EXAMTOPICS

- Expert Verified, Online, **Free.**

≡ MENU 🔍

⬅ **Google Discussions**

Exam Professional Machine Learning Engineer All Questions
View all questions & answers for the Professional Machine Learning Engineer exam

Go to Exam

📄 **EXAM PROFESSIONAL MACHINE LEARNING ENGINEER TOPIC 1 QUESTION 175 DISCUSSI...**

Actual exam question from Google's Professional Machine Learning Engineer
Question #: 175
Topic #: 1
[All Professional Machine Learning Engineer Questions]

You have recently trained a scikit-learn model that you plan to deploy on Vertex AI. This model will support both online and batch prediction. You need to preprocess input data for model inference. You want to package the model for deployment while minimizing additional code. What should you do?

A. 1. Upload your model to the Vertex AI Model Registry by using a prebuilt scikit-ieam prediction container.
2. Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job that uses the instanceConfig.instanceType setting to transform your input data.

B. 1. Wrap your model in a custom prediction routine (CPR). and build a container image from the CPR local model.
2. Upload your scikit learn model container to Vertex AI Model Registry.
3. Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job

C. 1. Create a custom container for your scikit learn model.
2. Define a custom serving function for your model.
3. Upload your model and custom container to Vertex AI Model Registry.
4. Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job.

D. 1. Create a custom container for your scikit learn model.
2. Upload your model and custom container to Vertex AI Model Registry.
3. Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job that uses the instanceConfig.instanceType setting to transform your input data.

Show Suggested Answer

Comments

Type your comment…

Submit

👤 **shadz10** `Highly Voted 👍` 9 months, 3 weeks ago

`Selected Answer: B`

B - Creating a custom container without CPR adds additional complexity. i.e. write model server write dockerfile and also build and upload image. Where as using a CPR only requires writing a predictor and using vertex SDK to build image. https://cloud.google.com/vertex-ai/docs/predictions/custom-prediction-routines

👍 🔙 🚩 upvoted 5 times

👤 **bobjr** `Most Recent ⊘` 5 months ago

`Selected Answer: B`

https://cloud.google.com/vertex-ai/docs/predictions/custom-prediction-routines

👍 🔙 🚩 upvoted 1 times

👤 **gscharly** 6 months, 2 weeks ago

`Selected Answer: B`

agree with shadz10

👍 🔙 🚩 upvoted 1 times

👤 **guilhermebutzke** 9 months ago

`Selected Answer: C`

My choose: C

Option C ensures that the scikit-learn model is properly packaged, deployed, and integrated with Vertex AI services while minimizing the need for additional code beyond what is necessary for customizing the serving function.

Option B is not considered correct because it suggests wrapping the scikit-learn model in a custom prediction routine (CPR), which might not be the most suitable approach for deploying scikit-learn models on Vertex AI.

Options A and D using InstanceConfig, that is limited for preprocessing. Uploading the container without a serving function won't work.

👍 🔙 🚩 upvoted 1 times

👤 **pikachu007** 9 months, 4 weeks ago

`Selected Answer: D`

Considering the goal of minimizing additional code and complexity, option D - "Create a custom container for your scikit-learn model, upload your model and custom container to Vertex AI Model Registry, deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job that uses the instanceConfig.instanceType setting to transform your input data" seems to be a more straightforward and efficient approach. It involves customizing the container for the scikit-learn model, leveraging the Vertex AI Model Registry, and utilizing the specified instance type for batch prediction without introducing unnecessary complexity like custom prediction routines.

👍 🔙 🚩 upvoted 1 times

👤 **b1a8fae** 10 months ago

`Selected Answer: B`

I go with B:

"Custom prediction routines (CPR) lets you build custom containers with pre/post processing code easily, without dealing with the details of setting up an HTTP server or building a container from scratch." (https://cloud.google.com/vertex-ai/docs/predictions/custom-prediction-routines). This alone makes B preferable to C and D, provided lack of complex model architecture.

Regarding A, pre-built containers only allow serving predictions, but not preprocessing of data (https://cloud.google.com/vertex-ai/docs/predictions/pre-built-containers#use_a_prebuilt_container). B thus remains the most likely option.

👍 🔙 🚩 upvoted 4 times

Social Media

EXAMTOPICS

We are the biggest and most updated IT certification exam material website.

Using our own resources, we strive to strengthen the IT professionals community for free.