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

INTEGRATION D'UN PROFIL UTILISATEUR MULTIDIMENSIONNEL DANS UN SYSTEME DE RECOMMANDATION BASE SUR LE CONTENU

Proposé et dirigé par :

M^me^ ACHEMOUKH Farida

Réalisé par :

AIT SAID Rosa

BOUTLENDJ Feriel

JURY:

Présidente : M^me^ S.FELLAG

Examinatrice : M^me^ M.BENTAYEB

Année universitaire : 2019/2020

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 ma chère grande mère que Dieu très haut lui accorde Santé et longue vie.

A mes chers

parents, A mes

chères sœurs,

A mon cher

oncle Mohand

A mon binôme Farfou,

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

Rosa

Du profond de mon cœur je dédie ce modeste travail à tous ceux qui me sont chers, A ma chère grande mère que Dieu très haut lui accorde Santé et longue vie,

A mes chers parents,

A mes chers frères,

A mes chères sœurs,

A mon très cher ami "Hocine",

A mon binôme Ro,

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

Feriel

Table des matières

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 Concept de base, notions et notations liées :……...3
1.3.1 L'utilisateur et l'item :……..4
1.3.2 Evaluation (note ou vote) :……4
1.3.3 Matrice d'évaluation utilisateur-item :……5
1.3.4 Filtrage d'information :……5
1.3.5 Notion de communauté :……..6
1.3.6 Notion de profil :……..6
1.3.7 La prédiction :……...7
1.3.8 La personnalisation :……8
1.3.9 La différence entre la personnalisation et la recommandation :……...8
1.4 Les fonctionnalités des systèmes de recommandation :……9
1.5 Les techniques de recommandation :……..9
1.5.1 Recommandation basé sur le contenu :……..10
1.5.2 Recommandation basée sur le filtrage collaboratif :……..18
1.5.3 La recommandation hybride :……21
1.6 Conclusion :……27

CHAPITRE II : MODELISATION DU PROFIL UTILISATEUR EN SYSTEMES DE RECOMMANDATION

2.1 Introduction :……..28
2.2 Approche de représentation du profil utilisateur :……28
2.2.1 Représentation ensembliste :……..28
2.2.2 Représentation sémantique à base d'ontologie :……29
2.2.3 Représentation multidimensionnelle :……29
2.3 Construction du profil utilisateur :……..30
2.3.1 Acquisition des données utilisateurs :……31
2.3.2 Le prétraitement des données :……34
2.4 Les techniques de construction du profil utilisateur :……35
2.4.1 Technique tf_idf (terms frequency_inverse document frequency):……35
2.4.2 Les méthodes de classification:……..36
2.4.3 Les méthodes de clustering :……...36
2.5 Adaptation et évaluation du profil utilisateur :……....37
2.6 Conclusion :……38

CHAPITRE III : SPECIFICATION D'UN PROFIL UTILISATEUR MULTIDIMENSIONNEL POUR LA RECOMMANDATION

3.1 Introduction :……39
3.2 Problématique et motivation :……...39
3.3 Architecture générale de l'approche de recommandation :……39
3.4 Description de l'approche de recommandation :……40
3.4.1 Définition du profil utilisateur :……..40
3.4.2 Définition du profil item :……...45
3.4.3 Exploitation et intégration du profil utilisateur dans le processus de recommandation :…… 46

3.4.4 Illustration de l'approche proposée :……...47
3.5 Conclusion :……53

CHAPITRE IV : EVALUATION ET VALIDATION EXPERIMENTALE DU MODELE PROPOSE

4.1 Introduction :……...54
4.2 Environnement de développement :……..54
4.3 Description détaillées des fichiers de données :……55
4.3.1 Les fichiers disponibles avec la base :……55
4.3.2 Les fichiers ajoutés :……...57
4.4 Implémentation de notre approche :……59
4.4.1 Sélection de données :……59
4.4.2 Construction du profil film :……..59
4.4.3 Construction du profil utilisateur :……63
4.4.4 Recommandation pour un utilisateur selon son numéro :……..67
4.5 Expérimentation : évaluation de la performance de l'approche proposée :…….69
4.5.1 Métriques d'évaluation :……69
4.6 Conclusion :……73

CONCLUSION GENERALE……74

BIBLIOGRAPHIE……..75

LISTE DES FIGURES

Figure(1.1): Architecture haut niveau d'un système de recommandation basé sur le contenu (d'après (Lops et al ; 2011)……11
Figure (3.1) : Architecture générale de l'approche de recommandation……40
Figure (3.2) :Liste des recommandations pour l'utilisateur 1……..……52
Figure (4.1) : Liste des films contenant dans le fichier « movie.dat »……59
Figure (4.2) : Le code effectué pour obtenir la représentation binaire des films……60
Figure (4.3) : la représentation binaire des films……60
Figure (4.4) : le code de la normalisation des vecteurs films……..61
Figure (4.5) : les vecteurs films normalisés……61
Figure (4.6) : la fonction de pondération des films……62
Figure (4.7) : les vecteurs pondérés des films……..62
Figure (4.8) : les traitements effectués sur le dataset ……..63
Figure (4.9) : la table pleine de dataset « df »……..63
Figure (4.10) : la fonction qui calcule le vecteur centre d'intérêt des utilisateurs……64
Figure (4.11) : le vecteur de la dimension centre d'intérêt de l'utilisateur numéro 1……64
Figure (4.12) : les méthodes qui génére le vecteur de la dimension genre……65
Figure (4.13) : le vecteur de la dimension genre (femme / homme) ……...65
Figure (4.14) : la méthode qui génére le vecteur la dimension âge……66
Figure (4.15) : le vecteur âge de l'utilisateur 1……...66
Figure (4.16) : le profil utilisateur multidimensionnel de l'utilisateur 1……66
Figure (4.17) : l'algorithme de recommandation……68
Figure (4.18) : la liste des recommandations pour l'utilisateur numéro 1……..68
Figure (4.19) : le rappel pour les 20 premiers utilisateurs……...71
Figure (4.20) : la précision pour les 20 premiers utilisateurs……..72
Figure (4.21) : la valeur de F1 mesure obtenue par notre système……..72

LISTE DES TABLEAUX

Tableau 3.1: représentation binaire des filmes……47
Tableau 3.2 : notes des utilisateurs sur les films…….47
Tableau 3.3 : Les vecteurs pondérés des films……48
Tableau 3.4 : Les vecteurs centres d'intérêts……..48
Tableau 3.5: degré de pertinence dimension centres d'intérêts -item……49
Tableau 3.6: données personnelles des utilisateurs……49
Tableau 3.7: notes des utilisateurs sur les films selon leur genre……50
Tableau 3.8 : Les vecteurs profils des genres utilisateurs……...50
Tableau 3.9: degré de pertinence dimension genre -item ……...50
Tableau 3.10: notes des utilisateurs sur les films selon leur âge……50
Tableau 3.11 : Les vecteurs profils des catégories d'âges des utilisateurs……51
Tableau 3.12: degré de pertinence dimension âge -item ……52
Tableau 4.1 : classement des possibilités de la recommandation d'un item pour un utilisateur……...68

Introduction générale

Introduction générale

Les systèmes de recommandation sont une forme spécifique du filtrage de l'information. Ils visent à présenter les éléments d'information, qui sont susceptibles d'intéresser l'utilisateur.

Ces systèmes facilitent la recherche d'une information pertinente pour les utilisateurs, on leur fournissant des recommandations en fonction de leurs préférences. Ces systèmes sont devenus populaires à la fois dans le e-commerce et dans la communauté de recherche d'information, où de nombreuses approches ont été suggérées pour fournir des recommandations.

Dans le cadre des systèmes de recommandations des films, nous adaptons dans ce travail une approche basée sur le contenu pour un profil utilisateur multidimensionnel. 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.

L'une des difficultés majeures des systèmes de recommandation est la construction des dimensions du profil utilisateur basé sur la représentation multidimensionnelle. Ce profil 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 :

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 les inconvénients de chaque approche.

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.

Dans le troisième chapitre, nous expliquons l'approche que nous avons proposée.

Dans le dernier chapitre, nous allons implémenter l'approche proposée puis évaluer notre système en se basant sur une collection de tests.

Nous terminant par une conclusion générale.

Chapitre I :

Etat de l'art sur les systèmes de recommandation

1.1 Introduction :

Le web permet de nouvelles techniques capable de guider l'utilisateur vers des ressources susceptibles de l'intéresser et de répondre efficacement à ses besoins en information. Ces techniques dites systèmes de recommandation.

La première apparition des ces systèmes date de l'année 1992, avec le système tapestry [Goldberg et al.1992], qui pouvait calculer et envoyer des recommandations aux utilisateurs par email.

Dans ce chapitre, nous allons voir d'abord c'est quoi un système 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 technique. La dernière section conclue le chapitre.

1.2 Définition des systèmes de recommandation :

Les systèmes de recommandation sont définis comme étant des outils logiciels et des techniques qui suggèrent aux utilisateurs des éléments utiles, mais vue la diversité des classifications proposées pour ces systèmes, il existe une définition générale de Robin Burke (Burke, 2002) qui les définit comme suit:

"Des systèmes capable de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sain d'un espace de données important".

1.3 Concept de base, notation et notion liées :

Dans cette section, nous définissons quelques concepts relatifs aux systèmes de recommandation qui seront utilisés par la suite.

1.3.1 L'utilisateur et l'item:

En général, l'entrée d'un système de recommandation se compose d'un ensemble d'utilisateurs, d'un ensemble d'items ou d'articles.

- **l'utilisateur** : est la personne qui utilise le système de recommandation, donne sont avis sur divers items et reçoit les nouvelles recommandations de système.

- **L'item** : ensemble des éléments possiblement recommandable aux utilisateurs.

1.3.2 Evaluation (note ou vote) :

- **Note :** une note est une valeur numérique représentant l'appréciation d'un utilisateur pour une ressource.
- **Vote** : un vote exprime une préférence pour une ressource.

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 *r* donné par un utilisateur *u* à un item *i* est représenté 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. On a deux types d'évaluation :

- **Évaluation explicite** : la note est attribuée directement par un utilisateur à un item en donnant une valeur numérique ou binaire à travers l'interface du système.

- **Évaluation implicite :** dans ce cas c'est le système qui s'occupe de la détection des préférences de l'utilisateur en utilisant des algorithmes et des techniques spécifiques.

1.3.3 Matrice d'évaluation utilisateur-item :

Une matrice creuse qui relie les utilisateurs et les items par des score (une note, nombre d'utilisation, durée), 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.

1.3.4 Filtrage d'information :

Le filtrage d'information est un processus qui permet de fournir l'information adéquate aux utilisateurs qui en ont besoin, son but principal est de suggérer des contenus pertinents en éliminant les données indésirables sur un flux entrant.

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 réfèrent à des informations telles que l'âge, le sexe, le pays et l'éducation des utilisateurs. Ce type de données est généralement difficile à obtenir et est normalement collecté explicitement.

- **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 [Margaritis and Vozalis, 2003].

1.3.5 Notion de communauté :

Une communauté ou groupe est un ensemble d'utilisateurs similaires qui partagent les mêmes préférences et goûts et qui sont regroupés relativement à un critère donné.

Plusieurs critères peuvent être utilisés par le processus de formation des communautés, citant par exemple le contenu des items évalués par les utilisateurs, les évaluations qu'ils ont données aux items, leurs données démographiques et leur domaine d'intérêt. Selon chacun de ces critères, les communautés créées par le système varient et les positions des utilisateurs dans ces communautés changent. Par conséquent, chaque utilisateur peut appartenir à autant de communautés qu'il y a de critères utilisés pour leur formation**.**

1.3.6 Notion de profil :

Le profil d'un objet est un ensemble de caractéristiques permettant de l'identifier ou de le représenter. Deux types de profils peuvent être exploités dans les systèmes de recommandation, correspondant aux deux entités utilisées dans ces systèmes : l'utilisateur et l'item.

1.3.6.1 Le profil utilisateur:

Il s'agit d'une description des caractéristiques de l'utilisateur qui peuvent être ses centres d'intérêt, ses données démographiques, ou bien ses préférences exprimées sous forme d'évaluations, etc.

- **Construction de profil utilisateur :**

Plusieurs approches d'acquisition des informations sur l'utilisateur afin de construire son profil :

- **Approches manuelles** : se fondent sur l'intervention de l'utilisateur, alors que les approches automatiques déduisent automatiquement le profil de l'utilisateur.

- **Approches automatiques ou semi-automatiques :** deux approches sont distinguées : le '*profiling*' et les approches par stéréotypes. Le '*profiling*' consiste à examiner, analyser et enregistrer les actions et successions d'actions d'un utilisateur, lors des différentes sessions de recherche et d'interactions avec le système pour déterminer son profil.

En revanche, l'approche par stéréotype est fondée sur l'identification des groupes d'utilisateurs et la détermination des critères de chaque groupe.

Un stéréotype d'un utilisateur dans un système de recommandation consiste en un vecteur d'items et leurs appréciations qui augmentent d'une façon continue tant que l'utilisateur interagit avec le système au cours du temps [BUR 02].

1.3.6.2 Le profil item :

Il correspond à la description de l'item avec un ensemble de caractéristiques, appelées aussi attributs ou propriétés, par exemple dans un système de recommandation de films, les items (films) sont représentés par leurs Id, titre, genre, réalisateur, année de production, les acteurs principaux. Tandis que dans un système de recommandation de document, les caractéristiques sont des mots clés qui décrivent le contenu sémantique du document. Ces mots clés et leurs poids sont obtenus en général par une opération d'indexation [RIJ 79].

Les documents peuvent être représentés également par d'autres informations qui ne sont pas liées à leur contenu appelées métadonnées, ces dernières peuvent se présenter sous forme de propriétés liées aux documents (type du document, profession de l'auteur, etc.), ou juste à des parties des documents (style, forme discursive, type d'unité documentaire, etc.) [**CRU 99**].

1.3.7 La prédiction :

Elle est subdivisée en deux partie : la première est la prédiction de la note probable que l'utilisateur va attribuer à un item qu'il n'a pas encore évalué. Cette prédiction est nécessaire

vu que la matrice utilisateurs-items est creuse alors toutes les cases à valeurs manquantes doit être compléter avec cette méthode.

La deuxième partie est la prédiction de la ressource, qui consiste à calculer une liste de ressources que l'utilisateur aimera le plus on utilisant la similarité entre la ressource et le profil utilisateur.

La prédiction de note est facilement exploitable pour effectuer de la recommandation : une fois que l'on a déterminé les notes de toutes les ressources qu'un utilisateur donné n'a pas encore consulté, il suffit de lui recommander les ressources ayant les meilleures notes prédites.

1.3.8 La personnalisation :

Une notion proche de la recommandation est la personnalisation. C'est une notion souvent vague dans la littérature, et qui peut se rapporter à des concepts différents. La définition donnée par l'académie française est la suivante :

"Action d'adapter quelque chose à une personne selon ses goûts, ses besoins ou ses moyens".

1.3.9 La différence entre la personnalisation et la recommandation :

La personnalisation représente la part d'adaptation des recommandations aux goûts et aux besoins de l'utilisateur (Hagen, Manning et Souza, 1999), par contre une recommandation pouvant être plus ou moins personnalisée.

En particulier, un système de recommandation, peut fournir à l'utilisateur une recommandation qui sort du cadre de ses préoccupations et centre d'intérêts habituels.

1.4 Fonctionnalités des systèmes de recommandation :

Les systèmes de recommandation sont caractérisés par différentes fonctionnalités, à savoir :

- **la prédiction de rang :** arrangement d'articles par ordre de pertinence.
- **la prédiction de score** : aide à prendre une décision si un utilisateur va apprécier un article i.
- **la recommandation article à articles** : aide à l'exploration de nouveaux articles similaire à un article i.
- **la recommandation personnalisée d'articles** : découverte de nouveaux articles qui sont susceptibles d'intéresser un utilisateur u.

1.5 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 deux classes de recommandations qui sont 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.

1.5.1 Recommandation basé sur le contenu :

Dans cette approche, chaque item est défini par un ensemble d'attributs et leur valeur. Les items ayant des valeurs proches pour leurs attributs sont considérés approchants (similaires). Quand un utilisateur attribue une bonne note à un item, les items approchants sont considérés comme des recommandations potentielles [Van Meteren and Van Someren2000, Basu *et al.*1998, Pazzani1999, Pazzani and Billsus2007].

L'approche basée sur le contenu, construit le profil d'un utilisateur en examinant les centres d'intérêt de ce dernier. Pour ce faire, il s'appuie sur les traits des items que l'utilisateur a noté.

1.5.1.1 Approche générale :

La recommandation des items basée sur le contenu nécessite deux ensembles à définir : 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 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 aimé et d'un profil "négatif" représentant les items qu'il a détesté.

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.

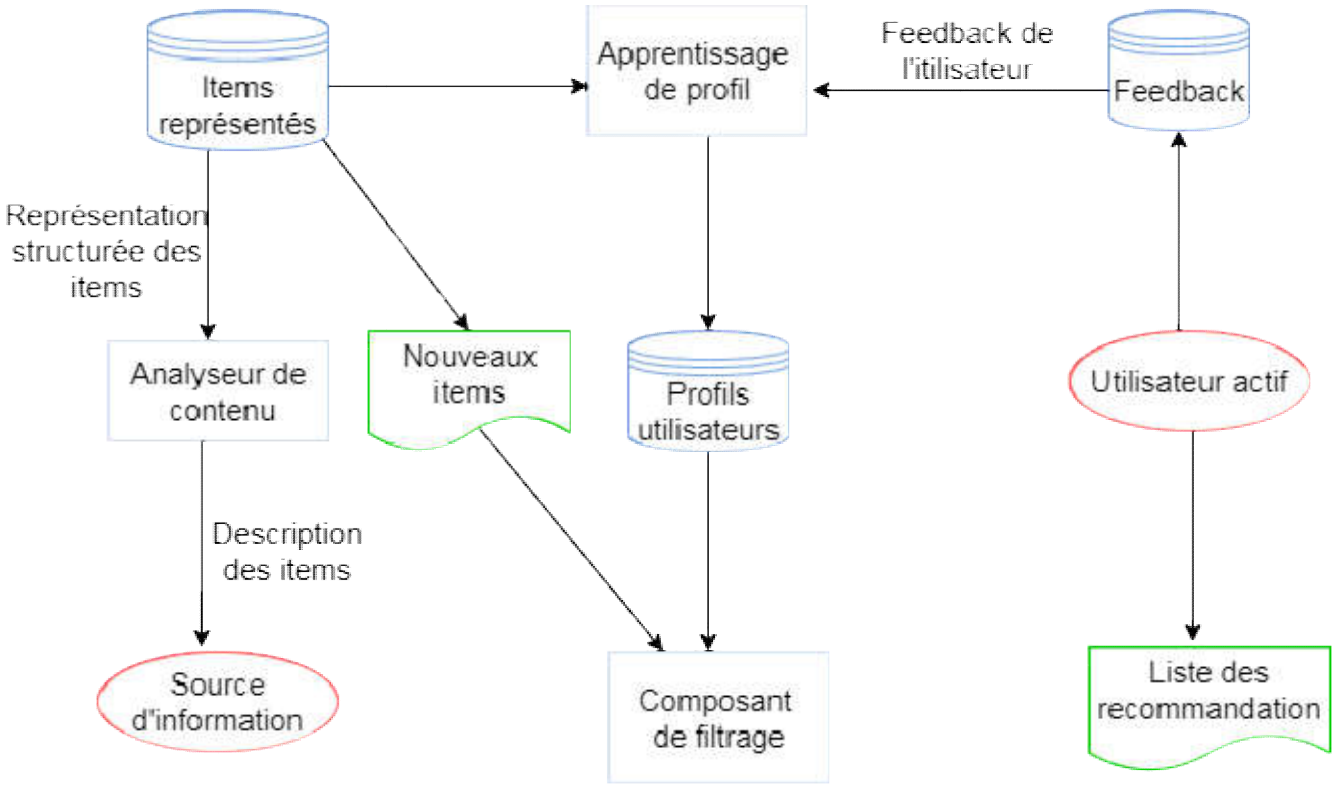

Figure 1: Architecture haut niveau d'un système de recommandation basé sur le contenu (d'après (Lops et al ; 2011).

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) 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 (Michal ski 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 aimé ou n'a pas aimé.

- **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.

1.5.1.2 Représentation 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.

1.5.1.3 Les types de recommandation basée sur le contenu:

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

1.5.1.3.1 Recommandation basée sur les vecteurs mots clés :

Le modèle de représentation vectoriel VSM (Vector Space Model) avec la pondération classique TF-IDF (Term Frequency-Inverse Document Frequency), considérer comme étant le modèle le plus utilisé dans la plupart des systèmes de recommandation basés sur le contenu. 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 = $\{d_1, d_2, \dots, d_n\}$ 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 la tokenization, l'élimination des mots vides de sens, et la troncature (stemming) (Baez-yates et al, 1999). Chaque document dj est représenté par un vecteur dans un espace vectoriel à *n* dimensions, tel que dj = $\{w_{1j}, w_{2j}, \dots \dots, w_{nj}\}$ où w_{kj} est le poids de terme t_k dans le document *dj*.

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):

$$\boldsymbol{TF.IDF}(\boldsymbol{t_k}, \boldsymbol{d_j}) = \boldsymbol{TF}(\boldsymbol{t_k}, \boldsymbol{d_j}) \ * \ \log\left(\frac{\boldsymbol{N}}{\boldsymbol{n_k}}\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.. (\mathbf{1.1})$$

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 :

$$\boldsymbol{TF}(\boldsymbol{t_k}, \boldsymbol{d_j}) = \frac{\boldsymbol{f_{k,j}}}{\boldsymbol{max_z f_{z,j}}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots. (\mathbf{1.2})$$

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

$max_z f_{z,j}$: est le maximum des fréquences $f_{z,j}$ des termes t_k 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 :

$$\boldsymbol{W_{kj}} = \frac{\boldsymbol{TF.IDF}(\boldsymbol{t_k}, \boldsymbol{d_j})}{\sqrt{\sum_{s=1}^{|T|} \boldsymbol{TF.IDF}(\boldsymbol{t_s}, \boldsymbol{d_j})^2}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (\mathbf{1.3})$$

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

$$\boldsymbol{content(d_j) = (w_{1j}, w_{2j}, \dots, w_{nj})} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (\mathbf{1.4})$$

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, citons la similarité cosinus :

$$\boldsymbol{Sim(d_i, d_j) = \frac{\sum_k w_{ki} w_{kj}}{\sqrt{\sum_k w_{ki}^2} * \sqrt{\sum_k w_{kj}^2}}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (\mathbf{1.5})$$

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 $ContentBasedProfil(u)$ le profil de l'utilisateur u contenant ses préférences. Ce profil est obtenu en analysant le contenu des items qu'il a noté 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, $ContentBasedProfil(u)$ 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 .Il peut être obtenu à partir des vecteurs du contenu des items que l'utilisateur a noté en utilisant différentes techniques.

Par exemple, l'algorithme de Rocchio (Rocchio et al, 1971) a largement été utilisé pour déterminer $ContentBasedProfil(u)$ 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.

1.5.1.3.2 Recommandation basée sur la sémantique :

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. 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.

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

Un descripteur d'article (ou profil d'un item) 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 ».

1.5.1.3.3 Exemple de système de recommandation basée sur le contenu :

Un système de recommandation basé sur le contenu a été proposé par (Chandrasekaran *et* al, 2008) 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.

1.5.1.3.4 Avantages et inconvénients des systèmes de recommandation basée sur le contenu :

Les approches basées sur le contenu présentent plusieurs avantages et inconvénients.

- **Les avantages :**

➢ **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.

- **Les 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 noté ou observé 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 à 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.

1.5.2 Recommandation basée sur le filtrage collaboratif:

Le filtrage collaboratif 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 et les méthodes basées sur un modèle.

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 stockées 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 l'item.

1.5.2.1 Les approches de filtrages collaboratifs basées 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 candidat *i* à partir des notes de *u* pour les items voisins de *i.*

1.5.2.2 Avantages et inconvénients de filtrage collaboratif:

- **Les avantages :**
  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 nécessitée 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.

- **Les inconvénients :**

➢ **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).

1.5.3 La recommandation hybride :

Les systèmes de recommandation hybrides combinent les techniques de recommandation dites pures pour atteindre une meilleure performance (**BUR, 2002**), en tirant profit des avantages des approches employées, et limitant les problèmes qui leurs sont liés. Les systèmes hybrides sont devenus les systèmes les plus communément utilisés ces dernières années.

En se basant sur les différentes techniques principales de recommandation (filtrage collaboratif, filtrage basé contenu et le filtrage démographique), diverses techniques de recommandation hybrides ont été introduites et testées, et qui ont surpassé les systèmes à composant unique.

L'hybridation peut être établie en combinant des techniques différentes de types différents, où l'hybridation la plus fréquente est celle qui combine le filtrage collaboratif avec le filtrage basé contenu **(BUR, 2002) (WEN *et al,* 2008).** En outre, il est également possible de fusionner les différentes techniques du même type, par exemple, en hybridant les deux types du filtrage basé contenu utilisant l'algorithme Naïve de Bayes et l'algorithme des k-plus proches voisins, respectivement, ou les deux types du filtrage collaboratif, basé mémoire et basé modèle **(CHE *et al,* 2009),** ou même le filtrage collaboratif basé utilisateur avec filtrage collaboratif basé item **(WAN *et al,* 2006).**

L'approche hybride a été conçue pour faire face aux problèmes des systèmes de recommandation classiques présentés précédemment.

Dont les principaux problèmes qui ont été largement abordés par les chercheurs dans ce domaine, sont le problème du démarrage à froid, les problèmes de la clairsemé et de l'évolutivité de données, et le problème de l'adaptabilité des systèmes au changement des profils des utilisateurs (la stabilité par rapport aux problèmes de plasticité).

En vue d'éviter ces problèmes, en particulier le problème du démarrage à froid (**BUR, 2002**) pour un nouvel utilisateur et pour un nouvel item, l'hybridation entre le filtrage collaboratif et le filtrage basé sur la confiance est généralement adoptée en utilisant le contenu des items, qui représente le profil des utilisateurs, pour détecter les similarités entre eux et construire les différentes communautés utilisées dans le filtrage collaboratif. Ou bien, pour détecter la similarité entre les items les plus préférés par les voisins de l'utilisateur actif, générés par un processus du filtrage collaboratif, et les items évalués par l'utilisateur afin de lui générer ceux qui sont intéressant pour lui.

Différentes possibilités de combinaison entre les méthodes de recommandation à donner naissance à sept différentes méthodes d'hybridation proposées par Burke (**BUR, 2002):** la

pondération, le mixage, la distribution, la combinaison de caractéristiques, cascade, l'augmentation de caractéristique, et le méta-niveau. Nous définissons dans les sous sections suivantes ces différentes possibilités d'hybridation.

1.5.3.1 L'Hybride Pondéré (weighted) :

Un système de recommandation hybride pondéré est celui dans lequel le score d'un item recommandé est calculé à partir des résultats de toutes les techniques de recommandation disponibles dans le système. Donc, le plus simple hybride pondéré est celui qui combine les scores de recommandation de chaque algorithme en utilisant une formule linéaire.

L'avantage d'un système hybride pondéré est que toutes les capacités du système sont amenées à porter sur le processus de la recommandation d'une manière simple, et il est facile d'effectuer a posteriori l'affectation de poids et d'ajuster en conséquence l'hybride.

Cependant, l'hypothèse implicite dans cette technique est que la valeur relative des différentes techniques est uniforme, plus ou moins uniforme dans l'espace des items possibles.

1.5.3.2 L'Hybride par Commutation (Switching) :

Un hybride par commutation utilise un critère de bascule entre les techniques de recommandation. La question de cet hybride est la sélection de l'algorithme adéquat à chaque cas parmi les algorithmes de recommandation. Donc, un critère pour la sélection de l'algorithme qui convient comme valeur de la confiance ou un critère externe devrait exister, avec lequel les composants de la méthode hybride pourraient avoir des performances différentes à des situations différentes.

Les commutations hybrides introduisent une complexité supplémentaire dans le processus de recommandation, puisque les critères de commutation doivent être déterminés, ce qui introduit un autre niveau de paramétrage. Cependant, l'avantage est que le système peut être sensible aux forces et aux faiblesses de ses recommandataires constitutifs.

1.5.3.3 L'Hybride Mixte (Mixed) :

Il s'agit d'un hybride qui est fondé sur l'intégration et la présentation de plusieurs listes de recommandation en une seule. Chacun des algorithmes de cet hybride devrait être en mesure de produire des listes de recommandation avec des rangs et l'algorithme de base de l'hybride mixte les fusionne en une seule liste classée. La question ici est de savoir comment les nouveaux scores classés (les scores de la méthode hybride mixte) devraient être produits.

L'avantage de l'hybride mixte est qu'il évite le problème du démarrage à froid d'un nouvel item: la composante basée sur le contenu peut être invoquée pour recommander de nouveaux items sur la base de leurs descriptions, même s'ils n'ont pas été notés par personne.

Cependant, il souffre du problème de démarrage à froid pour un nouvel utilisateur puisque les deux méthodes, basée contenu et collaborative, ont besoin de quelques données sur les préférences de l'utilisateur pour pouvoir démarrer.

1.5.3.4 L'Hybride par Combinaison de caractéristiques (Feature Combination) :

Un autre moyen pour la fusion contenu/collaboration est de traiter l'information collaborative comme simplement des données de caractéristiques supplémentaires associées à chaque exemple, et utiliser les techniques basées sur le contenu sur cet ensemble de données augmentées.

Ils existent deux composants de recommandation très différents pour cet hybride, la recommandation contribuée et la recommandation actuelle.

Les recommandations actuelles travaillent avec les données modifiées par la recommandation contribuée. La recommandation injecte les caractéristiques d'une source d'un algorithme à la source d'un autre algorithme.

L'hybride par combinaison de caractéristiques permet au système de tenir compte des données collaboratives sans compter exclusivement sur elles, alors il réduit la sensibilité du système au nombre d'utilisateurs qui ont noté un item. Inversement, il laisse le système avoir

des informations sur la similarité inhérente des items qui sont par ailleurs opaques pour un système collaboratif.

1.5.3.5 L'Hybride en Cascade (Cascade) :

Contrairement aux algorithmes hybrides précédents, cet algorithme implique un processus par étape. Dans cet algorithme, une technique de recommandation est employée en premier pour produire un classement en gros des items candidats et une deuxième technique affine la recommandation parmi l'ensemble des items candidats.

L'hybride en cascade permet au système d'éviter d'employer la deuxième technique, de faible priorité, sur les items qui sont déjà bien différenciés par la première, ou qui sont mal-évalués et qui ne seront jamais recommandés. Parce que la deuxième étape de l'hybride en cascade se concentre uniquement sur les items pour lesquels une discrimination supplémentaire est nécessaire.

En outre, cet hybride est de par nature tolérant au bruit dans le fonctionnement d'une technique de basse priorité (la deuxième), car seules les évaluations attribuées par le recommandataire de haute priorité (la première technique) peuvent être affinées, pas renversées.

1.5.3.6 L'Hybride par Augmentation de Caractéristiques (Feature augmentation) :

L'hybride par augmentation de caractéristiques emploie une technique pour produire une note ou le classement d'un item et l'information, et ensuite cette information est incorporée dans le processus de la technique suivante.

Ce type d'hybride est similaire à l'hybride par combinaison de caractéristiques, mais différent en ce que le contributeur génère de nouvelles caractéristiques.

Il utilise un algorithme pour calculer les caractéristiques qui seront utilisées comme entrée pour un autre algorithme. Il est plus flexible et ajoute de plus petite dimension que la méthode

par combinaison de caractéristiques. La méthode par augmentation est intéressante, car elle offre un moyen d'amélioration de la performance d'un système de base.

Une fonctionnalité supplémentaire est ajoutée par des intermédiaires qui peuvent utiliser d'autres techniques pour augmenter les données elles-mêmes.

Bien que les deux techniques, en cascade et par augmentation de caractéristiques, enchaînent deux recommandeurs, avec le premier possède une influence sur le second, elles sont fondamentalement très différentes.

Dans un hybride par augmentation, les caractéristiques utilisées par le second recommandeur incluent la sortie du premier, telles que les évaluations par exemple.

Dans un hybride en cascade, la deuxième technique n'utilise pas la sortie de la première technique dans la production de ses classements, mais les résultats des deux recommandeurs sont combinés de manière prioritaire.

1.5.3.7 L'Hybride de Méta-niveau (Meta-level) :

Une autre façon que deux techniques de recommandation peuvent être combinées est d'utiliser le modèle généré par l'un comme entrée pour un autre. Cela diffère de la méthode d'augmentation de caractéristiques, car dans un hybride d'augmentation, un premier modèle appris à générer des caractéristiques pour l'entrée d'un second algorithme, alors que dans un hybride de méta-niveau, l'ensemble du modèle devient l'entrée.

Pour cet hybride, une technique est employée pour produire une note ou un classement pour un item, ensuite l'information trouvée (note ou classement) sera incorporée dans le processus de l'autre technique de recommandation.

1.6 Conclusion :

Dans ce chapitre, nous avons présenté les systèmes de recommandation. En détaillant les approches les plus utilisées, ainsi que leur hybridation.

La mise en ouvre de ces système nécessite l'intégration d'un profil utilisateur .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 :

Modélisation du profil utilisateur en système de recommandation

2.1 Introduction :

Les profils utilisateurs jouent un rôle déterminant dans les systèmes de recommandation. Ils permettent de modéliser et stocker des informations relatives aux utilisateurs (préférences, intérêts ...).

Ce chapitre porte sur les étapes de la modélisation d'un profil utilisateur. Nous présentons en premier les approches de sa représentation. En deuxième partie, nous présentons les techniques de sa construction et de son évolution. Nous concluons le chapitre par une conclusion.

2.2 Approches de représentation du profil de l'utilisateur :

Le profil utilisateur peut être représenté en intégrant la sémantique ou encore en utilisant une représentation vectorielle ou multidimensionnelle. Dans cette section, nous présentons les principales représentations des modèles des profils utilisateurs.

2.2.1 Représentation ensembliste:

La modélisation d'un profil utilisateur de manière ensembliste a pour objectif de formaliser les données du modèle en un ensemble de termes pondérés ou en vecteurs de termes pondérés (Lieberman, 1998) ou en classes de vecteurs de termes pondérés (Gowan, 2003) souvent représentés selon le modèle vectoriel de Salton (Salton et Yang, 1973). Ces termes traduisent les centres d'intérêt de l'utilisateur. Le poids d'un terme est souvent calculé selon le schéma TF*IDF.

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).

2.2.2 Représentation sémantique à 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 pour ce concept. De plus, chaque concept est souvent représenté par un vecteur de termes pondérés. 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).

2.2.3 Représentation multidimensionnelle :

Un autre type de modélisation se propose de structurer le profil selon un ensemble de dimensions représentées selon divers formalismes : c'est la modélisation multidimensionnelle (Bouzeghoub and Kostadinov, 2005). C'est la première approche qui offre un modèle général et dont les informations sont structurées. Les catégories sont au nombre de cinq :

1. **Le domaine d'intérêt :** regroupe tous les attributs qui concernent les objets de contenu (informations ciblées). Cette dimension peut aussi bien définir le domaine d'expertise et le niveau de qualification de l'utilisateur que le contenu auquel il s'intéresse.

2. **Les données personnelles :** ces données représentent la partie statique du profil et contiennent des informations qui décrivent l'utilisateur et ne dépendent pas du système à interroger

3. **La qualité :** ces données décrivent la qualité attendue par l'utilisateur.

4. **Les données de livraison :** ces données concernent tout ce qui est lié à la présentation des résultats en fonction de la plateforme de l'utilisateur.

5. **Les données de sécurité :** La sécurité est une dimension fondamentale du profil. Elle peut concerner les données que l'on interroge ou modifie, les informations que l'on calcule, les requêtes utilisateurs ou les autres dimensions du profil.

2.3 Construction du profil utilisateur:

La construction du profil utilisateur repose sur deux phases. La première phase est la collecte d'informations sur l'utilisateur et la seconde, est la construction proprement dite du profil. Nous présentons dans cette section les techniques d'acquisition des données utilisateur

ainsi que les techniques de sa construction. Ensuite nous introduisons la notion de gestion de l'évolution du profile de l'utilisateur.

2.3.1 Acquisition des données utilisateurs :

La phase d'acquisition des données utilisateur consiste à collecter les informations qui serviront à l'instanciation du profil de l'utilisateur. Nous distinguons trois modes d'acquisition : acquisition explicite et acquisition implicite et leur hybridation.

Dans l'approche explicite, les informations sont directement obtenues de l'utilisateur.

Dans l'approche implicite, ce sont les données de comportement de l'utilisateur qui sont exploitées. Nous détaillons ces approches dans ce qui suit.

2.3.1.1 Acquisition explicite :

Dans l'acquisition explicite l'utilisateur est directement interrogé à travers des formulaires à remplir, par exemple. Ce dernier peut aussi être amené à émettre un jugement d'intérêt en fournissant une valeur sur une échelle graduée comme dans le système Ringo (upendra shardanand et al, 1995). L'acquisition explicite a été largement utilisée pour la personnalisation des interfaces des sites web en fonction des préférences des utilisateurs. 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, documents ...) dans deux classes « Intéressants » ou « Inintéressants ».

- ✓ **Le vote:** Dans les systèmes (Pazzani et al, M Pazzani, J Muramatsu, D Billsus Syskill et Webert) ; (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.

Ces limitations ont mené vers l'utilisation des techniques d'acquisition implicite des données utilisateur pour la construction du profil utilisateur.

2.3.1.2 Acquisition implicite :

Dans cette technique de collecte de données, l'utilisateur n'est pas directement impliqué. En effet, les données utilisées sont issues de l'observation des comportements et des interactions de l'utilisateur avec le système lors de ses recherches.

Les activités que peut avoir un utilisateur peuvent correspondre à l'utilisation de moteurs de recherche à travers la soumission de requêtes et la sélection des items, la navigation sur le web ou encore l'utilisation d'applications dans le contexte de sa recherche (applications de bureautique, éditeurs de texte, ...).

Cette approche présente le principal avantage de ne pas impliquer l'utilisateur et de ne pas lui imposer l'émission de jugements, ni un effort d'attention particulier lors de sa recherche. Néanmoins, la difficulté dans cette technique est de définir un processus d'interprétation du comportement observé dans un contexte d'application spécifique (Daoud, 2009).

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.

2.3.1.3 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.

2.3.2 Le Prétraitement des données :

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.

2.4 Les techniques de construction du profil utilisateur:

Les profils utilisateurs sont construits en utilisant les sources d'information collectées. Nous décrivons dans cette section les techniques les plus courantes utilisées dans la construction de ces profils.

2.4.1 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.

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 item est jugé pertinent par l'utilisateur, le système construit le vecteur tf-idf de ce dernier, après cette étape on obtient un ensemble de N + 1 vecteurs pondérés (N vecteurs profils et 1 nouveau vecteur item). Puis le système calcule la similarité cosinus entre chaque paire de vecteurs dans cet ensemble et combine les deux vecteurs les plus similaires.

2.4.2 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.

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).

2.4.3 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).

2.5 Adaptation et Evolution du profil utilisateur :

Une des caractéristiques des systèmes de recommandation est la gestion de l'évolution du profil de l'utilisateur. Cette gestion consiste à capturer les transformations des centres d'intérêt de l'utilisateur et à propager ces changements au niveau de la représentation du profil de l'utilisateur.

L'évolution du profil peut s'effectuer lors de sa modélisation par deux modèles, l'un à court terme et l'autre à long terme (Billsus and Pazzani, 1999) ou encore en appliquant des algorithmes d'évolution génétique.

L'adaptation du profil de l'utilisateur implique d'avoir des changements dans les centres d'intérêts de l'utilisateur ce qui peut comprendre aussi la suppression ou l'ajout de domaines.

En effet, faire évoluer le contenu d'un profil utilisateur consiste à adapter aussi bien la structure que le contenu aux évolutions des centres d'intérêts de l'utilisateur.

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 qu'Amazon (linden et al, 2003), Movie Lens (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.

2.6 Conclusion :

Dans ce chapitre, nous avons abordé la notion de profil utilisateur, les modèles de sa représentation, les méthodes d'acquisition des informations des utilisateurs, les techniques de construction et de mise à jour de ce profil dans les systèmes de recommandation.

Dans le chapitre qui suit nous allons spécifier un modèle de profil utilisateur multidimensionnel pour un système de recommandation.

Chapitre III :

Spécification d'un profil utilisateur multidimensionnel pour la recommandation

3.1 Introduction:

L'utilisateur joue un rôle principal dans le processus de recommandation pour cela une modélisation de son profil est nécessaire.

La recommandation basée contenu consiste à analyser le contenu des items candidats à la recommandation, en s'appuiant sur des évaluations effectuées par l'utilisateur.

Notre travail consiste à définir un profil utilisateur multidimensionnel pour un système de recommandation basé contenu. Pour ce faire nous avons exploité le jeu de données MovieLens portant sur les informations des utilisateurs et leurs évaluations sur un ensemble de films.

La construction du profil est donc basée sur les données recueillies.

Dans ce chapitre nous allons détailler notre approche de modélisation du profil utilisateur basée sur la représentation multidimensionnelle.

3.2 Problématique et motivation:

Le problème de la recommandation peut être formulé de la manière suivante : comment guider l'utilisateur dans son exploitation de données afin de trouver les informations pertinentes ? L'utilisateur doit donc être bien placé au centre du système de recommandation. Notre objectif est alors de définir un profil utilisateur performant en se basant sur l'approche de la représentation multidimensionnelle pour l'utiliser dans le processus de la recommandation.

3.3 Architecture générale de l'approche de recommandation:

Les principaux composants de notre approche de recommandation sont illustrés dans (la figure 3.1). Ils intègrent la définition du profil utilisateur, la définition des profils items et l'intégration de ces profils dans le processus de recommandation.

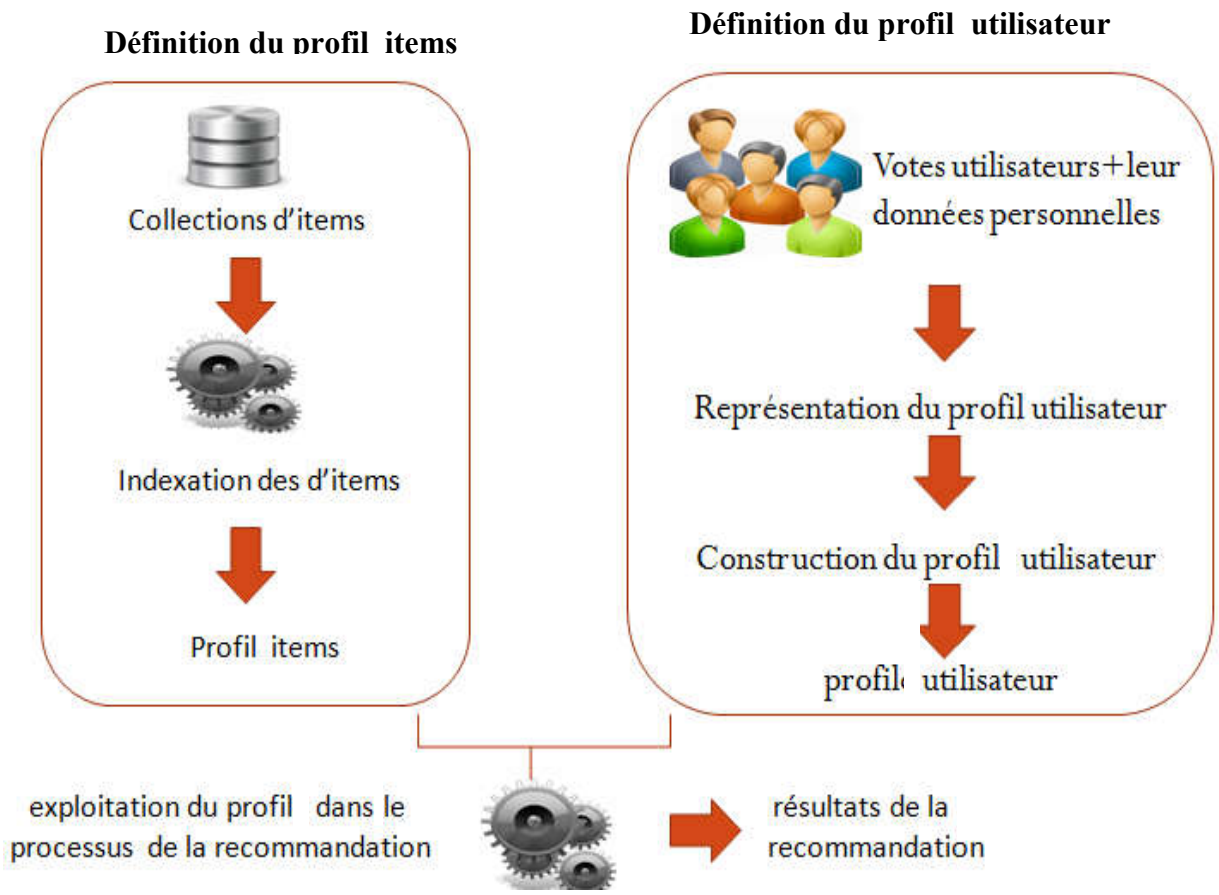

Figure 3.1 : Architecture générale de l'approche de recommandation

3.4 Description de l'approche proposée :

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

3.4.1 Définition du profil utilisateur :

La définition du profil utilisateur consiste en sa représentation et sa construction.

3.4.1.1 Représentation du profil utilisateur:

Pour la représentation du profil utilisateur nous allons utiliser une représentation basée sur l'approche multidimensionnelle qui palie aux limites du modèle vectoriel. Dans notre approche le profil utilisateur est définit selon trois dimensions :

1. la dimension centres d'intérêts,
2. la dimension âge,
3. la dimension genre.

Où chaque dimension sera représentée sous forme vectorielle.

- ✓ Pour construire la dimension centres d'intérêts, on va analyser un ensemble des items précédemment notés par l'utilisateur, et construire le vecteur des intérêts de cet utilisateur sur la base des caractéristiques des items aimés ou détestés par celui-ci.

- ✓ La dimension genre prend en considération le sexe de l'utilisateur.
  Si l'utilisateur est de genre homme (ou femme) on prend tout les items évaluées par l'ensemble des hommes (des femmes) puis construire le vecteur genre sur la base des caractéristiques des items aimés ou détestés par ces utilisateurs hommes (femmes). Ce vecteur exprime les préférences des utilisateurs hommes (femmes).

- ✓ Pour construire la dimension âge on procède comme suit :

  - ➢ Diviser l'ensemble des utilisateurs de système en sous-ensembles selon leurs âge, par exemple les utilisateurs ayant un âge comprit entre 18 et 24 appartiennent à le même sous-ensemble.

  - ➢ Déterminer le sous-ensemble à lequel l'utilisateur appartient, puis analyser les items notés par ce sous-ensemble.

  - ➢ Sur la base des caractéristiques des items aimés ou pas aimés par ces utilisateurs on construit le vecteur âge qui représente les préférences de ce sous-ensemble.

.

3.4.1.2 Construction du profil utilisateur:

Le profil utilisateur U suivant sa représentation multidimensionnelle, comprend trois vecteurs :

Le premier correspond aux centres d'intérêts de l'utilisateur, les deux autres vecteurs correspondent à la catégorie d'âge et le genre de l'utilisateur.

Le profil U est illustré comme suit:

$$\boldsymbol{U = (D_{CI}, D_{age}, D_{genre})} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (\mathbf{3.1})$$

Avec :

- D_{CI} : est la dimension centres d'intérêts, représentée sous forme de vecteur des poids de termes, où chaque terme correspond à une caractéristique d'item, cette dimension est représentée sous la forme suivante:

$$\boldsymbol{D_{CI} = (wc_{1k}, wc_{2k}, \dots, wc_{ik})} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (\mathbf{3.2})$$

wc_{ik}: représente le poids de la catégorie item Ca_i liée aux préférences de l'utilisateur u extrait en prenant en considération ses notes sur les items qui a déjà consulté. Il est calculé comme suit:

$$\boldsymbol{wc_{ik} = \sum_{Ij \in I} w_{ij} * n_j} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (\mathbf{3.3})$$

Où:

n_j: représente la note attribuée par l'utilisateur u pour l'item I_j

w_{ij}: Représente le poids de la catégorie item ca_i dans l'item I_j , calculé comme suit :

$$w_{ij} = \frac{occ_{i,j}}{Nc_j} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.4)$$

Avec :

$occ_{i,j}$: Occurrence de la catégorie ca_i dans l'item I_j.

Nc_j: Le nombre de catégorie dans l'item I_j.

- D_{age} : Est la dimension âge. L'idée est de discrétiser l'âge des utilisateurs du système en intervalles. Chaque intervalle représente une tranche d'âge.
  Dans notre approche nous avant décomposé l'âge en sept tranches comme suit :

  1. La « tranche 1 » contient les utilisateurs ayant un âge moins de 18 ans.
  2. La « tranche 2 » contient les utilisateurs ayant un âge compris entre 18 ans et 24 ans.
  3. La « tranche 3 » contient les utilisateurs ayant un âge compris entre 25 ans et 34 ans.
  4. La « tranche 4 » contient les utilisateurs ayant un âge compris entre 35 ans et 44 ans.
  5. La « tranche 5 » contient les utilisateurs ayant un âge compris entre 45 ans et 49 ans.
  6. La « tranche 6 » contient les utilisateurs ayant un âge compris entre 50 ans et 55 ans.
  7. La « tranche 7 » contient les utilisateurs ayant un âge supérieur à 56 ans.

Puis pour chacune de ces tranches d'âge, on regroupe tous les utilisateurs de la même tranche d'âge. Les poids d'une catégorie item d'une tranche d'âge sont calculées en utilisant les notes attribuées pour les items par les utilisateurs appartenant à cette tranche d'âge.

Le vecteur de la dimension âge est illustré sous cette forme :

$$\boldsymbol{D_{age}} = (\boldsymbol{wc_{1g}}, \boldsymbol{wc_{2g}}, \dots, \boldsymbol{wc_{ig}}) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (\mathbf{3.5})$$

wc_{ig}: représente le poids de la catégorie d'item Ca_i liée aux préférences des utilisateurs de la même tranche d'âge. Il est extrait en prenant l'ensemble des notes effectuées par ces utilisateurs sur les items. Il est calculé comme suit :

$$\boldsymbol{wc_{ig}} = \sum_{\boldsymbol{Ij \in I}} \boldsymbol{w_{ij}} * \boldsymbol{n_j} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (\mathbf{3.6})$$

Où:

n_j représente la note moyenne attribuée par les utilisateurs d'une même catégorie d'âge pour l'item I_j.

w_{ij}: représente le poids de la catégorie item ca_i dans l'item I_j.

- D_{genre} : est la dimension genre. Pour cette dimension on va créer deux groupes d'utilisateurs selon leur genre: les hommes et les femmes puis on procède exactement comme avec la dimension âge.

  On va extraire les préférences des hommes et celles des femmes on prenant l'ensemble des notes attribuées par les utilisateurs de même groupe pour obtenir les préférences des hommes ainsi que les préférences des femmes. Le vecteur genre est représenté comme suit :

$$\boldsymbol{D_{genre}} = (\boldsymbol{wc_{1y}}, \boldsymbol{wc_{2y}}, \dots, \boldsymbol{wc_{iy}}) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (\mathbf{3.7})$$

Avec :

wc_{iy}: représente le poids de la catégorie item Ca_i liée aux préférences des utilisateurs regroupés selon leur genre. Il est extrait en prenant les notes effectuées par ces utilisateurs sur les items. Il est calculé comme suit :

$$\boldsymbol{wc_{iy} = \sum_{Ij \in I} w_{ij} * n_j} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.8)$$

Où:

n_j: représente la note moyenne attribuée par les utilisateurs de même genre pour l'item I_j.

w_{ij}: représente le poids de la catégorie item ca_i dans l'item I_j.

3.4.2 Définition du profil item:

On a utilisé la représentation binaire 0/1. Si une catégorie ca_i appartienne à un item alors c'est 1 sinon c'est 0.

Chaque item I_j est représenté sous forme d'un vecteur de poids des catégories ca_i

$$\boldsymbol{I_J = (p_{1j}, p_{2j}, \ldots, p_{ij})} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.9)$$

p_{ij} : représente le poids d'une catégorie item dans l'item I_j . Il est calculé comme suit :

$$\boldsymbol{p_{ij} = tf_{ij} * \ log \frac{N}{n_i}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.9)$$

Avec :

.tf_{ij}: représente la fréquence d'une catégorie ca_i dans un item I_j.

N : est le nombre total des items.

n_i: est le nombre d'items qui contient la catégorie ca_i.

3.4.3 Exploitation et intégration du profil utilisateur dans le processus de recommandation:

Après avoir définit le profil utilisateur, il sera inclut dans le module de la recommandation. Pour qu'un item soit recommandé il doit être pertinent pour toutes les dimensions (centre d'intérêts, âge, genre).

La pertinence item-dimension est calculée entre les vecteurs profils utilisateurs et les vecteurs pondérés des items. Ceci est dénoté par la formule suivante:

$$\textbf{Similarité cosinus} = \frac{\sum_{Ij \in D} p_{ij} * w_{ik}}{||p_{ij}|| * ||w_{ik}||} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.11)$$

Où:

D : représente l'ensemble des items.

p_{ij}: représente le poids d'une catégorie item dans un item I_J.

w_{ik}: représente le poids d'une catégorie d'item dans un vecteur d'une dimension (centre d'intérêts, âge, genre)

3.4.4 Illustration de l'approche proposée:

Nous avons illustré notre approche sur une collection de 10 films.

F= {f1, f2, f3, f4, f5, f6, f7, f8, f9, f10}

L'index de la collection inclut 6 catégories (Action, Adventure, Animation, comedy, romance, thriller).

Les films sont représentés sous forme binaire, la valeur d'un film est de 1 si le film est de la catégorie i sinon elle est de 0.

Le tableau suivant montre la représentation binaire des films :

	Action	Adventure	Animation	comedy	romance	thriller
f1	1	1	0	0	1	0
f2	1	1	0	0	0	0
f3	0	1	0	1	0	0
f4	1	1	1	0	0	0
f5	0	0	1	1	0	0
f6	0	0	0	0	0	1
f7	0	0	0	0	0	1
f8	1	0	0	0	0	1
f9	0	0	1	0	1	0
f10	0	1	0	0	1	0

Tableau 3.1: représentation binaire des films

Pour illustré les notes attribuées par les utilisateurs aux films, nous avons utilisé une représentation 1 et -1 pour représenter les notes attribuées. Par exemple l'utilisateur 1 a aimé le film 1, pour cela la valeur 1 est attribuée et il n'a pas aimé le film 9, pour cela la valeur -1 est attribuée. Il n'a pas regardé les films f 3, f4, f5, f6, f7.

Les notes des utilisateurs sur les films sont illustrées dans le tableau (3.2) :

	f1	***f2***	***f3***	***f4***	***f5***	***f6***	***f7***	***f8***	***f9***	***f10***
U1	1	1						1	-1	-1
U2	1					-1	-1		1	1
U3		1					1	1		
U4			1		1	-1				
U5		1							1	1
U6					-1	1		1		

Tableau 3.2 : les notes des utilisateurs sur les films

3.4.4.1 Définition des profils films :

Le tableau suivant représente les vecteurs pondérés des films, calculés selon la formule (3.9):

	Action	Adventure	Animation	Comedy	Romance	Thriller
f1	0.132646	0.100343	0	0	0.174290	0
f2	0.198970	0.150514	0	0	0	0
f3	0	0.150514	0	0.349480	0	0
f4	0.132646	0.100343	0.174290	0	0	0
f5	0	0	0.261439	0.349480	0	0
f6	0	0	0	0	0	0.5228787
f7	0	0	0	0	0	0.5228787
f8	0.198970	0	0	0	0	0.261439
f9	0	0	0.261439	0	0.261439	0
f10	0	0.150514	0	0	0.261439	0

Tableau 3.3 : Les vecteurs pondérés des films

3.4.4.2 Définition du profil utilisateurs

1. Dimension centre d'intérêt :

En se basant sur la formule (3.2), les vecteurs centres d'intérêts sont représentés dans le tableau suivant :

	Action	Adventure	Animation	Comedy	Romance	Thriller
U1	1	0.497487	0.080402	0.331650	0	0.582914
U2	0.699699	0.849849	0.750750	0.600600	1	0
U3	0666666	0.333333	0	0	0	1
U4	0.5	0.75	0.75	1	0.5	0
U5	0.5	1	0.5	0	1	0
U6	0.5	0.25	0	0	0.25	1

Tableau 3.4 : Les vecteurs centres d'intérêts

Le degré de pertinence de chaque film est mesuré en se basant sur la formule (3.11).

Le tableau (3.5) représente les degrés de pertinences des films pour les utilisateurs

	U1	*U2*	*U3*	*U4*	*U5*	*U6*
f1	0.580565	0.825932	0.405618	0.585981	0.895082	0.477800
f2	0.840937	0.604619	0.587529	0.525391	0.645495	0.468687
f3	0.384123	0.501309	0.105716	0.749982	0.250171	0.084332
f4	0.625128	0.724110	0.405618	0.697611	0.666307	0.323572
f 5	0.240338	0.525508	0	0.771517	0.189425	0
f6	0.446583	0	0.801784	0	0	0.852803
f7	0.446583	0	0.801784	0	0	0.852803
f8	0.819349	0.239281	0.961741	0.186897	0.191512	0.936863
f9	0.043556	0.699053	0	0.545545	0.670821	0.150756
f10	0.190162	0.734330	0.133346	0.498400	0.863665	0.291141

Tableau 3.5: degré de pertinence dimension centres d'intérêts-films

Les résultats obtenus montrent que le degré de pertinence de film 8 est de 0.962. C'est le film que l'utilisateur 3 apprécie le plus .Ce concept est appliqué aux films 'n' et nous pouvons déterminer quel film un utilisateur appréciera le plus.

2 Dimension genre:

Le tableau ci-dessous contient les données personnelles (âge et genre) des six utilisateurs déjà présentés.

	Age	**Genre**
U1	32	Homme
U2	18	Femme
U3	45	Homme
U4	20	Homme
U5	25	Femme
U6	36	Homme

Tableau 3.6: données personnelles des utilisateurs

Le tableau (3.7) montre les résultats obtenus en effectuent une moyenne de tous les votes des utilisateurs de même genre.

Les utilisateurs *U1, U3, U4, U6* sont de genre homme et les utilisateurs *U2* et *U5* sont de genre femme.

	f1	*f2*	*f3*	*f4*	*f5*	*f6*	*f7*	*f8*	*f9*	*f10*
Votes homme	1	1	1		-1	-1	1	1	-1	-1
Votes femme	1	1				-1	-1		1	1

Tableau 3.7: notes des utilisateurs sur les films selon leur genre

On se basant sur la formule (3.7) on obtient les vecteurs de la dimension genre illustrés dans le tableau suivant :

	Action	**Adventure**	**Animation**	**Comedy**	**Romance**	**Thriller**
Vecteur CI homme	1	0.785407	0	0.429184	0.145922	0.858369
Vecteur CI femme	0.849849	1	0.750750	0.600600	1	0

Tableau 3.8: Les vecteurs genres des utilisateurs

De la même manière le degré de pertinence de chaque film est mesuré selon la formule (3.11) entre les vecteurs pondérés des films et le vecteur genre (homme /femme) de chaque utilisateur.

	Homme	**Femme**
f1	0.593770	0.841992
f2	0.767583	0.670855
f3	0.425557	0.493912
f4	0.529924	0.747564
f5	0.207488	0.487344
f6	0.518243	0
f7	0.518243	0
f8	0.778038	0.269523
f9	0.062296	0.648285
f10	0.312943	0.715189

Tableau 3.9: degré de pertinence dimension genre-films

3. Dimension âge :

D'une manière similaire avec la dimension genre ; on catégorise l'âge, mais vu que l'âge a des valeurs de cette dimension sont de type quantitatif discret alors on va les discrétiser en intervalles. Chaque intervalle représente une tranche d'âge.

Dans notre exemple on considère deux tranches :

Les moins et les plus de 30 ans (Catégorie 1 < 30 et catégorie 2>=30). Donc les utilisateurs U2, U4 et U5 appartient à la catégorie 1 et les utilisateurs U1, U3 et U6 appartient à la catégorie 2.

Le tableau (3.10) montre les résultats obtenus en effectuent une moyenne de tous les votes des utilisateurs d'une même tranche d'âge.

	***f*1**	***f*2**	***f*3**	***f*4**	***f*5**	***f*6**	***f*7**	***f*8**	***f*9**	***f*10**
Catégorie 1	1	1	1		1	-1	-1		1	1
Catégorie 2	1	1			-1	1	1	1	-1	-1

Tableau 3.10: notes des utilisateurs sur les films selon leur âge

On se basant sur la formule (3.5) on obtient les vecteurs de la dimension âge illustrés dans le tableau suivant :

	Action	**Adventure**	**Animation**	**Comedy**	**Romance**	**Thriller**
Vecteur âge catégorie 1	0.738903	1	0.783289	0.783289	0.869451	0
Vecteur âge catégorie 2	0.665714	0.38	0	0.142857	0.097142	1

Tableau 3.11 : Les vecteurs profils des catégories d'âges des utilisateurs

Donc, le degré de pertinence de chaque film est mesuré selon la formule (3.11) entre les vecteurs pondérés des films et le vecteur âge de chaque utilisateur. Les résultats obtenus sont illustrés dans le tableau (3.12):

Films	Catégorie 1	Catégorie 2
f1	0.773112	0.467915
f2	0.634839	0.597724
f3	0.644317	0.221668
f4	0.739930	0.412357
f5	0.583644	0.089945
f6	0	0.786300
f7	0	0.786300
f8	0.238209	0.942715
f9	0.622105	0.054010
f10	0.666698	0.215275

Tableau 3.12: degré de pertinence dimension âge-films

En se basant sur les degrés de pertinences calculés précédemment, on obtient les films les plus pertinents pour un utilisateur à partir de son centre d'intérêt et son genre (homme/femme) ainsi que son âge. Cela est obtenu en faisant une intersection entre les films pertinents pour le genre d'un utilisateur (homme/femme) et ceux pertinents pour sa catégorie d'âge et son centre d'intérêts.

Les films obtenus sont recommandés, si ces derniers ne sont pas vus par l'utilisateur.

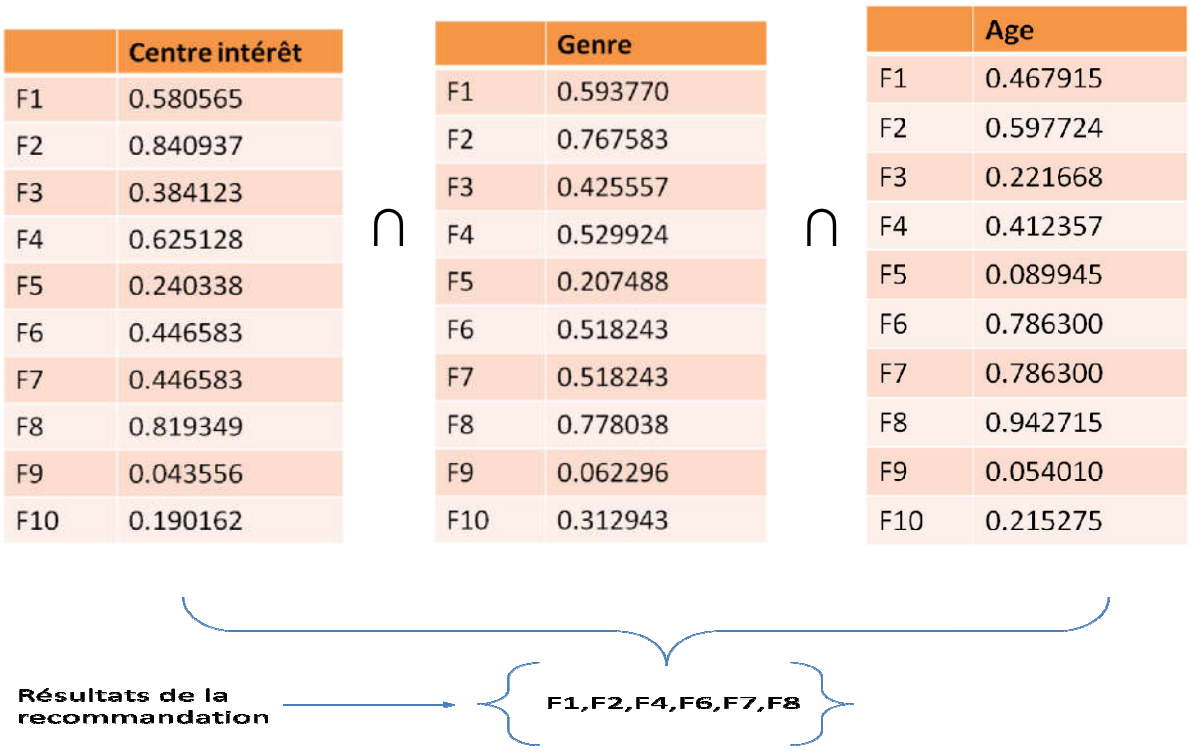

Figure 3.2 : liste des recommandations pour l'utilisateur 1

3.5 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 suivant l'évaluation et la validation expérimentale de l'approche proposée, en utilisant une collection de test.

Chapitre IV :

Evaluation et validation expérimental du modèle proposé

4.1 Introduction :

Dans ce chapitre nous allons décrire l'environnement de développement, ainsi que la collection de test qu'on a utilisé pour implémenter notre approche. En terminant avec une évaluation des résultats obtenus.

4.2 Environnement de développement:

✓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 est jupyter notbook** :

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 Description détaillée de la collection de test :

Dans notre exemple nous allons utiliser un jeu de donnée *MovieLens,* collecté par les membres du projet de recherche *GroupeLens* au Département d'informatique et de génie de l'Université de *Minnesot.*

La base de données *MovieLens* contient plusieurs fichiers pour représenter les différents ensembles de données et pour décrire les films, les utilisateurs, ainsi que la relation entre utilisateurs et items.

Nous présentons dans cette section une description détaillée des différents fichiers, ceux disponibles avec la base et ceux que nous avons créés et ajouté afin d'accomplir les expérimentations.

4.3.1 Les fichiers disponibles avec la base :

4486_6865_bundle_archive.zip : Fichier compressé de type .zip qui contient toute la base de données
téléchargée, contient trois tables (utilisateurs, évaluations, films):1000209 évaluations faites par 6040 utilisateurs pour 3900 films.

movies.dat: Ce fichier contient les informations sur les items (films), représenté sous le format suivant:

Movie_id :: Titre :: Genres

Les titres sont identiques aux titres fournis par l'IMDB (y compris année de sortie) - Les genres sont séparés par des tuyaux et sont sélectionnés parmi les genres suivants:

Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |

Comme exemple nous citons la description du premier film :

1:: Toy Story (1995) :: Animation | Children's | Comedy .

users.dat : Ce fichier contient les données démographiques sur les utilisateurs, sous le format suivant :

UserID | Gender | Age | Occupation | Code

Le genre est indiqué par « M » pour les hommes et « F » pour les femmes. L'âge est choisi parmi les plages suivantes :

1: «sous 18 ans »**, 18:** « 18-24 »**, 25:** « 25-34 »**, 35:** « 35-44 »**, 45:** « 45-49 »**, 50:** « 50-55 »**, 56:** « 56+ »

L'occupation est choisie parmi 20 occupations.

A titre d'exemple, l'utilisateur ayant l'identifiant 1, qui est un homme âgé de 24 ans et qui travaille comme technicien avec un code postal 85711, est représenté dans la base comme suit :

1 | 24 | M | technicien | 85711

ratings.dat : Ce fichier contient les évaluations des utilisateurs sur certains films (chaque utilisateur a évaluer au moins 20 films). Les enregistrements sons sous le format suivant :

UserID :: Movie_id :: Rating :: Timestamp.

- Les userID vont de 1 à 6040 (les mêmes identifiant utilisés dans le fichier user.dat).
- Les Movie_id compris entre 1 et 3952(les identifiant des films utilisés dans le fichier movie.dat.
- Les notes sont faites sur une échelle de 5 étoiles (notes par étoiles uniquement).
- L'horodatage est représenté en secondes depuis l'époque comme renvoyé par time (2).
- Chaque utilisateur a au moins 20 évaluations.

Un exemple de cet ensemble de données soit la ligne suivante:

1 :: 1 :: 5 :: 978824268 : Elle indique que l'utilisateur 1 a évalué le film 1 avec une note (vote) égale à 5 au moment 978824268s.

4.3.2 Les fichiers ajoutés :

Dans cette section, nous présentons une description des fichiers que nous avons créés et ajoutés à la base de données *MovieLens* pour être utilisés dans les expérimentations.

Movies: Cet ensemble contient tous les items. Nous avons pris les 18 colonnes correspondantes aux genres des items pour représenter les films une représentation binaire comme suit :

	Action	adventure	……	Western
Movie_id				

Donc, la liste des caractéristiques utilisées pour représenter le film devient comme suit: ***1 | 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 :*** le film ayant l'identifiant 1 est de genre animation, children's et comedy.

Df : L'ensemble de données *df* est l'ensemble plein, c.à.d. l'ensemble qui contient les 1000209 évaluations des 6040 utilisateurs sur les 3883 films. Cet ensemble, est une

jointure des trois tables déjà existées dans le dataset téléchargé, mais avec quelque modification sur les deux colonnes âge et rate :

L'âge sera représenté par un numéro entre 1 et 7 (**1:** «sous 18 ans »**, 2:** « 18-24 »**, 3:** « 25-34 »**, 4:** « 35-44 »**, 5:** « 45-49 »**, 6:** « 50-55 »**, 7:** « 56+ »).

Les votes seront notés par 1(a aimé) si la note attribuée par l'utilisateur pour un film est supérieur ou égale à 3, sinon la note sera -1 (n'a pas aimé).

Les colonnes *occupation*, *code* et *times* sont supprimées car ils ne seront pas inclut dans le processus de la recommandation.

L'ensemble de données *df* est représentée avec une tab séparée comme suit :

Movie_id | *userID* | *age* | *gender* | *title* | *movie_genre* | *rate*

Un exemple de cet ensemble de données est le suivant :

1* :: *1* :: *2* :: *M* :: *Toy Story (1995)* :: Animation | Children's | Comedy :: *1

Où cette ligne indique que l'utilisateur 1 (de genre homme, il a un âge compris entre 18 et 24) a évalué le film 1(le titre : toy story (1995) de genre : Animation, Children's et Comedy) avec une note 1 (alors on peut dire qu'il a aimé ce film).

Df_train : cet ensemble contient 60% des données de la table DF. Ces données seront exploitées dans la phase d'entrainement.

Df_test : cet ensemble contient 40% des données de la table DF. Ces données seront exploitées dans la phase de test et d'évaluation.

4.4 Implémentation de notre approche :

4.4.1 Sélection de données:

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

- Détecté les valeurs manquantes afin de les supprimer.
- Déterminé les valeurs inconnus et nulles afin de les supprimer.
- Supprimé les utilisateurs nuls et les livres nuls.

4.4.2 Construction du profil film :

Afin de construire les vecteurs pondérés des films, nous exploitons le fichier **movie.dat** qui contient tous les films de la collection. **movie.dat,** selon la forme suivante :

	movie_id	title	movie_genre
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
3878	3948	Meet the Parents (2000)	Comedy
3879	3949	Requiem for a Dream (2000)	Drama
3880	3950	Tigerland (2000)	Drama
3881	3951	Two Family House (2000)	Drama
3882	3952	Contender, The (2000)	Drama|Thriller

Figure (4.1) : Liste des films contenant dans le fichier « movie.dat »

Les vecteurs pondérés de ces films seront représentés dans un ensemble des catégories (les valeurs de la colonne « movie_genre »). Pour exploiter les catégories on transforme la table initiale **movie.dat** en une table binaire qui représente les films selon leurs catégories.

La figure suivante contient le code utilisé pour obtenir la représentation binaire des films :

```
#reptésentation binair de 'movies_copy'
movies_copy=pd.DataFrame(movies_copy.loc[:,['movie_id','movie_genre']])
movies_copy = pd.concat([movies_copy, movies_copy.movie_genre.str.get_dummies(sep='|')], axis=1)
movies_copy =movies_copy.drop(['movie_genre'],axis=1)
movies_copy=pd.DataFrame(movies_copy.set_index('movie_id'))
```

Figure (4.2) : Le code effectué pour obtenir la représentation binaire des films

La figure (4.3) représente le résultat d'exécution :

movie_id	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	My
1	0	0	1	1	1	0	0	0	0	0	0	0	
2	0	1	0	1	0	0	0	0	1	0	0	0	
3	0	0	0	0	1	0	0	0	0	0	0	0	
4	0	0	0	0	1	0	0	1	0	0	0	0	
5	0	0	0	0	1	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
3948	0	0	0	0	1	0	0	0	0	0	0	0	
3949	0	0	0	0	0	0	0	1	0	0	0	0	
3950	0	0	0	0	0	0	0	1	0	0	0	0	
3951	0	0	0	0	0	0	0	1	0	0	0	0	
3952	0	0	0	0	0	0	0	1	0	0	0	0	

Figure (4.3) : la représentation binaire des films

On normalise les vecteurs films avant d'effectuer la pondération tf_idf. La figure suivante contient le code utilisé dans la normalisation des vecteurs films :

```
#normalisation de 'movies_copy'
movies_copy_norm=movies_copy/movies_copy.sum(axis=1)[:,None]
```

Figure (4.4) : le code de la normalisation des vecteurs films

La figure (4.5) représente les vecteurs films normalisés :

movie_id	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	M
1	0.0	0.000000	0.333333	0.333333	0.333333	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
2	0.0	0.333333	0.000000	0.333333	0.000000	0.0	0.0	0.0	0.333333	0.0	0.0	0.0	
3	0.0	0.000000	0.000000	0.000000	0.500000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
4	0.0	0.000000	0.000000	0.000000	0.500000	0.0	0.0	0.5	0.000000	0.0	0.0	0.0	
5	0.0	0.000000	0.000000	0.000000	1.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
3948	0.0	0.000000	0.000000	0.000000	1.000000	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	
3949	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.0	1.0	0.000000	0.0	0.0	0.0	
3950	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.0	1.0	0.000000	0.0	0.0	0.0	
3951	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.0	1.0	0.000000	0.0	0.0	0.0	
3952	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.5	0.000000	0.0	0.0	0.0	

Figure (4.5) : les vecteurs films normalisés

La figure (4.6) contient le code de la fonction de pondération TF_IDF des vecteurs films :

```
#ponderation tfidf
n=movies_copy_norm.sum(axis=0)
df_df=pd.Series(n,index=movies_copy_norm.columns)

def idf(i):
    k=np.log(movies_copy_norm.size/i)
    return k

df_idf=pd.Series(idf(df_df[0:18]))
df_movies_penderation=df_idf[:]*movies_copy_norm
```

Figure (4.6) : la fonction de pondération des films

La figure (4.7) représente les vecteurs pondérés des films :

movie_id	Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical
1	0.0	0.000000	1.0	0.875745	0.598468	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
2	0.0	0.799782	0.0	0.811353	0.000000	0.0	0.0	0.000000	1.0	0.0	0.0	0.0
3	0.0	0.000000	0.0	0.000000	0.776976	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
4	0.0	0.000000	0.0	0.000000	1.000000	0.0	0.0	0.918101	0.0	0.0	0.0	0.0
5	0.0	0.000000	0.0	0.000000	1.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
3948	0.0	0.000000	0.0	0.000000	1.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
3949	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	1.000000	0.0	0.0	0.0	0.0
3950	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	1.000000	0.0	0.0	0.0	0.0
3951	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	1.000000	0.0	0.0	0.0	0.0
3952	0.0	0.000000	0.0	0.000000	0.000000	0.0	0.0	0.730530	0.0	0.0	0.0	0.0

Figure (4.7) : les vecteurs pondérés des films

4.4.3 Construction du profil utilisateur :

Avant d'entamer la construction des profils utilisateurs, on doit d'abord crée la table **DF** qui est une jointure des trois tables de base (user.dat, movie.dat et rating.dat).

Sachant que l'attribut âge est discrétiser en intervalle comme c'était expliquer précédemment. L'attribut rate qui contient les notes attribuées par les utilisateurs sur les films, est représenter par 1 si sa valeur est supérieur ou égale à 3, -1 sinon.

La figure suivante représente les traitements effectués :

```
#descretiser l'age
df_users['age']=pd.cut(df_users['age'],bins=[1,18,25,35,45,50,56,np.inf],labels=[1,2,3,4,5,6,7],right=False)
# aimé==1 n'a pas aimé==-1
df_ratings['rate'].replace([1, 2,3,4,5],[-1, -1, 1, 1, 1],inplace=True)
#jointure des trois tables == "df"
df=movies.merge(df_ratings,on='movie_id')
df=df_users.merge(df,on='userID')
df.set_index('movie_id')
```

Figure (4.8) : les traitements effectués sur le dataset

La figure (4.9) montre la table pleine « DF »

movie_id	userID	age	gender	title	movie_genre	rate
1	1	2	M	Toy Story (1995)	Animation|Children's|Comedy	1
48	1	2	M	Pocahontas (1995)	Animation|Children's|Musical|Romance	1
150	1	2	M	Apollo 13 (1995)	Drama	1
260	1	2	M	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Fantasy|Sci-Fi	1
527	1	2	M	Schindler's List (1993)	Drama|War	1
...	...	...	...	...	...	...
3720	943	2	M	Sunshine (1999)	Drama	1
3787	943	2	M	Shower (Xizhao) (1999)	Comedy	1
3822	943	2	M	Girl on the Bridge, The (La Fille sur le Pont)...	Drama|Romance	1
3844	943	2	M	Steel Magnolias (1989)	Drama	1
3952	943	2	M	Contender, The (2000)	Drama|Thriller	1

Figure (4.9) : la table pleine de dataset « df »

La Table **DF** ne sera pas exploitée entièrement dans l'entrainement. On extrait un sous ensemble qu'on appelle **DF_train** qui représente 60% de données inclut dans **DF**.

4.4.3.1 Construction de la dimension centres d'intérêts :

Le vecteur des centres d'intérêt pour un utilisateur est calculé en utilisant les vecteurs items normalisés et les évaluations faites par cet utilisateur. La figure suivante représente la fonction qui calcule le vecteur centre d'intérêt de l'utilisateur 1 :

```
def user_vect_rate(i) :
    categorie_poids=movies_copy_norm[movies_copy_norm.index.isin(user_table(i).movie_id)]
    df_user_rate=categorie_poids.T.mul(df['rate']).sum(axis=1)
    df_user_rate=pd.DataFrame(df_user_rate)
    return df_user_rate
```

Figure (4.10) : la fonction qui calcule le vecteur centre d'intérêt des utilisateurs

La figure (4.11) illustre le vecteur centres d'intérêts de l'utilisateur 1 :

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thrille
0.119403	0.059701	0.701493	0.716418	0.298507	0.059701	0.0	1.0	0.134328	0.0	0.0	0.522388	0.0	0.19403	0.044776	0.17910

Figure (4.11) : le vecteur centre d'intérêt de l'utilisateur 1

4.4.3.2 Construction de la dimension genre :

Cette dimension est représentée par un vecteur, selon le genre de l'utilisateur (homme / femme). Les deux méthodes qui calculent le vecteur homme et femme sont illustrées dans la figure (4.12):

```
def homme_vect():
    df_h=df_trainset[df_trainset['gender']=='M']
    categorie_poids=movies_copy_norm[movies_copy_norm.index.isin(df_h.movie_id)]
    df_homme_rate=categorie_poids.T.mul(df_h['rate']).sum(axis=1)
    df_homme_rate=pd.DataFrame(df_homme_rate)
    return df_homme_rate
```

```
def femme_vect():
    df_f=df_trainset[df_trainset['gender']=='F']
    categorie_poids=movies_copy_norm[movies_copy_norm.index.isin(df_f.movie_id)]
    df_femme_rate=categorie_poids.T.mul(df_f['rate']).sum(axis=1)
    df_femme_rate=pd.DataFrame(df_femme_rate)
    return df_femme_rate
```

Figure (4.12) : les méthodes qui génére le vecteur de la dimension genre

les vecteurs genre sont illustrés dans la figure (4.13) :

Vecteur de préférences des hommes

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi
0.226245	0.094126	0.015601	0.065462	0.715459	0.094797	0.043666	0.944017	0.000149	0.001866	0.209898	0.005598	0.027618	0.168321	0.091364

Vecteur de préférences des femmes

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi
0.217101	0.119203	0.026598	0.128255	0.596435	0.004735	0.021167	0.933714	0.005849	0.017686	0.214316	0.009191	0.0	0.160006	0.113355

Figure (4.13) : le vecteur de la dimension genre (homme / femme)

4.4.3.3 Construction de la dimension âge :

Cette dimension est représentée par un vecteur, selon la tranche d'âge de l'utilisateur. La figure (4.14) représente la méthode qui calcule le vecteur âge.

```
def age_vect(i):
    if i<=7:
        df_age=df_trainset[df_trainset['age']==i]
        categorie_poids=movies_copy_norm[movies_copy_norm.index.isin(df_age.movie_id)]
        df_age_rate=categorie_poids.T.mul(df_age['rate']).sum(axis=1)
        df_age_rate=pd.DataFrame(df_age_rate)
    return df_age_rate
```

Figure (4.14) : la méthode qui génére le vecteur la dimension âge

La figure (4.15) montre le vecteur résultat de l'utilisateur 1 qui appartient à la tranche d'âge numéro 2 (2 : « 18-24 »)

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thril
0.136364	0.069519	0.07754	0.09893	0.505348	0.010695	0.064171	1.0	0.040107	0.0	0.032086	0.005348	0.026738	0.21123	0.005348	0.267

Figure (4.15) : le vecteur âge de l'utilisateur 1

La figure suivante illustre le profil utilisateur multidimensionnel de l'utilisateur 1 qui est de genre homme avec un âge égal à 24 :

Dimension centres d'intérêts

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thrille
0.119403	0.059701	0.701493	0.716418	0.298507	0.059701	0.0	1.0	0.134328	0.0	0.0	0.522388	0.0	0.19403	0.044776	0.17910

Dimension genre

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi
0.226245	0.094126	0.015601	0.065462	0.715459	0.094797	0.043666	0.944017	0.000149	0.001866	0.209898	0.005598	0.027618	0.168321	0.091364

Dimension âge

Action	Adventure	Animation	Children's	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	Musical	Mystery	Romance	Sci-Fi	Thril
0.136364	0.069519	0.07754	0.09893	0.505348	0.010695	0.064171	1.0	0.040107	0.0	0.032086	0.005348	0.026738	0.21123	0.005348	0.267

Figure (4.16) : le profil utilisateur multidimensionnel de l'utilisateur 1

4.4.4 Recommandation pour un utilisateur selon son numéro :

Après avoir définit les trois dimensions du profil utilisateur on va les exploités en recommandation.

La figure (4.17) décrit l'algorithme de recommandation et l'ntegration de profil utilisateur multidimensionnel en recommandation :

```
# recommandation selon l'ID de l'utilisateur
def recommander(userID):
    #construire le vecteur de la dimension centres d'interets
    df_ratings_user=df_trainset[df_trainset['userID']==userID]
    categorie_poids=movies_copy_norm[movies_copy_norm.index.isin(df_ratings_user.movie_id)]
    df_user_rate=categorie_poids.T.mul(user_table(userID).rate).sum(axis=1)
    user_preference=df_user_rate.T
    user_preference=pd.DataFrame(user_preference)
    #calculer la similarité cosinus entre le vecteur centre d'interets et les profils films
    c=cosine_similarity(user_preference.T,df_movies_penderation)
    #ordonner les films selon leurs degré de pertinence
    R = np.argsort(c)[:, ::-1]
    #eliminer les films déjà vu par l'utilisateur
    interet = [i for i in R[0] if  i not in user_table(userID)['movie_id'].values]
    #retourner les titres des films pertinents pour la dimension centre d'interets
    centre_interet=movies['title'][interet]
    df_users_id=df_users[df_users['userID']==userID]
    #déterminer la tranche d'age de l'utilisateur
    j=df_users_id['age'].values
    #determiner son sexe
    if df_users_id['gender'].values=='M':
#l'intersection des films pertinents pour la dim age et ceux pertinents pour la dimension genre(homme)
```

```
        homme_age=age(j[0])[age(j[0]).isin(homme())]
#intersection des films pertinents pour la dim CI et les résultats obtenu dans l'instruction précédente
        recommandation=centre_interet[centre_interet.isin(homme_age)]
    else:
#l'intersection des films pertinents pour la dim age et ceux pertinents pour la dimension genre(femme)
        femme_age=age(j[0])[age(j[0]).isin(femme())]
#intersection des films pertinents pour la dim CI et les résultats obtenu dans l'instruction précédente
        recommandation=centre_interet[centre_interet.isin(femme_age)]
    return recommandation
```

Figure (4.17) : l'algorithme de recommandation

La figure (4.18) illustre les titres des films proposés recommandation pour l'utilisateur numéro 1 :

```
3882                          Contender, The (2000)
1274                                  M*A*S*H (1970)
1302    Amityville 1992: It's About Time (1992)
1301     American Werewolf in London, An (1981)
1300                                  Alien³ (1992)
                              ...
2583          Curse of Frankenstein, The (1957)
2582     Frankenstein Meets the Wolf Man (1943)
2581          Ghost of Frankenstein, The (1942)
2580                 Son of Frankenstein (1939)
0                                   Toy Story (1995)
```

Figure (4.18) : la liste des recommandations pour l'utilisateur numéro 1

4.5 Expérimentation : évaluation de la performance de l'approche proposée

Pour l'évaluation de notre approche, nous avons diviser notre ***dataset*** en ***trainset*** pour effectuer les traitements et un ***testset*** pour évaluer les résultats obtenus.

Le **testset** contient des items déjà évalués par l'utilisateur. Ces items sont les éléments pertinents attendus en résultats de la recommandation. On aura donc quatre possibilités, illustrées dans le tableau (4.1) :

	Recommandé	**Non recommandé**
Pertinent	Vrai_positif (tp)	Faux_négatif (fn)
Non pertinent	Faux_positif (fp)	Vrai_négatif (tn)

Tableau (4.1) : classement des possibilités de la recommandation d'un item pour un utilisateur

Afin d'évaluer la performance de notre système de recommandation, nous avons utilisé des métriques d'évaluation qui mesurent la capacité de l'algorithme de recommandation à prédire correctement les évaluations connues des utilisateurs.

Ces métriques utilisent des évaluations correctes d'un dataset de test et les comparent avec des prédictions proposées par l'algorithme de recommandation, en se basant sur un dataset d'apprentissage. La majorité de ces métriques proviennent du domaine de la recherche d'information.

.

4.5.1 Métriques d'évaluation :

Pour évaluer notre algorithme de recommandation nous avons utilisé le rappel et la précision. Ces deux métriques composent la formule de calcul d'une autre métrique qui est F1 **(Sarwar et al, 2000).**

- **Rappel :** le rappel mesure la proportion des items pertinents retrouvés parmi tous les items pertinents dans la base.

- **Précision :** la précision mesure la proportion des items pertinents retrouvés parmi tous les items retrouvés par le système.

- **F1 :** mesure la performance d'un système ou d'un algorithme qui correspond à la moyenne harmonique du taux de précision et du rappel.

Où, F1 est calculée comme suit :

$$\boldsymbol{F1 = \frac{2PR}{P + R}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{4.1})$$

P et *R* sont la *Précision* et le *Rappel* respectivement. Ils sont calculés comme suit :

$$\boldsymbol{P = \frac{tp}{tp + fp}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{4.2})$$

$$\boldsymbol{R = \frac{tp}{tp + fn}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{4.3})$$

Avec :

tp, ***fp*** et ***fn*** sont illustrés dans le tableau (4.1).

Les valeurs de *R, P* et *F1* sont comprises entre 0 et 1.

Les figures suivantes montrent les valeurs des deux métriques : *rappel* et *précision*, calculées pour un échantillon de vingt utilisateurs dans notre système de recommandation.

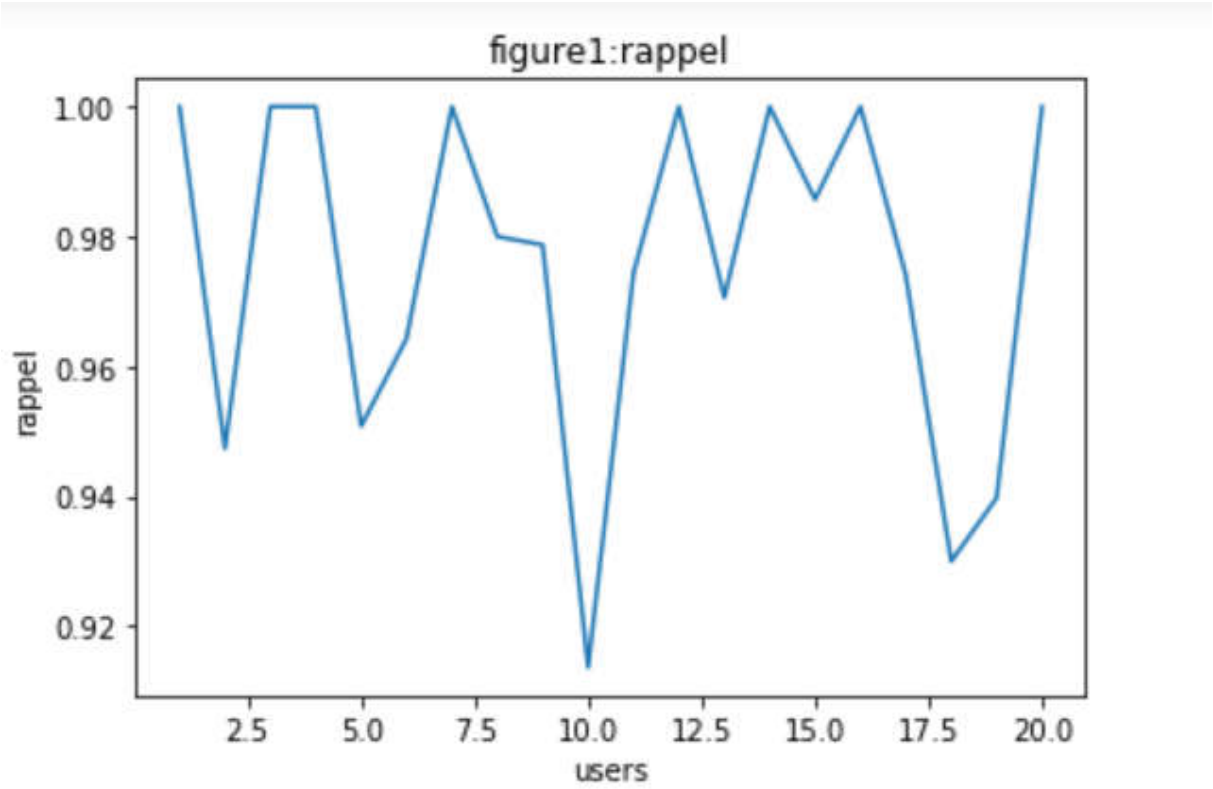

Figure (4.19) : le rappel pour les 20 premiers utilisateurs

La figure (4.19) montre le rappel calculé pour les vingt premiers utilisateurs. Les valeurs retrouvées sont variés entre 0.92 et 1, on aura donc un rappel moyen égale à 0.97. Cela signifié que le système retourne presque la totalité des items pertinents.

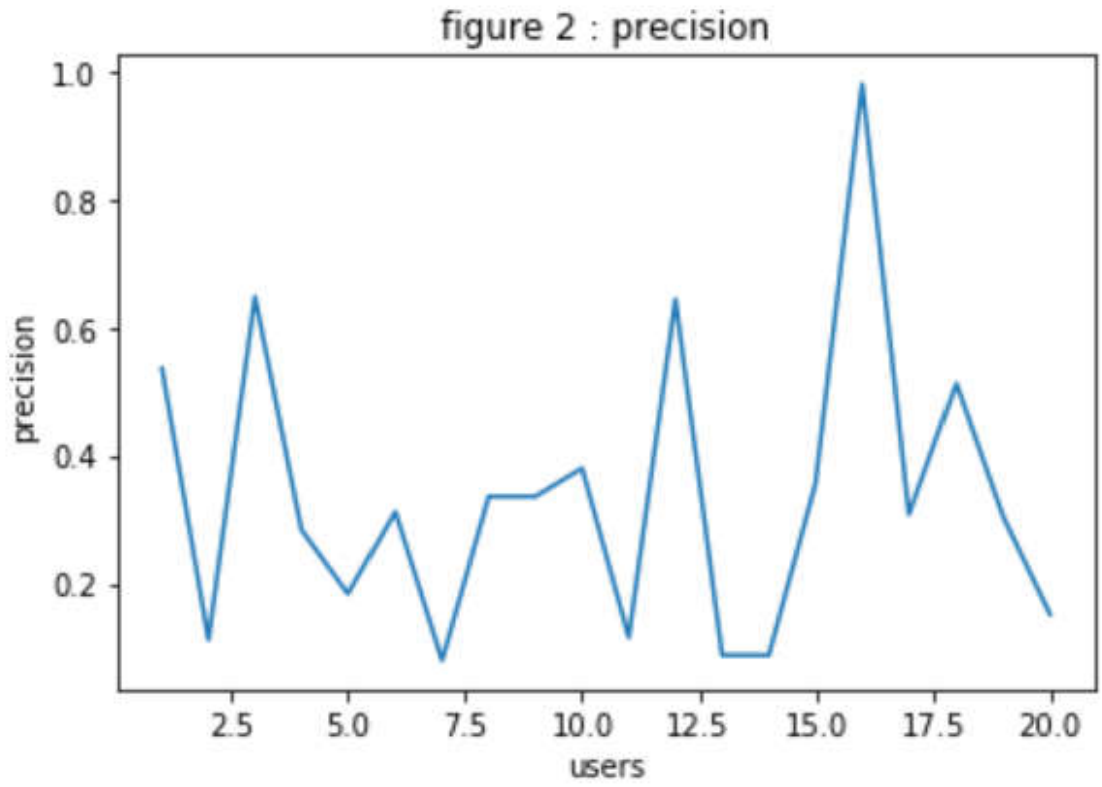

Figure (4.20) : la précision pour les 20 premiers utilisateurs

La figure (4.20) montre la précision calculée pour les vingt premiers utilisateurs. Les valeurs retrouvées sont variés entre 0.08 et 0.98, on aura donc une précision moyenne égale à 0.33. Cela signifié que le système retourne en réponse de nombreux documents erronés en plus de ceux pertinents.

La figure suivante illustre le résultat de calcule de la F1 mesure qui est égale à 0.50. Cela signifié que notre système est performent a 50%.

```
def F1():
    f1=2*rappel*precision /(rappel+precision)
    return f1
```

```
F1()
```

```
0.5018212356699551
```

Figure (4.21) : la valeur de F1 mesure obtenue par notre système

4.6 Conclusion :

Au long de ce chapitre, nous avons présenté notre collection de test, les outils nécessaires pour la réalisation de notre approche et nous avons présenté un aperçu de notre implémentation avec le résultat de la recommandation dans la première partie, afin de l'évaluer par la suite dans la deuxième partie en utilisant les métriques d'évaluations des systèmes de recommandation.

Conclusion générale

Conclusion générale

Les systèmes de recommandation ont reconnu une grande importance dans les différentes applications web pour aider les utilisateurs à sélectionner les items et produits qui correspondent à leur besoins. Comme tout autre domaine, les systèmes de recommandation ont leurs problèmes qui réduisent leur efficacité, où l'un de ces problèmes est : comment guider l'utilisateur dans son exploration de données afin de trouver des informations pertinentes ?

L'utilisateur doit donc être bien placé au centre du système de recommandation.

Notre objectif est donc de définir un profil utilisateur multidimensionnel à savoir : la dimension centre d'intérêt, la dimension âge et la dimension genre, et de l'intégrer dans un système de recommandation basé sur le contenu.

Dans notre travail nous avons présenté les étapes de définition du profil utilisateur ainsi que son intégration dans le processus de la recommandation puis nous terminons par une évaluation de l'approche proposée.

Comme perspectives à notre travail nous envisageons :

- L'évolution de profil utilisateur au cours de temps.
- De prendre en compte le contexte émotionnel.

BIBLIOGRAPHIE

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

Amato, G. and Straccia, U. (1999). User prole modeling and applications to digital libraries. In European Conference on Digital Libraries (ECDL),Paris. Springer.

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

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.

BER (2002) Berti-Equille L, (2002). Annotaion et recommandation collaboratives des documents selon leur qualité, *RSTI série ISI-NIS Recherche de filtrage d'information*, 7(12), pp. 125-155.

[BEZ, 73] Bezdek J. C., (1973). Fuzzy mathematics in pattern classification. *ph_D. dissertation.* Cornell University Ithaca, NY.

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

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

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

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 visualization. The Journal of Machine Learning Research,1 :49–75

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

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.

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

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. (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.

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 locationbased recommandation services for tour planing in mobile tourism applications . *In international Confernce on Electronic Commerce and web Tenchnologies.*

CHA et al (2008) Chandrasekaran K., Gauch S., Lakkaraju P., & Luong H. P., (2008). Concept-Based Document Recommendations for CiteSeer Authors. *In Proc. Of the 5th Inter. Conf.*, 5149, pp. 83-92.

CHE et al (2009) Chen G., Wang F., Zhang C., (2009). Collaborative Filtering Using Orthogonal Nonnegative Matrix Tri-factorization. *JIPM*, Vol. 45.

HAM (1989) Hammond, K. (1989). Case-based Planning: Viewing Planning as a Memory Task.
Boston, MA: Academic Press.

RIE et SCH (1989)Riesbeck, C., & Schank, R. C. (1989). Inside Case-Based Reasoning. *Hillsdale, NJ: Lawrence Erlbaum*.

RIJ (1979) Rijsbergen C.J.V, (1979). Information Retrieval. *Second edition, Butterworks.*

TOW et QUI (2000) Towle B., Quinn C., (2000). Knowldge Bsed Recommender Systems Using Explicit User Models, *In KBEM, AAAI Technical Report WS-00-04*, pp. 74-44.

TRE et JAR (2012), Treerattanapitak K., and Jaruskulchai C., (2012). Exponential fuzzy Cmeans for collaborative Filtering. *Journal of Computer Science and Technology*, 27(3), pp.567-576.

WAN et al (2008)Wang, J., De Vries, A. P., & Reinders, M. J. (2008). Unified relevance models for rating prediction in collaborative filtering. *ACM Trans. on Information Systems (TOIS),* 26(3), pp. 16.

WAN et al (2010), Wang J., Zhang N. J., Yin J., (2010). Collaborative filtering recommendation
based on fuzzy clustering of user preferences. *In 7th Inter. Conf. on FSKD*, Vol. 5, pp. 1946–1950.

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

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

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

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

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

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

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

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

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.

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 basedo a reinforcement method.

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.

BIBLIOGRAPHIE

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

Amato, G. and Straccia, U. (1999). User prole modeling and applications to digital libraries. In European Conference on Digital Libraries (ECDL),Paris. Springer.

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

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.

BER (2002) Berti-Equille L, (2002). Annotaion et recommandation collaboratives des documents selon leur qualité, *RSTI série ISI-NIS Recherche de filtrage d'information*, 7(12), pp. 125-155.

[BEZ, 73] Bezdek J. C., (1973). Fuzzy mathematics in pattern classification. *ph_D. dissertation.* Cornell University Ithaca, NY.

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

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

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

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 visualization. The Journal of Machine Learning Research,1 :49–75

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

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.

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

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. (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.

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 locationbased recommandation services for tour planing in mobile tourism applications . *In international Confernce on Electronic Commerce and web Tenchnologies.*

CHA et al (2008) Chandrasekaran K., Gauch S., Lakkaraju P., & Luong H. P., (2008). Concept-Based Document Recommendations for CiteSeer Authors. *In Proc. Of the 5th Inter. Conf.*, 5149, pp. 83-92.

CHE et al (2009) Chen G., Wang F., Zhang C., (2009). Collaborative Filtering Using Orthogonal Nonnegative Matrix Tri-factorization. *JIPM*, Vol. 45.

HAM (1989) Hammond, K. (1989). Case-based Planning: Viewing Planning as a Memory Task.
Boston, MA: Academic Press.

RIE et SCH (1989)Riesbeck, C., & Schank, R. C. (1989). Inside Case-Based Reasoning. *Hillsdale, NJ: Lawrence Erlbaum*.

RIJ (1979) Rijsbergen C.J.V, (1979). Information Retrieval. *Second edition, Butterworks.*

TOW et QUI (2000) Towle B., Quinn C., (2000). Knowldge Bsed Recommender Systems Using Explicit User Models, *In KBEM, AAAI Technical Report WS-00-04*, pp. 74-44.

TRE et JAR (2012), Treerattanapitak K., and Jaruskulchai C., (2012). Exponential fuzzy Cmeans for collaborative Filtering. *Journal of Computer Science and Technology*, 27(3), pp.567-576.

WAN et al (2008)Wang, J., De Vries, A. P., & Reinders, M. J. (2008). Unified relevance models for rating prediction in collaborative filtering. *ACM Trans. on Information Systems (TOIS),* 26(3), pp. 16.

WAN et al (2010), Wang J., Zhang N. J., Yin J., (2010). Collaborative filtering recommendation
based on fuzzy clustering of user preferences. *In 7th Inter. Conf. on FSKD*, Vol. 5, pp. 1946–1950.

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

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

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

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

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

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

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

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

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.

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 basedo a reinforcement method.

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.