Free Questions for MLS-C01 by ebraindumps

Shared by Contreras on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

An ecommerce company has developed a XGBoost model in Amazon SageMaker to predict whether a customer will return a purchased item. The dataset is imbalanced. Only 5% of customers return items

A data scientist must find the hyperparameters to capture as many instances of returned items as possible. The company has a small budget for compute.

How should the data scientist meet these requirements MOST cost-effectively?

Options:

A- Tune all possible hyperparameters by using automatic model tuning (AMT). Optimize on {'HyperParameterTuningJobObjective': {'MetricName': 'validation:accuracy', 'Type': 'Maximize'}}

B- Tune the csv_weight hyperparameter and the scale_pos_weight hyperparameter by using automatic model tuning (AMT). Optimize on {'HyperParameterTuningJobObjective': {'MetricName': 'validation:f1', 'Type': 'Maximize'}}.

C- Tune all possible hyperparameters by using automatic model tuning (AMT). Optimize on {'HyperParameterTuningJobObjective': {'MetricName': 'validation:f1', 'Type': 'Maximize'}}.

D- Tune the csv_weight hyperparameter and the scale_pos_weight hyperparameter by using automatic model tuning (AMT). Optimize on {'HyperParameterTuningJobObjective': {'MetricName': 'validation:f1', 'Type': 'Minimize'}).

Answer:

B

Explanation:

The best solution to meet the requirements is to tune the csv_weight hyperparameter and the scale_pos_weight hyperparameter by using automatic model tuning (AMT). Optimize on {"HyperParameterTuningJobObjective": {"MetricName": "validation:f1", "Type": "Maximize"}}.

The csv_weight hyperparameter is used to specify the instance weights for the training data in CSV format. This can help handle imbalanced data by assigning higher weights to the minority class examples and lower weights to the majority class examples. The scale_pos_weight hyperparameter is used to control the balance of positive and negative weights. It is the ratio of the number of negative class examples to the number of positive class examples. Setting a higher value for this hyperparameter can increase the importance of the positive class and improve the recall. Both of these hyperparameters can help the XGBoost model capture as many instances of returned items as possible.

Automatic model tuning (AMT) is a feature of Amazon SageMaker that automates the process of finding the best hyperparameter values for a machine learning model. AMT uses Bayesian optimization to search the hyperparameter space and evaluate the model performance based on a predefined objective metric. The objective metric is the metric that AMT tries to optimize by adjusting the hyperparameter values. For imbalanced classification problems, accuracy is not a good objective metric, as it can be misleading and biased towards the majority class. A better objective metric is the F1 score, which is the harmonic mean of precision and recall. The F1 score can reflect the balance between precision and recall and is more suitable for imbalanced data. The F1 score ranges from 0 to 1, where 1 is the best possible value. Therefore, the type of the objective should be "Maximize" to achieve the highest F1 score.

By tuning the csv_weight and scale_pos_weight hyperparameters and optimizing on the F1 score, the data scientist can meet the requirements most cost-effectively. This solution requires tuning only two hyperparameters, which can reduce the computation time and cost compared to tuning all possible hyperparameters. This solution also uses the appropriate objective metric for imbalanced classification, which can improve the model performance and capture more instances of returned items.

References:

* XGBoost Hyperparameters

* Automatic Model Tuning

* How to Configure XGBoost for Imbalanced Classification

* Imbalanced Data

Question 2

Question Type: **MultipleChoice**

A law firm handles thousands of contracts every day. Every contract must be signed. Currently, a lawyer manually checks all contracts for signatures.

The law firm is developing a machine learning (ML) solution to automate signature detection for each contract. The ML solution must also provide a confidence score for each contract page.

Which Amazon Textract API action can the law firm use to generate a confidence score for each page of each contract?

Options:

A- Use the AnalyzeDocument API action. Set the FeatureTypes parameter to SIGNATURES. Return the confidence scores for each page.

B- Use the Prediction API call on the documents. Return the signatures and confidence scores for each page.

C- Use the StartDocumentAnalysis API action to detect the signatures. Return the confidence scores for each page.

D- Use the GetDocumentAnalysis API action to detect the signatures. Return the confidence scores for each page

Answer:

A

Explanation:

The AnalyzeDocument API action is the best option to generate a confidence score for each page of each contract. This API action analyzes an input document for relationships between detected items. The input document can be an image file in JPEG or PNG format, or a PDF file. The output is a JSON structure that contains the extracted data from the document. The FeatureTypes parameter specifies the types of analysis to perform on the document. The available feature types are TABLES, FORMS, and SIGNATURES. By setting the FeatureTypes parameter to SIGNATURES, the API action will detect and extract information about signatures from the document. The output will include a list of SignatureDetection objects, each containing information about a detected signature, such as its location and

confidence score. The confidence score is a value between 0 and 100 that indicates the probability that the detected signature is correct. The output will also include a list of Block objects, each representing a document page. Each Block object will have a Page attribute that contains the page number and a Confidence attribute that contains the confidence score for the page. The confidence score for the page is the average of the confidence scores of the blocks that are detected on the page. The law firm can use the AnalyzeDocument API action to generate a confidence score for each page of each contract by using the SIGNATURES feature type and returning the confidence scores from the SignatureDetection and Block objects.

The other options are not suitable for generating a confidence score for each page of each contract. The Prediction API call is not an Amazon Textract API action, but a generic term for making inference requests to a machine learning model. The StartDocumentAnalysis API action is used to start an asynchronous job to analyze a document. The output is a job identifier (JobId) that is used to get the results of the analysis with the GetDocumentAnalysis API action. The GetDocumentAnalysis API action is used to get the results of a document analysis started by the StartDocumentAnalysis API action. The output is a JSON structure that contains the extracted data from the document. However, both the StartDocumentAnalysis and the GetDocumentAnalysis API actions do not support the SIGNATURES feature type, and therefore cannot detect signatures or provide confidence scores for them.

References:

* AnalyzeDocument

* SignatureDetection

* Block

* Amazon Textract launches the ability to detect signatures on any document

Question 3

Question Type: MultipleChoice

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?

Options:

A- Use Amazon SageMaker Data Wrangler. Import the data from Amazon S3. Impute hourly missing data. 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. Resample data by using the aggregate daily total. 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:

A

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.

References:

* Analyze and Visualize

* Transform and Export

* Amazon SageMaker Autopilot

* ARIMA Model -- Complete Guide to Time Series Forecasting in Python

Question 4

Question Type: **MultipleChoice**

A global bank requires a solution to predict whether customers will leave the bank and choose another bank. The bank is using a dataset to train a model to predict customer loss. The training dataset has 1,000 rows. The training dataset includes 100 instances of customers who left the bank.

A machine learning (ML) specialist is using Amazon SageMaker Data Wrangler to train a churn prediction model by using a SageMaker training job. After training, the ML specialist notices that the model returns only false results. The ML specialist must correct the model so that it returns more accurate predictions.

Which solution will meet these requirements?

Options:

A- Apply anomaly detection to remove outliers from the training dataset before training.

B- Apply Synthetic Minority Oversampling Technique (SMOTE) to the training dataset before training.

C- Apply normalization to the features of the training dataset before training.

D- Apply undersampling to the training dataset before training.

Answer:

B

Explanation:

The best solution to meet the requirements is to apply Synthetic Minority Oversampling Technique (SMOTE) to the training dataset before training. SMOTE is a technique that generates synthetic samples for the minority class by interpolating between existing samples. This can help balance the class distribution and provide more information to the model. SMOTE can improve the performance of the model on the minority class, which is the class of interest in churn prediction. SMOTE can be applied using the SageMaker Data Wrangler, which provides a built-in analysis for oversampling the minority class1.

The other options are not effective solutions for the problem. Applying anomaly detection to remove outliers from the training dataset before training may not improve the model's accuracy, as outliers may not be the main cause of the false results. Moreover, removing outliers may reduce the diversity of the data and make the model less robust. Applying normalization to the features of the training dataset before training may improve the model's convergence and stability, but it does not address the class imbalance issue. Normalization can also be applied using the SageMaker Data Wrangler, which provides a built-in transformation for scaling the

features2. Applying undersampling to the training dataset before training may reduce the class imbalance, but it also discards potentially useful information from the majority class. Undersampling can also result in underfitting and high bias for the model.

References:

* Analyze and Visualize

* Transform and Export

* SMOTE for Imbalanced Classification with Python

* Churn prediction using Amazon SageMaker built-in tabular algorithms LightGBM, CatBoost, TabTransformer, and AutoGluon-Tabular

Question 5

Question Type: **MultipleChoice**

A company is setting up a mechanism for data scientists and engineers from different departments to access an Amazon SageMaker Studio domain. Each department has a unique SageMaker Studio domain.

The company wants to build a central proxy application that data scientists and engineers can log in to by using their corporate credentials. The proxy application will authenticate users by using the company's existing Identity provider (IdP). The application will then route users to the appropriate SageMaker Studio domain.

The company plans to maintain a table in Amazon DynamoDB that contains SageMaker domains for each department.

How should the company meet these requirements?

Options:

A- Use the SageMaker CreatePresignedDomainUrl API to generate a presigned URL for each domain according to the DynamoDB table. Pass the presigned URL to the proxy application.

B- Use the SageMaker CreateHuman TaskUi API to generate a UI URL. Pass the URL to the proxy application.

C- Use the Amazon SageMaker ListHumanTaskUis API to list all UI URLs. Pass the appropriate URL to the DynamoDB table so that the proxy application can use the URL.

D- Use the SageMaker CreatePresignedNotebookInstanceUrl API to generate a presigned URL. Pass the presigned URL to the proxy application.

Answer:

A

Explanation:

The SageMaker CreatePresignedDomainUrl API is the best option to meet the requirements of the company. This API creates a URL for a specified UserProfile in a Domain. When accessed in a web browser, the user will be automatically signed in to the domain, and granted access to all of the Apps and files associated with the Domain's Amazon Elastic File System (EFS) volume. This API can only

be called when the authentication mode equals IAM, which means the company can use its existing IdP to authenticate users. The company can use the DynamoDB table to store the domain IDs and user profile names for each department, and use the proxy application to query the table and generate the presigned URL for the appropriate domain according to the user's credentials. The presigned URL is valid only for a specified duration, which can be set by the SessionExpirationDurationInSeconds parameter. This can help enhance the security and prevent unauthorized access to the domains.

The other options are not suitable for the company's requirements. The SageMaker CreateHumanTaskUi API is used to define the settings for the human review workflow user interface, which is not related to accessing the SageMaker Studio domains. The SageMaker ListHumanTaskUis API is used to return information about the human task user interfaces in the account, which is also not relevant to the company's use case. The SageMaker CreatePresignedNotebookInstanceUrl API is used to create a URL to connect to the Jupyter server from a notebook instance, which is different from accessing the SageMaker Studio domain.

References:

* CreatePresignedDomainUrl

* CreatePresignedNotebookInstanceUrl

* CreateHumanTaskUi

* ListHumanTaskUis

Question 6

Question Type: **MultipleChoice**

A data scientist is trying to improve the accuracy of a neural network classification model. The data scientist wants to run a large hyperparameter tuning job in Amazon SageMaker.

However, previous smaller tuning jobs on the same model often ran for several weeks. The ML specialist wants to reduce the computation time required to run the tuning job.

Which actions will MOST reduce the computation time for the hyperparameter tuning job? (Select TWO.)

Options:

A- Use the Hyperband tuning strategy.

B- Increase the number of hyperparameters.

C- Set a lower value for the MaxNumberOfTrainingJobs parameter.

D- Use the grid search tuning strategy

E- Set a lower value for the MaxParallelTrainingJobs parameter.

Answer:

A, C

Explanation:

The Hyperband tuning strategy is a multi-fidelity based tuning strategy that dynamically reallocates resources to the most promising hyperparameter configurations. Hyperband uses both intermediate and final results of training jobs to stop under-performing jobs and reallocate epochs to well-utilized hyperparameter configurations. Hyperband can provide up to three times faster hyperparameter tuning compared to other strategies1. Setting a lower value for the MaxNumberOfTrainingJobs parameter can also reduce the computation time for the hyperparameter tuning job by limiting the number of training jobs that the tuning job can launch. This can help avoid unnecessary or redundant training jobs that do not improve the objective metric.

The other options are not effective ways to reduce the computation time for the hyperparameter tuning job. Increasing the number of hyperparameters will increase the complexity and dimensionality of the search space, which can result in longer computation time and lower performance. Using the grid search tuning strategy will also increase the computation time, as grid search methodically searches through every combination of hyperparameter values, which can be very expensive and inefficient for large search spaces. Setting a lower value for the MaxParallelTrainingJobs parameter will reduce the number of training jobs that can run in parallel, which can slow down the tuning process and increase the waiting time.

References:

* How Hyperparameter Tuning Works

* Best Practices for Hyperparameter Tuning

* HyperparameterTuner

* Amazon SageMaker Automatic Model Tuning now provides up to three times faster hyperparameter tuning with Hyperband

Question 7

Question Type: MultipleChoice

A machine learning engineer is building a bird classification model. The engineer randomly separates a dataset into a training dataset and a validation dataset. During the training phase, the model achieves very high accuracy. However, the model did not generalize well during validation of the validation dataset. The engineer realizes that the original dataset was imbalanced.

What should the engineer do to improve the validation accuracy of the model?

Options:

A- Perform stratified sampling on the original dataset.

B- Acquire additional data about the majority classes in the original dataset.

C- Use a smaller, randomly sampled version of the training dataset.

D- Perform systematic sampling on the original dataset.

Answer:

A

Explanation:

Stratified sampling is a technique that preserves the class distribution of the original dataset when creating a smaller or split dataset. This means that the proportion of examples from each class in the original dataset is maintained in the smaller or split dataset. Stratified sampling can help improve the validation accuracy of the model by ensuring that the validation dataset is representative of the original dataset and not biased towards any class. This can reduce the variance and overfitting of the model and increase its generalization ability. Stratified sampling can be applied to both oversampling and undersampling methods, depending on whether the goal is to increase or decrease the size of the dataset.

The other options are not effective ways to improve the validation accuracy of the model. Acquiring additional data about the majority classes in the original dataset will only increase the imbalance and make the model more biased towards the majority classes. Using a smaller, randomly sampled version of the training dataset will not guarantee that the class distribution is preserved and may result in losing important information from the minority classes. Performing systematic sampling on the original dataset will also not ensure that the class distribution is preserved and may introduce sampling bias if the original dataset is ordered or grouped by class.

References:

* Stratified Sampling for Imbalanced Datasets

* Imbalanced Data

* Tour of Data Sampling Methods for Imbalanced Classification

Question 8

Question Type: **MultipleChoice**

A machine learning (ML) developer for an online retailer recently uploaded a sales dataset into Amazon SageMaker Studio. The ML developer wants to obtain importance scores for each feature of the dataset. The ML developer will use the importance scores to feature engineer the dataset.

Which solution will meet this requirement with the LEAST development effort?

Options:

A- Use SageMaker Data Wrangler to perform a Gini importance score analysis.

B- Use a SageMaker notebook instance to perform principal component analysis (PCA).

C- Use a SageMaker notebook instance to perform a singular value decomposition analysis.

D- Use the multicollinearity feature to perform a lasso feature selection to perform an importance scores analysis.

Answer:

A

Explanation:

SageMaker 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 Quick Model visualization, which can be used to quickly evaluate the data and produce

importance scores for each feature. A feature importance score indicates how useful a feature is at predicting a target label. The feature importance score is between [0, 1] and a higher number indicates that the feature is more important to the whole dataset. The Quick Model visualization uses a random forest model to calculate the feature importance for each feature using the Gini importance method. This method measures the total reduction in node impurity (a measure of how well a node separates the classes) that is attributed to splitting on a particular feature. The ML developer can use the Quick Model visualization to obtain the importance scores for each feature of the dataset and use them to feature engineer the dataset. This solution requires the least development effort compared to the other options.

References:

* Analyze and Visualize

* Detect multicollinearity, target leakage, and feature correlation with Amazon SageMaker Data Wrangler