Getting Started with BigQuery ML: A Practical Tutorial for Beginners

Dipan Saha · Follow

6 min read · Jun 7, 2023

Photo by Arseny Togulev on Unsplash

Introduction

BigQuery ML is a powerful tool that brings machine learning capabilities to Google BigQuery, allowing users to build and deploy machine learning models directly within the BigQuery environment, thereby increasing the development speed by eliminating the need for data movement to a different GCP AI/ML service.

In this tutorial, I shall walk you through the steps to get started with BigQuery ML and demonstrate its practical applications.

Prerequisites

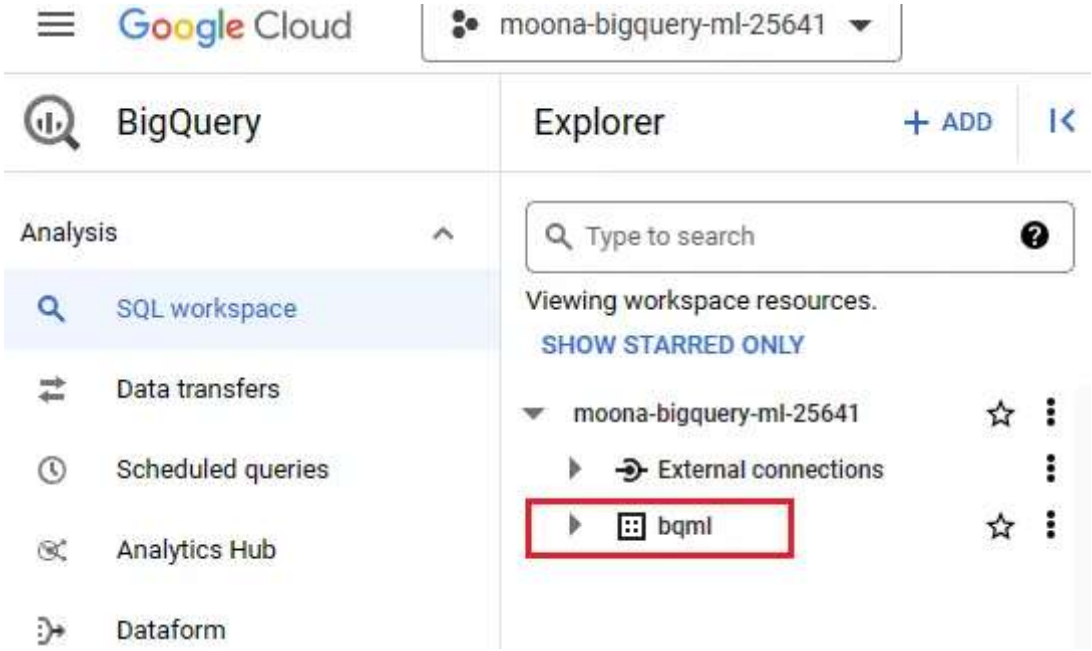
Before diving into BigQuery ML, make sure you have the following prerequisites in place:

1. **Google Cloud Platform (GCP) Account:** Sign up for a GCP account and create a project.
2. **BigQuery API Enabled:** In your GCP project, enable the BigQuery API by navigating to the API Library and searching for “BigQuery API.”

Tutorial

Step 1: Create a BigQuery Dataset

In the BigQuery console, create a dataset to store your data and machine learning models. This can be done by clicking on your project name and selecting “Create dataset.”

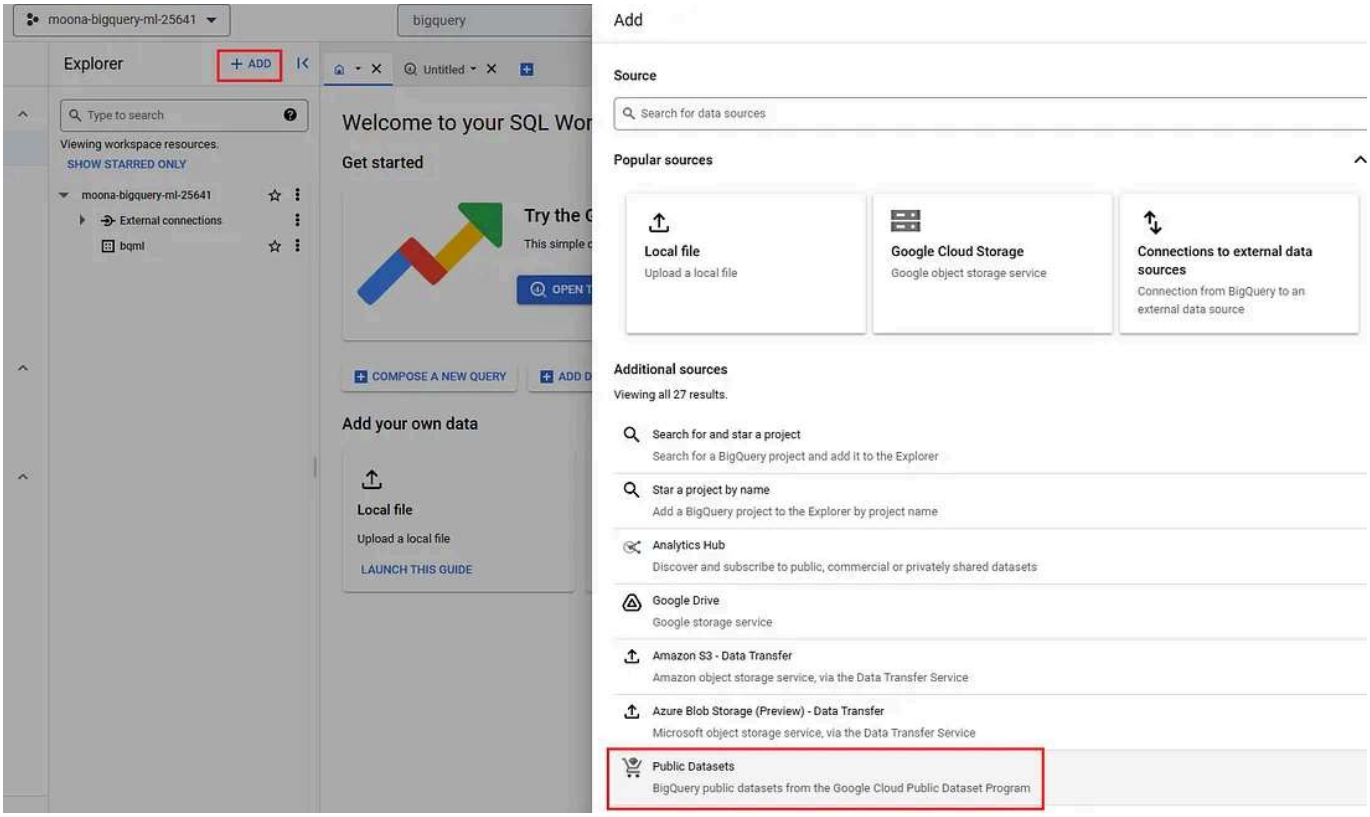


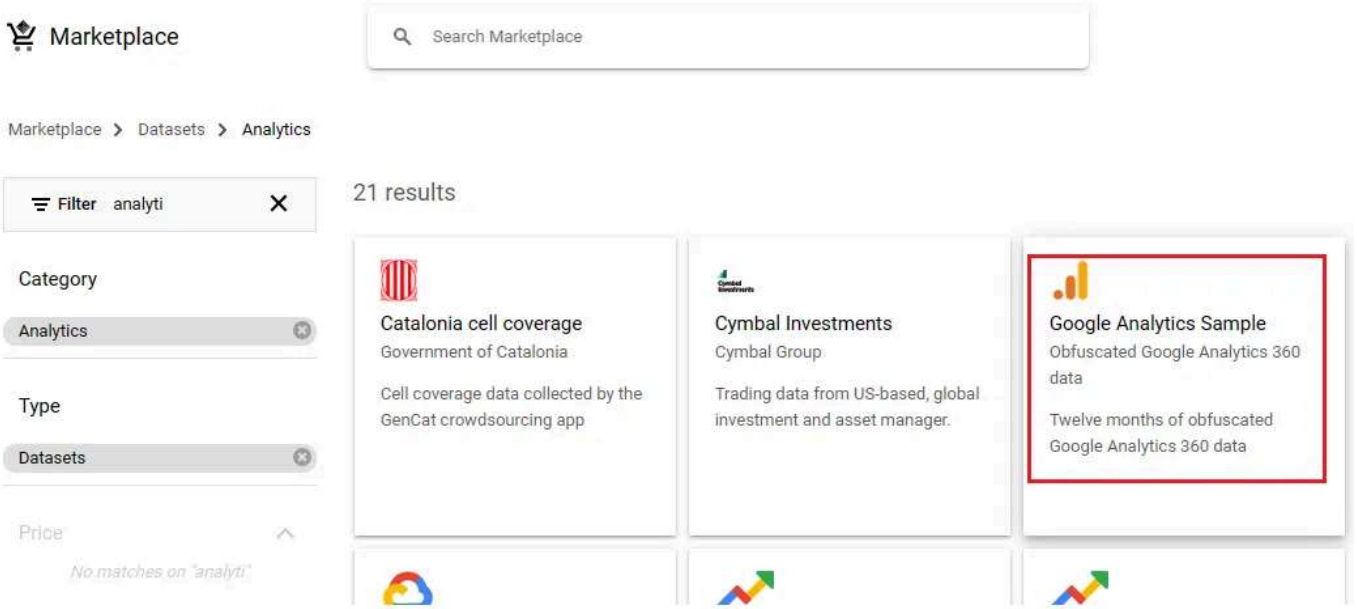
Step 2: Data Exploration and Preparation

Typically, the data exploration and preparation includes 2 steps.

- **Explore Your Dataset:** Use SQL queries in the BigQuery console to gain insights into your data. Understand the structure, relationships, and statistical properties of your dataset.
- **Data Preprocessing:** Perform any necessary preprocessing steps such as handling missing values, encoding categorical variables, or normalizing numerical features. BigQuery's SQL syntax provides various functions to facilitate these operations.

However, for our use case, we shall use a publicly available clean and processed data. Let's add the BigQuery Public Dataset 'Google Analytics Sample' to our project.





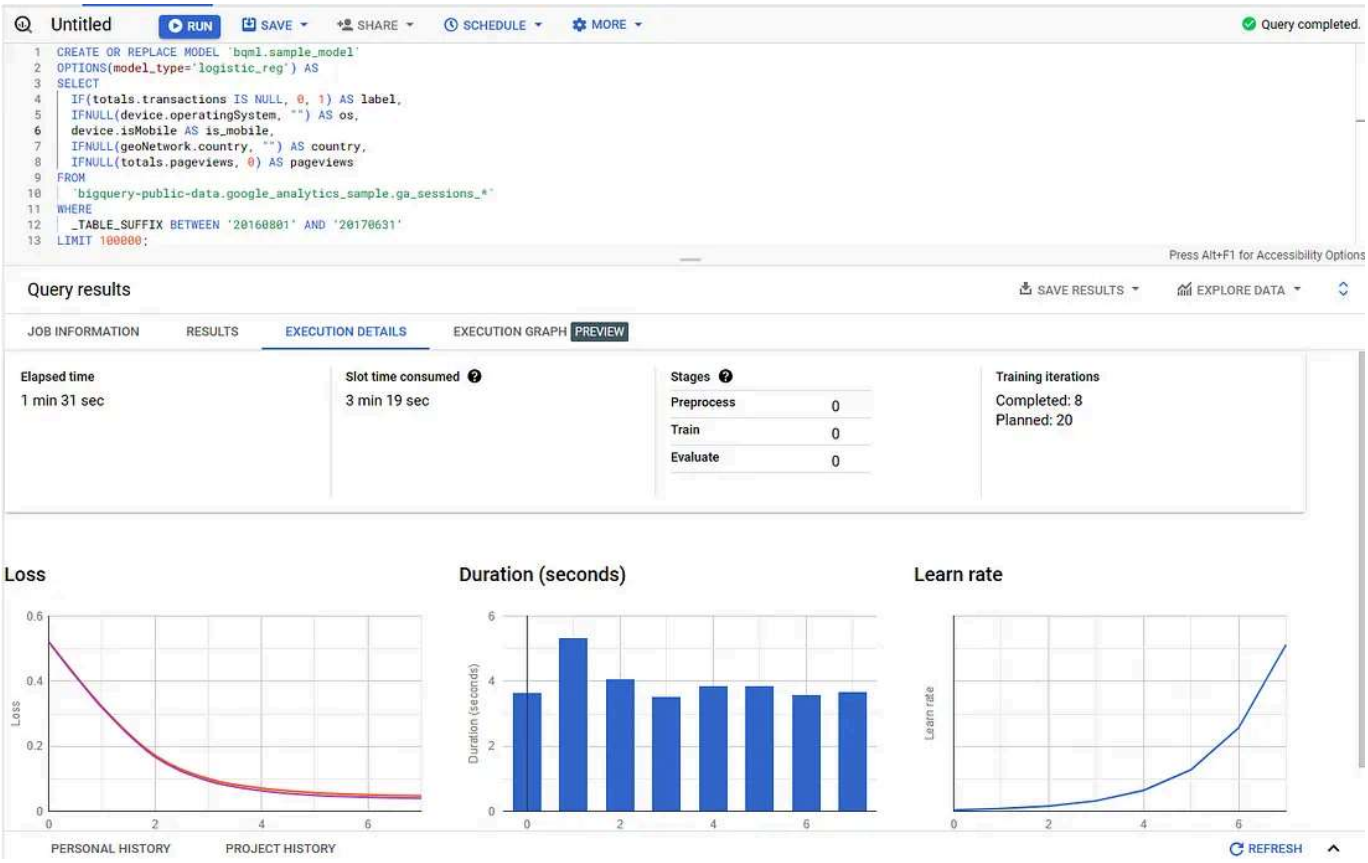
Step 3: Build our first Machine Learning Model

Model Creation: Use the CREATE MODEL statement in BigQuery to define and train your machine learning model. Specify the machine learning algorithm, target variable, and input features.

```
CREATE OR REPLACE MODEL `bqml.sample_model`
OPTIONS(model_type='logistic_reg') AS
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS country,
  IFNULL(totals.pageviews, 0) AS pageviews
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20160801' AND '20170631'
LIMIT 100000;
```

Few things to note:

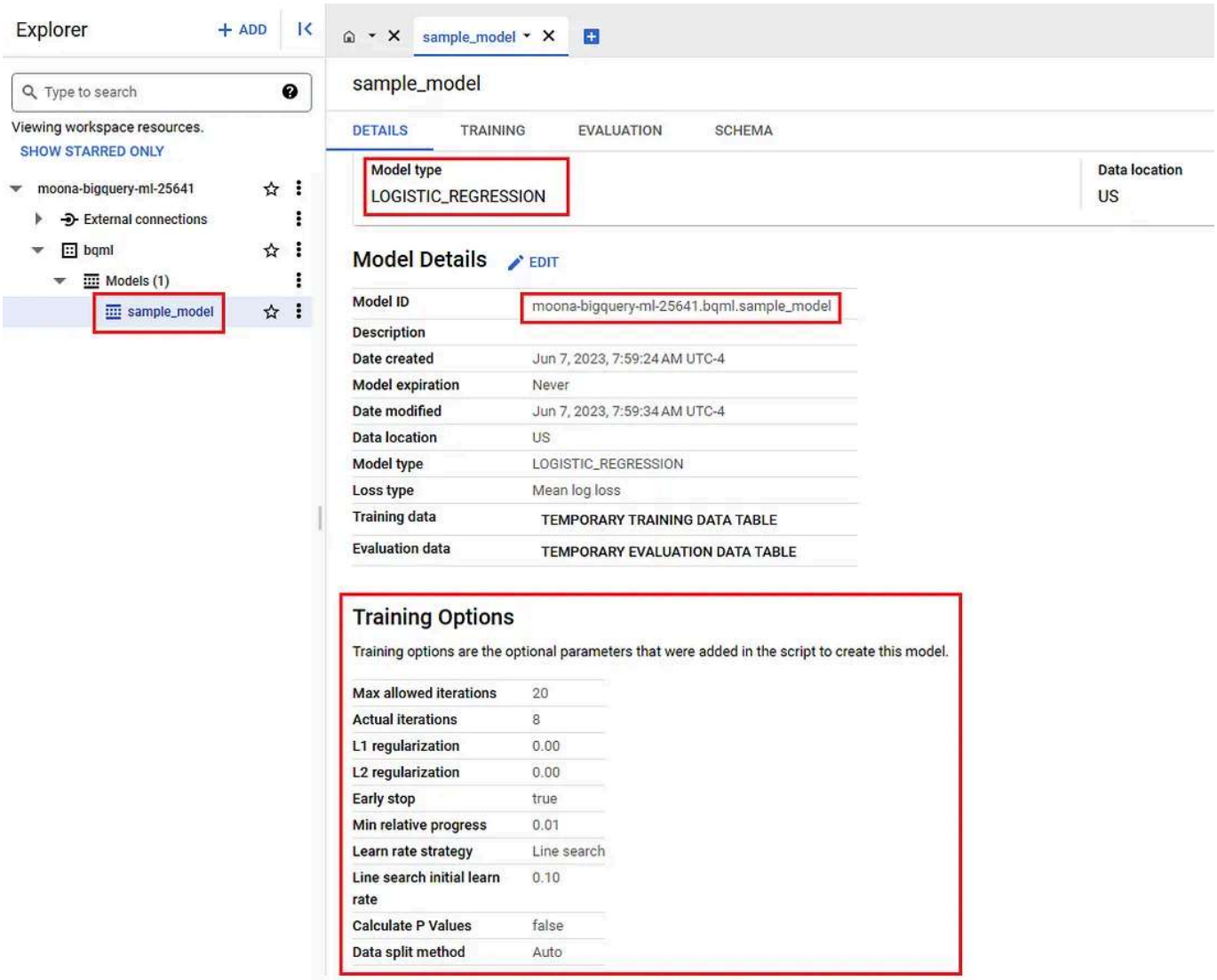
- We have named our model as `sample_model` which will get saved under the `bqml` dataset.
- The model type has been selected as Logistic Regression (`model_type='logistic_reg'`)
- The input data is being generated via a SELECT query from the BigQuery Public Dataset 'Google Analytics Sample'.
- The training data has been limited to those collected from 1 August 2016 to 31 June 2017. We shall use a different portion of the input data for evaluation and test.
- Furthermore, we're limiting to 100,000 data points to save us some processing time.
- As we haven't specified any target variable, BQML will take the `label` field as the default target variable.



Step 4: View Model information & training statistics

Now we can get information about our model by clicking on `sample_model` under `bqml` dataset in the UI.

Under Details, you should find some basic model info.



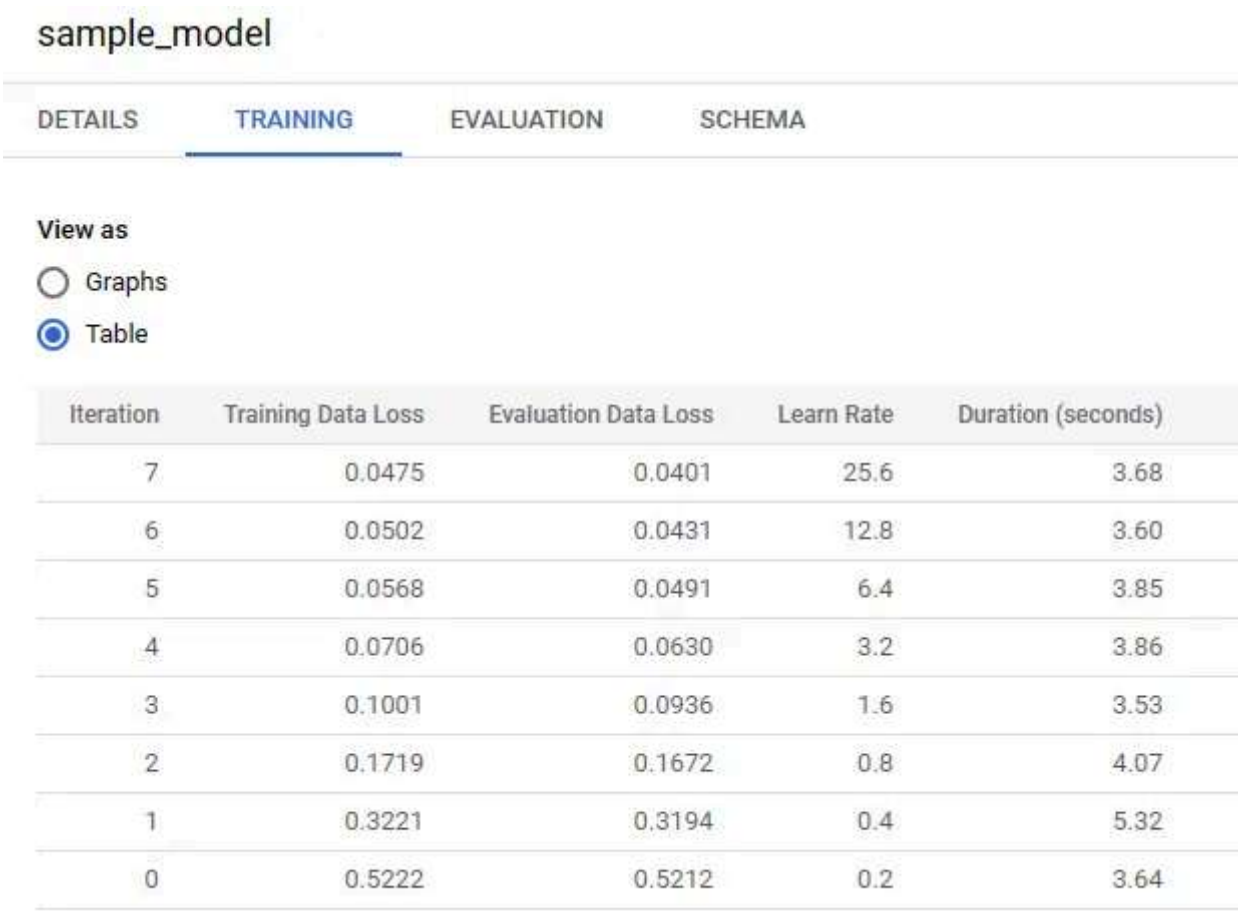
As you can see from above, the `Data split method` has been specified as `Auto`. This means that BigQuery ML will automatically split data for it's validation processes.

Alternatively, you can use the `DATA_SPLIT_METHOD` argument to tell BigQuery ML how you want to split the data. The default split is `AUTO_SPLIT` which is defined as follows:

> *When there are fewer than 500 rows in the input data, all rows are used as training data. When there are between 500 and 50,000 rows in the input data, 20% of the data is used as evaluation data in a RANDOM split. When there are*

> *more than 50,000 rows in the input data, only 10,000 of them are used as evaluation data in a RANDOM split.*

The training information can either be viewed in table format or as graphs under the training tab.



sample_model

DETAILS TRAINING EVALUATION SCHEMA

View as

Graphs

Table

Iteration	Training Data Loss	Evaluation Data Loss	Learn Rate	Duration (seconds)
7	0.0475	0.0401	25.6	3.68
6	0.0502	0.0431	12.8	3.60
5	0.0568	0.0491	6.4	3.85
4	0.0706	0.0630	3.2	3.86
3	0.1001	0.0936	1.6	3.53
2	0.1719	0.1672	0.8	4.07
1	0.3221	0.3194	0.4	5.32
0	0.5222	0.5212	0.2	3.64

The evaluation tab shows different relevant metrices. The most important of those is the `Accuracy` which tells you how accurately your model has predicted the results for the evaluation set.

sample_model

DETAILS TRAINING EVALUATION SCHEMA

Aggregate Metrics

Threshold	0.5000
Precision	0.4737
Recall	0.1667
Accuracy	0.9888
F1 score	0.2466
Log loss	0.0401
ROC AUC	0.9846

Score threshold

Positive class threshold	0.0000
Positive class	1
Negative class	0
Precision	0.0110
Recall	1.0000
Accuracy	0.0110
F1 score	0.0217

Precision-recall by threshold

Precision-recall curve

Area under curve: 0.457

ROC curve

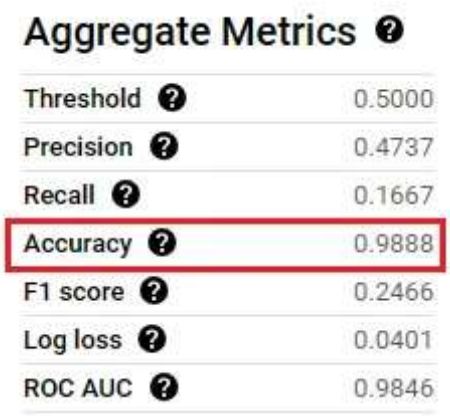
Area under curve: 0.985

Confusion matrix

Item counts

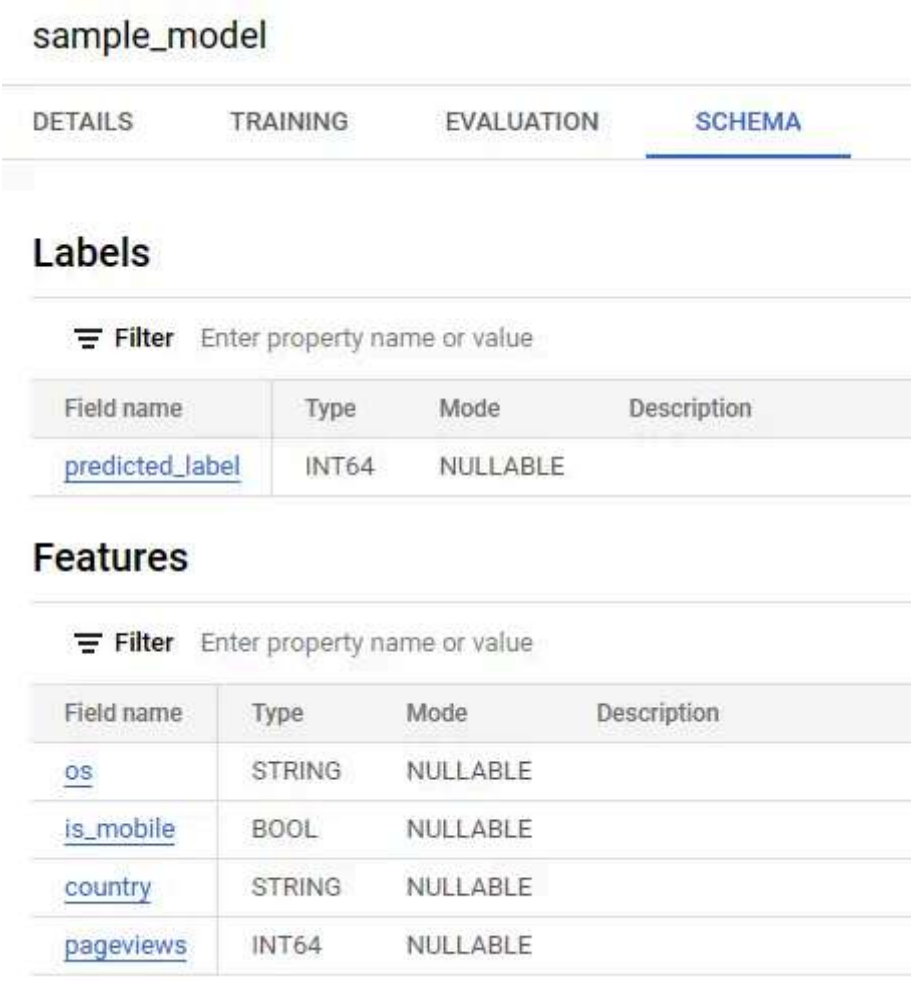
This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label \ Predicted label	1	0
1	100%	0%
0	100%	0%



Under the Schema tab you will notice that BQML has selected the `label` field as target variable. This field is generally named as `predicted_<target variable name>` for logistic regression models.

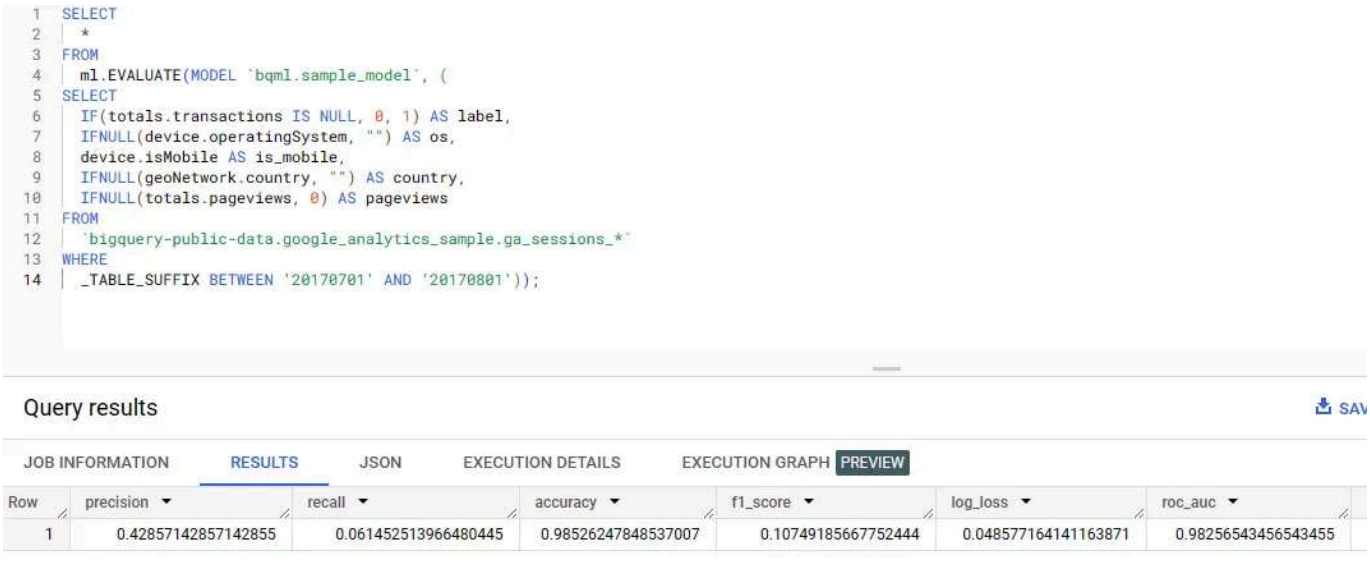
The remaining fields will be marked as features.



Step 5: Evaluate the model by using a separate set of input data

We can also evaluate our model ourselves by using a separate set of input data [`_TABLE_SUFFIX BETWEEN '20170701' AND '20170801'`]

```
SELECT
  *
FROM
  ml.EVALUATE(MODEL `bqml.sample_model`, (
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS country,
  IFNULL(totals.pageviews, 0) AS pageviews
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'));
```



As you can see, our model has predicted the results on this new input set with 98%+ accuracy.

Step 7: Model Selection and Tuning

At this point, if we prefer, we can try different machine learning algorithms or hyperparameter configurations to improve our model performance. This process should ideally be iterated until you find the optimal model for your task.

However, if we are fine with our model performance, we can skip this step too.

Step 8: Make Predictions with the Model

We can use the `ML.PREDICT` function in SQL queries to generate predictions from our deployed model. Just pass the input data to the model and retrieve the predicted outcomes.

Here, we are using a SELECT query to pass the input test set [`_TABLE_SUFFIX BETWEEN '20170701' AND '20170801'`] to our model which will provide the outcome through the `predicted_label` field. You can then standard SQL functions to see the total predicted purchases (which is nothing but the sun of the `predicted_label` field) group by countries.

```
SELECT
  country,
  SUM(predicted_label) as total_predicted_purchases
FROM
  ml.PREDICT(MODEL `bqml.sample_model`, (
SELECT
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(geoNetwork.country, "") AS country
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
GROUP BY country
```

```
ORDER BY total_predicted_purchases DESC
LIMIT 10;
```

```
SELECT
  country,
  SUM(predicted_label) as total_predicted_purchases
FROM
  ml.PREDICT(MODEL `bqml.sample_model`, (
SELECT
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(geoNetwork.country, "") AS country
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
GROUP BY country
ORDER BY total_predicted_purchases DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	country	total_predicted_purcl
1	United States	128
2	Taiwan	6
3	Canada	4
4	India	2
5	Turkey	2
6	Japan	2
7	Thailand	1
8	Brazil	1
9	United Kingdom	1
10	Australia	1

Another example: Predict purchases per user

This time we try to predict the number of transactions each visitor makes, sort the results and select the top 10 visitors by transactions.

```
SELECT
  fullVisitorId,
  SUM(predicted_label) as total_predicted_purchases
FROM
  ml.PREDICT(MODEL `bqml.sample_model`, (
SELECT
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(geoNetwork.country, "") AS country,
```



```
GROUP BY fullVisitorId
ORDER BY total_predicted_purchases DESC
LIMIT 10;
```

> *Note that we didn't use the `fullVisitorId` field while training our model as this has no impact on the model predictions. The true features which impact the model performance were os, is_mobile, pafeviews and country.*

Hence selecting the features is an integral part of the model creation process.

Congratulations!

As a part of this exercise, we have created a binary logistic regression model, evaluated the model and used the model to make predictions.

Hope this helped you on your machine learning journey!

Bigquery Ml Bigquery Machine Learning