**Objective**

Return an aggregated result with a machine-learning (ML) prediction using Google BigQuery ML and serve out results using Google App Engine.

*Introduction to BigQuery by Pol Ferrando: toward data science*

**Step 1:** **Create a GCP project**

Once logged in to the Google Cloud console, you will create a new project by naming it and making sure that it has a billing profile attached to it. Once created, make sure that you select that project so that you are working under it, you should see a page that prompts you that “You’re working in <name of project>”.

**Step 2: Activate Cloud shell**

Click on the cloud shell icon on the right hand side of the page otherwise you can search for it on the search bar. If you were able to open it up via the cloud shell icon in the project page it should automatically be configured to that project on the shell. However, if it’s not, you will need to configure it through the following commands.

| **gcloud** projects list |
| --- |

| **gcloud** config set project <name **of** project> |
| --- |

**Step 3: Load Data into BigQuery**

Click on BigQuery through the access page of your project and import your CSV data to that project. If you are following Dr. Ostrowski demo you can use the google analytics sample data which can be found through the search bar and clicking “View dataset”. This will then take you to the Query editor. If you face problems trying to run a query on the “Editor” tab, open up another Editor tab → “Query 2”.

**Step 4: Create Model**

We can use the CREATE MODEL function to create our model. Create a model based on the feature sets (attributes) you are interested in analyzing. The only options we need to specify are the model type (linear\_regression) and the target variable (homes\_sold).

We want to use [linear regression](https://onlinecourses.science.psu.edu/stat501/node/250/) because we are predicting a continuous quantity (i.e homes sold.) If we instead wanted to predict a category, we would use [logistic regression](https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc).

Additionally, since we are using a linear regression model, it can not have infinite or NaN values

Copy and paste the CREATE MODEL SQL query statement:

| # Provide name of model  CREATE OR REPLACE MODEL `real-estate-365520.redfin\_dataset.linearreg\_model`  # Specify options  OPTIONS  (model\_type='linear\_reg',  input\_label\_cols=['homes\_sold']) AS  # Provide training data  SELECT  inventory,  months\_of\_supply,  CAST(city as STRING) as city, #city is a category, not a #.  pending\_sales,  new\_listings,  sold\_above\_list,  homes\_sold  FROM  #project-id.dataset.table  `real-estate-365520.redfin\_dataset.city\_market`  WHERE  homes\_sold IS NOT NULL # Filter for rows containing data we want to predict. |
| --- |

**Step 5: Evaluate Model**

We can see how well our model performed by using the ML.EVALUATE function. The function takes a model name, and a table. The table should have the same schema as the table used to create the model.

Copy and paste the EVALUATE MODEL SQL query statement:

| SELECT \* FROM  ML.EVALUATE(  MODEL `real-estate-365520.redfin\_dataset.linearreg\_model`, # Model Name  # Table to evaluate against  (  SELECT  inventory,  months\_of\_supply,  CAST(city as STRING) as city, #city is a category, not a #.  pending\_sales,  new\_listings,  sold\_above\_list,  homes\_sold  FROM  #project-id.dataset.table  `real-estate-365520.redfin\_dataset.city\_market`  WHERE  homes\_sold IS NOT NULL  )  ) |
| --- |

**Here’s how to interpret the columns.**

* mean\_absolute\_error The average distance from the predicted value to the actual value. Lower is better.
* mean\_squared\_error Used for [evaluating statistical significance](https://en.wikipedia.org/wiki/Mean_squared_error#Regression). Lower is better.
* mean\_squared\_log\_error Used as a numerically stable cost function by [Gradient Descent](https://hackernoon.com/gradient-descent-aynk-7cbe95a778da) for training the model. Lower is better.
* median\_absolute\_error A measure more [robust to outliers](https://en.wikipedia.org/wiki/Median_absolute_deviation). Lower is better.
* r2\_score [Coefficient of determination](https://en.wikipedia.org/wiki/Coefficient_of_determination). Higher is better.
* explained\_variance The fraction of variance explained. Higher is better.

**Step 6: Perform Prediction with Model**

Making a prediction with the model is as easy as calling ML.PREDICT. Ensure the table you’re calling ML.PREDICT has a compatible schema.

Copy and paste the PREDICT MODEL SQL query statement:

| SELECT \* FROM  ML.PREDICT(MODEL `real-estate-365520.redfin\_dataset.linearreg\_model`,  (  SELECT  inventory,  months\_of\_supply,  CAST(city as STRING) as city, #city is a category, not a #.  pending\_sales,  new\_listings,  sold\_above\_list,  homes\_sold  FROM  `real-estate-365520.redfin\_dataset.city\_market`  WHERE  homes\_sold IS NOT NULL)  ) |
| --- |

Reference: (Model SQL code used as a template) <https://medium.com/eliiza-ai/introduction-to-linear-regression-with-bigquery-ml-b62d134e0c18>