

Cahier des charges



Projet : Amazon Review Analysis

Auteur : Ismaël SYLLA

A propos de ce document

Ce document reprend les exigences, les normes et les conditions spécifiques auxquels le projet doit répondre pour être mené à bien. Le but de ce document est de comprendre les objectifs et les limites du projet ainsi que de s'assurer que tous les acteurs impliqués savent ce qu'on attend d'eux.

Les éléments repris dans ce document se base sur un prototype développé afin d'étudier la faisabilité et de donner un premier aperçu de la future solution qui sera développée par la suite dans le cadre de ce projet.

Table des matières

1.	Introduction	4
1.1.	Contexte du projet	4
1.2.	Problématiques.....	4
1.3.	L'objectif du projet	4
1.4.	L'attendu du projet.....	4
2.	Périmètre	5
2.1.	Inclusions	5
2.2.	Exclusions	5
2.3.	Besoins fonctionnels	6
2.4.	Besoins non fonctionnels.....	6
3.	Contraintes.....	6
4.	Source de données	7
5.	Risques et matrice de risques.....	8
6.	Enjeux réglementaires et éthiques	9
7.	Critères de succès :	9

1. Introduction

1.1. Contexte du projet

Amazon, un des grands leaders du e-commerce, reçoit des milliers d'avis d'utilisateurs sur les produits vendus sur sa plateforme. Ces avis constituent une source précieuse pour les consommateurs dans leur processus de décision d'achat.

Toutefois, les produits peuvent avoir énormément d'avis et tous les avis n'ont pas la même valeur. Certains sont courts, certains sont biaisés et d'autres non pertinents. L'objectif de ce projet est de développer un système de scoring des commentaires pour tous les produits afin d'évaluer leurs pertinences et les mettre en avant.

1.2. Problématiques

- Tous les avis ne sont pas toujours pertinents.
- Les notes laissées par les consommateurs ne reflètent pas toujours la qualité réelle du produit.
- Amazon et les vendeurs ont besoin d'indicateur plus fiable basé sur la qualité des avis.
- Les commentaires les plus pertinents ne remontent pas forcément en priorité pour les consommateurs.

1.3. L'objectif du projet

Le projet vise à développer une solution intelligente capable d'analyser automatiquement les avis clients Amazon afin de déterminer les produits les plus pertinents selon un système de scoring.

La solution comprendra deux étapes principales : la catégorisation et la notation des avis selon un score de confiance, puis le calcul d'un score de pertinence global destiné à identifier et promouvoir les avis les plus pertinents pour chaque produit.

1.4. L'attendu du projet

- Classifier chaque avis des produits, en combinant plusieurs critères comme la pertinence du texte, sa longueur, la présence d'images, la confiance du modèle et la richesse en mots-clés.
- Mettre en avant les avis les plus utiles et fiables, afin que les utilisateurs puissent rapidement identifier les commentaires de qualité.
- Construire un tableau de bord pour visualiser les scores des avis, comparer leur pertinence et explorer les informations importantes.

2. Périmètre

2.1. Inclusions

Le projet couvre les activités suivantes :

- Collecte des données Amazon : utilisation d'un jeu de données d'avis produits Amazon stocké dans une base de données relationnelle.
- Nettoyage et préparation des données : traitement et harmonisation des données, notamment la suppression des doublons et la correction des incohérences.
- Catégorisation des avis : application d'un modèle d'intelligence artificielle pour classifier les avis avec un score de confiance associé. Les catégories identifiées :
 - "product quality or satisfaction" pour les commentaires qui parlent de la qualité des produits.
 - "product defect or damaged item" pour les avis qui parlent des produits défectueux ou endommagés.
 - "delivery issue or shipping delay" pour les avis qui dénoncent des problèmes de livraisons.
 - "customer service or support" pour les avis qui dénoncent des problèmes avec le service client.
- Calcul du poids des avis : définition et mise en œuvre d'une formule de pondération intégrant les critères suivants :
 - 30% taille de la description du produit (utilisation de la formule Gaussienne pour déterminer la taille).
 - 20% si la description contient une image.
 - 10% si l'utilisateur a passé une commande.
 - 15% si l'utilisateur a laissé un score de 1 ou 5 (extrémité).
 - 25% le score est calculé à l'aide d'un algorithme, qui évalue le sentiment du texte global. Plus la description exprime un sentiment positif ou négatif marqué, plus elle est considérée comme pertinente.
- Analyse et visualisation : création de graphiques représentant la distribution des notes et la classification des avis, ainsi que la comparaison des scores par catégorie ou type de produit.
- Adaptation de l'application Streamlit existante pour remonter les avis pertinents.

2.2. Exclusions

Le projet ne comprend pas les éléments suivants :

- Collecte automatique via API : seules des données statiques, récupérées une fois, seront utilisées, en raison de contraintes légales et de limites d'accès.
- Intégration avec les systèmes réels Amazon.
- Actions automatiques sur la plateforme réel Amazon

2.3. Besoins fonctionnels

En se basant sur le prototype, le projet doit permettre de :

- Charger les données Amazon : charger le jeu de données contenant les avis produits dans un data lake depuis une base de données relationnelle.
- Traitement des données (ETL) : récupération à partir du data lake, traitement des doublons, de texte et jointures et anonymisation des données personnelles.
- Stockage des données : les données sont ensuite stockées dans un datawarehouse et une base de données NoSQL.
- Classification et calcul des poids pour les avis : appliquer un algorithme de classification puis une formule de pondération basée sur plusieurs critères. Ces deux éléments vont permettre d'évaluer la pertinence des avis. La pondération des variables pourrait être amenée à changer.
- Stocker les résultats : les résultats sont ensuite enregistrés dans le datawarehouse.
- Visualiser les résultats : des visualisations permettant d'analyser les résultats.

2.4. Besoins non fonctionnels

Le projet doit également respecter les exigences suivantes :

- Sécurité : aucune donnée personnelle n'est affichée ou exposée dans les jeux de données.
- Accessibilité : La solution doit pouvoir être exécutée sur n'importe quel environnement de travail.
- Maintenabilité : le code doit être modulable, lisible et correctement commenté.
- Scalabilité : la solution doit pouvoir évoluer pour intégrer, à l'avenir, des flux de données via API ou d'autres sources externes.

3. Contraintes

Le projet doit tenir en compte les contraintes suivantes :

- Temps limité : le projet est réalisé dans le cadre d'un projet académique.
- Multilinguisme des avis : les avis étant rédigés en plusieurs langues, il est important de choisir un algorithme capable de prédire correctement les catégories.
- Données : accès en lecture seule sur la base de données relationnelle.
- Limite des données : seules des données simulées ou publiques sont utilisées, sans accès à l'API Amazon.
- Ressource humaine : sur certains points l'équipe est encore junior sur les sujets (ML).
- Infrastructure : le traitement se fait sur une infrastructure locale, en privilégiant des solutions gratuites et open source.
- Données personnelles : dans le projet les données à caractères personnelles identifiées doivent être anonymisées.

4. Source de données

Les données utilisées dans le projet proviennent exclusivement d'une base de données relationnelle PostgreSQL mise à disposition dans le cadre du projet. Elles contiennent des informations relatives aux avis produits, aux produits eux-mêmes, aux images associées et aux commandes des utilisateurs.

L'extraction des données nous permet de récupérer les champs pertinents pour l'évaluation et le scoring des avis. Cette extraction applique plusieurs opérations incluant jointures, filtrage et enrichissement (présence d'image, longueur du texte, existence d'une commande).

Extrait des données utilisées :

- review_id, buyer_id (qui sera anonymisé)
- title, description (texte de l'avis), rating
- text_length (calculé via LENGTH)
- has_image (déduit de la table review_images)
- has_orders (présence d'au moins une commande passée par l'utilisateur)
- p_id, product_name, category

Enfin, cette extraction nous permet d'avoir un dataset structuré et riche, facilitant la préparation du pipeline ETL et l'analyse ML.

5. Risques et matrice de risques

Risque	Probabilité	Gravité	Mesures de mitigation
Classification incorrecte due au zero-shot ML (avis multilingues + contexte varié)	Moyenne	Élevée	Tests sur corpus multilingue, vérification manuelle d'un échantillon, ajustement des seuils de confiance
Pondération non représentative (formule héritée du prototype)	Faible	Moyenne	Documenter chaque variable, réaliser des tests exploratoires, permettre l'ajustement des poids
Dépendance à la structure de la BD PostgreSQL	Moyenne	Faible	Documentation de la requête, vérification régulière du schéma, gestion d'erreurs dans l'ETL
Données incorrectement anonymisées (notamment buyer_id)	Très faible	Élevée	Hashage systématique du buyer_id, double validation conformité RGPD
Manque de données ou valeurs manquantes dans certaines colonnes	Moyenne	Faible	Contrôles qualité ETL, remplacement ou exclusion maîtrisée des valeurs manquantes
Performance limitée de l'infrastructure locale	Moyenne	Moyenne	Optimisation des requêtes SQL, traitement par lots, usage d'index PostgreSQL

6. Enjeux réglementaires et éthiques

Le projet doit se conformer aux règles en vigueur concernant le traitement des données, notamment la protection des données personnelles et la transparence des algorithmes utilisés.

Respect du RGPD/CCPA

- Les données personnelles provenant de la base PostgreSQL, en particulier buyer_id, font l'objet d'une anonymisation obligatoire avant tout traitement ou analyse.
- Aucun élément permettant d'identifier directement ou indirectement un utilisateur ne sera conservé dans les systèmes analytiques ou dans les visualisations.
- Le pipeline ETL inclut une étape spécifique dédiée à cette anonymisation.

Éthique et transparence de l'IA

- L'utilisation du modèle zero-shot classification implique de veiller à la réduction des biais linguistiques et culturels. Une vérification humaine sur un échantillon sera réalisée pour garantir la pertinence des classifications.
- Les règles de calcul du score pondéré sont documentées, transparentes et ajustables afin d'éviter toute discrimination involontaire entre catégories d'avis.
- La solution ne doit pas manipuler les avis mais simplement fournir une lecture plus structurée et objective de leur pertinence.

7. Critères de succès :

Le projet sera considéré comme réussi si :

- Exécution sans erreur : la solution peut être exécutée correctement dans son environnement prévu.
- Pondération justifiée et paramétrable : le calcul du poids des avis est documenté et peut être reproduit.
- Résultats cohérents : les scores pondérés sont en corrélation avec les notes globales des avis, permettant une analyse fiable.
- Documentation : la solution complète est documentée.
- Conformité réglementaire.