INDICE

INTRODUCCIÓN

Se van a intentar emplear técnicas de ML para la construcción de modelos de predicción que sean capaces de aumentar el rendimiento de dos casos de negocio. Para esto vamos a usar la aplicación de Machine Learning de BigQuery, se van a explorar los datasets de Google Analytics en BigQuery, se optimizarán y evaluarán distintos modelos hasta conseguir uno aceptable con el cual haremos predicciones de resultados. Después y por último, compararemos estas predicciones con los datos reales y se extraerán conclusiones.

CASO 1: USO DE BIGQUERY ML PARA PREDECIR USUARIOS COBRADOS

El objetivo de este ejercicio es predecir qué usuarios van a ser cobrados y cuáles no en un día concreto. El dataset empleado son los datos reales de Google Analytics de Digital Virgo Spain (DVS).

Como primer acercamiento a los datos, los visitantes del sitio web de DVS pueden suscribirse a sus servicios, lo que se contabilizará como transacción. Después DVS puede conseguir cobrarles la suscripción o no, dependiendo del estado de facturación de su línea telefónica. El % de conversión a transacción es menor a un 1%, y de la transacción al cobro (first billing) un 70% aproximadamente. El objetivo es predecir este first billing para invertir sólo en las transacciones que van a ser cobradas.

Para elaborar este ejercicio se usará la interfaz gráfica de BigQuery ML.

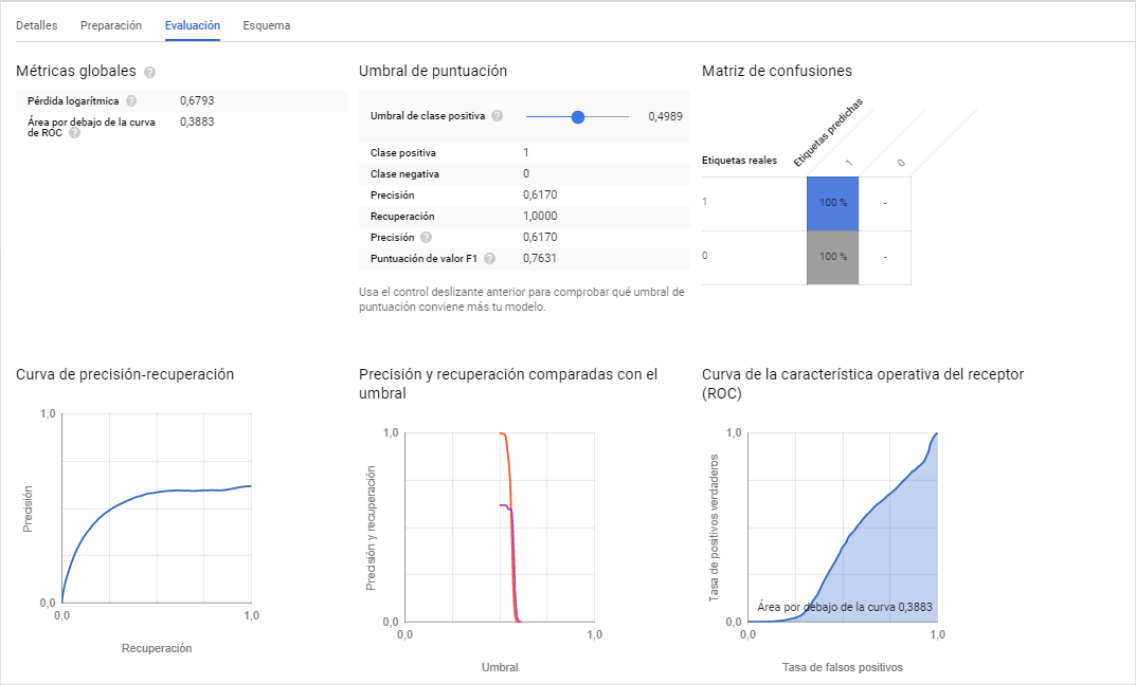
Modelo 1:

Para elaborar el primer modelo, se introdujeron solamente los usuarios que tenían suscripción, además de variables como la fecha, la hora en formato “momento del día”, marca del dispositivo, navegador; y métricas como número de sesiones y transacciones. La consulta quedó así:

Se usó solo a los usuarios suscritos ya que con el ratio de conversión actual el dataset es muy desbalanceado.

```
-- Modelo1
CREATE OR REPLACE MODEL `wide-oasis-135923.iamarketingdvs.modelodefinitivo1`
OPTIONS(model_type='logistic_reg') AS
SELECT
IFNULL(SUM(( SELECT value FROM UNNEST(hits.customMetrics) WHERE index = 2)),0) AS label,
date, case when hits.hour > 7 and hits.hour < 13 then 'manana' when hits.hour > 13 and
hits.hour < 20 then 'tarde' else 'noche' end as momentodia, hits.hour as hora, clientId,
geoNetwork.city as ciudad, device.deviceCategory as categoriaDispositivo,
device.mobileDeviceBranding as marca, device.browser as navegador, trafficSource.source
as fuente, trafficSource.campaign as campana, (SELECT value FROM
UNNEST(session.customDimensions) WHERE index = 14 GROUP BY 1) AS producto,
device.operatingSystem as OS, sum(totals.visits) as visitas, CASE WHEN
sum(totals.transactions) < 1 THEN 0 ELSE totals.transactions END as subs
FROM `wide-oasis-135923.140393857.ga_sessions_*` as session,
UNNEST(hits) AS hits
where _table_suffix BETWEEN '20190101' AND '20191001'
and trafficSource.campaign like '%range%'
and trafficSource.source like '%google%'
and totals.transactions > 0
GROUP BY date, clientId, ciudad, marca, categoriaDispositivo, totals.transactions,
navegador, campana, fuente, producto, OS, hora
```

Los resultados de la evaluación fueron estos:



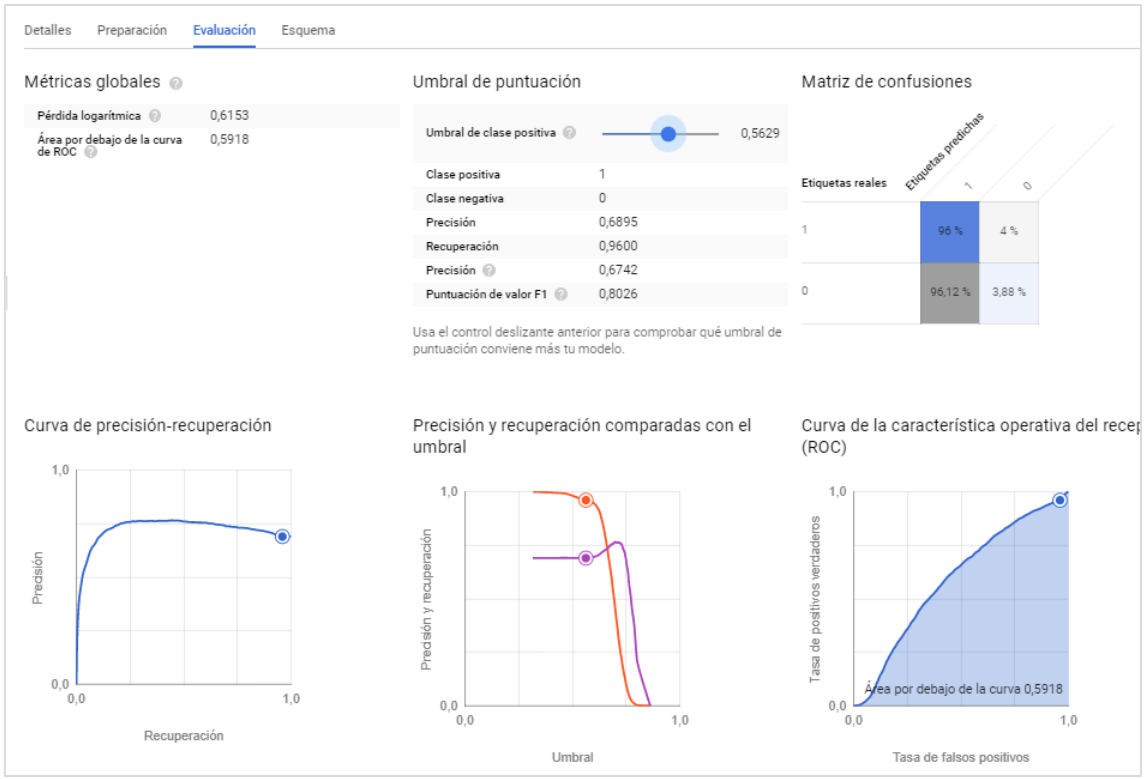
Como se puede ver, el modelo es malo. Predice el 61% de los casos de cobro sobre los suscritos, sin embargo, lo hace porque siempre dice que el suscrito es un cobro, con lo que acierta siempre el cobro, pero nunca el no cobro (recall 100%).

Lo siguiente fue afinar, se retiraron filas como las suscripciones (siempre 1), la campaña, la fuente de tráfico y la categoría del dispositivo entre otras, eran columnas que siempre tenían el mismo valor, por tanto no aportaban nada al modelo.

La consulta para la creación del segundo modelo quedó de esta forma:

```
CREATE OR REPLACE MODEL `wide-oasis-135923.iamarketingdvs.modelodefinitivo2`
OPTIONS(model_type='logistic_reg') AS
SELECT
IFNULL(SUM(( SELECT value FROM UNNEST(hits.customMetrics) WHERE index = 2)),0) AS label,
date, case when hits.hour > 7 and hits.hour < 13 then 'manana' when hits.hour > 13 and
hits.hour < 20 then 'tarde' else 'noche' end as momentodia, clientId, geoNetwork.city as
ciudad, device.mobileDeviceBranding as marca, device.browser as navegador,
trafficSource.campaign as campana, (SELECT value FROM UNNEST(session.customDimensions)
WHERE index = 14 GROUP BY 1) AS producto, device.operatingSystem as OS,
sum(totals.visits) as visitas
FROM `wide-oasis-135923.140393857.ga_sessions_*` as session,
UNNEST(hits) AS hits
where _table_suffix BETWEEN '20190101' AND '20191001'
and trafficSource.campaign like '%range%'
and trafficSource.source like '%google%'
and totals.transactions > 0
GROUP BY date, clientId, ciudad, marca, navegador, campana, producto, OS, momentodia
```

Su evaluación fue esta:



Se ha completado la consulta (tiempo transcurrido: 4,2 s; bytes procesados: 4,4 GB)

Información de la tarea | Resultados | JSON | Detalles de ejecución

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.5751609592236634	0.9961215255332903	0.5766740658527562	0.7292517006802721	0.689892207559442	0.562357

El recall en ambos modelos es del 100%, sospecho que el dataset tiene mucho que ver. No hay métricas entre la transacción y el billed, solo dimensiones. Cuando vimos el caso práctico en clase, la clave era la construcción de métricas personalizadas basándonos en datos existentes, sin embargo, por el modelo de negocio de este caso, no hay métricas intermedias disponibles entre la suscripción y el cobro.

De todas formas, vamos a hacer el predict para completar el proceso:

```
SELECT
  *
FROM ML.PREDICT(MODEL `wide-oasis-135923.iamarketingdvs.modelodefinitivo2`, (
SELECT
IFNULL(SUM(( SELECT value FROM UNNEST(hits.customMetrics) WHERE index = 2)),0) AS label,
date, case when hits.hour > 7 and hits.hour < 13 then 'manana' when hits.hour > 13 and
hits.hour < 20 then 'tarde' else 'noche' end as momentodia, hits.hour as hora, clientId,
geoNetwork.city as ciudad, device.deviceCategory as categoriaDispositivo,
device.mobileDeviceBranding as marca, device.browser as navegador, trafficSource.source
as fuente, trafficSource.campaign as campana, (SELECT value FROM
UNNEST(session.customDimensions) WHERE index = 14 GROUP BY 1) AS producto,
device.operatingSystem as OS, sum(totals.visits) as visitas, CASE WHEN
sum(totals.transactions) < 1 THEN 0 ELSE totals.transactions END as subs, IFNULL(SUM((
SELECT value FROM UNNEST(hits.customMetrics) WHERE index = 2)),0) AS Fbilled
FROM `wide-oasis-135923.140393857.ga_sessions_*` as session,
UNNEST(hits) AS hits
where _table_suffix BETWEEN '20191002' AND '20191030'
and trafficSource.campaign like '%range%'
and trafficSource.source like '%google%'
and totals.transactions > 0
```

```
GROUP BY date, clientId, ciudad, marca, categoriaDispositivo, totals.transactions,
navegador, campana, fuente, producto, OS, hora))
```

Mediante esta matriz de confusión improvisada en datastudio

predict	real	count ▾
1	1	6.164
1	0	4.553
0	0	71
0	1	24

Podemos comprobar la evaluación, muchas precisión, pero a base de predecir que todos los usuarios son cobrados, acierta un 60% de las veces porque hay un 60 % de cobro aproximadamente.

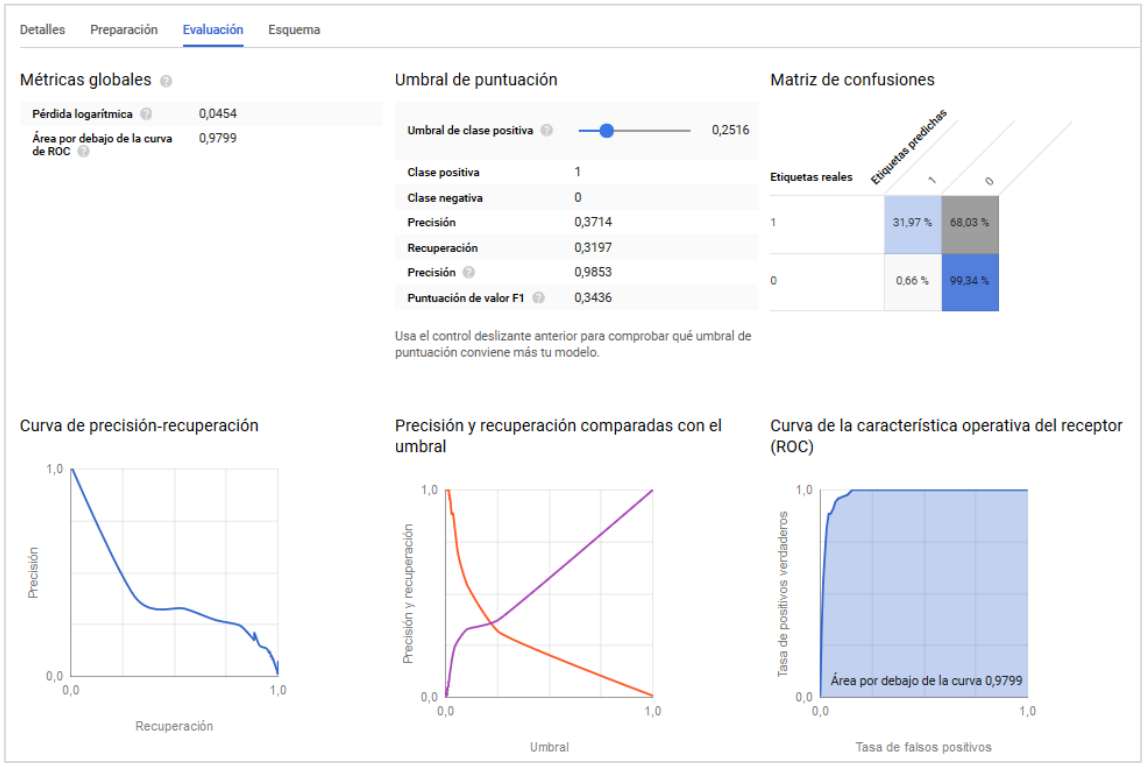
CASO 2. MEJORA DEL MODELO DE GOOGLE SOBRE EL GOOGLE MERCHANDISE STORE

Para poder trabajar en un modelo algo *mejor predictor*, hemos escogido el dataset de la Google Merchandise Store, con él pretendemos crear un modelo que funcione “normalmente” (mejor o peor) y predecir los usuarios que van a suscribirse un día concreto. Hay 1 año de datos, por lo que vamos a entrenarlo con los primeros 11 meses y predecir el día siguiente:

La consulta que generaría un primer modelo para hacer un acercamiento a los datos sería esta:

```
CREATE OR REPLACE MODEL `ml-small.MLsmalldataset.modeloGoogleTiendaGogle1`
OPTIONS(model_type='logistic_reg') AS
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS country,
  IFNULL(totals.pageviews, 0) AS pageviews
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE _TABLE_SUFFIX BETWEEN '20160801' AND '20170701'
```

La evaluación quedó así:



Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.468503937007874	0.11080074487895716	0.9853431583476764	0.17921686746987953	0.04624976980335791	0.98272

Los datos en esta ocasión son malos, pero constan de margen de mejora (o eso creo). Este *precission* nos dice que el modelo predice correctamente el 46% de las veces que lo hace, sin embargo, solo ha abarcado un 11% de los positivos reales (recall). Vamos a intentar aumentar estas métricas de rendimiento del modelo añadiendo algunas columnas más al dataset de entrenamiento del modelo:

```
CREATE OR REPLACE MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle1`
OPTIONS(model_type='logistic_reg') AS
SELECT

  IF(totals.transactions IS NULL, 0, 1) AS label,
  fullvisitorId as cliente,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS country,
  IFNULL(geoNetwork.city, "") AS ciudad,
  IFNULL(device.deviceCategory, "") AS categoriaDispositivo,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(totals.visits, 0) AS sesiones,
  IFNULL(totals.UniqueScreenViews, 0) AS visitasUnicas,
  IFNULL(totals.timeOnSite, 0) AS TiempoEnWeb,
  IFNULL(totals.sessionQualityDim, 0) AS calidadSesion

FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20160801' AND '20170630'
```

La evaluación quedó de esta manera:

```
SELECT
  *
FROM ML.EVALUATE(MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle1`, (
  SELECT

  IF(totals.transactions IS NULL, 0, 1) AS label,
  fullvisitorId as cliente,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS country,
  IFNULL(geoNetwork.city, "") AS ciudad,
  IFNULL(device.deviceCategory, "") AS categoriaDispositivo,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(totals.visits, 0) AS sesiones,
  IFNULL(totals.UniqueScreenViews, 0) AS visitasUnicas,
  IFNULL(totals.timeOnSite, 0) AS TiempoEnWeb,
  IFNULL(totals.sessionQualityDim, 0) AS calidadSesion

  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
```

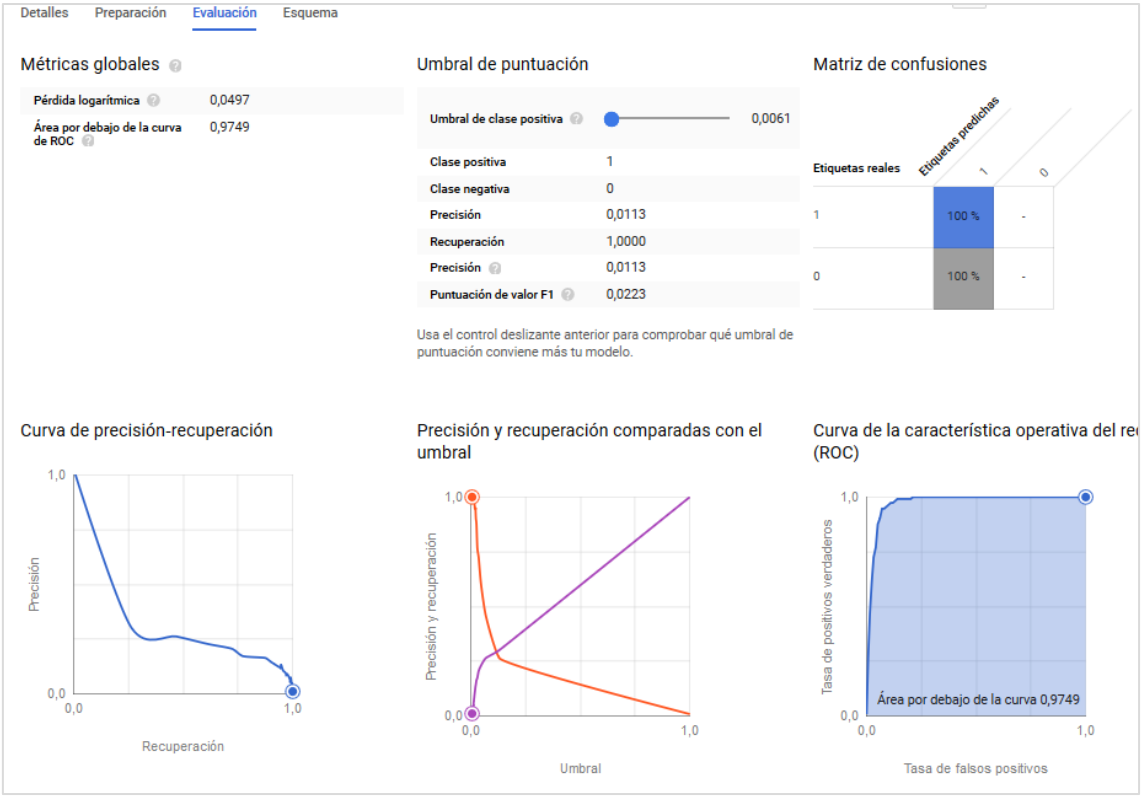
Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.3194444444444444	0.04283054003724395	0.9848590791738382	0.0755336617405583	0.05736572537282918	0.947676

Parece que hemos empeorado el modelo, puede ser por haber añadido dimensionalidad intrascendente en vez de métricas relevantes, vamos a crear un tercer modelo sólo añadiendo las métricas al original:

```
CREATE OR REPLACE MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle2`
OPTIONS(model_type='logistic_reg') AS
SELECT

  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(totals.visits, 0) AS sesiones,
  IFNULL(totals.UniqueScreenViews, 0) AS visitasUnicas,
  IFNULL(totals.timeOnSite, 0) AS TiempoEnWeb,
  IFNULL(totals.sessionQualityDim, 0) AS calidadSesion

FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20160801' AND '20170630'
```

Su correspondiente evaluación:

```
SELECT
  *
FROM ML.EVALUATE(MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle3`, (
  SELECT

  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(totals.visits, 0) AS sesiones,
  IFNULL(totals.UniqueScreenViews, 0) AS visitasUnicas,
  IFNULL(totals.timeOnSite, 0) AS TiempoEnWeb,
  IFNULL(totals.sessionQualityDim, 0) AS calidadSesion
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
```



Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.4020618556701031	0.07262569832402235	0.9850473321858864	0.12302839116719243	0.051748590533147114	0.979891

Este modelo efectivamente tiene menos dimensiones, sin embargo sigue siendo peor que el primero, el más sencillo.

Como no conseguimos mejorar el modelo añadiendo métricas ni dimensiones, vamos a modificar el primero modelo, es decir, mismo número de columnas, pero con diferentes valores:

Vamos a cambiar el país por ciudad, las visitas a páginas por sesiones y el sistema operativo por el navegador:

```
CREATE OR REPLACE MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle4`
OPTIONS(model_type='logistic_reg') AS
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.browser, "") AS navegador,
  IFNULL(geoNetwork.city, "") AS city,
  IFNULL(totals.visits, 0) AS sesiones
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE _TABLE_SUFFIX BETWEEN '20160801' AND '20170701'
```

La evaluación queda de esta forma:

```
SELECT
  *
FROM ML.EVALUATE(MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle4`, (
  SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.browser, "") AS navegador,
  IFNULL(geoNetwork.city, "") AS city,
  IFNULL(totals.visits, 0) AS sesiones
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
```

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.0	0.0	0.9855583046471601	0.0	0.06819285568749782	0.715836

Para encontrarle explicación a estos datos, vamos a explorar las columnas nuevas, es conocido que el dataset de muestra de la Merchandise Store de Google tiene algunos campos no disponibles por temas de privacidad o seguridad:

```
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.browser, "") AS navegador,
  IFNULL(geoNetwork.city, "") AS city,
  IFNULL(totals.visits, 0) AS sesiones
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE _TABLE_SUFFIX BETWEEN '20170801' AND '20170801'
```

Fila	label	navegador	city	sesiones
1	0	Edge	Houston	1
2	0	Chrome	Palo Alto	1
3	0	Chrome	Singapore	1
4	0	Firefox	Singapore	1
5	0	Chrome	Atlanta	1
6	0	Safari	Atlanta	1
7	0	Chrome	Tel Aviv-Yafo	1
8	0	Chrome	Kitchener	1
9	0	Safari	Houston	1

Todo parece correcto, sin embargo, vamos a hacer un último intento cambiando algunas métricas del modelo inicial, hasta el momento el mejor:

```
CREATE OR REPLACE MODEL `ml-small.MLsmalldataset.upgradeTiendaGogle5`
OPTIONS(model_type='logistic_reg') AS
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS pais,
  IFNULL(totals.timeOnSite, 0) AS tiempo,
  IFNULL(totals.pageviews, 0) AS pageviews
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE _TABLE_SUFFIX BETWEEN '20160801' AND '20170630'
```

La evaluación:

```
SELECT
  *
FROM ML.EVALUATE(MODEL `ml-small.MLsmalldataset.upgradeTiendaGogle5`, (
  SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS pais,
  IFNULL(totals.timeOnSite, 0) AS tiempo,
  IFNULL(totals.pageviews, 0) AS pageviews
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
```

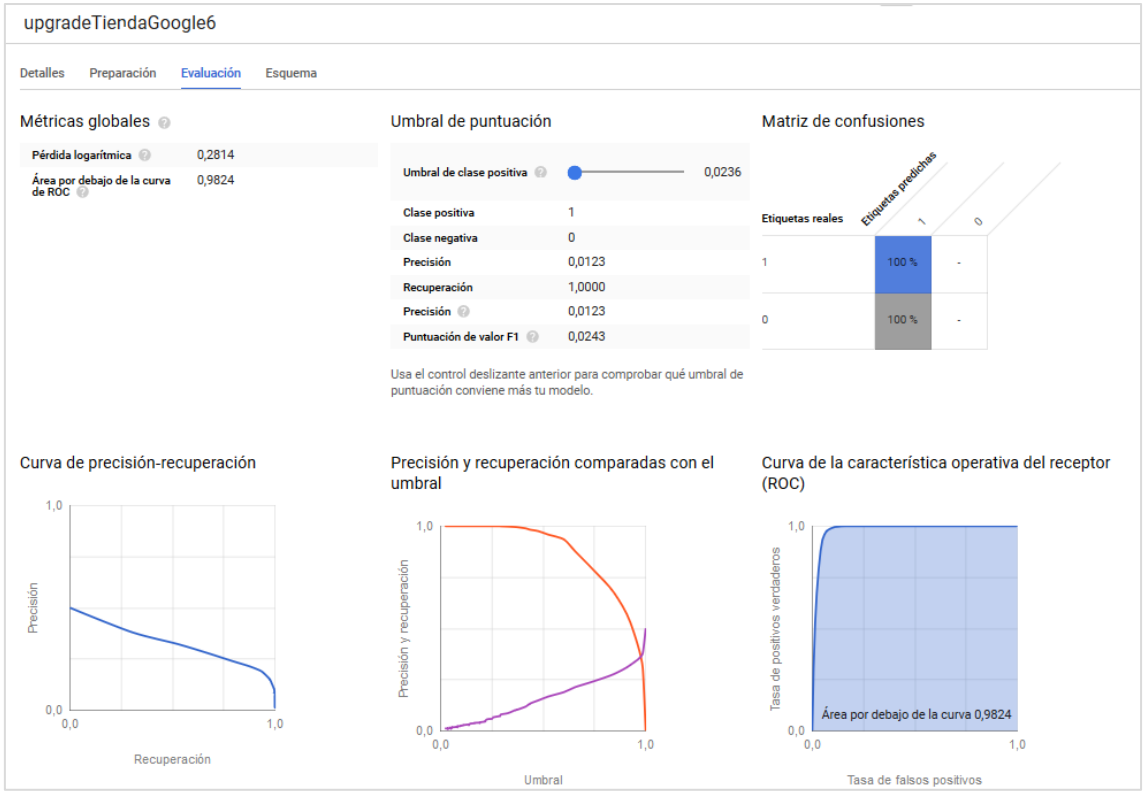
Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.45098039215686275	0.06424581005586592	0.9853566049913941	0.11246943765281174	0.04677313300865393	0.978796

No conseguimos añadir métricas y dimensiones que mejoren el modelo inicial. Sin embargo vamos a probar a cambiar los parámetros por defecto de la regresión logística:

```
CREATE OR REPLACE MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle6`
OPTIONS(model_type='logistic_reg',learn_rate_strategy='constant',data_split_method=
'random',data_split_eval_fraction = 0.15,learn_rate= 0.6,l1_reg=0.15,auto_class_weights
= true) AS
SELECT
  IF(totals.transactions IS NULL, 0, 1) AS label,
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(geoNetwork.country, "") AS country,
  IFNULL(totals.pageviews, 0) AS pageviews
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE _TABLE_SUFFIX BETWEEN '20160801' AND '20170701'
```

La evaluacion queda así:

```
SELECT
  *
FROM ML.EVALUATE(MODEL `ml-small.MLsmalldataset.upgradeTiendaGoogle6`, (
  SELECT
    IF(totals.transactions IS NULL, 0, 1) AS label,
    IFNULL(device.operatingSystem, "") AS os,
    device.isMobile AS is_mobile,
    IFNULL(geoNetwork.country, "") AS country,
    IFNULL(totals.pageviews, 0) AS pageviews
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
```



Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.18654981219817565	0.9711359404096834	0.9384278184165232	0.31297824456114026	0.2526054830864656	0.984171

Vemos como la precisión ha bajado, sin embargo el recall ha aumentado mucho, esto es, de los que predecimos como suscritos, el 18% lo son realmente, sin embargo, de todos los suscritos reales, hemos acertado un 97%.

Lo que realmente nos gusta es el primer modelo, no hemos conseguido mejorarlo de ninguna manera, así que vamos a predecir transacciones por usuario, sistema operativo y país en el mes de julio.

Para hacerlo usamos la función PREDICT de BQ ML, como ya usamos la de CREATE MODEL o la de EVALUATE.

```
SELECT
  os, country, fullvisitorId,
  SUM(predicted_label) as totalTransactionsPredicted
FROM ML.PREDICT(MODEL `ml-small.MLsmalldataset.modeloGoogleTiendaGogle1`, (
  SELECT
    IFNULL(device.operatingSystem, "") AS os,
    device.isMobile AS is_mobile,
    IFNULL(totals.pageviews, 0) AS pageviews,
    IFNULL(geoNetwork.country, "") AS country,
    fullvisitorId
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170731'))
  GROUP BY os, country, fullvisitorId
  ORDER BY totalTransactionsPredicted DESC
```

Vamos a hacer ahora un análisis de esta predicción y de la realidad. Para ello vamos a hacer la tabla de las suscripciones reales. Para esto haremos la misma consulta y guardaremos los resultados como otra métrica llamada suscripciones_reales. Después compararemos.

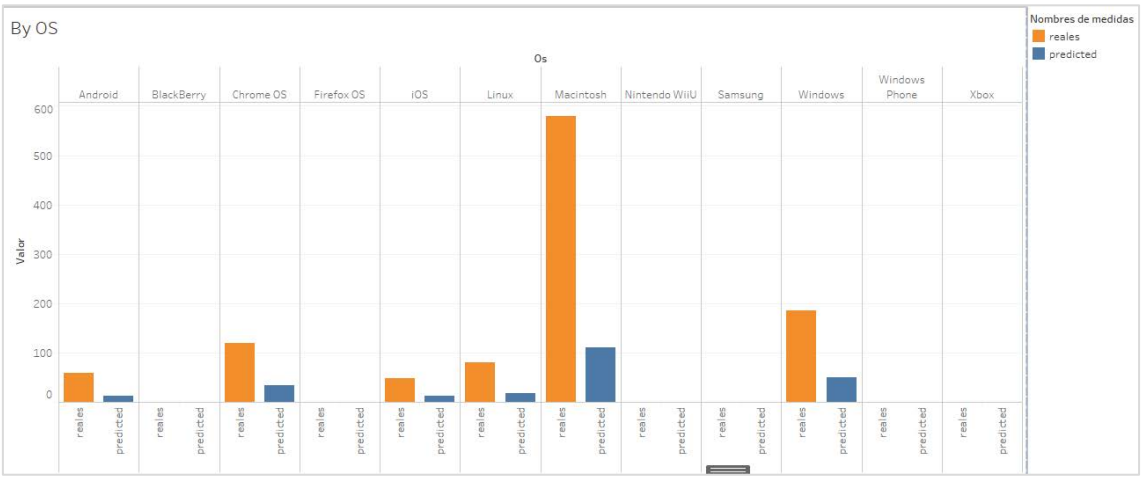
```
SELECT
    sum(totals.transactions) as realTransactions,
    IFNULL(device.operatingSystem, "") AS os,
    sum(totals.pageviews) AS pageviews,
    IFNULL(geoNetwork.country, "") AS country,
    fullvisitorId
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170731'
  GROUP BY os, country, fullvisitorId
  ORDER BY realTransactions DESC
```

Después de que el Join no funcione y pierda casi 2 horas de tiempo, caigo en la cuenta de que la tabla de transacciones reales tiene valores nulos en vez de ceros ☹ cambio la consulta:

```
SELECT
case when sum(totals.transactions) > 0 then sum(totals.transactions) else 0 end as
transaccionesReales,
    IFNULL(device.operatingSystem, "") AS os,
    IFNULL(geoNetwork.country, "") AS country,
    fullvisitorId
  FROM
    `bigquery-public-data.google_analytics_sample.ga_sessions_*`
  WHERE
    _TABLE_SUFFIX BETWEEN '20170701' AND '20170731'
  GROUP BY os, country, fullvisitorId
  ORDER BY transaccionesReales DESC
```

Vamos a guardar los resultados en dos nuevas tablas de BigQuery y las exploraremos en Tableau haciendo con este software la Join usando como clave las tres dimensiones, SO, coutry y fullvisitorId. Lo ideal es llevar la tabla a Tableau lo más limpia posible y preprocesada, pero como todo el proyecto es SQL, he querido hacer algo más en Tableau.

Subsreales vs subs predichas por SO:



Se puede ver, como ya notamos en la fase de evaluación, que hay más transacciones reales que predichas, se nota que el *recall* es relativamente bajo, sin embargo todos los usuarios

predichos hicieron transacción en el mes de Julio, una *precission* muy alta. Sin embargo, también puede notarse cierto buen funcionamiento del modelo, ya que no es sólo el número de transacciones lo que predice, sino que las asigna con bastante lógica a cada dimensión, es decir, el mayor número de transacciones predichas está en Macintosh, son menos de las reales, pero también son el máximo y así sucesivamente.

Con respecto a los datos por país, pasa lo mismo, la distribución es algo mayor entre dimensiones, pero los comentarios podrían ser los mismos.

Country	predict..	reales
United States	203	1.029
Canada	7	19
India	2	1
Germany	1	2
Indonesia	1	2
Italy	1	1
St. Lucia	1	1
United Kingdom	1	2
(not set)	0	1
Argentina	0	1
China	0	1
France	0	1
Greece	0	1
Israel	0	2
Malaysia	0	1
Mexico	0	2
Nigeria	0	2
Philippines	0	1
Poland	0	1
Switzerland	0	1

CONCLUSIÓN:

Se han entrenado modelos en dos datasets diferentes, dos modelos de negocio distintos, un modelo de suscripción doublé optin con un 97% de transacciones de usuarios en su primera sesión y un modelo de tienda online clásico como es el de la Google Merchandise Store.

En el primer caso, el modelo impide una predicción real de un objetivo ya que no existen a penas métricas disponibles entre los usuarios que visitan el site y los que son cobrados por las transacciones. El flujo de suscripción en doublé-optin, lo que quiere decir que se necesitan sólo 2 clics para realizar la transacción, lo que es bueno y malo. Bueno por performance de negocio, malo por la casi nula susceptibilidad de optimización aplicando técnicas de machine learning.

En el segundo caso se ha intentado mejorar el modelo de ejemplo que viene en los tutoriales de Google sobre BQ ML, el que se ha versionado para aplicar en el primer caso. En este segundo caso las expectativas eras buenas ya que el modelo usaba solo 3 dimensiones y 2 métricas, pensé que añadiendo más complejidad con nuevas métricas y dimensiones iba a poder conseguir una evaluación mejor que la primera, sin embargo, no fue así. Se aplicaron 3 tácticas, con estos resultados:

1.- Aumentar métricas y dimensiones:

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.3194444444444444	0.04283054003724395	0.9848590791738382	0.0755336617405583	0.05736572537282918	0.947676

2.- Aumentar sólo métricas:

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.4020618556701031	0.07262569832402235	0.9850473321858864	0.12302839116719243	0.051748590533147114	0.979891

3.- Mantener dimensionalidad, pero cambiando parámetros:

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.45098039215686275	0.06424581005586592	0.9853566049913941	0.11246943765281174	0.04677313300865393	0.978796

4.- mantener las dimensiones originales aplicando los parámetros de regresión logística aplicados en el caso práctico de Iberia:

Fila	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.18654981219817565	0.9711359404096834	0.9384278184165232	0.31297824456114026	0.2526054830864656	0.984171

La conclusión final es que no se ha conseguido generar ningún modelo aceptable más que el que venía dado de ejemplo con la Merchandise Store, parece que cuando nos enfrentamos a casos reales, el camino hasta la construcción de modelos no es tan llano y obvio como en los ejemplos preparados de los pétalos o los coches con la cilindrada, la velocidad máxima y la potencia que hemos ido haciendo en clase.

Aún más difícil que la construcción de un buen modelo, es prever y elegir qué tipo de modelo aplicar a cada caso de uso real en el negocio de cada uno.

DOCUMENTACION:

Sintaxis de en BQ ML:

https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create

How to query and calculate Google Analytics Data in BigQuery

https://towardsdatascience.com/how-to-query-and-calculate-google-analytics-data-in-bigquery-cab8fc4f396#f2c8

Esquema raw de campos de BigQuery Export

https://support.google.com/analytics/answer/3437719?hl=es

Primeros pasos con BigQuery ML en la IU web

https://cloud.google.com/bigquery-ml/docs/bigqueryml-web-ui-start

W3schools SQL Tutorial

https://www.w3schools.com/sql/default.asp