Building Demand Forecasting with BigQuery ML

Daniele Ciarlo (20018325)

Overview

This tutorial shows you how to create a time series model to forecast the demand of products using BigQuery ML.
BigQuery is a fully managed analytics database that offers built-in features like business intelligence, machine learning and geospatial analysis.
We will only use the ML features to train and validate a model on the NYC Citi Bike Trips public dataset. Later on, the same model will be used to forecast demand for bikes at each bike station in the next 30 days.

Set up your environment

Note: This step is subdivided in two different parts, one containing the instructions to follow if you are doing this tutorial on qwiklabs and the other containing instructions if you are doing this directly on your Google Cloud Platform (GCP) account.

On Qwiklabs

Before you click the Start Lab button read these instructions. Labs are **timed** and you **cannot pause them**. The timer shows you how long the resources on Google Cloud will be available to you.
This is due to the fact that Qwiklabs lets you use a temporary Google Cloud account just for the duration of the lab.

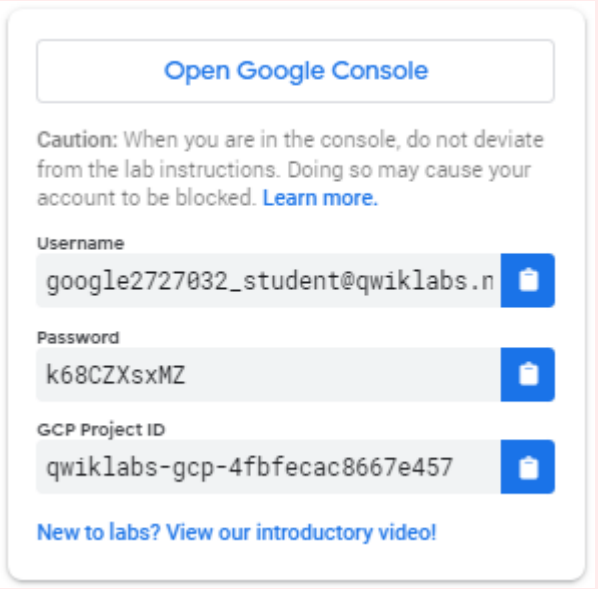
To complete this tutorial you will need:

- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

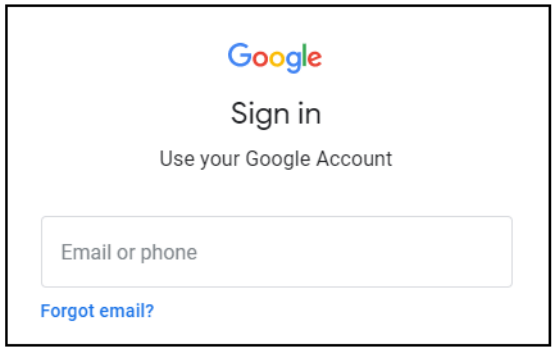
Note: If you are using a Chrome OS device, open an Incognito window to run this lab.

How to start:

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. It should appear a panel with the temporary credentials that you must use for this lab.



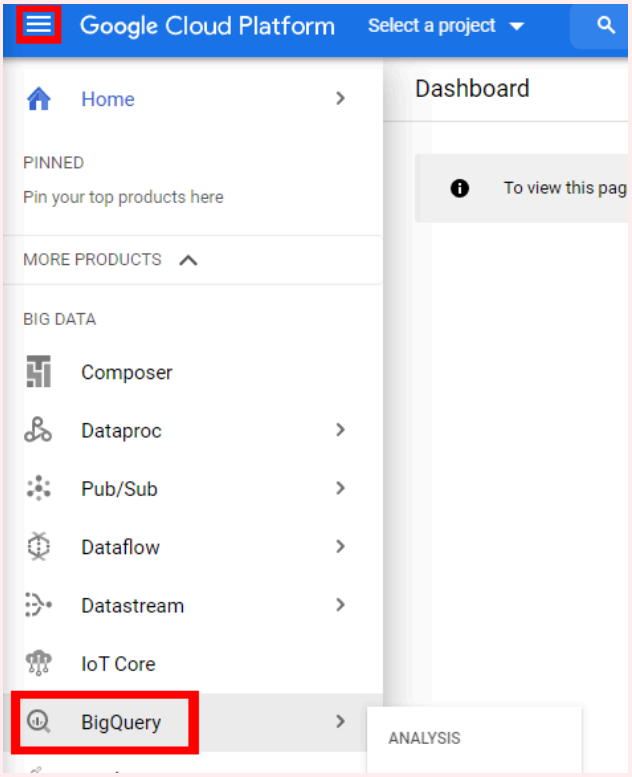
2. Copy the username and click **Open Google Console**. This will opens another tab showing the **Sign in** page.



If you see the **Choose an account** page, click **Use Another Account**.

3. Paste the username, then copy and paste the password.
4. Click through these 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 some time the Cloud Console opens in this tab.
5. From the left Navigation Menu click on BigQuery.



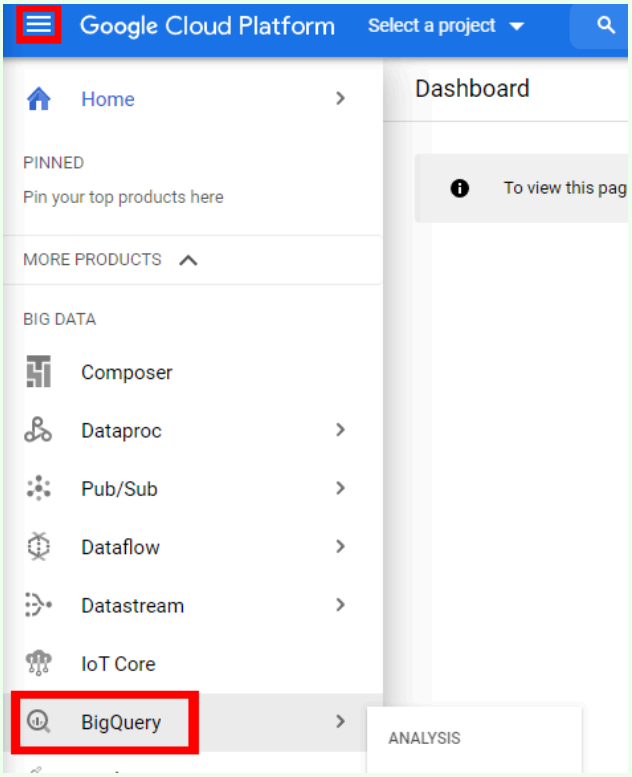
6. The Welcome to BigQuery in the Cloud Console message box opens. This message box provides a link to the quickstart guide and the release notes. Click **Done**.
   The BigQuery console opens.

Directly with your GCP account

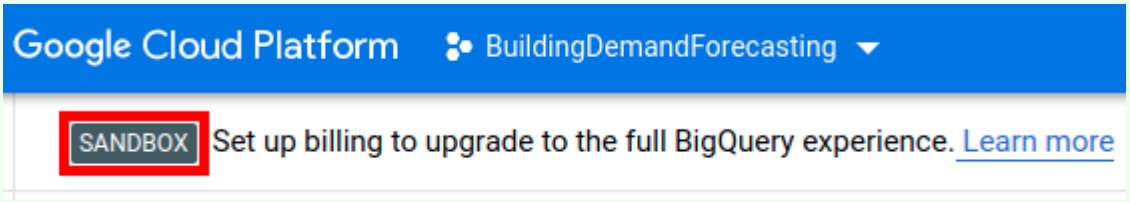
If you follow the steps of this tutorial on a free Google account that does not have billing options configured (no registered credit card or other means of payment), no charges will apply since you are in **BigQuery Sandbox**.

To see if you really are in "**Sandbox** Mode":

- Open up **Google Cloud Console.**
- From the left **Navigation Menu** click on **BigQuery.**



- Check the top left corner, if there is a label like the one below you are good to go:



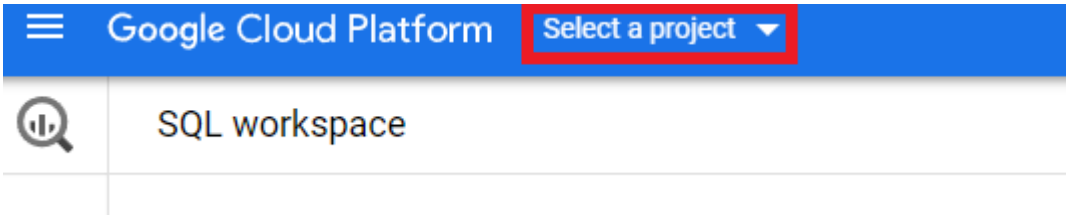
The sandbox lets you experience BigQuery and the Cloud Console without providing a credit card, creating a billing account, or enabling billing for your project.
This way you can replicate the entire tutorial without spending any money.
You can read more about BigQuery Sandbox here.

If you are not in **BigQuery Sandbox,** be aware of possible incurring charges!

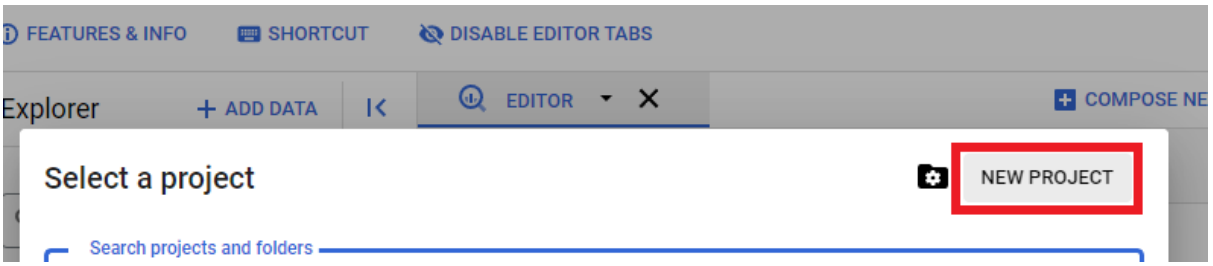
Before we begin (optional but recommended)

This step is not necessary but is suggested to improve the ease of use and to make the tutorial more manageable.
We recommend to create a new project in the following way:

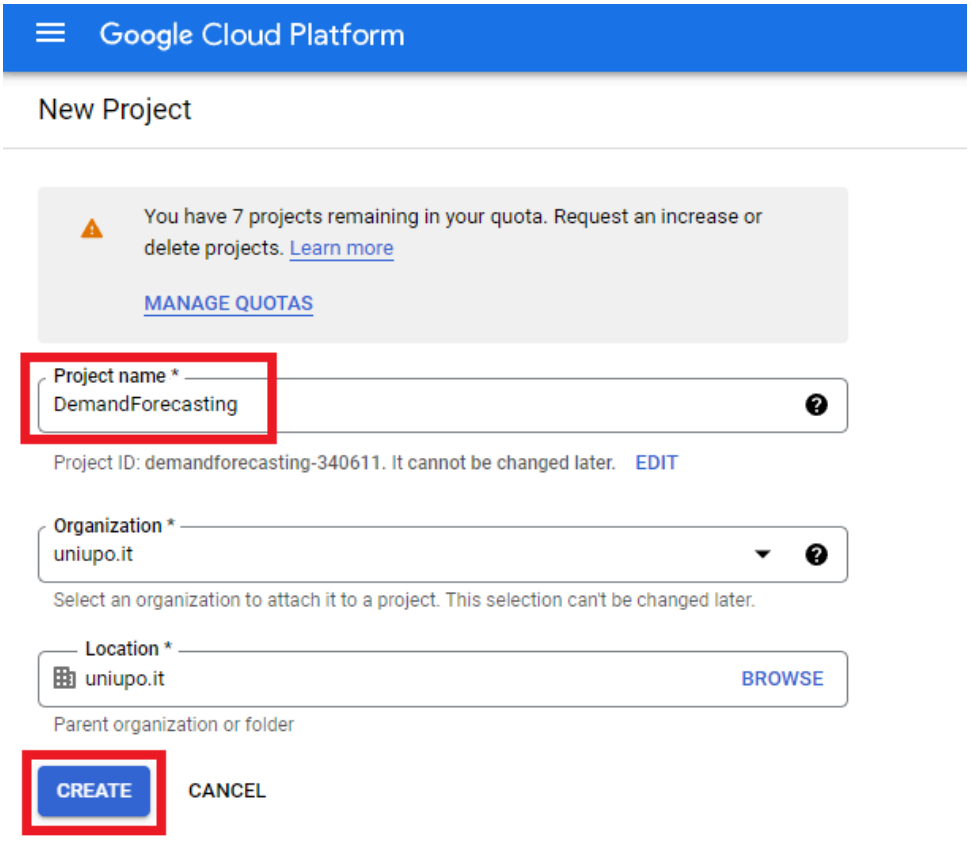
- Click on the button next to the Google Cloud Platform logo like shown below:



- This will prompt a new window, to create a new project press the **NEW PROJECT** button.



- In the following type `DemandForecasting` (or any other name you like) as **Project name** and click the **CREATE** button.



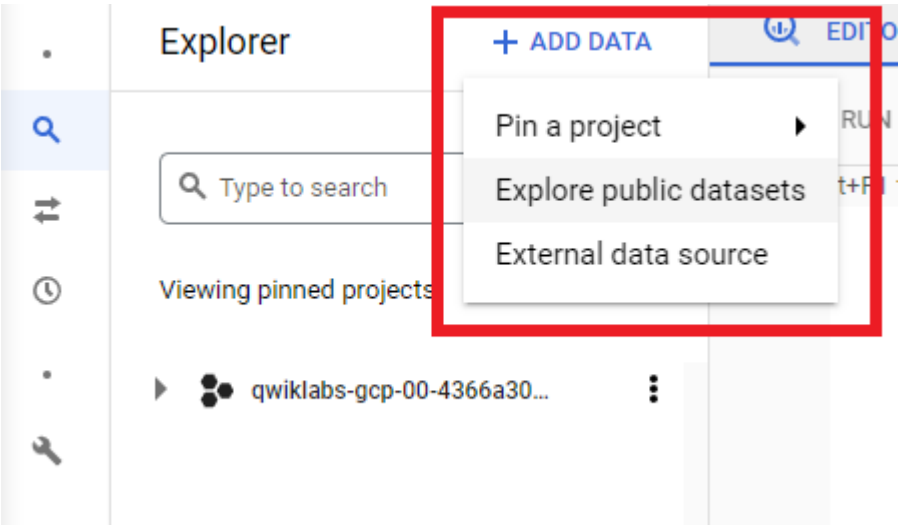
- Now that you have created the project, select it from the drop down menu on the right of the Google Cloud Platform logo (same button used above to create the project).

Explore the dataset

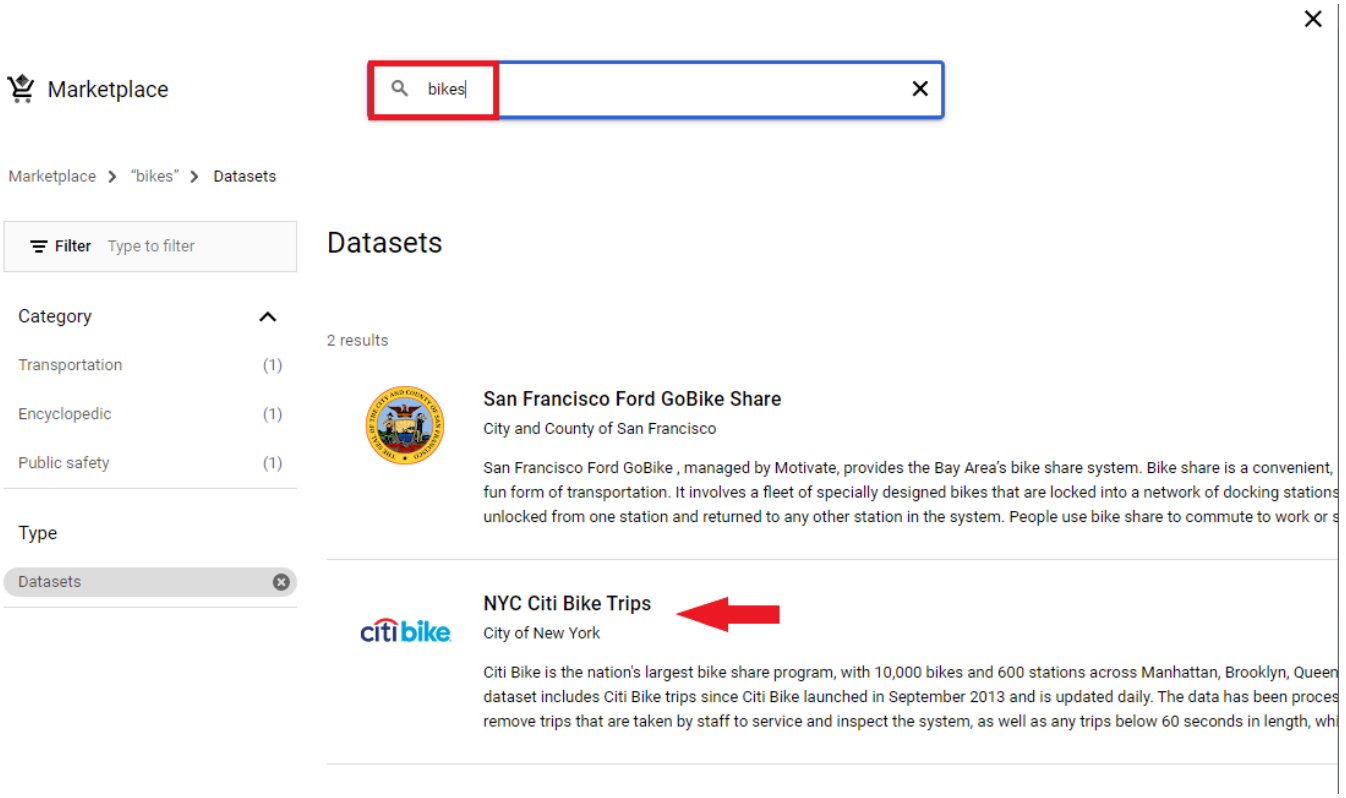
In this step we will explore one of the public datasets available on the Google Cloud Platform marketplace, specifically the NYC Bike Trips dataset. Analyzing it is a fundamental step towards understanding how the dataset is made and what we can achieve with it.

This is necessary since we will use those data during the tutorial to train and evaluate the forecasting model.

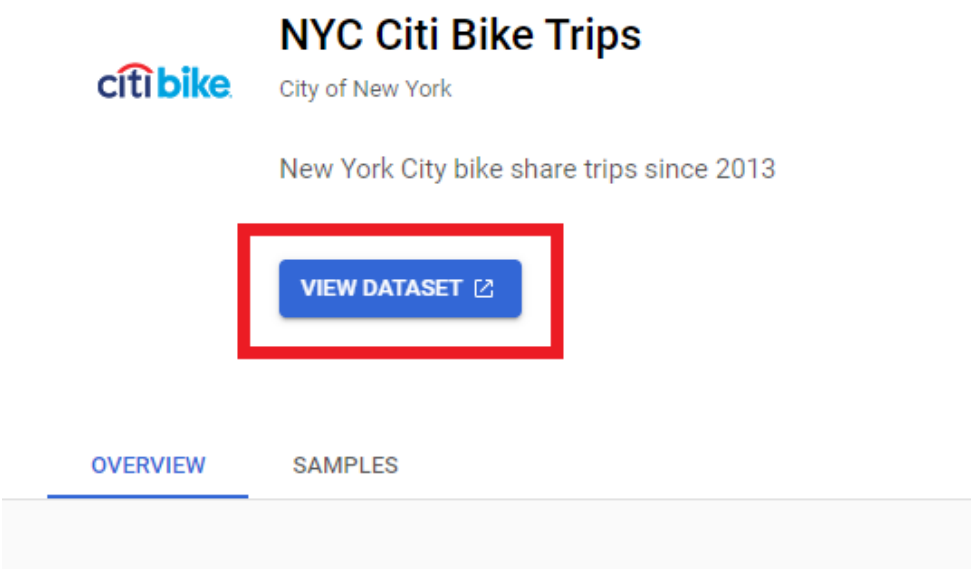
1. Let's start by selecting **Add Data** and then choosing **Explore public datasets**.



2. Write "bikes" in the search bar and hit enter. Among the results there will be the **NYC Bike Trips** dataset, select it.



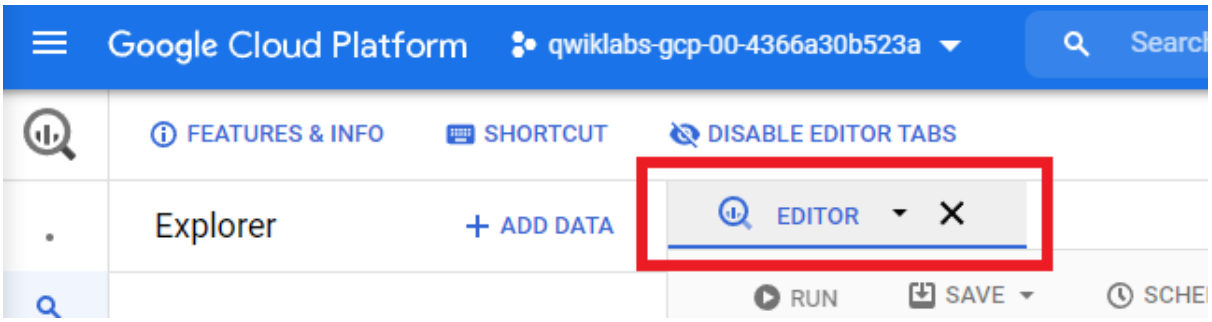
3. Open the dataset by clicking the **VIEW DATASET** button.



The dataset contains daily records about sharing stations and shared bikes in New York City. Now we will try to run some queries on the dataset to answer simple questions, this will help us to understand how to use the available data and what we can do with it.

The first question we can try to answer is: Could you name some New York locations where you can hire bikes?

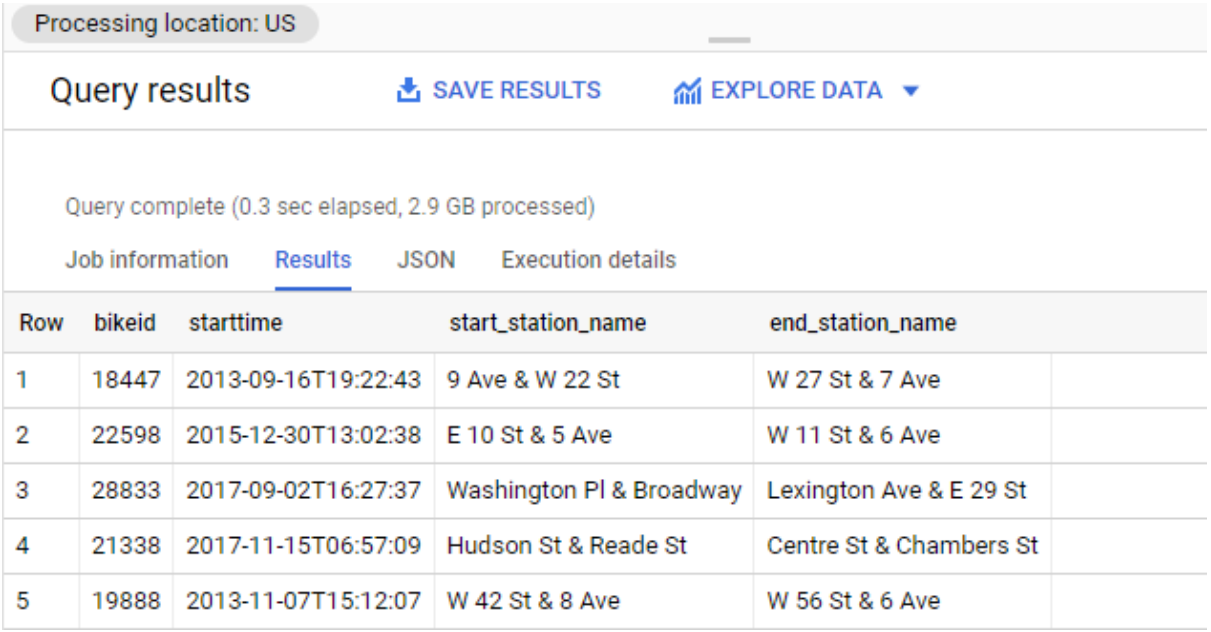
1. Open the query Editor by clicking on the **Editor** tab:



2. Add the following code:

```
SELECT
   bikeid,
   starttime,
   start_station_name,
   end_station_name,
FROM
  `bigquery-public-data.new_york_citibike.citibike_trips`
LIMIT 5
```

3. Then click **Run**.
4. At this point you should see a table similar to the one below:


Processing location: US

Query results SAVE RESULTS EXPLORE DATA

Query complete (0.3 sec elapsed, 2.9 GB processed)

Job information Results JSON Execution details

Row	bikeid	starttime	start_station_name	end_station_name
1	18447	2013-09-16T19:22:43	9 Ave & W 22 St	W 27 St & 7 Ave
2	22598	2015-12-30T13:02:38	E 10 St & 5 Ave	W 11 St & 6 Ave
3	28833	2017-09-02T16:27:37	Washington Pl & Broadway	Lexington Ave & E 29 St
4	21338	2017-11-15T06:57:09	Hudson St & Reade St	Centre St & Chambers St
5	19888	2013-11-07T15:12:07	W 42 St & 8 Ave	W 56 St & 6 Ave

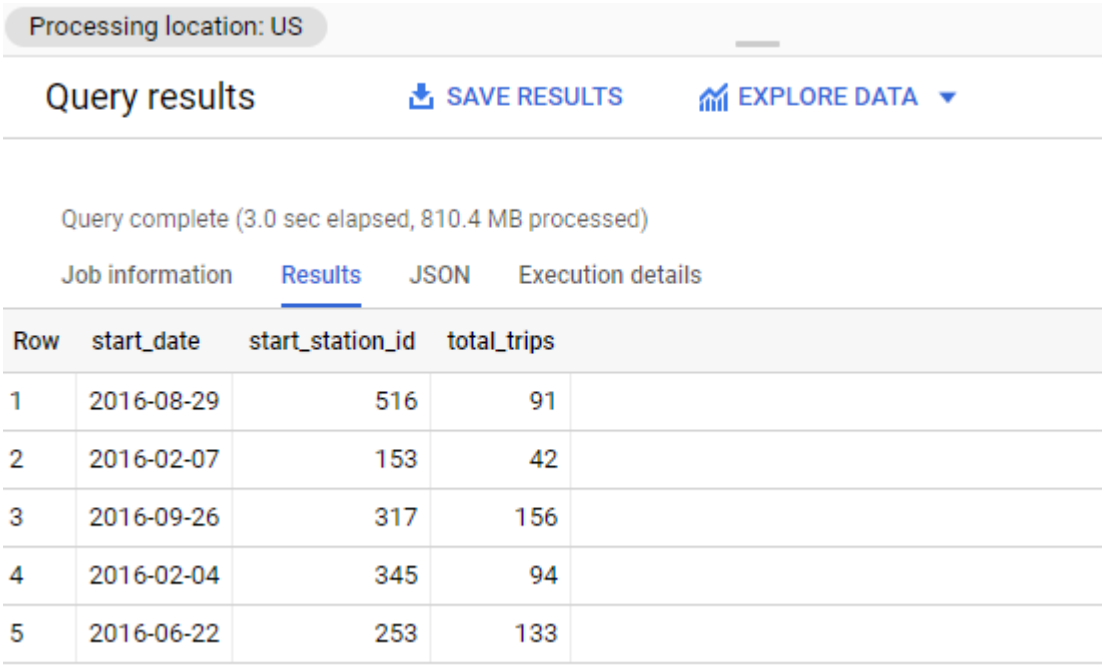
This table is made of five rows (due to the `LIMIT 5` flag) each containing information about the sharing session of one bike, specifically it contains the bike id, the starting time of the bike sharing, the starting station and the arrival station.

Let's try to answer another question: Could you list some stations and their total number of trips during 2016?

1. Start by replacing the previous query with this one:

```
SELECT
  EXTRACT (DATE FROM TIMESTAMP(starttime)) AS start_date,
  start_station_id,
  COUNT(*) as total_trips
FROM
 `bigquery-public-data.new_york_citibike.citibike_trips`
WHERE
   starttime BETWEEN DATE('2016-01-01') AND DATE('2017-01-01')
GROUP BY
    start_station_id, start_date
LIMIT 5
```

2. Then click **Run**.
3. At this point you should see a table similar to the one below:



Row	start_date	start_station_id	total_trips
1	2016-08-29	516	91
2	2016-02-07	153	42
3	2016-09-26	317	156
4	2016-02-04	345	94
5	2016-06-22	253	133

The table above contains, once again, five rows, each one showing the total number of trips started from the listed station, during that specific day.

Let's analyze the query to understand how it works:

- Using the WHERE clause we have selected only results from the time range specified by the initial question
- COUNT(*) , as the name suggest, count the number of rows
- GROUP BY combine non-distinct values, giving us distinct rows each with a specific station and the count of its trips

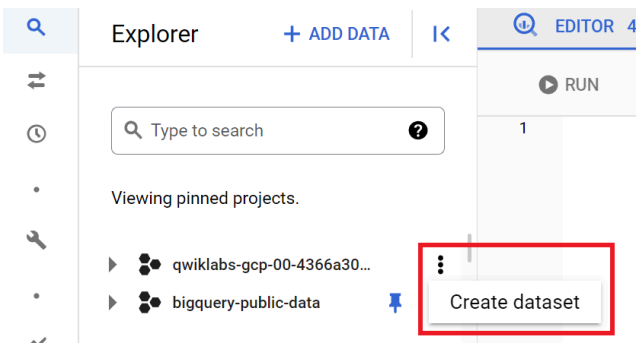
Now that you have carried out some tests with your own hands, you should have a broader understanding of the available data and how they could be used to build a forecasting model.

Cleaned training data

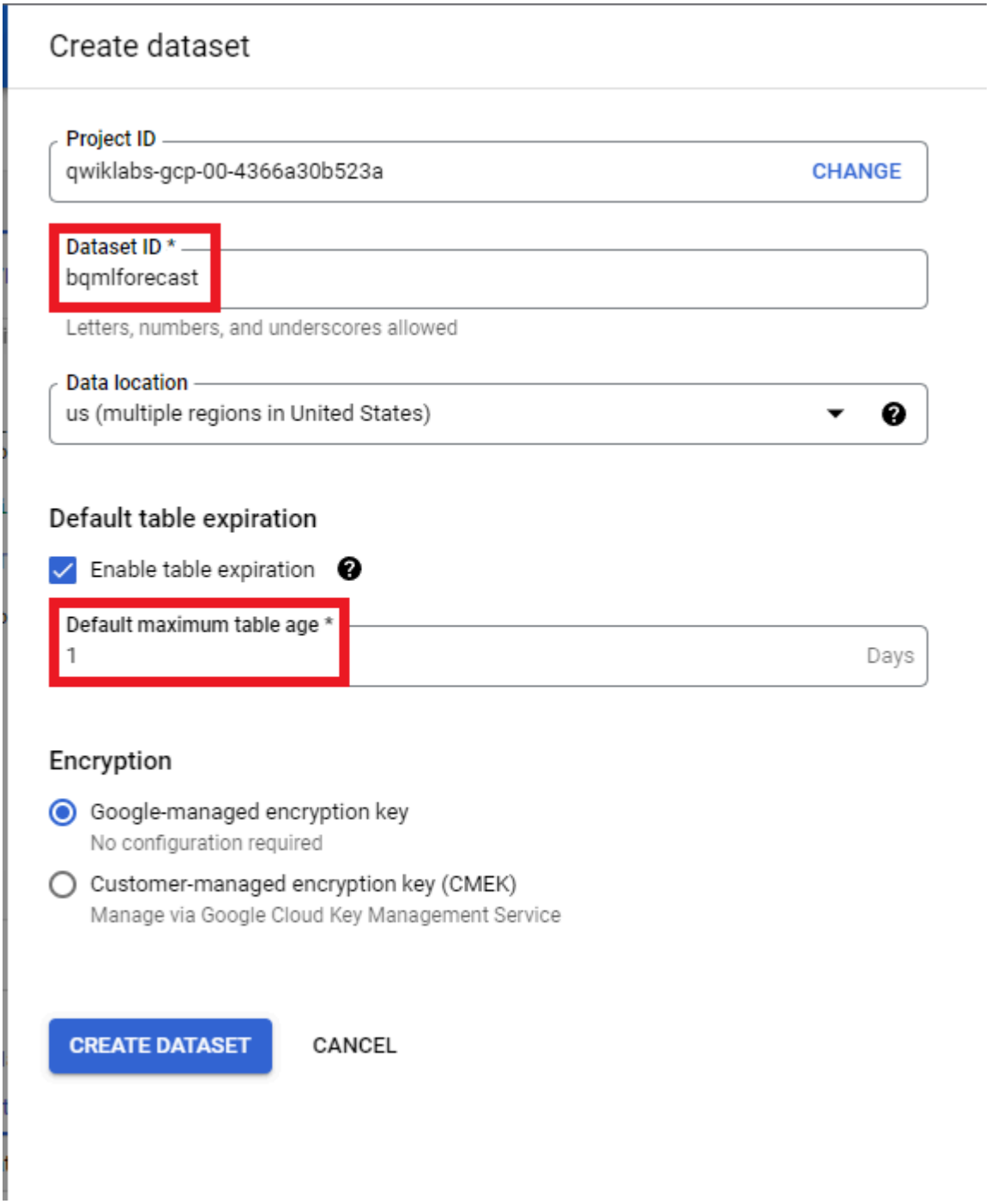
The data retrieved by queries can be stored as a table or a view. Since the information contained in the last query was exactly what we need to train our model, the next step will be to create a dataset using a subset of those data.

Create a dataset

- Start by clicking the **options** button (3 dots) of the project that starts with "Qwiklabs", at this point you should see the **Create Dataset** button.



- Click **Create Dataset**.
- This will open up a window listing the default configuration of the dataset: change the dataset name to `bqmlforecast` and the default table expiration as **1** day. Leave to default all the other options. (Of course you can use another dataset name, just remember to use it during the rest of the tutorial!)



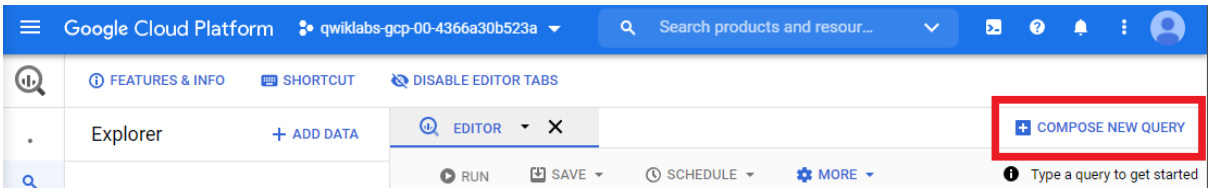
- Click **Create Dataset**.

You have successfully created an empty dataset!

Create the table

The next step will be to fill it with the result of a query.

1. Click the **Compose new query** button.



2. **Run** this slightly modified version of the previous query to generate data:

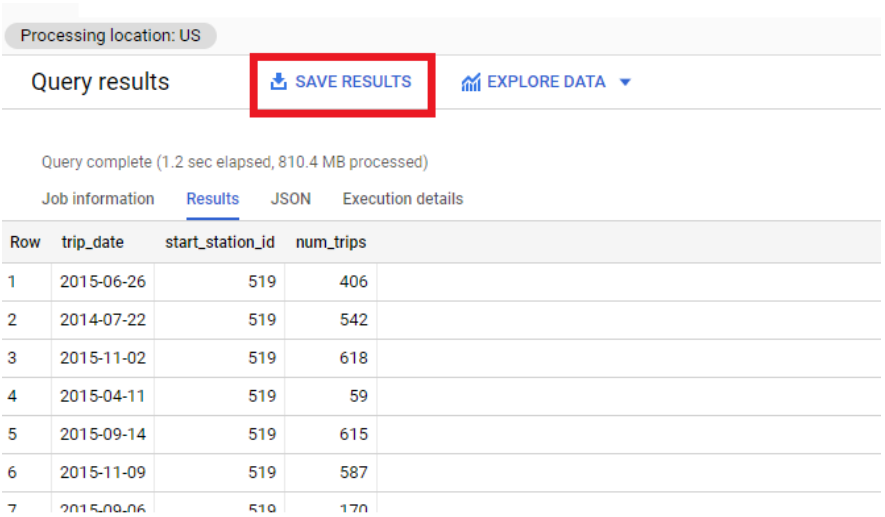
```
SELECT
 DATE(starttime) AS trip_date,
 start_station_id,
 COUNT(*) AS num_trips
FROM
 `bigquery-public-data.new_york_citibike.citibike_trips`
WHERE
 starttime BETWEEN DATE('2014-01-01') AND ('2016-01-01')
 AND start_station_id IN (521,435,497,293,519)
GROUP BY
 start_station_id,
 trip_date
```

NOTE:
This query differs from the second one by two aspects:

1. It broadens the time range for the retrieved results
2. Instead of using `LIMIT` that computes **all** the rows and just limits the visualized outcome, this query specifies 5 station ids in the `WHERE` clause, this way the query will retrieve **only** the data about those stations.

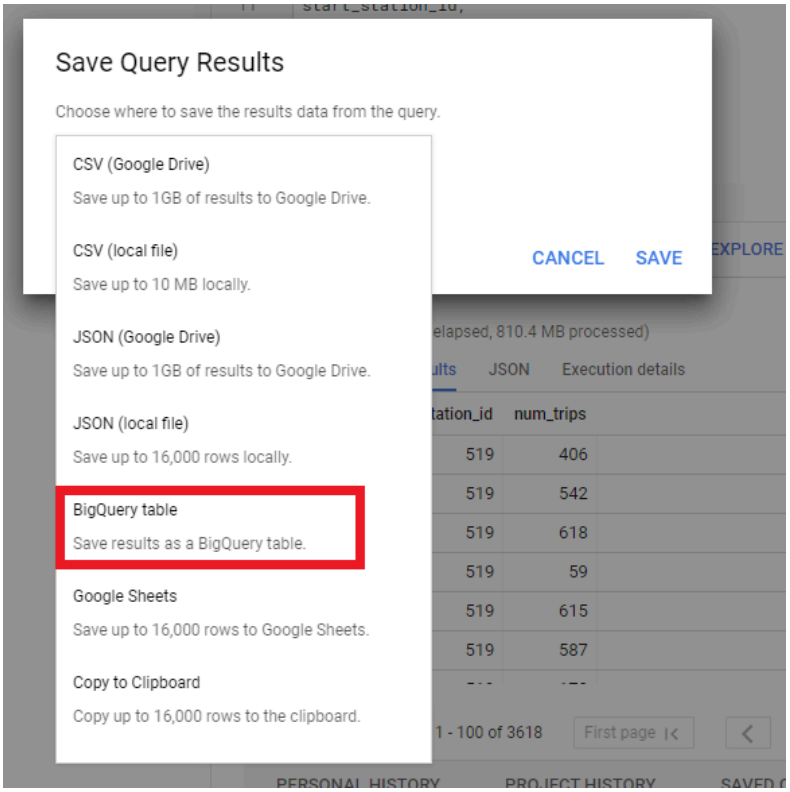
Other than saving computation, limiting the data to 5 stations helps with the model training time. Using the entire dataset would require more time than is available for this tutorial.

3. Click **Save Results**.



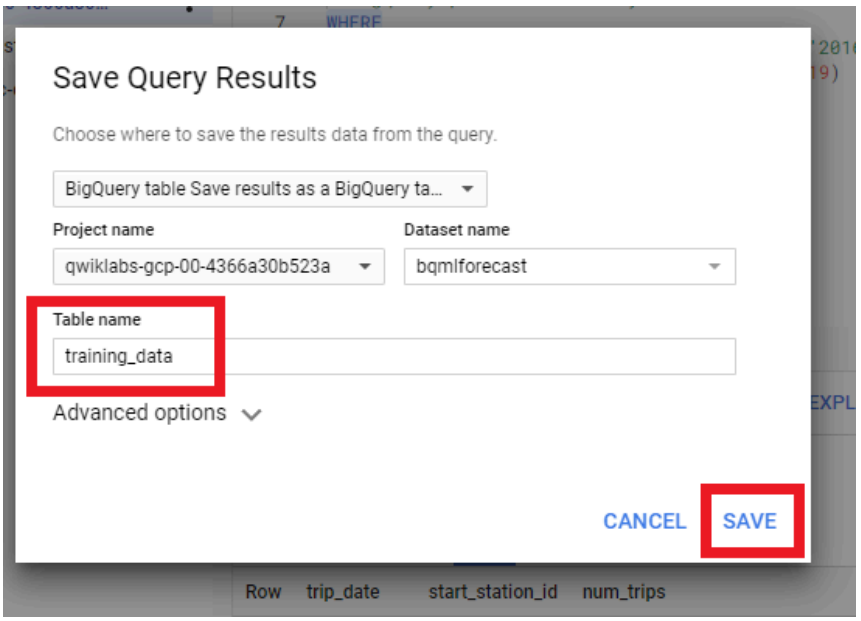
Row	trip_date	start_station_id	num_trips
1	2015-06-26	519	406
2	2014-07-22	519	542
3	2015-11-02	519	618
4	2015-04-11	519	59
5	2015-09-14	519	615
6	2015-11-09	519	587
7	2015-09-06	519	170

4. In the dropdown menu, select **BigQuery Table**.



5. Name the table `training_data`. (As before, you can set a different name, just remember to use it for the rest of the tutorial.)

6. Click **Save**.



Training a model

Now that we have a dataset, we can use it to train a Machine Learning model that will perform demand forecasting.

1. Write the following query into the Editor:

```
CREATE OR REPLACE MODEL bqmlforecast.bike_model
  OPTIONS(
    MODEL_TYPE='ARIMA',
    TIME_SERIES_TIMESTAMP_COL='trip_date',
    TIME_SERIES_DATA_COL='num_trips',
    TIME_SERIES_ID_COL='start_station_id',
    HOLIDAY_REGION='US'
  ) AS
  SELECT
    trip_date,
    start_station_id,
    num_trips
  FROM
    bqmlforecast.training_data
```

2. Press the **Run** button to start the training.
3. The training will take approximately 2 minutes. If you are curious about what is happening right now read the colored paragraph just below.

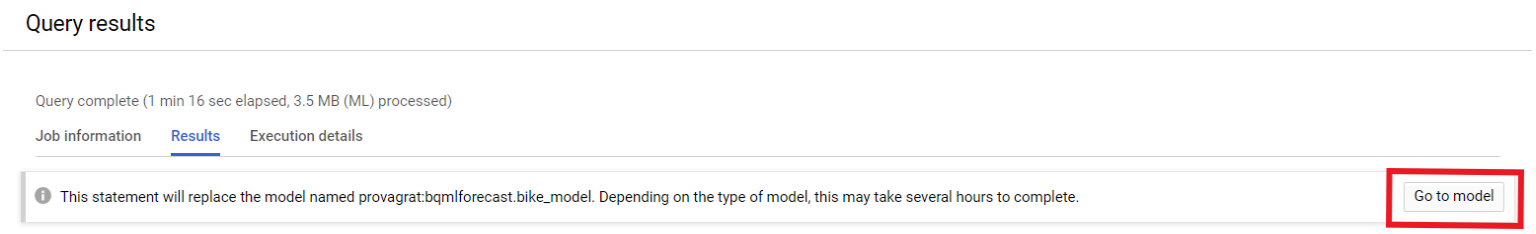
We are using the ARIMA (autoregressive integrated moving average) model. It uses time series data to predict future points in the series (forecasting).

The components of the model creation pipeline are listed below:

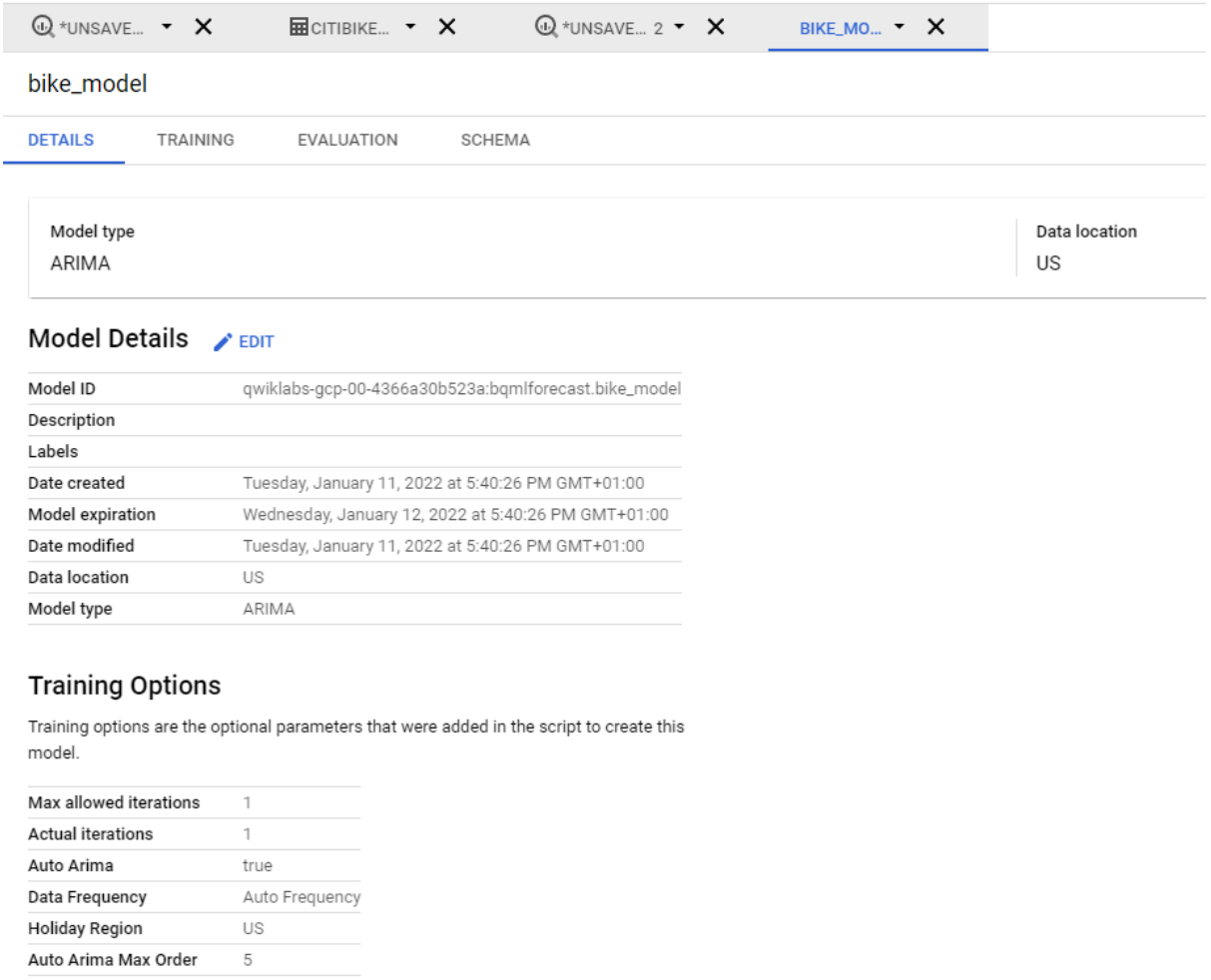
- **Pre-processing**: Adjustments to the input time series (for example: missing values and duplicated timestamp).
- **Holiday effects**: With this enabled spike and dip anomalies that appear during holidays won't be treated as anomalies.
- **Seasonal and trend decomposition**: Seasonality extrapolation using double exponential smoothing (exponential smoothing applied twice).
- **Trend modeling**: Using auto.ARIMA, this means dozens of candidate models are trained and evaluated in parallel.

When a green check mark appears your wait is over! The model has completed training successfully. You can now use it to perform forecasting.

4. Now that the model is trained, click on **Go to model** in the results tab.



This will open up a window that contains Model Details and Training Options used, like below.



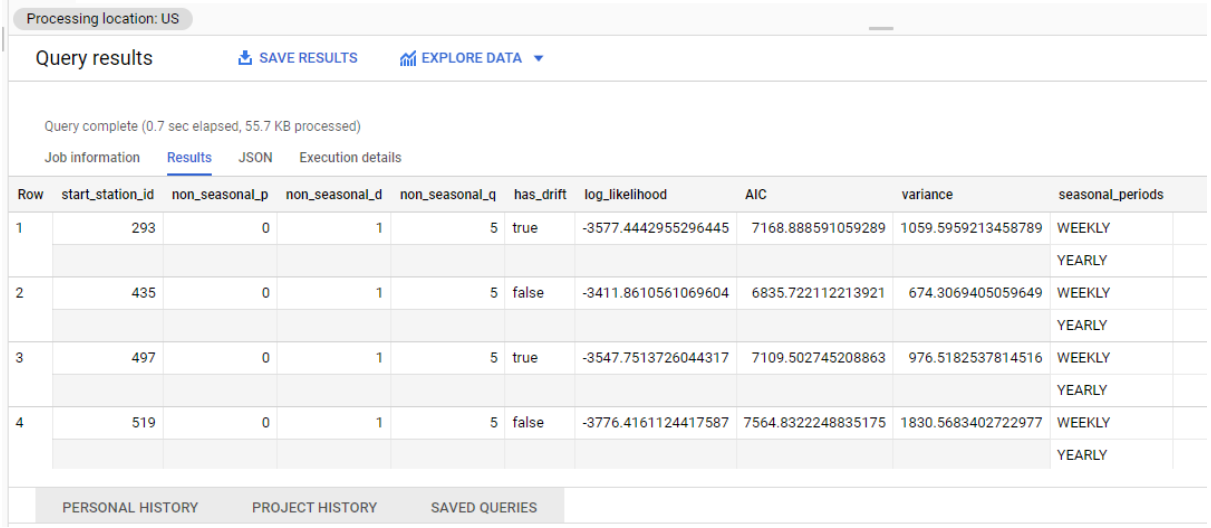
Evaluate the time series model

Now that we have a trained model, we can evaluate it.

1. Let's start by running the following query on the model:

```
SELECT
  *
FROM
  ML.EVALUATE(MODEL bqmlforecast.bike_model)
```

Running the above query should produce a new table containing evaluation metrics for every model provided to the function `ML.EVALUATE`.



Processing location: US

Query results SAVE RESULTS EXPLORE DATA

Query complete (0.7 sec elapsed, 55.7 KB processed)

Job information Results JSON Execution details

Row	start_station_id	non_seasonal_p	non_seasonal_d	non_seasonal_q	has_drift	log_likelihood	AIC	variance	seasonal_periods
1	293	0	1	5	true	-3577.4442955296445	7168.888591059289	1059.5959213458789	WEEKLY
									YEARLY
2	435	0	1	5	false	-3411.8610561069604	6835.722112213921	674.3069405059649	WEEKLY
									YEARLY
3	497	0	1	5	true	-3547.7513726044317	7109.502745208863	976.5182537814516	WEEKLY
									YEARLY
4	519	0	1	5	false	-3776.4161124417587	7564.8322248835175	1830.5683402722977	WEEKLY
									YEARLY

PERSONAL HISTORY PROJECT HISTORY SAVED QUERIES

2. As you can see from the table, we have five models, one for each station in the training data.
   The first four columns define each single model, the remaining ones are relevant for the fitting process.
   During the fitting process, the `auto.ARIMA` algorithm is used to determine the best ARIMA model between the dozens of candidate models trained and evaluated in parallel. This process is repeated for every time series so, in our case, five times.
   The metric that usually gives better results to evaluate which model fits better the data is `AIC`.

Make predictions using the model

At this point we have a trained and evaluated model, the only thing that we have left to do is use it to make some actual predictions.

Setting the `HORIZON`, the `CONFIDENCE_LEVEL` and using the `ML.FORECAST` function, we can now forecast the number of values we want, with the level of confidence we prefer.

1. Copy the following code in the query editor and run it to make some predictions:

```
DECLARE HORIZON STRING DEFAULT "30"; #number of values to forecast
DECLARE CONFIDENCE_LEVEL STRING DEFAULT "0.90";
EXECUTE IMMEDIATE format("""
    SELECT
        *
    FROM
      ML.FORECAST(MODEL bqmlforecast.bike_model,
                  STRUCT(%s AS horizon,
                         %s AS confidence_level)
                 )
    """, HORIZON, CONFIDENCE_LEVEL)
```

Since we set 30 as `HORIZON`, the model will make a prediction for the next whole month (30 days).

2. Press the second **VIEW RESULTS** button.

All results

Elapsed time	Statements processed	Job status
2.5 sec	2	SUCCESS

Job	Stages completed	Bytes processed	Action
5:52 PM [3:20]	1	0 B	VIEW RESULTS
5:52 PM Procedure	3	157.89 KB	VIEW RESULTS

This will open up a table containing 30 days of demand forecasts, each predicted value will also shows the upper and lower bound of the `prediction_interval` (given the `CONFIDENCE_LEVEL`), as seen in the table below:

Query complete (0.7 sec elapsed, 157.9 KB processed)

Job information | Results | JSON | Execution details

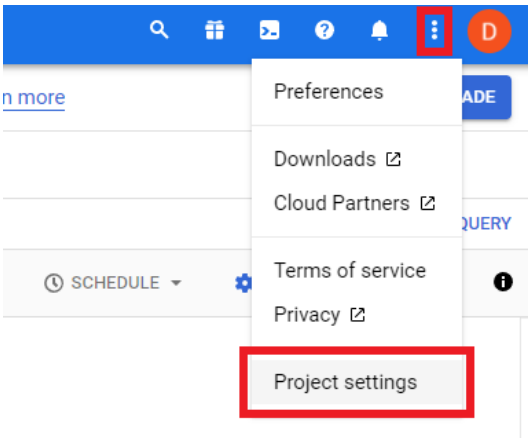
Row	start_station_id	forecast_timestamp	forecast_value	standard_error	confidence_level	prediction_interval_lower_bound	prediction_interval_upper_bound	confidence_interval_lower_bound	confidence_interval_upper_bound
1	293	2016-01-01 00:00:00 UTC	124.98517864991061	32.55146548997371	0.9	71.50080401164561	178.4695532881756	71.50080401164561	178.4695532881756
2	293	2016-01-02 00:00:00 UTC	61.52885757839809	33.10901888997855	0.9	7.128383020272885	115.9293321365233	7.128383020272885	115.9293321365233
3	293	2016-01-03 00:00:00 UTC	37.2061282576972	33.127975577858784	0.9	-17.225493487850315	91.63775000324472	-17.225493487850315	91.63775000324472
4	293	2016-01-04 00:00:00 UTC	182.27608843919887	38.16849280962786	0.9	119.56253812494833	244.9896387534494	119.56253812494833	244.9896387534494
5	293	2016-01-05 00:00:00 UTC	158.50284301197905	44.373314658280904	0.9	85.59432877454438	231.41135724941373	85.59432877454438	231.41135724941373
6	293	2016-01-06 00:00:00 UTC	147.23307339075438	44.40826173236646	0.9	74.26713862325236	220.1990081582564	74.26713862325236	220.1990081582564
7	293	2016-01-07 00:00:00 UTC	165.7168821720452	44.44318132647482	0.9	92.69357202603425	238.74019231805613	92.69357202603425	238.74019231805613
8	293	2016-01-08 00:00:00 UTC	172.35232102447083	44.47807350532931	0.9	99.27168054516433	245.43296150377734	99.27168054516433	245.43296150377734
9	293	2016-01-09 00:00:00 UTC	100.15376295281442	44.51293833339958	0.9	27.015837079497516	173.29168882613132	27.015837079497516	173.29168882613132
10	293	2016-01-10 00:00:00 UTC	101.30248311920477	44.54777587490301	0.9	28.107316685648897	174.49764955276063	28.107316685648897	174.49764955276063

Free the resources

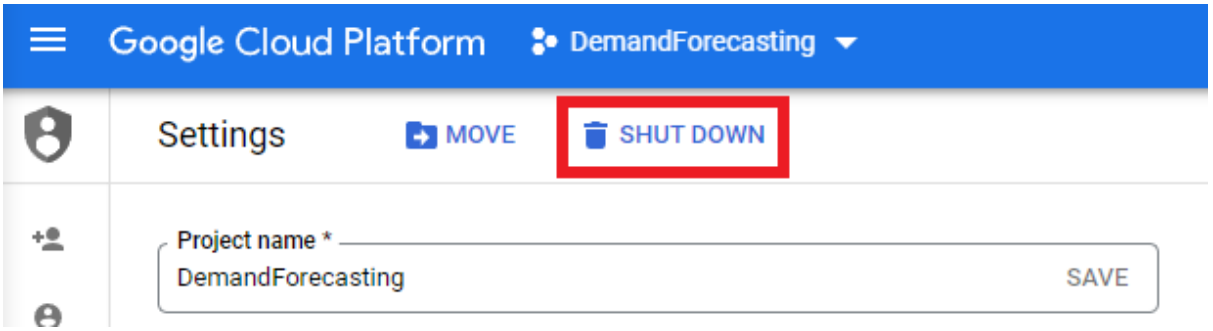
To avoid possible incurring charges we will now free all the resources. There are two ways to do it: delete the project or free all the resources one by one.

Delete the project

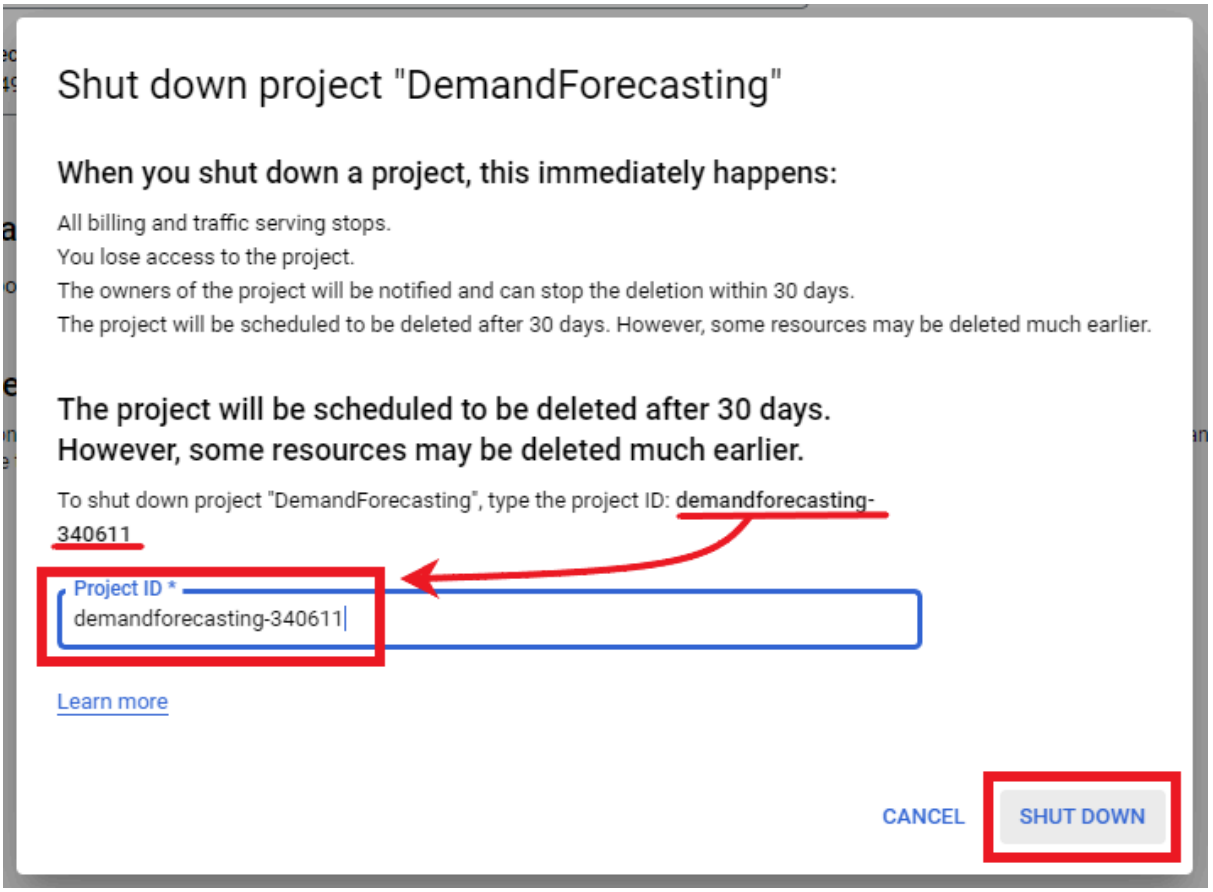
- Click the option button (3 stacked dots) at the top right of the screen, then press the **Project settings** button.



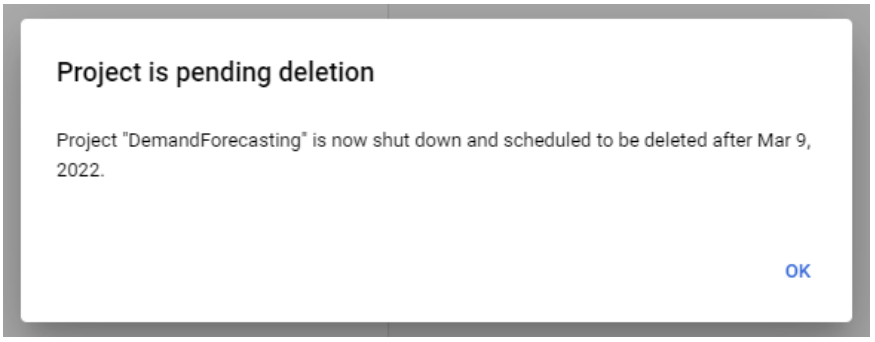
- Click the **SHUT DOWN** button at the top.



- This will open up a new window explaining what happens when you shut down a project and requesting you to insert the Project ID to confirm the elimination. Insert the ID and press the **SHUT DOWN** button.

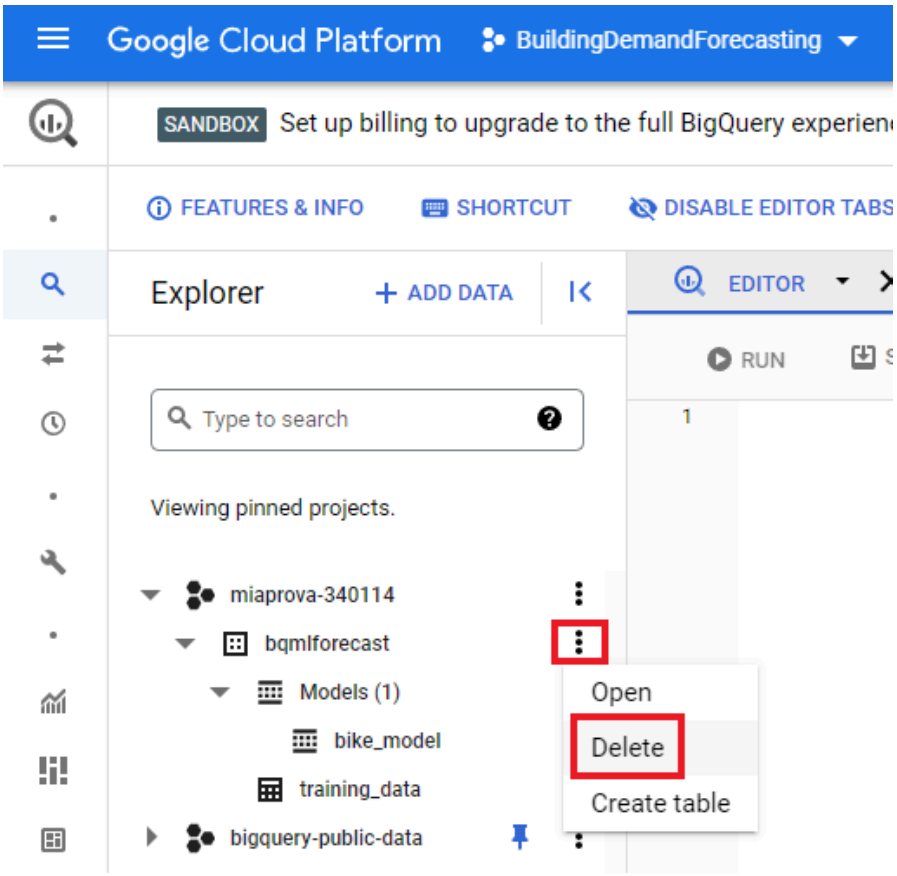


- Now, if everything is gone the right way, a window will open showing you the scheduled time of deletion of the project, like below.

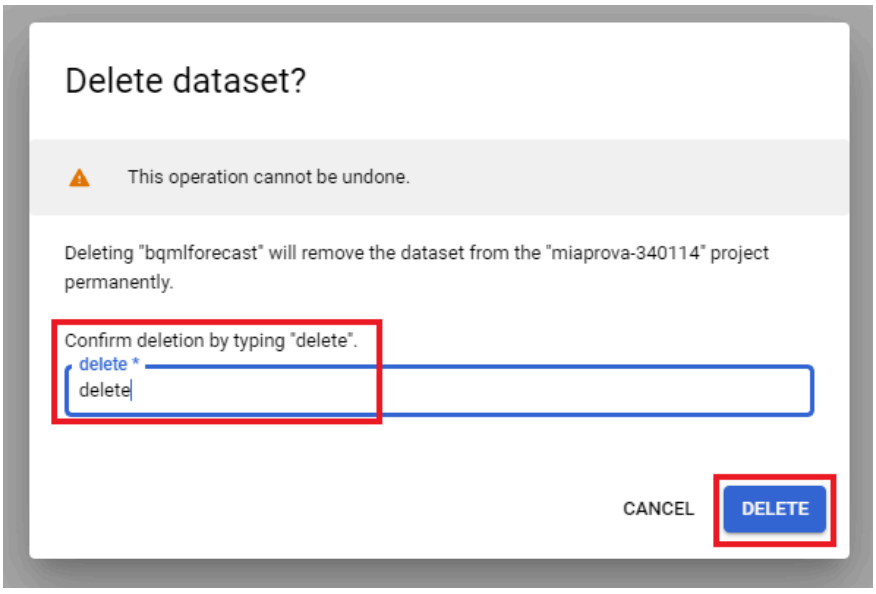


Free the resources

- In the explorer tab click the options button (3 stacked dots) on the right of the dataset name (`bqmlforecast`), then press the **Delete** button.



- This will open up a window asking you to confirm the deletion by typing **delete** and pressing the **DELETE** button, like below:



You have successfully deleted the dataset and the model. Both of them were created specifying an expiration date so, even without doing this step, they would get eliminated the day after their creation.

This concludes this tutorial, I hope it could have been of help to you.