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 210 DISCUSSI...**

Actual exam question from Google's Professional Machine Learning Engineer
Question #: 210
Topic #: 1

[All Professional Machine Learning Engineer Questions]

You have trained a model by using data that was preprocessed in a batch Dataflow pipeline. Your use case requires real-time inference. You want to ensure that the data preprocessing logic is applied consistently between training and serving. What should you do?

A. Perform data validation to ensure that the input data to the pipeline is the same format as the input data to the endpoint.

B. Refactor the transformation code in the batch data pipeline so that it can be used outside of the pipeline. Use the same code in the endpoint.

C. Refactor the transformation code in the batch data pipeline so that it can be used outside of the pipeline. Share this code with the end users of the endpoint.

D. Batch the real-time requests by using a time window and then use the Dataflow pipeline to preprocess the batched requests. Send the preprocessed requests to the endpoint.

Show Suggested Answer

by 👤 **pikachu007** at *Jan. 13, 2024, 5:36 a.m.*

Comments

Type your comment...

fitri001 6 months, 2 weeks ago

Selected Answer: B

Refactored Transformation Code: By refactoring the transformation code from the batch pipeline, you can create a reusable module that performs the same preprocessing steps.
Same Code in Endpoint: Utilize the refactored code within your real-time inference endpoint. This ensures the data is preprocessed identically to how it was preprocessed during training.

👍 ↩ 🚩 upvoted 1 times

fitri001 6 months, 2 weeks ago

A. Data Validation: While data validation is important, it doesn't guarantee consistent preprocessing logic. You need to ensure the same transformations are applied.
C. Share Code with End Users: Sharing code with end-users might not be ideal, especially if it requires specific libraries or configurations for execution outside of the pipeline.
D. Batching and Dataflow: Batching real-time requests for Dataflow processing might introduce latency and defeat the purpose of real-time inference.

👍 ↩ 🚩 upvoted 2 times

pinimichele01 6 months, 3 weeks ago

Selected Answer: B

agree with guilhermebutzke

👍 ↩ 🚩 upvoted 1 times

guilhermebutzke 8 months, 2 weeks ago

Selected Answer: B

My Answer B:

B. This option ensures that the preprocessing logic used during training, which has already been validated and tested, is applied consistently during real-time inference. By making the transformation code reusable outside of the batch pipeline and utilizing it in the endpoint, you ensure that the same preprocessing steps are applied to incoming data during inference, thus maintaining consistency between training and serving.

A: While data validation is essential, it only ensures the format. It doesn't guarantee consistent preprocessing logic between training and serving.

C: Sharing code with end-users might not be desirable for security or maintainability reasons.

D: Batching introduces latency and might not be suitable for real-time needs. Additionally, using the entire Dataflow pipeline might be inefficient for individual requests.

👍 ↩ 🚩 upvoted 3 times

shadz10 9 months, 3 weeks ago

Selected Answer: B

The transformation logic code in the serving_fn function defines the serving interface of your SavedModel for online prediction. If you implement the same transformations that were used for preparing training data in the transformation logic code of the serving_fn function, it ensures that the same transformations are applied to new prediction data points when they're served.
https://www.tensorflow.org/tfx/guide/tft_bestpractices

👍 ↩ 🚩 upvoted 3 times

pikachu007 9 months, 3 weeks ago

Selected Answer: B

A. Data validation: While essential, it doesn't guarantee consistency if the preprocessing logic itself differs between pipeline and endpoint.
C. Sharing code with end users: This shifts the preprocessing burden to end users, potentially leading to inconsistencies and errors, and isn't feasible for real-time inference.
D. Batching real-time requests: This introduces latency and might not align with real-time requirements, as users expect immediate responses.

👍 ↩ 🚩 upvoted 1 times

Start Learning for free

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.