

Sélection des données et cartographie des opportunités



Projet : Amazon Review Analysis

Auteur : Amara NAIT SAIDI

Date : 20 octobre 2025

Version 1.0

A propos de ce document

L'objectif de ce document est de sélectionner les données pertinentes afin de mener à bien une première analyse de notre cas d'utilisation.

Table des matières

1.	Introduction	4
2.	Contexte et objectifs du projet	4
2.1.	Objectifs techniques.....	4
2.2.	Périmètre.....	4
3.	Sélection des données pertinentes	5
3.1.	Tables principales	5
3.2.	Justification de la sélection des tables	5
3.3.	Dictionnaire des données.....	6
3.3.1.	Table REVIEW	6
3.3.2.	Table PRODUCT	6
3.3.3.	Table CATEGORY	6
3.3.4.	Table ORDERS	7
3.3.5.	Table REVIEW_IMAGES	7
3.3.6.	Table PRODUCT_REVIEWS	7
3.3.7.	Relation entre les tables et cardinalités	7
4.	Conclusion.....	7
5.	Références	8

1. Introduction

L'objectif de ce document est d'identifier et de sélectionner les données pertinentes issues de la base de données transactionnelle d'Amazon, afin de répondre à notre cas d'usage centré sur la pertinence des avis clients. L'enjeu principal est de déterminer quelles tables, champs et types de données sont nécessaires pour alimenter notre solution analytique et permettre une meilleure compréhension du comportement des utilisateurs vis-à-vis des produits.

Dans un premier temps, les données identifiées serviront à la conception d'un prototype fonctionnel. Ce prototype aura pour but de tester et de valider nos hypothèses sur la pertinence des avis, notamment en évaluant la capacité de notre modèle à isoler les commentaires les plus utiles et représentatifs pour chaque produit.

2. Contexte et objectifs du projet

Le projet vise à analyser et classifier automatiquement les avis clients laissés sur les produits de la plateforme Amazon. L'objectif est de développer un système capable d'identifier les avis les plus pertinents et de les catégoriser par thèmes afin d'améliorer la qualité des produits et la satisfaction client.

2.1. Objectifs techniques

Concevoir une architecture de données permettant l'extraction, le traitement et l'analyse des avis clients. Le système doit gérer la compatibilité des données, assurer la qualité, traiter les données rejetées, et respecter les normes réglementaires (RGPD, CNIL, CCPA).

2.2. Périmètre

Source unique : Base de données PostgreSQL contenant 25 tables.

Volumétrie totale : 1 888 728 enregistrements répartis sur l'ensemble des tables.

3. Sélection des données pertinentes

3.1. Tables principales

Dans notre cas d'utilisation, nous avons identifié les tables suivantes :

Table	Volumétrie	Rôle	Données extraites
REVIEW	111 322	Table centrale	Texte, titre, note
PRODUCT_REVIEWS	111 322	Liaison	Association avis-produit
PRODUCT	42 858	Contexte	Nom, description, prix
CATEGORY	2	Catégorisation	Catégorie produit
ORDERS	222 649	Enrichissement	Commandes passées
REVIEW IMAGES	119 382	Enrichissement	Images des avis

Note : nous aurions aimé exploiter la table SHIPMENT dans notre projet, toutefois, la volumétrie de cette table ne nous a pas permis d'aller plus loin. La table ne contient que 5 lignes.

3.2. Justification de la sélection des tables

Pour répondre à notre cas d'utilisation (automatiser le tri et la classification des avis), nous avons priorisé les tables qui apportent le contenu textuel, le contexte produit, et les enrichissements utiles pour ce cas d'usage. La sélection des tables pertinentes s'est faite sur la base des données nécessaire au calcul du score de pertinence qui combine plusieurs paramètres.

Justification des tables sélectionnées et logique d'intégration :

Pour répondre au besoin de classification et d'identification des avis pertinents, seules les tables réellement utiles à la construction d'un enregistrement complet et exploitable ont été retenues. La table **REVIEW** constitue le point d'entrée : elle concentre le texte, le titre, la note et l'identifiant de l'auteur. Autour de cette table centrale, plusieurs jointures permettent de reconstruire un contexte riche et structuré autour de chaque avis.

La table **PRODUCT_REVIEWS** lie chaque avis à un produit, rendant possible la récupération des informations issues de la table **PRODUCT**, telles que le nom ou les caractéristiques du produit. La table **CATEGORY**, jointe en complément, permet d'ajouter un niveau de catégorisation utile pour segmenter les analyses et enrichir le modèle par des variables plus générales sur le type de produit.

La table **REVIEW_IMAGES** ajoute un indicateur simple mais important : la présence ou non d'une image dans un avis. Cet élément constitue un signal de qualité souvent exploité dans les modèles de pertinence. De son côté, la table **ORDERS** permet de vérifier si l'auteur de l'avis a réellement passé une commande, offrant un indicateur d'authenticité particulièrement discriminant (acheteur vérifié vs non vérifié).

Cette approche garantit un usage ciblé et cohérent des données, en se limitant aux tables réellement nécessaires pour reconstruire un contexte complet autour de l'avis, tout en assurant la performance et la lisibilité du prototype.

3.3. Dictionnaire des données

Le dictionnaire de données détaille la structure des tables principales utilisées pour l'algorithme de classification et le calcul de score. Pour chaque table, sont précisés : nom de colonne, type de données, format attendu, contraintes d'intégrité et description fonctionnelle.

3.3.1. Table REVIEW

Colonne	Type	Contraintes	Description
REVIEW_ID	INTEGER	PK, NOT NULL	Identifiant unique de l'avis
BUYER_ID	INTEGER	FK, NOT NULL	Référence vers BUYER
DESC	TEXT	NOT NULL	Texte complet de l'avis
TITLE	VARCHAR(150)	NOT NULL	Titre de l'avis
RATING	INTEGER	CHECK (1-5)	Note de 1 à 5 étoiles !
SELLER_PRODUCT_FLAG	BOOLEAN	NOT NULL	0=Produit, 1=Vendeur

3.3.2. Table PRODUCT

Colonne	Type	Contraintes	Description
P_ID	INTEGER	PK, NOT NULL	Identifiant unique produit
P_NAME	VARCHAR(200)	NOT NULL	Nom du produit
DESC	TEXT	NULL	Description détaillée
PRICE	DECIMAL(10,2)	NOT NULL	Prix unitaire
QTY	INTEGER	NOT NULL	Quantité en stock
CATEGORY_ID	INTEGER	FK, NOT NULL	Référence vers CATEGORY

3.3.3. Table CATEGORY

Colonne	Type	Contraintes	Description
CATEGORY_ID	INTEGER	PK, NOT NULL	Identifiant catégorie
NAME	VARCHAR(100)	NOT NULL, UNIQUE	Nom de la catégorie
DESC	TEXT	NULL	Description catégorie

3.3.4. Table ORDERS

Colonne	Type	Contraintes	Description
ORDER_ID	INTEGER	PK, NOT NULL	Identifiant ORDER
BUYER_ID	VARCHAR(40)	FK, NOT NULL	Référence vers BUYER
DISCOUNT_ID	INTEGER	FK, NOT NULL	Référence vers DISCOUNT
PAYMENT_ID	INTEGER	FK, NOT NULL	Référence vers PAYMENT
ORDER_DATE	DATE	NOT NULL	Date de la commande

3.3.5. Table REVIEW_IMAGES

Colonne	Type	Contraintes	Description
REVIEW_ID	INTEGER(4)	FK, NOT NULL	Identifiant review
REVIEW_IMAGE	VARCHAR(250)	NOT NULL	Url image

3.3.6. Table PRODUCT_REVIEWS

Colonne	Type	Contraintes	Description
P_ID	VARCHAR(10)	FK, NOT NULL	Identifiant PRODUCT
REVIEW_ID	INTEGER	FK, NOT NULL	Identifiant REVIEW

3.3.7. Relation entre les tables et cardinalités

- BUYER (1 : N) REVIEW.
- REVIEW (N : N) PRODUCT via PRODUCT_REVIEWS.
- PRODUCT (N : 1) CATEGORY.
- ORDERS (N : 1) BUYER.
- ORDERS (0 : N) DISCOUNT.
- ORDERS (N : 1) PAYMENT.
- REVIEW (1 : N) REVIEW_IMAGES.

La hiérarchie de catégories est limitée (2 catégories dans la base actuelle).

Unicité garantie par les clés primaires ((REVIEW_ID, BUYER_ID, P_ID))

4. Conclusion

La sélection et la cartographie des données constituent une phase essentielle dans la mise en œuvre du projet sur la pertinence des avis Amazon. Cette étape a permis de mieux comprendre la structure de la base de données, d'en identifier les tables et relations clés, et de déterminer les jeux de données à exploiter pour répondre efficacement à notre cas d'usage.

Les informations recueillies serviront de fondation au développement du prototype, qui permettra de tester la faisabilité technique et analytique du modèle de classification des avis. Ce prototype jouera un rôle central dans la validation de nos hypothèses et dans la préparation d'une solution finale plus robuste et industrialisable.

5. Références

- [Functional Requirements](#)
- [EER Diagram](#)
- [Relational Schema](#)