



- 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 208 DISCUSSI...

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 208

Topic #: 1

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

You recently developed a wide and deep model in TensorFlow. You generated training datasets using a SQL script that preprocessed raw data in BigQuery by performing instance-level transformations of the data. You need to create a training pipeline to retrain the model on a weekly basis. The trained model will be used to generate daily recommendations. You want to minimize model development and training time. How should you develop the training pipeline?

- A. Use the Kubeflow Pipelines SDK to implement the pipeline. Use the BigQueryJobOp component to run the preprocessing script and the CustomTrainingJobOp component to launch a Vertex AI training job.
- B. Use the Kubeflow Pipelines SDK to implement the pipeline. Use the DataflowPythonJobOp component to preprocess the data and the CustomTrainingJobOp component to launch a Vertex AI training job.
- C. Use the TensorFlow Extended SDK to implement the pipeline Use the ExampleGen component with the BigQuery executor to ingest the data the Transform component to preprocess the data, and the Trainer component to launch a Vertex AI training job.
- D. Use the TensorFlow Extended SDK to implement the pipeline Implement the preprocessing steps as part of the input\_fn of the model. Use the ExampleGen component with the BigQuery executor to ingest the data and the Trainer component to launch a Vertex AI training job.

Show Suggested Answer

by [pikachu007](#) at Jan. 13, 2024, 5:29 a.m.

## Comments

Type your comment...

[Submit](#)

  **tdum76000** 1 month, 3 weeks ago

**Selected Answer: D**



"If you use TensorFlow in an ML workflow that processes terabytes of structured data or text data, we recommend that you build your pipeline using TFX."

<https://cloud.google.com/vertex-ai/docs/pipelines/build-pipeline>

Google recommends TFX for large amount of structured data. Use input\_fn for the Tensorflow model as it will output a tf.data.Dataset object.

Note: As it is not mentionned that we are working with terabytes of data, Kubeflow is a viable option and i would choose answer A but i'll stick to google's recommendations



   upvoted 1 times

  **forport** 2 months, 1 week ago

**Selected Answer: C**

Option C is the most suitable because TFX provides a comprehensive MLOps framework, seamlessly integrating data ingestion, preprocessing, and model training, while also offering strong support for Vertex AI, making it the most efficient solution for the given use case.


   upvoted 1 times

  **AK2020** 2 months, 2 weeks ago

**Selected Answer: C**

C. Use the TensorFlow Extended SDK to implement the pipeline. Use the ExampleGen component with the BigQuery executor to ingest the data, the Transform component to preprocess the data, and the Trainer component to launch a Vertex AI training job.

   upvoted 1 times

  **TanTran04** 3 months, 2 weeks ago

**Selected Answer: A**

I go with A

Kubeflow Pipelines SDK: supports machine learning and includes components specifically for tasks like data preprocessing, model training, and validation.

BigQueryJobOp: enabling you to preprocess data using SQL scripts efficiently within BigQuery.

   upvoted 1 times

  **SausageMuffins** 5 months ago

**Selected Answer: C**

Example Gen directly ingest data from BigQuery and the transform component makes it more efficient than using an input fn.


I chose C over A and B because kubeflow pipelines is more sophisticated and requires more setup and effort because of its customizability.

   upvoted 1 times

  **gscharly** 6 months ago

**Selected Answer: A**

agree with guilhermebutzke

   upvoted 1 times

  **pinimichele01** 6 months, 1 week ago

**Selected Answer: A**

agree with guilhermebutzke

   upvoted 1 times

  **Shark0** 6 months, 2 weeks ago

**Selected Answer: C**

Given the requirement to minimize model development and training time while creating a training pipeline for a wide and deep model trained on datasets preprocessed using a SQL script in BigQuery, the most suitable option is:

C. Use the TensorFlow Extended SDK to implement the pipeline. Use the ExampleGen component with the BigQuery

executor to ingest the data, the Transform component to preprocess the data, and the Trainer component to launch a Vertex AI training job.

This option leverages TensorFlow Extended (TFX), which is designed for scalable and production-ready machine learning pipelines. The ExampleGen component with the BigQuery executor efficiently ingests data from BigQuery. The Transform component applies preprocessing steps to the data, and the Trainer component launches a Vertex AI training job, minimizing the time and effort required for model development and training.

   upvoted 1 times

  **Carlose2108** 7 months, 3 weeks ago

Why not C?

   upvoted 1 times

  **guilhermebutzke** 8 months, 1 week ago

My Answer: A

According with this documentation:

<https://cloud.google.com/vertex-ai/docs/tabular-data/tabular-workflows/overview>

A: CORRECT: BigQueryJobOp for running the existing preprocessing script that already resides there, CustomTrainingJobOp for launching custom training jobs on Vertex AI, which aligns with the requirement of using the pre-trained TensorFlow model.

B: Not Correct: While DataflowPythonJobOp can be used for preprocessing this increasing development time compared to the simpler BigQueryJobOp approach.

C and D: Not Correct: While possible, using the TensorFlow Extended SDK with its components introduces unnecessary complexity for this specific scenario. For example, why use ExampleGen? Implementing preprocessing within the model's input\_fn is generally not recommended due to potential efficiency drawbacks and training-serving skew issues.

   upvoted 3 times



  **BlehMaks** 9 months ago

**Selected Answer: A**

D is wrong. Google doesn't recommend to use input\_fn for preprocessing

[https://www.tensorflow.org/tfx/guide/tft\\_bestpractices#preprocessing\\_options\\_summary](https://www.tensorflow.org/tfx/guide/tft_bestpractices#preprocessing_options_summary)

   upvoted 2 times

  **pikachu007** 9 months, 1 week ago

**Selected Answer: D**

Addressing Limitations of Other Options:

Kubeflow Pipelines (A and B): While Kubeflow offers flexibility, it might require more setup and configuration, potentially increasing development time compared to TFX's integrated approach.

Separate Preprocessing (C): Using a separate Transform component for preprocessing can add complexity and potential overheads, especially for instance-level transformations that can often be directly integrated within the model's input pipeline.

   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.