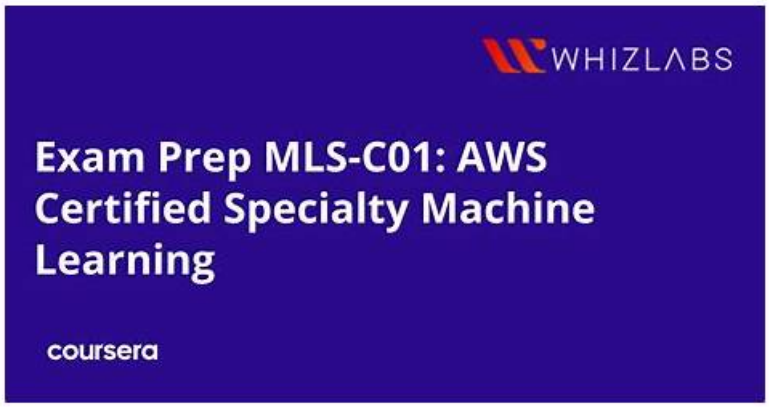
MLS-C01 Latest Exam Experience & MLS-C01 Latest Dumps Ppt



BONUS!!! Download part of TestKingFree MLS-C01 dumps for free: https://drive.google.com/open?id=1QEcueE57aqH2tl6UXJD21GHM1T-zqTL0

Web-based AWS Certified Machine Learning - Specialty (MLS-C01) practice test of TestKingFree is accessible from any place. You merely need an active internet connection to take this Amazon MLS-C01 practice exam. Browsers including MS Edge, Internet Explorer, Safari, Opera, Chrome, and Firefox support this AWS Certified Machine Learning - Specialty (MLS-C01) practice exam. Additionally, this AWS Certified Machine Learning - Specialty (MLS-C01) test is supported by operating systems including Android, Mac, iOS, Windows, and Linux.

To take the AWS-Certified-Machine-Learning-Specialty exam, candidates must have a basic understanding of AWS services and machine learning concepts. They should also have experience working with AWS services such as Amazon SageMaker, Amazon S3, and Amazon EC2. Candidates are also expected to have experience with at least one programming language, such as Python or R.

>> MLS-C01 Latest Exam Experience <<

MLS-C01 Latest Dumps Ppt | MLS-C01 Exam Overviews

The users of our MLS-C01 exam questions log on to their account on the platform, at the same time to choose what they want to attend the exam simulation questions, the MLS-C01 exam questions are automatically for the user presents the same as the actual test environment simulation MLS-C01 test system, the software built-in timer function can help users better control over time, so as to achieve the systematic, keep up, as well as to improve the user's speed to solve the problem from the side with our MLS-C01 test guide.

Who should take the Amazon MLS exam

Candidates who have the following should take this exam:

- Identify appropriate AWS services to implement ML solutions
- Design and implement scalable, cost-optimized, reliable, and secure ML solutions
- Select and justify the appropriate ML approach for a given business problem

If a candidate wants significant improvement in career growth needs enhanced knowledge, skills, and talents. TThe AWS Certified Solutions Architect - Associate examination is intended for individuals who perform a solutions architect role and have one or more years of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable distributed systems on AWS.

It asks for following experience:

- The ability to express the intuition behind basic ML algorithms
- The ability to follow model-training best practices
- Experience performing basic hyperparameter optimization
- Experience with ML and deep learning frameworks

- 1-2 years of experience developing, architecting, or running ML/deep learning workloads on the AWS Cloud

Amazon AWS Certified Machine Learning - Specialty Sample Questions (Q329-Q334):

NEW QUESTION # 329
A telecommunications company is developing a mobile app for its customers. The company is using an Amazon SageMaker hosted endpoint for machine learning model inferences.
Developers want to introduce a new version of the model for a limited number of users who subscribed to a preview feature of the app. After the new version of the model is tested as a preview, developers will evaluate its accuracy. If a new version of the model has better accuracy, developers need to be able to gradually release the new version for all users over a fixed period of time.
How can the company implement the testing model with the LEAST amount of operational overhead?

- A. Update the ProductionVariant data type with the new version of the model by using the CreateEndpointConfig operation with the InitialVariantWeight parameter set to 0. Specify the TargetVariant parameter for InvokeEndpoint calls for users who subscribed to the preview feature.
  When the new version of the model is ready for release, gradually increase InitialVariantWeight until all users have the updated version.
- B. Configure two SageMaker hosted endpoints that serve the different versions of the model. Create an Application Load Balancer (ALB) to route traffic to both endpoints based on the TargetVariant query string parameter. Reconfigure the app to send the TargetVariant query string parameter for users who subscribed to the preview feature. When the new version of the model is ready for release, change the ALB's routing algorithm to weighted until all users have the updated version.
- C. Configure two SageMaker hosted endpoints that serve the different versions of the model. Create an Amazon Route 53 record that is configured with a simple routing policy and that points to the current version of the model. Configure the mobile app to use the endpoint URL for users who subscribed to the preview feature and to use the Route 53 record for other users. When the new version of the model is ready for release, add a new model version endpoint to Route 53, and switch the policy to weighted until all users have the updated version.
- D. Update the DesiredWeightsAndCapacity data type with the new version of the model by using the UpdateEndpointWeightsAndCapacities operation with the DesiredWeight parameter set to 0. Specify the TargetVariant parameter for InvokeEndpoint calls for users who subscribed to the preview feature.
  When the new version of the model is ready for release, gradually increase DesiredWeight until all users have the updated version.

Answer: D

Explanation:
The best solution for implementing the testing model with the least amount of operational overhead is to use the following steps:
* Update the DesiredWeightsAndCapacity data type with the new version of the model by using the UpdateEndpointWeightsAndCapacities operation with the DesiredWeight parameter set to 0. This operation allows the developers to update the variant weights and capacities of an existing SageMaker endpoint without deleting and recreating the endpoint. Setting the DesiredWeight parameter to 0 means that the new version of the model will not receive any traffic initially1
* Specify the TargetVariant parameter for InvokeEndpoint calls for users who subscribed to the preview feature. This parameter allows the developers to override the variant weights and direct a request to a specific variant. This way, the developers can test the new version of the model for a limited number of users who opted in for the preview feature2
* When the new version of the model is ready for release, gradually increase DesiredWeight until all users have the updated version. This operation allows the developers to perform a gradual rollout of the new version of the model and monitor its performance and accuracy. The developers can adjust the variant weights and capacities as needed until the new version of the model serves all the traffic1 The other options are incorrect because they either require more operational overhead or do not support the desired use cases. For example:
* Option A uses the CreateEndpointConfig operation with the InitialVariantWeight parameter set to 0.
This operation creates a new endpoint configuration, which requires deleting and recreating the endpoint to apply the changes. This adds extra overhead and downtime for the endpoint. It also does not support the gradual rollout of the new version of the model3
* Option B uses two SageMaker hosted endpoints that serve the different versions of the model and an Application Load Balancer (ALB) to route traffic to both endpoints based on the TargetVariant query string parameter. This option requires creating and managing additional resources and services, such as the second endpoint and the ALB. It also requires changing the app code to send the query string parameter for the preview feature4
* Option D uses the access key and secret key of the IAM user with appropriate KMS and ECR permissions. This is not a secure way to pass credentials to the Processing job. It also requires the ML specialist to manage the IAM user and the keys.
1: UpdateEndpointWeightsAndCapacities - Amazon SageMaker
2: InvokeEndpoint - Amazon SageMaker
3: CreateEndpointConfig - Amazon SageMaker

4: Application Load Balancer - Elastic Load Balancing

NEW QUESTION # 330
A data scientist is training a text classification model by using the Amazon SageMaker built-in BlazingText algorithm. There are 5 classes in the dataset, with 300 samples for category A, 292 samples for category B,
240 samples for category C, 258 samples for category D, and 310 samples for category E.
The data scientist shuffles the data and splits off 10% for testing. After training the model, the data scientist generates confusion matrices for the training and test sets.

Training data confusion matrix

		Predicted class					
		A	B	C	D	E	Total
True class	A	270	0	0	0	0	270
	B	1	260	0	0	2	263
	C	0	0	111	100	5	216
	D	4	3	132	92	1	232
	E	0	0	2	3	274	279
	Total	275	263	245	195	282	1260

Test data confusion matrix

		Predicted class					
		A	B	C	D	E	Total
True class	A	9	1	0	0	0	10
	B	2	25	0	2	0	29
	C	10	2	11	10	1	34
	D	1	0	12	14	0	27
	E	9	1	4	1	25	40
	Total	31	29	27	27	26	140

What could the data scientist conclude form these results?

- A. The model is overfitting for classes B and E.
- B. The dataset is too small for holdout cross-validation.
- C. The data distribution is skewed.

- D. Classes C and D are too similar.

Answer: A

Explanation:
Explanation
A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It displays the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) produced by the model on the test data1. For multi-class classification, the matrix shape will be equal to the number of classes i.e for n classes it will be nXn1. The diagonal values represent the number of correct predictions for each class, and the off-diagonal values represent the number of incorrect predictions for each class1.
The BlazingText algorithm is a proprietary machine learning algorithm for forecasting time series using causal convolutional neural networks (CNNs). BlazingText works best with large datasets containing hundreds of time series. It accepts item metadata, and is the only Forecast algorithm that accepts related time series data without future values2.
From the confusion matrices for the training and test sets, we can observe the following:
The model has a high accuracy on the training set, as most of the diagonal values are high and the off-diagonal values are low. This means that the model is able to learn the patterns and features of the training data well.
However, the model has a lower accuracy on the test set, as some of the diagonal values are lower and some of the off-diagonal values are higher. This means that the model is not able to generalize well to the unseen data and makes more errors.
The model has a particularly high error rate for classes B and E on the test set, as the values of M_22 and M_55 are much lower than the values of M_12, M_21, M_15, M_25, M_51, and M_52. This means that the model is confusing classes B and E with other classes more often than it should.
The model has a relatively low error rate for classes A, C, and D on the test set, as the values of M_11, M_33, and M_44 are high and the values of M_13, M_14, M_23, M_24, M_31, M_32, M_34, M_41, M_42, and M_43 are low. This means that the model is able to distinguish classes A, C, and D from other classes well.
These results indicate that the model is overfitting for classes B and E, meaning that it is memorizing the specific features of these classes in the training data, but failing to capture the general features that are applicable to the test data. Overfitting is a common problem in machine learning, where the model performs well on the training data, but poorly on the test data3. Some possible causes of overfitting are:
The model is too complex or has too many parameters for the given data. This makes the model flexible enough to fit the noise and outliers in the training data, but reduces its ability to generalize to new data.
The data is too small or not representative of the population. This makes the model learn from a limited or biased sample of data, but fails to capture the variability and diversity of the population.
The data is imbalanced or skewed. This makes the model learn from a disproportionate or uneven distribution of data, but fails to account for the minority or rare classes.
Some possible solutions to prevent or reduce overfitting are:
Simplify the model or use regularization techniques. This reduces the complexity or the number of parameters of the model, and prevents it from fitting the noise and outliers in the data. Regularization techniques, such as L1 or L2 regularization, add a penalty term to the loss function of the model, which shrinks the weights of the model and reduces overfitting3.
Increase the size or diversity of the data. This provides more information and examples for the model to learn from, and increases its ability to generalize to new data. Data augmentation techniques, such as rotation, flipping, cropping, or noise addition, can generate new data from the existing data by applying some transformations3.
Balance or resample the data. This adjusts the distribution or the frequency of the data, and ensures that the model learns from all classes equally. Resampling techniques, such as oversampling or undersampling, can create a balanced dataset by increasing or decreasing the number of samples for each class3.
References:
Confusion Matrix in Machine Learning - GeeksforGeeks
BlazingText algorithm - Amazon SageMaker
Overfitting and Underfitting in Machine Learning - GeeksforGeeks

NEW QUESTION # 331
An online store is predicting future book sales by using a linear regression model that is based on past sales data. The data includes duration, a numerical feature that represents the number of days that a book has been listed in the online store. A data scientist performs an exploratory data analysis and discovers that the relationship between book sales and duration is skewed and non-linear. Which data transformation step should the data scientist take to improve the predictions of the model?

- A. Quantile binning
- B. One-hot encoding
- C. Cartesian product transformation
- D. Normalization

Answer: A

Explanation:
Explanation
Quantile binning is a data transformation technique that can be used to handle skewed and non-linear numerical features. It divides the range of a feature into equal-sized bins based on the percentiles of the data.
Each bin is assigned a numerical value that represents the midpoint of the bin. This way, the feature values are transformed into a more uniform distribution that can improve the performance of linear models. Quantile binning can also reduce the impact of outliers and noise in the data.
One-hot encoding, Cartesian product transformation, and normalization are not suitable for this scenario.
One-hot encoding is used to transform categorical features into binary features. Cartesian product transformation is used to create new features by combining existing features. Normalization is used to scale numerical features to a standard range, but it does not change the shape of the distribution. References:
Data Transformations for Machine Learning
Quantile Binning Transformation

NEW QUESTION # 332
A developer at a retail company is creating a daily demand forecasting model. The company stores the historical hourly demand data in an Amazon S3 bucket. However, the historical data does not include demand data for some hours.
The developer wants to verify that an autoregressive integrated moving average (ARIMA) approach will be a suitable model for the use case.
How should the developer verify the suitability of an ARIMA approach?

- A. Use Amazon SageMaker Data Wrangler. Import the data from Amazon S3. Resample data by using the aggregate daily total. Perform a Seasonal Trend decomposition.
- B. Use Amazon SageMaker Autopilot. Create a new experiment that specifies the S3 data location. Choose ARIMA as the machine learning (ML) problem. Check the model performance.
- C. Use Amazon SageMaker Data Wrangler. Import the data from Amazon S3. Impute hourly missing data. Perform a Seasonal Trend decomposition.
- D. Use Amazon SageMaker Autopilot. Create a new experiment that specifies the S3 data location. Impute missing hourly values. Choose ARIMA as the machine learning (ML) problem. Check the model performance.

Answer: C

Explanation:
The best solution to verify the suitability of an ARIMA approach is to use Amazon SageMaker Data Wrangler. Data Wrangler is a feature of SageMaker Studio that provides an end-to-end solution for importing, preparing, transforming, featurizing, and analyzing data. Data Wrangler includes built-in analyses that help generate visualizations and data insights in a few clicks. One of the built-in analyses is the Seasonal-Trend decomposition, which can be used to decompose a time series into its trend, seasonal, and residual components. This analysis can help the developer understand the patterns and characteristics of the time series, such as stationarity, seasonality, and autocorrelation, which are important for choosing an appropriate ARIMA model. Data Wrangler also provides built-in transformations that can help the developer handle missing data, such as imputing with mean, median, mode, or constant values, or dropping rows with missing values. Imputing missing data can help avoid gaps and irregularities in the time series, which can affect the ARIMA model performance. Data Wrangler also allows the developer to export the prepared data and the analysis code to various destinations, such as SageMaker Processing, SageMaker Pipelines, or SageMaker Feature Store, for further processing and modeling.
The other options are not suitable for verifying the suitability of an ARIMA approach. Amazon SageMaker Autopilot is a feature-set that automates key tasks of an automatic machine learning (AutoML) process. It explores the data, selects the algorithms relevant to the problem type, and prepares the data to facilitate model training and tuning. However, Autopilot does not support ARIMA as a machine learning problem type, and it does not provide any visualization or analysis of the time series data. Resampling data by using the aggregate daily total can reduce the granularity and resolution of the time series, which can affect the ARIMA model accuracy and applicability.

NEW QUESTION # 333
A large mobile network operating company is building a machine learning model to predict customers who are likely to unsubscribe from the service. The company plans to offer an incentive for these customers as the cost of churn is far greater than the cost of the incentive.
The model produces the following confusion matrix after evaluating on a test dataset of 100 customers:

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

- A. The precision of the model is 86%, which is less than the accuracy of the model.
- B. The precision of the model is 86%, which is greater than the accuracy of the model.
- C. The model is 86% accurate and the cost incurred by the company as a result of false negatives is less than the false positives.
- D. The model is 86% accurate and the cost incurred by the company as a result of false positives is less than the false negatives.

Answer: D

Explanation:
Based on the model evaluation results, this is a viable model for production because the model is 86% accurate and the cost incurred by the company as a result of false positives is less than the false negatives. The accuracy of the model is the proportion of correct predictions out of the total predictions, which can be calculated by adding the true positives and true negatives and dividing by the total number of observations. In this case, the accuracy of the model is (10 + 76) / 100 = 0.86, which means that the model correctly predicted
86% of the customers' churn status. The cost incurred by the company as a result of false positives and false negatives is the loss or damage that the company suffers when the model makes incorrect predictions. A false positive is when the model predicts that a customer will churn, but the customer actually does not churn. A false negative is when the model predicts that a customer will not churn, but the customer actually churns. In this case, the cost of a false positive is the incentive that the company offers to the customer who is predicted to churn, which is a relatively low cost. The cost of a false negative is the revenue that the company loses when the customer churns, which is a relatively high cost. Therefore, the cost of a false positive is less than the cost of a false negative, and the company would prefer to have more false positives than false negatives.
The model has 10 false positives and 4 false negatives, which means that the company's cost is lower than if the model had more false negatives and fewer false positives.

NEW QUESTION # 334
.....

What's more, part of that TestKingFree MLS-C01 dumps now are free: https://drive.google.com/open?id=1QEcueE57aqH2tl6UXJD21GHM1T-zqTL0