Predicting Visitor Purchases with a Classification Model with Big Query 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

Access to public dataset

Once BigQuery is open, open data-to-insights project in a new browser tab to bring this project into your BigQuery projects panel.
The field definitions for the **data-to-insights** ecommerce dataset are here. Keep the link open in a new tab for reference

Explore ecommerce data

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

```
#standardSQL
WITH visitors AS(
SELECT
COUNT(DISTINCT fullVisitorId) AS total_visitors
FROM `data-to-insights.ecommerce.web_analytics`  # public dataset on bigquery
),
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
```

The result: 2.69%

Explore ecommerce data

Question: What are the top 5 selling products?

```
#standardSQL
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;
```

Result is

Row	v2ProductName	v2ProductCategory	units_sold	revenue
1	Nest® Learning Thermostat 3rd Gen-USA - Stainless Steet	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.

Explore ecommerce data

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

```
# 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
```

Result is

Row	total_visitors	will_buy_on_return_visit
1	729848	0
2	11873	1

Select features and create your training dataset

```
SELECT
  * EXCEPT(fullVisitorId)
FROM
  # features
  (SELECT
    fullVisitorId,
    IFNULL(totals.bounces, 0) AS bounces,
    IFNULL(totals.timeOnSite, 0) AS time_on_site    # deal with missing values
  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;
```

Result is

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

ML model

Dataset ID, type **ecommerce** # Create a BigQuery dataset to store models

```
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)
;
```

Evaluate classification model performance

```
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)
));
```

Row	roc_auc	model_quality
1	0.724588	not great

Improve model performance with feature engineering

```
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
);
```

Evaluate classification model performance

```
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
  )
  ));
```

Row	roc_auc	model_quality
1	0.910382	good

Predict which new visitors will come back and purchase

```
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;
```

Row	predicted_will_buy_on_return_visit	predicted_will_buy_on_return_visit_probs.label	predicted_will_buy_on_return_visit_probs.prob	unique_session_id	will_buy_on_return_visit
1	1	1	0.5063877442980596	1138389983344638566-1501537260	0
		0	0.4936122557019404		
2	1	1	0.6177436820092239	273427315284151453-1499785490	0
		0	0.3822563179907761		
3	1	1	0.5608212570496836	9756202106186308060-1499477518	1
		0	0.4391787429503164		
4	1	1	0.5496589421617243	3584433599055417628-1500581559	0
		0	0.4503410578382757		
5	1	1	0.6745622736082219	8633380214002553788-1499313933	0
		0	0.3254377263917815		
6	1	1	0.5439317028160215	450153187928705091-1501016343	0
		0	0.4560682971839785		