République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERI de Tizi-Ouzou

Faculté de Génie Electrique et Informatique

Département Informatique

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention d'un diplôme de Master en Informatique

Spécialité : Ingénierie des Systèmes d'Informations

THEME

DEFINITION D'UN PROFIL UTILISATEUR POUR UN SYSTEME DE RECOMMANDATION EN RECHERCHE D'INFORMATION

Proposé et dirigé par :

Mme ACHEMOUKH Farida

Réalisé par:

DEKKAL Dyhia

MERIOUD Celia

JURY:

Présidente : Mme S.FELLAG

Examinatrice : Mme M.BENTAYEB

Session 2019

Remerciements

Nous tenons à remercier en premier lieu les personnes sans qui ce travail n'aurait jamais vu le jour.

Tout d'abord nous remercions notre promotrice ACHEMOUKH Farida, qui nous a offert l'opportunité de réaliser ce mémoire et pour sa patience, sa disponibilité et surtout ses judicieux conseils qui ont contribué à alimenter nos réflexions.

Nous tenons à remercier également les membres de jury d'avoir accepté d'évaluer notre travail.

Un énorme remerciement pour nos parents, frères et sœurs ainsi qu'à nos amis.

Dédicaces

Du profond de mon cœur je dédie ce modeste travail à tous ceux qui me sont chers,

A mon cher grand père que Dieu très haut lui accorde Santé et longue vie.

A mes chers parents,

A mes chers frères,

A mon binôme Dyhia,

Ainsi qu'à mes proches et mes amis(es).

Celia

Du profond de mon cœur je dédie ce modeste travail à tous ceux qui me sont chers,

A mes chers parents,

A mes chers frères,

A ma chère sœur,

A mon cher neveu "Aris" et chère cousine " Dalia ",

A mon binôme Celia,

A ma deuxième famille " Vo-Vietnam ",

Ainsi qu'à mes proches et mes amis(es).

Dyhia

TABLE DES MATIERES

TABLE DES MATIERES

INTRODUCTION GENERALE: 1

CHAPITRE I : ETAT DE L'ART SUR LES SYSTEMES DE RECOMMANDATION

1.1 Introduction : 3

1.2 Définition des systèmes de recommandation : 3

1.3 Concepts de base, notation et notions liées : 4

1.3.1 L'usager et l'item : 4

1.3.2 Evaluation (note ou vote) : 4

1.3.3 Filtrage d'information : 4

1.3.4 Matrice d'évaluation utilisateur-item : 6

1.3.5 La prédiction : 6

1.3.6 La personnalisation vs la recommandation : 7

1.4 Les techniques de recommandation : 7

1.4.1 Recommandation basé sur le contenu : 8

1.4.1.1 Approche générale : 8

1.4.1.2 Représentation d'un item : 10

1.4.1.3 Les types de recommandation basés sur le contenu : 11

1.4.1.3.1 Recommandation basé sur les vecteurs mots-clés : 11

1.4.1.3.2 Recommandation basée sur la sémantique : 13

1.4.1.3.3 Exemple de système de recommandation basé contenu : 14

1.4.1.3.4 Avantages et inconvénients des approches basées sur le contenu : 15

1.4.2 Recommandation basé sur le filtrage collaboratif : 16

1.4.2.1 Processus du filtrage collaboratif : 18

1.4.2.1.1 Evaluation des recommandations : 18

1.4.2.1.2 Formation des communautés : 19

1.4.2.1.3 Production des communautés : 19
1.4.2.1.3.1 Profils et communautés :.....19
1.4.2.2 Les approches du filtrage collaboratif basées sur les voisins : 20
1.4.2.3 Exemple de système de recommandation basé sur les filtrages collaboratif :....24
1.4.2.4 Avantages et inconvénients du filtrage collaboratif :.....25
1.4.3 Recommandation hybride : 26
1.5 Conclusion :.....28

CHAPITRE II : MODELISATION D'UN PROFIL UTILISATEUR

2.1 Introduction : 29
2.2 La modélisation des utilisateurs : 29
2.2.1 Représentation du profil utilisateur : 30
2.2.1.1 Représentation ensembliste : 30
2.2.1.2 Représentation sémantique à base d'ontologie :.....32
2.2.1.3 Représentation multidimensionnel :.....34
2.2.1.4 Représentation par matrice utilisateurs_items : 35
2.2.2 Construction du profil utilisateur :.....35
2.2.2.1 Acquisition d'information : 36
2.2.2.2 Prétraitement des données : 40
2.2.2.3 Les techniques de construction du profil utilisateur: 41
2.3 Adaptation et mise à jour du profil utilisateur :.....44
2.4 Conclusion :.....45

CHAPITRE III : MODELISATION DU PROFIL UTILISATEUR EN RECOMMANDATION

3.1 Introduction :……46

3.1 Problématique et motivations :……46

3.2 Description de l'approche proposée :……47

3.2.1 Architecture du système de recommandation : 47

3.2.2 Représentation du profil utilisateur : 47

3.2.3 Construction du profil utilisateur :.....48

3.2.4 Exploitation du profil utilisateur dans le processus de recommandation : 48

3.2.4.1 Inférence des centres d'intérêts :.....48

3.2.4.2 Calcul du poids pondérés des catégories pour chaque article :.....49

3.2.4.3 Intégration du profil utilisateur pour la prédiction:.....49

3.3 Illustration de l'approche proposée :……50

3.4 Conclusion :……..55

CHAPITRE 4 : IMPLEMENTATION DE NOTRE APPROCHE

4.1 Introduction : 56

4.2 Environnement de développement : 56

4.3 Aperçue de notre Implémentation: 57

4.3.1 Détail de l'approche :.....57

4.3.2 Tests et résultats :.....58

4.4 Conclusion :.....62

CONCLUSION GENERALE:……63

LISTE DES FIGURES

Figure 1.1 : Schéma général du filtrage d'information.....6
Figure 1-2: Architecture haut niveau d'un système de recommandation basé sur le contenu (d'après (Lops et al.,2011).....9
Figure 1-3 : Principe général du filtrage collaboratif.....18
Figure 2.1 : Un exemple de profil représenté par des mots clés.....31
Figure 2.2 : Exemple du profil utilisateur représenté par le modèle d'ontologie avec le processus de mise à jour des poids des concepts. (ahu sieg et al, 2007).....33
Figure 2.3 : Matrice utilisateurs items 35
Figure 2.4 : Les phases de construction du profil utilisateur 36
Figure 3.1 : Architecture du système de recommandation. 47

Tableau 3.1 : Représentation binaire des livres 50
Tableau 3.2 : Notes des utilisateurs sur les articles.....51
Tableau 3.3 : Total catégories 51
Tableau 3.4 : Les vecteurs articles 52
Tableau 3.5 : Les vecteurs profils utilisateur 52
Tableau 3.6 : Inverse document fréquence (IDF) 53
Tableau 3.7 : Vecteurs pondérés des articles 53
Tableau 3.8 : Degré de pertinence.....54

INTRODUCTION GÉNÉRALE

Les systèmes d'informations actuels sont caractérisés par leur volume croissant, leur hétérogénéité, et par le fait qu'ils ne sont pas suffisamment adaptés aux besoins des utilisateurs. Au vu de l'état actuel de ces systèmes en termes d'hétérogénéité de domaines, de sources, de représentation et de structuration des informations, l'accès a une information pertinente et adaptée aux utilisateurs est un vrai challenge. Les besoins de l'utilisateur sont difficiles a traiter, d'une part, parce qu'ils ne sont pas formulés explicitement et, d'autre part, parce qu'ils sont évolutifs.

L'utilisation des systèmes de recommandation est devenue une nécessité vu qu'ils permettent de fournir l'information pertinente avec moins d'effort et dans un délai de réponse satisfaisant, et les services de recommandation de nos jours ont propulsé la recherche d'information au premier plan vu que les systèmes de recherche d'information classique présentent un certain nombre de limites tel que l'expression du besoin de l'utilisateur par une requête qui s'avère complexe contrairement aux systèmes de recommandations qui offrent des informations qui correspondent aux besoins de l'utilisateur.

La majorité des systèmes de recommandation souffre du problème de démarrage à froid et plusieurs ressources d'informations sont exigées. Notre travail se situe dans ce contexte, notamment dans le cadre des systèmes de recommandation d'articles. Nous adoptons une approche basée sur le contenu qui évite le problème du démarrage à froid. Ce type de recommandation doit disposer d'informations sur les utilisateurs telles que leurs caractéristiques personnelles, leurs préférences, leurs centres d'intérêts etc, communément 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.

Notre travail est réparti sur quatre chapitres :

Chapitre 1 : Le premier chapitre présente une vue générale sur les systèmes de recommandation, nous définissons d'abord ce qu'est un système de recommandation. Ensuite

nous détaillons ses différents types, ainsi les avantages et inconvénients de chaque approche.

Chapitre 2: dans le deuxième chapitre nous verrons les modèles de représentation de profils utilisateurs, les méthodes d'acquisition des informations des utilisateurs, les techniques de constructions et de mise à jour des profils.

Chapitre 3 : Dans ce chapitre, nous expliquons l'approche que nous avons proposée.

Chapitre 4 : Implémentation de l'approche proposée, ce dernier chapitre se base sur les détails d'implémentation et de mise en œuvre de notre approche, ainsi qu'à la présentation des résultats obtenus.

Nous terminons notre mémoire par une conclusion générale.

CHAPITRE I

ETAT DE L'ART SUR LES SYSTÈMES DE RECOMMANDATION

CHAPITRE I

1.1 INTRODUCTION :

Le développement du Web a créé un besoin de nouvelles techniques pour aider les utilisateurs à trouver ce qu'ils recherchent mais aussi pour faire savoir qu'une information existe, ces techniques sont appelées ***les Systèmes de Recommandation*** .

Les préludes des systèmes de recommandation découlent de recherches menées dans la construction de modèles représentant les choix d'utilisateurs. Ces recherches sont issues de domaines distincts tels que la recherche documentaire, les sciences de gestion et marketing, les sciences cognitives et les théories d'approximation (Adomavicius et al., 2005) .

Dans ce premier chapitre nous donnons une définition des systèmes de recommandation ainsi que les concepts de base et notions liées au domaine. Ensuite, nous présentons les techniques de recommandation, ainsi que les avantages et les inconvénients de chaque approche.

1.2 DEFINITION DES SYSTEMES DE RECOMMANDATION :

Le terme «recommandation »consiste à: «*Conseiller particulièrement quelque chose, exhorter une personne à quelque chose* ».
Ainsi, la recommandation peut être vue comme un dialogue entre une personne experte d'un domaine donné et une autre personne qui souhaite acquérir des informations dans ce domaine.

Un système de recommandation est une forme spécifique de filtrage de l'information qui a pour but de présenter à un utilisateur des éléments qui sont susceptibles de l'intéresser, et ce, en se basant sur ses préférences et son comportement.

D'après la classification proposée dans (Burk R., 2002) un système de recommandation : *est tout* système *capable de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important*".

1.3 CONCEPTS DE BASE, NOTATION ET NOTIONS LIEES :

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

1.3.1 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.

1.3.2 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 triplet **<*u*, *i*, *r*>.** 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 all, 2003)**.**

1.3.3 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.

Les données d'entrée pour un système de recommandation dépendent du type de l'algorithme de filtrage employé. Généralement, elles appartiennent à l'une des catégories suivantes (Nguyen et al, 2006) :

- **Les estimations** :(également appelées les votes), expriment l'opinion des utilisateurs sur les articles (exemple : 1 mauvais à 5 excellent).

- **Les données démographiques** : se référant à des informations telles que l'âge, le sexe, le pays et l'éducation des utilisateurs.

- **Les données de contenu :** qui sont fondées sur une analyse textuelle des documents liés aux éléments évalués par l'utilisateur. Les caractéristiques extraites de cette analyse sont utilisées comme entrées dans l'algorithme de filtrage afin d'en déduire un profil d'utilisateur.

Pour réaliser le filtrage, le système de recommandation (SR) utilise les profils représentant des préférences relativement stables des utilisateurs pour calculer des recommandations. Ce calcul se fait par la prédiction des scores qu'un utilisateur est susceptible d'attribuer aux contenus.
Le SR adapte ce profil au cours du temps en exploitant au mieux le retour de pertinence que les utilisateurs fournissent sur les informations (documents) reçues.

Par exemple, dans la **figure 1.1**, la fonction de décision du système traite le flux entrant de document pour suggérer à l'utilisateur, en consultant son profil, les documents qu'il préfère. A son tour, l'utilisateur doit fournir des évaluations c'est-à-dire évaluer fréquemment les

recommandations, pour que le système comprenne mieux ses besoins en information, et lui fournisse par conséquent de meilleures nouvelles recommandations.

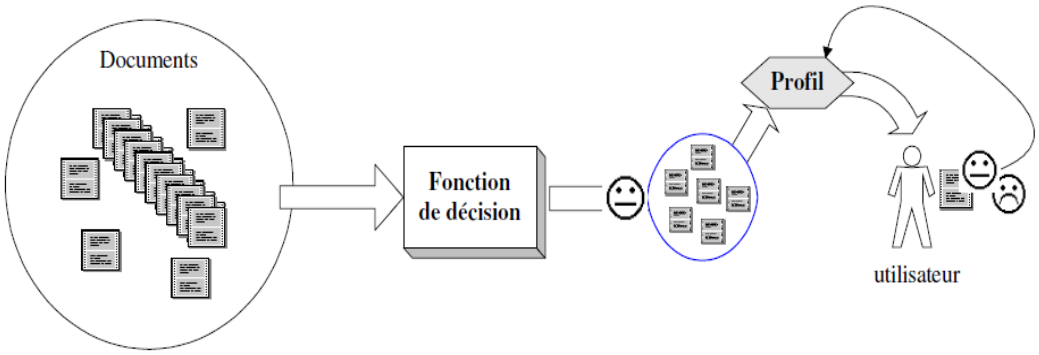

Figure 1.1 : Schéma général du filtrage d'information.

1.3.4 MATRICE D'EVALUATION UTILISATEUR-ITEM :

L'ensemble de tous les triplets du système <*u, i, r*> 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.

1.3.5 LA PREDICTION :

La prédiction est le calcul de la note probable que l'utilisateur va attribuer à un item qu'il n'a pas encore vu ou évalué.

En général, les matrices d'évaluation ont seulement quelques cellules contenant des valeurs tandis que les autres ont des valeurs inconnues et dans la majorité des cas elles ont à l'intérieur un"0", ce qui donne des matrices creuses. Donc, la densité de ces matrices ne sera pas suffisante pour générer des recommandations précises. Par conséquent, les méthodes de prédiction des évaluations manquantes sont utilisées pour augmenter la densité de la matrice utilisateur-item en vue de faire des recommandations plus puissantes et plus pertinentes.

1.3.6 LA PERSONNALISATION VS LA RECOMMANDATION :

La personnalisation est une notion proche de la notion de recommandation mais elle est moins générale et elle consiste à adapter un item aux goûts, aux besoins et parfois même au comportement de l'utilisateur.

Tandis qu'une recommandation génère une liste d'items plus ou moins adaptée aux besoins de l'utilisateur (c.à.d. elle ne garantie pas une personnalisation totale parce que les listes recommandées peuvent contenir des items nouveaux pour l'utilisateur ou différents de ces préférences, pour améliorer la satisfaction).

1.4 LES TECHNIQUES DE RECOMMANDATION :

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

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

De ces facteurs sont produits divers types de recommandations dont les plus utilisées dans la littérature sont les basés sur le contenu et le filtrage collaboratif.
Nous présentons ces deux approches ainsi que leurs hybridations.

Le filtrage basé sur le contenu compare les nouveaux documents au profil de l'utilisateur, et recommande ceux qui sont les plus proches. Le filtrage collaboratif compare les utilisateurs entre eux sur la base de leurs jugements passés pour créer des communautés, et chaque utilisateur reçoit les documents jugés pertinents par sa communauté. Le filtrage hybride combine le filtrage basé sur le contenu et le filtrage collaboratif pour exploiter au mieux les avantages de chacun.

1.4.1 RECOMMANDATION BASE SUR LE CONTENU :

Les systèmes de recommandation basés sur le contenu s'appuient sur des évaluations effectuées par un utilisateur sur un ensemble de documents ou items. L'objectif est alors de comprendre les motivations l'ayant conduit à juger comme pertinent ou non un item donnée.

La recommandation basée sur le contenu consiste à analyser le contenu des items candidats à la recommandation ou les descriptions de ces items. Les méthodes de recommandation basées sur le contenu utilisent des techniques largement inspirées du domaine de la recherche d'information. La différence se trouve essentiellement dans l'absence de requêtes explicites formulées par l'utilisateur. Les approches basées contenu infèrent plutôt les préférences de l'utilisateur et lui recommandent les items dont le contenu est similaire au contenu des items qu'il a aimés auparavant (Balabanovic et al., 1997), (Adomavicius et al., 2005), (Zhang et al., 2002), (Pazzani et al., 2007) .Ainsi, quand de nouveaux items sont introduits dans le système, ils peuvent être recommandés directement, sans que cela ne nécessite un temps d'intégration.

1.4.1.1 APPROCHE GENERALE :

Pour recommander des items en se basant sur le contenu, deux ensembles doivent être constitué: les profils des items et les profils des utilisateurs.

La notion de contenu ne se rapporte donc pas uniquement au contenu des items, mais également aux attributs descriptifs des utilisateurs. Une approche basée contenu analyse un ensemble d'items précédemment notés ou consultés par un utilisateur, et construit un modèle ou un profil des intérêts de l'utilisateur sur la base des caractéristiques des items aimés ou détestés par celui-ci. En fonction de ses feedbacks, le profil de l'utilisateur est construit il est souvent constitué d'un profil "positif" représentant les items qu'il a aimés et d'un profil "négatif" représentant les items qu'il a détestés.

Le processus de recommandation consiste donc essentiellement à comparer les attributs des items candidats avec les attributs du profil "positif" et "négatif" de l'utilisateur. De ce fait, les items qui seront recommandés à l'utilisateur sont les items qui sont similaires à son profil "positif" et moins similaires à son profil "négatif". Plus le profil de l'utilisateur construit reflète les préférences de l'utilisateur, plus le système de recommandation peut être efficace.

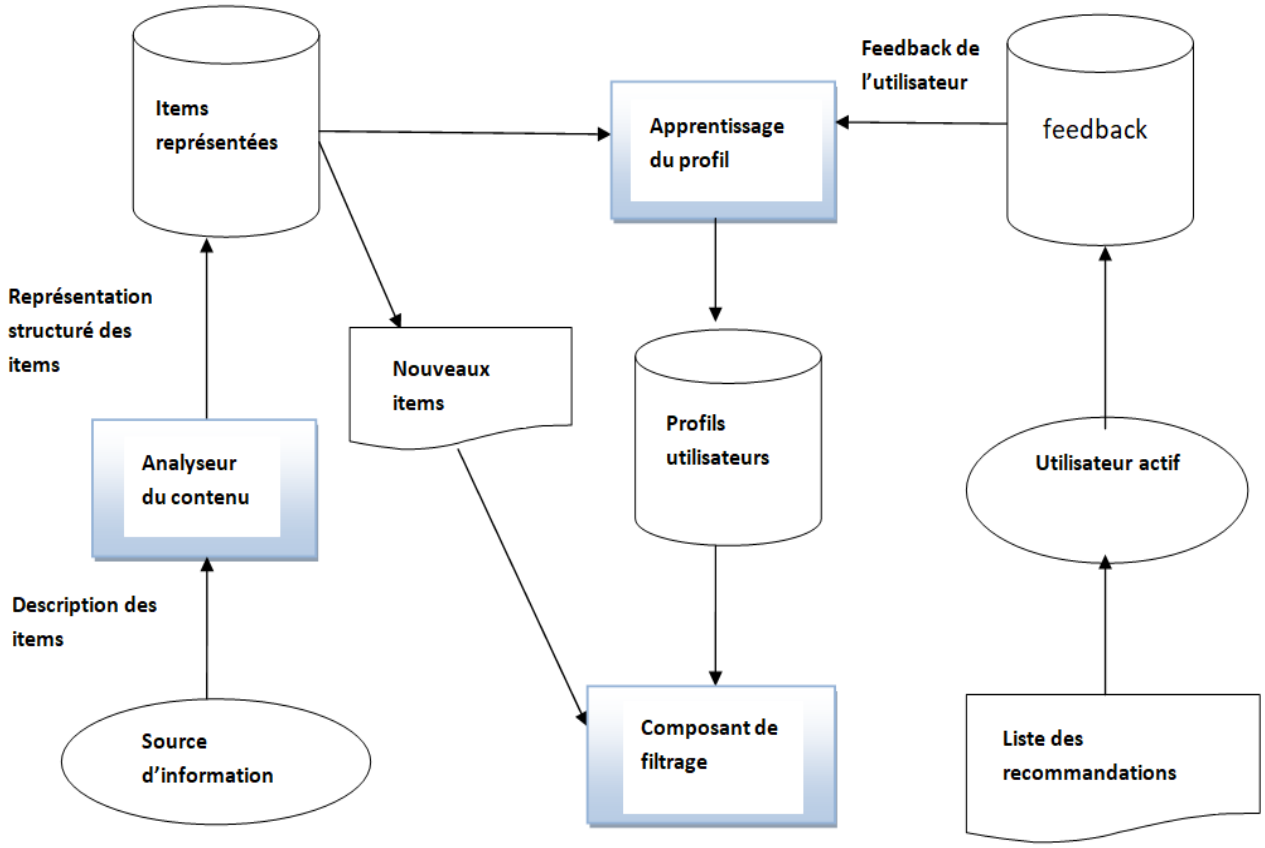

Figure 1.2: Architecture haut niveau d'un système de recommandation basé sur le contenu (d'après (Lops et al ., 2011).

Un système de recommandation basé sur le contenu a besoin de techniques pour produire une représentation efficace des items et du profil de l'utilisateur pour pouvoir les comparer. Ainsi (Lops et al ., 2011) proposent une architecture de haut niveau (**Figure 1.2**) dans laquelle le processus de recommandation est réalisé en trois étapes, chacune de ces étapes est gérée par un composant spécifique :

- **Analyseur du contenu :** Lorsque l'information n'est pas structurée (par exemple, un item représenté par un texte), ce module a pour but de réaliser le prétraitement pour extraire l'information pertinente, la structurer et la représenter dans une forme cible appropriée (par exemple un vecteur de mots clés).

- **Apprentissage du profil :** Ce module collecte les données représentatives des préférences de l'utilisateur et généralise ces données, afin d'apprendre et de construire le

profil de l'utilisateur. Des techniques d'apprentissage automatique (Michalski et al., 2013) peuvent être utilisées pour cela. On peut citer à titre d'exemple les arbres de décisions, les réseaux de neurones et la classification naïve de Bayes. Ces techniques visent à inférer un profil de l'utilisateur en utilisant l'information sur les items qu'il a aimés ou n'a pas aimés.

- **Composant de filtrage :** Ce module filtre les items pertinents en faisant correspondre la représentation du profil utilisateur aux items candidats à la recommandation. La pertinence de l'item est calculée en utilisant des métriques de similarité entre l'item considéré et le profil de l'utilisateur. Plus la similarité avec le profil "positif" est grande et plus la similarité avec le profil "négatif" est petite, plus l'item a des chances d'être recommandé.

Afin de construire et mettre à jour le profil de l'utilisateur actif, ses réactions aux items (notes) sont recueillies et enregistrées dans le composant Feedback. Ces notes d'intérêt sont exploitées au cours du processus d'apprentissage du modèle utile pour prédire la pertinence a priori d'un item que l'utilisateur n'a pas encore noté. Les utilisateurs peuvent aussi définir explicitement leurs domaines d'intérêt au préalable comme profil initial, mais ce cas est assez rare.

1.4.1.2 REPRESENTATION D'UN ITEM :

Dans la plupart des systèmes basés sur le contenu, la description de l'item est sous forme de texte ou extrait de pages web , email ...
Contrairement aux données structurées, il n'y a pas d'attributs avec des valeurs bien définies. Cela rend plus difficile l'apprentissage du profil utilisateur, en raison de l'ambiguïté du langage naturel, et notamment de la polysémie (multiples significations pour un mot) et de la synonymie (plusieurs mots qui ont le même sens).
Un prétraitement peut alors être nécessaire afin d'extraire l'information pertinente et de la structurer sous forme d'un ensemble d'attributs.

1.4.1.3 LES TYPES DE RECOMMANDATION BASES SUR LE CONTENU :

On distingue deux types de recommandation basé sur le contenu: recommandation basé sur les **mots clés** et recommandation basé sur la **sémantique**.

1.4.1.3.1 RECOMMANDATION BASEE SUR LES VECTEURS MOTS-CLES :

La plupart des systèmes de recommandation basés sur le contenu utilisent le modèle de représentation vectoriel VSM (Vector Space Model) avec la pondération classique TF-IDF (Term Frequency-Inverse Document Frequency). Dans ce modèle, chaque document (qui représente un item) est représenté par un vecteur de dimension n, où une dimension correspond à un terme de l'ensemble du vocabulaire d'une collection de documents. Formellement, tout document est représenté par un vecteur poids sur des termes, où chaque poids indique le degré d'association entre le document et le terme.

Soit $D = \{d_1, d_2, \dots\dots, d_n\}$ dénotant un ensemble de documents ou corpus, et $T = \{t_1, t_2, \dots, t_n\}$ le dictionnaire, c'est-à-dire l'ensemble des mots du corpus.

T est généralement obtenu en appliquant des opérations de traitement du langage naturel, comme l'atomisation (tokenization), l'élimination des mots vides de sens, et la troncature (stemming) (Baez-yates et al, 1999). Chaque document d_j est représenté par un vecteur dans un espace vectoriel à n dimensions, tel que $d_j = \{w_{1j}, w_{2j}, \dots\dots, w_{nj}\}$ où w_{kj} est le poids du terme t_k dans le document d_j.

La représentation de documents en utilisant le modèle d'espace vectoriel fait apparaître deux difficultés : la **pondération** des termes et la mesure de **similarité** des vecteurs représentant les documents.

La méthode de pondération de termes la plus couramment utilisée est la pondération TF.IDF, qui est basée sur des observations empiriques sur le texte (Salton, 1989):

- Des occurrences multiples d'un terme dans un document sont souvent plus pertinentes que de simples occurrences (TF) ;
- Les termes rares ne sont pas forcément moins discriminants par rapport aux termes fréquents (IDF) ;
- Des documents longs ne sont pas préférables à des documents plus courts.

Plus explicitement, les termes qui apparaissent fréquemment dans un document, mais rarement dans le reste du corpus ont plus de chances de représenter le sujet du document (Lops et al ., 2011).De plus, la normalisation des vecteurs résultats empêche les documents longs d'avoir plus de chances d'être retrouvés que les documents courts. Cela est bien pris en compte par la fonction TF-IDF (Sparck Jones, 1972):

$$TF.IDF\ (t_k, d_j) = TF\ (t_k, d_j) \times log\ \left(\frac{N}{n_k}\right) \quad (1.1)$$

Où N dénote le nombre de documents dans le corpus, et n_k représente le nombre de documents de la collection dans lesquels le terme t_k apparaît au moins une fois, avec :

$$TF\ (t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \quad (1.2)$$

Où : $f_{k,j}$ représente le nombre d'occurrences du terme t_k dans le document d_j,

$max_z f_{z,j}$ est le maximum des fréquences $f_{z,j}$ des termes t_z apparaissant dans le document d_j,

Afin que tous les poids appartiennent à l'intervalle $[0,1]$, et que tous les documents soient représentés par des vecteurs de même longueur, les poids obtenus par la fonction TF.IDF sont généralement normalisés en utilisant la normalisation cosinus :

$$w_{k,j} = \frac{TF\ IDF\ (t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} TF\ IDF\ (t_s, d_j)^2}} \quad (1.3)$$

Une fois que les poids sont calculés et normalisés. Le contenu d'un item d_j, est défini par :

$$Content(d_j) = (w_{1j}, w_{2j}, \dots\ w_{kj}) \quad (1.4)$$

Après cette étape de pondération des termes et de normalisation, il faut définir une mesure de similarité des vecteurs caractéristiques. Cette mesure de similarité est requise pour déterminer la proximité entre deux documents. Il existe de nombreuses mesures de similarité, mais la mesure la plus largement utilisée dans la littérature est la similarité cosinus :

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \times \sqrt{\sum_k w_{kj}^2}} \quad \text{..... (1.5)}$$

Dans les systèmes de recommandation basés sur le contenu s'appuyant sur un modèle d'espace vectoriel, les profils des utilisateurs et les items sont représentés comme des vecteurs de termes pondérés. Notons $ContentBasedProfile(u)$ le profil de l'utilisateur u contenant ses préférences. Ce profil est obtenu en analysant le contenu des items qu'il a notés auparavant, et il est souvent défini par un vecteur de mots clés en utilisant des techniques issues du domaine de la recherche d'information.

Plus formellement, $ContentBasedProfile(u)$ est défini comme un vecteur de poids $(w_{1u}, w_{2u}, \dots w_{ku})$ où chaque poids w_{iu} dénote l'importance de l'attribut k_i par rapport aux préférences de l'utilisateur u. Ce vecteur de poids représentant le profil de l'utilisateur peut être obtenu à partir des vecteurs du contenu des items que l'utilisateur a notés en utilisant différentes techniques. Par exemple, l'algorithme de Rocchio (Rocchio et al, 1971) a largement été utilisé pour déterminer $ContentBasedProfile(u)$ comme étant le vecteur moyen à partir des vecteurs des items notés. La prédiction de l'intérêt d'un utilisateur pour un item qu'il n'a pas encore noté peut être effectuée par le calcul de la similarité cosinus entre le vecteur du profil utilisateur et le vecteur de l'item.

1.4.1.3.2 RECOMMANDATION BASEE SUR LA SEMANTIQUE :

L'algorithme de filtrage basé sur le contenu peut réaliser le matching entre un descripteur de contenu (comme par exemple, documents, livre, etc.) et un profile utilisateur et détermine le degré de pertinence de chaque article (ou contenu) pour les utilisateurs potentiels. Si de nombreux articles s'accumulent dans un certain laps de temps, l'algorithme de filtrage de contenu peut ordonner les articles en fonction de leur pertinence pour chacun des utilisateurs potentiels.

- **Représentation des contenus - le descripteur d'article :** Un descripteur d'article se compose d'un ensemble de concept qui peuvent être représentés par une ontologie de domaine. Les concepts qui représentent un élément sont les plus spécialisés dans une branche de la hiérarchie. De toute évidence, un article peut être représenté avec de nombreux concepts de l'ontologie, chaque concept peut apparaître dans n'importe quelle branche de la hiérarchie

de l'ontologie et à tout niveau cela dépend du contenu réel de cet article. Il est à noter que le profil peut inclure des concepts frères, c'est-à-dire les fils d'un même concept.

- **Représentation des utilisateurs - le profil d'utilisateur :** Un profil utilisateur basé sur le contenu se compose d'une liste pondérée de concepts de l'ontologie, représentant ses préférences (ses intérêts). De toute évidence, le profil de l'utilisateur peut comporter de nombreux concepts de l'ontologie, chacun figurant dans les différentes branches et différents niveaux de la hiérarchie. Par exemple, le profil de l'utilisateur peut inclure uniquement « sport », ou « sport » et « football », ou « football » et « basketball », ou tous les trois - en plus de nombreux autres concepts. Cela signifie qu'un certain concept dans un descripteur d'article peut être comparé avec plus d'un concept équivalent dans le profil de l'utilisateur.

- **Similarités entre un descripteur d'article et un profil utilisateur** :

Un descripteur d'article et un profil utilisateur sont semblables à un certain degré si leurs profils comprennent des concepts communs (le même) ou des concepts relatifs, c'est-à-dire des concepts ayant une sorte de relation père-fils. Un descripteur d'article et un profil utilisateur peuvent avoir de nombreux concepts communs ou relatifs; de toute évidence, plus les concepts sont communs ou relatifs, plus forte est leur similitude. Par exemple, si le profil de l'utilisateur inclut « football » et « sport », ce profil est similaire (à un certain degré) à un article qui comprend ces deux concepts, mais il est moins semblable à un article incluant juste « sport », et il est plus semblable à un article, comprenant « sport » et « football ».

1.4.1.4 EXEMPLE DE SYSTEME DE RECOMMANDATION BASE CONTENU :

Un système de recommandation basé sur le contenu a été proposé par (Chandrasekaran *et* al.,) pour recommander les documents scientifiques susceptibles d'intéresser les auteurs connus de la base de données *CiteSeer*. Pour chaque auteur participant à l'étude, ils ont créé un profil utilisateur basé sur les documents publiés antérieurement. Sur la base de similitudes entre le profil utilisateur et les profils des documents de la collection, des documents seront recommandés à l'auteur. Contrairement à la représentation traditionnelle, les profils des utilisateurs et des items ont été représentés par des arbres de concepts dans ce système. Ensuite, la similarité entre les profils utilisateurs et les profils documents est calculée à travers un algorithme de *matching* d'arbre en utilisant une mesure de distance arbre-édit.

1.4.1.5 AVANTAGES ET INCONVENIENTS DES APPROCHES BASEES SUR LE CONTENU :

Les approches basées sur le contenu présentent plusieurs avantages et inconvénients. Les points forts de ces approches consistent à:

- **L'autonomie de l'utilisateur :** les techniques de recommandation basées sur le contenu traitent chaque utilisateur de façon indépendante. Ainsi, seules les évaluations de l'utilisateur lui-même sont prises en compte pour construire son profil utilisateur et faire la recommandation, ce qui n'est pas le cas pour les approches utilisant le filtrage collaboratif.

- **La prise en compte immédiate d'un nouvel item :** le filtrage basé sur le contenu peut recommander des items nouvellement introduits dans la base avant même qu'ils reçoivent une évaluation de la part d'un utilisateur, au contraire des approches collaboratives qui ne peuvent recommander un item que s'il a été préalablement évalué par un groupe d'utilisateurs.

Cependant les approches de recommandation basées sur le contenu présentent aussi de nombreux inconvénients :

- **Limite de l'analyse du contenu :** une limite naturelle de la recommandation basée sur le contenu est la nécessité de disposer d'une représentation variée et riche du contenu des items, ce qui n'est pas toujours le cas. La précision des recommandations est liée à la quantité d'informations dont dispose le système pour discriminer les items appréciés de ceux non appréciés par l'utilisateur (Lops et al ., 2011) .Contrairement au filtrage collaboratif qui peut traiter tout type d'items sans aucune information sur leur contenu, l'approche basée sur le contenu ne peut traiter que les items disposant d'un contenu pouvant être analysé.

- **Sur-spécialisation (Over-specialization) :** le système ne peut recommander que les items qui sont similaires au profil utilisateur. L'utilisateur ne peut donc recevoir que des recommandations proches des items qu'il a notés ou observés par le passé (Adomavicius et al., 2005).Or, la diversité des recommandations est souvent appréciée et s'avère être un critère d'évaluation important des systèmes de recommandation (Yu et al., 2009). Idéalement, l'utilisateur doit recevoir des recommandations pertinentes et diversifiées.

Par exemple, il n'est pas intéressant de recommander toutes les chansons de Jacques Brel à un utilisateur qui a aimé l'une de ces chansons.

- **Intégration d'un nouvel utilisateur non immédiate :** un utilisateur doit évaluer un certain nombre d'items avant que le système ne puisse interpréter ses préférences et lui fournir des recommandations pertinentes (Ricci et al., 2011). Ce problème est connu dans la littérature sous le nom du problème de démarrage à froid pour les utilisateurs (user cold start).

- **Les difficultés à recommander des documents multimédia** (images, vidéos, etc.). Ceci a cause de la difficulté d'indexer ce type de documents qui pose le problème de la prise en compte de l'information structurelle des documents pour aider à identifier les contenus multimédias pertinents. c'est en fait la même problématique dont souffrent les systèmes de recherche.

1.4.2 RECOMMANDATION BASEE SUR LE FILTRAGE COLLABORATIF :

Le filtrage collaboratif (*Collaborative Filtering* « CF ») a pour principe d'exploiter les évaluations faites par des utilisateurs sur certains documents, afin de recommander ces mêmes documents à d'autres utilisateurs, et sans qu'il soit nécessaire d'analyser le contenu des documents.

Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable, et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grâce à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible, soit difficile à analyser, et en particulier elle peut être utilisé pour tout type de données : texte, image, audio et vidéo.

On distingue généralement deux sous-familles principales du filtrage collaboratif : Les méthodes basées sur la mémoire (memory-based) et les méthodes basées sur un modèle (model-based).

Les algorithmes de filtrage collaboratif basés sur la mémoire (basés sur des heuristiques) selon (Adomavicius et al., 2005)ou plus fréquemment basés sur les voisins (Desrosiers et al, 2011) utilisent les notes des utilisateurs stockés en mémoire pour faire de la prédiction.

Les algorithmes basés sur un modèle construisent en offline une image réduite de la matrice des notes dans un objectif de réduire la complexité des calculs et/ou de traiter le problème des notes manquantes. Le modèle passe d'abord par une étape d'apprentissage, puis, il est utilisé pour faire de la recommandation.

Plusieurs méthodes ont été utilisées pour les algorithmes de recommandation basés sur un modèle. On peut citer, parmi les plus abouties:

- les méthodes de réduction de la dimension appelées SVD (décomposition en valeurs singulières) (Koren et al, 2011),
- les approches probabilistes (Breese et al, 1998),
- les approches basées sur le clustering (Ungar et al, 1998) ,
- les approches basées sur les règles d'association (Heckerman et al, 2001).

Dans le cadre des approches basées modèles, la prédiction peut être faite de deux façons différentes:

- A partir de la prédiction fournie par le modèle lui-même, en construisant par exemple un modèle probabiliste pour l'estimation des valeurs de prédiction ou directement à partir du modèle.
- Ou bien, en regroupant les utilisateurs\ items par les méthodes de *clustering* et par la suite, les méthodes basées mémoires (basées utilisateurs ou basées items) seront utilisées pour prédire les évaluations pour les items.

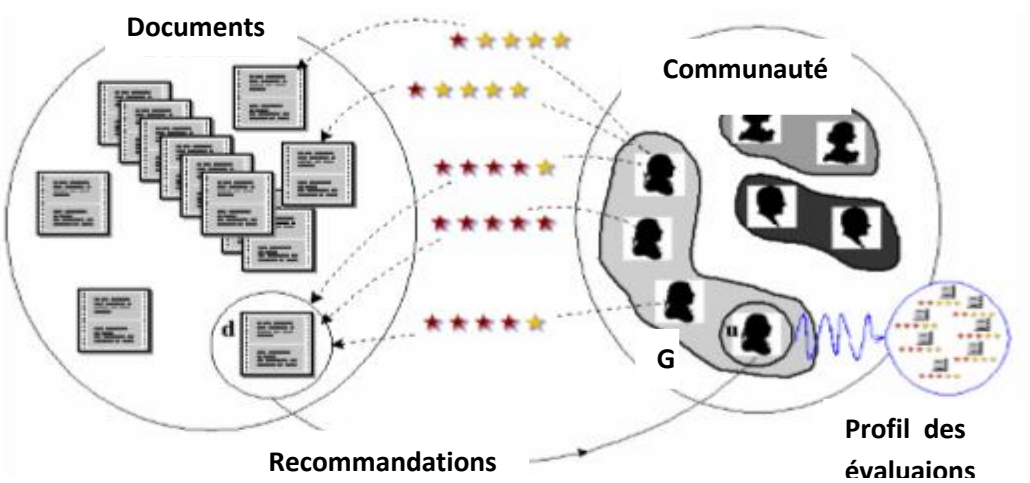

Figure 1-3 : Evaluation général du filtrage collaboratif.

Dans la Figure 4, supposons que l'on a des communautés formées par la proximité des évaluations des utilisateurs. Le document d sera recommandé à l'utilisateur u , car ce document est apprécié de la communauté G où se trouve l'utilisateur.

1.4.2.1 PROCESSUS DU FILTRAGE COLLABORATIF :

Le processus du filtrage collaboratif suit les étapes données ci-dessous :

1.4.2.1.1 EVALUATION DES RECOMMANDATIONS :

Selon le principe de base du filtrage collaboratif, les utilisateurs doivent fournir leurs évaluations sur des documents afin que le système forme les communautés. Evaluer une recommandation peut se faire de façon explicite ou implicite, comme suit :

- **Explicite** : L'utilisateur donne une valeur numérique sur une échelle donné (par exemple de 1 à 5, ou de 1 à 10, etc.), ou bien, une valeur qualitative de satisfaction, par exemple, mauvaise, moyenne, bonne et excellente.

- **Implicite** : Le système induit la satisfaction de l'utilisateur à travers ses actions. Par exemple, le système estimera qu'une recommandation supprimée correspond à une évaluation très mauvaise, alors qu'une recommandation imprimée ou sauvegardée peut être interprétée comme une bonne évaluation.

1.4.2.1.2 FORMATION DES COMMUNAUTES :

Le processus de formation des communautés est le noyau d'un système de filtrage collaboratif. Pour chaque utilisateur, le système doit calculer sa communauté, généralement cela se fait par la proximité des évaluations des utilisateurs. Pour ce faire, on peut calculer, dans un premier temps, la proximité entre un utilisateur donné et tous les autres. Ensuite, et afin de créer contrairement la communauté de l'utilisateur, en appliquant la méthode des voisins les plus proche et en utilisant un seuil pour le niveau de proximité ou un seuil pour la taille maximale de la communauté, en raison de sa performance et sa précision.

1.4.2.1.3 PRODUCTION DES COMMUNAUTES :

Dans ce derniers processus, une fois la communauté de l'utilisateur crée, le système prédit l'intérêt qu'un document particulier à présenter pour l'utilisateur en s'appuyant sur les évaluations que les membres de la communauté ont faites sur ce même document. Lorsque l'intérêt prédit dépasse un certain seuil, le système recommande le document à l'utilisateur.

1.4.2.1.4 PROFILS ET COMMUNAUTES :

Les profils basés sur l'historique des évaluations des utilisateurs, ainsi que les communautés, sont les deux facteurs clés d'un système de filtrage collaboratif (SFC). Le problème de la surcharge d'information peut être pallié par la personnalisation de l'accès aux informations, en utilisant des *profils* représentant des intérêts relativement stables des utilisateurs. En d'autres termes les profils des utilisateurs sont utilisés comme des critères persistant dans la recherche d'information.

1) Profil utilisateur :

Le profil utilisateur est composé de prédicats pondérés. Le poids d'un prédicat exprime son intérêt relatif pour l'utilisateur. Il est spécifié par un nombre réel compris entre 0 et 1. Le profil s'enrichit progressivement au fur et à mesure que l'utilisateur évalue des documents reçus. Outre les informations d'identification de base (par exemple, l'identifiant ou des éléments d'état civil), le profil de l'utilisateur peut regrouper des informations très diverses selon les besoins.

2) Communautés :

La notion de communauté dans un système de filtrage collaboratif est définie comme le regroupement des utilisateurs en fonction de l'historique de leurs évaluations, afin que le système calcule des recommandations.

Selon cette optique, les profils sont un facteur interactif, alors que les communautés sont considérées comme un facteur interne du système.

1.4.2.2 LES APPROCHES DU FILTRAGE COLLABORATIF BASEES SUR LES VOISINS :

Les algorithmes de filtrage collaboratif basés sur les voisins (Nakamura et al, 1998) (Delgado et al, 1999) utilisent généralement la totalité de la matrice des notes des utilisateurs pour faire la recommandation. On parle d'approche des k plus proche voisins (ou k-Nearest Neighbours - kNN).

Ces approches sont regroupées en deux familles : basés sur les utilisateurs (user-user collaborative filtering) ou basés sur les items (item-item collaborative filtering). Pour les algorithmes basés sur les utilisateurs tels que GroupLens (Resnick et al., 1994) ou Ringo (Shardanand and Maes et al ., 1995), l'appréciation estimée d'un utilisateur u pour un item i est prédite en utilisant les notes de ses voisins (ses utilisateurs similaires, avec lesquels ils partagent les mêmes préférences). De manière analogue, les algorithmes basés sur les items (Linden et al., 2003), (Sarwar et al, 2001)déterminent l'appréciation estimée d'un utilisateur u pour un item candidat i à partir des notes de u pour les items voisins de i.

Nous détaillons dans la suite ces deux types d'algorithmes.

On note par $\bar{r}_u$ la moyenne des notes données par l'utilisateur u sur les items qu'il a notés et par la $\bar{r}_i$ moyenne des notes reçues par l'item i .

$$\bar{r}_u = \frac{\sum_{i \in I_u} r_{u,i}}{|I_u|} \quad \text{(1.6)}$$

$$\bar{r}_i = \frac{\sum_{u \in U_i} r_{u,i}}{|U_i|} \quad \text{(1.7)}$$

On note également par $sim(u, v)$ la fonction mesurant la similarité entre les deux utilisateurs u et v, et par $sim(i, j)$ la similarité entre les deux items i et j. On définit $I_{UV} =$

$I_u \cap I_v$ comme étant l'ensemble des items notés à la fois par les utilisateurs u et v, et de façon équivalente $U_{ij} = U_i \cap U_j$ l'ensemble des utilisateurs ayant noté à la fois les items i et j.

A. Filtrage basé sur les utilisateurs :

Le filtrage collaboratif basé sur les utilisateurs a été introduit pour la première fois dans le système GroupLens (Resnick et al., 1994) , son principe de fonctionnement est très simple : déterminer les utilisateurs qui sont similaires à l'utilisateur courant, puis calculer une valeur de prédiction pour chaque item candidat à la recommandation en analysant les notes que les voisins de l'utilisateur courant ont exprimées sur cet item.

❖ Calcul de la similarité :

La similarité entre deux utilisateurs u et v peut être mesurée en utilisant la similarité Cosinus ou bien en utilisant le coefficient de corrélation de Pearson . Selon (Shafar et al, 2007), le coefficient de Pearson est le plus utilisé dans la littérature. C'est aussi le plus performant en termes de pertinence des recommandations.

❖ Calcul de la prédiction :

Pour le calcul de la prédiction, est adoptée par les auteurs de Groupelens (Resnick et al., 1994) la formule (formule 1.8) est comme suit :

$$pred(u,i) = \bar{r}_u + \frac{\sum_{w \in voisin(u) \cap U_i} sim(w,u) \times (r_{w,i} - \bar{r}_w)}{\sum_{w \in voisin(u) \cap U_i} |sim(w,u)|} \quad \text{..... (1.8)}$$

B. Filtrage basé sur les items :

Le filtrage collaboratif à base d'items a été introduit par (Sarwar et al., 2001). La prédiction de la note de l'utilisateur u pour un item candidat i est calculée à partir de ses notes pour les items voisins (similaires) de i. Son principe de fonctionnement est le suivant : pour l'item i candidat à la recommandation, on détermine les voisins les plus proches (les items similaires) en calculant sa similarité avec les autres items disponibles et on calcule

ensuite la prédiction de la note de l'utilisateur courant u pour l'item i à partir des notes que u a attribué à aux voisins de i .

❖ **Calcul de la similarité**:

La similarité entre deux items i et j peut être calculée en utilisant soit le Cosinus soit le coefficient de Pearson soit le cosinus ajusté. Cependant, une étude expérimentale menée par les auteurs de (Sarwar et al, 2001), comparant les trois mesures, a montré que le Cosinus ajusté est le plus performant en termes de pertinence de prédiction.

❖ **Calcul de la prédiction :**

la prédiction de la note de l'utilisateur courant u pour un item candidat à la recommandation i revient à calculer une moyenne pondérée de ses notes sur l'ensemble des items similaires à i. Chaque note $r_{u,j}$ est pondérée par la similarité de l'item j avec l'item i. Afin d'avoir une prédiction dans le même intervalle de valeurs que les notes, la prédiction est divisée par la somme des similarités. L'ajustement de la note est inutile dans ce cas puisqu'il s'agit du même utilisateur. L'équation (1.9) donne la formule exacte utilisée par (Sarwar et al, 2001).

$$pred(u,i) = \frac{\sum_{i \in I_u} sim(i,j) \times r_{u,j}}{\sum_{i \in I_u} |sim(i,j)|} \quad \text{(1.9)}$$

Comme pour le filtrage basé sur les utilisateurs, les auteurs (Sarwar et al, 2001) ont démontré que la pertinence des prédictions est très sensible au nombre de voisins considérés dans la formule. Ainsi, parmi les items notés par u seuls les k plus proches voisins de i sont pris en compte pour aboutir à de meilleures recommandations et gagner en temps de calcul.

C. Calcul de la similarité :

Le calcul de la similarité a pour objectif de déterminer dans quelle mesure deux utilisateurs ou deux items sont similaires. Il existe plusieurs façons de calculer cette similarité, cependant les méthodes les plus utilisées et qui présentent les meilleurs résultats sont présentées ici :

- **Mesure de cosinus** :

Cosinus est une mesure de similarité entre deux objets a et b de manière générale, très utilisée en recherche d'informations (Salton, 1989), qui consiste à représenter les deux objets par deux vecteurs $\vec{x}_a$ et $\vec{x}_b$ et de mesurer le cosinus de l'angle formé par les deux vecteurs (formule 1.10).

$$sim(a,b) = cos\,(\vec{x}_a\,,\vec{x}_b) = \frac{\vec{x}_a.\vec{x}_b}{\|\bar{x}_a\|.\|\bar{x}_b\|} \quad \text{.....} \quad (1.10)$$

Dans le cas du filtrage collaboratif, chaque utilisateur u est représenté par un vecteur x_u, où $x_{ui} = r_{u,i}$ Pour pouvoir calculer la similarité entre deux utilisateurs u et v, le cosinus est calculé sur l'ensemble des items notés par les deux utilisateurs (formule 1.11).

$$sim(u,v) = \cos(\vec{x}_u\,,\vec{x}_v) = \frac{\sum_{i\in I_{uv}} r_{u,i}\times r_{v,i}}{\sqrt{\sum_{i\in I_{uv}} r_{u,i}{}^2}.\sqrt{\sum_{i\in I_{uv}} r_{v,i}{}^2}} \text{.....} \quad (1.11)$$

Le cosinus peut aussi s'appliquer pour calculer la similarité entre deux items. En effet, il suffit de remplacer dans l'équation (la formule 2.8) les utilisateurs par leurs équivalents en items comme montré dans (la formule 1.12).

$$sim(i,j) = \cos\left(\vec{x}_i\,,\vec{x}_j\right) = \frac{\sum_{u\in U_{ij}} r_{u,i}\times r_{u,j}}{\sqrt{\sum_{u\in U_{ij}} r_{u,i}{}^2}.\sqrt{\sum_{u\in U_{ij}} r_{u,j}{}^2}} \text{.....} \quad (1.12)$$

Le cosinus varie entre 0 et 1. Une valeur égale à 1 indique que les deux utilisateurs ont des préférences identiques, une valeur égale à 0 indique qu'ils n'ont rien en commun. Un inconvénient majeur de l'utilisation du cosinus dans le filtrage collaboratif est qu'il ne tient pas compte de la variation dans le jugement des utilisateurs.

- **Coefficient de corrélation de Pearson :**

Ce coefficient a été utilisé notamment par les auteurs du système GroupLens (Resnick et al., 1994) pour calculer la similarité entre deux utilisateurs u et v. Le coefficient de corrélation de Pearson mesure le rapport entre la covariance et le produit de l'écart-type des notes données par les deux utilisateurs. Il permet ainsi de mesurer la similarité en utilisant les items notés à la fois par u et v. Plus les deux utilisateurs auront tendance à noter les mêmes items de façon équivalente, plus ils seront similaires comme illustré dans la formule (1.13).

$$sim(u,v) = \text{Pearson}(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u).(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2} * \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad \text{(1.13)}$$

Le coefficient de corrélation de Pearson peut également être utilisé pour mesurer la corrélation entre deux items i et j. L'équation (1.14) donne la similarité de Pearson entre deux items.

$$sim(i,j) = \text{Pearson}(i,j) = \frac{\sum_{u \in I_{ij}} (r_{u,i} - \bar{r}_i).(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in I_{ij}} (r_{u,i} - \bar{r}_i)^2} * \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \bar{r}_j)^2}} \quad \text{(1.14)}$$

1.4.2.3 EXEMPLE DE SYSTEME DE RECOMMANDATION BASE SUR LES FILTRAGES COLLABORATIF :

Le système BIPO (*Best Articles par Overlap*) (Bal et al., 2008) vise à améliorer les systèmes du FC par l'adaptation de la fonction de similarité utilisateur-à-utilisateur utilisée dans l'étape de sélection des voisins, en tenant compte de l'utilisateur actif. Le système se base sur l'hypothèse qu'un voisin peut être amélioré si la similarité utilisateur-à-utilisateur est basée sur une sélection d'un sous ensemble des articles co-évalués en commun: les articles fortement corrélés avec l'article cible. Une méthode de sélection adaptative des items locaux est utilisée pour la sélection du meilleur voisin, où le calcul change dynamiquement en fonction des profils des utilisateurs. Ensuite, la méthode sélectionne le sous-ensemble

d'articles ayant le plus grand poids de l'ensemble des articles co-évalués par les deux utilisateurs, dont la similarité est à déterminer.

1.4.2.4 AVANTAGES ET INCONVENIENTS DU FILTRAGE COLLABORATIF :

Les méthodes de filtrage collaboratif présentent plusieurs avantages dont les plus importants sont :

- **Effet de surprise (serendipity) :** l'effet de surprise que peut recevoir l'utilisateur en recevant une recommandation pertinente qu'il n'aurait pas trouvée seul est souvent souhaitable. Les algorithmes basés sur le filtrage collaboratif permettent généralement de faire des recommandations à effet de surprise. Par exemple, si un utilisateur u est proche d'un utilisateur v du fait qu'il ne regarde que des comédies, et si v apprécie un film d'un autre genre, ce film peut être recommandé à u du fait de sa proximité avec v.

- **Non nécessité de la connaissance du domaine** : les systèmes de recommandation basés sur le filtrage collaboratif ne requièrent aucune connaissance sur les items. Ces méthodes peuvent recommander des items sans avoir besoin de comprendre leurs sens ni disposer de leurs attributs. La recommandation est basée uniquement sur les notes données aux items.

Cependant, l'utilisation des techniques de filtrage collaboratif peut entrainer plusieurs problèmes :

- **Le démarrage à froid :** concerne à la fois les nouveaux utilisateurs et les nouveaux items qui sont introduits dans le système. Un nouvel utilisateur qui n'a noté aucun item ne peut pas recevoir de recommandation puisque le système ne connaît pas ses goûts. Ce problème est connu sous le nom de problème du démarrage à froid pour les utilisateurs (user cold start). Une solution à ce problème est de lui demander explicitement de noter un certain nombre d'items. D'autres solutions consistent à recommander au départ les items les plus populaires ou même des recommandations aléatoires. Ce problème du démarrage à froid se pose aussi lors de l'ajout d'un nouvel item.

Celui-ci ne peut pas être recommandé avant d'avoir été noté par un certain nombre d'utilisateurs.

- **La parcimonie (sparsity):** Le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous-ensemble des items disponibles. De ce fait, la matrice des notes est une matrice creuse avec un taux de valeurs manquantes pouvant atteindre 95% du total des valeurs (Papagelis et al., 2005) .Les systèmes de filtrage collaboratif ont des difficultés dans ce cas, le nombre de notes à prédire étant largement supérieur aux nombres de notes déjà connues. Le problème de la parcimonie peut être réduit en utilisant les approches par modèles qui réduisent la dimension de la matrice des notes.

- **Le problème du mouton gris (gray sheep) :** Les utilisateurs qui ont des goûts étranges (qui varient de la norme ou qui sortent du commun) n'auront pas beaucoup d'utilisateurs voisins. Il sera donc difficile de faire des recommandations pertinentes pour ce genre d'utilisateurs (Ghazanfar et al, 2014) .

1.4.3 RECOMMANDATION HYBRIDE :

Constatant les avantages et inconvénients de chacune des deux approches ci-dessus, on comprend que de nombreux systèmes reposent sur leur combinaison, ce qui en fait des systèmes de filtrage dits « hybrides ». En général, l'hybridation s'effectue en deux phases :

(i) Appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates,

(ii) 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.

Plus généralement, les systèmes hybrides gèrent des profils d'utilisateurs orientés contenu, et la comparaison entre ces profils donne lieu à la formation de communautés d'utilisateurs permettant le filtrage collaboratif. La meilleure description des méthodes hybrides a été faite par (Burk R., 2002) .Alors, selon Burke on peut distinguer sept façons de combiner les méthodes traditionnelles :

1) Pondération (Weighted): Une méthode hybride qui combine la sortie d'approches distinctes, utilisant, par exemple, une combinaison linéaire des scores de chaque technique de recommandation.

2) Commutation (Switching) *:* C'est une technique qui permet de faire le choix d'un modèle de recommandation parmi plusieurs, en se basant sur plusieurs critères. La détermination de la technique appropriée dépend de la situation. Le système se doit alors de définir les critères de commutation, ou les cas ou l'utilisation d'une autre technique est recommandée. Ceci permet au système de connaitre les points forts et les points faibles des techniques de recommandation qui le constituent.

3) Technique mixte (Mixed) : Dans cette approche, le recommandeur ne combine pas, mais augmente la description des ensembles de données, en prenant en considération les estimations des utilisateurs et la description des items. La nouvelle fonction de prédiction doit faire face aux deux types de descriptions et permet d'éviter les problèmes posés par le filtrage collaboratif, à savoir, le démarrage a froid.

4) Combinaison de caractéristiques (Features combination) : Dans un hybride basé sur la combinaison de caractéristiques, les données provenant des techniques collaboratives sont traitées comme une caractéristique, et une approche basée sur le contenu est utilise sur ces données.

5) Cascade : La cascade implique un processus étape par étape. Dans ce cas, une technique de recommandation est appliquée en premier, produisant un ensemble de candidats potentiels.
Puis, une deuxième technique raffine les résultats obtenus dans la première étape. Cette méthode a pour avantage que si la première technique génère peu de recommandations, ou si ces recommandations sont ordonnées afin de permettre une sélection rapide, la deuxième technique ne sera plus utilisée.

6) Augmentation de caractéristiques (Featur augmentation) : L'augmentation de caractéristiques est semblable à la cascade, mais dans ce cas-la les résultats obtenus (le classement ou la classification) de la première technique sont utilisées par le deuxième comme une caractéristique ajoutée.

7) Méta niveau (Meta-level) : Dans un hybride base sur méta niveau, une première technique est utilisée, mais différemment que la précédente méthode (augmentation de caractéristiques), non pas pour produire de nouvelles caractéristiques, mais pour produire un modèle. Et dans la deuxième étape, c'est le modèle entier qui servira d'entrée pour la deuxième technique.

1.5 CONCLUSION :

Dans ce chapitre, nous avons d'abord présenté la notion des systèmes de recommandation, en détaillant les trois approches les plus utilisées, à savoir le filtrage collaboratif et filtrage basé contenu ainsi que leur hybridation.

La mise en ouvre de ses système nécessite l'intégration d'un profil utilisateur alors nous abordons dans le chapitre suivant cette notion de profil utilisateur, à savoir sa représentation et sa construction dans les systèmes de recommandation.

CHAPITRE II

LA MODELISATION D'UN PROFIL UTILISATEUR

CHAPITRE II

2.1 INTRODUCTION :

Le profil utilisateur est un élément primordial dans les systèmes de recommandation. Il consiste en une structure qui permet de modéliser et stocker des informations relatives à l'utilisateur (préférences, intérêts …) afin de proposer un contenu pertinent en fonction de ses besoins et exigences spécifiques.

Dans ce chapitre, nous abordons le profil utilisateur. Nous présentons d'abord la notion de profil utilisateur. Nous présentons ensuite, les différentes phases et techniques de modélisation du profil utilisateur. La prise en compte de l'évolution du profil utilisateur sera également abordée dans la dernière section de ce chapitre.

2.2 LA MODELISATION DES UTILISATEURS :

Le modèle utilisateur a pour objectif de profiler l'utilisateur en modélisant ses goûts et ses préférences. Dans le cas des systèmes de recommandations. Plusieurs approches ont été définies pour représenter le profil de l'utilisateur, Ce profil peut contenir :

- **Données démographiques** : le profil utilisateur est constitué d'une liste de données démographiques telles que l'âge, le sexe, la profession, le niveau d'étude, le pays etc, décrivant le type de l'utilisateur. Les systèmes à filtrage démographique ne fournissent pas de recommandations individualisées, en effet les utilisateurs de même type auront les mêmes recommandations. Pour cette raison, les systèmes de recommandation démographiques ne sont pas considérés comme des systèmes de recommandation personnalisée mais des systèmes de recommandation de groupe (mastchoff, 2011).

- **Données sur le contenu des items** : le contenu sémantique des items est essentiellement exploité par les systèmes de recommandation basée sur le contenu. Plusieurs approches ont été utilisées pour représenter le profil utilisateur à partir du contenu de l'item (montaner et al, 2003) ; parmi lesquelles on peut citer le modèle vectoriel (VSM) dans lequel l'utilisateur est représenté par un vecteur de poids définit dans le même espace que celui représentant les items, chaque poids mesurant l'importance du terme correspondant pour l'utilisateur.

- **Données sur les usages :** la modélisation de l'utilisateur par les données issues de l'analyse des usages, et particulièrement les votes, est essentiellement utilisée par les systèmes de recommandation collaboratifs (lousan et al, 2009) .Les utilisateurs sont modélisés par la matrice des votes contenant l'historique de leurs votes sur la totalité des items à recommander.

2.2.1 REPRESENTATION DU PROFIL UTILISATEUR :

Comme dans les systèmes de RI, chaque type de système de recommandation demande un modèle de représentation différent du profil pour s'adapter à ses buts et son fonctionnement. Il existe donc plusieurs modèles tels que le modèle vectoriel, le modèle à base d'ontologie, le modèle multidimensionnel, etc. Dans ce qui suit nous décrivons les modèles les plus connus en recommandation.

2.2.1.1 REPRESENTATION ENSEMBLISTE :

L'approche ensembliste consiste à représenter le profil de l'utilisateur par des paquets de termes pondérés. On parle également de représentation vectorielle par analogie au modèle vectoriel de Salton **(salton, 1971)** sur lequel elle se base. Ces paquets de termes, traduisant les centres d'intérêts de l'utilisateur peuvent être regroupés différemment selon l'approche suivie pour exploiter le profil de l'utilisateur.

On distingue dans la littérature quatre grandes approches de représentation de profils utilisateurs basés sur la représentation ensembliste :

- Par liste de mots clés où chaque mot correspond à un centre d'intérêt spécifique (freitag et al, 1995).

- Par vecteur de termes pondérés pour chaque centre d'intérêts (Tebri et al, 2005) .

- Par ensemble de vecteurs de termes pondérés (ou non) indépendants, pour prendre en compte des centres d'intérêts multiples (Somlo et al, 2003) où chaque vecteur correspond à un domaine d'intérêt (pazzani et al, 1996).

➢ Par définition d'une relation d'ordre entre les centres d'intérêts du profil, on parle dans ce cas de préférences (kiebling, 2002) .

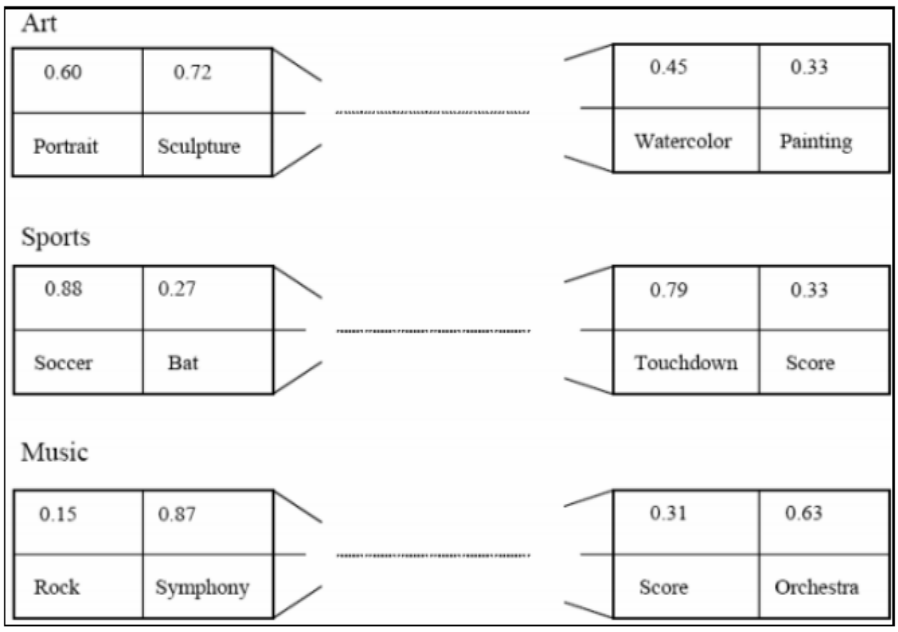

Figure 2.1 : Un exemple de profil représenté par des mots clés

La représentation ensembliste fait partie des premières représentations de profils utilisateurs qui ont été utilisées, plusieurs systèmes de recommandation et plus précisément en approche basé sur le contenu utilisent ce type de représentation. La pondération des termes est généralement basée sur le schéma tf.idf communément utilisé en recherche d'information (salton et al, 1973). Le poids associé à chaque terme permet de représenter son degré d'importance dans le profil de l'utilisateur. La **figure 2.1** donne un exemple de profil utilisateur représenté par des mots clés pondérés. Ce profil contient trois centres d'intérêts : Art, Sport, et Musique. Chaque centre d'intérêt est représenté par un ensemble de termes pondérés. Music = <(Rock, 0.15), (Symphony, 0.87)…> est un extrait de l'ensemble de termes pondérés représentant le centre d'intérêt Music.

2.2.1.2 REPRESENTATION SEMANTIQUE A BASE D'ONTOLOGIE :

Un autre modèle populaire de représentation de profils utilisateurs est le modèle sémantique à base d'ontologie. Dans ce modèle, un profil est une hiérarchie de concepts pondérés. Chaque nœud dans la hiérarchie est un concept. Le poids attaché avec un concept représente l'intérêt de l'utilisateur avec ce concept. Ce poids peut être changé pour mettre à jour l'intérêt de l'utilisateur. De plus, chaque concept est souvent représenté par un vecteur de termes pondérés. Le poids attaché avec un concept représente l'intérêt de l'utilisateur tandis que ce vecteur représente le contenu de ce concept. Ce vecteur peut être construit à partir d'un ensemble de documents assignés à ce concept.

Il existe plusieurs répertoires Web tels que celles d'ODP ou Yahoo qui peuvent être utilisés comme hiérarchie de concepts. Dans ce cas, un vecteur de termes pondérés qui représente un concept peut être construits à partir des documents (pages Web) indexés sous ce concept (Susan et al, 2003) (ou l'ensemble des documents indexés sous ce concept plus les documents indexés sous ses sous-concepts (ahu sieg et al, 2007). Un exemple de profil utilisateur représenté par le modèle à base d'ontologie avec le processus de mise à jour est illustré dans la figure 2.2 (Micro speretta et al, 2004) ; (Vistu kanth, 2004) ; (stuarte et al, 2001) ; (ahu sieg et al, 2007).

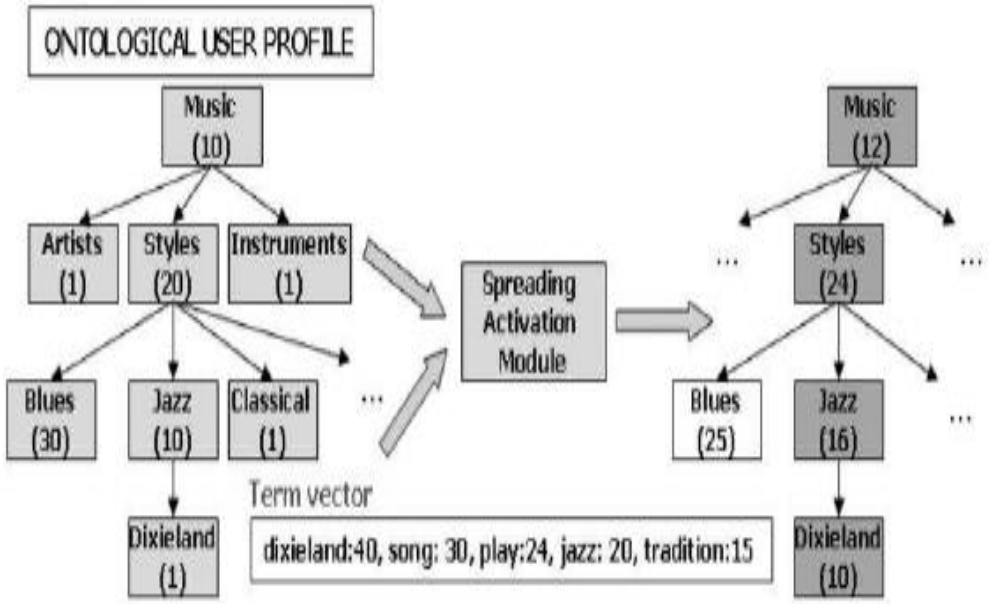

Figure 2.2– Exemple du profil utilisateur représenté par le modèle d'ontologie avec le processus de mise à jour des poids des concepts. (ahu sieg et al, 2007)

L'utilisation de ce type de modèle pour représenter les profils utilisateurs peut aider à mieux connaître les intérêts des utilisateurs par rapport au modèle vectoriel. Par exemple, si on représente un profil utilisateur par un vecteur de termes pondérés et dans ce vecteur contient un terme instrument, on ne sait pas exactement si ce terme concerne des instruments musiques ou les autres types d'instruments. Cependant, si on utilise le modèle ontologie (comme illustré dans la figure 2.2), on peut facilement régler ces ambiguïtés.
Dans cet exemple, le concept " instruments " est un concept " fils " du concept " music ", alors il s'agit des instruments de musiques. De plus, un autre avantage du modèle d'ontologie est qu'on peut propager la valeur d'intérêt d'un concept vers les autres concepts reliés (par exemple, son concept "père") afin de trouver des nouveaux centres d'intérêt.

Cette représentation est très souvent utilisé en recommandation nous citons comme exemple le système Foxtrotet Quickstep (stuarte et al, 2001) ; (Struarte et al, 2003) ,qui utilise un modèle à base d'ontologie des articles scientifiques pour représenter les centres d'intérêts des utilisateurs.

2.2.1.3 REPRESENTATION MULTIDIMENSIONNEL :

Le travail de (Amato et al, 1999) est un des premiers travaux vers la construction d'un modèle multidimensionnel pour représenter des profils utilisateurs. Cette représentation donne une description globale des utilisateurs en prenant en compte plusieurs dimensions différentes. Dans leur article, les informations concernant les utilisateurs peuvent être classifiées dans cinq catégories différentes, chaque catégorie est une dimension:

- La catégorie des données personnelles contient des données d'identification personnelles de l'utilisateur (nom, date de naissance, contact...)

- La catégorie de recherche contient des préférences et des restrictions sur les documents que l'utilisateur est en train de rechercher.

- La catégorie de livraison, c'est des spécifications concernant le mode de livraison des informations trouvées (courriel, fax, Web, temps de livraison etc.)

- La catégorie de données des actions contient des enregistrements sur l'interaction de l'utilisateur avec le système de recherche et les données de navigation (pages Web visités, documents lus, jugements de pertinence etc.)

- Enfin la catégorie de données de sécurité est une collection des préférences de l'utilisateur concernant des conditions d'accès aux informations du profil utilisateur.

Dans l'article (Nesrine zemerli et al, 2005), les auteurs proposent un autre modèle multidimensionnel pour représenter des profils utilisateurs. Dans ce modèle, le contenu d'un profil se compose de trois dimensions (ou catégories) principales:

- La catégorie des préférences concernant des préférences de l'utilisateur (domaine d'intérêt, préférences de recherche d'information)
- La catégorie des données personnelles permettant d'identifier l'utilisateur (son identité et sa profession)
- La catégorie des données d'environnement contenant des informations sur l'environnement de recherche de l'utilisateur (l'emplacement géographique, la configuration

logicielle et matérielle). Chaque dimension peut se décomposer en sous-dimensions qui sont plus détaillées.

Les auteurs proposent aussi la possibilité d'intégrer ce profil dans la phase de reformulation de la requête, dans la phase de réduction de l'espace de recherche pour restreindre l'espace de recherche aux documents qui correspondent le mieux aux besoins de l'utilisateur, dans la phase d'appariement document-requête, ou dans la phase de présentation des résultat.

2.2.1.4 REPRESENTATION PAR MATRICE UTILISATEURS ITEMS :

La représentation par matrice utilisateurs-items est souvent utilisé dans les systèmes de recommandation collaborative (voir **figure 2.3**). Chaque ligne de la matrice représente un utilisateur et chaque colonne représente un item. Une cellule [i,j] de la matrice contient le vote de l'utilisateur i pour l'item j sur une échelle quelconque dans cette exemple l'échelle est de [1,10] (ou rien si utilisateur n'a pas voté cet item) . Dans ce modèle, le profil d'un utilisateur est considéré comme un vecteur des votes de cet utilisateur pour les items.

	Item1	Item2	Item3	Item4
Utilisateur1	-	6	9	2
Utilisateur2	2	-	6	8
Utilisateur3	8	2	-	-

Figure 2.3 *:* Matrice utilisateurs items.

2.2.2 CONSTRUCTION DU PROFIL UTILISATEUR :

Indépendamment de son modèle de représentation, la construction du profil utilisateur repose sur deux phases principales : la phase de collecte des sources d'informations et la phase d'exploitation de ces sources d'informations pour construire et représenter le profil utilisateur avant son utilisation en recommandation **(Figure 2.4)**.

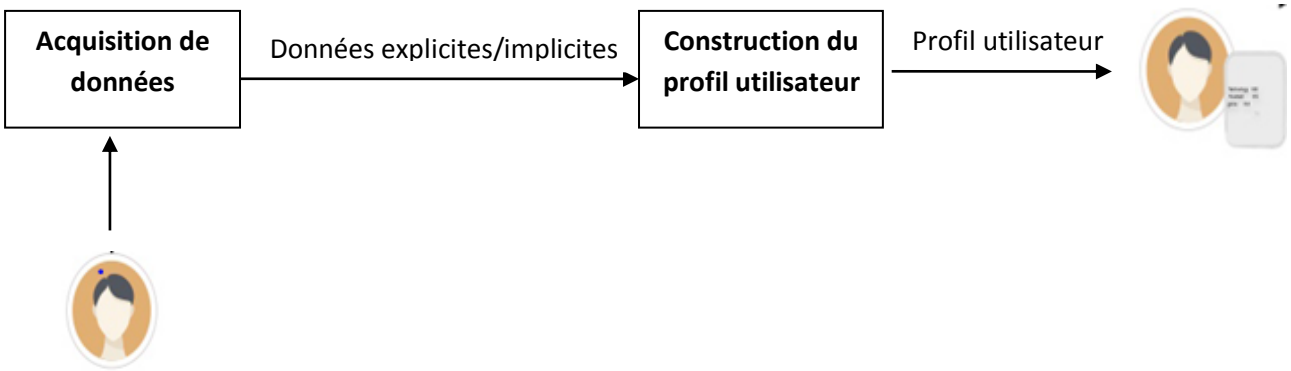

Figure 2.4 : **Les phases de construction du profil utilisateur.**

Nous présentons, dans un premier temps, les techniques d'acquisition des sources d'informations et le prétraitement des données. Ensuite, nous présentons les techniques de construction de profils utilisateurs.

2.2.2.1 ACQUISITION D'INFORMATION :

Dans cette partie, nous abordons les méthodes d'acquisition des informations nécessaires pour construire les profils utilisateurs. Il existe trois méthodes principales pour acquérir ces informations : la méthode d'acquisition implicite, la méthode d'acquisition explicite et la méthode hybride.

- ✓ **Acquisition explicite :**

Dans cette méthode, les utilisateurs doivent fournir explicitement les informations nécessaires au système. Une telle approche a été utilisée depuis longtemps dans plusieurs systèmes de recherche d'information (par exemple la méthode de retour de pertinence de Rocchio (Rocchio, 1971) . Il existe plusieurs formes d'acquisition explicite :

- ➢ **Entrée directe par les utilisateurs :** Les utilisateurs doivent entrer leurs intérêts sous la forme de mots clés. Malgré sa simplicité apparente, c'est une approche qui demande beaucoup d'effort du côté des utilisateurs pour clarifier leurs intérêts.

- **Classement binaire par les utilisateurs :** Dans les systèmes (Liren chen et al, 1998) ; (young woo seo et al, 2000) les utilisateurs doivent classer les items (pages Web, livres ...) dans deux classes « intéressants » ou « inintéressants »**.**

- **Le vote:** Dans les systèmes (Pazzani et al, M Pazzani, J Muramatsu, and D Billsus Syskill & Webert : Identifyinginteresting web sites., 1996) ; (marko balabanovic, 1997) ; (daniel billsus et al, 1999) ; (Paul Resnick et al, 1994) ; (Upendra shardanand et al, 1995),les utilisateurs doivent donner des votes explicites pour mesurer leurs intérêts avec les items qu'ils doivent évaluer. Par exemple, dans le système Syskill & Webert (Pazzani et al,1996), les pages Web sont évaluées selon trois niveaux hot, lukewarm, cold. Dans le système FAB (marko balabanovic, 1997), des notes de -3 à +3 sont données aux pages Web. Dans le système Ringo (upendra shardanand et al, 1995), les artistes et les albums de musique sont évalués par des notes de 1 à 7.

- **Stéréotype :** Quelques systèmes (Bruce krulwich et al, 1997) demandent aux utilisateurs de répondre à un ensemble de questions prédéfinies ensuite ils utilisent ces informations pour déduire leurs profils.

L'avantage des approches explicites est que les profils ainsi construits sont plus précis. Cependant, il existe également plusieurs inconvénients à cette approche (Miquel montaner et al, 2003):

- Les jugements de pertinence de l'utilisateur pour les items sont dépendants des changements de son besoin d'information. Par exemple, un utilisateur qui a déjà lu deux documents pertinents à son besoin d'information attribuera un vote plus faible à un troisième qui traite du même sujet parce que son niveau d'exigence a augmenté.

- Les échelles numériques peuvent être inadéquates pour décrire les réactions des hommes pour les items.

- Les utilisateurs sont souvent hésitants à donner des évaluations si elles ne sont pas directement reliées à leurs besoins immédiats.

À cause de ces inconvénients, plusieurs systèmes ont choisi d'acquérir implicitement des informations nécessaires pour le profilage. Ces techniques d'acquisition implicite sont présentées dans la partie ci-après.

- ✓ **Acquisition implicite :**

Les limitations dans l'acquisition des données explicites, ont orienté les travaux vers des techniques d'acquisition des données implicites de l'utilisateur. Il s'agit dans ce cas, de ne plus demander à l'utilisateur de fournir explicitement ses données, mais de trouver des sources et des données tierces, permettant d'extraire des connaissances sur l'utilisateur et de construire son profil.

Deux questions principales se posent alors et sont explorées dans les sous-sections suivantes : quelles données peuvent être exploitées puisqu'elles ne sont pas explicitement demandées à l'utilisateur ? Et qui produit ces données ?

a. Données disponibles :

Les données utilisées dans cette approche, sont les données collectées en observant le comportement et/ou en extrayant les informations des utilisateurs au travers de leurs activités. D'après (Kelly et al, 2011) , ces données sont aussi variées que : les documents propres à l'utilisateur, les requêtes et documents sélectionnés lors de l'utilisation d'un moteur de recherche, les pages Web consultées, les contenus publiés sur le web ou sur les réseaux sociaux (annotations, commentaires), les fichiers logs sur les applications telles que les applications de messagerie, les fichiers logs sur les consultations de bases de données, etc. Ces données sont généralement utilisées pour extraire les intérêts de l'utilisateur, dans la mesure où elles ont « un lien direct » avec lui puisqu'il les a consultées ou produites.

b. Producteurs de données :

Dans la littérature, les données implicites peuvent être produites soit par l'utilisateur lui-même, soit par d'autres utilisateurs. Dans ce dernier cas, ce sont des individus considérés comme « proches » de l'utilisateur : les individus similaires (Ekstrand et al, 2011) ; ou les individus dans son réseau social (Carmel et al, 2009) ; (Guy et al, 2009), (wen et al, 2011), (al Z. e., 2010). Nous analysons ces trois cas dans les paragraphes qui suivent.

- **Données produites par l'utilisateur :** L'acquisition des données à partir des données produites par l'utilisateur est le cas le plus courant dans la littérature. Toutefois, ce type d'acquisition peut poser des problèmes lorsque le système dispose de très peu de données générées par l'utilisateur. C'est souvent le cas des nouveaux utilisateurs ou des utilisateurs peu actifs comme évoqué dans (al Z. e., 2010).

- **Données produites par les individus similaires à l'utilisateur :** Dans ce cas, les données à acquérir sont extraites des données produites par les utilisateurs similaires à l'utilisateur courant. La similarité entre les utilisateurs est généralement calculée par croisement (cosinus de similarité par exemple) du profil de l'utilisateur avec ceux de tous les autres utilisateurs du système (Gao et al, 2010). Ensuite, les informations des individus similaires sont utilisées, pour calculer les informations ou les intérêts de l'utilisateur. Ce principe est à la base des différentes techniques de filtrage collaboratif. Cependant, cette technique nécessite beaucoup de temps de calcul pour des systèmes qui possèdent un nombre très élevé d'utilisateurs (temps de calcul de similarités entre les utilisateurs). De plus, elle ne peut pas être exploitée efficacement pour les utilisateurs ayant un profil vide ou très pauvre (nouvel utilisateur dans le système par exemple), car il devient impossible de retrouver des individus similaires à ce dernier.

- **Données produites par les individus dans le réseau social de l'utilisateur :** Dans ce cas, les données à acquérir sont extraites des données produites par les utilisateurs en lien avec l'utilisateur concerné dans un réseau social. Un réseau social est un graphe de relations entre individus. Les liens entre les individus dans le réseau social, représentent les relations entre eux et donc une certaine similarité. Cette approche se base sur les théories qui montrent qu'un utilisateur crée des relations avec ceux qui lui sont similaires (homophilie) car il partage avec eux des intérêts communs (Aral et al, 2013), (Carmel et al, 2009).Cette approche d'acquisition de données, restreint par construction le nombre d'individus similaires à l'utilisateur, en évitant d'explorer tous les utilisateurs du système. Ceci peut réduire drastiquement le temps de calcul de similarités entre l'utilisateur et les individus. De plus, cette approche peut être utilisée comme alternative, pour pallier les limites des méthodes précédentes, en cas de manque d'informations sur l'utilisateur (profil vide ou pauvre).

Le principal avantage de l'acquisition de données implicites est qu'elle ne nécessite aucune action explicite de la part de l'utilisateur. Cependant avec cette technique, on peut faire face au problème d'informations biaisées ou de manque d'informations. En effet, avec les données acquises sans vérification de la part de l'utilisateur, il se peut que ces dernières ne soient pas pertinentes pour lui (ex. les données que l'utilisateur produit par erreur ou qui contiennent des informations obsolètes). On peut également manquer d'informations importantes pour extraire des connaissances sur l'utilisateur par exemple, lorsque l'utilisateur n'interagit pas souvent avec le système, les données ne seront pas suffisantes pour extraire les préférences ou intérêts de cet utilisateur. Avec cette technique d'acquisition de données, il est donc nécessaire d'appliquer un prétraitement et un traitement de données qui s'avèrent plus complexes que la technique d'acquisition de données explicites. La problématique du prétraitement des données est abordée dans la section suivante.

- ✓ **Acquisition hybride :**

Quelques systèmes ont choisi de combiner les deux précédentes méthodes pour obtenir une meilleure performance. Dans (young woo seo et al, 2000), un profil utilisateur contient des termes pondérés, chaque fois qu'un document est jugé pertinent, le poids d'un terme dans son profil est mis à jour en utilisant les paramètres suivants : le vote explicite, le temps utilisé pour lire ce document, le nombre de liens suivis et l'action sauvegarder dans les signets de ce document.

Dans (Miquel montaner et al, 2003), les auteurs font une liste de 37 systèmes qui utilisent différentes approches d'acquisitions d'information d'utilisateurs. Parmi eux, 20 systèmes utilisent des approches explicites, 8 systèmes utilisent des approches implicites, les approches hybrides sont utilisées par 9 autres systèmes

2.2.2.2 PRETRAITEMENT DES DONNEES :

Les données issues de la sélection des données, comme expliqué précédemment, peuvent contenir de nombreuses inconsistances telles que : des données incomplètes (manque de valeurs ou attributs importants), des données biaisées (présence d'erreurs produites lors des saisies ou de la collection automatique de données), des incohérences (nommages ou codages différents dans les données, …). De plus, ces données brutes peuvent ne pas être conformes au modèle ou au format d'entrée de l'algorithme de construction du profil utilisateur. Après

l'étape de sélection des données, il est nécessaire de mettre en œuvre une étape de prétraitement avant l'étape finale de construction du profil utilisateur.

On utilise plusieurs types de prétraitement selon les données (Gracia et al, 2015); (Liu, 2007) .

- Pour les données incomplètes, biaisées ou incohérentes, on peut appliquer des techniques de nettoyage de données, qui consistent à ignorer les données manquantes ou à utiliser la valeur moyenne d'un attribut en remplacement ou encore à utiliser la valeur la plus probable (formule bayésienne ou arbre de décision) en remplacement, etc.

- La discrétisation des données peut être appliquée pour convertir des attributs continus vers des attributs ordinaux.

- La réduction des données peut être appliquée pour obtenir une représentation réduite du jeu de données, plus petite en volume, mais qui produit (ou presque) les mêmes résultats analytiques.

- Pour rendre les données conformes au modèle ou à l'algorithme utilisé, on peut appliquer des techniques de transformation de données qui permettent par exemple, de ne conserver qu'un résumé d'un texte à partir d'un texte entier, de traduire un texte dans une autre langue …

Après la collecte et le prétraitement des données, celles-ci sont utilisées en entrée de la phase de construction du profil utilisateur présentée dans la section suivante.

2.2.2.3 LES TECHNIQUES DE CONSTRUCTION DU PROFIL UTILISATEUR:

Dans cette partie, nous allons aborder les techniques de construction de profils utilisateurs à partir des informations collectées.

❖ **Technique tf_idf (terme frequency _inverse document frequency) :**

La méthode la plus utilisée pour construire des profils utilisateurs est la technique tf-idf et ses variantes. C'est une technique issue du domaine de la recherche d'information pour la pondération de termes dans le modèle vectoriel tel que nous l'avons vu précédemment dans

l'approche basé sur le contenu vu que le modèle vectoriel est le modèle le plus utilisé pour représenter des profils utilisateurs.

Quelques systèmes utilisant cette approche sont (Liren Chen et al, 1998) , (Daniel Billsus et al, 1999). Dans (Liren Chen et al, 1998) , un profil se compose de N vecteurs de termes pondérés. Chaque vecteur représente un domaine d'intérêt de l'utilisateur. Chaque fois qu'un document est jugé pertinent par l'utilisateur, le système construit le vecteur tf-idf de ce document, après cette étape on obtient un ensemble de N + 1 vecteurs pondérés (N vecteurs profils et 1 nouveau vecteur de document). Puis le système calcule la similarité cosinus entre chaque paire de vecteurs dans cet ensemble et combine les deux vecteurs les plus similaires. Dans cette approche, le profil est mis à jour incrémentalement chaque jour.

Dans le système NewsDude (Daniel Billsus et al, 1999), le profil à court-terme d'utilisateur se compose de plusieurs documents pour lesquels il a voté, chaque document est représenté par son vecteur tf-idf. Chaque fois qu'un nouveau document arrive, le système va d'abord extraire le vecteur tf-idf de ce document. Puis il compare la similarité cosinus de ce document avec les autres documents dans le profil : les documents ayant une similarité avec le nouveau document plus élevée qu'un seuil prédéfini seront filtrés (ou sélectionnés). La prévision de vote du nouveau document sera la valeur moyenne de tous les votes que l'utilisateur a effectués pour les documents filtrés. Le système utilise cette prévision de vote pour décider de recommander le nouveau document à l'utilisateur ou non.

- **Les méthodes de classification :**

Les méthodes de classification sont les méthodes d'apprentissage supervisé qui sont en charge d'affecter les éléments dans les groupes existants. Ces groupes contiennent déjà des exemples positifs qui sont nécessaires pour les algorithmes d'apprentissage supervisé. Par exemple, dans les travaux de (Gauch et al, 2003), le profil utilisateur est représenté par le modèle ontologie. Le poids d'un concept représente l'intérêt de l'utilisateur avec ce concept. Pour calculer ces poids, ils utilisent les pages Web que l'utilisateur a lues dans le passé. Ces pages Web sont enregistrées dans le répertoire cache du navigateur. Pour chaque page Web (d_k), ils calculent la similarité de cette page avec tous les concepts dans l'ontologie. La similarité est calculée en utilisant la mesure cosinus et en prenant en compte également le temps que l'utilisateur a utilisé pour lire cette page Web ainsi que la longueur de la page. La

page Web est classifiée dans cinq concepts (C_j) qui sont les plus similaires avec cette page Web. Les poids de ces concepts seront augmentés par les valeurs retournées par le classificateur. En bref, cet ajustement est calculé de la manière ci-dessous :

$$\textbf{Similarité } (\boldsymbol{C_j}, \boldsymbol{d_k}) = facteur\_temps\_longeur \times similarité\_cosinus(d_k, C_j) \quad \ldots (2.1)$$

Dans cette formule, le *facteur_temps_longeur* est calculé par une des quatre formules suivantes :

$$\frac{\boldsymbol{temps}}{\boldsymbol{longueur}} \quad \ldots\ldots (2.2)$$

$$\log \frac{\boldsymbol{temps}}{\boldsymbol{longueur}} \quad \ldots\ldots (2.3)$$

$$\log \frac{\boldsymbol{temps}}{\boldsymbol{log(longueur)}} \quad \ldots\ldots (2.4)$$

$$\log \frac{\boldsymbol{temps}}{\boldsymbol{log(log(longueur))}} \quad \ldots\ldots (2.5)$$

Où $\boldsymbol{temps}$ est le temps (en seconde) que l'utilisateur a utilisé pour lire le document et $\boldsymbol{longueur}$ est la longueur (en octet) du document.

Parmi les autres travaux qui utilisent les méthodes de classification pour construire et mettre à jour les profils utilisateurs nous pouvons citer à titre d'exemple (Mirco Speretta et al, 2004).

❖ **Les méthodes de clustering :**

Les méthodes de clustering (ou regroupement) sont les méthodes d'apprentissage non supervisé qui sont en charge d'attribuer les éléments dans des groupes qui n'existent pas à l'avance. Par exemple, dans (Hyoung R et al, 2003) les auteurs utilisent une hiérarchie d'intérêts de l'utilisateur pour représenter le profil utilisateur. Dans cette hiérarchie, les nœuds feuilles représentent les intérêts spécifiques et les nœuds internes représentent les utilisations de profils utilisateurs dans les systèmes personnalisés intérêts plus généraux.

Pour construire cette hiérarchie, ils utilisent un algorithme de regroupement hiérarchique. L'entrée de cet algorithme est un ensemble de pages Web que l'utilisateur a visitées. Ils enlèvent les mots vides et puis font la lemmatisation sur les mots de ces pages

Web. Puis ils calculent les similarités entre toutes les paires de mots en utilisant différentes fonctions de similarité (AEMI, AEMI-SP, Jaccard etc.). Ensuite, l'algorithme va grouper récursivement ces mots dans les sous-groupes, chaque sous-groupe représente un nœud dans la hiérarchie d'intérêts.

Parmi les autres travaux qui utilisent les méthodes de clustering pour construire des profils utilisateurs, nous pouvons citer à titre d'exemple (Gabriel et al, 2001) , (Sieg, 2004).

2.3 ADAPTATION ET MISE A JOUR DU PROFIL UTILISATEUR :

Les préférences de l'utilisateur évoluent et changent avec le temps. Par exemple, après une naissance, une mère peut être intéressée par des produits dédiés aux nourrissons, mais son intérêt va progressivement changer vers d'autres produits au cours du temps. Ce qui signifie, que les goûts ne sont pas statiques et que les observations (représentées sous forme de retour d'expérience) c'est à dire les évaluations des utilisateurs sur les recommandations effectuées par le système, les plus récentes, représentant les intérêts actuels de l'utilisateur, sont plus pertinentes que les anciennes observations. Par conséquent, il s'avère nécessaire, de disposer de techniques permettant la mise à jour des intérêts de l'utilisateur en conservant les préférences les plus récentes et en éliminant les plus anciennes. Plusieurs approches ont été définies parmi les quelles on trouve :

- **Adaptation manuelle** : l'utilisateur est invité à mettre à jour son propre profil si ses centres d'intérêts ont changé. En plus du fait qu'elle nécessite une implication directe de la part de l'utilisateur, cette solution est inadaptée lorsque les goûts de l'utilisateur changent fréquemment.

- **Adaptation par ajout d'information** : c'est la méthode la plus utilisée dans les systèmes de recommandation (montaner et al, 2003). La mise à jour du profil utilisateur est assurée par les retours d'expérience explicites, ou implicites recueillis par le système par analyse des usages qui consiste à analyser les données disponibles dans les fichiers log afin de modéliser les préférences de l'utilisateur. L'analyse des usages doit, par conséquent, aboutir à inférer les évaluations apportées par l'utilisateur sur les items qu'il a consultés. Une

évaluation indique l'utilité (l'intérêt ou le désintérêt) que porte l'utilisateur pour un item. Elle peut être soit sous forme d'une note (également appelée vote) qu'il attribue explicitement ou implicitement, soit sous forme d'annotations (tag) qu'il associe à l'item correspondant. Les évaluations des utilisateurs sont également considérées comme un retour d'expérience sur les recommandations effectuées par le système en mesurant le taux de recommandations pertinentes. Le retour d'expérience peut se présenter soit sous forme de vote explicite ou implicite ou sous forme d'annotations. Les systèmes tels que Amazon (linden et al, 2003) ,MovieLens (miller et al, 2003) et Tapestry (Goldberg et al, 1992) utilisent les retours d'expérience comme moyen d'adaptation de leurs systèmes de recommandation. Le problème posé par cette méthode est qu'elle ne permet pas d'éliminer les anciennes préférences.

2.4 CONCLUSION :

Dans cette partie, nous avons étudié la notion de profil utilisateur, les modèles de représentation de profils utilisateurs, les méthodes d'acquisition des informations des utilisateurs, les techniques de construction et de mise à jours des profils utilisateurs.

Parmi les modèles de représentation de profils utilisateurs, le modèle à base d'ontologie qui permet de mieux connaître les intérêts des utilisateurs Cependant, ce modèle est plus difficile à mettre en œuvre car il demande souvent beaucoup d'informations de l'utilisateur (par exemple, les documents qui représentent ses centres d'intérêt) pour construire l'ontologie qui représente son profil. Le modèle multidimensionnel semble être un bon modèle pour représenter les profils utilisateurs. Il peut contenir également les autres modèles comme le modèle vectoriel qui est facile à implémenter et qui permet de représenter les différents centres d'intérêts de l'utilisateur en utilisant plusieurs vecteurs pondérés Cependant, chaque système différent a besoin de différentes dimensions pour représenter ses utilisateurs.

La question principale est donc : comment définir un profil utilisateur qui soit pertinent pour un système de recommandation?

CHAPITRE III

MODÉLISATION DU PROFIL UTILISATEUR EN RECOMMANDATION

CHAPITRE III

3.1 INTRODUCTION :

L'utilisateur est l'acteur principal d'un système de recommandation pour cela une modélisation de son profil est nécessaire.

Notre travail consiste à représenter et à construire un profil utilisateur pour un système de recommandation. Pour ce faire nous avons exploité le jeu de données Book-Crossing qui a été collecté par le groupe de recherche GroupLens de l'université du Minnesota portant sur les évaluations faites par des utilisateurs sur un ensemble de livres pour avoir les centres d'intérêts des utilisateurs qui nous permettront de construire leurs profils.

Dans ce chapitre nous allons présenter l'approche proposée qui se base sur le modèle vectoriel.

3.1 PROBLEMATIQUE ET MOTIVATIONS :

Les approches proposées en recommandation sont confrontées à la question de la définition des informations nécessaires concernant l'utilisateur et la modélisation de son profil ainsi que son exploitation dans le processus de recommandation .Nous constatons que l'une des difficultés dont souffrent les systèmes de recommandations est le démarrage à froid lors de la première interaction de l'utilisateur avec le système, car son profil est inexistant par conséquent le système ne peut pas lui fournir des recommandations pertinentes, la réponse à ce problème est que l'utilisateur est mis à contribution , pour cela le système lui proposera des items (articles) les plus populaires afin de les évaluer pour avoir son profil.

Notre travail s'intitule autour des systèmes de recommandation basé contenu, vu que c'est le plus adéquat à notre domaine applicatif dans le cadre de la recommandation des articles. Nous montrons comment définir un profil utilisateur à partir des premières évaluations en utilisant le modèle vectoriel.

3.2 DESCRIPTION DE L'APPROCHE PROPOSEE :

3.2.1 ARCHITECTURE DU SYSTEME DE RECOMMANDATION :

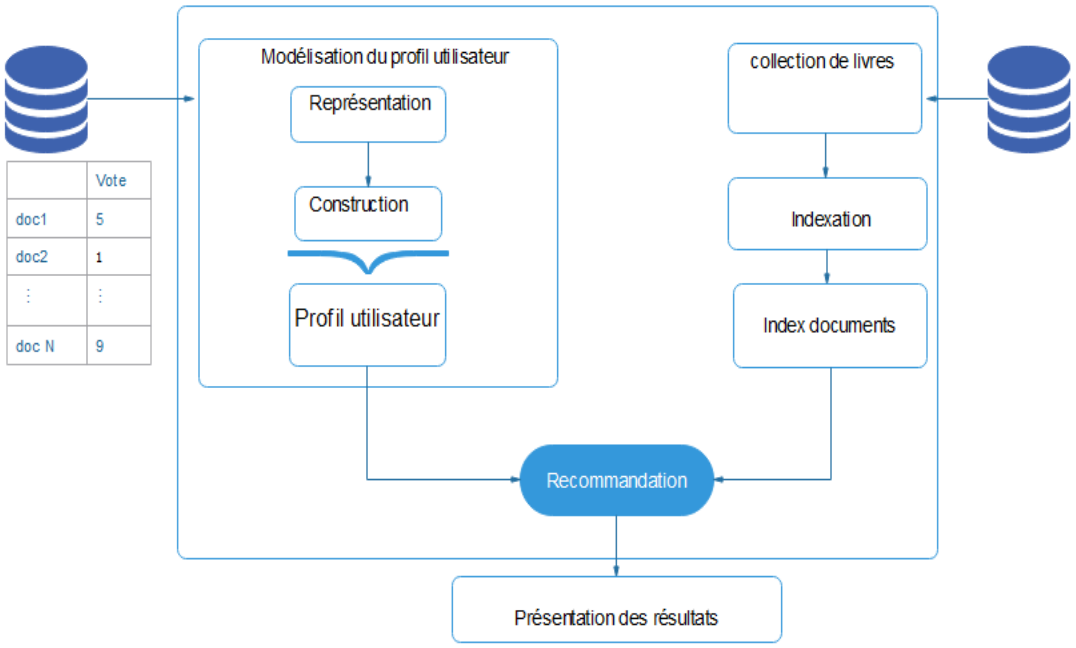

Figure 3. 1 : Architecture du système de recommandation

Les principaux composants dans notre approche de recommandation sont illustrés dans (la figure 3.1), Ils intègrent l'indexation des articles ainsi que la modélisation du profil utilisateur et leurs exploitation dans la phase de calcul du score de pertinence de ces derniers.

Nous allons commencer par la modélisation du profil utilisateurs qui consiste en sa représentation ainsi que sa construction.

3.2.2 REPRESENTATION DU PROFIL UTILISATEUR :

Pour la représentation du profil utilisateur nous allons utiliser une représentation basée sur le modèle vectoriel pour la simplicité de sa mise en œuvre.

3.2.3 CONSTRUCTION DU PROFIL UTILISATEUR :

Une fois le profil représenté, sa construction consiste à l'acquisition et la sélection des données nécessaire à sa représentation. Nous appliquerons la technique tf_idf qui est la plus utilisé dans le modèle basé contenu, cette approche tire parti de la description des attributs des livres avec lesquels l'utilisateur a interagi pour recommander des livres similaires.

Il est simple d'utiliser la catégorie du livre pour créer des profils d'articles et des profils utilisateurs, pour ceci nous utiliserons la représentation binaire 1/0 qui est efficace et donne de bons résultats.

3.2.4 EXPLOITATION DU PROFIL UTILISATEUR DANS LE PROCESSUS DE RECOMMANDATION :

L'objectif d'un système de recommandation est d'attribuer des notes aux documents ou articles qui seront appropriés aux goûts des utilisateurs. Ainsi ces données sont utilisées par le système afin de créer une représentation du profil utilisateur.

Dès lors, notre système consiste à inclure le profil d'un utilisateur définit par son centre d'intérêt, comme composante dans le module de prédiction du processus de recommandation.

3.2.4.1 INFERENCE D'UN PROFIL UTILISATEUR :

Un profil utilisateur est représenté sous forme d'un vecteur de termes pondérés chaque terme correspond à une catégorie d'article.

Ce dernier est inféré en se basant sur les notes attribuées par l'utilisateur, il est construit à partir de l'ensemble des articles les plus populaires, proposés lors de son interaction avec le système.

Un profil utilisateur u est alors représenté par un vecteur de catégories pondérées qui dénote l'importance de la catégorie ca_i par rapport au profil de l'utilisateur u :

$$u = (wc_{1k}, wc_{2k}, \dots\ wc_{ik}) \quad \dots\dots \quad (3.1)$$

$w_{i,k}$: Le poids de la catégorie ca_i par rapport au profil de l'utilisateur u . Il est calculé comme suit :

$$wc_{i,k} = \sum_{a_j \in A} w_{i,j} * n_j \quad \dots\dots \quad (3.2)$$

Où :

n_j : représente la note attribuée par l'utilisateur pour l'article a_j

$W_{i,j}$: représente le poids de la catégorie ca_i dans l'article a_j calculé comme suit :

$$W_{i,k} = \frac{Occ_{i,j}}{Nc_j^{\,2}} \quad \text{.....} \quad (3.3)$$

$Occ_{i,j}$: Occurrence de la catégorie ca_i dans l'article a_j

Nc_j : Le nombre de catégorie dans l'article a_j

3.2.4.2 CALCUL DU POIDS PONDERES DES CATEGORIES POUR CHAQUE ARTICLE :

Chaque article A_j est représenté sous forme d'un vecteur de catégories ca_j

$$A_j = (wc_{1J}, wc_{2J}, \dots wc_{ij}) \text{.....} \quad (3.4)$$

Le poids d'une catégorie dans l'article a_j est dénoté par la formule suivante :

$$w_{ij} = tf_{i,j} * \log \frac{N}{n_i} \text{.....} \quad (3.5)$$

Avec N est le nombre total des articles et n_i le nombre d'articles qui contiennent la catégorie ca_i.

3.2.4.3 INTEGRATION DU PROFIL UTILISATEUR POUR LA PREDICTION:

Après avoir définit le profil utilisateur par son centre d'intérêt, il sera inclut dans le module de recommandation.

Pour la prédiction des scores de pertinence on effectue le produit scalaire entre le vecteur profil utilisateur et les vecteurs pondérés des articles, ceci est dénoté par la formule suivante:

$$\text{Produit scalaire} = \sum_{a_j \in D} W_{i,j} * W_{i,k} \text{.....} \quad (3.6)$$

À la fin on aura le degré de pertinence d'un article pour l'utilisateur.

3.3 ILLUSTRATION DE L'APPROCHE PROPOSEE :

Nous allons dérouler un exemple de l'approche proposée, à partir des 6 meilleurs articles les plus populaires $A = \{A_1, A_2, A_3, A_4, A_5, A_6\}$.

L'index de la collection inclut 5 catégories, les articles sont représentés sous forme binaire, les articles qui sont de catégorie i sont donnés par 1 sinon ils sont de 0 (**voir la figure 3.1**).

Articles	Big Data	R	Python	Machine Learning	Learning Path
Article1	1	0	1	0	1
Article2	0	1	1	1	0
Article3	0	0	0	1	1
Article4	0	0	1	1	0
Article5	0	1	0	0	0
Article6	1	0	0	1	0

Tableau3.1: représentation binaire des livres

Le tableau ci-dessus (tableau 3.1) représente, l'article 1 concerne le Big Data, le python et machine Learning. De même, l'article 2 traite de R, de Python et de machine learning.

	Article1	Article2	Article3	Article4	Article5	Article6
User1	1	-1				1
User2	-1	1		1		

Tableau 3.2 : notes des utilisateurs sur les articles

Le tableau 3.2 spécifie si un utilisateur a aimé ou non un article recommandé, l'utilisateur a aimé l'article 1 pour cela la valeur 1 est attribuée et n'a pas aimé l'article 2 pour cela la valeur 0 est attribuée et il n'a pas consulté l'article 3,4,5.

- **Représentation du profil d'articles :**

	Article1	Article2	Article3	Article4	Article5	Article6
Total Catégorie	3	3	2	2	1	2

Tableau 3.3 : total catégories

	Big Data	R	Python	Machine Learning	Learning Path
Article1	**0.577350269**	**0**	**0.577350269**	**0**	**0.577350269**
Article2	**0**	**0.577350269**	**0.577350269**	**0.577350269**	**0**
Article3	**0**	**0**	**0**	**0.707106781**	**0.707106781**
Article4	**0**	**0**	**0.707106781**	**0.707106781**	**0**
Article5	**0**	**1**	**0**	**0**	**0**
Article6	**0.707106781**	**0**	**0**	**0.707106781**	**0**

Tableau 3.4 : Les vecteurs articles

- **Modélisation du profil utilisateur :**

Après avoir normalisé les articles on calcule les vecteurs profils :

	Article 1	Article 2	Article 3	Article 4	Article 5
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269
User2	-0.577350269	0.577350269	0.707106781	1.28445705	-0.577350269

Tableau 3.5 : Les vecteurs profils utilisateur

L'utilisateur 1 aime le plus les articles sur le Big Data (score le plus élevé de 1,28), suivis des learning paths, puis de machine learning. De même, **l'utilisateur 2** aime le plus les articles sur machine learning.

- **Calcul des vecteurs pondérés des articles :**

Nous allons calculer les IDF :

DF	**2**	**2**	**3**	**4**	**2**
IDF	0.69897004	0.69897004	0.52287875	0.397940009	0.69897004

Tableau 3.6 : Inverse document fréquence (IDF)

Maintenant que nous avons les vecteurs d'article et IDF nous allons calculer les vecteurs pondérés des articles.

Articles	**Big Data**	**R**	**Python**	**Machine Learning**	**Learning Path**
Article1	**0.40355052**	**0**	**0.301**	**0**	**0.4034**
Article2	**0**	**0.40355054**	**0.30188419**	**0.22975082**	**0**
Article3	**0**	**0**	**0**	**0.28138608**	**0.49424643**
Article4	**0**	**0**	**0.36973111**	**0.28138608**	**0**
Article5	**0**	**0.69897004**	**0**	**0**	**0**
Article6	**0.49424642**	**0**	**0**	**0.28138608**	**0**

Tableau 3 .7 : vecteurs pondérés d,

- **la prédiction des scores de pertinence :**

Pour la prédiction des articles le système utilise les vecteurs pondérés des articles pour un produit scalaire avec les vecteurs pondérés des utilisateurs. Cela donne le degré de pertinence qu'un utilisateur aime un article particulier.

Les résultats sont dans le tableau ci-dessous :

		
Article1	0.7513	-0.2502
Article2	-0.2032	0.739
Article3	0.3218	0.3492
Article4	0.03651	0 ,6227
Article5	-0.4035	0.4035
Article6	0.6712	0.0760

Tableau 3 .8 : degré de pertinence

Pour l'article 1, le degré de pertinence est de 0.7513 c'est l'article que l'utilisateur apprécie le plus .Ce concept est appliqué aux articles 'n' et nous pouvons déterminer quel article un utilisateur appréciera le plus.

Par conséquent, avec de nouveaux articles, une recommandation distincte peut être faite à un utilisateur particulier sur la base des articles qu'il n'a pas encore notés.

3.4 CONCLUSION :

Au long de ce chapitre, nous avons décrit notre approche. Nous avons illustré l'architecture de notre système de recommandation, et nous avons défini notre approche de modélisation du profil utilisateur à savoir sa représentation et sa construction ainsi que son exploitation dans le processus du système de recommandation. Nous avons illustré ces étapes avec un exemple.

Nous présentons dans le chapitre qui suit le résultat d'implémentation de notre approche.

CHAPITRE IV

IMPLÉMENTATION DE NOTRE APPROCHE

CHAPITRE IV

4.1 INTRODUCTION :

Dans ce chapitre, nous présentons l'implémentation de notre approche ainsi que les différents outils que nous avons exploités et enfin nous présentons les résultats obtenus.

4.2 ENVIRONNEMENT DE DEVELOPPEMENT :

✓ **Langage de programmation :** Python qui est un langage de programmation orienté objet interprété et un langage multiplateforme. La syntaxe de Python est simple et claire, elle respecte les standards du domaine. Python propose les principales fonctionnalités de la programmation (actions conditionnelles, boucles, programmation modulaire), y compris les mécanismes de classes (héritage, surcharge des méthodes, polymorphisme). Python se marie très bien avec un cours d'algorithme. La distribution Python intègre un grand nombre de librairies. Elles couvrent un large choix de domaines (bases de données, accès réseaux, multimédia, traitements systèmes, compression, multithreading, ...).Outre les librairies standards, un grand nombre de paquetages (packages) développés par des contributeurs indépendants donne accès à des fonctionnalités spécialisées performantes. Ils nous donnent la possibilité de programmer des applications dans quasiment tous les secteurs de l'informatique (machine learning, au Big data, la programmation statistique).

✓ **Plate-forme de développement est Anaconda (distribution Python) :** Anaconda est une distribution gratuite et à code source ouvert des langages de programmation Python et R pour l'informatique scientifique (informatique, applications de machine learning, traitement de données à grande échelle, analyse prédictive, etc.), qui vise à simplifier les packages, gestion et déploiement, intègre nativement un grand nombre de packages notamment ceux consacrés au calcul scientifique et aux statistiques(numpy, scipy, pandas, ,etc) .

✓ **Environnement de programmation** : Spyder et jupyter notebook intégrés dans la distribution Anaconda :

- **Spyder :** Spyder est un environnement scientifique puissant écrit en Python, pour Python, et conçu par et pour les scientifiques, les ingénieurs et les analystes de données. Il offre une combinaison unique des fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec l'exploration de données, l'exécution interactive, l'inspection approfondie et les superbes capacités de visualisation d'un progiciel scientifique. En outre, Spyder offre une intégration intégrée à de nombreux logiciels scientifiques populaires.

- **Jupyter notebook :** Jupyter Notebook est une application Web à source ouverte qui vous permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif. Les utilisations incluent: nettoyage et transformation de données, simulation numérique, modélisation statistique, visualisation de données, apprentissage automatique, etc.

4.3 APERÇUE DE NOTRE IMPLEMENTATION:

4.3.1 DETAIL DE L'APPROCHE :

✓ **Collection de données :**

Dans notre exemple nous allons utiliser un jeu de donnée Book-Crossing portant sur les évaluations faites par les utilisateurs sur un ensemble de livres.
Ce jeu de données contient 278 858 utilisateurs (anonymisés mais avec des informations démographiques) fournissant 1149780 évaluations collectées explicitement et implicitement d'environ 271379 livres.

Format : Le jeu de données Book-Crossing comprend 3 tables.

➢ **BX-Users :**

Contient les utilisateurs. Les ID d'utilisateur (`User-ID`) ont été anonymisés et mappés sur des entiers.
Les données démographiques sont fournies (`Location`, `Age`) si disponibles. Sinon, ces champs contiennent des valeurs NULL.

➢ **BX-Books :**

Les livres sont identifiés par leur ISBN respectif. Les ISBN non valides ont déjà été supprimés de l'ensemble de données. De plus, des informations basées sur le contenu sont données (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtenu à partir d' Amazon Web Services. URL reliant pour couvrir les images sont également données, apparaissant en trois saveurs différentes (Image-URL- S, Image-URL-M , Image-URL-L), à savoir, petite, moyenne, grande. Ces URL pointent vers le site Web Amazon.

➢ **BX-Book-Ratings :**

Contient les informations de notation du livre. Les notations (Book-Rating) sont soit explicites, exprimées sur une échelle de 1 à 10 (des valeurs plus élevées dénotant une appréciation supérieure), soit implicites, exprimées par 0.

❖ **Sélection de données :**

L'opération la plus importante qui nous permet de détecter et de corriger (ou de supprimer) les erreurs présentes sur les données stockées dans les fichiers CSV pour cela nous avons :

- Détecté les valeurs manquantes et les supprimer
- Déterminé les valeurs inconnus et nulles comme l'année de publication ne peut pas être nulle et les supprimer
- Filtré les données dont l'âge=0 et âge > à 122 pour cela
- Supprimé les utilisateurs nuls et les livres nuls.

Après ce traitement, nous avons un total

4.3.2 TESTS ET RESULTATS :

o **Générer du contenu pour les livres à classer :**

Importer les livres avec leur description ayant plus de 10 évaluations

	isbn	title	author	pub_year	publisher	category	descriptio
0	0002005018	Clara Callan	Richard Bruce Wright	2001.0	HarperFlamingo Canada	Actresses	In a small town in Canada, Clara Calla reluct.
1	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999.0	Farrar Straus Giroux	Medical	Describes the great flu epidemic of 1918, a o.
2	0399135782	The Kitchen God's Wife	Amy Tan	1991.0	Putnam Pub Group	Fiction	A Chinese immigrant who is convinced sh is dy.
3	0440234743	The Testament	John Grisham	1999.0	Dell	Fiction	A suicidal billionaire, a burnt-ou Washington.
4	0452264464	Beloved (Plume Contemporary Fiction)	Toni Morrison	1994.0	Plume	Fiction	Staring unflinchingly into the abyss slaver.

- **Représentation binaire selon la catégorie :**

Nous représentons les livres selon la catégorie sous forme binaire

	Accidents	Action and adventure	Actors	Actresses	Adoptees	Adventure stories	Affirmations	African American fiction	African American men	African American psychologists	...	Ryan, Jack (Fictitious character)	Savich, Dillon (Fictitious character)	Science fiction	Self-Help
0	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0

- **Vecteurs pondérés des articles :**

Nous calculons les vecteurs pondérés pour chaque article ainsi une fois générés ils seront inclut dans le module de recommandation.

```
#Affichage
idf_df_item
```

isbn	Accidents	Action and adventure	Actors	Actresses	Adoptees	Adventure stories	Affirmations	African American fiction	African American men	African American psychologists	...	Ryan, Jack (Fictitious character)	Savich, Dillon (Fictitious character)	Scie fict
0002005018	0.0	0.0	0.0	3.166134	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
000648302X	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
000649840X	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0020264763	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0020264801	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0020442408	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0060081597	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0060083298	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0060156732	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
006016767X	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
0060168013	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	

- **Les Vecteurs des profils utilisateurs :**

```
# vecteurs profil utilisateur
df_users.head(12)
```

	Accidents	Action and adventure	Actors	Actresses	Adoptees	Adventure stories	Affirmations	African American fiction	African American men	African American psychologists	...	Ryan, Jack (Fictitious character)	Savich, Dillon (Fictitious character)	Science fiction
10	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100004	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100009	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
10001	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100029	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100035	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
10005	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100053	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100066	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100067	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100088	0.0	0.0	0.0	0.0	0.777778	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
100098	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0

- **Calcul du degré de pertinence de chaque article pour l'ensemble des utilisateurs :**

isbn	10	100004	100009	10001	100029	100035	10005	100053	100066	100067	...	99885	99894	999	9991	99946	99955
0002005018	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000
000648302X	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000
000649840X	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000
0020264763	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000
0020264801	0.0	0.0	1.206194	1.507742	0.0	1.206194	0.0	1.356968	0.954903	1.05542	...	0.0	0.0	1.05542	0.0	0.829258	0.000000
0020442408	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	6.342534
0060081597	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000
0060083298	0.0	0.0	1.206194	1.507742	0.0	1.206194	0.0	1.356968	0.954903	1.05542	...	0.0	0.0	1.05542	0.0	0.829258	0.000000
0060156732	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000
006016767X	0.0	0.0	1.206194	1.507742	0.0	1.206194	0.0	1.356968	0.954903	1.05542	...	0.0	0.0	1.05542	0.0	0.829258	0.000000
0060168013	0.0	0.0	1.206194	1.507742	0.0	1.206194	0.0	1.356968	0.954903	1.05542	...	0.0	0.0	1.05542	0.0	0.829258	0.000000
006017143X	0.0	0.0	1.206194	1.507742	0.0	1.206194	0.0	1.356968	0.954903	1.05542	...	0.0	0.0	1.05542	0.0	0.829258	0.000000
0060172533	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.00000	...	0.0	0.0	0.00000	0.0	0.000000	0.000000

- **Recommandation pour un utilisateur selon son numéro :**

```
#recommander selon le numéro_user
def recommender(user_no):

    #retourner tout les isbn des livres

    isbn_no = df_predict.index

    #user predicted rating to all books
    user_predicted_rating = df_predict[user_no]

    #combiner book rating et book detail
    user_rating_book = pd.concat([user_predicted_rating,books_wd.set_index('isbn')], axis=1)

    #livres qui sont lus par les utilisateurs
    already_read = df_final[df_final['user'].isin([user_no])]['isbn']

    #recommandation des livres non lus par l'utilisateur
    all_rec = user_rating_book[~user_rating_book.index.isin(already_read)]

    return all_rec.sort_values(by=[user_no], ascending=False).iloc[0:10]
```

- **Résultat :**

```
recommender('278026')
```

```
C:\Users\CBS Computer\Anaconda3\lib\site-packages\ipykernel_launcher.py:11: FutureWarning: Sorting because non-concatenation ax
is is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  # This is added back by InteractiveShellApp.init_path()
```

	278026	title	author	pub_year	publisher	category	description
0449005410	1.145884	Horse Heaven (Ballantine Reader's Circle)	Jane Smiley	2001.0	Ballantine Books	Fiction	A novel set in the world of thoroughbred racin...
0449004503	1.145884	Death Rounds	PETER CLEMENT	1999.0	Fawcett	Fiction	The author, a former emergency room physician ...
0449130509	1.145884	Winterbourne	Susan Carroll	1987.0	Fawcett	Fiction	Beloved author Susan Carroll took the romance ...
0449006689	1.145884	Murder in Havana (Truman, Margaret, Capital Cr...	Margaret Truman	2002.0	Fawcett Books	Fiction	Asked to investigate an American pharmaceutica...
0449006344	1.145884	Angel Falls	KRISTIN HANNAH	2001.0	Ballantine Books	Fiction	Liam will do anything to break his wife out of...
0804112525	1.145884	Jazz Funeral (Skip Langdon Novels (Paperback))	Julie Smith	1994.0	Ivy Books	Fiction	During a sweltering summer, the popular produc...

4.4 CONCLUSION :

Dans ce chapitre, nous avons propos é un mod èle de profil utilisateur pour un syst ème de recommandation bas é sur le contenu.

Nous avons essayé de mettre en œuvre l'ensemble des idées qui caractérisent notre approche .Au long de ce chapitre, nous avons pr ésent é notre collection de test, les outils n écessaires pour la r éalisation de notre approche, nous avons pr ésent é un aper çu de notre impl émentation et enfin le r ésultat de la recommandation.

CONCLUSION GÉNÉRALE

CONCLUSION GENERALE

Dans notre travail nous nous sommes intéressés à l'étape de construction du profil utilisateur pour le système de recommandation, l'objectif principal de ce système est de fournir le nombre maximal des recommandations pertinentes.

Nous avons présenté les différentes étapes de modélisation de notre profil ainsi que son implémentation pour le système de recommandation et on a vu l'intérêt que l'on peut avoir en mettant en place un filtrage basé sur le contenu pour la recommandation d'articles.

Comme perspectives à notre travail nous envisageons :

- D'enrichir le profil utilisateur en utilisant le modèle multidimensionnel
- D'intégrer le facteur de temps dans la représentation du profil utilisateur
- La formalisation des conditions d'évaluation des systèmes de recommandation selon les caractéristiques du jeu de données et les objectifs de recommandations
- De prendre en compte le contexte émotionnel

BIBLIOGRAPHIE

BIBLIOGRAPHIE

Académie Française.Dictionnaire. (1932). Récupéré sur https://academie.atilf.fr.

Adomavicius et al. (2005). Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating Contextual Information in Recommender Systems Using a Multidimensional.

Baez-yates et al. (1999). Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). Modern information retrieval, volume 463. ACM press New York.

Bal et al. (2008). Baltrunas L. and Ricci F., (2008).Locally Adaptive Neighborhood Selection for Collaborative Filtering Recommendations. Proc. 5th Inter. Conf. AH 2008 , Germany, 5149, pp. 22-31.

Balabanovic et al. (1997). Balabanovic, M. and Shoham, Y. (1997).Fab :content-based collaborative recommandation.Communication of the ACM.

Bel et al, B. (2007). Bell, R. M., & Koren, Y.Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems. In Proc. of the 13th ACM SIGKDD, Inter. Conf. On Knowledge.

Breese et al, B. (1998). Breese, J. S., Heckerman, D., and Kadie, C. (1998).Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, pages 43–52.Morgan Kaufmann Publishers Inc.

Burk R. (2002). Burke, R. (2002).Hybrid recommender systems : Survey and experiments. User Modeling and User-Adapted Interaction,.

Delgado et al, D. (1999). Delgado, J. and Ishii, N. (1999).Memory-based weighted majority prediction. In SIGIR Workshop Recomm. Syst. Citeseer. Citeseer.

Desrosiers et al, D. (2011). Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In Recommender.

Ekstrand et al., 2. (2011). Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011).Collaborative filtering recommender systems .Foundations and trends in Human-Computer Interaction.

Ghazanfar et al, G. (2014). Ghazanfar, M. A. and Prügel-Bennett, A.
(2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. Expert Systems with Applications, 41(7) :3261–3275.

Goldberg et al. (1992). Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992).Using collaborative filtering to weave an information tapestry Communication of the ACM.

Heckerman et al. (2001). Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite,
R., and Kadie, C. (2001). Dependency networks for inference, collaborative filtering, and data visuajnlization. The Journal of Machine Learning Research,1 :49–75.

Kelly et all. (2003). Implicit feedback for inferring user preference: a bibliography. In ACM SIGIR Forum , Vol. 37(2), pp. 18-28. ACM.

Koren et al, K. (2011). Koren, Y. and Bell, R. (2011). Advances in collaborative filtering. In Recommender systems handbook, pages 145–186. Springer.

Lee et al. (2008). Lee, T. Q., Park, Y., & Park, Y. T. (2008). A time-based approach to effective recommender systems using implicit feedback. ESA, 34(4), pp. 3055-3062.

Linden et al. (2003). Linden, G., Smith, B., and York, J. (2003). Amazon.com Recommendations.

Lops et al . (2011). Lops, P., De Gemmis, M., and Semeraro, G. (2011).Content-based recommender systems: State of the art and trends.In recommender systems handbook.

Michalski et al. (2013). Michalski, R. S., Carbonell, J. G., and Mitchell, T. M.(2013) Machine learning:An artificial intelligence approach .Springer science & Business Media.

Nakamura et al. (1998). Nakamura, A. and Abe, N. (1998).Collaborative filtering using weighted majority prediction algorithms. In ICML, volume 98,pages 395–403.

Nguyen et al. (2006). Nguyen A. and Denos N.and Berrut C.,(2006)Modèle d'espaces de communautés basé sur la théorie des ensembles d'approximation dans un système de filtrage

hybride, Conf. en Recherche Information et Applications (CORIA), Lyon, France, pp. 303-314.

Nguyen.COCoFil2., A. T. (2006, Novembre). Nguyen A., Denos N., Berrut C., (2006).Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés. Thèse. Université joseph Fourier Grenoble I.

Ouard et al. (1998). Oard, D. W., & Kim, J. (1998, July). Implicit feedback for recommender systems. In Proc. of the AAAI workshop on recommender systems, pp. 81-83.

Pazzani et al. (2007). Pazzani, M. J. and Billsus, D. (2007).Content-based Recommandation systems.In the adaptative web.

Rao et al. (2008). Application domain and functional classification of recommender systems a survey. Desidoc journal of library and information technology.

Rendle et al. (2009). Bayesian personalized ranking from implicit feedback. In Proc. of the 25th Conf. on UAI. pp. 452-461.

Resnick et al. (1994). Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994).Recommender systems.

Ricci et al. (2011). Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommander systems handbook.

Rocchio et al. (1971). Rocchio, J. J. (1971). Relevance feedback in information retrieval.

Salton et al . (1965). THE SMART .automatic document retrievel systems an illustration .Communication of the ACM.

Salton et al. (1975a). A Vector Space of Model for Automatic Indexing .Commun.ACM.

Salton. (1989). Salton, G. (1989). The transformation, analysis, and retrieval of. Reading : Addison-Wesley.

Sarwar et al, S. (2001). Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001).Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web, pages 285–295. ACM.

Shafar et al, S. (2007). Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007).Collaborative filtering recommender systems. In The adaptive web, pages.

Shardanand and Maes et al . (1995). Shardanand, U. and Maes, P. (1995a). Social Information Filtering: Algorithms for Automating "Word of Mouth".

Sparck Jones. (1972). Sparck Jones, K. (1972).A statistical interpretation of term specificity and its application in retrieval. Journal of documentation, 28(1) :11–21.

Su et al. (2009).

Ungar et al. (1998). Ungar, L. H. and Foster, D. P. (1998). Clustering methods for collaborative filtering. In AAAI workshop on recommendation systems,volume 1, pages 114–129.

Yu et al. (2009). Yu, C., Lakshmanan, L., and Amer-Yahia, S. (2009).Personlized location-based recommandation services for tour planing in mobile tourism applications . *In international Confernce on Electronic Commerce and web Tenchnologies.*

Zhang et al. (2002). Zhang, Y., Callan, J., and Minka, T. (2002).Novelty and redundancy detection in adaptive filtering .In Processings of the 25th annual international ACM SIGR conference on Research and development in information retrievel.

A. Pretschner et al, P. o. (1999). Personalization on the web.

Adomavicius et al. (2005). incorporating contextual information in recommender systems using multidimentionnel approach.

ahu sieg et al. (2007). Ahu Sieg, Bamshad Mobasher, and Robin Burke. Web search personalization with ontological user profiles.

al, k. e. (2001). personalised hypermedia presentation techniques for improving online customer relations.

al, l. e. (2011). *Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro, content based recommender systems state of the art and trends in recommender system handbook* .

al, Z. e. (2010). ZHANG Z.-K., LIU C., ZHANG Y.-C., ZHOU T, solving the cold start problem in recommender system with social tags.

Amato et al. (1999). Amato and Umberto Straccia. User profile modeling and applications to digital libraries.

Aral et al. (2013). tie strength embeddedness and social influence.

Bruce krulwich et al. (1997). Bruce Krulwich. Lifestyle finder. Intelligent user profiling using large-scale demographic data.

Brusilovsky. (1996). Methods and techniques of adaptative hyper media, user modeling and User Adapted Interaction,.

Carmel et al. (2009). CARMEL D., ZWERDLING N., GUY I., OFEK-KOIFMAN S., HAR'EL N., RONEN I., UZIEL E., YOGEV S., CHERNOV S ,personalozed social search based of the users social networks.

Daniel Billsus et al. (1999). Daniel Billsus , Michael J. Pazzani. An hybrid user model for news story classification.

daniel billsus et al. (1999). Daniel Billsus and Michael J. Pazzani , An hybrid user model for news story classification.

Ekstrand et al. (2011). EKSTRAND M.D., RIEDL J.T., KONSTAN J.A., Collaborative filtering recommender systems.

freitag et al. (1995). D Freitag, R Armstrong, T Joachims, T Mitchell Web watcher: A learning apprentice for the word wide web.

Gabriel et al. (2001). Gabriel, Somlo and Adele E Incremental clustering for profil maintenance in information gathering web agents.

Gao et al. (2010). personnalisation in web computing and informatics theories thechniques applications and future research.

Gauch et al. (2003). Jeason Chaffee, and Alaxander Pretschner. Ontology-based personalized search and browsing.

Goldberg et al. (1992). Goldberg, David Nichols, Brian M. Oki, and Douglas Terry, Using collaborative filtering algorithm.

Gracia et al. (2015). Data processing in data mining.

Guy et al. (2009). GUY I., ZWERDLING N., CARMEL D., RONEN I., UZIEL E., YOGEV S., OFEK-KOIFMAN S, Personalized recommendation of social software items based on social relation.

Hyoung R et al. (2003). Hyoung R ,Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization.

isozaki et al. (2002). Efficient support vector classifers for named entity recognition.

Joachims. (1997). A probabilistic analysis of the rocchio algorithm.

Kelly et al. (2011). KELLY D., TEEVAN J , Implicit feed back for inferring user preference.

kiebling

searching.

Liu. (2007). explorong hyperlinks content and usage data.

Lops et al. (2011). Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro, Content based recommender systems state of the art and trends in recommender system handbook.

Lops et al. (2011). *Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems : State of the art and trends. In Recommender Systems Handbook* .

Lops et al. (2011). *Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems : State of the art and trends* .

Lops et al. (2011). content based recommender system state of the art stends .

lousan et al. (2009). Fabián P. Lousame and Eduardo Sánchez, a taxonomy of collaborative based recommender system in web personalisation in intelligent environnement.

marko balabanovic. (1997). An adaptative web page recommentation service.

Marko Balabanovic et al. (1997). content-based, collaborative recommendation.

mastchoff. (2011). Combining individual models .

mayfield et al. (2003). Single n-gram stemming.

Micro speretta et al. (2004). Mirco Speretta and Susan Gauch. personalizing search based on user search histories.

miller et al. (2003). BN Miller, I Albert, SK Lam, JA Konstan, Experiences with an occasionally connected recommander system.

Miquel montaner et al. (2003). Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa, A taxinomy of recommender agents of the internet.

Mirco Speretta et al. (2004). and Susan Gauch. Personalizing search based on user search histories.

montaner et al. (2003). *Miquel Montaner, Beatriz Lòpez, and JosepLluís de la Rosa, A taxonomy of recommander agents on the internet* .

Nesrine zemerli et al. (2005). Nesrine zemerli, LyndaTamine, and Mohand Boughanem. Accès personnalisé à l'information : Proposition d'un profil utilisateur multidimensionnel.

Paul Resnick et al. (1994). Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom,and John Ried l. An open architecture for collaborative filtering of netnews.

Pazzani. (1999). A framework for collaborative content based and demographic filtering.

Pazzani et al. (2007). content based recommendation systems .

Pazzani et al. (1996). M Pazzani, J Muramatsu, and D Billsus Syskill & Webert : Identifyinginteresting web sites.

pazzani et al. (1996). M.J.Pazzani,J.Muramatsu,andD.Billsus., Syskill Webert: Identifying interesting web sites.

porter. (2006). an algorithm for suffix stripping.

Robertson et al. (1988). document retreival systems.

Rocchio. (1971). The SMART Retrieval System : Experiments in Automatic Document Processing.

salton et al. (1973). G. Salton and C. Yan, On the specification of termes values in automatic indexing.

salton. (1971). The SMAT retrieval system: expirements in automatic document processing .

sarwar et al. (2001). items-based collaborativefiltering recommendation algorithm.

Sieg, A. (2004). Bamshad Mobasher, and Robin D. Burke. Inferring user's information context : Integrating user profiles and concept hierarchies.

Somlo et al. (2003). G. Somlo and A. Howe, Using web helper agent profiles in query generation l.

Struarte et al. (2003). Stuart E. Middleton, Nigel R. Shadbolt ,and David C. De Roure, Capturing knowledge of user preferences.

stuarte et al. (2001). Stuart E. Middleton, David C. De Roure, and Nigel R. Shadbolt, Ontologies in recommender systems.

Susan et al. (2003). Susan Gauch, Jeason Chaffee, and Alaxander Pretschner, Ontology based personalized search and browsing.

Tebri et al. (2005). H. Tebri, M. Boughanem, and C. Chrisment.Incremantal profile learning based on a reinforcement method.

Upendra shardanand et al. (1995). Implicit user modeling for personalized search .

upendra shardanand et al. (1995). implicit user modeling for personalized search.

Vistu kanth. (2004). Vishnu Kanth Reddy Challam, Contextual information retreival using based user profiles.

webb et al. (1995). feature based modelling .

wen et al. (2011). WEN Z., LIN C.-Y, Improving user internets inference from social neibors.

young woo seo et al. (2000). Young-Woo Seo and Byoung-Tak Zhang, A reinforcement learning agent for personalized information filtering.

Dieudonnée tchuente . (2013). Modélisation et dérivation de profil utilisateurs à partir de réseaux sociaux : approche à partir de communautés de réseaux k-égocentrique.

Thanh trung van . (2008). Utilisation de profils utilisateurs pour l'accées à une bibliothèque numérique.

Sonia ben ticha . (2015). Recommandation hybride personnalisée.

Sirinya .(2017). Temporalité et reseaux sociaux : prise en compte de l'évolution dans la construction de profils utilisateur.

Martin arnaud. (2012). L'évolution de profil multi attributs par l'apprentissage automatique et adaptative dans un système de recommandation pour l'aide à la décision.

Dieudonnée tchuente . (2013). Modélisation et dérivation de profil utilisateurs à partir de réseaux sociaux : approche à partir de communautés de réseaux k-égocentrique.

Thanh trung van . (2008). Utilisation de profils utilisateurs pour l'accées à une bibliothèque numérique.

Sonia ben ticha . (2015). Recommandation hybride personnalisée.

Sirinya .(2017). Temporalité et reseaux sociaux : prise en compte de l'évolution dans la construction de profils utilisateur.

Martin arnaud. (2012). L'évolution de profil multi attributs par l'apprentissage automatique et adaptative dans un système de recommandation pour l'aide à la décision.