
Titre du mini-projet

> **Mini-projet : Analyse de paniers et recommandations produits**
> **Algorithmes Apriori et FP-Growth sur données réelles**

---

1. Contexte

Une enseigne de e-commerce met à votre disposition ses données de ventes. Chaque ligne du fichier correspond à un produit vendu dans une transaction (facture). L'objectif de ce mini-projet est d'identifier des **associations d'articles** (règles du type :

> *SI un client achète X, ALORS il achète souvent Y*)

afin de proposer des **recommandations produits** (cross-selling, bundles, agencement des rayons, etc.).

Vous utiliserez principalement :

- l'algorithme **Apriori** ;
- l'algorithme **FP-Growth** ;
- les fonctions de génération de **règles d'association** (support, confiance, lift).

---

2. Objectifs pédagogiques

À l'issue de ce mini-projet, l'étudiant devra être capable de :

1. **Préparer un jeu de données transactionnel** à partir de données brutes (fichiers CSV / Excel).
2. Appliquer **Apriori** et **FP-Growth** pour extraire des **itemsets fréquents** et des **règles d'association**.
3. Interpréter correctement les mesures : **support**, **confiance**, **lift**.
4. Traduire ces règles en **recommandations métier concrètes** (offres combinées, placement produits, "Vous aimerez aussi…").
5. Comparer brièvement **Apriori vs FP-Growth** (temps de calcul, nombre de règles, lisibilité).

---

3. Données et environnement

- Jeux de données possibles (au moins un à utiliser) :
  - `order_data.csv` : petit dataset de démonstration (quelques dizaines de transactions).

  - `Online Retail.xlsx` ou `transaction_dataset.csv` : données réelles avec plusieurs milliers de transactions.

- Environnement de travail :
  - Python (Jupyter Notebook / Google Colab),
  - Bibliothèques : `pandas`, `numpy`, `matplotlib` ou `seaborn` (optionnel), `apyori` ou `mlxtend.frequent_patterns` pour Apriori, `mlxtend.frequent_patterns` pour FP-Growth.

---

4. Travail à réaliser

Partie 0 – Mise en route (Apriori sur petit dataset)

1. Charger le fichier **`order_data.csv`**.
2. Construire une **liste de listes** contenant, pour chaque transaction, la liste des articles achetés.
3. Appliquer l'algorithme **Apriori** (par exemple avec `apyori`) en utilisant des paramètres proposés par l'enseignant, par exemple :
   - `min_support = 0.25`
   - `min_confidence = 0.2`
   - `min_lift = 2`
   - `min_length = 2`
4. Afficher quelques règles d'association (3 à 5 règles) et, pour **au moins deux d'entre elles** :
   - écrire la règle sous la forme : *{X} ⇒ {Y}* ;
   - interpréter **support**, **confiance**, **lift** dans un langage simple (métier).

> *Objectif de la partie 0 :* se remettre en tête le fonctionnement d'Apriori et le sens des métriques.

Partie 1 – Préparation des données réelles

1. À partir du fichier **`Online Retail.xlsx`** ou **`transaction_dataset.csv`** :
   - garder uniquement les colonnes utiles (par exemple : ID de facture, ID produit, quantité, éventuellement pays ou date) ;
   - nettoyer les données : suppression des valeurs manquantes, des lignes incohérentes, etc.
2. Construire le jeu de données **transactionnel** :
   - chaque transaction = une facture ;
   - chaque transaction = liste des produits achetés dans cette facture.

3. Dans le rapport, indiquer :
    - **nombre total de transactions** ;
    - **nombre d'articles différents** ;
    - donner **3 exemples** de transactions (listes d'articles).

Partie 2 – Extraction de règles avec FP-Growth

1. Encoder les transactions :
    - utiliser `TransactionEncoder` ou les fonctions de `mlxtend` pour transformer la liste de listes en tableau booléen (one-hot) ;
    - obtenir un `DataFrame` dont chaque colonne correspond à un article et chaque ligne à une transaction (True/False).
2. Appliquer **FP-Growth** :
    - choisir une valeur de `min_support` raisonnable (ex. 0.02 ou 0.03, à ajuster selon la taille du dataset) ;
    - lister les **10 itemsets fréquents** ayant le plus grand support.
3. Générer des **règles d'association** :
    - utiliser `association_rules` (`mlxtend`) avec `metric="confidence"` et une valeur de `min_threshold` (ex. 0.5 ou 0.6) ;
    - trier les règles par **confiance** ou **lift**.
4. Sélectionner **5 règles intéressantes** :
    - support ni trop faible ni trop élevé ;
    - confiance élevée ;
    - lift > 1 (idéalement > 1.2).

Pour chaque règle sélectionnée, vous devez :

- la réécrire en toutes lettres (ex. : « Les clients qui achètent A et B achètent souvent C ») ;
- expliquer ce que signifient **support**, **confiance**, **lift** de cette règle ;
- proposer **au moins une action marketing** (bundle, promotion, rangement dans le magasin, recommandation en ligne, etc.).

Partie 3 – Comparaison Apriori vs FP-Growth

1. Appliquer **Apriori** sur le même dataset réel (ou sur un sous-échantillon) avec des paramètres compatibles (`min_support`, etc.).
2. Comparer Apriori et FP-Growth sur :
    - le **temps de calcul** (approximatif, par ex. mesure avec `%time`),
    - le **nombre d'itemsets fréquents** et/ou de **règles** générées,

- la **lisibilité** des résultats (trop de règles, règles redondantes, etc.).

3. Dans la conclusion du rapport, répondre aux questions suivantes :
   - Quel algorithme (Apriori ou FP-Growth) recommanderiez-vous pour ce type de données, et pourquoi ?
   - Quelles sont, selon vous, les **limites** des règles d'association sur ce dataset ?
   - Proposez **deux pistes d'amélioration** (segmentation par pays, par période, filtrage de certains articles, visualisation de graphes d'items, etc.).

---

5. Livrables attendus

1. **Un notebook Jupyter (.ipynb)** contenant :
   - le code de préparation des données,
   - le code Apriori,
   - le code FP-Growth,
   - l'extraction et la sélection des règles,
   - quelques graphiques simples (facultatif) : top des produits, distribution des supports, etc.
   - des **cellules Markdown** expliquant vos choix et vos résultats.
2. **Un rapport synthétique (3–4 pages)** au format PDF, comprenant :
   - une **introduction** (contexte et objectif du mini-projet),
   - la **méthodologie** (préparation des données, choix des paramètres, description des algorithmes utilisés),
   - les **principaux résultats** (tableaux ou captures des règles retenues avec interprétation),
   - des **recommandations métier** pour l'enseigne,
   - une **conclusion** incluant la comparaison Apriori / FP-Growth.

---

2. Adaptation / enrichissement

A. Objectifs pédagogiques (version Master)

1. **Configurer et justifier les hyperparamètres** (min_support, min_confidence, métrique de tri, taille minimale/maximale des itemsets), à partir d'arguments théoriques et empiriques.
2. Mener une **analyse de sensibilité** de ces hyperparamètres sur :
   - le nombre d'itemsets ;
   - le type de règles obtenues ;

- l'intérêt métier des règles.

3. **Segmenter** le dataset (par pays, période, type de client, etc.) et comparer les règles obtenues dans chaque segment.
4. Discuter les **limites** des règles d'association (corrélation vs causalité, biais de popularité, bruit, saisonnalité, etc.).
5. Proposer **une ébauche de protocole d'évaluation** si ces règles étaient utilisées dans un système de recommandation (A/B test, métriques offline…).

B. Extensions demandées (à ajouter à la Partie 2 ou 3)

1. **Analyse de sensibilité (obligatoire)**
   - Faire varier `min_support` (par exemple 0.01, 0.02, 0.03, 0.05) et observer :
     - le nombre d'itemsets fréquents ;
     - le nombre de règles générées ;
     - le profil des règles (règles triviales vs règles plus rares mais plus informatives).
   - Présenter les résultats dans un **tableau ou un graphique** et commenter.
2. **Segmentation du dataset (obligatoire)** Sur le dataset réel, choisir un critère de segmentation (par exemple :
   - un pays (UK vs autres),
   - une période (haut/bas saison),
   - une catégorie de produits.

   Pour **au moins deux segments**, répéter :
   - l'extraction d'itemsets fréquents ;
   - la génération de règles d'association ;
   - la sélection de 3 à 5 règles intéressantes par segment.

   Dans le rapport, comparer les segments :
   - Quelles règles sont communes ?
   - Quelles règles sont spécifiques à un segment ?
   - Quelles recommandations métier différentes peut-on proposer selon les segments ?
3. **Lien avec les systèmes de recommandation (obligatoire)** Rédiger une **courte section** (~1 page) qui explique :
   - comment ces règles d'association pourraient être intégrées dans un **système de recommandation** (par exemple : règles utilisées pour un module "Frequently bought together") ;

- quelles seraient les **métriques d'évaluation** pertinentes : taux de clic, taux de conversion, panier moyen, etc. ;
- quel protocole d'**expérimentation** (A/B testing, offline metrics) l'entreprise pourrait mettre en place.

4. **Discussion critique (obligatoire)** Inclure une discussion plus critique sur :
   - les limites méthodologiques (corrélation vs causalité, données manquantes, biais de saison) ;
   - les risques potentiels d'un mauvais usage des règles (sur-personnalisation, sur-exposition de certains produits, effets sur la diversité des ventes) ;
   - des pistes de couplage avec d'autres approches (collaborative filtering, modèles séquentiels, etc.).

C. Barème indicatif

- Préparation des données et clarté du notebook : **20 %**
- Application correcte des algorithmes (Apriori, FP-Growth) : **20 %**
- Analyse de sensibilité et segmentation : **25 %**
- Interprétation métier et recommandations : **20 %**
- Qualité du rapport (structure, figures, esprit critique, bibliographie éventuelle) : **15 %**
