Free Questions for MLS-C01

Shared by Valentine on 20-10-2022

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A media company wants to create a solution that identifies celebrities in pictures that users upload. The company also wants to identify the IP address and the timestamp details from the users so the company can prevent users from uploading pictures from unauthorized locations.

Which solution will meet these requirements with LEAST development effort?

Options:

A- Use AWS Panorama to identify celebrities in the pictures. Use AWS CloudTrail to capture IP address and timestamp details.

B- Use AWS Panorama to identify celebrities in the pictures. Make calls to the AWS Panorama Device SDK to capture IP address and timestamp details.

C- Use Amazon Rekognition to identify celebrities in the pictures. Use AWS CloudTrail to capture IP address and timestamp details.

D- Use Amazon Rekognition to identify celebrities in the pictures. Use the text detection feature to capture IP address and timestamp details.

Answer:

C

Explanation:

The solution C will meet the requirements with the least development effort because it uses Amazon Rekognition and AWS CloudTrail, which are fully managed services that can provide the desired functionality. The solution C involves the following steps:

Use Amazon Rekognition to identify celebrities in the pictures. Amazon Rekognition is a service that can analyze images and videos and extract insights such as faces, objects, scenes, emotions, and more. Amazon Rekognition also provides a feature called Celebrity Recognition, which can recognize thousands of celebrities across a number of categories, such as politics, sports, entertainment, and medi

a.Amazon Rekognition can return the name, face, and confidence score of the recognized celebrities, as well as additional information such as URLs and biographies1.

Use AWS CloudTrail to capture IP address and timestamp details. AWS CloudTrail is a service that can record the API calls and events made by or on behalf of AWS accounts. AWS CloudTrail can provide information such as the source IP address, the user identity, the request parameters, and the response elements of the API calls.AWS CloudTrail can also deliver the event records to an

Amazon S3 bucket or an Amazon CloudWatch Logs group for further analysis and auditing2.

The other options are not suitable because:

Option A: Using AWS Panorama to identify celebrities in the pictures and using AWS CloudTrail to capture IP address and timestamp details will not meet the requirements effectively. AWS Panorama is a service that can extend computer vision to the edge, where it can run inference on video streams from cameras and other devices. AWS Panorama is not designed for identifying celebrities in pictures, and it may not provide accurate or relevant results.Moreover, AWS Panorama requires the use of an AWS Panorama Appliance or a compatible device, which may incur additional costs and complexity3.

Option B: Using AWS Panorama to identify celebrities in the pictures and making calls to the AWS Panorama Device SDK to capture IP address and timestamp details will not meet the requirements effectively, for the same reasons as option A.Additionally, making calls to the AWS Panorama Device SDK will require more development effort than using AWS CloudTrail, as it will involve writing custom code and handling errors and exceptions4.

Option D: Using Amazon Rekognition to identify celebrities in the pictures and using the text detection feature to capture IP address and timestamp details will not meet the requirements effectively. The text detection feature of Amazon Rekognition is used to detect and recognize text in images and videos, such as street names, captions, product names, and license plates. It is not suitable for capturing IP address and timestamp details, as these are not part of the pictures that users upload.Moreover, the text detection feature may not be accurate or reliable, as it depends on the quality and clarity of the text in the images and videos5.

References:

1: Amazon Rekognition Celebrity Recognition

2: AWS CloudTrail Overview

3: AWS Panorama Overview

4: AWS Panorama Device SDK

5: Amazon Rekognition Text Detection

Question 2

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 3

Question Type: MultipleChoice

A data scientist at a financial services company used Amazon SageMaker to train and deploy a model that predicts loan defaults. The model analyzes new loan applications and predicts the risk of loan default. To train the model, the data scientist manually extracted loan data from a database. The data scientist performed the model training and deployment steps in a Jupyter notebook that is hosted on SageMaker Studio notebooks. The model's prediction accuracy is decreasing over time. Which combination of slept in the MOST operationally efficient way for the data scientist to maintain the model's accuracy? (Select TWO.)

Options:

A- Use SageMaker Pipelines to create an automated workflow that extracts fresh data, trains the model, and deploys a new version of the model.
B- Configure SageMaker Model Monitor with an accuracy threshold to check for model drift. Initiate an Amazon CloudWatch alarm when the threshold is exceeded. Connect the workflow in SageMaker Pipelines with the CloudWatch alarm to automatically initiate retraining.
C- Store the model predictions in Amazon S3 Create a daily SageMaker Processing job that reads the predictions from Amazon S3, checks for changes in model prediction accuracy, and sends an email notification if a significant change is detected.
D- Rerun the steps in the Jupyter notebook that is hosted on SageMaker Studio notebooks to retrain the model and redeploy a new version of the model.
E- Export the training and deployment code from the SageMaker Studio notebooks into a Python script. Package the script into an Amazon Elastic Container Service (Amazon ECS) task that an AWS Lambda function can initiate.

Answer:

A, B

Explanation:

Option A is correct because SageMaker Pipelines is a service that enables you to create and manage automated workflows for your machine learning projects.You can use SageMaker Pipelines to orchestrate the steps of data extraction, model training, and model deployment in a repeatable and scalable way1.

Option B is correct because SageMaker Model Monitor is a service that monitors the quality of your models in production and alerts you when there are deviations in the model quality. You can use SageMaker Model Monitor to set an accuracy threshold for your model and configure a CloudWatch alarm that triggers when the threshold is exceeded.You can then connect the alarm to the workflow in SageMaker Pipelines to automatically initiate retraining and deployment of a new version of the model2.

Option C is incorrect because it is not the most operationally efficient way to maintain the model's accuracy. Creating a daily SageMaker Processing job that reads the predictions from Amazon S3 and checks for changes in model prediction accuracy is a manual and time-consuming process. It also requires you to write custom code to perform the data analysis and send the email notification. Moreover, it does not automatically retrain and deploy the model when the accuracy drops.

Option D is incorrect because it is not the most operationally efficient way to maintain the model's accuracy. Rerunning the steps in the Jupyter notebook that is hosted on SageMaker Studio notebooks to retrain the model and redeploy a new version of the model is a manual and error-prone process. It also requires you to monitor the model's performance and initiate the retraining and deployment steps yourself. Moreover, it does not leverage the benefits of SageMaker Pipelines and SageMaker Model Monitor to automate and streamline the workflow.

Option E is incorrect because it is not the most operationally efficient way to maintain the model's accuracy. Exporting the training and deployment code from the SageMaker Studio notebooks into a Python script and packaging the script into an Amazon ECS task that an AWS Lambda function can initiate is a complex and cumbersome process. It also requires you to manage the infrastructure and resources for the Amazon ECS task and the AWS Lambda function. Moreover, it does not leverage the benefits of SageMaker Pipelines and SageMaker Model Monitor to automate and streamline the workflow.

References:

1:SageMaker Pipelines - Amazon SageMaker

2:Monitor data and model quality - Amazon SageMaker

Question 4

Question Type: MultipleChoice

A data science team is planning to build a natural language processing (NLP) application. The application's text preprocessing stage will include part-of-speech tagging and key phase extraction. The preprocessed text will be input to a custom classification algorithm that the data science team has already written and trained using Apache MXNet.

Which solution can the team build MOST quickly to meet these requirements?

Options:

A- Use Amazon Comprehend for the part-of-speech tagging, key phase extraction, and classification tasks.
B- Use an NLP library in Amazon SageMaker for the part-of-speech tagging. Use Amazon Comprehend for the key phase extraction. Use AWS Deep Learning Containers with Amazon SageMaker to build the custom classifier.
C- Use Amazon Comprehend for the part-of-speech tagging and key phase extraction tasks. Use Amazon SageMaker built-in Latent Dirichlet Allocation (LDA) algorithm to build the custom classifier.
D- Use Amazon Comprehend for the part-of-speech tagging and key phase extraction tasks. Use AWS Deep Learning Containers with Amazon SageMaker to build the custom classifier.

Answer:

D

Explanation:

Amazon Comprehend is a natural language processing (NLP) service that can perform part-of-speech tagging and key phrase extraction tasks. AWS Deep Learning Containers are Docker images that are pre-installed with popular deep learning frameworks such as Apache MXNet. Amazon SageMaker is a fully managed service that can help build, train, and deploy machine learning models. Using Amazon Comprehend for the text preprocessing tasks and AWS Deep Learning Containers with Amazon SageMaker to build the custom classifier is the solution that can be built most quickly to meet the requirements.

References:

Amazon Comprehend

AWS Deep Learning Containers

Amazon SageMaker

Question 5

Question Type: MultipleChoice

A company is building a demand forecasting model based on machine learning (ML). In the development stage, an ML specialist uses an Amazon SageMaker notebook to perform feature engineering during work hours that consumes low amounts of CPU and memory resources. A data engineer uses the same notebook to perform data preprocessing once a day on average that requires very high memory and completes in only 2 hours. The data preprocessing is not configured to use GPU. All the processes are running well on an ml.m5.4xlarge notebook instance.

The company receives an AWS Budgets alert that the billing for this month exceeds the allocated budget.

Which solution will result in the MOST cost savings?

Options:

A- Change the notebook instance type to a memory optimized instance with the same vCPU number as the ml.m5.4xlarge instance has. Stop the notebook when it is not in use. Run both data preprocessing and feature engineering development on that instance.

B- Keep the notebook instance type and size the same. Stop the notebook when it is not in use. Run data preprocessing on a P3 instance type with the same memory as the ml.m5.4xlarge instance by using Amazon SageMaker Processing.

C- Change the notebook instance type to a smaller general-purpose instance. Stop the notebook when it is not in use. Run data preprocessing on an ml. r5 instance with the same memory size as the ml.m5.4xlarge instance by using Amazon SageMaker Processing.

D- Change the notebook instance type to a smaller general-purpose instance. Stop the notebook when it is not in use. Run data preprocessing on an R5 instance with the same memory size as the ml.m5.4xlarge instance by using the Reserved Instance option.

Answer:

C

Explanation:

The best solution to reduce the cost of the notebook instance and the data preprocessing job is to change the notebook instance type to a smaller general-purpose instance, stop the notebook when it is not in use, and run data preprocessing on an ml.r5 instance with the same memory size as the ml.m5.4xlarge instance by using Amazon SageMaker Processing. This solution will result in the most cost savings because:

Changing the notebook instance type to a smaller general-purpose instance will reduce the hourly cost of running the notebook, since the feature engineering development does not require high CPU and memory resources.For example, an ml.t3.medium instance costs $0.0464 per hour, while an ml.m5.4xlarge instance costs $0.888 per hour1.

Stopping the notebook when it is not in use will also reduce the cost, since the notebook will only incur charges when it is running.For example, if the notebook is used for 8 hours per day, 5 days per week, then stopping it when it is not in use will save about 76% of the monthly cost compared to leaving it running all the time2.

Running data preprocessing on an ml.r5 instance with the same memory size as the ml.m5.4xlarge instance by using Amazon SageMaker Processing will reduce the cost of the data preprocessing job, since the ml.r5 instance is optimized for memory-intensive workloads and has a lower cost per GB of memory than the ml.m5 instance.For example, an ml.r5.4xlarge instance has 128 GB of memory and costs $1.008 per hour, while an ml.m5.4xlarge instance has 64 GB of memory and costs $0.888 per hour1. Therefore, the ml.r5.4xlarge instance can process the same amount of data in half the time and at a lower cost than the ml.m5.4xlarge instance. Moreover, using Amazon SageMaker Processing will allow the data preprocessing job to run on a separate, fully managed infrastructure that can be scaled up or down as needed, without affecting the notebook instance.

The other options are not as effective as option C for the following reasons:

Option A is not optimal because changing the notebook instance type to a memory optimized instance with the same vCPU number as the ml.m5.4xlarge instance has will not reduce the cost of the notebook, since the memory optimized instances have a higher cost per vCPU than the general-purpose instances.For example, an ml.r5.4xlarge instance has 16 vCPUs and costs $1.008 per hour, while an ml.m5.4xlarge instance has 16 vCPUs and costs $0.888 per hour1. Moreover, running both data preprocessing and feature engineering development on the same instance will not take advantage of the scalability and flexibility of Amazon SageMaker Processing.

Option B is not suitable because running data preprocessing on a P3 instance type with the same memory as the ml.m5.4xlarge instance by using Amazon SageMaker Processing will not reduce the cost of the data preprocessing job, since the P3 instance type is optimized for GPU-based workloads and has a higher cost per GB of memory than the ml.m5 or ml.r5 instance types.For example, an ml.p3.2xlarge instance has 61 GB of memory and costs $3.06 per hour, while an ml.m5.4xlarge instance has 64 GB of memory and costs $0.888 per hour1. Moreover, the data preprocessing job does not require GPU, so using a P3 instance type will be wasteful and inefficient.

Option D is not feasible because running data preprocessing on an R5 instance with the same memory size as the ml.m5.4xlarge instance by using the Reserved Instance option will not reduce the cost of the data preprocessing job, since the Reserved Instance option requires a commitment to a consistent amount of usage for a period of 1 or 3 years3. However, the data preprocessing job only runs once a day on average and completes in only 2 hours, so it does not have a consistent or predictable usage pattern. Therefore, using the Reserved Instance option will not provide any cost savings and may incur additional charges for unused capacity.

References:

Amazon SageMaker Pricing

Manage Notebook Instances - Amazon SageMaker

Amazon EC2 Pricing - Reserved Instances

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 car company is developing a machine learning solution to detect whether a car is present in an image. The image dataset consists of one million images. Each image in the dataset is 200 pixels in height by 200 pixels in width. Each image is labeled as either having a car or not having a car.

Which architecture is MOST likely to produce a model that detects whether a car is present in an image with the highest accuracy?

Options:

A- Use a deep convolutional neural network (CNN) classifier with the images as input. Include a linear output layer that outputs the probability that an image contains a car.

B- Use a deep convolutional neural network (CNN) classifier with the images as input. Include a softmax output layer that outputs the probability that an image contains a car.
C- Use a deep multilayer perceptron (MLP) classifier with the images as input. Include a linear output layer that outputs the probability that an image contains a car.
D- Use a deep multilayer perceptron (MLP) classifier with the images as input. Include a softmax output layer that outputs the probability that an image contains a car.

Answer:

A

Explanation:

A deep convolutional neural network (CNN) classifier is a suitable architecture for image classification tasks, as it can learn features from the images and reduce the dimensionality of the input. A linear output layer that outputs the probability that an image contains a car is appropriate for a binary classification problem, as it can produce a single scalar value between 0 and 1. A softmax output layer is more suitable for a multi-class classification problem, as it can produce a vector of probabilities that sum up to 1. A deep multilayer perceptron (MLP) classifier is not as effective as a CNN for image classification, as it does not exploit the spatial structure of the images and requires a large number of parameters to process the high-dimensional input.References:

AWS Certified Machine Learning - Specialty Exam Guide

AWS Training - Machine Learning on AWS

AWS Whitepaper - An Overview of Machine Learning on AWS