Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.Les bibliothéques utilisés :

**mlxtend :** Fournit des outils pour l'extraction de règles d'association, comme les algorithmes Apriori et FPGrowth, ainsi que pour générer des règles d'association.

**pandas** : Permet de manipuler et analyser des données sous forme de tableaux (DataFrames).

**scikit-learn :**

**StandardScaler :** Normalise les données pour qu'elles aient une moyenne de 0 et un écart-type de 1.

**KMeans :** Algorithme de clustering pour regrouper les données en k clusters.

DBSCAN : Algorithme de clustering basé sur la densité des points.

**NearestNeighbors :** Trouve les voisins les plus proches pour chaque point de données.

**matplotlib.pyplot :** Permet de créer des visualisations graphiques des données.

**numpy** : Bibliothèque pour les calculs numériques et la manipulation de tableaux multidimensionnels.

**Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect. 1 : Chargement et Nettoyage**

Ce codes permettent de charger et de nettoyer les données du fichier Online Retail.xlsx. Le fichier est d'abord chargé dans un DataFrame df pour être manipulé. Ensuite, les lignes contenant des valeurs manquantes dans les colonnes essentielles (InvoiceNo, Description et CustomerID) sont supprimées, garantissant que seules les transactions complètes et valides sont conservées.  
  
2 : RÈGLES D’ASSOCIATION  
Ces deux lignes de code ensemble filtrent d'abord les données pour ne conserver que les transactions effectuées en France **(df\_france = df[df['Country'] == 'France']).** Ensuite, elles regroupent ces données par numéro de facture **(InvoiceNo)** et par description de produit (Description), puis calculent la somme des quantités vendues pour chaque produit dans chaque factureUne image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect..  
  


Cette ligne transforme les quantités en un format binaire : si un produit a été acheté (quantité > 0), il est marqué par un 1, sinon par un 0. Cela transforme le tableau en une matrice binaire représentant la présence ou l'absence de chaque produit dans chaque transaction.

**Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.Apriori :**

Le code utilise l'algorithme Apriori pour extraire les ensembles d'items fréquents du panier d'achats basket\_fr avec un support minimum de 0,01. Ensuite, il génère des règles d'association à partir de ces ensembles fréquents en utilisant la métrique "lift" avec un seuil minimum de 1.

**Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.Output : (affichage de 5 premiers lignes)**

Interpretation :

**Règle 1 :**

Si un client achète : DOLLY GIRL BEAKER

Alors il achètera probablement aussi : CHARLOTTE BAG DOLLY GIRL DESIGN

Support : 1.29 % des transactions contiennent les deux articles.

Confidence : Dans 55.6 % des cas où quelqu’un achète le beaker, il achète aussi le sac.

Lift : 8.31 ➜ Forte corrélation, cette association est 8 fois plus probable qu’un achat aléatoire.

**Règle 2 :**

Inverse de la règle 1 :

Si on achète le sac, dans 19.2 % des cas, on prend aussi le beaker.

Même lift (8.31) car la force du lien est la même.

**Règle 3 :**

Si on achète : DOLLY GIRL BEAKER

Alors on achète aussi : DOLLY GIRL CHILDRENS BOWL

Support : 1.8 %

Confidence : Très élevée ➜ 77.8 %

Lift : 16.81 ➜ Très forte relation, ces articles sont très souvent achetés ensemble.

**Règle 4 :**

Inverse de la règle 3, avec confidence plus faible (38.9 %), mais le lift reste élevé (16.81) ➜ Cela montre une dépendance forte entre ces deux produits.

**Règle 5 :**

Si on achète : DOLLY GIRL CHILDRENS CUP

Alors on achète aussi : DOLLY GIRL BEAKER

Confidence : 37.5 %

Lift : 16.21 ➜ Encore une forte corrélation**.**

**Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.Clustering :**

Ce code agrège les données par client, en calculant la somme des quantités, la moyenne des prix unitaires, et le nombre de transactions pour chaque client. Il calcule également le montant total dépensé (TotalSpent) en multipliant la quantité par le prix unitaire, et supprime les clients avec des valeurs négatives pour la quantité ou le montant total. Ensuite, il nettoie les données en supprimant les outliers (valeurs aberrantes) en utilisant l'IQR (interquartile range) pour les colonnes Quantity et TotalSpent. Enfin, il normalise les données (quantité, prix unitaire, nombre d'achats, et total dépensé) à l'aide du StandardScaler pour les rendre prêtes à l'analyse.

Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.

Ce code applique l'algorithme de clustering KMeans pour diviser les clients en 3 groupes (ou clusters), en fonction des données normalisées (customer\_scaled). Le nombre optimal de clusters (optimal\_clusters = 3) est choisi, mais il peut être ajusté selon la méthode du coude. L'algorithme attribue ensuite chaque client à un cluster, et cette information est ajoutée à la colonne Cluster du DataFrame customer\_df.



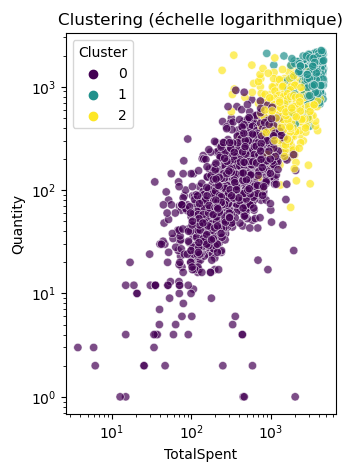
Silhouette Score: 0.46

Le **Silhouette Score** mesure la qualité du regroupement obtenu par un algorithme de clustering comme KMeans.

Dans ce cas, un **Silhouette Score de 0.46** indique une **bonne séparation entre les clusters**, même si ce n’est pas parfait. Cela signifie que les clients sont globalement bien segmentés, mais qu’il pourrait y avoir quelques zones d'amélioration.  
  
Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

Ce code trace un **nuage de points** (scatter plot) pour visualiser les résultats du clustering des clients, en fonction de leur **dépense totale** (TotalSpent) et de la **quantité achetée** (Quantity). Chaque point représente un client, et la couleur indique à quel **cluster** il appartient. Les axes sont mis à l’échelle **logarithmique** pour mieux visualiser les données lorsqu'elles couvrent une large plage de valeurs. Cela permet d’identifier visuellement les segments de clients selon leur comportement d’achat.



Une image contenant texte, Police, capture d’écran, ligne

Le contenu généré par l’IA peut être incorrect.Analyse des clusters :

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.Affichage des statistiques par cluster :

**Cluster 2 :**

Effectif (count) 997 clients

Profil : Acheteurs réguliers/modérés

Ils achètent beaucoup (608 produits en moyenne), et dépensent un montant moyen élevé (près de 1700 €).

C’est probablement le groupe cœur de clients fidèles.

**Cluster 1 :**

Effectif (count) 491 clients

Profil : Gros acheteurs, gros dépensiers

Ils achètent deux fois plus que le cluster 0, et dépensent presque le double aussi.

Ce sont les meilleurs clients, peut-être les grossistes ou des entreprises.

**Cluster 0 :**

Effectif (count) 2183 clients

Profil : Petits acheteurs réguliers

Le plus grand groupe (plus de 2000 clients), mais avec un panier moyen faible.

Ce sont peut-être des clients occasionnels, des particuliers, ou des nouveaux clients.