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 251 DISCUSSI...**

Actual exam question from Google's Professional Machine Learning Engineer

Question #: 251

Topic #: 1

[All Professional Machine Learning Engineer Questions]

You are developing a training pipeline for a new XGBoost classification model based on tabular data. The data is stored in a BigQuery table. You need to complete the following steps:

1. Randomly split the data into training and evaluation datasets in a 65/35 ratio
2. Conduct feature engineering
3. Obtain metrics for the evaluation dataset
4. Compare models trained in different pipeline executions

How should you execute these steps?

A. 1. Using Vertex AI Pipelines, add a component to divide the data into training and evaluation sets, and add another component for feature engineering.
2. Enable autologging of metrics in the training component.
3. Compare pipeline runs in Vertex AI Experiments.

B. 1. Using Vertex AI Pipelines, add a component to divide the data into training and evaluation sets, and add another component for feature engineering.
2. Enable autologging of metrics in the training component.
3. Compare models using the artifacts' lineage in Vertex ML Metadata.

C. 1. In BigQuery ML, use the CREATE MODEL statement with BOOSTED_TREE_CLASSIFIER as the model type and use BigQuery to handle the data splits.
2. Use a SQL view to apply feature engineering and train the model using the data in that view.

3. Compare the evaluation metrics of the models by using a SQL query with the ML.TRAINING_INFO statement.

D. 1. In BigQuery ML, use the CREATE MODEL statement with BOOSTED_TREE_CLASSIFIER as the model type and use BigQuery to handle the data splits.

2. Use ML TRANSFORM to specify the feature engineering transformations and tram the model using the data in the table.

3. Compare the evaluation metrics of the models by using a SQL query with the ML.TRAINING_INFO statement.

[Show Suggested Answer]

by 👤 **pikachu007** at *Jan. 13, 2024, 3 p.m.*

Comments

Type your comment...

Submit

👤 **pikachu007** `Highly Voted 👍` 9 months, 1 week ago

`Selected Answer: A`

Option B: While Vertex ML Metadata provides artifact lineage, it's less comprehensive for model comparison than Experiments.
Options C and D: BigQuery ML is powerful for in-database model training, but it has limitations in pipeline orchestration, complex feature engineering, and detailed model comparison features, making it less suitable for this scenario.

👍 ↩ 🚩 upvoted 6 times

👤 **wences** `Most Recent ⊘` 1 month ago

`Selected Answer: A`

Can anyone give a good reason for the answers without using ChatGPT or Gemini?

👍 ↩ 🚩 upvoted 1 times

👤 **tardigradum** 2 months, 1 week ago

`Selected Answer: A`

BQ ML falls a bit short when it comes to building pipelines that include feature engineering and experiment comparison (it's better to use Vertex Pipelines and do the comparisons using Vertex Experiments).

👍 ↩ 🚩 upvoted 1 times

👤 **fitri001** 6 months, 1 week ago

`Selected Answer: A`

Flexibility and Control: Vertex AI Pipelines allow you to define a custom pipeline with separate components for data splitting, feature engineering, and XGBoost training using your preferred libraries (like BigQueryClient and xgboost). This provides more control and customization compared to BigQuery ML's limited model types and functionality.
Feature Engineering and Data Splitting: Separate components enable clear separation of concerns and potentially parallel execution for efficiency.
Autologging and Model Comparison: Vertex AI autologging simplifies capturing evaluation metrics during training. Vertex AI Experiments offer a centralized interface to compare metrics across different pipeline runs (potentially with varying hyperparameter configurations).

👍 ↩ 🚩 upvoted 1 times

   👤 **fitri001** 6 months, 1 week ago

   why not C & D?
   C & D. BigQuery ML: While BigQuery ML offers some XGBoost functionality, it has limitations:
   Limited Model Types: BigQuery ML doesn't provide the full flexibility of using custom XGBoost libraries with advanced configurations.
   Less Control over Feature Engineering: Feature engineering using SQL views might be restrictive compared to a dedicated component in Vertex AI Pipelines.
   Limited Model Comparison: While ML.TRAINING_INFO provides some insights, Vertex AI Experiments offer a more comprehensive view for comparing models across pipeline runs.

   👍 ↩ 🚩 upvoted 1 times

👤 **pinimichele01** 6 months, 1 week ago

`Selected Answer: A`

see b1a8fae

👍 ↩ ⚑ upvoted 1 times

⊟ 👤 **omermahgoub** 6 months, 1 week ago

A: Leverage Vertex AI Pipelines and Experiments

👍 ↩ ⚑ upvoted 1 times

⊟ 👤 **guilhermebutzke** 8 months ago

My Answer: A

A: CORRECT: It involves proper data splitting into training and evaluation sets and conducting feature engineering within the pipeline, fulfilling steps 1 and 2. Enabling autologging of metrics ensures that you can track and compare the performance of different model executions, fulfilling step 3.

B: Not Correct: Better use Vertex AI Experiments

C and D: Not Correct: BigQuery ML lacks functionalities for comparing models across pipeline runs. You would need to rely on external tools or custom scripts to extract and compare evaluation metrics, making the process less streamlined.
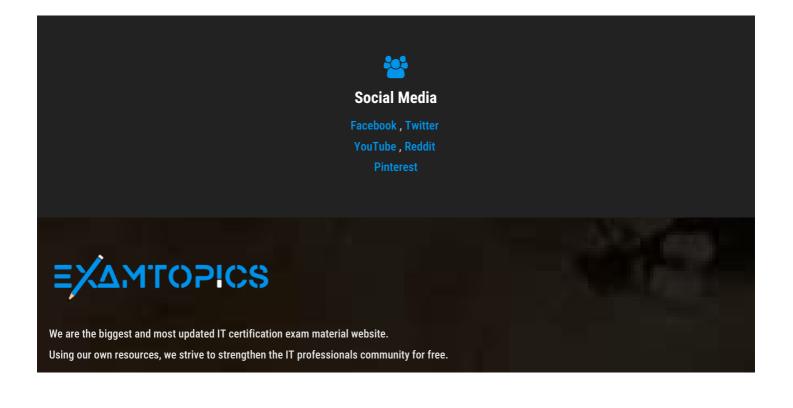
👍 ↩ ⚑ upvoted 2 times
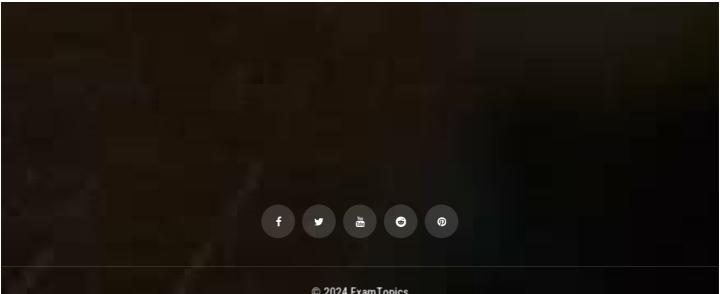
⊟ 👤 **b1a8fae** 9 months ago

Compare models in different pipeline executions -> go for Vertex AI experiments

👍 ↩ ⚑ upvoted 3 times