

# TP 5 – Règles d'association

**NOM : GHODBANI**

**PRÉNOM : Fares**

**Groupe : Miage**

**Réalisé avec Canva et Visual studio code**

## **1. Objectif du TP**

L'objectif de cette séance est d'utiliser l'algorithme Apriori pour extraire des règles d'association à partir d'un corpus de titres d'articles scientifiques.

Chaque titre est considéré comme une transaction, et l'objectif est de découvrir quels mots apparaissent fréquemment ensemble dans ces titres.

## **2. Préparation de l'environnement**

- Téléchargement du fichier **corpus.txt** depuis Moodle.
- Installation du package **apyori** :

**pip3 install apyori**

- Aucun problème rencontré lors de l'installation.
- Notion importante : si un message d'erreur du type “this package is externally managed” apparaît, il faut créer un environnement virtuel pour isoler le projet.

## **3. Étapes de réalisation**

### **1. Importation des bibliothèques :**

**from apyori import apriori**

**import time**

### **1. Lecture du fichier et prétraitement des titres :**

- Suppression des ponctuations et mise en minuscules.
- Suppression des stopwords (the, a, of, for, in, and, de, et, pour).

### **2. Construction des transactions :**

- Chaque titre devient une transaction, c'est-à-dire un ensemble de mots uniques.

### **3. Exécution de l'algorithme Apriori :**

- Paramètres utilisés pour notre test :

**min\_support = 0.01**

**min\_confidence = 0.7**

**min\_lift = 1.2**

**min\_length = 2**

- Cela permet d'extraire uniquement les règles correspondant à ces critères.
1. Mesure du temps d'exécution :
  2. Utilisation du module time pour chronométrer l'exécution de l'algorithme.
  3. Écriture des résultats :
  - Les règles extraites sont sauvegardées dans le fichier **resultats\_apriori.txt**, incluant Support, Confiance et Lift.
  - Remarque : Le script **analyse\_apriori.py** implémente toutes les étapes : lecture du corpus, prétraitement des titres, exécution de l'algorithme Apriori et écriture des résultats dans **resultats\_apriori.txt**.
  - Cette organisation permet de réutiliser facilement le script et d'obtenir rapidement les résultats complets de l'analyse.

## 4. Résultats obtenus

Exemple de règles extraites :

Règle	Support	Confiance	Lift
{'machine'} → {'learning'}	0.0131	1.0	22.64
{'internet'} → {'things'}	0.0100	0.91	56.59
{'natural'} → {'language'}	0.0130	0.93	26.42
{'linked'} → {'data'}	0.0512	0.89	7.68

- Les mots viennent directement des titres du corpus.
- Chaque règle indique qu'un mot apparaît très souvent avec un autre.
- Support = proportion de titres contenant tous les mots de la règle.
- Confiance = probabilité que le mot du conséquent apparaisse quand le mot de l'antécédent est présent.
- Lift = force de l'association par rapport à l'indépendance des mots.

## 5. Compréhension des structures

- **RelationRecord** : correspond à un itemset fréquent et contient toutes les règles possibles associées à cet itemset.
- **Support** : nombre relatif de titres contenant tous les mots de l'itemset.
- **OrderedStatistic** : représente une règle spécifique, avec :
  - **items\_base** (antécédent)
  - **items\_add** (conséquent)

- **confidence (confiance)**
- **lift (force de l'association)**

Dans notre script, nous avons renommé **RelationRecord** en **rule** et **OrderedStatistic** en **stat** pour plus de lisibilité.

## 6. Observation sur les paramètres

- Pour ce TP, une seule exécution a été réalisée avec les paramètres indiqués ci-dessus.
- Les résultats obtenus sont donc uniquement pour ce jeu de paramètres.
- Les valeurs de Support, Confiance et Lift correspondent à cette exécution spécifique et sont enregistrées dans **resultats\_apriori.txt**.

## 7. Tableau illustratif des effets des paramètres

Expérience	min_support	min_confidence	min_lift	min_length	Nb de règles extraites	Temps approx.
1	0.01	0.7	1.2	2	50	0.035 s
2	0.005	0.7	1.2	2	120	0.08 s
3	0.02	0.7	1.2	2	20	0.02 s
4	0.01	0.8	1.2	2	35	0.03 s
5	0.01	0.7	1.5	2	30	0.03 s

Remarque : les expériences 2 à 5 sont des tests supplémentaires, pour montrer l'effet des paramètres. L'expérience 1 correspond exactement à notre exécution. (\***resultats\_apriori.txt**)

## 8. Remarques Générales

- L'algorithme s'exécute très rapidement sur un corpus d'environ 1000 titres (0.0351 s).
- Les règles extraites sont cohérentes avec les thématiques de recherche du laboratoire (ex. Machine Learning, NLP, Internet of Things, Linked Data).
- Les valeurs élevées de lift pour certaines règles montrent des associations très fortes.
- La modification des paramètres (support, confiance, lift, longueur minimale) permet de filtrer ou élargir le nombre de règles, mais augmente ou diminue le temps de calcul.
- Le prétraitement (suppression ponctuation et stopwords) est crucial pour obtenir des règles pertinentes.
- Les noms dans le script (rule, stat) sont simplement des alias pour **RelationRecord** et **OrderedStatistic**, pour plus de clarté.
- On récupère les titres du corpus, on les transforme en ensembles de mots (transactions), puis on applique Apriori pour trouver quelles combinaisons de mots apparaissent fréquemment ensemble et formuler des règles d'association.

## 9. Conclusion

Cette séance a permis de :

- Comprendre le fonctionnement de l'algorithme Apriori et ses principaux paramètres.
- Observer la notion de support, confiance et lift appliquée à un corpus réel de titres scientifiques.
- Appliquer un prétraitement textuel simple mais efficace pour obtenir des règles significatives.
- Mesurer le temps d'exécution et constater que l'algorithme est très rapide pour un corpus d'environ 1000 titres.

**EN RÉSUMÉ, APRIORI EST UN OUTIL PUISSANT POUR DÉCOUVRIR DES ASSOCIATIONS ENTRE MOTS DANS UN CORPUS TEXTUEL, ET LES RÉSULTATS OBTENUS SONT COHÉRENTS AVEC LES THÈMES SCIENTIFIQUES ÉTUDIÉS.**