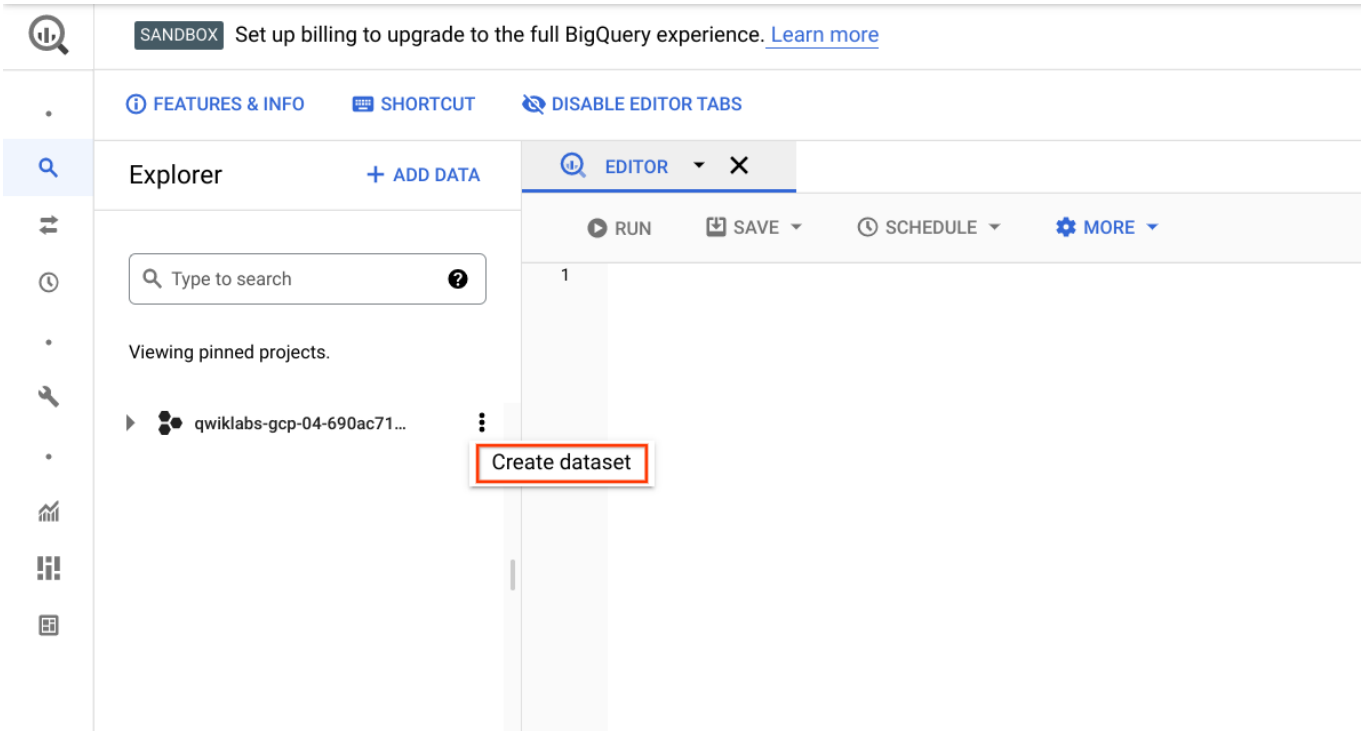
GCP – BigQuery 2 : apprentissage automatique

- Premiers pas avec BigQuery ML

Tâche 1. Créer un ensemble de données



Tâche 2. Créer un modèle

```
CREATE OR REPLACE MODEL `bqml_lab.sample_model`
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 '20170631'
LIMIT 100000;CRÉER  OU REMPLACER LE MODÈLE `bqml_lab.sample_model`
```

```
OPTIONS(model_type = 'logistic_reg' ) AS
```

(Facultatif) Informations sur le modèle et statistiques de formation

sample_model

QUERY MODEL | DELETE MODEL | EXPORT MODEL

DETAILS | TRAINING | EVALUATION | SCHEMA

View as

- ○ Graphs
- ◉ Table

Iteration	Training Data Loss	Evaluation Data Loss	Learn Rate	Duration (seconds)
10	0.0467	0.0342	25.6	4.63
9	0.0470	0.0343	12.8	4.70
8	0.0475	0.0350	25.6	5.31
7	0.0482	0.0354	25.6	5.03
6	0.0511	0.0393	12.8	5.05
5	0.0583	0.0471	6.4	6.10
4	0.0724	0.0624	3.2	6.96
3	0.1017	0.0934	1.6	5.93
2	0.1732	0.1673	0.8	6.01
1	0.3231	0.3197	0.4	6.33
0	0.5227	0.5214	0.2	5.04

Rows per page: 50 1 – 11 of 11

sample_model

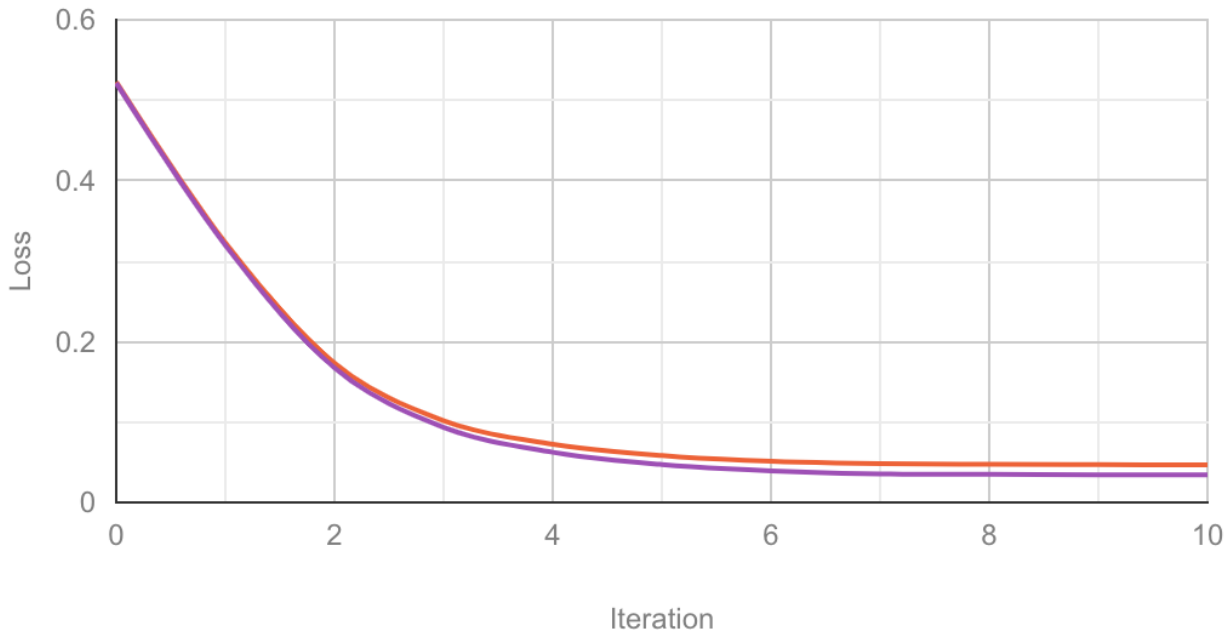
DETAILS | TRAINING | EVALUATION | SCHEMA

View as

(•) Graphs

( ) Table

Loss



Tâche 3. Évaluer le modèle

- Remplacez la requête précédente par la suivante, puis cliquez sur **Exécuter** :

#standardSQL

```
#standardSQL
SELECT
  *
FROM
  ml.EVALUATE(MODEL `bqml_lab.sample_model`, (
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'));
```

Row	precision	recall	accuracy	f1_score	log_loss	roc_auc
1	0.47368421052631576	0.10893854748603352	0.9853834982788297	0.17713853141559424	0.04552280390355375	0.9773986013986014

Tâche 4. Utiliser le modèle

Prédire les achats par pays

```
#standardSQL
SELECT
  country,
  SUM(predicted_label) as total_predicted_purchases
FROM
  ml.PREDICT(MODEL `bqml_lab.sample_model`, (
SELECT
  IFNULL(device.operatingSystem, "") AS os,
  device.isMobile AS is_mobile,
  IFNULL(totals.pageviews, 0) AS pageviews,
  IFNULL(geoNetwork.country, "") AS country
FROM
  `bigquery-public-data.google_analytics_sample.ga_sessions_*`
WHERE
  _TABLE_SUFFIX BETWEEN '20170701' AND '20170801'))
GROUP BY country
ORDER BY total_predicted_purchases DESC
LIMIT 10;
```

Vous devriez voir un tableau semblable à celui-ci :

Row	country	total_predicted_purchases
1	United States	140
2	Taiwan	5
3	India	2
4	Turkey	1
5	Venezuela	1
6	United Kingdom	1
7	Japan	1
8	Indonesia	1
9	Canada	1
10	St. Lucia	1

Prédire les achats par utilisateur

Voici un autre exemple. Cette fois, vous essaierez de prédire le nombre de transactions effectuées par chaque visiteur, de trier les résultats et de sélectionner les 10 premiers visiteurs par transactions :

- Remplacez la requête précédente par la suivante, puis cliquez sur **Exécuter :**

```
SELECT
 fullVisitorId, SUM (predicted_label) as total_predicted_purchases
FROM
 ml.PREDICT(MODEL `bqml_lab.sample_model`, ( SELECT
 IFNULL(device.operatingSystem, "") AS os, device.isMobile AS is_mobile,
IFNULL(totals.pageviews, 0 ) AS pages vues, IFNULL(geoNetwork.country, "")
AS country, fullVisitorId FROM
  `bigquery - public - data.google_analytics_sample.ga_sessions_ * `
 WHERE
  _TABLE_SUFFIX BETWEEN '20170701' AND '20170801' ))
 GROUP BY fullVisitorId
```

```
  ORDER BY total_predicted_purchases DESC
LIMIT dix ;
```

Vous devriez voir un tableau semblable à celui-ci :

Row	fullVisitorId	total_predicted_purchases
1	9417857471295131045	3
2	8069922490326866 50	2
3	0576935009275810 77	2
4	29694186761262587 98	2
5	0376394056092189113	2
6	83889310329550527 46	2
7	74203005015230124 60	2
8	12809936612043474 50	2
9	1122883309288959 42	2
10	86395516253142188 23	1

Tâche 5. Testez votre compréhension

BigQuery est un entrepôt de données d'entreprise entièrement géré qui permet des requêtes SQL ultra-rapides.

_vérifier_Vrai

False

Quelle option décrit le mieux ce que fait BigQuery ML ?

Creates machine learning models so you can export and use the model to re-evaluate the accuracy of other models.

_vérifier_Crée et exécute des modèles de machine learning dans BigQuery à l'aide de requêtes SQL standards.

Exports data from the warehouse, reformats the data, then executes the model using standard SQL queries.

Creates machine learning models using Python or Java in BigQuery, then executes the model using standard SQL queries.

GCP – BigQuery 2 : apprentissage automatique

- **Prédire les achats des visiteurs avec un modèle de classification dans BigQuery ML**

Ouvrir la console BigQuery

1. Dans la console Google Cloud, sélectionnez le **menu de navigation** > **BigQuery**.

Le message **Bienvenue sur BigQuery dans Cloud Console** s'affiche. Il contient un lien vers le guide de démarrage rapide et les notes de version.

2. Cliquez sur **OK**.

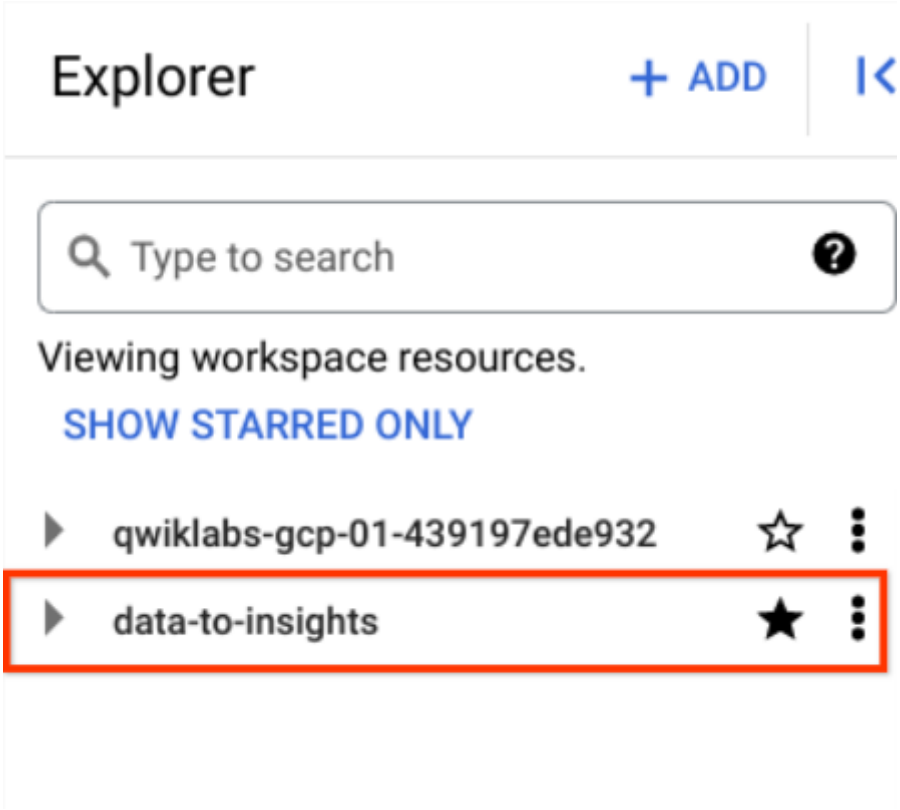
La console BigQuery s'ouvre.

Accéder à l'ensemble de données du cours

1. Dans le volet **Explorateur**, cliquez sur **+ AJOUTER**.

Le volet `Ajouter des données` s'ouvre.

2. Sous "Sources supplémentaires", cliquez sur **Ajouter un projet aux favoris en saisissant son nom**.
3. Saisissez `data-to-insights` et cliquez sur **Ajouter aux favoris**.



Cliquez sur le lien direct ci-dessous pour afficher le projet public **data-to-insights** :

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

Les définitions de champ correspondant à l'ensemble de données d'e-commerce **data-to-insights** sont disponibles sur cette page. Gardez la page ouverte dans un nouvel onglet pour référence.

Cliquez sur l'onglet **Requête**, puis sélectionnez **Dans un nouvel onglet** pour ouvrir l'éditeur de requête.

Tâche 1 : Explorer les données d'e-commerce

Scénario : Votre équipe d'analystes de données a exporté les journaux Google Analytics d'un site Web d'e-commerce dans BigQuery et a créé une table répertoriant toutes les données brutes relatives aux sessions visiteur du site afin que vous puissiez les explorer. Vous allez tenter de répondre à quelques questions à l'aide de ces données.

Question : Sur le nombre total de visiteurs de notre site Web, quel pourcentage a effectué un achat ?

1. Copiez la requête suivante et collez-la dans l'**Éditeur** BigQuery :

```
#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
```

2. Cliquez sur **Exécuter**.

Résultat : 2,69 %

Question : Quels sont les cinq produits qui se vendent le mieux ?

1. Effacez la requête précédente, puis ajoutez la requête suivante dans l'**Éditeur** :

```
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;
```

2. Cliquez sur **Exécuter**.

Vous obtenez le résultat suivant :

Ligne	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 : Combien d'internautes ont effectué un achat lors d'une nouvelle visite sur le site Web ?

1. Effacez la requête précédente, puis ajoutez la requête suivante dans l'**Éditeur** :

```
# 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
```

2. Cliquez sur **Exécuter**.

Résultats :

Ligne	total_visitors	will_buy_on_return_visit
1	729848	0
2	11873	1

L'analyse des résultats indique que 1,6 % (11 873/729 848) du nombre total de visiteurs revient sur le site Web et effectue un achat. Cela inclut le sous-ensemble de visiteurs qui ont effectué un achat lors de leur première session et lors d'une visite ultérieure.

Question : Pour quelles raisons un client lambda parcourt-il un site Web d'e-commerce, mais n'effectue un achat que lors d'une visite ultérieure ?

Réponse : Il n'y a pas de réponse unique à cette question, mais l'une des raisons souvent avancées est que les visiteurs comparent les offres sur différents sites d'e-commerce avant de prendre une décision d'achat. Cette pratique, très courante pour les achats de produits de luxe qui requièrent de nombreuses recherches et comparaisons préalables (achat d'une voiture, par exemple), s'applique aussi dans une moindre mesure à ce site de produits promotionnels (t-shirts, accessoires, etc.).

Dans l'univers du marketing en ligne, il est important d'identifier ces futurs clients et de tenir compte des caractéristiques de leur première visite pour augmenter les taux de conversion et limiter la fuite de clients potentiels vers des sites concurrents.

Tâche 2 : Identifier un objectif

Vous allez maintenant créer un modèle de machine learning dans BigQuery pour déterminer si un nouveau visiteur est susceptible d'effectuer un achat ultérieurement. Cette information peut aider votre équipe marketing à cibler les prospects intéressants avec des promotions et des campagnes publicitaires spéciales pour encourager les conversions si ceux-ci comparent les offres entre deux visites sur votre site e-commerce.

Tâche 3 : Sélectionner des caractéristiques et créer l'ensemble de données d'entraînement

Google Analytics capture un large éventail de dimensions et de mesures concernant les visites sur ce site Web d'e-commerce. Après avoir parcouru la liste complète des champs dans la documentation du schéma de BigQuery Export [UA], prévisualisez l'ensemble de données de démonstration pour identifier des caractéristiques pertinentes qui permettront à un modèle de machine learning d'établir une relation entre les données collectées lors de la première visite d'un internaute sur votre site Web et la probabilité qu'il y revienne pour effectuer un achat.

Votre équipe veut déterminer si les deux champs suivants conviendront pour votre modèle de classification :

- `totals.bounces` (visiteur qui quitte le site Web immédiatement)
- `totals.timeOnSite` (durée de la visite de l'internaute sur le site Web)

Question : Quels risques prendrions-nous en utilisant uniquement les champs ci-dessus ?

Réponse : Les résultats du machine learning dépendent des données qui lui sont fournies. Si le modèle ne dispose pas de données suffisantes pour déterminer et apprendre la relation entre vos caractéristiques d'entrée et votre étiquette (dans ce cas, si le visiteur a acheté un produit ultérieurement), il fournira des informations imprécises. Commencer à entraîner un modèle avec seulement ces deux champs n'est qu'un premier pas, mais cela vous permet tout de même de déterminer leur capacité à produire un modèle efficace.

- Dans l'**Éditeur** BigQuery, exécutez la requête suivante :

```
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;
```

Résultats :

Ligne	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 : Quels champs correspondent aux caractéristiques d'entrée et à l'étiquette ?

Réponse : Les entrées sont **bounces** (rebonds) et **time_on_site** (temps passé sur le site). L'étiquette est **will_buy_on_return_visit** (achètera lors d'une visite ultérieure).

Question : Quels sont les deux champs dont on connaît la valeur après la première session d'un visiteur ?

Réponse : Les valeurs des champs **bounces** et **time_on_site** sont disponibles après la première session d'un visiteur.

Question : Quel champ ne sera connu qu'à une date ultérieure ?

Réponse : La valeur de **will_buy_on_return_visit** n'est pas disponible après la première visite. Rappelez-vous que votre prédiction est basée sur un sous-ensemble d'utilisateurs qui sont revenus sur votre site Web pour effectuer un achat. Sachant qu'au moment de la prédiction, vous ne savez pas ce que l'avenir vous réserve, vous ne pouvez pas affirmer qu'un nouveau visiteur reviendra sur le site et effectuera un achat. La création d'un modèle de ML présente l'intérêt de calculer, à l'aide des données récupérées lors de la première session, la probabilité qu'un utilisateur donné effectue un achat ultérieurement.

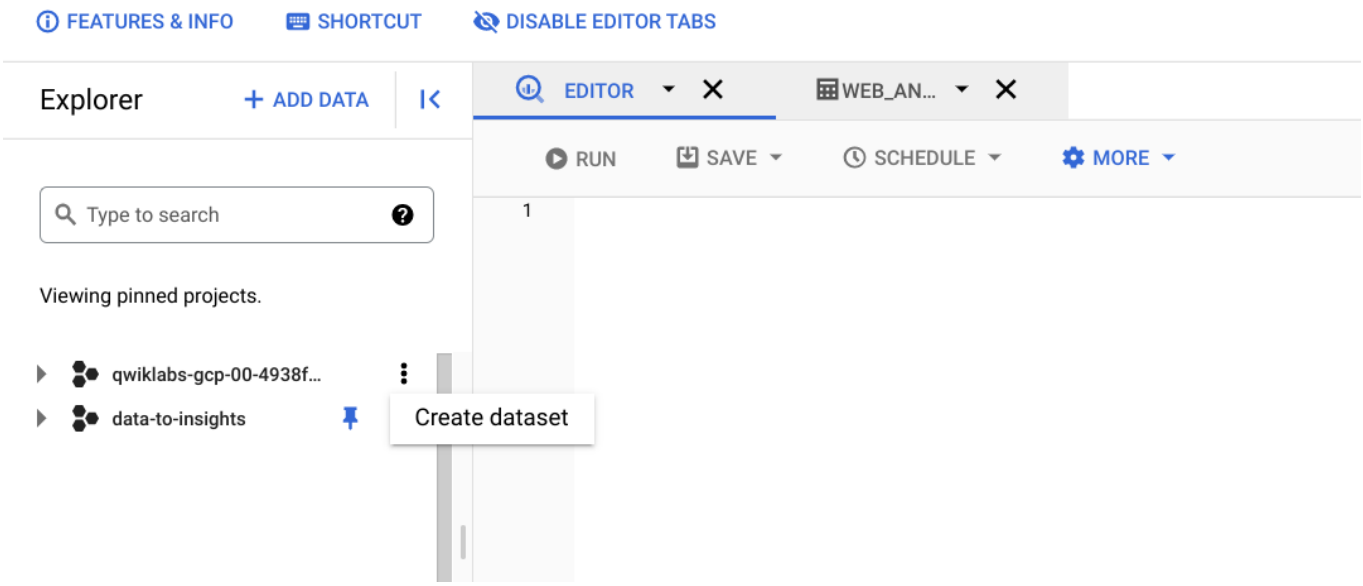
Question : Compte tenu des premiers résultats fournis par les données, pensez-vous que **time_on_site** et **bounces** sont de bons indicateurs pour déterminer si un utilisateur reviendra pour effectuer un achat ?

Réponse : Il faut généralement attendre que le modèle soit entraîné et évalué pour se prononcer. Cependant, on peut déjà voir à partir des 10 premières valeurs `time_on_site` qu'un seul client est revenu pour effectuer un achat, ce qui n'est pas très prometteur. Examinons à présent la qualité du modèle.

Tâche 4 : Créer un ensemble de données BigQuery pour y stocker vos modèles

Créons maintenant un autre ensemble de données BigQuery dans lequel stocker vos modèles de ML.

1. Dans le volet de gauche, dans la section **Explorateur**, cliquez sur l'icône **Afficher les actions** à côté du nom de votre projet (qui commence par `qwiklabs-gcp-...`), puis sur **Créer un ensemble de données**.



2. Dans la boîte de dialogue **Créer un ensemble de données** :

- Dans le champ **ID de l'ensemble de données**, saisissez "ecommerce".
- Conservez les valeurs par défaut dans les autres champs.

3. Cliquez sur **Créer un ensemble de données**.

Cliquez sur **Vérifier ma progression** pour valider l'objectif.

Créer un ensemble de données

Vérifier ma progression

Tâche 5 : Sélectionner un type de modèle BigQuery ML et définir les options correspondantes

Maintenant que vous avez sélectionné les caractéristiques de base, vous pouvez créer votre premier modèle de ML dans BigQuery.

Vous avez le choix entre deux types de modèles :

Modèle	**Type de modèle**	**Type de données de l'étiquette**	**Exemple**
Prévision	linear_reg (régression	Valeur numérique (nombre entier	Prévisions de ventes pour l'année prochaine d'après

	linéaire)	généralement ou à virgule flottante)	l'historique des données de ventes
Classification	logistic_reg (régression logistique)	0 ou 1 (classification binaire)	Classification ou non d'un e-mail dans la catégorie spam selon le contexte

Remarque : Il existe de nombreux autres types de modèles pour le machine learning (comme les réseaux de neurones et les arbres de décision). Ceux-ci sont disponibles dans des bibliothèques telles que TensorFlow. Au moment de la rédaction de cette page, BigQuery ML n'acceptait que les deux types mentionnés ci-dessus.

Quel type de modèle choisir ?

Puisque vous procédez au binning des visiteurs dans deux catégories ("achètera ultérieurement" et "n'achètera pas ultérieurement"), utilisez la régression logistique `logistic_reg` dans un modèle de classification.

La requête suivante permet de créer un modèle et d'en spécifier les options.

1. Exécutez-la pour entraîner votre modèle :

```
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)
;
```

2. Attendez que le modèle soit entraîné (cela prend entre 5 et 10 minutes).

Remarque : Vous ne pouvez pas transmettre la totalité des données disponibles au modèle pendant l'entraînement, car vous devez mettre de côté des points de données encore inconnus pour évaluer et tester le modèle. C'est la raison pour laquelle vous devez ajouter une clause WHERE, de manière à filtrer les données et cibler l'entraînement sur les neuf premiers mois de données de session, et non sur l'ensemble de données complet, qui porte quant à lui sur 12 mois.

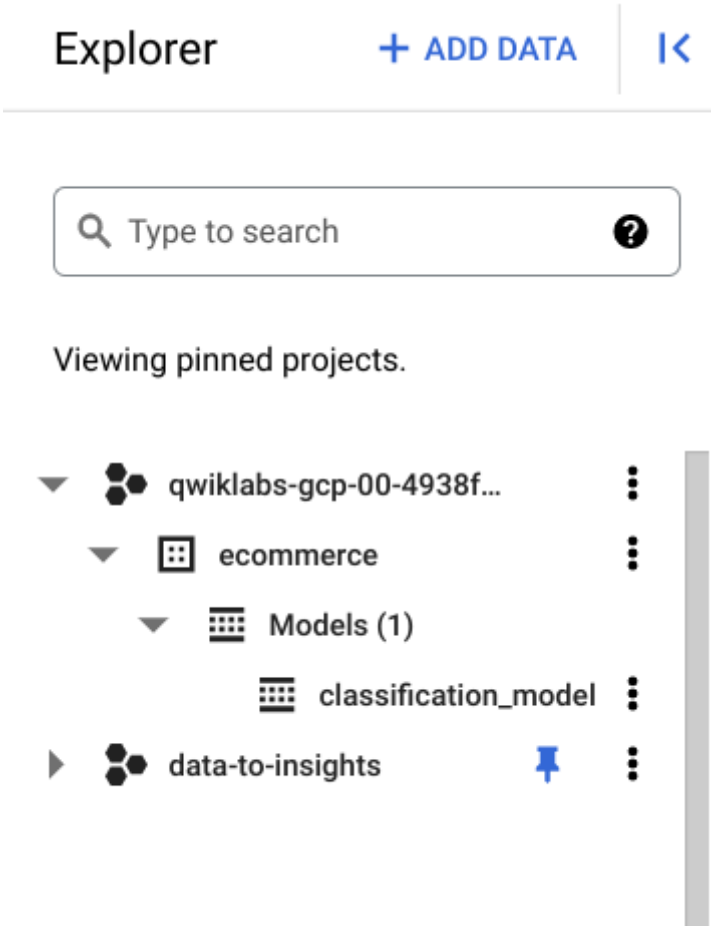
Cliquez sur **Vérifier ma progression** pour valider l'objectif.

Créer un modèle et définir ses options

Vérifier ma progression

Une fois votre modèle entraîné, le message suivant s'affiche : "This statement created a new model named **qwiklabs-gcp-xxxxxxxxx:ecommerce.classification_model**" (Cette instruction a créé un modèle nommé qwiklabs-gcp-xxxxxxxxx:ecommerce.classification_model).

3. Cliquez sur **Accéder au modèle**.
4. Vérifiez que le modèle **classification_model** apparaît bien au sein de l'ensemble de données "ecommerce".



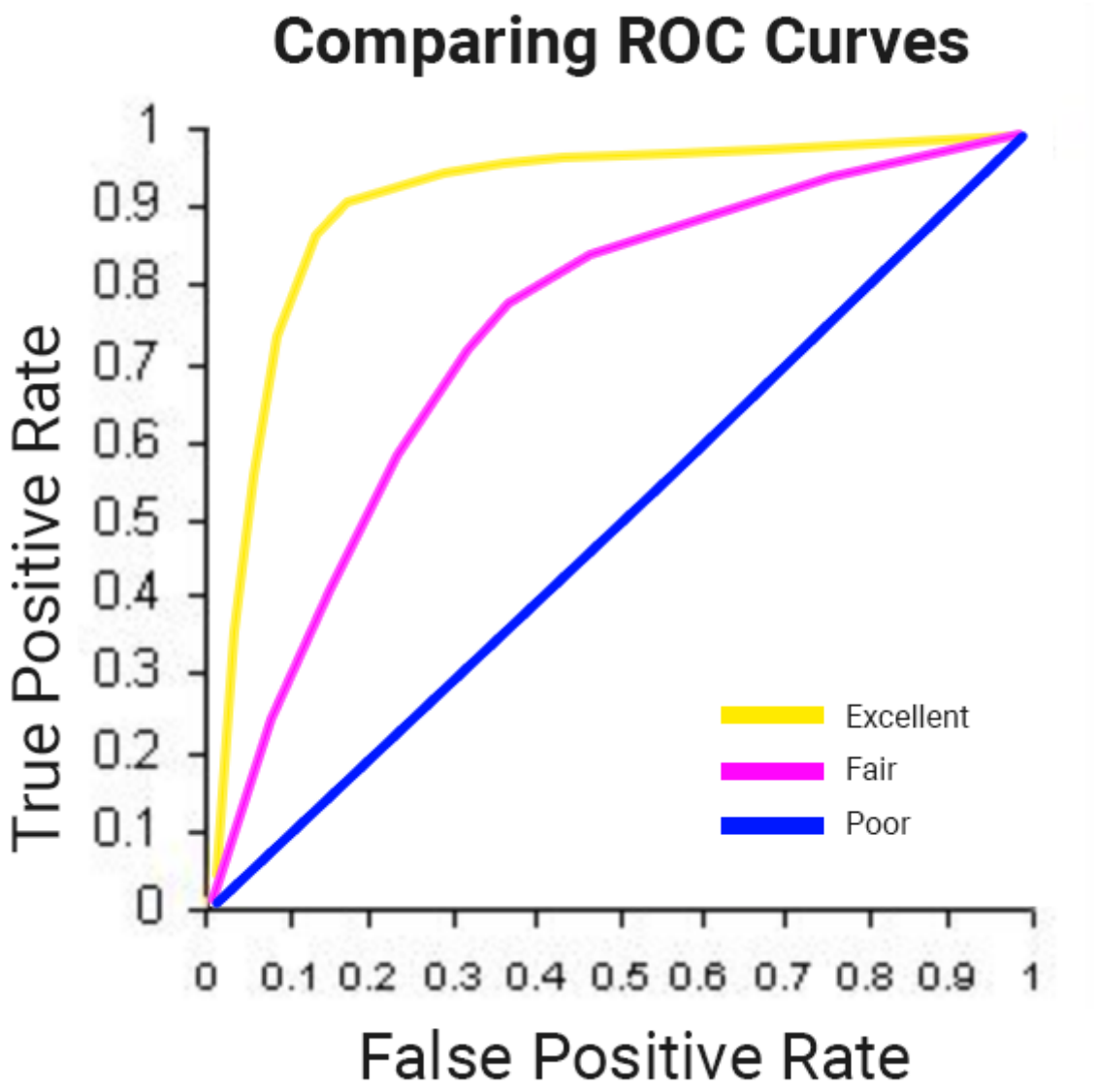
L'étape suivante consiste à évaluer les performances du modèle à partir de données d'évaluation non rencontrées précédemment.

Tâche 6 : Évaluer les performances du modèle de classification

Sélectionner vos critères de performances

Pour limiter les problèmes de classification en ML, il convient de réduire au minimum le taux de faux positifs (par exemple, prédire le retour d'un utilisateur et un achat, et que cela ne se produise pas) et d'obtenir un taux de vrais positifs maximal (prédire le retour d'un utilisateur et un achat, et que cela se produise).

Cette relation est représentée par une courbe ROC (receiver operating characteristic) telle qu'illustrée ci-dessous, l'objectif étant d'avoir un AUC (Area under the ROC - Espace sous la courbe ROC) le plus grand possible :



Dans BigQuery ML, **roc_auc** désigne simplement un champ interrogeable lors de l'évaluation du modèle de ML entraîné.

- Maintenant que l'entraînement est terminé, utilisez la fonction `ML.EVALUATE` pour évaluer les performances du modèle. Pour ce faire, exécutez cette requête :

```
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)

));
```

Vous devriez obtenir le résultat suivant :

Ligne	roc_auc	model_quality
1	0.7238561438561438	decent

L'évaluation de votre modèle indique un **roc_auc** de 0,72, correspondant à des performances prédictives correctes, mais largement améliorables. Le but étant d'obtenir un espace sous la courbe le plus proche possible de 1, vous avez une marge de progression.

Cliquez sur **Vérifier ma progression** pour valider l'objectif.

Évaluer les performances du modèle de classification

Vérifier ma progression

Tâche 7 : Améliorer les performances du modèle par extraction de caractéristiques

Comme indiqué précédemment, l'ensemble de données renferme de nombreuses autres caractéristiques susceptibles d'aider le modèle à mieux comprendre la relation entre la première session d'un visiteur et la probabilité qu'il effectue un achat lors d'une visite ultérieure.

1. Créez un second modèle de machine learning nommé `classification_model_2` comprenant les caractéristiques supplémentaires suivantes :

- Étape du processus de paiement à laquelle le visiteur s'est arrêté
- Provenance du visiteur (source du trafic : recherche naturelle, site référent, etc.)
- Catégorie d'appareil (mobile, tablette, ordinateur de bureau)
- Localisation géographique (pays)

2. Créez le second modèle en cliquant sur l'icône "**+**" (Saisir une nouvelle requête) :

```
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,
      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
```

```
);
```

3. Attendez que le nouveau modèle soit entraîné (5 à 10 minutes).

Cliquez sur **Vérifier ma progression** pour valider l'objectif.

Améliorer les performances du modèle par extraction de caractéristiques (Créer le second modèle)

Vérifier ma progression

4. Évaluez les performances prédictives du nouveau modèle :

```
#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) > 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
)
));
```

Résultat :

Ligne	roc_auc	model_quality

1	0.9094875124875125	good

Tâche 8 : Prédire quels nouveaux visiteurs reviendront effectuer un achat

Vous allez à présent élaborer une requête permettant de prédire quels nouveaux visiteurs reviendront pour effectuer un achat.

- La requête de prédiction ci-dessous emploie le modèle de classification amélioré afin de calculer la probabilité qu'un nouveau visiteur du Google Merchandise Store achète un produit lors d'une visite ultérieure :

```
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;
```

Les prédictions sont établies pour le dernier mois (sur les 12 mois que compte l'ensemble de données).

Cliquez sur **Vérifier ma progression** pour valider l'objectif.

Prédire quels nouveaux visiteurs reviendront effectuer un achat

Vérifier ma progression

Le modèle va maintenant fournir des prédictions pour les sessions d'e-commerce de juillet 2017. Comme vous le voyez, trois champs ont été ajoutés :

- predicted_will_buy_on_return_visit : indique si le modèle pense que le visiteur achètera un produit ultérieurement (1 = oui)
- predicted_will_buy_on_return_visit_probs.label : classificateur binaire pour oui/non
- predicted_will_buy_on_return_visit.probs.prob : taux de confiance du modèle dans sa prédiction (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	sour
1	1	Row label prob 1 1 0.596723644289942 2 0 0.40327635571005804	3052828106337222847-1499951313	0	6	0	3880	109	(dire
2	1	Row label prob 1 1 0.505748767363709 2 0 0.494251232636291	5847392129774736841-1501466665	0	6	0	685	21	gdea
3	1	(2 rows)	419329437059811162O-1499731450	0	4	0	423	21	mon
4	1	(2 rows)	0389413525404733314-1500754279	0	6	0	2557	46	gdea
5	1	(2 rows)	5213811450122638180-1500613030	0	6	0	729	24	gdea
6	1	(2 rows)	8946235742138524977-1501001777	0	4	0	5962	80	mall
7	1	(2 rows)	1662338536128600596-1501091808	0	6	0	892	28	gdea

Tâche 9 : Analyser les résultats et informations complémentaires

Résultats

- Sur les premiers 6 % de nouveaux visiteurs (classés par ordre décroissant de probabilité), plus de 6 % effectuent un achat lors d'une visite ultérieure.
- Ces utilisateurs représentent près de 50 % de tous les nouveaux visiteurs effectuant un achat lors d'une visite ultérieure.
- Globalement, 0,7 % seulement des nouveaux visiteurs effectuent un achat lors d'une visite ultérieure.
- Le ciblage des premiers 6 % de nouveaux visiteurs (plutôt que de l'ensemble de ces derniers) permet de dégager un ROI marketing neuf fois supérieur.

Informations supplémentaires

Conseil : Si vous relancez l'entraînement d'un modèle existant avec de nouvelles données, gagnez du temps en ajoutant `warm_start = true` à ses options. Notez qu'il est impossible de modifier les colonnes de caractéristiques. Cette opération nécessiterait un nouveau modèle.

roc_auc est une métrique de performances parmi d'autres pour évaluer des modèles. Il en existe d'autres telles que la justesse, la précision et le rappel. Le choix de la métrique adéquate dépend principalement de l'objectif global que vous vous êtes fixé.

Autres ensembles de données à explorer

Vous pouvez utiliser le projet **bigquery-public-data** si vous souhaitez découvrir la modélisation d'autres ensembles de données, par exemple pour prédire le tarif de courses en taxi.

1. Pour ouvrir l'ensemble de données **bigquery-public-data**, cliquez sur **+Ajouter**. Sous "Sources supplémentaires", cliquez sur **Ajouter un projet aux favoris en saisissant son nom**.
2. Indiquez ensuite le nom `bigquery-public-data`.
3. Cliquez sur **Ajouter aux favoris**.

Le projet `bigquery-public-data` apparaît désormais dans la section "Explorateur".

Tâche 10 : Tester vos connaissances

Testez vos connaissances sur Google Cloud Platform en répondant à notre quiz.

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

Vrai

Félicitations !

Vous venez de créer un modèle de ML dans BigQuery pour classifier les visiteurs d'un site d'e-commerce.