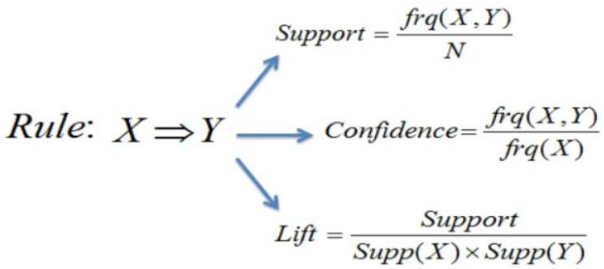
Algorithme d'Association – Analyse de Panier d'Achat

Définition:
L'algorithme d'association (comme Apriori) découvre les relations fréquentes entre articles dans un panier d'achat, permettant d'optimiser les stratégies marketing.

1. le principe de l'algorithme choisi:



Génération des itemsets fréquents :
L'algorithme parcourt les transactions pour identifier les combinaisons d'articles les plus fréquentes.

1. **Calcul des métriques** :
    - **Support :** Fréquence d'un groupe d'articles dans les transactions.
    - **Confiance :** Probabilité qu'un article soit acheté si un autre l'a été.
    - **Lift :** Dépendance entre articles (lift > 1 signifie une association positive).

Exemple :
Si "whole milk" et "tropical fruit" sont souvent achetés ensemble, la confiance et le lift seront élevés.

2. Dataset pour l'implémentation de cet algorithme

Un dataset couramment utilisé pour ce type d'analyse est le **"Groceries Dataset"**,Ce dataset contient des transactions clients anonymes d'un supermarché.

Exemple de dataset :

Member_number	Date	itemDescription
1808	21-07-2015	tropical fruit
2552	05-01-2015	whole milk
2300	19-09-2015	pip fruit

- **Les transactions** sont définies par les **Member_number** regroupant tous les articles achetés.
- Par exemple, si un client achète **"whole milk"**, **"butter"** et **"tropical fruit"**, l'algorithme peut découvrir des associations comme :
  - **"whole milk"** → **"tropical fruit"** avec une forte confiance.
  - **"butter"** → **"whole milk"** avec un lift positif, indiquant une relation fréquente entre ces produits.

3. Expliquer la nature du problème étudié

- **Type d'apprentissage** : **Apprentissage non supervisé**
  - L'objectif n'est pas de prédire une variable cible, mais d'identifier des relations entre les variables (articles).
- **Nature du problème** : Analyse d'association
  - Trouver des **règles d'association fréquentes** dans des données transactionnelles.

4. Implémentation de l'algorithme

a. Prétraitement des données :

- Convertir les données en un format binaire ou transactionnel.
  - Par exemple : chaque ligne représente une transaction, et chaque colonne un article. Si un article est présent dans une transaction, la valeur est `1`, sinon `0`.

b. Définir les variables :

- **Variables observées** : Les articles achetés (exemple : "Pain", "Lait").
- **Variables expliquées** : Les combinaisons fréquentes d'articles.

c. Partitionner le dataset :

- Le partitionnement (apprentissage/test) est optionnel ici, car l'objectif est d'explorer des associations dans l'ensemble de données global.

d. Phase de training :

La version d'Anaconda que j'utilise ne contient pas la bibliothèque que je souhaite utiliser a priori, donc je dois l'installer:

- Utiliser l'algorithme **Apriori** et **fpgrowth** pour extraire les itemsets fréquents.
- Définir un seuil minimal pour :
  - **Support** (exemple : 0.2 – l'itemset doit apparaître dans au moins 20 % des transactions).
  - **Confiance** (exemple : 0.7 – la règle doit avoir une probabilité d'au moins 70 %).

e. Phase de prédiction :

L'algorithme Apriori prédit implicitement l'achat d'articles supplémentaires en fonction des articles déjà présents dans une transaction.

5. Les métriques pour évaluer les performances :

Les métriques utilisées sont le Support, la Confiance et le Lift.

Justification des Métriques :

- Support : Fréquence d'apparition d'un groupe d'articles. Permet de filtrer les règles peu fréquentes.
- Confiance : Probabilité qu'un article soit acheté si un autre l'est. Évalue la fiabilité des règles.
- Lift : Quantifie la dépendance entre les articles. Un lift > 1 prouve une association significative.

6. Interprétation des résultats :

- Support moyen : 0.0115 (1.15% des transactions).
- Confiance moyenne : 0.7241 (72.41% de fiabilité).
- Lift moyen : 1.5804 (Augmentation de 58.04% de la probabilité d'achat).

7. Les paramètres de configuration :

- Algorithme FP-Growth
  - min_support=0.01 : seuil minimal pour identifier les itemsets fréquents.
  - use_colnames=True : utilise les noms des articles pour plus de lisibilité.
  - metric="lift" : évalue la dépendance entre articles.
  - min_threshold=1.5 : seuil minimum pour des règles pertinentes.
  - Filtrage des règles :
    - Confiance ≥ 0.7 et Lift > 1.
    - num_itemsets=2 : se concentre sur les associations de 2 articles.