République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 août 1955 - Skikda

Projet de fin d'études pour l'obtention du diplôme de

Master en Informatique

Option : Génie logiciel avancé et application

Thème :

Le filtrage collaboratif par modèle pour la recommandation de traitement médical

Présenté par :

Bouhafer Ines

Bouderoua Ikram

Devant le jury composé de :

Président : *SEGHIRI. R*

Examinateur : *NABET. A*

Encadreur : *Dr. MANSOUL.A*

-Session Juin 2023 – 2024 –

Remerciements

Tout d'abord, nous tenons à remercier Allah, le tout puissant de nous avoir. Dont né la santé, la volonté et la patience pour réaliser ce modeste travail.

Nous tenons tout d'abord à remercier notre encadrant, ***Dr. Mansoul.*** *A que dieu le protège, et lui accorde longue vie, Pour son aide son aide.et son observation objective a réalisation cette thèse.*

Nos sincères remerciements à l'étudiant du département de communication de masse ***Rami Ben Jamaa****.*

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

Nous adressons nos remerciements aux membres du jury, devant qui, nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu.

Nous réservons ici une place particulière pour remercier vivement tous ceux qui, d'une manière ou d'une autre, nous ont aidés et encouragés à la réalisation de ce modeste travail.

INES ET IKRAM.

Dédicaces

Je commence par remercier dieu le tout puissant et lui rendre grâce.

C'est avec un grand plaisir que je dédie ce modeste travail, fruit de

mes études en expriment ma profonde gratitude à tous mes parents et proches.

Je dédie ce mémoire avec plaisir À mon père ***Kamal*** *et ma mère* ***Zohra*** *en signe de reconnaissance pour leur soutien tout au long de mes études et leurs innombrables sacrifices en leur souhaitant une longue vie.*

À mes frères, mon collège : ***BOUHAFER YASSER, CHAKER, NOUDJOUD, FAROUK, TAKWA*** *et tous les membres de ma famille pour qui je souhaite tout le bonheur.*

À tous mes amis qui ont toujours fait preuve d'un esprit de collaboration et de serviabilité.

BOUHAFER INES

Dédicaces

Tout d'abord, je remercie Allah qui me facilite mon chemin jusqu'à l'arrivée à réaliser ce modeste travail et qui sans lui je ne peux rien faire.

Je dédie ce modeste travail et ma profonde gratitude à mon père ***Brahim*** *et ma mère* ***Zineb***.

À mes chères sœurs ***: SAFIA, CHAFIA, INES.***

À mes frères ***: DJAMEL, MOHAMED, MEZIANE, AYOUB***

À toute ma famille.

Que toute personne m'ayant aidé de près ou de loin, trouve ici l'expression de ma reconnaissance.

Finalement à tous ceux qui nous portent dans leurs cœurs

BOUDEROUA IKRAM

Résumé

Les progrès technologiques ont fait exploser la quantité d'informations médicales recueillies à chaque instant. Si bien qu'il est devenu difficile de savoir quelles sont les données à rechercher et où les trouver. Dans le cadre de ce mémoire de master, nous nous intéressons à élaborer un système de recommandation de traitement médical.

Parmi les techniques de recommandation, le filtrage collaboratif est la méthode la plus importante et la plus utilisée et les données médicale extrait de l'adresse web **www.kaggle.com.**

Ce travail consiste à étudier l'une technique ou l'on traite des données.

Puis extrait les motifs fréquents pour générer les règles d'association entre les concepts avec leurs supports et leurs confiances à l'aide de l'algorithme APRIORI.

Nous avons finalisé ce mémoire par l'implémentation de cet algorithme, l'évaluation des exemples et la discussion des résultats.

Mots-clés : fouille de donnes, Système de recommandation, Filtrage, Règles d'association, Préparation de données, Algorithme Apriori, support confiance.

ملخص

لقد أدى التقدم التكنولوجي إلى زيادة كمية المعلومات الطبية التي يتم جمعها في أي لحظة. لدرجة أنه أصبح من الصعب معرفة البيانات التي يجب البحث عنها ومكان العثور عليها. كجزء من رسالة الماجستير هذه، نحن مهتمون بتطوير نظام توصيات العلاج الطبي.

و من بين تقنيات التوصية، تعتبر التصفية التعاونية هي الطريقة الأكثر أهمية والأكثر استخداما و البيانات الطبية المستخرجة من عنوان الويب **www.kaggle.com**

يتكون عملنا من دراسة تقنية حيث نقوم بمعالجة البيانات، ثم استخراج الأنماط المتكررة لإنشاء قواعد الارتباط بين المفاهيم مع دعمها وثقتها باستخدام خوارزمية ابر يوري.

لقد أنهينا هذه الرسالة بتطبيق هذه الخوارزمية, وتقييم الأمثلة ومناقشة النتائج.

الكلمات المفتاحية: التنقيب عن البيانات، نظام التوصيات، التصفية، قواعد الارتباط، إعداد البيانات، خوارزمية أبر يوري، دعم الثقة.

Abstract

Technological advances have exploded the amount of medical information collected at any given moment. So much so that it has become difficult to know what data to look for and where to find it. As part of this master's thesis, we are interested in developing a medical treatment recommendation system.

.Among the recommendation techniques, collaborative filtering is the most important and most used method and the dataset extracted from the web address **www.kaggle.com.**

Our work consists of studying a technique where we process data, then extract frequent patterns to generate association rules between concepts with their supports and their confidences using the APRIORI algorithm.

We finalized this dissertation by implementing this algorithm, evaluating the examples and discussing the results.

Keywords: data mining, Recommendation system, Filtering, Association rules, Data preparation, Apriori algorithm, trust support.

Table de matières

Listes figures

Liste des tableaux

Introduction Générale

Problématique :

Les systèmes de recommandation dans le domaine médical sont utilisés pour suggérer des traitements, diagnostiquer des maladies, ou personnaliser des soins en fonction des données patientes. Bien que ces systèmes aient un potentiel énorme pour améliorer les soins de santé, ils présentent également des défis et des problématiques spécifiques.

Un des domaines de recherche principaux, relatifs à la problématique de la surcharge d'information est le domaine de la recherche d'information. Le principe général est d'élaborer des méthodes et des algorithmes afin de rechercher des ressources en fonction de requêtes formulées par des utilisateurs. Il n'est cependant pas toujours évident pour un utilisateur de savoir comment exprimer sa demande. De plus, sa requête correspond généralement à une quantité importante de ressources et il est difficile de savoir quels résultats lui présenter en premier, d'autant plus que d'un utilisateur à un autre, l'ordre de priorité peut changer.

Un autre domaine de recherche relatif à cette problématique est le domaine des systèmes de recommandation. Ces systèmes sont capables de fournir des recommandations adaptées aux préférences et aux besoins des utilisateurs. Ils se sont avérés être très satisfaisants pour aider les utilisateurs à accéder aux ressources désirées dans un temps limité. Initialement conçus pour la recommandation de ressources web, films, etc. les systèmes de recommandation sont devenus de plus en plus populaires et sont aujourd'hui un composant principal de beaucoup d'applications dans différents domaines. Un avantage très conséquent des systèmes de recommandation est que l'utilisateur n'a pas besoin de formuler de requêtes. Sa seule requête est implicite, elle peut se traduire par : "Quelles sont les ressources qui correspondent à mes préférences, mes besoins et mes contraintes ? ".

Mais le problème principal auquel sont confrontés les systèmes de recommandations est le problème de démarrage à froid, et cela parce que le système ne possède aucune information caractérisant l'utilisateur du système. Notre travail se situe dans ce contexte, notamment dans le cadre des systèmes de recommandation des documents. Nous adoptons le filtrage collaboratif dans d'autre cas. Ce type de recommandation repose en générale sur la contribution de l'utilisateur dans le système c'est- à-dire les notes et préférences attribuées par

cet utilisateur aux différents documents qu'il a consulté, l'ensemble de ces informations sont appelés profils utilisateurs.

Une des difficultés majeures est la construction de ce profil, dont la pertinence vis-à-vis des besoins/intérêts de l'utilisateur, joue un rôle important dans la qualité des recommandations produites. De ce fait, le profil utilisateur devient central dans les systèmes de recommandation, cette problématique fera objet de notre mémoire.

Objectifs de recherche :

L'objectif d'un système de recommandation par filtrage collaboratif dans le domaine médical est d'améliorer la qualité des soins et des traitements offerts aux patients en utilisant des données partagées et des retours d'expérience d'autres utilisateurs (patients, médecins, etc.).

L'organisation de mémoire:

Ce document est divisé en quatre chapitres organisés comme suit :

- **Introduction Générale**

Cette section présente le sujet principal de ce document, à savoir l'utilisation de la fouille de données pour recommander un traitement médical, identifie l'importance de recommander un traitement et donne un aperçu des méthodes qui seront discutées.

- **Chapitre 1 : la recherche d'association**

Ce chapitre présente différentes techniques d'exploration de données à des fins de recommandation.

Il couvre les règles d'association, les arbres de décision et la classification, et examine les systèmes de recommandation existants. Un état de l'art de la recommandation est également présenté, offrant un aperçu des avancées et des défis dans ce domaine.

- **Chapitre** 2 **: Approche de la recommandation par filtrage collaboratif**

Dans ce chapitre, une approche spécifique utilisant des règles d'association pour la recommandation médicale est présentée.

Il décrit l'ensemble des maladies, les données utilisées et les étapes de prétraitement des données. Le chapitre aborde également le modèle de formation et le processus de recommandation, et se termine par une discussion sur l'efficacité de cette approche.

- **Chapitre** 3 **: Implémentation et Résultats**

Ce chapitre se concentre sur la mise en œuvre pratique des méthodes de recommandation discutées. Il détaille l'environnement de développement, matériel et logiciel, ainsi que les étapes de prétraitement, d'apprentissage et de recommandation.

Les résultats obtenus sont présentés et discutés, en mettant l'accent sur les métriques de performance utilisées pour évaluer le modèle. Le chapitre conclut par une analyse critique des résultats.

- **Conclusion générale et perspectives**

Les travaux présentés dans ce mémoire ont porté sur l'extraction des connaissances à partir des textes. Au cours de ce travail on a tout d'abord présenté un état de l'art qui explique brièvement le concept de Fouille de texte (Text Mining) en précisant le processus, les techniques et les domaines d'application de ce dernier. On a choisi l'un des méthodes qui est l'extraction des règles d'association.

Puis, on a présenté le concept de notre travail en expliquant le fonctionnement des algorithmes qui permettent la recherche des motifs fréquents et la génération des règles d'association valides entre les concepts à l'aide de l'algorithme Apriori.

Finalement, on a présenté l'application en exécutant notre code python de l'algorithme APRIORI. On a aussi fait des expérimentations pour extraire les règles d'association à partir des bases textuelles dans différents domaines.

Malheureusement, le temps attribué à ce travail a passé rapidement, d'où il était difficile d'enrichir notre travail et étudier d'autres approches et algorithmes. Nous proposons comme perspectives :

- Appliquer d'autres méthodes de l'extraction de connaissances à partir des textes.

- Tester d'autres algorithmes de génération des règles d'association tels que : FPGROWTH, ECLAT et CLOSE ensuite comparer les résultats avec APRIORI.

CHAPITRE 1

La recherche d'association

1. Introduction

Durant les dernières années, une croissance importante des moyens de génération et de collection des données a été remarquée. Ceci est principalement dû à l'évolution de la technologie des supports de stockage. Du fait de l'informatisation rapide des entreprises, des administrations, du commerce, des télécommunications, la quantité de données disponibles augmente très rapidement. Cependant, l'analyse et l'exploitation de ces données restent très difficiles.

Cela créé un besoin d'acquisition de nouvelles techniques et méthodes intelligentes de gestion qui permettent d'extraire des données, des informations utiles appelées connaissances. C'est ainsi qu'on a commencé à parler de la découverte de connaissances à partir de données Knolwedge data Discovery (KDD) ou encore de Data Mining ou de fouille de données.

Les systèmes de recommandation ont été utilisés afin de faire face aux problèmes de surcharge et de richesse d'informations disponibles notamment à travers le Web ou les e-services. Les systèmes de recommandation visent à proposer à un utilisateur actif une ou des recommandations d'items susceptibles de l'intéresser. Ces recommandations peuvent concerner un article à lire, un livre à commander, un film à regarder, un restaurant à choisir, etc.

Dans ce chapitre, on va faire un tour d'horizon sur le concept du fouille de données (Data mining), son historique, ses définitions de bases, ainsi que des tâches et techniques. Parmi les techniques citées dans ce chapitre, la technique d'extraire des règles d'association est la plus adaptée à notre étude de cas qui consiste à extraire les règles séquentielles qui considère l'influence séquentielle dans la recommandation de POIs.

2. Les méthodes de fouille de données pour la recommandation

Les méthodes de fouille de données pour la recommandation sont diverses et peuvent varier en fonction du contexte et des données disponibles.

2.1 Définition

Le data mining ou fouilles de donnes, est un processus d'extraction des informations utiles à partir des grandes bases de données en utilisant des technique de statistique, de

Machine Learning et de visualisations de données, pour identifier dans ces données des relations, des modèles, des tendances cachée, ces informations après peuvent être utilisées pour prendre des décisions basées sur les données **[1]**.

2.2 Historique

Au début des années 1960, le terme la fouille de données est apparue et avait une signification insultante à cette époque. La fouille de données travaille sur des méthodes d'identification des données, en plus de la force croissante des nouvelles technologies, ce qui contribuait grandement à augmenter les ensembles de données, la manipulation et la capacité de stockage.

La fouille de donnés est un processus d'application de méthodes visant à découvrir des tendances cachées. Au fil des années, plusieurs méthodes et techniques ont émergé comme les réseaux de neurones de Mac Culloch et Pitt en 1941 **[2]**, et les arbres de décision en 1943 **[2]**.

A partir de 1984 ces technologies ont été améliorées pour qu'elles puissent exploiter et découvrir des modèles de plus en plus précis. De nos jour, la fouille de donnés se présente comme un outil essentiel dans les processus décisionnels. Elle combine un ensemble de techniques statistiques qui doivent être utilisées en fonction des problèmes descriptifs ou de la prise de décision.

2.3 Les étapes du processus de Textmining

Les étapes nécessaires pour effectuer le processus de Textmining sont :

- La sélection ou collection des données : Reçue il de données textuelles non structurées à partir de différentes sources, telles que des sites web, des plateformes de médias sociaux, des articles de presse, des avis de clients, etc.
- Prétraitement de données : Prétraiter les données textuelles, nettoyé et normalisé les textes par la tokenisation et la lemmatization et enlevant des informations irrelevantes, telles que des mots vides, des caractères spéciaux.
- Représentation du texte : Convertissez les données textuelles après le pré-traitement en un format numérique tel qu'un sac de mots (bag of words) ou une matrice de ter document, pour les préparer à l'analyse.
- Exploration du texte : Explorez les données textuelles pour identifier des motifs, des relations et des insights à l'aide de spécifiquement techniques telles que

l'analyse de la fréquence des mots, (les nuages de mots) et l'analyse de sentiment.

- Analyse et visualisation du texte : Appliquer des algorithmes d'apprentissage automatique tel que la classification, et visualisez les résultats pour les rendre plus faciles à comprendre et à interpréter.

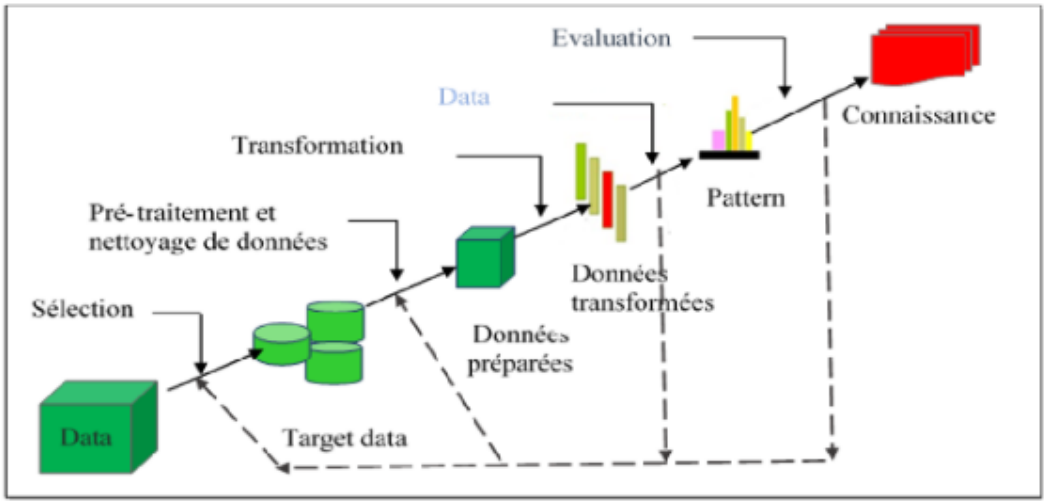


Figure 1. 1 : Le processus de l ECD Extraction des connaissances apprit des donnés [2].

2.4 Les règles d'association

2.4.1 Définition

La recherche des règles d'association est l'un des sérieux problèmes de l'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.

2.4.1.1 Item et Itemset

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

Exemple 1 : a représente un item.

Exemple 2 : c représente un item.

Un Item set est un ensemble d'items, d'objets ou d'articles d'une base de données. Exemple :{ item2, item3, item4, item6}

Un K-Item set est, ou k un ensemble de k éléments -Items, il est aussi un Itemset.

Exemple 1: {item2, item3, item4, item6} représente un 4-Itemset.

Exemple 2: {item2, item4, item6} représente un 3-Itemset

2.4.1.2 Motif

Nous définissons les motifs et les motifs fréquents en utilisant certaines définitions utilisées précédemment ([H. CHERFI, 2005], [1] Bastide, 2000] et [Pasquier et al. 1999]).Soit "T" et "D" deux ensembles et R une matrice. "T" est un ensemble de termes et "D" est un ensemble de texte.

La matrice R représente la relation binaire qui existe entre l'ensemble T et D

R	d1	d2	d3	d4	d5	d6
A	**1**	**0**	**1**	**0**	**1**	**0**
B	**0**	**1**	**1**	**1**	**1**	**1**
C	**1**	**1**	**1**	**0**	**1**	**1**
D	**1**	**0**	**0**	**0**	**0**	**0**

Tab.1.1 : Matrice Terme-Documents.

Si un motif ''t'' apparaît un nombre de fois et ce nombre est supérieur à un support minimal dans l'ensemble de textes D alors on dit qu'il est fréquent, i.e. support(t)> où min sup minimal qui est donné par l'utilisateur est le support:

2.4.1.3 Support d'une règle d'association

Le support d'une règle d'association (A→C) est une mesure de la fréquence d'apparition de la Règle, il représente le pourcentage de documents qui contiennent A et C "support (A U C)" divisé par le nombre total de document :

$$\text{Support}(A \rightarrow C) = \frac{\boldsymbol{SUPPORT(A \cup C)}}{|\boldsymbol{D}|}$$

$$\text{Support}(A \rightarrow C) \in [0, 1]$$

D : le nombre total de documents

Le support est un indicateur de fiabilité de la règle

2.4.1.4 Confiance d'une règle d'association

La confiance est une mesure permettant d'évaluer la validité d'une règle d'association. La confiance d'une règle d'association A → C, notée Confi (A → C) représente la proportion de documents qui contient A et qui contient aussi C. Elle est définie comme suit :

$$\text{Confi } (A \rightarrow C)=\frac{\boldsymbol{SUPPORT(A \cup C)}}{\boldsymbol{suport(A)}} \text{ .Confi}(A \rightarrow C) \in [0 ,1]$$

2.5 Les Algorithmes de Génération de Règles d'association

Il existe plusieurs algorithmes de génération de règles d'association. Ils utilisent suivants les notions de support et de confidence pour déterminer la pertinence des règles d'association. Parmi eux on site :

2.5.1 Algorithme Apriori :

Apriori est un algorithme classique de recherche de règles d'association introduit par Agrawal et Srikant en 1993. **[3]** C'est le premier algorithme destiné à la recherche de règles d'association. Apriori génère les motifs fréquents puis les relie entre eux pour générer les règles d'association. Il se base essentiellement sur la propriété d'anti-mono tonicité existante entre les motifs. Elle est utilisée à chaque itération de l'algorithme Apriori afin minimiser au maximum le nombre de motifs candidats à tester.

2.5.2 Le principe de l'algorithme Apriori :

La description de l'algorithme Apriori se résume dans les étapes suivantes :

- Trouver les 1-Itemsets : Parcourir la base de données pour identifier les éléments individuels et collecter les ensembles d'items ayant un support supérieur ou égal à min sup.
- Générer les (k + 1)-Itemsets : Générer des candidats pour les (k + 1) Item sets en combinant des k-Item sets fréquents en utilisant la propriété d'Apriori.
- Filtrer les candidats : Vérifier le support de chaque candidat (k + 1) Itemset et conserver uniquement ceux qui satisfont le seuil min sup.
- Répéter les étapes 2 et 3 : Itérer les étapes 2 et 3 jusqu'à ce qu'aucun nouveau k-Itemset fréquent ne puisse être trouvé.

L'algorithme Apriori explore de manière itérative les ensembles d'items de taille k en se basant sur les ensembles d'items fréquents de taille k-1 déjà trouvés. Il utilise la propriété d'Apriori, qui stipule que si un ensemble d'articles est infréquenté, tous ses ensembles supérieurs (ensembles plus larges le contenant) seront également infréquentés.

Cette propriété permet à l'algorithme de générer et de filtrer efficacement les candidats d'ensembles d'items, réduisant l'espace de recherche et améliorant l'efficacité.

En répétant le processus jusqu'à ce qu'aucun nouveau k-Items et fréquent ne puisse être trouvé, l'algorithme Apriori découvre les ensembles d'items fréquents dans la base de données.

2.5.3 Avantages et inconvénients

2.5.3.1 Avantage

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

- 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 **[4]**.

2.5.3.2 Inconvénients

Les inconvénients auxquels on fait face lors d'une utilisation de l'algorithme Apriori sont les suivants :

- 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.
- La recherche de règles d'association impose un temps considérable qui peut s'avérer désavantageux si l'on fait face à une énorme base de données.

2.6 Les arbres de décision

Les arbres de décision est un outil puissant utilisé beaucoup plus pour la classification que pour la prédiction. Ces arbres permettent de distinguer les différentes classes et de leur associer à une ou plusieurs règles. Les arbres de décision sont des outils d'aide à la décision qui permettent selon des variables discriminantes de répartir une population d'individus en groupes homogènes en fonction d'un objectif connu.

Les arbres de décisions permettent à partir des données connues sur le problème de donner des prédictions par réduction, niveau par niveau, du domaine des solutions. Chaque

nœud interne d'un arbre de décision permet de répartir les éléments à classifier de façon homogène entre ses différents fils en portant sur une variable discriminante de ces éléments.

Les branches qui représentent les liaisons entre un nœud et ses fils sont les valeurs discriminantes. Dans la littérature, plusieurs algorithmes d'induction des arbres de décision ont été proposés telle que l'algorithme CHAID développé par KASS et ID3 **[5]**, C4.5 développé par QUINLAN [5]. A titre d'illustration, Figure 1.2, présent un exemple d'un arbre de décision qui détermine si on va jouer au tennis ou non.

En commençant par le nœud racine, si les perspectives sont nuageuses alors nous devrions certainement jouer au tennis. S'il pleut, on ne devrait pas jouer tennis que si le vent est élevé. Et s'il fait beau alors on devrait jouer au tennis au cas où l'humidité serait normale.

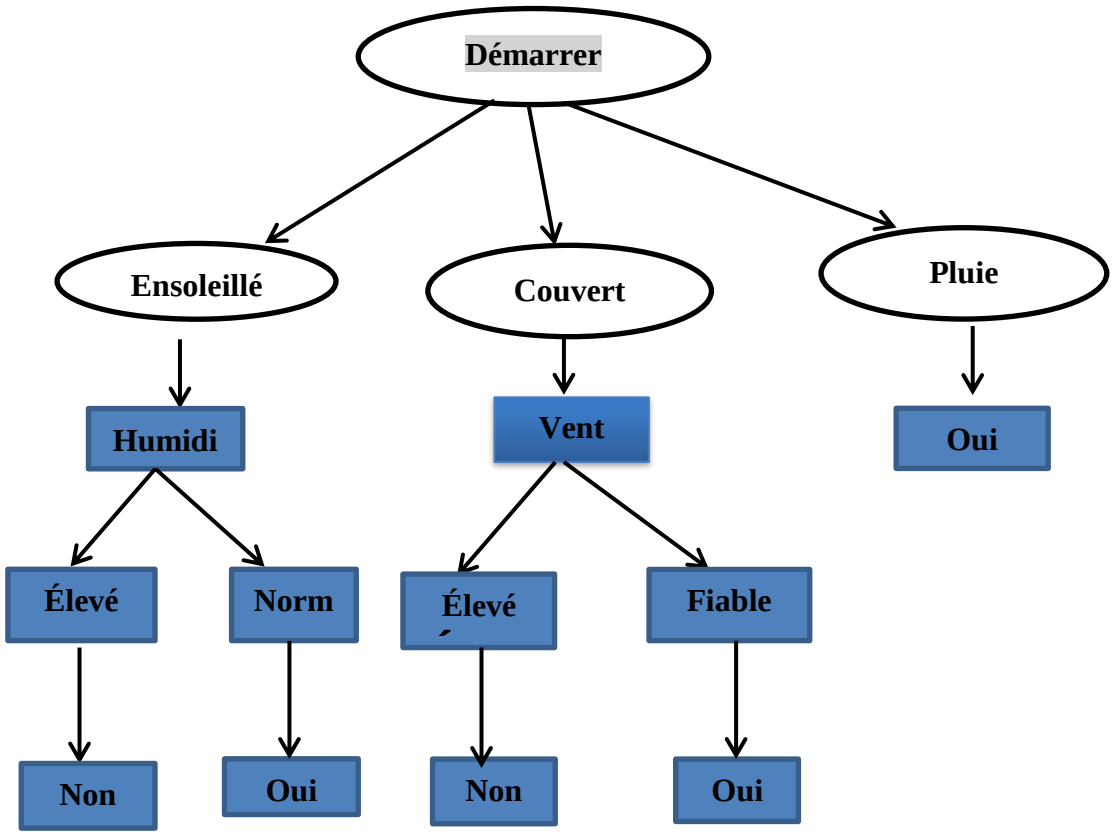


Figure 1.2 : Les arbres de décisions [28].

2.7 Les réseaux de neurones

Un réseau de neurones est un modèle de calcul dont le fonctionnement vise à simuler le fonctionnement des neurones biologiques. Un réseau neuronal est l'association, en un graphe

plus ou moins complexe, d'objets élémentaires, les neurones formels. Les principaux réseaux se distinguent par 1) l'organisation du graphe (en couches, complets, etc.), c'est-à-dire leur architecture, 2) son niveau de complexité (le nombre de neurones, présence ou non de boucles de rétroaction dans le réseau), et 3) par le type des neurones (leurs fonctions de transition ou d'activation), Figure 2.3.

Il existe deux types de réseaux :1) Réseaux à apprentissage supervisé où la réponse est connue à l'avance et 2) Réseaux à apprentissage non supervisé où le résultat n'est pas connu à l'avance.

Ces outils sont généralement utilisés pour la classification, l'estimation, la prédiction et la segmentation. Ceux-ci obtiennent de bonnes performances, en particulier, pour la reconnaissance de formes. Donc, ils sont bien adaptés pour des problèmes comprenant des variables continues éventuellement bruitées. Le principal inconvénient est qu'un réseau est défini par une architecture et un grand ensemble de paramètres réels (les coefficients synaptiques) ainsi qu'un, faible pouvoir explicatif **[6]**.

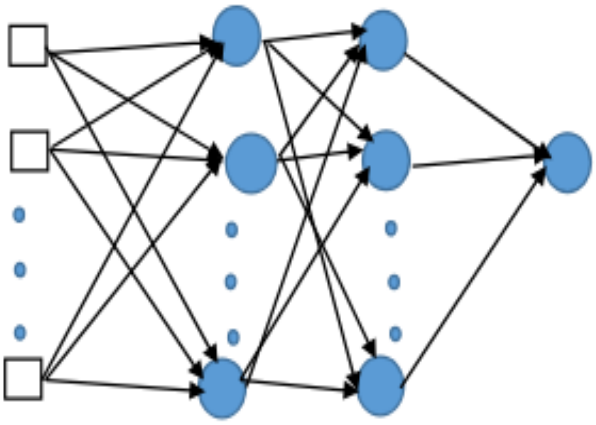


Figure 1. 3 : Les réseaux de neurones.

3. Les systèmes de recommandation

3.1 Définition

Définitions des systèmes de recommandation Les systèmes de recommandation est une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (film, musique…) qui sont susceptibles d'intéresser l'utilisateur.

Les systèmes de recommandation sont des techniques des logiciels fournissent des suggestions d'« Item » à un « Utilisateur ».Généralement ils peuvent connaitre les intérêts et les loisirs des utilisateur en fonction de leurs historiques puis prédire leurs notes ou leurs préférences pour un item donné **[7]**.

Les systèmes de recommandation jouent un rôle important dans des sites Internet aussi bien classés que Amazon1 , YouTube2 , Netflix3 , Yahoo4 , TripAdvisor5 , Last.fm6 ,IMDb7 et delicious8 pour des sites Web.

De plus, de nombreuses entreprises de médias développent et déploient actuellement des systèmes de recommandation dans le cadre des services qu'ils fournissent à leurs abonnés **[8]**.

Par exemple Netflix, le service de location de films en ligne, a décerné un prix de 1 million de dollars à l'équipe qui a réussi à améliorer sensiblement les performances de son système de recommandation.

L'entreprise a intégré les propositions les plus pertinentes dans sa version du système de recommandation mis en production.

En gros, il existe deux entités de base dans les systèmes de recommandation sont :

« Item » est le terme général utilisé pour dénoter ce que le système recommande aux utilisateurs.

« Utilisateur » est la personne qui utilise un système de recommandation, donne son avis sur les différents items et reçoit les nouvelles recommandations du système.

Un système de recommandation requiert généralement 3 étapes :

Etape 1 : Recueillir de l'information sur l'utilisateur.

Etape 2 : Construire une matrice contenant l'information recueillie.

Etape 3 : Extraire à partir de cette matrice une liste de recommandations.

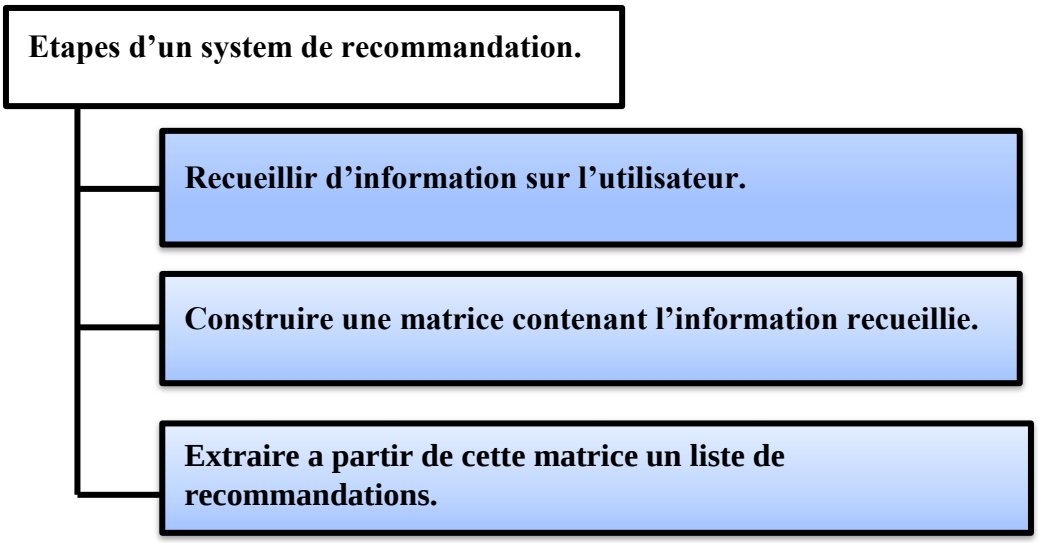


Figure 1 .4 : Etapes d’un système de recommandation [4].

3.2 Les étapes principales d'un SR

Les systèmes de recommandation sont des systèmes informatiques qui visent à prédire les préférences ou les intérêts d'un utilisateur pour des éléments tels que des produits, des services ou des informations, et à recommander les éléments les plus pertinents. Voici les étapes principales impliquées dans la construction et le fonctionnement de ces systèmes :

3.2.1 Collecte de données

Cette étape consiste à collecter des données sur les utilisateurs, les éléments et leurs interactions. Ces données peuvent inclure des historiques d'achats, des évaluations, des clics, des avis, etc.

3.2.2 Prétraitement des données

Les données collectées peuvent être brutes et contenir des erreurs ou des informations inutiles. Le prétraitement des données implique le nettoyage, la normalisation et éventuellement la réduction de dimensionnalité pour préparer les données à être utilisées dans le système de recommandation.

3.2.3 Choix du modèle

Il existe différents types de modèles de systèmes de recommandation, notamment les systèmes de filtrage collaboratif, les systèmes basés sur le contenu, les systèmes hybrides, etc.

Le choix du modèle dépend souvent de la nature des données et des objectifs spécifiques du système de recommandation.

3.3 Types de donnés de système de recommandation

Nous distinguons deux types de collecte des données : collecte explicite et collecte implicite :

3.3.1 Collecte explicite

Les utilisateurs sont sollicités pour émettre leurs avis sur des items via un système de notation (ex. une grille de 5 étoiles, un questionnaire de satisfaction), ou bien en publiant leurs avis sur un élément donné (ex. La fonction “J’aime” sur le réseau social Facebook9).

3.3.2 Collecte Implicite

La collecte implicite repose sur les interactions des utilisateurs sur le système. Par exemple le nombre de visites sur une page, le nombre de vues sur une vidéo, le temps passé sur une section donnée.

3.4 Objectifs d'un SR

- Les objectifs des systèmes de recommandation sont :
- Améliorer l’expérience de l’utilisateur.
- Augmenter continuellement les performances clés (durée de visionnement, temps de lecture, panier moyen, raccourcissement des délais de recherche de contenus/produits, etc.
- Gérer un volume croissant de données impossible à traiter manuellement.
- Analyser rigoureusement les données pour des recommandations personnalisées pertinentes.

3.5 Quelques systèmes de recommandations

Afin d’aider l’utilisateur ou l’utilisatrice à trouver des informations pertinentes, nous trouvons de nombreux systèmes dans des différents domaines que nous utilisons au quotidien, parmi ces systèmes, nous citons, Netflix et Spotify. Nous proposons une brève description de ces deux systèmes dans ce qui suit **[9]**.

3.5.1 Netflix

Est une Application de streaming basé sur la recommandation par contenu grâce aux algorithmes qu'elle a implémentée, l'utilisateur sur Netflix est plutôt passif car le choix du prochain film ou série sont déterminé par l'algorithme lui-même.

La recommandation du prochain film ou série est influencée par le contenu déjà visionné, Netflix fait ce qu'on appelle une analyse de contenu et une analyse comportementale qui est liée à un certain nombre de données :

- Le temps de lecture pour une série, afin de déterminer votre niveau d'appréciation ;
- Celles que vous avez abandonnées, et au bout de combien de temps ;
- Celle que vous avez regardée avant et les suivantes en tentant de trouver une corrélation ;
- L'heure à laquelle vous avez regardé ce contenu ;

Ce que vous avez regardé la semaine précédente, le mois précédent, l'année précédente à la même époque (1).

Figure 1.5 : Logo de l'application Netflix [21].

3.5.2 Spotify

Est une plateforme de streaming audio et de médias suédoise fondée en 2006, leur système repose sur les trois caractéristiques.

Filtrage collaboratif : utilise le comportement d'un utilisateur et celui d'utilisateurs similaires.

Traitement du langage naturel (NLP): pour les paroles de chansons, les listes de lecture, les articles de blog, les commentaires sur les réseaux sociaux.

Modèles audio : utilisés sur l'audio brut.

En résumé Spotify utilise des algorithmes d'apprentissage en profondeur à fin de profiler l'utilisateur et lui recommander des chansons qui peuvent l'intéresser basé sur la music qu'il a déjà écouté.

Figure 1 .6 : Logo de l'application Spotify [22].

3.6 Concepts de base, notation et notions liées

Nous définissons dans cette partie quelques concepts relatifs aux systèmes de recommandation, qui seront utilisés par la suite.

3.6.1 L'usager et l'item

Les deux entités de base qui apparaissent dans tous les systèmes de recommandations sont l'usager et l'item.

L'« usager » est la personne qui utilise un système de recommandation, donne son opinion sur diverses items et reçoit les nouvelle recommandations du système.

L'« Item » est le terme général utilisé pour désigner ce que le système recommande aux usagers.

3.6.2 Evaluation (note ou vote)

Une évaluation est une valeur numérique dans une échelle quelconque (la plus utilisée est [1-5]) ou binaire (aimer\ Ne pas aimer, bon\ mauvais, etc.) qui représente la préférence ou non d'un item donné par un utilisateur. L'évaluation donné par un utilisateur u à un item i est représenté par ou par un triplé. Où, une note de 5, par exemple, exprime une grande préférence et une note de 1 indique une faible préférence i.e. l'utilisateur n'a pas aimé l'item.

Une note peut être attribuée directement par un utilisateur à un item en donnant une valeur numérique ou binaire à travers l'interface du système appelée évaluation explicite (Burk R., 2002). En outre, les préférences de l'utilisateur peuvent être déduites par le système en utilisant des algorithmes et techniques spécifiques (Rendle et al, 2009) (Lee et al, 2008), et

dans ce cas appelée évaluation implicite (Ouard et al, 1998) (Burk R., 2002) (Kelly et al, 2003).

3.6.3 Filtrage d'information

Le filtrage d'information est l'expression utilisée pour décrire une variété de processus dédiés à la fourniture de l'information adéquate aux personnes qui en ont besoin (Bel et al, 2007). Son but est de sélectionner et suggérer aux utilisateurs, à partir de larges volumes d'informations générés dynamiquement, les informations jugées pertinentes pour eux.

Par conséquent, le filtrage d'information peut être vu aussi comme étant le processus d'élimination de données indésirables sur un flux entrant, plutôt que la recherche de données spécifiques sur ce flux. Le filtrage commence donc après la définition du besoin de l'utilisateur, il permet d'éliminer les documents qui peuvent ne pas intéresser l'utilisateur. Le filtrage offre à l'utilisateur un gain d'effort et de temps.

3.6.4 Matrice d'évaluation utilisateur-item

L'ensemble de tous les triplets du système sont enregistrés dans une base de données creuse appelée Matrice d'Evaluation (Rating Matrix) ou encore Matrice utilisateur item (user-item Matrix) et elle est notée par R, où chaque ligne correspond aux évaluations fournies par un seul utilisateur et une colonne correspond aux évaluations qu'a eu un seul item par l'ensemble des utilisateurs.

La matrice d'évaluation utilisateur-item est l'entrée pour les systèmes de recommandation et la base des techniques du filtrage collaboratif, qui utilisent les préférences (votes) pour la génération des recommandations.

3.6.5 Communauté

L'ensemble d'utilisateurs similaires et partageant les mêmes centres d'intérêt, préférences et goût, peuvent être regroupés selon un critère donné, est appelé Groupe ou Communauté. [Bou, 05].

3.6.6 Recommandation

Le calcul d'une liste d'items les plus aimés / préférés par l'utilisateur est appelée recommandation. Celle-ci est faite en attribuant des scores pour les items selon leurs

popularités ou leurs préférences [Wen et al, 08], par exemple. Contrairement à la prédiction, le calcul des recommandations ne se base pas strictement sur les évaluations.

3.6.7 Notion du profil

• Profil utilisateur : c'est un portail incluant une description des caractéristiques de l'utilisateur, telles que, ses centres d'intérêt, ses données démographiques, ou bien ses préférences exprimées sous forme d'évaluations, etc. Il existe certaines approches de construction des profils utilisateurs. Les travaux des auteurs [Bur, 02], [Cho et al, 02] et [Sha et al, 97] peuvent constituer un référentiel de ces approches.

• Profil item : il s'agit ici, de décrire l'item avec un ensemble de propriétés, appelées aussi attributs [Rij, 79].

Outre les attributs d'items, on trouve aussi les métadonnées qui présentent des informations (données) sur les caractéristiques elles-mêmes [Ber, 02].

3.4 Les techniques de recommandation

Plusieurs facteurs entrent en considération afin de catégoriser les systèmes de recommandation.

- La connaissance de l'utilisateur (c.-à-d. son profil en fonction de ses goûts).
- Le positionnement d'un utilisateur par rapport aux autres (la notion de classes ou réseaux d'utilisateurs).
- La connaissance des items à recommander.
- La connaissance des différentes classes d'items à recommander.

De ces facteurs sont produits divers types de recommandations dont les plus utilisés dans la littérature sont le filtrage basé sur le contenu et le filtrage collaboratif. Ce document présente dans un premier temps ces deux approches ainsi que leur hybridation.

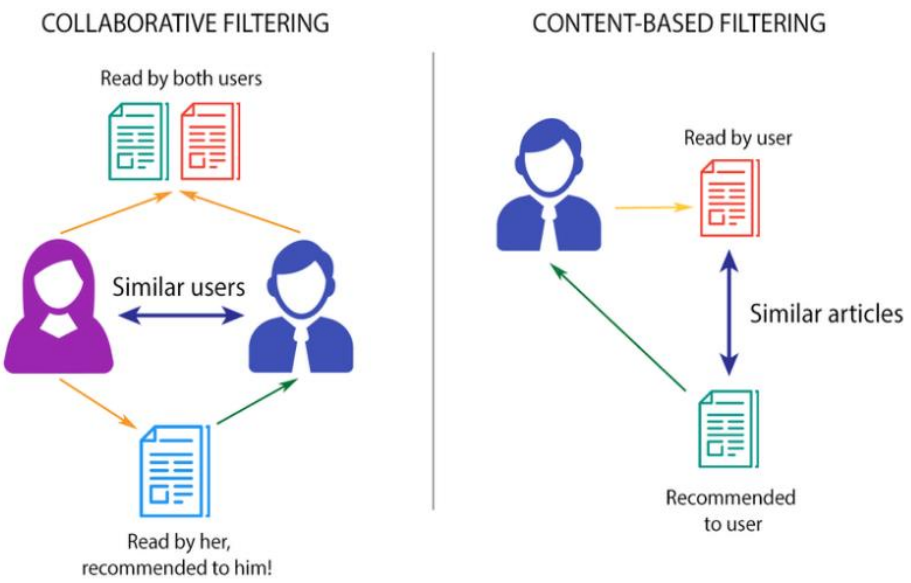


Figure 1.7 : Filtrage Collaborative / Filtrage Basé sur le Contenu [5].

4. Classification des systèmes de recommandation

Les techniques de recommandation peuvent être classées de différentes manières. Parfois plusieurs termes sont utilisés pour désigner une même méthode ou approche. La classification la plus utilisée repose sur trois types : filtrage basé sur le contenu, filtrage collaboratif et filtrage hybride (Adomavicius & Tuzhilin, 2005).

En plus de ces deux approches, Robin Burke (Burke, 2002) propose de considérer trois autres approches : la recommandation basée sur les données démographiques, la recommandation basée sur la connaissance (knowledge-based) et la recommandation basée sur l'utilité (utility-based).

Mais il note que ces trois approches sont des cas particuliers des approches classiques. L'objectif ici est de s'appuyer sur la classification la plus connue sur laquelle nous basons notre étude. Nous présentons dans la suite les approches basées contenu et le filtrage collaboratif, puis les approches hybrides (figure 1.7). Nous allons détailler la technique de filtrage collaboratif que nous allons utiliser dans notre travail.

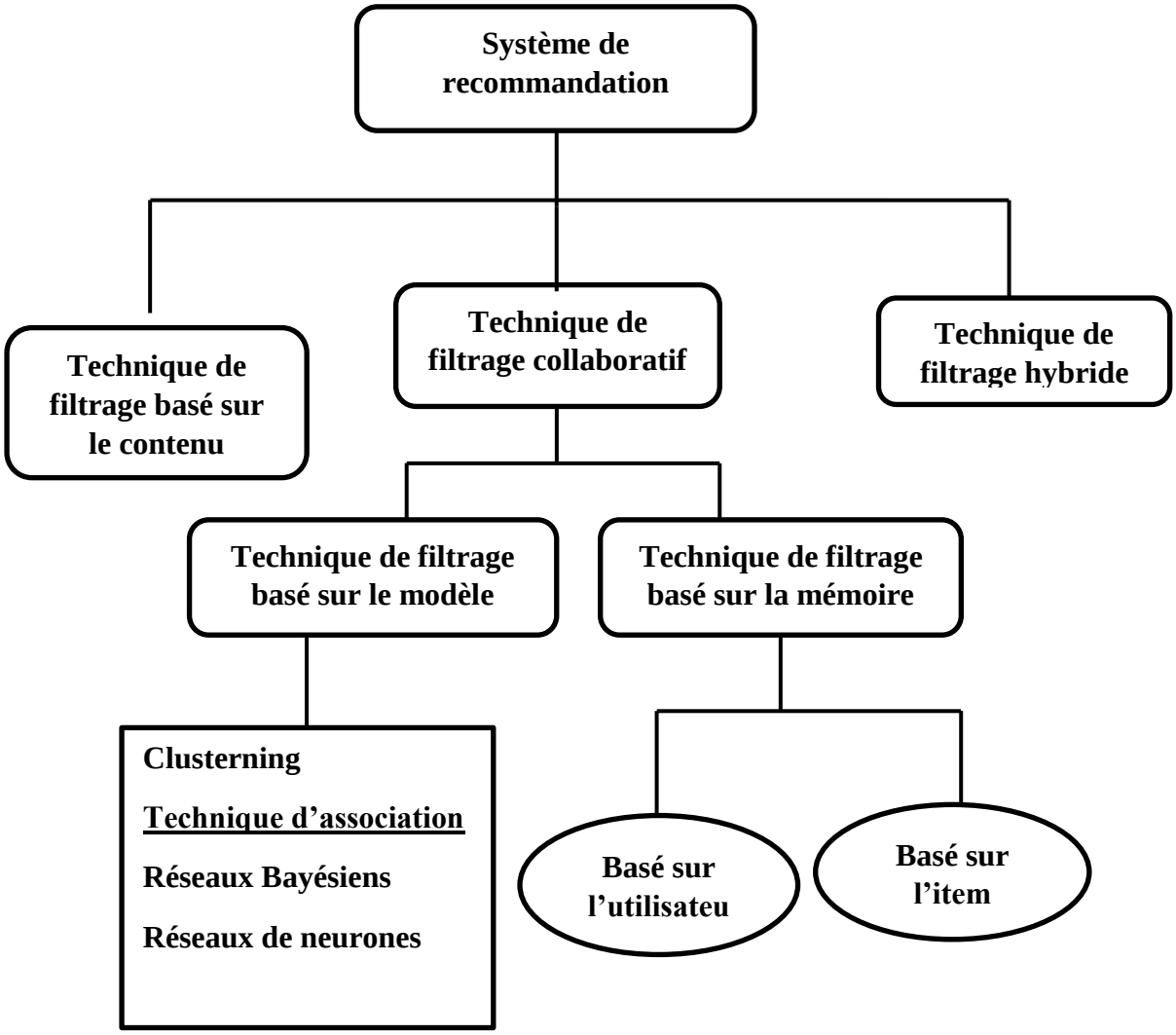


Figure 1.8 : Classification des systèmes de recommandation (Isinkaye et al. 2015) [9].

4.1 Recommandation basée sur les usages

Les systèmes de recommandation basée sur les usages calculent les recommandations en se basant sur les usages passés que les utilisateurs ont fait du système. Cette approche ne nécessite pas de considérer le contenu des ressources, ce qui présente plusieurs avantages. Le premier avantage est que cela évite l'extraction de profils de ressource et d'utilisateur. Le terme « profil » est utilisé ici dans le même sens que celui employé pour la recommandation basée sur le contenu de la section précédente.

En réalité des profils d'utilisateurs peuvent être définis dans le cadre de la recommandation basée sur les usages, mais se rapportent à une forme différente de profil. Le

second avantage est que les approches basées sur les usages ne sont pas aussi dépendantes de la nature des données que les approches basées sur le contenu.

En particulier, elles sont applicables aussi bien aux données graphiques que sonores, deux types de données pour lesquels les recommandations basées sur le contenu ont une efficacité très limitée. Parmi les critères exploitables pour effectuer des recommandations basées sur les usages, les deux critères principaux sont les appréciations et les motifs.

L'utilisation d'appréciations correspond au filtrage collaboratif, et celle des motifs à différentes approches issues du domaine de la fouille de données. Dans cette section, nous présentons les approches principales exploitant ces deux critères ; puis nous présentons les limitations de ces approches.

4.2 Basé sur le contenu

Les systèmes de recommandation basés sur le contenu (content based) fonctionnent en analysant les caractéristiques des items à recommander (lieux, produits, etc.) puis en les regroupant **[8]**. Dans ce genre de SR, des mots-clés sont utilisés afin de décrire les ressources ; un profil est ensuite créé afin d'indiquer pour chaque utilisateur le type de ressources qu'il aime.

Par exemple, nous construisons un profil d'un utilisateur A qui préfère les séries dont les genres préférés sont actions et romances (cas de Netflix). Et, nous essayons de recommander des produits qui de la même section que A préfère.

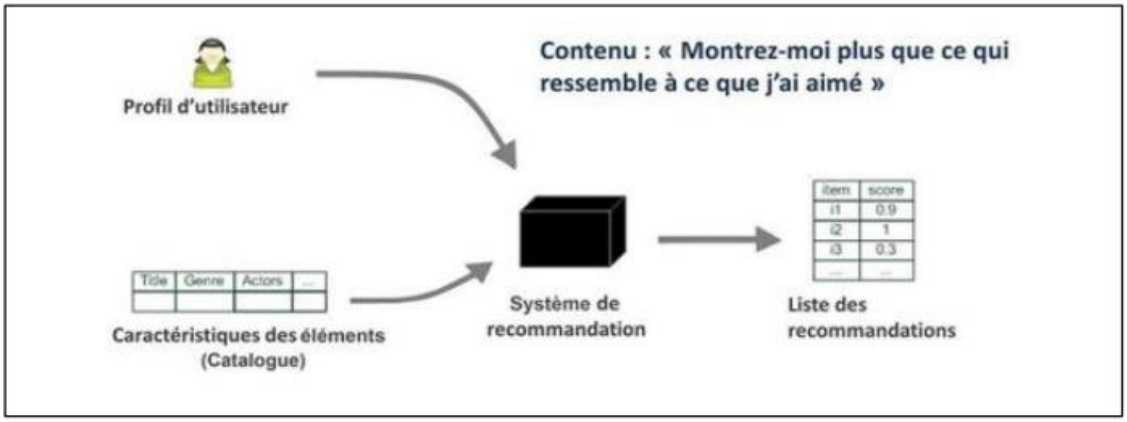


Figure 1.9 : Filtrage basé sur le contenu [5].

4.3 Filtrage collaboratif

Les méthodes basées sur le filtrage collaboratif sont basées sur la collection et l'analyse d'informations sur le comportement des utilisateurs, leurs activités et leurs

préférences pour ensuite prédire ce que les utilisateurs sont susceptibles d'aimer d'après leurs similarités avec d'autres utilisateurs **[7]**.

L'idée clef est que la note d'un utilisateur U pour un nouvel item i est susceptible d'être similaire à celle donnée par un autre utilisateur V, si U et V ont noté d'autres items d'une manière similaire. De même, U est susceptible de noter deux items i et j de la même façon, si d'autres utilisateurs ont donné des notes similaires à ces deux items. Les approches collaboratives dépassent certaines limitations des approches basées sur le contenu.

Par exemple, des items dont le contenu n'est pas défini, peuvent quand même être recommandés aux utilisateurs grâce aux feedbacks des autres utilisateurs. De plus, les recommandations collaboratives sont basées sur la qualité des items évaluée par les utilisateurs, au lieu de s'appuyer sur le contenu qui peut être un mauvais indicateur de qualité **[7]**.

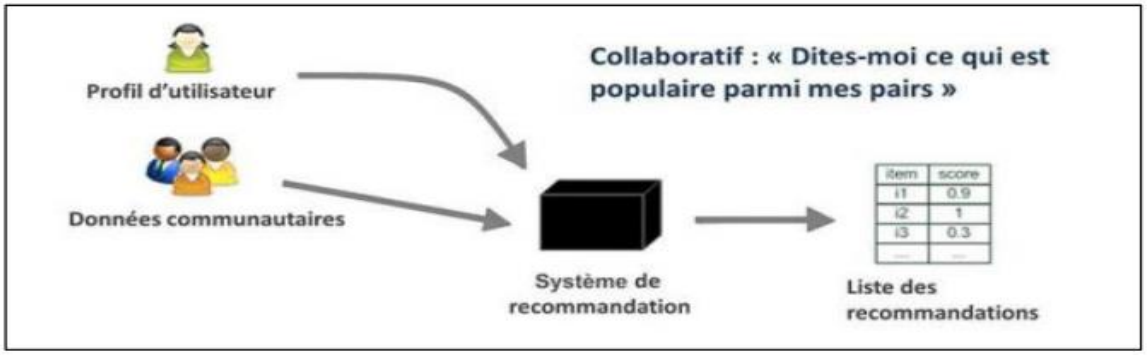


Figure 1.10 : Filtrage collaboratif [5].

Le filtrage collaboratif est composé de deux méthodes :

- FC Basé sur la mémoire
- FC basé sur le modèle

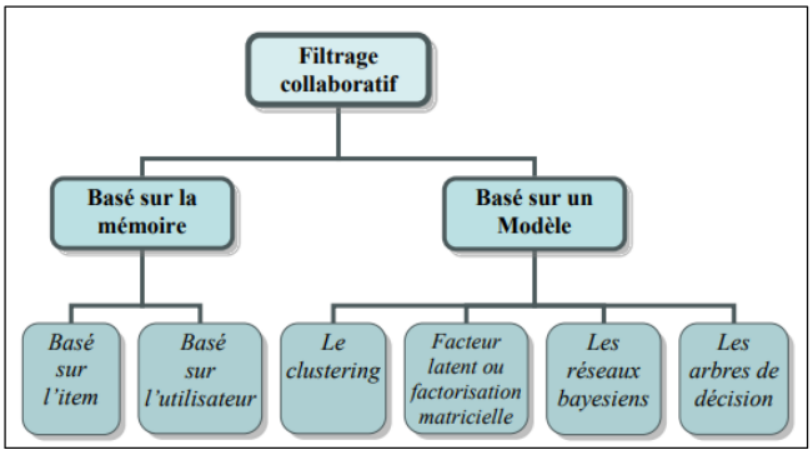


Figure 1.11 : Méthodes de filtrage collaboratif [2].

4.3.1 Filtrage collaboratif basé sur la mémoire

Cette méthode utilise un système de notation pour prédire les préférences d'un utilisateur en tenant compte des préférences d'un utilisateur similaire, ou du « voisin » **[10]**. Il existe deux façons de calculer les préférences ici, le FC basé sur l'utilisateur et le FC basé sur l'item.

Le FC à base de mémoire souffre essentiellement de deux problèmes : le passage à l'échelle et le manque de données. Le passage à l'échelle fait que le temps de calcul est énormément long lors de son application sur des données nombreuses. Le manque de données vient du fait qu'il n'est pas toujours facile de trouver assez de notes communes entre les utilisateurs, particulièrement sur des grandes listes d'items **[11]**.

- **FC basée sur l'utilisateur**

Cette approche est basée sur la similarité entre les utilisateurs. Nous construisons une matrice M [Utilisateur x Item] :

M	Item 1	Item 2	Item 3	Item 4	Item 5
Dina	3	8	9		7
Amelia	3		8		8
Rahim	2	5	7	9	

Tab 1.2 : Utilisateur x Item [7].

Dina et Amelia ont des avis similaires sur trois items (item 1, item3 et item 5) et Dina aime l'item 2 donc il sera une bonne recommandation pour Amelia.

- **FC basée sur l'item**

La similarité entre items ici est basée sur le jugement des utilisateurs. Pour que deux items soient considérés voisins (similaires), ils doivent être appréciés par les mêmes utilisateurs **[10]**. Nous construisons aussi la même matrice M. Dina et Amelia aiment l'item 3 et 5. Cela suggère que les personnes qui aiment l'item 3 aiment aussi l'item 5, donc l'item 5 pourra être recommandé à Rahim qui aime l'item 3.

4.3.2 Filtrage collaboratif basé sur le modèle

L'objectif principal du FC à base de modèle est de réduire l'impact du problème de passage à l'échelle et de celui du manque de données.

Dans cette méthode de filtrage collaboratif, différents algorithmes d'exploration de données et d'apprentissage automatique sont utilisés pour développer un modèle pour prédire l'évaluation d'un utilisateur d'un élément non estimé. Quelques exemples de ces modèles sont les réseaux bayésiens **[12]**, les modèles de cluster ING **[13]**, la factorisation matricielle **[14]** et les arbres de décision **[15]**.

Dans notre travail, nous allons nous intéresser essentiellement sur la factorisation matricielle, qui consiste à décomposer une matrice en plusieurs autres matrices, et faire le produit de ces matrices entre elles, pour retrouver à la fin la matrice originale.

4.4 Comparaison entre le filtrage collaboratif et le filtrage basés sur le contenu

Les approches de recommandation présentées ont des avantages et des inconvénients :

- Le problème principal de ces deux approches réside dans le fait qu'il faut une base d'utilisateurs ayant déjà fait des choix pour faire des recommandations.

- Le SR basé sur le contenu requiert la connaissance de l'utilisateur seulement, contrairement aux systèmes basés sur le filtrage collaboratif qui nécessite une connaissance sur tous les utilisateurs.

- Le filtrage collaboratif se distingue par la diversité des items à recommander à l'utilisateur. Par exemple, un utilisateur qui a des voisins similaires du point de vue des politiques, peut se voir recommander des articles de sport si ces voisins aiment les articles de sport, même si cet utilisateur n'a jamais exprimé ce genre de favoris. Au contraire des SR basés sur le contenu si par exemple un utilisateur ne s'intéresse qu'aux articles parlant de la médecine, il ne se verra jamais proposer un article de cuisine.

4.5 Filtrage Hybride

Un système de recommandation hybride utilise des composants de différents types d'approches de recommandation ou s'appuie sur leur logique. En général, l'hybridation s'effectue en deux phases :

- Appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates.

- Combiner ces ensembles de recommandations préliminaires selon certaines méthodes telles que la pondération, la mixtion, la cascade, la commutation, etc. afin de produire les recommandations finales pour les utilisateurs.

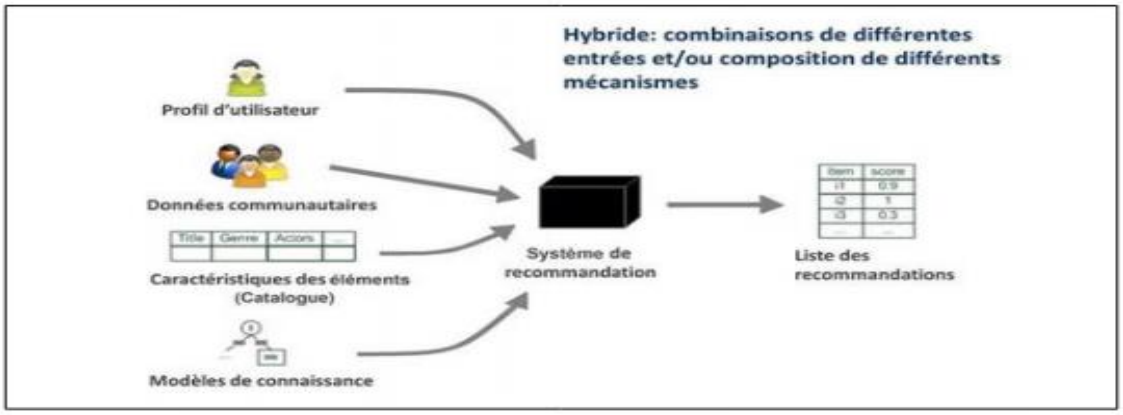


Figure 1.12 : Filtrage hybride [5].

5. Etat de l'art de la recommandation

Les systèmes de recommandation appliquent plusieurs techniques d'exploration de données telles que le filtrage collaboratif, le filtrage basé sur le contenu, les techniques hybrides ou les méthodes basées sur les connaissances en fonction des caractéristiques du domaine, de la qualité des données disponibles et des objectifs commerciaux. Dans ce chapitre nous allons dénombrer un certain nombre de travaux connexes aux systèmes de recommandation.

5.1 Les travaux connexes aux systèmes de recommandation « Information Lens System »

[Mal et al, 87] est le premier système de recommandation développé. A l'époque, l'approche la plus commune pour le problème de partage d'informations dans l'environnement de messagerie électronique était la liste de distributions basée sur les groupes d'intérêt. La technique du filtrage est utilisée dans un sens plus général qui consiste à sélectionner les choses à partir d'un ensemble plus large de possibilités (filtrage positif : sélection).

Tapestry [Gol et al, 92] est parmi les premiers systèmes à émettre l'hypothèse qu'une intervention humaine pourrait améliorer les résultats de filtrage d'items, des e-mails dans leur cas. L'objectif est donc d'améliorer les résultats de filtrage d'e-mail en proposant le système de filtrage collaboratif Tapestry. Le principe, novateur à l'époque, est de s'appuyer sur les précédentes requêtes d'utilisateurs du système Tapestry et sur leurs appréciations en fonction des documents qu'ils ont jugés pertinents afin de filtrer les résultats proposés à de futurs utilisateurs qui feront la même requête.

Dans GroupLens [Res et al, 94], les auteurs proposent une approche assez similaire en se focalisant sur la recherche documentaire par le biais du système GroupLens. Notons cependant que comme lors de l'utilisation de l'outil Tapestry, GroupLens se fonde sur une appréciation humaine active d'article. Le terme d'actif signifie dans ce cas que l'utilisateur doit faire la démarche d'annoter un document afin de construire des classes de documents pertinents. Ces approches ne sont donc pas fondées sur un apprentissage implicite du comportement d'utilisateurs comme dans le cas d'Amazon [Lin et al, 03].

Les applications web ont également fait l'objet d'intégration des systèmes de recommandation. Les travaux de [Lie, 95] présentent le système Letizia qui vise à aider les utilisateurs à naviguer sur le Web. Son authenticité réside su fait qu'il soit l'un des premiers à proposer des recommandations via un système de filtrage basé sur le contenu sans intervention

explicite de l'utilisateur. Plus formellement, le système utilise des techniques statistiques pour créer un profil d'utilisateur basé sur des outils d'extraction de descripteurs ou de mots-clés. Les informations ciblées sont les liens vers les pages que les utilisateurs visitent.

Encore, dans le même système de filtrage par contenu, [Paz et al, 97] proposent d'identifier les sites pertinents des utilisateurs dans un domaine fixe, en démontrant la qualité du classificateur bayésien [Dud et Har, 73] par rapport aux autres systèmes de classification. Les données apprises sont affichées en premier dans un espace vectoriel de cooccurrences des mots selon les pages Web dans lesquels elles apparaissent. Le classificateur bayésien est utilisé pour construire des classes d'individus. Les auteurs suggèrent également de saisir des informations sémantiques à l'aide de WordNet [Mil et al, 90], ce qui augmente les performances.

Le système WebMate [Che et Syc, 98] permet à ses utilisateurs de rendre la navigation Web plus efficace, en proposant notamment un agent les guidant dans leurs recherches. La méthode utilisée est vectorielle et propose de construire, dans un premier temps, les profils utilisateurs avec un système Saltonien reposant sur une pondération de type Tf-Idf. Les mots clés représentatifs des choix des utilisateurs sont alors extraits. Des lors, l'originalité de la méthode est de réintroduire dans la requête de l'utilisateur des résultats jugés pertinents par d'autres utilisateurs proches de lui, utilisant le principe du retour de pertinence (Relevance Feed-back). Finalement, lors de la première utilisation du système par un utilisateur, son profil est alors vide et aucune recommandation ne peut lui être faite. Notons que leur système a été évalué avec les moteurs de recherche Altavista et Lycos.

L'auteur Cunningham et ses collègues [Cun et al, 00] ont proposé Websell, un système d'aide à la navigation sur le Web s'intéressant à la vente en ligne. Le principe est de produire un assistant de vente en ligne pour un utilisateur en ciblant dans un premier temps ces choix et en lui recommandant dans un second temps des produits relatifs à son profil. Le projet Websell utilise notamment un filtrage collaboratif.

[Fis et Ste, 91] ont proposé le système Infoscope afin de faire la recommandation à des utilisateurs de groupes de discutions en fonction des requêtes qu'ils soumettent. Le principe est de construire une représentation sémantique d'un utilisateur en fonction de ses préférences en s'appuyant sur une structure d'arbre afin de catégoriser les articles des NewsGroups.

[She et Mae, 93] présentent le système Newt, se basant sur le retour de pertinences d'utilisateurs (relevance feedback) et d'un algorithme de classification afin d'effectuer un filtrage personnalisé d'informations. Le retour de pertinence utilisateur a été choisi afin de permettre une mise à jour dynamique des choix de l'utilisateur. Un algorithme génétique, a été utilisé pour sa capacité de sortir des minimums locaux lors des phases de mutations.

[Moo et Roy, 99], partent de la constatation que les systèmes de recommandations collaboratifs peuvent être limités, ils ont proposé le système Libra qui est destiné à faire de la recommandation de livres à partir du site Web d'Amazon. Le processus de recommandation de Libra se déroule en plusieurs phases, telles que, la phase d'extraction d'informations, la phase d'évaluation de l'utilisateur, une phase d'apprentissage de profils et la phase de recommandation.

RARE (Recommender system based on Association RulEs) [Ben et al, 06] est un système de recommandation de cours. RARE se base sur le principe de faire profiter ses utilisateurs d'une expérience collaborative d'étudiants. Vu que le volume des données mémorisées des étudiants qui ont été enregistrés et qui ont achevé leurs études est important, des techniques de forage de données sont appliquées sur cet ensemble de données. Elles permettraient de capturer le comportement des étudiants dans leurs choix de cours et, par conséquent, révéler des relations cachées entre les cours suivis.

[Alv et al ,14] ont proposé un nouveau système de recommandation basé sur la qualité. Ce système utilise la qualité des items pour estimer leur pertinence, cette mesure de qualité est prise comme un nouveau facteur à considérer dans le processus de recommandation. Cela aide les utilisateurs à accéder aux ressources de recherche pertinentes. Ce système de recommandation est développé en utilisant une approche linguistique floue et il a été testé de manière satisfaisante dans une bibliothèque numérique universitaire.

Le système Cora [Flo et al, 20] est un système conversationnel qui recommande des recettes alignées sur les habitudes alimentaires et les préférences actuelles de ses utilisateurs. Les utilisateurs peuvent interagir avec Cora en cliquant sur des boutons ou en écrivant du texte en langage naturel. De son côté, Cora peut engager les utilisateurs via un dialogue social

ou aller droit au but. Une expérience a été menée pour évaluer l'impact du style conversationnel de Cora et du mode d'interaction avec les utilisateurs sur la qualité perçue de l'interaction et du système, et sur l'intention de l'utilisateur de cuisiner les recettes recommandées. Les résultats montrent qu'un système de recommandation conversationnel qui engage ses utilisateurs à travers un dialogue social améliore la qualité perçue de l'interaction ainsi que celle du système.

[Jar et al ,16], proposent dans leur étude une architecture d'un système de recommandation basé sur le contenu visant à sélectionner des examinateurs (experts) pour évaluer des propositions de recherche ou des articles. Ils introduisent un cadre algorithmique complet supporté par diverses techniques de recherche d'informations. Ils proposent une méthodologie complète qui explore les concepts de données, d'informations, de connaissances et de relations entre eux pour soutenir la formation d'une recommandation appropriée.

Chaque partie essentielle du système est traitée comme un module séparé, tandis que chaque couche prend en charge une certaine fonctionnalité du système. La modularité de l'architecture facilite la maintenabilité.

Entrée est un exemple de ce qu'on appelle un système FindMe [Bur et al, 97]. Les systèmes FindMe sont considérés comme des systèmes de recommandations avec les caractéristiques distinctives suivantes :

1. Ils sont principalement basés sur des connaissances,
2. Possibilité de critiquer une suggestion de la part de l'utilisateur,
3. Ils classent les produits en fonction des objectifs attendus d'utilisateurs.

Dans l'article [Ric, 79], l'auteur propose le système Grundy qui visait à l'époque la construction de profils utilisateurs sans parler nécessairement de recommandation mais plutôt de personnalisation. Le principe est de construire des profils "stéréotypé" et d'activer ces derniers en fonction du comportement d'un utilisateur, selon le type de requête qu'il formule. Cette approche reste toujours utilisable afin de construire des profils initiaux d'utilisateurs comme dans [Kob et al, 01], où les auteurs proposent une approche collaborative favorisant ainsi les relations entre clients de sites en ligne. Ce travail a été étendu par un large panel d'approches de recommandation telles que, les approches hybrides et celles fondées sur le contenu.

[Sch et al, 01] s'intéressaient à l'apprentissage de profils en se basant sur un filtrage collaboratif et une analyse du contenu du profil de l'utilisateur. A noter que cette approche peut être apparentée à une méthode hybride, utilisant une méthode de filtrage collaboratif et une méthode basée sur le contenu.

[Bas et al, 01] qui proposent une méthode à base de règle regroupant dans un même classificateur des notions basées sur le contenu et collaboratives. A noter que la partie collaborative est dans cette approche passive, les auteurs récupérant des informations sur l'utilisateur via le Web, sans interaction avec eux.

5.2 Comparaison

Nous avons tenté de procéder à une comparaison des approches des systèmes de recommandation en se basant sur quelques critères. Cette comparaison est présentée dans le tableau suivant.

Approche	**Objectifs**	**Algorithme**	**Technique**	**Sémantique**
[Gol et al ,92] Système Tapestry	Recommandation comparaison	Filtrage collaboratif filtrage l'information	Annotation	Non
[Che et Syc, 98] Système WebMate	Recommandation Sélection	Filtrage collaboratif	Relevance feed-back	Non
[Cun et al, 00] C	Choix/ Recommandation	Filtrage collaboratif	/	Oui
[Ric, 79] Système Grundy	Personnalisation	Filtrage collaboratif		

			/	
[Sch et al, 01]	Analyse	Filtrage collaboratif / filtrage basée sur le contenu	Apprentissage	Non
[Fis et Ste, 91] Système Infoscope	Recommandation	Filtrage collaboratif	/	Oui
[She et Mae, 93] SystèmeNewt	Choix	filtrage personnalisé d'informations	Relevance feedback	Non
[Lie, 95] Système Letizia	Recommandation / Choix	filtrage basée sur le contenu	Extraction de descripteurs ou de mots-clés	Non
[Bas et al, 01]	Regroupement	Méthode à base de règle / Hybride		Non
Bouhafer Ines Bouhafer Ines Bouhafer Ines [Jar et al ,16] Système des examinateurs (experts)	Recommandation / Recommandation	basé sur le contenu/ connaissance	/	/

Tab 1.3 : Comparaison entre les systèmes de recommandation [56].

5.3 Synthèse

Les systèmes collaboratifs et basés sur le contenu sont faciles à configurer car seules les informations de base sur les noms d'éléments, les descriptions et les représentations graphiques sont nécessaires.

Les méthodes de filtrage hybride (HF) peuvent être plus précises que les méthodes conventionnelles ; cependant, la mise en œuvre efficace de telles solutions peut être très difficile pour des problèmes complexes. Les deux approches, si elles ne sont pas formées avec de nombreux exemples (évaluations des éléments ou modèle de préférences des utilisateurs), fournissent de mauvais ré Cette limitation a principalement motivé une quatrième approche, basée sur les connaissances, qui tente de mieux utiliser les connaissances préexistantes du domaine d'application pour construire un modèle plus précis nécessitant moins d'instances de formation.

4. Conclusion

Dans ce chapitre, nous avons donné une définition des systèmes de recommandation en citant les différentes approches et les détaillées en donnant des exemples pour mieux comprendre la notion de ces systèmes. Après nous avons cité les problèmes de ces systèmes, tel que le manque de donnés et le passage à l'échelle.

Dans le prochain chapitre nous présentations notre résultat expérimental en appliquant l'approche proposée, qui consiste à étudier la filtration coopérative, pour conclure ce qui est le mieux**.**

CHAPITRE 2

Approche de la recommandation par filtrage collaboratif

1. Introduction

Notre travail consiste à construire un système de recommandation base sur l'hybridation mixée en utilisant un filtrage collaboratif et une recommandation basée contenu qu'ont vas détailler dans la suite du chapitre, tout en se basant sur le modèle vectoriel pour la représentation du profil utilisateur.

Nous commençons tout d'abord par la présentation de l'approche suivie dans son développement et nous schématisons son architecture. Ensuite, nous y décrivons les deux composantes principales de l'architecture proposée, et nous expliquons les rôles des différentes bases de données utilisées par le système

2. Le système de recommandations utilisées

2.1 Architecture général de system de recommandation

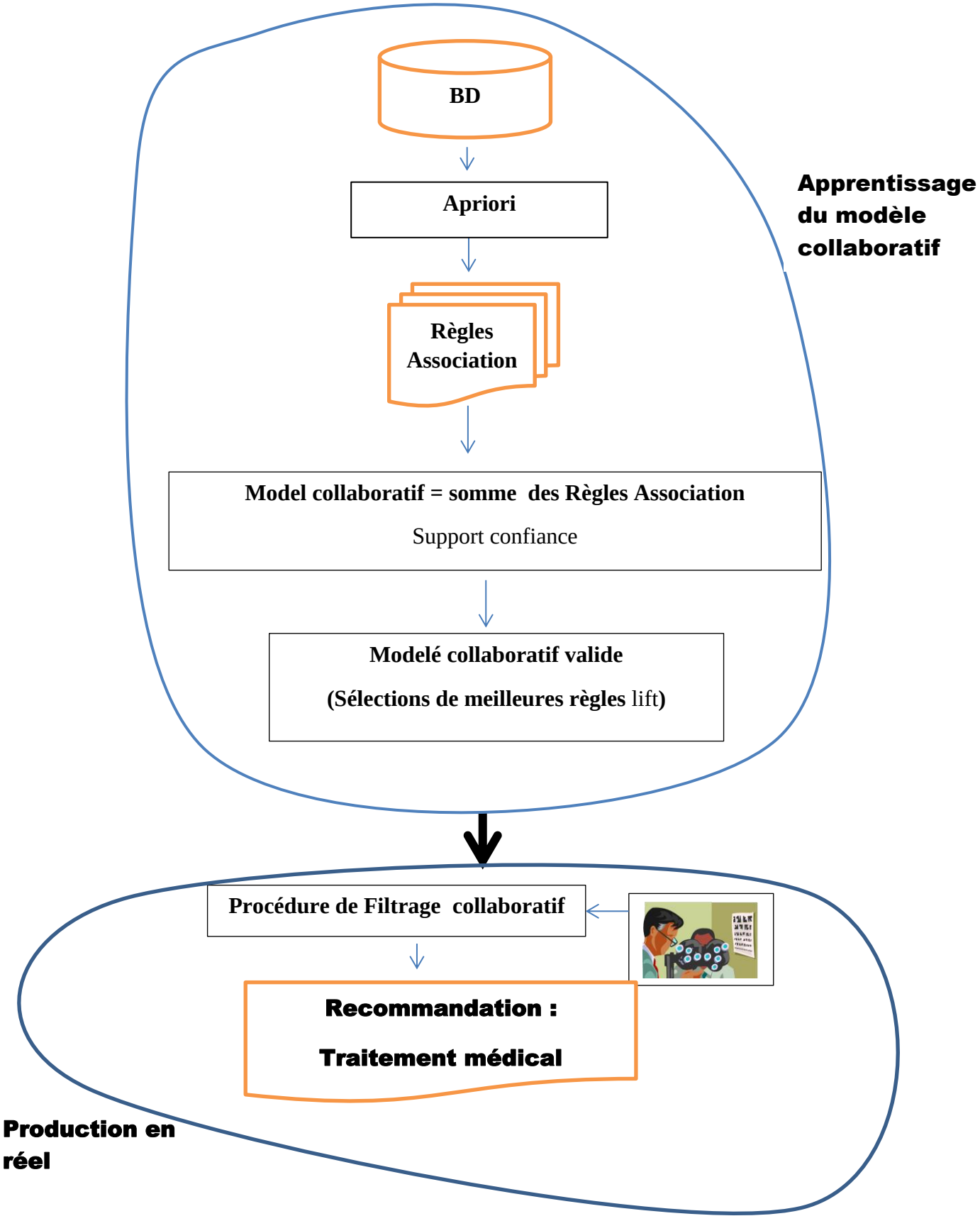


Figure 2.1 : Architecture général de system de recommandation.

Dans son sens récent, le filtrage collaboratif est sous-jacent aux systèmes de recommandation. Il regroupe des techniques qui visent à opérer une sélection sur les éléments à présenter aux utilisateurs (filtrage) en se basant sur le comportement et les goûts exprimés de très nombreux autres utilisateurs (collaboration).

Par exemple

- Pierre, Paul, Jacques, et les autres aiment les produits A et B.
- René, qui aime le produit A, va sûrement aimer le produit B…

Le recueil d'information joue un rôle crucial dans le processus, il peut être :

- Explicite. L'utilisateur attribue des notes au produit ou indique son appréciation (like).

2.1.1 Avantage :

pas d'ambiguïtés sur les goûts et les centres d'intérêts de l'utilisateur.

Inconvénients : biais de déclaration, exagération souvent.

- Implicite. Recueil basé sur le comportement (achats, clics, durée sur une page).

2.1.2 Avantages : objectivité.

Inconvénients : grande volumétrie, aucune indication sur l'appréciation.

Nous sommes dans le cadre du data mining. On cherche à identifier :

– à partir d'une base de données

– les régularités de comportement ou les associations (patterns) entre les utilisateurs et les items (produits, thèmes, etc.) en se basant sur leurs évaluations (notes) et leurs attitudes passées (clics, achats).

Mais, les données présentant des caractéristiques spécifiques (dimensionnalité, mises à jour fréquentes, données manquantes à profusion, matrices creuses, …), nous utiliserons préférentiellement certaines techniques.

Plutôt que les techniques sophistiquées, on privilégiera souvent les approches simples rapides, pouvant appréhender les fortes dimensionnalités / volumétries et résistants au sur apprentissage.

Le tableau de données pour le filtrage collaboratif se présente souvent comme suit :

	Item 1	Item 2	Item 3	Item 4
Fièvre	**?**	**2**	**7**	**8**
Toux	**4**	**1**	**?**	**7**
Maux de tète	**3**	**8**	**?**	**4**

Tab 2.1 : Le tableau de données pour le filtrage collaboratif.

- Les valeurs correspondent à une note (par ex. allant de 0 à 10). A la place d'une note, nous pouvons avoir une pondération binaire.
- A la place de l'utilisateur, nous pouvons avoir une transaction. Nous nous rapprochons du cadre de l'analyse des associations.
- « ? » Cela signifie que nous n'avons pas l'opinion de l'utilisateur sur le médicament et que nous souhaitons l'estimer avec précision.

2.1.3 Principe :

Les meilleures recommandations proviennent des individus qui présentent des goûts ou comportements similaires.

Point de départ. Nous devons disposer d'une base où les préférences d'un grand nombre d'utilisateurs sont disponibles**.**

2.2 Approche centrée utilisateur

Quelle note attribuer à René pour l'item n°4 ?

	Item 1	Item 2	Item 3	Item 4
Fièvre	**?**	**2**	**7**	**8**
Toux	**4**	**1**	**?**	**7**
Maux de tète	**3**	**8**	**?**	**4**
Nausée	**4**	**1**	**6**	**?**

Tab 2.2 : le tableau de Approche centrée utilisateur.

Identifier les utilisateurs dont le profil de notes est le plus proche de René. Se servir des notes de ces individus pour l'item n°4 pour estimer la note de René.

La recommandation ne tient absolument aucun compte de la nature du médicament ni de son contenu.

Les éléments clés de l'algorithme sont :

- Disposer d'une mesure de similarité ;
- Décider du nombre de voisins ;
- Calcul de la note agrégée, avec possiblement une pondération tenant compte de la proximité.

2.2.1 Avantage :

- Les calculs sont simples. Les résultats sont faciles à expliquer.
- On peut travailler toujours à partir d'une base constamment mise à jour.

2.2.2 Inconvénients :

- Parcourir la base à chaque recommandation à effectuer pour identifier le voisinage est coûteux. Heureusement des heuristiques permet de réduire les temps de calcul (ex. Localité-sensitive hashing).
- Quand il y a trop de données manquantes (au démarrage du système notamment), problème d'estimation. Cold Start problème.
- L'absence d'une note peut signifier une opinion. Ça ne m'intéresse pas donc je ne note pas.

2.3 Approche centrée item

Idée : production les valeurs d'un item à partir des autres. Problème de régression classique avec prise en compte des données manquantes.

	Item 1	Item 2	Item3	Item 4
Fièvre	**?**	**2**	**7**	**8**
Toux	**4**	**1**	**?**	**7**
Maux de tète	**3**	**8**	**?**	**4**
Nausée	**4**	**1**	**6**	**?**

Tab 2.3 : le tableau Approche centrée item.

- Régression multiple : Dimensionnalité et volumétrie rendent l'approche impossible. Nombre de paramètres à estimer trop élevé.
- Régression simple. On pourrait tester les 3 régressions simples pour prédire « Item 1 » et ne conserver que la meilleure. Idem pour les autres. Même là, le volume de calcul est considérable. Et on se heurte de plus au problème du sur apprentissage.

Construire une régression simple de type

$$y = x + b$$

- Un seul paramètre à estimer. Simplicité des calculs. La constante b est estimée par la moyenne des écarts entre les notes des items y et x.
- La contrainte sur la pente introduit une forme de régularisation, prévenant les problèmes de sur apprentissage.
- Une augmentation d'une manière significative du nombre d'utilisateurs ne pénalise pas le système.

2.3.1 Avantages

- Les calculs sont simples. Les résultats sont faciles à expliquer.
- On peut travailler toujours à partir d'une base constamment mise.
- à jour en déploiement.
- La mise à jour des constantes des modèles peut être lissée dans le temps.

2.3.2 Inconvénients :

- On doit parcourir toute la base et consolider les prédictions en déploiement
- Il y a quand même $p\times (p-1)$ 2 constantes à estimer à partir des données.
- Quand il y a trop de données manquantes (au démarrage du système notamment), problème d'estimation toujours. Cold Start problème.

2.4 Filtrage collaboratif par modèle

Dans le FC par voisinage, pas de modèle, les données orientent l'apprentissage

- Généralisation d'instance-based learning
- Ici, on a un apprentissage (training), une construction du modèle
- Ensuite on a une phase distincte de prédiction
- Apprentissage classique : arbre de décision, règles, classifieur Bayes, réseaux de neurones
- La plupart ont été adapté au contexte "recommandation"

La complétion de matrices généralise la classification "classique", avec deux différences :

- Pas de séparation claire entre variables dépendantes et indépendantes (colonnes)
- Pas de séparation claire entre données d'entraînement et de test (lignes)

2.5 Règles d'association

- Historiquement, règles d'association proposées dans le contexte du supermarché.
- On considère un ensemble de transactions T = {T1 ...Tm} définies sur n éléments de I.

Chaque T est un sous-ensemble d'éléments de I, on cherche les corrélations de sous-ensembles.

Exemple : {Bread, Butter, Milk} et {Fish, Beef, Ham} sont fréquents.

Mary a acheté {Butter, Milk}, on peut penser qu'elle achètera Bread.

item ⟶ / Gustomer ↓	Bread	Butter	Milk	Fish	Beef	Ham
Jack	1	1	1	0	0	0
Mary	0	1	1	0	1	0
Jane	1	1	0	0	0	0
Sayani	1	1	1	1	1	1
John	0	0	0	1	0	1
Tom	0	0	0	1	1	1
Peter	0	1	0	1	1	0

Tab 2.4 : Le tableau des règles d'association.

- Une règle se note X ⇒ Y ({Butter, Milk} ⇒ {Milk})
  - On utilise deux notions pour caractériser les fréquences des sous-ensembles (itemset) Le support de X ⊆ I, défini comme la fraction des transactions dont X est un sous-ensemble.

La confiance, définie comme la probabilité conditionnelle que T contienne Y sachant qu'elle contient X. Obtenue en divisant le support de X∪Y par celui de X.

On dit qu'une règle est une règle d'association de support minimal s et de confiance minimale c si :

Le support de X∪Y est d'au moins s

La confiance X ⇒ Y est au moins c

Pour trouver les règles d'associations, on procède en 2 étapes :

- On cherche tous les itemsets Z de support $s > s_0$

– Pour chacun, on calcule toutes les partitions (X, Z-X), pour créer des règles X ⇒ Z−X

On garde toutes les règles de confiance $c > c_\theta$

– c_θ et s_θ sont des seuils

La première phase est couteuse, c'est un champ de recherche (frequent itemset mining)

Adaptation au CF :

– Ces règles sont utiles avec des matrices unaires.
– On garde celles dont le 2nd membre contient 1 élément (exactement)
– On cherche les règles d'un utilisateur ($X \cup X_u$), on trie par confiance décroissante
– Les k premiers éléments sont le top- kpour cet utilisateur
– (plein de variantes)

2.6 Classifieur Bayesien naïf

On oublie l'ordre des notes, on considère qu'il s'agit de l catégories v1 ...vl.

On cherche r_{uj}, avec une valeur parmi v1 ...vl.

On doit déterminer la probabilité que ruj prenne chacune de ses valeurs, sachant que l'on dispose d'un ensemble de notes l_u :

$$P\ (r_{uj} = v_s \mid \text{notes de } I_u) \quad \forall s \in \{v1,..., vl\}$$

On transforme avec :

$$\mathbf{P(A|B)} = \frac{\mathbf{P(A).P(B/A)}}{\boldsymbol{P(B)}}\mathbf{(A|B)PA.P(B/A)}\Big/\mathbf{P(B)}$$

On gardera la plus grande probabilité.

Le dénominateur ne dépend pas de s :

$$P\ (r_{uj} = v_s \mid \text{notes de } I_u) \propto P\ (r_{uj} = v_s) \cdot P\ (\text{notes de } I_u | r_{uj} = v_s)$$

$P\ (r_{uj} = v_s)$, Prior, est la fraction des utilisateurs ayant donné vs comme note, parmi ceux ayant noté j.

Avec l'hypothèse naïve (indépendance des notes), on a :

$$P\ (\text{notes de } I_u \mid r_{uj} = v_s) = \Pi\ k\epsilon\ l_u P\ (r_{uk} \mid r_{uj} = v_s)$$

Chaque P (r_{uk} | r_{uj} = v_s) est la fraction des utilisateurs qui ont noté r_{uk} l'élément k, sachant qu'ils ont noté vs l'élément j On peut ensuite estimer $\hat{r}_{uj}$:

En calculant les probabilités pour tous les s, prendre la plus grande, garder v_s

Raisonnable si l'est petit.

Utiliser une moyenne pondérée, dont les poids des valeurs possibles sont la probabilité.

2.6.1 Classifier arbitraire comme boîte noire

Dans le cas non unaire, travailler sur une matrice incomplète est délicat.

Réduction de dimension possible.

Dans l'espace original, on peut adopter une solution itérative :

- On initialise les valeurs manquantes (moyenne ligne, colonne), éventuellement centrage
- On fixe une colonne (cible), les autres servent d'entrée (facture variable)
- On apprend avec les notes existantes de la colonne, on prédit le reste
- On met à jour la colonne avec les prédictions
- On répète, jusqu'à convergence

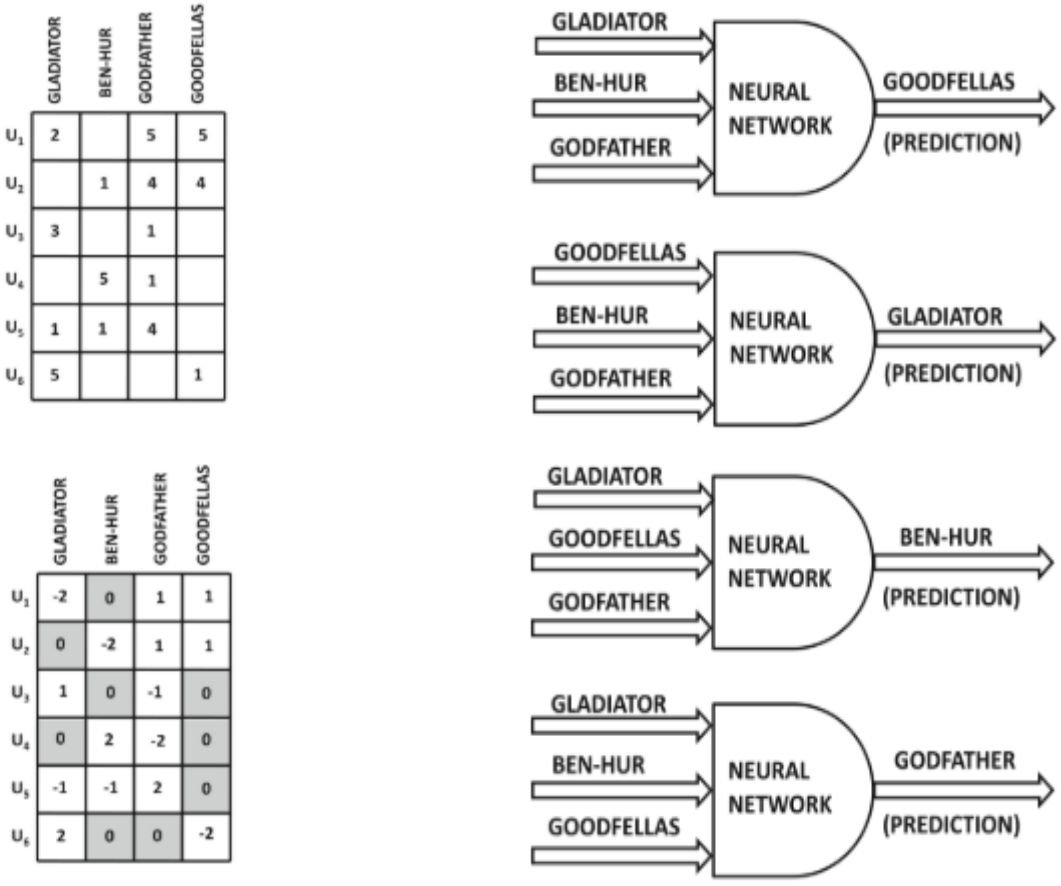


Figure 2.2 : Classifier arbitraire comme boîte noire.

2.6.2 Modèles à facteurs latents

- L'idée est de bénéficier du fait que des parties importantes des lignes/colonnes sont grandement corrélées
- Redondances
- On peut approximer la matrice complète raisonnablement avec une matrice de rang faible
- Un sous-ensemble d'entrées de la matrice suffit pour obtenir une matrice complète de rang faible
- Cette matrice de rang faible fournit des estimations robustes des entrées manquantes de la matrice initiale
- Ces méthodes sont l'état de l'art pour les systèmes de recommandation

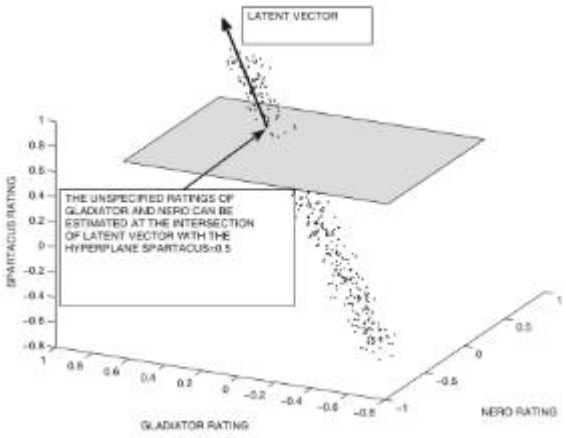


Figure 2.3 : Modèles à facteurs latents.

Notes corrélées, rang Ĩ (droite)

- Ici, une seule note (Spartacus 0.5) suffit pour estimer Nero et Gladiator !
- Intersection entre le vecteur latent (droite) et l'hyperplan (parallèle aux axes) tel que Spartacus ait 0.5

En pratique : on n'a pas besoin de toute la matrice pour estimer les vecteurs latents principaux
Orthogonalité des vecteurs latents

2.6.3 Principe de la factorisation de matrices

Une matrice R de rang k ≪ min (m,n) peut toujours être écrite sous la forme :

$$R = UVT$$

- U est une matrice m×k, V n×k
- Les colonnes de U sont des vecteurs d'une base de l'espace de dim k des colonnes de R
- Une ligne de V contient les coefficients pour combiner ces vecteurs en une colonne de R.
- Il existe un nombre infini de factorisations, correspondants à divers ensembles de vecteurs
- La SVD est un exemple, dans lequel il y a orthogonalité des colonnes/lignes

2.6.4 Un peu d'algèbre linéaire

- Si la matrice R a un rang plus grand que k, on peut l'approximer par R ≈ UVT
- U est une matrice m×k, V n×k

- l'erreur d'approximation est ||R−UVT ||2
- ||·||2 représente la somme des carrés des entrées de R−UVT , c'est la norme de Frobenius

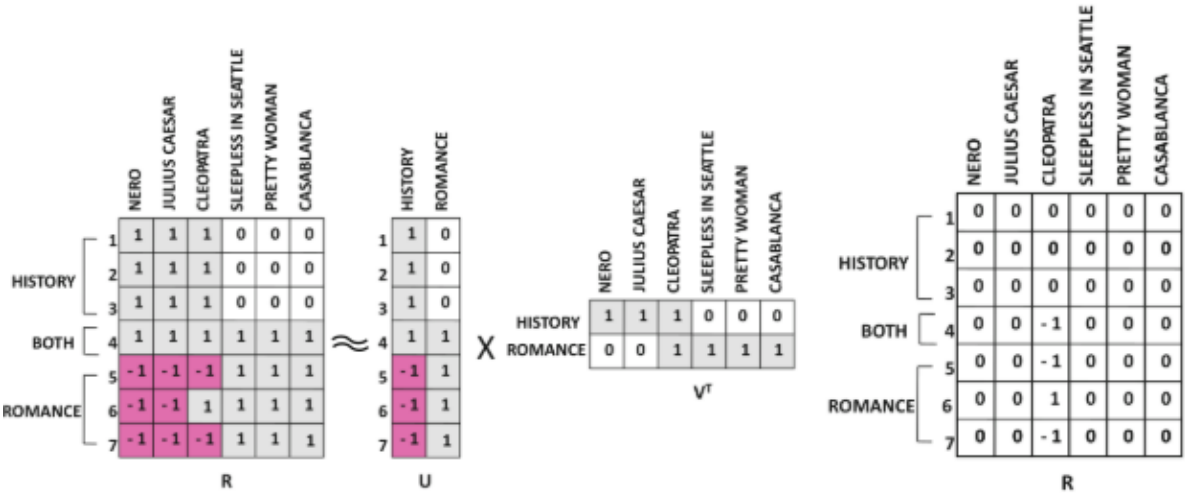


Figure 2.4 : Un peu d'algèbre linéaire.

- Les utilisateurs 1-3 aiment les films historiques, neutre sur Romance
- 4 aime les 2 genres
- 4-7 aiment Romance, pas films historiques
- Nombreuses corrélations !
- Si on multiplie par -1, interprétations plus délicates

Chaque colonne de U est appelée vecteur latent, chaque ligne facteur latent

- la ie ligne u_i de U est un facteur utilisateur, contenant "l'affinité" de l'utilisateur i pour chacun des k concepts (genres)
- chaque ligne vi de V est un facteur "item", contenant l'appartenance des films à chaque genre.

2.7 Apprentissage

On cherche à résoudre :

$$\text{minimiser } J = \frac{1}{2}||R - UV^T||^2$$

Sans contraintes sur U et V.

- ”Loss” quadratique, diverses SGD peuvent le résoudre
- MAIS : valeurs manquantes !
- Si on pouvait factoriser R en UVT complètes, on pourrait prédire les entrées manquantes de R :

$$\hat{r}_{ij} = \sum_{s=1}^{k} u_{is} \cdot v_{js}$$

- L’erreur pour un item est :

$$e_{ij} = rij - \hat{r}_{ij}$$

- D’où :

$$\text{minimiser } J = \frac{1}{2} \sum_{(i,j) \in S} e_{ij}^2$$

Noter le ”seulement sur les valeurs de S.

Pour mettre en œuvre un modèle de filtrage collaboratif basé sur ces concepts, nous devons suivre les étapes suivantes :

- Préparation des données : Structurer les données sous forme de transactions (par exemple, des paniers d'achat ou des évaluations d'éléments).
- Calcul du support : Déterminer la fréquence d'apparition de chaque ensemble d'éléments dans les transactions.
- Calcul de la confiance et du lift : Utiliser les supports calculés pour déterminer les valeurs de confiance et de lift pour différentes règles d'association.
- Recommandation : Utiliser les règles avec une haute confiance et un lift élevé pour recommander des éléments.

2.8 Calculer : batch update méthode

Algorithm *GD*(Ratings Matrix: R, Learning Rate: α)
begin
  Randomly initialize matrices U and V;
  $S = \{(i, j) : r_{ij}$ is observed$\}$;
  while not(convergence) **do**
  begin
    Compute each error $e_{ij} \in S$ as the observed entries of $R - UV^T$;
    for each user-component pair (i, q) **do** $u_{iq}^{+} \Leftarrow u_{iq} + \alpha \cdot \sum_{j:(i,j)\in S} e_{ij} \cdot v_{jq}$;
    for each item-component pair (j, q) **do** $v_{jq}^{+} \Leftarrow v_{jq} + \alpha \cdot \sum_{i:(i,j)\in S} e_{ij} \cdot u_{iq}$;
    for each user-component pair (i, q) **do** $u_{iq} \Leftarrow u_{iq}^{+}$;
    for each item-component pair (j, q) **do** $v_{jq} \Leftarrow v_{jq}^{+}$;
    Check convergence condition;
  end
end

2.9 Calculer : SGD

U et V mises à jour simultanément, convergence plus rapide que batch

- SGD préférable pour données larges et temps de calcul goulet d'étranglement
- $\alpha = 0.005$ (learning rate), peut être adapté à chaque étape
- Seuil de convergence à fixer astucieusement, trop d'itérations peuvent dégrader la qualité
- Initialisation dans $(-1, 1)$?

2.10 Validation et Interprétation des Règles avec le support et confiance

- Validation Croisée : Diviser les données en ensembles de formation et de test pour valider les règles générées.
- Interprétation des Règle : Visualiser et interpréter les règles pour comprendre les relations entre les items.
- Recommandation : Utiliser les règles générées pour recommander des items aux utilisateurs en fonction de leurs interactions passées.

2.11 Validation des Règles avec le Lift Calcul du Lift

Pour chaque règle d'association générée, calculer le lift afin de déterminer si la présence de X augmente significativement la probabilité de présence de Y.

2.11.1 Interprétation du Lift

- Un lift > 1 indique que la présence de X augmente la probabilité de Y.

- Un lift = 1 indique que X et Y sont indépendants.
- Un lift < 1 indique que la présence de X diminue la probabilité de Y.

L'apprentissage collaboratif avec des règles d'association est une méthode puissante pour les systèmes de recommandation.

En analysant les interactions entre les utilisateurs et les items, ainsi que les relations entre les items eux-mêmes, on peut fournir des recommandations personnalisées qui améliorent l'expérience utilisateur. Les techniques telles que l'algorithme Apriori permettent de découvrir des relations cachées dans les données, ce qui est crucial pour développer des modèles de recommandation efficaces.

L'apprentissage collaboratif est une méthode puissante qui, lorsqu'elle est bien mise en œuvre, peut transformer l'expérience éducative des étudiants.

2.12 Production réel

Pour produire des règles d'association en temps réel, il est essentiel de concevoir un système capable de traiter et d'analyser rapidement de grandes quantités de données transactionnelles au fur et à mesure qu'elles sont générées.

La recommandation de traitements médicaux utilisant le filtrage collaboratif est une approche avancée qui utilise des techniques de machine learning pour aider à la prise de décision en santé. Voici une explication détaillée sur comment cela fonctionne et comment une telle application pourrait être mise en œuvre en production :

2.12.1 Concept du Filtrage Collaboratif

Le filtrage collaboratif repose sur l'idée que les utilisateurs qui ont eu des comportements similaires dans le passé auront des comportements similaires à l'avenir. Dans le contexte médical, cela peut être appliqué en se basant sur des données de patients et de leurs réponses à différents traitements.

2.13 Étapes pour Construire et Déployer un Modèle de Filtrage Collaboratif

2.13.1 Collecte et Préparation des Données

- Collecte des Données : Rassemblez les données sur les interactions utilisateur-élément, telles que les évaluations, les clics, les achats, etc.

- Préparation des Données : Nettoyez les données, gérez les valeurs manquantes et effectuez une transformation des données en une matrice utilisateur-élément.

2.13.2 Exploration des Données

- Analysez les données pour comprendre la distribution des évaluations et la parité de la matrice.
- Utilisez des visualisations pour explorer les interactions utilisateur-élément.

2.13.3 Sélection de l'Algorithme

Choisissez l'algorithme de filtrage collaboratif approprié : basé sur les utilisateurs, basé sur les éléments, ou une méthode de factorisation matricielle telle que la décomposition en valeurs singulières (SVD).

2.13. 4 Construction du Modèle

Implémentez l'algorithme choisi en utilisant une bibliothèque de machine Learning comme scikit-learn, Surprise, ou TensorFlow.

2.13.5 Évaluation du Modèle

- Utilisez des métriques telles que l'erreur quadratique moyenne (RMSE), la précision, le rappel et le F1-score pour évaluer les performances du modèle.
- Effectuez une validation croisée pour assurer la robustesse du modèle.

2.13.6 Optimisation du Modèle

- Réglez les hyper paramètres du modèle pour améliorer ses performances.
- Essayez différentes techniques d'amélioration telle que l'ajustement des biais utilisateur/élément, l'intégration de contextes, etc.

2.13. 7 Déploiement du Modèle

- Déployez le modèle dans un environnement de production en utilisant des Framework comme Flask, Django, ou FastAPI pour créer une API RESTful.
- Utilisez des outils de déploiement comme Docker, Kubernetes pour la capabilité et la gestion des versions.

2.13. 8 Surveillance et Maintenance du Modèle

- Surveillez les performances du modèle en production et collectez des retours utilisateurs.
- Mettez à jour et ré entraînez le modèle régulièrement avec de nouvelles données pour maintenir sa pertinence.

3. Architecture du system de recommandation utilise

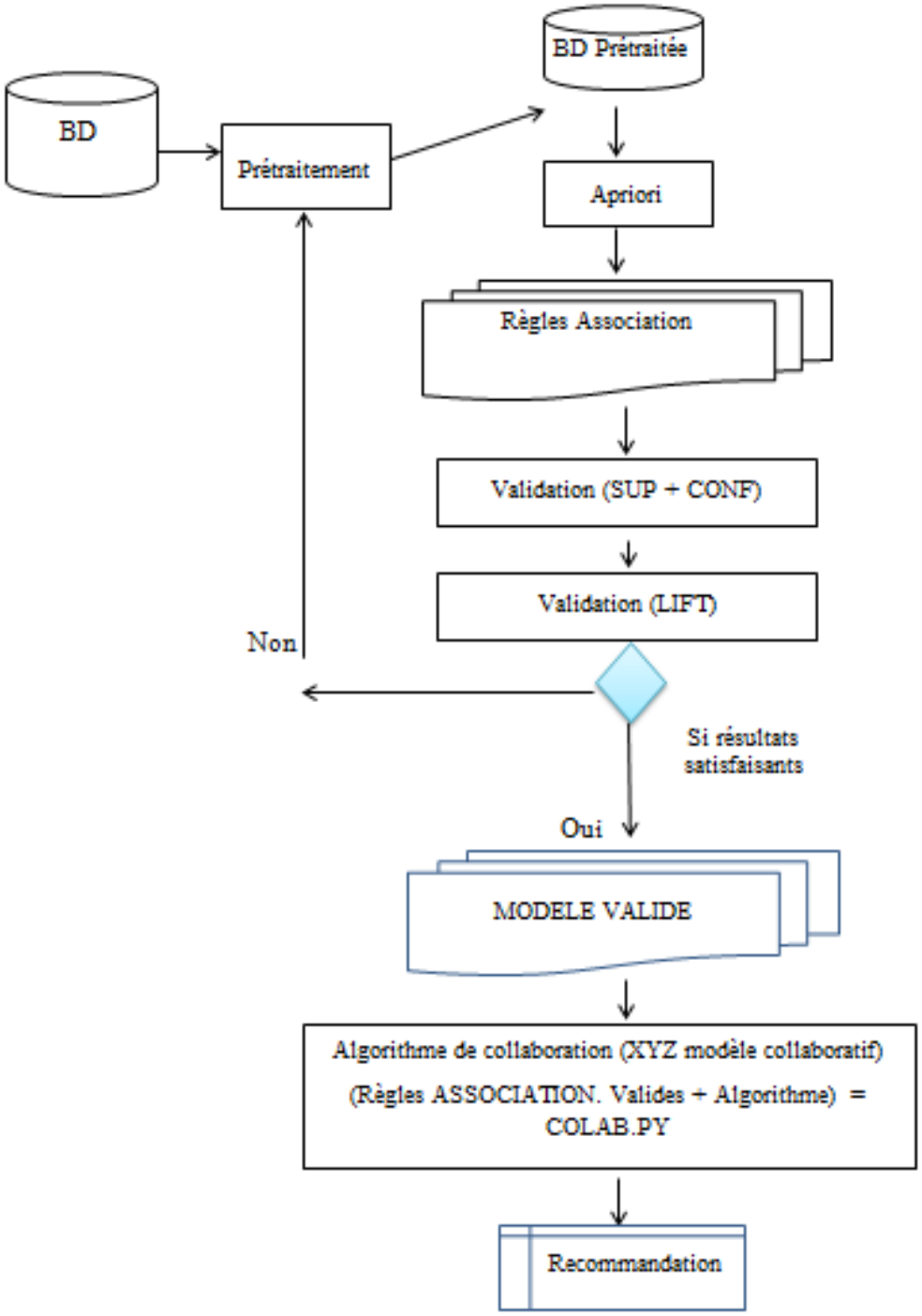


Figure 2.5 : Architecture du system de recommandation utilisé.

4. Préparation de données

La préparation de données couvre toutes les activités pour préparer, à partir des données initiales, l'ensemble de données final qui sera utilisé pour la construction de modèles.

Elle est constituée de cinq étapes la sélection, le nettoyage, la construction, l'intégration et le formatage des données.

4.1 Sélection de données

Les sources de données qui sont utiles et disponibles varient d'un problème à un autre. Il est donc important d'éliminer les données inutiles ou non pertinentes. Les critères pouvant aidé à décider quelles sont les données importantes peuvent inclure la pertinence des données par rapport aux objectifs, leur qualité et même les contraintes techniques telles que des limites sur le volume ou les types de données [Shearer, 2000], [URL6].

4.2 Nettoyage de données

L'étape de nettoyage de données est habituellement nécessaire. Elle consiste à détecter et supprimer les erreurs et les contradictions figurant dans les données. Cette étape prend tout son sens lorsque plusieurs sources de données doivent être combinées du fait qu'il y a souvent des redondantes dans différentes représentations [Chen et al. 2004].

Le nettoyage tient compte aussi d'autres types de problèmes de qualité comme les valeurs de champs incorrectes (par exemple le numéro d'identification de sécurité sociale se trouvant à la place du revenu d'une personne), des valeurs qui semblent être correctes (par exemple les mâles enceintes), les valeurs absentes, un même nom employé pour différentes entités ou différents noms employés pour la même entité, etc. [URL3].

4.3 Construction de données

La construction de données consiste à développer des enregistrements entièrement nouveaux ou produire des attributs dérivés des attributs existants. Ainsi, certaines variables qui ont peu d'effet lorsque prises seules, peuvent être combinées avec d'autres variables en utilisant l'arithmétique ou les opérations algébriques, telles que l'utilisation du ratio «dette / revenu » au lieu de simplement la dette ou du revenu.

Quelques variables qui sont définies sur un grand intervalle peuvent aussi être modifiées en créant une meilleure variable telle que l'utilisation du Ïog du revenu au lieu du revenu {URL3]. Un autre type d'attribut dérivé est la transformation simple-attribut comme la transformation des champs symboliques en des valeurs numériques. Le but de cette

transformation est de se conformer au format requis par les outils de construction de modèle [Shearer, 2000], [Zaima et al. 2003].

4.4 Intégration de données

L'intégration de données combine les données de différentes sources en une seule base de données. Ceci consiste en la combinaison de l'information provenant des différentes tables ou enregistrements. L'intégration de données couvre également les agrégations qui consistent à calculer des nouvelles valeurs en récapitulant l'information de plusieurs enregistrements et/ou plusieurs tables [URL6]. Par exemple, une agrégation pourrait inclure la conversion d'une table des achats de client, où il y a un enregistrement pour chaque achat, en une nouvelle table où il y a un enregistrement pour chaque client [Shearer, 2000].

4.5 Formatage de données

Le formatage de données est souvent nécessaire afin de convertir les données en un faonnât requis par la technique de forage de données qui sera employée. Il est commun pour certains types d'information, d'être représentées dans différents formats si provenant de différentes sources de données. La date est probablement le cas le plus commun. Le champ date extrait à partir d'une source de données européenne dans le format «DD MM-YYYY», pourrait être transformé dans le format «MM-DD-YYYY» pour faciliter l'assimilation des résultats de forage de données [Mendonca et al. 1999]. Il est nécessaire, donc, de s'assurer que ce type d'information est représenté dans un format cohérent et connu par l'entrepôt de forage de données.

5. Architecture de l'algorithme APRIORI

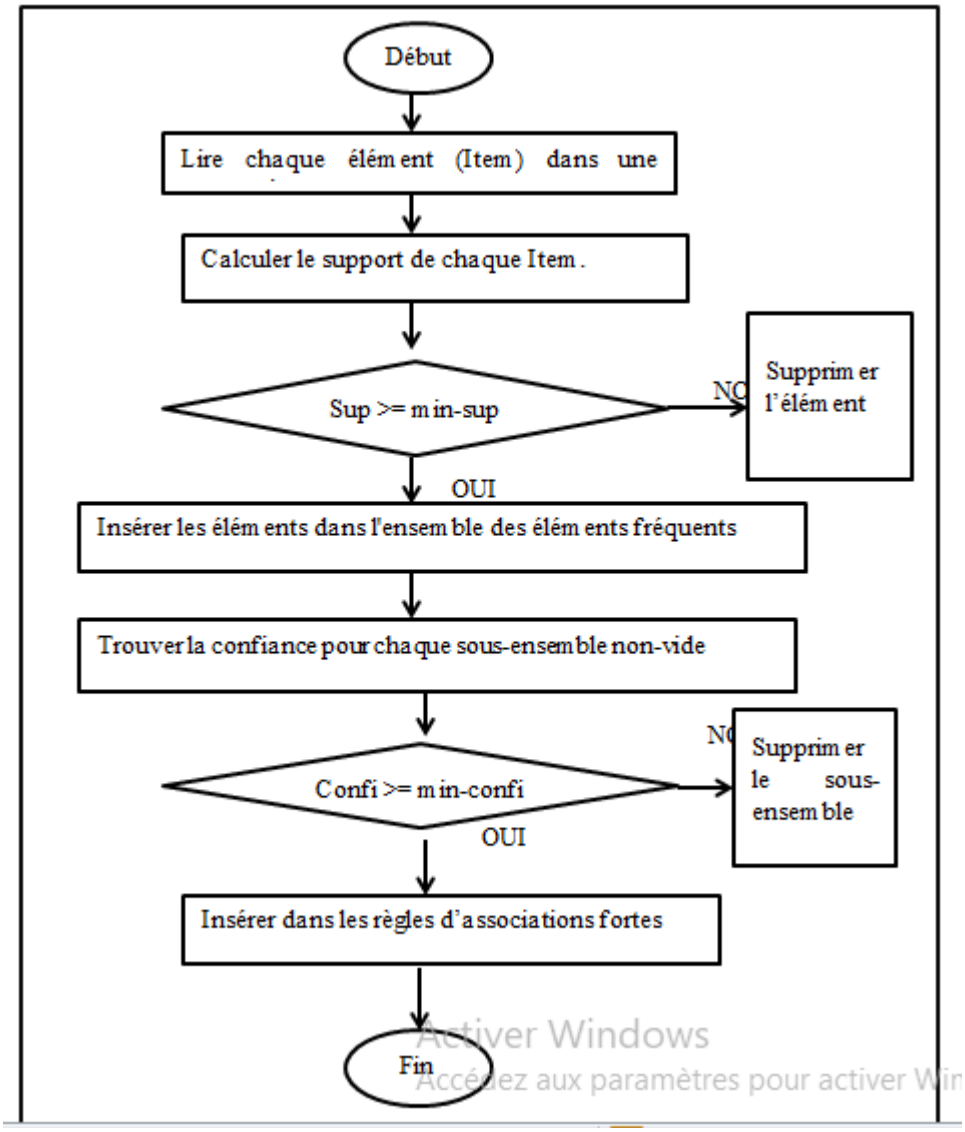


Figure 2.6 : Étapes d'extraction de règles d'associations (algorithme Apriori).

6. Conclusion

A travers ce deuxième chapitre, nous avons effectué une analyse des systèmes de recommandation existants dans le domaine de l'e-santé. Cette analyse va nous aider dans la conception et l'implémentation de notre système de recommandation qui vont faire l'objet du chapitre suivant.

CHAPITRE 3

Implémentation et résultats

1. Introduction

Dans le cadre d notre travail, nous avons traité notre jeu de donnée pour une meilleur. Dans ce dernier chapitre, nous présentons d'abord une étude technique dans laquelle nous définissons l'environnement logiciel utilisée pour construire notre application, puis nous définirons notre data set avec une description de ses caractéristiques et les étapes de prétraitement des données (explorer, nettoyer, sélection de modèle ...) pour corriger les valeurs aberrantes et choisir le meilleur modèle à suivre.

A la fin, c'est la partie application ou nous fournissons des interfaces graphiques importantes développées pour clarifier les performances des activités du système et nous terminerons par une conclusion.

2. Environnement de développement

2.1 Python

C'est un langage de programmation multi-paradigme et le langage de programmation dominant dans la data science avec de nombreuses implémentations ce qui le rend encore plus intéressant. Concernant le domaine de l'apprentissage automatique Python se distingue tout particulièrement en orant une pléthore de librairies de très grande qualité, couvrant tous les types d'apprentissages disponibles qui combine la facilite d'utilisation et d'apprentissage avec la puissance des librairies qu'elles possèdent. Parmi ces bibliothèques, nous avons utilisé **[16]**.

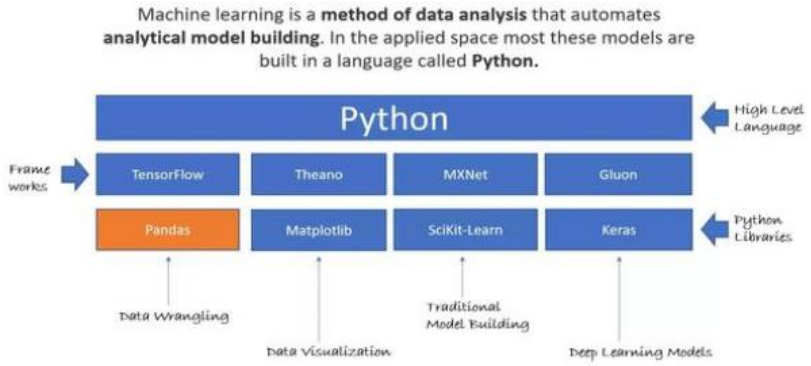


Figure 3.1 : Aperçu des Framework et libraires de python [22].

2.2 Le navigateur Anaconda

C'est une distribution libre et open source des langages de programmation Python et R, appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique.

Une distribution est un langage de programmation, certaines librairies et autres fonctionnalités. Anaconda est donc une distribution Python, faite pour la Data Science **[21]**.

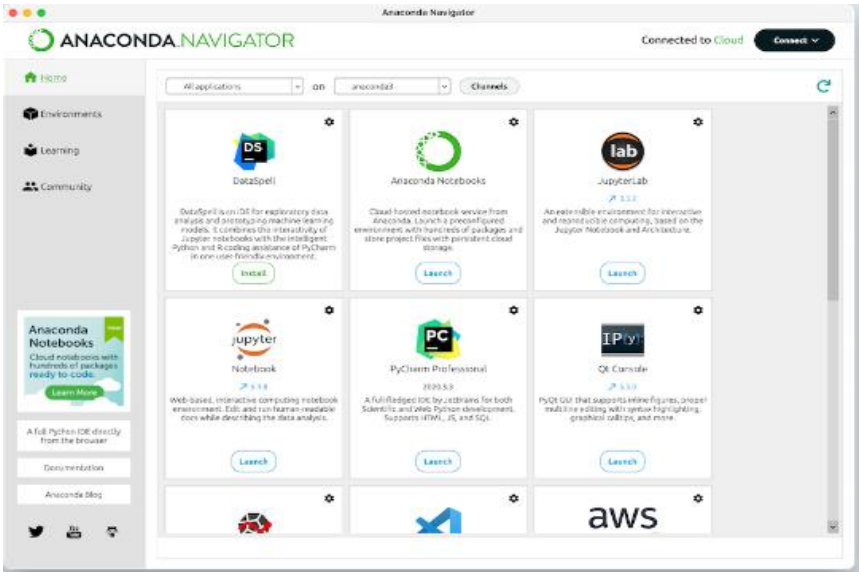


Figure 3.2 : Navigateur Anaconda.

2.3 Jupyter Notebook

C'est une interface Web dans laquelle nous pouvons taper du code Python, l'exécuter et voir directement les résultats, y compris une visualisation à l'aide de graphiques **[17]**.

2.4 Visual Studio Code

C'est un éditeur du code source développé par Microsoft pour Windows, Linus et MacOs, qui prend immédiatement en charge presque tous les principaux langages de programmation. Plusieurs d'entre eux sont inclus par défaut, par exemple JavaScript, TypeScript, CSS et HTML, mais d'autres extensions de langage peuvent être trouvées et téléchargées gratuitement à partir de VS Code Marketplace. Il a été présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code multiplateforme.

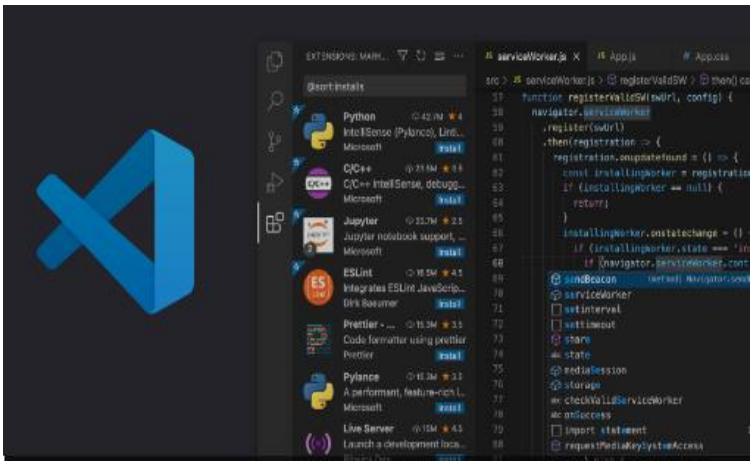


Figure 3.3 : visuel Studio code.

3. Bibliothèques essentielles pour l'apprentissage automatique en Python

Bibliothèque de programmes ou bien librairie logicielle est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Les fonctions sont regroupées de par leur appartenance à un même domaine conceptuel (mathématique, graphique, tris, etc.). La bibliothèque standard de Python est très grande, elle offre un large éventail d'outils.

Dans ce qui suit, nous allons définir les bibliothèques utilisées dans notre implémentation :

3.1 Pandas

Pandas est une bibliothèque Python open source pour l'analyse de données hautement spécialisée. C'est actuellement le point de référence que tous les professionnels utilisant le langage Python doivent étudier à des fins statistiques d'analyse et de prise de décision. Cette bibliothèque a été conçue et développée principalement par Wes McKinney à partir de 2008. En 2012, Sien Chang, l'un de ses collègues, a été ajouté au développement. Ensemble, ils ont mis en place l'une des bibliothèques les plus utilisées de la communauté Python [18].

3.2 Flask

Est un petit Framework web Python léger, qui fournit des outils et des fonctionnalités utiles qui facilitent la création d'applications web en Python. Il offre aux développeurs une certaine flexibilité et constitue un cadre plus accessible pour les nouveaux développeurs puisque vous pouvez construire rapidement une application web en utilisant un seul fichier

Python. Flaks est également extensible et ne force pas une structure de répertoire particulière ou ne nécessite pas de code standard compliqué **[19]**.

3.3 ML xtend

Pourrait être une expression pour décrire l'extension ou l'expansion de l'utilisation de Python dans le domaine de l'apprentissage automatique (Machine Learning). Cela pourrait signifier une augmentation de l'utilisation de Python pour des tâches plus avancées en apprentissage automatique ou pour des applications plus complexes. Cela pourrait également impliquer une utilisation de Python dans des domaines connexes à l'apprentissage automatique, tels que le traitement du langage naturel, la vision par ordinateur, l'analyse de données massives, etc. En bref, "ML xtend code Python" pourrait indiquer une tendance à étendre l'utilisation de Python pour répondre à des besoins croissants ou émergents dans le domaine de l'apprentissage automatique et des domaines connexes.

3.4 Html

Un langage de balisage : il utilise une structure du code sous forme de balises. Il permet la représentation des pages Web statiques, c’est un langage permettant d’écrire de l’hypertexte.

4. Les interfaces du système

4.1 Visualisation des données

La visualisation des données médicales est une approche puissante pour analyser et interpréter des informations complexes de manière compréhensible. Les tableaux sont souvent utilisés pour organiser et présenter ces données de manière structurée.

jupyter regles_association.csv Last Checkpoint: 28 days ago

File Edit View Settings Help

Delimiter: ,

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leve
1	douleurs corporelles	grippe	0.1111111111111111	0.4444444444444444	0.1111111111111111	1.0	2.25	0.0617283950617
2	douleurs corporelles	paracétamol	0.1111111111111111	0.4444444444444444	0.1111111111111111	1.0	2.25	0.0617283950617
3	douleurs corporelles	sirop contre la toux	0.1111111111111111	0.5555555555555556	0.1111111111111111	1.0	1.7999999999999998	0.0493827160493
4	fièvre	grippe	0.6666666666666666	0.4444444444444444	0.4444444444444444	0.6666666666666666	1.5	0.1481481481481
5	fièvre	migraine	0.6666666666666666	0.2222222222222222	0.2222222222222222	0.3333333333333333	1.5	0.0740740740740
6	fièvre	ibuprofène	0.6666666666666666	0.3333333333333333	0.3333333333333333	0.5	1.5	0.111111111111
7	fièvre	paracétamol	0.6666666666666666	0.4444444444444444	0.4444444444444444	0.6666666666666666	1.5	0.1481481481481
8	maux de gorge	rhume	0.2222222222222222	0.2222222222222222	0.2222222222222222	1.0	4.5	0.172839506172
9	maux de gorge	sirop contre la toux	0.2222222222222222	0.5555555555555556	0.2222222222222222	1.0	1.7999999999999998	0.0987654320987
10	maux de tête	intoxication alimentaire	0.5555555555555556	0.2222222222222222	0.2222222222222222	0.39999999999999997	1.8	0.0987654320987
11	maux de tête	migraine	0.5555555555555556	0.2222222222222222	0.2222222222222222	0.39999999999999997	1.8	0.0987654320987
12	maux de tête	antiacide	0.5555555555555556	0.2222222222222222	0.2222222222222222	0.39999999999999997	1.8	0.0987654320987
13	maux de tête	ibuprofène	0.5555555555555556	0.3333333333333333	0.3333333333333333	0.6	1.8	0.1481481481481
14	nausée	intoxication alimentaire	0.2222222222222222	0.2222222222222222	0.2222222222222222	1.0	4.5	0.172839506172

Figure 3.4 : Visualisation des données.

4.2 Le prétraitement de données

La base de données est une base de données temporelle. Avec les notes des items données par les utilisateurs, on trouve le temps de l'attribution de note en seconde.

De ce fait, la première étape à faire dans le prétraitement de données est de ne garder que les données qui sont importantes pour la réalisation de notre système. L'utilisateur de notre système peut charger le fichier de la base de données, comme il peut charger le fichier que nous utilisons après prétraitement.

4.3 Importation des Librairies

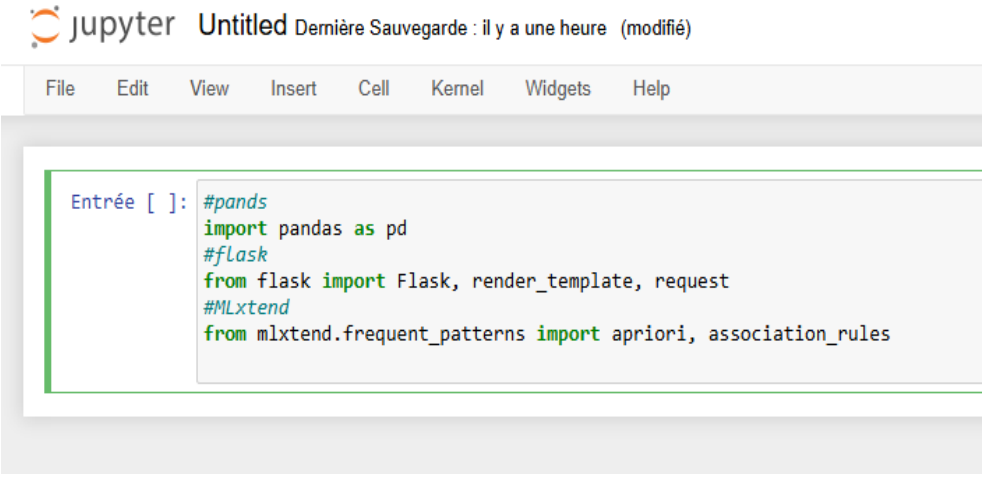
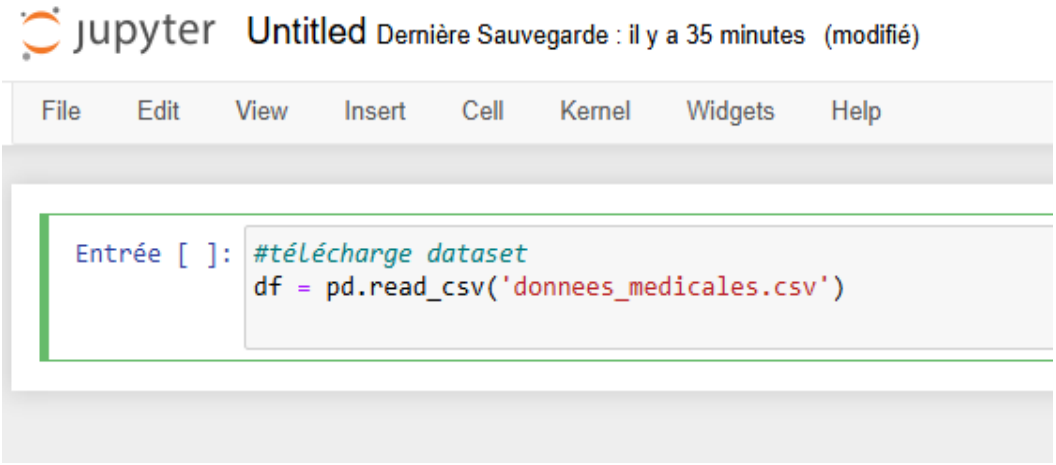


Figure 3.5 : Importer les Librairies.

4.4 Téléchargement des données



jupyter donnees_medicales.csv Last Checkpoint: 22 days ago

File Edit View Settings Help

Delimiter: ,

	Symptômes	Maladies	Médicaments
1	fièvre,toux	grippe	tamol,sirop contre la toux
2	maux de tête,fièvre	migraine	ibuprofène
3	toux,maux de gorge	rhume	sirop contre la toux
4	aux de tête,fièvre,nausée	ne,intoxication alimentaire	ibuprofène,antiacide
5	,toux,douleurs corporelles	grippe	tamol,sirop contre la toux
6	fièvre,toux	grippe	tamol,sirop contre la toux
7	fièvre,maux de tête	grippe	paracétamol
8	fièvre,toux,maux de tête	grippe	contre la toux,ibuprofène
9	maux de gorge,toux	rhume	sirop contre la toux
10	nausée,maux de tête	intoxication alimentaire	antiacide

Figure 3.6 : Télécharger les données.

4.5 Nettoyage et traitement des données

```
[3]: df.dropna(inplace=True)
```

```
[4]: df.drop_duplicates(inplace=True)
```

```
[5]: df['Symptômes'] = df['Symptômes'].str.lower()
     df['Maladies'] = df['Maladies'].str.lower()
     df['Médicaments'] = df['Médicaments'].str.lower()
```

```
[6]: df_symptomes = df['Symptômes'].str.get_dummies(sep=',')
     df_maladies = df['Maladies'].str.get_dummies(sep=',')
     df_medicaments = df['Médicaments'].str.get_dummies(sep=',')
```

```
[7]: df_combine = pd.concat([df_symptomes, df_maladies, df_medicaments], axis=1)
```

```
[8]: df_combine.to_csv('donnees_medicales_pretraitees.csv', index=False)
```

```
[9]: itemsets_frequents = apriori(df_combine, min_support=0.1, use_colnames=True)

[10]: regles = association_rules(itemsets_frequents, metric='lift', min_threshold=1)

[11]: regles = regles[(regles['antecedents'].apply(lambda x: all(item in df_symptomes.columns for item in list(x)))) &
                      (regles['consequents'].apply(lambda x: any(item in df_maladies.columns for item in list(x)) or
                                                  any(item in df_medicaments.columns for item in list(x))))]

[12]: regles['antecedents'] = regles['antecedents'].apply(lambda x: ','.join(list(x)))
      regles['consequents'] = regles['consequents'].apply(lambda x: ','.join(list(x)))

[13]: regles.to_csv('regles_association.csv', index=False)
```

Figure 3.7 : Nettoyage et traitement des données.

```
[14]: print(regles)
```

```
              antecedents                                        consequents  \
5     douleurs corporelles                                            grippe
7     douleurs corporelles                                       paracétamol
9     douleurs corporelles                             sirop contre la toux
13                 fièvre                                            grippe
15                 fièvre                                          migraine
...                   ...                                               ...
4097         toux,fièvre  sirop contre la toux,ibuprofène,grippe,maux de...
4099  maux de tête,fièvre  sirop contre la toux,ibuprofène,grippe,toux,pa...
4104                toux  sirop contre la toux,ibuprofène,grippe,maux de...
4105        maux de tête  sirop contre la toux,ibuprofène,grippe,toux,pa...
4107               fièvre  sirop contre la toux,ibuprofène,grippe,toux,ma...

      antecedent support  consequent support   support  confidence  lift  \
5               0.111111            0.444444  0.111111    1.000000  2.25
7               0.111111            0.444444  0.111111    1.000000  2.25
9               0.111111            0.555556  0.111111    1.000000  1.80
13              0.666667            0.444444  0.444444    0.666667  1.50
15              0.666667            0.222222  0.222222    0.333333  1.50
...                  ...                 ...       ...         ...   ...
4097            0.333333            0.111111  0.111111    0.333333  3.00
4099            0.444444            0.111111  0.111111    0.250000  2.25
4104            0.555556            0.111111  0.111111    0.200000  1.80
4105            0.555556            0.111111  0.111111    0.200000  1.80
4107            0.666667            0.111111  0.111111    0.166667  1.50
```

	antecedent support	consequent support	support	confidence	lift	\
5	0.111111	0.444444	0.111111	1.000000	2.25	
7	0.111111	0.444444	0.111111	1.000000	2.25	
9	0.111111	0.555556	0.111111	1.000000	1.80	
13	0.666667	0.444444	0.444444	0.666667	1.50	
15	0.666667	0.222222	0.222222	0.333333	1.50	
...	...	...	...	...	...	
4097	0.333333	0.111111	0.111111	0.333333	3.00	
4099	0.444444	0.111111	0.111111	0.250000	2.25	
4104	0.555556	0.111111	0.111111	0.200000	1.80	
4105	0.555556	0.111111	0.111111	0.200000	1.80	
4107	0.666667	0.111111	0.111111	0.166667	1.50	

	leverage	conviction	zhangs_metric
5	0.061728	inf	0.625
7	0.061728	inf	0.625
9	0.049383	inf	0.500
13	0.148148	1.666667	1.000
15	0.074074	1.166667	1.000
...	...	...	...
4097	0.074074	1.333333	1.000
4099	0.061728	1.185185	1.000
4104	0.049383	1.111111	1.000
4105	0.049383	1.111111	1.000
4107	0.037037	1.066667	1.000

[614 rows x 10 columns]

Figure 3 .8 : cette figure représente les règles association.

5. Modèle

Nous proposons ci-dessous des interfaces de recommandations médicales pour permettre aux personnes de savoir si elles risquent de contracter une maladie avec un taux de recommandation bien défini.

Cette application contient une adresse locale après l'installation.

http://127.0.0.1:5000

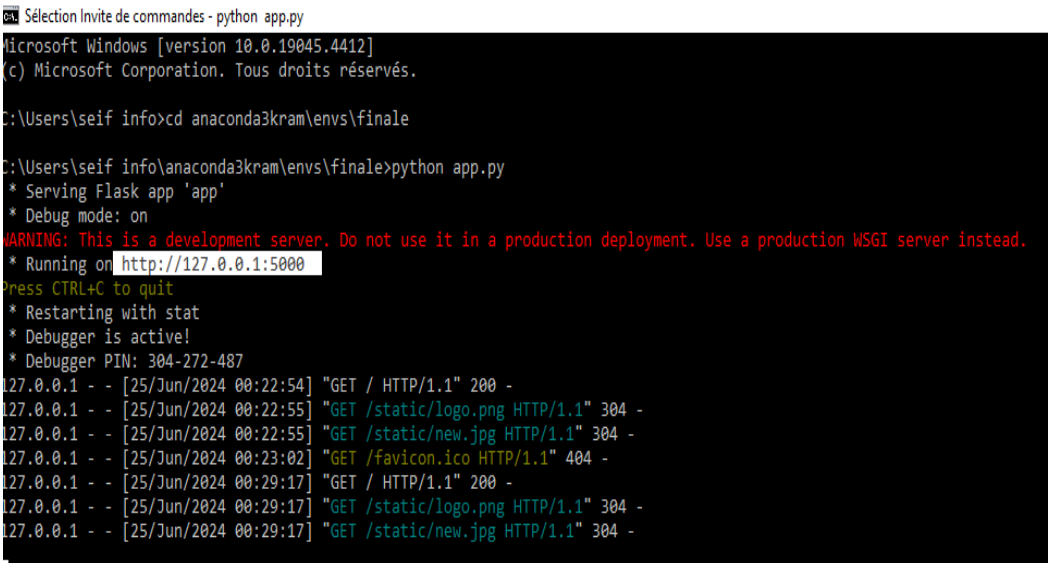


Figure 3 .9 : adresse locale après l'installation**.**

5.1 La page d'accueil

La page d'accueil de notre application web est conçue pour offrir une navigation simple et intuitive grâce à une barre de navigation (navbar) située en haut de la page. Cette barre de navigation comprend trois éléments principaux : "Home", "Contact" et "About".

1. **Home :** Cette section sert de point d'entrée principal pour les visiteurs. Elle présente un aperçu global du site et guide les utilisateurs vers les autres sections.

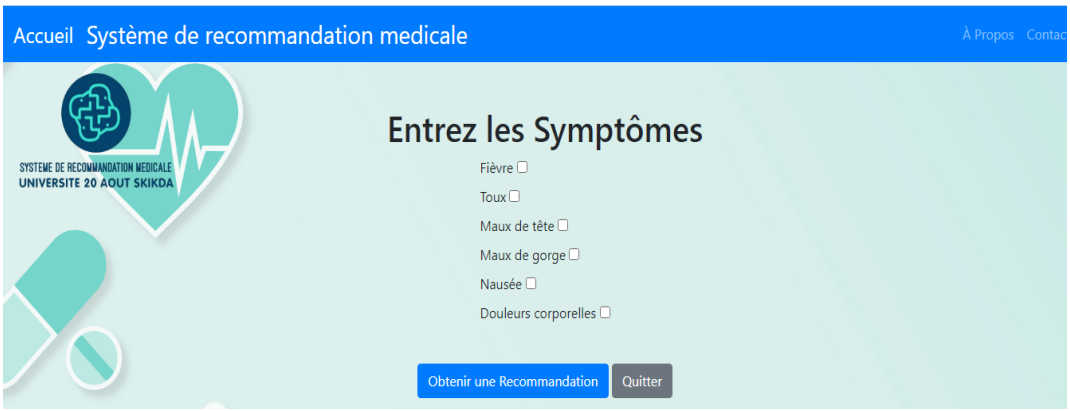


Figure 3.11 : Interface d'homme.

2. **Contact :** Cette page permet aux utilisateurs de nous contacter facilement.

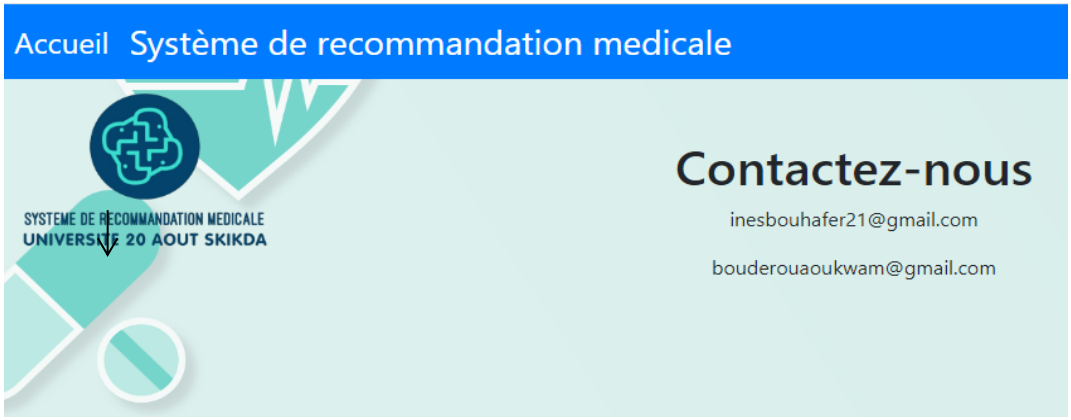


Figure 3.12 : Interface de contact.

3. About : Cette section fournit des informations sur les créateurs de l'application.

Figure 3.13 : Interface about.

5.2 Interface des Symptômes

En plus de la barre de navigation, la page d'accueil inclut une interface dédiée aux symptômes. Elle présente une liste de symptômes.

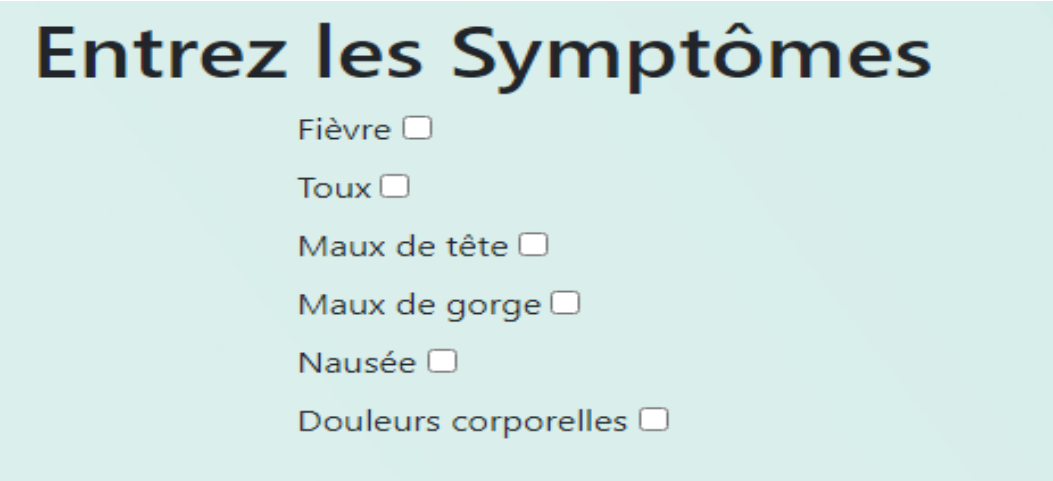


Figure 3.14 : Interface des Symptômes.

6. La recommandation

L'interface de recommandation de traitement médical vise à aider les professionnels de la santé à prendre des décisions éclairées en suggérant des traitements optimaux basés sur les données disponibles sur les patients et les connaissances médicales actuelles. Voici une vue détaillée sur comment une telle interface pourrait fonctionner, ses composantes clés et ses avantages.

6.1 La forme de la recommandation

Le but de cette interface est la recommandation de traitement pour cela il doit remplir le formulaire ci-dessous qui contient les informations suivantes :

Médicamentes, Symptômes, Maladies.

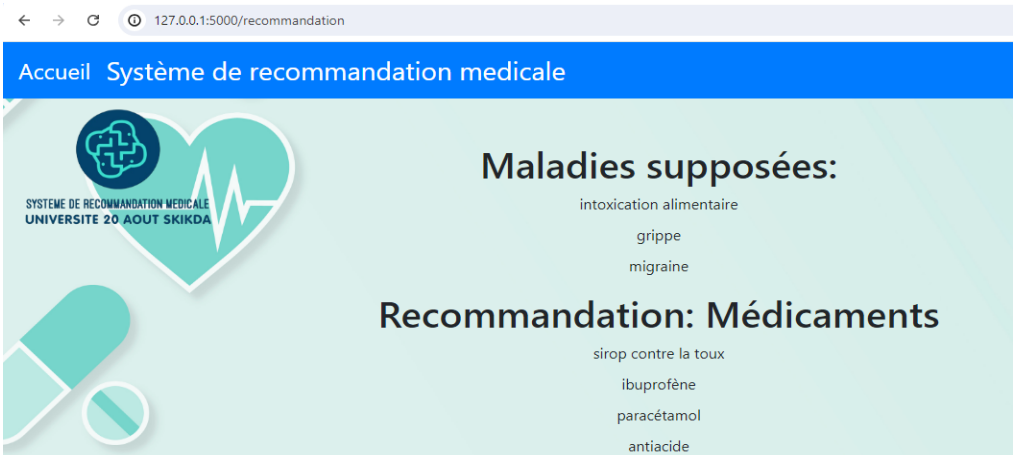


Figure 3.15 : Interface de recommandation.

7. Résultat et discussions

Résultat 1 de l'exemple

Support=100% (3/3), confiance=100%.

Algorithme : Apriori

Item set	Support
Maux de tète	**100%(3/3)**
Toux	**100%(3/3)**
Maux de gorge, toux	**100%(3/3)**
Fièvre	**100%(3/3)**
Maux de tête, nausée	**100%(3/3)**
Fièvre, Maux de tête, nausée	**100%(3/3)**
Douleur corporal	**100%(3/3)**

Id	La règle d'association	Support	Confiance	Lift
1	Toux, Fièvre =>grippe	33, 33	100%	1.0
2	Doleur corporelles, fièvres=> sirop contre la toux	33, 33	100%	1.0
3	Fièvre =>ibuprofène, toux	33, 33	100%	1.0

Règle (1…3) : ces règles d'association suggèrent une relation potentielle entre toux et Fièvre**.**

Règle (**2**) Cette règle concernant les associations est une règle forte règles d'association**.**

Résultat 1 de l'exemple 2

Support=100% (1/1), confiance=100%,

Item set	Support
grippe	**100%(1/1)**
paracétamol	**100%(1/1)**
douleurs corporelles	**100%(1/1)**
fièvre	**100%(1/1)**
toux	**100%(1/1)**
maux de tête	**100%(1/1)**

Id	La règle d'association	Support	Confiance	Lift
1	douleurs corporelles =>grippe	100%(1/1)	100%	2.25
2	douleurs corporelles => paracétamol	100%(1/1)	100%	2.25
3	douleurs corporelles, fièvre=>grippe	100%(1/1)	100%	2.25
4	douleurs corporelles=> grippe, fièvre	100%(1/1)	100%	2.25

5	Douleurs corporelles, toux =>paracétamol	100%(1/1)	100%	2.25
6	Douleurs corporelles=> paracétamol, grippe	100%(1/1)	100%	2.25
7	toux, fièvre=> grippe	100%(3/3)	100%	2.25
8	toux, fièvre=> paracétamol	100%(3/3)	100%	2.25
9	Maux de tête, toux =>grippe	100%(1/1)	100%	2.25
10	maux de tête, toux=> paracétamol	100%(1/1)	100%	2.25
11	douleurs corporelles, toux, fièvre=> grippe	100%(1/1)	100%	2.25
12	douleurs corporelles, toux=>grippe, fièvre	100%(1/1)	100%	2.25
13	Douleurs corporelles, toux, fièvre=>paracétamol	100%(1/1)	100%	2.25
14	douleurs corporelles, toux =>paracétamol, fièvre	100%(1/1)	100%	2.25
15	douleurs corporelles, fièvre=> paracétamol, grippe	100%(1/1)	100%	2.25
16	maux de tête, toux=> paracétamol, grippe	100%(1/1)	100%	2.25
17	douleurs corporelles,	100%(1/1)	100%	2.25

	toux, fièvre=> paracétamol, grippe			

Règle (1…3) Ces règles indiquent qu'il existe des relations fortes entre douleurs corporelles et grippe.

Règle (13…17) Ces règles indiquent qu'il existe des relations fortes entre douleurs corporelles et paracétamol.

Règle (13…15) : ces règles d'association suggèrent une relation potentielle entre Douleurs corporelles et paracétamol.

8. Conclusion

Dans ce chapitre, nous avons présenté les différents étapes de prétraitement des données tels que l'exploration et la visualisation des données ainsi le nettoyage des valeurs aberrantes.

Conclusion Générale

Les systèmes de recommandation automatique sont devenus à l'instar des moteurs de recherche, un outil incontournable pour tout site Web focalisé sur un certain type d'articles disponibles dans un catalogue riche, que ces articles soient des objets, des produits culturels (livres, films, morceaux de musique, etc.), des éléments d'information (news) ou encore simplement des pages (liens hypertextes). L'objectif de ces systèmes est de sélectionner, dans leur catalogue les items les plus susceptibles d'intéresser un utilisateur particulier, ont répertorié un vaste ensemble de systèmes de recommandation pour différents domaines applicatifs, dans des contextes académiques et industriels.

La tendance actuelle des systèmes de recommandation est plutôt axée sur des méthodes nouvelles, multicritères, multidimensionnelles ou encore se fondant sur des notions psychologiques comme les émotions, les opinions. Notons cependant qu'un système de recommandation doit avant tout s'adapter aux données, celles-là mêmes que l'on proposera à un utilisateur. Ainsi, le choix d'une méthode de recommandation doit en premier lieu êtres dirigé par ce critère. Le travail présenté dans ce mémoire rentre dans le cadre du filtrage collaboratif qui est la méthode la plus importante et la plus utilisée dans les systèmes de recommandation. Nous avons présenté expérimentalement, une étude comparative entre les deux méthodes de filtrage collaboratif.

Nos résultats ont montré que l'algorithme basé sur modèle est le meilleur de tous les autres algorithmes, ainsi que cet algorithme peut résoudre le problème scalabilité et de rareté des données, ainsi il permet d'améliorer la qualité et la performance du système. Comme perspectives nous envisageons d'apporter quelques améliorations à savoir : nous envisageons de rechercher sur comment donner une explication à une recommandation pour améliorer la confiance des systèmes de recommandation.

Bibliographie

[1] M. Nemiche. "Data mining," Master, Faculté des Sciences d'Agadir, Morocco, 2015.

[2] W. McCulloch, and W. Pitts, "a Logical Calculus of Ideas Immanent in Nervous Activity", Bulletin of Mathematical Biophysics 5:115-133, 1943.

[3] R. Agrawal, T. Imieliński, and A. Swami. "Mining association rules between sets of items in large databases, " in Proceedings of the 1993 ACM SIGMOD international conférence on Management of data, pp. 207–216, 1993.

[4] Pagé. C. "Bases de règles multi-niveaux", Université du Québec à Montréal, Février 2008.

[5] R. Rakotomala. "Arbre de décision", Revue MODULAD, numéro 33, 2005.

[6] H. Jiawei, and M. Kamber, "Data Mining Concepts and Techniques", published by Morgan Kauffman, 2nd Ed, 2006.

[7] Ain Picot-Clemente. Une architecture générique de Systèmes de recommandation de combinaison d'items. Application au domaine du tourisme. Recherche d'information [cs.IR]. Université de Bourgogne. Français. tel-00688994v1, 2011.

[8] S. Taouli, and W. Benachenhou. Utilisation de factorisation matricielle sans les systèmes de recommandation sensible au contexte. Mémoiremaster, université Abou Bakr Belkaid – Tlemcen, 2017.

[9] Netflix : comment fonctionne l'algorithme de recommandations. Futura Tech. [En ligne] [Consulté le : 20 avril 2020.] https://www.futura-sciences.com/tech/questions-reponses/informatique-netflixfonctionne-algorithme-recommandations-8640/.

[10] Filtrage collaborative. All You Need to Know About Collaborative Filtering (digitalvidya.com) (consulté le 02/02/2021).

[11] Charif ALCHIEKH HAYDAR. Les systèmes de recommandation à base de confiance. Université de Lorraine. Thèse de doctorat, 2014.

[12] Les réseaux bayésiens. PowerPoint Presentation (cnrs.fr) (consulté le 03/02/2021)

[13] Les modèles de clustering. Microsoft PowerPoint - ClusteringAssas.ppt (lip6.fr) (consulté le 03/02/2021).

[14] Factorisation matricielle. Aide de PTC Mathcad (consulté le 03/02/2021)

[15] Les arbres de décision. Cours - Arbres de décision — Cours Cnam RCP209 (consulté le 03/02/2021).

{16] Sharma, N.heartbeat.Understanding the Mathematics behind Support Vector Machines. [en ligne].Disponiblesur : https : //heartbeat.fritz.ai/understanding-themathematics-behindsupport-vector-machines-5e20243d64d5.

[17]Benjamin Marlé and Alexis PERRIER. Initiez-vous à python pour l'analyse de données, https : //openclassrooms.com/fr/courses/6204541-initiez-vous-a-pythonpour-lanalyse-de-donnees/6204548-installez-python-et-anaconda.Consulté le 10/05/2020, 2020.

[18] Fabio Nelli. The pandas library—an introduction. In Python Data Analytics, pages 87–139. Springer, 2018.

[19] laptrinhx. Naive Bayes Unlocked. [en ligne].Disponible sur :https://laptrinhx.com/naive-bayes-unlocked-1301819179. (2019).

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليـــــــــــم العالـــــــــــي و البحث العلمـــــــــــي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955 - Skikda
Faculté des Sciences
Bibliothèque de la faculté

جـامعة 20 أوت 1955- سكيكدة
كـــــــلية العلـــــوم
مكتبـة الكليـة

سكيكدة في / /2024

بطاقة معلومات خاصة بمذكرة التخرج

اسم و لقب الطالب : Bouderoua ikram

رقم التسجيل : 191936010383

Bouhafer ines

181836008365

اسم و لقب المشرف على المذكرة : Mansoul abdelhak

عنوان المذكرة : Le filtrage collaborative par modéle pour la recommandation de traitement médical

القسم : Informatique

المستوى : Master 2

التخصص : Génie logiciel avancé et application

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليـــــم العالـــــي و البحث العلمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955- skikdu-

Faculté des Sciences

Département d'Informatique

جـامعة 20 اوت 1955- سكيكدة

كـــلية العلـــوم

قسم الاعلام الالي

الرقم : ..../ ق/ ا/ ا/ ك.ع/2024

Autorisation de Dépôt de Mémoire de Master

Je soussigné: ...MANSOUL.....A.....

Certifie que l'étudiant(e) :...BOUDERAOA....Ikram...

Spécialité : Génie logiciel avancé et application.....

Ayant soutenu le projet intitulé : Le filtrage collaboratif par modèle pour la recommandation de traitement médical.....

A apporté les corrections nécessaires sur son manuscrit de Master

Signature de l'encadreur

le 17/07/24

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955- skikda- جامعة 20 أوت 1955- سكيكدة

Faculté des Sciences كلية العلوم

Département d'Informatique قسم الاعلام الالي

الرقم : ..../ و.ا.ت.ا/ك ع/2024

Autorisation de Dépôt de Mémoire de Master

Je soussigné: ...MANSOUL A...

Certifie que l'étudiant(e) :...BOUHAFER Ines...

Spécialité : ..Génie Logiciel avancé et application...

Ayant soutenu le projet intitulé : Le filtrage collaboratif par modèle pour la recommandation de traitement médical...

A apporté les corrections nécessaires sur son manuscrit de Master

Signature de l'encadreur

le 17/07/2024