



- 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 226 DISCUSSI...

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 226

Topic #: 1

[\[All Professional Machine Learning Engineer Questions\]](#)

You recently trained a XGBoost model that you plan to deploy to production for online inference. Before sending a predict request to your model's binary, you need to perform a simple data preprocessing step. This step exposes a REST API that accepts requests in your internal VPC Service Controls and returns predictions. You want to configure this preprocessing step while minimizing cost and effort. What should you do?

- A. Store a pickled model in Cloud Storage. Build a Flask-based app, package the app in a custom container image, and deploy the model to Vertex AI Endpoints.
- B. Build a Flask-based app, package the app and a pickled model in a custom container image, and deploy the model to Vertex AI Endpoints.
- C. Build a custom predictor class based on XGBoost Predictor from the Vertex AI SDK, package it and a pickled model in a custom container image based on a Vertex built-in image, and deploy the model to Vertex AI Endpoints.
- D. Build a custom predictor class based on XGBoost Predictor from the Vertex AI SDK, and package the handler in a custom container image based on a Vertex built-in container image. Store a pickled model in Cloud Storage, and deploy the model to Vertex AI Endpoints.

Show Suggested Answer

by [pikachu007](#) at Jan. 13, 2024, 7:35 a.m.

Comments

Type your comment...

Submit


  **fitri001** 6 months, 3 weeks ago

Selected Answer: D

why not c?

While it utilizes the XGBoost Predictor, packaging the pickled model in the container increases image size and requires redeploying the container for model updates.

   upvoted 3 times

  **fitri001** 6 months, 3 weeks ago

why D?

Reduced Code Footprint: You only need to write the custom predictor logic, not a full Flask application. This minimizes development effort and container size.

Leverages Vertex AI Features: By using the XGBoost Predictor from the Vertex AI SDK, you benefit from pre-built functionality for handling XGBoost models.

Cost-Effective Deployment: Utilizing Vertex built-in container images reduces the need for custom image maintenance and potentially lowers container runtime costs.

Separate Model Storage: Storing the pickled model in Cloud Storage keeps the model separate from the prediction logic, allowing for easier model updates without redeploying the entire container.

   upvoted 2 times

  **guilhermebutzke** 8 months, 2 weeks ago



Selected Answer: D

My Answer: D

This option involves using the Vertex AI SDK to build a custom predictor class, which allows for easy integration with the XGBoost model. Packaging the handler in a custom container image based on a Vertex built-in container image ensures compatibility and smooth deployment. Storing the pickled model in Cloud Storage provides a scalable and reliable way to access the model. Deploying the model to Vertex AI Endpoints allows for easy management and scaling of inference requests, while minimizing cost and effort.

The main difference between C and D is where the model is saved. So, is a good practice to save models in GCS because Separation of Concerns, Flexibility, and Reduced Image Size

   upvoted 1 times

  **pikachu007** 9 months, 3 weeks ago

Selected Answer: D

Minimal Custom Code: Leverages the pre-built XGBoost Predictor class for core model prediction, reducing development effort and potential errors.

Optimized Container Image: Utilizes a Vertex built-in container image, pre-configured for efficient model serving and compatibility with Vertex AI Endpoints.

Separated Model Storage: Stores the model in Cloud Storage, reducing container image size and simplifying model updates independently of the container.

VPC Service Controls: Vertex AI Endpoints support VPC Service Controls, ensuring adherence to internal traffic restrictions.

   upvoted 1 times

Start Learning for free



Social Media

[Facebook](#) , [Twitter](#)

[YouTube](#) , [Reddit](#)

[Pinterest](#)



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.



© 2024 ExamTopics

ExamTopics doesn't offer Real Microsoft Exam Questions. ExamTopics doesn't offer Real Amazon Exam Questions. ExamTopics Materials do not contain actual questions and answers from Cisco's Certification Exams.

CFA Institute does not endorse, promote or warrant the accuracy or quality of ExamTopics. CFA® and Chartered Financial Analyst® are registered trademarks owned by CFA Institute.