Explore NYC taxi cab data

Question: How many trips did Yellow taxis take each month in 2015?
Click the **Compose Query** button and add the following to the New Query field:

```
#standardSQL
SELECT
  TIMESTAMP_TRUNC(pickup_datetime,
    MONTH) month,
  COUNT(*) trips
FROM
  `bigquery-public-data.new_york.tlc_yellow_trips_2015`
GROUP BY
  1
ORDER BY
  1
```

Then click **Run Query**.
The result:

month	trips
2015-01-01 00:00:00 UTC	12748986
2015-02-01 00:00:00 UTC	12450521
2015-03-01 00:00:00 UTC	13351609
2015-04-01 00:00:00 UTC	13071789
2015-05-01 00:00:00 UTC	13158262
2015-06-01 00:00:00 UTC	12324935
2015-07-01 00:00:00 UTC	11562783
2015-08-01 00:00:00 UTC	11130304
2015-09-01 00:00:00 UTC	11225063
2015-10-01 00:00:00 UTC	12315488
2015-11-01 00:00:00 UTC	11312676
2015-12-01 00:00:00 UTC	11460573

Question: What was the average speed of Yellow taxi trips in 2015?
Replace the previous query with the following, and then **Run Query**:

```
#standardSQL
SELECT
  EXTRACT(HOUR
  FROM
    pickup_datetime) hour,
  ROUND(AVG(trip_distance / TIMESTAMP_DIFF(dropoff_datetime,
        pickup_datetime,
        SECOND))*3600, 1) speed
FROM
```

```
  `bigquery-public-data.new_york.tlc_yellow_trips_2015`
WHERE
  trip_distance > 0
  AND fare_amount/trip_distance BETWEEN 2
  AND 10
  AND dropoff_datetime > pickup_datetime
GROUP BY
  1
ORDER BY
  1
```

The result:

hour	speed
0	15.8
1	16.3
2	16.8
3	17.5
4	20.0
5	21.6
6	17.6
7	13.7
8	11.6
9	11.4
10	11.5
11	11.3
12	11.2
13	11.3
14	11.2
15	11.0
16	11.5
17	11.2
18	11.1
19	11.8
20	12.9

During the day, the average speed is around 11-12 MPH; but at 5:00 AM the average speed almost doubles to 21 MPH. Intuitively this makes sense since there is likely less traffic on the road at 5:00 AM.

Identify an objective

You will now create a Machine Learning model in BigQuery to predict the price of a cab ride in New York city given the historical dataset of trips and trip data. Predicting the fare before the ride could be very useful for trip planning for both the rider and the taxi agency.

Select features and create your training dataset

The New York City Yellow Cab dataset is a public dataset provided by the city and has been loaded into BigQuery for your exploration. Browse the complete list of fields here and then preview the dataset to find useful features that will help a machine learning model understand the relationship between data about historical cab rides and the price of the fare.
Your team decides to test whether these below fields are good inputs to your fare forecasting model:

- Tolls Amount
- Fare Amount
- Hour of Day
- Pick up address
- Drop off address
- Number of passengers

Replace the query with the following:

```
#standardSQL
WITH params AS (
    SELECT
    1 AS TRAIN,
    2 AS EVAL
    ),

  daynames AS
    (SELECT ['Sun', 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat'] AS daysofweek),
```

```
taxitrips AS (
SELECT
  (tolls_amount + fare_amount) AS total_fare,
  daysofweek[ORDINAL(EXTRACT(DAYOFWEEK FROM pickup_datetime))] AS dayofweek,
  EXTRACT(HOUR FROM pickup_datetime) AS hourofday,
  pickup_longitude AS pickuplon,
  pickup_latitude AS pickuplat,
  dropoff_longitude AS dropofflon,
  dropoff_latitude AS dropofflat,
  passenger_count AS passengers
FROM
  `nyc-tlc.yellow.trips`, daynames, params
WHERE
  trip_distance > 0 AND fare_amount > 0
  AND MOD(ABS(FARM_FINGERPRINT(CAST(pickup_datetime AS STRING))),1000) = params.TRAIN
)

SELECT *
FROM taxitrips
```

Note a few things about the query:

1. The main part of the query is at the bottom: (SELECT * from taxitrips).
2. taxitrips does the bulk of the extraction for the NYC dataset, with the SELECT containing your training features and label.
3. The WHERE removes data that you don't want to train on.
4. The WHERE also includes a sampling clause to pick up only 1/1000th of the data.
5. We define a variable called TRAIN so that you can quickly build an independent EVAL set.

Then **Run Query**.
Sample Results:

Row	total_fare	dayofweek	hourofday	pickuplon	pickup
1	14.9	Fri	0	-73.980763	40.730
2	22.1	Fri	0	-74.003837	40.750
3	15.3	Fri	0	-73.983574	40.730
4	7.5	Fri	0	-73.978727	40.74

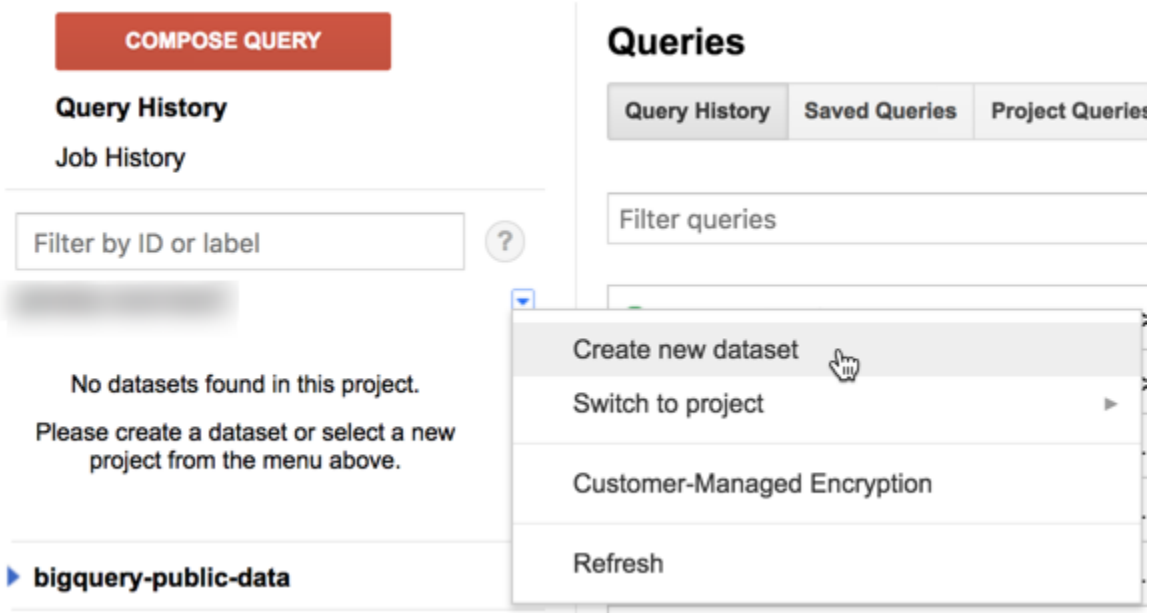
What is the label (correct answer)?

total_fare is the label (what we will be predicting). You created this field out of **tolls_amount** and **fare_amount**, so you could ignore customer tips as part of the model as they are discretionary.

Create a BigQuery dataset to store models

Next, create a new BigQuery dataset which will also store your ML models.

1. In the left pane, click the down arrow icon ( ) next to your project name, and then click **Create new dataset**.



2. In the **Create Dataset** dialog:

- For **Dataset ID**, type **taxi**.
- Leave the other values at their defaults.

3. Click **OK**.

Select a BQML model type and specify options

Now that you have your initial features selected, you are now ready to create your first ML model in BigQuery.

There are the two model types to choose from:

Model	Model Type	Label Data type	Example
Forecasting	linear_reg	Numeric value (typically an integer or floating point)	Forecast sales figures for next year given historical sales data.
Classification	logistic_reg	0 or 1 for binary classification	Classify an email as spam or not spam given the context.

Note: There are many additional model types used in Machine Learning (like Neural Networks and decision trees) and available using libraries like TensorFlow. At this time, BQML supports the two listed above.
Which model type should you choose? Since you are predicting a numeric value (cab fare) you want to use **linear regression**.
Enter the following query to create a model and specify model options, replacing `-- paste the previous training dataset query here` with the training dataset query you created earlier (omitting the `#standardSQL` line):

```
CREATE or REPLACE MODEL taxi.taxifare_model
OPTIONS
  (model_type='linear_reg', labels=['total_fare']) AS
-- paste the previous training dataset query here
```

Next, click **Run Query** to train your model.
Wait for the model to train (5 - 10 minutes).

After your model is trained, you will see the message "This was a CREATE operation. Results will not be shown" which indicates that your model has been successfully trained.

Look inside your taxi dataset and confirm __taxifae_model __now appears.

Next, you will evaluate the performance of the model against new unseen evaluation data.

Evaluate classification model performance

Select your performance criteria

For linear regression models you want to use a loss metric like Root Mean Squared Error. You want to keep training and improving the model until it has the lowest RMSE.

In BQML, **mean_squared_error** is simply a queryable field when evaluating your trained ML model. Simply add a SQRT() to get RMSE.
Now that training is complete, you can evaluate how well the model performs with this query using ML.EVALUATE:

```
#standardSQL
SELECT
  SQRT(mean_squared_error) AS rmse
FROM
  ML.EVALUATE(MODEL taxi.taxifare_model,
  (

  WITH params AS (
    SELECT
    1 AS TRAIN,
    2 AS EVAL
    ),

  daynames AS
    (SELECT ['Sun', 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat'] AS daysofweek),

  taxitrips AS (
  SELECT
    (tolls_amount + fare_amount) AS total_fare,
```

```
  daysofweek[ORDINAL(EXTRACT(DAYOFWEEK FROM pickup_datetime))] AS dayofweek,
  EXTRACT(HOUR FROM pickup_datetime) AS hourofday,
  pickup_longitude AS pickuplon,
  pickup_latitude AS pickuplat,
  dropoff_longitude AS dropofflon,
  dropoff_latitude AS dropofflat,
  passenger_count AS passengers
FROM
  `nyc-tlc.yellow.trips`, daynames, params
WHERE
  trip_distance > 0 AND fare_amount > 0
  AND MOD(ABS(FARM_FINGERPRINT(CAST(pickup_datetime AS STRING))),1000) = params.EVAL
)

SELECT *
FROM taxitrips

))
```

You are now evaluating the model against a different set of taxi cab trips with your params.EVAL filter.
After the model runs, review your model results (your model RMSE value will vary slightly).

Row	rmse
1	9.477056435999074

After evaluating your model you get a **RMSE** of $9.47. Knowing whether or not this loss metric is acceptable to productionalize your model is entirely dependent on your benchmark criteria, which is set before model training begins. Benchmarking is establishing a minimum level of model performance and accuracy that is acceptable. How you can continue to improve the model performance through feature engineering will be discussed at the end of this lab.

Predict taxi fare amount

Next you will write a query to use your new model to make predictions:

```
#standardSQL
```

```
SELECT
*
FROM
  ml.PREDICT(MODEL `taxi.taxifare_model`,
   (

 WITH params AS (
    SELECT
    1 AS TRAIN,
    2 AS EVAL
    ),

  daynames AS
    (SELECT ['Sun', 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat'] AS daysofweek),

  taxitrips AS (
  SELECT
    (tolls_amount + fare_amount) AS total_fare,
    daysofweek[ORDINAL(EXTRACT(DAYOFWEEK FROM pickup_datetime))] AS dayofweek,
    EXTRACT(HOUR FROM pickup_datetime) AS hourofday,
    pickup_longitude AS pickuplon,
    pickup_latitude AS pickuplat,
    dropoff_longitude AS dropofflon,
    dropoff_latitude AS dropofflat,
    passenger_count AS passengers
  FROM
    `nyc-tlc.yellow.trips`, daynames, params
  WHERE
    trip_distance > 0 AND fare_amount > 0
    AND MOD(ABS(FARM_FINGERPRINT(CAST(pickup_datetime AS STRING))),1000) = params.EVAL
  )

  SELECT *
  FROM taxitrips

));
```

Now you will see the model's predictions for taxi fares alongside the actual fares and other features for those rides.

Improve model performance

- Looking to build in more sophisticated features into your model? Continue reading this blog post which covers how you can leverage geospatial functions in BigQuery for more accurate ML models

Additional information

Tip: add `warm_start = true` to your model options if you are retraining new data on an existing model for faster training times. Note that you cannot change the feature columns (this would necessitate a new model).

Other datasets to explore

You can use this below link to bring in the **bigquery-public-data** project if you want to explore modeling on other datasets like forecasting fares for Chicago taxi trips:

- https://bigquery.cloud.google.com/table/bigquery-public-data:chicao_tax_trips.taxitrips

Congratulations!

You've successfully built a ML model in BigQuery to forecast taxi cab fare for New York City cabs.

This self-paced lab is part of the Qwiklabs Data Engineering Quest. A Quest is a series of related labs that form a learning path. Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in this Quest and get immediate completion credit if you've taken this lab. See other available Qwiklabs Quests.

Take your Next Lab

Continue your quest with Working with Google Cloud Dataprep, or one of these:

- Building an IoT Analytics Pipeline on Google Cloud Platform
- Analyzing Natality Data Using Datalab and BigQuery

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.
Manual Last Updated August 29, 2018

Lab Last Tested July 25, 2018