|  |
| --- |
| **SIMPLON.CO DEVELOPPEUR DATA**  **juin à décembre 2019** |
| Données Google Analytics  Analyse de performance du site  Google Merchandise Store |
| *Structuration de données*  *Conception d’une base de données*  *Traitement et visualisation* |
|  |
|  |
|  |

|  |
| --- |
| Projet de Sophie Soizeau Saint-Martin  Certifications “Développer une base de données” (CNCP 3497)  et “Exploiter une base de données” (CNCP 3508) |

TABLE DES MATIÈRES

**Introduction 3-5**

I. Présentation du projet 3

II. Cahier des charges 4

III. Langages et outils utilisés 5

**Partie 1 : Conception et Construction de la base de données 6-20**

I. Source des données 6

II. Description et exploration des données 7-8

III. Modèle conceptuel des données 9-10

IV. Exploration et extraction des données 11-16

V. Importation des données vers MySQL 17

VI. Intégrité et Sécurisation des données 18-19

VII. Optimisations 20

**Partie 2 : Analyse et visualisation des données 21-22**

I. Rappel des objectifs et outils 21

III. Visualisation 22

**Bilan de projet et pistes d’amélioration 24**

**Annexe 1 – Détail des attributs Google Analytics 25**

**INTRODUCTION**

1. **Présentation du projet et résumé**

La Data générée par les utilisateurs sur le web représente une grande partie de la Data disponible. L’analyse des données utilisateurs sur un site web ou une application permet d’améliorer la connaissance client, la qualité des services et les performances des actions marketing.

Les données collectées sur l’audience, les parcours de navigation des utilisateurs et leurs interactions avec le site jusqu’à la conversion sont analysables via Google Analytics.

Je vais traiter les données Google Analytics issues du site e-commerce Google Store, en particulier les informations suivantes, pour analyser la performance du site:

* **Les sources de trafic**
* **L’origine géographique des visiteurs**
* **Le comportement des utilisateurs sur le site**
* **Les achats enregistrés**

Les données sont mises à disposition par Google sur Kaggle, je vais extraire les informations qui me sont utiles et les structurer dans une base de données.

1. **Cahier des charges**

**Enjeux :**

La direction Marketing souhaite:

* améliorer la connaissance client
* dégager des leviers d’action pour améliorer les performances du site (acquisition de trafic, ergonomie du site pour réduire le taux de rebond, campagnes promotionnelles...)
* améliorer les performances des actions marketing

L’intégration de l’information client provenant de différentes sources: CRM, communication ...

est essentielle pour avoir une information client à 360°.

Cette connaissance client sera partagée au sein de l’entreprise ainsi que les indicateurs de

performance.

**Stratégie :**

* Historiser toutes les données de navigation et de transaction.
* Constituer une base relationnelle client : données structurées avec un schéma établi et des valeurs normalisées.
* Fournir un outil de visualisation des indicateurs de performance

**Résultats :**

Les informations attendues pour analyser les performances et les leviers d’action sont :

* Nombre de sessions, tendance, pics de trafic
* Canaux d’acquisition de trafic\*
* Taux de conversion (taux moyen : 3%), tendance, selon la source / la durée de session
* Taux de rebond (doit être inférieur à 50%), tendance, analyse par source
* Durée de session, analyse par source
* Nombre de pages visitées, analyse par source
* Top 5 des pages produit visitées
* Provenance géographique

\*Définition des canaux d’acquisition de trafic : *organique* = visites venant des moteurs de recherche, *direct* = via l’URL du site web, *référence* = visiteurs qui proviennent de liens depuis d’autres sites autres que les moteurs de recherche et les réseaux sociaux, *réseaux sociaux , payant*

**Contraintes :**

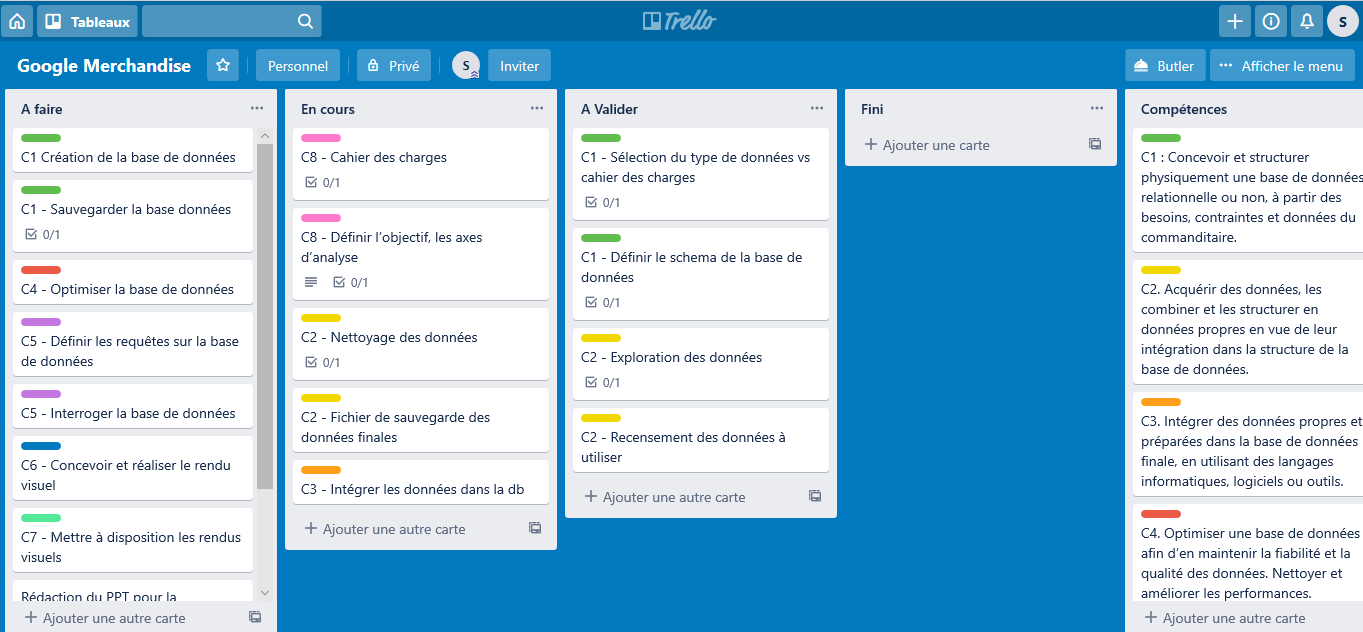
Les données mises à disposition sont épurées. Le montant des transactions est fourni mais la composition du panier par produit n’est pas fourni.

Les informations clients respectent la réglementation sur la protection des données individuelles (RGPD): collecte au préalable du consentement écrit, clair et explicite de l’internaute avant tout traitement de données personnelles et anonymisation.

# Langages et outils utilisés

|  |  |
| --- | --- |
| Nom outil | Utilisation |
| **Trello** | Gestion de projets agiles, permet de visualiser les tâches selon la méthode Kanban et de limiter le travail en cours |
| **Python** | Langage orienté objet pour traiter la Data et développe, en particulier avec pandas, et de développer des applications |
| **Jupyter**  **Notebook** | Application web pour exécuter du code: utilisé pour explorer les données, les traiter et les charger vers la base de données |
| **Pyzo** | IDE pour Python, permet de constituer des scripts directement exécutables |
| **MySQL** | SGBDR: Système de Gestion de Bases de Données Relationnelles qui stocke des données de façon organisée et cohérente |
| **MySQL Workbench 8.0** | IDE SQL pour la conception de la base de données, la visualisation via des requêtes et l’administration de la base de données |
| **Matplotlib / Seaborn** | Modules Python pour la visualisation des données |
| **Flask** | Framework open-source de développement web en Python,  permet de créer une application web pour visualiser les données |

# Gestion du projet



CHAPITRE 1 :

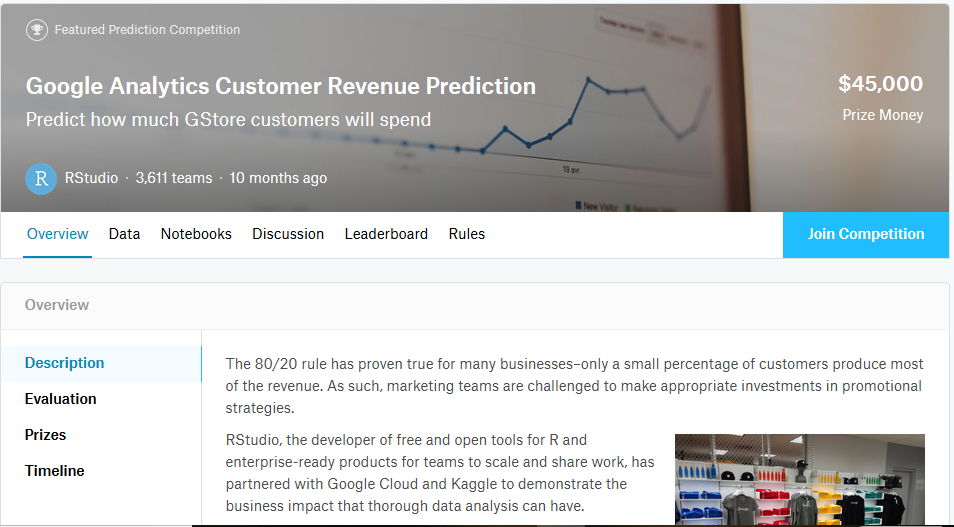
CONCEPTION ET CONSTRUCTION DE LA BASE DE DONNÉES

1. **Source des données**

Des données sont mises à disposition dans Kaggle par Google et proviennent de la boutique

[Google Merchandise Store](https://shop.googlemerchandisestore.com/?utm_source=Partners&utm_medium=affiliate&utm_campaign=Data%20Share%20Promo), le site e-commerce de Google qui vend les produits dérivés (sacs,

gourdes, vêtements, produits d’écritures…) des marques Google: Google, Android, YouTube…

****

Le concours sur Kaggle concerne la prédiction de revenu. J’ai utilisé le fichier ‘test\_v2.csv’ qui

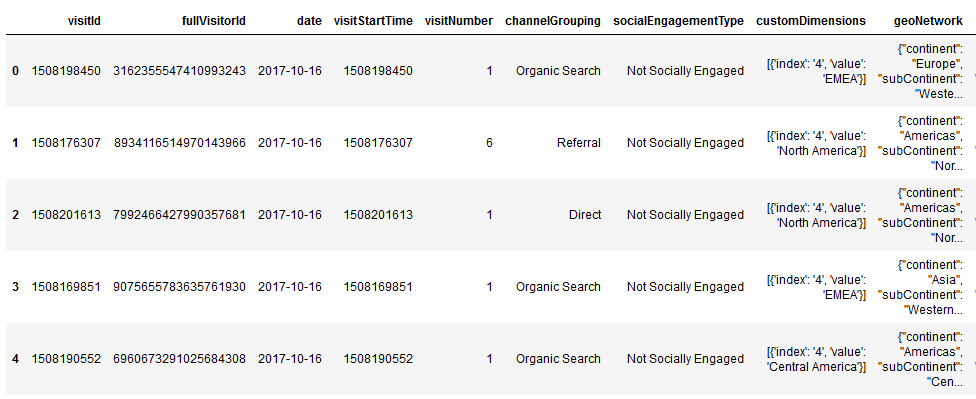
contient les informations de mai 2018 à mi-octobre 2018, c’est-à-dire 400 000

enregistrements.

1. **Description des données**

Les données sont constituées de 13 colonnes et 400 000 enregistrements ( “Visualisation des

inputs 1” et “Visualisation des inputs 2” ci-dessous).



**Visualisation des inputs 1**



**Visualisation des inputs 2**

Le détail complet des indicateurs Google Analytics utilisés dans le cadre de ce projet est en

Annexe 1.

7 colonnes correspondent à des attributs / valeurs:

* visitId: Identifiant défini pour une session. Il fait partie de la valeur généralement stockée sous la forme du cookie \_utmb. Il est propre à cet utilisateur. Pour obtenir un identifiant parfaitement unique, il faut combiner les valeurs des champs fullVisitorId et visitId.
* fullVisitorId: ID de visiteur unique
* date: Date de la session au format AAAAMMJJ
* visitStartTime: Horodatage (spécifié au format POSIX)
* visitNumber: Numéro de session associé à cet utilisateur. S'il s'agit de la première session, la valeur utilisée est 1
* channelGrouping: Canal d’acquisition de trafic
* socialEngagementType: Type d'interaction, soit "Intéressé sur le réseau social", soit "Pas intéressé sur le réseau social"

6 colonnes correspondent à des sections comportant de multiples informations, sous forme de

fichiers json représentant soit un objet (commençant par { et se terminant par }), soit un tableau

(commençant par [ et terminant par ]).

* customDimensions: contient des dimensions personnalisées, ici des informations géographiques concernant l'utilisateur.
* geoNetwork: contient des informations sur la zone géographique de l'utilisateur.
* trafficSource: contient des informations sur la source de trafic ayant généré la session.
* device: contient des informations sur les appareils de l'utilisateur.
* totals: contient les valeurs cumulées de la session sur la navigation ( rebond, durée, nombre de pages…) et les montants achetés
* hits: les champs imbriqués permettent d’accéder au détail des appels (= click), en particulier sur les pages visitées.

Je retiendrai une partie des données, en extrayant les attributs qui me sont utiles des sections

geoNetwork, device, totals et hits.

1. **Modèle conceptuel des données (MCD)**
   1. **Choix du type de base de données**

Même si la présence de fichiers json en input pourrait me conduire à constituer une base de

données NoSQL, je choisis de constituer une base de données relationnelle pour répondre aux

enjeux :

* constituer une base de données client avec un schéma établi et des valeurs normalisées
* contrôler l’intégrité des données

La base de données client Google respectera les principales caractéristiques attachées à une base

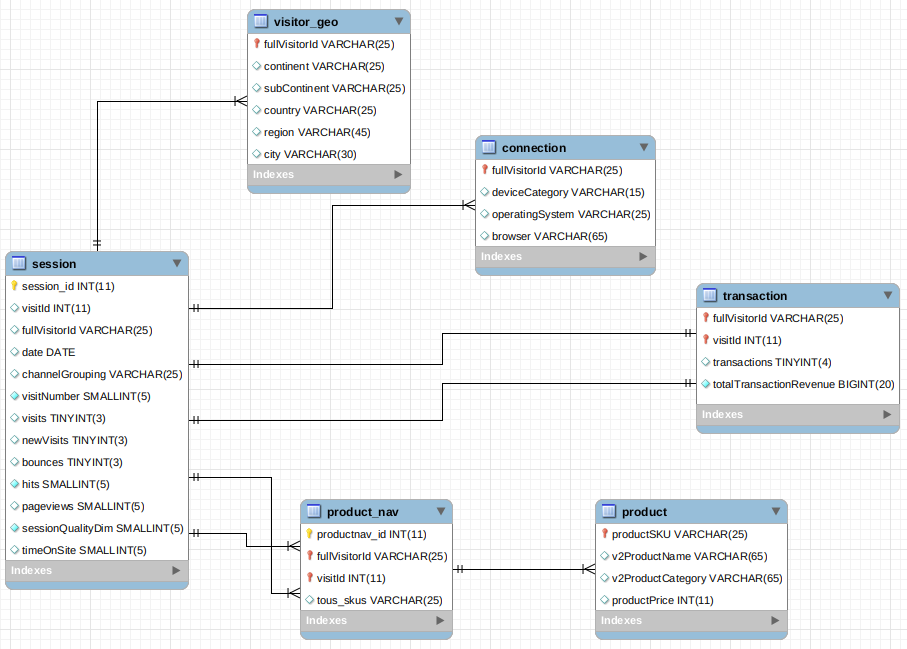
de données relationnelle :

* La structure de données est identifiable
* L’intégrité des données peut être contrôlée
* Le caractère transactionnel est fortement présent (eCommerce)
  1. **Optimisation de la base de données**

Je valide que le schéma respecte :

* la première forme normale (1FN), c’est-à-dire que chaque attribut soit **atomique**.
* la deuxième forme normale (2FN) concernant les tables avec une **clé primaire composite,** c’est-à-dire qu’aucun attribut ne faisant pas partie de la clé primaire ne doit dépendre que d'une partie de la clé primaire
* la troisième forme normale, c’est-à-dire qu’aucun attribut ne faisant pas partie de la clé primaire ne doit **dépendre d'une partie des autres attributs** ne faisant pas non plus partie de la clé primaire.

A partir des données sélectionnées du fichier en input, j’organise mes tables. Le schéma peut être visualisé ci-dessous **MCD de la base googleMerchandise:**



**MCD de la base googleMerchandise**

J’ai généré le schéma ci-dessus à l’aide de MySQL Workbench 8.0 via la fonctionnalité « Reverse

Engineering ».

* **table session**: chaque ligne correspond à une session de navigation et comporte les données quantitatives concernant la session, provenant de la section totals de l’input, ainsi que la source du trafic issue de channelGrouping

|  |  |
| --- | --- |
| Clé primaire | session\_Id, identifiant unique pour chaque ligne, sachant que fullVisitorId et visitId ne sont pas uniques |
| Index | sur visitId et fullVisitorId: les données sont organisées pour accélérer les requêtes SQL |

* **tables connection et visitor\_geo:** chaque ligne correspond à un client **=** fullVisitorId. Ce sont les données sur le type d’appareil utilisé pour se connecter, avec l’OS et le browser, ainsi que les données d’identification géographique attachées au client

|  |  |
| --- | --- |
| Clé primaire | fullVisitorId |
| Clé étrangère | fullVisitorId pour faire le lien avec la table session : c’est une relation one to many |

* **table transaction:** chaque ligne correspond à un client **=** fullVisitorId et une session, ce sont les données d’achat attachées au client et à la session, ici seuls le montant total des achats et le nombre d’article est disponible

|  |  |
| --- | --- |
| Clé primaire | Clé composite fullVisitorId et visitId |
| Clé étrangère | fullVisitorId avec la table session : c’est une relation one to one  visitId avec la table session : c’est une relation one to one |

* **table product\_nav:** chaque ligne correspond à un produit pour une session de navigation, elle permet d’identifier les pages produits vues lors de la session

|  |  |
| --- | --- |
| Clé primaire | productnav\_id, identifiant unique pour chaque ligne |
| Clé étrangère | fullVisitorId avec la table session, c’est une relation one to many  visitId avec la table session, c’est une relation one to many |
| Index | sur tous\_skus : les données sont organisées pour accélérer les requêtes SQL |

* **table product:** chaque ligne correspond à un produit du catalogue de produits Google et contient le descriptif du produit

|  |  |
| --- | --- |
| Clé primaire | productSKU |
| Clé étrangère | productSKU, pour faire le lien avec la table product\_nav : c’est une relation one to many |

1. **Exploration et extraction des données**

Sur **Jupyter Notebook,** j’ai lu le fichier CSV et l’ai transformé en dataframe pour faire l’exploration

de données via **Pandas**.

Le fichier comportant 400 000 enregistrements, j’ai utilisé un échantillon aléatoire de 25% des

données.

J’ai regardé la proportion de valeurs à NA ainsi que la distribution des valeurs pour éliminer les

informations non pertinentes.

Les colonnes d’origine conservées sont:

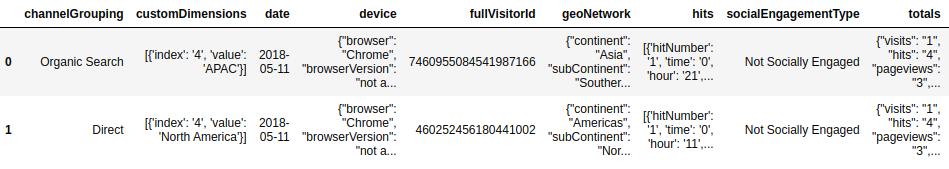
* visitId : Identifiant défini pour une session
* fullVisitorId : ID de visiteur unique
* date : Date de la session
* visitNumber : Numéro de session associé à l’utilisateur
* channelGrouping : Canal d’acquisition de trafic

Les sections conservées sont:

* totals : contient les valeurs cumulées de la session de navigation (rebond, durée, nombre de pages…) et les montants achetés.
* geoNetwork : contient des informations sur la zone géographique de l'utilisateur.
* device : contient des informations sur les appareils de l'utilisateur.
* hits : les champs imbriqués permettent d’accéder au détail des appels (= click), en particulier sur les pages visitées. Je n’ai conservé qu’une petite partie de cette section, en particulier celle qui permet d’identifier les pages produit visitées.

1. **Traitement d’un premier dataframe avec visitId, fullVisitorId, date, visitNumber, channelGrouping et les sections geoNetwork, device et totals:**

Dans le cas des sections geoNetwork, device et totals, les informations sont inclues dans des objets json constitués de clés / valeurs :



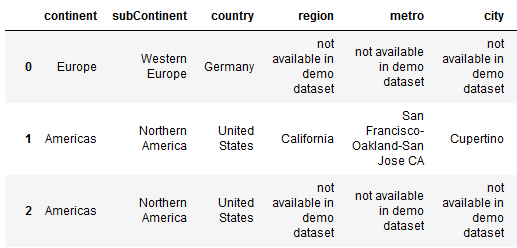
**Contenu des colonnes en input**

J’ai affiché les champs des json en colonne dans le dataframe en utilisant la fonction apply

(json.loads) et la méthode tolist(). Voici pour exemple l’extraction des données de la section geoNetwork:



Et la restitution sur le dataframe:



Je sélectionne les attributs qui constitueront ensuite mes tables **session, connection,**

**visitor\_geo et transaction.**

Via drop\_duplicates, je supprime les doublons des informations de connection et de provenance géographique des dataframes spécifiques qui me serviront pour les tables.

Je supprime les transactions à NA du dataframe transaction via .dropna() avant de constituer la table.

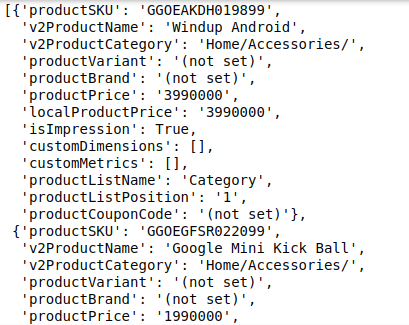
1. **Traitement de visitId, fullVisitorId avec la section hits:**

Pour hits, le json est constitué d’un tableau comportant 23 clés/valeurs. Il répertorie l’ensemble

des actions de navigation.



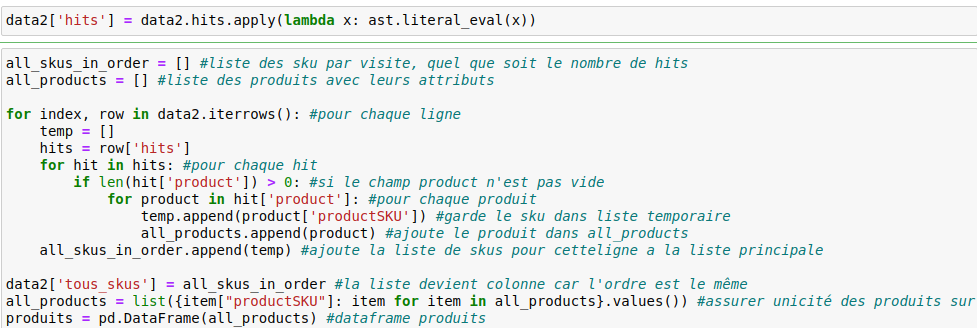
Je m’intéresse à la clé “product” de laquelle je vais extraire toutes les sous-sections et conserver les attributs qui m’intéressent: productSKU, v2ProductName, v2ProductCategory et productPrice.



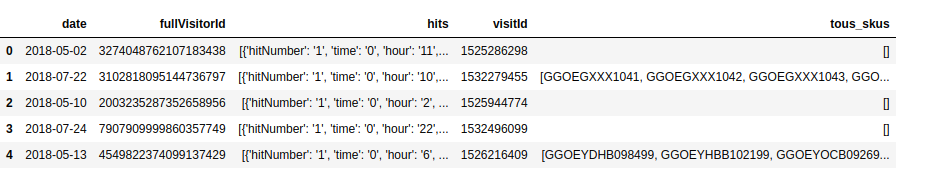
Je vais utiliser productSKU pour créer une colonne ‘tous\_skus’ qui liste tous les produits associés à une session.

Tous les attributs de ‘product’ vont être extraits pour constituer un dataframe.

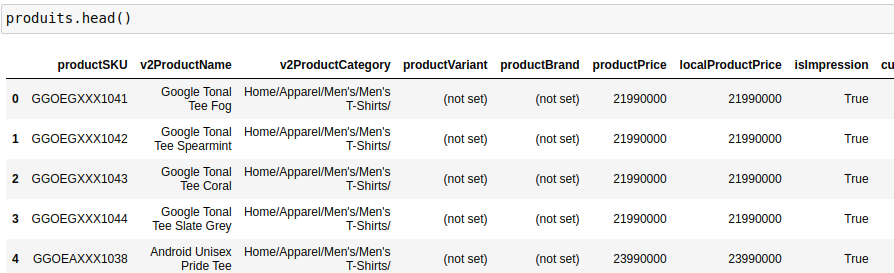
Les lignes de code ont été faites par Ioan Hodor, apprenant de la promo DDN avec moi :



Le dataframe comporte une nouvelle colonne ‘tous\_skus’ avec tous les productSKU par session:



Le nouveau dataframe **produits** comporte tous les produits du site avec leurs caractéristiques:

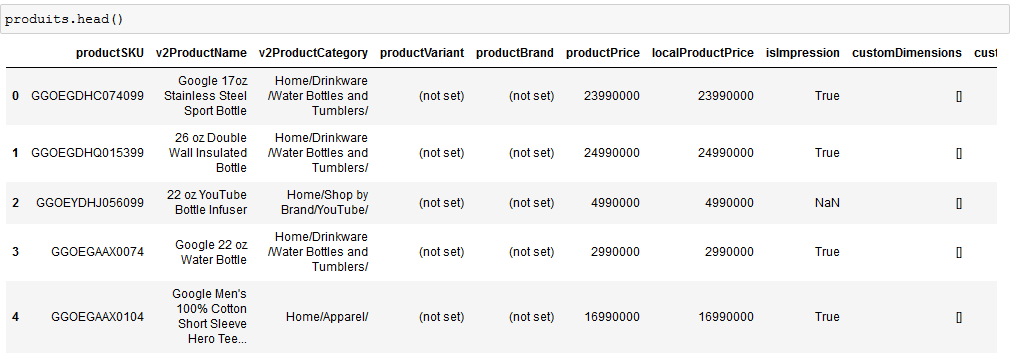
****

* 1. Pour pouvoir obtenir ensuite la table **product\_nav** je constitue un dataframe dont chaque ligne représente un couple session / produit:

****



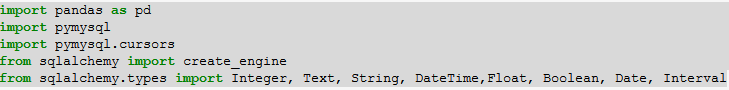
* 1. Pour obtenir la table **product** à partir du dataframe produits, je ne vais conserver que les colonnes qui m’intéressent et supprimer les lignes productSKU en doublon.



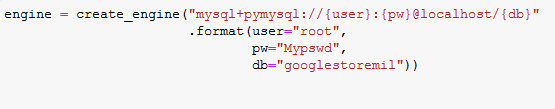
1. **Importation des données vers MySQL**

J’utilise **pymysql et sqlalchemy** pour faire la connexion vers MySQL :

Les imports des modules nécessaires:

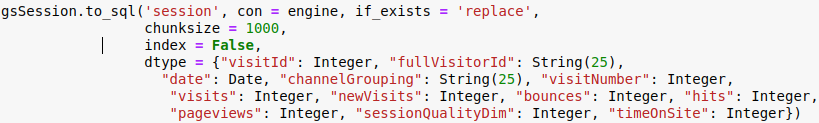


La connexion:

****

Après création de la database sur MySQL, j’utilise la fonction .to\_sql() pour créer les tables.

Exemple pour la table session:



1. **Intégrité et Sécurisation de la base de données**

Les contraintes appliquées vont permettre d’assurer l’intégrité des informations :

1. **Les clés primaires** :

Elles permettent d'identifier chaque enregistrement dans une table de base de données.

Par exemple pour la table session la clé primaire est session\_Id que j’ai créée pour avoir un identifiant unique pour chaque ligne, sachant que chaque client (=fullVisitorId) peut avoir plusieurs sessions et qu’un même code de visitId peut être affecté à plusieurs clients.

1. **Les clés étrangères** :

Elles permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données.

1. **Les datatypes :**

Ils spécifient le type de données que l'objet peut contenir : nombre de type entier ou

décimal, chaînes de caractères, données de date et d'heure, etc

Avec la fonction df.to\_sql, j’ai pu définir les datatypes des colonnes, en prenant les valeurs des types de SQLAlchemy.

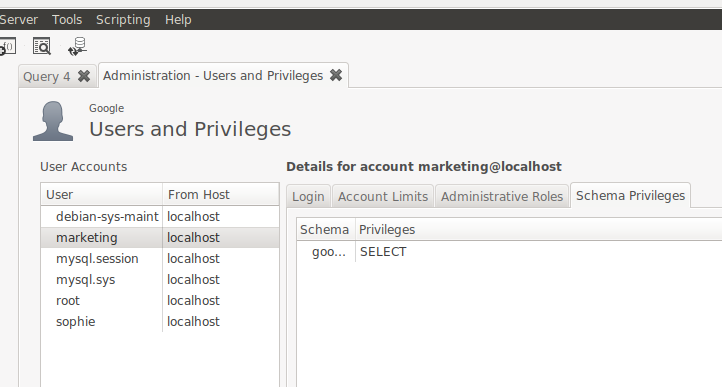
Sur MySQL je peux optimiser en particulier en utilisant tinyint ou smallint au lieu de int.

Pour fullVisitorId, j’ai conservé un datatype VARCHAR même si c’est une suite de chiffres car le type Integer ou Float créait une confusion.

La sécurisation de la base de données :

1. **Création de compte utilisateurs** “marketing” via Workbench Server > User and Privilege

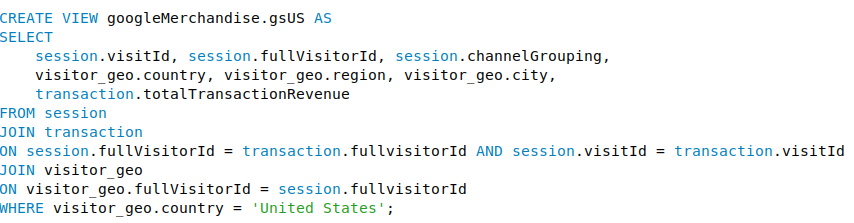
On peut ainsi permettre l’accès à la base et aux données en consultation uniquement et restreindre la mise à jour des données pour les utilisateurs habilités.



1. **Création de vues**

Elles permettent de sélectionner une partie des informations pour restreindre l’accès à une ou plusieurs tables selon la sensibilité des données.

Ci-dessous la vue gsUS permet de restreindre l’accès aux données de chiffre d’affaires par pays:



1. **Sauvegarde de la base de données** via Workbench ou via mysqldump en ligne de commande
   1. Sur workbench, Server > Data Export

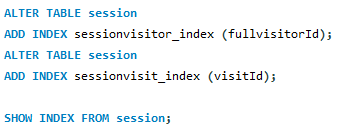
création d’un dossier /home/sophie/dumps/Dump20200101

* 1. En ligne de commande sur Ubuntu:

$ mysqldump -u sophie -p googleMerchandise > /home/sophie/Documents/Certif/googleMerchandise\_backup.sql

Il est aussi possible de planifier des sauvegardes régulières en utilisant le package automysqlbackup

1. **Sauvegarde des instructions** concernant les contraintes appliquées aux tables : Primary Key, Index, Foreign Key, datatypes dans des fichiers .sql
2. **Optimisations**
3. **Optimisation de la base de données** :
   1. Création d’Index: L’index est utile pour accélérer l’exécution d’une requête SQL qui lit des données et ainsi améliorer les performances d’une application utilisant une base de données.



1. **Optimisation des requêtes** : Création de vues

La vue représente une sorte d'intermédiaire entre la base de données et l'utilisateur. Elle permet en plus de l’aspect sécurité des données décrit plus haut :

* de faciliter l’accès à une sélection de données qui peuvent être issues d’une requête complexe avec de multiples tables jointes.
* de regrouper des informations au sein d'une entité.

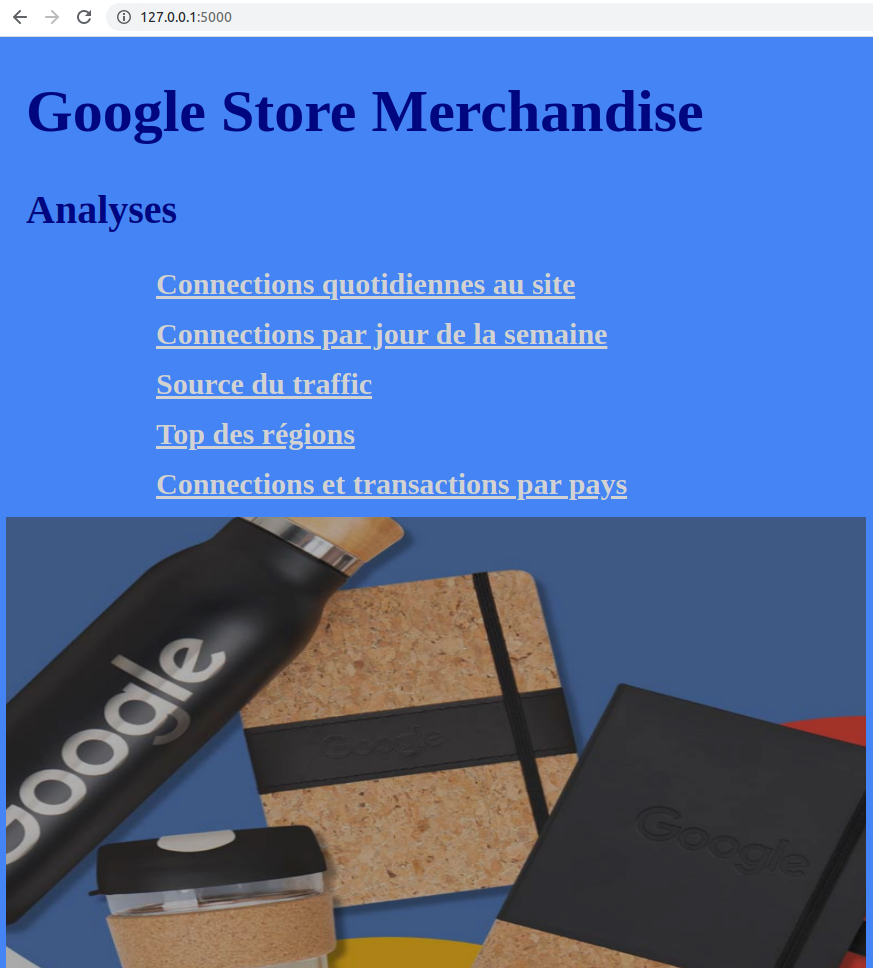
**CHAPITRE 2 :**

ANALYSE DE DONNÉES ET VISUALISATION

1. **Rappel des objectifs et outils**

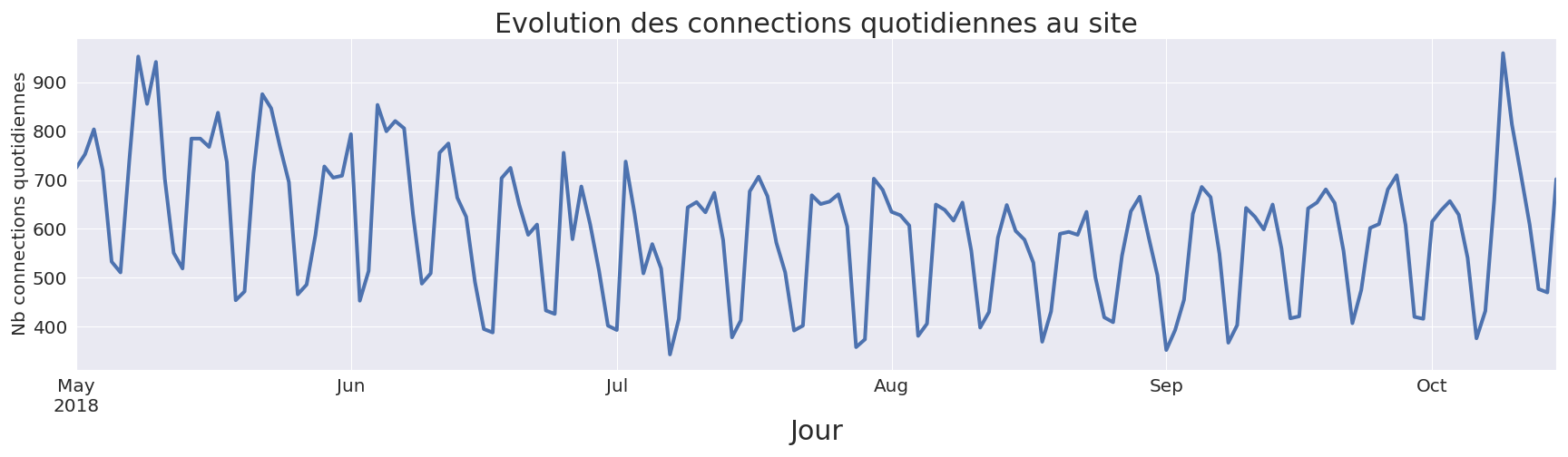
L’objectif est de comprendre le comportement des visiteurs sur le site et d’analyser les performances du site.

J’ai réalisé les analyses sur Python, fait la visualisation avec Matplotlib et Seaborn, avec une mise à disposition des graphiques sur une application web avec Flask.

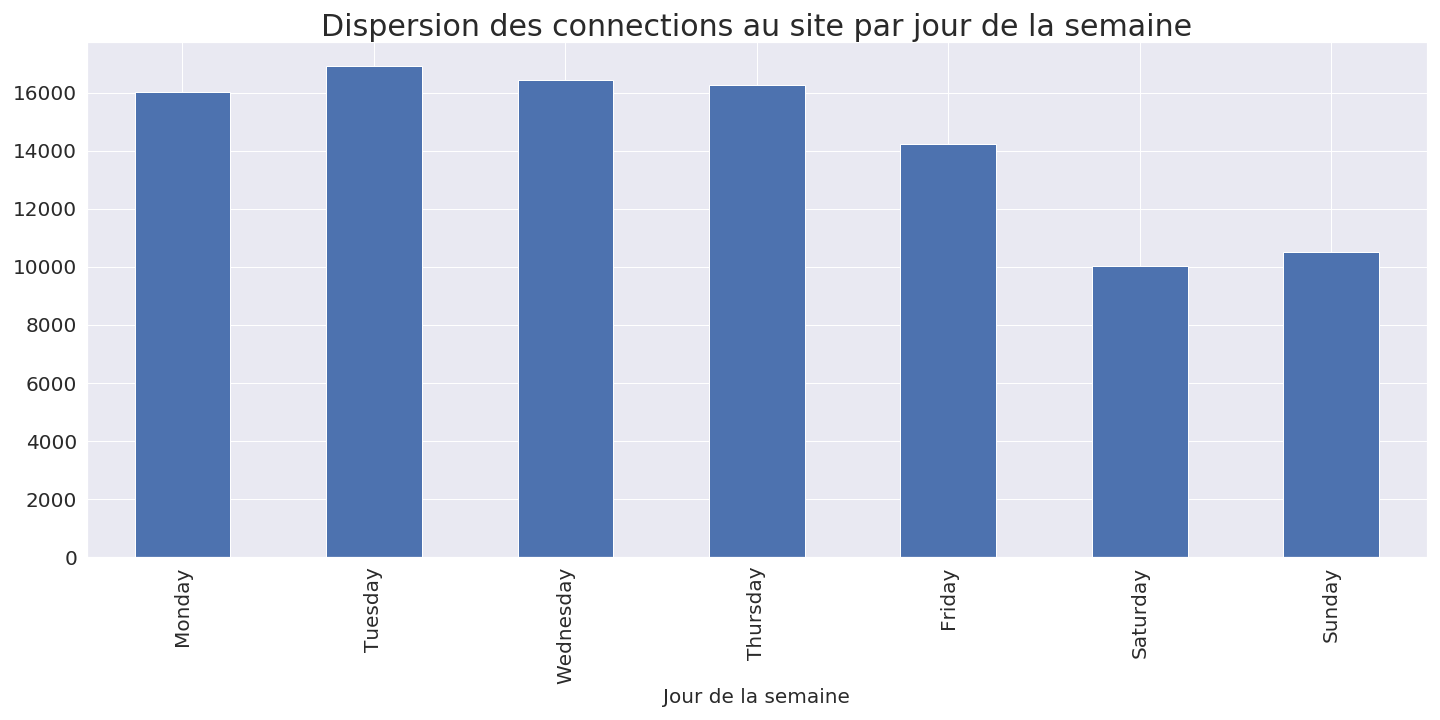


1. **Visualisation**

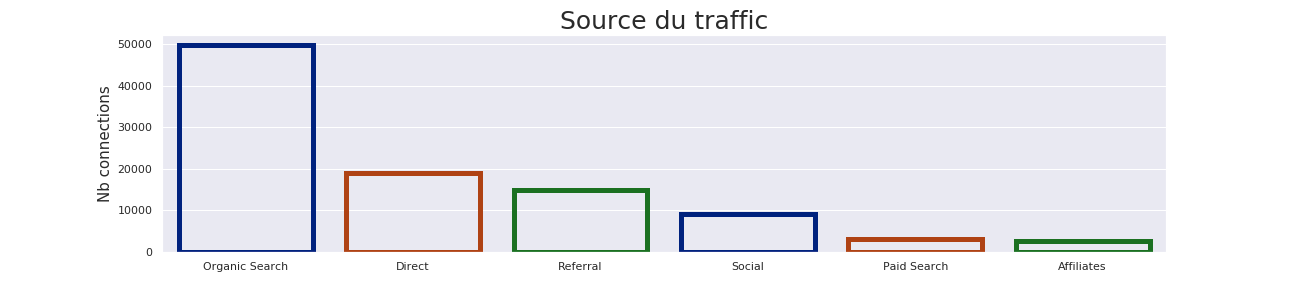
Le nombre de connexions au site est en moyenne de 600 par jour avec quelques pics d’activité autour de 900.



Les creux d’activité sur le site correspondent aux samedi et dimanche.

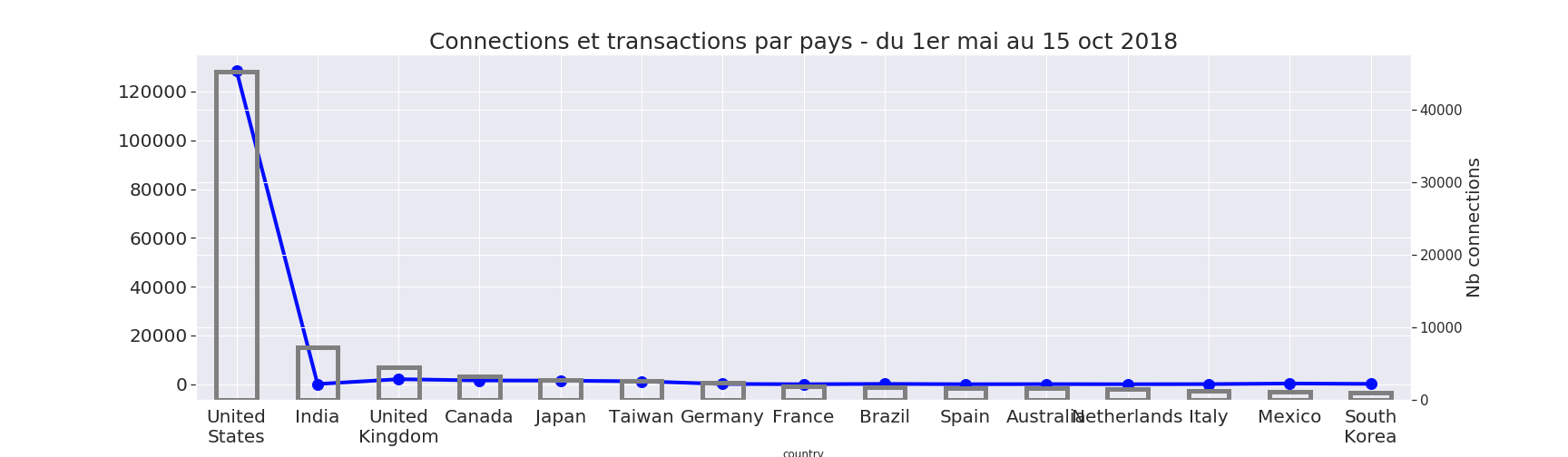


Le trafic vient en premier de l’ « Organic Search »: trafic qui provient des résultats naturels issus des moteurs de recherche reconnus par Google. Il y a un bon SEO mis en place sur le site de Google Merchandise Store!

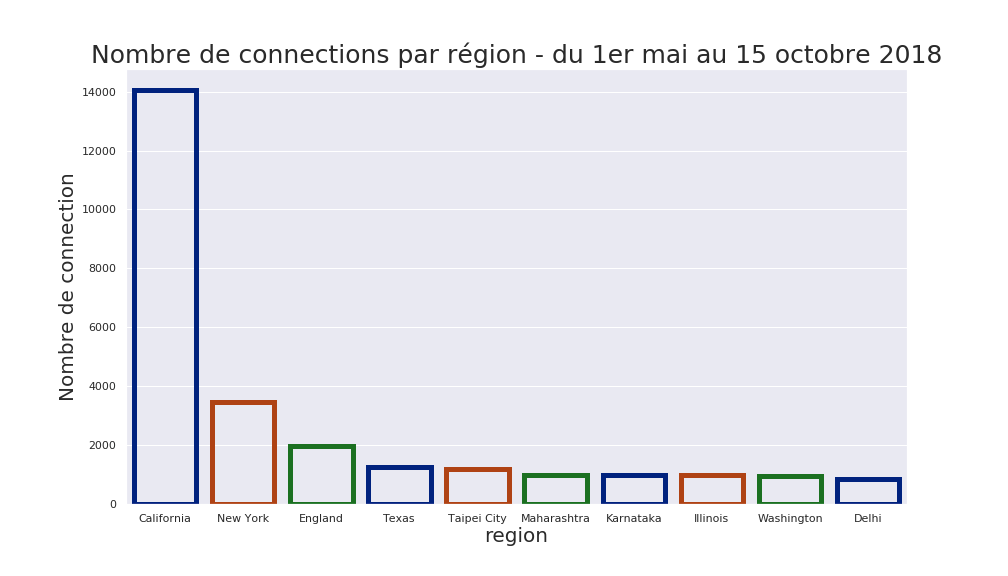


Les Etats-Unis sont les premiers contributeurs en nombre de visiteurs et en chiffre d’affaires.

L’Inde est le 2eme pays en terme de connexions mais le chiffre d’affaires est relativement faible.



La Californie et New York sont les régions avec le plus de connexions.



BILAN DE PROJET ET PISTES D'AMÉLIORATION

Le projet m’a permis d’approfondir mes connaissances sur Google Analytics.

J’ai pu continuer à me perfectionner pleinement sur Python pour le nettoyage, l’exploration et la

visualisation des données avec Flask.

La constitution de la base de données m’a permis d’avoir une expérience concrète avec toutes les

subtilités de SQL.

En terme de gestion de projet, j’ai totalement sous estimé le temps à consacrer à l’exploration et le nettoyage d’un nouveau jeu de données.

Je tiens à remercier nos professeurs Simplon: Manel Boumaiza, Sayf Bejaoui, David Azria et Yacine Aslim. La pédagogie Simplon et leur accompagnement pendant les 7 mois de la formation m’a permis de réaliser ce projet et d’acquérir les compétences pour aller plus loin à l’issue de la formation.

Je remercie également Ioan Hodor pour son aide sur l’extraction des informations imbriquées dans les json Google Analytics.

Pistes d’amélioration:

Pour aller plus loin je souhaite construire le process pour actualiser la base de façon hebdomadaires avec les nouvelle connexions.

Je voudrais également travailler sur la segmentation des clients à partir de comportements

homogènes sur le site. Cela permettrait d’établir des cibles et les leviers marketing

correspondants afin d’optimiser les investissements.

ANNEXES

**ANNEXE 1 – LES INDICATEURS GOOGLE ANALYTICS**

|  |  |  |
| --- | --- | --- |
| **Nom de la colonne** | **Datatype**  **/Format** | **Description / Explication** |
| fullVisitorId | Texte | ID de visiteur unique |
| visitId | Nombre entier | Identifiant défini pour cette session |
| visitNumber | Nombre entier | Numéro de session associé à cet utilisateur. S'il s'agit de la première session, la valeur utilisée est 1. |
| date | Chaîne de caractères | Date de la session au format AAAAMMJJ. |
| channelGrouping | Texte | Groupe de canaux par défaut associé à la session d'un utilisateur final pour cette vue. |
| totals |  | Contient les valeurs cumulées de la session de navigation ( rebond, durée, nombre de pages…) et les transactions |
| totals.bounces | Nombre entier | Pour une session avec rebond, la valeur est de 1. Sinon, elle est nulle. |
| totals.hits | Nombre entier | Nombre total d'appels durant la session. |
| totals.newVisits | Nombre entier | S'il s'agit de la première visite, la valeur est égale à 1. Sinon, elle est nulle. |
| totals.pageviews | Nombre entier | Nombre total de pages vues au cours de la session. |
| totals.sessionQualityDim | Nombre entier | Estimation indiquant dans quelle mesure une session spécifique était proche de générer une transaction. Ce chiffre est calculé pour chaque session. Il est compris entre 1 et 100. Une valeur proche de 1 indique une faible qualité de session ou une session qui n'était pas en passe de générer une transaction, tandis qu'une valeur proche de 100 indique une qualité de session élevée ou une session qui était tout près de générer une transaction. Une valeur de 0 indique que la qualité de la session n'est pas calculée pour la plage de dates sélectionnée. |
| totals.timeOnSite | Nombre entier | Durée totale de la session (en secondes). |
| totals.transactions | Nombre entier | Nombre total de transactions de commerce électronique au cours de la session. |
| totals.totalTransactionRevenue | Nombre entier | Chiffre d'affaires total des transactions. Il correspond à la valeur transmise à Analytics, multipliée par 10^6 (par exemple, la valeur 2,40 est restituée sous la forme "2 400 000") |

|  |  |  |
| --- | --- | --- |
| **Nom de la colonne** | **Datatype**  **/Format** | **Description / Explication** |
| device |  | Cette section contient des informations sur les appareils de l'utilisateur. |
| device.browser | Chaîne de caractères | Navigateur utilisé (par exemple, "Chrome" ou "Firefox"). |
| device.deviceCategory | Chaîne de caractères | Type d'appareil (ordinateur de bureau, tablette ou appareil mobile). |
| device.operatingSystem | Chaîne de caractères | Système d'exploitation de l'appareil (par exemple, "Macintosh" ou "Windows"). |
| geoNetwork |  | Cette section contient des informations sur la zone géographique de l'utilisateur. |
| geoNetwork.continent | Chaîne de caractère | Continent à l'origine des sessions, déterminé par l'adresse IP. |
| eoNetwork.subContinent | Chaîne de caractère | Sous-continent à l'origine des sessions, déterminé par l'adresse IP du visiteur. |
| geoNetwork.country | Chaîne de caractère | Pays à l'origine des sessions, déterminé par l'adresse IP. |
| geoNetwork.region | Chaîne de caractère | Région à l'origine des sessions, déterminée par l'adresse IP. Aux États-Unis, une région correspond à un État (celui de New York, par exemple). |
| geoNetwork.city | Chaîne de caractère | Ville de l'utilisateur, déterminée par son adresse IP ou l'ID géographique. |
| hits.product |  | Cette ligne et les champs imbriqués seront renseignés pour chaque appel qui contient des données PRODUCT |
| hits.product.productSKU | Chaîne de caractère | Code produit. |
| hits.product.v2ProductCategory | Chaîne de caractère | Catégorie du produit. |
| hits.product.v2ProductName | Chaîne de caractères | Nom du produit. |
| hits.product.productPrice | Nombre entier | Prix du produit. Il correspond à la valeur transmise à Analytics, multipliée par 10^6 (par exemple, la valeur 2,40 est restituée sous la forme "2 400 000"). |