# Predicting Visitor Purchases with a Classification Model with BigQuery ML

### **Objectives**

In this lab, you learn 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 the performance of your machine learning model
* Predict and rank the probability that a visitor will make a purchase

## 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. Click the query **EDITOR**.
2. Add the following to the New Query field:

#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

Copied!

content\_copy

1. Click **Run**.

The result: 2.69%

**Question:** What are the top 5 selling products?

1. Add the following query in the query **EDITOR**, and then click **Run**:

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;

Copied!

content\_copy

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-USA | Nest-USA | 6340 | 178572.4 |

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

1. Run the following query to find out:

# 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

Copied!

content\_copy

The results:

|  |  |  |
| --- | --- | --- |
| **Row** | **total\_visitors** | **will\_buy\_on\_return\_visit** |
| 1 | 729848 | 0 |
| 2 | 11873 | 1 |

Analyzing the results, you can see that (11873 / 729848) = 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.

## Task 2. Select features and create your training dataset

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 target them with special promotions and ad campaigns to ensure a conversion while they comparison shop between visits to your ecommerce site.

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 [here](https://support.google.com/analytics/answer/3437719?hl=en) and then [preview the demo dataset](https://bigquery.cloud.google.com/table/data-to-insights:ecommerce.web_analytics?tab=preview) 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)

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



Whether a user bounces is highly correlated with their time on site (e.g. 0 seconds)



Only using time spent on the site ignores other potential useful columns (features)



Both of the above

Submit

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 query **EDITOR**, add the following query and then click **Run**.

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)

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;

Copied!

content\_copy

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 |

Which fields are the model features? What is the label (correct answer)?



The features are bounces and time\_on\_site. The label is will\_buy\_on\_return\_visit



The features are bounces and will\_buy\_on\_return\_visit. The label is time\_on\_site



The feature is will\_buy\_on\_return\_visit. The labels are bounces and time\_on\_site

Submit

Which fields are known after a visitor's first session? (Check all that apply)



time\_on\_site



will\_buy\_on\_return\_visit



bounces



visitId

Submit

Which field isn't known until later in the future after their first session?



will\_buy\_on\_return\_visit



visitId



time\_on\_site



bounces

Submit

**Discussion:** **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 comes back and purchases. The value of building a ML model is to get the probability of future purchase based on the data gleaned about their first session.

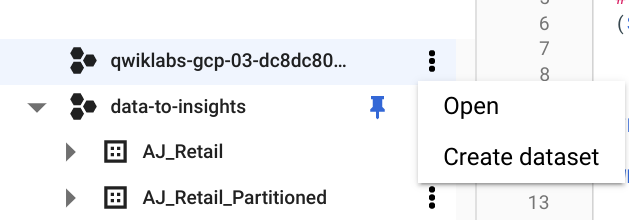
**Question:** Looking at the initial data results, do you think **time\_on\_site** and **bounces** will be a good indicator of whether the user will return and purchase or not?

**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 3. 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 on your project name, and then click on the View action icon (three dots) and select **Create Dataset**.



1. In the **Create Dataset** dialog:

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

1. Click **Create dataset**.

## Task 4. 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](https://www.tensorflow.org/tutorials/). At the time of writing, BigQuery ML supports the two listed above.

Which model type should you choose that will buy or won't buy?



Classification model (like logistic\_reg etc.)



Recommendation model (like matrix\_factorization etc.)



Forecasting model (like linear\_reg etc.)

Submit

1. Enter the following query to create a model and specify model options:

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

`data-to-insights.ecommerce.web\_analytics`

GROUP BY fullvisitorid)

USING (fullVisitorId)

;

Copied!

content\_copy

1. Next, click **Run** to train your model.

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.

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

1. Click **Go to model**.

Look inside the ecommerce dataset and confirm **classification\_model** now appears.

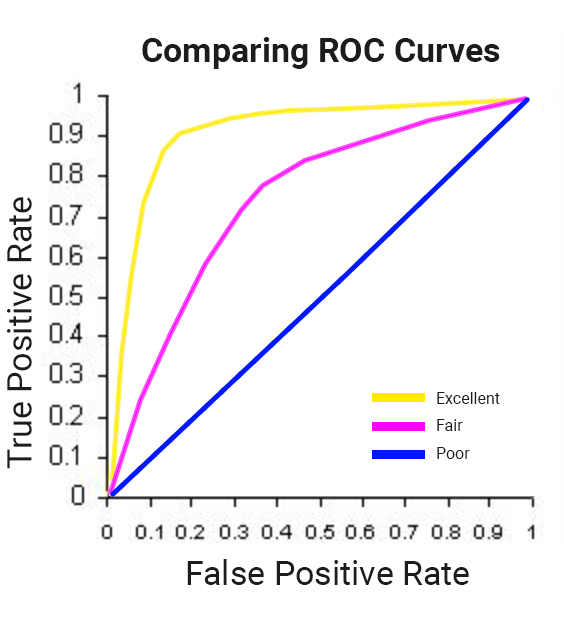
Next, you will evaluate the performance of the model against new unseen evaluation data.

## Task 5. 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:



In BigQuery ML, **roc\_auc** is simply a queryable field when evaluating your trained ML model.

Now that training is complete, you can evaluate how well the model performs by running this query using ML.EVALUATE:

SELECT

roc\_auc,

CASE

WHEN roc\_auc > .9 THEN 'good'

WHEN roc\_auc > .8 THEN 'fair'

WHEN roc\_auc > .7 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)

));

Copied!

content\_copy

You should see the following result:

|  |  |  |
| --- | --- | --- |
| **Row** | **roc\_auc** | **model\_quality** |
| 1 | 0.724588 | not great |

After evaluating your model you get a **roc\_auc** of 0.72, which shows that the model has 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.

## Task 6. 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.

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)

1. Create this second model by running the below query:

CREATE OR REPLACE MODEL `ecommerce.classification\_model\_2`

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,

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 '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

);

Copied!

content\_copy

**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 key new 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](https://support.google.com/analytics/answer/3437719?hl=en) you will see the field mapping of 6 = Completed Purchase.

As an aside, the web analytics dataset has nested and repeated fields like [ARRAYS](https://cloud.google.com/bigquery/docs/reference/standard-sql/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.

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

1. Evaluate this new model to see if there is better predictive power by running the below query:

#standardSQL

SELECT

roc\_auc,

CASE

WHEN roc\_auc > .9 THEN 'good'

WHEN roc\_auc > .8 THEN 'fair'

WHEN roc\_auc > .7 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) > 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

)

));

Copied!

content\_copy

(Output)

|  |  |  |
| --- | --- | --- |
| **Row** | **roc\_auc** | **model\_quality** |
| 1 | 0.910382 | 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.

## Task 7. 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.

Run the prediction query below which 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;

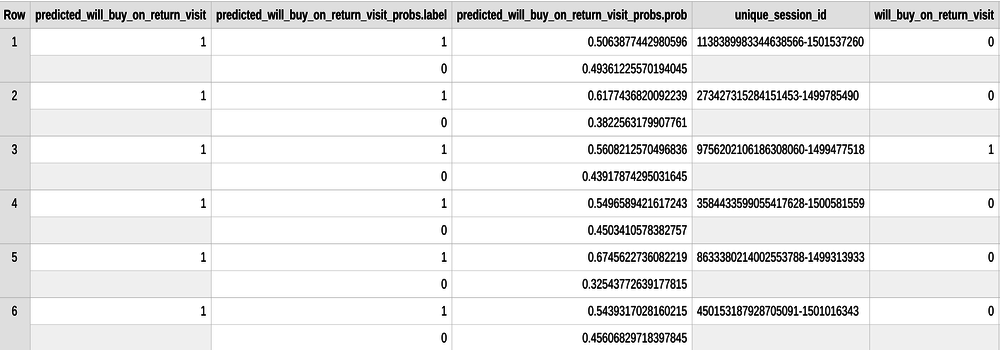
Copied!

content\_copy

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

Your model will now output the predictions it has 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%)



## 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**

**roc\_auc** is just one of the performance metrics available during model evaluation. Also available are [accuracy, precision, and recall](https://en.wikipedia.org/wiki/Precision_and_recall). Knowing which performance metric to rely on is highly dependent on what your overall objective or goal is.

### **Congratulations!**

You created a machine learning model using just SQL.

## Challenge

### **Summary**

In the previous two tasks you saw the power of feature engineering at work in improving our models performance. However, we still may be able to improve our performance by exploring other model types. For classification problems, BigQuery ML also supports the following model types:

* [Deep Neural Networks .](https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-dnn-models)
* [Boosted Decision Trees (XGBoost) .](https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-boosted-tree)
* [AutoML Tables Models .](https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-automl)
* [Importing Custom TensorFlow Models .](https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-tensorflow)

### **Task**

Though our linear classification (logistic regression) model performed well after feature engineering, it may be too simple of a model to fully capture the relationship between the features and the label. Using the same dataset and labels as you did in Task 6 to create the model ecommerce.classification\_model\_2, your challenge is to create a XGBoost Classifier.

**Hint :** Use following options for Boosted\_Tree\_Classifier:

1. L2\_reg = 0.1

2. num\_parallel\_tree = 8

3. max\_tree\_depth = 10

You may need to look at the documentation linked above to see the exact syntax. The model will take around 7 minutes to train. The solution can be found in the solution section below if you need help writing the query.

### **Solution:**

This is the solution that you require in order to create a XGBoost Classifier.

CREATE OR REPLACE MODEL `ecommerce.classification\_model\_3`

OPTIONS

(model\_type='BOOSTED\_TREE\_CLASSIFIER' , l2\_reg = 0.1, num\_parallel\_tree = 8, max\_tree\_depth = 10,

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,

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 '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

);

Copied!

content\_copy

Let us now evaluate our model and see how we did.

#standardSQL

SELECT

roc\_auc,

CASE

WHEN roc\_auc > .9 THEN 'good'

WHEN roc\_auc > .8 THEN 'fair'

WHEN roc\_auc > .7 THEN 'not great'

ELSE 'poor' END AS model\_quality

FROM

ML.EVALUATE(MODEL ecommerce.classification\_model\_3, (

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,

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

)

));

Copied!

content\_copy

Our roc\_auc has increased by about .02 to around .94!

**Note :** Your exact values will differ due to the randomness involved in the training process.

It’s a small change in the roc\_auc, but note that since 1 is a perfect roc\_auc, it gets more difficult to improve the metric the closer to 1 it gets.

This is a great example of how easy it is in BigQuery ML to try out different model types with different options to see how they perform. We were able to use a much more complex model type by only changing one line of SQL.

One may reasonably ask “Where did the choices for these options come from?”, and the answer is experimentation! When you are trying to find the best model type for your problems, then one has to experiment with different sets of options in a process known as hyperparameter tuning.

Let’s finish up by generating predictions with our improved model and see how they compare to those we generated before. By using a **Boosted tree classifier model**, you can observe a slight improvement of 0.2 in our ROC AUC compared to the previous model. The query below will predict which new visitors will come back and make a purchase.

SELECT

\*

FROM

ml.PREDICT(MODEL `ecommerce.classification\_model\_3`,

(

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

)

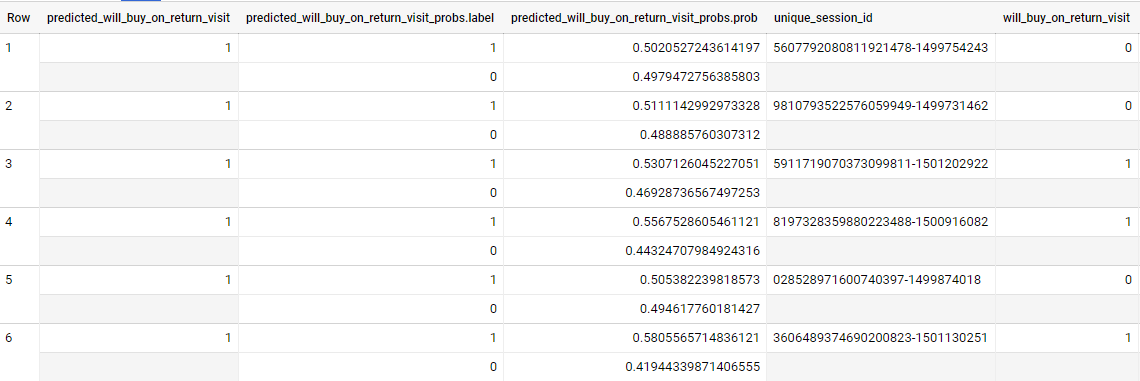
)

ORDER BY

predicted\_will\_buy\_on\_return\_visit DESC;

Copied!

content\_copy



The output now shows a classification model that can better predict the probability that a first-time visitor to the Google Merchandise Store will make a purchase in a later visit. By comparing the result above with the previous model shown in Task 7, you can see the confidence the model has in its predictions is more accurate when compared to the logistic\_regression model type.