Google

ASSOCIATE-DATA-PRACTITIONER Exam

Google Cloud Associate Data Practitioner

Questions & Answers
Demo

Version: 4.0

Question: 1

Your retail company wants to predict customer churn using historical purchase data stored in BigQuery. The dataset includes customer demographics, purchase history, and a label indicating whether the customer churned or not. You want to build a machine learning model to identify customers at risk of churning. You need to create and train a logistic regression model for predicting customer churn, using the customer_data table with the churned column as the target label. Which BigQuery ML query should you use?

A)

```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT *
FROM customer_data;
```

B)

```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT * EXCEPT(churned),
       churned AS label
FROM customer_data;
```

C)

```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT * EXCEPT(churned)
FROM customer_data;
```

D)

```
CREATE OR REPLACE MODEL churn_prediction_model
OPTIONS(model_type='logistic_reg') AS
SELECT churned as label
FROM customer_data;
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: B

Explanation:

In BigQuery ML, when creating a logistic regression model to predict customer churn, the correct query should:

Exclude the target label column (in this case, churned) from the feature columns, as it is used for training and not as a feature input.

Rename the target label column to label, as BigQuery ML requires the target column to be named label.

The chosen query satisfies these requirements:

SELECT * EXCEPT(churned), churned AS label: Excludes churned from features and renames it to label.

The OPTIONS(model_type='logistic_reg') specifies that a logistic regression model is being trained.

This setup ensures the model is correctly trained using the features in the dataset while targeting the churned column for predictions.

Question: 2

Your company has several retail locations. Your company tracks the total number of sales made at each location each day. You want to use SQL to calculate the weekly moving average of sales by location to identify trends for each store. Which query should you use?

A)

```
SELECT store_id, date, total_sales, AVG(total_sales) OVER (
PARTITION BY store_id
ORDER BY total_sales RANGE BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_av
FROM store_sales_daily
```

B)

```
SELECT store_id, date, total_sales, AVG(total_sales)
OVER (
PARTITION BY date ORDER BY store_id ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ) as roll
FROM store_sales_daily
```

C)

```
SELECT store_id, date, total_sales, AVG(total_sales)
OVER (
PARTITION BY store_id
ORDER BY date ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_avg
FROM store_sales_daily
```

D)

```
SELECT store_id, date, total_sales, AVG(total_sales)
OVER (
PARTITION BY total_sales
ORDER BY date RANGE BETWEEN 6 PRECEDING AND CURRENT ROW ) as rolling_avg
FROM store_sales_daily
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C

Explanation:

To calculate the weekly moving average of sales by location:

The query must group by store_id (partitioning the calculation by each store).

The ORDER BY date ensures the sales are evaluated chronologically.

The ROWS BETWEEN 6 PRECEDING AND CURRENT ROW specifies a rolling window of 7 rows (1 week if each row represents daily data).

The AVG(total_sales) computes the average sales over the defined rolling window.

Chosen query meets these requirements:

- PARTITION BY store_id groups the calculation by each store.
- ORDER BY date orders the rows correctly for the rolling average.
- ROWS BETWEEN 6 PRECEDING AND CURRENT ROW ensures the 7-day moving average.

Question: 3

Your company is building a near real-time streaming pipeline to process JSON telemetry data from small appliances. You need to process messages arriving at a Pub/Sub topic, capitalize letters in the serial number field, and write results to BigQuery. You want to use a managed service and write a minimal amount of code for underlying transformations. What should you do?

A. Use a Pub/Sub to BigQuery subscription, write results directly to BigQuery, and schedule a transformation query to run every five minutes.

B. Use a Pub/Sub to Cloud Storage subscription, write a Cloud Run service that is triggered when objects arrive in the bucket, performs the transformations, and writes the results to BigQuery.

C. Use the “Pub/Sub to BigQuery” Dataflow template with a UDF, and write the results to BigQuery.

D. Use a Pub/Sub push subscription, write a Cloud Run service that accepts the messages, performs the transformations, and writes the results to BigQuery.

Answer: C

Explanation:

Using the "Pub/Sub to BigQuery" Dataflow template with a UDF (User-Defined Function) is the optimal choice because it combines near real-time processing, minimal code for transformations, and scalability. The UDF allows for efficient implementation of custom transformations, such as capitalizing letters in the serial number field, while Dataflow handles the rest of the managed pipeline seamlessly.

Question: 4

You want to process and load a daily sales CSV file stored in Cloud Storage into BigQuery for downstream reporting. You need to quickly build a scalable data pipeline that transforms the data while providing insights into data quality issues. What should you do?

A. Create a batch pipeline in Cloud Data Fusion by using a Cloud Storage source and a BigQuery sink.

B. Load the CSV file as a table in BigQuery, and use scheduled queries to run SQL transformation scripts.

C. Load the CSV file as a table in BigQuery. Create a batch pipeline in Cloud Data Fusion by using a BigQuery source and sink.

D. Create a batch pipeline in Dataflow by using the Cloud Storage CSV file to BigQuery batch template.

Answer: A

Explanation:

Using Cloud Data Fusion to create a batch pipeline with a Cloud Storage source and a BigQuery sink is the best solution because:

Scalability: Cloud Data Fusion is a scalable, fully managed data integration service.

Data transformation: It provides a visual interface to design pipelines, enabling quick transformation of data.

Data quality insights: Cloud Data Fusion includes built-in tools for monitoring and addressing data quality issues during the pipeline creation and execution process.

Question: 5

You manage a Cloud Storage bucket that stores temporary files created during data processing. These temporary files are only needed for seven days, after which they are no longer needed. To reduce storage costs and keep your bucket organized, you want to automatically delete these files once they are

older than seven days. What should you do?

A. Set up a Cloud Scheduler job that invokes a weekly Cloud Run function to delete files older than seven days.

B. Configure a Cloud Storage lifecycle rule that automatically deletes objects older than seven days.

C. Develop a batch process using Dataflow that runs weekly and deletes files based on their age.

D. Create a Cloud Run function that runs daily and deletes files older than seven days.

Answer: B

Explanation:

Configuring a Cloud Storage lifecycle rule to automatically delete objects older than seven days is the best solution because:

Built-in feature: Cloud Storage lifecycle rules are specifically designed to manage object lifecycles, such as automatically deleting or transitioning objects based on age.

No additional setup: It requires no external services or custom code, reducing complexity and maintenance.

Cost-effective: It directly achieves the goal of deleting files after seven days without incurring additional compute costs.