

## Utilisation de l'algorithme Apriori

Dans le cadre de notre projet de data mining visant à améliorer l'expérience utilisateur d'une boutique en ligne, nous avons exploré une alternative à l'algorithme FP-Growth. En plus de l'approche basée sur FP-Growth que nous avons déjà présentée, nous avons également développé un code utilisant l'algorithme Apriori pour extraire des règles d'association à partir de nos données e-commerce.

### Implémentation de l'Algorithme Apriori :

Le code utilisant l'algorithme Apriori suit une démarche similaire à celle de FP-Growth. Nous avons effectué des étapes de prétraitement, telles que la suppression des valeurs manquantes et des produits offerts en cadeau. Ensuite, nous avons regroupé les produits par facture et client pour préparer les données à l'application de l'algorithme Apriori.

### Paramètres d'Exécution :

Les paramètres utilisés pour l'algorithme Apriori, tels que le seuil de support minimal (**min\_support\_threshold**) et le seuil de confiance minimale (**min\_confidence\_threshold**), ont été ajustés pour optimiser la recherche de règles d'association pertinentes.

### Comparaison entre FP-Growth et Apriori :

Afin de comparer l'efficacité des deux approches, nous avons également développé un code dédié à cette comparaison. Ce dernier a permis d'évaluer les performances en termes de temps d'exécution pour les deux algorithmes. Les résultats obtenus montrent que l'algorithme FP-Growth, et plus précisément son implémentation dans la bibliothèque mlxtend, est plus rapide que l'algorithme Apriori pour nos données.

### Conclusion sur la Comparaison :

Bien que les résultats obtenus par les deux algorithmes soient similaires, la différence de temps d'exécution en faveur de FP-Growth suggère que cette approche peut être plus efficace pour notre ensemble de données spécifique. Cette information est cruciale dans le choix de l'algorithme à déployer, en tenant compte des contraintes de temps et des ressources disponibles.