This tutorial introduces data analysts to BigQuery ML. BigQuery ML enables users to create and execute machine learning models in BigQuery using SQL queries. This tutorial introduces feature engineering by using the `TRANSFORM` clause. Using the `TRANSFORM` clause, you can specify all preprocessing (/bigquery-ml/docs/reference/standard-sql/bigqueryml-preprocessing-functions) during model creation. The preprocessing is automatically applied during the prediction and evaluation phases of machine learning.

In this tutorial, you use the `natality` sample table (https://console.cloud.google.com/bigquery?p=bigquery-public-data&d=samples&t=natality&page=table) to create a model that predicts the birth weight of a child based on the baby's gender, the length of the pregnancy, and bucketized demographic information about the mother. The `natality` sample table contains information about every birth in the United States over a 40 year period.

In this tutorial, you use:

- BigQuery ML to create a linear regression model using the `CREATE MODEL` statement with the `TRANSFORM` clause
- The `ML.FEATURE_CROSS` and `ML.QUANTILE_BUCKETIZE` preprocessing functions
- The `ML.EVALUATE` function to evaluate the ML model
- The `ML.PREDICT` function to make predictions using the ML model

This tutorial uses billable components of Google Cloud, including:

- BigQuery
- BigQuery ML

For more information about BigQuery costs, see the BigQuery pricing (/bigquery/pricing) page.

1. Sign in (https://accounts.google.com/Login) to your Google Account.

   If you don't already have one, sign up for a new account (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

Go to the project selector page (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (/billing/docs/how-to/modify-project).

4. BigQuery is automatically enabled in new projects. To activate BigQuery in a pre-existing project, go to Enable the BigQuery API.

   Enable the API (https://console.cloud.google.com/flows/enableapi?apiid=bigquery)

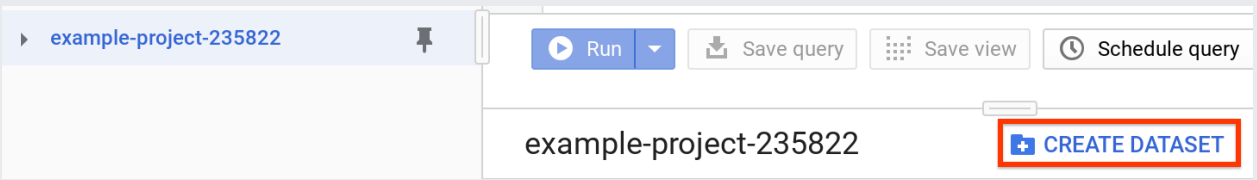
The first step is to create a BigQuery dataset to store your ML model. To create your dataset:

1. In the Google Cloud Console, go to the BigQuery web UI.

   Go to the BigQuery web UI (https://console.cloud.google.com/bigquery)

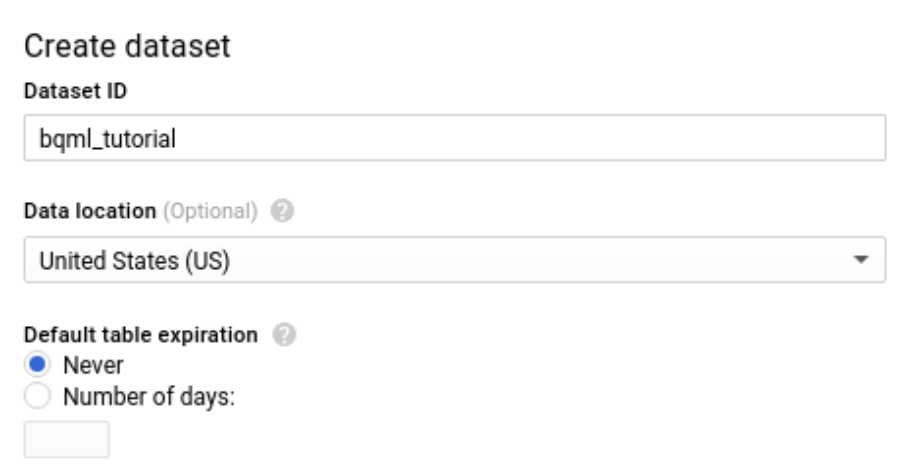
2. In the navigation panel, under the **Resources** section, click your project name.

3. On the right side, in the details panel, click **Create dataset**.



4. On the **Create dataset** page:

- For **Dataset ID**, enter `bqml_tutorial`.
- For **Data location**, choose **United States (US)**. Currently, the public datasets are stored in the `US` multiregional location (/bigquery/docs/locations). For simplicity, place your dataset in the same location.



5. Leave all of the other default settings in place and click **Create dataset**.

Next, create a linear regression model using the natality sample table for BigQuery. The following standard SQL query is used to create the model you use to predict the birth weight of a child.

In addition to creating the model, running the `CREATE MODEL` command trains the model you create.

The `CREATE MODEL` (/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create) clause is used to create and train the model named `bqml_tutorial.natality_model`.

The `OPTIONS(model_type='linear_reg', input_label_cols=['weight_pounds'])` clause indicates that you are creating a linear regression (https://en.wikipedia.org/wiki/Linear_regression) model. A linear regression is a type of regression model that generates a continuous value from a linear combination of input features. The `weight_pounds` column is the input label column. For linear regression models, the label column must be real valued (that is, the column values must be real numbers).

This query's `TRANSFORM` clause uses the following columns from the `SELECT` statement:

- `weight_pounds`: The weight, in pounds, of the child (FLOAT64).
- `is_male`: The sex of the child. TRUE if the child is male, FALSE if female (BOOL).
- `gestation_weeks`: The number of weeks of the pregnancy (INT64).
- `mother_age`: The age of the mother when giving birth (INT64).
- `mother_race`: The race of the mother (INT64). This integer value is the same as the `child_race` value in the table schema. To force BigQuery ML to treat `mother_race` as a non-numeric feature, with each distinct value representing a different category, the query casts `mother_race` to a STRING. This is important because race is more likely to have more meaning as a category than an integer, which has ordering and scale.

Through the `TRANSFORM` clause, the original features are preprocessed to feed in training. The generated columns are:

- `weight_pounds`: Passed as is, without any change.
- `is_male`: Passed through to feed in training.

- `gestation_weeks`: Passed through to feed in training.
- `bucketized_mother_age`: Generated from `mother_age` by bucketizing `mother_age` based on quantiles using the `ML.QUANTILE_BUCKETIZE()` analytic function.
- `mother_race`: String format of the original `mother_race`.
- `is_male_mother_race`: Generated from crossing `is_male` and `mother_race` using the `ML.FEATURE_CROSS` function.

The query's `SELECT` statement provides the columns that you can use in the `TRANSFORM` clause. However, you do not need to use all columns in the `TRANSFORM` clause. As a result, you can do both feature selection and preprocessing inside the `TRANSFORM` clause.

The `FROM` clause—`bigquery-public-data.samples.natality`—indicates that you are querying the natality sample table in the samples dataset. This dataset is in the `bigquery-public-data` project.

The `WHERE` clause—`WHERE weight_pounds IS NOT NULL AND RAND() < 0.001`—excludes rows where weight is NULL and uses the `RAND` function to draw a random sample of the data.

To run the `CREATE MODEL` query to create and train your model:

1. In the upper right of the BigQuery web UI, click the **Compose new query** button.
2. Enter the following standard SQL query in the **Query editor** text area.

3. Click **Run**.

   The query takes about 30 seconds to complete, after which your model (`natality_model`) appears in the navigation panel of the BigQuery web UI. Because the query uses a `CREATE MODEL` statement to create a table, you do not see query results.

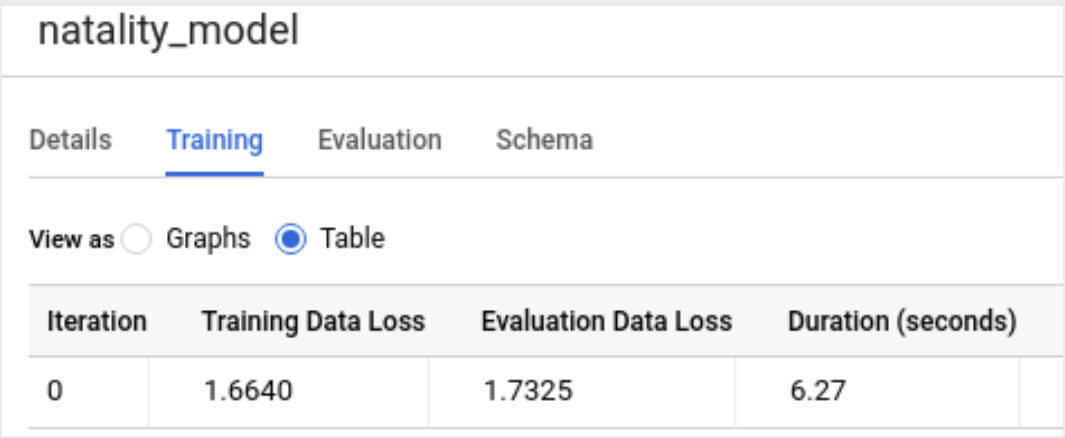
To see the results of the model training, you can use the `ML.TRAINING_INFO` (/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-train) function, or you can view the statistics in the BigQuery web UI. In this tutorial, you use the BigQuery web UI.

A machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss. This process is called *empirical risk minimization*.

Loss is the penalty for a bad prediction—a number indicating how bad the model's prediction was on a single example. If the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples.

To see the model training statistics that were generated when you ran the `CREATE MODEL` query:

1. In the BigQuery web UI, in the **Resources** section, expand ***project-name* > bqml_tutorial** and then click **natality_model**.
2. Click the **Training** tab and for **View as**, select the **Table** option. The results should look like the following:



The **Training Data Loss** column represents the loss metric calculated after the model is trained on the training dataset. Because you performed a linear regression, this column is the mean squared error (https://developers.google.com/machine-learning/glossary/#MSE).

The **Evaluation Data Loss** column is the same loss metric calculated on the holdout dataset (data that is held back from training to validate the model). The default optimize strategy used for the training is "normal_equation", so only one iteration is required to converge to the final model.

For more information about the `optimize_strategy` option, see the CREATE MODEL statement (/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create).

For more information about the `ML.TRAINING_INFO` function and the "optimize_strategy" training option, see the BigQuery ML syntax reference (/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-train).

After creating your model, you evaluate the performance of the classifier by using the `ML.EVALUATE` function. The `ML.EVALUATE` function evaluates the predicted values against the actual data.

The query used to evaluate the model is as follows:

The upper `SELECT` statement retrieves the columns from your model.

The `FROM` clause uses the `ML.EVALUATE` (/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-evaluate) function against your model: `bqml_tutorial.natality_model`.

This query's nested `SELECT` statement and `FROM` clause are the same as those in the `CREATE MODEL` query. Because the `TRANSFORM` clause is used in training, you don't need to specify the specific columns and transformations. They are automatically restored.

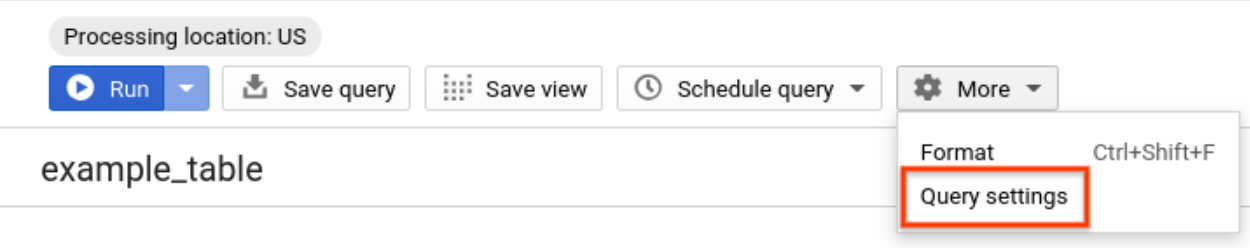
The `WHERE` clause—`WHERE weight_pounds IS NOT NULL`—excludes rows where weight is NULL.

Note: You can also call the `ML.EVALUATE` function without providing the input data. The function uses the evaluation metrics calculated during training:

To run the `ML.EVALUATE` query that evaluates the model, complete the following steps:

1. In the BigQuery web UI, click the **Compose new query** button.
2. Enter the following standard SQL query in the **Query editor** text area.

3. (Optional) To set the processing location, on the **More** drop-down list, click **Query settings**. For **Processing location**, choose **United States (US)**. This step is optional because the processing location is automatically detected based on the dataset's location.



4. Click **Run**.

5. When the query is complete, click the **Results** tab below the query text area. The results should look like the following:

Query complete (5.502 sec elapsed, 4.12 GB processed)

Job information | Results | JSON | Execution details

Row	mean_absolute_error	mean_squared_error	mean_squared_log_error	median_absolute_error	r2_score	explained_variance
1	0.9566580179970666	1.6756289722442677	0.034241471462096516	0.7385590721661188	0.04650972930257946	0.046516832131241026

Because you performed a linear regression, the results include the following columns:

- `mean_absolute_error`
- `mean_squared_error`
- `mean_squared_log_error`
- `median_absolute_error`

- `r2_score`
- `explained_variance`

An important metric in the evaluation results is the R^2 score (https://en.wikipedia.org/wiki/Coefficient_of_determination). The R^2 score is a statistical measure that determines whether the linear regression predictions approximate the actual data. A `0` value indicates that the model explains none of the variability of the response data around the mean. A `1` value indicates that the model explains all the variability of the response data around the mean.

Now that you have evaluated your model, the next step is to use it to predict an outcome. You can use your model to predict the birth weight of all babies born in Wyoming.

The query used to predict the outcome is as follows:

The top-most `SELECT` statement retrieves the `predicted_weight_pounds` column. This column is generated by the `ML.PREDICT` function. When you use the `ML.PREDICT` function, the output column name for the model is `predicted_`***`label_column_name`***. For linear regression models, `predicted_label` is the estimated value of `label`. For logistic regression models, `predicted_label` is one of the two input labels, depending on which label has the higher predicted probability.

The `ML.PREDICT` (/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-predict) function is used to predict results using your model: `bqml_tutorial.natality_model`.

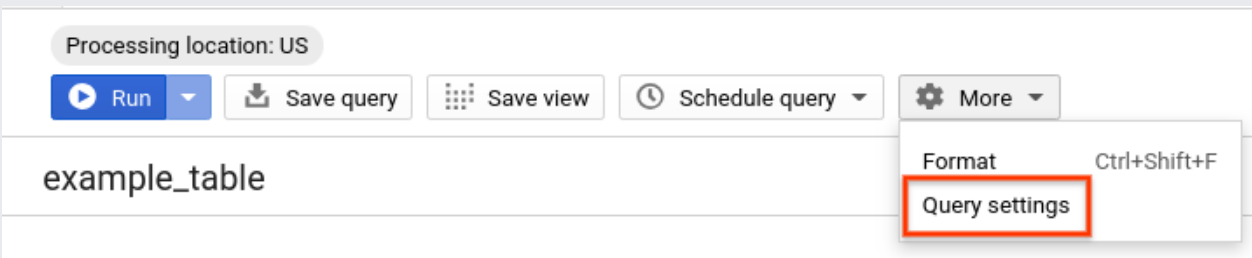
This query's nested `SELECT` statement and `FROM` clause are the same as those in the `CREATE MODEL` query. Note that you don't necessarily need to pass in all columns as in training, and only the ones used in the `TRANSFORM` clause are required. Similar to `ML.EVALUATE`, the transformations inside the `TRANSFORM` are automatically restored.

The `WHERE` clause—`WHERE state = "WY"`—indicates that you are limiting the prediction to the state of Wyoming.

To run the query that uses the model to predict an outcome:

1. In the BigQuery web UI, click the **Compose new query** button.
2. Enter the following standard SQL query in the **Query editor** text area.

3. (Optional) To set the processing location, on the **More** drop-down list, click **Query settings**. For **Processing location**, choose **United States (US)**. This step is optional because the processing location is automatically detected based on the dataset's location.



4. Click **Run**.
5. When the query is complete, click the **Results** tab below the query text area. The results should look like the following:

Job information | **Results** | JSON

Row	predicted_weight_pounds
1	7.735962399307027
2	7.728855793480761
3	7.383850250400428
4	7.4132677633242565
5	7.734971309702814
6	7.828010317909502
7	7.675314172840444
8	7.454863195482176
9	7.35046837905611
10	7.667216477406328

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial:

- You can delete the project you created.
- Or you can keep the project and delete the dataset.

Deleting your project removes all datasets and all tables in the project. If you prefer to reuse the project, you can delete the dataset you created in this tutorial:

1. If necessary, open the BigQuery web UI.

   Go to the BigQuery web UI (https://console.cloud.google.com/bigquery)

2. In the navigation panel, click the **bqml_tutorial** dataset you created.
3. On the right side of the window, click **Delete dataset**. This action deletes the dataset, the table, and all the data.
4. In the **Delete dataset** dialog box, confirm the delete command by typing the name of your dataset (`bqml_tutorial`) and then click **Delete**.

To delete the project:

Caution: Deleting a project has the following effects:

- **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
- **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such as an `appspot.com` URL, delete selected resources inside the project instead of deleting the whole project.

If you plan to explore multiple tutorials and quickstarts, reusing projects can help you avoid exceeding project quota limits.

1. In the Cloud Console, go to the **Manage resources** page.

   Go to the Manage resources page (https://console.cloud.google.com/iam-admin/projects)

2. In the project list, select the project you want to delete and click **Delete** .
3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

- To learn more about machine learning, see the Machine learning crash course (https://developers.google.com/machine-learning/crash-course/).
- For an overview of BigQuery ML, see Introduction to BigQuery ML (/bigquery-ml/docs/bigqueryml-intro).
- To learn more about the BigQuery web UI, see Using the BigQuery web UI (/bigquery/bigquery-web-ui).