BigQuery ML

David Han

February 2021

1 Introduction

In our world, there is so much data out there that can be put to use for improving society, such as weather data, traffic data, and etc. However, what use is it if we are not able to quickly manipulate the tons of data that we have? This is where machine learning comes in. Machine learning is a powerful tool that can analyze and break down data when conventional techniques cannot. Recent computer technologies have increased the possibility of data collection, storage, and processing. BigQuery ML enables data scientists and data analysts to build and operationalize ML models on planet-scale structured or semi-structured data, directly inside BigQuery, using simple SQL—in a fraction of the time.

2 What is Big Query ML?

Machine learning on large datasets requires extensive programming and knowledge of ML frameworks. These requirements restrict solution development to a very small set of people within each company, and they exclude data analysts who understand the data but have limited machine learning knowledge and programming expertise.

BigQuery ML empowers data analysts to use machine learning through existing SQL tools and skills. Analysts can use BigQuery ML to build and evaluate ML models in BigQuery. Analysts don't need to export small amounts of data to spreadsheets or other applications. It also eliminates the need to move data, increasing development speed.

3 Different Models in BigQuery ML

3.1 What is a Model?

The representation of what a machine learning system has learned from the training data.

3.2 Supported Models

- **Linear regression** for forecasting; for example, the sales of an item on a given day. Labels are real-valued (they cannot be +/- infinity or NaN).
- **Binary logistic regression** for classification; for example, determining whether a customer will make a purchase. Labels must only have two possible values.
- **Multiclass** logistic regression for classification. These models can be used to predict multiple possible values such as whether an input is "low-value," "medium-value," or "high-value." Labels can have up to 50 unique values.
- **K-means clustering** for data segmentation; for example, identifying customer segments. K-means is an unsupervised learning technique, so model training does not require labels nor split data for training or evaluation.
- **Matrix Factorization** for creating product recommendation systems. You can create product recommendations using historical customer behavior, transactions, and product ratings and then use those recommendations for personalized customer experiences.
- **Time series** for performing time-series forecasts. You can use this feature to create millions of time series models and use them for forecasting. The model automatically handles anomalies, seasonality, and holidays.
- **Deep Neural Network (DNN)** for creating TensorFlow based Deep Neural Networks for classification and regression models.
- **AutoML Tables** to create best-in-class models without feature engineering or model selection. AutoML Tables searches through a variety of model architectures to decide the best model.
- **TensorFlow model** importing. This feature lets you create BigQuery ML models from previously trained TensorFlow models, then perform prediction in BigQuery ML.

4 Benefits

The primary benefit of using BigQuery ML is being able to use machine learning tools in relation to a cloud-based data warehouse. This service allows an ML model to be created only using SQL, without necessarily using another programming language, such as Python or Java.

Oftentimes, it can take lots of time to export data for usage in a machine learning model. With BigQuery ML, there is no need to export or reformat data. Some of the disadvantages of needing to export data are:

- Increases complexity because multiple tools are required.

- Reduces speed because moving and formatting large amounts data for Python-based ML frameworks takes longer than model training in BigQuery.
- Requires multiple steps to export data from the warehouse, restricting the ability to experiment on your data.

5 Working With Models

This section will provide a brief overview and flow of how to work with BigQuery ML models. There are more tutorials on the google cloud website, and will cover significantly more in depth.

5.1 Listing Models

There are three different ways you can list BigQuery ML models in a dataset:

- Using the Cloud Console.
- Using the bq ls command in the bq command-line tool.
- Calling the API method directly or by using the client libraries.

Before listing models, you must have permissions from the bigquery.models.list permissions. The following roles gives these permissions:

- bigquery.dataViewer
- bigquery.dataEditor
- bigquery.dataOwner
- bigquery.metadataViewer
- bigquery.user
- bigquery.admin

For example, you can use the following code to list models in a dataset:

```
from google.cloud import bigquery

client = bigquery.Client()

# TODO(developer): Set dataset_id to the ID of the dataset that contains
#                  the models you are listing.
# dataset_id = 'your-project.your_dataset'

models = client.list_models(dataset_id)  # Make an API request.
```

```
print("Models contained in '{}':".format(dataset_id))
for model in models:
    full_model_id = "{}.{}.{}".format(
        model.project, model.dataset_id, model.model_id
    )
    friendly_name = model.friendly_name
    print("{}: friendly_name='{}'".format(full_model_id, friendly_name))
```

5.2 Getting Model Metadata

In the previous section, the three different methods for listing models hold the same for obtaining model metadata. The same permissions are also required.

```
from google.cloud import bigquery

client = bigquery.Client()

model_id = 'your-project.your_dataset.your_model'
model = client.get_model(model_id)

full_model_id = "{}.{}.{}".format(model.project, model.dataset_id, model.model_id)
friendly_name = model.friendly_name
print(
    "Got model '{}' with friendly_name '{}'.".format(full_model_id, friendly_name)
)
```

5.3 Managing Models

This section will show you how to copy different models. You can copy model using the bq command-line (recommended) or using the API.

For example, you can use the following command to copy a model:

```
bq --location=US cp mydataset.mymodel mydataset2.mymodel
```

The location tag refers to the name of your location, but is optional. The first model is the model you are copying and the second model is the model in the destination dataset.

6 Tutorials

There are several tutorials on the website for using BigQuery ML, such as creating a k-means clustering model or making recommendations from movie ratings. Visit this link to try them: https://cloud.google.com/bigquery-ml/docs/tutorials