

Predict Visitor Purchases with a Classification Model in BigQuery ML

1 hour 15 minutes

No cost

[Rate Lab](#)

GSP229



Google Cloud Self-Paced Labs

Overview

BigQuery is Google's fully managed, NoOps, low cost analytics database. With BigQuery you can query terabytes and terabytes of data without having any infrastructure to manage or needing a database administrator. BigQuery uses SQL and can take advantage of the pay-as-you-go model. BigQuery allows you to focus on analyzing data to find meaningful insights.

BigQuery ML is a feature in BigQuery that data analysts can use to create, train, evaluate, and predict with machine learning models with minimal coding.

In this lab, you use a special ecommerce dataset that has millions of Google Analytics records for the Google Merchandise Store loaded into BigQuery. You use this data to create a classification (logistic regression) model in BigQuery ML that predicts customers' purchasing habits.

What you'll learn

In this lab, you learn how to perform the following tasks:

- Use BigQuery to find public datasets
- Query and explore the ecommerce dataset
- Create a training and evaluation dataset to be used for batch prediction
- Create a classification (logistic regression) model in BigQuery ML
- Evaluate and improve the performance of your machine learning model
- Predict and rank the probability that a visitor will make a purchase

Setup and requirements

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).

Note: Use an Incognito or private browser window to run this lab. This prevents any conflicts between your personal account and the Student account, which may cause extra charges incurred to your personal account.

- Time to complete the lab---remember, once you start, you cannot pause a lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab to avoid extra charges to your account.

How to start your lab and sign in to the Google Cloud console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is the **Lab Details** panel with the following:

- The **Open Google Cloud console** button
- Time remaining
- The temporary credentials that you must use for this lab
- Other information, if needed, to step through this lab

2. Click **Open Google Cloud console** (or right-click and select **Open Link in Incognito Window** if you are running the Chrome browser).

The lab spins up resources, and then opens another tab that shows the **Sign in** page.

Tip: Arrange the tabs in separate windows, side-by-side.

Note: If you see the **Choose an account** dialog, click **Use Another Account**.

3. If necessary, copy the **Username** below and paste it into the **Sign in** dialog.

"**Username**"

content_c

You can also find the **Username** in the **Lab Details** panel.

4. Click **Next**.

5. Copy the **Password** below and paste it into the **Welcome** dialog.

"Password"

content_c

You can also find the **Password** in the **Lab Details** panel.

6. Click **Next**.

Important: You must use the credentials the lab provides you. Do not use your Google Cloud account credentials.

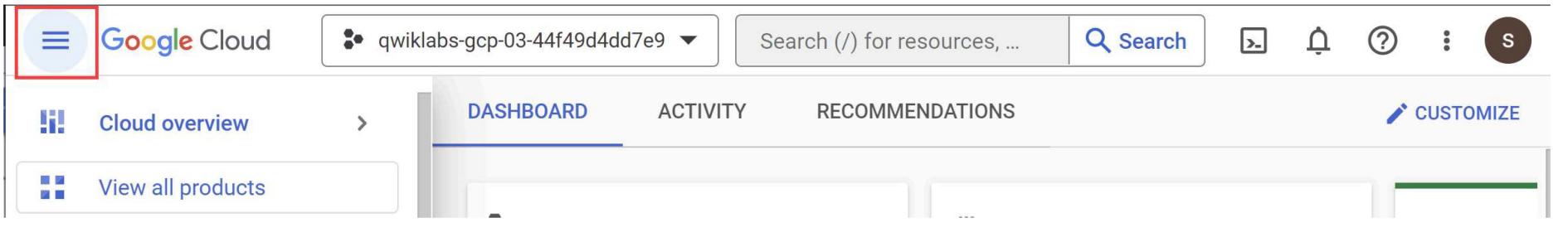
Note: Using your own Google Cloud account for this lab may incur extra charges.

7. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Google Cloud console opens in this tab.

Note: To view a menu with a list of Google Cloud products and services, click the **Navigation menu** at the top-left.



Open the BigQuery console

1. In the Google Cloud Console, select **Navigation menu** > **BigQuery**.

The **Welcome to BigQuery in the Cloud Console** message box opens. This message box provides a link to the quickstart guide and the release notes.

2. Click **Done**.

The BigQuery console opens.

Access the course dataset

1. In the **Explorer pane**, click **+ ADD**.

The **Add data** pane opens.

2. Click **Star a project by name** under Additional sources.

3. Enter **data-to-insights** and click **Star**.

Explorer

+ ADD

K

Type to search

?

Viewing workspace resources.

SHOW STARRED ONLY

▶ qwiklabs-gcp-01-439197ede932 ★ ⋮

▶ data-to-insights ★ ⋮

Click on the below direct link to view the public **data-to-insights** project:

- https://console.cloud.google.com/bigquery?p=data-to-insights&d=ecommerce&t=web_analytics&page=table

The field definitions for the **data-to-insights** ecommerce dataset are here. Keep the link open in a new tab for reference.

Click on the **Query** tab, then select **In new tab** to open the Query Editor.

Task 1. Explore ecommerce data

Scenario: Your data analyst team exported the Google Analytics logs for an ecommerce website into BigQuery and created a new table of all the raw ecommerce visitor session data for you to explore. Using this data, you'll try to answer a few questions.

Question: Out of the total visitors who visited our website, what % made a purchase?

1. Copy and paste the following query into the BigQuery Editor:

```
#standardSQL
WITH visitors AS(
SELECT
COUNT(DISTINCT fullVisitorId) AS total_visitors
FROM `data-to-insights.ecommerce.web_analytics`
),
purchasers AS(
SELECT
COUNT(DISTINCT fullVisitorId) AS total_purchasers
FROM `data-to-insights.ecommerce.web_analytics`
WHERE totals.transactions IS NOT NULL
)
SELECT
total_visitors,
total_purchasers,
total_purchasers / total_visitors AS conversion_rate
FROM visitors, purchasers
```

content_c

2. Click Run.

The result: 2.69%

Question: What are the top 5 selling products?

1. Clear the previous query, and then add the following query in the Editor:

```
SELECT
  p.v2ProductName,
  p.v2ProductCategory,
  SUM(p.productQuantity) AS units_sold,
  ROUND(SUM(p.localProductRevenue/1000000),2) AS revenue
FROM `data-to-insights.ecommerce.web_analytics`,
UNNEST(hits) AS h,
UNNEST(h.product) AS p
GROUP BY 1, 2
ORDER BY revenue DESC
LIMIT 5;
```

content_c

2. Click Run

The result:

Row	v2ProductName	v2ProductCategory	units_sold	revenue
1	Nest® Learning Thermostat 3rd Gen-USA - Stainless Steel	Nest-USA	17651	870976.95
2	Nest® Cam Outdoor Security Camera - USA	Nest-USA	16930	684034.55
3	Nest® Cam Indoor Security Camera - USA	Nest-USA	14155	548104.47
4	Nest® Protect Smoke + CO White Wired Alarm-USA	Nest-USA	6394	178937.6
5	Nest® Protect Smoke + CO White Battery Alarm-	Nest-USA	6340	178572.4

Question: How many visitors bought on subsequent visits to the website?

1. Clear the previous query, and then add the following query in the **Editor**:

```
# visitors who bought on a return visit (could have bought on first as well
WITH all_visitor_stats AS (
SELECT
    fullvisitorid, # 741,721 unique visitors
    IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) >
0, 1, 0) AS will_buy_on_return_visit
    FROM `data-to-insights.ecommerce.web_analytics`
    GROUP BY fullvisitorid
)

SELECT
    COUNT(DISTINCT fullvisitorid) AS total_visitors,
    will_buy_on_return_visit
FROM all_visitor_stats
GROUP BY will_buy_on_return_visit
```

content_c

2. Click **Run**.

The results:

Row	total_visitors	will_buy_on_return_visit
1	729848	0
2	11873	1

Analyzing the results, you can see that $(11873 / 741721) = 1.6\%$ of total visitors will return and purchase from the website. This includes the subset of visitors who bought on their very first session and then came back and bought again.

Question: What are some of the reasons a typical ecommerce customer will browse but not buy until a later visit?

Answer: Although there is no one right answer, one popular reason is comparison shopping between different ecommerce sites before ultimately making a purchase decision. This is very common for luxury goods where significant up-front research and comparison is required by the customer before deciding (think car purchases) but also true to a lesser extent for the merchandise on this site (t-shirts, accessories, etc).

In the world of online marketing, identifying and marketing to these future customers based on the characteristics of their first visit will increase conversion rates and reduce the outflow to competitor sites.

Task 2. Identify an objective

Now you will create a Machine Learning model in BigQuery to predict whether or not a new user is likely to purchase in the future. Identifying these high-value users can help your marketing team to target them with special promotions and ad campaigns to ensure a conversion while they comparison shop between visits to your ecommerce site.

Task 3. Select features and create your training dataset

Google Analytics captures a wide variety of dimensions and measures about a user's visit on this ecommerce website. Browse the complete list of fields in the [UA] BigQuery Export schema documentation and then preview the demo dataset to find useful features that will help a machine learning model understand the relationship between data about a visitor's first time on your website and whether they will return and make a purchase.

Your team decides to test whether these two fields are good inputs for your classification model:

- `totals.bounces` (whether the visitor left the website immediately)
- `totals.timeOnSite` (how long the visitor was on our website)

Question: What are the risks of only using the above two fields?

Answer: Machine learning is only as good as the training data that is fed into it. If there isn't enough information for the model to determine and learn the relationship between your input features and your label (in this case, whether the visitor bought in the future) then you will not have an accurate model. While training a model on just these two fields is a start, you will see if they're good enough to produce an accurate model.

- In the BigQuery Editor, run the following query:

```
SELECT * EXCEPT(fullVisitorId) content_...
FROM
# features
(SELECT
fullVisitorId,
IFNULL(totals.bounces, 0) AS bounces,
IFNULL(totals.timeOnSite, 0) AS time_on_site
FROM
`data-to-insights.ecommerce.web_analytics`
WHERE
totals.newVisits = 1)
```

```

JOIN
(SELECT
  fullvisitorid,
  IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) >
0, 1, 0) AS will_buy_on_return_visit
FROM
  `data-to-insights.ecommerce.web_analytics`
GROUP BY fullvisitorid)
USING (fullVisitorId)
ORDER BY time_on_site DESC
LIMIT 10;

```

Results:

Row	bounces	time_on_site	will_buy_on_return_visit
1	0	15047	0
2	0	12136	0
3	0	11201	0
4	0	10046	0
5	0	9974	0
6	0	9564	0
7	0	9520	0
8	0	9275	1
9	0	9138	0
10	0	8872	0

Question: Which fields are the input features and the label?

Answer: The inputs are **bounces** and **time_on_site**. The label is **will_buy_on_return_visit**.

Question: Which two fields are known after a visitor's first session?

Answer: **bounces** and **time_on_site** are known after a visitor's first session.

Question: Which field isn't known until later in the future?

Answer: **will_buy_on_return_visit** is not known after the first visit. Again, you're predicting for a subset of users who returned to your website and purchased. Since you don't know the future at prediction time, you cannot say with certainty whether a new visitor will come back and purchase. The value of building an ML model is to get the probability of future purchase based on the data gleaned about their first session.

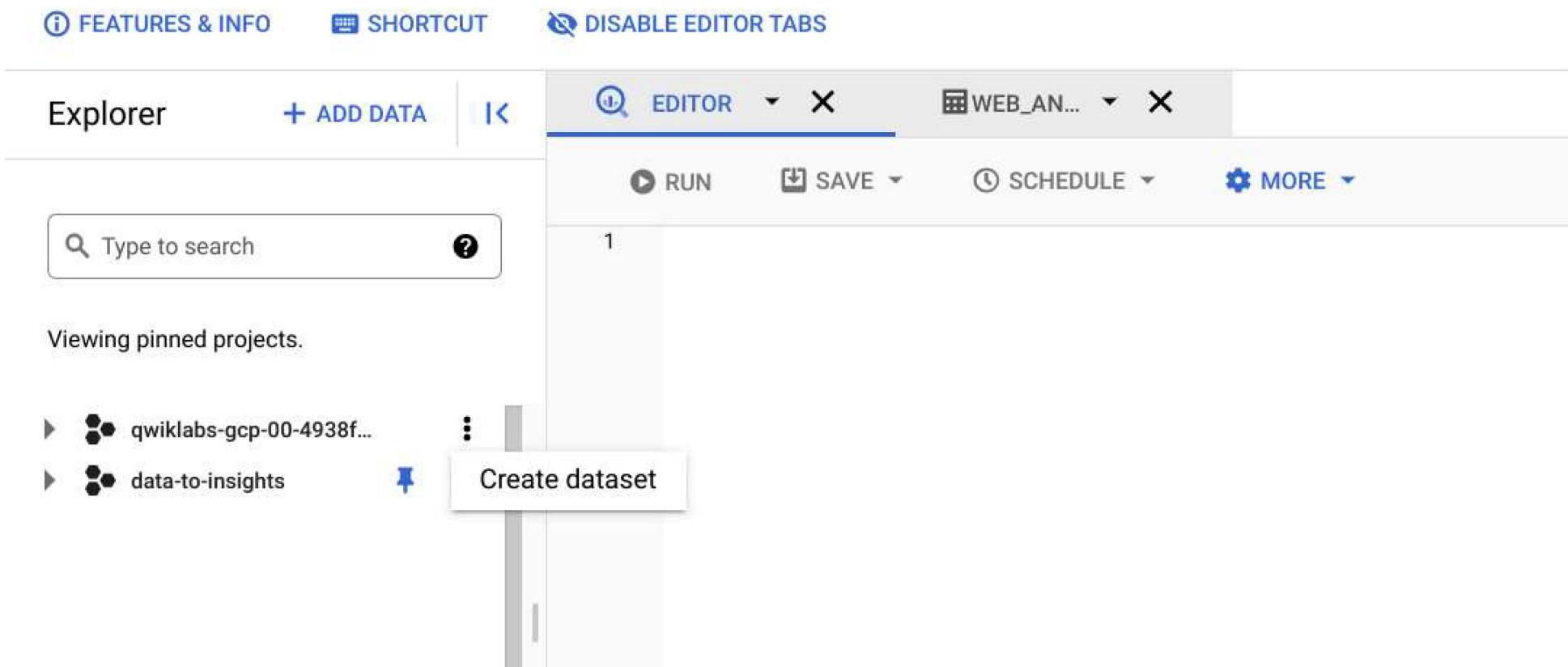
Question: Looking at the initial data results, do you think **time_on_site** and **bounces** will be a good indicator of whether the user will return and purchase or not?

Answer: It's often too early to tell before training and evaluating the model, but at first glance out of the top 10 **time_on_site**, only 1 customer returned to buy, which isn't very promising. Let's see how well the model does.

Task 4. 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, under **Explorer** section, click on the **View actions** icon next to your project name (starts with `qwiklabs-gcp-...`), and then click **Create dataset**.



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

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

3. Click **Create dataset**.

Click **Check my progress** to verify the objective.

Create a new dataset

Check my progress

Task 5. Select a BigQuery ML 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 the time of writing, BigQuery ML supports the two listed above.

Which model type should you choose?

Since you are bucketing visitors into "will buy in future" or "won't buy in future", use `logistic_reg` in a classification model.

The following query creates a model and specifies model options.

1. Run this query to train your model:

```
CREATE OR REPLACE MODEL `ecommerce.classification_model`  
OPTIONS  
(  
  model_type='logistic_reg',  
  labels = ['will_buy_on_return_visit'])  
AS  
  
#standardSQL  
SELECT  
  * EXCEPT(fullVisitorId)  
FROM  
  
  # features  
  (SELECT  
    fullVisitorId,  
    IFNULL(totals.bounces, 0) AS bounces,  
    IFNULL(totals.timeOnSite, 0) AS time_on_site  
  FROM  
    `data-to-insights.ecommerce.web_analytics`  
  WHERE  
    totals.newVisits = 1  
    AND date BETWEEN '20160801' AND '20170430') # train on first 9  
months  
JOIN  
  (SELECT  
    fullvisitorid,  
    IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) >  
    0, 1, 0) AS will_buy_on_return_visit  
  FROM
```

content_c

```
`data-to-insights.ecommerce.web_analytics`  
GROUP BY fullvisitorid)  
USING (fullVisitorId)  
;
```

2. Wait for the model to train (5 - 10 minutes).

Note: You cannot feed all of your available data to the model during training since you need to save some unseen data points for model evaluation and testing. To accomplish this, add a WHERE clause condition is being used to filter and train on only the first 9 months of session data in your 12 month dataset.

Click **Check my progress** to verify the objective.



Create a model and specify model options

[Check my progress](#)

After your model is trained, you will see the message "This statement created a new model named **qwiklabs-gcp-xxxxxxxxx:ecommerce.classification_model**".

3. Click **Go to model**.

4. Look inside the ecommerce dataset and confirm **classification_model** now appears.

Type to search

Viewing pinned projects.

- ▼ **qwiklabs-gcp-00-4938f...** ⋮
- ▼ **ecommerce** ⋮
- ▼ **Models (1)** ⋮
- classification_model** ⋮
- ▶ **data-to-insights** ⋮

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

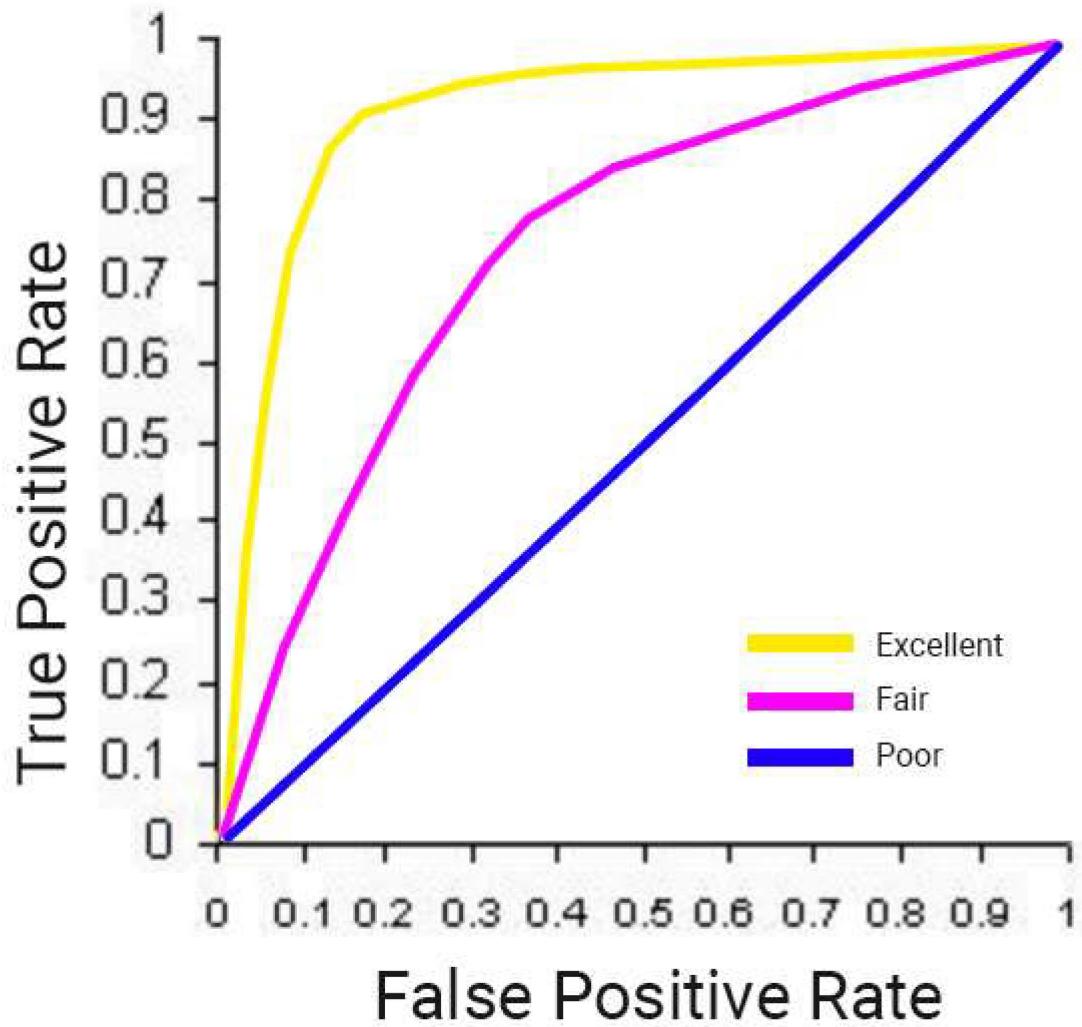
Task 6. Evaluate classification model performance

Select your performance criteria

For classification problems in ML, you want to minimize the False Positive Rate (predict that the user will return and purchase and they don't) and maximize the True Positive Rate (predict that the user will return and purchase and they do).

This relationship is visualized with a ROC (Receiver Operating Characteristic) curve like the one shown here, where you try to maximize the area under the curve or AUC:

Comparing ROC Curves



In BigQuery ML, `roc_auc` is simply a queryable field when evaluating your trained ML model.

- Now that training is complete, run this query to evaluate how well the model performs using `ML.EVALUATE`:

```

SELECT
  roc_auc,
  CASE
    WHEN roc_auc > .9 THEN 'good'
    WHEN roc_auc > .8 THEN 'fair'
    WHEN roc_auc > .7 THEN 'decent'
    WHEN roc_auc > .6 THEN 'not great'
    ELSE 'poor' END AS model_quality
FROM
  ML.EVALUATE(MODEL ecommerce.classification_model, (
    SELECT
      * EXCEPT(fullVisitorId)
    FROM
      # features
      (SELECT
        fullVisitorId,
        IFNULL(totals.bounces, 0) AS bounces,
        IFNULL(totals.timeOnSite, 0) AS time_on_site
      FROM
        `data-to-insights.ecommerce.web_analytics`
      WHERE
        totals.newVisits = 1
        AND date BETWEEN '20170501' AND '20170630') # eval on 2 months
    JOIN
      (SELECT
        fullvisitorid,
        IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) >
        0, 1, 0) AS will_buy_on_return_visit
      FROM
        `data-to-insights.ecommerce.web_analytics`
      GROUP BY fullvisitorid)
      USING (fullVisitorId)
));

```

You should see the following result:

Row	roc_auc	model_quality
1	0.7238561438561438	decent

After evaluating your model you get a **roc_auc** of 0.72, which shows the model has decent, but not great, predictive power. Since the goal is to get the area under the curve as close to 1.0 as possible, there is room for improvement.

Click **Check my progress** to verify the objective.



Evaluate classification model performance

[Check my progress](#)

Task 7. Improve model performance with Feature Engineering

As was hinted at earlier, there are many more features in the dataset that may help the model better understand the relationship between a visitor's first session and the likelihood that they will purchase on a subsequent visit.

1. Add some new features and create a second machine learning model called `classification_model_2`:

- How far the visitor got in the checkout process on their first visit
- Where the visitor came from (traffic source: organic search, referring site etc..)

- Device category (mobile, tablet, desktop)
- Geographic information (country)

2. Create this second model by clicking on "+" (Compose new query) icon:

```

CREATE OR REPLACE MODEL `ecommerce.classification_model_2`content_c
OPTIONS
  (model_type='logistic_reg', labels = ['will_buy_on_return_visit'])
AS

WITH all_visitor_stats AS (
SELECT
  fullvisitorid,
  IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) >
0, 1, 0) AS will_buy_on_return_visit
  FROM `data-to-insights.ecommerce.web_analytics`
  GROUP BY fullvisitorid
)

# add in new features
SELECT * EXCEPT(unique_session_id) FROM (
  SELECT
    CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS
unique_session_id,
    # labels
    will_buy_on_return_visit,
    MAX(CAST(h.eCommerceAction.action_type AS INT64)) AS
latest_ecommerce_progress,
    # behavior on the site
    IFNULL(totals.bounces, 0) AS bounces,
    IFNULL(totals.timeOnSite, 0) AS time_on_site,
    IFNULL(totals.pageviews, 0) AS pageviews,
    # where the visitor came from
    trafficSource.source,

```

```

trafficSource.medium,
channelGrouping,

# mobile or desktop
device.deviceCategory,

# geographic
IFNULL(geoNetwork.country, "") AS country

FROM `data-to-insights.ecommerce.web_analytics`,
UNNEST(hits) AS h

JOIN all_visitor_stats USING(fullvisitorid)

WHERE 1=1
# only predict for new visits
AND totals.newVisits = 1
AND date BETWEEN '20160801' AND '20170430' # train 9 months

GROUP BY
unique_session_id,
will_buy_on_return_visit,
bounces,
time_on_site,
totals.pageviews,
trafficSource.source,
trafficSource.medium,
channelGrouping,
device.deviceCategory,
country
);

```

Note: You are still training on the same first 9 months of data, even with this new model. It's important to have the same training dataset so you can be certain a better model output is attributable to better input features and not new or different training data.

A new key feature that was added to the training dataset query is the maximum checkout progress each visitor reached in their session, which is recorded in the field `hits.eCommerceAction.action_type`. If you search for that field in the field definitions you will see the field mapping of 6 =

Completed Purchase.

Note: The web analytics dataset has nested and repeated fields like ARRAYS which need to be broken apart into separate rows in your dataset. This is accomplished by using the UNNEST() function, which you can see in the above query.

3. Wait for the new model to finish training (5-10 minutes).

Click **Check my progress** to verify the objective.

○ Improve model performance with Feature Engineering(Create second model)

[Check my progress](#)

4. Evaluate this new model to see if there is better predictive power:

```
#standardSQL
SELECT
  roc_auc,
CASE
  WHEN roc_auc > .9 THEN 'good'
  WHEN roc_auc > .8 THEN 'fair'
  WHEN roc_auc > .7 THEN 'decent'
  WHEN roc_auc > .6 THEN 'not great'
  ELSE 'poor' END AS model_quality
FROM
  ML.EVALUATE(MODEL ecommerce.classification_model_2, (
WITH all_visitor_stats AS (
SELECT
  fullvisitorid,
  IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) >
```

content_c

```
0, 1, 0) AS will_buy_on_return_visit
  FROM `data-to-insights.ecommerce.web_analytics`
  GROUP BY fullvisitorid
)

# add in new features
SELECT * EXCEPT(unique_session_id) FROM (
  SELECT
    CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS
unique_session_id,
    # labels
    will_buy_on_return_visit,
    MAX(CAST(h.eCommerceAction.action_type AS INT64)) AS
latest_ecommerce_progress,
    # behavior on the site
    IFNULL(totals.bounces, 0) AS bounces,
    IFNULL(totals.timeOnSite, 0) AS time_on_site,
    totals.pageviews,
    # where the visitor came from
    trafficSource.source,
    trafficSource.medium,
    channelGrouping,
    # mobile or desktop
    device.deviceCategory,
    # geographic
    IFNULL(geoNetwork.country, "") AS country
  FROM `data-to-insights.ecommerce.web_analytics`,
  UNNEST(hits) AS h
  JOIN all_visitor_stats USING(fullvisitorid)
WHERE 1=1
  # only predict for new visits
  AND totals.newVisits = 1
  AND date BETWEEN '20170501' AND '20170630' # eval 2 months
```

```
GROUP BY
unique_session_id,
will_buy_on_return_visit,
bounces,
time_on_site,
totals.pageviews,
trafficSource.source,
trafficSource.medium,
channelGrouping,
device.deviceCategory,
country
)
));
```

Output:

Row	roc_auc	model_quality
1	0.9094875124875125	good

With this new model you now get a **roc_auc** of 0.91 which is significantly better than the first model.

Now that you have a trained model, time to make some predictions.

Click **Check my progress** to verify the objective.

Improve model performance with Feature Engineering(Better predictive power)

 Check my progress

Task 8. Predict which new visitors will come back and purchase

Next you will write a query to predict which new visitors will come back and make a purchase.

- The prediction query below uses the improved classification model to predict the probability that a first-time visitor to the Google Merchandise Store will make a purchase in a later visit:

```
SELECT * FROM ml.PREDICT(MODEL `ecommerce.classification_model_2`, (
  WITH all_visitor_stats AS (
    SELECT fullvisitorid,
      IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will_buy_on_return_visit
    FROM `data-to-insights.ecommerce.web_analytics`
    GROUP BY fullvisitorid
  )
  SELECT CONCAT(fullvisitorid, '-' ,CAST(visitId AS STRING)) AS unique_session_id,
    # labels
    will_buy_on_return_visit,
    MAX(CAST(h.eCommerceAction.action_type AS INT64)) AS latest_ecommerce_progress,
    # behavior on the site
    IFNULL(totals.bounces, 0) AS bounces,
    IFNULL(totals.timeOnSite, 0) AS time_on_site,
    totals.pageviews,
```

```

# where the visitor came from
trafficSource.source,
trafficSource.medium,
channelGrouping,

# mobile or desktop
device.deviceCategory,

# geographic
IFNULL(geoNetwork.country, "") AS country

FROM `data-to-insights.ecommerce.web_analytics`,
UNNEST(hits) AS h

JOIN all_visitor_stats USING(fullvisitorid)

WHERE
# only predict for new visits
totals.newVisits = 1
AND date BETWEEN '20170701' AND '20170801' # test 1 month

GROUP BY
unique_session_id,
will_buy_on_return_visit,
bounces,
time_on_site,
totals.pageviews,
trafficSource.source,
trafficSource.medium,
channelGrouping,
device.deviceCategory,
country
)

)

ORDER BY
predicted_will_buy_on_return_visit DESC;

```

The predictions are made in the last 1 month (out of 12 months) of the dataset.

Click **Check my progress** to verify the objective.

Predict which new visitors will come back and purchase

 Check my progress

Your model now outputs its predictions for those July 2017 ecommerce sessions. You can see three newly added fields:

- `predicted_will_buy_on_return_visit`: whether the model thinks the visitor will buy later (1 = yes)
- `predicted_will_buy_on_return_visit_probs.label`: the binary classifier for yes / no
- `predicted_will_buy_on_return_visit_probs.prob`: the confidence the model has in it's prediction (1 = 100%)

JOB INFORMATION		RESULTS			JSON		EXECUTION DETAILS									
Row	predicted_will_buy_on_return_visit	predicted_will_buy_on_return_visit_probs		unique_session_id			will_buy_on_return_visit	latest_ecommerce_progress	bounces	time_on_site	pageviews	sourc				
1	1		Row	label	prob	3052828106337222847-1499951313	0	6	0	3880	109	(dire				
			1	1	0.596723644289942											
			2	0	0.40327635571005804											
2	1		Row	label	prob	5847392129774736841-1501466665	0	6	0	685	21	gde				
			1	1	0.505748767363709											
			2	0	0.494251232636291											
3	1	▼ (2 rows)		4193294370598111620-1499731450		0	0		4	0		423	21	mon		
4	1	▼ (2 rows)		0389413525404733314-1500754279		0	0		6	0		2557	46	gde		
5	1	▼ (2 rows)		5213811450122638180-1500613030		0	0		6	0		729	24	gde		
6	1	▼ (2 rows)		8946235742138524977-1501001777		0	0		4	0		5962	80	mall		
7	1	▼ (2 rows)		1662238536128600596-1501001808		0	0		6	0		802	28	order		

Task 9. Analyze results and additional information

Results

- Of the top 6% of first-time visitors (sorted in decreasing order of predicted probability), more than 6% make a purchase in a later visit.
- These users represent nearly 50% of all first-time visitors who make a purchase in a later visit.
- Overall, only 0.7% of first-time visitors make a purchase in a later visit.
- Targeting the top 6% of first-time increases marketing ROI by 9x vs targeting them all!

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).

`roc_auc` is just one of the performance metrics available during model evaluation. Also available are accuracy, precision, and recall. Knowing which performance metric to rely on is highly dependent on what your overall objective or goal is.

Other datasets to explore

You can use the **bigrquery-public-data** project if you want to explore modeling on other datasets like forecasting fares for taxi trips.

1. To open the **bigrquery-public-data** dataset, click **+Add**. Click **Star a project by name** under Additional sources.
2. Then write the **bigrquery-public-data** name.

3. Click Star.

The `bigrquery-public-data` project is listed in the Explorer section.

Task 10. Test your knowledge

Test your knowledge about Google Cloud Platform by taking our quiz.



With BigQuery you can query terabytes and terabytes of data without having any infrastructure to manage or needing a database administrator.

- True
- False

Congratulations!

You've successfully built a machine learning model with BigQuery ML to classify ecommerce visitors and predict their purchasing habits.

Next steps / Learn more