Université Claude Bernard Lyon 1

UE Data Mining

Rapport Projet Data Mining

Étudiants :
Axel COLMANT
Matéo MUNOZ
Mohamed Massamba SENE

Enseignant :
Rémy Cabazet

Table des matières

1 Introduction

Le commerce électronique a transformé la manière dont les entreprises et les particuliers achètent et vendent des biens, en exploitant la puissance d'Internet pour atteindre une vaste audience. Toutefois, la diversité des profils de clients dans l'e-commerce rend complexe la caractérisation du marché potentiel. Cette hétérogénéité souligne l'importance d'une connaissance approfondie de la clientèle cible, essentielle pour des communications efficaces et personnalisées. Une compréhension fine des comportements des consommateurs permet aux entreprises de segmenter leur clientèle en groupes distincts, afin de mieux répondre à leurs besoins spécifiques.

Dans cette étude, nous utilisons les données d'Olist Store, une entreprise brésilienne spécialisée dans la vente en ligne. Ces données, disponibles sur la plateforme Kaggle, regroupent près de 100 000 commandes passées entre 2016 et 2018 sur différents marchés au Brésil. Les données fournies sont anonymisées, et les références à des entreprises ou partenaires ont été remplacées par des termes issus de l'univers de Game of Thrones. Ces informations comprennent les détails des clients, leur localisation géographique, et les modalités de livraison, permettant ainsi une analyse complète des transactions effectuées.

La figure 1 illustre la structure du jeu de données, mettant en évidence les différentes tables et la manière dont elles sont reliées. Ces données réelles constituent une base solide pour notre étude exploratoire.

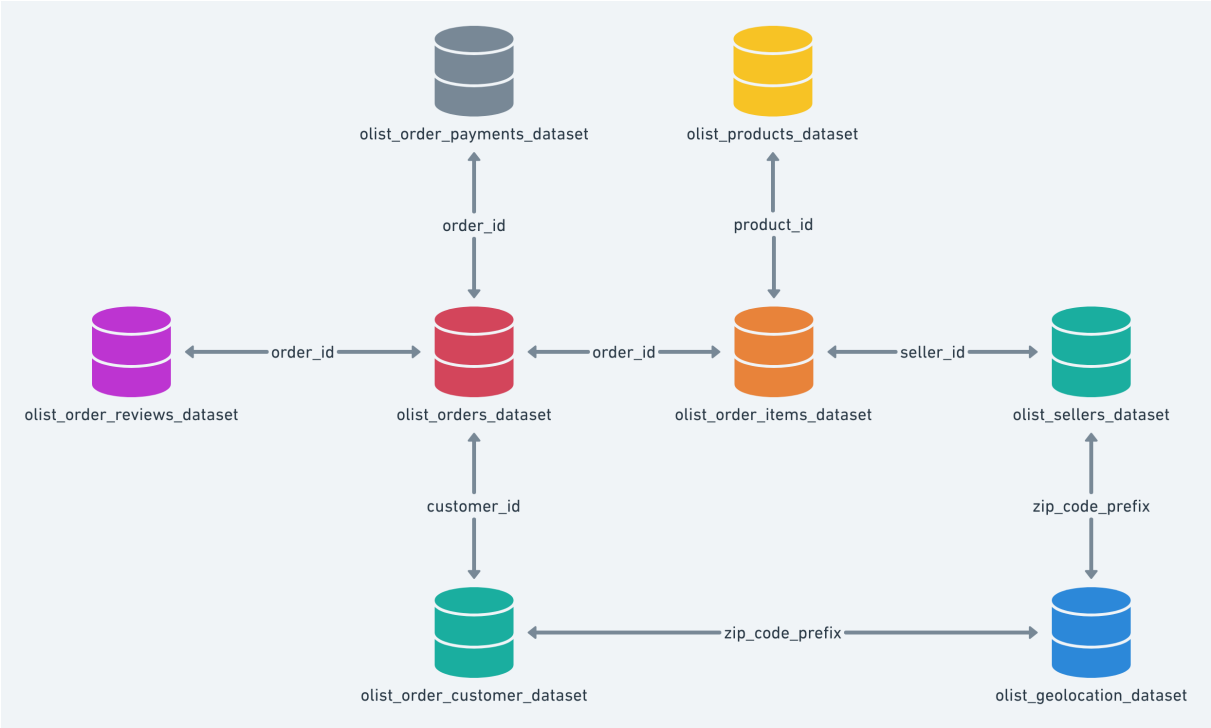

FIGURE 1 – Structure du jeu de données Olist

Dans le cadre de notre projet, nous adopterons trois approches principales pour exploiter

ce jeu de données :

- **Segmentation des clients par clustering** : En utilisant les données des commandes, nous appliquerons des techniques de clustering afin de regrouper les clients en segments homogènes, facilitant une personnalisation des offres.
- **Recommandation de produits** : Nous analyserons les avis clients pour construire un système de recommandation de produits, contribuant à améliorer l'expérience utilisateur et augmenter les ventes.
- **Analyse par graphes** : Nous appliquerons des méthodes d'analyse basées sur les graphes pour extraire des informations supplémentaires à partir des relations entre les différentes entités du jeu de données, telles que les clients, les produits et les marchés.

Le code source et les scripts utilisés pour ce projet sont accessibles sur notre dépôt Github, disponible via ce lien.

2 Clustering

Dans cette section, l'objectif est de réaliser une segmentation des clients par le biais de techniques de clustering. Cette approche repose sur une analyse approfondie des comportements des utilisateurs et de leurs données personnelles. Le processus de segmentation s'articule autour de deux grandes étapes, qui seront décrites dans la suite de cette section.

2.1 Préparation et exploration des données

Comme indiqué dans l'introduction, le jeu de données contient des informations relatives aux commandes des clients ainsi qu'à leur localisation géographique. Dans le cadre de cette étude, nous avons choisi de charger l'ensemble des jeux de données, à l'exception des tables concernant la géolocalisation et les avis clients. En effet, notre objectif est de nous focaliser sur l'analyse des comportements d'achat des utilisateurs en nous basant exclusivement sur leur historique d'achats. Les deux jeux de données omis seront exploités dans d'autres sections de notre travail.
Pour chaque dataframe obtenu, les étapes suivantes ont été appliquées :

- Suppression des duplicatas et gestion des valeurs manquantes ;
- Élimination des colonnes jugées non pertinentes pour notre analyse (par exemple, les propriétés des produits) ;
- Conversion des colonnes contenant des dates, initialement au format chaîne de caractères, en format datetime.

Après avoir nettoyé individuellement chaque jeu de données, nous les avons fusionnés pour constituer un jeu de données intermédiaire. Ce dernier a ensuite fait l'objet d'une suppression des valeurs manquantes, leur proportion étant négligeable par rapport à la taille du jeu de données. Par conséquent, l'impact de cette suppression sur l'apprentissage devrait être minimal. Le dataframe final obtenu comporte 115 903 lignes et 21 colonnes. Les caractéristiques présentes dans ce dataframe incluent les éléments suivants.

- Des identifiants uniques ;
- Des informations relatives au paiement ;
- Les détails de la localisation des clients ;
- Les informations de localisation des vendeurs ;
- Le nom de la catégorie du produit ;
- Le prix des articles et les frais de transport ;
- Le statut des commandes ;
- Les dates associées aux commandes.

Nous avons ensuite procédé à la de visualisation des données afin d'en apprendre plus sur le jeu de données. Nous avons commencé par séparer les variables qualitatives et quantitatives avant de créer nos graphiques.
La Figure 2 présente les différentes distributions de nos variables quantitatives.

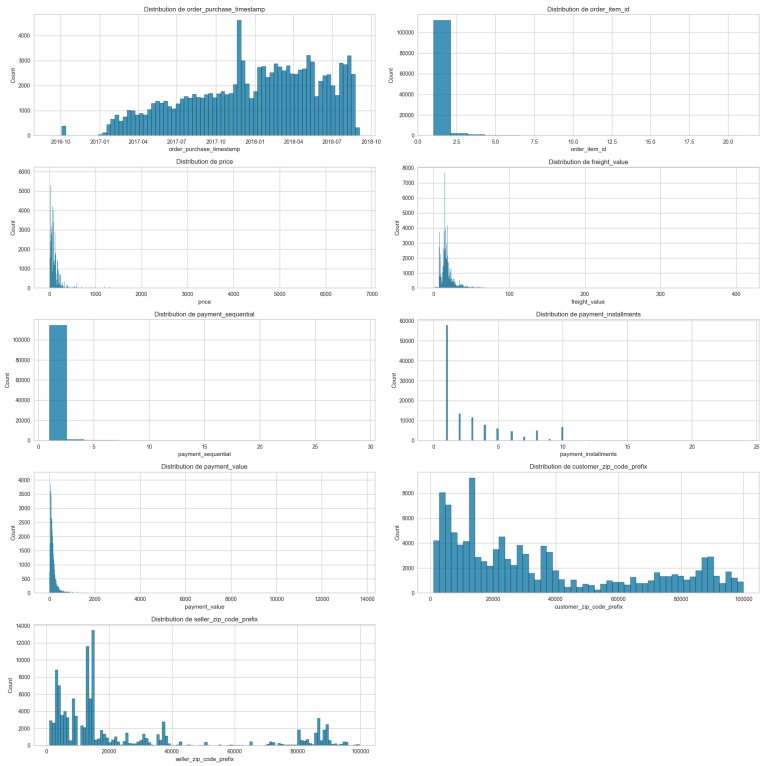

FIGURE 2 – Distributions des variables quantitatives

La Figure 3 présente quelques visualisations effectuées sur les variables qualitatives.

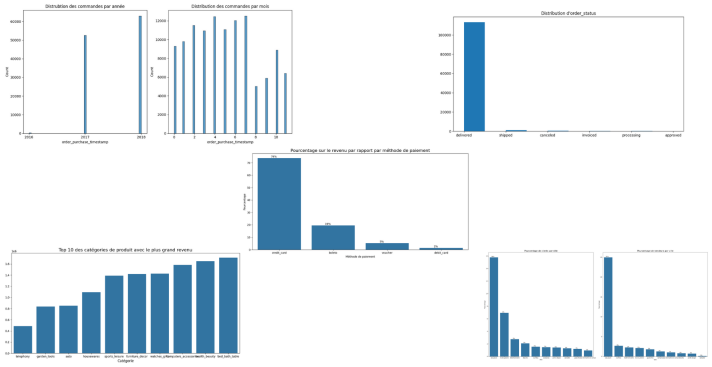

FIGURE 3 – Distributions des variables qualitatives

Sur la base des visualisations effectuées, plusieurs actions ont été entreprises, telles que la suppression des données de l'année 2016, celles-ci étant en quantité négligeable par rapport aux données de 2017 et 2018. De plus, nous avons appliqué une transformation logarithmique aux variables price, freight_value et payment_value afin de les normaliser. Cependant, dans son état actuel, le jeu de données est centré autour des commandes. Nous avons donc jugé nécessaire de créer de nouvelles variables, agrégées autour des clients, afin de mieux comprendre leurs comportements. Nous avons donc procéder à quelques visualisations afin de nous assurer de la pertinence de voir les distributions des données ainsi générées. Les résultats obtenus sont visibles dans la Figure 4.

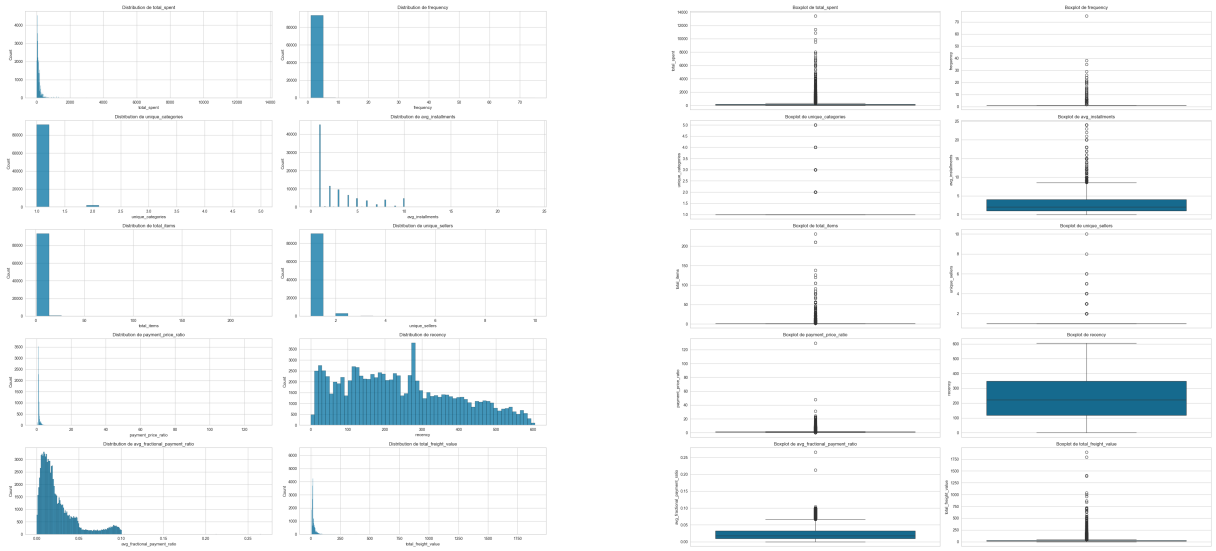

FIGURE 4 – Distributions (gauche) et Box-plots (droite) des variables créées

Nous avons ainsi eu à supprimer les variables unique_categories et unique_sellers car elle ne prenait majoritairement qu'une seule valeur et donc cela ne comportait pas une information pertinente pour la construction de notre segmentation. Nous avons également visualisé la corrélation entre nos variables ce qui nous a permis d'identifier les facteurs qui influencent les dépenses totales et la fréquence d'achat. Cette matrice est présente dans la Figure 5.

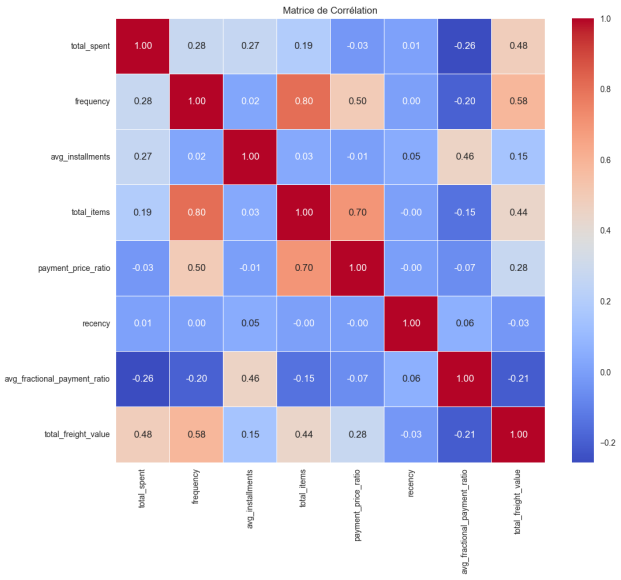

FIGURE 5 – Matrice de corrélation des variables créées

2.2 Choix des modèles et évaluation

Après la première étape de préparation et d'exploration des données ainsi que le feature engineering réalisé, nous allons donc procéder au clustering des clients.

2.2.1 Modèles testés

Nous avons décider de tester trois approches de clustering :

- **KMeans** [1] qui minimise la variance intra-cluster en optimisant la distance euclidienne entre les points et leur centre de cluster ;
- **GaussianMixture** [2] permet de modéliser chaque cluster comme une distribution gaussienne en autorisant des clusters de formes ellipsoïdales ou de différentes tailles ;
- **DBSCAN** [3] est choisi pour sa capacité à détecter des clusters de forme arbitraire et à identifier les outliers, sans nécessiter de définir à priori le nombre de clusters.

2.2.2 Métriques d'évaluation

Notre but est de comparer ces trois approches sur nos données afin de voir laquelle donne le meilleur clustering, pour cela nous utilisons donc comme métriques :

- **Silhouette Score** évalue la compacité et la séparation des clusters. Il mesure à quel point les points sont proches des autres points dans le même cluster par rapport aux points des clusters voisins ;
- **Calinski-Harabasz Score** évalue la séparation inter-clusters par rapport à la compacité intra-cluster. Un score plus élevé indique que les clusters sont denses et bien séparés ;
- **Davies-Bouldin Score** mesure le rapport entre la distance intra-cluster et la distance inter-cluster. Un score plus bas indique une meilleure séparation et des clusters plus compacts.

2.2.3 Optimisation des hyperparamètres

Nous déterminons les hyperparamètres à utiliser pour les modèle au travers de la méthode du coude pour KMeans, le Knee point pour DBSCAN et BIC score pour Gaussian Mixture. Nous utilisons également une pipeline qui effectue d'abord la normalisation et l'ACP avant de faire le clustering. En effet la normalisation est essentielle pour uniformiser l'échelle des variables et éviter que celles à grande amplitude ne dominent le calcul des distances dans les algorithmes de clustering. L'ACP permet de réduire la dimensionnalité tout en conservant les informations clés, ce qui rend les données plus faciles à traiter. Nous effectuons également une réduction de dimensionnalité avec TSNE.

2.3 Résultats

Après avoir entrainé les différents modèles, nous avons obtenus les résultats consignés dans le Tableau 1.

1. https ://en.wikipedia.org/wiki/K-means_clustering
2. https ://fr.wikipedia.org/wiki/Modèle_de_mélange_gaussien
3. https ://en.wikipedia.org/wiki/DBSCAN

Modèle	Silhouette Score	Calinksi-Harabasz Score	Davies-Bouldin Score
KMeans	0.15	11699.36	3.64
GaussianMixture	-0.03	6330.48	10.44
DBSCAN	-0.84	41.28	3.18

TABLE 1 – Comparaison des modèles

Sur la base des résultats, nous pouvons voir que le modèle KMeans présente les meilleurs performances avec le plus haut Silhouette Score et Davies-Bouldin Score bien que GuassianMixture ait un Calinski-Harabasz Score plus élevé. Nous avons donc poursuivi avec ce modèle en afin de construire nos clusters que nous pouvons voir dans la Figure 6.

segments	total_spent	frequency	avg_installments	total_items	payment_price_ratio	recency	avg_fractional_payment_ratio	total_freight_value	#observations	Percentage
0	110.634750	1.120612	1.821361	1.186958	1.374357	391.859870	0.018300	19.877196	28231	0.300935
1	114.649922	1.043548	6.545977	1.073587	1.356946	257.686693	0.062378	19.794447	16878	0.179915
2	697.614984	7.133441	3.541767	21.368167	7.418621	230.554662	0.003029	152.128424	622	0.006630
3	540.072907	2.225655	5.081877	3.392778	2.177333	229.376483	0.008686	66.466257	7671	0.081771
4	107.202166	1.107699	1.721428	1.168824	1.376395	129.715212	0.017734	20.278010	40409	0.430749

FIGURE 6 – Clusters de clients obtenues

Notre interprétation des segments ainsi obtenus est la suivante :

- Segment 0 : Clients occasionnels avec faible dépense moyenne
- Segment 1 : Clients achetant principalement en plusieurs fois avec faible fréquence
- Segment 2 : Clients à haute valeur avec achats fréquents et importants
- Segment 3 : Clients réguliers avec paniers d'achat moyens
- Segment 4 : Clients récents et peu engagés

Cependant il faut noter que les clusters ne sont pas bien séparés et que de nombreux points sont proches des frontières entre clusters marqué par le faible silhouette_score (0.15). La Figure 7 montre que les clusters clients sont difficilement discernables avec une faible séparation des clusters.

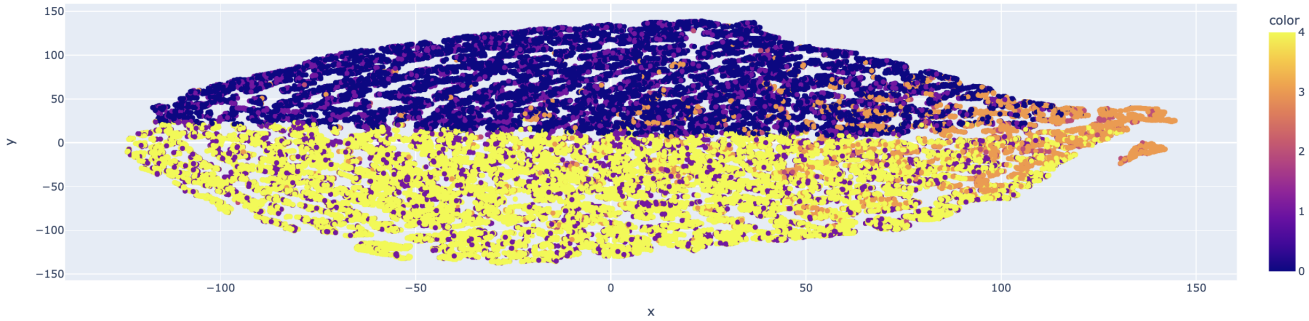

FIGURE 7 – Visualisation TSNE des clusters

3 Recommendation

L'objectif principal de cette section est de développer un système de recommandation capable de proposer des produits pertinents aux utilisateurs en se basant sur leur historique d'achat et les évaluations disponibles. Cette démarche s'appuie sur une analyse approfondie des données et l'application de modèles avancés d'apprentissage automatique.

3.1 Préparation des données

Pour construire le système de recommandation, les étapes suivantes ont été réalisées :

- **Nettoyage des données :** Suppression des doublons et des valeurs manquantes dans les jeux de données liés aux produits, commandes, clients, paiements, et évaluations.
- **Fusion des données :** Combinaison des jeux de données via des jointures successives pour constituer un dataframe final contenant :
  - Identifiants des clients et des produits ;
  - Scores d'évaluation des produits par les clients ;
  - Informations sur les catégories de produits ;
  - Détails des paiements effectués.
- **Filtrage des utilisateurs et des produits :**
  - Exclusion des clients ayant effectué une seule commande pour améliorer la personnalisation.
  - Suppression des produits avec moins de deux évaluations pour garantir une robustesse dans l'apprentissage.
- **Encodage des identifiants :** Conversion des identifiants des clients et des produits en indices numériques à l'aide de la classe `LabelEncoder`, afin de les utiliser dans les algorithmes de recommandation.

Le tableau final utilisé pour le système de recommandation comporte 31 958 lignes et trois colonnes principales : `customer_unique_id`, `product_id`, et `review_score`. Les données ont été normalisées à l'aide de `MinMaxScaler`, garantissant que les scores soient dans l'intervalle $[0, 1]$. Une colonne supplémentaire, `product_category_name`, a été ajoutée pour visualiser les catégories de produits dans l'espace latent. Les Figures 8, 9 et 10 présentent les visualisations effectuées.

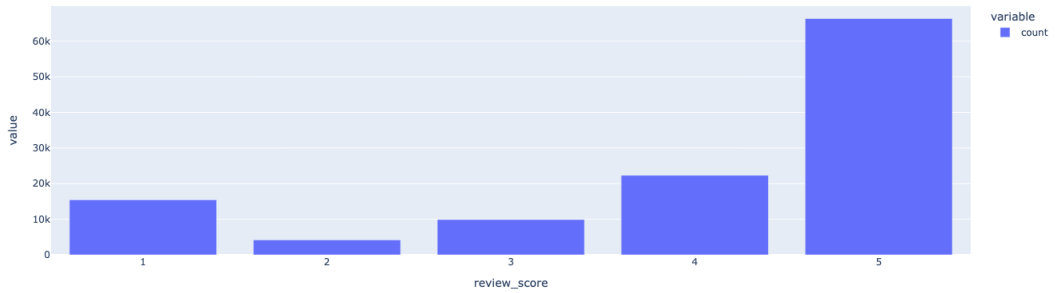

FIGURE 8 – Répartition des notes

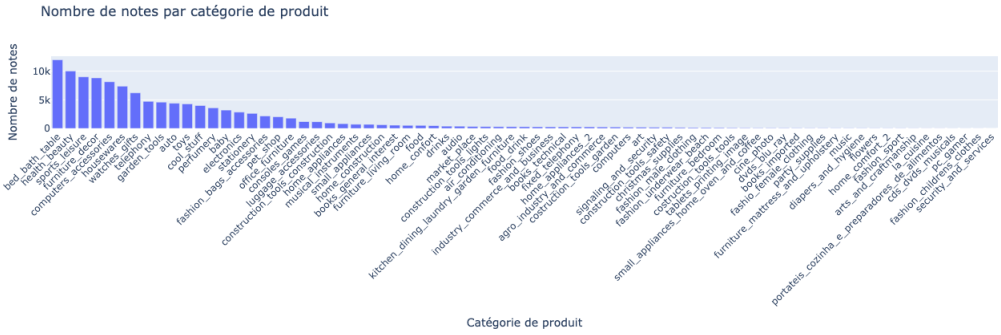

FIGURE 9 – Répartition du nombre de notes par catégorie

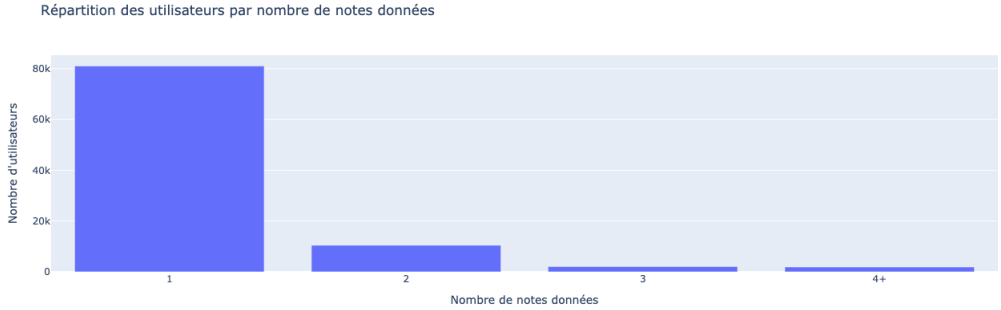

FIGURE 10 – Répartition du nombre de votes par note

Pour le score, une centralisation a été réalisée en standardisant les données : les scores ont été recentrés autour de 0 et leur écart-type uniformisé à 1, produisant une colonne `standardized_score`. Cette étape a permis de corriger un biais initial important, où les scores étaient majoritairement concentrés autour de la valeur 1.0 (5 étoiles). Cette normalisation améliore la capacité du modèle à capturer les relations latentes entre utilisateurs et produits, tout en facilitant la convergence et la stabilité de l'algorithme de factorisation matricielle.

3.2 Choix des modèles et évaluation

Plusieurs algorithmes de recommandation ont été testés afin de sélectionner celui offrant les meilleures performances.

3.2.1 Méthodes de recommandation testées

— **SVD (Singular Value Decomposition) :** Factorisation de la matrice utilisateur-produit en composantes latentes pour prédire les scores des produits non notés.
— **NMF (Non-negative Matrix Factorization) :** Méthode similaire, avec des composantes latentes contraintes à des valeurs non négatives pour une meilleure interprétation.

SVD++ n'a pas été utilisé, car l'absence d'interactions implicites dans les données disponibles le rend peu utile dans ce contexte.

3.2.2 Optimisation des hyperparamètres

— **SVD :** Les valeurs pour le nombre de facteurs latents (`n_factors`) ont été testées dans l'ensemble {2, 8, 16, 32, 64, 128, 512}.
— **NMF :** Les dimensions testées sont {5, 10, 15, 20, 30, 40, 50, 100}, avec une limite pour préserver les ressources mémoire.

Le meilleur modèle a été choisi en fonction de la **Root Mean Square Error (RMSE)** moyenne calculée par validation croisée (5 plis). Une optimisation supplémentaire a été réalisée via `Grid Search` avec la bibliothèque `Surprise`, améliorant les résultats.

3.2.3 Métriques d'évaluation

Les performances des modèles ont été évaluées à l'aide de la RMSE, mesurant l'écart entre les scores réels et prédits. Le modèle SVD a obtenu la meilleure RMSE avec une valeur minimale de 0.585. Les résultats complets sont présentés dans le tableau 2.

Modèle	Dimension	RMSE (meilleur)
SVD	512	**0.585**
NMF	30	1.159

TABLE 2 – Résultats des tests pour obtenir le meilleur algorithme

Les meilleurs hyperparamètres pour SVD, selon l'optimisation par Grid Search, sont :
— Nombre de dimensions : 10 ;
— Nombre d'epochs : 500 ;
— Taux d'apprentissage global : 0.005.

3.3 Implémentation du système de recommandation

3.3.1 Formation et extraction des caractéristiques

Le meilleur modèle a été formé sur l'ensemble des données. Les vecteurs latents des produits et des utilisateurs ont ensuite été extraits pour analyser leurs relations.

3.3.2 Visualisation des similarités

Les similarités entre produits ont été visualisées en réduisant la dimensionnalité des vecteurs latents :
— Une analyse en composantes principales (PCA) a réduit les vecteurs à 5 dimensions.
— Une réduction supplémentaire via t-SNE a permis de visualiser les produits dans un espace bidimensionnel.

La Figure 11 montre les clusters formés, révélant des regroupements naturels de produits similaires.

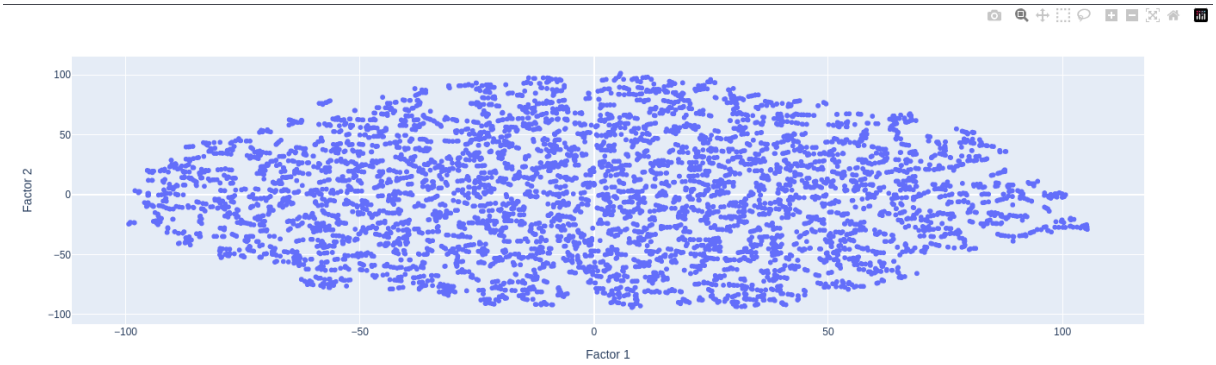

FIGURE 11 – Visualisation t-SNE des similarités entre produits.

3.4 Résultats

Le système génère des recommandations spécifiques pour chaque utilisateur :

1. Les produits déjà achetés sont exclus.
2. Les scores prédits sont triés par ordre décroissant.
3. Les n produits ayant les scores les plus élevés sont proposés.

Cependant, les résultats ne sont pas entièrement satisfaisants (voir Figure 12).

```
Client ID aléatoire : 6062
Rated Product:
5281 - health_beauty: 1.0
5339 - health_beauty: 1.0

Recommended Product:
5767 - housewares: Estimated Rating = 1.50
3344 - furniture_decor: Estimated Rating = 1.47
810 - home_construction: Estimated Rating = 1.39
1993 - pet_shop: Estimated Rating = 1.37
5117 - agro_industry_and_commerce: Estimated Rating = 1.36
499 - bed_bath_table: Estimated Rating = 1.35
3307 - furniture_decor: Estimated Rating = 1.35
2689 - sports_leisure: Estimated Rating = 1.34
809 - housewares: Estimated Rating = 1.30
1638 - sports_leisure: Estimated Rating = 1.29
```

FIGURE 12 – Exemple de résultats pour le système de recommandation

Les recommandations, bien que basées sur des scores élevés, manquent de pertinence vis-à-vis des préférences explicites des clients. Par exemple, un client ayant attribué une note maximale à des produits de la catégorie `health_beauty` reçoit des recommandations issues de catégories différentes (`housewares`, `furniture_decor`, `sports_leisure`). Cela met en évidence une limitation dans la personnalisation fine, le système ne tenant pas pleinement compte des préférences catégorielles. En conséquence, malgré des recommandations statistiquement pertinentes, le système peut ne pas répondre aux attentes des utilisateurs, limitant ainsi son efficacité globale.

4 Network Data Mining

Les graphes sont des outils pour modéliser et analyser les relations complexes au sein d'un résea.

Dans le cadre de cette étude, nous nous concentrons sur les interactions entre les clients et les produits, en représentant ces relations sous la forme d'un graphe biparti. Ce dernier permet de capturer les comportements d'achat en connectant les clients aux produits qu'ils ont achetés, révélant ainsi les structures sous-jacentes et les tendances d'achat.

L'objectif de cette analyse est double : identifier les produits stratégiques et les clients influents, et découvrir des regroupements naturels au sein du réseau, tels que des communautés de clients partageant des préférences similaires ou des groupes de produits souvent achetés ensemble. En combinant la détection de communautés et les mesures de centralité, cette analyse vise à fournir des insights exploitables pour optimiser les stratégies de recommandation et de segmentation.

4.1 Préparation et Exploration des Données

Pour cette analyse, nous avons construit un graphe biparti afin de modéliser les interactions entre les clients et les produits. Les données ont été extraites des tables olist_customers_dataset, olist_orders_dataset, et olist_order_items_dataset. Ces tables ont été jointes pour établir les relations entre les clients (identifiés par customer_id) et les produits (identifiés par product_id) via les commandes effectuées.

Le graphe est constitué de deux types de nœuds : les clients et les produits, reliés par des arêtes représentant un achat. Chaque arête peut être pondérée pour refléter soit la fréquence d'achat, soit le montant total dépensé pour un produit donné. Le graphe final contient environ 132 000 nœuds et 102 425 arêtes, avec une structure hétérogène. Une exploration préliminaire a révélé des produits très populaires ayant de nombreuses connexions, ainsi que des clients avec des comportements d'achat variés.

Cette étape a permis de structurer les données de manière à capturer les relations complexes entre les clients et les produits, préparant le terrain pour des analyses approfondies.

4.2 Méthodes d'Analyse et Évaluation

L'analyse des graphes repose sur une variété de méthodes permettant de comprendre les structures et les relations au sein du réseau. Ces méthodes peuvent être regroupées en deux grandes catégories : **l'analyse structurelle**, qui examine les propriétés globales et locales du graphe, et **l'analyse communautaire**, qui se concentre sur les regroupements naturels au sein du réseau.

4.2.1 Analyse de Centralité

Les mesures de centralité fournissent des indicateurs sur l'importance des nœuds dans le réseau.

La **Centralité de degré**, identifie les noeuds les plus connectés, révélant les produits les plus populaires ou les clients les plus actifs.

La **Centralité d'autovecteur** met en lumière les noeuds qui influencent le plus le réseau globalement, souvent liés à d'autres noeuds influents.

La **Centralité de proximité** mesure la distance moyenne d'un nœud aux autres, indiquant les nœuds stratégiquement proches et accessibles.

4.2.2 Détection de Communautés

La détection de communautés vise à regrouper les nœuds en groupes fortement connectés. Ces communautés représentent des segments naturels dans le réseau, comme des groupes de clients partageant des préférences similaires ou des ensembles de produits fréquemment achetés ensemble. L'algorithme de modularité gourmande a été utilisé pour identifier ces regroupements, optimisant la densité des connexions au sein des communautés.

4.3