République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et De la Recherche Scientifique
Université Mohamed El Bachir El Ibrahimi de Bordj-Bou-Arreridj

Faculté des mathématiques et d'informatique

Département d'Informatique

Mémoire de Fin d'études

En vue d'obtention du diplôme

Master en informatique

Spécialité : Technologies de l'information et de la communication

Thème

Système de recommandation par filtrage Collaboratif

Présenté par :

- Mr. BAHLOULI ACHRAF
- Mr. TOUAFEK AYOUB

Jugé le : ** Septembre 2020 devant le jury composé de :

- **Président** :
- **Examinateur** :
- **Encadreur** : Mr : BOUMAAZA Farid MAA à L'U. El Bachir El Ibrahimi BBA

Promotion 2019/2020

Remerciements

Nous remercions le bon Allah tout-puissant, qui nous a donné la force, la volonté et le courage pour terminer ce modeste travail.

Nous adressons notre profond remerciement à Mr **BOUMAAZA FARID** pour son encadrement, son écoute, ses élucidations, ses conseils, ses directives et encouragements qu'il nous a afflué.

Nous remercions Mrs les jurés pour l'intérêt qu'ils ont porté à ce travail en acceptant d'être examinateurs.

Ainsi, nous adressons nos remerciements les plus chaleureux à toutes les personnes qui ont aidé de près ou de loin par fruit de leur connaissance pendant toute la durée de notre parcours éducatif.

A tous les enseignants dans le département d'informatique

Tout simplement à tous ceux et celles qui méritent nos remerciements.

Dédicace

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

Je dédie ce modeste travail et ma profonde gratitude à ma mère Malika et mon père Djamel

À mes chères sœurs : Khaoula, Ritadj Sara.

À mon frère : Seifeddine

À toute ma famille.

À tous mes amis :

Ammar, Ilyas, Abderrahim, Larbi, Abdrezak, Abderrahmane, Akram, Walid, Abdesslam.

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

À Mon collège : Touafek Ayoub.

Finalement a tous ceux qui nous portent dans leurs cœurs.

BAHLOULI ACHRAF

Dédicace

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

À mes frères, mon collège : BAHLOULI ACHRAF et tous les membres de ma famille pour qui je souhaite tout le bonheur.

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

TOUAFEK AYOUB

Résumé

Les systèmes de recommandations sont des systèmes automatiques qui permettent, par des algorithmes d'apprentissage automatique, de fournir à des utilisateurs des suggestions qui répondent à leurs exigences .Parmi les techniques de recommandation, le filtrage collaboratif est la méthode la plus importante et la plus utilisée.

Dans ce mémoire, nous avons fait une étude comparative entre les deux méthodes de filtrage collaboratif pour enfin conclure quelle est la meilleure.

Mots-clés : apprentissage automatique, système de recommandation, filtrage collaboratif.

Abstract

Recommendation systems are automated systems that, through automatic learning algorithms, provide users with suggestions that meet their requirements. Collaborative filtering is the most important and most widely used method in recommendation systems.

In this paper, we have made a comparative study between the two collaborative filtering methods to finally conclude which is the best.

Keywords: machine learning, recommendation system, collaborative filtering.

ملخص

أنظمة التوصية أنظمة تلقائية توفر للمستخدمين، من خلال خوارزميات التعلم الآلي ، اقتراحات تلبي متطلباتهم.

من بين تقنيات التوصية، التصفية التعاونية هي الطريقة الأكثر أهمية والأكثر استخدامًا.

في هذه الأطروحة، قمنا بإجراء دراسة مقارنة بين طريقتين التصفية التعاونيتين لنستنتج أخيرًا أيهما الأفضل

الكلمات الرئيسية: التعلم الآلي، نظام التوصيات، التصفية التعاونية.

Table des matières

Chapitre IV : Expérimentation

Table des figures

Liste des tableaux

Introduction générale

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

Les systèmes de recommandations (SRs) sont des outils puissants aidant les utilisateurs, en ligne, à résoudre le problème de la surcharge d'informations auquel ils sont confrontés aujourd'hui, avec l'avènement d'internet, en leur fournissant personnalisées. Ce sont des systèmes de personnalisation qui présentent aux utilisateurs les contenus les plus pertinents, en utilisant certaines informations concernant leurs préférences passées.

Les méthodes les plus importantes et les plus utilisées pour le calcul de la recommandation sont basées sur le filtrage collaboratif. Mais les travaux dans la littérature de recherche, montrent que le FC classique n'est pas adapté à la recommandation d'intérêts multiples. En fait, la qualité de sa recommandation est très faible lorsque les utilisateurs des systèmes de recommandation ont des intérêts totalement différents.

Dans ce mémoire, nous allons expérimenter un système de recommandation, et nous allons faire une étude comparative entre les deux méthodes de filtrage collaborative pour enfin conclure quelle est la meilleure avec le langage de programmation python.

Pour ce faire, après une introduction générale notre travail est réparti en trois chapitres :

- **Le chapitre I :** ce premier chapitre est consacré pour présenter l'apprentissage automatique à travers quelques définitions et concepts de base.
- **Le chapitre II** : est consacré à la présentation des systèmes de recommandation en général.
- **Le chapitre III** : portera sur la présentation des différentes techniques des systèmes de recommandation.
- **Le chapitre IV:** est consacré pour la présentation de l'approche proposée.

Et enfin une conclusion générale qui résume notre travail et présente quelques perspectives.

CHAPITRE I

Apprentissage Automatique

CHAPITRE I

I.1 Introduction

Dans ce mémoire, nous nous intéressons aux théories, algorithmes et applications liés à un aspect particulier de l'intelligence artificielle (IA) : la faculté d'apprentissage.

De cette notion d'apprentissage, il devient facile de constater que le paradigme de programmation classique a évolué, ainsi :

- Avant, programmer consiste à prescrire une logique pour faire exécuter (une tâche).
- Maintenant, on programme pour rendre intelligent, autrement dit, programmer pour faire exécuter de manière intelligente des tâches nouvelles réputées nécessitant du raisonnement et un jugement. Par un tel mécanisme de programmation, nous serons en mesure de faire doter un programme d'une aptitude d'apprentissage.

I.2 Aperçu sur l'état actuel

Un bref historique dans le domaine de l'apprentissage artificiel, aussi communément appelé apprentissage automatique AA (ou Machine Learning, en anglais), nous amènent à parler des trois grandes époques de l'ordinateur, plus précisément, au tout début de l'informatique de son évolution au fil du temps et enfin au monde d'aujourd'hui et de demain.

De nos jours, nous pouvons constater, et ce n'est qu'un point de vue, que l'évolution de l'informatique s'est faite principalement sur deux axes [1] :

- Gain en capacité à cumuler de l'information et à sa diffusion dans des domaines tels que les fouilles de données (Data Mining), les entrepôts de données, les réseaux et services web, sans oublier leurs applications sous-jacentes sous smartphones.
- Gain en intelligence des systèmes informatiques, en particulier, les domaines liés à l'intelligence artificielle lesquels sont les plus touchés par cette avancée technologique et comprennent particulièrement, les jeux, la parole, la vision par ordinateur, etc.

I.3 AA, qu'est-ce que c'est ?

"L'apprentissage automatique, c'est la capacité d'un ordinateur à apprendre sans avoir été explicitement programmé."

Arthur Samuel

L'apprentissage automatique est un des champs de l'Intelligence Artificielle (IA) qui consiste en l'automatisation de l'apprentissage d'un algorithme, notamment par l'analyse, la sélection et le traitement de données. Ainsi, les IA auront la possibilité, grâce au Machine Learning, d'expérimenter et de tirer des conclusions de leurs expérimentations. Il peut aussi être utilisé pour reconnaitre des éléments (images, objets, ...) à partir d'une base de données [34].

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage**.**

Donc il consiste à apprendre, en tirant des prévisions de fonctionnement ou de comportement à partir de masses de données gigantesques. Il aboutit à la mise en place de programmes informatiques qui n'auraient jamais pu être écrits en passant par l'algorithmique classique.

On écrit des programmes à partir de ce que l'on a retiré de l'observation des données et qui permettent de faire la même chose. "Historiquement, cette théorie a pris son essor avec les travaux des mathématiciens Vapnik et Chervonenkis dans les années 60", rappelle Stéphan Clémençon [4]. "Ses travaux se sont développés un peu à l'écart du monde des probabilités et des statistiques. Ils n'ont pas été reconnus tout de suite, car avec le Machine Learning, le point de vue est différent de celui de la statistique traditionnelle. Le Machine Learning ne se concentre plus sur la façon de retrouver des objets abstraits comme une loi de probabilité par exemple, mais se concentre avant tout sur le côté opérationnel, c'est-à-dire la prise de décision à partir des données en faisant le moins d'erreurs possible."

I.4 Modèles de l'apprentissage automatique

Le modèle va permettre à l'apprenant d'apprendre efficacement selon un biais pour la réalisation de la tâche désirée. Le choix du modèle à employer est donc primordial pour réussir un apprentissage optimal. Ce biais est d'une nature différente selon le modèle retenu :

- **fonctionnel** : il est alors basé sur une fonction de décision sur les attributs d'entrées permettant d'établir une séparatrice. La plus célèbre et la plus simple est sans doute la fonction linéaire (exemple : SVM support vecteur machine).
- **probabiliste** : il est alors basé sur une distribution de probabilité sur les attributs des entrées. Les réseaux bayésiens sont sans doute l'exemple typique de ce genre de modèle.
- **connexionniste** : il est alors basé sur un réseau de neurones. Ce type de modèle s'inspire à la base du fonctionnement du cerveau humain Nous citerons les perceptrons et les réseaux de neurones multicouches (MLP) comme exemples typiques connexionnistes.
- **temporelle** : il est alors basé sur un couplage temporel entre les entrées. Ce type de modèle décrit différents états temporels dans lesquels on est susceptible de se trouver l'exemple typique est le modèle de MARKOV Caché.

I.5 Domaines de l'apprentissage automatique

Les principaux domaines d'application de l'apprentissage automatique (AA) sont les fouilles de données et l'intelligence artificielle.

- La fouille de données (Data Mining, en anglais) est le processus d'extraction de la connaissance :
  Il consiste à sélectionner les données à étudier à partir de bases de données (hétérogènes ou homogènes), à épurer ces données et enfin à les utiliser en apprentissage pour construire un modèle.

Exemples

- Trouver une prescription pour un malade (patient) à travers des fichiers médicaux antérieurs.

- Apprentissage de la reconnaissance de transactions frauduleuses par carte de crédit, par examen des transactions passées avérées frauduleuses.
- L'intelligence artificielle, la vision par ordinateur, la robotique, l'analyse et la compréhension des images, la reconnaissance de formes, reconnaître des objets dans les vidéos et extraire des contenus sémantiques des images sont autant d'applications qui requièrent la construction de modèles par apprentissage automatique [1].

Exemples

- Systèmes de vidéo surveillance pour la détection des intrus.
- Logiciel biométrique de reconnaissance de visages et d'empreintes digitales.

I.6 Apprentissages et élaboration du modèle

Le fonctionnement de le Machine Learning se base sur deux types de techniques : l'apprentissage supervisé et l'apprentissage non-supervisé. Ces deux techniques s'inscrivent dans les phases d'apprentissage et de prédiction qui caractérisent le fonctionnement du Machine Learning.

I.6.1 Apprentissage supervisé

L'apprentissage est dit supervisé lorsque les données qui entrent dans le processus sont déjà catégorisées et que les algorithmes doivent s'en servir pour prédire un résultat en vue de pouvoir le faire plus tard lorsque les données ne seront plus catégorisées. On peut par exemple donner au système une liste de profils clients contenant des habitudes d'achats, et expliquer à l'algorithme quels sont les clients habituels et les clients occasionnels. Une fois l'apprentissage terminé, l'algorithme devra pouvoir déterminer tout seul à partir d'un profil client à quelle catégorie celui-ci appartient**.**

I.6.2 Apprentissage non-supervisé

Comme son nom l'indique, il est l'opposé du premier. Dans le premier, l'apprentissage s'effectue de manière subjective c'est-à-dire qu'une personne décrit au préalable quels critères existent dans son jeu de données à analyser et classer. Dans cet apprentissage-ci, le programme ne reçoit que des valeurs et doit créer les classes dans lesquelles les attribuer. Il va donc "décider" lui-même le nombre de classes à créer pour ensuite ranger les données dans chaque classe. Ces algorithmes sont utilisés lorsque nous n'avons pas d'échantillon à disposition. Il est très intéressant d'utiliser cette technique lorsque vous ne savez pas ce que vous cherchez. Typiquement quand vous voulez faire éclore quelque chose qui ne vous viendrait pas à l'esprit. Ce type d'apprentissage est aussi appelé « clustering », qui signifie « groupement » [2].

Pour l'exemple, on l'utilise dans un traitement d'images particulier, qui sélectionne les pixels similaires et en déduit qu'ils sont liés à un objet précis. Une reconstitution 3D à partir d'une photo peut être réalisée ensuite par ce biais (on appelle ce domaine-là « Computer Vision »).

Dans l'illustration ci-dessous, un algorithme de clustering a permis de regrouper les pixels par région et d'en séparer les différents composants. D'autres algorithmes sont ensuite utilisés pour recréer une projection en trois dimensions [2].

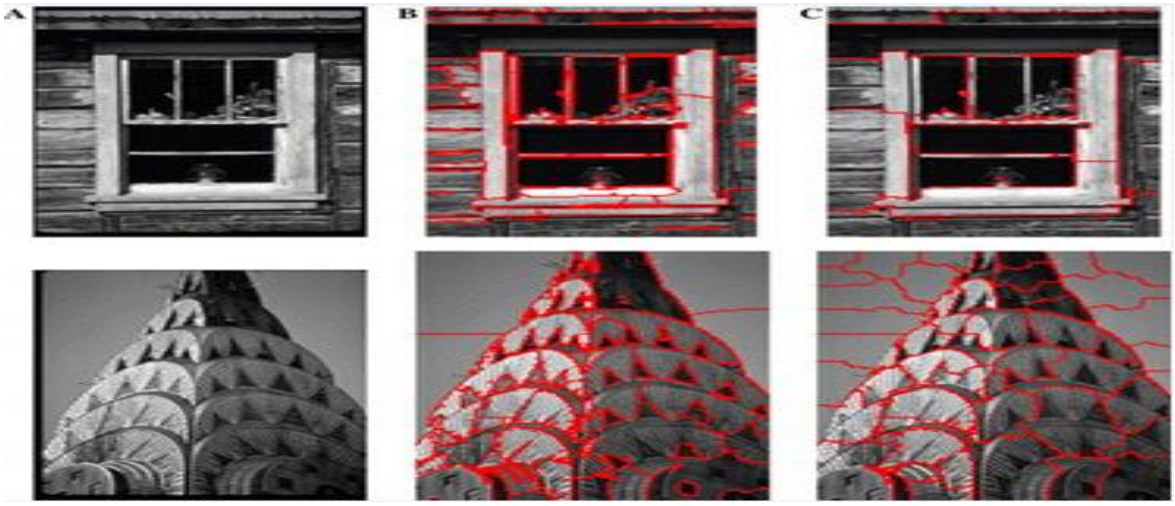


FIGURE I.1 - Exemple d'apprentissage non-supervisé

I.7 Algorithmes mis en œuvre par l'AA

Il existe trois grandes familles d'algorithme de Machine Learning :

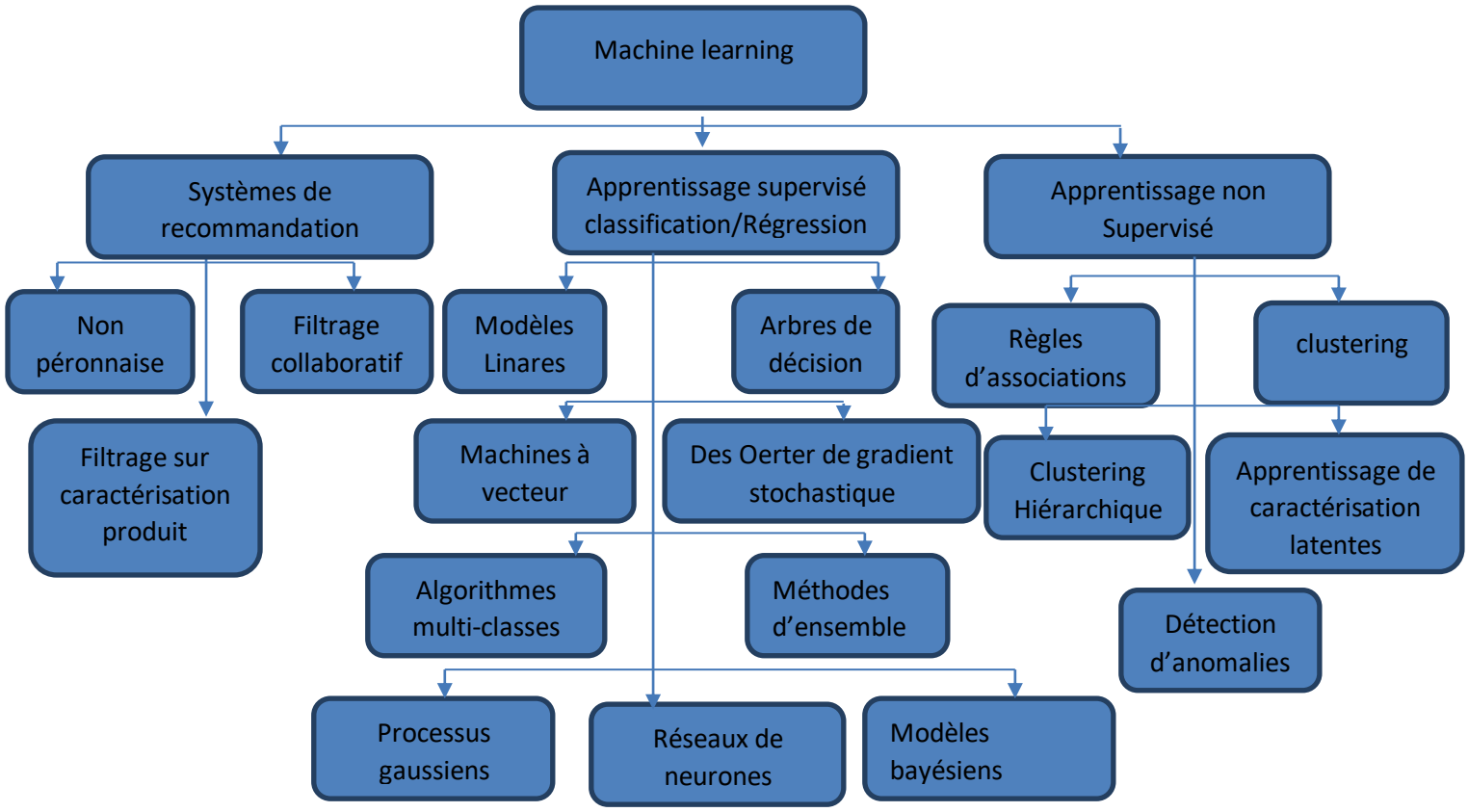


FIGURE I.2- Algorithmes mis en œuvre par le Machine Learning

La figure (I.2) montre qu'il y en a beaucoup d'algorithme mis en œuvre par le ML, tout l'enjeu ça va être de choisir le bon algorithme pour le bon cas d'utilisation, parce qu'il n'y a pas un algorithme qui marche pour tous les cas.

Les Data Scientiste disposent déjà de tout un attirail méthodologique pour développer leurs algorithmes prédictifs. À eux de choisir le ou les outils les plus adaptés et de bien les paramétrer. Les entreprises qui souhaitent utiliser le Machine Learning recourent donc de plus en plus au service des Data Scientiste, et fait appel à des algorithmes et modèles mis à disposition par des 'brokers' d'algorithmes en les adaptant à leurs domaines spécifiques.

Non seulement il n'existe pas un algorithme universel capable de prédire un comportement, un phénomène, mais tout reste encore à inventer comme aime à le souligner Stéphan Clémençon"[4] en termes d'algorithmes, tout reste à faire. Par exemple, aujourd'hui, nous sommes fréquemment amenés à gérer des graphes de données. La théorie des graphes, ce n'est pas quelque chose de neuf en mathématiques appliquées. Mais auparavant, on traitait des graphes qui n'ont rien à voir avec ceux que nous sommes amenés à traiter aujourd'hui pour identifier des communautés, mettre en avant des attachements communautaires. Les réseaux sociaux d'aujourd'hui n'ont plus rien des graphes que l'on utilisait dans les années50. Il n'existe même pas de méthode qui permettrait de simuler des graphes réalistes d'une telle taille."

De même en ce qui concerne les moteurs de recommandation et d'analyse des données de préférence des utilisateurs : leur principe n'est pas nouveau. "En revanche, aujourd'hui un internaute exprime ses préférences sur un nombre d'objets qui est colossal", poursuit Stéphan Clémençon [4]. "Toutes les méthodes qui étaient basées sur une modélisation des lois sur les permutations ne passent pas du tout à l'échelle. Si vous avez n objets, vous avez factorielle n façon de les classer. De plus, les internautes n'expriment pas leur préférence sur les mêmes objets. Certains font un rating sur beaucoup d'objets, d'autres sur très peu. Les données sont très hétérogènes, très complexes, et on doit réfléchir aux bons modes de représentation."

1.8 Applications du Machine Learning

Si aujourd'hui on parle énormément du Machine Learning, c'est essentiellement pour ses applications pour le web, notamment dans les systèmes de recommandation des sites marchands ou des jeux en ligne. Néanmoins, la technique est apparue bien avant, notamment pour la compression de données ou encore la visualisation de grandes quantités d'informations. "Le problème phare du Machine Learning a été la reconnaissance de formes qui a engendré des applications dans la biométrie, la reconnaissance de visages ou de caractères manuscrits" rappelle Stéphan Clémençon [4]. Autre exemple évoqué par le chercheur, la transmission des images et vidéos sur Internet. "Si leur transmission est beaucoup plus rapide aujourd'hui, c'est grâce au format JPEG 2000 et la compression par ondelettes. C'est un mode de représentation qui permet de s'adapter à la variabilité des données à la volée".

Depuis ces premières applications, des algorithmes de Machine Learning ont été déployés à grande échelle pour le filtrage anti-spam, pour optimiser les stocks dans la distribution, et bien évidemment pour la segmentation et le ciblage des clients. Mais des applications de la Machine Learning commencent aussi à apparaître dans la maintenance industrielle, notamment la maintenance prédictive des matériels installés sur les plateformes pétrolières, les moteurs d'avions... De leur côté, les objets connectés, des compteurs aux bracelets connectés, laissent augurer de multiples autres applications.

Autre secteur avide de nouveaux algorithmes, le domaine de la gestion des risques qui s'est intéressé à la Machine Learning voici déjà plusieurs années. Janvier Régis Habimana [4]

souligne : "des modèles se basent sur le Machine Learning afin de fournir une prédiction de risques pris, par exemple, par des établissements financiers dans le cadre de prêts ou de contrats d'assurance" Enfin, le Machine Learning peut trouver des applications dans la lutte contre la criminalité. Plusieurs villes américaines font aujourd'hui appel à IBM pour prédire les zones où pourraient survenir les prochaines agressions.

I.9 Conclusion

Dans ce chapitre, nous avons présenté tout d'abord la définition de l'apprentissage automatique et ses origines, ensuite nous avons donné un rapide aperçu de ce qui existe déjà grâce au domaine de l'apprentissage automatique. Parmi les innombrables applications du Machine Learning, nous nous intéressons aux systèmes de recommandation. Ces systèmes sont très développés aujourd'hui, mais demeurent au final peu visible, l'utilisateur ne percevant que le résultat, à savoir une liste de suggestions. Les champs d'application de ces systèmes de recommandation sont divers et variés (suggestion de films, de produits marchands, de services...).

Dans le chapitre suivant nous allons présenter les fondements des systèmes de recommandation.

CHAPITRE II

Système de recommandation

CHAPITRE II

II.1 Introduction

De nos jours, un grand nombre de systèmes de recommandation sont utilisés dans divers domaines. Le principal but de ces systèmes est de filtrer le flux d'informations de façon à fournir pour chaque utilisateur les ressources qui répondent à son besoin en information. Afin de satisfaire ces contraintes, les moteurs de ces systèmes gèrent les profils des utilisateurs pour choisir les ressources à transmettre à chacun, et mettre à jour ces profils en fonction des retours et des interactions des utilisateurs. Un système de recommandation fournit à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnels. Les applications de recommandation peuvent être trouvé dans une grande variété d'industries, entreprises, service financier, musique / radio en ligne, tv et vidéos, les publications en ligne, et d'innombrables autres.

II.2 Historique

Les systèmes de recommandation sont reconnus assez tôt dans l'histoire de l'informatique, «Information Lens System» [17] peut être considérée comme le premier système de recommandation à l'époque, l'approche la plus commune pour le problème du partage d'informations dans l'environnement de messagerie électronique était la liste de distribution basée sur les groupes d'intérêt.

La première définition pour le filtrage a été donnée aussi par Malone : " Même si le terme a une connotation littérale de laisser les choses dehors (filtrage négatif : enlèvement), nous l'utilisons ici dans un sens plus général qui consiste à sélectionner les choses à partir d'un ensemble plus large de possibilités (filtrage positif : sélection)".

La littérature académique a introduit le terme de filtrage collaboratif par le système « Tapestry » [17] , Il a été développé en 1992 par le centre de recherche de "Xerox" aux États-Unis, il s'agit d'un système de recommandation intégré à une application de mail électronique qui a permis aux utilisateurs de créer des requêtes permanentes, basées sur les annotations (les tags) des utilisateurs.

Quelques années plus tard, un certain nombre de systèmes académiques de recommandation ont vu le jour en 1994 et en 1995, tels que le système de recommandation d'articles d'actualités et de films développé par «Group Lens » [27] et le système de recommandation de musique "Ringo". Ces deux systèmes sont également basés sur le filtrage collaboratif, des livres [27], des vidéos, des films, des pages Web, des articles et des liens Internet.

Ensuite, en 2006, Netflix a lancé Netflix Prize pour améliorer l'état de recommandation des films, aussi Netflix a plus de 17.000 films dans sa sélection. Aujourd'hui les systèmes de recommandation sont devenus très populaires et sont utilisés dans diverses applications Web.

II.3 Définition

Les systèmes de recommandation peuvent être définis de plusieurs façons, vu la diversité des classifications proposées pour ces systèmes, mais il existe une définition générale de Robin Burke [Burke, 2002] qui les défini comme suit :

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

Les deux entités de base qui apparaissent dans tous les systèmes de recommandations sont l'usager et l'item. L'«usager» est la personne qui utilise un système de recommandation, donne son opinion sur divers items et reçoit les nouvelles recommandations du système. L'«item» est le terme général utilisé pour désigner ce que le système recommande aux usagers.

Les données d'entrée pour un système de recommandation dépendent du type de l'algorithme de filtrage employé.

Le système de recommandation identifie quatre caractéristiques clés :

- Aide à décision : prédire une note à un utilisateur pour un article.
- Aide à la comparaison : classer une liste d'articles d'une manière personnalisée pour un utilisateur.
- Aide à la découverte : fournir à un utilisateur des articles inconnus qui seront appréciés.
- Aide à l'exploration : donner des articles similaires à un article cible donné.

II.4 Objectifs d'un SR

Nous avons précédemment défini des systèmes de recommandation comme des outils et des méthodes fournissant aux utilisateurs des suggestions d'articles qu'ils aimeraient acheter ou utiliser.

Dans cette section, nous visons à donner une définition plus fine en illustrant les utilisations possibles d'un SR. La première distinction qui doit être faite est entre l'utilisateur de SR et le fournisseur de services [28]. Par exemple, un restaurant ou un système de recommandation d'hôtel est généralement utilisé par un intermédiaire (par exemple, TripAdvisor) afin d'augmenter son taux de conversion, c'est-à-dire augmenter le nombre de personnes allant à un restaurant donné ou vendre plus de chambres d'hôtel.

De l'autre côté, les motivations de l'utilisateur pour utiliser un système comme TripAdvisor sont de trouver un restaurant ou un hôtel qui correspond à ses goûts et besoins, augmentant ainsi sa satisfaction.

En fait, il existe plusieurs raisons pour lesquelles les fournisseurs de services emploient des moteurs de recommandation :

- **Augmenter les revenus :** en d'autres termes, cela signifie augmenter le nombre d'articles qui sont vendus. Cette fonction est probablement la plus importante dans un contexte SR. Le but ici est de vendre réellement plus d'articles qu'il n'y en aurait eu sans aucune recommandation. Pour atteindre cet objectif, le système recommande des articles qui sont censés satisfaire les goûts et les besoins de l'utilisateur. Cependant, nous devons faire une distinction entre la prédiction des intérêts des utilisateurs dans un élément et la probabilité que les utilisateurs choisissent /sélectionnent réellement l'élément recommandé.

- **Augmenter la diversité des articles vendus :** le but de cette fonction est d'inciter les utilisateurs à sélectionner des éléments qui resteraient inconnus sans recommandation. Par exemple, dans le cas d'un SR des livres (par exemple librairie Amazon), le fournisseur de services veut pouvoir vendre des livres de tous ses catalogue et pas seulement les 5 plus populaires.
- **Comprendre les utilisateurs :** une autre fonction importante d'un SR est de pouvoir décrire les préférences des utilisateurs. Ces préférences peuvent être recueillies explicitement ou en les prédisant. Ces données peuvent être utilisées par le fournisseur de services pour mieux gérer sa production ou son stock.

Nous expliquons maintenant les fonctions auxquelles les utilisateurs pourraient être intéressés lors de l'utilisation d'un SR.

- **Classement d'une liste d'éléments :** C'est probablement l'une des fonctions les plus importantes pour un SR, c'est-à-dire de fournir certains bons éléments à l'utilisateur actuel, selon les prévisions de notation. En d'autres termes, recommander des articles que l'utilisateur devrait aimer.
- **Recommander une séquence :** Cette fonction vise davantage à s'adapter aux préférences à long terme des utilisateurs. Le principe est de générer une suite cohérente de recommandations au lieu de fournir une succession d'indépendantes. Par exemple, il serait logique de recommander Matrix 2 Reloaded après avoir recommandé Matrix 1 [31].
- **Navigation améliorée :** Étant donné un grand catalogue, la tâche d'un SR peut être améliorée l'expérience de navigation de l'utilisateur en l'aidant à trouver des articles qui correspondent à ses goûts et à ses besoins.

II.5 Concepts de base, et Notions liées

Dans cette section, nous définissons quelques concepts relatifs aux systèmes de recommandation qui seront utilisés dans cette thèse.

II.5.1 Les entités utilisateur et item

Dans tout système de recommandation, il existe deux entités importantes qui sont les utilisateurs et les items (articles).

- **Utilisateur** : est une personne qui accède au système et fait l'enregistrement en saisissant ses informations démographiques, ses centres d'intérêt et d'autres informations personnelles. L'ensemble des utilisateurs dans le système est représenté par U, un utilisateur est donné par U.
- **Item** : dans les systèmes de recommandation, un item est l'entité qui représente tout élément constituant une liste de recommandation et qui correspond aux besoins de l'utilisateur, incluant tout produit susceptible d'être vendu (livre, produits....etc. dans les sites de l'e-commerce tel que Amazon.com), vu (les films dans les sites de TV en ligne tel que Netflix), écouté (la musique) ou lu (tel que les informations dans les journaux en ligne, les revues dans les bibliothèques numériques), ainsi que les destinations de vacance, des restaurants, etc. L'ensemble des items disponibles dans le système est représenté par I, où $i \in I$.

II.5.2 Matrice d'évaluation utilisateur-item

L'ensemble du système <u, i> sont enregistrés dans une base de données creuse appelée Matrice d'évaluation (Rating Matrix) ou encore Matrice utilisateur-item (user-item Matrix) et elle est notée par R, où chaque ligne correspond aux évaluations fournies par un seul utilisateur et une colonne correspond aux évaluations qu'a eues 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 FC, qui utilisent les préférences (votes) pour la génération des recommandations. Afin d'améliorer la performance des systèmes de recommandation, plusieurs recherches sont apparues offrant de nouveaux scénarios pour introduire de nouvelles informations à la matrice d'évaluation [31]. Ces informations peuvent être divisées en deux catégories, selon leurs sources ou bien leurs associations à l'interaction avec le système.

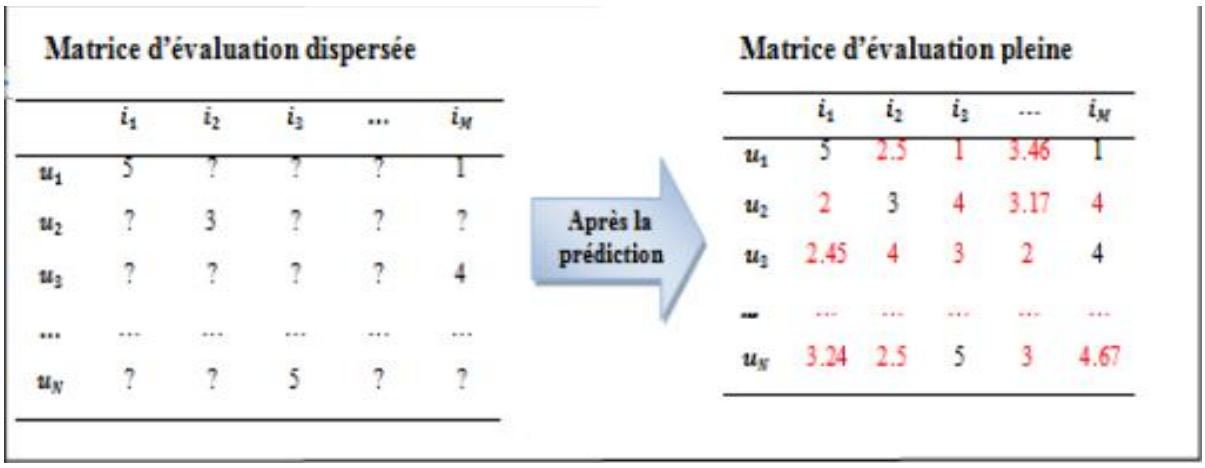
Le premier type est celui de l'information littérale riche sur les utilisateurs (genre, âge, loisirs... Etc.) Ou sur les items (catégorie, contenu…etc.) [31], qui est devenue importante et très utilisée surtout dans les réseaux sociaux et les technologies du Web 2.0 (tags, commentaire, contenu multimédia) [31].

Le deuxième type d'information correspond à l'information associée à l'interaction de l'utilisateur avec le système, incluant le temps d'évaluation ou d'achat des items, l'emplacement (local) de l'utilisateur ainsi que les commentaires et les avis des utilisateurs. [18].

II.5.3 La prédiction

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

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



	i_1	i_2	i_3	...	i_M
u_1	5	?	?	?	1
u_2	?	3	?	?	?
u_3	?	?	?	?	4
...	...	...	...	...	...
u_N	?	?	5	?	?

	i_1	i_2	i_3	...	i_M
u_1	5	2.5	1	3.46	1
u_2	2	3	4	3.17	4
u_3	2.45	4	3	2	4
...	...	...	...	...	...
u_N	3.24	2.5	5	3	4.67

FIGURE II.1- Un exemple illustratif de la prédiction des évaluations manquantes

II.5.4 La recommandation

La recommandation est l'action de calculer une liste d'items (Top- N items) que l'utilisateur aimera le plus. Le calcul des listes de recommandation se fait en attribuant des scores pour les items selon leurs popularités ou leurs préférences par exemple [34]. Contrairement à la prédiction le calcul des recommandations ne se base pas strictement sur les évaluations.

II.6 Les étapes principales d'un SR

Un système de recommandation requiert généralement trois (03) étapes :

1. La première consiste à recueillir de l'information sur l'utilisateur.
2. La deuxième consiste à bâtir modèle utilisateur contenant l'information recueillie.
3. La troisième consiste à extraire à partir de cette matrice une liste de recommandations.

II.6.1 La collecte d'informations

Pour être pertinent, un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Une distinction peut être faite entre deux formes de collecte de données :

1. Collecte de données explicites - Filtrage actif :

La collecte repose sur le fait que l'utilisateur indique explicitement au système ses intérêts.

Exemple : demander à un utilisateur de commenter, taguer/étiqueter, noter, liker ou encore ajouter comme favoris des contenus (objets, articles...) qui l'intéressent. On utilise souvent une échelle de ratings allant de 1 étoile (je n'aime pas du tout) à 5 étoiles (j'aime beaucoup) qui sont ensuite transformées en valeurs numériques afin de pouvoir être utilisées par les algorithmes de recommandation.

Avantage : capacité à reconstruire l'historique d'un individu et capacité à éviter d'agréger une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).

Inconvénient : Les informations recueillies peuvent contenir un biais dit de déclaration. [14]

2. Collecte de données implicite - Filtrage passif : elle repose sur une observation et une analyse des comportements de l'utilisateur effectué de façon implicite dans l'application qui embarque le système de recommandation, le tout se fait en "arrière-plan" (en gros sans rien demander à l'utilisateur)

Exemple :

- Obtenir la liste des éléments que l'utilisateur a écoutés, regardés ou achetés en ligne.
- Analyser la fréquence de consultation d'un contenu par un utilisateur, le temps passé sur une page.
- Montrer le comportement en ligne de l'utilisateur.
- Analyser son réseau social.

Avantage : aucune information n'est demandée aux utilisateurs, toutes les informations sont collectées automatiquement. Les données récupérées sont a priori justes et ne contiennent pas de biais de déclaration.

Inconvénient : les données récupérées sont plus difficilement attribuables à un utilisateur et peuvent donc contenir des biais d'attribution (utilisation commune d'un même compte par plusieurs utilisateurs). Un utilisateur peut ne pas aimer certains livres qu'il a achetés, ou il peut l'avoir acheté pour quelqu'un d'autre.

II.6.2 Modèle utilisateur

Le modèle utilisateur se présente généralement sous forme de matrice appelée "matrice d'évaluation utilisateur-item ", cette dernière contient des données recueillies sur l'utilisateur associées aux produits disponibles sur le site web.

Un autre point important est comment le temps influence le profil de l'utilisateur. Les intérêts des utilisateurs, généralement, évoluent au cours du temps. Les données du modèle utilisateurs devraient donc constamment être réajustées pour rester conformes aux nouveaux centres d'intérêt de l'utilisateur.

Le tableau II.1 illustre un exemple d'une matrice de notes pour 3 utilisateurs et 3 films.

Les valeurs marquées " ? " indiquent que l'utilisateur n'a pas donné d'avis.

	Inception	Batman begins	Tianic	Star wars
User A	?	2	5	4
User B	4	1	?	5
User C	3	2	?	4
User D	2	4	5	3

TABLE II.1 - Exemple de matrice de note

II.6.3 Liste de recommandations

Pour extraire une liste de suggestions à partir d'un modèle utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre 2 choses. Plus la ressemblance est forte, plus la valeur de la similarité sera grande. À l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite.

On a l'habitude de présenter 4 approches possibles pour un système de recommandation:

- Recommandation Personnalisée.
- Recommandation Objet (Content-Based filtering CB).
- Recommandation Sociale (Collaborative Filtering CF – Context Aware).
- Recommandation Hybride.

II.7 Classification des SRs

Durant ces dix dernières années, plusieurs classifications sont apparues, suivant les données à recommander, suivant les informations disponibles et bien évidemment suivant l'objectif visé. Ce qui conduit l'émergence d'un débat sur quelle classification est la meilleure. (Voir la figure II.2) :

1. La classification classique : cette classification est reconnue par trois types de filtrage.

- Un filtrage collaboratif(CF).
- Un filtrage basé sur le contenu(CBF).
- Le filtrage hybride.

2. La Classification de [Su et al, 2009] [6] : elle est utilisée dans les systèmes collaboration. Ils proposent une sous classification qui comprend les techniques hybrides les classer dans les méthodes de collaboration hybrides. [Su et al, 2009] classe filtrage collaboratif en trois catégories :

- Approches FC à base de mémoire : pour K-plus proches voisins.

- Approches FC basé sur un modèle englobant une variété de techniques telles que : clustering, les réseaux bayésiens, factorisation de matrices, les processus de décision de Markov.

3. La classification de Burke [12] : propose une classification très complète des techniques de recommandation existantes en identifiant les données d'entrée de chaque méthode et son algorithme utilisé. Il définit cinq types de Techniques de recommandation :

- Filtrage collaboratif.
- Filtrage basé sur contenu.
- Filtrages démographiques.
- Filtrage basé connaissances.
- Filtrage communautaire.

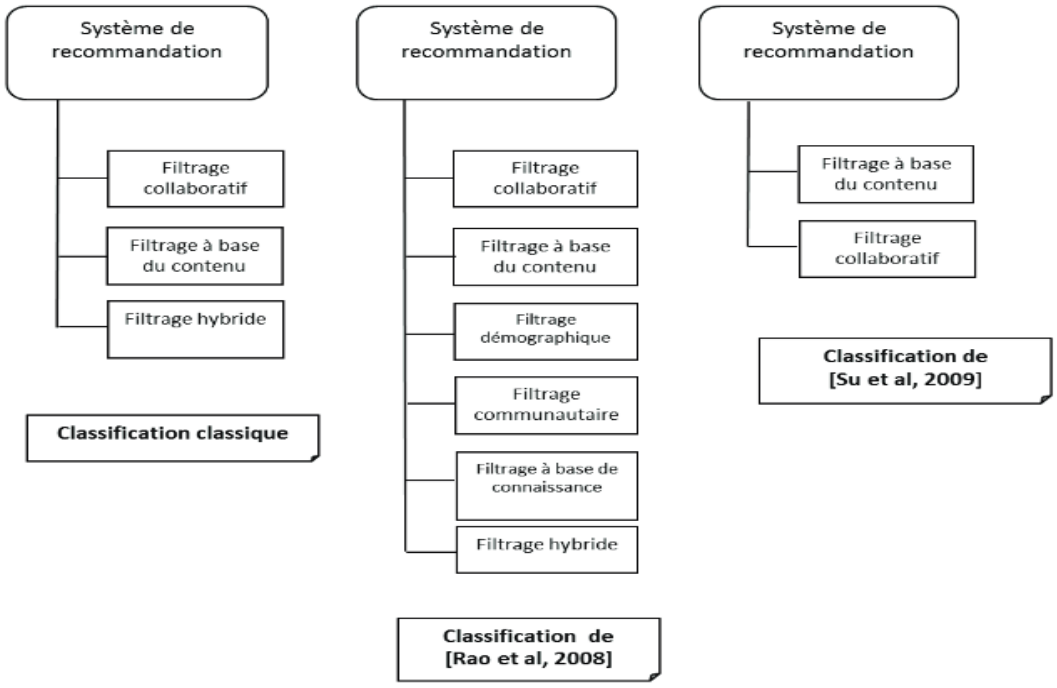


FIGURE II.2 - Classification principale des systèmes de recommandations.

II.8 Moteurs de recommandation : un nombre assez important de systèmes de recommandation sont proposés dans la littérature. Nous citons entre autres :

II.8.1 Amazon

Basé sur plus de 20 ans d'expérience dans le domaine des recommandations et de recherche en machine learning chez Amazon, Amazon Personalize vous permet d'améliorer l'engagement et la conversion des clients grâce à des recommandations personnalisées de produits et de contenus en temps réel, et à des promotions marketing ciblées. Cela revient à avoir 24 heures sur 24 votre propre système de recommandations de machine learning Amazon.com.

Il n'est pas nécessaire d'avoir une expérience en machine learning puisque grâce à des API simples, vous pouvez intégrer facilement des fonctionnalités de personnalisation sophistiquées dans vos systèmes et votre plate-forme. Amazon Personalize automatise les étapes complexes nécessaires à la création, à la formation, au réglage et au déploiement d'un modèle de recommandation, pour vous permettre d'offrir plus rapidement des expériences utilisateur personnalisées.

Toutes vos données sont chiffrées. Elles sont par conséquent privées et sécurisées, et ne sont utilisées que pour créer des recommandations à l'intention de vos utilisateurs. Vous payez uniquement pour ce que vous utilisez ; il n'y a pas de frais minimums ni d'engagement initial [3].

Mais comment Amazon a-t-il développé un outil aussi sophistiqué et performant ?

FIGURE II.3 - Amazon recommandation

Les consommateurs aiment être conseillés et avisés par des personnes partageant leurs goûts, c'est sur cette théorie qu'Amazon a développé son système de recommandation automatique, misant ainsi sur du contenu (2 produits peuvent être plus en moins liés en fonction du nombre de mots similaires dans leur description par exemple.) et sur le filtrage collaboratif (la construction d'une matrice de relation entre les différents achats des consommateurs). Amazon a développé son mystérieux algorithme item-toitem collaborative filtering, la priorité est clairement la personnalisation des produits recommandés en fonction des intérêts de chaque consommateur. La tendance serait de recommander les produits similaires entre des clients ayant des similitudes (Âge, ville, sexe, CSP...) mais Amazon a préféré faire le choix de recommander des produits en fonction d'une liste établie avec les consultations de produits, achats et classements réalisés par les consommateurs eux-mêmes [5].

II.8.2 Pigdata-La recommandation produit orienté métier et visuel

L'outil de recommandation produit Pigdata fonctionne en mode Saas avec un algorithme de collaborative filtering avec plus de 100 paramètres analysés. Il permet également de réaliser du Cross sell et de l'Up Sell automatiques. Pigadata est customizable avec 120 paramètres analysés comme le temps passé sur la page, les clics effectués ou, bien entendu, les achats réalisés [5]. La solution assure également des fonctions de cross-selling et up-selling. Elle est customizable (templates personnalisés ou template Pigdata) et paramétrable.

FIGURE II.4 - Pigdata-La recommandation produit orienté métier et visuel.

II.8.3 Google

Google personnalise nos résultats de recherche quand cela est possible, en se basant sur notre localisation et/ou nos dernières recherches. Lorsqu'on est connecté à notre compte Google, il propose un contenu encore plus pertinent en fonction de notre historique de recherche. L'algorithme du PageRank est de manière intrinsèque un outil basé sur de la recommandation sociale dans la mesure où il utilise les liens entre les pages web. L'utilisation de contenus provenant de nos cercles de Google+ est aussi une forme de recommandation sociale [26].

II.8.4 Nuukik-Plusieurs approches d'analyse prédictive

Le lancement de Nuukik fait suite à 3 ans de "R and D" menée avec l'INRIA pour développer les algorithmes de recommandation de ce service. Celui-ci panache le Collaborative filtering, c'est-à-dire une recommandation qui s'appuie sur les ventes déjà réalisées ainsi que la navigation des internautes avec une recommandation basée sur les contenus, typiquement les attributs de la fiche produite. Ce mélange de deux approches doit permettre selon les ingénieurs de la start-up de rester pertinent même lorsqu'il y a peu d'historique sur un produit au moment des changements de collection dans le textile [30].

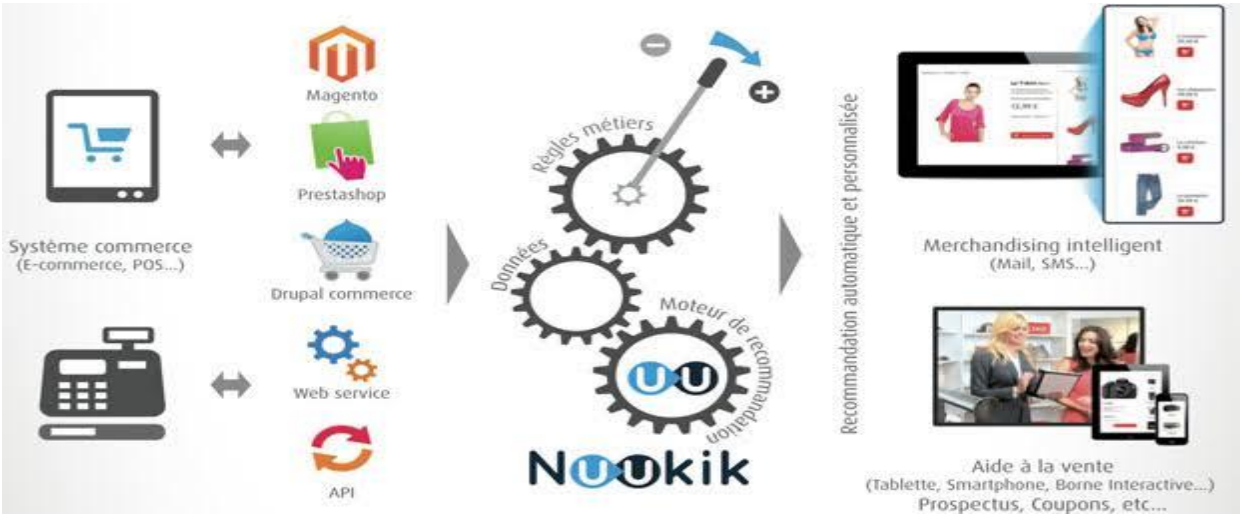


FIGURE II.5 - Moteur Nuukik.

De base, Nuukik propose sa technologie sous forme de service Cloud avec une facturation à la requête. Néanmoins, l'éditeur n'exclut pas la possibilité de déployer son logiciel en mode on premise pour les sites qui en exprimeront le besoin [30].

II.8.5 Ezako-Un surplus d'intelligence

L'outil Ezako collecte des informations et permet de former des groupes de clients aux goûts similaires mais aussi de découvrir les goûts individuels des clients. Le tableau de bord est conçu pour piloter le plus simplement possible les actions. Les recommandations basées sur l'historique de ventes et les clients similaires peuvent être orientées par des règles métiers (fin de stock.) et une stratégie de vente (offres spéciales, top ventes…).

La version custom plus puissante propose une personnalisation du moteur de recherche, une recherche instantanée et une autocorrection des requêtes [30].

FIGURE II.6 - Moteur Ezako

II.9 Conclusion

Trouver des informations sur un grand site peut être un processus long et difficile. Un système de recommandation peut aider l'utilisateur à trouver des informations en leur fournissant des suggestions personnalisées.

Dans ce chapitre, nous avons présenté tout d'abord la définition de la recommandation en général et ses objectifs principaux. Ensuite, nous avons souligné les étapes de construction d'un système de recommandation, comme nous avons convenu qu'un système de recommandation a pour objectif de fournir à un utilisateur des ressources pertinentes en fonction de ses préférences. Ce dernier voit ainsi réduit son temps de recherche mais reçoit également des suggestions de la part du système auxquelles il n'aurait pas spontanément prêté attention. Aussi nous avons cité les différentes classifications des systèmes de recommandation et enfin nous avons donné quelques exemples de systèmes de recommandation du domaine de l'e-commerce.

Dans le chapitre suivant nous allons essayer d'expliquer les principales techniques visant à produire des systèmes de recommandation.

CHAPITRE III

Techniques de recommandation

CHAPITRE III

III.1 Introduction

Les techniques des systèmes de recommandation fonctionnent avec deux types de données qui sont :

(i) Interactions utilisateur-article : comme le comportement d'achat ou d'évaluations.

(ii) Les informations d'attribut sur les utilisateurs et les articles : tels que les profils textuels ou les mots-clés pertinents.

Les méthodes qui utilisent le premier type sont appelées méthodes de filtrage collaboratif, alors que les méthodes qui utilisent le deuxième type sont appelées méthodes de recommandation basées sur le contenu [28].

Une autre technique appelé recommandation basée sur la connaissance, est fondée sur les exigences de l'utilisateur spécifiées explicitement. Au lieu d'utiliser les historiques des notations ou les données d'achat, des bases de connaissances externes sont utilisées pour créer la recommandation.

Certains systèmes de recommandation combinent ces différents aspects pour créer des systèmes hybrides. Les systèmes hybrides peuvent combiner les forces de différents types de systèmes de recommandation pour créer des techniques qui peuvent être robuste dans une grande variété de paramètres.

Dans ce chapitre, nous allons discuter brièvement de ces techniques.

III.2 Filtrage basé sur le contenu

Le filtrage basé sur le contenu (Content-Based Filtering), qui est une évolution générale des études sur le filtrage d'information, s'appuie sur le contenu des documents (thèmes abordés) pour les comparer à un profil lui-même constitué de thèmes. Chaque utilisateur du système possède alors un profil qui décrit des centres d'intérêts. Par exemple, le profil peut contenir une liste des thèmes ou préférences que l'utilisateur aime bien ou qu'il n'aime pas. Lors de l'arrivée d'un nouveau document, le système compare le descriptif du document avec le profil de l'utilisateur pour prédire l'utilité de ce document pour cet utilisateur. Ce dernier permet d'améliorer la qualité des recommandations au cours du temps.

L'avantage des systèmes de filtrage cognitifs basés contenu est qu'ils permettent d'associer des documents à un profil utilisateur. Notamment, en utilisant des techniques d'indexation et d'intelligence artificielle. L'utilisateur est indépendant des autres ce qui lui permet d'avoir des recommandations même s'il est le seul utilisateur du système [10]. Afin de recommander par exemple des films a un utilisateur, le système analyse les corrélations entre ces films et les films consultés antérieurement par cet utilisateur.

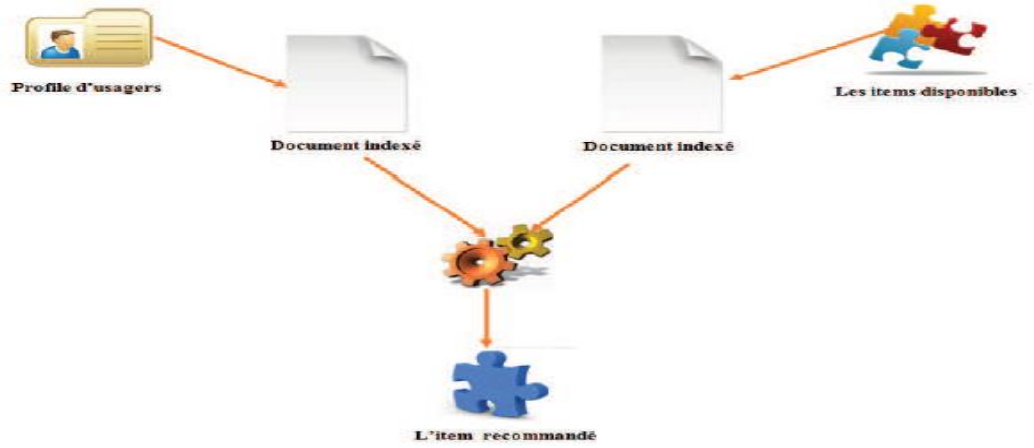


FIGURE III.1 - Recommandation basé sur le contenu.

Ces corrélations sont évaluées en considérant des attributs comme le titre et le genre. De ce fait, parmi ces films, ceux qui seront recommandés à l'utilisateur sont les plus similaires (En termes d'attribut) aux films consultés par cet utilisateur. Cependant, ce type de systèmes présente certaines limitations.

- L'effet "entonnoir" : les besoins de l'utilisateur sont de plus en plus spécifiques, ce qui l'empêche d'avoir une diversité de sujets. Même pire, un nouvel axe de recherche dans un domaine bien précis peut ne pas être pris en compte car il ne fait pas partie du profil explicite de l'utilisateur.
- Filtrage basé sur le critère thématique uniquement, absence d'autres facteurs comme la qualité scientifique, le public visé, l'intérêt porté par l'utilisateur, etc.
- Les difficultés à recommander des documents multimédia (images, vidéos, etc.) et ceci à cause de la difficulté à indexer ce type de documents, c'est en fait la même problématique dont souffrent les systèmes de recherche.
- Problème de démarrage à froid : un nouvel utilisateur du système éprouve des difficultés à exprimer son profil en spécifiant des thèmes qui l'intéressent. Ceci malgré les techniques d'apprentissage ou l'utilisateur fournit des textes exemples.

III.2.1 Exemple de systèmes de recommandation basés sur le contenu

Un système de recommandation basé sur le contenu a été proposé par Chandrasekaran et al [13] 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.

III.3 Filtrage démographique

C'est une recommandation simple qui propose des items par rapport au profil démographique d'usager (figure III.2). Elle comporte à partager les usagers en plusieurs classes ou groupes par rapport aux informations démographiques telles que le sexe, l'âge, la profession, la localisation, la langue, le pays, etc. Le principe de cette approche est que deux utilisateurs ayant évolué dans un environnement similaire partagent des goûts communs que deux utilisateurs ayant évolué dans des environnements différents et ne partageant donc pas les mêmes codes

Ce qui conduit à limiter la performance de ce type du filtrage du fait que ces systèmes ne fonctionnent pas bien que pour un utilisateur avec beaucoup de proches voisins (similaires). De plus, les informations démographiques doivent être accumulées.

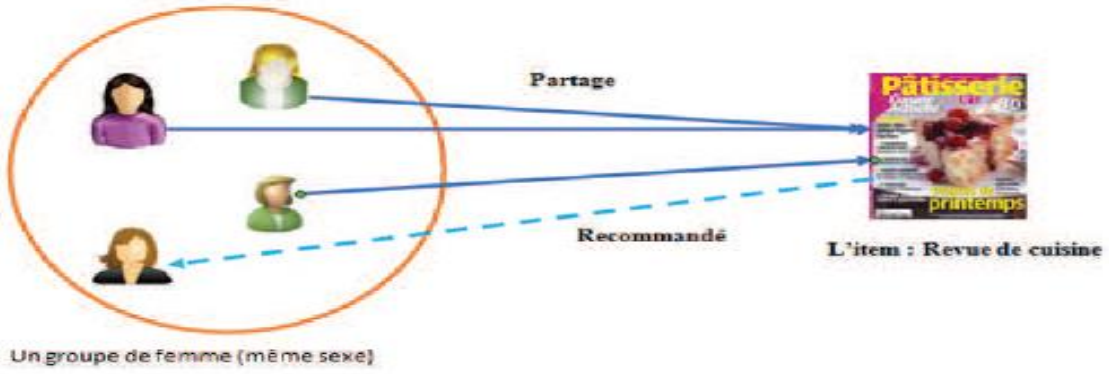


FIGURE III.2 - Recommandation démographique.

III.3.1 Exemple de système de recommandation basé sur le filtrage démographique

Le système de recommandation ALAMBIC: ALAMBIC [8] est un système de recommandation basé sur le filtrage démographique, qui génère des recommandations en se basant sur les commentaires précédents des utilisateurs sur les mêmes caractéristiques démographiques (l'âge, le sexe, le niveau d'éducation, la richesse, la situation géographique, etc.). Ce système obtient de manière adéquate les objectifs de la protection de la vie privée dans les systèmes de recommandation de l'e-commerce, et il est basé sur une partie du tiers semi-confiant dans lequel les utilisateurs n'ont besoin que d'une confiance limitée. L'une des principales originalités de ce système est de séparer les données de l'utilisateur entre cette partie et le fournisseur de services, car il ne peut pas tirer les informations sensibles de la part de l'utilisateur seulement.

III.4 Filtrage basé connaissances

Les SRs basés sur la connaissance font des suggestions en se basant sur des inférences des besoins et des préférences d'un utilisateur. Dans un certain sens, toutes les techniques de recommandation peuvent être décrites comme faisant un certain type d'inférence. Cette classe des systèmes s'inspire de la recherche en raisonnement à base de cas (Case-Based Reasoning CBR) [19].

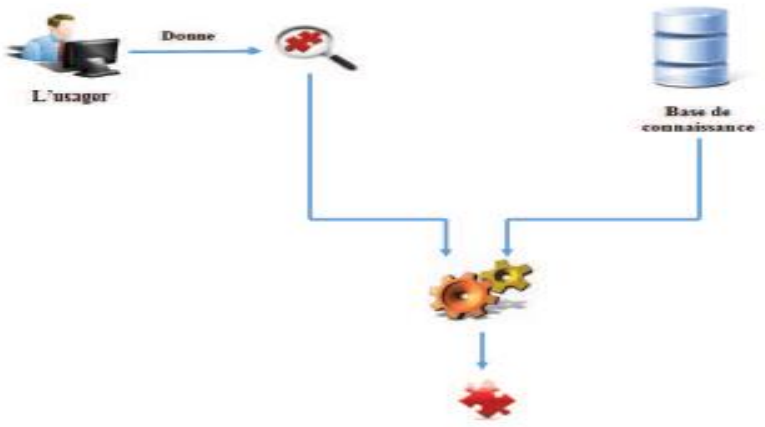


FIGURE III.3 – Recommandation basé connaissances.

Contrairement aux deux premières méthodes qui utilisent des algorithmes d'apprentissage, le filtrage basé connaissance exploite des connaissances du domaine et fait des inférences sur les besoins et préférences des utilisateurs, ce qui rend cette méthode plus appropriée dans certains cas que les autres surtout quand le système souffre d'un manque d'informations sur les utilisateurs ou sur les items (le problème de démarrage à froid) du fait qu'elle n'est pas sensible au manque d'informations.

Cependant, les approches dans ces systèmes se distinguent par le fait d'avoir une connaissance fonctionnelle, i.e. elles possèdent une connaissance décrivant comment un item particulier satisfait le besoin d'un utilisateur [12]. Par exemple, l'utilisateur peut recommander tout simplement un thé et le système va connaître que le thé fait partie de la cuisine asiatique.

Le profil de l'utilisateur peut être n'importe quelle structure de connaissances qui supporte cette inférence. Dans le cas le plus simple, comme dans Google, elle peut être simplement la requête que l'utilisateur a formulée. Dans d'autres cas, elle peut être une représentation plus détaillée des besoins des utilisateurs [19]. Ils existent trois types de connaissances qui sont impliqués dans un tel système :

- **Catalogue des connaissances** : représente les connaissances sur les items à recommander et leurs caractéristiques. Par exemple, le système de recommandation Entree devrait contenir la connaissance qui indique que la cuisine « Thai » est une sorte de cuisine « asiatique ».
- **Connaissance fonctionnelle :** le système doit être capable de relier entre les besoins de l'utilisateur et l'item qui pourrait satisfaire ces besoins. Par exemple, Entree sait que le besoin d'une place pour un dîner romantique pourrait être satisfait par un restaurant qui a comme description « Calme, avec une vue sur mer ».
- **Connaissance sur l'utilisateur :** le système doit avoir des connaissances sur l'utilisateur qui pourraient prendre la forme d'informations démographiques générales ou des informations spécifiques sur les nécessités, pour lesquels une recommandation est demandée. De ces types de connaissances, la dernière est la plus difficile, comme il est, dans le pire des cas, une instance du problème général de la modélisation de l'utilisateur [33]. La connaissance utilisée dans un tel système peut prendre aussi différentes formes. Google par exemple utilise une information sur les liens entre les pages Web pour inférer la valeur de la popularité et d'authenticité.

III.4.1 Exemple de systèmes de recommandation basés connaissances

Le système de recommandation EntreeC [12]: Le système EntreeC est un système de recommandation des restaurants, et il a été basé sur le système de recommandation de restaurant.

Le système génère ses recommandations par la recherche de restaurants dans une nouvelle ville similaire aux restaurants que l'utilisateur connaît et aime. Le système permet aux utilisateurs de naviguer, en indiquant leurs préférences par rapport à un restaurant donné, ainsi il affine leurs critères de recherche.

III.5 Filtrage collaboratif

Une des premières techniques utilisées et qui reste encore aujourd'hui parmi les plus simples et efficaces est le filtrage collaboratif.

Cette technique est basée sur le partage d'opinions entre les utilisateurs. Bien que le terme n'ait été introduit que depuis moins de deux décennies, il implémente le principe du " bouche-à-oreille" pratiqué depuis toujours par les humains pour se construire une opinion sur un produit ou un service [21].

Ce procédé en trois étapes :

1. Commence par la collecte d'information sur les utilisateurs.
2. Puis on forme une matrice afin de calculer des associations.
3. Finalement nous sommes en mesure de faire une recommandation avec un niveau de confiance assez élevé [7].

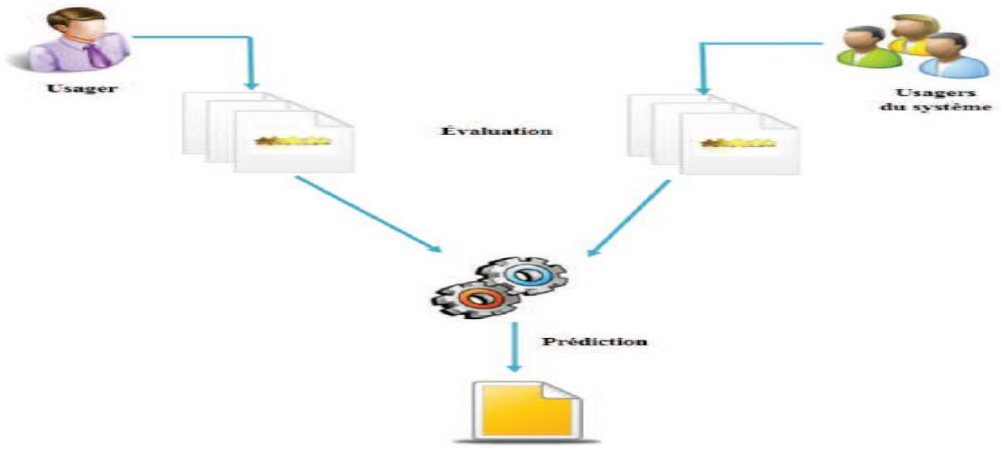


FIGURE III.4 - Recommandation basé sur le filtrage collaboratif

La figure (III.4) représente un tableau de films avec sur un axe les utilisateurs d'un même système (ex : un groupe d'amis sur MovieLens) et sur un autre les films. Chaque cellule de la matrice contient l'avis donné par un utilisateur pour un film, la cellule vide signifie qu'il n'a pas d'avis particulier sur ce film. Afin de prédire si Illyes apprécierait le film "Harry Potter" et probablement lui recommander ce film, on compare les votes de llyes à ceux des autres utilisateurs choisis. On peut alors voir que Illyes et Imen ont des votes identiques, et que Imen n'a pas aimé le film «Harry Potter », on pourrait alors prédire que Illyes n'aimera pas aussi ce film et de ne lui pas faire cette suggestion.

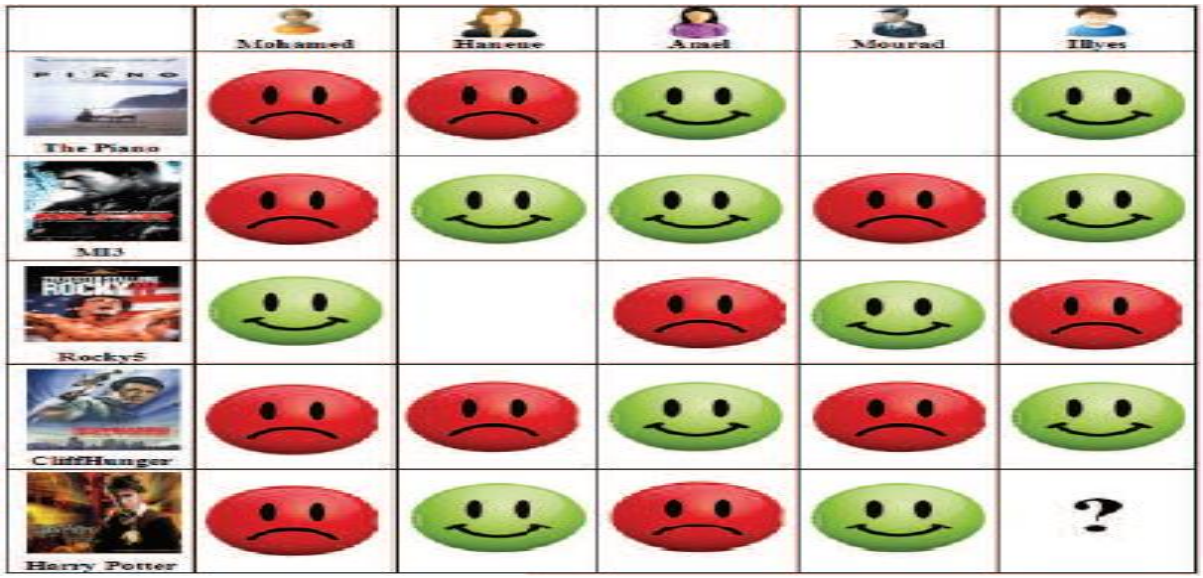


FIGURE III.5 - Exemple de recommandation base sur le filtrage collaboratif.

Principaux algorithmes du filtrage collaboratif

Les auteurs de Su and Khoshgoftaar [32] ont regroupé les méthodes de filtrage collaboratif en deux classes : les algorithmes basés sur la mémoire et les algorithmes basés sur un modèle.

Les algorithmes de filtrage collaboratif basés sur la mémoire, appelés également basés sur des heuristiques ou basés sur les voisins [16] utilisent les votes des utilisateurs stockés en mémoire pour faire de la prédiction. Les algorithmes basés sur un modèle construisent en offline une image réduite de la matrice des votes dans un objectif de réduire la complexité des calculs.

III.5.1 Méthodes basés sur le modèle :

Le premier type d'algorithmes, est comme le nom l'indique basés sur des modèles, supposés réduire la complexité. Ces modèles peuvent être probabilistes et utilisent l'espérance de l'évaluation pour calculer la prédiction. Comme ils peuvent être basés sur des classificateurs permettant de créer des classes pour réduire la complexité.

Modèle de Clustering

Les méthodes de Clustering permettent de limiter le nombre d'individus considérés dans le calcul de la prédiction. Le temps de traitement sera donc plus court et les résultats seront potentiellement plus pertinents puisque les observations porteront sur un groupe le plus proche de l'utilisateur actif. Autrement dit, au lieu de consulter l'ensemble de la population, nous estimons la préférence d'un groupe de personnes ayant les mêmes goûts que l'utilisateur.

K-Means

La méthode des plus proches voisins K-Means consiste dans un premier temps à choisir aléatoirement k centres dans l'espace de représentation utilisateurs/ressources. Ensuite, chaque utilisateur est mis dans le cluster du centre le plus proche. Quand les groupes de personnes sont formés, nous recalculons la position des centres pour chaque cluster et réitérons l'opération depuis le début jusqu'à obtenir un état stable où les centres ne bougent plus. L'algorithme est certes simple à mettre en œuvre mais présente certains inconvénients, lié à la criticité du choix des clusters initiaux, pouvant influencer sur la qualité de la classification.

RecTree

RecTree est un algorithme de filtrage collaboratif appelé l'arbre de recommandation (Recommandation Tree). L'algorithme RecTree fractionne les données dans des cliques d'utilisateurs approximativement semblables. L'objectif est de maximiser les similarités entre les membres d'une même clique et à minimiser celles entre les membres de deux cliques différentes.

III.5.2 Filtrage collaboratif basé sur la mémoire :

Contrairement aux approches basées sur un modèle, les algorithmes de filtrage collaboratif basés sur la mémoire ne nécessitent pas de phases d'apprentissage coûteuses à renouveler fréquemment. Toutefois, la recommandation basée sur les voisins est plus coûteuse en raison du calcul des similarités entre les items (ou les utilisateurs). Une solution consiste à précalculer les similarités, et à ne conserver que les k plus proches voisins. Le stockage des k plus proches voisins ne nécessitant pas un espace important, ce qui permet à de telles approches de passer l'échelle même pour des applications ayant des millions d'utilisateurs et d'items [16].

Les méthodes basées sur la mémoire maintiennent une base de données des votes de tous les utilisateurs. Un score de similarité [7] est déterminé entre l'utilisateur courant et chacun des autres membres de la base. Chaque prédiction entraîne ensuite un calcul sur l'ensemble de cette source de données.

Ces techniques basées sur la mémoire sont divisées en deux analyses : FC à base d'items, et FC à base d'utilisateurs, nous allons expliquer chacune de ces deux analyses, aussi nous présentons les principales méthodes utilisées par la recommandation collaborative pour le calcul de la similarité entre deux objets (items ou utilisateurs).

En plus des notations définies dans ce chapitre, on note par $\overline{v_u}$ la moyenne des votes de l'utilisateur U sur les items qu'il a notés (formule III.1) et par ū la moyenne des votes de l'item i (formule III.2).

$$\overline{v_u} = \frac{\sum u \epsilon U_i v_i}{|U_i|} \quad \text{(III.1)}$$

$$\overline{v_i} = \frac{\sum i \epsilon I_i v_i}{|I_i|} \quad \text{(III.2)}$$

On note également par sim(u, v) la fonction mesurant la similarité entre les deux utilisateurs u et v (resp. sim(i, j) celle mesurant la similarité entre les deux items i et j). On définit $I_{uw} = I_u \cap I_w$ l'ensemble des items notés à la fois par les utilisateurs u et w, et de façon équivalente

$U_{ij} = U_i \cap U_j$ l'ensemble des utilisateurs ayant notés à la fois les items i et j. Enfin, on note par voisins(u) l'ensemble des utilisateurs $w \in U$ définis comme les voisins de l'utilisateur u.

A : L'analyse basée sur l'utilisateur :

Cette technique de recommandation se base sur le principe de trouver des utilisateurs similaires à l'utilisateur courant puis d'utiliser leurs évaluations pour prédire ce que l'utilisateur courant peut aimer. Les utilisateurs similaires à l'utilisateur courant, appelés voisins de cet utilisateur, sont ceux qui ont un comportement d'évaluation similaire à celui de l'utilisateur courant.

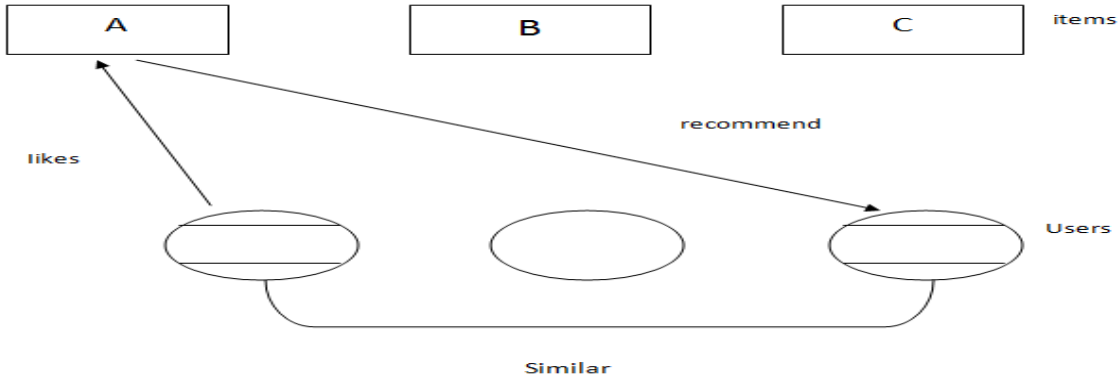


FIGURE III.6- Analyse basée sur l'utilisateur

Cette approche basée sur les utilisateurs (user-based) a été introduite pour la première fois dans le système GroupLens [27], son principe de fonctionnement se résume ainsi :

1. Déterminer les voisins de l'utilisateur courant en calculant sa similarité avec les autres utilisateurs de la base.

2. Calculer la prédiction du vote de l'utilisateur courant u_a pour un item $I \in II_{u_a}$ candidat à la recommandation, en analysant les votes de ses voisins sur ce même item.

Calcul de la similarité : la similarité entre deux utilisateurs u et w peut être mesurée en utilisant, soit le coefficient de corrélation de Pearson (formule (III.10)), soit le Cosinus (formule (III.8)).

Calcul de la prédiction : pour le calcul de la prédiction, le plus naïf est de calculer la moyenne des votes de tous les voisins de l'utilisateur courant u_a comme l'illustre l'équation (III.3).

$$pred(u_a, i) = \frac{\sum_{u \epsilon voisins(u_a)} v_{ui}}{|(u_a) \cap u_i|} \quad \text{(III.3)}$$

La formule donnée en (III.3) est dite naïve parce qu'elle considère tous les voisins au même pied d'égalité et ne tient pas compte du fait que certains voisins soient plus proches que d'autres de l'utilisateur courant u_a . Afin de tenir compte de cette diversité, dans la somme de la formule (III.3), on pondère le vote de chaque voisin par la valeur de sa similarité avec l'utilisateur courant. Ainsi, les votes des plus proches voisins auront un poids plus important que celui des voisins les moins proches. Vu que la somme des poids de tous les voisins n'est

pas égale à 1, et afin de normaliser la valeur de la prédiction, cette somme est divisée par la somme des similarités de l'utilisateur courant avec ses voisins.

L'équation (III.4) donne la formule correspondante pour le calcul de la prédiction.

$$\boldsymbol{pred(u_a, i)} = \frac{\sum_{\boldsymbol{u \epsilon voisins(u_a) \cap u_i}} sim(u_a,u) v_{ui}}{\sum_{\boldsymbol{u \epsilon voisins(u_a) \cap u_i}} |sim(u_a,u)|} \quad \text{(III.4)}$$

Par ailleurs, tous les utilisateurs sont différents dans leurs façons d'évaluer un item. En effet, il existe des utilisateurs qui notent large en affectant la valeur de 5 sur une échelle de 1 à 5 pour un item qu'ils jugent satisfaisant alors que d'autres, qui ont tendance à noter de façon plus stricte, attribueront la valeur 3 à un item qu'ils jugent satisfaisant. Pour compenser la variation dans le jugement des utilisateurs, le vote de chaque utilisateur u est ajusté par la moyenne de ses votes Uz. L'équation (III.5) donne la formule adoptée par les auteurs de pour le calcul de la prédiction.

$$\boldsymbol{pred(u_a, i)} = \frac{\sum_{\boldsymbol{u \epsilon voisins(u_a) \cap u_i}} sim(u_a,u)(v_{ui} - v_u)}{\sum_{\boldsymbol{u \epsilon voisins(u_a) \cap u_i}} |sim(u_a,u)|} \quad \text{(III.5)}$$

Dans l'algorithme initial tel qu'implanté dans le GroupLens système [20], tous les voisins sont pris en compte dans le calcul de la prédiction. Il a été démontré par la suite que la prise en compte des k plus proches voisins améliore, non seulement la pertinence des prédictions, mais également l'efficacité de l'algorithme [11].

Recommandation Top-N basée sur les utilisateurs :

Les algorithmes de recommandation Top-N sont généralement utilisés lorsque le système ne dispose pas de votes numériques définis sur une échelle de valeur mais de votes binaires (aime, n'aime pas) ou unaire (a acheté, a consulté). Comme pour la prédiction basée sur les utilisateurs, une première étape consiste à définir les plus proches voisins de l'utilisateur courant en utilisant soit le coefficient de Pearson soit le Cosinus. Une fois les plus proches voisins de l'utilisateur courant u_a identifiés, une deuxième étape consiste à déterminer pour chaque voisin w, la liste L_w des items pertinents (items achetés). Les items sont par la suite triés selon la fréquence de leur présence dans les listes de tous les voisins. Le Top-N des items les plus fréquents seront alors recommandés à u_a . Les algorithmes de recommandation Top-N basés sur les utilisateurs souffrent du problème de passage à l'échelle et du problème de performance pour les applications en temps réel.

B : L'analyse basée sur l'item :

Les approches dans cette classe appliquent les calculs statistiques sur les items. L'idée fondamentale est de proposer à l'utilisateur des items similaires à ceux qu'il a aimés [9].

Le filtrage collaboratif basé sur les utilisateurs souffre de problèmes de montée en charge si la base d'utilisateurs est importante. La technique du filtrage collaboratif basé sur les items (Sarwar et al. 2001) a été développée pour répondre à cette problématique. Cette technique est utilisée lorsqu'il s'agit de trouver des items similaires à l'item courant. Cette technique utilise les similarités entre les patterns des évaluations des items. Si deux items ont tendance à avoir les mêmes utilisateurs qui les aiment et les mêmes utilisateurs qui ne les aiment pas, alors ces items sont similaires. Les utilisateurs ont des préférences similaires pour les items similaires.

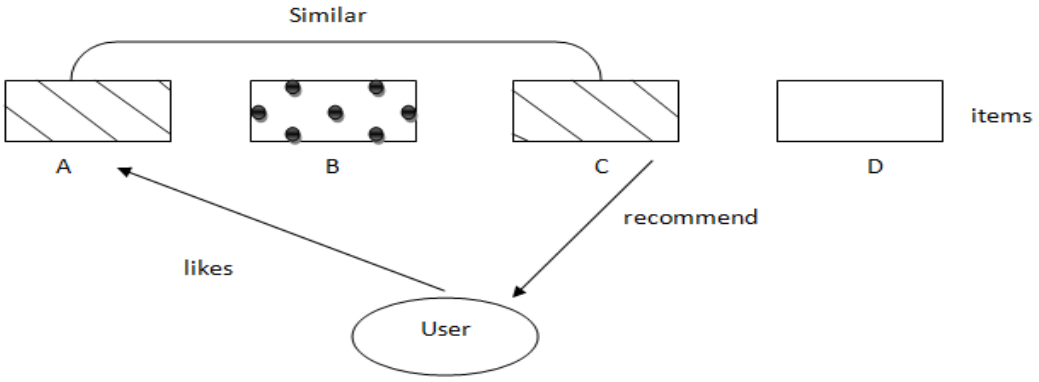


FIGURE III.7 - Analyse basée sur l'item

Cette approche basée sur les items (item-based) a été introduite par Sarwar et al [29]. La prédiction du vote de l'utilisateur u pour un item candidat $i \in II_u$ est calculée à partir de ses évaluations pour les items voisins de i. Son principe de fonctionnement est le suivant :

1. Pour tout item i de I, déterminer les voisins les plus proches en calculant sa similarité avec les autres items.

2. Calculer la prédiction du vote de l'utilisateur courant u a pour l'item i en analysant les votes de ce dernier sur les voisins de i.

Détermination des voisins : la similarité entre deux items i et j peut être calculée en utilisant soit le Cosinus (voir formule (III.8)), soit le coefficient de Pearson (voir formule (III.11)), soit le cosinus ajusté (voir formule (III.12)). Cependant, une étude expérimentale menée par les auteurs de Sarwar et al comparant les trois mesures, a montré que le Cosinus ajusté reste le plus performant en termes de pertinence de prédiction.

Calcul de la prédiction : la prédiction du vote de l'utilisateur courant u_a pour un item candidat à la recommandation i revient à calculer une moyenne pondérée de ses votes sur l'ensemble des items similaires à i. Chaque vote $v_{u_a,j}$ est pondéré par la similarité de l'item j avec l'item i. Afin d'avoir une prédiction dans le même intervalle de valeurs que les votes, la prédiction est divisée par la somme des similarités. L'ajustement du vote est inutile dans ce cas puisqu'il s'agit du même utilisateur. L'équation (III.6) donne la formule de calcul de la prédiction telle que donnée dans Sarwar et al.

$$\boldsymbol{pred(u_a, i)} = \frac{\sum_{j \epsilon I_{v_a}} sim(i,j)(v_{uj})}{\sum_{j \epsilon I_{v_a}} |sim(i,j)|} \quad \text{(III.6)}$$

Les algorithmes basés sur les items sont moins sensibles aux données manquantes et plus performants en termes d'efficacité (complexité de calcul) que les algorithmes basés sur les utilisateurs. Cependant, et malgré leur lenteur, les expérimentations ont montré que les algorithmes basés sur les utilisateurs sont plus performants en terme de précision [25].

Recommandation Top-N basée sur les items :

Le système de recommandation d'Amazon utilise un algorithme de recommandation Top-N basé sur les items. Après une opération d'achat, le système affiche à l'utilisateur un message de la forme "les clients ayant acheté ce produit ont également consulté ces produits" suivi d'une liste de recommandations. Les algorithmes Top-N basés sur les items ont été développés pour contrecarrer les problèmes de performance et de passage à l'échelle dont souffraient les algorithmes Top-N basés sur les utilisateurs. Le principe de fonctionnement de tels algorithmes suit deux étapes :

- Une première étape consiste à calculer les voisins de chaque item de I. Une fois les k plus proches voisins, voisins k (j) de chaque item j identifiés,
- une deuxième étape consiste à construire la liste des items à recommander à l'utilisateur courant u_a.

Soit I_{u_a} la liste des items achetés par u_a, pour chaque item $j \in II_{u_a}$ candidat à la recommandation, on calcule la somme de ses similarités, $x_{ua}(j) = \sum_{i \in I_{ua} voisins_k(j)}(sim(j,i))$avec ses k plus proches voisins ($voisins_k(j)$) parmi les items achetés par u_a. La liste des Top-N items à recommander à l'utilisateur courant est constituée des N items ayant les plus grandes valeurs de$x_{ua}(j)$.

Calcul de la similarité

Le calcul de la similarité a pour objectif de déterminer les voisins d'un utilisateur (ou d'un item). Dans les algorithmes basés sur les voisins tels que définis dans cette section, la similarité entre deux utilisateurs ou items est calculée en mesurant la corrélation existante entre leurs votes. C'est pour cette raison que ces approches sont dites également approches basées sur la corrélation des votes des voisins (neighborhood approches based on rating correlation) [15].

Cosinus : une mesure de similarité entre deux objets a et b, très utilisée en recherche et filtrage d'information, consiste à représenter les deux objets par deux vecteurs Ta et de mesurer le cosinus de l'angle formé par les deux vecteurs.

$$sim(a,b) = \cos(\overrightarrow{x_a}, \overrightarrow{x_b}) = \frac{\overrightarrow{x_a}.\overrightarrow{x_b}}{\|\overrightarrow{x_a}\|\|\overrightarrow{x_b}\|} \text{ (III.7)}$$

Dans le cas du filtrage collaboratif, chaque utilisateur u est représenté par un vecteur $x_u \in R|I|$, où $x_{ui} = v_{ui}$si l'utilisateur u a évalué l'item i. Étant donné les valeurs manquantes dans la matrice des votes, le cosinus est calculé sur l'ensemble des items notés par les deux utilisateurs. La similarité entre deux utilisateurs u et w est alors donnée par la formule (III.8).

$$sim(u,w) = \cos(\overrightarrow{x_u}, \overrightarrow{x_w}) = \frac{\sum_{i \in I_{uw}} V_{ui} V_{wi}}{\sqrt{\sum_{i \in I_{uw}} {V_{ui}}^2} \sqrt{\sum_{i \in I_{uw}} {V_{wi}}^2}} \quad \text{(III.8)}$$

La formule (III.9) applique le cosinus pour calculer la similarité entre deux items. En fait, il suffit de remplacer dans l'équation (III.8) les utilisateurs par leurs équivalents en items.

$$sim(i,j) = \cos(\overrightarrow{x_i}, \overrightarrow{x_j}) = \frac{\sum_{u \in U_{ij}} V_{ui} V_{uj}}{\sqrt{\sum_{i \in U_{ui}} {V_{ui}}^2} \sqrt{\sum_{i \in I_{uj}} {V_{uj}}^2}} \quad \text{(III.9)}$$

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

Coefficient de corrélation de Pearson :

Utilisé par les auteurs du système GroupLens [27] pour calculer la similarité entre deux utilisateurs u et w de U. Le coefficient de corrélation de Pearson mesure la liaison linéaire entre deux variables numériques en calculant le rapport entre leur covariance et le produit non nul de leur écart type. Il permet ainsi de mesurer la similarité en supprimant l'inconvénient de la variation des votes. L'équation (III.10) donne la formule de calcul de la mesure de corrélation entre deux utilisateurs u et w. En raison des données manquantes, seuls les items notés à la fois par u et w sont pris en compte.

$$sim(u,w) = Pearson(u,w) = \frac{\sum_{i \in I_{uw}} (v_{ui} \bar{v}_u)(v_{wi} \bar{v}_w)}{\sqrt{\sum_{i \in I_{uw}} (v_{ui} \bar{v}_u)} \sqrt{\sum_{i \in I_{uw}} (v_{wi} \bar{v}_w)}} \quad \text{(III.10)}$$

Le coefficient de corrélation de Pearson varié entre -1 et 1. Une valeur égale à 1 indique que les utilisateurs partagent exactement les mêmes goûts, une valeur de -1 indique qu'ils ont des goûts totalement opposés. Le coefficient de Pearson est généralement reconnu comme étant la meilleure mesure pour calculer la similarité entre les utilisateurs. Il peut être également utilisé pour mesurer la corrélation entre deux items. L'équation (III.12) donne le coefficient de Pearson entre deux items i et j.

$$sim(i,j) = Pearson(i,j) = \frac{\sum_{u \in U_{ij}} (v_{ui} \bar{v}_j)(v_{uj} \bar{v}_j)}{\sqrt{\sum_{u \in U_{uj}} (v_{uj} \bar{v}_u)} \sqrt{\sum_{u \in U_{uj}} (v_{ij} \bar{v}_j)}} \quad \text{(III.11)}$$

Le cosinus ajusté (Adjusted cosine) : lorsqu'on applique le coefficient de Pearson pour mesurer la similarité entre deux items (voir formule (III.11)), les votes d'un même utilisateur sont centrés par rapport à la moyenne de ses votes. Or, la variation de notation pour un même utilisateur n'est pas aussi importante que la variation entre les différents utilisateurs. C'est pour cette raison qu'il est plus intéressant, lors du calcul de la similarité entre deux items, d'ajuster les votes par rapport à la moyenne des votes des utilisateurs plutôt que par rapport à la moyenne des votes des items. C'est ce que fait le Cosinus ajusté.

	Item 1	Item 2	Item 3	Item 4
Jean	-	2	7	8
Marie	4	1	-	7
Christian	3	8	-	4

TABLE III.1 – Exemple de matrice d'évaluation

Introduit par Sarwar et al c'est la plus populaire et la plus pertinente des mesures utilisées dans le calcul de la similarité entre deux items dans les algorithmes de filtrage collaboratifs. Le cosinus ajusté est en fait une amélioration du cosinus en ajustant la valeur des votes d'un utilisateur par rapport à la moyenne de ses votes.

$$sim(i,j) = CosinusAjusté(i,j) = \frac{\sum_{u\in U_{ij}}(v_{ui}\bar{v}_u)(v_{uj}\bar{v}_u)}{\sqrt{\sum_{u\in U_{uj}}(v_{ui}\bar{v}_u)}\sqrt{\sum_{u\in U_{uj}}(v_{ij}\bar{v}_u)}} \quad \text{(III.12)}$$

(III.12) Comme le coefficient de Pearson, le Cosinus ajusté varient entre -1 et 1. Une valeur égale à 1 indique que les deux items sont identiques, une valeur égale a -1 indique qu'ils sont opposés.

Exemple

La table (III.1) présente la correspondance entre les utilisateurs et les items.

On suppose que les appréciations vont de 0 à 10 (très mauvais à excellent) et que le tiret « - » représente l'absence d'évaluation. En effet, la plupart des utilisateurs ne vont noter qu'un très petit nombre d'items (par exemple Amazon vend des millions de livres, un lecteur même assidu ne peut en noter au plus que quelques milliers). Dans de nombreux cas, la matrice sera donc creuse, voire très creuse. Supposons que nous voulions maintenant utiliser cette matrice pour calculer une similarité entre utilisateurs puis pour induire $Pred(u_l, i_j)$ pour un utilisateur u_l et un item i_j donnés.

1. Approche basée sur l'utilisateur

Calcul de la similarité

Dans cet exemple on va appliquer la similarité de Pearson. Selon cette mesure, la similarité entre Jean et Marie est :

$$sim(Jean, Marie) = \frac{(2-5)(1-4)+(8-5)(7-4)}{\sqrt{(2-5)^2+(8-5)^2}\sqrt{(1-4)^2+(2-4)^2}} = \frac{18}{18} = 1$$

Avec $X_{Jean} = (2 + 8)/2 = 5$ et $X_{Marie} = (1 + 7)/2 = 4$ quand on prend les articles notés en commun. Entre Jean et Christian, elle est : Sim (Jean, Christian) = $\frac{-12}{\sqrt{18}\sqrt{8}}$= 12 /(3 · 2 · 2) = -1

Et entre Marie et Christian : Sim (Marie, Christian) =$\frac{-12}{\sqrt{18}\sqrt{8}} \approx$ -0.756 Il apparaît ainsi que Jean et Marie sont positivement corrélés, tandis que Jean et Christian, de même que Marie et Christian, sont des paires corrélées négativement. Il faut noter que s'il peut sembler étrange que Jean et Christian soient parfaitement négativement corrélés, il faut prendre en compte leur moyenne sur les composantes 2 et 4 à savoir 5 pour Jean et 6 pour Christian. Par rapport à ces moyennes, les notes de Jean pour les items 2 et 4 sont -3 et +3, tandis que pour Christian elles sont 2 et -2. Il y a bien tendance exactement inverse.

Calcul d'une recommandation

Supposons que nous voulions prédire la note que donnerait Jean à l'item 1 et que nous prenions Marie et Christian comme voisins.

$$Pred\ (Jean, item\ 1) = \frac{17}{3} + \frac{(1.(4-4)) + (-1.(3-5))}{1+1} = 5.67 + 1 \approx 6.67$$

Intuitivement, puisque Christian n'a pas aimé l'item 1, et qu'il est négativement corrélé avec Jean, alors cela devrait amener à penser que Jean va plutôt aimer item 1, d'où une valeur

accordée à item 1 supérieure à la moyenne de Jean. L'évaluation de Marie, quant à elle, ne modifie rien car elle évalue l'item 1 à sa moyenne : 4.

2. Approche basée sur l'item ;

Supposons que nous voulions prédire la note que donnerait Jean à l'item 1 et que nous prenions Item 2 et Item 4 comme voisins (Item 3 ne peut pas être voisin d'Item 1 car il ne partage aucune composante utilisateur). En utilisant la mesure de corrélation de Pearson, on trouve que Sim (Item 1, Item 2) = 1 et

$$Pred(Jean, Item\ 1) = \frac{4+3}{2} + \frac{(-1.(2-11/3) + (1.(8-19/3))}{1+1} = 3.5 + 1.67 = 5.17$$

On observe que, en utilisant ces formules de similarité et de combinaison, le résultat n'est pas le même que dans l'approche centrée utilisateur.

Les points forts des algorithmes basés sur les utilisateurs

Les algorithmes de filtrage collaboratif basé sur les utilisateurs ont de nombreux avantages parmi lesquels on peut citer :

La simplicité : une telle approche est intuitive et simple à implémenter. Dans sa forme la plus simple, seul le nombre k de voisins à considérer est à définir.

Effet de surprise : (serendipity) : serendipity en anglais désigne l'effet de surprise que peut ressentir un utilisateur en recevant une recommandation pertinente qu'il n'aurait jamais soupçonné intéressante pour lui. Des études Good et all [24] ont montré que "l'effet de surprise" est un facteur important pour mesurer la satisfaction des utilisateurs. Les algorithmes basés sur les utilisateurs permettent de faire des recommandations à effet de surprise. En effet, si un utilisateur A est proche d'un utilisateur B du fait qu'il ne regarde que des comédies, et si B apprécie un film d'un autre genre, ce film peut être recommandé à A du fait de sa proximité avec B.

Les points forts des algorithmes basés sur les items

Les algorithmes de filtrage collaboratif basé sur les items ont de nombreux avantages parmi lesquels on peut citer :

Explication : de récents travaux ont montré la nécessiter d'expliquer les raisons des recommandations proposées aux utilisateurs pour les inciter à ajuster leur profil. Tel est le cas par exemple des sites commerciaux qui affichent à la suite de la consultation d'un produit par l'utilisateur courant, un message de la forme "les clients ayant consulté ce produit ont également consulté ces produits". Les approches basées sur les items permettent aisément de fournir des justifications. Ainsi, le système de recommandation de livres de Amazon, qui utilise un algorithme basé sur les items, affiche comme explication "Ces recommandations sont basées sur les articles que vous possédez et plus encore". L'utilisateur peut alors affiner son profil en fournissant une liste plus complète des livres qu'il possède.

Stabilité : des observations sur les sites de e-commerce ont montré que ces approches ne sont pas très sensibles à l'ajout constant d'utilisateurs, d'items ou de votes [16]. Lorsqu'un nouvel utilisateur évalue un certain nombre d'items, seule sa similarité avec les utilisateurs du système est à recalculer.

III.6 Avantages et Inconvénients des SRS

Le tableau III.2 résume les forces et faiblesses des méthodes traditionnelles utilisé par les systèmes de recommandation, en l'occurrence le filtrage collaboratif, le filtrage démographique, le filtrage à base de contenu, et le Filtrage basé sur la connaissance.

- **Démarrage à froid :** Les systèmes de filtrage collaboratif dépendent des évaluations des items par les utilisateurs. Ainsi, un nouvel item ne peut pas être recommandé tant qu'aucun utilisateur ne l'a évalué. Dans les systèmes de recommandation basés sur le filtrage collaboratif et les systèmes basés sur le contenu, il est impossible de prédire les préférences des utilisateurs sans connaitre leurs historiques d'évaluations d'items. Ainsi, les nouveaux utilisateurs ne recevront pas de recommandations précises avant d'avoir évalué un certain nombre d'items.
- **Sparsity :** Un système de recommandation souffre de la sparsity quand le nombre d'items évalués par les utilisateurs est très faible par rapport au nombre d'items total présent dans le système. Ce fait conduit à avoir une très faible densité dans la matrice d'évaluation utilisateurs/items. Cela a des conséquences sur la capacité du système de recommandation à recommander toutes les items disponibles et sur l'exactitude des recommandations générée

Categories	**Avantages**	**Inconvénients**
❖ Filtrage collaborative	- Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique. - La qualité de la recommandation peut être évaluée. - Plus les nombres d'utilisateur est grand plus la recommandation est meilleur	- Démarrage à froid - La complexité : dans les systèmes avec un grand nombre d'items et d'utilisateur le calcul croit linéairement.
❖ Technique basée sur le contenu	- Une liste de recommandations peut être générée même s'il n'y a qu'un seul données. - La qualité croit avec le temps. - Pas besoin d'information sur les autres utilisateurs. - Prendre en considération les goûts uniques des utilisateurs	- Manque de diversité des recommandations - Problème de recommandation des images et de vidéos en absence de Métadonnées. – Nécessité du profil d'utilisateur.
❖ Filtrage démographique	- N'exige aucun historique d'estimations	- Problème de confidentialité. - nouvel item - Utilisateur avec un goût unique.
❖ Filtrage basés connaissance	- Sensible avec le changement des preferences	- La Sparsity

TABLE III.2 - Les avantages et les inconvénients des techniques de recommandations.

III.7 Conclusion

Dans ce chapitre nous avons présenté un certain nombre d'approches visant à produire des systèmes de recommandation.

Les systèmes basés sur le filtrage collaboratif reposent seulement sur les évaluations des utilisateurs et peuvent être utilisés afin de recommander des articles sans aucune manipulation de contenu.

Nous nous sommes intéressés au filtrage collaboratif. En effet, les avantages du filtrage collaboratif incluent la capacité à filtrer n'importe quel type de contenu : du texte, de la vidéo, de la musique..., dont la manipulation est complexe et demande du temps.

Ce filtrage ne demande pas de connaissances particulières sur les méthodes de fouille de contenu, de recherche d'information ou d'indexation.

Le filtrage collaboratif est une technique dont le principe est simple à appliquer et bien adaptée aux articles dont le contenu est complexe (vidéo, son, images...).

Dans le chapitre suivant, nous allons présenter nos résultats expérimentaux en appliquant l'approche proposée, une étude comparative entre les deux méthodes de filtrage collaboratif pour enfin conclure quelle est la meilleure.

CHPITRE IV

EXPÉRIMENTATION

CHAPITRE IV

IV.1 Introduction

Nous allons expérimenter un système de recommandation, et nous allons faire une étude comparative entre les deux méthodes de filtrage collaboratif pour enfin conclure quelle est la meilleure avec le langage de programmation python. Mais avant cela, nous allons présenter notre environnement de travail.

IV.2 Environnement de travail

IV.2.1 Matériel

Mémoire (RAM): 8 GO.

Processeur : Intel ® Core™ i5-5300U CPU 2.30 GHz 2.29 GHz.

Système d'exploitation : Système d'exploitation 64 bits, processeur x64.

IV.2.2 Python

Python est un langage de script de haut niveau, portable, dynamique, extensible, gratuit, structuré et open source conçu pour être orienté objet. Il est multi-paradigme et multi-usage. Développé à l'origine par Guido Van Rossum en 1993, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.

IV.2.3 Caractéristiques du langage :

– Python est portable, non seulement sur les différentes variantes d'UNIX, mais aussi sur les OS propriétaires : MacOS, BeOS, NeXTStep, MS-DOS et les différentes variantes de Windows.

– Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux

– La syntaxe de Python est très simple et combinée des types des données évolués (listes, dictionnaires...), conduit à des programmes à la fois très compacts et très lisibles.

– Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de comptage de références (proche, mais différent, d'un garbage collector).

– Python est orienté-objet. Supporte l'héritage multiple et la surcharge des opérateurs.

– Python est dynamique (l'interpréteur peut évaluer des chaînes de caractères représentant des expressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflectif (il supporte la méta programmation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même).

IV.3 Description de l'approche proposée

L'objectif de ce projet était la création d'un système de recommandation performant et scalable. On a mesuré la performance avec les indices d'évaluation tels que le RMSE (Root Mean Square Error) et la scalabilité par la possibilité d'appliquer nos algorithmes sur un jeu de données de 10 millions d'observations. Pour le faire, on a utilisé des modèles qui se basent sur la similarité (Memory Based) et des modèles qui se basent sur la factorisation des matrices (Model Based).

IV.4 L'évaluation des systèmes de recommandation

Dans le cadre de l'évaluation expérimentale du Framework proposé, nous avons effectué une expérience pour illustrer sa validité et son efficacité, et pour évaluer également ses performances et son évolutivité. Pour réaliser cette expérimentation, nous avons utilisé la métrique prédictive RMSE, et les métriques d'aide à la décision (la précision et le rappel). La métrique prédictive calcule la précision des prédictions par rapport à l'évaluation réelle effectuée par l'utilisateur par contre les métriques d'aide à la décision n'évaluent pas la qualité de la prédiction, mais la pertinence des recommandations.

IV.4.1 Erreur quadratique moyenne (RMSE)

L'erreur quadratique moyenne de la racine calcule la valeur moyenne de toutes les différences au carré entre les notes réelles et prédites, puis procède au calcul de la racine carrée sur le résultat. Par conséquent, de grosses erreurs peuvent affecter considérablement la note RMSE, ce qui rend la métrique RMSE plus précieuse lorsque des erreurs significativement importantes sont indésirables. L'erreur quadratique moyenne entre les notes réelles et les notes prédites est donnée par :

$$\textbf{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}(\mathbf{d\,i} - \hat{\mathbf{d}}\,\mathbf{i})^2}{n}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

- d i : est la note réelle.
- ˆd i : est la note prévue.
- n : est le nombre de notes.

IV.4.2 Précision et Rappel

Lorsqu'on cherche à prédire si un utilisateur est intéressé ou non par un item, quatre possibilités sont offertes par la matrice de confusion.

Item	Pertinent	Non pertinent
Recommandé	Vrai Positif (vp)	Faux Positif (fp)
Non recommandé	Faux Négatif (fn)	Vrai Négatif(vn)

TABLE IV.1– Matrice de confusion

La précision (2) correspond au pourcentage ou au nombre des items suggérés et s'avérant véritablement pertinentes pour l'utilisateur. Par exemple, si l'on considère une liste des Top-N recommandations, la précision correspond à la proportion d'items véritablement consommés, appréciés ou achetés par l'utilisateur courant. Il est calculé en utilisant l'expression suivante :

$$Précision = \frac{vp}{vp+fp} \text{.....(2)}$$

Le rappel (recall) (2) mesure le nombre de recommandations pertinentes émises au regard du nombre total de recommandations pertinentes. Concrètement, on énumère le nombre d'items dont la mesure associée est non nulle et se retrouvant parmi les items suggérés, il est calculé par la formule ci-dessous :

$$Rappel = \frac{vp}{vp+fn} \text{.....(3)}$$

Si la précision est faible, l'utilisateur sera insatisfait, car il devra perdre du temps à lire des items qui ne l'intéressent pas. Si le rappel est faible, l'utilisateur n'aura pas accès à un item qu'il souhaiterait avoir.

Un système de recommandation parfait doit avoir une précision et un rappel près de la valeur 1, mais ces deux exigences sont souvent contradictoires et une très forte précision ne peut être obtenue qu'au prix d'un rappel faible et vice-versa.

IV.5 Apprentissage :

Quel que soit le jeu de données utilisé, et quel que soit l'approche utilisée nous devons impérativement passer par les étapes suivantes :

- Télécharger le fichier, puis le lire avec la bibliothèque pandas du python.
- Construire la matrice d'évaluation utilisateur-item.
- Diviser les données en ensembles d'apprentissage et de test.
- Construire une matrice de similarité.

Avec notre matrice de similarité en main, nous pouvons maintenant prédire les notations qui n'ont pas été incluses dans les données.

En utilisant ces prédictions, nous pouvons ensuite les comparer avec les données de test pour essayer de valider la qualité de notre modèle de recommandation.

IV.5.1 Résultats expérimentaux

Modélisation : il existe deux types de recommandation :

- Approche basée sur le contenu (content Based): On ne va pas l'utiliser vu qu'on n'a pas les caractéristiques de chaque item et de chaque utilisateur.

-Approche de filtrage collaboratif (collaboratif filtering): Qui se base sur la note attribuée par l'utilisateur sur l'item.

Il existe deux types de systèmes de recommandation par filtrage collaboratif : Modèle basé sur la mémoire et méthode basé sur le modèle :

Pour le modèle basé sur la mémoire, on a utilisé le FC à base utilisateur et FC à base item pour les métriques "cosinesimilarity" et "cityblock".

Pour filtrage collaboratif basé sur le modèle, on a utilisé les SVDs et le gradient stochastique descendant pour l'améliorer.

Problématique :

Python n'a pas réussi à créer des matrices avec 7636 lignes et 1264 colonnes (pour construire la matrice user-item). C'est pour cela on a voulu expliquer et résolut le problème dans un cas de jeu de données plus petit dis ont de l'ordre 100.000 au lieu de 10.000.000. À la fin on a réussi à résoudre le problème avec tout le jeu de donnés. On a aperçu que la base de données présente les notes utilisateur par utilisateur. Donc c'est on prend 100.000 observations aléatoirement de la base, on peut tomber sur le même problème vu qu'on peut ne pas réduire vraiment le nombre des utilisateurs et le nombre des items, pour ça on a décidé de travailler sur les premiers 100.000 lignes.

	userId	movieId	rating	timestamp
0	1	31	2.5	1260759144
1	1	1029	3.0	1260759179
2	1	1061	3.0	1260759182
3	1	1129	2.0	1260759185

FIGURE IV.1– Matrice d'évaluation utilisateur-item.

1. Filtrage collaboratif basé sur la mémoire :

1.1 La mise en place du modèle:

On va commencer par créer les modèles basés sur la mémoire :

- L'analyse basée sur l'utilisateur (User-Based) : "Les utilisateurs qui sont similaires à vous, ont aimé aussi "

- L'analyse basée sur l'item (Item-Based): "Les utilisateurs qui ont aimé ça, ont aimé aussi"

Pour expliquer plus:

- L'analyse basée sur l'utilisateur: va prendre un utilisateur, trouve les utilisateurs les plus similaires à lui en se basant sur la note, puis recommande les items aimés par ces utilisateurs (ça prend un user et retourne des items).

- L'analyse basée sur l'item: prend un item, cherche les utilisateurs qui ont aimé cet item, trouve les items aimés par ces utilisateurs (Ça prend un item et retourne une liste des items).

Pour le faire, on utilise 2 métriques le cosine similaire et cityblock. On a commencé par créer les matrices user-item train et test. Ce sont les deux matrices qui vont croiser les notes des utilisateurs et des items. Puis, on a créé nos 4 modèles basés sur la mémoire, à la fin, on a créé une fonction pour faire les prédictions selon le modèle.

1.2 La comparaison des RMSE :

Après le calcul des valeurs de RMSE on a trouvé les résultats suivants :

```
User-based CF RMSE: 1.4910328668717732
Item-based CF RMSE: 1.495252514238234
User-based1 CF RMSE: 1.5235943464287394
Item-based1 CF RMSE: 1.5348231521032742
```

FIGURE IV.2 –Résultats des RMSE.

Le meilleur modèle est celui qui a le RMSE le plus petit, pour notre cas c'était Item-Based pour la métrique cityblock.

1.3 La généralisation de notre meilleur modèle:

Dans un vrai problème, ce qu'on veut faire c'est d'avoir des recommandations pour un utilisateur précis. On a créé la fonction getrecom qui prend en paramètres :

-iduser: l'id d'un utilisateur.

-n: nombre de résultat à affiché.

-ch== {"all"|"discover"}: par défaut "all", on utilise "discover" pour avoir seulement des recommandations sur les items non notés par l'utilisateur. On a construit le modèle à partir de tout le jeu de données.

```
(getrecom_membased_for_item(6373,10,"all"))
40     5.000000
117    4.998489
663    4.994203
24     4.985279
84     4.982837
495    4.957466
710    4.955156
532    4.951005
426    4.939143
225    4.934024
Name: 6372, dtype: float64
```

```
(getrecom_membased_for_user(730,10,"discover"))
2139    4.911902
1910    4.910203
1591    4.900214
2457    4.894726
3574    4.888690
2522    4.884848
2235    4.882589
813     4.881029
1773    4.879601
1112    4.879552
Name: 729, dtype: float64
```

FIGURE IV.3– Exemples de fonctionnement de fonction getrecom.

-Les modèles basés sur la mémoire sont faciles à implémenter et générer des bons résultats.- Ce type de modèle n'est pas scalable (n'est pas vraiment pratique dans un problème d'une grande base de données vu qu'il calcule à chaque fois la corrélation entre tous les utilisateurs et les items) et ne résolut pas le problème de cold start (lorsqu'on commence avec un nouvel utilisateur/item dont on n'a pas assez d'information)

-Pour répondre au problème de scalabilité on crée les modèles basés sur le modèle (partie suivante).

-Pour répondre au problème de cold start, on utilise la recommandation basée sur le contenu (on ne va pas l'utiliser vu qu'on n'a pas ces données).

2. Filtrage collaboratif basé sur le model :

Dans cette partie du projet, nous appliquons le deuxième sous-type du filtrage collaboratif : "Model-based".

Il consiste à appliquer la matrice de factorisation (MF) : c'est une méthode d'apprentissage non supervisé de décomposition et de réduction de dimensionnalité pour les variables cachées.

Le but de la matrice de factorisation est d'apprendre les préférences cachées des utilisateurs et les attributs cachés des items depuis les ratings connus dans notre jeu de données, pour enfin prédire les ratings inconnus en multipliant les matrices de variables cachées des utilisateurs et des items.

Il existe plusieurs techniques de réduction de dimensionnalité dans l'implémentation des systèmes de recommandations.

Dans notre projet, nous avons utilisés :

- SVD : Singular Value Decomposition.

- SGD : Stochastic Gradient Descent.

- NMF (sklearn) : Non-Negative Matrix Factorization.

-ALS : Alternating Least Squares.

2.1 Décomposition en valeurs singulières (SVD) :

Cette technique, comme toutes les autres, consiste à réduire la dimensionnalité de la matrice User-Item calculée précédemment.

Posons R la matrice User-Item de taille m x n (m : nombre de users, n: nombre d'items) et k: la dimension de l'espace des caractères cachés.

L'équation générale de SVD est donnée par : R=USV^T.

- La matrice U des caractères cachés pour les utilisateurs : de taille m*k.

- La matrice V des caractères cachés pour les items : de taille n*k.

- La matrice diagonale de taille k x k avec des valeurs réelles non-négatives sur la diagonale. On peut faire la prédiction en appliquant la multiplication des 3 matrices.

On a trouvé 1.49 comme RMSE, c'est plus grand que le RMSE des modèles basés sur la mémoire, mais ça prend énormément moins du temps.

Ce qu'on va dans la partie qui suit c'est d'améliorer notre modèle par le gradient stochastique.

2.2 Algorithme du gradient stochastique(SGD)

L'algorithme du gradient stochastique est une méthode de descente de gradient (itérative) utilisée pour la minimisation d'une fonction objectif qui est écrite comme une somme de fonctions différentiables**.**

Quand on utilise le filtrage collaboratif pour SGD, on veut estimer deux matrices P et Q:

- La matrice P des caractères cachés pour les utilisateurs : de taille m*k (m: nombre d'utilisateurs, k: dimension de l'espace des caractères cachés)

- La matrice Q des caractères cachés pour les items : de taille n*k (m: nombre d'items, k: dimension de l'espace des caractères cachés)

Après l'estimation de P et Q, on peut alors prédire les ratings inconnus en multipliant les matrices P et la transposée.

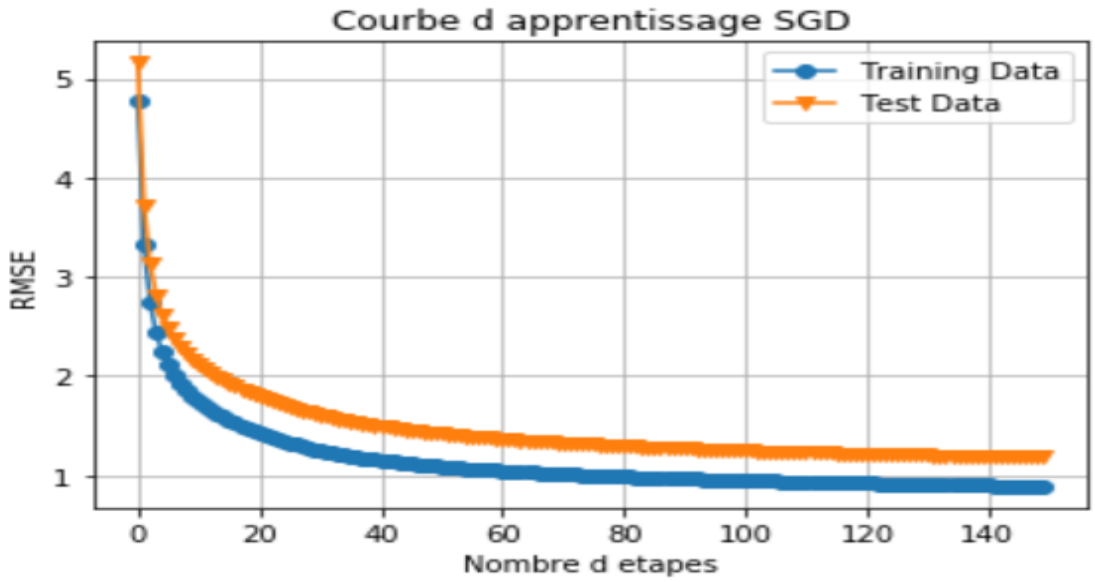


FIGURE IV.4 –Courbe d'apprentissage SGD.

Le modèle semble fonctionner bien avec, relativement, une basse valeur de RMSE après convergence.

La performance du modèle peut dépendre des paramètres (gamma), (lambda) et k qu'on a varié à plusieurs reprises afin d'obtenir le meilleur RMSE.

Après cette étape, on peut comparer le rating réel avec le rating estimé, pour ce faire, on utilise la matrice User-item qu'on a déjà calculée et utilisé la fonction prédiction (P, Q) implémentée précédemment.

2.3 Factorisation matricielle non négative (NMF) :

La factorisation matricielle non négative (NMF) est un algorithme d'extraction de fonctionnalités de pointe. NMF est utile lorsqu'il existe de nombreux attributs et que les

attributs sont ambigus ou ont une faible prévisibilité. En combinant des attributs, NMF peut produire des modèles, des sujets ou des thèmes significatifs. Voici la formule de la fonction de NMF avec le résultat :

```
from sklearn.decomposition import NMF
nmf_model = NMF(n_components=5, init='random', random_state=0)
nmf_model.fit(train_data_matrix)
```

```
NMF(alpha=0.0, beta_loss='frobenius', init='random', l1_ratio=0.0, max_iter=200,
    n_components=5, random_state=0, shuffle=False, solver='cd', tol=0.0001,
    verbose=0)
```

FIGURE IV.5 –Résultat de NMF.

2.4 Alternating Least Square (ALS)

ALS est également un algorithme de factorisation de matrice et il s'exécute en parallèle. ALS est implémentée dans Apache Spark ML et construit pour un problème de filtrage collaboratif à grande échelle. L ALS fait un assez bon travail pour résoudre l'évolutivité et la rareté des données d'évaluation, et c'est simple et s'adapte bien à de très grands ensembles de données. On a appliqué cet algorithme et on a trouvé les résultats suivants :

	Actual Rating	Predicted Rating
1	5.0	4.704053
8	4.0	4.837665
43	2.0	4.904602
73	4.0	4.799663
171	4.0	4.943187
210	4.0	4.732062
223	5.0	4.635833
239	4.0	4.225383
282	2.5	4.411363
311	3.0	4.146663

FIGURE IV.6– Résultats d'ALS.

Résultat final :

Pour filtrage collaboratif basé sur la mémoire: les modèles sont faciles à implémenter, ils génèrent des bons résultats mais ils ne résolvent pas les problèmes de Cold Start (pas d'informations sur les utilisateurs et les items), de rareté des données et de scalabilité (monter dans les dimensions).

Pour le filtrage collaboratif basé sur le modèle: Ils ont résolu le problème de rareté des données. Et chaque algorithme a ces points faibles et ces points forts. Pour l'utilisation de la matrice de factorisation finale : On a résolu le problème de scalabilité et de rareté des données. Mais on n'a pas exploité la variable date_ratings pour que nos tests soient plus significatifs.

On a trouvé comme résultat final que l'algorithme basé sur modèle est le meilleur de tous les autres algorithmes. Dans deux itération on a trouvé une erreur de train qui est égale à x et une erreur de test qui est égal à y. Comme c'est l'algorithme le plus rapide et le plus efficace, On a décidé de le généralisé sur tout le jeu de données.

IV.6 Conclusion:

Dans ce chapitre, nous avons présenté le cadre applicatif de notre travail. Nous avons décrit la mise en œuvre des différents modules de l'architecture proposée, en précisant le rôle de chaque élément. Ensuite, une présentation était faite des outils utilisés et du prototype que nous avons utilisé comme support à notre approche. Enfin nous avons présenté et discuté l'expérimentation mise en place. Le but principal de cette implémentation est de comparer les deux algorithmes à fin de conclure quel est le meilleur et de démontrer que cet algorithme peut résoudre le problème scalabilité et de rareté des données, ainsi il permet d'améliorer la qualité et la performance du système.

Conclusion général et perspectives

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

La tendance actuelle des systèmes de recommandation est plutôt axée sur des méthodes nouvelles, multicritères, multidimensionnelles ou encore se fondant sur des notions psychologiques comme les émotions, les opinions. Notons cependant qu'un système de recommandation doit avant tout s'adapter aux données, celles là mêmes que l'on proposera à un utilisateur. Ainsi, le choix d'une méthode de recommandation doit en premier lieu êtres dirigé par ce critère.

Le travail présenté dans ce mémoire rentre dans le cadre du filtrage collaboratif qui est la méthode la plus importante et la plus utilisée dans les systèmes de recommandation. Nous avons présenté expérimentalement, une étude comparative entre les deux méthodes de filtrage collaboratif**.** Nos résultats ont montré que l'algorithme basé sur modèle est le meilleur de tous les autres algorithmes, ainsi que cet algorithme peut résoudre le problème scalabilité et de rareté des données, ainsi il permet d'améliorer la qualité et la performance du système.

Comme perspectives nous envisageons d'apporter quelques améliorations à savoir :

- En premier lieu, nous souhaitons combines le filtrage collaboratif avec le filtrage à base du contenu dans le but d'avoir de meilleurs résultats.

- Et en second lieu, nous envisageons de rechercher sur comment donner une explication à une recommandation pour améliorer la confiance des systèmes de recommandation.

Liste des abréviations

IA : Intelligence Artificielle.

AA : Apprentissage Automatique.

SR : Système de Recommandation.

FC : Filtrage Collaboratif.

FC-I: Filtrage Collaboratif basé sur Item.

FC-U : Filtrage Collaboratif basé sur Utilisateur.

CBR : Raisonnement basé sur les cas.

CBF : Filtrage basé sur le contenu.

RMSE : Erreur quadratique moyenne.

MF : Matrice de factorisation.

SVD : Décomposition en valeurs singulières.

SGD : Algorithme du gradient stochastique.

NMF : Factorisation matricielle non négative.

BIBLIOGRAPHIE

[1]http://elearning.univ-jijel.dz/elearning/pluginfile.php/4333/mod-resource/content/1/SupportCours Mokhtar-Taffar-ApprAuto.pdf.

[2] https://rai2020.blogspot.com/2016_02_28_archive.html.

[3] https://aws.amazon.com/fr/personalize/

[4]http://www.pearltrees.com/damienmilin/article-voir/id16676292/item1905249791590/vous avez dit machine learning?

[5] L'efficacité de la recommandation produit chez amazon, 24,25

[6] Gediminas Adomavicius and YoungOk Kwon. New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems, 2007.22

[7] Charu C. Aggarwal. 34,35,36

[8] Fernandez . M. Mani Onana F. S. Aimeur E., Brassard G. Privacy-preserving demographic filtering, proc. of the ACM Sym. on Applied computing, pp.872 – 878, 2006. 31

[9] Hirsh H. Cohen W. Basu, C. and N. C. Manning.39

[10] A. Belloui. L'usage des concepts du web sémantique dans le filtrage d'information collaboratif. PhD thesis, Institut National d'Informatique d'Alger., 2008.29

[11] Heckerman D. Breese, J. and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In 14th Annual Conference on Uncertainty in Artificial Intelligence, pages 43-52, 1998.38

[12] R. Burke. Hybrid recommender systems : Survey and experiments. User Modeling and User-Adapted Interaction, 2002. 16,22,32,33

[13] Lakkaraju P. Luong H. P. Chandrasekaran K., Gauch S. Concept-based document recommendations for citeseer authors. In Proc. of the 5 th Inter.Conf., 5149, pp. 83-92., 2008.30

[14] Rijsbergen C.J.V. Information retrieval. second edition. Butterworks, 1979.20

[15] Steve Lawrence David M. Pennock, Eric Horvitz and C. Lee Giles. collaborative filtering by personality diagnosis :a hybrid memory- and model-based approach. In Proceedings of the sixteenth Conference on Uncertainty in Artificial Intelligence, 2000. 39,41

[16] Christian Desrosiers and George Karypis. A survey of collaborative filtering techniques. adv. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, Recommender Systems Handbook, pages 107–144., 2011. 35,36,44

[17] Nichols D. Oki M. Goldberg, D. and D. Terry. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35:61-70, 1992. 15

[18] Gasparetti F. Micarelli A. Sansonetti G. Gurini, D. F. A sentiment-based approach to twitter user recommendation. In RSWeb@ RecSys, 2013. 19

[19] K. Hammond. Case-based planning : Viewing planning as a memory task. Boston, MA : Academic Press, 1989. 32,33

[20] Jon Herlocker J. Ben Schafer, Dan Frankowski and Shilad Sen. Collaborative filtering recommender systems. in Lecture Notes in Computer Science, pages 291–324. Springer Berlin Heidelberg., 2007. 38

[21] Jon Herlocker J. Ben Schafer, Dan Frankowski and Shilad Sen. Collaborative filtering recommender systems. in peter brusilovsky, alfred kobsa, and wolfgang nejdl, editors, the adaptive web, in Lecture Notes in Computer Science, pages 291-324., 2007. 33

[22] et J. Riedl J. Herlocker, J. Konstan. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval, 2002. 16

[23] F. Meyer. Recommender systems in industrial contexts. PhD thesis, University of Grenoble, France, 2012. 16

[24] Joseph A. Konstan Al Borchers Nathaniel Good, J. Ben Schafer. Combining collaborative filtering with personal agents for better recommendations. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence.pages 439–446, Menlo, Park,, 1999. 44

[25] Elsa Negre. 16,40

[26] D. Poirier. Des textes communautaires à la recommandation. PhD thesis, Université d'Orléans et Université Pierre et Marie Curie - Paris 6., 2011. 25

[27] P. Resnick and H. Varian. Recommender systems. Communications of the ACM, 40 :56-58, 1997. 15,16,37,41

[28] F. Ricci. Travel recommender systems. IEEE Intelligent Systems, 2002. 17,29

[29] Karypis G. Konstan J. Sarwar, B. and J. Riedl. Item-based collaborative filtering recommendation algorithms. In WWW10, pages 285-295, Hong Kong. ACM., 2001. 39,40,42

[30] Senee. 5 moteurs de recommandation de merchandising à la loupe, 2015. 26

[31] Larson M. Hanjalic A. Shi, Y. Collaborative filtering beyond the user-item matrix : A survey of the state of the art and future challenges. ACM Computing Surveys (CSUR), 2014. 18,19

[32] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. adv. in Artif. Intell., 2009, pp.4:2-4:2., 2009. 35

[33] Quinn C. Towle B. Knowldge based recommender systems using explicit user models. In KBEM, Technical Report WS-00-04, pp. 74-44., 2000. 33

[34]https://sites.google.com/site/tpemachinelearning/lintelligenceartificielle/machine_learning