SPÉCIFICATIONS FONCTIONNELLES ET TECHNIQUES

Projet Amazon Reviews :

Analyse et classification des avis Clients

BONNAT Jonathan

Data Engineer - B2B-LP-DESFL

💬 INTRODUCTION GENERALE

Ce document présente les spécifications techniques et fonctionnelles du projet Amazon Reviews : Analyse et classification des avis clients.

Il décrit les fonctionnalités attendues, les exigences techniques et les contraintes associées au déploiement d'une solution de valorisation automatisée des avis clients au sein de l'écosystème Amazon.

L'objectif global du projet est d'identifier, classifier et hiérarchiser les avis les plus pertinents parmi un grand volume de retours clients.

Le système repose sur l'intégration d'un modèle de classification zero-shot pré-entraîné, combiné à des règles de scoring de pertinence fondées sur plusieurs indicateurs (longueur du texte, présence d'image, statut d'abonnement, note attribuée…).

Ce document sert de référence fonctionnelle et technique avant la phase finale de développement et de validation du système.

🧱 PÉRIMÈTRE FONCTIONNEL

FONCTIONS INCLUSES

FONCTIONS	DESCRIPTION	DONNÉES EN ENTRÉE → DONNÉES EN SORTIE
Extraction des avis	Connexion au serveur PostgreSQL et récupération des tables pertinentes	Identifiants, texte des avis, métadonnées produit → Dataframe consolidé
Prétraitement des données	Nettoyage, normalisation, et enrichissement des données	Données brutes extraites → Données normalisées avec champs calculés
Classification thématique	Attribution automatique d'une catégorie d'avis via un modèle zero-shot	Textes d'avis → Catégorie et score de confiance du modèle
Scoring de pertinence	Calcul d'un score global basé sur la longueur du texte, les mots-clés, la présence d'image, l'abonnement et la note	Variables textuelles et binaires → Score de pertinence pour chaque avis
Analyse exploratoire	Visualisation des distributions, scores moyens et corrélations entre critères	Données enrichies → Graphiques et statistiques
Export et visualisation	Génération d'un rapport et sauvegarde des résultats au format .csv ou graphique	Résultats finaux → Fichiers d'export

FONCTIONS EXCLUES

Certaines fonctionnalités ne sont pas incluses dans ce prototype :

- Apprentissage supervisé ou fine-tuning d'un modèle existant
- Analyse multilingue des avis (uniquement en anglais)
- Analyse en temps réel (traitement batch uniquement)
- Gestion des faux avis ou modération automatique
- Interface utilisateur ou intégration front-end
- Connexion directe à l'environnement de production AWS

HYPOTHÈSES ET CONTRAINTES

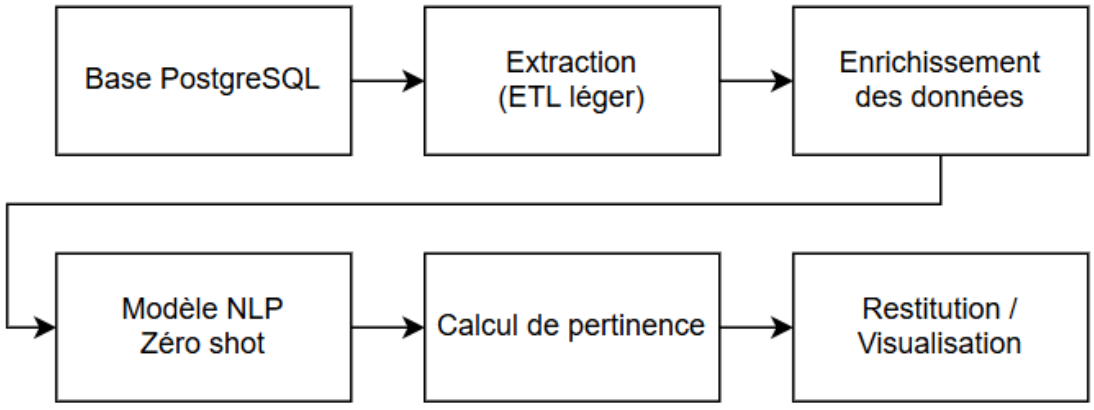
- Le dataset est figé (lecture seule depuis la base PostgreSQL).
- Tous les avis proviennent de clients authentifiés Amazon.
- Le modèle zero-shot opère sans supervision ni apprentissage local.
- Le scoring repose sur des pondérations fixes, définies arbitrairement.
- Le traitement est exécuté sur CPU (option GPU non activée dans le prototype).

🧠 ARCHITECTURE FONCTIONNELLE ET TECHNIQUE

SCHÉMA GLOBAL DE L'ARCHITECTURE

L'architecture fonctionnelle repose sur un pipeline analytique orienté vers l'exploitation des avis clients d'Amazon.

Elle est conçue de manière modulaire, assurant la transmission des données transactionnelles vers un module d'analyse sémantique basé sur le traitement automatique du langage (NLP).



Chaque composant a été conçu pour rester interchangeable et scalable, afin de pouvoir être intégré ultérieurement à un environnement cloud (AWS, GCP, ou autre).

DESCRIPTION DES COMPOSANTS

- **Base de données (PostgreSQL) :**

  La base transactionnelle fournie contient les tables utilisées pour notre problématique (review, product, orders, subscription et review_images). Les données sont extraites en lecture seule, puis nettoyées et jointes selon les besoins du use case (“Analyse de la pertinence des avis clients”).

- **Pipeline de traitement (Python) :**

  Le traitement repose sur un pipeline Python utilisant les librairies pandas, numpy et psycopg2 pour la connexion et la manipulation des données. Ce pipeline effectue l’extraction SQL, l’enrichissement des avis (longueur du texte, présence d’image, statut abonné, etc) et le calcul des scores intermédiaires nécessaires au calcul de pertinence.

- **L’utilisation du modèle NLP (Zero-Shot Classification) :**

  Le modèle utilisé est DistilBART MNLI via la bibliothèque Transformers , développée par Hugging Face. Il permet de catégoriser les avis sans entraînement supervisé, selon les labels définis : “product quality”, “delivery issue”, “customer service”, “product defect”, “general satisfaction”, “price value” et “other”.

- **Stockage et restitution des résultats :**

  Les résultats du modèle sont stockés dans le DataFrame pandas et peuvent être exportés au format CSV. Ils sont composés des informations suivantes : “review_id”, “category”, “confidence_score”, “relevance_score”. La restitution visuelle est assurée par Matplotlib et Seaborn pour l’analyse exploratoire et les visualisations globales.

FLUX DE DONNÉES

- **Extraction :**

  Les données sont récupérées depuis la base PostgreSQL via une requête SQL optimisée pour extraire les champs pertinents (avis, note, texte, image, abonnement, etc).

- **Préparation :**

  Nettoyage et enrichissement des données, calcul de variables dérivées (“text_length”, “is_subscribed”, “has_image”).

- **Classification :**

  Passage du texte dans le modèle zero-shot pour obtenir la catégorie et le score de confiance.

- **Scoring :**

  Calcul du score de pertinence basé sur la pondération des critères suivants : texte (25%), image (20%), abonnement (20%), note extrême (15%), mots-clés (20%).

- **Restitution :**

  Génération de graphiques, rapports et statistiques agrégées. Les résultats peuvent être exportés pour une future intégration à un tableau de bord ou une API interne.

MATRICE RACI

FONCTIONNALITÉS / ACTIVITÉ	CHEF DE PROJET	DATA ENGINEER	DATA ANALYST	DATA SCIENTIST	RESPONSABLE QUALITE / SECURITE
Récupération des données	C	R / A	I	I	I
Analyse des données brutes	I	R	A	C	I
Traitement des données	I	R / A	C	I	I
Implémentation des algorithmes de classification	I	C	I	R / A	I
Implémentation des algorithmes de pondération	I	C	I	R / A	I
Création de la visualisation des résultats	I	C	R / A	C	I
Rédaction des tests (unitaires, validation des règles)	C	R / A	C	I	I
Rédaction de la documentation technique	A	R	C	C	I
Rédaction du rapport final	A	C	R	C	I
Mise en production	A	R	I	C	C
Définir les règles de qualité	C	R	A	I	C
Gérer la sécurité et les accès	I	C	I	I	R / A
Gérer la conformité et la sécurité réglementaire	I	I	I	I	R / A

Légende :

- R (Responsible) : réalise la tâche.
- A (Accountable) : porte la responsabilité finale de la tâche et valide le résultat.
- C (Consulted) : Consulté pour avis ou expertise.
- I (Informed) : Informé du déroulement ou du résultat de la tâche

SÉCURITÉ ET CONFORMITÉ

L'architecture du prototype respecte les principes de sécurité et de conformité liés au traitement de données client internes à Amazon.

Aucune donnée personnelle sensible n'est extraite ni stockée localement : seules les informations nécessaires à l'analyse (avis, texte, note, métadonnées produit) sont manipulées.

Le stockage et le traitement respectent les bonnes pratiques RGPD et CNIL suivantes :

- Minimisation des données : seules les colonnes utiles au use case sont extraites.
- Traçabilité : toutes les étapes du pipeline (extraction, transformation, scoring) peuvent être auditées.
- Stockage sécurisé : le traitement local est chiffré au niveau du système de fichiers, et la base PostgreSQL nécessite une connexion SSL (sslmode=require).
- Conformité cloud : l'architecture est compatible avec les exigences AWS (S3, IAM, VPC) pour un déploiement conforme à la politique de protection des données d'Amazon.

Ces mesures garantissent que la solution reste conforme au RGPD et intégrable sans risque dans un environnement de production sécurisé.

MÉTRIQUES DE VALIDATION

Avant de considérer le prototype comme déployable en production, il lui faudra atteindre les résultats suivants :

DOMAINE	INDICATEUR	PRÉREQUIS
Performance NLP	Score moyen de confiance du modèle	> 0.5
Pertinence des avis	Score moyen de pertinence	> 0.6
Efficacité du pipeline	Temps de traitement par review	< 3s (CPU) ou < 1s (GPU)
Stabilité des résultats	Écart-type du score de pertinence	< 10
Préparation production	Taux d'exécution sans erreur du pipeline	100%

TECHNOLOGIES ET ENVIRONNEMENTS

COMPOSANT	TECHNOLOGIE + VERSION	USAGE
Langage	Python 3.12	Pipeline et modélisation
Base de données	PostgreSQL (lecture seule)	Source de données transactionnelle
Connexion	psycopg2 2.9.9	Accès et requêtes SQL
Traitement	pandas 2.2.3 / numpy 1.26.4	Manipulation et traitement des données
Modèle NLP	transformers 4.57.1 / DistilBART MNLI	Classification zero-shot
Visualisation	matplotlib 3.9.2 / seaborn 0.13.2	Analyses et visualisations
Environnement	Local (CPU) + compatible GPU / Cloud (AWS)	Exécution et scalabilité

JUSTIFICATION TECHNOLOGIQUE

TECHNOLOGIE	JUSTIFICATION
Python	Langage standard de la data science, riche en librairies pour NLP et interopérable avec PostgreSQL
PostgreSQL	Fiable, open-source, conforme SQL, parfaite pour extraire et agréger des données textuelles à grande échelle.
psycopg2	Stable et sécurisé, gère les connexions SSL et les exceptions SQL efficacement.
pandas / numpy	Librairies standards de data science, très efficace et très bien documenté
transformers / DistilBART MNLI	Compromis idéal entre performance et vitesse. Le mode “MNLI” permet la classification sans jeu d’entraînement spécifique, adaptée à un prototype non supervisé.
matplotlib / seaborn	Libraries connexes à Python et Pandas, permettant des visualisations rapides, intégrées au code
Jupyter Notebook / AWS	Le notebook permet d'itérer et documenter facilement, parfait pour un prototype. Le déploiement sur AWS nous permet d’utiliser une autre force d’Amazon, pour industrialiser et permettre la scalabilité du projet

🏁 CONCLUSION

Le présent document a permis de définir le périmètre fonctionnel et technique du prototype de classification et de valorisation des avis clients d'Amazon.

Ce système valide la faisabilité d'une approche non supervisée basée sur un modèle zéro-shot de type DistilBART MNLI, capable de catégoriser automatiquement les avis selon leur contenu textuel, tout en intégrant des indicateurs complémentaires comme la présence d'image ou le statut d'abonnement.

Les premiers résultats obtenus démontrent une cohérence globale du pipeline et une capacité réelle de tri automatisé des retours clients. La confiance moyenne du modèle (48 %) et la distribution logique des scores de pertinence confirment la viabilité du processus, malgré une performance limitée sur les avis courts ou ambigus.

Des pistes d'amélioration ont été identifiées :

- affiner le scoring textuel pour mieux valoriser les avis synthétiques ;
- renforcer la pondération dynamique entre critères (texte, image, abonnement, rating) ;
- envisager un fine-tuning partiel du modèle sur un jeu d'avis annotés afin d'améliorer la précision des catégories.

À moyen terme, le système pourra être intégré à l'écosystème analytique Amazon au sein d'un pipeline ETL automatisé, alimentant les équipes produit et marketing via des tableaux de bord internes. Une orchestration cloud (AWS + S3) assurerait la scalabilité du traitement et la mise à jour continue des résultats.

Enfin, une supervision continue des performances permettra d'adapter le modèle aux évolutions du langage et des comportements clients. Cette évolution ouvrirait la voie à une approche supervisée globale combinant apprentissage automatique, détection de sentiment et analyse contextuelle, garantissant la pertinence et la fiabilité durable de la solution au sein de l'écosystème Amazon.