Free Questions for Professional-Machine-Learning-Engineer by certscare

Shared by Brennan on 24-05-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: **MultipleChoice**

You have recently developed a custom model for image classification by using a neural network. You need to automatically identify the values for learning rate, number of layers, and kernel size. To do this, you plan to run multiple jobs in parallel to identify the parameters that optimize performance. You want to minimize custom code development and infrastructure management. What should you do?

Options:

A- Create a Vertex AI pipeline that runs different model training jobs in parallel.

B- Train an AutoML image classification model.

C- Create a custom training job that uses the Vertex AI Vizier SDK for parameter optimization.

D- Create a Vertex AI hyperparameter tuning job.

Answer:

D

Question 2

Question Type: **MultipleChoice**

You work at an organization that maintains a cloud-based communication platform that integrates conventional chat, voice, and video conferencing into one platform. The audio recordings are stored in Cloud Storage. All recordings have an 8 kHz sample rate and are more than one minute long. You need to implement a new feature in the platform that will automatically transcribe voice call recordings into a text for future applications, such as call summarization and sentiment analysis. How should you implement the voice call transcription feature following Google-recommended best practices?

Options:

A- Use the original audio sampling rate, and transcribe the audio by using the Speech-to-Text API with synchronous recognition.

B- Use the original audio sampling rate, and transcribe the audio by using the Speech-to-Text API with asynchronous recognition.

C- Upsample the audio recordings to 16 kHz. and transcribe the audio by using the Speech-to-Text API with synchronous recognition.

D- Upsample the audio recordings to 16 kHz. and transcribe the audio by using the Speech-to-Text API with asynchronous recognition.

Answer:

D

Question 3

Question Type: **MultipleChoice**

You have created a Vertex AI pipeline that automates custom model training You want to add a pipeline component that enables your team to most easily collaborate when running different executions and comparing metrics both visually and programmatically. What should you do?

Options:

A- Add a component to the Vertex AI pipeline that logs metrics to a BigQuery table Query the table to compare different executions of the pipeline Connect BigQuery to Looker Studio to visualize metrics.

B- Add a component to the Vertex AI pipeline that logs metrics to a BigQuery table Load the table into a pandas DataFrame to compare different executions of the pipeline Use Matplotlib to visualize metrics.

C- Add a component to the Vertex AI pipeline that logs metrics to Vertex ML Metadata Use Vertex AI Experiments to compare different executions of the pipeline Use Vertex AI TensorBoard to visualize metrics.

D- Add a component to the Vertex AI pipeline that logs metrics to Vertex ML Metadata Load the Vertex ML Metadata into a pandas DataFrame to compare different executions of the pipeline. Use Matplotlib to visualize metrics.

Answer:

C

Explanation:

Vertex AI Experiments is a managed service that allows you to track, compare, and manage experiments with Vertex AI. You can use Vertex AI Experiments to record the parameters, metrics, and artifacts of each pipeline run, and compare them in a graphical interface. Vertex AI TensorBoard is a tool that lets you visualize the metrics of your models, such as accuracy, loss, and learning curves. By logging metrics to Vertex ML Metadata and using Vertex AI Experiments and TensorBoard, you can easily collaborate with your team and find the best model configuration for your problem.Reference:Vertex AI Pipelines: Metrics visualization and run comparison using the KFP SDK,Track, compare, manage experiments with Vertex AI Experiments,Vertex AI Pipelines

Question 4

Question Type: MultipleChoice

You work at a mobile gaming startup that creates online multiplayer games Recently, your company observed an increase in players cheating in the games, leading to a loss of revenue and a poor user experience. You built a binary classification model to determine whether a player cheated after a completed game session, and then send a message to other downstream systems to ban the player that cheated Your model has performed well during testing, and you now need to deploy the model to production You want your serving solution to provide immediate classifications after a completed game session to avoid further loss of revenue. What should you do?

Options:

A- Import the model into Vertex AI Model Registry. Use the Vertex Batch Prediction service to run batch inference jobs.

B- Save the model files in a Cloud Storage Bucket Create a Cloud Function to read the model files and make online inference requests on the Cloud Function.

C- Save the model files in a VM Load the model files each time there is a prediction request and run an inference job on the VM.

D- Import the model into Vertex Al Model Registry Create a Vertex Al endpoint that hosts the model and make online inference requests.

Answer:

D

Explanation:

Online inference is a process where you send a single or a small number of prediction requests to a model and get immediate responses1. Online inference is suitable for scenarios where you need timely predictions, such as detecting cheating in online games.Online inference requires that the model is deployed to an endpoint, which is a resource that provides a service URL for prediction requests2.

Vertex AI Model Registry is a central repository where you can manage the lifecycle of your ML models3.You can import models from various sources, such as custom models or AutoML models, and assign them to different versions and aliases3.You can also deploy models to endpoints, which are resources that provide a service URL for online prediction2.

By importing the model into Vertex AI Model Registry, you can leverage the Vertex AI features to monitor and update the model3. You can use Vertex AI Experiments to track and compare the metrics of different model versions, such as accuracy, precision, recall, and AUC. You can also use Vertex AI Explainable AI to generate feature attributions that show how much each input feature contributed to the model's prediction.

By creating a Vertex AI endpoint that hosts the model, you can use the Vertex AI Prediction service to serve online inference requests2.Vertex AI Prediction provides various benefits, such as scalability, reliability, security, and logging2.You can use the Vertex AI API or the Google Cloud console to send online inference requests to the endpoint and get immediate classifications4.

Therefore, the best option for your scenario is to import the model into Vertex AI Model Registry, create a Vertex AI endpoint that hosts the model, and make online inference requests.

The other options are not suitable for your scenario, because they either do not provide immediate classifications, such as using batch prediction or loading the model files each time, or they do not use Vertex AI Prediction, which would require more development and maintenance effort, such as creating a Cloud Function or a VM.

Online versus batch prediction | Vertex AI | Google Cloud

Deploy a model to an endpoint | Vertex AI | Google Cloud

Introduction to Vertex AI Model Registry | Google Cloud

Get online predictions | Vertex AI | Google Cloud

Question 5

Question Type: **MultipleChoice**

You work for a pet food company that manages an online forum Customers upload photos of their pets on the forum to share with others About 20 photos are uploaded daily You want to automatically and in near real time detect whether each uploaded photo has an animal You want to prioritize time and minimize cost of your application development and deployment What should you do?

Options:

A- Send user-submitted images to the Cloud Vision API Use object localization to identify all objects in the image and compare the results against a list of animals.

B- Download an object detection model from TensorFlow Hub. Deploy the model to a Vertex AI endpoint. Send new user-submitted images to the model endpoint to classify whether each photo has an animal.

C- Manually label previously submitted images with bounding boxes around any animals Build an AutoML object detection model by using Vertex AI Deploy the model to a Vertex AI endpoint Send new user-submitted images to your model endpoint to detect whether each photo has an animal.

D- Manually label previously submitted images as having animals or not Create an image dataset on Vertex AI Train a classification model by using Vertex AutoML to distinguish the two classes Deploy the model to a Vertex AI endpoint Send new user-submitted images to your model endpoint to classify whether each photo has an animal.

Answer:

A

Explanation:

Cloud Vision API is a service that allows you to analyze images using pre-trained machine learning models1.You can use Cloud Vision API to perform various tasks, such as face detection, text extraction, logo recognition, and object localization1.Object localization is a feature that allows you to detect multiple objects in an image and draw bounding boxes around them2.You can also get the labels and confidence scores for each detected object2.

By sending user-submitted images to the Cloud Vision API, you can use object localization to identify all objects in the image and compare the results against a list of animals.You can use theOBJECT_LOCALIZATIONfeature type in theAnnotateImageRequestto request object localization3. You can then use thelocalizedObjectAnnotationsfield in theAnnotateImageResponseto get the list of detected objects, their labels, and their confidence scores. You can compare the labels with a predefined list of animals, such as dogs, cats, birds, etc., and determine whether the image has an animal or not.

This option is the best for your scenario, because it allows you to automatically and in near real time detect whether each uploaded photo has an animal, without requiring any manual labeling, model training, or model deployment. You can also prioritize time and minimize cost of your application development and deployment, as you can use the Cloud Vision API as a ready-to-use service, without needing any machine learning expertise or infrastructure.

The other options are not suitable for your scenario, because they either require manual labeling, model training, or model deployment, which would increase the time and cost of your application development and deployment, or they use object detection models, which are more complex and computationally expensive than object localization models, and are not necessary for your simple task of detecting whether an image has an animal or not.

Cloud Vision API | Google Cloud

Object localization | Cloud Vision API | Google Cloud

AnnotateImageRequest | Cloud Vision API | Google Cloud

[AnnotateImageResponse | Cloud Vision API | Google Cloud]

Question 6

Question Type: **MultipleChoice**

You have deployed a scikit-learn model to a Vertex AI endpoint using a custom model server. You enabled auto scaling; however, the deployed model fails to scale beyond one replica, which led to dropped requests. You notice that CPU utilization remains low even during periods of high load. What should you do?

Options:

A- Attach a GPU to the prediction nodes.

B- Increase the number of workers in your model server.

C- Schedule scaling of the nodes to match expected demand.

D- Increase the minReplicaCount in your DeployedModel configuration.

Answer:

B

Explanation:

Auto scaling is a feature that allows you to automatically adjust the number of prediction nodes based on the traffic and load of your deployed model1.However, auto scaling depends on the CPU utilization of your prediction nodes, which is the percentage of CPU resources used by your model server1.If your CPU utilization is low, even during periods of high load, it means that your model server is not fully utilizing the available CPU resources, and thus auto scaling will not trigger more replicas2.

One possible reason for low CPU utilization is that your model server is using a single worker process to handle prediction requests3.A worker process is a subprocess that runs your model code and handles prediction requests3.If you have only one worker process, it can only handle one request at a time, which can lead to dropped requests when the traffic is high3.To increase the CPU utilization and the throughput of your model server, you can increase the number of worker processes, which will allow your model server to handle multiple requests in parallel3.

To increase the number of workers in your model server, you need to modify your custom model server code and use the--workersflag to specify the number of worker processes you want to use3. For example, if you are using a Gunicorn server, you can use the following command to start your model server with four worker processes:

```
gunicorn --bind :$PORT --workers 4 --threads 1 --timeout 60 main:app
```

By increasing the number of workers in your model server, you can increase the CPU utilization of your prediction nodes, and thus enable auto scaling to scale beyond one replica.

The other options are not suitable for your scenario, because they either do not address the root cause of low CPU utilization, such as attaching a GPU or scheduling scaling, or they do not enable auto scaling, such as increasing the minReplicaCount, which is a fixed number of nodes that will always run regardless of the traffic1.

Scaling prediction nodes | Vertex AI | Google Cloud

Troubleshooting | Vertex AI | Google Cloud

Using a custom prediction routine with online prediction | Vertex AI | Google Cloud

Question 7

Question Type: **MultipleChoice**

You are developing a model to identify traffic signs in images extracted from videos taken from the dashboard of a vehicle. You have a dataset of 100 000 images that were cropped to show one out of ten different traffic signs. The images have been labeled accordingly for model training and are stored in a Cloud Storage bucket You need to be able to tune the model during each training run. How should you train the model?

Options:

A- Train a model for object detection by using Vertex AI AutoML.

B- Train a model for image classification by using Vertex AI AutoML.

C- Develop the model training code for object detection and tram a model by using Vertex AI custom training.

D- Develop the model training code for image classification and train a model by using Vertex AI custom training.

Answer:

D

Explanation:

Image classification is a task where the model assigns a label to an image based on its content, such as ''stop sign'' or 'speed limit'1.Object detection is a task where the model locates and identifies multiple objects in an image, and draws bounding boxes around them2. Since your dataset consists of images that were cropped to show one out of ten different traffic signs, you are dealing with an image classification problem, not an object detection problem. Therefore, you need to train a model for image classification, not object detection.

Vertex AI AutoML is a service that allows you to train and deploy high-quality ML models with minimal effort and machine learning expertise3.You can use Vertex AI AutoML to train a model for image classification by uploading your images and labels to a Vertex AI dataset, and then launching an AutoML training job4.However, Vertex AI AutoML does not allow you to tune the model during each training run, as it automatically selects the best model architecture and hyperparameters for your data4.

Vertex AI custom training is a service that allows you to train and deploy your own custom ML models using your own code and frameworks5. You can use Vertex AI custom training to train a model for image classification by writing your own model training code, such as using TensorFlow or PyTorch, and then creating and running a custom training job. Vertex AI custom training allows you to tune the model during each training run, as you can specify the model architecture and hyperparameters in your code, and use Vertex AI Hyperparameter Tuning to optimize them .

Therefore, the best option for your scenario is to develop the model training code for image classification and train a model by using Vertex AI custom training.

Image classification | TensorFlow Core

Object detection | TensorFlow Core

Introduction to Vertex AI AutoML | Google Cloud

AutoML Vision | Google Cloud

Introduction to Vertex AI custom training | Google Cloud

[Custom training with TensorFlow | Vertex AI | Google Cloud]

[Hyperparameter tuning overview | Vertex AI | Google Cloud]

Question 8

Question Type: MultipleChoice

You work for a large retailer and you need to build a model to predict customer churn. The company has a dataset of historical customer data, including customer demographics, purchase history, and website activity. You need to create the model in BigQuery ML and thoroughly evaluate its performance. What should you do?

Options:

A- Create a linear regression model in BigQuery ML and register the model in Vertex Al Model Registry Evaluate the model performance in Vertex Al.

B- Create a logistic regression model in BigQuery ML and register the model in Vertex Al Model Registry. Evaluate the model performance in Vertex Al.

C- Create a linear regression model in BigQuery ML Use the ml. evaluate function to evaluate the model performance.

D- Create a logistic regression model in BigQuery ML Use the ml.confusion_matrix function to evaluate the model performance.

Answer:

B

Explanation:

Customer churn is a binary classification problem, where the target variable is whether a customer has churned or not. Therefore, a logistic regression model is more suitable than a linear regression model, which is used for regression problems.A logistic regression model can output the probability of a customer churning, which can be used to rank the customers by their churn risk and take appropriate actions1.

BigQuery ML is a service that allows you to create and execute machine learning models in BigQuery using standard SQL queries2.You can use BigQuery ML to create a logistic regression model for customer churn prediction by using theCREATE MODELstatement and specifying theLOGISTIC_REGmodel type3.You can use the historical customer data as the input table for the model, and specify the features and the label columns3.

Vertex AI Model Registry is a central repository where you can manage the lifecycle of your ML models4.You can import models from various sources, such as BigQuery ML, AutoML, or custom models, and assign them to different versions and aliases4. You can also deploy models to endpoints, which are resources that provide a service URL for online prediction.

By registering the BigQuery ML model in Vertex AI Model Registry, you can leverage the Vertex AI features to evaluate and monitor the model performance4. You can use Vertex AI Experiments to track and compare the metrics of different model versions, such as accuracy, precision, recall, and AUC. You can also use Vertex AI Explainable AI to generate feature attributions that show how much each input feature contributed to the model's prediction.

The other options are not suitable for your scenario, because they either use the wrong model type, such as linear regression, or they do not use Vertex AI to evaluate the model performance, which would limit the insights and actions you can take based on the model results.

Logistic Regression for Machine Learning

Introduction to BigQuery ML | Google Cloud

Creating a logistic regression model | BigQuery ML | Google Cloud

Introduction to Vertex AI Model Registry | Google Cloud

[Deploy a model to an endpoint | Vertex AI | Google Cloud]

[Vertex AI Experiments | Google Cloud]

Question 9

Question Type: MultipleChoice

You work for a retail company that is using a regression model built with BigQuery ML to predict product sales. This model is being used to serve online predictions Recently you developed a new version of the model that uses a different architecture (custom model) Initial analysis revealed that both models are performing as expected You want to deploy the new version of the model to production and monitor the performance over the next two months You need to minimize the impact to the existing and future model users How should you deploy the model?

Options:

A- Import the new model to the same Vertex AI Model Registry as a different version of the existing model. Deploy the new model to the same Vertex AI endpoint as the existing model, and use traffic splitting to route 95% of production traffic to the BigQuery ML model and 5% of production traffic to the new model.

B- Import the new model to the same Vertex AI Model Registry as the existing model Deploy the models to one Vertex AI endpoint Route 95% of production traffic to the BigQuery ML model and 5% of production traffic to the new model

C- Import the new model to the same Vertex AI Model Registry as the existing model Deploy each model to a separate Vertex AI endpoint.

D- Deploy the new model to a separate Vertex AI endpoint Create a Cloud Run service that routes the prediction requests to the corresponding endpoints based on the input feature values.

Answer:

A

Explanation:

Vertex AI Model Registry is a central repository where you can manage the lifecycle of your ML models1.You can import models from various sources, such as BigQuery ML, AutoML, or custom models, and assign them to different versions and aliases1.You can also deploy models to endpoints, which are resources that provide a service URL for online prediction2.

By importing the new model to the same Vertex AI Model Registry as a different version of the existing model, you can keep track of the model versions and compare their performance metrics1.You can also use aliases to label the model versions according to their readiness for production, such asdefaultorstaging1.

By deploying the new model to the same Vertex AI endpoint as the existing model, you can use traffic splitting to gradually shift the production traffic from the old model to the new model2.Traffic splitting is a feature that allows you to specify the percentage of prediction requests that each deployed model in an endpoint should handle2.This way, you can minimize the impact to the existing and future model users, and monitor the performance of the new model over time2.

The other options are not suitable for your scenario, because they either require creating a separate endpoint or a Cloud Run service, which would increase the complexity and maintenance of your deployment, or they do not allow you to use traffic splitting, which would create a sudden change in your prediction results.Reference:

Introduction to Vertex AI Model Registry | Google Cloud

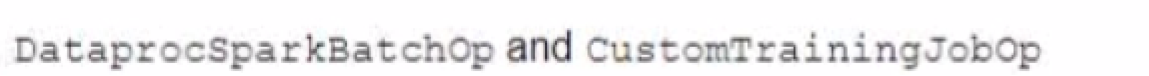
Deploy a model to an endpoint | Vertex AI | Google Cloud

Question 10

Question Type: **MultipleChoice**

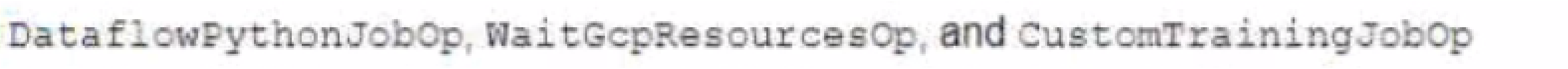
You are building a custom image classification model and plan to use Vertex AI Pipelines to implement the end-to-end training. Your dataset consists of images that need to be preprocessed before they can be used to train the model. The preprocessing steps include resizing the images, converting them to grayscale, and extracting features. You have already implemented some Python functions for the preprocessing tasks. Which components should you use in your pipeline'?

A.



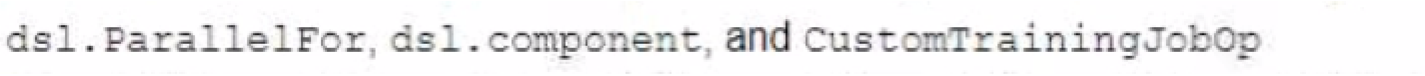
`DataprocSparkBatchOp` and `CustomTrainingJobOp`

B.



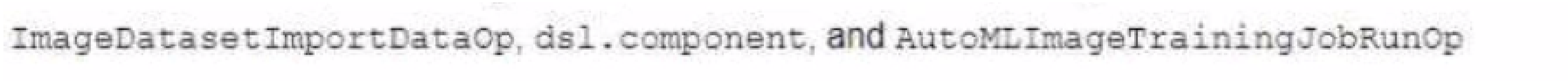
`DataflowPythonJobOp`, `WaitGcpResourcesOp`, and `CustomTrainingJobOp`

C.



`dsl.ParallelFor`, `dsl.component`, and `CustomTrainingJobOp`

D.



`ImageDatasetImportDataOp`, `dsl.component`, and `AutoMLImageTrainingJobRunOp`

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

To Get Premium Files for Professional-Machine-Learning-Engineer Visit

https://www.p2pexams.com/products/professional-machine-learning-engineer

For More Free Questions Visit

https://www.p2pexams.com/google/pdf/professional-machine-learning-engineer