# EDA - Rapport de projet

## Exercice de Fouille interactive de motifs avec préférences de l'utilisateur

---


École d'ingénieurs pour l'informatique et les techniques avancées

Specialization SCIA – Graphs

**Auteurs**

Aymeric Le Riboter
Abel Aubron
Erwann Lesech
Nathan Claude

Enseignants : Lamine Diop & Marc Plantevit

November 30, 2025

# 1 Introduction

Ce projet s'inscrit dans le cadre du module EDA – SCIA-G et vise à développer une solution complète de fouille interactive de motifs permettant à un utilisateur non expert d'extraire, filtrer, explorer et affinefpr des motifs dans des données transactionnelles. L'objectif est de combiner extraction classique de motifs fréquents, échantillonnage guidé, feedback utilisateur et une interface interactive accessible à distance. Notre implémentation repose sur une architecture moderne **FastAPI + Streamlit + Docker** telle que définie dans notre dépôt GitHub, ce qui garantit reproductibilité, modularité et facilité de déploiement. Dépôt GitHub du projet

# 2 Démarche suivie

## 2.1 Prétraitement et transformation des données

Dans un premier temps, nous avons permis à l'utilisateur d'importer plusieurs types de jeux de données : formats transactionnels, séquentiels, mais aussi des formats supplémentaires tels que les matrices item–transaction ou leurs versions transposées. Le type du dataset ainsi que le séparateur utilisé lors du chargement sont automatiquement détectés, avec la possibilité pour l'utilisateur de les modifier selon ses besoins (séparateurs pris en charge : virgule, point-virgule, tabulation, pipe ou espace).
Une fois le format identifié, les données sont normalisées et converties dans une représentation interne unifiée. Pour les transactions, les listes d'items sont transformées en matrice binaire indiquant la présence ou l'absence de chaque item, afin de préparer efficacement les calculs de support et l'extraction de motifs. Des exemples illustrant les différents formats acceptés et leur transformation sont fournis en annexes (voir Appendices A–B).

## 2.2 Extraction du pool de motifs

L'extraction initiale est réalisée dans le backend. Cette extraction est paramétrable par l'utilisateur via le frontend. Il a le choix entre différents algorithmes dépendamment du type de jeu de données en entrées:

- FP-Growth, TwoStep Sampling et GDPS si le jeu de données est composé de transactions
- PrefixSpan et TwoStep Sampling si le jeu de données est composé de séquences

Chacun de ces modèles nécessitent des paramètres de configuration spécifiques qui sont ajustable via l'interface utilisateur. Le résultat constitue un pool $P$ de plusieurs centaines à milliers de motifs, stockés sous forme de dataframe.

## 2.3 Stratégie de scoring et échantillonnage interactif

Une fonction de scoring composite a été mise en place pour classer les motifs. Elle repose sur :

$$Score(m) = \lambda \cdot Support(m) + \gamma \cdot Surprise(m) - \delta \cdot Redondance(m)$$

Cette fonction de scoring composite permet d'apporter plus de diversité dans lors de l'échantillonnage.

Les coefficients sont ajustables via l'interface Streamlit. Le nombre de motifs extraits par les algorithmes présenter précédemment pouvant être très élevé il est nécessaire de réaliser ensuite un échantillonnage avant d'afficher le résultats à l'utilisateur.

L'échantillonnage utilise un **importance sampling pondéré**, offrant :

- tirage avec ou sans remise,
- sélection d'un sous-ensemble de taille $k$,
- exploration priorisant les motifs les plus informatifs.

Une fois cet échantillonnage effectué ces résultats sont ensuites transmis au frontend pour affichage à l'utilisateur.

### 2.4 Boucle de feedback utilisateur

L'utilisateur peut **liker/disliker** un motif. Cela modifie dynamiquement son poids dans la distribution :

$$w'_m = \begin{cases} w_m + e^{-\alpha} & \text{si like} \\ w_m - e^{-\beta} & \text{si dislike} \end{cases}$$

Pour s'assurer que le feedback utilisateur ne déséquilibre pas trop le score les coefficients paramétrables sur l'intervalle [0,1] sont ensuite ramené sur l'intervalle [2,5] ainsi avec un paramètre alpha/beta tres faible alors le feedback ne modifiera le score que de l'ordre de 0.007 alors que lorsque qu'il sera très élevé cela se rapprochera de 0.14.

Cette ré-pondération permet une interaction en temps réel, adaptée à la logique de fouille exploratoire.

### 2.5 Interface interactive

L'interface Streamlit permet à l'utilisateur d'effectuer rapidement et facilement une extraction de motifs/patterns sur l'ensemble des formats de jeux de données acceptés. Il peut ensuite paramétrer son extraction tout d'abord en choisissant l'algorithme utilisé pour l'extraction mais aussi en ayant la possibilité de paramétrer cet algorithme. Pour permettre une fouille interactive l'utilisateur a aussi à sa disposition un système de feedback modélisé par un bouton like et un bouton dislike présent sous chaque motif résultant de l'extraction. L'utilisateur aura la possibilité de réaliser plusieurs fois l'extraction de sous ensembles à partir du même pool de motif. Cela permet par exemple de prendre en compte les feedbacks données sur les motifs extraits lors de la précédente extraction. Le mécanisme de feedback ne permet pas de liker ou disliker plusieurs fois le même motif lors d'une même extraction, en revanche si un motif réapparaît lors d'une extraction ultérieur l'utilisateur pourra à nouveau interagir avec celui-ci.

La latence est maintenue sous 2 secondes grâce à la communication optimisée avec le backend FastAPI, conteneurisé via Docker. Toutefois en cas de jeu de données très volumineux le temps de réponse peut augmenter, cela est dû au temps de traitement nécessaire à l'extraction des motifs.

## 3 Choix techniques et méthodologiques

### 3.1 Choix des algorithmes

- **FP-Growth** : efficace pour extraire de nombreux motifs dans des jeux de données transactionnels sans explosion combinatoire.
- **PrefixSpan** : particulièrement adapté aux données séquentielles, il évite la génération explicite de candidats et explore efficacement l'espace des motifs séquentiels via une croissance par préfixes projetés.

- **GDPS (Guided Discovery of Pattern Sampling)** : méthode d'échantillonnage guidé permettant de sélectionner des motifs représentatifs en intégrant directement les préférences de l'utilisateur ou des critères d'intérêt dans le processus de sampling.
- **TwoStep Pattern Sampling** : procédure en deux phases combinant sélection rapide d'un ensemble de motifs candidats puis raffinage ciblé, offrant un bon compromis entre diversité, qualité et temps de calcul.

### 3.2 Choix des paramètres

Les différents algorithmes retenus nécessitent d'être paramétrés pour donner des résultats concluant. Ainsi pour les algorithmes de fouilles exhaustives de motifs (PrefixSpan et FP-Growth) nous avons affecté la valeur minimale du support pour accepter un motif à 0.05 cela permet d'obtenir des motifs suffisamment représentatif des jeux de données, une valeur plus basse entraînerait à la fois une explosion du temps de calcul mais aussi un nombre de motifs générés beaucoup trop important.

Ensuite, pour les algorithmes d'échantillonnage en sortie il est possible de définir un nombre de motifs uniques que l'on souhaite extraire. Ces algorithmes essaieront d'atteindre ce nombre en revanche s'il est trop élevé il est possible que le pool extrait réellement comprenne moins de motifs uniques que demander. Pour GDPS on peut en plus de la taille du pool de motifs parametrer la taille des motifs extraits. De nouveau si la valeur des bornes minimales et maximales sont mal définis (valeur minimale trop élevé, intervalle de taille trop petit) alors il est possible que la taille du pool de motifs soit plus faible que celle demandé.

Pour finir, les valeurs par défaut du feedback positif ($\alpha$) et négatif ($\beta$) sont affectées à 0.03, une valeur faible qui permet que le feedback n'influence pas de facon trop importantes le score affecté au motif.

Ces valeurs ont été ajustées empiriquement pour obtenir un équilibre entre diversité et pertinence.

### 3.3 Métriques d'évaluation

Pour évaluer notre travail, nous avons utilisé cinq métriques principales :

1. **Diversité** : Calculée via la distance de Jaccard entre toutes les paires de motifs :

$$\text{Diversité} = 1 - \frac{|A \cap B|}{|A \cup B|}$$

   Cette métrique permet de vérifier si notre pipeline extrait des motifs suffisamment différents ou s'ils sont trop similaires.

2. **Couverture** : Déclinée en trois sous-métriques :
   - *Couverture des motifs* : proportion de motifs échantillonnés par rapport au pool complet.
   - *Couverture des items* : proportion d'items uniques du dataset présents dans les motifs échantillonnés.
   - *Couverture du support* : proportion du support total capturée par l'échantillon.
3. **Stabilité** : Pour mesurer la sensibilité à l'aléatoire, nous réalisons 10 échantillonnages avec des graines différentes, puis calculons la similarité de Jaccard moyenne entre les échantillons. Un score élevé indique une faible sensibilité à l'aléatoire.

4. **Taux d'acceptation** : Proportion de retours positifs (likes) par rapport au total des feedbacks utilisateurs :

$$\text{Taux d'acceptation} = \frac{\text{Nombre de likes}}{\text{Nombre total de feedbacks}}$$

5. **Temps de réponse** : Temps d'exécution moyen de l'échantillonnage sur 5 runs, afin de garantir une expérience interactive (objectif : $<$ 2–3 secondes).

Ces cinq métriques sont ensuite combinées dans un score global défini comme une moyenne pondérée :

$$\text{Score global} = 0.30 \times \text{Acceptation} + 0.25 \times \text{Diversité} + 0.25 \times \text{Couverture} + 0.20 \times \text{Stabilité}$$

# 4 Résultats

## 4.1 Analyse du pool de motifs

Le pool initial contient typiquement entre 100 et 4000 motifs selon les paramètres. On observe :

- une distribution très asymétrique des supports,
- quelques motifs très fréquents, mais peu informatifs,
- des motifs rares mais intéressants.

## 4.2 Résultats de l'échantillonnage

L'importance sampling sélectionne systématiquement :

- des motifs plus variés,
- des motifs ayant un meilleur ratio surprise/support,
- une réduction claire de la redondance structurelle.

Le feedback utilisateur améliore significativement la pertinence du top-$k$ après 2–3 itérations.

## 4.3 Performance et stabilité

- temps moyen d'extraction initiale : 1.2–1.6 s,
- échantillonnage : $<$ 0.4s,
- rafraîchissement de l'interface : $<$ 2s,
- variance faible entre seeds : stabilité satisfaisante.

# 5 Discussion critique

## 5.1 Limites

- FP-Growth et PrefixSpan peuvent devenir lourd sur des datasets massifs.
- Le feedback reste limité à like/dislike.
- Le scoring composite dépend d'hyperparamètres encore empiriques.

## 5.2 Pistes d'amélioration

- intégrer un moteur d'échantillonnage en sortie type *MCMC-pattern sampling*,
- améliorer les explications fournies à l'utilisateur (XAI),
- tester d'autres méthodes de scoring pour l'importance sampling,
- envisager d'autres méthodes d'échantillonage pondérés comme un MCMC léger

# 6 Conclusion

Ce projet démontre qu'il est possible de proposer une fouille de motifs interactive, rapide et accessible à un utilisateur non expert. L'approche combinant extraction exhaustive, scoring dynamique, échantillonnage pondéré et feedback utilisateur offre une exploration efficace d'un espace de motifs potentiellement très vaste. L'architecture modulaire FastAPI + Streamlit garantit une solution propre, extensible et prête à être utilisée en contexte réel.

# 7 Annexes

# A Formats de données acceptés

## A.1 Format transactionnel

```
transaction_id    items
1                 bread,milk,eggs
2                 bread,butter
...
```

## A.2 Format séquentiel

```
session   step   event
0         0      home
0         1      products
0         2      cart
...
```

## A.3 Format matriciel (item × transaction)

```
transaction_id   1       2       3       4       5
items            bread   bread   milk    bread   bread
                 milk             butter  milk
                 eggs             cheese
...
```

# B Représentation binaire après prétraitement

```
Transaction_ID   bread   milk   butter   cheese   eggs
T1               1       1      1        0        0
T2               1       0      1        0        0
T3               0       1      0        1        1
...
```