EXAM PROFESSIONAL MACHINE LEARNING ENGINEER TOPIC 1 QUESTION 244 DISCUSSI...

Actual exam question from Google's Professional Machine Learning Engineer
Question #: 244
Topic #: 1
[All Professional Machine Learning Engineer Questions]

Your work for a textile manufacturing company. Your company has hundreds of machines, and each machine has many sensors. Your team used the sensory data to build hundreds of ML models that detect machine anomalies. Models are retrained daily, and you need to deploy these models in a cost-effective way. The models must operate 24/7 without downtime and make sub millisecond predictions. What should you do?

A. Deploy a Dataflow batch pipeline and a Vertex AI Prediction endpoint.

B. Deploy a Dataflow batch pipeline with the RunInference API, and use model refresh.

C. Deploy a Dataflow streaming pipeline and a Vertex AI Prediction endpoint with autoscaling.

D. Deploy a Dataflow streaming pipeline with the RunInference API, and use automatic model refresh.

Show Suggested Answer

by b1a8fae at *Jan. 20, 2024, 5:35 p.m.*

Comments

fitri001 Highly Voted 6 months, 1 week ago

Selected Answer: D

why D?
Real-time Predictions: Dataflow streaming pipelines continuously process sensor data, enabling real-time anomaly detection with sub-millisecond predictions. This is crucial for immediate response to potential machine issues.
RunInference API: This API allows invoking TensorFlow models directly within the Dataflow pipeline for on-the-fly inference. This eliminates the need for separate prediction endpoints and reduces latency.
Automatic Model Refresh: Since models are retrained daily, automatic refresh ensures the pipeline utilizes the latest version without downtime. This is essential for maintaining model accuracy and anomaly detection effectiveness.
Why not C?
Dataflow Streaming Pipeline with Vertex AI Prediction Endpoint with Autoscaling: While autoscaling can handle varying workloads, Vertex AI Prediction endpoints might incur higher costs for real-time, high-volume predictions compared to invoking models directly within the pipeline using RunInference.

upvoted 7 times

gscharly Most Recent 6 months ago

Selected Answer: D

agree with fitri001

upvoted 1 times

pinimichele01 6 months, 2 weeks ago

Selected Answer: D

With the automatic model refresh feature, when the underlying model changes, your pipeline updates to use the new model. Because the RunInference transform automatically updates the model handler, you don't need to redeploy the pipeline. With this feature, you can update your model in real time, even while the Apache Beam pipeline is running.

upvoted 1 times

pinimichele01 6 months, 1 week ago

and also ai endpoint not good for online inference

upvoted 1 times

guilhermebutzke 8 months ago

Selected Answer: C

My Answer: C

The phrase: “The models must operate 24/7 without downtime and make sub millisecond predictions” configures a case of online prediction (option B or C)

The phrase: “Models are retrained daily, and you need to deploy these models in a cost-effective way”, choose between “ Vertex AI Prediction endpoint with autoscaling” instead “RunInference API, and use automatic model refresh” looks better because always update with retrained models, and the scalability.

https://cloud.google.com/blog/products/ai-machine-learning/streaming-prediction-with-dataflow-and-vertex

upvoted 3 times

sonicclasps 8 months, 3 weeks ago

Selected Answer: C

low latency - > streaming
C & D could both work, but C is the GCP solution. So I chose C

upvoted 2 times

asmgi 3 months, 1 week ago

I don't think autoscaling is relevant to this task, since we have the same amount of sensors at any time.

upvoted 1 times

vaibavi 8 months, 2 weeks ago

i think autoscaling will lead to downtime atleast when the replicas are updating .

upvoted 2 times

pinimichele01 5 months, 3 weeks ago

i agree, D is better

upvoted 1 times

b1a8fae 9 months ago

Selected Answer: D

Needs to be active 24/7 -> streaming.
RunInference API seems like the way to go here, using automatic model refresh on a daily basis.
https://beam.apache.org/documentation/ml/about-ml/

upvoted 4 times