



**Free Questions for Professional-Machine-Learning-Engineer
by certsinside**

Shared by Meyers on 15-04-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

You have a custom job that runs on Vertex AI on a weekly basis. The job is implemented using a proprietary ML workflow that produces the datasets, models, and custom artifacts, and sends them to a Cloud Storage bucket. Many different versions of the datasets and models were created. Due to compliance requirements, your company needs to track which model was used for making a particular prediction, and needs access to the artifacts for each model. How should you configure your workflows to meet these requirements?

Options:

- A-** Configure a TensorFlow Extended (TFX) ML Metadata database, and use the ML Metadata API.
- B-** Create a Vertex AI experiment, and enable autologging inside the custom job.
- C-** Use the Vertex AI Metadata API inside the custom Job to create context, execution, and artifacts for each model, and use events to link them together.
- D-** Register each model in Vertex AI Model Registry, and use model labels to store the related dataset and model information.

Answer:

D

Question 2

Question Type: MultipleChoice

You developed a Python module by using Keras to train a regression model. You developed two model architectures, linear regression and deep neural network (DNN), within the same module. You are using the `-- training_method` argument to select one of the two methods, and you are using the `Learning_rate` and `num_hidden_layers` arguments in the DNN. You plan to use Vertex AI's hyperparameter tuning service with a Budget to perform 100 trials. You want to identify the model architecture and hyperparameter values that minimize training loss and maximize model performance. What should you do?

Options:

- A-** Run one hyperparameter tuning job for 100 trials. Set `num_hidden_layers` as a conditional hyperparameter based on its parent hyperparameter `training_method`, and set learning rate as a non-conditional hyperparameter.
- B-** Run two separate hyperparameter tuning jobs: a linear regression job for 50 trials, and a DNN job for 50 trials. Compare their final performance on a common validation set, and select the set of hyperparameters with the least training loss.
- C-** Run one hyperparameter tuning job for 100 trials. Set `num_hidden_layers` and `learning_rate` as conditional hyperparameters based on their parent hyperparameter `training_method`.
- D-** Run one hyperparameter tuning job with `training_method` as the hyperparameter for 50 trials. Select the architecture with the lowest training loss, and further hyperparameter tune it and its corresponding hyperparameters for 50 trials.

Answer:

C

Question 3

Question Type: MultipleChoice

You have developed an AutoML tabular classification model that identifies high-value customers who interact with your organization's website.

You plan to deploy the model to a new Vertex AI endpoint that will integrate with your website application. You expect higher traffic to the website during

nights and weekends. You need to configure the model endpoint's deployment settings to minimize latency and cost. What should you do?

Options:

A- Configure the model deployment settings to use an n1-standard-32 machine type.

B- Configure the model deployment settings to use an n1-standard-4 machine type. Set the minReplicaCount value to 1 and the maxReplicaCount value to 8.

C- Configure the model deployment settings to use an n1-standard-4 machine type and a GPU accelerator. Set the minReplicaCount value to 1 and the maxReplicaCount value to 4.

D- Configure the model deployment settings to use an n1-standard-8 machine type and a GPU accelerator.

Answer:

B

Explanation:

Deploying a model to an endpoint in Vertex AI associates physical resources with the model so it can serve online predictions with low latency¹.By configuring the model deployment settings to use an n1-standard-4 machine type and setting the minReplicaCount value to 1 and the maxReplicaCount value to 8, you can ensure that the model scales according to the traffic, thereby minimizing latency and cost¹.The n1-standard-4 machine type provides a balance between computing power and cost, and the dynamic scaling allows the model to handle higher traffic during nights and weekends without incurring unnecessary costs during off-peak times

Question 4

Question Type: MultipleChoice

You are using Kubeflow Pipelines to develop an end-to-end PyTorch-based MLOps pipeline. The pipeline reads data from BigQuery,

processes the data, conducts feature engineering, model training, model evaluation, and deploys the model as a binary file to Cloud Storage. You are

writing code for several different versions of the feature engineering and model training steps, and running each new version in Vertex AI Pipelines.

Each pipeline run is taking over an hour to complete. You want to speed up the pipeline execution to reduce your development time, and you want to

avoid additional costs. What should you do?

Options:

- A- Delegate feature engineering to BigQuery and remove it from the pipeline.
- B- Add a GPU to the model training step.
- C- Enable caching in all the steps of the Kubeflow pipeline.
- D- Comment out the part of the pipeline that you are not currently updating.

Answer:

C

Explanation:

Kubeflow Pipelines allows for efficient use of compute resources through parallel task execution and caching, which eliminates redundant executions¹. By enabling caching in all the steps of the Kubeflow pipeline, you can avoid re-running the same steps when you execute the pipeline multiple times. This can significantly speed up the pipeline execution and reduce your development time without incurring additional costs

Question 5

Question Type: MultipleChoice

You trained a model, packaged it with a custom Docker container for serving, and deployed it to Vertex AI Model Registry. When you submit a batch prediction job, it fails with this error "Error model server never became ready Please validate that your model file or container configuration are valid. There are no additional errors in the logs What should you do?

Options:

- A-** Add a logging configuration to your application to emit logs to Cloud Logging.
- B-** Change the HTTP port in your model's configuration to the default value of 8080
- C-** Change the health Route value in your models configuration to /healthcheck.
- D-** Pull the Docker image locally and use the docker run command to launch it locally. Use the docker logs command to explore the error logs.

Answer:

B

Explanation:

When you deploy a custom container to Vertex AI Model Registry, you need to follow some requirements for the container configuration. One of these requirements is to use the HTTP port 8080 for serving predictions. If you use a different port, the model server might not be able to communicate with Vertex AI and cause the error "Error model server never became ready". To fix this error, you need to change the HTTP port in your model's configuration to the default value of 8080 and redeploy the container. Reference:

[Custom container requirements documentation](#)

[Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate](#)

Question 6

Question Type: MultipleChoice

You work on a team that builds state-of-the-art deep learning models by using the TensorFlow framework. Your team runs multiple ML experiments each week which makes it difficult to track the experiment runs. You want a simple approach to effectively track, visualize

and debug ML experiment runs on Google Cloud while minimizing any overhead code. How should you proceed?

Options:

- A-** Set up Vertex AI Experiments to track metrics and parameters Configure Vertex AI TensorBoard for visualization.
- B-** Set up a Cloud Function to write and save metrics files to a Cloud Storage Bucket Configure a Google Cloud VM to host TensorBoard locally for visualization.
- C-** Set up a Vertex AI Workbench notebook instance Use the instance to save metrics data in a Cloud Storage bucket and to host TensorBoard locally for visualization.
- D-** Set up a Cloud Function to write and save metrics files to a BigQuery table. Configure a Google Cloud VM to host TensorBoard locally for visualization.

Answer:

A

Explanation:

Vertex AI Experiments is a service that allows you to track, compare, and optimize your ML experiments on Google Cloud. You can use Vertex AI Experiments to log metrics and parameters from your TensorFlow models, and then visualize them in Vertex AI TensorBoard. Vertex AI TensorBoard is a managed service that provides a web interface for viewing and debugging your ML experiments. You can use Vertex AI TensorBoard to compare different runs, inspect model graphs, analyze scalars, histograms, images, and more. By using

Vertex AI Experiments and Vertex AI TensorBoard, you can simplify your ML experiment tracking and visualization workflow, and avoid the overhead of setting up and maintaining your own Cloud Functions, Cloud Storage buckets, or VMs. Reference:

[Vertex AI Experiments documentation]

[Vertex AI TensorBoard documentation]

[Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate](#)

Question 7

Question Type: MultipleChoice

Your team is training a large number of ML models that use different algorithms, parameters and datasets. Some models are trained in Vertex AI Pipelines, and some are trained on Vertex AI Workbench notebook instances. Your team wants to compare the performance of the models across both services. You want to minimize the effort required to store the parameters and metrics. What should you do?

Options:

- A-** Implement an additional step for all the models running in pipelines and notebooks to export parameters and metrics to BigQuery.
- B-** Create a Vertex AI experiment. Submit all the pipelines as experiment runs. For models trained on notebooks log parameters and

metrics by using the Vertex AI SDK.

C- Implement all models in Vertex AI Pipelines Create a Vertex AI experiment, and associate all pipeline runs with that experiment.

D- Store all model parameters and metrics as model metadata by using the Vertex AI Metadata API.

Answer:

B

Explanation:

Vertex AI Experiments is a 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 model training run, and compare them in a graphical interface. Vertex AI Experiments supports models trained in Vertex AI Pipelines, Vertex AI Custom Training, and Vertex AI Workbench notebooks. To use Vertex AI Experiments, you need to create an experiment and submit your pipeline runs or custom training jobs as experiment runs. For models trained on notebooks, you need to use the Vertex AI SDK to log the parameters and metrics to the experiment. This way, you can minimize the effort required to store and compare the model performance across different services. Reference: Track, compare, manage experiments with Vertex AI Experiments, Vertex AI Pipelines: Metrics visualization and run comparison using the KFP SDK, [Vertex AI SDK for Python]

**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>

