République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : ISIL

THEME

Découverte des Motifs Profitables dans les Base de Données Transactionnelles

Présenté par :

Makhloufi Sena

Chetouana Aya

Soutenu publiquement le : jj/mm/aaaa

Devant le jury composé de :

Président : Dr. Zouache Djaafer

Examinateur : Dr. Saifi Lynda

Encadreur : Dr : Nouioua Mourad

2022/2023

Dédicace

Aya :

Je tien à dédier ce mémoire avec grand plaisir :

Aux être les plus chers de ma vie , ma mère et mon père . je tien à les remercier pour leur patience infinie, leur amour inconditionnel, leur soutien constant et leur encouragements j'usqu'à la fin de mes études. Ils ont été le pilier solide et le soutien essentiel dans mon parcour éducatif , et je leur suis profondément reconnaissante pour tout .

À mes chers frères (Chamseddine , Takieddine , Youcef bahaeddine) , ainsi qu'à ma famille , mes amis proches et ma binome.

Sans oublier mon encadreur (Nouioua Mourad) , je vous suis très reconnaissante pour votre aide précieuse, votre soutien inestimable,

ainsi que vos conseils et orientations qui ont été d'une grande valeur dans mon développement.

Sena :

Je dédie ce mémoire :

À mes chers parents ma mère et mon pére pour leur patience, leur amour , leur soutien et leur encouragement .

À la famille makhloufi .

À mes chères frères Oussama , Aziz et Aymen .

À ma binome Aya Chetouana .

À mes milleures amies Chaima , Fariel et Célia.

À mes amis et mes camarades .

Sans oublier mon encadreur Mr.Nouioua Mourad.

Remerciement

Tout d'abord, nous remercions Dieu de nous avoir donné le pouvoir d'apprendre et de faire ce travail.

Nous remercions notre encadreur. Dr. Nouioua Mourad pour son aide, sa disponibilité et les précieux conseils qu'il nous a donnés.

Nos remerciements aux membres de jury pour avoir accepté l'évaluation de notre travail.

A cette occasion, nous voudrions remercier tous les professeurs qui ont contribué à notre formation et à la qualité de l'éducation qu'ils nous ont donnée.

Nous n'oublions pas nos parents pour leur soutien et leur patience, nos familles et amis qui nous ont soutenus et

Encouragés, ainsi que toutes les personnes qui nous ont aidés de près ou de loin.

Merci à tous.

Résumé

Notre recherche parle d’un thème de découverte de motifs rentables dans les bases de données transactionnelles. Il s'agit de trouver des motifs à haute utilité qui ont une forte corrélation avec une variable cible, telle que le profit ou la satisfaction du client.

L'objectif est d'aider les entreprises à prendre des décisions éclairées en identifiant les modèles de comportement des clients qui ont un impact positif sur leur rentabilité. Le mémoire examine différentes méthodes de découverte de motifs rentables, telles que l'analyse de corrélation, l'analyse de séquence en utilisant deux algorithmes différentes, et propose des approches pour améliorer la précision et l'efficacité de ces méthodes.

Abstract

Our research is about the discovery of profitable pattern discovery in transactional databases. It's about finding high-utility patterns that have a strong correlation with a target variable, such as profit or customer satisfaction.

The goal is to help businesses make informed decisions by identifying customer behavior patterns that positively impact their bottom line. The thesis examines different cost-effective pattern discovery methods, such as correlation analysis, sequence analysis using two different algorithms, and proposes approaches to improve the accuracy and efficiency of these methods.

ملخص

يتناول بحثنا موضوع اكتشاف الأنماط المربحة في قواعد البيانات التجارية، حيث يهدف البحث إلى العثور على الأنماط الشائعة التي ترتبط بشكل قوي بمتغير هدف مثل الربح أو رضا العملاء. يهدف البحث الى إلى مساعدة الشركات في اتخاذ قرارات مدروسة من خلال تحديد أنماط سلوك العملاء التي تؤثر إيجابيًا على ربحهم. يتضمن البحث دراسة طرق مختلفة لاكتشاف الأنماط الرابحة، مثل تحليل الترابط وتحليل التسلسل باستخدام خوارزميتين مختلفتين، كما يقترح البحث نهجًا لتحسين دقة وكفاءة هذه الطرق.

Table des Matières

Liste des Abréviations

CHUIM ou Correlated HUIM: Correlated High Utility Itemsets Mining

ECD : Extraction de Connaissances à partir de Données/ Knowledge Extraction

EUCS : Estimated Utility Co-Occurrence Structure

FCHM: Fast Correlated High-Utility Itemset Mining

FHM: Faster High-Utility Itemset Mining Algorithm

FIM: Frequent Itemset Mining

HUIM: High-Utility Itemset Mining

HUI-Miner: High Utility Itemset-Miner.

HUP-Growth: High Utility Pattern Growth Algorithm

HUP-Miner: High-Utility Pattern-miner.

IDE: Integrated Development Environment

IHUP: Incremental High-Utility Pattern.

KDD: Knowledge Discovery in Databases

SPMF: Sequential Pattern Mining Framework

TWU: Transaction Weighted Utilization

UP-Growth: Utility Pattern Growth Algorithm

ULB-Miner: Efficient high utility itemset mining using buffered utility-lists algorithm

Liste des Figures

Liste des Tableaux

Chapitre 1: Introduction Générale

1.1 Contexte Général

La fouille de données est une discipline qui consiste à extraire des connaissances utiles à partir des grandes quantités de données. Les techniques de fouille de données peuvent être généralement décrites comme prédictives ou descriptives [1]. Les premiers sont utilisés pour effectuer des prédictions, tandis que les seconds peuvent résumer ou révéler des informations intéressantes à partir des données pour aider les utilisateurs à comprendre les données. L'un des principaux types de fouille de donnée descriptive est fouille de motifs, en anglais *Pattern Mining*, qui vise à révéler des modèles intéressants, utiles ou inattendus dans les bases de données.

La fouille des itemsets fréquents, en anglais FIM (*Frequent Itemset Mining*), est l'un des problèmes classiques du *pattern mining* où l'objectif principal est de découvrir l'ensemble d'items, c'est-à-dire les itemsets, qui sont fréquemment achetés ensemble par les clients dans une base de données de transactions. Les itemsets fréquents sont des ensembles d'articles qui apparaissent souvent ensemble dans les transactions. La recherche d'itemsets fréquents est importante car elle peut aider à identifier des modèles dans les données qui peuvent être utilisés pour prendre des décisions éclairées.

1.2 Problématique, Motivation et Contributions

1.2.1 Problématique

Malgré plusieurs problèmes du monde réel peuvent être formulées comme un problème de FIM spécialement les problèmes issus de l'analyse du panier de consommation (*Market basket analysis)*, FIM a certaines limites et elle ne peut pas couvrir tous les aspects de certains problèmes dans la pratique [9].

Plus précisément, deux limitations importantes ont été identifiées :

1. FIM ne prend pas en considération la quantité achetés par les clients. Par exemple, l'achat d'une, deux ou 10 bouteilles est interprété de la même façon par FIM.
2. Dans FIM, tous les items (produits) sont considérés comme ayant la même importance. Par exemple, le profit de vente d'un jeu électronique est le même que le profit de vente d'un stylo.

Pour surmonter ces limitations, une extension plus difficile de FIM appelée High Utility Itemset Mining (HUIM).

1.2.2 Motivation

L'extraction d'itemsets à haute utilité (HUIM) a été développée pour trouver l'ensemble d'itemsets qui peuvent générer un profit élevé dans une base de données quantitative. Contrairement à FIM, chaque item a une quantité d'achat (utilité interne) dans HUIM. De plus, nous devons également définir la table d'utilité qui contient le profit (utilité externe) de chaque item de la base de données. L'objectif principal de HUIM est de trouver les itemsets qui ont des utilités non inférieures au seuil prédéfini (*minutil*).

En comparaison avec FIM, HUIM est plus intéressant car il prend en considération la quantité et l'utilité des éléments dans la base de données qui peuvent refléter de nombreux scénarios réels.

L'utilité peut être utilisée pour modéliser des critères intéressants dans divers problèmes. Par exemple, pour étudier les habitudes d'achat dans une base de données transactionnelle, l'utilité d'un modèle (un ensemble d'éléments) peut être mesurée en termes de profit qu'il rapporte, tandis que pour analyser les données de flux de clics, l'utilité peut représenter le temps passé sur les pages Web, etc.

Vue des avantages de HUIM par rapport à FIM, récemment, l'exploration de modèles à haute utilité est devenue une tâche clé d'exploration de modèles et plusieurs algorithmes ont été proposés pour faire face à cette tâche. Plusieurs problèmes de monde réel ont été modélisés comme des problèmes de HUIM. De plus, le tache de HUIM a été entendue vers d'autre tâches plus intéressantes tels que le fouille des motifs à haute utilité corrélée, la fouille des motifs séquentiels à haute utilité, etc.

L’avancement rapide de ce domaine de recherche, fouille de motif à haute utilité, nous a motivé à essayer ce type d’algorithmes en appliquant ces algorithmes sur des base de données du monde réels.

1.2.3 Contribution

Dans ce mémoire, nous nous intéressons à appliques les algorithmes des fouille des motifs avec utilité dans deux bases de données transactionnelles du monde réel.

La première base de données contient les transactions faites par des clients d’une boutique de cosmétique. La boutique contient 16303 produits et les transactions sont enregistrées pendant 3 ans. La deuxième base contient les achats faites dans une boutique en ligne pendant 9 mois. Le nombre de produits vendus par cette boutique est 3468.

Nous nous intéressons à découvrir deux types de motifs : Les motifs à haute utilité ainsi que les motifs à haute utilité avec corrélation entre les itemsets. Pour atteindre le premier objectif nous appliquons l’algorithme FHM (*Faster High Utility Itemset Mining)* alors que l’algorithme FCHM (*Fast Correlated High Utility Itemset Mining)* est adopté pour obtenir le deuxième type de motifs.

Nous allons comparer les performances des deux algorithmes et nous allons essayer d’extraire des motifs intéressants à partir des résultats obtenus.

1.2.4 Organisation du Mémoire

Le premier chapitre de ce mémoire offre une introduction générale sur la fouille de données. Nous commençons par introduire le contexte général de notre travail, nous présentons ensuite la problématique et notre motivation.

Le deuxième chapitre de ce mémoire est consacré pour présenter le domaine de la fouille de données. Nous commençons par donner sa définition, sa position par rapport au processus d’extraction des connaissances (KDD). Ensuite, les différentes taches de fouille de données sont présentées. Nous citons aussi les défis et les objectifs de la fouille de données. Nous finissons par la présentation de quelque domaine d’applications de la fouille de données.

Dans le troisième chapitre, nous nous intéressons à présenter en détail les taches de la fouille des itemsets à haute utilité (HUIM) ainsi que la fouille des motifs à haute utilités corrélés (Correlated HUIM ou CHUIM). Nous présentons les concepts de base de ces techniques, les algorithmes FHM et FCHM qui sont utilisés pour effectuer ces taches ainsi que les stratégies d'élagage adoptées par ces algorithmes.

Le quatrième chapitre est consacré à l'application des algorithmes FHM et FCHM sur deux bases de données transactionnelles, la première est d'une boutique de cosmétique et l'autre est une base d'une boutique en ligne. Nous présentons les résultats obtenus selon plusieurs critéres d'évaluation et nous donnons une interprétation des motifs obtenus par les deux algorithmes.

Enfin, nous conclurons ce mémoire par une conclusion générale ou nous donnons un résume sur notre travail et nous présentons aussi les perspectives futures de cette étude.

Chapitre 2: Fouille de Données

2.1 Introduction

Le *data-mining* ou « *fouille de données* » regroupe un ensemble des techniques permettant l'extraction, à partir d'un important volume de données brutes, des connaissances originales auparavant inconnues [1].

Dans ce premier chapitre nous traitons le sujet de « *La fouille de données* ». Nous commençons par la présentation du processus *KDD* et *data-mining* aussi avec ses différentes tâches et ses techniques. En fait, les techniques de *data-mining* peuvent être décomposé de deux catégories : Les techniques prédictives et les techniques descriptives. Parmi les techniques descriptives, on trouve *la fouille de motifs* (*Pattern Mining)*. Ensuite, on cite les défis, les objectifs de data-mining. Nous concluons notre chapitre par la présentation des objectifs de fouille de donnée.

2.2 Processus d'Extraction de Connaissances

2.2.1 Définition

Les termes *Data-Mining* et KDD *(Knowledge Discovery in Databses)* sont habituellement interchangés : *Knowledge Discovery in Databses* est le processus de trouver information et/ou partons utiles à partir de données. Data Mining est l'utilisation des algorithmes pour extraire information et/ou partons comme partie du processus KDD. En d'autres termes, data-mining est une partie du processus KDD et data-mining sont le cœur du processus d'extraction de connaissances [2].

2.2.2 Autre définition

ECD (Extraction de Connaissances à partir de Données/Knowledge extraction) : est l'ensemble du processus de découverte et d'interprétation de régularités dans des données [3].

2.3 Les Etapes de Processus KDD

Les étapes de processus KDD est composé essentiellement de 8 étapes qui sont illustrées dans la Figure 2.1. Les étapes du processus KDD sont les suivantes :

1. Fixation des objectifs selon la compréhension du domaine.
2. Création des données concernées par la découverte.
3. Prétraitement et nettoyage des données.
4. Transformation des données.
5. Choix de la tache appropriée de Data-Mining.
6. Choix de l'algorithme de Data-Mining.
7. Mise en œuvre de l'algorithme.
8. Evaluation et interprétation [4].

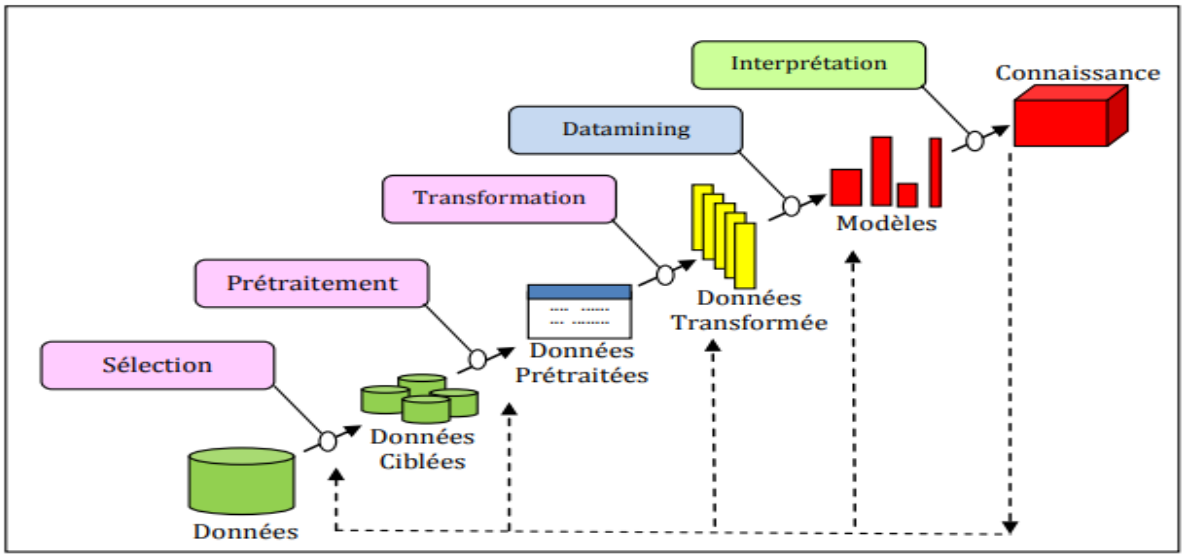

Figure 2.1: Le processus d'extraction de connaissances.

2.4 La Fouille de Données

2.4.1 Définition

Le terme data-mining signifie littérairement **forage de données.** Data-mining est un processus non élémentaire de mises à jour de relations, corrélations, dépendances, associations, modèles,

structures, tendances, classes, facteurs, obtenus en navigant à travers de grands ensembles de données, généralement consignées dans des bases de données (relationnelles ou pas), navigation réalisée au moyen de méthodes mathématiques, statistiques ou algorithmiques [3].

Dans nos jours, les techniques de data-mining démontrent leurs efficacités. La preuve est la large gamme d'utilisation de ces techniques dans plusieurs domaines tel que la santé, l'éducation et le marketing…etc.

2.5 Les Taches de Fouille de Données (Data-Mining)

Beaucoup de problèmes du monde réel tel que les problèmes intellectuels, économiques ou même commerciaux peuvent être exprimés en termes des taches suivantes : Classification, règles d'association, estimation et description.

2.5.1 La Classification

La classification est la tâche la plus commune de data-mining et qui semble être une obligation humaine. La classification se base sur des classes prédéfinies, c'est-à-dire des classes connues ou ont été construites déjà [5].

Dans la classification, la population est sous-devisée, en effectuant chaque élément ou enregistrement à une classe prédéfinie sur la base d'un modèle obtenue par apprentissage avec des exemples pré-classé.

2.5.2 Règle d'Association

Une règle d'association est un énoncé du type Si A alors B. Par exemple, « 60% des clients qui achètent du lait achètent aussi du pain ». Le domaine d'application classique des règles d'association concerne l'analyse du « panier de la ménagère » par les grandes surfaces de distribution [4].

Ces règles leur permettent de réorganiser la disposition de leurs produits dans les rayons, et aussi d'offrir des promotions en fonction des habitudes d'achats découvertes, etc. Pour extraire les règles d'association, une étape préliminaire est d'abord la découverte des patterns fréquents.

2.5.3 Description

La méthode de description en data-mining est une approche analytique utilisée pour extraire des informations significatives à partir de grands ensembles de données ou à partir des bases de données complexes. Elle vise à découvrir des modèles et des structures cachés dans les données, ce qui permet de générer des descriptions compréhensibles et utiles des caractéristiques des données.

2.5.4 L'Estimation

La tâche d'estimation est souvent utilisée pour effectuer la classification, et pour obtenir la valeur d'une telle variable comme l'état du compte d'une carte de crédit, etc.

2.6 Les Techniques de Fouille de Données

Les principaux algorithmes de la fouille de données sont classés en deux ensembles de techniques :

2.6.1 Les Techniques Descriptives

Ces techniques permettant de décrire, classifier, résumer des données et mettre en évidence les informations présentes qui sont cachées dans le volume de données :

1. **La visualisation des données :** Ces technique descriptives permettent de représenter graphiquement les données de manière compréhensible et informative, facilitant ainsi la détection de schémas, de tendances ou d'anomalies.
2. **Le clustering :** Cette technique regroupe les données similaires en fonction de leurs similarités ou de leurs distances. Elle permet d'identifier des groupes ou des segments homogènes au sein des données, ce qui facilite la compréhension des relations entre les instances de ces données.
3. **Les règles d'association :** les règles d'associations consistent à découvrir des relations intéressantes et fréquentes entre les éléments d'un ensemble de données. Elles permettent d'identifier des combinaisons d'items qui se produisent souvent ensemble.

4. **Pattern Mining :** le *pattern mining* vise à révéler des modèles intéressants, utiles ou inattendus dans les bases de données. De nombreux algorithmes d'exploration de motifs ont été conçus pour trouver divers types de motifs tels que les motifs fréquents, les règles d'association et les motifs séquentiels fréquents et les motifs profitables.

2.6.2 Pattern Mining (La Fouille de Motifs ou Recherche des Patterns)

Pattern mining est un sous-domaine clé de data-mining, son but et de développer des algorithmes pour découvrir des modèles intéressants dans les bases de données. Les modèles découverts peuvent être utilisés pour aider à comprendre les données et également pour effectuer d'autres tâches comme la classification et la prédiction.

La recherche des patterns peut être utilisée pour des fins commerciales pour analyser le panier de la ménagère puis développer des stratégies marketing sur les achats alimentaires des consommateurs. A partir de l'analyse de nombreuses transactions.

2.6.3 Les Techniques Prédictives

Ces techniques permettant d'extrapoler et de prédire de nouvelles informations et connaissances à partir des données [6]. Il existe plusieurs techniques prédictives tel que :

1. **La Classification** : L'objectif de la classification supervisée est principalement de définir des règles permettant de classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets [7].
2. **La Régression :** le but de la régression est d'estimer une valeur (numérique) de sortie à partir des valeurs d'un ensemble de caractéristiques en entrée. Par exemple, estimer le prix d'une maison en se basant sur sa surface, nombre des étages, son emplacement, etc [8].

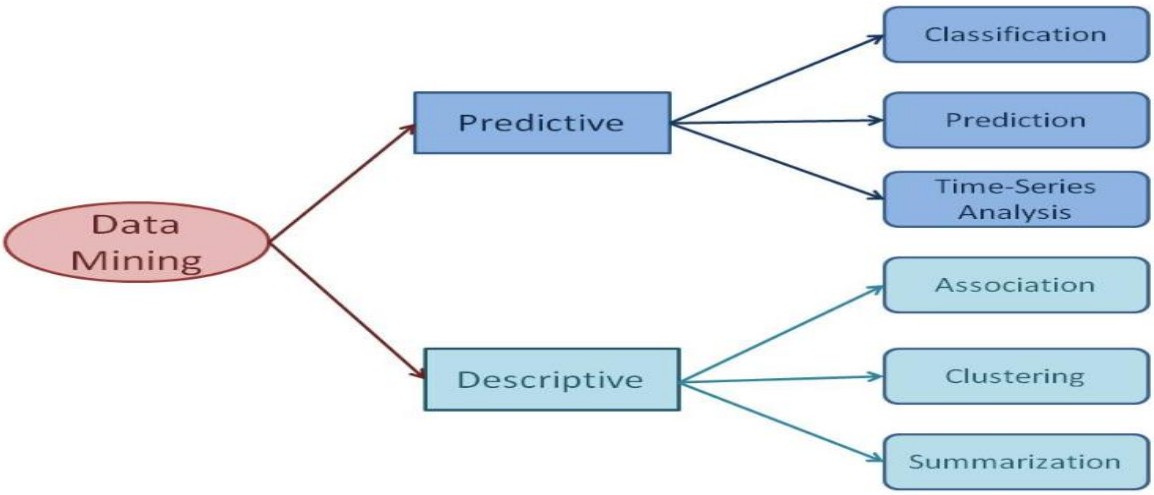

Figure 2.2: Les techniques de Data-Mining [9].

Dans la réalité, il n'y a pas de meilleures méthodes de fouilles. Il faudra faire des compromis selon les besoins dégagés et les caractéristiques connues des outils. Donc le choix de la méthode approprié se fait en fonction de plusieurs facteurs tel que :

- La nature et la disponibilité des données.
- Les connaissances disponibles.
- L'environnement de l'entreprise.
- Les différentes tâches.
- La finalité du modèle construit.
- La complexité de la construction du modèle.
- La complexité de l'utilisation du modèle et sa performance.

2.7 Les Défis de Fouille de Données

La mise en œuvre de data-mining rencontre les difficultés suivantes :

- **L'hétérogénéité des données** : Cette réalité implique une phase très chronophage de préparation. En fait, il a été démontrer que le temps de préparation influe négativement sur le temps global du projet.
- **Le volume des données :** Le volume de données est parfois difficile à traiter.
- **Evaluation des résultats :** Lors de la définition du problème, un objectif principal est de déterminer s'il y'a un aspect important du problème à résoudre qui n'a pas été suffisamment considéré. À la fin de cette phase, une décision est prise par l'utilisateur en fonction des résultats fournis par les outils de data-mining.

2.8 Les Objectifs de Fouille de Données

Les techniques de data-mining généralement servent à :

- Comprendre le processus de la fouille de données.
- Extraction de l'information pour décider.
- Connaitre les limites légales de l'utilisation des données.
- Découvrir un des outils informatiques majeurs d'aide à la décision.
- Avoir un aperçu du data-mining dans le domaine du marketing.

2.9 Les Domaines d'Application

Il y'a plusieurs secteurs utilisant la fouille de données. Parmi ces secteurs, on cite :

2.9.1 Secteur Bancaire

Les techniques de data-mining sont utilisées dans le secteur bancaire pour réduire le risque de prêts bancaires ; la création de modèles à partir des caractéristiques des clients permet de discriminer les clients à risque élevé [10].

2.9.2 Secteur d’Education

Data-mining est employé dans les établissements scolaires pour améliorer la qualité d’enseignement. Par exemple, répartir les élèves ayant une grande capacité d’assimilation dans la même classe[10].

2.9.3 Secteur Médical

Plusieurs types d’utilisation peuvent être faites dans le secteur médical :

- Recherche scientifique.
- Découverte des maladies d’après les symptômes du patient.
- Mettre en évidence des facteurs de risque ou de rémission de certaines maladies.
- Choix du médicament le plus approprié pour guérir une maladie donnée, pour un individu donné, etc.

2.10 Conclusion

La fouille de données est une technique d'analyse de données qui permet de découvrir des modèles, des relations et des tendances cachées dans de grands ensembles de données.

La fouille de données est utilisée dans de nombreux domaines, tels que le marketing, la finance, la santé, la sécurité, l'analyse de réseaux sociaux, etc. Elle permet de prendre des décisions éclairées en identifiant des modèles et des tendances qui seraient difficiles à détecter autrement.

Chapitre 3: Les Algorithmes de HUIM

3.1 Introduction

La recherche des algorithmes d'extraction d'itemsets a commencé dans les années 1990, avec des algorithmes pour découvrir des modèles fréquents (*frequent pattern*s) dans les bases de données pour le but de faciliter la compréhension des humains et soutenir la prise de décision dans de nombreux domaines surtout le marketing.

Dans ce chapitre, on va parler des principaux problèmes d'extraction des itemsets à haute utilité (HUIM), et de deux types d'algorithmes pour résoudre ces problèmes.

3.2 Fouille des Itemsets Fréquents *(Frequent Itemset Mining)*

Les techniques d'extraction des itemsets fréquents et les règles d'association sont parmi les outils les plus répandus en data-mining. Elle permet la découverte de règles intelligibles et exploitables dans un ensemble de données volumineux, règles exprimant des associations entre items (attributs) dans une base de données transactionnelle ou relationnelle. C'est un sujet de recherche attractif et très actif vu ses larges champs d'applications à divers domaines tels que : Le marketing, aide au diagnostic médical, télécommunication, analyse de données spatiales, etc [11].

Avant de définir d'un façon exacte le problème de FIM, on présente les concepts de base FIM.

3.2.1 Concepts Préliminaires

- **Définition 1 (Base de données transactionnelle).** Une base de données transactionnelle est un ensemble de transactions représenté par $D = \{T1, T2, ..., Tn\}$, tel que n est le nombre de transactions de la base. Chaque transaction a un identifiant et elle contient un ensemble d'items [11].
- **Exemple.** Le Tableau 3-1 représente un exemple d'une base de données transactionnelle *D.* Cette base contient 5 transactions. La première transaction par exemple est composée de 5 items *a, b, c, d et e.*

Tableau 3-1. Base de données transactionnelle

TID	***Transaction***
T_0	***a, b, c, d, e***
T_1	***b, c, d, e***
T_2	***a, c, d***
T_3	***a, c, e***
T_4	***b, c, e***

- **Définition 2 (Support d'un itemset).** Le support d'un itemset *X* dans une base *D*, noté *Supp(X)*, est la proportion de transactions de *D* contenant *X* : $Supp(X)=\frac{Freq(X)}{|D|}$ ou *Freq(X)* est le nombre des transactions dans D qui contiennent l'itemset *X* [11].
- **Exemple.** Le support de *{ad}* dans la base *D* est $Supp(ad)=\frac{Freq(ad)}{|D|}=\frac{3}{5}$.

3.2.2 Problème de FIM (*Frequent Itemset Mining)*

FIM consiste à rechercher toutes les combinaisons d'articles (itemsets) qui apparaissent fréquemment dans une base de données transactionnelle. Un ensemble d'articles est considéré comme fréquent s'il dépasse un seuil de fréquence prédéfini, appelé *minSupp*, généralement déterminé par l'utilisateur. Formellement, on appelle *X* un *itemset fréquent* si $Supp(X) \geq minSupp$ *[11]*.

3.2.3 Les Limitations de FIM

Un problème majeur de FIM est qu'il considère que tous les éléments (items) ayant la même importance (utilité), peu importe si l'item a une forte utilité ou non [11]. Un autre problème de FIM est que les quantités d'éléments ne sont pas prises en compte. Par exemple, si un client achète cinq pains, ou dix pains, il est considéré comme identique. Alors, l'exploration de motifs fréquents (*frequent pattern mining*) peut trouver beaucoup de motifs fréquents qui ne sont pas intéressants [11].

Donnant un exemple réel, l'itemset {pain, lait} est un itemset fréquent (ce itemset est largement acheté par les clients) mais il n'est pas intéressant pour un commerçant puisqu'il ne génère pas

beaucoup de profit (utilité). Les algorithmes d'exploration de modèles fréquents peuvent manquer les modèles rares qui génèrent un profit élevé [11].

3.3 Fouille de Motifs à Haute Utilité (High Utility Itemset Mining)

Contrairement au problème de FIM, HUIM est la tâche d'extraction d'itemsets à haute utilité dans une base de données transactionnelle quantitative. Pour pouvoir définir le problème de HUIM, on a besoin d'abord de définir les concepts de base liée à ce problème.

3.3.1 Concepts Préliminaires

- **Définition 4 (Base de données transactionnelle quantitative).** Une base de données transactionnelle quantitative contient des informations supplémentaires tel que les quantités d'items dans les transactions (utilité interne), et des pondérations indiquant l'importance relative de chaque item à l'utilisateur (utilité externe) [12].
- **Exemple.** Les Tableaux 3-2 et 3-3 donnent un exemple d'une base de données transactionnelle avec les quantités externes des items [12]. La transaction T_3 par exemple indique qu'un client a acheté 2 unités de l'item a, 2 unités de l'item c et une seule unité de l'item e.

Tableau 3-2. Base de données transactionnelle quantitative [12]

TID	***Transaction***
T_0	(a, 1), (b, 5), (c, 1), (d, 3), (e, 1)
T_1	(b, 4), (c, 3), (d, 3), (e, 1)
T_2	(a, 1), (c, 1), (d, 1)
T_3	(a, 2), (c, 6), (e, 2)
T_4	(b, 2), (c, 2), (e, 1)

Tableau 3-3. Utilités des items

Item	**Utilité externe**
a	5
b	2
c	1
d	2
e	3

- **Définition 5 (Utilité d'un item *i* dans une transaction T_c)**. L'utilité d'un itemset *i* dans une transaction T_c est notée comme $u(i, T_c)$ et défini comme $p(i) \times q(i, Tc)$. Où $p(i)$ est l'utilité externe de l'item *i* et $q(i,T_c)$ est la quantité de *i* dans la transaction T_c [12].
- **Exemple.** L'utilité de l'item *a* dans la transaction T_3 est : $u(a, T_3)=5 \times 2 = 10$.
- **Définition 6 (Utilité d'un itemset X dans une transaction T_c)** . L'utilité d'un itemset *X* dans une transaction T_c est notée $u(X, T_c)$ et définie comme $u(X, T_c) = \sum_{i \in X} u(i, T_c)$ si $X \subseteq T_c$.
- **Exemple.** L'utilité de l'itemset *{ac}* dans la transaction T_3 est : $u(a, T_3) + u(c, T_3) = 5 \times 2 + 1 \times 6 = 16$.
- **Définition 7 (Utilité d'un itemset X dans une base *D*).** L'utilité d'un itemset *X* dans une base de données *D* est noté $u(X)$ et défini comme $u(X) = \sum_{Tc \in g(X)} u(X, T_c)$, où $g(X)$ est l'ensemble des transactions contenant *X*. Il représente le profit généré par la vente de l'itemset *X* dans la base de données transactionnelle quantitative *D*.
- **Exemple.** L'utilité de l'itemset *{ac}* dans la base de données *D* est : $u(\{ac\}) = u(a)+u(c) = u(a, T_0) + u(a, T_2) + u(a, T_3) + u(c, T_0) + u(c, T_2) + u(c, T_3) = 5 + 5 + 10 + 1 + 1 + 6 = 28$.
- **Definition 8 (Itemset à haute utilité ou High utility itemset).** *X* est un itemset à haute utilité si son utilité $u(X)$ n'est pas inférieur à un seuil d'utilité *minutil* prédéfini par l'utilisateur ; (c'est-à-dire $u(X) \geq minutil$). Sinon, *X* est un itemset de faible utilité ou *low-utility itemset*. [11]

3.3.2 Problème de HUIM (High Utility Itemset Mining)

Le problème de fouille des itemsets à haute utilité consiste à découvrir tous les itemset à haute utilité prenant en considération un seuil *minutil* défini par l'utilisateur. Des études ont montré que l'utilité d'un itemset s'exprime en pourcentage de l'utilité total dans la base de données que ce qu'on appelle « Utilité absolue » [12].

HUIM à de nombreuses applications, en prend un exemple vivant comme « l'application de l'analyse du panier de consommateur ». Alors le problème ici est de trouver tous les itemsets qui

ont généré un profit supérieur ou égal à *minutil*. Il existe plusieurs algorithmes qui ont été proposés pour découvrir des itemsets à haute utilités tels que : *HUI-Miner, FHM* et *Two-phase*.

3.3.3 Difficulté de HUIM Algorithmes

Pour une base de données quantitative et un seuil d'utility minimum donnés spécifié par l'utilisateur, le problème de HUIM a toujours une seule solution. Il s'agit d'énumérer tous les modèles qui avoir une utilité supérieure ou égale au seuil d'utilité minimum. Le problème de l'extraction d'itemset à haute utilité est difficile pour deux raisons principales [12]:

1. La première raison est que le nombre d'itemsets à considérer peut-être très important pour trouver ceux qui ont une grande utilité. Généralement, si une base de données contient m éléments distincts, il y a 2^m - 1 itemsets possibles qui peuvent être générés. Une approche naïve pour résoudre le problème HUIM est de compter les utilités de tous les itemsets possibles en parcourant la base de données, pour ensuite conserver les itemsets à haute utilité. Bien que cette approche produise le résultat correct, elle est inefficace. La raison en est que le nombre d'itemsets possibles peut être très important. Par exemple, si un magasin de détail a *10 000* articles *(m = 10000)*, n'a besoin de calculer les utilités de $(2^{10000} -1)$ itemset, ce qui est non praticable.
2. Les itemsets à haute utilité sont souvent dispersés dans l'espace de recherche. Ainsi, de nombreux itemsets doit être pris en compte par un algorithme avant de pouvoir trouver les itemsets à haute utilité réels. Cette difficulté est présente paracerque contrairement au problème de FIM ou le support d'itemset est toujours supérieur ou égal au support de l'un de ses sur-ensembles (super-sets), dans le cas de HUIM, l'utilité d'un itemset peut être supérieure, inférieure ou égale à l'utilité de n'importe lequel de ses sur-ensembles/sous-ensembles. En d'autres termes, l'utilité a une propriété est qu'elle n'est ni monotone ni anti-monotone.

- **Propriété 1 (La mesure d'utilité n'est ni monotone ni anti-monotone).** Etant donné deux itemsets *X* et *Y* tels que $X \subset Y$, la relation entre les utilités de *X* et *Y* sont soit : $u(X) < u(Y)$, $u(X) > u(Y)$ ou $u(X) = u(Y)$.

3.4 Algorithmes Proposés pour HUIM

Plusieurs algorithmes ont été proposés pour extraire les itemsets à haute utilité, nous mentionnons : UMining, IHUP, UP-Growth, HUP-Growth, MU Growth, HUI-Miner, FHM, ULB-Miner, HUI-Miner [12]. Ces algorithmes ont la même entrée (base de données transactionnelle quantitative plus la valeur de minimum utilité *minutil*) et la même sortie (l'ensemble des itemsets à haute utilité).

Tous ces algorithmes représentent l'espace de recherche de HUIM sous forme d'un graphe et ils utilisent soit une stratégie de profondeur (DFS) ou une stratégie de largeur (BFS) pour explorer l'espace de recherche. L'exploration d'espace de recherche n'est pas faite d'une façon exhaustive mais en utilisant des stratégies d'élagage *(pruning strategies)*. Les stratégies d'élagages permettent aux algorithmes d'explorer intelligemment l'espace de recherche par l'élimination autant que possible des itemset qui sont certainement à faible utilité avec leur sur-ensembles. Les premiers algorithmes ont été conçue pour réaliser deux phases pour découvrir l'ensemble de tous itemsets à haute utilité ans la base. Ensuite, d'autres algorithmes plus performants ont été développer qui sont basés en une seule phase.

Tous les points essentiels pour la conception d'un algorithme pour le HUIM sont les suivants :

1. Si l'algorithme utilise une recherche en profondeur ou en largeur ?
2. Si l'algorithme est basé sur deux phases ou une seule phase.
3. Les stratégies d'élagages qui sont utilisés pour rechercher des itemsets à haute utilité.
4. Représentation des itemsets utilisé par l'algorithme.
5. Comment ils génèrent ou déterminent les prochains itemsets à explorer dans l'espace de recherche ?

Ces choix de conception influencent les performances de ces algorithmes en termes de : Temps d'exécution, utilisation de la mémoire et la facilité avec laquelle ces algorithmes peuvent être implémenté et étendu pour d'autres tâches d'exploration de données [13].

Nous allons présenter maintenant les deux types d'algorithmes de HUIM : les algorithmes basés sur deux phases et les algorithmes basé sur une seule phase.

3.4.1 Algorithmes Basés sur Deux Phases (Two-Phase based Algorithms) :

Comme il est indiqué dans leurs noms, ces algorithmes consistent à réaliser deux phases pour trouver les itemsets à haute utilité. Généralement, la première phase est utilisée pour générer un ensemble d'itemsets candidats à haute utilité en surestimant leurs valeurs d'utilité. Ensuite, les algorithmes effectuent une analyse de la base de données pour calculer l'utilité exacte des candidats et filtrer les ensembles d'éléments à faible utilité dans la deuxième phase. Basé sur cette idée, plusieurs algorithmes ont été développés tels que l'algorithme *Two-Phase*, *IHUP* et *UPGrowth [12]*.

Pour faire face à la difficulté de HUIM et pour pouvoir réduire l'espace de recherche, les algorithmes à deux phases adoptent la mesure *d'utilité pondérée par les transactions,* en anglais *Transaction-Weighted Utilization (TWU)* qui est une borne supérieure sur la mesure d'utilité [14]. *TWU* peut être utilisé pour réduire en toute sécurité l'espace de recherche et ignorer les ensembles d'éléments à faible utilité sans manquer aucun ensemble d'éléments à utilité élevée. Les algorithmes basés sur deux phases utilisent la mesure *TWU* dans leurs stratégies d'élagage pour éliminer les itemsets non promoteurs.

Pour pouvoir calculer la mesure *TWU* des itemsets, on doit d'abord calculer les utilités des transactions :

- **Définition 9 (Utilité d'une transaction).** L'utilité d'une transaction T_c, notée par $u(T_c)$, est la somme des utilités de tous les éléments inclus dans T_c. Mathématiquement, $TU(T_c) = \sum_{x \in T_c} u(x, T_c)$ [12].
- **Exemple.** On prend toujours la base représentée par les Tableaux 3.2 et 3.3, $TWU(T_3) = u(a, T_3) + u(c,T_3) + u(e,T_3) = (2\times5)+(6\times1)+(2\times3)=22.$
- **Définition 10 (TWU d'un itemset X).** *TWU* d'un itemset *X* est la somme des utilités des transactions qui contiennent X. En d'autres termes, $TWU(X) = \sum_{T_c \in g(X)} tu(T_c)$ ..
- **Exemple. L'itemset {bc} apparaît en trois transactions dans la base** représentée par les Tableaux 3.2 et 3.3, il apparait dans T_0, T_1 et T_4. Alors, $TWU(bc) = u(T0)+u(T1)+u(T4)= 25+20+9= 54.$

Deux spécificités importantes de la mesure *TWU* est que : (1) $\forall X \in D$, on a toujours, $TWU(X) \geq u(X)$. (2) De plus, l'utilité d'un itemset X est toujours supérieur ou égale aux utilités de toute les sur ensembles (supersets) de X, C-à-d, $\forall X \in D, TWU(X) \geq TWU(Y) / X \subseteq Y$. Ces deux spécificités font que *TWU* soit une borne sur la mesure d'utilité [14]. Et donc, La mesure *TWU* est utilisée par les algorithmes ci-dessus pour réduire l'espace de recherche en fonction de la propriété suivante :

- **Propriété 2 (Élagage de l'espace de recherche à l'aide de TWU).** Pour tout itemset *X*, si $TWU(X) < minutil$, alors *X* est un itemset de faible utilité ainsi que tous ses sur-ensembles.

Les algorithmes basés sur deux phases utilisent la *propriété 2* comme propriété principale pour élaguer l'espace de recherche.

3.4.1.1 Principe de fonctionnement des algorithmes en deux phases

Le principe de fonctionnement des algorithmes en deux phases est le suivant [14]:

Phase 1 : Dans cette phase, ces algorithmes premièrement calculent le *TWU* des itemsets dans l'espace de recherche, puis vérifier que : Pour un itemset X :

1. Si $TWU(X) < X$, alors X et ses supersets ne peuvent pas être itemsets à haute utilité. Alors, ils peuvent être éliminés de l'espace de recherche et leur TWU n'ont pas besoin d'être calculés.
2. Sinon, *X* et ses supersets peuvent être sélectionnés comme itemsets à haute utilité. Ainsi, *X* est conservé et sauvegardé en mémoire en tant qu'itemset candidat à haute utilité et ses supersets seront explorés.

Phase 2 : L'utilité exacte de chaque itemset candidat X trouvé dans la phase 1 est calculée en parcourant la base de données. Si $u(X) \geq minutil$, alors *X* sera conservé car *X* est un itemset à haute utilité.

3.4.1.2 Avantages de processus en deux phases

- Il est garanti que seuls les itemsets à faible utilité sont supprimés de l'espace de recherche [14].

- Les algorithmes à deux phases peuvent trouver tous les itemsets à haute utilité tout en réduisant l'espace de recherche pour améliorer leurs performances.

3.4.1.3 Les limites de processus en deux phases

- Il génère des itemsets en combinant des itemsets sans recourir à la base de données, alors il peut génère des modèles qui n'apparaissent même pas dans la base de données[12].
- Il prend beaucoup de temps pour traiter des itemsets qui n'existent dans la base de données.
- Il scanne à plusieurs reprises la base de données pour calculer les *TWU* et les utilités exactes des itemsets. Ces scannes répétitives de la base font que e temps de ces algorithmes soit très long.
- L'utilisation d'une recherche en largeur peut être assez couteuse en termes de mémoire.

3.4.2 Algorithmes Basés sur Une Phase

Plus tard et pour éviter les limitations des algorithmes basés sur deux phases, la deuxième catégorie d'algorithmes a émergé avec l'apparition de l'algorithme *HUI_Miner* qui ne nécessite qu'une seule phase pour découvrir *HUIM*.

HUI_Miner utilise une structure qui s'appelle *utility-list* pour stocker à la fois les informations d'utilité sur un les itemsets et des informations heuristiques pour élaguer l'espace de recherche et éviter la génération coûteuse de nombreux itemsets candidats comme le font les algorithmes de deux phases [15].

Sur la base de l'idée de *HUI_miner*, d'autres algorithmes plus efficaces ont été proposés tels que *FHM [16]*et *ULB-Miner* [17], etc.

L'efficacité des algorithmes basés sur une seule phase par rapport aux algorithmes de deux phases réside dans le fait que les algorithmes basés sur une seule phase ont également introduit des nouvelles bornes supérieures (upper-bounds) sur l'utilité des qui sont plus performante que *TWU* car ces bornes supérieures sont basées sur l'utilité exacte des itemsets, et peuvent donc élaguer une plus grande partie de l'espace de recherche par rapport à la mesure *TWU*.

Ces bornes supérieures incluent les utilités restantes (*remaining utility*), et des mesures plus récentes telles que l'utilité locale (*local utility*) et l'utilité de sous-arbre *(sub-tree utility).*

3.4.2.1 La Structure Utilité-Liste (Utility-list Structure)

Avant de présenter l'algorithme FHM que nous avons utilisé dans notre étude, nous allons présenter d'abord *l'utility-list* structure qui est la brique de base de cet algorithme. La structure *Utility-list* a été adoptée la première fois avec l'algorithme HUI-Miner. Ensuite, d'autre algorithmes plus efficaces ont été proposés et qui sont aussi base sur la structure *utility-list* tels que FHM.

L'utility-list structure est une façon pour représenter les itemsets. Cette représentation permet de calculer rapidement l'utilité des différents itemsets sans besoin de scanner la base de données. De plus, *utility-lists* de itemsets larges peuvent être rapidement créées en joignant des *utility-lists* de itemsets courts [15].

- **Définition 11 (Utility-list d'un itemset X).** Étant donné une base de données quantitative D, un ensemble d'items $I = \{i_1, i_2, \ldots, i_n\}$ et une relation ordre total $\succ$ prédéfinie sur l'ensemble des items I. *utility-list* de *X*, *ul(X)*, dans *D* est une ensemble de tuples où chaque tuple $(T_{id}, iutil, rutil)$ est définie pour chaque transaction T_{id} contenant X. Le *iutil* est l'utilité de *X* dans la transaction T_{id} tandis que le *rutil* est l'utility restante, *remaining utility*, et elle est définie par : $rutil(X) = \sum_{i \in T_{id} \text{ et } i \succ x \ \forall x \in X} u(i, T_{id})$ [12].
- **Exemple.** On se base toujours sur la base présentée dans les Tableaux 3.2 et 3.3, les *utility-lists* des itemsets *{a}, {d}* et *{a,d}* sont illustrées dans la Figure 3.1.

The utility-list of $\{a\}$

tid	iutil	rutil
T_0	5	20
T_2	5	3
T_3	10	12

The utility-list of $\{d\}$

tid	iutil	rutil
T_0	6	3
T_1	6	3
T_2	2	0

The utility-list of $\{a, d\}$

tid	iutil	rutil
T_0	11	3
T_2	7	0

Figure 3.1. Utility-lists des itemsets {a}, {d} et {ad}

L'utilité exacte d'un itemset *X* peut facilement être calculée à l'aide de sa *utility-list* correspondante. Plus précisément, l'utilité de l'itemset *X* est la somme de toutes les valeurs *iutil* de sa liste d'utilités.

- **Exemple.** Dans la Figure 3.1, l'utilité exacte de l'itemset *{ad}, u{ad}=11+7=18.*

De plus, *utility-lists* peuvent être utilisées pour élaguer l'espace de recherche en utilisant la propriété 3. Cette propriété est basée sur le calcule d'une nouvelle borne supérieure sur la mesure d'utilité et plus efficace que la borne supérieure *TWU*. Cette borne supérieure est appelée *la borne supérieure de l'utilité restante*, en anglais, *remaining utility upper bound*. Elle est définie comme suit :

- **Définition 12. (Remaining utility upper-bound (reu)).** Étant donné un itemset *X* avec sa *utility-list* correspondante *ul(X)*, *Remaining utility upper-bound* de *X* est la somme des toutes les valeurs de *iutil* et *rutil* de X. Elle est calculée par : $ru(X) = \sum_{e \in ul(X)}(e.iutil + e.rutil)$.
- **Exemple.** Dans la Figure 3.1, *Remaining utility* de l'itemset *{ad}* est, *ru({ad})=11+7+3+0=21*.

remaining utility upper bound peut etre utilisée pour elageuer l'espace de recherche en en utilisant la propriété 3.

- **Propriété 3 (Élagage de l'espace de recherche à l'aide d'une liste d'utilitaires en utilisant la borne supérieure *remaining utility*).** Étant donné un itemset *X* avec sa *utility-list* correspondante *ul(X),* si la somme des valeurs *iutil* et *rutil* dans *ul(X)* est

inférieure au seuil d'utilité minimum *(minutil), X* et toutes ses extensions (supersets) sont des itemsets de faible utilité.

- **Exemple.** On voie dans la Figure 3.1 que *ru({ad})=11+7+3+0=21*. Si l'utilisateur choisis 25 comme une valeur de seuille d'utilité, *minutil=25,* alors l'itemset *{ad}* avec tous ses sur-ensemble seront éliminées parce que ce sont certainement des itemsets à faible utilité.

3.5 L'algorithme FHM (Faster High-Utility Itemset Mining Algorithm)

Parmi les algorithmes populaires d'HUIM l'algorithme FHM, FHM est l'acronyme de « Faster High-Utility Itemset Mining Algorithm ». FHM est un algorithme basé en une seule phase et il utilise la structure *utility-list.* Des études ont révélé que l'algorithme FHM est 7 fois plus rapide qu'HUI-Miner.

L'algorithme FHM est présenté par la Figure 3.2. FHM prend en entré deux paramètres : (1) la base de données transactionnelle quantitative D et (2) le seuille minimal d'utilité (*minutil*). FHM adopte une recherche en profondeur (*DFS search*) pour explorer l'espace de recherche des itemsets [13].

Algorithm 1: The FHM algorithm

input : D: a transaction database, *minutil*: a user-specified threshold
output: the set of high-utility itemsets

1 Scan D to calculate the TWU of single items;
2 $I^* \leftarrow$ each item i such that TWU$(i) <$ *minutil*;
3 Let $\succ$ be the total order of TWU ascending values on I^*;
4 Scan D to built the utility-list of each item $i \in I^*$ and build the $EUCS$ structure;
5 `Search` $(\emptyset, I^*, minutil, EUCS)$;

Figure 3.2. L'algorithme FHM

Durant le processus d'exploration des itemsets, FHM utilise trois stratégies pour réduire l'espace de recherche et éliminer autant que possible les itemsets non promoteurs. En d'autres termes, les itemsets à faible utilité.

Les étapes de FHM sont comme suit :

3.5.1 Etape 1 : Calculer la Mesure TWU des Items Initiaux

FHM commence par un scan de la base de données pour calculer la mesure de *TWU* de tous les itemsets initiaux de la base *D* (Voir line 1 dans Figue 3.2). Les itemsets initiaux sont les itemsets de longueur 1 ou tout simplement items.

3.5.2 Etape 2 : Eliminer les Items Non Promoteurs

L'algorithme ensuite effectue la première stratégie d'élagage pour éliminer tous les items ayant TWU inférieur au seuille *minutil* (line 2). Cette tâche est faite en utilisant la *propriété 1* illustrée précédemment.

3.5.3 Etape 3 : Construire les Utility-lists des Itemset Promoteurs et la Structure EUCS

A ce point, FHM effectue un deuxième scan de la base de données pour construire les *utility-lists* des items promoteurs qui n'ont été pas éliminés par la première stratégie d'élagage (line 4). De plus, FHM construire aussi une structure qui s'appelle, *EUCS (Estimated Utility Co-Occurrence Structure)*. EUCS est défini comme suit :

- **Définition 13 (La structure *EUCS* (*Estimated Utility Co-Occurrence Structure*).** *EUCS* d'une base de données *D* est composée d'un ensemble de triplets de la forme *(a, b, c)* sachant que : *a* et *b* sont de items et *c* est une valeur qui corresponde à la valeur TWU de itemset ({ab}). En d'autres termes, *c=TWU({ab}) [12]*.
- **Exemple.** La figure suivante présente la structure EUCS correspondante à la base D des Tableaux 3.2 et 3.3.

Item	a	b	c	d	e	f
b	30					
c	65	61				
d	38	50	58			
e	57	61	88	50		
f	30	30	30	30	30	
g	27	11	38	0	38	0

Figure 3.3. La structure EUCS de la base D

EUCS contient tous les co-occurrences de la base D avec les valeurs TWU correspondantes. *EUCS* sera utilisé par FHM après pour élaguer d'une façon rapide l'espace de recherche durant la procédure récursive.

3.5.4 Etape 4 : Effectuer la procédure de recherche récursive

Après la construction de *EUCS*, l'exploration de recherche en profondeur des itemsets commence en appelant la procédure récursive *Search (line 5).* La procédure récursive *search* est présentée dans la Figure 3.4. L'idée de cet algorithme est de représenter l'ensemble de tous les itemsets possibles sous forme d'un graphe et ensuite l'algorithme effectue une recherche par profondeur pour explorer ce graphe. L'algorithme part des itemsets de taille *1* et explore les itemsets de plus grande taille en effectuant ce qu'on appelle le processus de construction (*construct process*) (Voir line 10 de Figure 3.4).

Avant d'explorer des itemsets plus large l'algorithme vérifie d'abord l'utilité exacte de l'itemset courant et enregistrer ce dernier si son utilité est supérieure au seille *minutil*. L'algorithme vérifie l'utilité exacte des itemsets en utilisé son utilité liste correspondante.

Algorithm 2: The *Search* procedure

input : P: an itemset, *ExtensionsOfP*: a set of extensions of P, the *minutil* threshold, the *EUCS* structure
output: the set of high-utility itemsets

```
1  foreach itemset Px ∈ ExtensionsOfP do
2      if SUM(Px.utilitylist.iutils) ≥ minutil then
3          output Px;
4      end
5      if SUM(Px.utilitylist.iutils)+SUM(Px.utilitylist.rutils) ≥ minutil then
6          ExtensionsOfPx ← ∅;
7          foreach itemset Py ∈ ExtensionsOfP such that y ≻ x do
8              if ∃(x, y, c) ∈ EUCS such that c ≥ minutil) then
9                  Pxy ← Px ∪ Py;
10                 Pxy.utilitylist ← Construct (P, Px, Py);
11                 ExtensionsOfPx ← ExtensionsOfPx ∪ Pxy;
12             end
13         end
14         Search (Px, ExtensionsOfPx, minutil);
15     end
16 end
```

Figure 3.4: La procédure récursive Search

Le processus de construction est illustré par la Figure 3-5 Cette opération serte à construire les *utility-lists* d'itemsets plus grands en joignant les *utility-lists* des petits itemsets. Cette construction est faite sans besoin de scanner la base de données. La Figure 3.1 donne un exemple de construction de *l'utility-list* de l'itemset *{ad}* à partir des itemsets *{a}* et *{d}*.

Algorithm 3: The Construct procedure

input : P: an itemset, Px: the extension of P with an item x, Py: the extension of P with an item y
output: the utility-list of Pxy

```
1  UtilityListOfPxy ← ∅;
2  foreach tuple ex ∈ Px.utilitylist do
3      if ∃ey ∈ Py.utilitylist and ex.tid = exy.tid then
4          if P.utilitylist ≠ ∅ then
5              Search element e ∈ P.utilitylist such that e.tid = ex.tid.;
6              exy ← (ex.tid, ex.iutil + ey.iutil − e.iutil, ey.rutil);
7          end
8          else
9              exy ← (ex.tid, ex.iutil + ey.iutil, ey.rutil);
10         end
11         UtilityListOfPxy ← UtilityListOfPxy ∪ {exy};
12     end
13 end
14 return UtilityListPxy;
```

Figure 3.5. Le processus de construction

Il est important de noter que, la procédure récursive *search* ne fait pas naïvement le processus de construction mais il vérifie d'abord s'il est nécessaire de faire la construction ou non. Cette vérification est faite grâce à deux stratégies d'élagage qui permette d'éviter de construire des *utility-lists* pour des itemsets qui sont certainement non promoteurs. Les deux stratégies d'élagage sont : (1) Elagage par utilité restante *(remaining utility)* et (2) Elagage par co-occurrence d'utilité *(Co-occurrence-based Pruning).* Nous avons déjà expliqué en détail l'élagage par co-occurrence, le lecteur peut se référer à la *Définition 12* et *Propriété 3* pour avoir plus de détails.

L'élagage par co-occurrence, *Co-occurrence based Pruning*, est basé sur la structure EUCS et il est définie par la propriété suivante :

- **Propriété 4 (Élagage de l'espace de recherche à l'aide l'co-occurrence d'utilité).** Étant donné une structure *EUCS* d'une base *D* et deux items *a* et *b* de *D*, on extrait le triple *(a,b,c)* de *EUCS. Si c* est inférieure au seuil d'utilité minimum *minutil*, alors l'itemset {ab} avec tous ces supersets sont des itemsets à faible utilité.

Mathématiquement, si $c=twu\{ab\} \leq minutil$, alors l'itemset $\{ab\}$ avec tous ses supersets seront éliminés.

3.5.5 Etape 5. Retourner l'ensemble de tous les itemsets à haute utilité

Une fois que l'algorithme a terminé la recherche récursive, l'ensemble de tous les itemsets à haute utilisé est affichée sur l'écran et enregistrée dans un fichier texte.

3.6 Fouille des Motifs à Haute Utilité Corrélés (Correlated HUIM)

3.6.1 Pourquoi *Correlated HUIM ?*

Les algorithmes de *HUIM* peuvent découvrir des itemsets générant un profit élevé mais contenant des items faiblement corrélés. Ces itemsets sont parfois inutiles pour prendre des décisions de marketing.

Par exemple, considérons une base de données transactionnelle d'un magasin de vente aux détails, Les algorithmes de HUIM peuvent trouver que l'achat d'une machine à laver avec un jouet pour enfants est un itemset à haute utilité, car ces deux articles ont globalement généré un profit élevé lorsqu'ils sont vendus ensemble. Mais ce ne serait pas utile d'utiliser ce itemset pour promouvoir la machine à laver des jouets d'enfants car si l'on y regarde de près, ces deux articles sont rarement vendus ensemble. En d'autres termes, malgré la vente de ce deux items donne un profit élevé, ces deux items sont faiblement corrélés [18].

Pour avoir des itemsets plus significatifs que les itemsets à haute utilité, il est préférable d'utiliser le *Correlated HUIM* ou *CHUIM* qui va nous permettre de découvrir l'ensemble des itemsets à haute utilité et corrélés, en anglais *correlated high utility itemsets*.

3.6.2 Concepts Préliminaires

- **Définition 14 (Le *support conjonctif* d'un itemset *X*).** Le support conjonctif *consup* d'un itemset *X* dans une base de données *D* est noté comme *consup (X)* et défini par : $consup(X)=|g(X)|$, où *g(X)* est les transactions qui contiennent *X [19]*.

- **Définition 15 (Le *support disjonctif* d'un itemset *X*)**. Le support disjonctif dans une base de données *D* est noté comme *dissup(X)* et défini par : *dissup(X) = {|T ∈ D| X ∩ T= ∅} [19]*.
- **Définition 16 (La mesure *bond* de l'itemset X).** Le *Bond* d'un itemset X est défini comme : $Bond(X) = \frac{consup(X)}{dissup(X)}$. La mesure de *Bond* d'un itemset est toujours entre *0* et *1 [19]*.
- **Définition 17 (Itemset corrélé ou *Correlated itemset*).** Etant donné une base de données transaction *D* et un seuil de corrélation minimal définie par l'utilisateur *minbond*, On dit qu'un itemset *X* est corrélé si, *bond(X) ≥ minbond*. Notez que le seuil de corrélation minimal *minbond* est toujours entre 0 est 1 (*0 ≤ minbond ≤ 1) [19]*.

Une spécificité importante de la mesure de *Bond* est qu'elle est *anti-monotone*. En d'autres termes, cette mesure est une borne supérieure. Par conséquent, cette mesure peut être comme stratégie d'élagage utilisée pour élaguer les itemsets non corrélés avec tous leurs sur-ensembles (supersets).

- **Propriété 5 (Élagage de l'espace de recherche à l'aide la stratégie d'élagage basé sur la mesure de *bond*).** Étant donné un itemset *X* avec sa correspondante mesure de *Bond*, si *Bond(X)< minbond*, alors l'itemset *X* avec tous ses sur-ensembles sont des itemsets faiblement corrélés. Donc, on doit les éliminés [19].
- **Définition 17 (Itemset à haute utilité corrélé ou *Correlated high utility itemset (CHUI)*).** Etant donné une base de données transactionnelle *D*, un seuil de corrélation minimal *minbond* et un seuil d'utilité minimal *minutil*, on dit qu'un itemset *X* est itemset à haute utilité et corrélé *(CHUI)* si, (*bond(X) ≥ minbond et u(X) ≥ minutil) [19]*.

En se basant sur ces concepts ci-dessus, on peut définir la fouille de itemsets à haute utilité et corrélé ou *correlated HUIM* comme suit.

3.6.3 Problème de *Correlated High Utilité Itemset Mining (CHUIM)*

Etant donnée une base de données transaction *D*, un seuil de corrélation minimal *minbond* et un seuil d'utilité minimal *minutil*, La tâche de *CHUQIM (Correlated High Utility Quantitative*

Itemset Mining) vise à énumérer tous les itemsets qui sont à la fois à haute utilité et corrélés (CHUQI) [18, 19].

3.7 L'algorithme FCHM (Fast Correlated High-Utility Itemset Mining)

L'algorithme FCHM est un algorithme de fouille de données qui permet de découvrir des itemsets à haute utilité corrélés dans un ensemble de données transactionnelles. L'algorithme FCHM est basé sur l'algorithme FHM avec des stratégies d'élagage additionnelle pour éliminer les itemsets non corrélée. FCHM est présenté dans la Figure 3.6. FCHM prend en entré trois paramètres : (1) la base de données transactionnelle quantitative D, (2) le seuille minimal d'utilité (*minutil*) et (3) le seuil minimal de corrélation *(minbound) [19]*.

Algorithm 1: The FCHM algorithm

input : D: a transaction database, $minutil$: a user-specified threshold, $min\_measure$: a user-specified threshold for the $all\text{-}confidence$ or $bond$ measure

output: the set of correlated high-utility itemsets

1 Scan D to calculate the TWU of single items;
2 $I^* \leftarrow$ each item i such that TWU$(i) \geq minutil$;
3 Let $\succ$ be the order of TWU ascending values on I^*;
4 Scan D to built the utility-list of each item $i \in I^*$ and build the $EUCS$;
5 Output each item $i \in I^*$ such that SUM$(\{i\}.utilitylist.iutils) \geq minutil$;
6 **Search** $(\emptyset, I^*, minutil, EUCS, min\_measure)$;

Figure 3.6. L'algorithme FCHM

3.7.1 Etapes 1, 2 et 3

Les trois premières étapes de FCHM sont exactement les même que FHM, l'étape 1 consiste à calculer la mesure TWU des items initiaux, l'étape 2 consiste à éliminer les itemsets non promoteurs et l'étape 3 consiste à construire les *Utility-lists* des itemset promoteurs et la structure EUCS. La principale différence réside dans l'étape 4.

3.7.2 Etape 4 : La procédure de recherche récursive de FCHM

La procédure de recherche récursive de FCHM est présentée dans la Figure 3.7. FCHM suivie une procédure de recherche similaire à celle de FHM avec des stratégies d'élagage supplémentaires pour ne conserver que les itemsets à haute utilité et corrélé en même temps.

Algorithm 2: The *Search* procedure

input : P: an itemset, $ExtensionsOfP$: a set of extensions of P, $minutil$: a user-specified threshold, $EUCS$: the $EUCS$ structure
output: the set of high-utility itemsets

```
1  foreach itemset Px ∈ ExtensionsOfP do
2      if SUM(Px.utilitylist.iutils)+SUM(Px.utilitylist.rutils) ≥ minutil then
3          ExtensionsOfPx ← ∅;
4          foreach itemset Py ∈ ExtensionsOfP such that y ≻ x do
5              if ∃(x, y, c) ∈ EUCS such that c ≥ minutil then
6                  Pxy ← Px ∪ Py;
7                  Pxy.utilitylist ← Construct (P, Px, Py);
8                  ExtensionsOfPx ← ExtensionsOfPx ∪ Pxy;
9                  if SUM(Pxy.utilitylist.iutils) ≥ minutil then output Px;
10             end
11         end
12         Search (Px, ExtensionsOfPx, minutil);
13     end
14 end
```

Figure 3.7. La procédure récursive Search de FCHM

Pour pouvoir appliquer les stratégies d'élagage des itemsets non corrélés, une modification est faite sur la structure *utility-list* en ajoutant ce qu'on appelle *Vecteur binaire disjonctif,* en anglais *disjunctive bit vector.* Un *vecteur binaire disjonctive* nous permet de calculer la mesure de *Bond* d'un itemset à partir de son *utility-list* sans besoin de scanner la base.

A partir de cette nouvelle structure de l*'utility-list,* la procédure récursive va appliquer les trois stratégies d'élagage de FHM présentées précédemment. Ainsi que quatre nouvelles stratégies d'élagage qui sont conçue pour éliminer les itemsets faiblement corrélés.

Ces strategies sont : *Strategy 1. Directly Outputting Single items (DOS), Strategy 2 Pruning Supersets of Non correlated itemsets (PSN), Strategy 3. Pruning using the Bond Matrix (PBM)* et *Strategy 4. Abandoning Utility-List construction early (AUL).*

3.7.3 Etape 5 : Retourner l'ensemble de tous les itemsets à haute utilité et corrélé.

Dès que l'algorithme termine la recherche récursive, l'ensemble de tous les *itemsets à haute utilisé corrélés* est affichée sur l'écran et enregistrée dans un fichier texte.

3.8 Comparaison entre FHM et FCHM

La différence entre FHM et FCHM réside dans les points suivants :

1. L'algorithme FCHM (Fast Correlated high-utility itemset Miner) est une extension de l'algorithme FHM (Faster high utility itemset mining) qui permet de découvrir des ensembles d'items à haute utilité corrélés [18].
2. L'algorithme FHM est utilisé pour extraire des ensembles d'items à haute utilité à partir d'une base de données transactionnelle. Il utilise une approche basée sur la génération de candidats et l'élagage pour réduire l'espace de recherche. Il utilise également une technique de tri pour accélérer le processus d'extraction.
3. L'algorithme FCHM utilise une approche similaire, mais il prend en compte la corrélation entre les items. Il utilise une mesure de corrélation pour identifier les ensembles d'items qui ont une forte corrélation et une haute utilité. Il utilise également une technique de tri pour accélérer le processus d'extraction [18].

3.9 Conclusion

Dans ce chapitre nous avons donné une étude détaillée sur le domaine de la fouille de motifs. Nous avons présenté premièrement le problème de base de ce domaine qui est la fouille des itemsets fréquents. Ensuite, nous avons expliqué en détails la fouille des itemsets à haute utilité en prenant FHM, un des algorithmes les plus populaires dans ce domaine. Finalement, nous avons

parlé d’une extension très intéressante de la fouille des itemsets à haute utilité. Cette extension consiste à découvrir les itemsets qui à la fois ayant haute utilité et sont corrélés.

Chapitre 4: Implémentation et Validation des Résultats

4.1 Introduction

Dans le domaine de l'exploitation de données, une tâche importante est de rechercher des motifs intéressants dans une base de données transactionnelle. Les algorithmes FHM et FCHM sont deux algorithmes efficaces pour effectuer cette tâche.

Dans ce chapitre, nous avons implémenté ces deux algorithmes sur deux bases de données transactionnelles différentes. Les base de données sont des bases réelles. La première base de données *cosmétique* concerne les vente des produits dans une boutique de cosmétique, tandis que la deuxième base de données *e-commerce* est une base contenant les transactions des clients d'une boutique en ligne. Nous avons comparé les performances des deux algorithmes sur ces deux bases de données en termes de temps d'exécution, mémoire utilisation et nombre de motif découverts ainsi que la qualité des résultats obtenus.

4.2 Outils de Développement

4.2.1 NetBeans

NetBeans est un environnement de développement intégré (IDE) populaire pour le langage de programmation *Java*. *NetBeans* offre un ensemble complet d'outils et de fonctionnalités pour aider les développeurs à écrire, déboguer et déployer des applications Java. *NetBeans* est open source et disponible gratuitement, ce qui en fait un choix populaire parmi les développeurs Java. *NetBeans* offre un large éventail de fonctionnalités telles que la complétion de code, le débogage, le profilage et l'intégration du contrôle de version. Il prend également en charge divers *frameworks* et technologies tels que JavaFX, Spring, Hibernate et Maven. *NetBeans* est disponible pour les systèmes d'exploitation Windows, macOS et Linux.

4.3 La bibliothèque SPMF *(Sequential Pattern Mining Framework)*

4.3.1 Un Aperçu sur SPMF

SPMF est un logiciel open source et une bibliothèque d'exploration de données écrite en Java, spécialisée dans la fouille de données de données et la fouille de motifs (la découverte des motifs intéressants dans les données) [20].

SPMF offre un large éventail d'algorithmes et d'outils pour découvrir différent type de motifs à partir de différents types de données tels que les bases de données transactionnelles, les journaux d'événements, les séries chronologiques et les graphes. La bibliothèque est implémentée en Java et peut être facilement intégrée dans des projets Java. De plus, SPMF peut être facilement intégrée dans des projets Java est et peut être utilisé pour une variété d'applications, y compris l'exploration de l'utilisation du Web, la bio-informatique et l'analyse financière.

La version actuelle de SPMF est v2.59, cette version offre des implémentations de 254 algorithmes de fouille de données pour :

- Extraction de règles d'association.
- Fouille des itemsets.
- Fouille de motifs séquentiels.
- Fouille des règles séquentielle.
- Prédiction de séquence.
- Fouille de motifs périodique.
- Fouille d'épisodes.
- Fouille de motifs à haute utilité.
- Fouille de série chronologique.
- Regroupement (clustering) et classification.

4.3.2 Variants de SPMF

SPMF est disponible sous deux formats :

1. **La version du code source** inclut tous les algorithmes. Il nécessite une expérience préalable avec Java pour compiler le code source et exécuter les exemples.
2. **La version graphique** fournit une interface utilisateur graphique et une interface de ligne de commande. Cette version est plus facile à utiliser. La figure suivante illustre l'interface de la version graphique de SPMF.

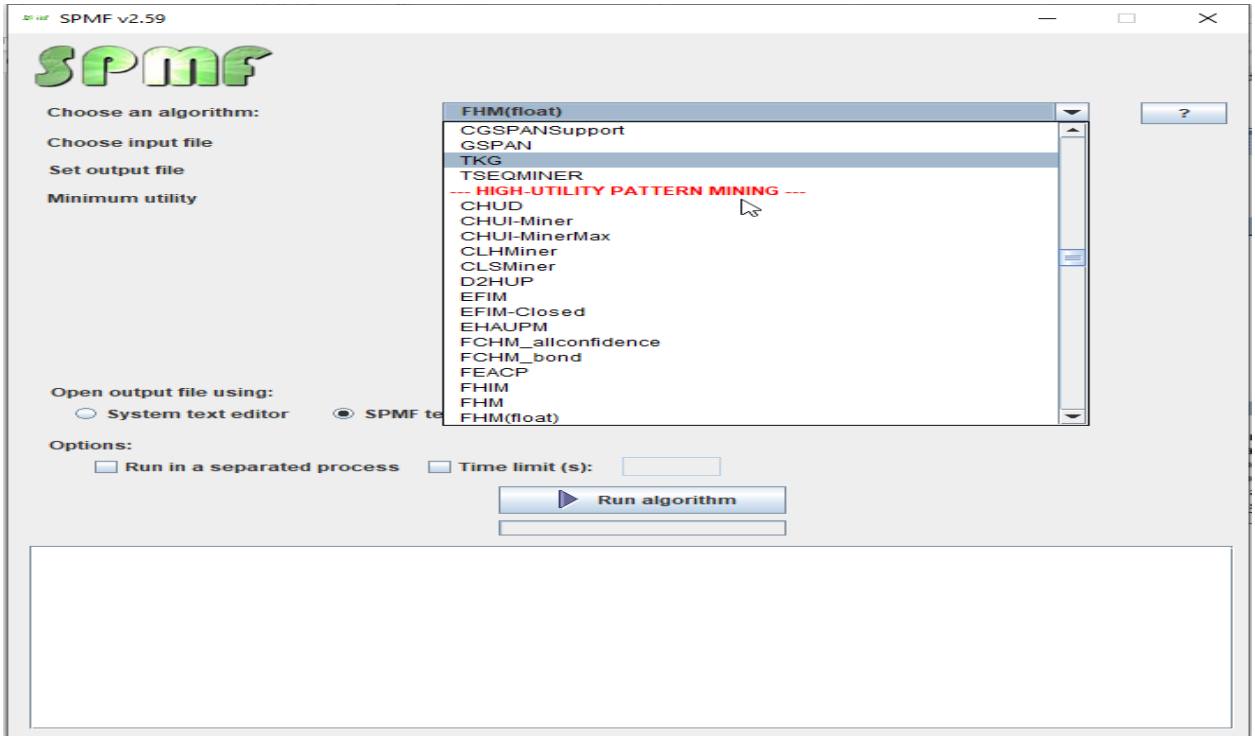

Figure 4.1. L'interface principale de SPMF

4.3.3 Avantages de SPMF

SPMF présente plusieurs avantages significatifs pour les tâches d'extraction de motifs. Voici les avantages de SPMF :

1. **Large choix d'algorithmes :** SPMF offre une vaste collection d'algorithmes pour l'extraction de déférents type de motifs.

2. **Facilité d'utilisation :** SPMF est présente sous format graphique. La version graphique offre une interface conviviale et bien documentée, ce qui facilite la mise en œuvre et l'utilisation des algorithmes.
3. **Variété des motifs découverts par SPMF** : SPMF prend en charge la découverte de plusieurs types de motifs, tels que : les motifs fréquents, les motifs à haute utilité, les épisodes, les motifs périodiques, etc.
4. **Variété des sources données traité par SPMF :** SPMF découvert les motifs dans plusieurs formes de données tel que : les bases de données transactionnelles, les séries chronologiques, les journaux d'événements, les graphes, etc.
5. **Variété d'outils pour l'évaluations des motifs :** SPMF fournit des mesures et des métriques permettant d'évaluer la qualité et l'intérêt des motifs découverts. Les utilisateurs peuvent ainsi évaluer la pertinence des motifs extraits par rapport à leurs critères spécifiques.
6. **Outils pour la visualisation des résultats :** SPMF propose des fonctionnalités de visualisation pour faciliter l'interprétation et l'analyse des motifs séquentiels extraits. Cela permet aux utilisateurs de mieux comprendre les motifs découverts et de les présenter de manière graphique ou sous forme de rapports.
7. **Extensibilité :** SPMF offre une architecture extensible qui permet aux utilisateurs de développer et d'intégrer leurs propres algorithmes personnalisés.

4.4 Le Premier Cas d'Etude : Base de Cosmétique

4.4.1 Présentation de la Base de Cosmétique

Nous avons pris les données que nous allons traiter à partir du magasin de cosmétiques. Les données extraites viennent en deux fichier Excel. Un fichier contient les détails sous forme des lignes : Chaque ligne représente les détails d'un produit qui a été acheté par un client. Chaque ligne est caractérisée par : La référence des produit, le nom de produit, le numéro de bon, la quantité vendue de produit, et le montant de vente de ce produit. Le deuxième ficher contient la liste des produits avec leurs informations. La base contient 16303 produits.

Les deux figures suivantes donnent un aperçu sur les deux fichiers de cette base.

IDDetailBon	IDBon	IDProduit	Designation	Qte	Prix_Achat	PrixV_Detail	profile
1573	18	4407	KASSA HAODA NOIRE	1	100	150	50
4548	40	548	SAVON DOVE	1	155	170	15
4541	40	1377	BAD SENSODYNE DOUCEUR	1	360	400	40
4564	40	1483	CRE EMILY	1	160	200	40
4544	40	1663	PORTE SAVON FLOWER	1	43	70	27
4543	40	2621	GLD JOHNSONS BEBE 200ML	1	350	400	50
4547	40	2790	COTON TIEGE FAMILY	4	73	90	68
4539	40	3122	DNT COLGATE MAX WHITE ONE INTENSE	1	255	350	95
4563	40	3240	EYLINER KISS BEAUTY LOVELY	4	125	200	300
4552	40	4951	COUCHES DEROIT FRELAX PS	1	130	160	30
4567	40	5854	SAVON LIQUIDE VENUS 400ML	8	99	120	168
4553	40	7396	PROTEGE SLIPE SLIM	1	71	80	9
4538	40	7732	LINGETTE MIO BEBE 72PS FA	6	118	130	72
4545	40	8885	BROSSE BOIS METAL	1	285	350	65
4566	40	9054	EDT EVIDENCE 2EM	2	350	650	600
4557	40	10637	CIRE EL SAIN 250ML	4	140	180	160
4536	40	10814	SERVIETTE HIYA 3PS	16	147,5	180	520
4537	40	10881	SERVIETTE HIYA 2PS	48	105	120	720
4559	40	11283	SERUME GOODY 60ML	6	175	200	150
4558	40	11494	FDT POUDER FLORMAR	1	110	200	90
4551	40	11668	ESOUI TOUSCOTEX MAXI CITRON 2PS	5	200	240	200

Figure 4.2. Un extrait du fichier contenant la liste des produits de la base cosmétique

A	B	C	D	E	F	G
IDDetailBon	IDBon	IDProduit	Designation	Qte	Prix_Achat	PrixV_Detail
1573	18	4407	KASSA HAODA NOIRE	1	100	150
4548	40	548	SAVON DOVE	1	155	170
4541	40	1377	BAD SENSODYNE DOUCEUR	1	360	400
4564	40	1483	CRE EMILY	1	160	200
4544	40	1663	PORTE SAVON FLOWER	1	43	70
4543	40	2621	GLD JOHNSONS BEBE 200ML	1	350	400
4547	40	2790	COTON TIEGE FAMILY	4	73	90
4539	40	3122	DNT COLGATE MAX WHITE ONE INTENSE	1	255	350
4563	40	3240	EYLINER KISS BEAUTY LOVELY	4	125	200
4552	40	4951	COUCHES DEROIT FRELAX PS	1	130	160
4567	40	5854	SAVON LIQUIDE VENUS 400ML	8	99	120
4553	40	7396	PROTEGE SLIPE SLIM	1	71	80
4538	40	7732	LINGETTE MIO BEBE 72PS FA	6	118	130
4545	40	8885	BROSSE BOIS METAL	1	285	350
4566	40	9054	EDT EVIDENCE 2EM	2	350	650
4557	40	10637	CIRE EL SAIN 250ML	4	140	180
4536	40	10814	SERVIETTE HIYA 3PS	16	147,5	180
4537	40	10881	SERVIETTE HIYA 2PS	48	105	120
4559	40	11283	SERUME GOODY 60ML	6	175	200
4558	40	11494	FDT POUDER FLORMAR	1	110	200
4551	40	11668	ESOUI TOUSCOTEX MAXI CITRON 2PS	5	200	240
4555	40	11854	DEO VENUSE NOUVEAU 200ML	4	165	200
4565	40	11871	VERNI GLAM 10ML	1	100	150

Figure 4.3. Un extrait du fichier contenant la liste des transactions de la base cosmétique

4.4.2 Pré-Traitement de la Base

Les fichiers fournis pour cette base ne sont pas adaptables par la bibliothèque SPMF. D'où un processus de pré-traitement est nécessaire pour transformer cette base en base de données transactionnelle sous format SPMF qui est adaptée par l'outil SPMF. Le format SPMF est un fichier texte qui contient un ensemble de ligne, chaque ligne contient les transactions qui ont été faite par un seul client.

Chaque ligne est de la forme :

$$\{item1\ \ item2\ item3\ \ \ldots\}: total\ utilité : \{profit\ 1\ profit\ 2\ profit\ 3\ \ldots\}$$

Où {*item1 item2 item3…*} sont les produits achetés par un client *et {profit1 profit2 profit3…}* représente les prix de ces produits. $total\ utilité$ est l'utilité totale de cette transaction (ligne).

Pour convertir les fichiers de base en format spmf, nous avons utilisé un script *python*. Les transactions sont déterminées par le numéro de bon *(IDBon)*. En d'autres termes, une transaction contient tous les produits achetés et écrits dans un ticket.

Voici un aperçu de la base après la transformation en format SPMF :

```
4407:50.0:50.0
548 1377 1483 1663 2621 2790 3122 3240 4951 5854 7396 7732 8885 9054 10637 10814 10881 11283 11494 11668 11854 11871 11895
12246 12306 12327 12384 12507 12542 12599:4511.0:15.0 40.0 40.0 27.0 50.0 68.0 95.0 300.0 30.0 168.0 9.0 72.0 65.0 600.0
160.0 520.0 720.0 150.0 90.0 200.0 140.0 50.0 20.0 380.0 160.0 60.0 50.0 20.0 112.0 100.0
61 109 2957 12384:425.0:70.0 55.0 200.0 100.0
6149 7623 12650:145.0:50.0 30.0 65.0
44 1552 8531 10612 12616 12616:202.0:60.0 12.0 25.0 45.0 40.0 20.0
4 78 475 5033 10105 10418 11668:500.0:60.0 55.0 80.0 70.0 55.0 60.0 120.0
10967:1060.0:1060.0
15 81 501 11288 12042:337.0:106.0 106.0 86.0 25.0 14.0
15 81 12042:240.0:106.0 106.0 28.0
587 2588 2790 6168 6581 10460 12507 12634 12650:1019.0:30.0 65.0 17.0 470.0 250.0 52.0 20.0 50.0 65.0
548 5410:140.0:60.0 80.0
4 264 529 11668 12121:560.0:30.0 220.0 250.0 40.0 20.0
514:40.0:40.0
11 79 104 126 320 398 435 616 1287 5189 6875 6947 8131 11234 12231:1975.0:50.0 50.0 30.0 50.0 110.0 30.0 50.0 115.0 70.0
400.0 55.0 50.0 20.0 45.0 850.0
2077 9678:135.0:35.0 100.0
9 17 75 284 2076 10856 11388:444.5:35.0 45.0 55.0 70.0 17.5 102.0 120.0
619 2494 5864 7460:605.0:50.0 55.0 150.0 350.0
408 409 534 2395:88.0:10.0 50.0 10.0 18.0
265 492 501 2121 3670 10644 12135 12471:1011.0:30.0 180.0 43.0 430.0 50.0 26.0 92.0 160.0
76 2021 7623:133.0:75.0 28.0 30.0
2444 11870 12055:258.0:48.0 10.0 200.0
356 1365 8255:132.0:27.0 70.0 35.0
1382 1603 6341 12490:172.0:40.0 40.0 22.0 70.0
32 1844 2147 4377:111.0:50.0 10.0 40.0 11.0
238 11854:69.0:34.0 35.0
65 514 7658:124.0:44.0 20.0 60.0
1375 1895 7935 10114 10142 10275 11437:450.0:12.0 200.0 45.0 40.0 53.0 60.0 40.0
1582 4675:90.0:30.0 60.0
65 586 5550:154.0:44.0 40.0 70.0
```

Figure 4.4. Un extrait du ficher SPMF de la base de cosmétique

À ce stade, nous allons appliquer les algorithmes FHM et FCHM sur la base de données et on va effectuer une analyse des résultats pour tirer des conclusions utiles.

4.4.3 Exécution de l'Algorithme FHM :

La Figure 4.5 illustre l'exécution de l'algorithme FHM avec la base de cosmétique en utilisant sa class main dans SPMF, *MainTestFhm_Float.java.* Dans l'exemple de la Figure 4.5, le fichier sous format SPMF est nommé data_3.txt et la valeur de seuil minimal d'utilité *(minutil)* est 35000.

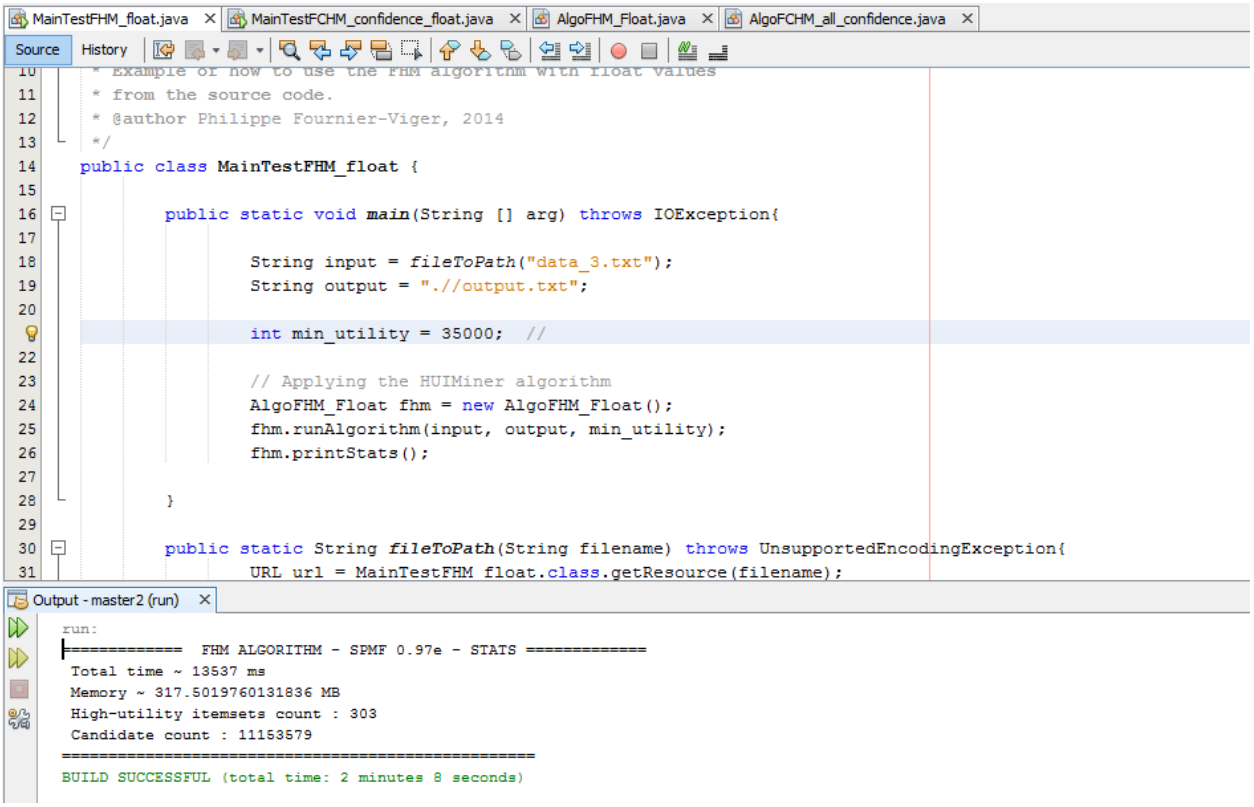

Figure 4.5. Exécution d'algorithme FHM avec la base cosmétique.

Notez que, les résultats obtenus par FHM sont aussi automatiquemnet enregistés par l'algorithme dans un fichier text des resultats. Un extrait de ficher des résultats est présenté par la Figure 4.6.

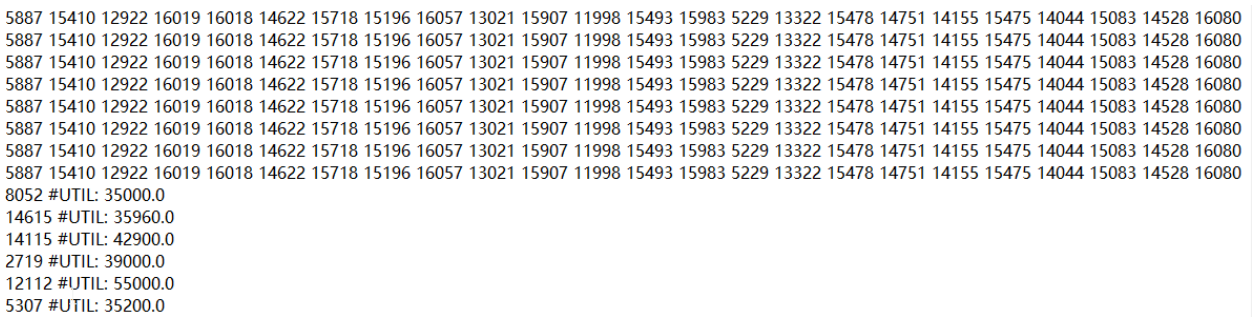

Figure 4.6. Un extrait de fichier des résultats de FHM avec minutil=35000

Le fichier résultant contient 303 lignes. Chaque ligne contient un itemset a haute utilité avec as utilité correspondante. Par exemple, l'itemset 5307 est un itemset à haute utilité qui est égale à 35200.

4.4.3.1 Les Résultats Obtenus

Durant l'application des algorithmes FHM et FCHM, nous avons évaluer ces algorithmes par rapport à trois critères : Le temps d'exécution, l'utilisation du mémoire et le nombre de patterns découverts.

FHM est tester selon plusieurs valeurs de seuil *minutil*. La valeur de seuil est changée de 34990 à 375080. Le résultat de l'algorithme FHM est présenté par le Tableau 4-1 et résumée par les Figures 4.7, 4.8 et 4.9.

Tableau 4-1. Les résultats obtenus par l'algorithme FHM avec la base de cosmétique.

Utility	**34990**	**35000**	**35010**	**35020**	**35030**	**35040**	**35050**	**35060**	**35070**	**375080**
Temps (ms)	3848	2947	2470	2260	2029	2001	1790	1872	1742	1711
Mémoire (MB)	626,43	622, 36	409,98	412,59	278,18	276,67	263, 39	262,93	263,06	232,05
Nombre de High utility itemsets	335	303	271	267	259	259	258	258	258	258

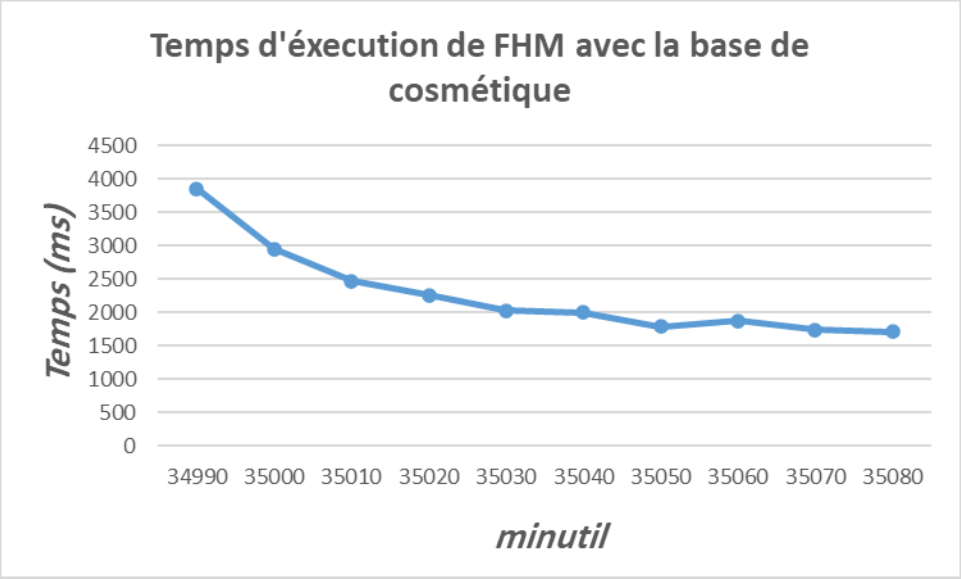

Figure 4.7. Temps d'exécution de FHM avec la base de cosmétique

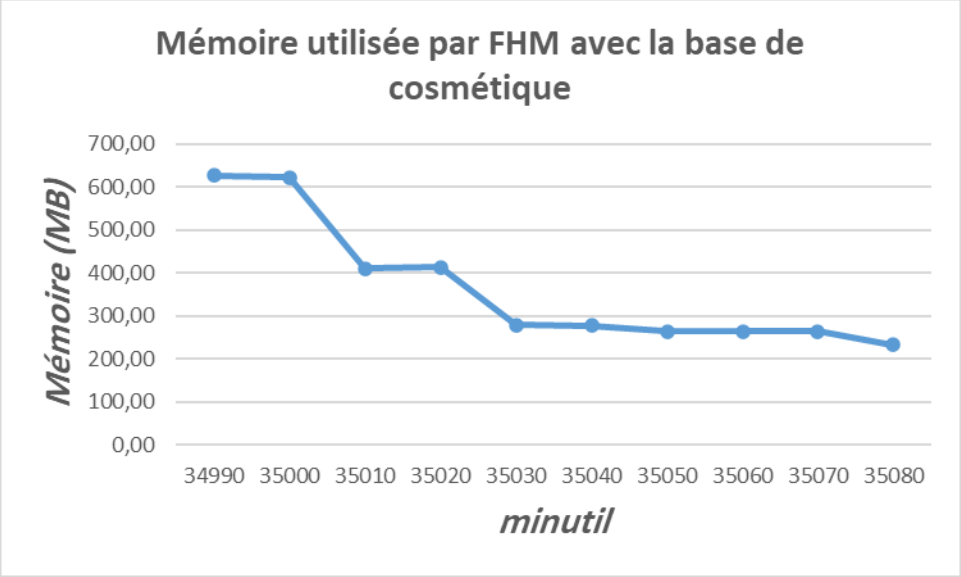

Figure 4.8. Mémoire utilisée par FHM avec la base de cosmétique

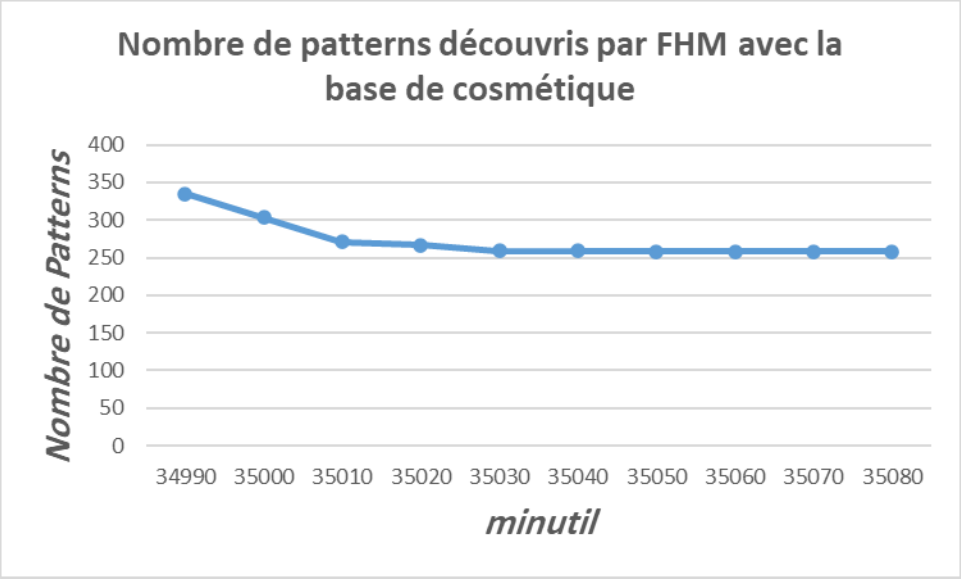

Figure 4.9. Nombre de patterns découverts par FHM avec la base de cosmétique

Nous voyons dans la Figure 4.9 que le nombre de patterns découverts par FHM déminue avec l'augmentation de la valeur de seuil *minutil*. En fait, l'augmentation de *minutil* fait que le nombre des motifs non promoteurs (leur utilité est inférieure à *minutil*) augmente et le nombre des itemsets à haute utilité diminue. Alors, au cas où la valeur de seuil est haute, les stratégies d'élagage vont éliminer plus d'itemsets non promoteurs. Suite à cela, le temps d'exécution est l'utilisation de mémoire vont aussi diminuer avec l'augmentation de *minutil* comme il a été illustré dans les Figures 4.7 et 4.8.

4.4.3.2 Exemples de motifs découverts par FHM

Dans ces expérimentations, nous nous intéressons à vérifier les patterns extraits par FHM pour voire leurs utilités.

a) Exemple 1 (FHM avec minutil=35050)

La figure suivante est capturée de fichier de résultats de FHM avec la valeur de seuil *minutil=35050*. Notez que, le nombre total de motifs est *258*.

```
249 #UTIL: 93335.0
2019 #UTIL: 72943.0
12504 #UTIL: 80990.0
4 #UTIL: 49260.0
11664 #UTIL: 64900.0
6552 #UTIL: 78936.0
2021 #UTIL: 107240.0
246 #UTIL: 137970.0
246 168 #UTIL: 45080.0
246 6969 #UTIL: 39405.0
3593 #UTIL: 41664.0
10066 #UTIL: 78660.0
10066 542 126 #UTIL: 39720.0
10066 126 #UTIL: 72150.0
10066 126 271 #UTIL: 42180.0
10066 271 #UTIL: 36780.0
1423 #UTIL: 51540.0
1617 #UTIL: 74655.0
1422 #UTIL: 43560.0
12398 #UTIL: 65100.0
126 #UTIL: 79450.0
243 #UTIL: 101439.0
1275 #UTIL: 107298.0
1656 #UTIL: 92000.0
1656 6969 #UTIL: 37616.0
2076 #UTIL: 47705.0
1211 #UTIL: 86490.0
408 #UTIL: 83760.0
```

Figure 4.10. Un extrait de fichier des résultats de FHM avec minutil=*35050*

En consultant le ficher de résultat, on voie que tous les itemsets découverts par FHM ont une longueur petite. La langueur de tous ces itemset est entre 1 et 3. Le tableau donne les détails des 3 itemsets qui sont sélectionné en bleu dans la Figure 4.10.

Tableau 4-2. Motifs à haute utilité découverts par FHM avec minutil=35050.

SH Nivea / Savon Nivea / GLD Nivea	Utility=39720
SH Nivea / GLD Nivea / Stick Nivea Roll	Utility=42180
DEO Nivea Men / SH Nivea / GLD Nivea	Utility =41530

Interprétation des résultats

Suite à notre analyse sur les motifs de Tableau 4.2, il a été observé que *Nivea* est la marque préférée des consommateurs en raison de la qualité exceptionnelle de ses produits et de leur prix abordable.

De plus, il a été déduit que la majorité des consommateurs qui achètent le *shampooing Nivea* optent également pour le *gel douche Nivea*. Par conséquence, l'achat du shampooing implique automatiquement l'achat du gel douche.

b) Exemple 2 (FHM avec minutil=35000)

Pour savoir si la diminution de *minutil* va nous permettre d'obtenir des motifs plus intéressants que ceux qui ont été présentés dans le premier exemple, nous avons met dans cet exemple la valeur de *minutil* à 35000. La Figure 4.11 présente quelques motifs découverts par FHM. Notez que, le nombre total de motifs dans ce cas est 303.

Malgré que le nombre de motifs extraits dans cet exemple est plus grand que le nombre de motifs extrait dans le premier exemple, on peut voir dans la Figure 4.11 que les nouveaux motifs résultants ne sont pas très intéressants que les motifs de premier exemple car leurs longueur est très long.

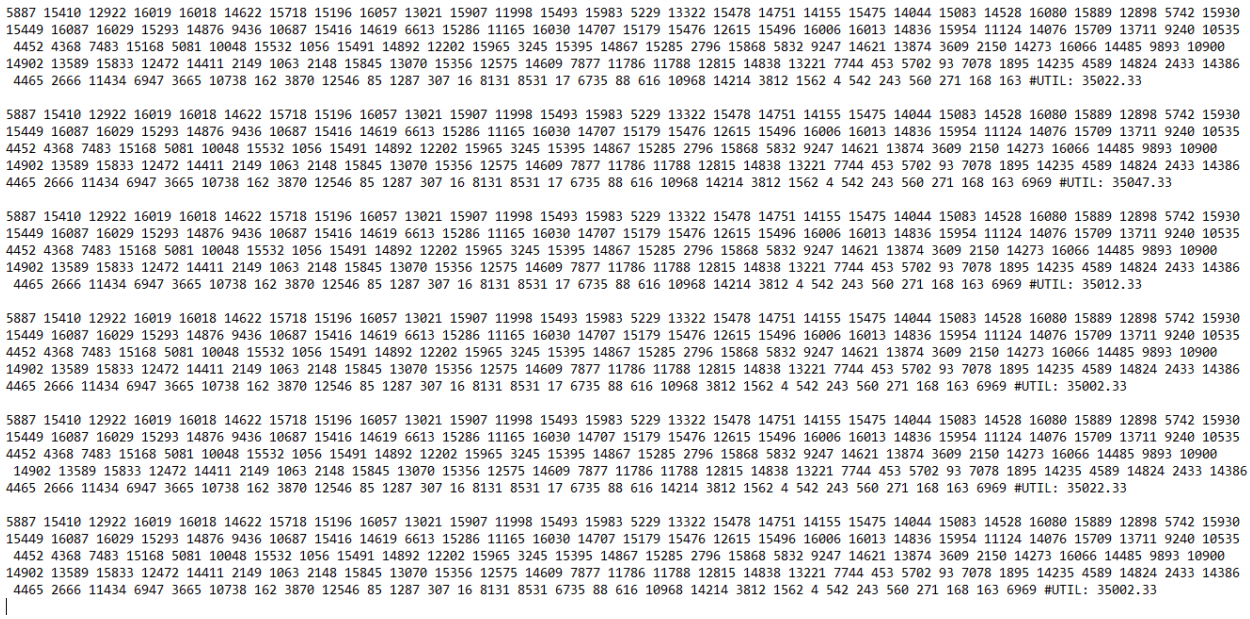

Figure 4.11. Un extrait de fichier des résultats de FHM avec minutil=35000

Tableau 4-3. Motifs à haute utilité découverts par FHM avec minutil=35000.

Séchoir / BabyLiss /badeve coigette electric/ masque elvive hialuromice/eyliner mitalic top/sh elve/selaka /savon pure white/ brl xnotex/ l'eau de rose hemani / bad colgate blancheur/ RGL diana matte/ B rose banat /faire a repacer craft / GEL nettoyant niall/stick higeen/angle fancy press/contoure palette hypo/HB / crème Nivea souffle perle/arap lovly / papier resouit tous Sofia/ HB/ Arap Paston/ crème elvive/ Hadjra plastic/ cherit / sac chanel / porte makiage/ FRP révolution /note skimm perfecting primer/DNT paradontax /BRL top face /DEO nivea /FDT fullcover/anti sern bourjois/sac channel/GLD corine/Gel creme / Pronzer pawder diana/eau miclaire neutrogina/terra cotta blash sh diana/GLD zarra /DEO stick cien/kassa /Fixateur kiss beauty/kit sourcils/Arvea / GON / eau / FRP revolution/FDT bourjois/mascara bourjois/eyliner noir / savon sanex/ sh ultra doux/FDT golden rose/savon classic/savon monsavon/GLD sadermo/BRL matte a diana/ Eye brow gel 3d note /bad oral /creyon sephora/Creme nivea/gel nettoyant/crem de soin/ brume zara/stick dove/ GEL detoxifiant /Pouder libre top face/BRL postel/DNTEmail diamant replenium/tassa hamman/dessoulvant adams/highlighter chanlanya/brosse/remington/mascara maybelline/ bad colgate/ FDT tailaine shine/DNT oral B /Apsh elseve /Tassa hammam/ creme le petit marseil/Tassa hammam/ Lait demaquillant/RGL lip stick top face/Fixature Kiss beauty /verni maria beaty/Brume EJ / compacte pawder note/ FRJ piove/FDT pouder piovie/ SH elvive/ Kassa hammam/Porte savon/ FDT Maybelline baby skin/masque elseve/ lait de corps vaseline/APSH elseve/Bain de douche listerine/creme netoyante/SH elvive/savon lux FR/ hadjra/ Gel nettoyant deep/ HB / savon petit Marseilia/ Savon le petit marseilias / Disque dimaquiller tami / SH dove / lait de core le petit marseilia / creme Nevia soft /mascara essence / deo sanex / lait nevia / crayon mki / GLD dove /talk enchanteur / DNT sen/SH	Utility =35022, 33

elseve / DEO le petit marseilias /savon abusaad / SH dove /creme aven / APSH dove / savon palmolive / eau micellaire venus / RGL even beauty creyon/ SH glis FR / Linghettet nivea baby /SH ultra doux / savon nivea / DEO nivea women / savon dove / stick Nivea roll / DEO rexona new	
Sechoir / Babyliss /badeve coigette electric/ masque elvive hialuromice/eyliner mitalic top/sh elve/selaka /savon pure white/ brl xnotex/ l'eau de rose hemani / bad colgate blancheur/ RGL diana matte/ B rose banat /faire a repacer craft / GEL nettoyant niall/stick higeen/angle fancy press/contoure palette hypo/HB / crème nivea souffle perle/arap lovly / papier resouit tous sofia/ HB/ Arap Paston/ creme elvive/ Hadjra plastic/ cherit / sac chanel / porte makiage/ *FRP reveulution/ note skimm perfecting primer/DNT paradontax /BRL top face /DEO nivea /FDT fullcover/anti sern bourjois/sac channel/GLD corine/Gel creme / Pronzer pawder diana/eau miclaire neutrogina/terra cotta blash sh diana/GLD zarra /DEO stick cien/kassa /Fixateur kiss beauty/kit sourcils/Arvea / GON / eau / FRP revolution/FDT bourjois/mascara bourjois/eyliner noir / savon sanex/ sh ultra doux/FDT golden rose/savon classic/savon monsavon/GLD sadermo/BRL matte a diana/ Eye brow gel 3d note /bad oral /creyon sephora/* *Creme nivea/gel nettoyant/crem de soin/ brume zara/stick dove/GEL detoxifiant /Pouder libre top face/BRL postel/DNTEmail diamant replenium/tassa hamman/dessoulvant adams/highlighter chanlanya/brosse/remington/mascara maybelline/ bad colgate/ FDT tailaine shine/DNT oral B /Apsh elseve /Tassa hammam/ creme le petit marseil/Tassa hammam/ Lait demaquillant/RGL lip stick top face/Fixature Kiss beauty /verni maria beaty/Brume EJ / compacte pawder note/ FRJ piove/FDT pouder piovie/ SH elvive/ Kassa hammam/Porte savon/ FDT Maybelline baby skin/masque elseve/ lait de corps vaseline/APSH elseve/Bain de douche listerine/creme netoyante/SH elvive/savon lux FR/ hadjra/ Gel nettoyant deep/ HB / savon petit Marseilia/* *Savon le petit marseilias / Disque dimaquiller tami / SH dove / lait de core le petit marseilia / creme Nevia soft /mascara essence / deo sanex / lait nevia / crayon mki / GLD dove /talk enchanteur / DNT sen/SH elseve / DEO le petit marseilias /savon abusaad / SH dove /creme aven / APSH dove / savon palmolive / eau micellaire venus / RGL even beauty creyon/ SH glis FR / Linghettet nivea baby /SH ultra doux / savon nivea / DEO nivea women / savon dove / stick Nivea roll / DEO rexona new / acsecoire*	Utility= 35047,33

Interprétation des résultats

En consultant le ficher des résultats, nous avons remarqué qu'on a deux types de motifs : Soit des motifs ayant une petite langueur (3 au max), soit des motifs trop longs comme les motifs présenter dans Tableaux 4.3.

Ces derniers ne sont sélectionnés par FHM que paracerque ce sont des itemsets avec une grande longueur. Il se peut que ces itemsets n'apparaissent qu'une seule ou quelque fois dans la base de données.

Malgré que les itemsets haute utilité mais avec grande langueur sont fréquemment non désirable par l'utilisateur car ce type d'itemset est par approprié pour faire des promotions ou stratégies de marketing, nous avons essayé quand même de vérifier ces itemsets : nous constatons que les produits les plus populaires et les plus vendus appartiennent à la catégorie des maquillages de haute qualité tels que : *Contour Palette HYPO, Bronzer Powder DIANA, Mascara essence LASH,* et *FDT Maybelline Skin.*

Ils sont suivis de près par les produits de soins personnels comme les produits de soins du visage : *Gel nettoyant Niall, eau de rose HEMANI ROZE, Lait demaquillant* ainsi que les produits de soins du corps comme : *Gel douche ZARA, stick HIGEEN,* etc. Ces produits sont trop demandés sur le marché et jouissent d'une grande popularité parmi les consommateurs.

4.4.4 Exécution de l'Algorithme FCHM

Nous avons vu que la variation de *minutil* dans FHM ne donne pas toujours un plus par rapport à les motifs découverts. Dans cette partie, nous allons essayer FCHM dont le but est d'avoir plus de motifs significatifs. La Figure 4.12 présente un exemple d'exécution de FCHM avec la base de cosmétique en utilisant sa class main dans SPMF, *MainTestFCHM_Confidence.java.* Le fichier sous format SPMF est nommé *data_3.txt* et la valeur de seuil minimal d'utilité *minutil* est 10020 et la valeur de de seuil minimal de corrélation est *minbond* est 0.5.

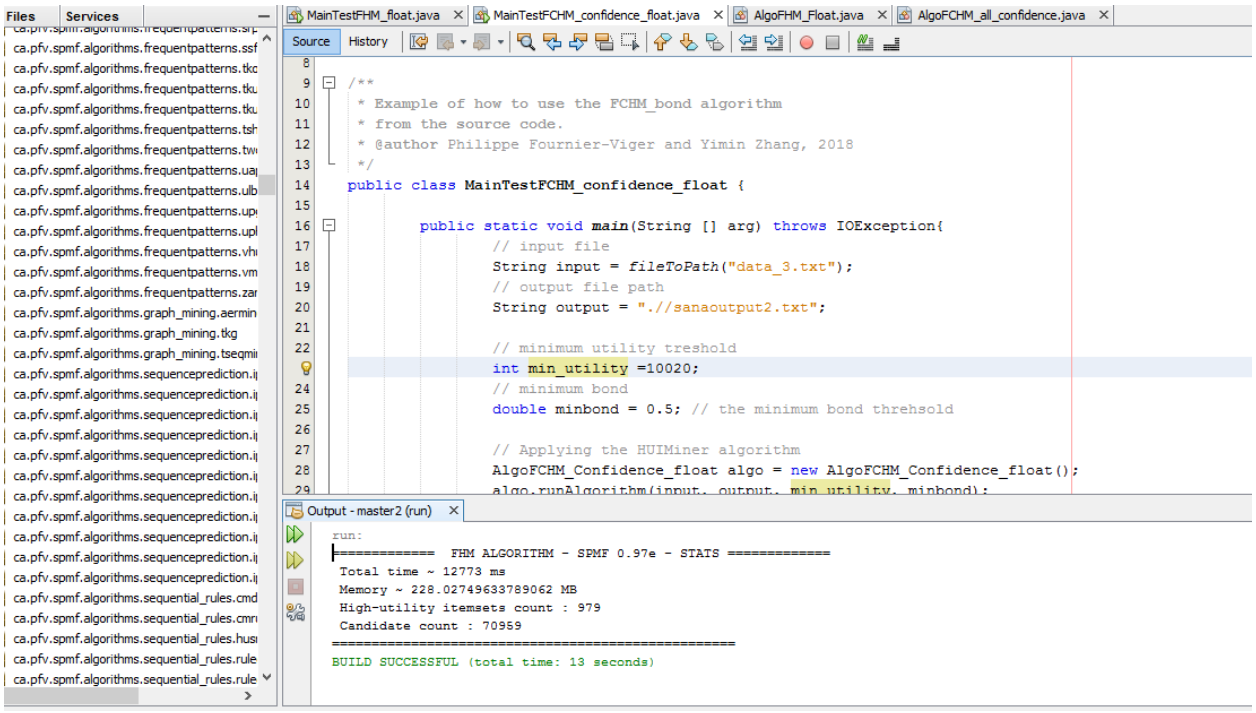

Figure 4.12. Exécution d'algorithme FCHM avec la base cosmétique

Les résultats obtenus par FCHM sont enregistrés dans un fichier texte des résultats. Le fichier résultant contient 979 itemset. Ces itemsets sont à la fois ayant une haute utilité et aussi sont corrélés.

Un extrait de ficher des résultats est présenté par la Figure 4.13. On voie que pour chaque motif, on a une information supplémentaire concernant sa mesure de corrélation. Par exemple, l'itemset {11817, 13510,13697,14747} est à haute utilité (11300) et corrélé à 100%.

```
13828 #UTIL: 13232.958 #ALLCONF: 1.0
12496 #UTIL: 13800.0 #ALLCONF: 1.0
6746 #UTIL: 11550.0 #ALLCONF: 1.0
2803 #UTIL: 10800.0 #ALLCONF: 1.0
11817 13510 13697 14747 #UTIL: 11300.0 #ALLCONF: 1.0
11817 13510 13697 14747 14428 #UTIL: 14300.0 #ALLCONF: 0.5
11817 13510 13697 14747 14428 11950 #UTIL: 16600.0 #ALLCONF: 0.5
11817 13510 13697 14747 11950 #UTIL: 13600.0 #ALLCONF: 0.5
11817 13510 13697 14428 #UTIL: 11300.0 #ALLCONF: 0.5
11817 13510 13697 14428 11950 #UTIL: 13600.0 #ALLCONF: 0.5
11817 13510 13697 11950 #UTIL: 10600.0 #ALLCONF: 0.5
11817 13510 14747 14428 #UTIL: 12100.0 #ALLCONF: 0.5
11817 13510 14747 14428 11950 #UTIL: 14400.0 #ALLCONF: 0.5
11817 13510 14747 11950 #UTIL: 11400.0 #ALLCONF: 0.5
11817 13510 14428 11950 #UTIL: 11400.0 #ALLCONF: 0.5
11817 13697 14747 14428 #UTIL: 12200.0 #ALLCONF: 0.5
11817 13697 14747 14428 11950 #UTIL: 14500.0 #ALLCONF: 0.5
11817 13697 14747 11950 #UTIL: 11500.0 #ALLCONF: 0.5
11817 13697 14428 11950 #UTIL: 11500.0 #ALLCONF: 0.5
11817 14747 14428 11950 #UTIL: 12300.0 #ALLCONF: 0.5
13510 13697 14747 14428 #UTIL: 10300.0 #ALLCONF: 0.5
13510 13697 14747 14428 11950 #UTIL: 12600.0 #ALLCONF: 0.5
13510 14747 14428 11950 #UTIL: 10400.0 #ALLCONF: 0.5
13697 14747 14428 11950 #UTIL: 10500.0 #ALLCONF: 0.5
13234 #UTIL: 16500.0 #ALLCONF: 1.0
15895 #UTIL: 12350.0 #ALLCONF: 1.0
9766 #UTIL: 10500.0 #ALLCONF: 1.0
13126 #UTIL: 10640.0 #ALLCONF: 1.0
11335 #UTIL: 10250.0 #ALLCONF: 1.0
7385 #UTIL: 12950.0 #ALLCONF: 1.0
10565 #UTIL: 13000.0 #ALLCONF: 1.0
13829 #UTIL: 10080.0 #ALLCONF: 1.0
```

Figure 4.13. Un extrait de fichier des résultats de FCHM avec minutil*=10020*

4.4.4.1 Les Résultats Obtenus :

Comme pour le cas de FHM, nous présentons les résultats d'évaluation de FCHM en termes de temps d'exécution, mémoire utilisation et nombre de motifs. Les résultats sont présentés par le Tableau 4-4 et visualisés par les Figures 4.14, 4.15 et 4.16. Notez que la valeur de *minconf* est

fixée à 0.5. Ceci signifie qu'on a besoin de découverts les itemsets qui ont une corrélation de 50% ou plus.

Tableau 4-4. Les résultats obtenus par l'algorithme FCHM avec la base de cosmétique.

Utility	**10010**	**10020**	**10030**	**10040**	**10050**	**10060**	**10070**	**10080**	**10090**	**10100**
Time (ms)	3589	3256	3234	3286	3203	3246	3256	3202	3257	3178
Memory (MB)	247,5	250, 5	250,8	250,7	248	248	250,7	248	250,7	247,9
High utility itemset Count	980	979	979	977	977	976	975	975	973	973

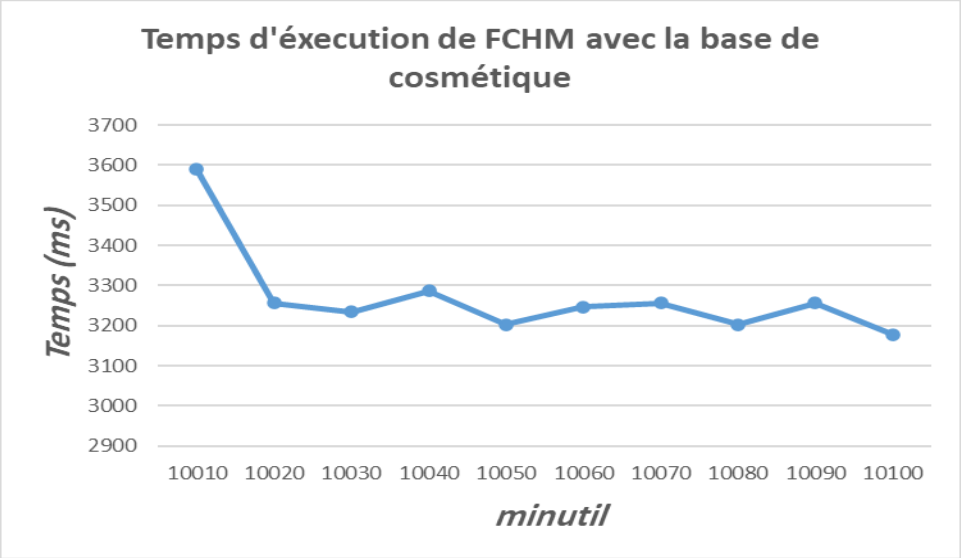

Figure 4.14. Temps d'exécution de FCHM avec la base de cosmétique

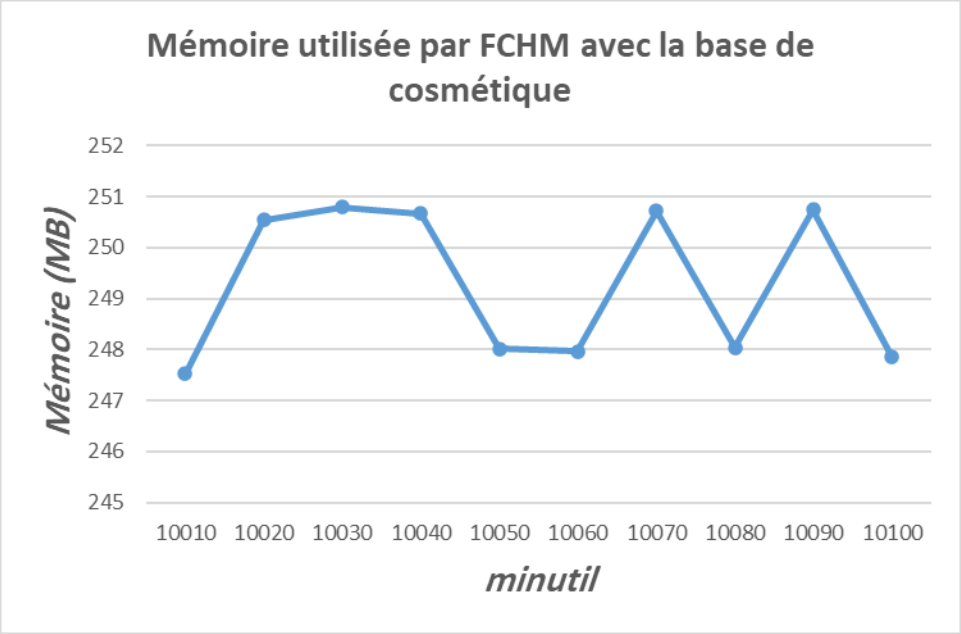

Figure 4.15. Mémoire utilisée par FCHM avec la base de cosmétique

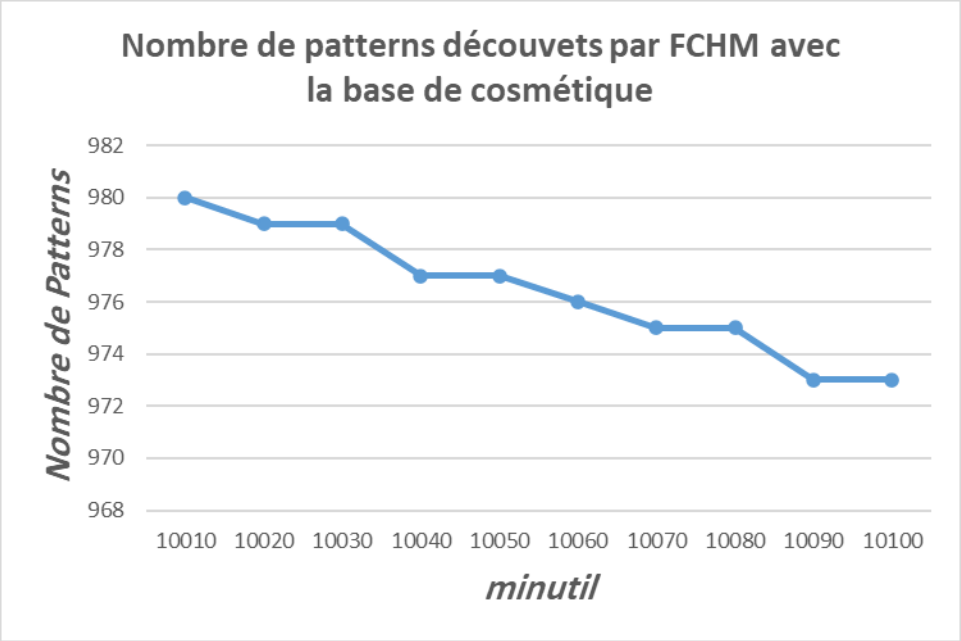

Figure 4.16. Nombre de patterns découverts par FCHM avec la base de cosmétique

Les résultats d'évaluations de FCHM sont un peu similaires au cas de FHM dans le fait que le temps d'exécution et le nombre de patterns diminuent avec l'augmentation de *minutil*. Par contre, on voie que, l'utilisation de mémoire est variée d'un cas à un autre.

4.4.4.2 Exemple de motifs découverts par FCHM

a) Exemple 3 (FCHM avec minutil 10020 et minconf=0.5)

Tableau 4-5. Motifs à haute utilité découverts par FCHM avec minutil=35050.

EDT Chanel allure perfume / EDT SI Nacre / COF Givenchy Live irresistible /EDT coco Chanel privee / EDT SI rouge / EDT Miss Dior EDP	Utility= 16600	Confidence=0.5
EDT Chanel allure perfume / EDT SI Nacre / EDT coco Chanel privee / EDT SI rouge / EDT Miss Dior EDP	Utility= 14400	Confidence=0.5
EDT Chanel allure perfume/ COF Givenchy Live irresistible /EDT coco Chanel privee / EDT SI rouge / EDT Miss Dior EDP	Utility= 14500	Confidence=0.5

Interprétation des résultats

Le tableau ci-dessous donne 3 motifs extraits du fichier de résultat. Après avoir analyser cet échantillon, nous avons constaté que le *parfum* est le produit le plus populaire et rentable par rapport les autres produits de cosmétiques, avec une préférence pour les marques *Céline, Chanel* et *Givenchy,* pour sa qualité exceptionnelle et son prix raisonnable. En plus, on voie que la majorité totale des items sont des parfums.

Ce qui signifie que les clients de parfum ont l'habitude d'acheter plus d'un parfum à la fois dans une seule transaction et les parfums achetées par un client n'ont pas de la même marque.

A partir de cette analyse, on peut proposer des stratégies de marketing tel que le lancement des promotions sur les parfums le plus achetées.

4.5 Le Deuxième Cas d'Etude (Base des Ventes d'une Boutique en Ligne)

4.5.1 Présentation de la Base de Vente d'une Boutique en Ligne

Le deuxième cas d'étude est fait sur une base de données d'un magasin de vente d'une boutique en ligne, on nomme cette base, *boutique en ligne [21]*.

La base boutique en ligne est une base réel contenant les transactions des clients du 01/12/2010 au 09/12/2011 d'une boutique en ligne basé à UK. Le nombre total de produits est 3468. Le nombre total de transactions est 14975. La base est publiée sur le site web de SPMF [21]. La base est téléchargeable via le site web de SPMF

Pour cette base, on n'a pas besoin d'une phase de pré-traitement car la base est déjà préparée et déposée sous format SPMF. Alors, on va directement tester cette base avec FHM est FCHM. Des aperçus sur la base de transaction et la liste de produits sont respectivement présentés par les Figures 4.17 et 4.18.

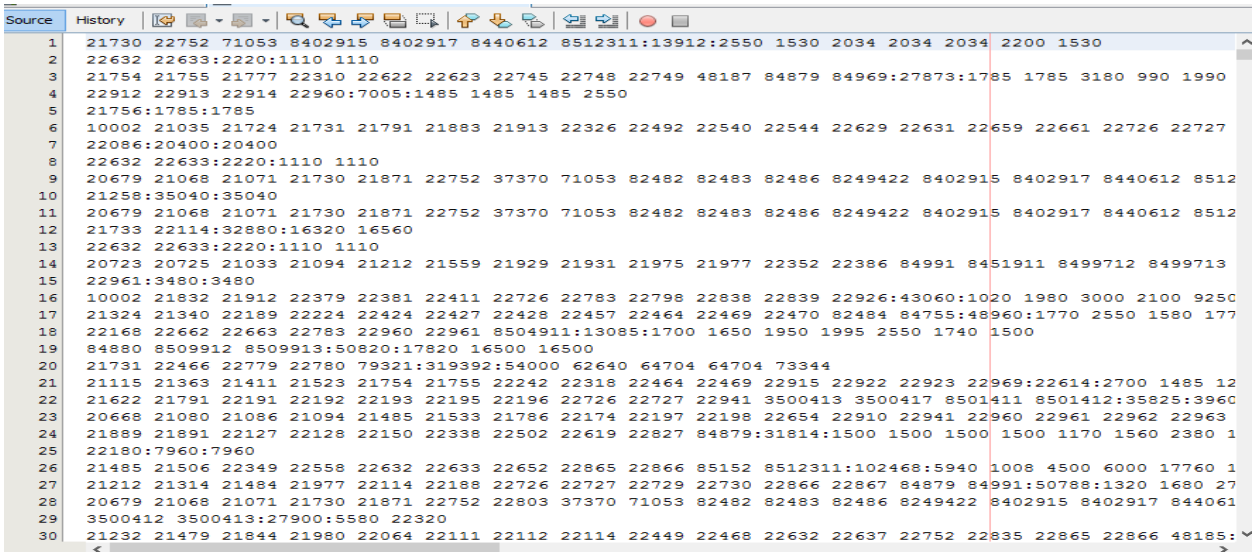

Figure 4.17. Un extrait du ficher SPMF de la base de boutique en ligne

```
@ITEM=90115=SUMMER BUTTERFLIES BAG CHARM
@ITEM=90114=SUMMER DAISIES BAG CHARM
@ITEM=90116=FRUIT SALAD BAG CHARM
@ITEM=90119=METALIC LEAVES BAG CHARMS
@ITEM=90118=PINK DAISY BAG CHARM
@ITEM=12=Adjust bad debt
@ITEM=14=Discount
@ITEM=6209612=PURPLE/TURQ FLOWERS HANDBAG
@ITEM=90131=PINK/AMETHYST/GOLD NECKLACE
@ITEM=8503443=3 GARDENIA MORRIS BOXED CANDLES
@ITEM=8249454=WOODEN FRAME ANTIQUE WHITE
@ITEM=90133=TEAL/FUSCHIA COL BEAD NECKLACE
@ITEM=6209611=PINK/YELLOW FLOWERS HANDBAG
@ITEM=90132=LIGHT TOPAZ TEAL/AQUA COL NECKLACE
@ITEM=8503444=3 WHITE CHOC MORRIS BOXED CANDLES
@ITEM=90135=ORANGE/WHT/FUSCHIA STONES NECKLACE
@ITEM=23=Manual
@ITEM=90134=OLD ROSE COMBO BEAD NECKLACE
@ITEM=90137=PINK COMBO MINI CRYSTALS NECKLACE
@ITEM=90136=PALE PINK/AMETHYST STONE NECKLACE
@ITEM=90138=WHITE/PINK MINI CRYSTALS NECKLACE
@ITEM=90140=PINK SWEETIE NECKLACE
@ITEM=29=SAMPLES
@ITEM=90143=SILVER BRACELET W PASTEL FLOWER
@ITEM=90145=SILVER HOOP EARRINGS WITH FLOWER
@ITEM=90144=SILVER DROP EARRINGS WITH FLOWER
@ITEM=90147=CHUNKY SILVER NECKLACE PASTEL FLOWE
@ITEM=90146=FINE SILVER NECKLACE W PASTEL FLOWE
@ITEM=90149=SILVER FLOWR PINK SHELL NECKLACE
@ITEM=90148=LONG SILVER NECKLACE PASTEL FLOWER
```

Figure 4.18. Un extrait du fichier contenant la liste des produits de la base boutique en ligne

4.5.2 Exécution de l'Algorithme FHM

Comme pour le premier cas d'étude, les algorithme FHM et FCHM sont testés sur la base de boutique en ligne. Le lecteur peut se référer au premier cas d'étude pour voir les détails à propos de l'exécution de ces algorithmes.

4.5.2.1 Les Résultats Obtenus

Les résultats d'exécution de FHM avec la base de boutique en ligne sont présentés dans le tableau suivant. Nous avons varié la valeur de *minutil* de *1000000* à *1009000* et nous enregistrons le temps d'exécution, l'utilisation de mémoire et le nombre des patterns.

Tableau 4-6. Les résultats obtenus par l'algorithme FHM avec la base de boutique en ligne.

Utility	**1000000**	**1001000**	**1002000**	**1003000**	**1004000**	**1005000**	**1006000**	**1007000**	**1008000**	**1009000**

Time (ms)	980	901	925	902	927	949	954	949	1420	1298
Memory (MB)	89,33	89,22	88,1	87,91	86,8	86,89	88,17	85,85	87,82	84,4
High utility itemset Count	15524	15292	15080	14899	14692	14465	14268	14100	13758	13551

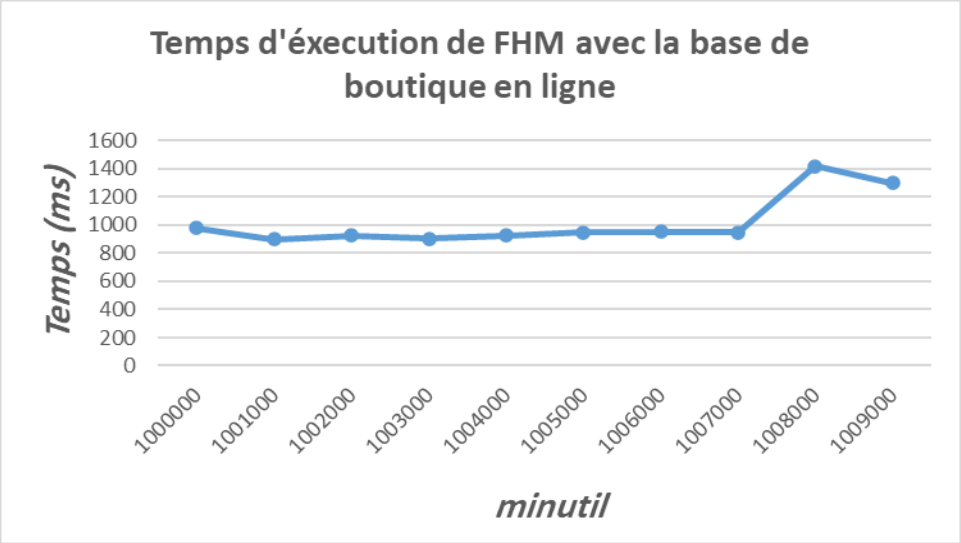

Figure 4.19. Temps d'exécution de FHM avec la base de boutique en ligne

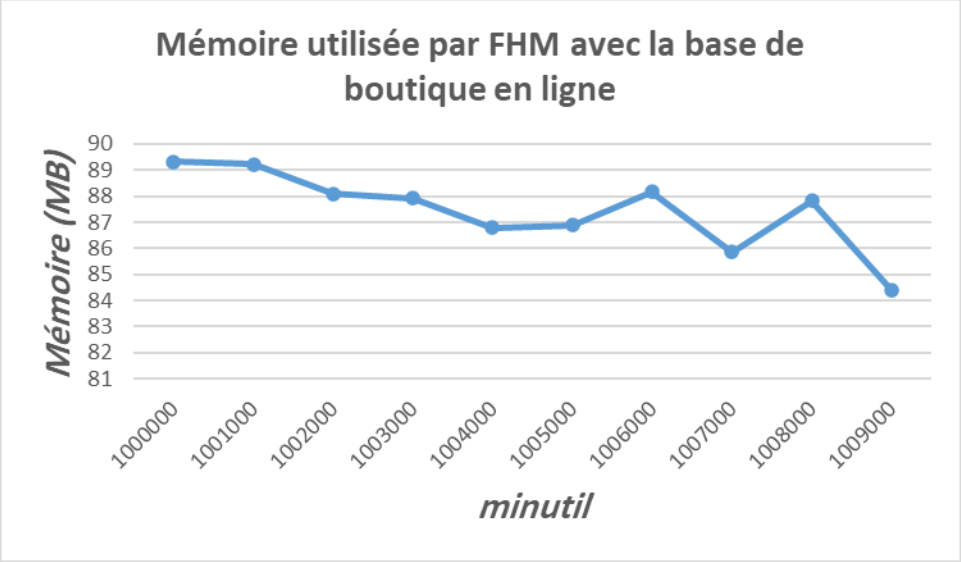

Figure 4.20. Mémoire utilisée par FHM avec la base de boutique en ligne

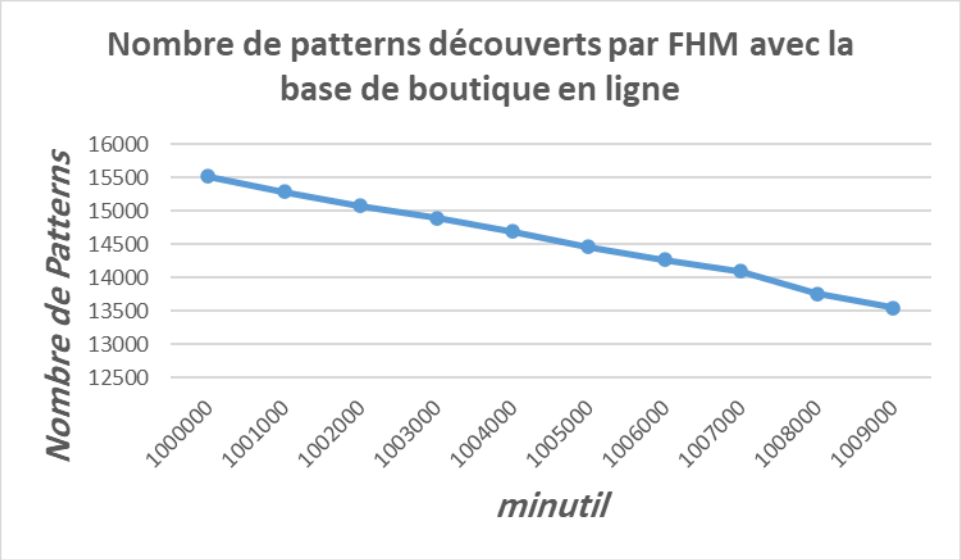

Figure 4.21. Nombre de patterns découverts par FHM avec la base de cosmétique

Nous remarquons que le temps d'exécution est stable avec les différentes valeurs de *minutil.* Nous remarquons aussi que le nombre des patterns obtenus dans cette base est largement grand par rapport au premier cas d'étude.

4.5.2.2 Exemple de motifs découverts par FHM

a) Exemple 4 (FHM avec minutil 1002000)

Tableau 4-7. Motifs à haute utilité découverts par FHM avec …

Sac géant design Space boy/ Sac jumbo hiboux /sac jumbo de fraises / Sac géant à pois roses	Utility=1034950
Sac géant design Space boy/ Sac jumbo hiboux / sac jumbo de fraises / Sac géant à pois roses / Sac jumbo à pois rouges rétrosport	Utility=1380190
Sac géant design Space boy/ Sac jumbo hiboux / sac jumbo de fraises /Sac jumbo à pois rouges retrosport	Utility=1199115
Sac géant design Space boy/ Sac jumbo hiboux / Sac géant à pois roses/ Sac jumbo à pois rouges rétrosport	Utility=1138840

Interprétation des résultats

Après l'analyse de cet exemple on trouve que : Ces sacs spacieux offrent une multitude d'utilisations et conviennent à différentes occasions, que ce soit pour le rangement à la maison, les voyages, les courses ou les cadeaux. Ils jouent un rôle essentiel dans le maintien de l'ordre et de

l'organisation, réduisant ainsi le désordre et facilitant la recherche des objets nécessaires. Leurs designs attrayants et leur grande capacité de stockage en font des options pratiques et esthétiques.

b) Exemple 5 (FHM avec minutil 1003000)

Tableau 4-8. Motifs à haute utilité découverts par FHM avec ...

tasse et soucoupe de style régence rose/ tasse et soucoupe de style régence vert/	Utility= 1569089
tasse et soucoupe de style régence rose/ tasse et soucoupe de style régence vert/ tasse et soucoupe de style régence aux roses/	Utility= 2047791
tasse et soucoupe de style régence rose/ tasse et soucoupe de style régence vert/ tasse et soucoupe de style régence aux roses/ Plateau à gâteaux de style régence à 3 étages	Utility= 2163200
tasse et soucoupe de style régence rose/ tasse et soucoupe de style régence vert/Plateau à gâteaux de style régence à 3 étages	Utility= 1908630

Interprétation des résultats

Ces produits, comprenant les ensembles de tasses à thé et soucoupes Regency avec des motifs de roses dans les couleurs verte et rose, offrent une expérience de dégustation de thé élégante et raffinée. Ils apportent une touche de beauté et d'élégance à chaque moment de thé. Fabriqués à partir de matériaux de haute qualité, ces ensembles sont à la fois durables et résistants. Ils conviennent parfaitement à une utilisation quotidienne ou à des occasions spéciales, ajoutant une note d'élégance à chaque table de thé. Profitez de moments agréables en dégustant votre thé préféré avec style et sophistication, donc ils achètent ensemble.

Autrement dit que, Les ustensiles sont vendus ensemble en fonction de la cohérence, des couleurs et de la forme, ainsi que de la même série.

4.5.3 Exécution de l'Algorithme FCHM

4.5.3.1 Les Résultats Obtenus

Tableau 4-9. Les résultats obtenus par l'algorithme FCHM avec la base de boutique en ligne.

Utility	**1000**	**2000**	**3000**	**4000**	**5000**	**6000**	**7000**	**8000**	**9000**	**10000**
Time (ms)	1937	1672	2093	1656	1609	1578	1766	1609	1703	2094
Memory (MB)	88,26	88, 45	88,35	88,25	88,24	87,25	87	86,03	86,02	85,79

High utility itemset Count	7837	7260	6713	6239	5817	5324	4783	4271	3744	3281

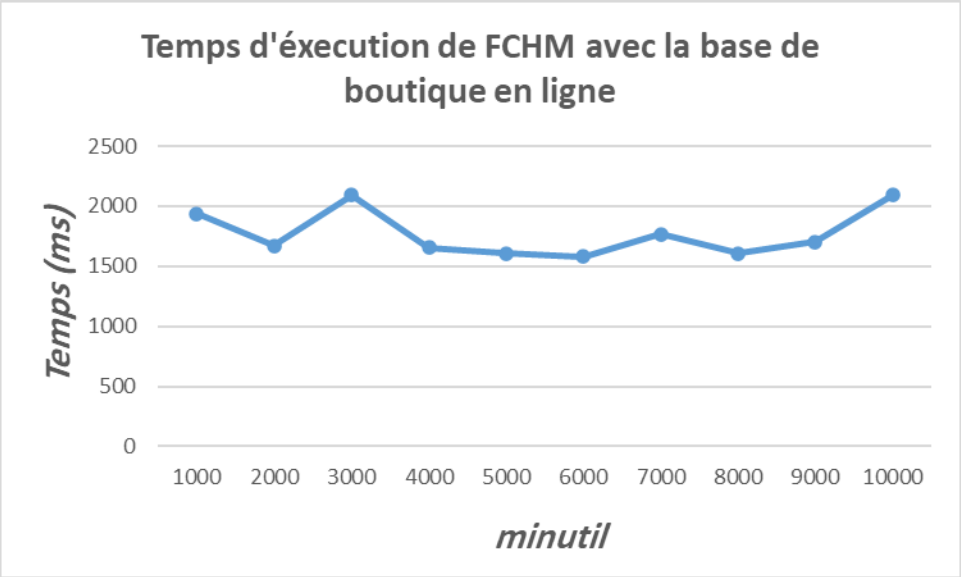

Figure 4.22. Temps d'exécution de FCHM avec la base de boutique en ligne

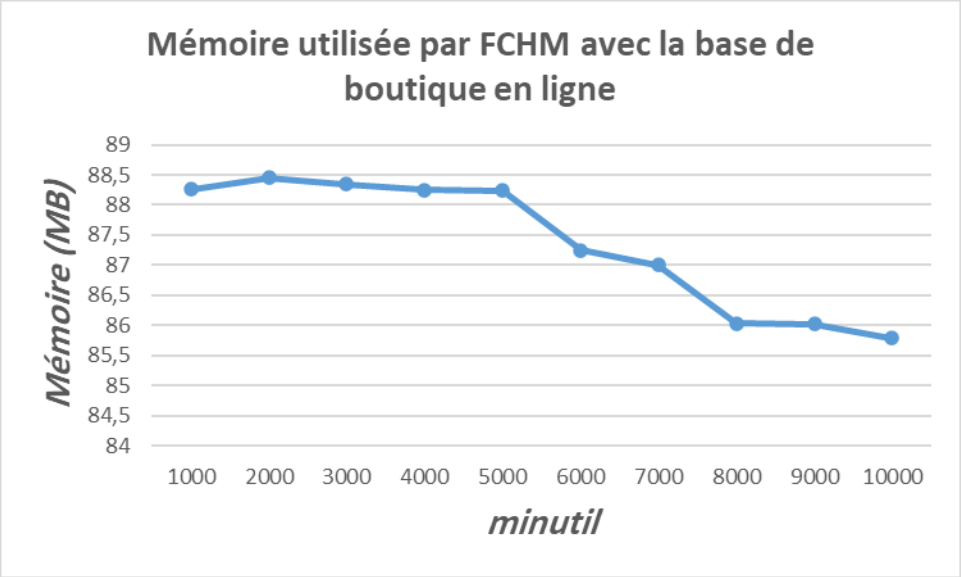

Figure 4.23. Mémoire utilisée par FCHM avec la base de boutique en ligne

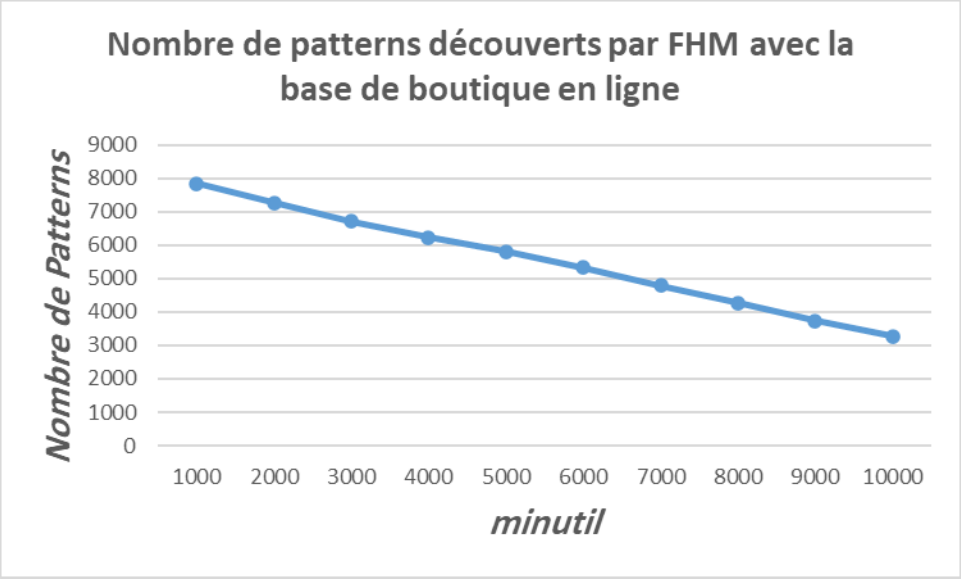

Figure 4.24. Nombre de patterns découverts par FCHM avec la base de boutique en ligne

4.5.3.2 Exemple de motifs découverts par FCHM

a) Exemple 7 (FHM avec minutil)

```
9017811 90062 9008113 9008212 9006014 #UTIL: 12180 #ALLCONF: 0.5
9017811 90062 9008113 9008212 9006014 9019511 #UTIL: 13970 #ALLCONF: 0.5
9017811 90062 9008113 9008212 9019511 #UTIL: 13375 #ALLCONF: 0.5
9017811 90062 9008113 9006014 #UTIL: 7100 #ALLCONF: 0.5
9017811 90062 9008113 9006014 9019511 #UTIL: 8890 #ALLCONF: 0.5
9017811 90062 9008113 901911 #UTIL: 8295 #ALLCONF: 0.5
9017811 90062 9008211 #UTIL: 7560 #ALLCONF: 0.5
9017811 90062 9008211 9008212 #UTIL: 12640 #ALLCONF: 0.5
9017811 90062 9008211 9008212 9006014 #UTIL: 13235 #ALLCONF: 0.5
9017811 90062 9008211 9008212 9006014 9019511 #UTIL: 15025 #ALLCONF: 0.5
9017811 90062 9008211 9008212 9019511 #UTIL: 14430 #ALLCONF: 0.5
9017811 90062 9008211 9006014 #UTIL: 8155 #ALLCONF: 0.5
9017811 90062 9008211 9006014 9019511 #UTIL: 9945 #ALLCONF: 0.5
9017811 90062 9008211 9019511 #UTIL: 9350 #ALLCONF: 0.5
9017811 90062 9008212 #UTIL: 10100 #ALLCONF: 0.5
9017811 90062 9008212 9006014 #UTIL: 10695 #ALLCONF: 0.5
9017811 90062 9008212 9006014 9019511 #UTIL: 12485 #ALLCONF: 0.5
9017811 90062 9008212 9019511 #UTIL: 11890 #ALLCONF: 0.5
9017811 90062 9006014 #UTIL: 5615 #ALLCONF: 0.5
9017811 90062 9006014 9019511 #UTIL: 7405 #ALLCONF: 0.5
9017811 90062 9019511 #UTIL: 6810 #ALLCONF: 0.5
9017811 9008113 9008211 #UTIL: 5220 #ALLCONF: 0.5
9017811 9008113 9008211 9008212 #UTIL: 10300 #ALLCONF: 0.5
9017811 9008113 9008211 9008212 9006014 #UTIL: 10895 #ALLCONF: 0.5
9017811 9008113 9008211 9008212 9006014 9019511 #UTIL: 12685 #ALLCONF: 0.5
9017811 9008113 9008211 9008212 9019511 #UTIL: 12090 #ALLCONF: 0.5
9017811 9008113 9008211 9006014 #UTIL: 5815 #ALLCONF: 0.5
9017811 9008113 9008211 9006014 9019511 #UTIL: 7605 #ALLCONF: 0.5
9017811 9008113 9008211 9019511 #UTIL: 7010 #ALLCONF: 0.5
9017811 9008113 9008212 #UTIL: 7760 #ALLCONF: 0.5
9017811 9008113 9008212 9006014 #UTIL: 8355 #ALLCONF: 0.5
9017811 9008113 9008212 9006014 9019511 #UTIL: 10145 #ALLCONF: 0.5
9017811 9008113 9008212 9019511 #UTIL: 9550 #ALLCONF: 0.5
```

Figure 4.25: Résultats obtenus d'algorithme FCHM pour l'utilité minimum égale 4000

Tableau 4-10. Motifs à haute utilité découverts par FCHM avec

Ambre chunky /Bracelet de carnaval / broche de lys olive	Utility = 6505
Ambre chunky / Bracelet de carnaval /broche de nœud vert	Utility = 7560

Ambre chunky / Bracelet de carnaval /collier en verre poli au feu bronze	Utility = 5615
Ambre chunky / Bracelet de carnaval / collier en verre poli au feu bronze/ bracelet en pierre violet	Utility = 7405

Interprétation des résultats

Après l'analyse de ces données, nous avons trouvé qu'il y a une relation entre ces éléments, de sorte que tous ces matériaux sont des choses qui contribuent à la fabrication d'accessoires,

Par conséquent, nous concluons que les artisans qui fabriquent des accessoires, s'ils achètent du fil de broderie, ils prennent également des pierres précieuses avec eux.

b) Exemple 8 (FHM avec minutil)

Tableau 4-11. Motifs à haute utilité découverts par FCHM

Nombre de tuiles Cottage Garden 3, Nombre de tuiles Cottage Garden 9	Utility=14140
Nombre de tuiles Cottage Garden 3, Nombre de tuiles Cottage Garden 4	Utility=7020
Nombre de tuiles Cottage Garden 3 Nombre de tuiles Cottage Garden 4, Nombre de tuiles Cottage Garden 8	Utility=10530
Nombre de tuiles Cottage Garden 3, Nombre de tuiles Cottage Garden 8	Utility=7020

Interprétation des résultats

Nombre de tuiles Cottage Garden 3= Tuiles de gazon artificiel de jeu.

Nombre de tuiles Cottage Garden 4= Tuiles en bois pour le jeu.

Nombre de tuiles Cottage Garden 8= Les tuiles de pont sont spécifiques au jeu.

Nombre de tuiles Cottage Garden 9= Carrelage de jeu.

Cottage Garden a pour objectif de remplir son jardin de manière optimale en plantant des fleurs et des légumes tout en évitant les espaces vides. Les participants doivent planifier leur jardin avec soin afin de maximiser leur score en obtenant des points pour les plantations de fleurs et de légumes, ainsi que pour les pots de fleurs et les plates-bandes complètes.

Donc les joueurs achètent beaucoup la pièce de « Tuiles de gazon artificiel » parce qu'ils offrent des avantages stratégiques particuliers dans le jeu.

4.6 Conclusion

Dans ce chapitre, nous avons présenté les résultats de l'application des deux algorithmes FHM et FCHM dans des bases de données réelles. Une évaluation des résultats obtenus est faite d'abord en utilisant trois critères : le temps d'exécution, mémoire utilisée et le nombre de motifs découverts. Ensuite, nous avons effectué une analyse sur les résultats des motifs résultants en donnant plusieurs exemples avec leurs interprétations.

Chapitre 5: Conclusion Générale

La fouille de motifs est une discipline qui consiste à extraire des connaissances utiles à partir des grandes quantités de données sous forme de motifs significatifs. Dans cette étude, nous avons utilisé deux algorithmes de la fouille de motifs FHM et FCHM pour extraire deux types de motifs dans des base données réels.

Nous avons constaté que : Les bases de données réelles présentent généralement des défis supplémentaires par rapport aux bases de données artificielles lors de leur utilisation avec les algorithmes de l'extraction des motifs. Deux principales difficultés associées aux bases de données réelles :

(1) Volume des données : Les bases de données réelles contiennent souvent des volumes massifs de données. Par exemple, après le pré-traitement de la base donnée de cosmétiques, le nombre total de transactions est plus de 170000 transactions avec de 16000 produits (item).

(2) La nature des données : À cause du grand nombre d'items dans les bases de données réelles, ces bases de données sont généralement éparses, en anglais *sparse datasets*. Par exemple, nous pouvons voir que pour la base de cométique contient plus de 16000 produits. Ce grand nombre d'items fait que la fréquence d'apparition de n'importe quels deux items dans la même transaction est petite. En d'autres termes, la corrélation entre les items est extrêmement faible. Suite à cela, l'application des algorithmes de fouille des motifs parfois ne donnent pas des motifs importants comme le cas de l'exemple 2.

Pour les travaux futurs, nous nous intéressons à la découverte des autres types de motifs à haute utilité tel que les motifs séquentiels à haute utilité et le teste de ces algorithmes dans des bases plus complexes.

Le choix de l'algorithme dépendra de la taille de la base de données et de la complexité des itemsets recherchés. FHM est plus adapté aux bases de données de taille moyenne à grande, Tandis que FCHM est plus adapté aux bases de données de petite à moyenne taille.

Ces algorithmes ont l'avantage d'être rapides et efficaces ils peuvent être utilisés dans divers domaines tels que le marketing, la santé, la finance, etc. pour identifier des modèles.

L'extraction d'itemsets fréquents peut être utilisée dans de nombreux domaines, tels que la recommandation de produits, la détection de fraudes, la segmentation de clients, etc. Elle permet de découvrir des relations intéressantes entre les différents éléments de la base de données.

Cette recherche contribue à l'amélioration de la compréhension des techniques d'extraction d'itemsets fréquents et leur application dans divers domaines.

Références

[1] D. Mokeddem, "Fouille de Données et Médias Sociaux," 2022.

[2] M. NEMICHE, "Data mining. ," Faculté des Sciences d Agadir, 2014.

[3] K. Thearling, "An introduction to data mining," *Direct Marketing Magazine*, pp. 28-31, 1999.

[4] M. l. Merrouche , & Miloudi , T. , "Mémoire de fin d'étude. ," Master, Université de B B Arreridj, 2013.

[5] T. Belkaaloul, & Safi,, "Mémoire de fin d'études.," Master, Université de B B Arreridj, 2018.

[6] B. Houmadi, "Étude exploratoire d'outils pour le Data Mining," Université du Québec à Trois-Rivières, 2007.

[7] A. Konate, "Un aperçu sur quelques méthodes en apprentissage automatique supervisé," 2018.

[8] L. DE MATTEIS, "Introduction à l'apprentissage automatique."

[9] L. Benahcene , & Belaifa,, "Mémoire de fin d'étude. ," Master, University de B B Arreridj 2021.

[10] T. MENOUER, & Dermouche, M, "Mémoire de fin d'études. ," 2010.

[11] P. Fournier‐Viger *et al.*, "A survey of itemset mining," vol. 7, no. 4, p. e1207, 2017.

[12] P. Fournier-Viger, J. Chun-Wei Lin, T. Truong-Chi, and R. Nkambou, "A survey of high utility itemset mining," *High-utility pattern mining: Theory, algorithms and applications*, pp. 1-45, 2019.

[13] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, "FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning," in *Foundations of Intelligent Systems: 21st International Symposium, ISMIS 2014, Roskilde, Denmark, June 25-27, 2014. Proceedings 21*, 2014: Springer, pp. 83-92.

[14] Y. Liu, W.-k. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," in *Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings 9*, 2005: Springer, pp. 689-695.

[15] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 55-64.

[16] P. Fournier-Viger, J. C.-W. Lin, Q.-H. Duong, and T.-L. Dam, "FHM: Faster high-utility itemset mining using length upper-bound reduction," in *Trends in Applied Knowledge-Based Systems and Data Science: 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings*, 2016: Springer, pp. 115-127.

[17] Q.-H. Duong, P. Fournier-Viger, H. Ramampiaro, K. Nørvåg, and T.-L. Dam, "Efficient high utility itemset mining using buffered utility-lists," *Applied Intelligence*, vol. 48, pp. 1859-1877, 2018.

[18] P. Fournier-Viger, Y. Zhang, J. C.-W. Lin, D.-T. Dinh, and H. Bac Le, "Mining correlated high-utility itemsets using various measures," *Logic Journal of the IGPL*, vol. 28, no. 1, pp. 19-32, 2020.

[19] P. Fournier-Viger, J. C.-W. Lin, T. Dinh, and H. B. Le, "Mining correlated high-utility itemsets using the bond measure," in *Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings 11*, 2016: Springer, pp. 53-65.

[20] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng, "Spmf: a java open-source pattern mining library," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3389-3393, 2014.

[21] P. Fournier-Viger, Y. Zhang, J. C.-W. Lin, H. Fujita, and Y. S. Koh, "Mining local and peak high utility itemsets," *Information Sciences*, vol. 481, pp. 344-367, 2019.