**USTHB** **2024-2025**

**Faculté d'informatique** **3ème Année CS**

**Cycle Ingéniorat**

# *Mini projet BI*

# *THÈME DU PROJET*

## *"Analyse des performances de vente et des comportements des clients dans un site e-commerce"*


**Présenté par :**

- **BENHAIK Meriem**
- **DAHMANI NAILA**
- **SADAOUI SARAH RAHMA**

**Encadré par :**

**Mme BERKANI LAMIA**

# Introduction

Dans un monde où les sites e-commerce jouent un rôle central dans le commerce moderne, nous avons identifié la nécessité d'analyser les performances de vente et les comportements des clients pour optimiser la compétitivité et la satisfaction des utilisateurs. Ce rapport vise à explorer ces dimensions clés en mettant en œuvre des approches avancées de gestion et d'analyse des données.

Nous avons conçu un entrepôt de données (data warehouse) pour centraliser les informations relatives aux ventes et aux comportements des clients. Nous avons également élaboré des requêtes OLAP (Online Analytical Processing) pour extraire des informations pertinentes et mis en place des outils de reporting afin de visualiser les résultats de manière claire et exploitable.

En parallèle, nous avons appliqué des techniques de Data Mining (DM), telles que le clustering, la classification et la génération de règles d'association. Ces algorithmes ont été développés à l'aide de Python, afin de répondre à des problématiques spécifiques, comme la segmentation des clients, la prédiction des comportements ou la découverte de relations cachées dans les données.

Enfin, nous avons illustré nos travaux par des exemples concrets, accompagnés d'explications détaillées et de visualisations, afin de démontrer leur utilité pour améliorer les performances commerciales et orienter les décisions stratégiques.

# Exploration et préparation des données

# Processus ETL :

**Nettoyage et analyse préliminaire des datasets Online Retail II et Customer Behavior**

Dans cette étude, nous avons utilisé deux datasets complémentaires pour analyser les transactions et comportements des clients dans un contexte de commerce électronique :

1. **Online Retail II** :
   Disponible sur Kaggle, ce dataset contient 525 461 lignes et 8 colonnes représentant des transactions réalisées sur une période donnée.

2. **Customer Behavior** :
   Disponible sur le UCI Machine Learning Repository, ce dataset fournit des informations détaillées sur le comportement des clients, telles que leurs interactions avec les produits, les fréquences d'achats, et d'autres indicateurs clés.

Les principales colonnes identifiées dans le dataset *Online Retail II* sont :

- **InvoiceNo** (Numéro de facture)
- **StockCode** (Code produit)
- **Description** (Description du produit)
- **Quantity** (Quantité achetée)
- **InvoiceDate** (Date de transaction)
- **UnitPrice** (Prix unitaire)
- **CustomerID** (Identifiant client)
- **Country** (Pays du client)

Ces données permettent d'étudier à la fois les performances commerciales et les comportements des clients.

Cependant, une première analyse a révélé plusieurs problèmes de qualité nécessitant un nettoyage approfondi :

1. **Doublons** :
   Nous avons détecté des lignes en double dans le dataset *Online Retail II*, ce qui a nécessité leur suppression pour éviter toute distorsion dans les résultats de l'analyse.

2. **Valeurs manquantes** :
   Des valeurs manquantes ont été identifiées dans les colonnes *CustomerID* et *Description*. Les lignes avec des identifiants clients manquants ont été supprimées, tandis que celles sans description ont été annotées pour un traitement ultérieur.

3. **Quantités négatives** :
   Des quantités négatives ont été observées, indiquant des retours ou des erreurs de saisie. Ces lignes ont été supprimées afin de ne pas fausser les résultats relatifs aux ventes.

4. **Incohérences de format** :
   Le format des dates dans la colonne *Invoice Date* était incohérent. Nous avons standardisé cette colonne pour garantir une analyse correcte des transactions dans le temps.

Après avoir identifié ces problèmes, nous avons procédé à un nettoyage des données en supprimant et en ajustant les lignes problématiques. Une version annotée et nettoyée des datasets a été générée, permettant ainsi de croiser les informations issues des deux sources et d’enrichir les analyses ultérieures.

## la Conception de l'Entrepôt de Données

Dans le cadre de ce projet , un entrepôt de données a été conçu pour centraliser et structurer les informations relatives aux transactions. Ce projet vise à fournir des insights permettant de maximiser les ventes et d'améliorer l'expérience client en identifiant les tendances d'achat et les produits les plus populaires. Afin de faciliter les analyses multidimensionnelles, un modèle de données en étoile a été choisi.

**Conception de l'Entrepôt de Données**

Le modèle en étoile a été adopté, ce qui consiste en une table des faits centrale qui est entourée de plusieurs tables de dimensions. La table de faits contient des données quantifiables sur les transactions (telles que le montant total des ventes, la quantité d'articles achetés, etc.), tandis que les tables de dimensions offrent un contexte descriptif pour ces transactions (comme des informations sur les produits, les clients, le temps, la géographie, etc.).

### A. Table des Faits

La table des faits est au cœur du modèle en étoile et contient les informations suivantes :

- **Price** : Le prix de l'article.
- **Quantity** : La quantité d'articles achetés dans la transaction.
- **CustomerID** : L'identifiant unique du client.
- **InvoiceID** : L'identifiant unique de la facture.
- **StockCode** : Le code produit de l'article acheté.
- **TotalSales** : Le montant total de la vente (calculé comme `Price * Quantity`).

**Structure de la Table des Faits (fact_sales) :**

```
CREATE TABLE fact_sales (
    InvoiceID INT,
    StockCode VARCHAR(50),
    Quantity INT,
    Price DECIMAL(10, 2),
    TotalSales DECIMAL(15, 2),
    CustomerID INT,
    PRIMARY KEY (InvoiceID, StockCode)
);
```

Cette table contient les mesures de chaque transaction réalisée sur le site e-commerce. Chaque ligne représente une vente unique et inclut des informations sur le produit acheté, la quantité, le client, la facture et le montant total de la vente.

## B. Tables de Dimensions

Les tables de dimensions fournissent des informations contextuelles qui aident à analyser les données transactionnelles dans la table des faits.

**Table des Produits (dim_product)**
Cette table contient des informations sur chaque produit vendu sur le site.

```
CREATE TABLE dim_product (
    StockCode VARCHAR(50) PRIMARY KEY,
    Price DECIMAL(10, 2)
);
```

1.
    - **StockCode** : Identifiant unique du produit.
    - **Price** : Prix du produit.

**Table des factures (dim_invoice)**
Cette table contient des informations sur les factures.

```
CREATE TABLE dim_invoice (
    id INT AUTO_INCREMENT PRIMARY KEY,
    InvoiceID INT,
    InvoiceDate DATETIME,
    TotalAmount DECIMAL (10, 2),
    UNIQUE KEY (InvoiceID, InvoiceDate)
);
```

- **id** : Identifiant unique et automatique pour chaque enregistrement dans la table.
- **InvoiceID** : Identifiant unique de la facture.
- **InvoiceDate** : Date et heure associées à la facture.
- **TotalAmount** : Montant total de la facture.

**Table du Temps (dim_time)**
Cette table permet de segmenter les ventes par périodes temporelles.

```
CREATE TABLE dim_time (
    DateID INT PRIMARY KEY AUTO_INCREMENT,
    Date DATE,
    Year INT,
    Month INT,
    Day INT,
    Weekday VARCHAR(10)
);
```

- **DateID** : Identifiant unique de la date.
- **Date** : La date de la transaction.
- **Year** : Année de la transaction.
- **Month** : Mois de la transaction.
- **Day** : Jour de la transaction.
- **Weekday** : Jour de la semaine, utile pour analyser les ventes selon le jour de la semaine.

**Table Géographique (dim_geography)**
Cette table contient des informations sur la géographie des clients. (table_2)

```
CREATE TABLE dim_geography (
   GeographyID INT PRIMARY KEY,
   Country VARCHAR(100),
   City VARCHAR(100)
);
```

2.

- **GeographyID** : Identifiant unique pour la géographie.
- **Country** : Pays du client.
- **City** : Ville du client.

**Table Géographique 2 (dim_geo)**
Cette table contient des informations sur la géographie des clients. (tableF)

```
CREATE TABLE dim_geography (
    GeographyID INT PRIMARY KEY,
    Country VARCHAR(100)
);
```

3.

- **GeographyID** : Identifiant unique pour la géographie.
- **Country** : Pays du client.
-

### C. Relations entre la Table des Faits et les Tables de Dimensions

Les tables de dimensions sont liées à la table des faits par des clés étrangères, permettant de croiser les données transactionnelles avec des informations contextuelles.

- **StockCode** dans la table des faits fait référence à **StockCode** dans la table des produits.
- **InvoiceID** dans la table des faits fait référence à **InvoiceID** dans la table des factures.
- **InvoiceDate** dans la table des faits fait référence à **DateID** dans la table du temps.
- **Country** dans la table des faits fait référence à **GeographyID** dans la table géographique.

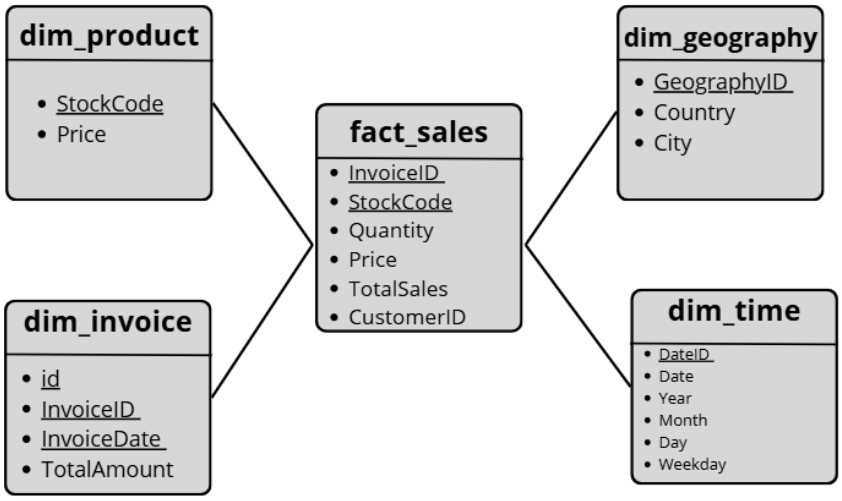



Figure 1 : **schéma en étoile de l'entrepôt de données**

**Analyse des Données**

En utilisant ce modèle, nous avons pu répondre à diverses questions analytiques, telles que :

1. **Analyse des ventes par produit** : Identifier les produits les plus vendus et analyser leur rentabilité.
2. **Segmentation des clients** : Analyser le comportement d'achat en fonction du genre, de l'âge, de la ville, et du total dépensé par les clients.
3. **Analyse temporelle des ventes** : Étudier les tendances des ventes sur différentes périodes (mensuelles, annuelles, par jour de la semaine, etc.).
4. **Analyse géographique des ventes** : Étudier les ventes par pays, ville, ou région pour comprendre les zones géographiques les plus rentables.

## Rôle de l'entrepôt de données dans cette étude

L'objectif principal de la création de cet entrepôt de données est de fournir aux décideurs du site e-commerce un accès direct à des informations cruciales pour prendre des décisions stratégiques basées sur des données solides et détaillées. L'entrepôt regroupe des informations clés telles que le prix, la quantité, le StockCode, l'ID du client, l'ID de la facture et le TotalSales, permettant d'obtenir une vision complète et précise des comportements d'achat des clients et de la performance des produits. Voici comment cet entrepôt de données soutient concrètement les décisions stratégiques des décideurs :

### 1. Optimisation des Stratégies de Pricing

L'analyse de la relation entre le **Price** et le **TotalSales** permet au décideur d'ajuster les prix en fonction du comportement des clients. Par exemple, l'examen de cette relation peut révéler si certains produits génèrent davantage de ventes ou de marges bénéficiaires. En fonction des résultats, il sera possible de définir des politiques de réductions de prix adaptées pour maximiser les revenus tout en maintenant la rentabilité. Cette stratégie est primordiale pour les périodes de forte concurrence ou lors de la mise en œuvre de nouvelles offres.

### 2. Personnalisation des Campagnes Marketing

Les données relatives aux clients (**CustomerID**, **TotalSales**, **Quantity**) permettent au décideur de segmenter efficacement les clients. Par exemple, les clients fréquents ou ceux ayant une forte valeur d'achat peuvent être ciblés par des offres spéciales et des campagnes de fidélisation. De même, les clients inactifs peuvent recevoir des offres de réactivation. En utilisant cette segmentation, le décideur peut maximiser l'efficacité des campagnes marketing et atteindre des segments spécifiques, augmentant ainsi le taux de conversion et la rentabilité des promotions.

### 3. Prévision des Ventes et Gestion des Stocks

En analysant les tendances des ventes à partir du **StockCode** et des **TotalSales**, l'entrepôt permet de prévoir la demande future pour chaque produit. Cela aide à anticiper les tendances et à ajuster les niveaux de stock pour éviter les ruptures ou les excédents. Par ailleurs, l'analyse des données temporelles liées aux ventes (**InvoiceDate**) permet de prédire les pics de demande durant certaines périodes de l'année (fêtes, soldes, etc.), facilitant ainsi une gestion plus précise des stocks.

### 4. Amélioration de l'Expérience Client

L'analyse des comportements d'achat des clients à travers les colonnes **Quantity**, **StockCode** et **TotalSales** permet au décideur de personnaliser l'expérience d'achat en fonction des préférences individuelles. Cela inclut la mise en place de recommandations de produits basées sur les achats passés, ou la création de promotions sur mesure pour des segments de clients spécifiques, améliorant ainsi la satisfaction et la fidélisation des clients.

### 5. Évaluation de la Rentabilité des Produits

L'accès aux données relatives aux produits (**StockCode**) et aux **TotalSales** permet au décideur de déterminer quels produits sont les plus rentables. Par exemple, certains produits peuvent générer un faible volume de ventes mais une marge bénéficiaire élevée. En analysant ces données, le décideur peut ajuster les stratégies de marketing pour mettre davantage en avant ces produits ou adapter les promotions en fonction de la rentabilité plutôt que du volume de vente seul.

### 6. Suivi de la Performance et Prise de Décisions Rapides

Grâce à un accès en temps réel aux données des ventes, des quantités et des performances des produits, les décideurs peuvent prendre des décisions rapides et adaptées. Par exemple, si une promotion ne donne pas les résultats attendus, elle peut être ajustée ou remplacée immédiatement, permettant ainsi une gestion dynamique et réactive de l’offre commerciale.

### 7. Segmentation Avancée et Ciblage

Les données sur les clients (**CustomerID**, **TotalSales**, **Quantity**) permettent au décideur de créer des segments comportementaux et démographiques plus avancés. Ces segments peuvent inclure des groupes spécifiques, comme des acheteurs de produits de luxe ou des clients qui n’ont pas acheté depuis plusieurs mois. Cette segmentation fine permet de créer des campagnes marketing plus efficaces et mieux ciblées, en maximisant le retour sur investissement des efforts publicitaires.

### 8. Détection des Tendances et Anticipation des Besoins

L’analyse des ventes (**TotalSales** et **Quantity**) par produit et par période permet au décideur de repérer les tendances émergentes. Par exemple, une forte augmentation de la demande pour un certain type de produit ou une catégorie particulière peut signaler un besoin croissant de réapprovisionnement ou d’adaptation des stratégies de marketing. Cette capacité à anticiper les besoins aide le décideur à rester proactif dans la gestion des offres et des stocks.

### 9. Optimisation de la Stratégie de Fidélisation

L’analyse de la fréquence des achats et des **TotalSales** par client permet de repérer les clients les plus fidèles. Ces clients peuvent ensuite être récompensés par des offres exclusives, des

réductions ou des accès à des programmes de fidélité. Cela contribue à renforcer la relation client et à favoriser un cycle d’achat répétitif, augmentant ainsi la valeur vie client (CLV).

### 10. Prédictions Basées sur les Données

Avec les données historiques des ventes et des comportements d'achat, le décideur peut utiliser des modèles prédictifs pour anticiper les ventes futures. Cela permet de mieux planifier les promotions, la gestion des stocks et l'allocation des ressources, en réduisant les risques liés à une mauvaise planification ou à des promotions mal ciblées.

L'entrepôt de données conçu pour l’analyse du comportement des clients sur le site e-commerce est un outil stratégique puissant, permettant aux décideurs de prendre des décisions plus éclairées et plus efficaces. En ayant accès à des informations détaillées sur les produits, les clients et les transactions, le décideur peut ajuster ses stratégies de pricing, personnaliser ses campagnes marketing, optimiser la gestion des stocks, et améliorer l’expérience client, tout en maximisant la rentabilité du site e-commerce. Cet entrepôt devient ainsi un levier incontournable pour une prise de décision agile et stratégique dans un environnement e-commerce de plus en plus compétitif.

## Choix Technologiques pour la Mise en Œuvre de l'Entrepôt de Données

Pour la mise en œuvre de cet entrepôt de données, nous avons choisi d'utiliser **MySQL** comme système de gestion de base de données relationnelle. Cette solution offre une grande souplesse pour gérer les relations entre les tables, effectuer des jointures complexes, et garantir la performance des requêtes. MySQL est également bien adapté pour stocker et manipuler de grandes quantités de données tout en assurant la fiabilité et la scalabilité. Afin d'automatiser l'intégration des données provenant des sources existantes, nous avons opté pour l'utilisation d'outils **ETL (Extract, Transform, Load)**, permettant de rationaliser le processus d'extraction, de transformation et de chargement des données dans l'entrepôt.

Les étapes techniques pour réaliser l'entrepôt de données étaient les suivantes :

1. **Création des tables** : Nous avons utilisé des requêtes SQL pour créer la table des faits ainsi que les différentes tables de dimensions. À cette étape, nous avons défini les types de données appropriés pour chaque colonne, afin d'assurer la cohérence des données et d'optimiser la gestion des informations.

2. **Chargement des données** : Pour intégrer les données dans l'entrepôt, nous avons utilisé des outils ETL pour extraire les informations des sources existantes, les transformer (par exemple, en calculant des agrégations telles que les ventes totales, les quantités vendues, etc.), puis les charger dans les tables correspondantes de l'entrepôt.

3. **Optimisation des performances** : Afin d'améliorer les performances des requêtes, nous avons créé des index sur les clés primaires et étrangères des tables de faits et des dimensions. Cela a permis de réduire le temps de réponse des requêtes complexes et d'assurer une analyse rapide des données.

## Requetes OLAP

1. **Ventes Totales par Produit**

```sql
SELECT dp.StockCode,
       dp.Price,
       SUM(fs.TotalSales) AS TotalSalesByProduct
FROM fact_sales fs
JOIN dim_product dp ON fs.StockCode = dp.StockCode
GROUP BY dp.StockCode, dp.Price
ORDER BY TotalSalesByProduct DESC;
```

Cette requête permet de calculer les ventes totales pour chaque produit. Elle regroupe les transactions de la table des faits par **StockCode**, et la somme des **TotalSales** est calculée pour chaque produit. Le prix de chaque produit est également récupéré de la table `dim_product`. Les résultats sont triés de manière décroissante pour identifier les produits ayant généré les ventes les plus importantes. Cela permet de repérer les produits les plus populaires et les plus rentables sur le site e-commerce.

**2. Ventes Moyennes Mensuelles**

```sql
SELECT dt.Year,
       dt.Month,
       AVG(fs.TotalSales) AS AverageMonthlySales
FROM fact_sales fs
JOIN dim_time dt ON fs.InvoiceDate = dt.DateID
GROUP BY dt.Year, dt.Month
ORDER BY dt.Year, dt.Month;
```

Cette requête calcule la vente moyenne par mois. En regroupant les ventes mensuelles grâce aux informations de la table `dim_time` (Year et Month), elle calcule la moyenne des TotalSales pour chaque mois. Cette analyse est utile pour observer les tendances des ventes sur une période donnée et identifier des mois plus performants que d'autres. Par exemple, cela peut aider à détecter les périodes de promotions ou de forte activité.

### 3. Ventes Totales par Mois

```sql
SELECT dt.Year,
       dt.Month,
       SUM(fs.TotalSales) AS TotalMonthlySales
FROM fact_sales fs
JOIN dim_time dt ON fs.InvoiceDate = dt.DateID
GROUP BY dt.Year, dt.Month
ORDER BY dt.Year, dt.Month;
```

Cette requête calcule les ventes totales réalisées chaque mois en faisant la somme des **TotalSales** par mois, en fonction des données de la table `dim_time`. Cela permet d'analyser l'évolution des ventes sur plusieurs mois, d'identifier les pics de vente (par exemple, en raison de promotions ou de fêtes de fin d'année) et de mieux comprendre la saisonnalité des produits. Ces informations sont essentielles pour la planification des stocks et des campagnes marketing.

### 4. Ventes Totales par Pays

```sql
SELECT dg.Country,
       SUM(fs.TotalSales) AS TotalSalesByCountry
FROM fact_sales fs
JOIN dim_geography dg ON fs.Country = dg.GeographyID
GROUP BY dg.Country
ORDER BY TotalSalesByCountry DESC;
```

Cette requête permet d'analyser les ventes totales par pays en croisant les données de la table des faits avec les informations géographiques de la table `dim_geography`. Elle permet de comprendre dans quelles zones géographiques les produits sont les plus populaires et où les

revenus sont les plus importants. Ces informations aident à orienter les efforts marketing et les stratégies commerciales en fonction des régions géographiques les plus rentables.

À partir des informations extraites de ces requêtes OLAP, les décideurs peuvent élaborer des stratégies plus efficaces pour augmenter les ventes et maximiser la rentabilité dans le secteur du e-commerce.

Voici comment ces analyses se traduisent en actions concrètes :

1. **Gestion des produits populaires** : L'identification des produits les plus rentables grâce aux **ventes totales par produit** permet de prioriser leur mise en avant sur le site. Ces produits peuvent être présentés sur la page d'accueil, dans les newsletters, ou intégrés à des offres groupées pour encourager les achats additionnels. Par ailleurs, cette analyse guide les décisions d'approvisionnement afin de maintenir un stock suffisant des articles les plus demandés.

2. **Optimisation des campagnes marketing et publicitaires** : Les résultats des **ventes mensuelles moyennes et totales** mettent en évidence les périodes de forte activité, comme les fêtes de fin d'année ou les périodes de soldes. Ces insights permettent de planifier des campagnes publicitaires ciblées (via Google Ads, Facebook Ads, etc.) et des promotions spéciales pendant ces périodes. Par exemple, en intégrant les mots-clés des produits les plus recherchés, on peut optimiser le **SEO** (Search Engine Optimization) pour améliorer le classement du site dans les résultats de recherche, augmentant ainsi le trafic organique.

3. **Offres personnalisées et fidélisation** : L'analyse des **ventes par pays** permet de mieux comprendre les préférences régionales et d'adapter les offres en conséquence. Par exemple, des promotions spécifiques ou des remises peuvent être proposées dans les régions où les ventes sont encore faibles. De plus, les informations sur les produits les plus populaires dans chaque zone géographique peuvent être utilisées pour des campagnes e-mailing personnalisées ou des recommandations automatiques sur le site, améliorant l'expérience client et la fidélisation.

4. **Promotion des jours stratégiques** : L'analyse des ventes par période temporelle, en croisant les données de la table temporelle (dim_time), met en lumière les jours ou les moments de la semaine où les ventes sont les plus élevées. Cela permet de concentrer les efforts publicitaires, comme des promotions éclairs ou des codes de réduction

valables à des moments précis, pour maximiser l’impact des campagnes.

5. **Stratégies de remarketing** : Les données issues de ces analyses peuvent également alimenter des campagnes de remarketing. Par exemple, cibler les clients ayant acheté des produits populaires ou ayant consulté des pages spécifiques avec des offres sur mesure contribue à augmenter le taux de conversion.

Grâce à ces analyses, les décideurs disposent d'une base solide pour ajuster leurs stratégies de marketing, de publicité et de gestion des produits. En intégrant ces données à des outils comme Google Analytics, des CRM ou des plateformes publicitaires, ils peuvent automatiser des actions ciblées qui non seulement augmentent les ventes, mais renforcent également la satisfaction et la fidélité des clients.

## Les algorithmes de DM

**Combinaison des 2 algorithmes de K-means et Apriori (Hybridation)**

**Objectif**

L'objectif de cette analyse est de segmenter les clients et d'identifier des relations entre les produits achetés. Nous avons utilisé deux algorithmes de machine learning :

1. **K-Means** pour segmenter les clients en groupes homogènes.
2. **Apriori** pour découvrir des règles d'association entre les produits achetés ensemble.

Ces analyses ont été effectuées sur une base de données transactionnelle d'e-commerce afin de fournir des insights utiles pour optimiser les stratégies marketing et la gestion des stocks.

## 1. Segmentation des Clients avec K-Means

**Objectif de la Segmentation**

Nous avons cherché à segmenter les clients en fonction de deux critères principaux :

- **Total des dépenses** (`TotalSpending`).
- **Nombre de transactions** (`TransactionCount`).

Ces informations ont été extraites à partir de la table `fact_sales` contenant les données des ventes.

**Méthode**

1. **Chargement des Données** : La requête SQL récupère le `CustomerID`, le total des dépenses et le nombre de transactions pour chaque client.
2. **Normalisation des Données** : Les variables `TotalSpending` et `TransactionCount` ont été normalisées avec un **StandardScaler** afin d'éviter les biais dus aux échelles différentes.

3. **Application de K-Means** : L'algorithme **K-Means** a été appliqué pour diviser les clients en **3 clusters** :

    **Cluster 0** : Clients fréquents avec des dépenses faibles.

    **Cluster 1** : Clients modérés avec des dépenses moyennes.

    **Cluster 2** : Clients premium avec des dépenses élevées mais moins fréquentes.

**Visualisation des Résultats**

Un graphique de dispersion a été généré pour visualiser la répartition des clients dans les différents clusters, en utilisant les variables `TotalSpending` et `TransactionCount`.

**Actions Recommandées**

- **Cluster 0 (Clients fréquents, faibles dépenses)** : Mettre en place un programme de fidélité et offrir des réductions sur les achats groupés.
- **Cluster 1 (Clients réguliers, dépenses moyennes)** : Proposer des promotions personnalisées et encourager les achats répétitifs.
- **Cluster 2 (Clients premium, dépenses élevées)** : Offrir des services VIP, comme la livraison gratuite ou un accès prioritaire aux nouveaux produits.

## 2. Analyse des Règles d'Association avec Apriori

**Objectif de l'Analyse**

L'objectif de cette analyse est de découvrir des associations entre les produits achetés ensemble. Cela permet de déterminer quelles combinaisons de produits sont fréquemment achetées dans le cadre des mêmes transactions, facilitant ainsi la mise en place de stratégies de **vente croisée**.

**Méthode**

1. **Chargement des Données** : Une requête SQL est utilisée pour récupérer les transactions (référencées par `InvoiceID`) et les codes des produits achetés (`StockCode`).
2. **Transformation des Données** : Les données sont transformées en une matrice binaire où chaque ligne représente une facture, et chaque colonne représente un produit. Un produit est marqué par 1 si acheté, sinon par 0.
3. **Application de l'Algorithme Apriori** : L'algorithme **Apriori** a été appliqué pour identifier les ensembles fréquents de produits, avec un seuil de **support minimal de 5%**.
4. **Génération des Règles d'Association** : Les règles d'association ont été extraites à l'aide de la métrique **lift**, ce qui nous permet de déterminer la force de l'association entre les produits.

**Résultats**

Les règles d'association identifiées indiquent quelles paires de produits sont fréquemment achetées ensemble. Par exemple, une règle pourrait indiquer qu'un client qui achète un produit A a également de fortes chances d'acheter un produit B.

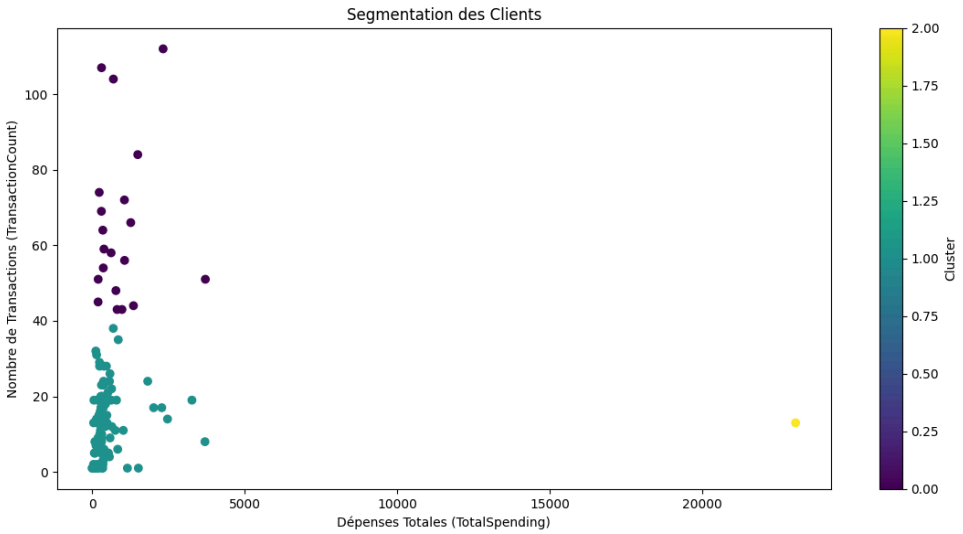
**Actions Recommandées**

- **Vente Croisée** : Proposer des produits complémentaires lors de l'achat d'un produit principal.
- **Packs Promotionnels** : Créer des offres combinées pour des produits fréquemment achetés ensemble.
- **Optimisation des Points de Vente** : Placer les produits associés près les uns des autres dans les rayons ou les catalogues en ligne pour faciliter les achats groupés.

## 3. Sauvegarde des Résultats

Les résultats des analyses de segmentation et des règles d'association ont été exportés dans des tables SQL pour un usage ultérieur :

- **Segmentation des clients** : Une nouvelle table `customer_segmentation` a été créée pour stocker les clients segmentés par cluster.
- **Règles d'association** : Les règles générées ont été sauvegardées dans la table `association_rules` pour une consultation ultérieure.




Cette analyse nous a permis de mieux comprendre les comportements des clients grâce à la segmentation et à la découverte des associations de produits. Les résultats permettent de cibler les actions marketing en fonction des profils clients et d'optimiser la vente croisée. En

appliquant ces insights, l'entreprise pourra améliorer sa stratégie de fidélisation, augmenter ses revenus et maximiser l'efficacité de ses promotions.

## La réalisation du Dashboard

Dans le cadre de notre projet d'analyse du comportement des clients, nous avons conçu et développé un dashboard interactif à l'aide de la bibliothèque Dash de Python.

### Outils et technologies

Nous avons utilisé les outils et technologies suivants pour réaliser ce Dashboard :

- **Python** : Langage principal pour l'analyse et la visualisation des données.
- **Pandas** : Bibliothèque pour manipuler les données tabulaires.
- **SQLAlchemy** : Outil pour créer une connexion avec la base de données MySQL.
- **Dash** : Framework pour développer des applications web interactives.
- **Plotly Express** : Bibliothèque pour créer des visualisations avancées.

### Étapes de réalisation

1. **Connexion à la base de données** Nous avons utilisé la bibliothèque SQLAlchemy pour établir une connexion à la base de données MySQL. La requête SQL « SELECT * FROM `table 2` » a permis de charger les données dans un DataFrame Pandas pour un traitement ultérieur.

2. **Préparation des visualisations** Nous avons conçu cinq visualisations interactives :

   **Graphique circulaire** : Répartition des clients par genre.

   **Graphique à barres** : Dépenses totales des clients par ville.

   **Histogramme** : Distribution des évaluations moyennes.

   **Graphique circulaire** : Répartition des types de membres.

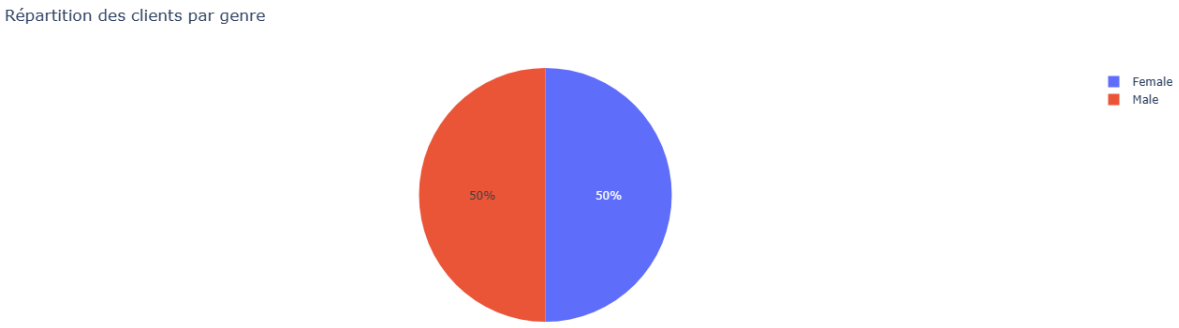
**Graphique à barres** : Relation entre le niveau de satisfaction et les dépenses totales.

3. **Conception de la mise en page** La mise en page du dashboard a été conçue pour offrir une navigation claire et intuitive. Nous avons utilisé des grilles pour organiser les visualisations en plusieurs sections.

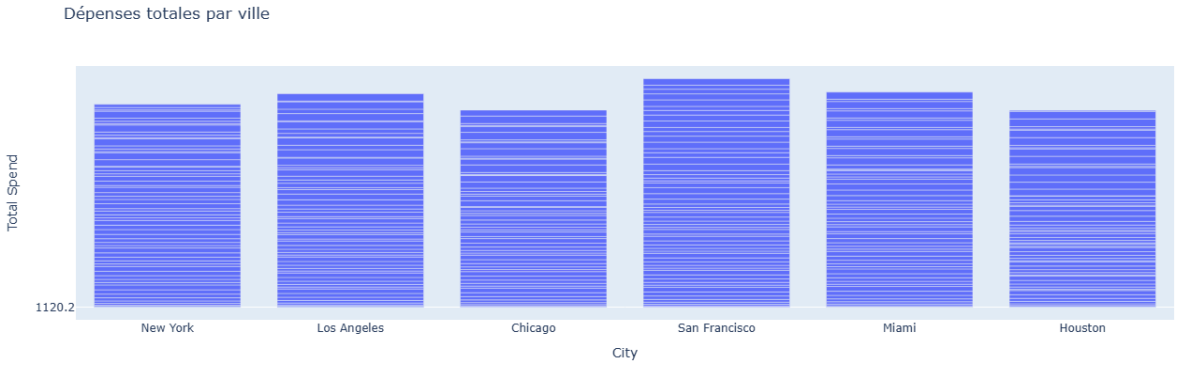
4. **Déploiement** L'application est exécutée localement à l'aide du serveur Dash, avec l'option « debug=True » activée pour faciliter le développement.

## Résultats obtenus

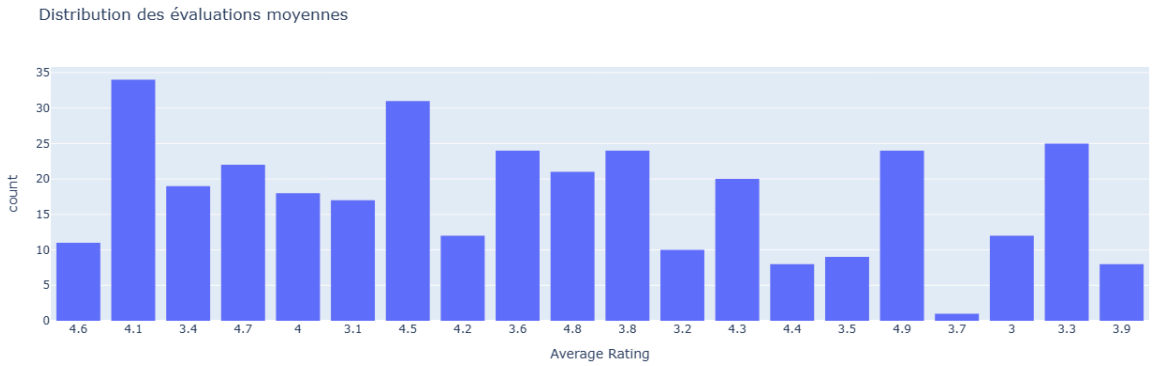
Les visualisations permettent de tirer les enseignements suivants :



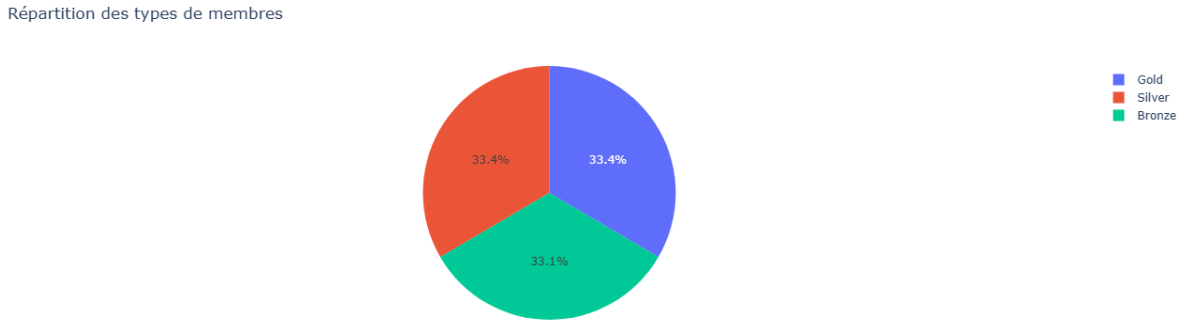

- **Genre** : La répartition des clients montre un équilibre entre les genres ou des dominances significatives.



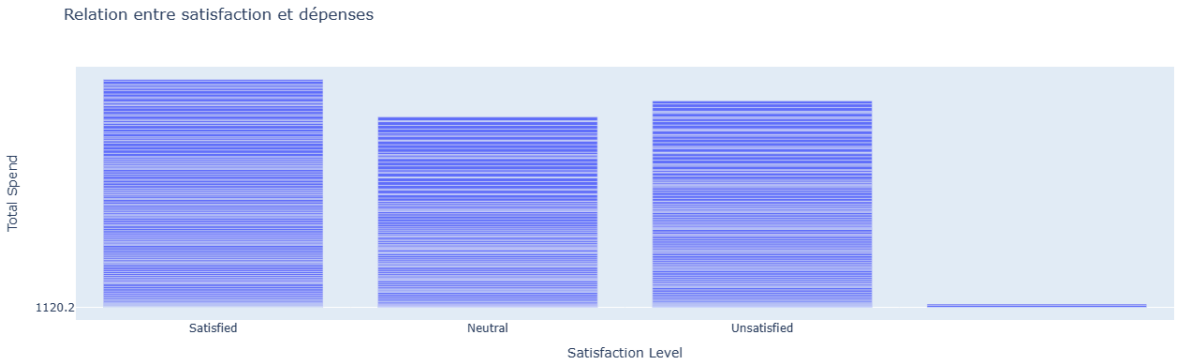
- **Dépenses par ville** : Certaines villes se distinguent par des dépenses plus élevées.




- **Évaluations moyennes** : La plupart des clients attribuent des notes comprises dans une plage spécifique.




- **Types de membres** : Les clients appartiennent majoritairement à certains types de membres.




- **Satisfaction et dépenses** : Les clients plus satisfaits tendent à dépenser davantage.

# Conclusion

En conclusion, ce rapport démontre que l'analyse des données est un levier fondamental pour améliorer les performances des sites e-commerce, un domaine où la concurrence est de plus en plus forte. Nous avons développé un entrepôt de données permettant de centraliser et de structurer les informations relatives aux ventes et aux comportements des clients, facilitant ainsi leur analyse. À travers l'utilisation des requêtes OLAP, nous avons pu extraire des informations stratégiques, telles que les produits les plus rentables, les habitudes d'achat des clients et les tendances de consommation. Ces analyses ont permis de dresser un portrait plus précis des comportements des utilisateurs.

De plus, en appliquant des techniques avancées de Data Mining, comme le clustering, la classification et la génération de règles d’association, nous avons pu approfondir la segmentation des clients et prédire leurs futurs comportements d’achat. Ces modèles ont été mis en œuvre à l'aide de Python, offrant une approche agile et personnalisée face aux besoins du projet. Les résultats obtenus ont révélé des relations insoupçonnées, permettant d'identifier des opportunités d'optimisation pour chaque segment de client.

Les exemples concrets, accompagnés de visualisations interactives, ont montré comment ces données peuvent être transformées en actions tangibles pour les décideurs. Par exemple, nous avons illustré comment une meilleure segmentation des clients pourrait mener à des campagnes marketing plus ciblées et à une gestion des stocks plus efficace. De plus, les analyses de performance ont permis de mieux orienter les décisions stratégiques, garantissant ainsi une compétitivité accrue sur le marché.

En somme, ce travail ouvre la voie à une gestion plus fine et plus précise des opérations e-commerce. En exploitant pleinement les données disponibles, les entreprises peuvent non seulement maximiser leurs résultats commerciaux mais aussi offrir une expérience utilisateur enrichie et personnalisée, renforçant ainsi leur position sur le marché.

## Bibliographie


- **MySQL** : Système de gestion de bases de données relationnelles utilisé pour stocker, structurer et interroger les données transactionnelles.
  **Référence** : Oracle Corporation. (2025). *MySQL Documentation*. Disponible en ligne : https://dev.mysql.com/doc/.

- **Python et bibliothèques associées** : Langage utilisé pour l'analyse des données et la création d'applications interactives.

  **Pandas** : Pour la manipulation et l'analyse des données tabulaires.
  **Référence** : Wes McKinney. (2017). *Python for Data Analysis* (2nd ed.). O'Reilly Media.

  **Matplotlib et Seaborn** : Pour la création de visualisations statiques et informatives.
  **Référence** : Hunter, J. D. (2007). "Matplotlib: A 2D Graphics Environment." *Computing in Science & Engineering*, 9(3), 90–95.

  **NumPy** : Pour les calculs numériques et la manipulation de tableaux multidimensionnels.
  **Référence** : Oliphant, T. E. (2015). *Guide to NumPy* (2nd ed.).

  **Dash** : Framework Python pour développer des applications analytiques interactives, avec des composants pour la visualisation de données et l'interaction utilisateur.
  **Référence** : Plotly Technologies Inc. (2025). *Dash User Guide and Documentation*. Disponible en ligne : https://dash.plotly.com/.

- **Outils ETL** : Techniques appliquées pour extraire les données brutes, effectuer des transformations (agrégations, nettoyages, calculs) et les charger dans un entrepôt de données.
  **Référence** : Kimball, R., & Caserta, J. (2004). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley.

- **Entrepôt de données** : Conception d'un entrepôt de données centralisé, incluant des tables dimensionnelles et des tables de faits, avec optimisation des performances via des index.
  **Référence** : Inmon, W. H. (2005). *Building the Data Warehouse* (4th ed.). Wiley.

- **Modèles de données** : Modélisation des données e-commerce en étoile, permettant une navigation facile pour les analyses OLAP.
  **Référence** : Kimball, R. (1996). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (2nd ed.). Wiley.

- **Visualisation des données et reporting interactif** : Création de tableaux de bord interactifs avec Dash pour permettre aux décideurs d'explorer les données en temps réel.
  **Référence** : Plotly Technologies Inc. (2025). *Dash Enterprise*. Disponible en ligne : https://plotly.com/dash/.

- **Données de retail online** : Jeu de données simulant des transactions clients dans un environnement e-commerce, utilisé pour extraire des indicateurs clés comme les ventes par pays, les dépenses moyennes par client, etc.
  **Référence** : UCI Machine Learning Repository. (n.d.). *Online Retail Dataset*. Disponible en ligne : https://archive.ics.uci.edu/ml/.