

[Start Lab](#)

01:30:00

# Summarize Text using SQL and LLMs in BigQuery ML

 Lab  1 hour  No cost  Introductory**GSP835****Google Cloud Self-Paced Labs**[Lab instructions and tasks](#)

GSP835

Overview

Setup and requirements

Task 1. Preparing the data

Task 2. Create the BigQuery dataset

Task 3. Create the external connection

Task 4. Create a remote ML model

Task 5. Generate text using the ML model

Congratulations!



## Overview

In this lab, you take steps to perform summarization of source code from GitHub, a popular open-source, source code repository, and identify the primary programming language using Vertex AI's Large Language Model (LLM) for text generation (text-bison) and hosted remote functions in BigQuery. The source data is from the GitHub Archive Project, which contains a full snapshot of over 2.8 million open source GitHub repositories in Google BigQuery Public Datasets.

## Learning objectives

In this lab, you learn how to create:

- A BigQuery dataset to contain a model.
- A BigQuery model that hosts the Vertex AI PaLM API as a remote function.
- An external connection to establish the connection between BigQuery and Vertex AI.

## Setup and requirements

### Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause

## Setup and requirements

### Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause

## Setup and requirements

### Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

**Note:** Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause

"Password"



You can also find the **Password** in the **Lab Details** panel.

6. Click **Next**.

**Important:** You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.

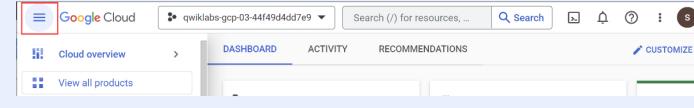
**Note:** Using your own Google Cloud account for this lab may incur extra charges.

7. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Google Cloud console opens in this tab.

**Note:** To view a menu with a list of Google Cloud products and services, click the **Navigation menu** at the top-left.



## Task 1. Preparing the data

You will be using the source code content from the `github_repos` dataset in the Google BigQuery Public Datasets. To do so, do the following:

1. From the **Navigation menu** choose **BigQuery**. If prompted click **Done**.
2. In the BigQuery console, search for `github_repos` and press **ENTER**.

**Note:** If you do not see the `github_repos` dataset after performing a search you may need to select **SEARCH ALL PROJECTS** to add the dataset.

3. Click on the star next to the dataset that is listed as the search result.

A screenshot of the BigQuery search interface. A search bar at the top contains the query "`github_repos`". Below the search bar, it says "Found 2 results." and there is a "SEARCH ALL PROJECTS" button. Two datasets are listed:

- `bigrquery-public-data` (with a star icon)
- `github_repos` (with a star icon)

A "SHOW MORE" link is located below the second item.

4. Expand the `github_repos` dataset and select the `sample_contents` table. This table contains sample data containing 10% of the full data in the contents table. Click **PREVIEW**.

## Task 2. Create the BigQuery dataset

A [BigQuery dataset](#) is a collection of tables. All tables in a dataset are stored in the same data location.

1. Clear the search filter in the BigQuery console and select **Create dataset** using the three dots next to your lab's project.



2. In the **Create dataset** form enter "bq\_llm" as the **Dataset ID** then click **CREATE DATASET**.

**Create dataset**

**Project ID**: `qwiklabs-gcp-01-62d8f9928762` [CHANGE](#)

**Dataset ID \***: `bq_llm`  
Letters, numbers, and underscores allowed

**Location type** [?](#)

- Region  
Specify a region to colocate your datasets with other Google Cloud services.
- Multi-region  
Allow BigQuery to select a region within a group to achieve higher quota limits.

**Multi-region \***: US (multiple regions in United States)

**Default table expiration**

Enable table expiration [?](#)

Default maximum table age  Days

**Advanced options** [▼](#)

**CREATE DATASET** **CANCEL**

The dataset is going to be used to house the model that you create in the next tasks of this lab.

Typically, data that is used by an ML application is stored in a table in the dataset as well. Since the data is in a BigQuery public dataset, you reference that data directly from the public data using an external connection. You create the external connection in the next task of this lab.

Click **Check my progress** to verify the objective.

Create a dataset to house the ML model.

**Check my progress**

## Task 3. Create the external connection

Now, create an external connection and save the Service Account ID from the connection configuration details.

1. Click the **+ ADD** button on the BigQuery Explorer pane, then click **Connections to external data sources** in the popular sources listed.
2. Select **Connection type** as Vertex AI remote models, remote functions and BigLake (Cloud Resource) and set **Connection ID** to `llm-connection`.
3. Click **CREATE CONNECTION**.
4. Select `us.llm-connection` under the External connections section of the project's datasets. Copy the Service Account ID generated from the external connection configuration details to your clipboard. You use it in the next step.

## Grant permissions to the Service Account

You need to grant the Service Account generated by the external connection access to the Vertex AI service. To do so:

1. From the **Navigation menu** select **IAM & Admin**.
2. Click **+ GRANT ACCESS** on the IAM page.
3. Paste the Service Account ID generated by the external connection in the **New principals** form input.
4. Set the **Role** to `Vertex AI User` then click **SAVE**.

Click **Check my progress** to verify the objective.



## Task 4. Create a remote ML model

In this task, you create a remote model that represents a hosted Vertex AI large language model (LLM).

1. From the **Navigation menu** select **BigQuery**. Click on **+ Compose new query**.

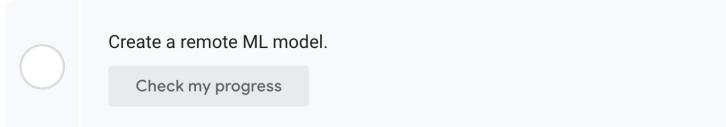
2. Run the following SQL query in a new tab in BigQuery Explorer:

```
CREATE OR REPLACE MODEL bq_llm.llm_model
REMOTE WITH CONNECTION `us.llm-connection`
OPTIONS (remote_service_type =
'CLOUD_AI_LARGE_LANGUAGE_MODEL_V1');
```

This creates a model named `llm_model` in the dataset `bq_llm` created earlier in the lab. The model leverages the `CLOUD_AI_LARGE_LANGUAGE_MODEL_V1` of Vertex AI as a remote function. Once completed you see the model appear in the BigQuery console.

The screenshot shows the BigQuery Explorer interface. On the left, there's a sidebar with 'Explorer' and a search bar. The main area shows a list of resources under 'bq\_llm'. A specific item, 'llm\_model', is selected and highlighted in blue. The right side has two tabs: 'DETAILS' (selected) and 'SCHEMA'. Under 'DETAILS', it shows 'Model type: Remote'. Below that is the 'Model Details' section with fields like Model ID (qwiklabs-gcp-04-61398b71a136.bq\_llm.llm\_model), Description, Date created (Aug 23, 2023, 4:31:36PM UTC-4), Model expiration (Never), Date modified (Aug 23, 2023, 4:31:36PM UTC-4), Data location (US), Remote connection (us.llm-connection), and Remote service type (REMOTE\_SERVICE\_TYPE\_UNSPECIFIED). At the bottom, there's a 'Training Options' section with a note that training options were added in the script and an 'Actual iterations' field set to 1.

Click **Check my progress** to verify the objective.



## Task 5. Generate text using the ML model

Use the ML model created to generate, summarize, or categorize text.

1. Run the following query in a new tab of the BigQuery console:

```
SELECT
  ml_generate_text_result['predictions'][0]['content'] AS
```

```

generated_text,
    ml_generate_text_result['predictions'][0]['safetyAttributes']
        AS safety_attributes,
    * EXCEPT (ml_generate_text_result)
FROM
    ML.GENERATE_TEXT(
        MODEL `bq_llm.llm_model`,
        (
            SELECT
                CONCAT('Can you read the code in the following text and
                generate a summary for what the code is doing and what language
                it is written in:', content)
                    AS prompt from `bigquery-public-
                data.github_repos.sample_contents`
                limit 5
),
STRUCT(
    0.2 AS temperature,
    100 AS max_output_tokens));

```

2. Review the following to understand the SQL query in more depth:

`ml_generate_text_result` is the response from the text generation model in JSON format that contains both **content** and **safety** attributes:

- **Content** represents the generated text result.
- **Safety** attributes represent built-in content filters with an adjustable threshold that is enabled in the Vertex AI Palm API to avoid any unintended or unforeseen responses from the LLM. Responses are blocked if they violate the safety thresholds defined.

**ML.GENERATE\_TEXT** is the construct used in BigQuery to access the Vertex AI LLM to perform text generation tasks.

**CONCAT** appends the supplied **PROMPT** statement to a database record.

`github_repos` is the public dataset name and `sample_contents` is the name of the table that holds the data you use in the prompt design.

**Temperature** is the prompt parameter to control the randomness of the response - the lesser, the better the relevance.

**Max\_output\_tokens** is the number of words you want in response.

The query response should look similar to the following:

The screenshot shows a BigQuery job results page with several rows of generated text and their associated safety scores. The columns include 'Job', 'Generated\_Text', 'Safety\_Attributes', and 'Prompt'. The generated text varies slightly between rows, reflecting the different prompts used for each row. The safety scores range from 0.0 to 0.2, with most being 0.0 or 0.1. The prompt for each row is identical, asking for a summary of the provided code.

Job	Generated_Text	Safety_Attributes	Prompt
1	1. This is a simple PHP script to check if a user has set a preference which enables the service to log them in. If they have, it will log them in. It also checks if they have a token stored in their session, and if they do, it logs them in. It then creates a new session for them. It also checks if there is a table named 'tokens' in the database, and if there is, it inserts a new row into it. Finally, it returns a success message.	0.0	Can you read the code in the following text and generate a summary for what the code is doing and what language it is written in?: content
2	2. This is a Go code. It is reading the file GetAndUpdate function in the environments.go file. It is using the OpenAPI client to make a GET request to the file. It then parses the response to an environment object. It then updates the environment object with the new values. It then returns the updated environment object.	0.0	Can you read the code in the following text and generate a summary for what the code is doing and what language it is written in?: content
3	3. This is Go code. It is reading the file GetAndUpdate function in the environments.go file. It is using the OpenAPI client to make a GET request to the file. It then parses the response to an environment object. It then updates the environment object with the new values. It then returns the updated environment object.	0.0	Can you read the code in the following text and generate a summary for what the code is doing and what language it is written in?: content
4	4. This is Go code. It is reading the file GetAndUpdate function in the environments.go file. It is using the OpenAPI client to make a GET request to the file. It then parses the response to an environment object. It then updates the environment object with the new values. It then returns the updated environment object.	0.0	Can you read the code in the following text and generate a summary for what the code is doing and what language it is written in?: content
5	5. This is Go code. It is reading the file GetAndUpdate function in the environments.go file. It is using the OpenAPI client to make a GET request to the file. It then parses the response to an environment object. It then updates the environment object with the new values. It then returns the updated environment object.	0.0	Can you read the code in the following text and generate a summary for what the code is doing and what language it is written in?: content

Click **Check my progress** to verify the objective.

Generate text using the ML model.

**Check my progress**

# Congratulations!

Congratulations! You have successfully used a Vertex AI Text Generation LLM programmatically to perform text analytics on your data using only SQL-queries. Check out [Vertex AI LLM product documentation](#) to learn more about available models.

## Google Cloud training and certification

...helps you make the most of Google Cloud technologies. [Our classes](#) include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. [Certifications](#) help you validate and prove your skill and expertise in Google Cloud technologies.

**Manual Last Updated: March 21, 2024**

**Lab Last Tested: March 21, 2024**

Copyright 2024 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.