République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de 20 août 1955 Skikda

Faculté des sciences

Département d'Informatique

Mémoire de Fin d'Études

Pour l'obtention du diplôme de Master en Informatique

Option:Systèmes d'informations avancés et applications -SIIA-

Thème

Extraction des règles d'association à partir d'une base de données en utilisant l'algorithme Apriori

Réalisé par :

- M^me^Benmerabet Sihem
- M^me^Boughiout Hakima

Encadré par:

Pr:Mazouzi Smaine

Soutenu publiquement le : 02/07/2023 devant le jury composé de

Mr. Bouhouche	Université de Skikda	Président
Mm. Bendana	Université de Skikda	Examinatrice

Année Universitaire : 2022/2023

Remerciements

الحمد لله الذي وفقنا و أعاننا على إكمال هذا العمل

Nous tenons à remercier d'abord notre encadreur
le professeur **M.** ***MazouziSmaine*** pour son aide, ses conseils, et ses remarques objectives qui ont contribué à la réalisation de ce mémoire.

Nos sincères gratitudes aux enseignants *du*
département d'Informatique.

Nous n'oublierons pas de remercier **nos parents** pour leur contribution, leur soutien, leur patience et leur encouragement morale.

Dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire.

Dédicaces

Je dédie ce travail à ma famille particulièrement à **mes très chers parents** qui m'encouragent toujours.

À **mon mari**, **mes enfants Oussama abdRahmen , Mohamed Nadjib ,Rouaya ,Oumaima.**

À **mes frère**, **mes sœurs**.

À **mes oncles et tantes.**

À toutes mes amies

Sihem

Dédicaces

Je dédie ce travail à ma famille particulièrement à **mes très chers parents** qui m'encouragent toujours.

À **mon mari**, **mes petites filles Douâa, Ritedj, Merieme, et Sarra.**

À **mon frère**, et **mes sœurs**.

À **mes oncles et tantes.**

À toutes mes amies

Hakima

ملخص

تتيح قواعد الارتباط المستخرجة من مجموعات البيانات الكبيرة تطبيقات متنوعة مثل التوصية في تطبيقات المبيعات ، بالإضافة إلى العديد من التطبيقات الأخرى. هذه واحدة من أهم عمليات المعالجة في مجال التنقيب عن البيانات، والغرض منها هو استخراج الأنماط المخفية في البيانات وكذلك العلاقات التي قد توجد هناك.

في أطروحة الماجستير هذه في نظم المعلومات المتقدمة والتطبيقات، درسنا التنقيب في البيانات وبشكل أكثر تحديدًا مشكلة استخراج قواعد الارتباط من قاعدة البيانات. بعد عرض تقديمي قصير لمجال تجميع البيانات، قمنا بدراسة وتقديم قواعد الارتباط والخوارزميات الرئيسية التي تسمح باستخراجها.

في نهاية هذه الدراسة ، اخترنا خوارزمية Apriori ، والتي قمنا على أساسها بتصميم وتنفيذ نظام لاستخراج قواعد الارتباط لتطبيق سلة التسوق.

من أجل تنفيذ واختبار النظام ، استخدمنا لغة Python في بيئة تطوير Google Colab ، والتي كانت ذات فائدة كبيرة لنا لسهولة استخدامها وأمانها.

الكلمات المفتاحية: التنقيب عن البيانات ، قواعد الرابطة ,Apriori ، التوصيات ، Python، Google Colab.

Résumé

Les règles d'association extraites de larges ensembles de données permettent diverses applications telles que la recommandation dans les applications de ventes, ainsi que plein d'autres applications. Il s'agit d'un des traitements les plus importants relevant du domaine data mining, dont le but et d'extraire des modèles enfouillés dans les données ainsi que les relations qui peuvent y exister.

Dans ce mémoire de master en Systèmes d'information avancés et applications, nous avons étudié le data mining et plus particulièrement le problème d'extraction de règles d'association au sein d'une base de données. Après une courte présentation du domaine du data mining, nous avons étudié et présenté les règles d'association et les principaux algorithmes permettant leur extraction.

Au terme de cette étude, nous avons opté pour l'algorithme Apriori, sur lequel nous avons modélisé et implémenter un système d'extraction de règles d'association pour l'application du panier de la ménagère.

Pour l'implémentation et le test du système, nous avons utilisé le langage Python sous l'environnement de développement Google Colab, qui nous a été d'une grande utilité pour sa convivialité et sa sécurité.

Mots-clés : Data mining, Règles d'association, Apriori, Recommandations, Python, Google Colab.

Abstract

Association rules extracted from large datasets enable various applications such as recommendation in sales applications, as well as many other applications. This is one of the most important processing in the data mining field, the purpose of which is to extract patterns within data as well as the relationships that may exist there.

In this master's thesis in Advanced Information Systems and Applications, we studied data mining and more particularly the problem of extracting association rules from a database. After a short presentation of the field of data mining, we studied and presented association rules and the main algorithms allowing their extraction.

At the end of this study, we opted for the Apriori algorithm, on which we modeled and implemented a system for extracting association rules for the application of the shopping basket.

For the implementation and testing of the system, we used the Python language under the Google Colab development environment, which has been of great use to us for its user-friendliness and security.

Keywords: Data mining, Association rules, Apriori, Recommendations, Python, Google Colab.

Sommaire

Liste des figures

Liste des Tableaux

Introduction générale

Introduction générale :

Les bases de données représentent la colonne vertébrale de tout système d'information de l'entreprise. C'est dans une base de données que les informations sont représentées et stuquées pour être utilisé pour toutes les activités de l'entreprise. Jusqu'à récemment, les bases de données sont exclusivement utilisées pour extraire les données nécessaires aux différentes tâches au sein de l'entreprise, et pour y stocker les informations qui en résultent. Cependant, et avec l'avènement de la fouille de données, communiquèrent appelée data mining, les bases de données ont trouvé une autre issue importante pour l'entreprise, à savoir l'extraction de nouvelles connaissance enfouie dans les données stuquées.

Le data minig est le processus d'analyse de grands ensembles de données pour identifier les modèles et les relations qui peuvent aider à résoudre les problèmes de l'entreprise grâce et ce en utilisant les techniques d'analyse des données. Ces techniques permettent aux entreprises de prévoir les tendances futures et de prendre des décisions commerciales plus éclairées. Il s'agit d'un domaine de première importance pour l'analyse de données dans son ensemble et l'une des disciplines fondamentales de la science des données, qui utilise des techniques d'analyse avancées pour trouver des informations utiles dans des ensembles de données. En plus de détail, le data minig est une étape du processus de découverte de connaissances dans les bases de données, une méthodologie de science des données pour la collecte, le traitement et l'analyse des données.

Dans ce mémoire de master, nous nous sommes intéressés à une tâche particulière du domaine du data minig à savoir l'extraction des règles d'association à partir des données stuquées dans une base de données, typiquement relationnelle. L'extraction des règles d'association consiste à trouver des associations et des relations intéressantes parmi de grands ensembles d'éléments de données, dits items. Ces règles indiquent la fréquence à laquelle un ensemble d'éléments se produit dans une transaction. Un exemple typique est une analyse du panier de la ménagère.

L'analyse du panier de la ménagère est l'une des principales applications d'extraction des relations pour montrer les associations entre les items (articles). Elle permet aux gestionnaires de ventes d'identifier les relations entre les articles que les gens achètent fréquemment et ensemble .Étant donné un ensemble de transactions, nous pouvons trouver des règles qui prédiront l'occurrence d'un item en fonction des occurrences d'autres éléments dans l'ensemble des transactions.

Notre travail consister à survoler les algorithmes d'extraction des règles d'association, puis de choisir le plus prometteur pour l'implémenter et fonctionner avec les données et les transactions issues d'une base de données relationnelle. Après cette étude, nous avons opté pour l'algorithme Apriori, largement utilisé pour l'extraction des règles d'association. Ces règles nous permettent d'associer les items dont il a une relation de cooccurrence mutuelle.

Notre mémoire est organisé comme suit :

Au **chapitre 1**, nous présentons le domaine du data mining, ses principes, ses techniques, et ses applications.

Au **chapitre 2**, nous abordons les règles d'association en montrant des exemples, et en présentant les principaux algorithmes de calcul de ses règles, dont l'algorithme Apriori, largement utilisé dans ce domaine.

Le **chapitre 3** est consacré à la conception de notre système, en unissant le langage de modélisation UML.

Le **chapitre 4** présente l'implémentation de notre système d'extraction des règles d'association et de recommandations, et ce en utilisant le langage Python sous l'environnement de développement en ligne Google Colab.

En **conclusion générale**, nous sommes revenues sur notre travail, notre contribution, et nous avons souligné quelques perspectives.

Chapitre 1

Data mining

1. 1 Introduction

Dans ce chapitre, nous survolons le domaine du data mining (fouille de données), en montrant son importance dans les différents domaines des sciences de données et plus particulièrement les bases de données et les systèmes d'information. En effet, ces dernières années, les outils traditionnels d'analyse statistique ont eu des difficultés à gérer l'énorme volume de données collectées. En outre, les méthodes statistiques nécessitent une connaissance plus large des données afin de définir les principales hypothèses pour l'analyse. Ce qui rend l'analyse plus coûteuse en termes de temps et de stockage cela a motivé le développement d'un nouveau champ de recherche appelé fouille de données ou l'Extraction de Connaissances dans les bases de données (ECD). Ce champ est issu des bases de données, des statistiques, et de l'intelligence artificielle **[1]**.

1.2Contexte historique du data mining

Le data mining a ses racines dans les années 1960 et 1970, lorsque les entreprises ont commencé à utiliser des systèmes de gestion de bases de données pour stocker et gérer leurs données. Le concept apparaît en 1989 sous un premier nom de KDD (Knowledge Discovery in Data bases, en français ECD pour Extraction de Connaissances à partir des Données), l'augmentation de la puissance de traitement informatique a permis aux chercheurs de développer des algorithmes de data mining plus sophistiqués.

Avant qu'en 1991 apparaisse pour la première fois le terme de Data Mining ou « fouille des données ». Comme l'expliquent fort bien Michael Berry et Gordon Lin off, ce concept – tel qu'on l'entend aujourd'hui, et surtout tel qu'on l'applique dans les services marketing – est étroitement lié au concept du « one-to-one relation ship ». C'est à dire la personnalisation des rapports entre l'entreprise et sa clientèle **[3].**

Au cours de ces années, le data mining est devenu un domaine de recherche et de pratique à part entière, avec l'émergence de logiciels spécialisés dans ce domaine. Les entreprises ont commencé à reconnaître l'importance de l'analyse de données pour améliorer leur performance et leur compétitivité, et ont commencé à investir dans des technologies de data mining.

Au fil des ans, le data mining est devenu de plus en plus sophistiqué, avec l'introduction de nouvelles techniques telles que l'apprentissage automatique, la détection de fraudes et l'analyse des réseaux sociaux. Avec la prolifération des données numériques à grande échelle (Big data) au cours des dernières années, le data mining est devenu encore plus important pour les entreprises qui cherchent à exploiter cette richesse de données pour améliorer leurs performances et leur compétitivité **[19].**

1.3 Définition générale

Le data mining, ou fouille de données, est l'ensemble des méthodes et techniques destinées à l'exploration et l'analyse de bases de données informatiques (souvent grandes), de façon automatique ou semi-automatique, en vue de détecter dans ces données des règles, des associations, des tendances inconnues ou cachées, des structures particulières restituant l'essentiel de l'information utile tout en réduisant la quantité de données. L'idée de base de la fouille de données est d'extraire des informations implicites, précédemment inconnues et potentiellement utiles à partir d'ensemble de données. **[2]**, de point de vue utilisateur ces informations peuvent être des règles d'association, des concepts, des modèles décrit le processus de l'ECD comme un processus itératif semi-automatique constitué de plusieurs étapes allant de l'objectif d'extraction et la sélection des données jusqu'à la visualisation et l'interprétation des résultats, en passant par la phase de recherche de connaissances. Les différentes étapes de ce processus sont présentées dans la Figure 1.1 **[4]**.

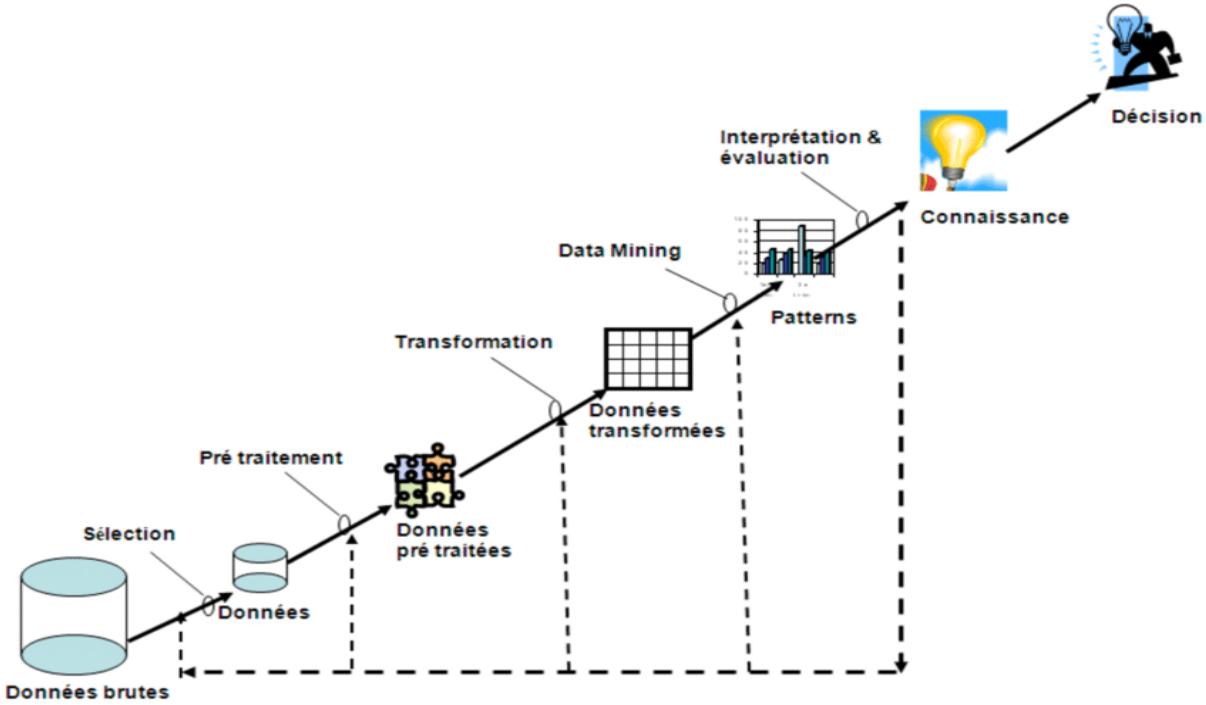

Figure1.1 : Processus de l'extraction de connaissance dans les bases de données

Les étapes principales du processus de l'ECD sont comme suit :

1.3.1 La sélection : Une fois l'objectif de l'extraction est fixé, l'étape de base est la sélection des échantillons significatifs de données. Toutes les données brutes ne sont pas nécessairement pertinentes pour une application des algorithmes de la fouille de données, il est nécessaire de sélectionner un sous-ensemble adapté à l'étude à mener et déterminer la structure générale des données ainsi que les règles utilisées en identifiant les informations exploitables et vérifier leur qualité et leur facilité d'accès.

1.3.2 Le Prétraitement : Les erreurs de saisie, les champs nuls, et les valeurs manquantes, impose généralement une phase de nettoyage de données, celle-ci a pour objectif de corriger ou de contourner l'inexactitude et les erreurs de données comme suit :

- Exclure les enregistrements incomplets.
- Identifier et traiter les valeurs manquantes, les valeurs erronées ou incertaines et les inconsistances.
- Remplacer les données manquantes.
- Prédire les valeurs (valeur moyenne des objets similaires, la régression).
- Utiliser l'absence de valeur comme une information.

1.3.3 Transformation : En vue d'appliquer un traitement spécifique aux données précédemment sélectionnées, il est nécessaire d'adapter leur structure dans un format approprié à la tâche de la fouille de données choisies dans la troisième étape.

1.3.4 Fouille de données : Dans cette phase, des méthodes intelligentes sont utilisées afin d'extraire les connaissances utiles à partir de données et les présenter sous une forme synthétique. Lors de cette étape plusieurs techniques peuvent être utilisées à savoir, le clustring, la classification, la régression, les règles d'association, etc.

1.3.5 Interprétation : Cette étape identifie les modèles intéressants représentant les connaissances, en se basant non seulement sur des mesures d'intérêt, mais aussi sur l'avis de l'expert. Cette évaluation prend généralement une forme graphique ou textuelle et contribue fortement à améliorer la lisibilité et la compréhension des résultats et facilite le partage de la connaissance. Les résultats produits par les algorithmes de fouille de données ne sont pas toujours exploitables directement. En effet, il est utile de définir des nouvelles mesures de qualité afin d'assister le décideur a utiliser les règles d'association les plus pertinentes **[2].**

1.4 Applications du data mining

1.4.1Business intelligence

La business intelligence est l'une des applications les plus courantes du data mining. Les entreprises utilisent le data mining pour extraire des informations à partir de leurs données pour améliorer leurs performances et leur prise de décision. Voici quelques exemples de la façon dont le data mining peut être utilisé pour améliorer la business intelligence :

- **Analyse de la clientèle** : les entreprises peuvent utiliser le data mining pour analyser les données des clients afin de comprendre les comportements d'achat, les préférences et les tendances. Les informations obtenues peuvent être utilisées pour créer des campagnes de marketing plus efficaces, personnaliser les offres pour chaque client et améliorer l'expérience client globale.
- **Détection de fraudes** : le data mining peut être utilisé pour détecter les fraudes dans les transactions financières. Les modèles de détection d'anomalies peuvent être utilisés pour identifier les transactions suspectes et réduire les pertes financières pour l'entreprise.
- **Prévisions de ventes** : le data mining peut être utilisé pour prédire les ventes futures en analysant les données historiques des ventes et en identifiant les

tendances saisonnières, les comportements des clients et les facteurs économiques.

- **Analyse des coûts** : le data mining peut être utilisé pour analyser les coûts de production et d'exploitation afin d'identifier les domaines où des économies peuvent être réalisées. Les informations obtenues peuvent être utilisées pour améliorer l'efficacité opérationnelle et réduire les coûts.

En utilisant le data mining pour améliorer la business intelligence, les entreprises peuvent prendre des décisions plus éclairées et améliorer leur performance globale.

1.4.2 Santé

Le data mining est également utilisé dans le domaine de la santé pour améliorer la qualité des soins, la recherche médicale et la gestion des données de santé.

Nous citons à titre d'exemple :

- **Prévision de l'évolution de maladies** : le data mining peut être utilisé pour prédire l'évolution de maladies chroniques telles que le diabète, les maladies cardiovasculaires et le cancer en analysant les données des patients et en identifiant les facteurs de risque et les tendances.
- **Détection d'anomalies médicales** : le data mining peut être utilisé pour détecter les anomalies dans les données médicales telles que les résultats de tests anormaux, les erreurs de prescription de médicaments et les résultats de diagnostic incorrects.
- **Recherche médicale** : le data mining peut être utilisé pour analyser les données des essais cliniques et des études épidémiologiques afin d'identifier les facteurs de risque, les traitements efficaces et les tendances de santé.
- **Gestion des données de santé** : le data mining peut être utilisé pour gérer les données de santé des patients en identifiant les tendances de santé, les traitements efficaces et les facteurs de risque pour améliorer les soins aux patients.

En utilisant le data mining dans le domaine de la santé, les professionnels de la santé peuvent prendre des décisions plus éclairées, améliorer les résultats pour les patients et contribuer à la recherche médicale.

1.4.3 E-commerce

Le data mining est largement utilisé dans le domaine de l'e-commerce pour améliorer la prise de décision, personnaliser les offres et améliorer l'expérience client.

Telque :

- **Recommandations de produits** : le data mining peut être utilisé pour recommander des produits aux clients en analysant les données d'achat et en identifiant les tendances et les préférences des clients.
- **Analyse de la concurrence** : le data mining peut être utilisé pour analyser les données de la concurrence, y compris les prix, les promotions et les stratégies marketing pour aider les entreprises à prendre des décisions éclairées.
- **Prévisions de demande** : le data mining peut être utilisé pour prédire la demande future en analysant les données historiques d'achat, les tendances saisonnières et les facteurs économiques.
- **Analyse des tendances du marché** : le data mining peut être utilisé pour analyser les tendances du marché en examinant les données de l'industrie, les rapports financiers et les indicateurs économiques.

En utilisant le data mining dans l'e-commerce, les entreprises peuvent améliorer la personnalisation des offres, la prise de décision et l'expérience client, ce qui peut se traduire par une augmentation des ventes et de la fidélité des clients.

1.4.4 Finance

Le data mining est largement utilisé dans le secteur financier pour améliorer la gestion des risques, la détection de fraudes et la prise de décisions éclairées. Voici quelques exemples d'applications du data mining dans le domaine de la finance :

- **Analyse du risque de crédit** : le data mining peut être utilisé pour prédire le risque de défaut de paiement en analysant les données des emprunteurs et en identifiant les facteurs de risque.
- **Détection de fraudes** : le data mining peut être utilisé pour détecter les fraudes dans les transactions financières en analysant les données des transactions et en identifiant les modèles de comportement suspects.
- **Prévisions de marché** : le data mining peut être utilisé pour prédire les tendances du marché et les mouvements des prix en analysant les données

historiques et en identifiant les tendances saisonnières et les facteurs économiques.

1.4.5 Télécommunications

Le data mining est utilisé dans l'industrie des télécommunications pour améliorer la qualité de service, la gestion de réseau et la personnalisation des offres.

Telque :

- **Analyse de la qualité de service** : le data mining peut être utilisé pour analyser les données de performance du réseau et identifier les goulots d'étranglement, les zones à faible couverture et les défaillances du réseau.
- **Personnalisation des offres** : le data mining peut être utilisé pour recommander des offres personnalisées aux clients en analysant les données d'utilisation et les préférences des clients.
- **Gestion de réseau** : le data mining peut être utilisé pour prédire les défaillances du réseau et les pannes en analysant les données de performance et en identifiant les tendances.

1.4.6 Détection de la fraude

Le data mining est utilisé dans divers secteurs pour détecter les fraudes. Tel que :

- **Détection de la fraude financière** : le data mining peut être utilisé pour détecter les fraudes dans les transactions financières en identifiant les modèles de comportement suspects.
- **Détection de la fraude dans les assurances** : le data mining peut être utilisé pour détecter les fraudes dans les demandes d'indemnisation en analysant les données des sinistres et en identifiant les modèles de comportement suspects.

1.4.7 Marketing et publicité

Le data mining est utilisé dans le marketing et la publicité pour améliorer la personnalisation des offres, la segmentation du marché et la mesure de l'efficacité des campagnes publicitaires. Telque :

- **Segmentation du marché** : le data mining peut être utilisé pour segmenter le marché en analysant les données démographiques, les comportements d'achat et les préférences des clients.
- **Personnalisation des offres** : le data mining peut être utilisé pour recommander des offres personnalisées aux clients en analysant les données d'achat et les préférences des clients.
- **Mesure de l'efficacité des campagnes publicitaires** : le data mining peut être utilisé pour mesurer l'efficacité des campagnes publicitaires en analysant les données d'interaction des clients.

1.4.8 Analyse des médias sociaux

Le data mining est utilisé dans l'analyse des médias sociaux pour comprendre les tendances, les opinions et les préférences des clients. Voici quelques exemples d'applications du data mining dans l'analyse des médias sociaux :

- **Analyse des sentiments** : le data mining peut être utilisé pour analyser les sentiments des clients envers une marque, un produit ou un service en analysant les données des médias sociaux.
- **Analyse des tendances** : le data mining peut être utilisé pour analyser les tendances des conversations des médias sociaux en identifiant les sujets chauds et les tendances émergentes.
- **Identification des influenceurs** : le data mining peut être utilisé pour identifier les influenceurs clés dans les médias sociaux en analysant les données d'interaction et les réseaux sociaux.

I.4.9 Gestion de la relation client

Le data mining est utilisé dans la gestion de la relation client pour améliorer la fidélité des clients et la satisfaction client. Telque :

- **Analyse de la rétention client** : le data mining peut être utilisé pour prédire la probabilité de départ des clients en analysant les données d'interaction et en identifiant les facteurs de risque.
- **Analyse de la satisfaction client** : le data mining peut être utilisé pour mesurer la satisfaction des clients en analysant les données d'interaction et en identifiant les problèmes et les préférences.

- **Personnalisation des offres** : le data mining peut être utilisé pour recommander des offres personnalisées aux clients en analysant les données d'achat et les préférences des clients **[19].**

1.5Objectifs du data mining

Les objectifs du data mining sont multiples. Les entreprises peuvent utiliser le data mining pour :

- **Détecter des modèles cachés** : Le data mining peut aider à détecter des modèles, des tendances et des relations cachées dans les données qui ne sont pas visibles à première vue.
- **Prédire les résultats futurs** : Le data mining peut être utilisé pour prédire les résultats futurs en se basant sur des données historiques. Par exemple, une entreprise de vente au détail peut utiliser le data mining pour prédire les ventes futures en fonction des données de vente passées.
- **Améliorer la prise de décision** : Le data mining peut aider les décideurs à prendre des décisions plus éclairées en fournissant des informations précises et fiables sur les données.
- **Optimiser les processus** : Le data mining peut aider à identifier les goulots d'étranglement, les inefficacités et les opportunités d'amélioration dans les processus métier.
- **Améliorer la qualité des produits et services** : Le data mining peut aider à identifier les problèmes de qualité des produits et services en examinant les données de satisfaction client et les données d'utilisation.
- **Détecter la fraude** : Le data mining peut être utilisé pour détecter les activités suspectes et les modèles de fraude dans les transactions financières.

Les objectifs du data mining sont d'extraire des connaissances et des informations précieuses à partir de grandes quantités de données pour aider les entreprises à prendre des décisions plus éclairées, à améliorer leur performance et à rester compétitives dans un environnement commercial en constante évolution**[19].**

1.6Techniques de data mining

Les techniques de data mining sont des méthodes d'analyse de données utilisées pour extraire des informations utiles à partir de grandes quantités de données. Ces techniques sont divisées en plusieurs catégories, notamment l'apprentissage

supervisé et non supervisé, l'extraction de règles d'association, le regroupement, la classification, la régression, la détection d'anomalies et les réseaux neuronaux. Chaque technique a ses propres avantages et inconvénients, et est adaptée à des problèmes de data mining spécifiques **[19].**

1.6.1 La catégorisation (Classification supervisée)

Dans l'apprentissage supervisé, on fournit à l'ordinateur des exemples d'entrées qui sont étiquetés avec les sorties souhaitées. Le but de cette méthode est que l'algorithme puisse 'apprendre' en comparant sa sortie réelle avec les sorties 'enseignées' pour trouver des erreurs et modifier le modèle en conséquence. L'apprentissage supervisé utilise donc des modèles pour prédire les valeurs d'étiquettes sur des données non étiquetées supplémentaires. Parmi les méthodes de classification supervisée, on peut citer : les arbres de décision, les réseaux neurones, la méthode des k plus proche voisins (KNN) ou la classification bayésienne **[5]**.

L'avantage de l'apprentissage supervisé est qu'il peut fournir des prédictions précises pour de nouvelles données. Cependant, il est important de disposer d'un ensemble de données d'entraînement représentatif pour que le modèle soit précis. De plus, l'apprentissage supervisé peut ne pas être adapté pour les situations où il n'y a pas suffisamment de données étiquetées disponibles **[19].**

1.6.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé ou le clustering, les données ne sont pas étiquetées à l'avance. L'idée étant que l'algorithme d'apprentissage est censé trouver tout seul des points communs parmi ses données d'entrée. Les données non étiquetées étant plus abondantes que les données étiquetées, les méthodes d'apprentissage automatique qui facilitent l'apprentissage non supervisé sont particulièrement utiles **[5]**.

L'objectif de l'apprentissage non supervisé est de découvrir des structures ou des relations intéressantes entre les données. Les exemples courants d'apprentissage non supervisé sont la classification non supervisée et le clustering.

La classification non supervisée consiste à trouver des structures sous-jacentes dans les données en créant des groupes de données similaires. Elle est utilisée pour segmenter les données en groupes homogènes, en fonction des caractéristiques communes entre les données. Par exemple, dans une analyse de marché, on pourrait utiliser la classification non supervisée pour regrouper les clients en segments homogènes en fonction de leurs comportements d'achat.

Le clustering, quant à lui, consiste à regrouper des données similaires dans des clusters. Les clusters sont des groupes de données qui ont des caractéristiques similaires. Cette technique peut être utilisée pour trouver des groupes de clients similaires ou pour regrouper des produits similaires **[19].**

L'avantage de l'apprentissage non supervisé est qu'il peut être utilisé pour découvrir des structures et des modèles cachés dans les données sans avoir besoin de données étiquetées. Cependant, les résultats peuvent être plus difficiles à interpréter et nécessitent souvent une analyse supplémentaire pour déterminer leur pertinence et leur utilité.

I.6.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé est une technique de data mining qui combine à la fois des données étiquetées et non étiquetées pour améliorer les performances du modèle. Cette approche est utile lorsque les données étiquetées sont rares ou coûteuses à obtenir.

L'objectif de l'apprentissage semi-supervisé est de tirer parti des données non étiquetées pour améliorer la précision du modèle. Le modèle est d'abord entraîné à partir des données étiquetées, puis il est ajusté à l'aide des données non étiquetées pour améliorer la précision des prédictions.

Les exemples courants d'apprentissage semi-supervisé sont la classification et la régression. Dans la classification semi-supervisée, le modèle utilise à la fois des données étiquetées et non étiquetées pour prédire la classe de nouvelles données. Dans la régression semi-supervisée, le modèle utilise à la fois des données étiquetées et non étiquetées pour prédire une variable de sortie continue.

L'avantage de l'apprentissage semi-supervisé est qu'il peut améliorer la précision du modèle en utilisant des données non étiquetées qui seraient autrement inutilisées. Cependant, il peut être plus complexe à mettre en œuvre et nécessiter des ressources supplémentaires pour l'annotation des données étiquetées **[19].**

1.6.4. Extraction de règles d'association

La recherche des règles d'association est l'un des sérieux problèmes du ECD. Le principe est de trouver des règles dans les données de types « si *Condition*, alors *Résultats*», notées *Conditions*→ *Résultats*. Cette technique permet la découverte de

règles intelligibles et exploitables dans un ensemble de données volumineux, règles exprimant des associations entre items ou attributs dans une base de données **[6]**.

L'extraction de règles d'association est une technique de data mining qui vise à découvrir des relations entre les éléments d'un ensemble de données. Cette technique est souvent utilisée dans les domaines du marketing, du commerce électronique et de l'analyse de panier d'achat.

L'objectif de l'extraction de règles d'association est de découvrir des règles du type « si A alors B », où A et B sont des ensembles d'éléments. Par exemple, si les clients achètent du pain, alors ils sont susceptibles d'acheter également du lait. Ces règles peuvent être utilisées pour prédire le comportement des clients, pour optimiser les promotions de vente croisée, ou pour améliorer l'organisation des produits en magasin.

La technique d'extraction de règles d'association repose sur l'utilisation de mesures telles que le support et la confiance pour évaluer l'importance des règles. Le support mesure la fréquence à laquelle un ensemble d'éléments apparaît dans l'ensemble de données, tandis que la confiance mesure la probabilité que B apparaisse lorsque A est présent.

Les algorithmes couramment utilisés pour l'extraction de règles d'association sont l'algorithme Apriori et l'algorithme FP-Growth. Ces algorithmes sont conçus pour explorer l'ensemble des règles possibles à partir de l'ensemble de données, en utilisant des techniques de prune pour éliminer les règles non pertinentes ou redondantes.

L'avantage de l'extraction de règles d'association est qu'elle peut révéler des relations cachées entre les données qui seraient autrement difficiles à détecter. Cependant, elle peut également générer un grand nombre de règles qui nécessitent une analyse et une évaluation supplémentaires pour déterminer leur pertinence et leur utilité.**[19]**

1.6.5. Regroupement (clustering)

Le regroupement, également connu sous le nom de clustering, est une technique de data mining qui vise à regrouper des données similaires en fonction de leurs caractéristiques communes. Cette technique est souvent utilisée dans le domaine de la segmentation de marché, de l'analyse de données géographiques et de la reconnaissance de formes.

L'objectif du regroupement est de diviser un ensemble de données en groupes homogènes, de sorte que les données dans chaque groupe soient similaires les unes aux autres, mais différentes des données dans les autres groupes. Les algorithmes de regroupement sont conçus pour trouver des groupes de données similaires en utilisant des mesures telles que la distance euclidienne ou la similarité cosinus.

Il existe plusieurs types de méthodes de regroupement, notamment le regroupement hiérarchique et le regroupement partitionné. Le regroupement hiérarchique crée une hiérarchie de groupes emboîtés, tandis que le regroupement partitionné divise l'ensemble de données en un nombre fixe de groupes.

Les algorithmes couramment utilisés pour le regroupement sont k-means, DBSCAN et hierarchial clustering. L'algorithme k-means divise l'ensemble de données en k clusters, où k est un nombre fixe spécifié à l'avance. L'algorithme DBSCAN utilise une approche basée sur la densité pour trouver des groupes, tandis que le regroupement hiérarchique utilise une approche basée sur la similarité pour créer une hiérarchie de groupes.

L'avantage du regroupement est qu'il peut être utilisé pour identifier des groupes de données similaires qui peuvent être utilisés pour prendre des décisions commerciales éclairées. Cependant, le choix de l'algorithme de regroupement approprié et la sélection des caractéristiques pertinentes sont des tâches importantes pour garantir la qualité des résultats de regroupement **[19].**

1.6.6. Classification

La classification est une technique de data mining qui consiste à prédire la classe d'un nouvel ensemble de données en utilisant un modèle préalablement entraîné. Cette technique est souvent utilisée dans les domaines de la reconnaissance de formes, de la détection de spam et de la prédiction de la probabilité de défaut de paiement.

L'objectif de la classification est de créer un modèle qui peut être utilisé pour prédire la classe de nouveaux ensembles de données. Les modèles de classification peuvent être créés à l'aide de techniques telles que l'arbre de décision, les réseaux de neurones artificiels, les machines à vecteurs de support et la régression logistique.

Les exemples courants de modèles de classification incluent les modèles de classification binaire, qui prédisent si un nouvel ensemble de données appartient à une

des deux classes possibles, et les modèles de classification multi classe, qui prédisent la classe d'un nouvel ensemble de données à partir de plusieurs classes possibles.

L'entraînement des modèles de classification implique l'utilisation d'un ensemble de données étiquetées pour apprendre à classer les données en fonction de leurs caractéristiques. Les mesures de performance telles que la précision, le rappel et la F-mesure sont utilisées pour évaluer la qualité du modèle de classification.

L'avantage de la classification est qu'elle peut être utilisée pour prédire la classe d'un nouvel ensemble de données avec une précision élevée. Cependant, le choix de la technique de classification appropriée et la sélection des caractéristiques pertinentes sont des tâches importantes pour garantir la qualité des résultats de classification **[4].**

1.6.7. Régression

La régression est une technique de data mining qui consiste à prédire une variable de sortie continue à partir de variables d'entrée. Cette technique est souvent utilisée dans les domaines de la finance, de l'économie et de la météorologie pour prédire des valeurs numériques telles que les prix des actions, les taux de croissance économique et les températures.

L'objectif de la régression est de créer un modèle qui peut être utilisé pour prédire la valeur d'une variable de sortie en fonction de ses relations avec les variables d'entrée. Les modèles de régression peuvent être créés à l'aide de techniques telles que la régression linéaire, la régression logistique et la régression polynomiale.

Les exemples courants de modèles de régression incluent les modèles de régression linéaire simple, qui prédisent une variable de sortie en fonction d'une seule variable d'entrée, et les modèles de régression linéaire multiple, qui prédisent une variable de sortie en fonction de plusieurs variables d'entrée.

L'entraînement des modèles de régression implique l'utilisation d'un ensemble de données étiquetées pour apprendre à prédire la valeur de la variable de sortie en fonction de ses relations avec les variables d'entrée. Les mesures de performance telles que l'erreur quadratique moyenne et le coefficient de détermination sont utilisées pour évaluer la qualité du modèle de régression.

L'avantage de la régression est qu'elle peut être utilisée pour prédire une valeur continue de la variable de sortie avec une précision élevée. Cependant, le choix de la technique de régression appropriée et la sélection des variables d'entrée pertinentes sont des tâches importantes pour garantir la qualité des résultats de régression **[19].**

1.6.8Détection d'anomalies

La détection d'anomalies, également connue sous le nom de détection de valeurs aberrantes, est une technique de data mining qui vise à identifier les valeurs atypiques dans un ensemble de données. Cette technique est souvent utilisée dans les domaines de la sécurité, de la finance et de la maintenance prédictive pour identifier les anomalies dans les transactions, les flux de données et les équipements.

L'objectif de la détection d'anomalies est de trouver des valeurs qui diffèrent considérablement des autres valeurs dans un ensemble de données. Les méthodes de détection d'anomalies comprennent l'approche statistique, l'approche basée sur la densité et l'approche basée sur l'apprentissage automatique.

Les exemples courants de techniques de détection d'anomalies incluent la méthode des distances localisées, qui mesure la distance entre les points de données, la méthode des k plus proches voisins, qui trouve les k voisins les plus proches de chaque point de données et la méthode des densités locales, qui identifie les régions à haute densité de points de données.

L'entraînement des modèles de détection d'anomalies implique l'utilisation d'un ensemble de données pour identifier les valeurs qui diffèrent considérablement des autres valeurs dans l'ensemble de données. Les mesures de performance telles que la sensibilité et la spécificité sont utilisées pour évaluer la qualité du modèle de détection d'anomalies.

L'avantage de la détection d'anomalies est qu'elle peut être utilisée pour identifier rapidement les valeurs qui diffèrent considérablement des autres valeurs dans un ensemble de données, permettant ainsi aux entreprises de prendre des mesures pour résoudre les problèmes potentiels. Cependant, le choix de la technique de détection d'anomalies appropriée et la sélection des caractéristiques pertinentes sont des tâches importantes pour garantir la qualité des résultats de détection d'anomalies **[19].**

1.6.9. Réseaux neuronaux

Les réseaux de neurones ont été développés comme un modèle mathématique générique afin de modéliser les neurones biologiques. Ils comportent un certain nombre d'éléments de traitement d'information appelés neurones **[7]**. Cette technique est souvent utilisée dans les domaines de la reconnaissance d'image, du traitement du langage naturel, de la classification et de la prédiction.

L'objectif des réseaux neuronaux est de créer un modèle qui peut apprendre à partir de données pour résoudre des problèmes de manière autonome. Les réseaux neuronaux sont constitués de couches de neurones interconnectés qui traitent les informations et les transmettent à travers le réseau pour produire une sortie.

Les exemples courants de modèles de réseaux neuronaux incluent les réseaux de neurones à propagation avant, qui transmettent les informations de la couche d'entrée à la couche de sortie, et les réseaux de neurones récurrents, qui permettent aux informations de circuler dans le réseau et de se connecter à des couches précédentes.

L'entraînement des modèles de réseaux neuronaux implique l'utilisation d'un ensemble de données pour apprendre les relations entre les entrées et les sorties. Les mesures de performance telles que la précision, le rappel et la F-mesure sont utilisées pour évaluer la qualité du modèle de réseaux neuronaux.

L'avantage des réseaux neuronaux est leur capacité à apprendre des structures de données complexes et à résoudre des problèmes qui seraient difficiles à résoudre par d'autres méthodes de data mining. Cependant, l'entraînement des modèles de réseaux neuronaux peut être coûteux en temps et en ressources, et le choix de la structure et de l'architecture du réseau neuronal est une tâche importante pour garantir la qualité des résultats.

En suivant ces étapes clés, les entreprises peuvent extraire des connaissances précieuses à partir de leurs données pour prendre des décisions éclairées et améliorer leurs performances **[19].**

1.7 .Défis du data mining

Le data mining peut être confronté à plusieurs défis qui peuvent entraver sa performance et son efficacité. Voici quelques-uns des principaux défis du data mining :

- **Qualité des données** : Les données utilisées pour l'analyse doivent être de bonnes qualités, fiables et précises pour que les résultats soient significatifs et utiles.
- **Confidentialité et sécurité** : L'utilisation des données personnelles pour l'analyse peut soulever des préoccupations en matière de confidentialité et de sécurité. Il est important de prendre des mesures pour protéger les données sensibles et garantir la confidentialité des individus.
- **Complexité des données** : Les données utilisées pour l'analyse peuvent être très complexes, comportant plusieurs variables et relations qui peuvent être difficiles à comprendre et à interpréter.
- **Scalabilité des algorithmes** : Les algorithmes de data mining peuvent être très gourmands en ressources, ce qui peut rendre difficile leur utilisation pour de grandes quantités de données.
- **Considérations éthiques** : L'utilisation des données pour l'analyse doit être effectuée de manière éthique et responsable, en tenant compte des droits et des intérêts des individus concernés. Les entreprises doivent être transparentes quant à l'utilisation de leurs données et s'assurer que les résultats de l'analyse sont utilisés de manière responsable **[19].**

1.8 Avenir du data mining

Le data mining est en constante évolution et de nouvelles tendances émergent continuellement. Voici quelques-unes des tendances émergentes dans le domaine du data mining :

- **Utilisation de l'apprentissage en profondeur (deep learning)** : le deep learning est une technique d'apprentissage automatique qui permet d'analyser des données non structurées telles que des images, des vidéos et du texte.
- **Utilisation de l'Internet des objets** : l'Internet des objets permet de collecter des données à partir de différents appareils connectés et de les analyser à l'aide du data mining pour obtenir des informations précieuses.
- **Utilisation de l'analyse prédictive** : l'analyse prédictive utilise des modèles statistiques pour prédire les résultats futurs à partir de données historiques.
- **Utilisation de la visualisation des données** : la visualisation des données est un outil important pour l'analyse des données, qui permet de représenter les données de manière graphique et intuitive.

Les avancées dans l'apprentissage automatique et l'intelligence artificielle permettent d'améliorer les performances du data mining. L'utilisation du big data permet également d'élargir les capacités du data mining et d'explorer de nouvelles applications dans diverses industries telles que la santé, la finance, l'e-commerce, les télécommunications et bien d'autres.

Cependant, avec ces avancées, il est important de considérer les implications éthiques et légales liées à l'utilisation des données personnelles et de s'assurer que le data mining est utilisé de manière responsable et éthique **[19].**

1.9 Conclusion

Dans ce chapitre nous avons passé en revue le domaine du data mining avec ses différentes applications. Il s'agit un outil précieux pour les entreprises et la société en général. Il permet de découvrir des modèles et des tendances cachés dans les données, ce qui peut conduire à des améliorations significatives dans divers domaines tels que la santé, les affaires, la finance, le marketing et bien d'autres.

Cependant, le data mining peut également être confronté à des défis tels que la qualité des données, les préoccupations en matière de confidentialité et de sécurité, la complexité des données, la scalabilité des algorithmes et les considérations éthiques. Pour que le data mining reste un outil utile et efficace, il est important que ces défis soient pris en compte et résolus.

Au final, le data mining est un domaine en constante évolution et ses perspectives futures sont passionnantes. L'utilisation de l'apprentissage en profondeur, de l'Internet des objets, de l'analyse prédictive et de la visualisation des données ouvre la voie à de nouvelles applications potentielles dans diverses industries. Cependant, il est crucial que ces avancées soient abordées avec une réflexion éthique et responsable pour garantir que les avantages du data mining soient réalisés tout en respectant les droits et les intérêts des individus.

Chapitre02

Règles d’association

2.1 Introduction

Le problème de l'extraction de règles d'association fut introduit par Agrawal et al. Ce problème, développé à l'origine pour l'analyse de bases de données de transactions de ventes, a pour but de découvrir des relations significatives entre les données de la base. Par exemple, une règle découverte dans les données de ventes dans un supermarché pourrait indiquer qu'un client achetant des hamburgers et des pommes de terre simultanément, serait susceptible d'acheter la mayonnaise. Une telle information peut être utilisée comme base pour prendre des décisions marketing telles que par exemple des promotions ou des emplacements bien choisis pour les produits associés.

2.2 Définition :

Une règle d'association est une application de la forme X → Y dans laquelle X et Y sont des ensembles des items.

* X est appelé condition ou l'antécédent.

* Y est appelé conclusion ou conséquent.

Voici quelques exemples de règles :

- Si un client achète du fromage alors il achète du pain (85%).
- Si maladie A et traitement B alors guérison (80%).
- Si un client achète thé et café il achète du sucre (99%).

À partir de deux variables X et Y, il est possible de construire huit règles différentes (voir Table 2.1)**[8]**.

$1: X \rightarrow Y$	$2: X \rightarrow \overline{Y}$	$3: \overline{X} \rightarrow Y$	$4: \overline{X} \rightarrow \overline{Y}$
$5: Y \rightarrow X$	$6: Y \rightarrow \overline{X}$	$7: \overline{Y} \rightarrow X$	$8: \overline{Y} \rightarrow \overline{X}$

Table 2.1: Règles d'association générées à partir de deux variables X et Y

Pour une règle X → Y, nous utilisons les notations suivantes :

• n = |T|, le nombre de transactions ou enregistrements.

• n_x = |X|, le nombre de transactions satisfaisant X.

• n_y = |Y|, le nombre de transactions satisfaisant Y.

• n_{xy} = |X \ Y|, le nombre de transactions satisfaisant à la fois X et Y.

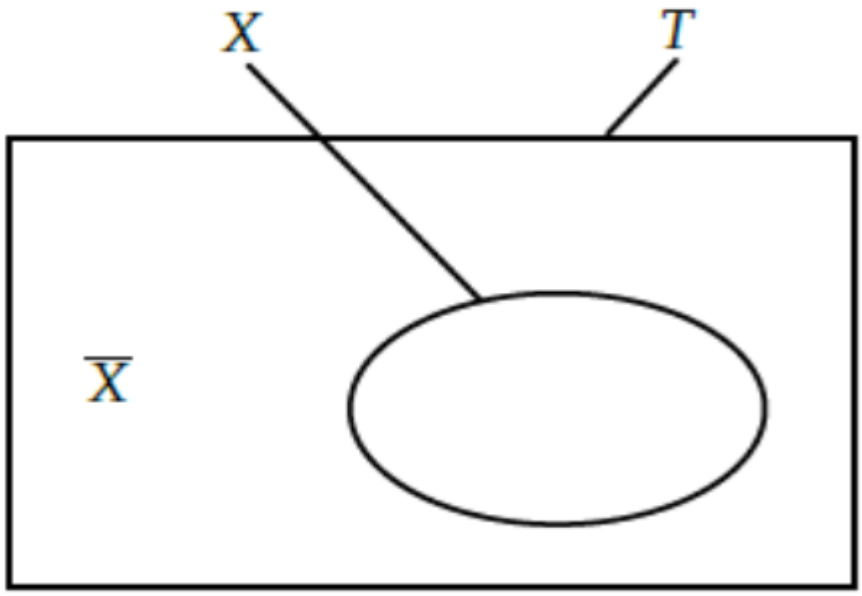

Figure 2.1 : Diagramme de Venn de l'ensemble X dans celui des transactions T

2.3 Terminologie:

Considérons un ensemble T de n transactions décrites par un ensemble I de variables

qualitatives ou quantitatives. T est stocké sous forme de table dans une base de données.

L'ensemble I= {i1, i2, . . . im} représente un ensemble de m attributs décrivant les n transactions de la base de données. Ces variables sont appelées des items dans la terminologie des règles d'association. L'ensemble T={t1,t2, . . . tn} représente un ensemble de transactions, une transaction étant un sous ensemble de I. Une conjonction

d'items Z= {i1, i2, . . . ik} non vide de I est appelé un itemset. Le nombre d'items k de l'ensemble Z constitue sa longueur. Un itemset de longueur k est nommé k-itemset.

Dans ce qui suit nous allons élucider notre travail avec un exemple d'une base de données qui comporte 10 transactions :

i1	i2	i3	i4
1	0	1	0
0	1	0	0
0	0	0	1
0	1	1	1
0	1	1	0
0	1	1	0
1	1	1	1
1	0	1	0
1	1	1	0
1	1	1	0

Table 2.2: Une base de données qui comporte 10 transactions

2.4 Quelques définitions :

2.4.1 Un Item :

est un objet, élément ou un article d'une base de données.

Exemple : i1 représente un item.

2.4.2 Un Itemset (motif) :

est un ensemble d'items, d'objets ou d'articles d'une base de données.

Exemple :{ i2, i3, i4}.

2.4.3 Un K-Itemset :

est un ensemble de k éléments, ou k-Items, il est aussi un Itemset.

Exemple 1 : {i1, i2, i3, i4} représente un 4-Itemset.

Exemple 2 : {i1, i2, i3} représente un 3-Itemset.

2.4.4 Contexte d'extraction de règles

Un contexte d'extraction des règles d'association est un triplet D= (O,I,R) dans lequel O et I sont respectivement des ensembles finis d'objets et d'items, et $T \subseteq O * I$ est une relation binaire entre les objets et les items. Un couple (o, i) $\subseteq$ T dénote le fait que l'objet o $\in$O est en relation avec l'item i $\in$I. Donc, un contexte d'extraction de taille m est une partie d'une base de données dans laquelle s'effectué le traitement.

2.4.5 Transaction

Une transaction est un itemset identifié par un identificateur unique Tid. L'ensemble de tous les identificateurs des transactions Tids sera désigné par l'ensemble T.

2.4.6 La fréquence d'un item ou d'un itemset

est le nombre de transactions ou d'identifiant unique qui contient au moins une fois l'item ou l'*itemset* en question.

2.4.7 Motif fréquent :

Un itemset est dit fréquent si son support est supérieur à un seuil défini à l'avance.

Dans notre exemple, en fixant le support minimum à 2 (ou 20% en relatif),

2.4.8 Superset :

Un superset est un itemset défini par rapport à un autre itemset,
par exemple: {i1, i2, i3} est un superset de {i1, i2}.

2.4.9 Itemset fermé (close ditemset) :

Un itemset fréquent est dit fermé si aucun de ses supersets n'a de support identique, autrement dit, tous ses supersets ont un support strictement plus faible.

Pour l'exemple ci-dessus, {i1, i3} est fermé car aucun de ses supersets n'a de support égal 5/10 Support ({i1, i2, i3})=3/10, Support ({i1, i3, i4})=1/10.

2.4.10 Maximal itemset :

Un itemset est dit maximal si aucun de ses supersets n'est fréquent.
Pour l'exemple ci-dessus, {i1, i2, i3} est maximal car son superset {i1, i2, i3,i4} (il n'y en a qu'un) n'est pas fréquent avec un support de 1/10.

2.4.11Generatoritemset :

Un itemset est générateur si tous ses sous-itemsets ont un support strictement supérieur **[20].**

Dans notre exemple :

{i1, i2, i3} de support 4/10 n'est pas générateur puisqu'on trouve {i1, i2} avec un support identique.

En revanche, {i2, i4}, de support 2/10, est générateur car Support ({i2}) = 7/10 et Support ({i4}) = 3/10.

2.5 Support d'une règle d'association

Le support d'une règle d'association est le nombre de transactions dans T contenant ce motif divisé par le cardinal de T, nommé n. Un seuil minimal de support minSup est fixé à partir duquel un ensemble d'items est dit fréquent.

$$support(X \rightarrow Y) = \frac{n_{xy}}{n}$$

ou Support (X →Y) Nombre de fois où X et Y apparaissent ensemble dans les transactions T.

le support prend sa valeur dans l'intervalle [0,1].

Par exemple, le support de {i1} est égal à 5/10.

Support({i1,i2})=3/10=0.3=30%

2.6 La confiance d'une règle d'association

La confiance évalue la validité d'une règle et permet de réduire le nombre de règles à l'aide d'un élagage en fonction d'un seuil minConf. Cependant, cette mesure ne permet pas de détecter l'indépendance des variables. Elle estime la probabilité conditionnelle que la variable Y soit réalisée sachant que la variable X l'est.

$$confiance(X \rightarrow Y) = \frac{n_{xy}}{n_x}$$

La confiance prend sa valeur dans l'intervalle [0,1].

Donc pour confiance ({i1,i2})= 3/5=0.6=60%.

Le support et la confiance sont deux indices élémentaires, mais ils constituent les mesures les plus communément utilisées pour évaluer les règles d'association. Malheureusement, ces indices présentent certaines limites : l'énorme quantité de règles d'association limite l'utilité de la technique et la trivialité d'une grande partie des règles générées implique une réelle nécessité de proposer de nouvelles mesures **[9].**

2.7 Règle d'association exacte et approximative

Si une règle d'association à une confiance de 100%, elle est dite exacte sinon elle est dite approximative.

2.8 Règles d'association fortes :

Soient (minSup un seuil de support minimal et minConf et un seuil de confiance minimal, tous deux définis par l'utilisateur. Une règle d'association est dite forte si le support et la confiance sont plus grands ou égaux à leurs seuils respectifs **[10].**

2.9 Le lift (ou l'intérêt)

La mesure du lift a été introduite par Brin et al. en 1997. Le lift correspond à une mesure d'écart par rapport à l'indépendance. La mesure du lift met en évidence l'intérêt d'une règle d'association comme suit :

$$lift(X \rightarrow Y) = \frac{nn_{xy}}{n_x n_y} = \frac{confiance(X,Y)}{n_y}$$

La sélection de règles de manière purement statistique (mesures objectives) se heurte à deux écueils principaux : la production d'un trop grand nombre de règles et l'élimination de règles intéressantes si des critères trop restrictifs sont appliqués.[9]

2.10 Utilité des règles d'association :

Les règles d'associations sont appliquées dans plusieurs domaines

- **Marketing et Planification commerciale :** placement des articles achetés fréquemment ensemble (**elles permettent de** collecter des données sur les habitudes d'achat des clients).
- **Internet:** collecter des données sur la façon dont les visiteurs utilisent un site web ,utiliser des associations dans les données pour optimiser l'interface utilisateur du site web en analysant où les utilisateurs ont tendance à cliquer dont le but de maximiser les chances pour que les visiteurs s'engagent dans un appel à l'action**.**
- **Analyse de données spatiales :** identification des relations entre les caractéristiques des données et la prédiction d'évènements.

2.11 Processus d'extraction de règles d'association:

L'extraction des règles d'association peut être décomposée en quatre étapes **[12]**

:

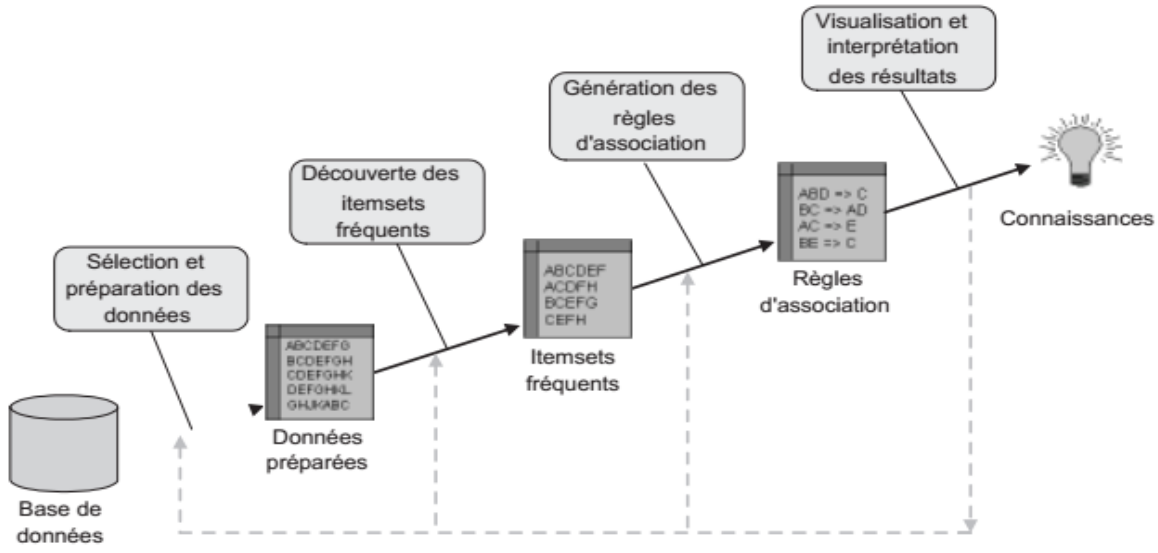

Figure 2.2: Étapes du processus d'extraction de règles d'association

2.11.1 Préparation des données

Cette phase consiste à sélectionner les données (attributs et objets) de la base de données utiles à l'extraction des règles d'association et transformer ces données en un contexte d'extraction. Ce contexte, ou jeu de données, est un triplet B=(O, A, T) dans lequel O est un ensemble d'objets, A est un ensemble d'attributs, également appelés items, et T est une relation binaire entre O et A. Un contexte d'extraction de règles

d'association D constitué de six objets, chacun identifié par son TID, et cinq items est représenté dans la Table 2.3. Cette phase est nécessaire afin qu'il soit possible d'appliquer les algorithmes d'extraction des règles d'association sur des données de natures différentes provenant de sources différentes, de concentrer la recherche sur les données utiles pour l'application et de minimiser les temps d'extraction **[11]** .

Tid	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E
6	B C E

Table 2.3 : Contexte d'extraction des règles d'association D.

2.11.2 Extraction des ensembles fréquents d'attributs

Cette phase consiste à extraire du contexte tous les ensembles d'attributs binaires l ⊆ A, appelés itemsets, qui sont fréquents dans le contexte B. Un itemset l est fréquent si son support, qui correspond au nombre d'objets du contexte qui «contiennent » l, est supérieur ou égal au seuil minimal de support minsupport défini par l'utilisateur. L'ensemble des itemsets fréquents dans le contexte est noté F. Le problème de l'extraction des itemsets fréquents est de complexité exponentielle dans la taille m de l'ensemble d'items puisque le nombre d'itemsets fréquents potentiels est 2^m. Ces itemsets forment un treillis dont la représentation sous forme de diagramme de Hasse pour le contexte D est présentée dans la Figure 2.3 **[11].**

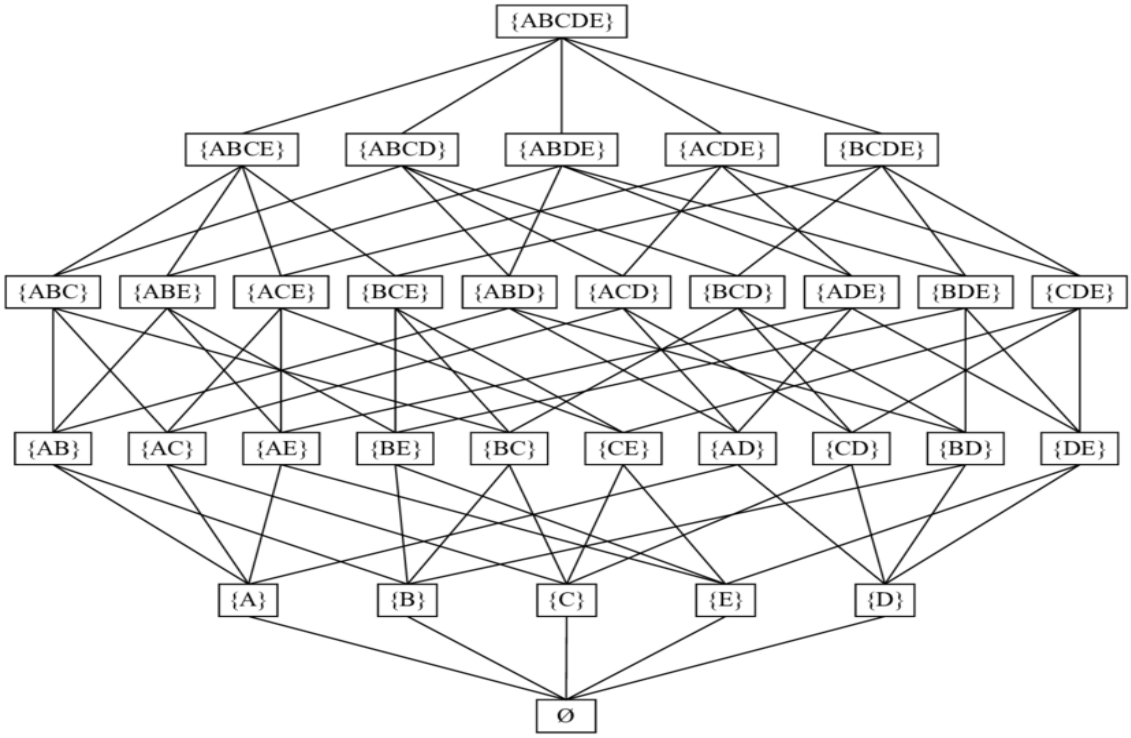

Figure 2.3 : Diagramme de Hasse représentant le treillis des itemsets.

2.11.3 Génération des règles d'association:

Durant cette phase, les itemsets fréquents extraits durant la phase précédente sont utilisés afin de générer les règles d'association qui sont des implications entre deux itemsets fréquents l1, l2 ∈ F tels que l1 ⊂ l2, de la forme r : l1 → (l2 \ l1). Afin de limiter l'extraction aux règles d'association les plus informatives, seules celles qui possèdent une confiance supérieure ou égale au seuil minimal minConfiance défini par l'utilisateur sont générées. La confiance d'une règle T : l1 → (l2 \ l1) est définie comme la proportion d'objets contenant la conséquence (l2 \ l1) de T parmi ceux qui contiennent l'antécédent l1 de T. Cette valeur est égale au rapport entre le support de l'itemset l2 et le support de l'itemset l1 **[11].**

2.11.4 Interprétation des résultats

Cette phase consiste en la visualisation par l'utilisateur des règles d'association extraites du contexte et leur interprétation afin d'en déduire des connaissances utiles pour l'amélioration de l'activité concernée. Le nombre important de règles d'association extraites en général impose le développement d'outils de classification des règles, de sélection par l'utilisateur de sous-ensembles de règles, et de leur visualisation sous une forme intelligible. Les connaissances de l'utilisateur concernant le domaine d'application sont nécessaires lors des phases de prétraitement, afin d'assister la sélection et la

préparation des données, et de post-traitement, pour l'interprétation et l'évaluation des règles extraites. En fonction de l'évaluation des règles extraites, les paramètres utilisés lors des précédentes phases (critères de sélection et préparation des données et seuils minimaux de support et de confiance) peuvent être modifiés avant d'effectuer à nouveau l'extraction des règles d'association, ceci afin d'améliorer la qualité du résultat**[11].**

2.12 Algorithme d'extraction des règles d'association:

2.12.1 Historique

Depuis l'algorithme de référence d'Agrawal et Srikant, de nombreux algorithmes ont été développés et sont synthétisés dans Hipp et al.. Différents algorithmes ont également été développés pour réduire le nombre d'itemsets en générant des itemsets fermés, maximaux ou optimaux. D'autres algorithmes ont été développés pour réduire le nombre de règles .Enfin, différentes méthodes complémentaires de post-traitement ont été proposées telles que l'élagage (pruning), les résumés (summarizing) ou les regroupements (grouping) [**3].**

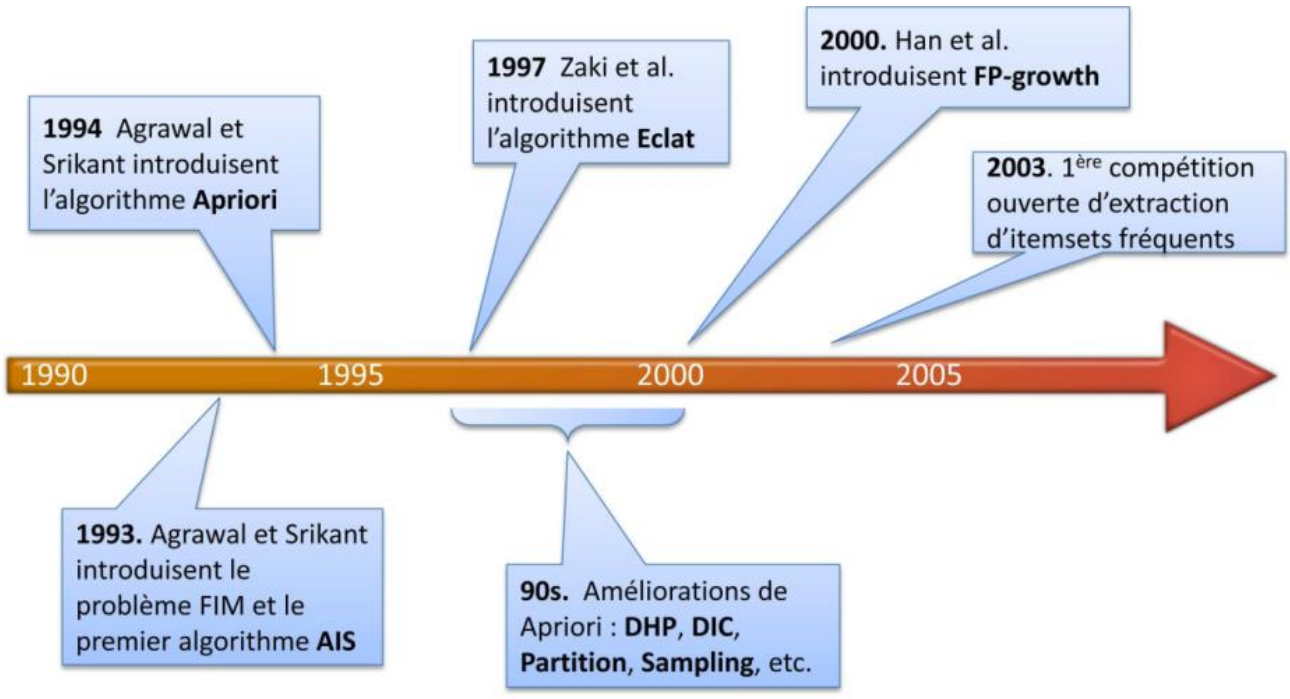

Figure 2.4 : Historique des algorithmes d'extraction d'itemsets fréquents

2.12.2Algorithme apriori

l'algorithme Apriori est un algorithme d'exploration de donnée proposé en 1994 , par Agrawal et all, La plupart des algorithmes de recherche de règles d'association se basant sur Apriori, il se base sur les deux propriétés suivantes :

Propriété 1 : Tous les sous-ensembles d'un itemset fréquent sont fréquents.

Cette propriété permet de limiter le nombre de candidats de taille *k* générés lors de la $k^{ème}$ itération (parmi les 2^kitemsets de taille *k* existants) en réalisant une jointure conditionnelle des itemsets fréquents de taille *k-1* découverts lors de l'itération précédente.

Propriété 2 : Tous les sur-ensembles d'un itemset infréquent sont infréquents.

Cette propriété permet de supprimer un candidat de taille *k* lorsque au moins un de ses sous-ensembles de taille *k-1* ne fait pas partie des itemsets fréquents découverts lors de l'itération précédente **[12].**

2.12.2.1Fonctionnement de l'algorithme Apriori [13]:

On considérera les notations suivantes :

C_k désigne l'ensemble des k-itemsets candidats (i. e. potentiellement fréquents).
F_k désigne l'ensemble des k-itemsets fréquents.

L'algorithme Apriori procède de manière itérative pour identifier les k−itemsets fréquents, c'est à dire que l'ensemble des itemsets fréquents est parcouru par niveau (parcours en largeur du haut vers le bas du treillis d'itemsets). Lors de la $k^{ème}$ itération, on recherche les k-itemsets fréquents, c'est-à dire les itemsets de longueur k qui sont fréquents. A l'itération suivante, la $(k+1)^{ème}$, on recherchera les (k+1)-itemsets fréquents et ainsi de suite.

Pour chaque itération k, on génère les k-itemsets candidats (à être fréquents) C_k à partir de l'ensemble des itemsets fréquents identifiés à l'itération précédente F_{k-1}.

On scanne les données (balayage) afin de calculer le support de chaque itemset candidat c ($c \in C_k$).
Puis les éléments c de C_k ayant un support suffisant (les éléments c de C_k fréquents) sont ajoutés à l'ensemble des itemsets fréquents F_k(voir table 2.4).

Durant la première itération de l'algorithme (table 2.4, ligne 1), tous les itemsets de taille1 sont considérés, et un balayage est réalisé pour déterminer l'ensemble des 1-itemsets fréquents F_1.

Chaque itération k suivante (Table 2.4, lignes 2 à 8) se subdivise en deux phases.

Durant la première phase (Table 2.4, ligne 3), l'ensemble C_k des k-itemsets candidats est construit par association des (k −1) itemsets fréquents de F_{k-1}.Cette phase est réalisée par la fonction Apriori-Gen (voir table 2.5).

Durant la deuxième phase (Table 2.4, lignes 4 à 8), un balayage des données est réalisé afin de déterminer le support de chacun des k−itemsets candidats de C_k. Les k−itemsets fréquents sont insérés dans F_k.

Remarque: L'opération consistant à ne pas introduire les k-itemsets infréquents dans F_k ,et donc à ne pas les prendre en compte pour la suite du traitement, est appelée pruning ou élagage.

La procédure APrioriGen (F_{k-1}) :

La procédure AprioriGen reçoit un ensemble F_{k-1} de (k - 1)-itemsets fréquents comme paramètre. Elle retourne un ensemble C_k de k-itemsets candidats. Les éléments de C_k sont générés par combinaison des éléments de F_{k-1}.

L'algorithme Apriori

Entrée : la base, le seuil de support minimal
Sortie : l'ensemble F_k des k-itemsets fréquents
(1) F_1 = 1-itemsets fréquents
(2) **pour**(k = 2; $F_{k-1} \neq 0$; k + +)**faire**
(3) C_k = AprioriGen(F_{k-1})
(4) **pour** chaque instance lue *inst***faire**
(5) **pour** chaque élément c de C_k tel que c $\in$*inst* **faire**
(6) c.support + +
(7) **fin pour**
(8) **fin pour**
(9) F_k = {c $\in C_k$\c.support ≥ minsupport}
(10) **fin pour**
(11) retourner $\cup F_k$

Table 2.4 Principe de l'algorithme Apriori

La procédure se décompose en deux étapes, la première visant à générer des candidats, la deuxième visant à en éliminer une partie. Durant une première partie de la procédure, les (k - 1)-itemsets fréquents de F_{k-1} sont combinés, et les résultats de ces combinaisons sont insérés dans C_k. Durant la deuxième partie, les éléments de C_k dont l'un des sous-ensembles est infréquent sont supprimés de C_k(élagage de C_k).

Au départ (voir Table 2.5, ligne 1 à 4), on effectue des combinaisons de deux (k-1) - itemsets fréquents p et q, ayant k-2 items communs. Il en découle un k-itemsets c dont les de k- 2 ^{eme} premiers items sont les items communs à p et q , et donc les k-1 ^{eme} et k ^{eme}items sont les items distincts de p et q. c est donc un candidat à être fréquent, ilest ajouté à l'ensemble itemsets candidats que l'on est en train de construire, C_k. Une fois l'ensemble C_k des k-itemsets candidats générés par combinaison des (k-1)-itemsets fréquents (F_{k-1}), on supprime une partie de ces candidats (Table 4.5 , lignes 5 à 9) en se basant sur la **Propriété 1 : Tout sous-ensemble d'un ensemble fréquent doit être fréquent**, (donc si un ensemble à un de ses sous-ensembles infréquent, il est infréquent,**Propriété2**).
Ainsi, tous les éléments c de C_k dont l'un des sous-ensembles s n'est pas fréquent ($s \notin F_{k-1}$) sont supprimés de C_k, soit un élagage de C_k(voir Table 2.5).

La procédure AprioriGen :

Entrée : l'ensemble F_{k-1} des (k-1)-itemsets fréquents
Sortie : l'ensemble C_k des k-itemsets candidats
(1) insérer dans C_k
(2) select p[1], p[2], ..., p[k-1], q[k-1]
(3) from Fk−1 p , Fk−1 q
(4) where p[1] = q[1],..., p[k-2]= q[k-2], p[k-1] < q[k-1]
(5) pour chaque itemset candidat c de C_k faire
(6) pour chaque sous-ensemble s de c tel que |s| = k − 1 faire
(7) si $s \notin F_{k-1}$ alors supprimer c de C_k
(8) fin pour
(9) fin pour
(10) retourne C_k

Table 2.5 La procédure AprioriGen

2.14 Exemple de l'algorithme Apriori :

Voici un exemple détaillé des étapes suivies par l'algorithme *Apriori*. La base de données utilisée (D) contenant 4 transactions, le support minimal est fixé à 20 %(ou 2/4). Les résultats détaillés de chacune des étapes sont illustrés à la figure 5.

1. Lors de la première itération, l'algorithme compte le support de chaque 1-*itemset* de la base de données. Ces *itemsets* forment l'ensemble des candidats C_1 qui sert à générer l'ensemble L_1, c'est à dire l'ensemble des 1-itemsets fréquents.

2. Premier élagage : L'algorithme compare ensuite le support de chaque 1-*itemset* avec le support minimal prédéterminé ici à 20 % ou une fréquence de 2. Tous les *itemsets* ayant un support inférieur à 2/4 sont retirés et ceux dont le support est supérieur ou égal à 2/4 sont conservés afin de générer l'ensemble L_1. Les 1-itemsets fréquents qui forment L_1 sont dans ce cas {A}, {B}, {C}, {D}, {E}.

3. **Étape de jointure :** Les 1-*itemsets* de l'ensemble L_1 sont utilisés pour générer les candidats C_2. La génération des candidats est réalisée en liant l'ensemble L_1 avec lui-même. Comme il s'agit de 1-*itemsets*, le nombre de combinaison possible est de $n(n-1)/2$,*n*étant le nombre d'*itemset*s. Dans cet exemple, le nombre d'itemsets étant de 4, six candidats sont formés. Les candidats sont {A,B}, {A,C}, {A,E}, {B,C}, {B,E} et {C,E}.

4. **Calcul du support**. Lorsque les 2-*itemsets* candidats ont été générés, l'algorithme effectue un autre balayage de la base de données afin de calculer la fréquence respective des candidats. Cette fréquence est inscrite dans une table comme illustré à la figure 2.5.

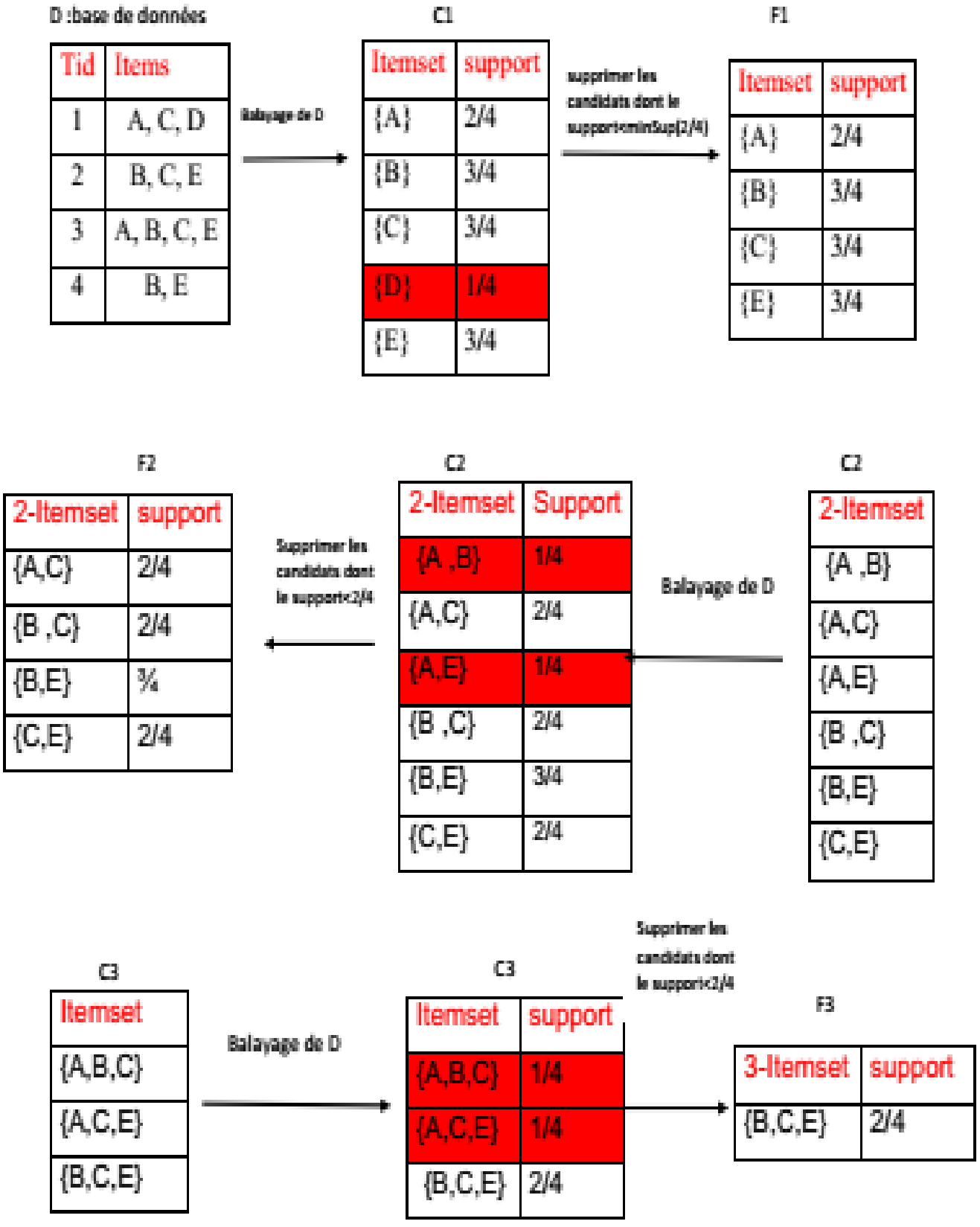

Figure: 2.5 Exemple de l'algorithme Apriori

2.15 Avantages et inconvénients de l'algorithme Apriori:

Avantages :

Il existe une multitude d'avantages dans l'utilisation de l'algorithme Apriori. On en énumère quelques-uns **[14][15]** :

- Est un algorithme facile à comprendre.
- La découverte rapide de règles d'association pertinentes entre objets.
- La facilité d'interprétation des résultats lors de l'extraction des règles d'association, malgré le nombre important de ces dernières.

Inconvénients :

Les inconvénients auxquels on fait face lors d'une utilisation de l'algorithme Apriori sont les suivants **[16]**

- Les algorithmes d'extraction liés à l'approche support / confiance génèrent un grand nombre de règles d'association.
- Un nombre important de configurations d'items ne peuvent pas engendrer de règles d'association.

2.16 Conclusion

Dans ce chapitre, nous avons présenté le problème de l'extraction des règles d'association. Il était question d'abord de les définir en fonction des items et des transactions qui y sont impliqués. Ensuite, nous avons présenté les caractéristiques des règles d'association, notamment les deux principales, à savoir le support et la confiance. Il a été relevé de la littérature du domaine que l'algorithme Apriori est le plus optimal et le plus utilisé dans les différents travaux ayant traité des règles d'association.

Chapitre 3

Conception

3.1 Introduction

Après avoir présenté au chapitre 2 les règles d'association, leur intérêt, et les principaux algorithmes permettant leur calcul à partir des bases de données, via l'ensemble des transactions, nous présentons dans ce chapitre la conception de notre système du calcul de l'ensemble des règles d'association, à partir d'un schéma de base de données. A partir de la table adéquate, l'ensemble des transactions est extrait, puis l'algorithme apriori est appliqué.

Pour exprimer notre conception, nous avons utilisé le langage de modélisation UML, via les 3 principaux diagrammes, à savoir le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classes. Cependant, nous commençons le chapitre par la présentation du langage de modélisation UML.

3.2 U.M.L (UNIFIED MODELING LANGUAGE)

U.M.L est une Technique de modélisation UNIFIEE issue de méthodes plus anciennes comme O.M.T, d'O.O.S.E et d'O.O.D. De plus, UML a choisi une notation supplémentaire : il s'agit d'une forme visuelle fondée sur des diagrammes.

UML n'est pas une méthode (i.e. une description normative des étapes de la modélisation) : ses auteurs ont en effet estimé qu'il n'était pas opportun de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré se borner à définir un langage graphique qui permet de représenter, de communiquer les divers aspects d'un système d'information.

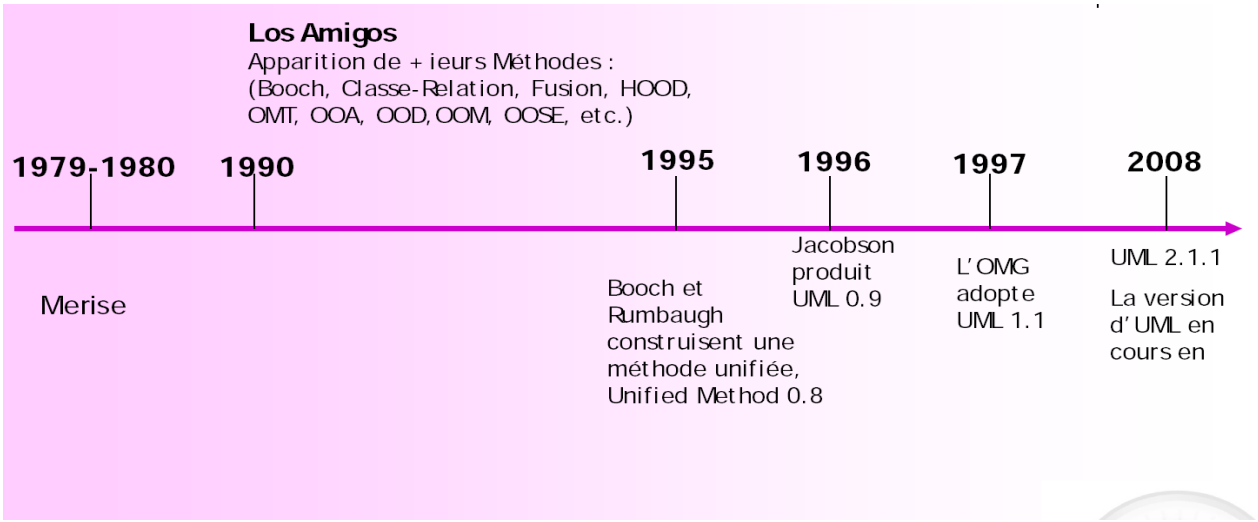

Figure 3.1 : Historique UML

UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet **[17].**

UML 2.5 offre ainsi quatorze diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes :

Diagrammes structurels ou diagrammes statiques (UML Structure)

– diagramme de classes (Class diagram)
– diagramme d'objets (Object diagram)
– diagramme de composants (Component diagram)
– diagramme de déploiement (Deploymentdiagram)
– diagramme de paquetages (Package diagram)
– diagramme de structures composites (Composite structure diagram)
– diagramme de profils (Profiles diagram)

Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)

– diagramme de cas d'utilisation (Use case diagram)
– diagramme d'activités (Activitydiagram)
– diagramme d'états-transitions (State machine diagram)

Diagrammes d'interaction (Interaction diagram)

– diagramme de séquence (Sequencediagram)
– diagramme de communication (Communication diagram)
– diagramme global d'interaction (Interaction overviewdiagram)
– diagramme de temps (Timing diagram)

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états-transitions. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique**[17].**

3.3 Fonctionnement du système:

Un système d'extraction des règles d'association à partir d'une base de données procède pour la détection des Motifs qui se répètent au sein de l'ensemble des attributs formants les tables de la base de données. Si nous considérons une base de données commerciale, où la base de données contient une table dite "ventes " par exemple, le problème d'extraction des règles d'association consiste à trouver et calculer les fréquences des co-occurrence des produits qui sont acquis simultanément par les clients. Ces associations permettent au gérant du magasin de prévoir les quantités d'approvisionnements des différents produits, et permet au chargé du Marketing à définir une politique de recommandation : un client qui achète un produit donné, on lui propose les produits qui ressortent les règles d'association avec le produit acheté.

Un schéma conceptuel d'une Base de données d'une application de vente au sein d'un magasin pourra être présenté selon le diagramme de classe suivant:

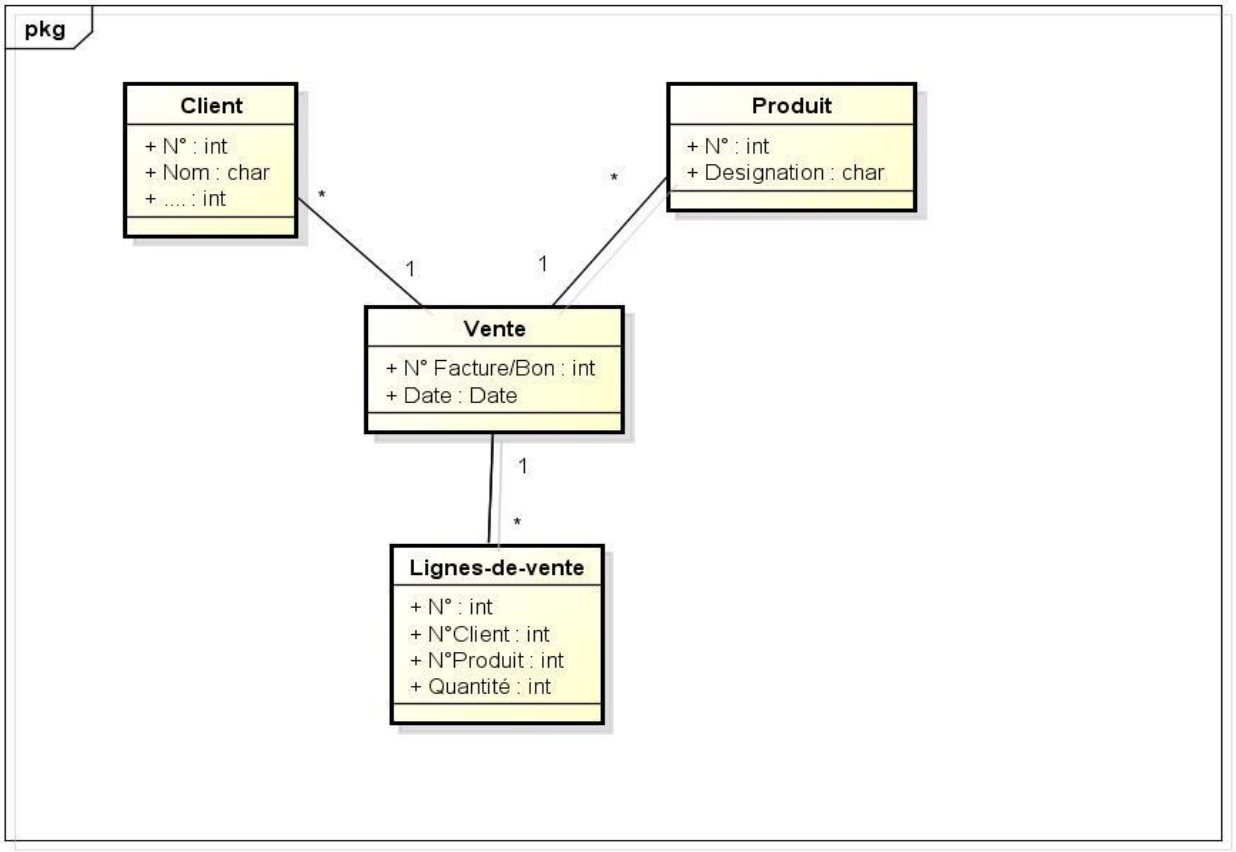

Figure 3.2. Schéma de la base de données d'une application de vente.

A partir des tables "Vente" et "lignes de vente" on produit la liste de toutes les transactions où chacune de ces dernière et la suite des articles achetés par un client donné :

N°	$Article_1$	$Article_2$	……..	$Article_j$		$Article_N$
01 . . i .	0/1	0/1		0/1		0/1

Table3.1 : une base de données qui comporte des transactions

La transaction à la ligne "i" correspond à un client qui aurait acheté un sous-ensemble de tous les articles du magasin {article1,……,$article_N$} , où il est enregistré **1** à l'élément (i, j) si le client "i" a acheté l'article "j" ,sinon **0** est enregistré" .

On peut représenter les transactions par énumération des articles qui la composent:

Transaction	$Article_1$	$Article_2$	.....	$Article_j$
01	$Article_{11}$	$Article_{12}$	.....	$Article_{1N1}$
.	.	.	.....	.
.	.	.	.....	.
i	$Article_{i1}$	$Article_{i2}$	.....	$Article_{iNi}$
.	.	.	.....	.
M	$Article_{M1}$	$Article_{M2}$	.....	$Article_{MNM}$

Table 3.2 : Représentation compacte des transactions

Cette représentation, que nous allons adopte dans notre travail, nous permet d'économiser considérablement en mémoire le stockage des transactions, car les valeurs N_i sont très inférieures à la valeur N, représentant le nombre de tous les articles que le magasin propose à la vente.

3.4 Diagrammes d'UML :

3.4.1 Les acteurs du système:

Un acteur représente un rôle joué par une entité (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié **[18].**

Pour notre système, nous envisageons deux acteurs, à savoir **le gestionnaire de la Base de données**, jouent le rôle de **l'administrateur du système**, et **l'utilisateur**, qui calcule les règles d'association, et les exploiteurs pour un système de recommandation par exemple.

3.4.2 Diagramme de cas d'utilisation:

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre **[17].**

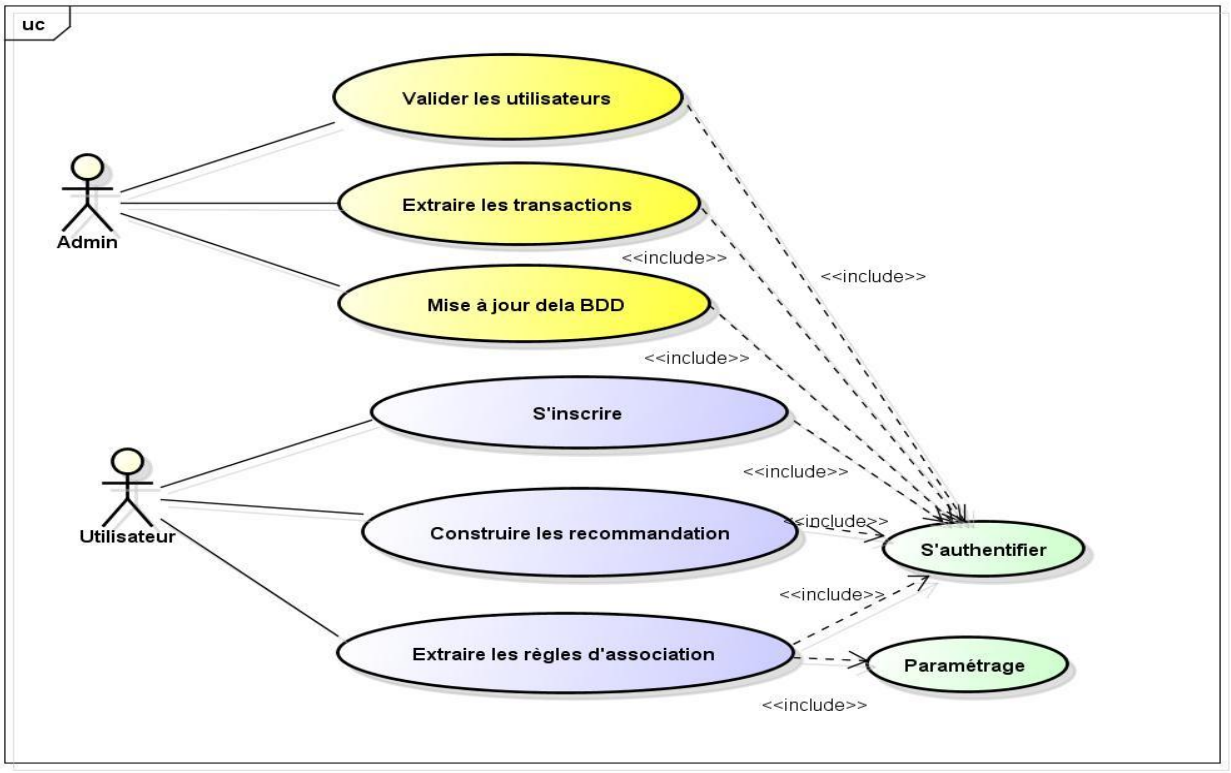

Figure 3.3 Diagramme de cas d'utilisation de système.

3.4.3 Scénarios

3.4.3.1 Administrateur:

- **Valider les utilisateurs:**

L'administrateur du système accepte ou refuse l'inscription d'un utilisateur en fonction de ses attributs.

- **Extraire les transactions :**

L'administrateur, exécute un utilitaire, Permettant d'extraire la liste de toutes les transactions à partir des tables " ventes" et "lignes de vente".

- **Mise à jour de la Base de données**

En fonction de l'ajout ou de la suppression d'article dans le magasin, l'administrateur procède aux mises à jour nécessaires.

3.4.3.2 Utilisateur:

- **Inscription:**

  Un utilisateur du système doit s'inscrire pour pouvoir l'utiliser. Pour se faire il doit fournir des données d'identification tel qu'un mail valide, et doit définir un mot de passe accepté.

- **Authentification:**

  Avant toute utilisation du système, l'utilisateur doit s'authentifier, en fournissant son user name (mail) et son mot de passe.

- **Paramétrage :**

  L'utilisateur peut définir quelques paramètres relatifs aux règles d'association tel que, le support minimale, et la confiance minimale (voir chapitre2).

- **Extraire les règles d'association**:

  Ça consiste à exploiter la liste des transactions (ventes) pour en extraire les co-occurrences d'articles, et selon les deux paramètres, support minimal et confiance minimale.

- **Construire les recommandations :**

  Toute co-occurrence d'article, obtenue à partir d'une règle d'association, constituera une potentielle recommandation. Ce pendant l'utilisateur sélectionne parmi les recommandations celles qui sont les pertinentes.

3.4.4 Diagrammes de séquences:

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre. On peut représenter les mêmes opérations par un diagramme de communication, graphe dont les noeuds sont des objets et les arcs (numérotés selon la chronologie) les échanges entre objets. En fait, diagramme de séquence et diagramme de communication sont deux vues différentes mais logiquement équivalentes (on peut construire l'une à partir de l'autre) d'une même chronologie. Ce sont des diagrammes d'interaction **[17].**

3.4.4.1. Inscription :

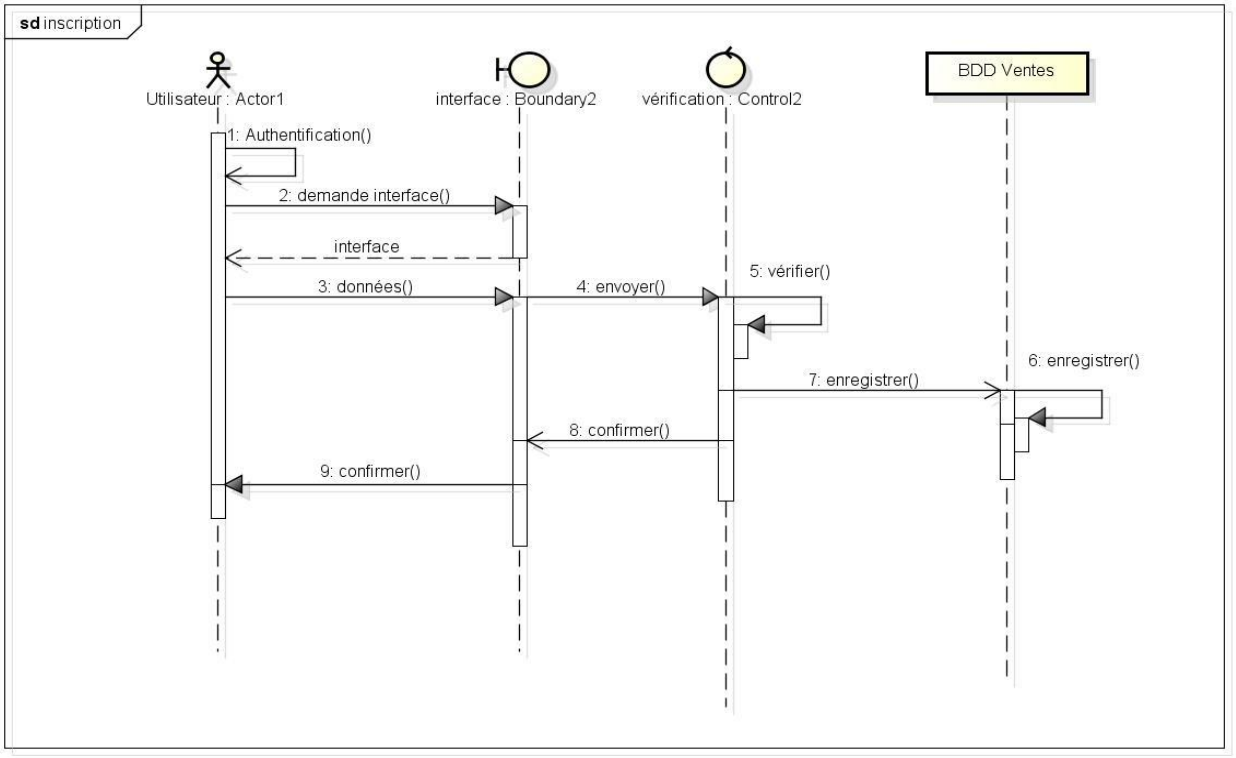

Figure 3.4 Diagramme de séquence de cas d'utilisation « Inscription »

3.4.4.2. Valider les utilisateurs :

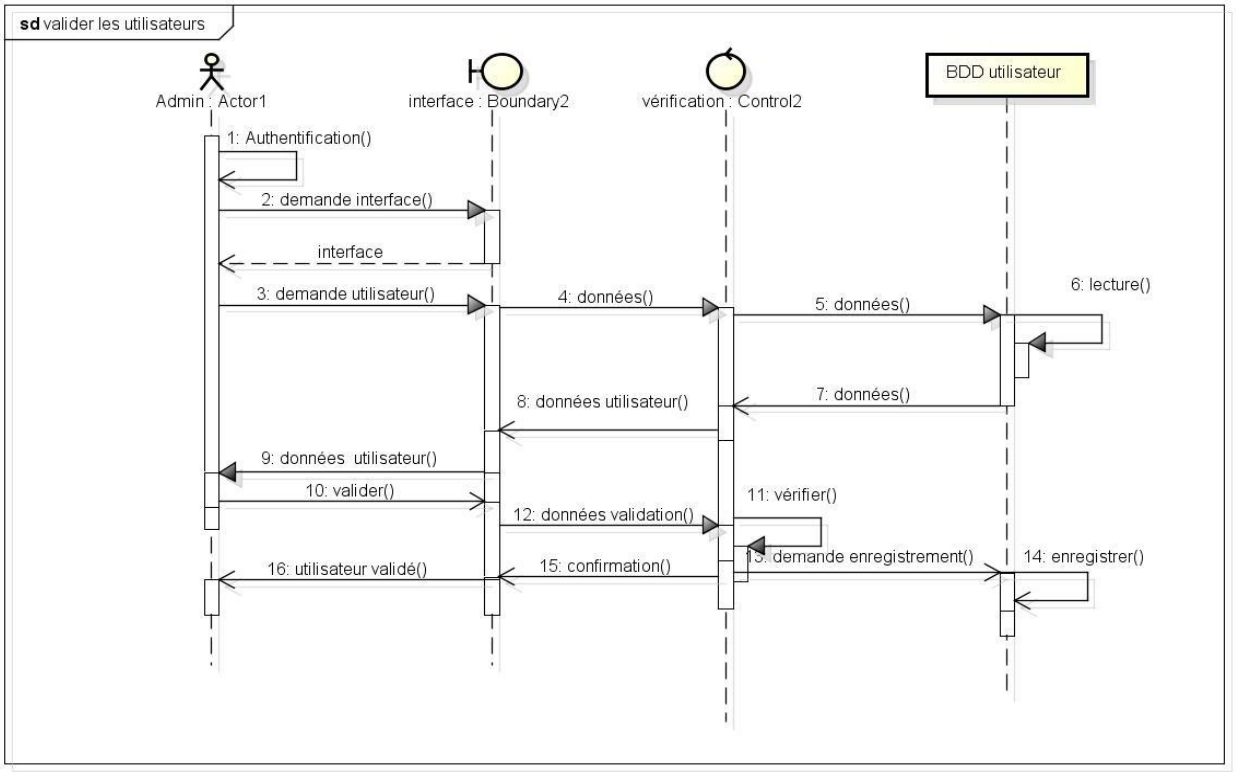

Figure 3.5 Diagramme de séquence de cas d'utilisation « Valider les utilisateurs »

3.4.4.3. Extraire les transactions :

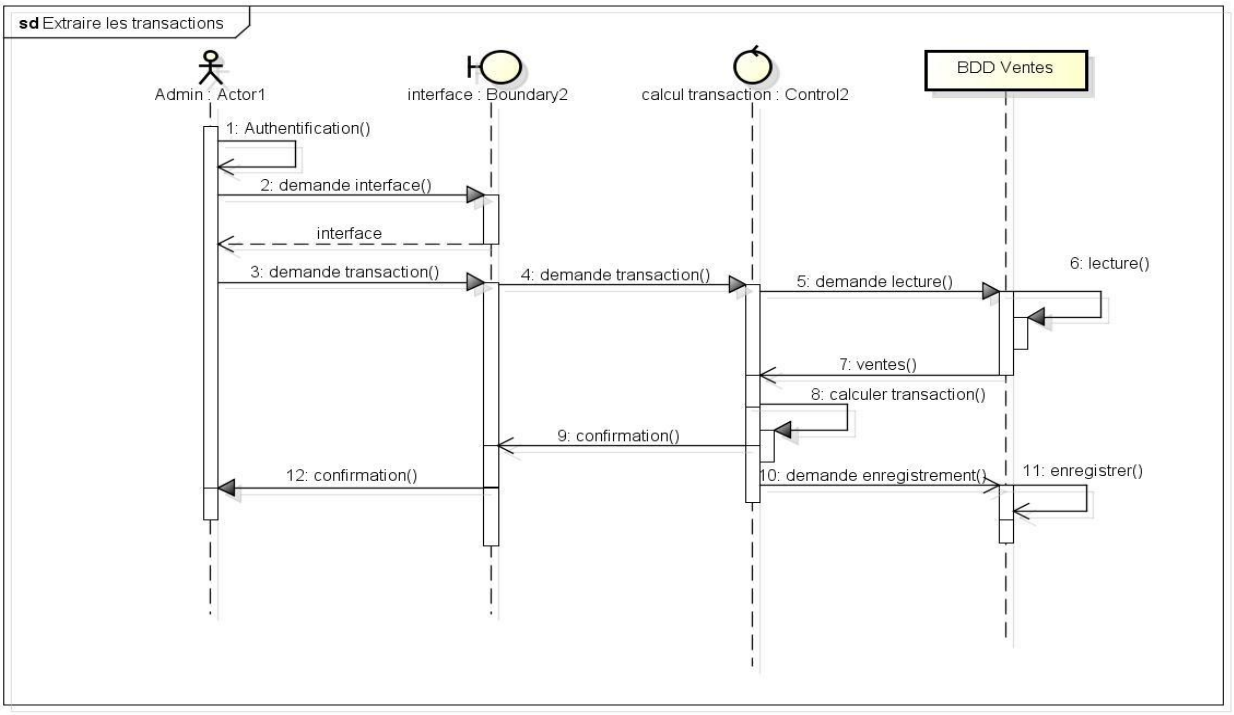

Figure 3.6 Diagramme de séquence de cas d'utilisation « Extraire les transactions »

3.4.4.4 Extraire les règles d'association:

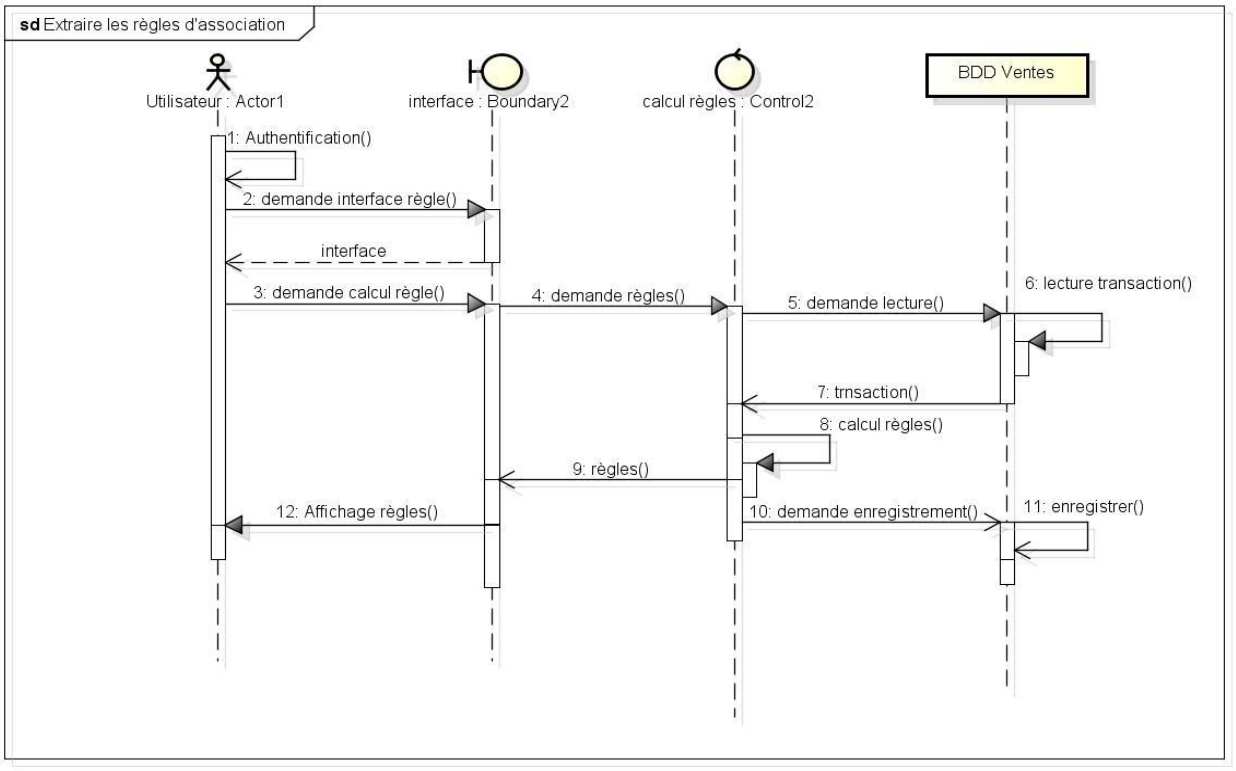

Figure 3.7. Diagramme de séquence de cas d'utilisation « Extraire les règles d'association »

3.4.4.5Construire les recommendations

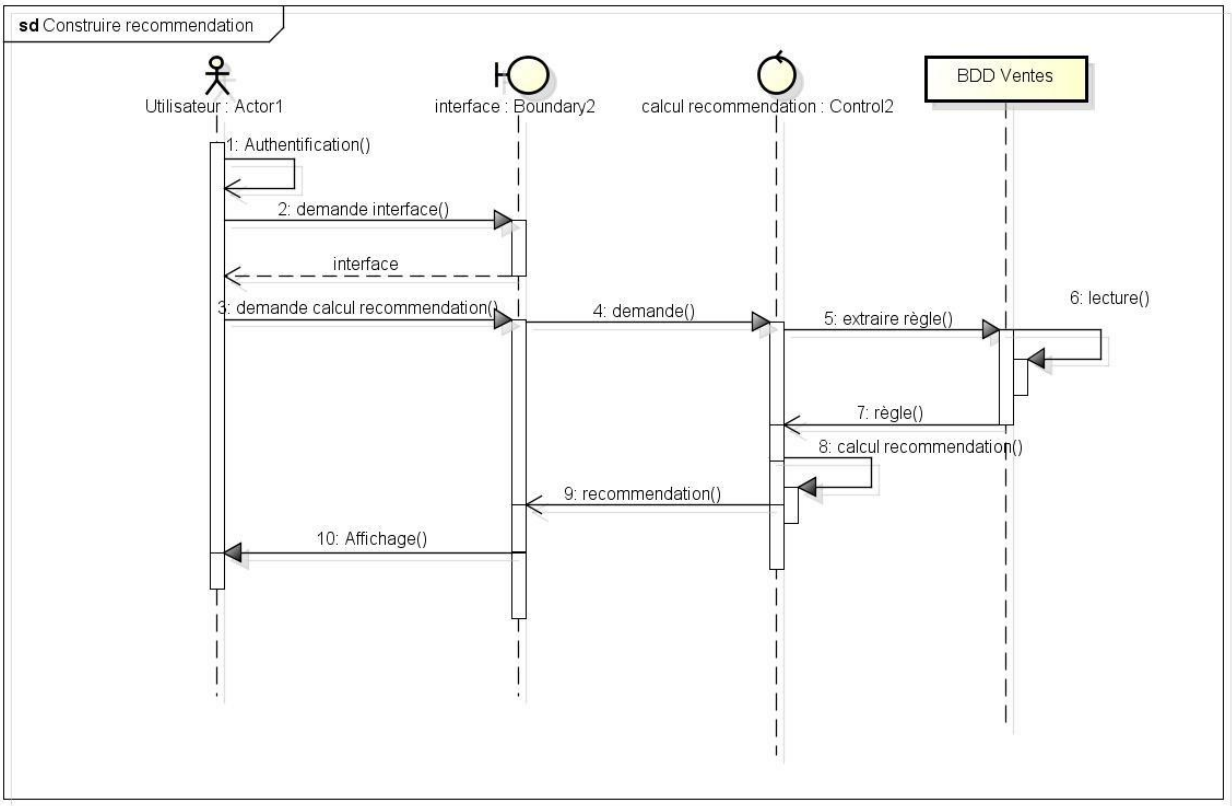

Figure 3.8 Diagramme de séquence de cas d'utilisation « construire les recommendation »

3.4.4.6Diagramme de classe

Le diagramme de classes est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens, que ceux-ci représentent un emboîtage conceptuel (héritage)ou une relation organique (agrégation)**[17].**

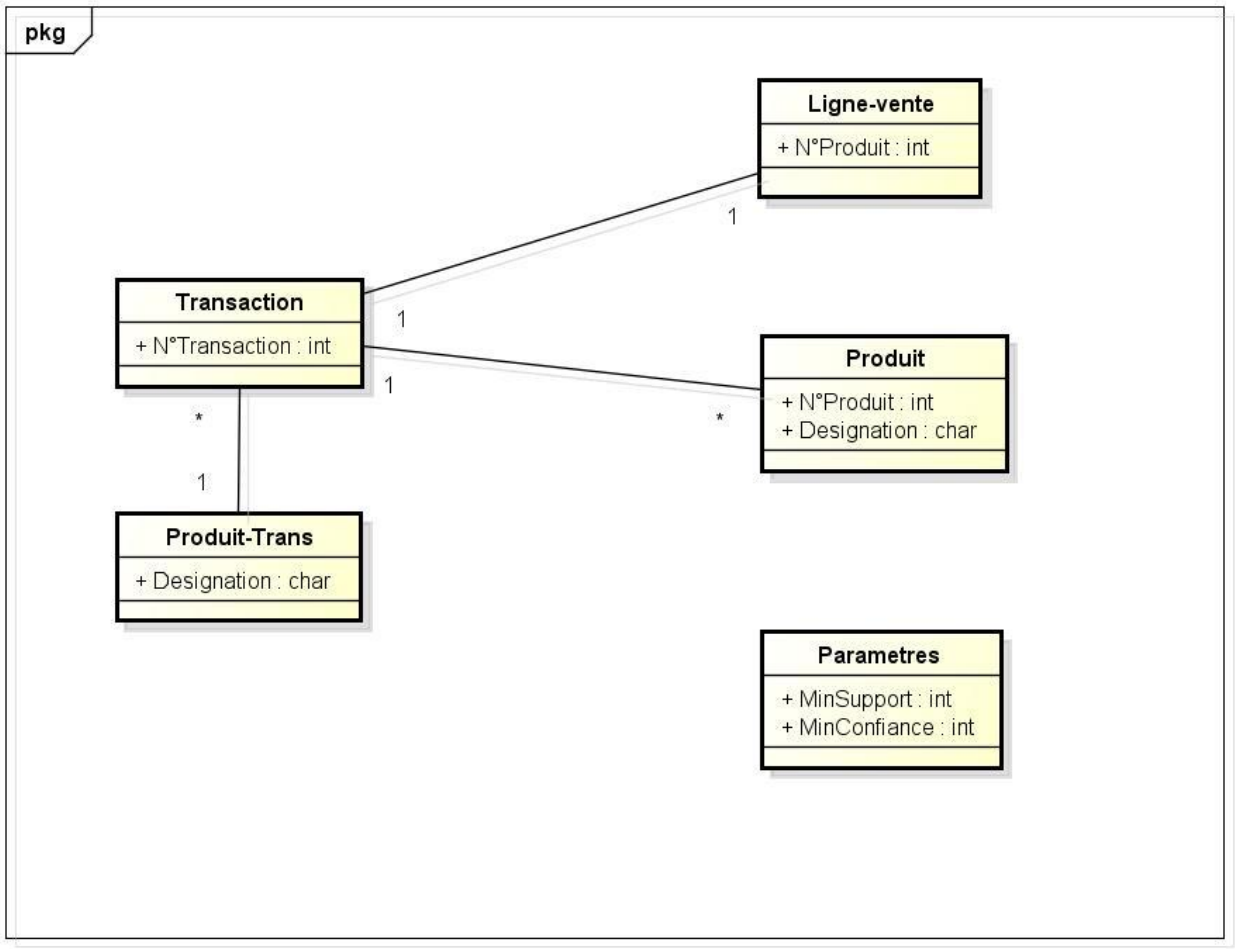

Figure 3.9. Diagramme de classes

3.5 Conclusion

Dans ce chapitre nous avons présenté la conception de notre système pour le calcul des règles d'association dans l'objectif de construire des recommandations à proposer aux clients, dans une application de vente dans un magasin. Pour ce faire, nous avons opté pour le langage de modélisation UML, dont il nous a permis d'avancer méthodiquement pour comprendre et exprimer le système à développer. En effet, via le langage UML, nous avons pu exprimer les trois aspects que nous considérons les plus importants dans notre cas, à savoir, les acteurs, les séquences d'opérations, et les différentes tables qui seront impliquées dans le calcul des règles d'association et des recommandations .

Au chapitre suivant, nous montrons la mise en œuvre de notre système, en utilisant le langage « Python », sous l'environnement de développement en ligne «Google Colab ».

Chapitre 4

Implémentation

4.1 Introduction

Après avoir présenté au chapitre précédent la conception de notre système pour le calcul des règles d'association et les recommandations, nous présentons dans ce chapitre son implémentation et ses tests. Tous d'abord, nous commençons par l'introduction des outils logiciels utilisés, à savoir le langage de programmation Python, et l'environnement de développement Google Colab. Ensuite, nous montrons les éléments d'implémentation et des cas de test de notre système, en analysons et en montrant les résultats du test du problème du panier de la ménagère.

4.2. Python

Python est un langage de programmation populaire. Il a été créé par Guido van Rossum et publié en 1991.

Il est utilisé pour

- ✓ le développement web (côté serveur),
- ✓ le développement de logiciels,
- ✓ les mathématiques,
- ✓ l'écriture de scripts système **[22].**

Le langage Python est placé sous une licence libre proche et fonctionne sur la plupart des plateformes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser **[21].**

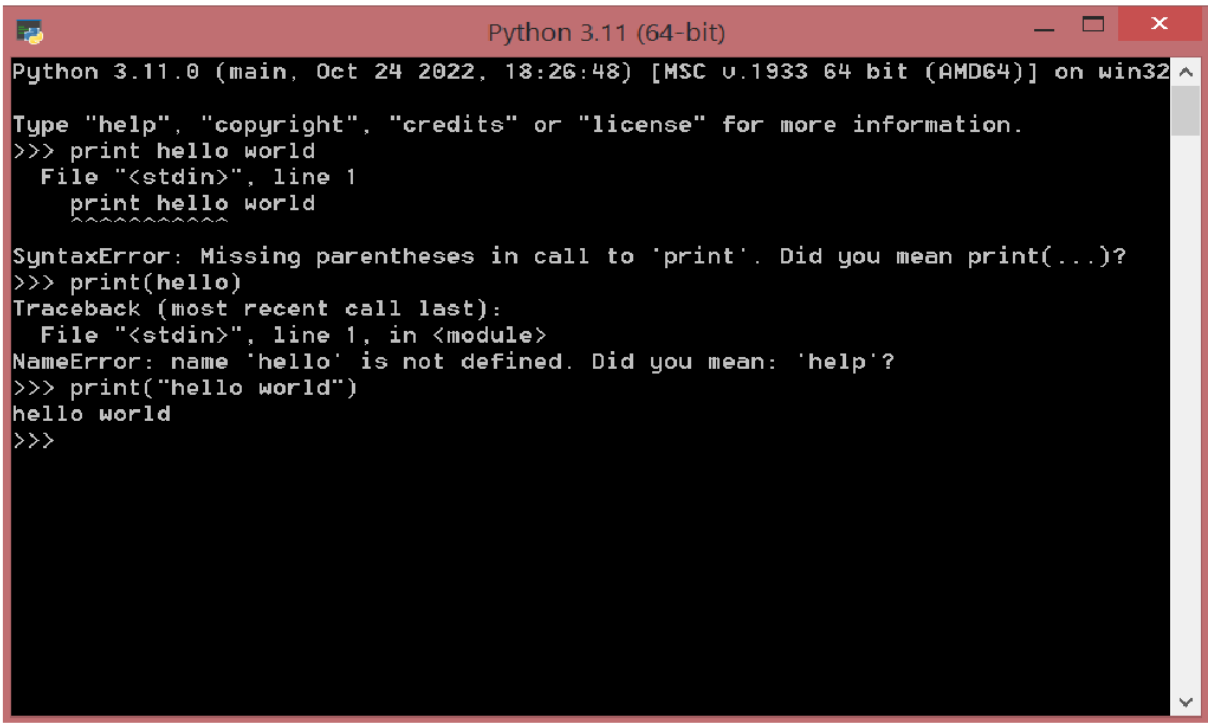

Figure 4.1 Editeur Python

La syntaxe du langage Python permet aux développeurs de réaliser des programmes avec moins de lignes que d'autres langages de programmation**[22].**

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation**[21].**

Il peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives . **[23]**

On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique **[21].**

4.3 Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur **[24].**

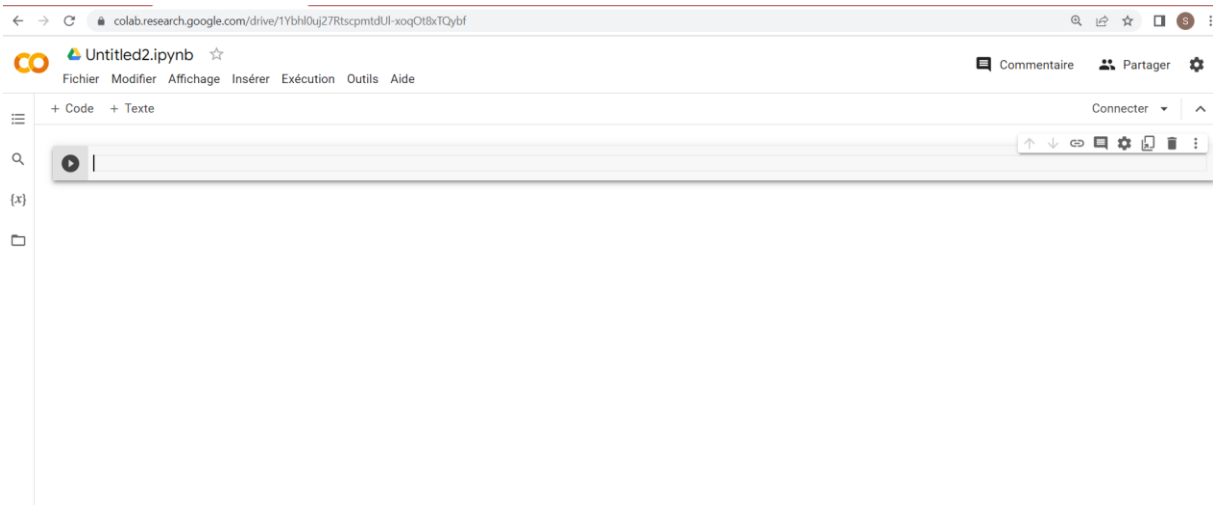

Figure 4.2 Interface Google Colab

Google Colab est un outil complet pour entraîner rapidement et tester rapidement des modèles d'apprentissage automatique sans avoir de contrainte matérielle. Sa particularité est que tout le monde peut l'utiliser. **[24]** Colab rend la science des données, l'apprentissage en profondeur, les réseaux de neurones et l'apprentissage automatique accessibles aux chercheurs individuels qui ne peuvent pas se permettre une infrastructure de calcul coûteuse **[1].**

Colab vous permet d'écrire et d'exécuter du code Python dans votre navigateur avec

- Aucune configuration requise
- Accès sans frais aux GPU
- Partage facile **[26]**

Colab est un produit de type Jupyter Notebook de Google Research. Un développeur de programme Python peut utiliser ce bloc-notes pour écrire et exécuter des codes de programme Python aléatoires simplement à l'aide d'un navigateur Web **[25].**

En un mot, Colab est une version hébergée dans le cloud de Jupyter Notebook. Pour utiliser Colab, vous n'avez pas besoin d'installer et d'exécuter ou de mettre à niveau votre matériel informatique pour répondre aux exigences de charge de travail intensives du CPU/GPU de Python. De plus, Colab vous donne un accès gratuit à l'infrastructure informatique comme le stockage, la mémoire, la capacité de traitement, les unités de traitement graphique (GPU) et les unités de traitement de tenseur (TPU).

Google a spécialement programmé cet outil de codage Python basé sur le cloud en gardant à l'esprit les besoins des programmeurs d'apprentissage automatique, des analystes de données volumineuses, des scientifiques des données, des chercheurs en IA et des apprenants Python.**[25]**

4.4. Implémentation

Nous montrons d'abord au lecteur qui voudrez tester notre code python pour le calcul des règles d'association, la procédure d'installation et d'exécution est la suivante:

1- se connecter à gmail.

2 - Accéder à Google drive.

3 - Créer un dossier sous drive de nom: RegAssociation.

4 - Accéder à ce dossier puis télécharger le fichier Transactions.csv. Ce fichier contient la liste des transactions obtenues à partir de la table des ventes.

5 - Ouvrir avec Wordpad le fichier RegAsso.ipynb, contenant le code python pour le calcul des règles d'associations.

6 - Copier le texte du code Python.

7 - Accéder à Google Colab par https://colab.research.google.com/.

8 - Cliquer sur nouveau note book.

9 - coller le texte (le code python copié)

10 - Renommer le fichier "RegAsso.ipynb"

11- Activer le drive sous Google Colab

12 - Exécuter la cellule

Le code, bien commenté, permettant le calcul des règles d'association est le suivant :

```
# Importation des modules utilisés
importnumpyasnp
import pandas aspd
# Importaion du module Apriori
fromapyoriimport apriori

# Lecteur de la base des transaction sous format CSV
# Toute transaction est la suite des articles achetés
Donnees = pd.read_csv("/content/drive/MyDrive/RegAssociatio/Transactions4.csv",
delimiter=";", header=None)
print(Donnees)

# Nombre de lignes et nombre de colones
NLignes = 6
NColones = 8

# Convertion des données en suite de liste
# chacune avec les items achetés
Lignes = []
foriinrange(0,NLignes):
  Lignes.append([str(Donnees.values[i,j]) for j inrange(0,NColones)])

print("avant supp")
print(Lignes)
# Effacer les cellules vides
for i inrange(len(Lignes)):
  Lignes[i] = [x for x inLignes[i] if x!='nan']
```

```
# Affichage des Transactions au format liste
print(Lignes)

# Apriori
# avec support minimal = 0.16
# et confiance minimale = 0.20
ReglesdAssociation = apriori(Lignes, min_support=0.16, min_confidence=0.2, min_lift=3, max_length=2)
# Restructuiration en liste
Regles = list(ReglesdAssociation)

# Affichage des resultats
forregleinRegles:
  elements = regle[0]
  # éléments contient la suite des éléments définissant les associations
  items = [x for x in elements]
  print()
  # Affichage d'une règle
  print('Règle: ' + items[0] + ' avec ' + items[1])
  print('Support: ' + str(regle[1]))
  print('Confiance: ' + str(regle[2][0][2]))
  print('Lift: ' + str(regle[2][0][3]))
```

Comment implémenter l'algorithme Apriori ?

Étape 01 : Installation du module « apyori »

```
!pip install apyori
```

Nous aurons la confirmation de l'installation par les messages suivants :

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/

Collecting apyori

Downloading apyori-1.1.2.tar.gz (8.6 kB)

Preparing metadata (setup.py) ... done

Building wheels for collected packages: apyori

Building wheel for apyori (setup.py) ... done

Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5956 sha256=15191b8f8e16017032cbbd1460b1e8c649271d405f3c8730fa2b03cb23dbeb8c

Stored in directory: /root/.cache/pip/wheels/c4/1a/79/20f55c470a50bb3702a8cb7c94d8ada15573538c7f4baebe2d

Successfully built apyori

Installing collected packages: apyori

Successfullyinstalled apyori-1.1.2

Étape 2 : Importer les bibliothèques

```
importnumpyasnp
import pandas aspd
fromapyoriimport apriori
```

Étape 3 : Importation de l'ensemble de données

```
Donnees = pd.read_csv("/content/drive/MyDrive/RegAssociatio/Transactions4.csv", delimiter=";", header=None)
print(Donnees)
```

	0	1	2	3	4	5	6	7
0	Viande	Jus	Poulet	Cafe	Pain	NaN	NaN	NaN
1	Jus	Poulet	Frites	Gateau	Confiture	Oeux	The	NaN
2	Jus	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Viande	Frite	Cafe	Poulet	Steak	NaN	NaN	NaN
4	Confiture	Viande	Frites	Jus	Poulet	Oeux	Pain	NaN
5	Confiture	Jus	Frites	Steak	Cafe	Gateau	The	Chocolat

Table4.1 table de données 1

Etape 4 :Prétraitement des données

Avant d'effectuer une analyse du panier d'achat, il est important de nettoyer et de prétraiter les données de transaction pour s'assurer qu'elles sont dans un format approprié, avant d'appliquer l'algorithme

La bibliothèque Apriori que nous utilisons nécessite que notre ensemble de données soit sous la forme d'une liste imbriquée, où l'ensemble de données entier est une grande liste et chaque transaction dans l'ensemble de données est une liste interne dans la grande liste externe.

Actuellement, les données se présentent sous la forme d'une trame de données pandas. Nous devons donc maintenant convertir notre Pandas DataFrame en une liste de listes.

```
enregistrements = []

Lignes = []
foriinrange(0,NLignes):
  Lignes.append([str(Donnees.values[i,j]) for j inrange(0,NColones)])
```

- avant supp des champs null

[['Viande', 'Jus', 'Poulet', 'Cafe', 'Pain', 'nan', 'nan', 'nan'], ['Jus', 'Poulet', 'Frites', 'Gateau', 'Confiture', 'Oeux', 'The', 'nan'], ['Jus', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan', 'nan'], ['Viande', 'Frite', 'Cafe', 'Poulet', 'Steak', 'nan', 'nan', 'nan'], ['Confiture', 'Viande', 'Frites', 'Jus', 'Poulet', 'Oeux', 'Pain', 'nan'], ['Confiture', 'Jus', 'Frites', 'Steak', 'Cafe', 'Gateau', 'The', 'Chocolat']]

Après suppression des champs nuls, nous obtenons les transactions suivantes :

[['Viande', 'Jus', 'Poulet', 'Cafe', 'Pain'], ['Jus', 'Poulet', 'Frites', 'Gateau', 'Confiture', 'Oeux', 'The'], ['Jus'], ['Viande', 'Frite', 'Cafe', 'Poulet', 'Steak'], ['Confiture', 'Viande', 'Frites', 'Jus', 'Poulet', 'Oeux', 'Pain'], ['Confiture', 'Jus', 'Frites', 'Steak', 'Cafe', 'Gateau', 'The', 'Chocolat']]

Étape 05 : application d'apriori

Pour fonctionner, la classe apriori a besoin de certains paramètres.

- premier paramètre : liste de la liste dont on veut extraire les règles
- deuxième paramètre : le paramètre min_support. Il est utilisé pour sélectionner les éléments dont la valeur de support est supérieure à la valeur spécifiée par le paramètre.
- Troisième paramètre : le paramètre min_confidence permet de filtrer les règles dont la confiance est supérieure au seuil de confiance spécifié par l'utlisateur.
- Quatrième paramètre : le paramètre min_lift spécifie la valeur d'élévation minimale pour les règles présélectionnées.
- Cinquième paramètre : max_length le paramètre spécifie le nombre maximum d'éléments que vous voulez dans vos règles.

Ces derniers 4 paramètres pour l'exemple ci-après sont respectivement, 0.16, 0.2, 3, et 2.

```
ReglesdAssociation = apriori(Lignes, min_support=0.16, min_confidence=0.2, min_lift=3, max_length=2)
```

Étape 06 : Afficher les résultats

Avant de visualiser les règles, nous devons stocker les règles dans la variable de résultat. Préparez ensuite correctement les règles à afficher.

```
Regles = list(ReglesdAssociation)

        # Affichage des resultats
        forregleinRegles:
            elements = regle[0]
    forregleinRegles:
                    elements = regle[0]
                    # éléments contient la suite des éléments définissant les associations
                    items = [x for x in elements]
```

```
print()
```

Comme dernière étape, nous pouvons imprimer et vérifier les règles conscientes.

Règle: Chocolat avec Gâteau

Support: 0.16666666666666666

Confiance: 1.0

Lift: 3.0

Règle: Chocolat avec Steak

Support: 0.16666666666666666

Confiance: 1.0

Lift: 3.0

Règle: Chocolat avec The

Support: 0.16666666666666666

Confiance: 1.0

Lift: 3.0

Règle: Steak avec Frite

Support: 0.16666666666666666

Confiance: 1.0

Lift: 3.0

Règle: The avec Gâteau

Support: 0.3333333333333333

Confiance: 1.0

Lift: 3.0

Un autre exemple

	0	1	2	3	4	5	6	7
0	pain	beurre	poulet	olive	NaN	NaN	NaN	NaN
1	the	sucre	pain	confiture	poulet	olive	steak	katchap
2	olive	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	pain	beurre	confiture	NaN	the	sucre	NaN	NaN
4	steak	katchap	poulet	olive	NaN	NaN	NaN	NaN
5	the	sucre	NaN	NaN	NaN	NaN	NaN	NaN
6	poulet	olive	the	sucre	steak	katchap	NaN	NaN
7	steak	katchap	pain	beurre	confiture	NaN	the	sucre

Table4.2 table de données 2

On a obtenu les résultats suivants :

Règle: pain avec beurre
Support: 0.3333333333333333
Confiance: 1.0
Lift: 3.0

Règle: pain avec confiture
Support: 0.16666666666666666
Confiance: 0.5
Lift: 3.0

Règle: pain avec ketchup
Support: 0.16666666666666666
Confiance: 0.5
Lift: 3.0

Règle: steak avec ketchup
Support: 0.3333333333333333

Confiance: 1.0

Lift: 3.0

Règle: pain avec steak

Support: 0.16666666666666666

Confiance: 1.0

Lift: 3.0

Dans les deux exemples, nous avons fait une analyse du panier de la ménagère réalisée sur un ensemble de données de transactions de vente. Les résultats de cette analyse peuvent fournir des informations précieuses pour les processus de prise de décision et les stratégies de marketing basées sur les données.

- ✓ Cette analyse permet de déterminer quels sont les articles les plus fréquemment vendus, ce qui peut permettre à l'entreprise d'optimiser ses opérations de gestion des stocks, ses promotions et ses placements de produits.
- ✓ L'analyse permet de repérer les produits qui sont le plus fréquemment achetés ensemble, ce qui peut permettre à L'entreprise de regrouper ces produits et de proposer des remises ou des promotions afin de motiver les clients à acheter davantage.
- ✓ L'analyse peut servir à classer les clients en fonction de leurs habitudes d'achat, ce qui peut aider à mettre en place des stratégies de marketing plus ciblées et à personnaliser l'expérience des clients.

4.5 Conclusion

Nous avons présenté dans ce dernier chapitre de notre mémoire, la mise en œuvre de notre système pour l'extraction des règles d'association à partir d'un ensemble de transactions issu d'une base de données. Nous avons analysé le problème du panier de la ménagère pour bien illustrer l'extraction des règles, et montrer comment l'algorithme a été implémenté et testé.

Nous avons opté pour le langage de programmation Python, qui est le langage par excellence pour coder les problèmes en « data sciences », et nous avons utilisé l'environnement de développement en ligne « Google Colab », et ce pour sa portabilité, car ne nécessitant aucun outil installé en local, en plus de ses garanties en sécurité.

Conclusion générale

Conclusion générale :

Dans ce mémoire de master, nous nous intéressées au problème d'extraction des règles d'association à partir d'un ensemble de transactions, issu d'une base de données relationnelle. Cette problématique relève du domaine du data mining, qui s'intéresse à la couverte de modèles et de relations dans les grandes ensembles de données, telles que emmagasinées dans les bases de données des grandes entreprises. Notre objectif était l'étude des algorithmes d'extraction des règles d'associations qui sont publiés dans la littérature et d'en choisir un pour l'implémenter et l'utiliser dans le but de proposer des recommandations dans une application de ventes d'articles ménagers.

Après l'étude du domaine du data minig et des algorithmes d'extraction des règles d'association, nous avons opté pour l'algorithme Apriori qui est un célèbre algorithme pour la communauté du domaine. Notre contribution, consistant à modéliser en utilisant le langage UML un système d'extraction des règles d'association d'une base de données relationnelle, puis d'en extraire les recommandations d'achat pour l'application du panier de la ménagère.

En termes de modélisation, nous avons présenté les diagrammes UML les plus significatifs pour notre application, à savoir le diagramme de cas d'utilisation, les diagrammes de séquences et le diagramme de classe.

En termes d'application, nous avons implémenté l'algorithme Apriori sur un ensemble de transactions que nous extrairons à partir de la table des ventes dans un magasin. Le paramétrage de l'algorithme nous permet de sélectionner que les règles les plus pertinentes et qui consisterons aux recommandations adressées aux clients du magasin.

Pour ce faire, nous avons utilisé le langage Python, langage d'excellence du domaine du data mining, et des sciences de données en général. Ce langage nous a permet de facilement manipulé les données du problème, exécuter le module Apriori et structurer l'affichage des recommandations. Nous avons utilisé Python sous l'environnement de développement en ligne Google Colab, et ce pour sa portabilité, sa flexibilité et ses garantie de termes de sécurité, car ne nécessitant aucune installation de codes exécutables.

En perspective à notre travail, nous comptons implémenter d'autres algorithmes et de comparer les résultats, ainsi que de tester l'algorithme Apriori, et les autres

algorithmes sur des bases de données avancées, telles que les bases de données No sql.

Bibliographie

[1] Ait-Mlouk, A. (2018). Fouille de données et analyse de qualité des règles d'association dans les bases de données massives: Application dans le domaine de la sécurité routière (Doctoral dissertation, Université Cadi Ayyad Marrakech (Maroc)).

[2] Fayyad, U., Piatetsky-Shapiro, G., &Smyth, P. (1996). From data mining to knowledgediscovery in databases. AI magazine, 17(3), 37-37.

[3] Nedjma, B., & Benlaiter, B. (2012). Extraction des règles d'association pour l'aide à la décision étude de cas : Pharmacie (Doctoral dissertation, Universite Mohamed Boudiaf : Faculté des mathématiques et de l'informatique :Département d'Informatique.

[4] Ltifi, H. (2011). Démarche centrée utilisateur pour la conception de SIAD basés sur un processus d'Extraction de Connaissances à partir de Données, Application à la lutte contre les infections nosocomiales (Doctoral dissertation, Université de Valenciennes et du Hainaut-Cambresis; Ecole Nationale d'Ingénieurs de Sfax).

[5] Raheel, S. (2010). L'apprentissage artificiel pour la fouille de données multilingues: application à la classification automatique des documents arabes (Doctoral dissertation, Lyon 2).

[6] Agrawal, R., Imieliński, T., &Swami, A. (1993, June). Mining association rulesbetween sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conferenceon Management of data (pp. 207-216).

[7] Gurney, K. (1997). An introduction to neural networks. CRC press.

[8] Bayardo Jr, R. J., & Agrawal, R. (1999, August). Mining the most interesting rules. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 145-154).

[9] Brin, S., Motwani, R., Ullman, J. D., &Tsur, S. (1997, June). Dynamic itemset counting and implication rules for market basket data. In Proceedings of the 1997 ACM SIGMOD international conference on Management of data (pp. 255-264).

[10] Boulanger, A. (2009). Génération des règles d'association : treillis de concepts denses/mémoireprésenté comme exigence partielle de la maîtrise en informatique par Alain Boulanger ; [directeur de recherche, Robert Godin].

[11] Pasquier, N. (2000, May). Extraction de Bases pour les Règles d'Association à partir des Itemsets Fermés Fréquents. In Inforsid'2000 Congress (pp. 56-77).

[12] Pasquier, N. (2000). Data mining: algorithmes d'extraction et de réduction des règles d'association dans les bases de données (Doctoral dissertation, Université Blaise Pascal-Clermont-Ferrand II).

[13] Fiolet, V. Algorithme distribues d'extraction de connaissances. 2006.

[14] Achouri, A. (2012). Extraction de relations d'associations maximales dans les textes : représentation graphique (Doctoral dissertation, Université du Québec à Trois-Rivières).

[15]Pagé, C. (2008). Bases de règles multi-niveaux/mémoire présenté comme exigence partielle de la maîtrise en informatique par Christian Pagé;[directeur de recherche, Robert Godin].

[16] Nouasria, A. (2016). Extraction d'associations lexicales fortes dans les commentaires (Doctoral dissertation, Université du Québec à Trois-Rivières).

[17]Laurent AUDIBERT.UML 2.0 (Institut Universitaire de Technologie de Villetaneuse).

[18] Roques, P., & Vallée, F. (2004). UML 2 en action. De l'analyse des besoins ı*** la conception J2EE, 15.

Webographie

[19] https://analytics.fr/definitions/data-mining/#le-data-mining-crsquoest-quoi

[20]http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Itemset_Mining.pdf

[21] https://fr.wikipedia.org/wiki/Python_(langage).

[22] https://www.w3schools.com/python/python_intro.asp

[23] https://www.w3schools.com/python/python_intro.asp

[24] https://ledatascientist.com/google-colab-le-guide-ultime/

[25] https://geekflare.com/fr/google-colab/

[26]https://colab.research.google.com/#scrollTo=5fCEDCU_qrC0