République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. MIRA-BEJAIA

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Faculté des Sciences Exactes
Département d'Informatique

MÈMOIRE

EN VUE DE L'OBTENTION DU DIPLÔME DE

MASTER RECHERCHE

Domaine : Mathématiques et Informatique **Filière :** Informatique
Spécialité : Systèmes d'Information Avancés

Présenté par

M.BOUAKKAZ Younes

M. MAYOUF Farid

Thème

Système de recommandation de livres basé sur des outils d'intelligence artificielle

Soutenu le 03 Juillet 2024 **Devant le jury composé de :**

Nom et Prénom	Grade		
Mme GADOUCHE Hania	MCB	Université de Béjaia	Présidente
Mme BATTAT Nadia	MCB	Université de Béjaia	Examinatrice
Mme AIT HACENE Souhila	MAA	Université de Béjaia	Encadrante
Mme CHIBANI Samia	MCA	Université de Béjaia	Co-encadrante

Année Universitaire : 2023/2024

Remerciements

Avant tout, nous tenons à exprimer notre profonde gratitude envers notre *Dieu*, le miséricordieux et tout-puissant, qui nous a octroyé le courage nécessaire pour mener à bien notre parcours universitaire. Que sa guidance continue de nous éclairer tout au long de notre vie future.

Nous souhaitons également remercier particulièrement Madame *AIT HACENE Souhila* et Madame *CHIBANI Samia*. Sans leur aide et leur encadrement, ce travail n'aurait pas pu être réalisé et ne serait pas aussi riche. Nous les remercions sincèrement de nous avoir soutenus dans cette démarche, d'avoir consacré le temps nécessaire à sa réalisation, et de nous avoir prodigué des remarques et des retours pertinents, ainsi que leur souci du détail, contribuant ainsi de manière significative à l'aboutissement de ce mémoire.

Nous remercions Madame *GADOUCHE Hania* pour l'honneur qu'elle nous a fait en acceptant de présider le jury. Nos remerciements s'adressent également a Madame *BATTAT Nadia*, membre du jury, pour l'intérêt qu'elle a manifestée envers notre travail en acceptant de l'examiner et de l'enrichir par ses précieuses suggestions.

Nous exprimons notre reconnaissance a nos famille pour leur soutien moral inconditionnel et ‘ leur présence réconfortante lors des moments difficiles.

Nous tenons à remercier chaleureusement l'ensemble des enseignants qui ont assuré notre formation. Leur encadrement pédagogique de qualité et leur présence d'esprit remarquable ont été déterminants dans notre réussite. Nous leur adressons notre plus sincère gratitude.

Dédicaces

Je dédie ce mémoire ....

À mes chers parents

Je profite de cette occasion pour exprimer ma profonde gratitude envers vous. Votre soutien continu et votre amour infini ont été ma source de force et d'inspiration tout au long de ce parcours académique. Vous êtes le socle sur lequel je me suis construit et cette réussite est aussi la vôtre.

À ma petite famille

Pour leur présence constante et inébranlable, qui a été une véritable source de force et de motivation. Leur amour et leur soutien m'ont guidé et encouragé tout au long de ce parcours académique. Je leur dédie ce travail en signe de reconnaissance pour tout ce qu'ils ont fait pour moi.

À la famille de ma mère

Pour leur soutien indéfectible et leurs encouragements constants, qui ont été essentiels à la réalisation de ce mémoire. Leur affection et leurs conseils m'ont toujours inspiré à donner le meilleur de moi-même. Je leur suis profondément reconnaissant pour leur présence tout au long de ce voyage.

À mon encadrante

Madame *Souhila AIT HACENE*, pour son encadrement, ses conseils avisés et son dévouement tout au long de la réalisation de ce projet. Sa passion pour l'enseignement et la recherche a été une source de motivation constante.

À mes amis

Merci pour votre soutien indéfectible et vos encouragements constants. Vos moments de réconfort, de détente et de rire ont été essentiels pour m'aider à traverser ce parcours académique. Votre amitié et votre présence ont été inestimables.

MAYOUF Farid

Dédicaces

Je dédie ce mémoire ....

À mes chers parents

Je prends ce moment pour vous exprimer ma profonde gratitude. Votre soutien continu et votre amour infini ont été mes sources de force et d'inspiration tout au long de mon parcours académique. Vous avez été le socle sur lequel je me suis construit, et cette réussite est également la vôtre.

À mon encadrante

Madame *Souhila AIT HACENE*, pour son encadrement, ses conseils avisés et son dévouement tout au long de la réalisation de ce projet. Sa passion pour l'enseignement et la recherche a été une source de motivation constante.

À mes amis

Merci d'être toujours là, de partager les rires, les peines et les moments précieux. Vous êtes vraiment spéciaux pour moi.

BOUAKKAZ Younes

Table des matières

Table des figures

Liste des tableaux

Liste des acronymes

SR Système de Recommandation

CF Collaborative Filtering

CB Content-Based Filtering

IA Intelligence Artificielle

ML Machine Learning

DL Deep Learning

SVM Support Vector Machine

ANN Artificial Neural Network

KNN k-Nearest Neighbors

CNN Convolutional Neural Network

RNN Recurrent Neural Network

DNN Deep Neural Network

API Application Programming Interface

RMSE Root Mean Squared Error

BRS BooK Recommendation System

CARS Les systèmes de recommandation contextuels

QoS Quality of Service

Kg-WSR Knowledge Graph-based Web Service Recommendation

SVD Singular Value Decomposition

SR-HSDF Système de Recommandation Hybride proposant une solution au problème de Sur-spécilisation et de Démarrage à Froid

Bow Bag of words

NLP Natural Language Processing

Introduction générale

Les systèmes de recommandation sont des outils logiciels qui utilisent des algorithmes et des techniques d'intelligence artificielle pour proposer aux utilisateurs des contenus, des produits ou des services en fonction de leurs préférences, comportements passés et données contextuelles. En analysant les interactions et les choix antérieurs des utilisateurs, ces systèmes prédisent ce qu'ils pourraient trouver intéressant ou utile, offrant ainsi des recommandations personnalisées et pertinentes. Les avantages de ces systèmes incluent la personnalisation, qui permet de fournir des suggestions adaptées aux préférences et besoins individuels des utilisateurs, améliorant leur expérience utilisateur. De plus, ils facilitent la découverte de nouveaux produits, services ou contenus que les utilisateurs n'auraient peut-être pas trouvés par eux-mêmes, enrichissant ainsi leur expérience globale. Enfin, en fournissant des recommandations pertinentes et personnalisées, ces systèmes augmentent la satisfaction des clients, ce qui peut conduire à une plus grande fidélité et à des avis positifs.

Cependant, malgré leurs avantages considérables, les systèmes de recommandation rencontrent plusieurs défis. Parmi les plus notables, on trouve le problème de sur-spécialisation, où les recommandations deviennent trop spécifiques et limitent ainsi la découverte de nouveaux contenus. L'évolutivité est également un problème majeur, car les systèmes doivent pouvoir gérer des volumes de données et des utilisateurs en constante augmentation sans perte de performance. Enfin, le problème du démarrage à froid où le système doit fournir des recommandations à des nouveaux utilisateurs ou pour de nouveaux articles en l'absence de données historiques suffisantes.

Dans notre mémoire, nous avons développé une approche intitulée « Système de Recommandation Hybride proposant une solution au problème de Sur-spécilisation et de Démarrage à Froid (SR-HSDF) ». Cette approche repose sur quatre contributions principales :
Contribution 1 : La collecte et pré-traitement des données : Cette étape consiste à rassembler et à préparer les données nécessaires pour le système de recommandation, assurant ainsi leur qualité et leur pertinence.
Contribution 2 : Système de recommandation avec filtrage basé sur le contenu : Utilisation d'algorithmes qui recommandent des articles (livres) en se basant sur les caractéristiques similaires aux articles que l'utilisateur a appréciés dans le passé.
Contribution 3 : Système de recommandation avec filtrage collaboratif : Cette méthode repose sur les interactions et les préférences d'utilisateurs similaires pour fournir des recommandations.
Contribution 4 : Résolution du problème du démarrage à froid avec k-means : Implémentation d'une technique de clustering permettant de catégoriser les utilisateurs et les livres pour améliorer les recommandations dès les premières interactions.

Ce mémoire est structuré en quatre chapitres :

Chapitre 1 : Les Systèmes de recommandation Introduction aux concepts fondamentaux des systèmes de recommandation, incluant une classification et une compa-

raison des différentes approches existantes.

Chapitre 2 : L'intelligence artificielle exploration des outils d'intelligence artificielle utilisés dans les systèmes de recommandation, mettant l'accent sur les algorithmes du Machine Learning et du Deep Learning.

Chapitre 3 : Etat de l'art Présentation d'un état de l'art des travaux connexes sur les systèmes de recommandation. Cette revue de travaux antérieurs nous permettra de positionner notre approche proposée par rapport aux solutions existantes et d'identifier les lacunes potentielles dans les approches actuelles.

Chapitre 4 : Approche proposée et évaluation présentation de notre approche SR-HSDF à travers la description des divers contributions et des exemples de leur mise en œuvre.

Chapitre 1

Les Systèmes de recommandation

1.1 Introduction

Dans ce premier chapitre, nous fournissons une définition des systèmes de recommandation et soulignons leur importance dans divers domaines. Ensuite, nous présentons une classification des systèmes de recommandations, ainsi qu'une comparaison des différents types de systèmes de recommandations déjà vus. Enfin, nous terminons le chapitre avec les défis et limites d'un système de recommandations.

1.2 Les systèmes de recommandation (SR)

1.2.1 Définition

Les systèmes de recommandation sont des outils essentiels dans le filtrage de l'information, car ils s'engagent à proposer des éléments jugés pertinents pour un utilisateur spécifique. Pour ce faire, une variété d'algorithmes est utilisée, allant des approches simples basées sur des mots-clés à des techniques plus complexes. En d'autres termes, ces systèmes offrent des recommandations personnalisées qui répondent aux intérêts spécifiques de chaque utilisateur [8].

Dans le fonctionnement de ces systèmes, deux composants clés se distinguent : l'utilisateur et l'élément. Dans ce mémoire, le terme élément peut être un objet physique comme une voiture, un ordinateur, un livre, un tee-shirt, ou un produit non physique, comme les films, la musique... où encore un service tel que : un médecin, un hôpital, un avocat. Le RS renvoi des suggestions aux utilisateurs, qui fournissent des avis par le biais de votes ou de commentaires. Ces avis sont souvent enregistrés sous forme de triples (utilisateur, élément, avis) et utilisés pour créer une matrice de scores représentant les interactions entre les utilisateurs et les éléments. Les avis peuvent prendre différentes formes, comme une note numérique ou une option binaire (j'aime/je n'aime pas). En somme, les systèmes de recommandation utilisent les retours des utilisateurs pour affiner et améliorer continuellement leurs recommandations personnalisées. Afin d'illustrer ce processus, nous présentons dans cette figure 1.1 un exemple concret dans le domaine de la santé [1].

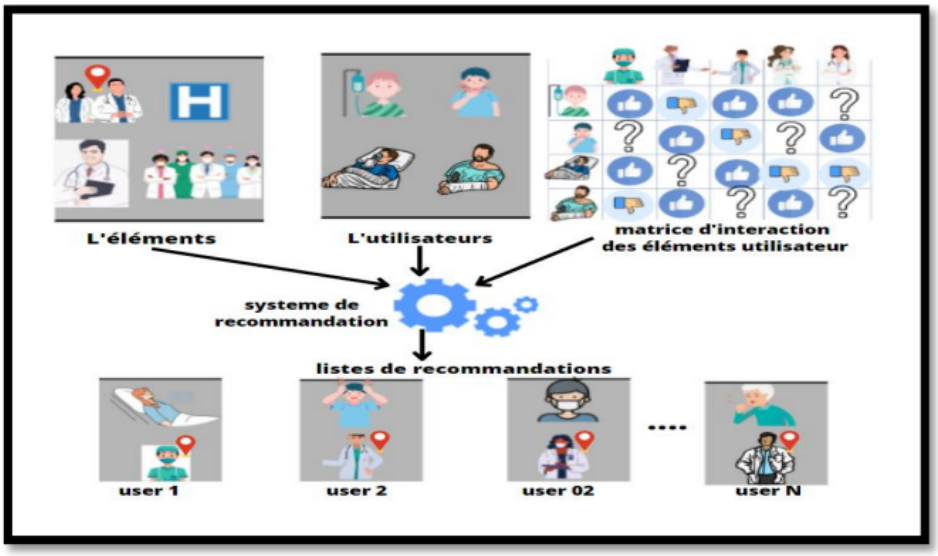

FIGURE 1.1 – Architecture générale des systèmes de recommandation[1].

1.2.2 Les types de systèmes de recommandation

1.2.2.1 Les systèmes de recommandation basé sur le filtrage collaboratif (collaborative filtering (CF))

Un système de recommandation basés sur le filtrage collaboratif est une technique de recommandation qui fait appel aux historiques d'interaction de plusieurs utilisateurs dans le but de recommander un élément ou un produit à un utilisateur donné. Ces systèmes partent du principe que les préférences d'un utilisateur peuvent être déduites des préférences d'un autre.

Le filtrage collaboratif, utilisé pour des recommandations personnalisées, offre des avantages notables comme des suggestions précises basées sur les préférences des utilisateurs similaires et une amélioration continue avec l'accumulation de données. Cependant, il présente des inconvénients, notamment le problème de démarrage à froid, la tendance à favoriser les items populaires au détriment de la diversité, et la nécessité de ressources informatiques importantes pour gérer un grand nombre d'utilisateurs et d'items[9]. La figure 1.2 représente un système de recommandation de livres qui se base sur le filtrage collaboratif .

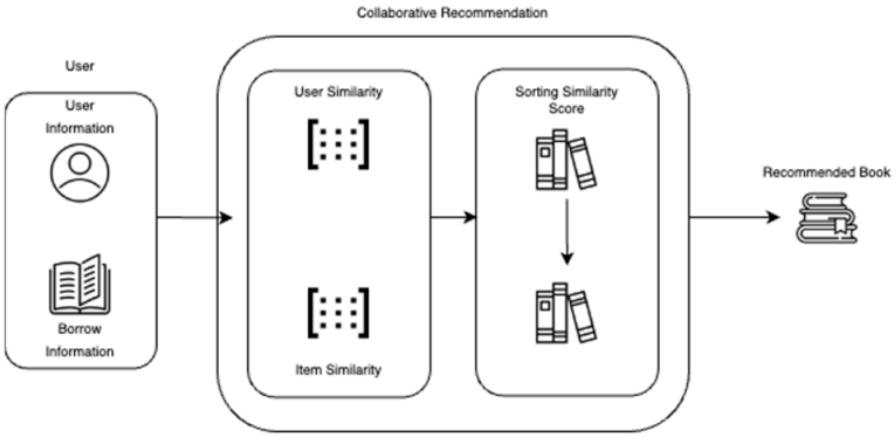

FIGURE 1.2 – Système de recommandation basé sur le filtrage collaboratif[2].

1.2.2.2 Système de recommandation avec filtrage basé sur le contenu (content-based filtering (CB))

Un système de recommandation basé sur le contenu est un type de système de recommandation qui suggère des éléments similaires à un utilisateur en se basant sur les caractéristiques et les attributs des éléments eux-mêmes. Contrairement aux systèmes de recommandation collaboratifs qui se fondent sur les comportements passés des utilisateurs, les systèmes basés sur le contenu analysent les attributs intrinsèques des éléments pour déterminer leur similitude. Par exemple, dans le cas d'un système de recommandation de films basé sur le contenu, les attributs pourraient inclure le genre, les acteurs, le réalisateur, le synopsis, etc. Ce type de système est souvent utilisé dans des domaines comme le commerce électronique, la musique, la vidéo et la recommandation d'articles [10].

Les systèmes de recommandation basés sur le contenu offrent des recommandations pertinentes pour les nouveaux utilisateurs et peuvent suggérer des éléments de niche. Cependant, leur dépendance aux métadonnées peut conduire à des recommandations redondantes, et ils ne prennent pas en compte les préférences changeantes de l'utilisateur. De plus, ils risquent de souffrir de la sur-spécialisation en ne considérant que des caractéristiques spécifiques plutôt que l'ensemble des intérêts de l'utilisateur[11]. La figure 1.3 représente un système de recommandation de livres qui se base sur le contenu.

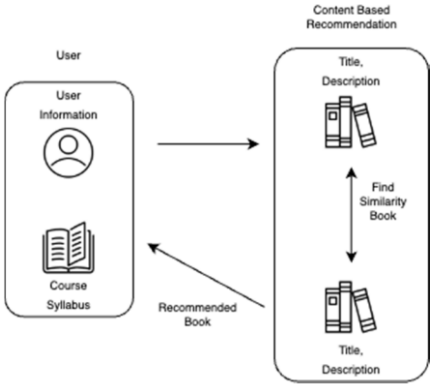

FIGURE 1.3 – Système de recommandations de livres basé sur le contenu[2].

1.2.2.3 Système de recommandation hybride "filtrage hybride" (hybrid filtering)

Un système de recommandation hybride est un type de système de recommandation qui combine plusieurs méthodes de recommandation différentes pour offrir des suggestions personnalisées aux utilisateurs. Typiquement, ces systèmes intègrent des techniques de filtrage collaboratif, qui se basent sur les préférences et les comportements d'autres utilisateurs similaires, avec des méthodes de filtrage basées sur le contenu, qui analysent les caractéristiques des éléments recommandés. En combinant ces approches, les systèmes de recommandation hybrides cherchent à pallier les limitations individuelles de chaque méthode, offrant ainsi des recommandations plus précises et diversifiées [12] . La figure 1.4 représente un système de recommandation hybride.

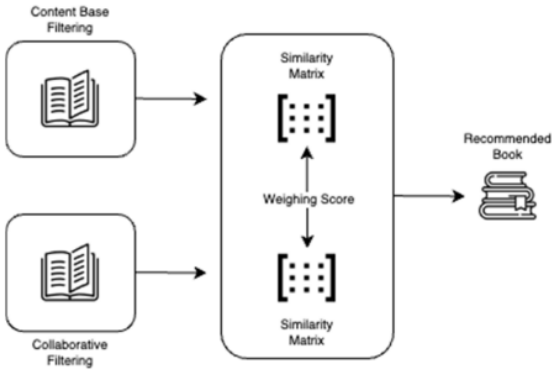

FIGURE 1.4 – Système de recommandation hybride (Filtrage hybride)[2].

Les systèmes de recommandation hybrides utilisent diverses stratégies pour combiner différentes techniques de filtrage, telles que la fusion simple, la pondération, la cascade, le niveau de confiance, le switching, les modèles intégrés et l'apprentissage dynamique. Chaque approche présente des avantages et des inconvénients en fonction du contexte d'application et des objectifs du système de recommandation [11].

1.2.2.4 Système de recommandations basées sur l'apprentissage automatique (Machine learning (ML))

Ces dernières années, l'apprentissage automatique a suscité un grand intérêt dans différents domaines de recherche tels que la vision par ordinateur (consiste à permettre aux machines de voir, analyser et comprendre les informations visuelles comme images et vidéos) et le traitement du langage naturel (NLP), non seulement en raison de ses excellentes performances, mais également en raison de son aspect attrayant consistant à apprendre à partir de zéro. Un système de recommandation basé sur l'apprentissage automatique est un modèle algorithmique qui analyse les modèles de données pour fournir des recommandations ou des prédictions personnalisées aux utilisateurs. Les algorithmes d'apprentissage automatique peuvent produire des systèmes de recommandation efficaces qui proposent des suggestions utiles en fonction des données de l'utilisateur, de l'élément et de l'interaction de l'utilisateur avec l'élément [13]. Aussi, les techniques d'apprentissage automatique sont utilisées pour fournir des recommandations sur différentes applications, dans le but de renforcer la confiance de l'utilisateur en fournissant des recommandations respectant la confidentialité [14].

1.2.3 Domaines d'application des systèmes de recommandation

Les systèmes de recommandation sont une solution performante de personnalisation, qui s'appuie continuellement sur les préférences du consommateur. Ils sont essentiels pour personnaliser l'expérience de l'utilisateur, augmenter la pertinence des informations présentées et faciliter la découverte de nouveaux contenus et services dans une variété de domaines. Selon Ketki Kinkar [15], les systèmes de recommandation peuvent améliorer l'expérience utilisateur dans divers domaines tels que le e-commerce, l'éducation, le cinéma, la musique, la littérature, en fournissant des données pertinentes basées sur les préférences des consommateurs.

- **l'e-commerce** Dans le domaine du commerce électronique, les SR jouent un rôle crucial dans l'amélioration de l'expérience d'achat en ligne et dans la stimulation des ventes en présentant aux consommateurs de nouveaux produits ou services qui correspondent à leurs préférences antérieures [16].
- **L'éducation** Dans le domaine de l'éducation, Les SR contribuent à l'apprentissage des élèves en proposant des recommandations personnalisées de ressources pédagogiques adaptées à leurs besoins individuels et à leur niveau de compétence[17].

- **Le divertissement** Les SR sont d'une grande importance dans l'univers du divertissement, couvrant le cinéma, la musique, la littérature et les jeux vidéo, car ils aident les utilisateurs à explorer et à trouver de nouveaux contenus qui correspondent à leurs préférences, améliorant ainsi leur parcours de divertissement global [18].
  Les systèmes de recommandation jouent un rôle crucial dans la recherche en ligne, en permettant aux utilisateurs de filtrer les résultats en fonction de leurs intérêts et comportements, facilitant ainsi la découverte d'informations pertinentes dans un vaste océan de données [19].
- **D'autres domaines** les SR sont particulièrement utiles dans des domaines tels

que les voyages, la restauration et la santé, où ils aident les utilisateurs à trouver des services et des produits adaptés à leurs besoins spécifiques, améliorant ainsi la prise de décision et la satisfaction client [20].

1.2.4 Les défis et limites d'un système de recommandations

Les systèmes de recommandation sont confrontés à plusieurs défis majeurs qui peuvent affecter leur efficacité et leur pertinence. Dans cette section, nous allons présenter certains défis des systèmes de recommandations.

— Problème de démarrage à froid (Cold start)
Le problème du démarrage à froid survient lorsqu'un nouvel utilisateur ou un nouvel élément est introduit dans le système et qu'il n'y a pas suffisamment d'informations pour formuler des recommandations pertinentes. En effet, pour un nouvel utilisateur, il n'y a aucune donnée historique sur ses préférences, ce qui rend difficile de trouver des utilisateurs similaires et de faire des recommandations appropriées. De même, pour un nouvel élément, il n'y a pas encore d'évaluations du produit par des utilisateurs, ce qui empêche le SR de déterminer sa pertinence pour différents utilisateurs [21].

— Rareté des données (Data sparsity)
La rareté des données fait référence à la situation où le nombre d'utilisateurs ayant évalué les éléments est très faible par rapport au nombre total d'éléments disponibles. Cela entraîne un chevauchement limité entre les évaluations des utilisateurs, rendant difficile de trouver des utilisateurs similaires et de formuler des recommandations précises. Ce problème a un impact négatif particulièrement important sur les techniques de filtrage Collaboratif, qui reposent sur les évaluations d'autres utilisateurs pour faire des prédictions[22].

— Évolutivité (Scalability)
La mise à l'échelle est un défi majeur pour les systèmes de recommandation traditionnels basés sur le filtrage collaboratif. Lorsque le nombre d'utilisateurs et d'éléments augmente de manière significative, les calculs nécessaires pour trouver des utilisateurs similaires et prédire les préférences deviennent très coûteux en termes de temps et de ressources. Cela peut conduire à des résultats imprécis et à des performances médiocres du système [23].

— Sur-spécialisation (Over-specialization)
La sur-spécialisation est un problème fréquent dans les systèmes de recommandation basés sur le contenu. Ces systèmes ont tendance à recommander des éléments trop similaires à ceux que l'utilisateur a déjà appréciés dans le passé, empêchant ainsi la découverte de nouveaux contenus et limitant la diversité des recommandations. Cela peut rapidement devenir ennuyeux pour l'utilisateur et réduire l'utilité du système [23, 24].

— Confidentialité (Privacy)
La confidentialité est une préoccupation majeure, en particulier pour les systèmes de recommandation démographique. Ces systèmes nécessitent la collecte d'informations personnelles sensibles sur les utilisateurs, telles que leur âge, leur sexe, leur localisation, etc. La divulgation ou l'utilisation inappropriée de ces données peut constituer une violation de la vie privée des utilisateurs [21, 25].

Nous concluons ce chapitre par la présentation d'un tableau comparatif des systèmes

de recommandations présentant les points forts et les points faibles de chaque SR.

1.2.5 Comparaison des systèmes de recommandation

	Points forts (Avantages)	Points faibles (Inconv´enients)
Le filtrage collabratife	**Simplicité et Efficacité :** Le filtrage collaboratif est facile à mettre en œuvre et peut fournir des recommandations de haute qualité en utilisant le comportement des utilisateurs. **Diversité des Recommandations :**Puisqu'il s'appuie sur le comportement des utilisateurs, il peut offrir une grande variété de recommandations, parfois inattendues.	**Problème du Démarrage à Froid :**Il rencontre des difficultés avec les nouveaux utilisateurs ou articles ayant peu ou pas de données d'interaction, rendant difficile la génération de recommandations. **Problèmes de Scalabilité :** À mesure que le nombre d'utilisateurs et d'articles augmente, le coût et la complexité computationnels augmentent de manière significative.
Le filtrage base sur le contenu	**Pas de Démarrage à Froid pour les Nouveaux Articles :**Les recommandations peuvent être faites pour de nouveaux articles tant que leurs caractéristiques sont connues. **Indépendance de l'Utilisateur :**Il ne nécessite pas de données utilisateur, ce qui facilite la gestion des préoccupations en matière de confidentialité.	**Limite de la Nouveauté :**Les recommandations ont tendance à être similaires à ce que l'utilisateur a déjà vu, manquant de surprise. **Sur-Spécialisation :**Peut trop se concentrer sur les préférences existantes de l'utilisateur, limitant ainsi la diversité et l'exploration de nouveaux contenus
Le filtrage hybride	**Amélioration de la Précision :** Combine plusieurs techniques de recommandation pour fournir des recommandations plus précises et robustes. **Réduction des Limitations :** Atténue les faiblesses des approches individuelles, comme les problèmes de démarrage à froid et le manque de nouveauté	**Mise en Œuvre Complexe :** La combinaison de plusieurs systèmes augmente la complexité en termes de conception et de mise en œuvre. **Coût Computationnel Élevé :** Nécessite plus de ressources computationnelles pour traiter et intégrer différents types de données et algorithmes.

TABLE 1.1 – Table de comparaison des systèmes de recommandation

1.3 Conclusion

Les systèmes de recommandation sont devenus des outils incontournables pour guider les utilisateurs à travers le vaste éventail de choix en ligne. Grâce aux percées de l'intelligence artificielle, en particulier l'apprentissage profond, ces systèmes peuvent fournir des recommandations personnalisées et pertinentes en exploitant efficacement les données riches et complexes sur les utilisateurs, les items et le contexte.

Les techniques d'IA permettent d'analyser de grandes quantités de données d'utilisateurs et d'items, afin d'identifier les modèles et les préférences cachés , l'IA jouera un rôle encore plus central dans le développement de systèmes de recommandation plus intelligents, robustes et performants capables de s'adapter aux besoins et préférences en constante évolution des utilisateurs.

Chapitre 2

L'intelligence Artificielle

2.1 Introduction

Dans ce chapitre, nous explorons certaines techniques de l'intelligence artificielle (IA), en mettant l'accent sur ses différentes branches telles que l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning).

2.2 Définition de l'Intelligence Artificielle

L'intelligence artificielle (IA) fait référence aux efforts visant à rendre les machines aussi intelligentes que le cerveau humain. En informatique, l'IA désigne l'étude des « agents intelligents », c'est-à-dire tout dispositif capable de percevoir son environnement et de prendre des mesures maximisant ses chances d'atteindre avec succès ses objectifs. De manière informelle, on parle d'intelligence artificielle lorsqu'une machine arrive à exécuter des tâches que les humains associent généralement à l'intelligence et au raisonnement, comme l'apprentissage ou la résolution de problèmes complexes [26].

L'IA englobe des domaines comme l'apprentissage automatique (machine learning) et l'apprentissage profond (deep learning). L'apprentissage automatique est un sous-domaine de l'IA qui permet aux systèmes d'apprendre et de s'améliorer à partir de données, sans être explicitement programmés. L'apprentissage profond, quant à lui, est un sous-ensemble de l'apprentissage automatique basé sur des réseaux de neurones artificiels avec de nombreuses couches cachées. La figure 2.1 illustre l'intelligence artificielle et ses techniques.

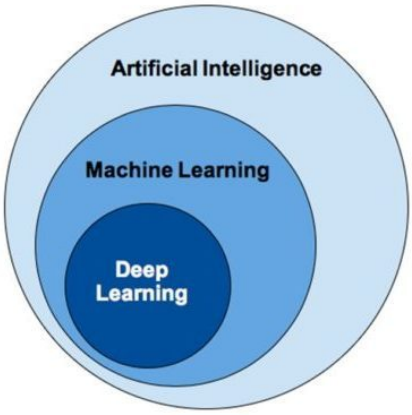

FIGURE 2.1 – Intelligence artificielle, machine learning et deep learning

2.3 Apprentissage automatique (Machine Learning (ML))

2.3.1 Définition de l'Apprentissage automatique

L'apprentissage automatique (machine learning) est défini comme un domaine de recherche et de développement au sein de l'intelligence artificielle (IA), axé sur la conception et l'implémentation de méthodes permettant aux systèmes informatiques d'apprendre automatiquement à partir de données et de réaliser des tâches spécifiques sans une programmation explicite pour chaque tâche. Cette discipline repose sur l'élaboration de modèles mathématiques et statistiques sophistiqués capables de détecter des structures et des motifs dans les données et de les utiliser pour prendre des décisions ou faire des prévisions. L'objectif fondamental de l'apprentissage automatique est de permettre aux systèmes informatiques de s'améliorer progressivement à travers l'expérience accumulée, facilitant ainsi une adaptabilité accrue et une performance optimisée dans divers domaines d'application, de la reconnaissance de motifs complexes à la prise de décision autonome [27].

2.3.2 Importance de l'apprentissage automatique

Le ML révolutionne divers secteurs en apportant des solutions innovantes pour résoudre des problèmes complexes et améliorer l'efficacité des opérations. Il permet non seulement d'automatiser des tâches répétitives, mais aussi de fournir des analyses prédictives précieuses, facilitant ainsi une prise de décision plus éclairée et une meilleure allocation des ressources. Sa capacité à traiter et analyser de vastes ensembles de données conduit à des avancées significatives dans des domaines tels que la santé, la biologie et les géosciences. De plus, l'intégration des connaissances spécifiques à chaque domaine améliore la signification et la cohérence scientifique des modèles, en faisant d'eux des outils inestimables pour la découverte scientifique et les applications pratiques. Avec l'augmentation continue des capacités de calculs et de la disponibilité des données, l'importance et l'impact du machine learning devraient encore s'accroître, promettant des contributions encore plus significatives à l'avenir [28].
Les systèmes de recommandation utilisent généralement les techniques du machine learning pour analyser les comportements passés des utilisateurs dans le but de personnaliser

les suggestions de produits, améliorant ainsi de manière significative l'expérience utilisateur.

2.3.3 Les diverses approches de l'apprentissage automatique

Diverses approches d'apprentissage automatique existent et chacune utilise des algorithmes spécifiques. Parmi les approches les plus connues figurent l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage par renforcement.

2.3.3.1 Apprentissage supervisé

L'apprentissage supervisé est une méthode d'apprentissage automatique caractérisée par la création d'un algorithme qui apprend une fonction prédictive. Cela est possible grâce à un entraînement à partir de données étiquetées, qui incluent un groupe de variables d'entrée accompagnées de leurs variables de sortie respectives. L'ensemble de données d'entrée est divisé en un ensemble d'entraînement et un ensemble de test. L'ensemble d'entraînement contient les variables de sortie qui doivent être prédites ou classifiées. Les algorithmes apprennent à partir de motifs présents dans l'ensemble de données d'entraînement et appliquent ces connaissances à l'ensemble de test pour effectuer des prédictions ou des classifications.

La classification en apprentissage automatique a pour objectif de prédire une variable cible catégorique ou discrète. Par exemple, elle permet de classer une image dans des catégories telles que chat, chien, oiseau. Les algorithmes de classification apprennent une fonction qui associe les variables d'entrée à une classe ou catégorie de sortie. Parmi les types courants d'algorithmes de classification, nous citons ; les classifieurs linéaires, les machines à vecteurs de support (SVM), les arbres de décision et les forêts aléatoires. La figure 2.2 représente un Exemple d'algorithme de classification.

FIGURE 2.2 – Exemple d'algorithme de classification

La régression en apprentissage automatique a pour objectif la prédiction d'une variable cible sous forme de valeur numérique continue. Par exemple, elle permet de prédire le prix d'une maison en fonction de sa surface, du nombre de chambres, etc. Les algorithmes de régression apprennent une fonction qui associe les variables d'entrée à une valeur de

sortie numérique. Parmi les algorithmes de régression populaires, nous citons la régression linéaire, la régression logistique et la régression polynomiale. La figure 2.3 représente un exemple d'algorithme de régression.

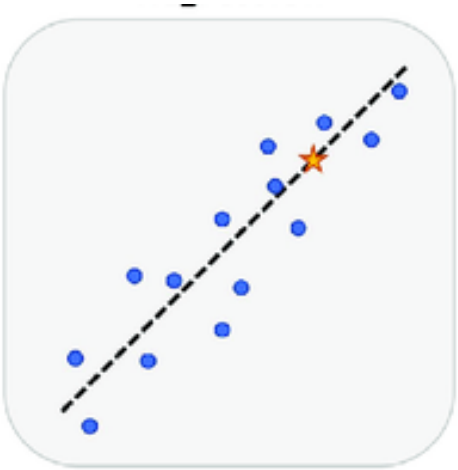

FIGURE 2.3 – Exemple d'algorithme de régression

Algorithmes d'apprentissage supervisé
Voici quelques exemples d'algorithmes d'apprentissage supervisé.

- **K-NN (k plus proches voisins)** est un algorithme de classification où la classe d'un nouvel exemple est déterminée par vote majoritaire parmi ses k voisins les plus proches dans l'espace des caractéristiques [29]
- **SVM (Support Vector Machine)** est un algorithme de classification qui trouve l'hyperplan optimal séparant les classes dans un espace multidimensionnel, en maximisant la marge entre les classes [29].
- **Régression linéaire**est un algorithme de régression qui modélise la relation linéaire entre une variable dépendante et une ou plusieurs variables indépendantes [30].
- **Régression polynomiale** étend la régression linéaire en incluant des termes polynomiaux (par exemple, quadratiques, cubiques) pour capturer des relations non linéaires entre les variables [30].

2.3.3.2 Apprentissage non supervisé

L'apprentissage non supervisé, contrairement à l'apprentissage supervisé, ne nécessitent pas de données étiquetées ni d'intervention d'un enseignant pour fournir des réponses correctes. En d'autres termes, dans l'apprentissage non supervisé, les données ne sont pas associées à des étiquettes ou des sorties cibles spécifiques. Les algorithmes non supervisés explorent les données pour découvrir des structures, des relations ou des groupes cachés, sans qu'un enseignant ne leur indique explicitement ce qu'ils doivent apprendre. Les algorithmes d'apprentissage non supervisés apprennent quelques caractéristiques à partir des données. Lorsque de nouvelles données sont introduites, ils utilisent les caractéristiques précédemment apprises pour reconnaître la classe de la nouvelle donnée. Les modèles d'apprentissage non supervisé sont utilisés pour trois tâches principales : le clus-

tering, l'association et la réduction de dimensionnalité. La figure 2.4 présente un exemple d'application d'un algorithme de clustering.

FIGURE 2.4 – Exemple d'application d'un algorithme de clustering.

Algorithmes d'apprentissage non supervisé
Voici quelques exemples d'algorithmes d'apprentissage non supervisé.

- **K-means** est un algorithme de clustering qui partitionne les données en k clusters en minimisant la variance intra-cluster. Il est efficace pour les ensembles de données où le nombre de clusters est connu à l'avance [29].
- **Hierarchical clustering(Clustering hiérarchique)**le clustering hiérarchique construit une hiérarchie de clusters en regroupant progressivement les données en sous-groupes, soit de manière ascendante (agglomératif) soit de manière descendante (divisive) [31].

2.3.3.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé se situe à mi-chemin entre l'apprentissage supervisé et l'apprentissage non supervisé. Son objectif principal est de pallier les inconvénients de ces deux types d'apprentissage. L'apprentissage supervisé nécessite une grande quantité de données d'entraînement étiquetées pour classifier les données de test. Ce processus est coûteux en temps. À l'inverse, l'apprentissage non supervisé ne requiert aucune donnée étiquetée. Il regroupe les données en fonction de la similarité des points de données en utilisant soit le clustering, soit l'approche du maximum de vraisemblance [32].

2.3.3.4 Apprentissage par renforcement

L'apprentissage par renforcement étudie comment un agent logiciel doit choisir ses actions dans un environnement afin de maximiser une notion de récompense cumulative. Il se base sur l'idée que le comportement d'un agent doit être façonné par les conséquences de ses actions, plutôt que d'être explicitement programmé avec le comportement correct. L'agent apprend en interagissant avec l'environnement, en recevant des récompenses positives ou négatives pour ses actions, et en ajustant son comportement en conséquence pour

maximiser la récompense cumulative attendue à long terme [33]. La figure 2.5 présente un Exemple d'algorithme d'apprentissage par renforcement.

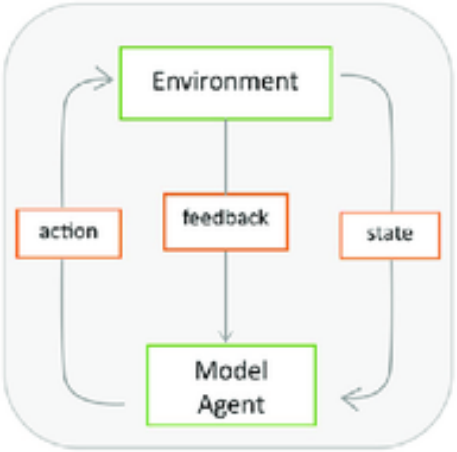

FIGURE 2.5 – Exemple d'algorithme d'apprentissage par renforcement.

2.4 Apprentissage profond (Deep Learning (DL))

2.4.1 Définition

L'apprentissage profond, également appelé deep learning, est un sous-domaine de l'IA qui utilise des réseaux de neurones possédant plusieurs couches de neurones cachées. Ces réseaux, dotés de nombreux paramètres, sont entraînés sur de vastes quantités de données pour résoudre des tâches complexes. Parmi les architectures couramment utilisées, on trouve les auto-encodeurs empilés, les réseaux de croyance profonde, les machines Boltzmann profondes et les réseaux de neurones convolutifs. En exploitant la puissance des réseaux de neurones artificiels, l'apprentissage profond ouvre de nouvelles perspectives pour l'intelligence artificielle et la résolution de problèmes complexes. En effet, l'apprentissage profond offre une ressource précieuse pour les chercheurs et une référence précieuse pour les applications en intelligence artificielle [34], en permettant à divers appareils d'apprendre à reconnaître des schémas complexes et d'effectuer des tâches cognitives avancées sans ou avec peu d'intervention humaine. Ainsi, le DL est utilisé dans divers domaines tels que la reconnaissance d'images, la reconnaissance vocale, la traduction automatique, le traitement automatisé du langage, et la recommandation de contenu [35].

2.4.2 L'importance de l'apprentissage profond

L'apprentissage profond est particulièrement efficace pour traiter de grandes quantités de données non structurées ou complexes. Il permet aussi d'automatiser des tâches auparavant difficiles ou impossibles à réaliser de manière algorithmique. Dans le domaine des systèmes de recommandation, l'apprentissage profond est essentiel pour améliorer la personnalisation, la précision et l'efficacité des SR. L'apprentissage profond suscite un grand intérêt au sein de la communauté des chercheurs en IA, favorisant ainsi l'émergence de nouvelles idées, d'innovations et de progrès dans le domaine de l'IA.

En résumé, l'apprentissage profond est important, car il fournit des solutions puissantes à des problèmes complexes, ouvre les portes à de nouvelles opportunités et révolutionne de nombreux aspects de notre vie quotidienne et de nos industries.

2.4.3 Réseaux de neurones artificiels (Artificial neural networks (ANN)

Les réseaux de neurones artificiels, inspirés du fonctionnement du cerveau humain, sont des modèles informatiques composés de plusieurs unités de traitement interconnectées, appelées neurones artificiels. Ces neurones sont organisés en couches successives, recevant des signaux d'entrée, effectuant des calculs sur ces signaux, et transmettant des signaux de sortie à d'autres neurones. L'évolution du modèle de réseau de neurones artificiels en tant que méthode d'apprentissage automatique imitant le cerveau humain est illustrée par la pyramide de l'intelligence artificielle suivante, montrant le cheminement du ML vers l'ANN et aboutissant au Deep Learning [3]. La figure 2.6 illustre pyramide de l'intelligence artificielle.

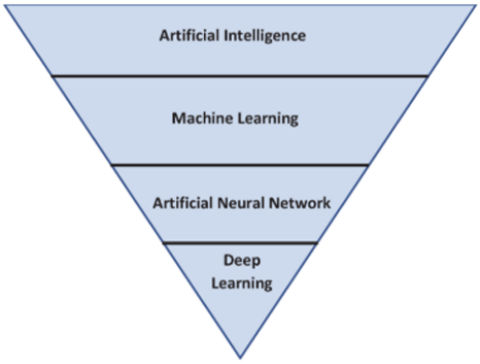

FIGURE 2.6 – Pyramide de l'intelligence artificielle [3].

2.4.4 L'architecture des réseaux de neurones profonds

Dans le domaine du DL, les réseaux de neurones profonds sont utilisés pour représenter et apprendre les hiérarchies de fonctionnalités. L'architecture d'un réseau neuronal profond se compose de plusieurs couches, parmi lesquelles figurent la couche d'entrée, la couche cachée et la couche de sortie.

2.4.4.1 La couche d'entrée

C'est la couche initiale qui est chargée de recevoir les données d'entrée non traitées, qui peuvent inclure des images, du texte ou des signaux audio. Au sein de cette couche, chaque nœud correspond à un attribut ou à une caractéristique distincte des données.

2.4.4.2 La couche cachée

Elle se trouve entre la couche d'entrée et la couche de sortie. Le rôle d'une couche de sortie est d'effectuer des calculs visant à extraire des représentations plus complexes et

conceptuelles des données. Ces couches sont constituées de nombreux neurones interconnectés, chacun étant chargé de calculer les poids et les biais pour traiter avec précision les signaux d'entrée.

2.4.4.3 La couche de sortie

Les résultats finaux ou prédictions du modèle sont produits par la couche de sortie. La structure de la couche de sortie est déterminée par la nature du problème à résoudre à savoir, une classification multi-classe ou d'une prédiction numérique.

La figure 2.7 présente l'architecture des réseaux de neurones profonds.

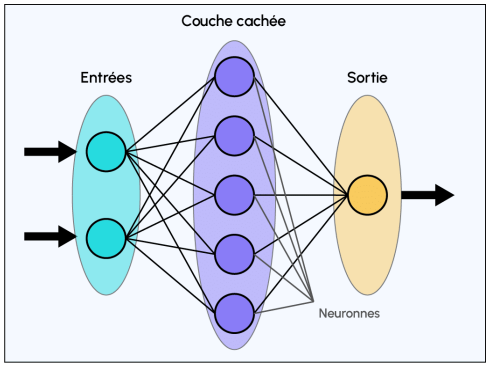

FIGURE 2.7 – Architecture des réseaux de neurones profonds [4].

2.4.5 Les différentes d'architecture dans les réseaux de neurones profonds

Il existe plusieurs architectures de réseaux de neurones profonds, chacune adaptée à des tâches spécifiques et présentant des structures organisationnelles différentes. nous citions dans ce qui suit le architectures les plus couramment utilisées.

2.4.5.1 Réseaux de neurones convolutifs (Convolutional Neural Networks (CNN)

Les CNN sont une classe de réseaux neuronaux artificiels devenus dominants dans diverses tâches de vision par ordinateur et de traitement d'images dans divers domaines notamment la radiologie, la reconnaissance d'objets, la classification d'images, la détection d'objets, la segmentation sémantique, et bien d'autres.

Les CNN sont conçus pour apprendre automatiquement et de manière adaptative les hiérarchies spatiales des caractéristiques par rétropropagation en utilisant plusieurs blocs de construction, tels que des couches de convolution, des couches de pooling et des couches entièrement connectées. La fonction principale des couches de convolution est d'analyser les données d'entrée afin d'identifier et d'isoler des modèles particuliers, tandis que les couches de pooling servent à réduire la complexité des données tout en préservant leurs éléments cruciaux. Les couches entièrement connectées, en revanche, jouent un rôle crucial dans la fusion des fonctionnalités extraites des couches précédentes pour générer le résultat final [36]. La figure 2.8 présente Réseaux de neurones convolutifs (CNN).

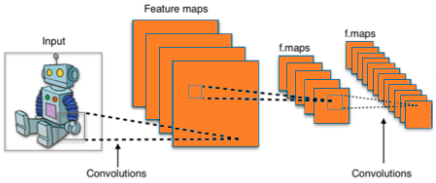

FIGURE 2.8 – Réseaux de neurones convolutifs (CNN) [5].

2.4.5.2 Les réseaux de neurones récurrents (Recurrent Neural Networks (RNN))

Un autre type d'architecture de réseau neuronal, appelé réseaux de neurones récurrents (RNN), a été spécialement conçu pour résoudre les problèmes d'apprentissage impliquant une relation directe entre les informations passées et les prédictions futures. Ces types d'exemples séquentiels sont courants dans diverses applications du monde réel, telles que la modélisation linguistique, où les mots précédents d'une phrase sont utilisés pour prédire le mot suivant. De même, dans les prévisions boursières, l'évolution des actions au cours de la dernière heure, de la dernière journée ou de la dernière semaine est utilisée pour prévoir les tendances futures. Les RNN sont particulièrement adaptés aux tâches impliquant des séries chronologiques ou des données séquentielles [37]. La figure 2.9 illustre Les réseaux de neurones récurrents (RNN).

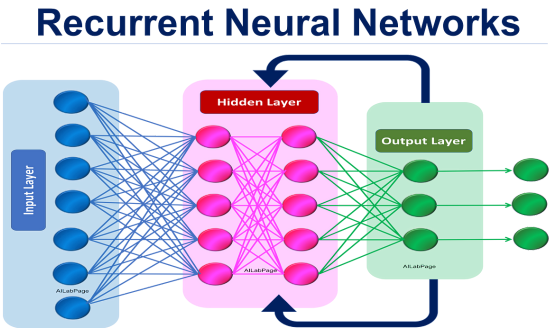

FIGURE 2.9 – Les réseaux de neurones récurrents (RNN) [6].

2.4.5.3 Les réseaux de neurones profonds (Deep Neural Network (DNN))

Les réseaux de neurones profonds (DNN) constituent une catégorie avancée de modèles d'apprentissage automatique ayant radicalement transformé le domaine de l'intelligence artificielle et des sciences cognitives. Leur architecture complexe, caractérisée par de multiples couches d'unités de traitement, leur permet d'acquérir des représentations de données complexes. Les DNN inspirés par la neurobiologie et les modèles de réseau neuronal profond, ont évolué pour devenir un outil puissant dans les domaines de l'apprentissage automatique et de l'intelligence artificielle, capables d'estimer les fonctions et les dynamiques en se basant sur des exemples [38]. La figure 2.10 présente Les réseaux de neurones profonds.

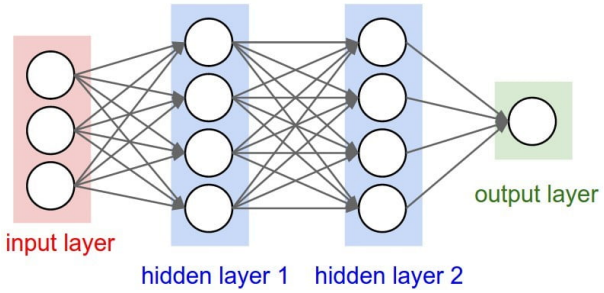

FIGURE 2.10 – Les réseaux de neurones profonds (DNN)[7].

2.5 Conclusion

En conclusion, l'intelligence artificielle, englobant l'apprentissage automatique et le deep learning, joue un rôle central et crucial au sein de la révolution numérique contemporaine. Cette évolution rapide ouvre de vastes horizons pour l'innovation technologique, transformant profondément divers secteurs de notre quotidien. Ces avancées ne se limitent pas à l'optimisation des systèmes existants, mais favorisent également la création de nouvelles applications et solutions novatrices. Les technologies avancées en IA et en apprentissage automatique offrent des outils puissants pour améliorer les processus opérationnels et affiner les décisions stratégiques, en répondant aux évolutions des besoins sociétaux et en repoussant les limites de ce qui est technologiquement réalisable. Les systèmes de recommandation basés sur l'IA, présentés dans le chapitre précédent, illustrent comment ces technologies continuent à redéfinir les interactions entre l'homme et la machine vers davantage de fluidité et de productivité. Le chapitre suivant présentera certains travaux de recherches récentes consacrés aux systèmes de recommandation basés sur l'IA.

Chapitre 3

État de l'art

3.1 Introduction

Les systèmes de recommandation se révèlent être un outil précieux, assistant les utilisateurs dans leur quête de produits, services, informations ou contenus en parfaite adéquation avec leurs besoins et préférences spécifiques. Cela fait des SR un axe de recherche dynamique qui a suscité l'intérêt de nombreux chercheurs et praticiens. En conséquence, ce chapitre présente divers travaux qui se sont axés sur les SR et plus particulièrement les SR basés sur des techniques de l'IA.

3.2 État de l'art

L'utilisation de l'intelligence artificielle dans le domaine de l'informatique a considérablement amélioré les performances des systèmes de recommandation. Dans cette section, nous présentons quelques travaux de recherches portant sur la recommandation basée sur l'IA.

Dans cet article [39], Takumi Fujimoto et al traitent de la problématique des recommandations de livres pour les personnes lisant peu ou n'ayant pas d'historique de lecture. L'approche proposée associe de manière innovante l'analyse du contenu textuel des livres à celle des émotions des utilisateurs. Le processus consiste à collecter des tweets et des critiques de livres, puis à les analyser avec Sentence-BERT pour générer des vecteurs de contenu et ML-ASK pour des vecteurs d'émotions/sentiments. Des scores de similarité de contenu et d'émotions avec les centres d'intérêt de l'utilisateur sont calculés, permettant de recommander les livres avec les meilleurs scores. Contrairement aux techniques classiques de filtrage collaboratif, basé sur le contenu ou hybride. L'évaluation repose sur huit critères à savoir ; précision, nouveauté, intention, reflet du contenu/sentiments, diversité, utilité, et satisfaction. Pour cela, les auteurs utilisent 33 822 critiques de BookMeter et des tweets ciblés. Les résultats montrent une performance supérieure aux recommandations Amazon sur tous les critères sauf la diversité, validant l'efficacité pour les lecteurs réguliers et occasionnels en combinant le contenu et les émotions. Les points forts de cette approches sont la personnalisation émotionnelle et la recommandation sans historique, tandis que ces limites incluent la dépendance aux données textuelles et le manque de diversité. En conclusion, l'évaluation a démontré l'efficacité de cette approche pour aligner les recommandations de livres avec les centres d'intérêt et des émotions des utilisateurs.

Arunruviwat et al [2] traitent le problème des utilisateurs qui ont du mal à localiser les livres pertinents dans les bibliothèques universitaires en raison de la croissance exponentielle du nombre de livres et de l'inefficacité des systèmes de recommandation actuels face au problème du démarrage à froid. La solution proposée pas ces auteurs implique la création d'un système de recommandation hybride combinant filtrage collaboratif et filtrage basé sur le contenu. Pour améliorer la précision des recommandations, ils utilisent les livres recommandés dans les programmes de cours. La méthodologie comprend la collecte de données de la bibliothèque via le système Sierra, l'acquisition de programmes de cours auprès de la Faculté d'ingénierie, et l'obtention de descriptions de livres via l'API Google Books. Ces données sont ensuite préparées et exploitées pour créer des systèmes de filtrage basés sur le contenu et collaboratifs. Ces deux systèmes sont par la suite intégrés grâce à une méthode de moyenne pondérée. En analysant une grande quantité de données, notamment 139 214 relevés de livres, des historiques d'emprunt, et 553 relevés de programmes de cours, une évaluation de perfomance a été menée en utilisant le RMSE (Root Mean Squared Error) comme mesure. La technique hybride proposée a démontré une précision de prédiction supérieure avec un RMSE de 1,2247, surpassant les techniques de filtrage basé sur le contenu (1,6945) et de filtrage collaboratif (1,4137). Ces résultats démontrent l'efficacité de l'intégration hybride dans la fourniture de recommandations précises et adaptées aux besoins des utilisateurs.

L'étude présentée dans [40] par Yizhu Zhao et al, traite la problématique du manque de personnalisation et de pertinence des systèmes de recommandation de livres traditionnels dans les bibliothèques. Pour répondre à ces défis, les auteurs proposent une approche novatrice qui utilise la détection d'expressions faciales. Cette méthode consite en premier à réaliser en temps réel l'acquisition des données faciales via une caméra tandis que l'utilisateur parcourt plusieurs livres. Ensuite vient le pré-traitement des images faciales (recadrage, conversion en niveaux de gris, normalisation, égalisation) recueillies et de la détection des visages à l'aide de l'algorithme Haar-Adaboost implémenté dans OpenCV. Les expressions faciales sont alors segmentées grâce à un modèle CNN pré-entraîné appelé « mini-Xception », qui prédit les probabilités de 7 émotions différentes (heureux, surpris, triste, dégoûté, effrayé, en colère, neutre). Si l'expression prédite est « heureux » ou « surpris », le système en déduit que l'apprenant aime le type de livre consulté et lui proposera d'autres livres similaires. Les travaux connexes ont été menés sur la reconnaissance d'expressions, les CNNs pour la Vision par ordinateur et les approches de recommandation de livres existantes. La validation de l'approche s'est faite suivant deux critères : le taux de précision déduisant les préférences du type de livres à partir des expressions faciales (expérience I) et le taux de succès pour recommander des livres qui plaisent aux utilisateurs (expérience II). Les données ont été collectées en temps réel dans le cadre de l'expérience auprès de 10 utilisateurs volontaires. Pour cela les auteurs utilisent une interface pour afficher les informations concernant les livres, des caméras home pour capturer les expressions faciales en temps réel, et un PC exécutant le pipeline d'analyse d'expressions OpenCV et le modèle CNN mini-Xception. Les bénéfices d'une telle approche sont la personnalisation en temps réel et l'authenticité du data-donneur, et le point négatif d'une telle approche est l'aspect flou des expressions et l'aspect des implications éthiques.

Dans [41] les auteurs présentent une étude sur les systèmes de recommandation littéraire. Le problème étudié est de savoir comment recommander des livres pertinents aux

utilisateurs, dans un contexte d'abondance d'information. L'approche présentée dans cet article est de faire une revue des méthodes appliquées aux BooK Recommendation System (BRS). Cela va des anciennes approches de filtrage collaboratif ou de contenu aux plus récentes méthodes basées sur l'apprentissage machine. La validation des stratégies se fait généralement en regard des métriques standards comme la précision, le rappel, F1-score, etc. Dans l'ensemble, la force des BRS est une meilleure gestion des catalogues des bibliothèques, une recommandation des utilisateurs au service des librairies numériques, une montée en flèche de leurs ventes en ligne, etc. Mais il subsiste toujours des problèmes tels que le démarrage à froid, les information rare, la recommandation faiblement diversifiée (la sur-spécialisation), etc. pour lesquels l'on peut encore améliorer les performances et la confiance des utilisateurs qui utilisent les BRS.

Dans cet article [42], Neha Rani et al traitent la problématique de la confiance des utilisateurs envers les systèmes de recommandation contextuels (CARS) par rapport aux SR standards dans le domaine éducatif. L'approche proposée est une étude par enquêtes visant à évaluer et comparer la confiance et la perception des utilisateurs des CARS et des SR standards. Elle consiste à recruter des participants, à concevoir l'enquête, présenter des exemples de CARS et SR, évaluer la confiance et la perception, puis effectuer une analyse quantitative et qualitative des résultats obtenus. L'évaluation des résultats a permis d'identifier une différence significative de confiance des utilisateurs, soulignant l'importance de traiter les problématiques de confidentialité, de sécurité des données, et de transparence pour renforcer la confiance envers les CARS. Les données utilisées sont des questionnaires soumis à des étudiants de l'Université de Floride. En termes d'avantages, cette approche mesure directement la perception des utilisateurs et évalue les facteurs spécifiques de méfiance. Ses limites résident dans la taille d'échantillon réduite et le manque de diversité de l'échantillon. Bien que les CARS puissent améliorer l'expérience d'apprentissage, la méfiance des utilisateurs pourrait limiter leur efficacité.

Qomariyah et al [43] s'intéressent au développement de SR dédiés à la recommandation de formations en ligne prenant en compte le style (la méthode) d'apprentissage préférés des étudiants inspirés des cinq sens. L'approche proposée utilise le modèle APARELL basé sur la programmation logique binaire. Le module d'apprentissage exploite APRIREL, un cadre existant utilisant des modèles de classification pour apprendre les préférences des utilisateurs exprimées sous forme de relations « betterthan » entre paires de styles d'apprentissage caractérisés par cinq critères (organisation, interaction, approche, présentation, interaction sociale). Le module de recommandation sélectionne ensuite les contenus correspondant au style d'apprentissage non dominé par d'autres selon les règles apprises. Les avantages de cette approche incluent l'utilisation d'un modèle connu et déjà testé, le respect de la confidentialité, une perspective d'application future et une solution au problème du démarrage à froid. Cependant, l'article ne fournit pas de détails sur les données, l'implémentation, les mesures de performance utilisées, et la résolution du démarrage à froid reste manuelle. Cette approche originale vise à personnaliser les recommandations d'E-learning en fonction des préférences d'apprentissage des étudiants.

Dans [44], Dhiman Sarma et al examinent les problèmes liés aux systèmes de recommandation de livres personnalisés basés sur le filtrage collaboratif. Ces systèmes exploitent les notations des utilisateurs pour générer des recommandations qui se révèlent inefficace lorsque des données manquent, notamment lorsque les utilisateurs cessent de noter les

livres après avoir quitté le service. Pour pallier cette lacune, les auteurs proposent une approche fondée sur une technique de clustering non supervisé utilisant l'algorithme K-means et pour partitionner les livres en clusters homogènes en fonction de leurs scores et des préférences des utilisateurs. La similarité Cosinus entre les clusters permet ensuite de recommander aux utilisateurs les livres analogues à ceux d'un cluster spécifique. Pour valider leur méthodologie, ils évaluent les performances prédictives du système en calculant sur dix jeux de données distincts des métriques telles que la sensibilité (taux de vrais positifs), la spécificité (taux de vrais négatifs) et le score F1 (moyenne harmonique de la précision et du rappel). Ils tracent également la courbe ROC (caractéristique opérationnelle du récepteur) afin de visualiser graphiquement le compromis sensibilité/spécificité des classificateurs. Les résultats démontrent une spécificité supérieure à la sensibilité, indiquant une meilleure capacité à filtrer les livres non pertinents. La proximité des courbes ROC avec la diagonale du classificateur parfait témoigne également de bonnes performances prédictives globales. Les auteurs en déduisent que leur approche basée sur le contenu produit des recommandations plus précises que les systèmes collaboratifs classiques. Parmi les avantages, on peut citer l'utilisation du contenu pour pallier l'absence de notations, l'emploi d'algorithmes robustes de clustering et de similarité cosinus, ainsi qu'une évaluation rigoureuse sur diverses métriques. Cependant, certaines limites subsistent, comme l'absence de prise en compte d'informations contextuelles supplémentaires, des performances potentiellement dégradées pour les nouveaux livres/utilisateurs et un manque de comparaison directe aux systèmes existants.

Dans cet article [45], Ashwini KB et al présentent un système de recommandation de film hybride combinant différentes techniques d'apprentissage machine. Après une phase de recherche multicritères par clustering des films avec l'algorithme des K-means sur la base de paramètres utilisateurs comme l'acteur et le genre, le SR proposé exploite des méthodes de factorisation matricielle et de filtrage collaboratif. Les algorithmes SVD (Singular Value Decomposition) et son extension SVD++ réalisent une décomposition pour réduire la dimensionnalité et déduire les préférences implicites des utilisateurs. Le filtrage collaboratif par similarité Cosinus identifie quant à lui les profils d'évaluation proches pour recommander les films les mieux notés par les utilisateurs similaires. Cette approche hybride vise à tirer parti des avantages complémentaires des différents algorithmes pour améliorer la pertinence des résultats. Cependant, l'article ne présente pas d'évaluation quantitative pour la comparaison des performances du SR proposé par rapport à d'autres systèmes de référence. De plus, certaines données contextuelles supplémentaires comme les métadonnées des films ou les informations démographiques des utilisateurs ne sont pas prises en compte, alors qu'elles pourraient potentiellement accroître la qualité des recommandations.

Cette étude [46] présentée par N. Giridharan et al. traite la problématique de la conception de systèmes de recommandation de contenus cinématographiques personnalisés aux préférences et à l'historique des utilisateurs en ligne. L'approche proposée implémente une méthodologie hybride combinant les techniques de filtrage collaboratif et de filtrage basé sur les caractéristiques du contenu. Le processus de cette approche utilise les ensembles de données de MovieLens et TMDB, contenant les évaluations et les métadonnées de films. Le filtrage collaboratif repose sur un algorithme de clustering par K-means permettant d'identifier les groupes d'utilisateurs aux préférences homogènes. Le filtrage par le contenu effectue une correspondance entre les attributs des films et le profil

de l'usager. La validation empirique porte sur l'analyse de la pertinence et de la précision prédictive des recommandations générées, en étudiant notamment les problématiques du démarrage à froid utilisateur et élément. Les avantages majeurs mis en évidence sont la personnalisation de l'expérience utilisateur, la prise en compte des préférences implicites et la capacité à formuler des recommandations sans historique initial. Néanmoins, certaines limitations subsistent, comme les problèmes résiduels de démarrage à froid, les données déséquilibrées pour le filtrage collaboratif et la nécessité d'accroître davantage la précision prédictive. Des pistes d'amélioration sont proposées, impliquant l'utilisation d'algorithmes de clustering et l'hybridation avec d'autres caractéristiques discriminantes.

Dans cette étude [47] présentée par Pegah Malekpour et al sur les systèmes de recommandation dans le domaine du e-commerce, la problématique centrale réside dans la nécessité de développer des ontologies spécifiques pour améliorer la pertinence et la personnalisation des recommandations fournies aux utilisateurs, en se concentrant particulièrement sur les plateformes de commerce électronique des petites et moyennes entreprises en Thaïlande. L'approche proposée combine des techniques d'analyse de contenu avec des méthodes de machine learning, telles que le clustering et l'analyse de motifs, pour extraire des règles de connaissance pertinentes et soutenir le moteur de recommandation. Cette approche permet de recommander des sites Web pertinents qui répondent aux critères et aux besoins des utilisateurs de manière plus ciblée. La validation de cette approche soulève des défis liés à la collecte et à l'analyse des données essentielles provenant des sites de e-commerce des PME (Petites et Moyennes Entreprises), en raison de la diversité et de l'évolution constante de leurs structures. Les points positifs de cette approche résident dans l'amélioration de la personnalisation des recommandations, offrant ainsi une expérience utilisateur plus enrichissante. Cependant, les points négatifs de cette approche incluent la complexité accrue liée à la collecte et à l'analyse des données spécifiques à ces sites, ce qui peut potentiellement limiter la pertinence et l'efficacité des recommandations en ligne.

Zhiying et al [48] examinent les algorithmes de recommandation de services Web, mettant l'accent sur les approches basées sur les graphes de connaissances, en soulignant la prédominance du filtrage collaboratif (CF) malgré ses limitations comme la rareté des données et les problèmes de démarrage à froid. Les travaux antérieurs ont tenté d'améliorer les recommandations en intégrant des facteurs supplémentaires comme la localisation, les avis textuels et la qualité de service (QoS), tandis que l'utilisation de l'ontologie reste rare en raison de sa complexité. L'apprentissage sur les graphes de connaissances, via l'algorithme Trans H, représente une nouvelle perspective pour résoudre ces défis en intégrant des informations sémantiques. L'approche proposée représente les services et les utilisateurs sous forme de vecteurs dans un graphe de connaissances, analyse les relations entre les services, calcule des similarités pour identifier les services voisins, et utilise la descente de gradient pour calculer les préférences QoS des utilisateurs. Les utilisateurs sont classifiés en positif ou négatif selon la distance entre ces services, et deux ensembles de services notés par les voisins sont générés pour obtenir le classement final des recommandations. L'évaluation montre que Kg-WSR améliore la précision de 5 à 10% par rapport à six autres algorithmes (SCOR, UPCC, etc.), bien que son temps d'exécution soit plus élevé. Kg-WSR améliore significativement la qualité des recommandations et le démarrage à froid en utilisant des connaissances sémantiques, comme démontré sur un dataset de 873 services web liés aux voyages et à la cartographie, avec les évaluations QoS de

345 utilisateurs. Les avantages incluent une meilleure gestion du démarrage à froid et une adaptation aux habitudes des utilisateurs, tandis que les limites concernent le temps de calcul élevé et l'absence de détails techniques sur l'implémentation et les hyperparamètres.

Pour conclure cette section, nous présenterons un tableau récapitulatif des articles étudiés et présenté ci dessous.

TABLE 3.1 – Tableau synthétisant les travaux présentés dans l'état de l'art

Article	Approche proposée	Domaine	Data-set	Les avantages	Les limites
Fujimoto et al. 2022 [39]	Contenu + sentiment Sentence-BERT + ML-Ask	Livres	33822 avis + tweets	considère les centres d'intérêt et les émotions des utilisateurs, Recommandation pour lecteurs réguliers et non-lecteurs	Dépendance aux données textuelles, problème de diversité
Kavitha et Koteeswaran 2023 [41]	Filtrage divers (contenu, collaboratif, démographique) + ML	Livres	Non mentionné	Personnalisation, précision améliorée, adaptation aux préférences changeantes, optimisation de l'expérience utilisateur	Dépendance aux données, risque de sur-apprentissage, biais algorithmiques, complexité de mise en œuvre, besoin de mises à jour constantes, manque de transparence
Sarma et al. 2021 [44]	K-means + similarité cosinus	Livres	10 000 livres	Bonne pertinence des recommandations, Réduction de la dépendance aux notations utilisateurs, utilisation du contenu réel	Besoin de données exhaustives, incapacité à suivre les préférences changeantes, vulnérabilité aux données bruitées
Cao et al. 2019 [48]	TransH + relations sémantique	Services Web	873 services Web	meilleure précision, rappel, couverture, et gestion du problème de démarrage à froid	Temps de calcul plus long

TABLE 3.2 – Tableau synthétisant les travaux présentés dans l'état de l'art

Article	L'approche proposée	Domaine	Dataset	Les avantages	Les limites
Yizhu Zhao et al. 2020 [40]	Haar-Adaboost (OpenCV) + CNN	Livres	10 volontaires	La connaissance des données en temps réel	Capturer la subtilité des expressions, implications éthiques
Neha Rani et al. 2023 [42]	confiance et perception des CARS vs RS via enquêtes	Livres	Questionnaires	Mesure directe de la perception des utilisateurs, évaluation des facteurs de méfiance	Taille d'échantillon restreinte, manque de diversité des participants.
Arunruviwat et al. 2022 [2]	filtrage collaboratif + contenu	Livres	139,214 enregistrements	Résolution automatique du problème de démarrage à froid, utilisation d'un système hybride	Complexité de fusion des techniques, déploiement et maintenance difficiles
Qomariyah et al. 2019 [43]	APARELL	E-learning	Non mentionné	Personnalisation des recommandations d'apprentissage en ligne	Le problème de surspécialisation
Jadhav et al. 2022 [45]	K-means, SVD, SVD++, filtrage collaboratif	Films	Informations sur les films	Recherche précise, recommandations personnalisées, combinaison efficace des approches	Problème de démarrage à froid, disponibilité limitée des données d'entraînement, explicabilité réduite des recommandations.
Alamdari et al. 2020 [47]	K-means, filtrage basé sur le contenu	E-commerce	Les plateformes de e-commerce	Amélioration de la personnalisation des recommandations	La complexité accrue liée à la collecte et à l'analyse des données.

Les techniques de recommandation couramment adoptées incluent le filtrage collaboratif, le filtrage basé sur le contenu et leurs hybridations. Cependant, ces systèmes souffrent fréquemment du problème de démarrage à froid. En l'absence de suffisamment d'évaluations ou d'informations de la part des nouveaux utilisateurs, il devient difficile d'appliquer efficacement le filtrage collaboratif. Ce phénomène entrave l'efficacité des recommandations initiales.

Pour pallier ces défis, les approches basées sur l'IA et les techniques hybrides émergent comme des solutions prometteuses, générant des recommandations plus précises et pertinentes en tenant compte des préférences et des émotions des utilisateurs, ce qui accroît leur satisfaction et leur engagement. Les travaux sur les systèmes de recommandation s'étendent au-delà des livres et des films, touchant divers domaines comme l'apprentissage en ligne, démontrant un potentiel significatif pour transformer la manière dont les recommandations sont effectuées. Ces systèmes deviennent ainsi plus intelligents, personnalisés et adaptatifs aux besoins des utilisateurs.

Suite à l'analyse des travaux présentés dans ce chapitre, nous avons décidé de proposer une approche combinant le filtrage collaboratif et le filtrage basé sur le contenu. Cette méthode hybride vise à exploiter les avantages des deux techniques pour fournir des recommandations plus précises et personnalisées. En plus, nous envisageons d'utiliser l'algorithme K-means pour résoudre le problème du démarrage à froid. En classant les utilisateurs et les items en clusters significatifs, K-means permet de générer des recommandations initiales même en l'absence de données historiques suffisantes. Cette approche hybride améliore non seulement la qualité des recommandations, mais aussi leur pertinence dès les premières interactions avec le système.

3.3 Conclusion

Dans ce chapitre, nous avons présenté divers travaux existants dans la littérature traitant les systèmes de recommandation dans divers domaines tels que les bibliothèques universitaires, les plate formes d'apprentissage en ligne, les sites de commerce électronique et les services Web. Nous avons examiné des approches incluant l'analyse des émotions, la reconnaissance faciale et l'intégration de techniques hybrides combinant le filtrage collaboratif et le filtrage basé sur le contenu. De plus, nous avons constaté que l'utilisation de techniques d'intelligence artificielle, telles que l'apprentissage profond et l'apprentissage automatique, est prometteuse pour améliorer la qualité et la pertinence des recommandations. Cependant, ces avancées nécessitent des ressources de calcul importantes et une gestion rigoureuse des données, posant des défis significatifs pour leur mise en œuvre efficace.

Chapitre 4

Approche proposée et évaluation

4.1 Introduction

Dans ce chapitre, nous détaillons notre approche dédiée à la recommandation des livres qui renvoi des suggestions correspondant aux préférences des utilisateurs. Notre méthode combine le filtrage par contenu et le filtrage collaboratif (Hybride) pour fournir des recommandations personnalisées, en plus de prendre en charge le problème du démarrage à froid.

4.2 Plateforme et outils de développement

Dans cette partie, nous présentons les plateformes et outils de développement utilisés pour la réalisation de notre travail.

Anaconda est une distribution open source conçue pour simplifier la gestion des environnements de développement Python et R, particulièrement utiles en science des données. Elle comprend Conda, un gestionnaire de paquets et d'environnements permettant de gérer les bibliothèques et les environnements isolés. Anaconda inclut plus de 1500 paquets scientifiques, tels que NumPy, Pandas, et Scikit-learn. Elle propose également des outils comme Jupyter Notebook pour le prototypage interactif et Spyder, un IDE pour la programmation scientifique.

Google Collab Environnement de développement basé sur Jupyter Notebook, hébergé gratuitement par Google. Permet d'exécuter du code Python dans le cloud, avec accès à des GPUs.

Jupyter notebook est une application web interactive utilisée pour créer des documents intégrant du code exécutable, des visualisations et du texte explicatif, favorisant ainsi la collaboration et la reproductibilité des analyses. Sa polyvalence permet l'intégration de plusieurs langages de programmation, offrant une flexibilité essentielle pour explorer, expérimenter et communiquer des résultats de manière interactive.

Python est un langage de programmation de haut niveau prisé pour sa lisibilité et sa simplicité. Il est largement utilisé dans divers domaines, y compris la science des données et le développement Web, grâce à sa vaste bibliothèque standard et sa communauté ac-

tive. Sa portabilité et sa polyvalence en font un outil indispensable pour les développeurs à travers différentes plates-formes.

Scikit-Learn Scikit-Learn : Bibliothèque Python pour l'apprentissage automatique. Offre des outils simples et efficaces pour la classification, régression, clustering et réduction de dimensionnalité.

Matplotlib Bibliothèque de visualisation de données en Python. Permet de créer des graphiques statiques, animés et interactifs de haute qualité.

Pandas Bibliothèque Python pour la manipulation et l'analyse de données. Fournit des structures de données flexibles comme les DataFrames pour un traitement efficace.

NumPy Bibliothèque fondamentale pour le calcul scientifique en Python. Propose des outils pour travailler avec des tableaux multidimensionnels et des fonctions mathématiques avancées.

4.3 Problématique

Les systèmes de recommandation jouent un rôle essentiel dans des domaines variés tels que la gestion des objets (hôtel, véhicule), des produits culturels (livres, films, musique) et des pages Web. Cependant, ces systèmes sont confrontés à des défis majeurs qui compromettent leur efficacité. Nous nous sommes focaliser dans ce travail sur deux défis des SR à savoir la sur-spécialisation et le problème de démarrage à froid. La sur-spécialisation survient lorsque les systèmes basés sur le contenu recommandent des éléments trop similaires à ceux déjà appréciés par l'utilisateur, limitant ainsi la découverte de nouveaux contenus et réduisant la diversité des recommandations. Le problème du démarrage à froid survient lorsqu'un nouvel utilisateur ou un nouvel élément est ajouté au système sans données historiques suffisantes, rendant compliqué la formulation de recommandations pertinentes.

Dès lors, notre travail de recherche consiste à déterminer comment les systèmes de recommandation basés sur l'IA peuvent surmonter ces défis pour offrir des recommandations diversifiées et pertinentes, tout en assurant une évolutivité efficace et en atténuant le problème du démarrage à froid.

4.4 Approche proposée : Système de Recommandation Hybride proposant une solution au problème de Sur-spécilisation et de Démarrage à Froid (SR-HSDF)

L'approche proposée dans ce travail combine quatre contributions. Tout d'abord, la collecte et le Pré-traitement des données. Ensuite, le filtrage basé sur le contenu, qui prend en entrée un livre et fournit en sortie des livres similaires à recommander. Puis, le système collaboratif qui vise à diversifier les recommandations de livres et à résoudre le problème de la sur-spécialisation. Ce système utilise l'ID utilisateur en entrée pour générer

des recommandations de livres. Enfin, la résolution du problème de démarrage à froid, basée sur l'algorithme K-means permettant de surmonter les défis posés par les nouveaux utilisateurs en prenant en entrée l'âge et le pays de l'utilisateur. Ces quatre contributions sont combinées pour former notre approche SR-HSDF offrant des recommandations de livres personnalisées et variées. Le schéma présenté dans la figure 4.1 résume l'architecture de l'approche proposée SR-HSDF.

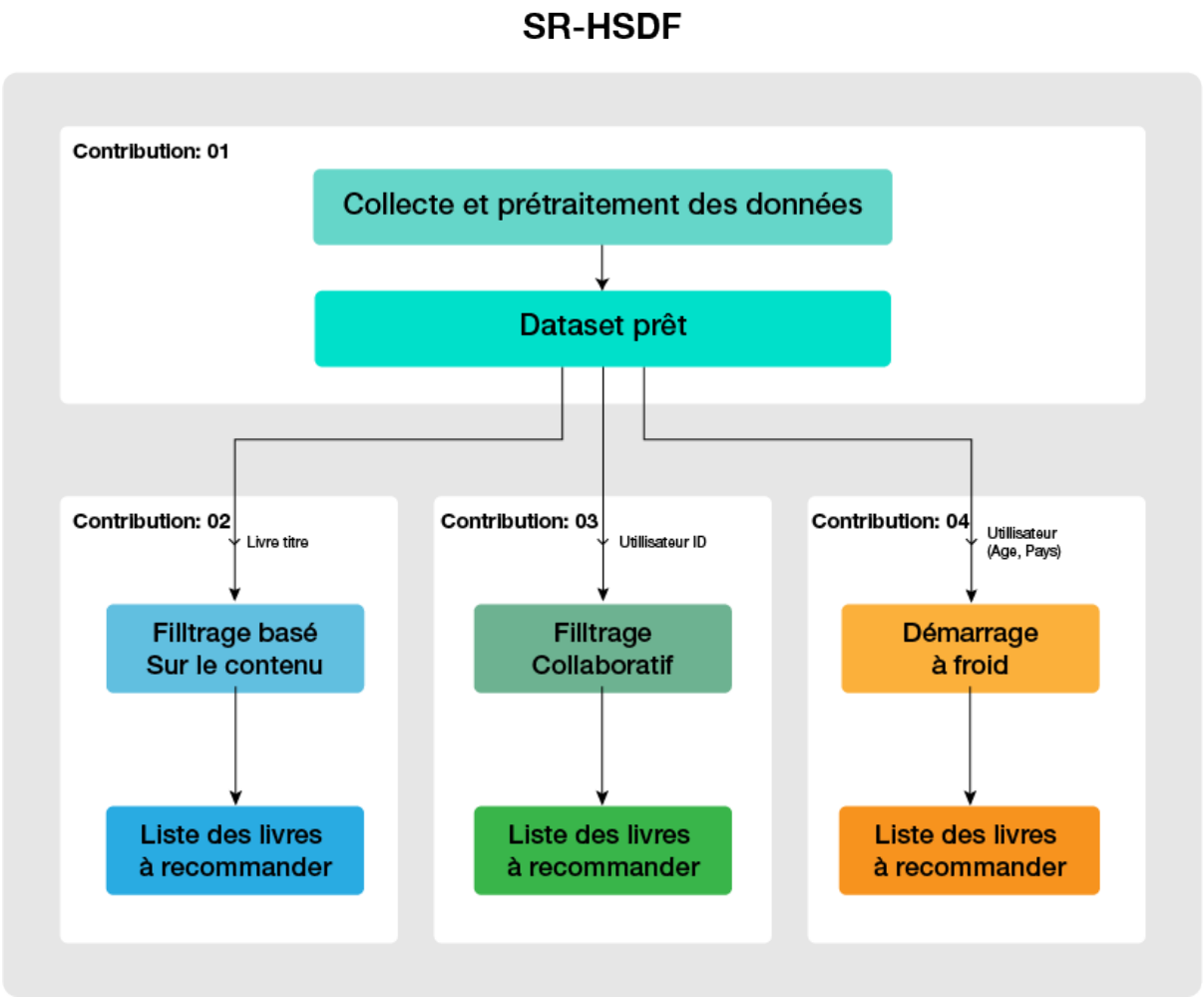

FIGURE 4.1 – L'architecture de SR-HSDF

4.4.1 Contribution 01
Collecte et Pré-traitement des données

Dans cette partie, nous nous concentrons sur la collecte de données pertinentes et leur Pré-traitement.

4.4.1.1 La Collecte des données

Nous avons tenté de trouver un dataset local à Bejaia dans la bibliothèque de l'université de Bejaia ainsi que dans la bibliothèque centrale de Bejaia, située juste à côté de la radio Soummam. Les deux bibliothèques disposent d'un système d'information qui sauvegarde les ouvrages empruntés par les utilisateurs. Le problème est que ces deux

sources de données manquent d'une information cruciale pour les SR à savoir, les avis des utilisateurs après la lecture des livres. Nous avons donc opté pour l'utilisation d'un dataset (Book Recommendation Dataset [49]) comprenant trois tables Books (Livres), Users (Utilisateurs) et Ratings (Avis). Ce dataset est très riche et contient plus de 270 000 livres.

- La table Books inclut les attributs suivants : ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, et Image-URL-L (4.2).
- La table Users qui contient : User-ID, Location, Age, (Figure 4.3).
- La table Ratings contient : User-ID, ISBN, Book-Rating (Figure 4.4).

Nous avons chargé les données à partir d'un fichier CSV en utilisant la fonction read_csv de la bibliothèque Pandas. Le résultat est présenté dans la figure 4.2.

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...

FIGURE 4.2 – Aperçu du contenu de la table livre

	User-ID	Location	Age
1	2	stockton, california, usa	18.0
3	4	porto, v.n.gaia, portugal	17.0
5	6	santa monica, california, usa	61.0
9	10	albacete, wisconsin, spain	26.0
10	11	melbourne, victoria, australia	14.0

FIGURE 4.3 – Aperçu du contenu de la table livre utilisateurs

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3

FIGURE 4.4 – Aperçu du contenu de la table des Avis

4.4.1.2 Pré-traitement des données

Dans cette section, nous allons présenter les différentes étapes de Pré-traitement des données utilisées dans notre approche SR-HSDF, qui sont comme suit :

- **Gestion des valeurs manquantes**
  Pour gérer les valeurs manquantes, nous avons appliqué la méthode **dropna()** afin de supprimer les lignes du dataset contenant des valeurs manquantes. Ce processus garantit que les données utilisées dans notre approche sont complètes et fiables.
- **Ajout de la colonne catégories**
  La catégorisation des livres dans la table Livres est essentielle pour le filtrage basé

sur le contenu. Notre Data initial comptant 30 374 livres ne comportait pas la colonne des catégories. Afin de remédier à cela, nous avons utilisé l'API de Google Books en conjonction avec la bibliothèque Requests de Python. Cette API nous permet de récupérer les catégories de chaque livre en utilisant son titre. Pour gérer la charge de calcul élevée requise par ce processus, nous avons découpé notre ensemble de données en 10 parties équivalentes. Chaque partie a été traitée individuellement pour l'ajout des catégories. Nous avons également utilisé Google Colab, une plateforme plus puissante, pour résoudre efficacement ce problème de codage. Une fois cette étape terminée, les résultats ont été fusionnés pour construire la nouvelle table contenant les catégories attribuées à chaque livre, Dans la figure 4.5 nous présentons le dataset après avoir ajouté les catégories pour chaque livre.

Image-URL-S	Image-URL-M	Image-URL-L	categories
images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...	Fiction|History|History|Literary Collections|R...
images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...	Fiction|Fiction|Biography & Autobiography|Fict...
images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...	History|History|History|History|History|Biogra...
images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...	Social Science|History|Medical|History|Social ...
images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...	Social Science|Juvenile Nonfiction|Social Scie...

FIGURE 4.5 – Table des livres après l'ajout de la colonne Catégories

— **Ajouter de la colonne pays**
Dans la table utilisateur, notre ensemble de données, en particulier la table des utilisateurs, manque de la colonne pays (country), qui est une colonne très importante pour résoudre le problème du démarrage à froid. Comme nous le montrerons dans la figure 4.6, nous avons la colonne adresse (location) à partir de laquelle nous pouvons extraire l'information concernant le pays.

	User-ID	Location	Age	country
1	2	stockton, california, usa	18.0	usa
3	4	porto, v.n.gaia, portugal	17.0	portugal
5	6	santa monica, california, usa	61.0	usa
9	10	albacete, wisconsin, spain	26.0	spain
10	11	melbourne, victoria, australia	14.0	australia

FIGURE 4.6 – Table du Livres avec la colonne Catégories

4.4.2 Contribution 02
SR avec Filltrage basé sur le contenu

Ce type de recommandation est l'un des plus répandus et utilisés. Son principe est simple : si un utilisateur apprécie les livres historiques, davantage d'ouvrages de ce genre lui seront recommandés. Lorsque l'utilisateur a des préférences multiples, telles que les livres historiques, politiques et de fiction, cela permet de lui offrir une diversité plus large de recommandations adaptées à ses préférences. La figure 4.7 illustre un exemple du filtrage basé sur le contenu.

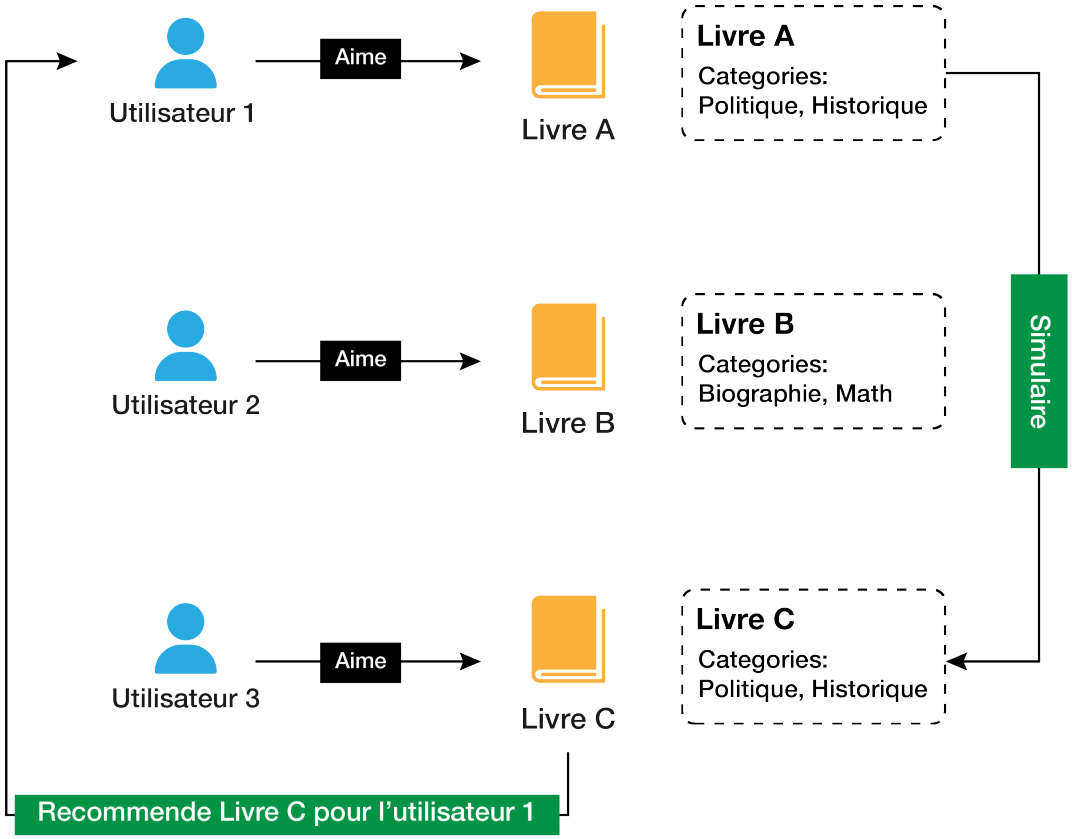

FIGURE 4.7 – Exemple d'SR avec Filtrage basé sur le contenu

Dans la table Livres récoltée nous avons constaté l'absence de la colonne "catégories" que nous avons ajouter lors de l'étape de Pré-traitement des données.

4.4.2.1 Les étape du Filtrage basé sur le contenu

Pour le filtrage basé sur le contenu, il existe plusieurs méthodes couramment utilisées pour analyser et représenter les données textuelles. Parmi ces méthodes, on retrouve le TF-IDF (Term Frequency-Inverse Document Frequency), qui permet de mesurer l'importance d'un terme dans un document par rapport à un corpus de documents. Une autre méthode est le traitement du langage naturel (NLP), qui comprend diverses techniques

pour comprendre et générer le langage humain, facilitant ainsi l'analyse sémantique des textes, nous avons utilisé la méthode Bag of words (Bow) présentée ci dessous.

— **Crée Bag of words (Bow)**

Bow est une méthode de représentation des données textuelles sous forme numérique, où chaque mot dans un document texte est traité comme une caractéristique distincte. Dans le contexte de la recommandation de livres, Bow est généralement utilisé pour représenter les descriptions, les résumés ou les informations sur les catégories des livres.

Dons notre modelé, Lors de la création d'une représentation Bow pour les livres dans la table livres, les données textuelles sont traitées pour chaque catégories de livre (Il existe 50 catégories dans la tables livres) et un vocabulaire de tous les mots uniques est alors crée. Ce vocabulaire sera ensuite utilisé pour représenter chaque livre sous forme de vecteur numérique, la longueur du vecteur étant égale à la taille du vocabulaire.

Chaque élément du vecteur correspond à un mot du vocabulaire, et la valeur de chaque élément indique l'existence de ce mot dans la description, le résumé, ou les informations sur le genre du livre.

Un exemple simple, présenté ci dessous, d'une représentation Bow pour deux livres pourrait ressembler à ceci :

Livre 1 : "Action", "Aventure", "Thriller"
Livre 2 : "Comédie", "Romance", "Drame"

Vocabulaire : "Action", "Aventure", "Thriller", "Comédie", "Romance", "Drame"
Vecteur du Livre 1 : [1, 1, 1, 0, 0, 0]
Vecteur du Livre 2 : [0, 0, 0, 1, 1, 1]

Dans cet exemple, le vocabulaire englobe l'ensemble des mots uniques issus des différentes catégories de livres. Les vecteurs associés à chaque livre représentent la présence ou l'absence de chacun de ces mots dans le vocabulaire correspondant à ce livre.

Voici les éléments du vocabulaire liés à la table Livres : [Fiction, History, Literary Collections, Reference, Religion, Literary Criticism, Biography & Autobiography, Social Science, Medical, Juvenile Nonfiction, Health & Fitness, Law, American literature, Juvenile Fiction, Business & Economics, Performing Arts, Humor, Young Adult Fiction, Self-Help, Science, Nature, Cooking, Family & Relationships, Music, Technology & Engineering, Art, Philosophy, Poetry, Drama, Sports & Recreation, Language Arts & Disciplines, Body, Mind & Spirit, Architecture, Comics & Graphic Novels, Computers, Political Science, Foreign Language Study, Education, Psychology, Pets, Detective and mystery stories, True Crime, Travel, English language, Children's stories, Crafts & Hobbies, Gardening, House & Home, Mathematics, Games & Activities]

Le résultat obtenu est une table avec 30 000 lignes (nombre des livres) et 50 colonnes (vocabulaire ou catégories de la table livres). Si le livre appartient à la catégorie, il prend 1 ; sinon, il prend 0. La figure 4.8 illustre la représentation des catégories avec Bow.

	Fiction	History	Literary Collections	Reference	Religion	Literary Criticism	Biography & Autobiography	Social Science	Medical	Juvenile Nonfiction	...	Detective and mystery stories	True Crime	Travel	Englis languag
Classical Mythology	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.
Clara Callan	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.
Decision in Normandy	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.
Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.
The Mummies of Urumchi	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	0.0	0.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
Illustrated Encyclopedia of Cacti	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.
Basic Scientific Subroutines	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.

FIGURE 4.8 – Représentation Bow des catégories existant dans la table livres

— **Calculer la matrice de similarité cosinus entre les livres**
Pour mesurer la similarité entre les catégories des livres, nous utilisons la similarité cosinus :

$$\text{sim}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \cos(A, B)$$

La similarité cosinus mesure l'angle entre les deux vecteurs. Si les vecteurs pointent dans la même direction, la similarité est maximale (1). Si les vecteurs sont orthogonaux, la similarité est nulle (0). Pour faciliter le calcul de la similarité cosinus, nous utilisons la fonction cosine_similarity de la bibliothèque scikitlearn. Cette fonction calcule automatiquement la similarité cosinus entre chaque paire de vecteurs dans une matrice, simplifiant ainsi le processus de comparaison des descriptions des livres.
Voici un exemple pratique utilisant 3 livres :
Livre 1 : "Action", "Aventure", "Thriller"
Livre 2 : "Comédie", "Romance", "Drame"
Livre 3 : "Action", "Aventure", "Comédie", "Romance"

Les vecteurs correspondants pour ces livres sont :
Vecteur du Livre 1 : A=[1, 1, 1, 0, 0, 0]
Vecteur du Livre 2 : B=[0, 0, 0, 1, 1, 1]
Vecteur du Livre 3 : C=[1, 1, 0, 1, 1, 1]

Calculons la similarité cosinus enter Livre 1 et 2
Produit scalaire : Calculons le produit scalaire des vecteurs A et B.
$A \cdot B = 1 \times 0 + 1 \times 0 + 1 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$

Normes euclidiennes : Calculons les normes euclidiennes des vecteurs A et B.
$\|A\| = \sqrt{1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2} = \sqrt{3} \approx 1.732$
$\|B\| = \sqrt{0^2 + 0^2 + 0^2 + 1^2 + 1^2 + 1^2} = \sqrt{3} \approx 1.732$

Similarité cosinus :
$sim(A, B) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{0}{1.732 \times 1.732} = \frac{0}{3} = 0$

Donc, la similarité cosinus entre le Livre 1 et le Livre 2 est de 0, ce qui signifie qu'il n'y a pas de similarité entre les catégories des deux livres.

Calculons la similarité cosinus enter Livre 1 et 3

Produit scalaire : Calculons le produit scalaire des vecteurs A et C.
$A \cdot C = 1 \times 1 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 1 + 1 + 0 + 0 + 0 + 0 = 2$

Normes euclidiennes : Calculons les normes euclidiennes des vecteurs A et C.
$\|A\| = \sqrt{1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2} = \sqrt{3} \approx 1.732$
$\|C\| = \sqrt{1^2 + 1^2 + 0^2 + 1^2 + 1^2 + 1^2} = \sqrt{5} \approx 2.236$

Similarité cosinus :
$sim(A, C) = \frac{A \cdot C}{\|A\|\|C\|} = \frac{2}{1.732 \times 2.236} = \frac{2}{3.872} \approx 0.516$

Donc, la similarité cosinus entre le Livre 1 et le Livre 3 est d'environ 0.516, ce qui indique une certaine similarité entre les catégories des deux livres.

Calculons la similarité cosinus enter Livre 1 et lui-même

Produit scalaire : Calculons le produit scalaire du vecteur A et lui-même.
$A \cdot A = 1 \times 1 + 1 \times 1 + 1 \times 1 + 0 \times 0 + 0 \times 0 + 0 \times 0 = 1 + 1 + 1 + 0 + 0 + 0 = 3$

Normes euclidiennes : Calculons la norme euclidienne du vecteur A.
$\|A\| = \sqrt{1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2} = \sqrt{3} \approx 1.732$

Similarité cosinus :
$sim(A, A) = \frac{A \cdot A}{\|A\|\|A\|} = \frac{3}{1.732 \times 1.732} = \frac{3}{3} = 1$

Donc, la similarité cosinus entre le Livre 1 et lui-même est de 1, ce qui est attendu puisque tout vecteur est parfaitement similaire à lui-même.

— **Construire une dataframes avec score cosinus**

Après avoir calculé les cosinus entre les livres, nous avons construit un ensemble de données contenant les valeurs de cosinus entre chaque livre et tous les autres livres. Cela nous a permis d'obtenir une matrice carrée de dimensions 30 166 x 30 166, représentant le cosinus entre chaque paire de livres. En remarquant que le cosinus entre un livre et lui-même est égal à 1, nous pouvons confirmer que notre tableau des scores de cosinus est correct.(La figure 4.9 représente La Matrice des cosinus score enter les livres)

	Classical Mythology	Clara Callan	Decision in Normandy	Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	The Mummies of Urumchi	The Kitchen God's Wife	What If?: The World's Foremost Military Historians Imagine What Might Have Been	PLEADING GUILTY	Under the Black Flag: The Romance and the Reality of Life Among the Pirates	Where You'll Find Me: And Other Stories	...	Missouri Madhouse (American Chillers)	Skinned Alive: Stories	/ Whistlin Woma
Classical Mythology	1.000000	0.288675	0.471405	0.235702	0.235702	0.471405	0.577350	0.288675	0.408248	0.333333	...	0.204124	0.204124	0.23570
Clara Callan	0.288675	1.000000	0.816497	0.000000	0.000000	0.408248	0.000000	0.500000	0.000000	0.577350	...	0.353553	0.707107	0.40824
Decision in Normandy	0.471405	0.816497	1.000000	0.333333	0.333333	0.333333	0.408248	0.408248	0.577350	0.471405	...	0.288675	0.577350	0.33333
Flu: The Story of the Great Influenza Pandemic of 1918 and the Search for the Virus That Caused It	0.235702	0.000000	0.333333	1.000000	0.666667	0.000000	0.408248	0.000000	0.577350	0.000000	...	0.000000	0.000000	0.33333
The Mummies of Urumchi	0.235702	0.000000	0.333333	0.666667	1.000000	0.000000	0.408248	0.000000	0.577350	0.235702	...	0.288675	0.000000	0.33333
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Illustrated Encyclopedia of Cacti	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.00000
Basic Scientific Subroutines	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.00000
Amok.	0.500000	0.577350	0.707107	0.235702	0.235702	0.235702	0.288675	0.288675	0.408248	0.333333	...	0.204124	0.408248	0.23570
Petite histoire de la dÃ? Â©sinformation	0.000000	0.288675	0.235702	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.166667	...	0.204124	0.204124	0.00000
Republic (World's Classics)	0.547723	0.000000	0.258199	0.258199	0.258199	0.258199	0.316228	0.000000	0.447214	0.182574	...	0.000000	0.000000	0.00000

FIGURE 4.9 – La Matrice des cosinus score enter les livres

— **Fonction générique de Recommandation**
Apres avoir obtenu la matrice des score cosinus, une fonction qui permet de prendre un titre de livre en entrée ainsi qu'un nombre n de livres à recommander. Cette fonction vérifie d'abord si le titre existe déjà dans notre liste de livres. Si le titre n'existe pas, elle affiche le message "Aucun livre n'est similaire au livre entré". Dans le cas contraire, la fonction sélectionne les n meilleurs livres triés par score de similarité, en commençant par le 2^{me} élément de la liste (pour éviter d'afficher le livre déjà entré) et allant jusqu'au rang $n + 1$.

```
def recommend_tfd_Id_function(livre, n):
    if livre in similarity_df.index:
        # Trouver l'index du livre dans le dataframe de
            ↪ similarite
        livre_index = similarity_df.index.get_loc(livre)

        # Obtenez les n livres les plus similaires au
            ↪ livre entre
        top_n = similarity_df.iloc[livre_index].
            ↪ sort_values(ascending=False)[1:n+1]
        # Renvoie un tableau de titres de livres
        return top_n.index.tolist()
    else:
        print(f'Aucun␣livre␣ne␣ressemble␣a{livre}')
        return []
```

4.4.2.2 Implémentation d'un exemple d'application du SR avec filtrage basé sur le contenu

La figure 4.10 présente une capture d'écran du résultat d'application de notre SR pour

la recherche des livres similaires au livre intitulé "Basic Scientific Subroutines".

```
Top 10 similaire livre to Basic Scientific Subroutines:
The Essential Guide to Computing: The Story of Information Technology    1.000000
Human Factors for Technical Communicators                                0.894427
Number: 1 2 3 4 5 6 7 8 9 10                                             0.894427
Tunisie, 2002                                                            0.894427
IT by Moose ...that's Instructional Techniques                           0.866025
Education in a New Era                                                   0.866025
The Web of Inclusion                                                     0.866025
The Glitter Game                                                         0.866025
500 Of the Coolest Sites for Cyberkids                                   0.866025
Strategies Of Ze                                                         0.816497
```

FIGURE 4.10 – Les 10 livres recommandés par le SR avec filtrage basé sur le contenu pour un utilisateur ayant lu le livre "Basic Scientific Subroutines".

4.4.2.3 Analyse

Cette partie de notre approche SR-HSDF a été consacrée à l'élaboration et la mise en œuvre d'un système de recommandation de livres avec filtrage basé sur le contenu. Ce type de filtrage repose sur l'analyse des catégories des livres pour proposer des recommandations personnalisées aux utilisateurs en fonction de leurs préférences spécifiques.

Cependant, bien que le filtrage basé sur le contenu présente de nombreux avantages, notamment sa capacité à fournir des recommandations précises et personnalisées, il souffre également de certaines limites. L'une de ces principales limites est la sur-spécialisation. En effet, en se basant uniquement sur les préférences passées de l'utilisateur, le système tend à proposer des recommandations trop similaires, limitant ainsi la diversité des suggestions et le potentiel de découverte de nouveaux ouvrages.

Pour pallier ce problème, une solution déjà utilisée dans le domaine des SR est l'utilisation du filtrage collaboratif. Contrairement au filtrage par contenu, le filtrage collaboratif exploite les préférences et les comportements d'un ensemble d'utilisateurs pour identifier des recommandations pertinentes. La section suivante sera consacrée à la présentation de notre troisième contribution.

4.4.3 Contribution 03
SR avec Filltrage collaboratif

Le filtrage collaboratif est une technique largement utilisée dans les systèmes de recommandation modernes. Il englobe plusieurs approches, notamment le filtrage collaboratif basé sur les utilisateurs (user-based), le filtrage collaboratif basé sur les articles (item-based), et le filtrage collaboratif basé sur la factorisation matricielle (Matrix Factorization). Dans cette section, nous nous focalisons sur le filtrage collaboratif basé sur la factorisation matricielle en utilisant l'algorithme de décomposition en valeurs singulières (SVD). La Figure 4.11 illustre le fonctionnement d'un SR avec filtrage collaboratif à travers un exemple

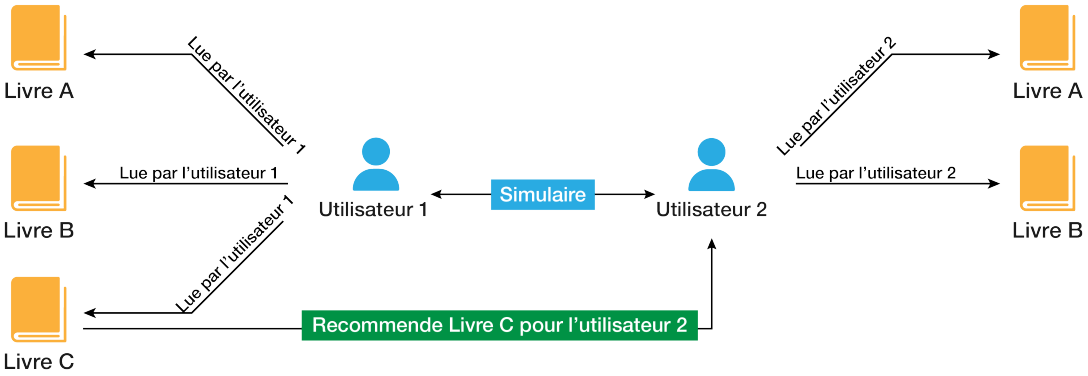

FIGURE 4.11 – Exemple d'SR avec Filtrage collaboratif

4.4.3.1 Algorithme de Décomposition en Valeurs Singulières (SVD)

La décomposition en valeurs singulières (SVD) est une méthode de réduction de dimensionnalité largement utilisée dans les systèmes de recommandation collaboratifs basés sur la factorisation matricielle. Elle permet de réduire la taille de la matrice utilisateur-élément tout en préservant les informations essentielles sur les préférences des utilisateurs et la pertinence des éléments, ce qui améliore la qualité des recommandations. Le SVD décompose la matrice de notation M en trois matrices : U, et Vt (transposée de V). Ces matrices permettent d'identifier les caractéristiques latentes importantes qui décrivent les préférences des utilisateurs et les attributs des éléments. La formule générale de la SVD est :

$$R \approx U \times V^T$$

R : C'est la matrice originale que nous voulons décomposer (matrice utilisateur-élément), dans laquelle il manque certaines des évaluations des livres.
U : est la matrice singulière gauche dont les colonnes sont des vecteurs singuliers gauches représentant les utilisateurs.
est une matrice diagonale contenant des valeurs singulières.
V : est la matrice singulière droite dont les colonnes sont des vecteurs singuliers droits représentant les livres.
Après cette factorisation, SVD permet de prédire les évaluation manquante dans matrice R en utilisant les matrices U, et V. Notre approche permet de prédire avec précision les notes que les utilisateurs attribueront aux produits qu'ils n'ont pas encore évalués, offrant ainsi des recommandations personnalisées basées sur les préférences individuelles des utilisateurs.

En outre, le SVD permettra de gérer les données éparses (Le manque de données). Elle comble de manière fiable ces valeurs manquantes, améliorant la qualité globale des recommandations.

4.4.3.2 L'application du filtrage collaboratif

Dans le cadre de notre étude des SR pour les livres, nous avons adopté l'algorithme SVD pour mettre en œuvre le filtrage collaboratif. Pour ce faire, nous avons suivi le processus suivant :

- **Installation et importation des bibliothèques**
  Nous avons installé la bibliothèque scikit-surprise pour gérer les systèmes de recommandation. Ensuite, nous avons importé les modules nécessaires de scikit-surprise, tels que dataset, Reader, SVD, train_test_split et accuracy, afin de faciliter le développement et l'évaluation de notre modèle de recommandation.
- **Chargement des données**
  Pour le chargement des données, nous avons défini une fonction qui récupère tous les titres de livres uniques dans notre dataframe. Cette fonction filtre ensuite les livres déjà notés par un utilisateur spécifique et pour chaque livre non noté, le modèle SVD prédit la note que l'utilisateur lui attribuerait. Les livres sont alors triés par ordre décroissant sur la note prédite, et les n meilleures recommandations seront sélectionnées et retournées.
- **Préparation des données**
  Pour la préparation des données, nous avons d'abord défini l'échelle de notation allant de 0 à 10. Ensuite, nous avons chargé les données de notation des livres dans un format compatible avec scikit-surprise.
- **Division des données**
  Nous avons divisé les données en un ensemble d'entraînement (80%) et un ensemble de test (20%) afin d'évaluer les performances du modèle.
- **Entraînement du modèle**
  Pour l'entraînement du modèle, nous avons d'abord initialisé le modèle SVD. Ensuite, nous avons entraîné ce modèle sur l'ensemble d'entraînement afin de définir les relations latentes entre les utilisateurs et les livres.
- **Fonction générique de recommandation**
  Nous avons défini une fonction générique pour générer des recommandations pour un utilisateur donné et un nombre spécifié de recommandations (n).

```
def recommend_Svd_function (user_id,n):
    titre_recommender =[]
    recommend_Svd_1 = recommend_Svd(user_id,n)
    for i, (title, _) in enumerate(recommend_Svd_1,
       ↪ start=1):
        titre_recommender.append(title)
    return titre_recommender
```

- **Évaluation du modèle**
  Les métriques d'évaluation que nous utilisons pour notre modèle SVD sont le RMSE (Root Mean Square Error) et le MAE (Mean Absolute Error).
  - **RMSE (Root Mean Square Error)**Le RMSE mesure l'erreur quadratique moyenne des prédictions de notre modèle. Nous le calculons comme la racine carrée de la moyenne des carrés des écarts entre les notes prédites et les notes

réelles.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

— **MAE (Mean Absolute Error)** Le MAE mesure l'erreur absolue moyenne des prédictions de notre modèle. Nous le calculons comme la moyenne des valeurs absolues des écarts entre les notes prédites et les notes réelles.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Le modèle que nous avons développé présente une valeur d'RMSE égal à 3,5233 et de MAE égal à 2,9118 sur une échelle de 10, ces valeurs d'erreur peuvent être considérées comme modérément élevées. Une RMSE et une MAE plus proches de zéro indiqueraient une meilleure performance du modèle. Ainsi, bien que le modèle fournit des prédictions relativement précises, il y a encore une marge de progression pour améliorer la précision. Il pourrait être bénéfique de réévaluer les variables d'entrée, de tester d'autres algorithmes de modélisation, ou d'optimiser davantage les paramètres pour réduire ces erreurs et obtenir un modèle plus performant.

4.4.3.3 Amélioration du modèle

Après l'analyse de l'ensemble de données, nous avons constaté que la fréquence des notes de 0 était extrêmement élevée par rapport aux autres notes (de 1 à 10), comme le montre la figure 4.12 . Après avoir effectué des recherches sur des systèmes similaires, nous avons découvert qu'une note de 0 pour un produit n'existe généralement pas, le minimum étant 1. Par conséquent, nous avons décidé de supprimer tous les avis ayant une note de 0.

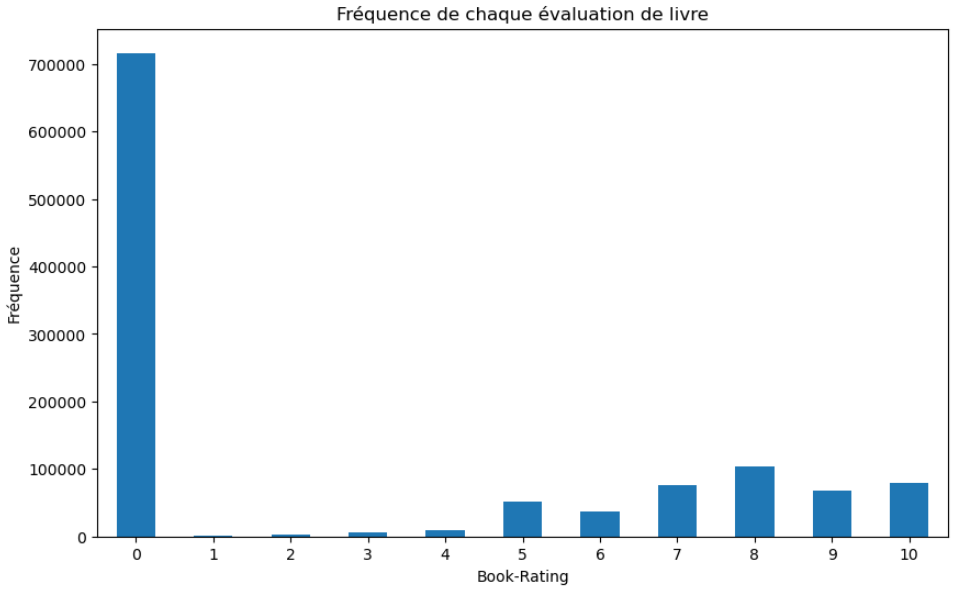

FIGURE 4.12 – Fréquence d'apparition de chaque évaluation de livre

Suite à cette modification, nous avons obtenu des résultats nettement améliorés, comme présenté dans la figure 4.13. En effet, la valeur du RMSE passe de 3,5233 à 1,6980 et celle du MAE passe de 2,9118 à 1,3360. Ces nouvelles valeurs indiquent une amélioration significative de la précision du modèle, avec une RMSE et une MAE plus proches de zéro. Le modèle fournit désormais des prédictions plus précises et fiables. Par conséquent, nous pouvons affirmer que notre modèle est désormais d'une bonne qualité.

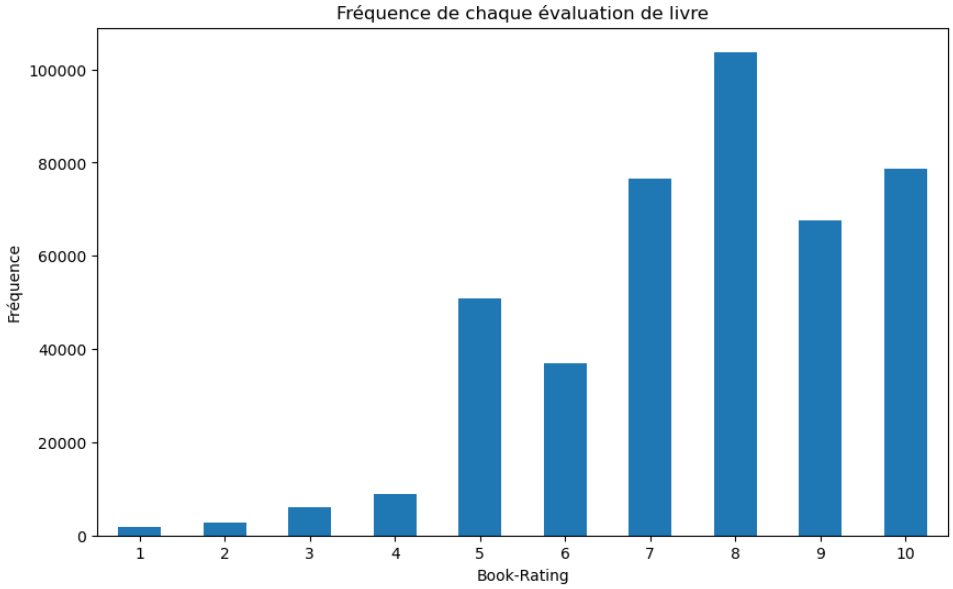

FIGURE 4.13 – Fréquence d'apparition de chaque évaluation de livre sans le Zéro

4.4.3.4 Système Hybride

Pour bénéficier des avantages des deux approches présentées précédemment, il est judicieux de combiner le filtrage basé sur le contenu et le filtrage collaboratif en un système de recommandation hybride. Cette combinaison permet de profiter de la simplicité et de la précision du filtrage basé sur le contenu, tout en exploitant la capacité du filtrage collaboratif à résoudre le problème de sur-spécialisation et à apporter de la variété dans les recommandations de livres. Un tel système hybride offrirait ainsi des suggestions plus riches et équilibrées, en alliant la pertinence thématique à l'exploration de nouveaux horizons littéraires.

```
def hybrid_recommender(user_id, livre, n):
    n = int(n)//2  # Convert n to an integer
    cb_recommendations = recommend_tfd_Id_function(livre, n)
    cf_recommendations = recommend_Svd_function(user_id, n)
    return cf_recommendations + cb_recommendations
```

4.4.3.5 Implémentation d'un exemple d'application du SR avec filtrage collaboratif et du système Hybride

La figure 4.14 présente une capture d'écran du résultat d'application de notre SR collaboratif pour la recherche des livres à recommander à l'utilisateur ayant l'ID "276747".

```
recommend_Svd_function(276747, 10)
```

```
['To Kill a Mockingbird',
 'Harry Potter and the Goblet of Fire (Book 4)',
 'The Talisman',
 'Johnny Got His Gun',
 'Piercing the Darkness',
 'Key of Knowledge (Key Trilogy (Paperback))',
 'Mrs. Frisby and the Rats of Nimh',
 'The Waste Lands (The Dark Tower, Book 3)',
 'Die unendliche Geschichte: Von A bis Z',
 'The Lion, the Witch, and the Wardrobe (The Chronicles of Narnia, Book 2)']
```

FIGURE 4.14 – Les 10 livres recommandés pour l'utilisateur 276747 avec le filtrage collaboratif

Comme l'illustre la Figure 4.15, en utilisant la fonction hybride pour le même utilisateur (ID = "276747") une liste de 20 livres sera retournée. Les 10 première recommandation sont retournées par le SR avec filtrage collaboratif tandis que les 10 deniers livres de la Figure 4.15 sont le résultat d'application du SR avec filtrage basé sur le contenu (présenté dans la Figure 4.10) et cela sachant que cet utilisateur a lu le livre "Basic Scientific Subroutines".

```
hybrid_recommender(276747, 'Basic Scientific Subroutines', 20)
```

```
['To Kill a Mockingbird',
 'Harry Potter and the Goblet of Fire (Book 4)',
 'The Talisman',
 'Johnny Got His Gun',
 'Piercing the Darkness',
 'Key of Knowledge (Key Trilogy (Paperback))',
 'Mrs. Frisby and the Rats of Nimh',
 'The Waste Lands (The Dark Tower, Book 3)',
 'Die unendliche Geschichte: Von A bis Z',
 'The Lion, the Witch, and the Wardrobe (The Chronicles of Narnia, Book 2)',
 'The Essential Guide to Computing: The Story of Information Technology',
 'Human Factors for Technical Communicators',
 'Number: 1 2 3 4 5 6 7 8 9 10',
 'Tunisie, 2002',
 "IT by Moose ...that's Instructional Techniques",
 'Education in a New Era',
 'The Web of Inclusion',
 'The Glitter Game',
 '500 Of the Coolest Sites for Cyberkids',
 'Strategies Of Ze']
```

FIGURE 4.15 – Les 20 livres recommandés en utilisant le système hybride

4.4.3.6 Analyse

En conclusion, les systèmes de recommandation basés sur le filtrage collaboratif et le filtrage basé sur le contenu présentent chacun des avantages distincts pour la suggestion de livres. Le filtrage collaboratif, en particulier, offre une solution efficace au problème de surspécialisation en introduisant de la diversité dans les recommandations. Cette approche permet de suggérer des ouvrages que l'utilisateur n'aurait peut-être pas découverts par lui-même, élargissant ainsi son horizon de lecture.
Néanmoins, il convient de noter que le filtrage collaboratif introduit un nouveau défi à savoir le problème du démarrage à froid. Ce problème se manifeste lorsque le système

manque de données suffisantes sur les nouveaux utilisateurs , limitant ainsi sa capacité à fournir des recommandations pertinentes dans ces cas spécifiques.

4.4.4 Contribution 04 : Solution proposée pour la résolution du problème de démarrage à froid

Le problème du démarrage à froid (cold start) dans les systèmes de recommandation peut être classé en deux catégories, le démarrage à froid pour les nouveaux articles et le démarrage à froid pour les nouveaux utilisateurs. Pour les nouveaux articles, le problème survient lorsque le système ne dispose pas de suffisamment d'évaluations antérieures liées à cet article. En ce qui concerne les nouveaux utilisateurs, il est difficile de leur recommander des articles car le système ne possède aucune information sur leurs préférences passés ou il est possible qu'ils n'aient encore évalué aucun article, ce qui signifie que leurs préférence sont inconnus du système. En effet, le problème du démarrage à froid se produit lorsque le système de recommandation manque de données sur les nouveaux éléments ou utilisateurs, rendant ainsi difficile la génération de recommandations précises.
Étant donné que nous avons utilisé le filtrage basé sur le contenu, nous avons déjà résolu le problème du démarrage à froid pour les livres. En effet, chaque livre appartient à une ou plusieurs catégories, ce qui nous fournit au moins une information utilisée pour formuler des recommandation avec laquelle nous pouvons travailler. Cependant, il nous reste à résoudre le problème du démarrage à froid pour les utilisateurs..

4.4.4.1 Solution proposée

Généralement, les systèmes de recommandation suggèrent aux nouveaux utilisateurs les meilleurs livres de toute la base de données, ce qui est très général. Notre idée est de minimiser le nombre de livres à recommander à un nouvel utilisateur. Nous remarquons que dans la table des utilisateurs, nous avons les deux attributs âge et pays, ce qui nous aide à créer des clusters en utilisant ces deux caractéristiques. Bien que nous puissions également utiliser l'algorithme KNN, nous avons choisi l'algorithme K-means pour cette tâche.

4.4.4.2 Les étape de notre solution pour la résolution du problème du démarrage a froid

- **Encoder la colonne pays (LabelEncoder())**
  Cette première étape consiste à encoder les noms des pays en valeurs numériques. Comme les algorithmes de clustering ne fonctionnent pas directement avec des données catégorielles, nous utilisons le LabelEncoder de la bibliothèque scikit-learn pour transformer les noms de pays en entiers. Chaque pays se voit attribuer un numéro unique, ce qui facilite le traitement des données par l'algorithme de clustering.
- **Sélectionner les caractéristique pour le clustering à savoir (age et pays)**
  Pour effectuer le clustering, nous devons sélectionner les caractéristiques pertinentes du dataset à utiliser. Dans ce cas, nous avons choisis l'âge des utilisateurs et le code numérique représentant leur pays. Ces deux caractéristiques sont jugées

pertinentes pour regrouper les utilisateurs en fonction des similarités démographiques et géographiques.

— **Choisir le nombre de clusters d'après la méthode Elbow**
La méthode Elbow est utilisée pour déterminer le nombre optimal de clusters à utiliser pour le K-means. Elle consiste à exécuter l'algorithme K-means pour un ensemble de valeurs possibles de k (nombre de clusters) et à tracer la somme des carrés des distances au centre des clusters pour chaque k. Le point où la courbe commence à se stabiliser (forme un coude) indique le nombre optimal de clusters. Dans ce cas, d'après le graphe présente dans la figure 4.16 il est clair que la meilleur valeur de k est 4.

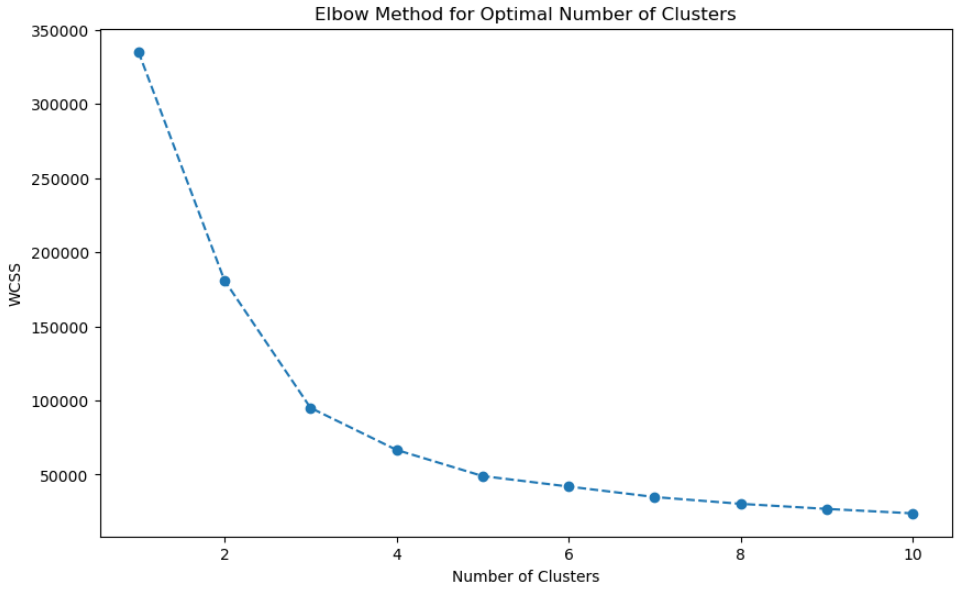

FIGURE 4.16 – Méthode du Elbow pour un nombre optimal de clusters

— **Appliquer le clustering K-means avec le nombre de clusters choisi**
Une fois le nombre optimal de clusters déterminé, nous appliquons l'algorithme K-means qui partitionne les utilisateurs en groupes où chaque utilisateur appartient au cluster avec le centre le plus proche. Cela permet de regrouper les utilisateurs selon des caractéristiques similaires définies précédemment (âge et pays). La figure 4.17 représente les diffèrent clusters obtenus.

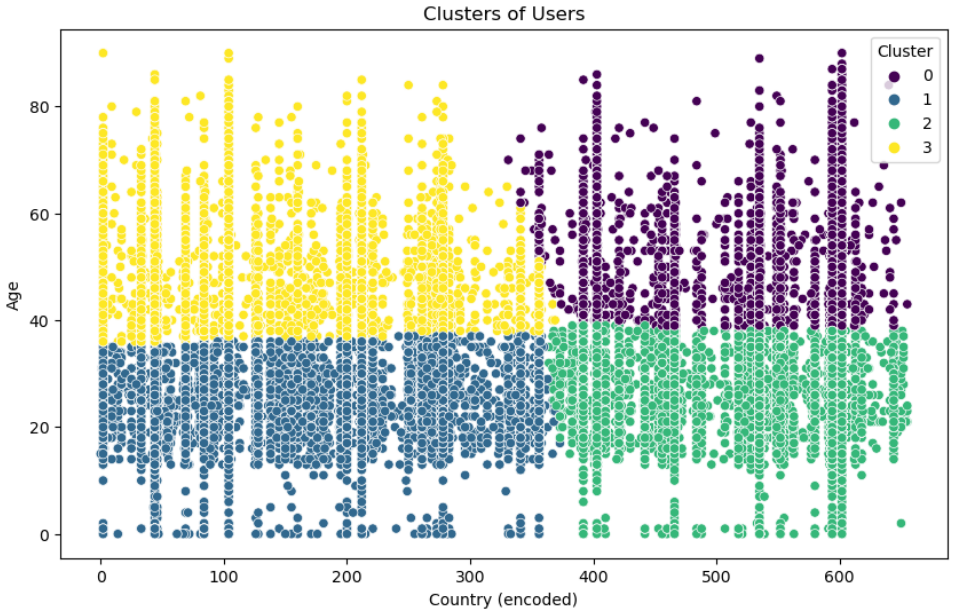

FIGURE 4.17 – Résultat d'application de k-means

— **Récupérer tous les ID des utilisateurs du cluster choisi**
Après avoir exécuté le clustering, nous identifions le cluster d'intérêt (par exemple, celui des utilisateurs âgés de 20 ans habitant en Tunisie) afin de récupérer les ID des utilisateurs appartenant à ce cluster.

— **Sélectionner les meilleurs livres**
Une fois que nous avons identifié les utilisateurs appartenant au cluster d'intérêt, nous récupérons tous les livres lus par ces utilisateurs en utilisant leurs IDs. Ensuite, nous calculons le nombre de lectures pour chaque livre. En triant ces livres par le nombre de lectures (du plus grand au plus petit), nous pouvons déterminer les livres les plus lus par les membres du cluster.

Pour améliorer la sélection des livres recommandés, nous ne retenons que ceux qui ont une note moyenne (moyenne des évaluations) supérieure à 7. Cela garantit que les livres recommandés sont non seulement populaires au sein du cluster mais qu'ils ont été bien évalués par ces membres, reflétant ainsi une appréciation positive générale. Enfin, ces livres seront présentés aux utilisateurs du cluster comme recommandations personnalisées. Cette approche assure que les livres recommandés répondent ainsi aux attentes des utilisateurs en matière de qualité et de pertinence.

4.4.4.3 Implémentation du modèle proposé avec un exemple

Supposons qu'un utilisateur U1 ait 20 ans et habite en Tunisie. Après avoir appliqué l'algorithme de clustering K-means avec les caractéristiques d'âge et de pays, nous identifions que U1 appartient au cluster 2 et nous récupérons les ID de ce cluster. La figure 4.18 représente le cluster 2 et tous les ID des utilisateurs de ce cluster.

```
The user belongs to cluster: 2
User IDs in the same cluster:[     2      4     10 ... 278843 278844 278853]
```

FIGURE 4.18 – le cluster de l'utilisateur U1 et tous les ID des utilisateurs de ce cluster

Aussi, nous calculons le nombre de lectures de chaque livre parmi ceux lus par les utilisateurs du cluster 2. En triant ces livres selon le nombre de lectures (du plus grand au plus petit), nous pouvons déterminer les livres les plus lus par les membres du cluster. La figure 4.19 représente les livres classés par nombre de lectures.

	Book-Title	books_count
0	Wild Animus	557
1	The Catcher in the Rye	127
2	Jurassic Park	110
3	To Kill a Mockingbird	110
4	The Joy Luck Club	105
...	...	...
8789	Love by Design (Arabesque)	1
8790	A Concise Anglo-Saxon Dictionary (Medieval Aca...	1
8791	The Cambridge Companion to Old English Literat...	1
8792	The Riverside Chaucer (Oxford Paperback)	1
8793	Gould's Book of Fish: A Novel in Twelve Fish	1

8794 rows × 2 columns

FIGURE 4.19 – Dataframe des livres classés par nombre de lectures

La figure 4.20 représente les livres retenus (les livres ayant une note moyenne supérieure à 7).

	Book-Title	books_count	Book_average_ratings
0	Wild Animus	557	0.872531
1	The Catcher in the Rye	127	5.472441
2	Jurassic Park	110	3.690909
3	To Kill a Mockingbird	110	5.736364
4	The Joy Luck Club	105	2.923810
...	...	...	...
4767	God's Little Instruction Book II: More Inspira...	1	10.000000
4766	Micro Fiction: An Anthology of Really Short St...	1	10.000000
4765	The Art of Tarot	1	10.000000
4764	Julie and Me and Michael Owen Make Three	1	0.000000
8793	Gould's Book of Fish: A Novel in Twelve Fish	1	0.000000

FIGURE 4.20 – Dataframe des livres classés par nombre de lectures avec note moyenne du livre

Enfin, nous présentons les livres retenus à utilisateur U1 du cluster comme recommandations personnalisées.

```
In [137]: print(f"Les meilleur livre a recommender pour le utilisateur de l'age de {age} ans et qui habite a {location}")
          for i, title in enumerate(Top_Books_10['Book-Title'], 1):
              print(f"{i}. {title}")

          Les meilleur livre a recommender pour le utilisateur de l'age de 20 ans et qui habite a tunisia
          1. Harry Potter and the Goblet of Fire (Book 4)
          2. El Hobbit
          3. Taliesin : Book One of the Pendragon Cycle (Pendragon Cycle)
          4. Johnny Got His Gun
          5. Harry Potter y el prisionero de Azkaban
          6. El Senor De Los Anillos: El Retorno Del Rey (Tolkien, J. R. R. Lord of the Rings. 3.)
          7. Rule of the Bone : Novel, A
          8. Behind the Attic Wall (Avon Camelot Books (Paperback))
          9. Whitney, My Love
          10. The Crystal Cave
```

FIGURE 4.21 – Les livres recommander pour l'utilisateur U1

4.5 Conclusion

L'approche SR-HSDF (Système de Recommandation Hybride proposant une solution au problème de Sur-spécialisation et de Démarrage à Froid) combine plusieurs techniques pour offrir des recommandations de livres personnalisées et variées tout en surmontant des défis courants des systèmes de recommandation. Les principales contributions de cette approche sont :
Un pré-traitement des données, incluant l'ajout de catégories de livres et l'extraction des pays des utilisateurs.
Un système de filtrage basé sur le contenu utilisant la méthode Bag of Words et la similarité cosinus pour recommander des livres similaires.
Un système de filtrage collaboratif basé sur la factorisation matricielle en utilisante

l'algorithme SVD pour diversifier les recommandations et résoudre le problème de sur-spécialisation.
Une solution au problème du démarrage à froid pour les nouveaux utilisateurs, utilisant le clustering K-means basé sur l'âge et le pays des utilisateurs.

SR-HSDF permet de générer des recommandations pertinentes et variées, même pour de nouveaux utilisateurs, tout en atténuant les problèmes courants tels que la sur-spécialisation. L'utilisation de techniques d'apprentissage automatique et de traitement de données avancées contribue à l'amélioration de l'efficacité et la précision de SR-HSDF.

Conclusion générale et perspectives

Les systèmes de recommandation jouent un rôle crucial dans la personnalisation de l'expérience utilisateur en fournissant des suggestions adaptées aux préférences individuelles. Dans notre mémoire, nous avons cherché à résoudre deux problèmes majeurs associés à ces systèmes : le problème de la sur-spécialisation et le problème de démarrage à froid. Pour répondre à ces défis, nous avons proposé une approche hybride appelée « Système de Recommandation Hybride proposant une solution au problème de Sur-spécilisation et de Démarrage à Froid (SR-HSDF) ». Cette approche combine le filtrage collaboratif et le filtrage basé sur le contenu pour tirer parti des avantages des deux méthodes tout en minimisant leurs inconvénients. Pour rappel, le problème de la sur-spécialisation survient lorsque le système de recommandation propose des éléments très similaires aux préférences antérieures de l'utilisateur, limitant ainsi la diversité des recommandations. En intégrant le filtrage collaboratif, nous avons pu atténuer ce problème en exploitant les préférences d'utilisateurs similaires pour offrir des recommandations plus variées et pertinentes. En ce qui concerne le démarrage à froid, notre solution utilise l'algorithme K-means pour prendre ne charge les nouveaux utilisateurs en prenant en compte des attributs comme l'âge et le pays. Cette méthode permet de fournir des recommandations précises même en l'absence de données historiques sur les préférences des nouveaux utilisateurs. De plus, le filtrage basé sur le contenu permet de surmonter le démarrage à froid pour les nouveaux livres en utilisant les catégories auxquelles ils appartiennent. Il est important de noter qu'il n'existe pas de système de recommandation capable de résoudre tous les problèmes associés aux SR. Chaque système tente de répondre au mieux à un maximum de défis, mais aucune solution n'est exhaustive.

Finalement, notre approche garantit non seulement des recommandations qui respectent les préférences des utilisateurs, mais aussi des suggestions diversifiées et adaptées aux nouveaux utilisateurs. Cette approche hybride améliore la qualité des recommandations tout en résolvant les problèmes de sur-spécialisation et de démarrage à froid, assurant ainsi une satisfaction accrue des utilisateurs.

Dans le cadre de l'évolution et l'amélioration de notre travail, plusieurs perspectives peuvent être envisagées. Une première direction prometteuse serait l'intégration de méthodes avancées de traitement du langage naturel (NLP) pour améliorer la compréhension contextuelle et les préférences des utilisateurs à partir de données textuelles non structurées telles que les avis et les commentaires. Par exemple, l'analyse sémantique pourrait permettre une adaptation plus fine des recommandations en prenant en compte le sentiment exprimé dans ces textes. Un autre domaine prometteur est l'exploration de l'utilisation de graphes de connaissances pour améliorer les systèmes de recommandation. Ces graphes permettent de représenter et de comprendre les relations complexes entre les éléments recommandés et les utilisateurs. En intégrant cette dimension sémantique plus riche, les systèmes de recommandation pourraient générer des recommandations plus pertinentes et mieux adaptées aux besoins et préférences individuels des utilisateurs.

Enfin, l'éthique et la transparence doivent être prioritaires pour assurer des systèmes de recommandation équitables, non biaisés et respectueux de la vie privée, favorisant ainsi une acceptation sociale et des recommandations justes.

Bibliographie

[1] H. El Bouhissi, D. Tagzirt, F. Bouredjioua, and O. Pavlova, "Health recommender system for smart cities.," in *MoMLeT+ DS*, pp. 334–343, 2023.

[2] P. Arunruviwat and V. Muangsin, "A hybrid book recommendation system for university library," in *2022 26th International Computer Science and Engineering Conference (ICSEC)*, pp. 291–295, IEEE, 2022.

[3] W. S. Alaloul and A. H. Qureshi, "Data processing using artificial neural networks," *Dynamic data assimilation-beating the uncertainties*, 2020.

[4] Sudarshan S, "Recurrent neural network (rnn)." https ://sudarshan-s.medium.com/recurrent-neural-network-rnn-20a619190586, Consulté le 15/03/2024.

[5] "Les rÉseaux de neurones convolutifs." https ://www.natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-rseaux-de-neurones-convolutifs, Consulté le 15/03/2024.

[6] Sudarshan S, "Recurrent neural network (rnn)." https ://sudarshan-s.medium.com/recurrent-neural-network-rnn-20a619190586, Consulté le 15/03/2024.

[7] https ://datascientest.com/wp-content/uploads/2020/06/Fichier-95.png, Consulté le 15/03/2024.

[8] A. R. Krishnan and R. Remya, "A case study on various recommendation systems," *International Journal of Computer Applications*, vol. 133, no. 15, pp. 5–8, 2016.

[9] D. Sukhanov, A. Galkin, and E. Khabibullina, "Collaborative filtering algorithm for recommender systems," in *2023 5th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*, pp. 557–560, IEEE, 2023.

[10] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web : methods and strategies of web personalization*, pp. 325–341, Springer, 2007.

[11] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation systems : Algorithms, challenges, metrics, and business opportunities," *applied sciences*, vol. 10, no. 21, p. 7748, 2020.

[12] G. Lekakos and P. Caravelas, "A hybrid approach for movie recommendation," *Multimedia tools and applications*, vol. 36, pp. 55–70, 2008.

[13] A. Fanca, A. Puscasiu, D.-I. Gota, and H. Valean, "Recommendation systems with machine learning," in *2020 21th International Carpathian Control Conference (ICCC)*, pp. 1–6, IEEE, 2020.

[14] D. Kumar. D, "Secure user recommendation using machine learning techniques," in *2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 233–240, IEEE, 2023.

[15] K. Kinkar, "Product recommendation system : A systematic literature review," *International Journal for Research in Applied Science and Engineering Technology*, 2021.

[16] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data mining and knowledge discovery*, vol. 5, pp. 115–153, 2001.

[17] M. C. Urdaneta-Ponte, A. Mendez-Zorrilla, and I. Oleagordia-Ruiz, "Recommendation systems for education : Systematic review," *Electronics*, vol. 10, no. 14, p. 1611, 2021.

[18] G. Dror, N. Koenigstein, and Y. Koren, "Web-scale media recommendation systems," *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2722–2736, 2012.

[19] F. Alyari and N. Jafari Navimipour, "Recommender systems : A systematic review of the state of the art literature and suggestions for future research," *Kybernetes*, vol. 47, no. 5, pp. 985–1017, 2018.

[20] M. G. Campana and F. Delmastro, "Recommender systems for online and mobile social networks : A survey," *Online Social Networks and Media*, vol. 3, pp. 75–97, 2017.

[21] S. Jain, A. Grover, P. S. Thakur, and S. K. Choudhary, "Trends, problems and solutions of recommender system," in *International conference on computing, communication & automation*, pp. 955–958, IEEE, 2015.

[22] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation : a review of ontology-based recommender systems for e-learning," *Artificial intelligence review*, vol. 50, pp. 21–48, 2018.

[23] M. Madhukar, "Challenges & limitation in recommender systems," *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, vol. 4, no. 3, pp. 138–142, 2014.

[24] J. J. Kizakkethottam, A. Margret, and R. Suma, "Web log mining using wap tree mine and web log mining tool," in *International Conference on Soft Computing and Network Security*, 2015.

[25] L. R. Sebastian, S. Babu, and J. J. Kizhakkethottam, "Challenges with big data mining : A review," in *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, pp. 1–4, IEEE, 2015.

[26] Mehak, R. Kumar, and D. A. Mehta, "Artificial intelligence," *International Journal of Advanced Research in Science, Communication and Technology*, 2023.

[27] W. Schneider and H. Guo, "Machine learning.," *The journal of physical chemistry. B*, vol. 122 4, p. 1347, 2018.

[28] S. K. Selvaraj, A. Raj, R. Rishikesh Mahadevan, U. Chadha, and V. Paramasivam, "A review on machine learning models in injection molding machines," *Advances in Materials Science and Engineering*, vol. 2022, no. 1, p. 1949061, 2022.

[29] M. Kubát, "An introduction to machine learning," pp. 1–348, 2017.

[30] S. Patil and S. Patil, "Linear with polynomial regression : Overview," *International Journal of Applied Research*, 2021.

[31] R. Gil-García and A. Pons-Porrata, "Dynamic hierarchical algorithms for document clustering," *Pattern Recognit. Lett.*, vol. 31, pp. 469–477, 2010.

[32] Y. Reddy, P. Viswanath, and B. E. Reddy, "Semi-supervised learning : A brief review," *Int. J. Eng. Technol*, vol. 7, no. 1.8, p. 81, 2018.

[33] A. G. Barto, "Reinforcement learning : Connections, surprises, and challenge," *AI Magazine*, vol. 40, no. 1, pp. 3–15, 2019.

[34] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville : Deep learning : The mit press, 2016, 800 pp, isbn : 0262035618," *Genetic programming and evolvable machines*, vol. 19, no. 1, pp. 305–307, 2018.

[35] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[36] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks : an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611–629, 2018.

[37] E. Bisong and E. Bisong, "Recurrent neural networks (rnns)," *Building Machine Learning and Deep Learning Models on Google Cloud Platform : A Comprehensive Guide for Beginners*, pp. 443–473, 2019.

[38] N. Kriegeskorte and T. Golan, "Neural network models and deep learning," *Current Biology*, vol. 29, no. 7, pp. R231–R236, 2019.

[39] T. Fujimoto and H. Murakami, "A book recommendation system considering contents and emotions of user interests," in *2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 154–157, IEEE, 2022.

[40] Y. Zhao and J. Zeng, "Library intelligent book recommendation system using facial expression recognition," in *2020 9th International Congress on Advanced Applied Informatics (IIAI-AAI)*, pp. 55–58, IEEE, 2020.

[41] V. K. Kavitha and S. Koteeswaran, "Study on book recommendation system," *2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, pp. 1–8, 2023.

[42] N. Rani and S. L. Chu, "Does the type of recommender system impact users' trust ? exploring context-aware recommender systems in education," in *2023 IEEE International Conference on Advanced Learning Technologies (ICALT)*, pp. 41–43, IEEE, 2023.

[43] N. N. Qomariyah and A. N. Fajar, "Recommender system for e-learning based on personal learning style," in *2019 international seminar on research of information technology and intelligent systems (ISRITI)*, pp. 563–567, IEEE, 2019.

[44] D. Sarma, T. Mittra, and M. S. Hossain, "Personalized book recommendation system using machine learning algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, 2021.

[45] O. N. Jadhav and A. Kb, "Movie recommendation system using machine learning algorithms," *Journal of Machine and Computing*, 2022.

[46] N. Giridharan, K. S. Nathan, and M. Swetha, "Movie recommendation system using machine learning," *International journal of health sciences*, 2022.

[47] P. M. Alamdari, N. J. Navimipour, M. Hosseinzadeh, A. Safaei, and A. Darwesh, "A systematic study on the recommender systems in the e-commerce," *IEEE Access*, vol. 8, pp. 115694–115716, 2020.

[48] Z. Cao, X. Qiao, S. Jiang, and X. Zhang, "An efficient knowledge-graph-based web service recommendation algorithm," *Symmetry*, vol. 11, no. 3, p. 392, 2019.

[49] "Book recommendation dataset." `https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset`, 2021.

RÉSUMÉ

Notre projet de fin de cycle se concentre sur le développement d'un système de recommandation de livres personnalisé utilisant des techniques d'apprentissage automatique. Notre approche SR-HSDF utilise le filtrage basé sur le contenu pour cibler les préférences des utilisateurs. Pour surmonter le problème de la sur-spécialisation, nous avons intégré le filtrage collaboratif au filtrage basé sur le contenu. Par a suite, nous utilisons l'algorithme de clustering K-means pour traiter le problème du démarrage à froid. Cette étape basé sur l'apprentissage automatique permet de fournir des recommandations diversifiées et pertinentes, tout en offrant des suggestions adaptées aux nouveaux utilisateurs. Le résultat est un système de recommandation robuste basé sur l'apprentissage automatique qui améliore l'expérience utilisateur en proposant des livres correspondant à leur préférence tout en introduisant de la variété dans les suggestions.

Mots clés : Apprentissage automatique, filtrage basé sur le contenu, le filtrage collaboratif, sur-spécialisation, démarrage à froid, K-means.

ABSTRACT

Our final-year project focuses on developing a personalized book recommendation system using machine learning techniques. Our approach, SR-HSDF, utilizes content-based filtering to target user preferences. To overcome the over-specialization issue, we integrated collaborative filtering with content-based filtering. Subsequently, we employ the K-means clustering algorithm to address the cold start problem. This machine learning-based step enables us to provide diverse and relevant recommendations while offering tailored suggestions for new users. The result is a robust machine learning-based recommendation system that enhances user experience by suggesting books aligned with their preferences while introducing variety in recommendations.

Keywords : Machine learning, content-based filtering, collaborative filtering, over-specialization, cold start, K-means.