REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur

Et de la recherche scientifique

Université "Yahia FARES" de Médéa

Faculté des sciences

Département des mathématiques et informatique

Mémoire de Master

Spécialité: Informatique

Option :Ingénierie des systèmes et technologies du web

Thème

Modèle de filtrage collaboratif hybridebasé sur l'apprentissage en profondeur

Réalisé par :

▪ BOUDJELLALAbdelkader▪SADOUKI Fatiha

▪DJEKKOUNE Chams Eddine

Proposé et Dirigé par :

2019 - 2020

Dédicace

Je dédie ce modeste travail à ...

Mes très chers parents, Qu'Allah leur procure bonne santé et longue vie.

Mes frères et sœurs, toute ma famille, mes amis, mon binôme Abd El Kader et à tous ceux qui ont contribué de près ou de loin afin que ce projet soit finalisé.



Chams Eddine

Dédicace

Ce mémoire est dédié à :

Mes chers Parents qui m'ont toujours entouré, motivé, aimé,

soutenue et ouvré pour ma réussite, Tous mes frères et ma sœur,

Tous mes Professeurs.



Abdelkader

Remerciement

*Louange à **Allah**, Seigneur de l'univers, Le Tout Miséricordieux, le Très Miséricordieux, pour nous avoir accordé le savoir, le courage, la patience, la volonté et la force nécessaire, pour affronter toutes les difficultés et les obstacles, qui se sont hissés au travers de notre chemin, durant toutes nos années d'études.*

*Nous tenons à remercier notre promotrice Mme **SADOUKI Fatiha** pour son encadrement, son écoute, ses élucidations, ses conseils, ses directives et encouragements qu'elle nous a afflué. Et nos vifs remerciements vont également aux membres du jury qui ont accepté de juger notre travail.*

Et enfin, merci à tous ceux qui ont contribué de près ou de loin pour l'accomplissement de ce travail, et en particulier à l'ensemble de nos enseignants, qui sont à l'origine de tout notre savoir.

RESUME

En raison de la modernisation récente des données, les systèmes de recommandation (SR) jouent un rôle primordiale dans de nombreux produits et services, c'est pour cette raison les grandes entreprises comme Amazon, Google, YouTube, Facebook, Netflix, Alibaba ... ont développé leurs propres systèmes de recommandation plus avancés, cela explique notre motivation de mettre en œuvre un système de recommandation pour les films en appliquant des algorithmes d'apprentissage automatique (ML) avec l'objectif de recommander les éléments les plus appropriés en fonction des préférences des utilisateurs.

Nous présentons dans ce travail, une proposition pour améliorer la liste des recommandations et rendre les recommandations plus adaptées aux changements des préférences des utilisateurs. Nous proposons une hybridation entre le filtrage collaboratif basé sur les réseaux de neurones profonds avec la méthode de segmentation basée sur le niveau de similarité entre les utilisateurs. La performance de notre approche est testée à travers plusieurs expérimentations menées sur la base de films *MoviLens*, afin de prouver l'efficacité des propositions et fournir de nouvelles perspectives.

Nous avons également créé une interface simple pour illustrer la mise en œuvre du modèle proposé avec une nouvelle recommandation de film.

Mots clé : Apprentissage automatique, Système de recommandation, Réseaux de neurones, Apprentissage en profondeur.

ABSTRACT

Due to the recent modernization of data, recommendation systems (SR) play a key role in many products and services, that's why large companies like Amazon, Google, YouTube, Facebook, Netflix, Alibaba ... has been developed their own advanced recommendation systems, this explains our motivation to develop a film recommendation system by applying machine learning algorithms (ML) with the goal of recommending the most appropriate elements based on user preferences.

In this work, we present a proposal to improve the recommendation list and make recommendations more responsive to changes in user preferences. We propose an hybridization between collaborative filtering based on deep neural networks with the segmentation method based on the level of similarity between users. The performance of our approach is tested through several experiments conducted on the datasetof MoviLens films, in order to prove the effectiveness of the proposals and provide new perspectives.

We have also created a simple interface to illustrate the implementation of the proposed model with a new film recommendation.

Key words: Machine Learning, Recommendation System, Neural Networks, Deep Learning.

ملخص

بسبب التحديث الأخير للبيانات، تلعب أنظمة التوصية (SR) دورًا رئيسيًا في العديد من المنتجات والخدمات، وهذا هو السبب في أن الشركات الكبيرة مثل Amazon و Google وYouTube و Facebook وNetflix وAlibaba... تقوم بتطوير أنظمة التوصية أكثر تقدما، وهو ما يفسر الدافع لدينا لتنفيذ نظام توصية للأفلام من خلال تطبيق خوارزميات التعلم الآلي (ML) بهدف التوصية بالعناصر الأكثر ملاءمة على أساس تفضيلات المستخدم.

وفي هذا العمل، نقدم اقتراحاً لتحسين قائمة التوصيات وجعل التوصيات أكثر استجابة للتغييرات في تفضيلات المستخدمين. نقترح تهجين بين الترشيح التعاوني على أساس الشبكات العصبية العميقة مع طريقة التقسيم على أساس مستوى التشابه بين المستخدمين. يتم اختبار أداء نهجنا من خلال العديد من التجارب التي أجريت على قاعدة أفلام MoviLens، من أجل إثبات فعالية المقترحات وتقديم وجهات نظر جديدة.

كما أنشأنا واجهة بسيطة لتوضيح تنفيذ النموذج المقترح مع توصية فيلم جديد.

الكلمات الأساسية: التعلم الالي، نظام التوصية، الشبكات العصبية، التعلم المتعمق.

Table des matières

Remerciement **4**
RESUME **5**
ABSTRACT **6**
ملخص **7**
Liste des Abréviation **11**
Introduction générale **13**
1.1 Introduction : 17
1.2 Historique : 17
1.3 Définition des systèmes de recommandation : 18
1.4 Objectif des systèmes de recommandation : 18
1.5 Filtrage d'information Vs Recherche d'information : 19
1.6 Processus de recommandation : 20
1.7 Les étapes principales de la recommandation : 21
1.7.1 Collecte d'Information : 21
1.7.2 Mise en œuvre d'une matrice utilisateur d'informations récoltées : 22
1.7.3 Extraction de la liste de recommandations : 23
1.8 Les principales approches de filtrage : 23
1.8.1 Filtrage basé sur le contenu (content based filtering) : 23
1.8.2 Filtrage collaboratif (collaborative filtering) : 24
1.8.3 Filtrage hybride (hybrid filtering) : 26
1.9 Avantages et inconvénients des systèmes de recommandations : 29
1.10 Evaluation des Systèmes de recommandation : 30
1.10.1 Évaluation online : 30
1.11 Conclusion : 32
L'apprentissage automatique, l'apprentissage profond et l'apprentissage profond pour la recommandation **33**
2.1 Introduction : 34
2.2 Apprentissage automatique : 34
2.2.1 Apprentissage supervisé : 35
2.2.2 Apprentissage non supervisé : 39
2.3 L'apprentissage en profondeur (Deep Learning) : 42

2.3.1 Définition : 43
2.3.2 Historique : 44
2.3.3 Pour quoi le choix du Deep Learning : 44
2.3.4 Les différentes Architectures du Deep Learning : 45
2.4 L'apprentissage profond pour la recommandation : 47
2.4.1 Recommandation basée sur le perceptron multicouche : 48
2.4.2 Recommandation basée sur Auto-Encoder : 50
2.4.3 Recommandation basée sur les réseaux neuronaux convolutionnels : 51
2.4.4 Recommandation basée sur les réseaux neuronaux récurrents : 53
2.5 Conclusion : 55
Approche proposée 57
3.1 Introduction : 58
3.2 Architecture générale de l'approche proposée : 58
3.3 Principe de fonctionnement : 59
3.3.1 Pré-traitement : 59
3.3.2 L'apprentissage : 60
3.3.3 La recommandation : 63
3.4 Conclusion : 63
Evaluation expérimentale de l'approche proposée 64
4.1 Introduction : 65
4.2 Cadre d'évaluation : 65
4.2.1 Corpus de données (Collection de données) 65
4.2.2 Langage de programmation 66
4.2.3 Métriques d'évaluation 67
4.3 Partie application : 70
4.3.1 Recommandation en ligne : 71
4.3.2 Recommandation hors ligne : 72
4.4 Comparaison entre notre modèle et les autres modèles étudiés [105] : 75
4.5 Conclusion : 77
Conclusion générale et perspectives 78
Références Bibliographique 79

Table des figures

Figure 1.1 : Les étapes d'un système de recommandation.....21
Figure 1.2 : Exemple de matrice utilisateur 22
Figure 1.3 : Un système de recommandation basé sur le contenu 23
Figure 1.4 : Un système de recommandation collaboratif 25
Figure 1.5 : Le système de recommandation hybride 27
Figure 1.6 : Conception d'hybridation monolithique 28
Figure 1.7 : Conception d'hybridation parallèle 28
Figure 1.8 : Conception d'hybridation tubulaire 28
Figure 2.1 : problème linéairement séparable.....37
Figure 2.2 : problème n'est pas linéairement séparable.....37
Figure 2.3 : Structure d'un réseau de neurone 38
Figure 2.4 : Illustration d'un arbre de décision 39
Figure 2.5 : Fonction d'appartenance dans K-Means/Fuzzy C-means 40
Figure 2.6 : le dendrogramme de la hiérarchie H de la suite de partitions d'un ensemble 42
Figure 2.7 : Relation entre l'intelligence artificielle, le ML et le Deep Learning 43
Figure 2.8 : Schéma illustratif de DL avec plusieurs couches.....43
Figure 2.9 : Comparaison entre la machine Learning et le Deep Learning. 45
Figure 2.10 : Différents modèles du Deep Learning 47
Figure 2.11 : Multi-Layer Perceptron 49
Figure 2.12 : Auto-Encoder 50
Figure 2.13 : Convolutional Neural Network.....52
Figure 2.14 : Recurrent Neural Networks.....54
Figure 3.1: Architecture générale de l'approche proposée.....58
Figure 3.2 : Architecture de prétraitement. 60
Figure 3.3 : Architecture de Multi Layer Perceptron (MLP).....62
Figure 3.4 : Regroupement par l'algorithme K-Means 62
Figure 3.5 : L'architecture de la recommandation. 63
Figure 4.1: Logos de quelques librairies utilisées.....67
Figure 4.2 : Les résultats de RMSE pour le MLP.....68
Figure 4.3 : Les résultats de MAE pour le MLP 69
Figure 4.4 : Processus d'évaluation offline.....71
Figure 4.5 : Recommandation en ligne.....71
Figure 4.6 : Capture de la partie saisis des informations d'un nouvel utilisateur.....71
Figure 4.7 : Capture des résultats de la recommandation de films à un utilisateur 73
Figure 4.8 : Capture sur les premiers 20 films recommandés à l'utilisateur par notre modèle hybride…….....74
Figure 4.9 : Précision et Rappel du modèle MLP...…….....75
Figure 4.10 : Précision et Rappel du modèle hybride...……76
Figure 4.11 : Résultats d'évaluations des techniques avec différents algorithmes avec dataset MovieLens1M…….....……..…….....…….….....76

Liste des tableaux

Tableau 1.1 : Les avantages et les inconvénients des techniques de recommandation 30
Tableau 2.1 : Ensemble d'apprentissage sur les symptômes 39
Tableau 3.1: Une description des formes utilisées dans l'architecture.....59
Tableau 4.1: Tableau de description des champs du fichier users 65
Tableau 4.2 : Tableau de description des champs du fichier movies.....66
Tableau 4.3 : Tableau de description des champs du fichier ratings 66
Tableau 4.4 : Le résultat de RMSE pour le MLP 68
Tableau 4.5 : Le résultat de MAE pour le MLP 68
Tableau 4.6 : Moyenne de Précision et Rappel à N éléments du modèle MLP 75
Tableau 4.7 : Moyenne de Précision et Rappel à N éléments du modèle hybride 75
Tableau 4.8 : Résultats d'évaluations des techniques avec différents algorithmes avec dataset MovieLens1M. 75

Liste des Abréviation

Acronyme	Signification
ACP	Analyse en Composantes Principales.
FC	Filtrage Collaboratif.
FBC	Filtrage basé contenue.
SDAE	stacked denoising auto-encoder
PMF	probabilistic matrix factorization
POI	Point-of-Interest
NCF	Neural Collaborative Filtering
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
AE	Autoencoder
ID3	Itérative Dichotomiser 3
C4.5, C5	Successeurs d'ID3
FCM	FuzzyC-Means
CHA	La classification ascendante hiérarchique

MLP	multilayer perceptron
LSTM	Long Short-Term Memory
PMF	Probabilistic Matrix Factorization
CDAE	Collaborative Denoising Auto-encoder
POI	Point-of-Interest
IA	Intelligence artificielle
GRU	Gated Recurrent Unit

Introduction générale

Durant les dernières décennies et avec l'évolution technologique ainsi que l'apparition d'internet, la base de données à exploiter ou analyser est devenue très volumineuse. Ce qui rend difficile d'accéder les données à rechercher. Certaines méthodes informatiques ont été proposées pour rendre cette recherche facile aussi que l'extraction des informations appropriées. Ces approches présentent une aide très utile lors de la prise de décision.

Nous nous focalisons dans ce travail sur le processus de recommandation qui oriente l'utilisateur lors de son exploration, en cherchant pour lui les informations appropriées. Ce processus présente d'une manière spécifique de filtrage visant à offrir les éléments d'information que l'utilisateur peut y intéresser.

Les systèmes de recommandation ont été l'objet de plusieurs études dans divers domaines, à savoir : le web, le e-commerce, etc. Un système de recommandation représente un outil logiciel spécialement conçu pour générer et proposer des éléments ou du contenu qu'un utilisateur donné veut utiliser. À l'aide de certaines techniques issues d'apprentissage automatique et de diverses données relatives à des produits individuels et à des utilisateurs individuels, le système crée un réseau avancé de connexions complexes entre ces produits et ces personnes.

A l'instar des approches informatiques traditionnelles, l'apprentissage automatique est un nouvel outil puissant dans le domaine en développement continu orienté vers la résolution de divers problèmes. Avec L'apparition des systèmes de recommandations récents via les algorithmes d'apprentissage, il a été montré que ceux-ci peuvent donner des résultats pertinents et ainsi améliorer les performances des recommandations proposées aux utilisateurs.

L'application d'apprentissage automatique sur les systèmes de recommandation a fait une révolution dans ce domaine avec des recommandations adaptées à l'intérêt des utilisateurs mais :

- Qu'est ce qu'on doit recommander à un utilisateur qui n'a vu aucun film ?
- Comment minimiser le temps nécessaire pour la recommandation ?
- Comment donner une chance à des nouveaux items qui peuvent intéresser l'utilisateur ?

Dans ce projet, nous proposons un nouveau modèle de recommandation qui adopte des algorithmes d'apprentissage automatique, au début, nous parlerons du processus de recommandation et d'apprentissage automatique ainsi que ses travaux connexes. Puis, nous détaillerons la méthode de la création de ce modèle ainsi que ses étapes, et en parallèle en montrant la nouveauté de ce modèle par rapport aux systèmes traditionnelles, dont l'objectif est d'augmenter ou améliorer les résultats obtenus. A la fin de ce rapport de mémoire, nous

allons présenter nos résultats obtenus pour montrer l’efficacité et les performances de l‘approche proposée.

Ce mémoire est divisé en quatre chapitres ;Le premier présente les systèmes de recommandations, les différentes approches et les métriques d'évaluation ; Le second, l’apprentissage automatique, les différents types d’apprentissage, et quelque d’algorithmes ; Le troisième consiste à la représentation du modèle proposé, et le dernier présente l’expérimentation et l’évaluation de notre système de recommandation.

Partie 1 : Etat de l’art

Chapitre1 :présentation des systèmes de recommandation

Chapitre2 : L’apprentissage automatique, l’apprentissage profond et l’apprentissage profond pour la recommandation

Chapitre 1

Les systèmes de recommandation

1.1 Introduction :

Avec le problème de la croissance phénoménale de la quantité d'informations disponibles sur Internet et le besoin de changer d'un état de la rareté à l'état de l'abondance, il est nécessaire de disposer d'un système qui peut recommander des bons produits en tenant compte de certains paramètres. Il s'agit là du système de recommandation qui est devenu plus populaire durant ces 10 à 20 dernières années. Un système de recommandation fournit à des utilisateurs des suggestions qui répondent à leurs besoins et préférences informationnels. Les applications de recommandation peuvent être trouvé dans une grande variété d'industries, entreprises, service financier, musique, radio en ligne, publications en ligne, TV et vidéos, et d'innombrables autres.

1.2 Historique :

Les systèmes de recommandation ont émergé comme un domaine de recherche à part entière au milieu des années 1990. Les systèmes de recommandation ont été utilisés dans de nombreux domaines recommandant toutes natures d'items : pages Web, images, livres, news, films, musique, restaurants, objets, etc. Les systèmes de recommandation ont vu leur popularité croître ces dernières années en raison de la démocratisation du web et de l'augmentation exponentielle de la quantité de ressources disponibles et accessibles tels qu'Amazon, Netflix [01].

Les systèmes de recommandation ont été introduits comme une technique intelligente pour faire le filtrage de l'information afin de présenter les éléments d'information qui sont susceptibles d'intéresser l'utilisateur. Ils peuvent être utilisés pour fournir efficacement des services personnalisés dans la plupart des domaines [02].

Les origines des systèmes de recommandation sont issues des travaux étendus dans les sciences cognitives, la théorie d'approximation, la recherche d'informations, la théorie de la prévoyance sont aussi liées avec la science de la gestion et le marketing, dans la modélisation des choix du consommateur [03].

« Information Lens System » peut être considérée comme le premier système de recommandation. À l'époque, l'approche la plus commune pour le problème de partage d'informations dans l'environnement de messagerie électronique était la liste de distributions basée sur les groupes d'intérêt. La première définition pour le filtrage a été donnée aussi par Malone : « Même si le terme a une connotation littérale de laisser les choses dehors (filtrage négatif : enlèvement), nous l'utilisons ici dans un sens plus général qui consiste à sélectionner les choses à partir d'un ensemble plus large de possibilités (filtrage positif : sélection) » [04].

Au cours des années 90, les systèmes de recommandation sont devenus un important domaine de recherche avec la publication des premiers articles dans le domaine du filtrage collaboratif. La littérature académique a introduit le terme de filtrage collaboratif par le système Tapes Try[05]. Il a été développé en 1992 par le centre de recherche de "Xerox" aux Etats Unis et

qui a permis aux utilisateurs de créer des requêtes permanentes, basées sur les annotations des utilisateurs.

À la fin des 1990, Les déploiements de la technique de recommandations commerciales ont commencé à émerger. Peut-être la plus célèbre application surune large gamme de systèmes techniques de recommandation est Amazon.com, En suit, en 2006, Netflix a lancé NetflixPrize pour améliorer l'état de recommandation des films ; aussi Netflix a plus de 17.000 films dans sa sélection [06].

Aujourd'hui les systèmes de recommandation sont devenus très populaires et sont utilisés dans diverses applications Web.

1.3 Définition des systèmes de recommandation :

Les systèmes de recommandation peuvent être définis de plusieurs façons, vue la diversité des classifications proposées pour ces systèmes qui peuvent se rapporter à différents types de données ou approches spécifiques, mais il existe une définition générale de Robin Burke [07] que nous utiliserons dans ce mémoire et qui les définit comme suit :

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

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

Les données d'entrée pour un système de recommandation dépendent du type de l'algorithme de filtrage employé. Généralement, elles appartiennent à l'une des catégories suivantes [08] :

- **Les estimations :** également appelées les votes, expriment l'opinion des utilisateurs sur les articles (exemple : de 1 mauvais à 5 excellent).
- **Les données démographiques :** se réfèrent à des informations telles que l'âge, le sexe, le pays et l'éducation des utilisateurs. Ce type de données est généralement difficile à obtenir et est normalement collecté explicitement.
- **Les données de contenu :** qui sont fondées sur une analyse textuelle des documents liés aux éléments évalués par l'utilisateur. Les caractéristiques extraites de cette analyse sont utilisées comme entrées dans l'algorithme de filtrage afin d'en déduire un profil d'utilisateur.

1.4 Objectif des systèmes de recommandation :

Un système de recommandation a pour objectif de fournir à un utilisateur des ressources pertinentes en fonction de ses préférences. Ce dernier voit ainsi réduit son temps de recherche

mais reçoit également des suggestions de la part du système auxquelles il n'aurait pas spontanément prêté attention. L'essor du Web et sa popularité ont notamment contribué à la mise en place de tels systèmes comme dans le domaine du e-commerce. Les systèmes de recommandation peuvent être vus initialement comme une réponse donnée aux utilisateurs ayant des difficultés à prendre une décision dans le cadre d'utilisation d'un système de recherche d'information « classique »[09].

1.5 Filtrage d'information Vs Recherche d'information :

La différence entre le filtrage d'information et la recherche d'information n'est pas toujours claire ; mais une première différence peut être tirée de leur définition. Généralement, un système de recherche d'information est un système ayant pour fonction d'apporter aux utilisateurs les documents qui leur permettent de satisfaire leurs besoins en information, en se basant sur les requêtes qu'ils fournissent [10], tandis qu'un système de filtrage d'information fournit des documents (ou des items) à une personne ou ensemble de personnes, en se basant sur leurs profils à long terme [11].

Un système de recherche d'information a pour but de guider les utilisateurs à retrouver les items qui leur permettent de répondre à leurs besoins, alors qu'un système de filtrage leur recommande directement les items intéressants qui conviennent à leurs préférences et besoins[12].

Dans un système de recherche d'information, l'utilisateur saisie une requête et le système la compare avec les sources d'information disponibles, pour trouver un ensemble de documents répondant à sa requête. Les sources d'information utilisées par le système de recherche d'information sont un ensemble de données collectées dans des bases de données et souvent indexés par un ensemble de mots clés[12].

Dans un système de filtrage d'information, le système compare l'ensemble des informations disponibles aux profils des individus ou groupe d'individus, afin de leur suggérer les items et/ou informations correspondantes à leurs besoins. Dans ces systèmes, les items et/ou informations proviennent directement de la part des producteurs ou bien collectés dans des bases de données[12].

Par conséquent, un certain nombre de points de distinction entre la recherche d'information et le filtrage d'information peuvent être cités [13] :

- Dans un système de recherche d'information, les données sont sélectionnées à partir d'une base de données statique, alors que le filtrage d'information établit une sélection ou bien une élimination d'information à partir d'une source de données dynamique.

- Un système de recherche d'information ne peut traiter qu'une seule requête à la fois, ce qui le rend capable d'être utilisé que par une seule personne à un moment donné, tandis qu'un système de filtrage peut être utilisé par une ou plusieurs personnes en même temps.
- Un système de recherche d'information collecte et organise les documents selon la recherche des individus, tandis qu'un système de filtrage d'information distribue automatiquement des documentset/ou des items aux personnes qui en ont besoin.
- Un système de recherche d'information utilise le principe de la découverte de l'information dans la base de données alors que le filtrage d'information utilise un processus d'élimination d'information du flux de données entrant.
- Un système de recherche d'information utilise les requêtes fournies par l'utilisateur qui représentent des intérêts à court terme. Par contre, le filtrage d'information utilise les profils des utilisateurs qui représentent généralement des intérêts à long terme, et qui peuvent être considérés comme des descriptions plus exactes des intérêts des utilisateurs que les requêtes.
- Un système de recherche d'information autorise l'utilisateur à interagir avec un document pendant une seule session de recherche, alors que le filtrage d'information permet l'interaction dans différentes sessions et prend en considération les changements de profils à travers les différentes sessions.

1.6 Processus de recommandation :

Le processus de génération d'une recommandation dans un système de recommandation est basé sur les points suivants [14] :

- Le type de donnée disponible dans sa base de données (les évaluations, les informations d'inscription des utilisateurs, caractéristiques et contenus des items qui peuvent être classés, et les relations sociales entre les utilisateurs, …) ;
- L'algorithme de filtrage utilisé (par exemple : démographique, basé contenu, collaboratif, basé social, hybride, …) ;
- Le modèle choisi (basé sur l'utilisation directe des données : "basé-mémoire", ou un modèle généré à l'aide des données : " basé-modèle") ;
- La technique employée : approche probabiliste, les réseaux bayésiens, les algorithmes des plus proches voisins, les réseaux de neurones, les algorithmes génétiques, les modèles flous, … ;
- La performance du système (temps et mémoire) ;
- L'objectif recherché (par exemple : la prédiction des K Top recommandations) ;

- La qualité des résultats souhaités (par exemple : la nouveauté et la précision).

1.7Les étapes principales de la recommandation :

Un système de recommandation requiert généralement trois étapes, et qui sont illustré dans la figure suivante :

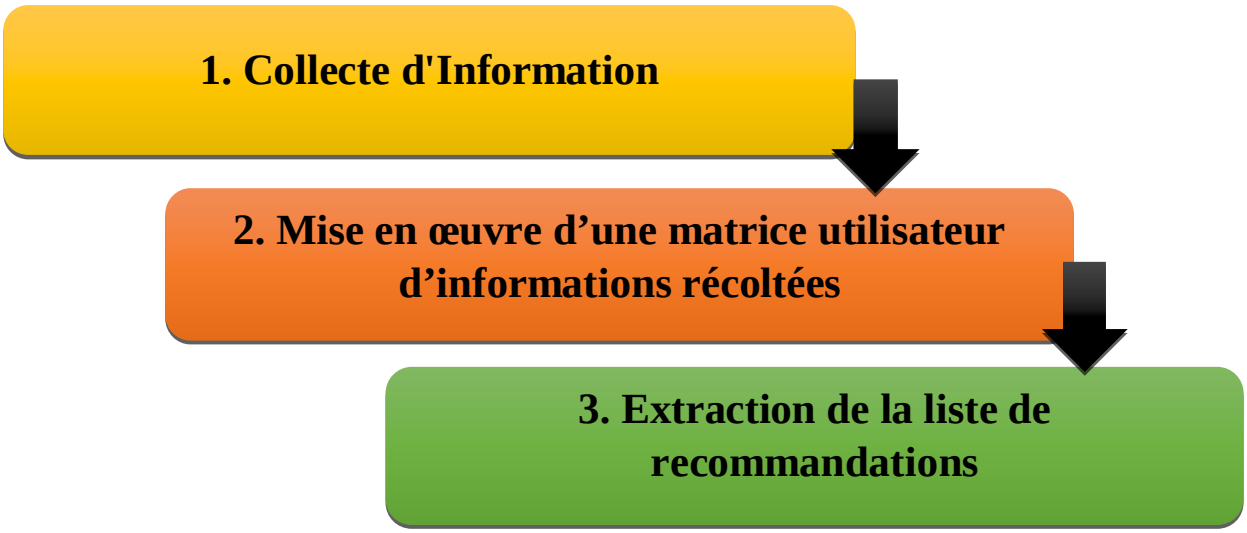


Figure 1.1 : Les étapes d'un système de recommandation[15]

1.7.1 Collecte d'Information :

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

Une distinction peut être faite entre deux formes de collecte de données [15] :

- **Collecte de données explicites – Filtrage actif :**

Repose sur le fait que l'utilisateur indique explicitement au système ses intérêts.

- **Exemple :** demander à un utilisateur de commenter, taguer/étiqueter, noter, aimer où encore ajouter comme favoris des contenus (objets, articles, …) qui l'intéressent. On utilise souvent une échelle de ratings allant de 1 étoile (je n'aime pas du tout) à 5 étoiles (j'aime beaucoup) qui sont ensuite transformées en valeurs numériques afin de pouvoir être utilisées par les algorithmes de recommandation.
- **Avantage :** capacité à reconstruire l'historique d'un individu et capacité à éviter d'agréger une information qui ne correspond pas à cet unique utilisateur (plusieurs personnes sur un même poste).
- **Inconvénient :** Les informations recueillies peuvent contenir un biais dit de déclaration. Il s'agit de l'écart entre la donnée déclarée et celle mesurée.

- **Collecte de données implicite – Filtrage passif :**

La collecte des données implicite repose sur une observation et une analyse des comportements de l'utilisateur, effectué de façon implicite dans l'application qui embarque le système de recommandation, le tout se fait en “arrière-plan” (sans rien demander à l'utilisateur).

- **Exemples :**
  - Obtenir la liste des éléments que l'utilisateur a écoutés, regardés ou achetés en ligne.
  - Analyser la fréquence de consultation d'un contenu par un utilisateur, le temps passé sur une page.
  - Monitorer le comportement en ligne de l'utilisateur.
  - Analyser son réseau social.
- **Avantage :** Aucune information n'est demandée aux utilisateurs, toutes les informations sont collectées automatiquement. Les données récupérées sont a priori justes et ne contiennent pas de biais de déclaration.
- **Inconvénient :** Les données récupérées sont plus difficilement attribuables à un utilisateur et peuvent donc contenir des biais d'attribution (utilisation commune d'un même compte par plusieurs utilisateurs). Un utilisateur peut ne pas aimer certains livres qu'il a achetés, ou il peut l'avoir acheté pour quelqu'un d'autre.

1.7.2 Mise en œuvre d’une matrice utilisateur d’informations récoltées :

La mise en œuvre d’une matrice appelée " matrice utilisateur " ou bien " modèle utilisateur " incluant les informations concernant les utilisateurs récoltées durant la précédente étape qui est la collecte d’informationspeut être représenté comme un tableau qui contient des données recueillies sur l’utilisateur associées aux produits disponibles sur le site web.

	Item 1	Item 2	Item 3	Item 4	Item 5	...
User 1	😃			😃		...
User 2		☹	😃			...
User 3		😃				...
User 4			☹	😃	☹	...
...	...	...	...	...	...	...

Figure 1.2 : Exemple de matrice utilisateur [15]

Lafigure ci-dessus présente un exemple fictif de matrice binaire contenant des informations de type "l’utilisateur U a apprécié/n’a pas apprécié l’item I". Ces informations peuvent également être " a acheté / n’a pas acheté ", "a consulté/n’a pas consulté ", ...Elles peuvent également se mesurer sur un nombre plus élève de classes : " a mis 1/2/3/4/5 étoiles ",...Dans ce sens, un autre point également important consiste en connaître comment le temps influence

le profil de l'utilisateur. Les intérêts des utilisateurs, évoluent en générale en fonction du temps. Les données du modèle utilisateur devraient donc constamment êtreréadaptéespour qu'elles soient toujours conformes aux nouveaux centres d'intérêts de l'utilisateur [15, 16].

1.7.3 Extraction de la liste de recommandations :

Pour extraire une liste de suggestions c.à.d. de recommandation à partir d'une matrice utilisateur, les algorithmes utilisent la notion de mesure de similarité entre objets ou personnes décrits par le modèle utilisateur. La similarité a pour but de donner une valeur ou un nombre (au sens mathématique du terme) à la ressemblance entre deux choses. Plus la ressemblance est forte, plus la valeur de la similarité sera grande. A l'inverse, plus la ressemblance est faible, et plus la valeur de la similarité sera petite [15,16].

1.8 Les principales approches de filtrage :

Un système de recommandation doit pouvoir faire des prédictions sur les intérêts des utilisateurs. Il faut donc pouvoir collecter un certain nombre de données sur ceux-ci afin d'être capable de construire un profil pour chaque utilisateur. Nous pouvons distinguer trois grandes classes d'approches selon la nature de ces données.

1.8.1 Filtrage basé sur le contenu (content basedfiltering) :

Pour les recommandations basées sur le contenu [17], la tâche consiste à déterminer quels éléments du catalogue coïncident le mieux avec les préférences de l'utilisateur. Une telle approche ne requiert pas une grande communauté d'utilisateurs ou un gros historique d'utilisation du système. La figure 1.3 illustre ce processus.

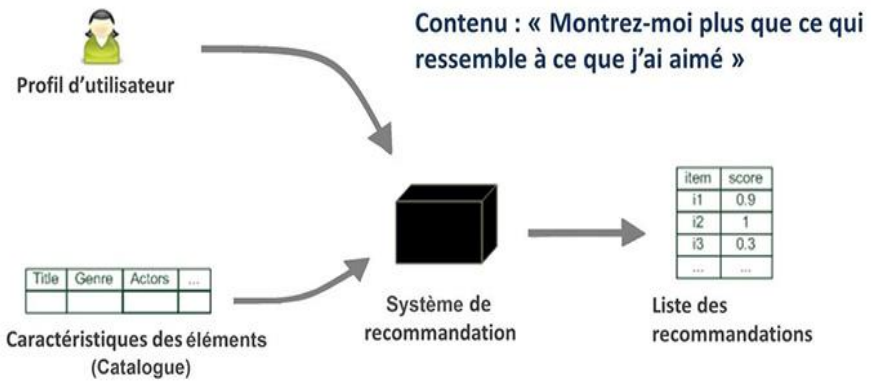


Figure 1.3 : Un système de recommandation basé sur le contenu [18]

La manière la plus simple de décrire un catalogue d'éléments est d'avoir une liste explicite des caractéristiques de chaque élément (on parle aussi d'attributs, de profil d'élément, …).

Pour un livre par exemple, on peut utiliser le genre, le nom des auteurs, l'éditeur ou toute autre information relative au livre, puis stocker ces caractéristiques (dans une base de données par exemple).Le profil de l'utilisateur est exprimé sous forme d'une liste d'intérêts basée sur

les mêmes caractéristiques. La coïncidence entre les caractéristiques des éléments et le profil de l'utilisateur peut être mesurée de différentes manières :

- L'indice de Dice ou d'autres mesures de similarité ;
- le TF-IDF (Term Frequency-Inverse Document Frequency);
- Les techniques basées sur la similarité des espaces vectoriels (les approches bayésiennes [06], les arbres de décision, …) couplées avec des techniques statistiques, lorsqu'il y a trop de mots-clés.

Exemple

U(c,i)	Harry Potter	L'Âge de glace	L'Âge de glace 2	OSS 117	Bienvenue chez les Ch'tis
Elsa	**?**	**8**	**?**	**2**	**7**
Marie	**9**	**8**	**?**	**3**	**6**
Arnaud	**3**	**5**	**?**	**5**	**?**
Patrick	**5**	**3**	**?**	**3**	**3**

L'utilisateur se verra recommander des éléments semblables (au sens d'une mesure desimilarité entre éléments) à ceux qu'il a préférés dans le passé.

Dans l'exemple ci-dessus, les films « L'Âge de glace » et « Bienvenue chez les Ch'tis » ont été évalués quasiment de la même manière par Elsa. On observe par ailleurs que les caractéristiques des films « L'Âge de glace » et « L'Âge de glace 2 » sont très « proches ». Le système recommande donc le film « L'Âge de glace 2 » à Elsa, en lui assignant le meilleur score qu'elle a attribué aux films « L'Âge de glace » et « Bienvenue chez les Ch'tis », c'est-à-dire 8.

1.8.2 Filtrage collaboratif (collaborative filtering) :

Les systèmes basés sur le filtrage collaboratif [19] produisent des recommandations en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs. De tels systèmes ne tentent pas d'analyser ou de comprendre le contenu des éléments à recommander. La méthode consiste à faire des prévisions automatiques sur les intérêts d'un utilisateur en collectant des avis de nombreux utilisateurs. L'hypothèse sous-jacente de cette approche est que ceux qui ont aimé un élément spécifique dans le passé auront tendance à aimer cet élément spécifique, ou un autre très « proche », à nouveau dans l'avenir. La figure 1.4 illustre ce processus.

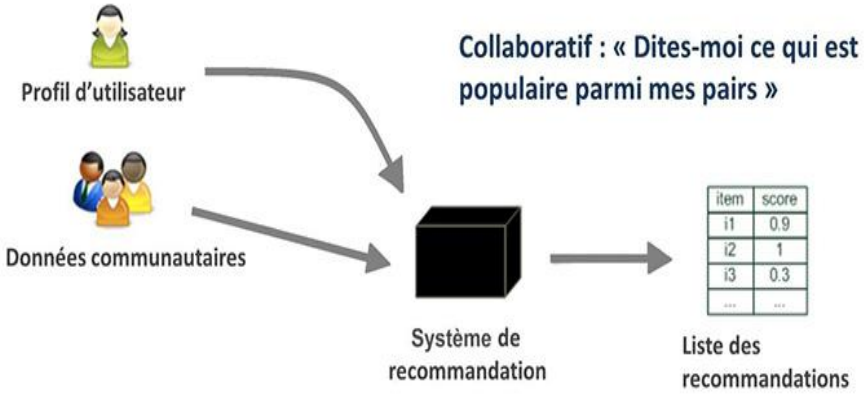


Figure 1.4 : Un système de recommandation collaboratif [18]

L'idée des approches collaboratives est d'essayer de prédire l'opinion d'un utilisateur sur les différents éléments. La recommandation est basée sur les goûts et avis précédents de l'utilisateur et sur une mesure de similarité avec d'autres utilisateurs.

Les principales étapes de cette approche sont :

- De nombreuses préférences d'utilisateurs sont enregistrées ;
- N sous-groupe d'utilisateurs est repéré dont les préférences sont similaires à celles de l'utilisateur qui cherche la recommandation ;
- Une moyenne des préférences pour ce groupe est calculée ;
- La fonction de préférence qui en résulte est utilisée pour recommander des éléments à l'utilisateur qui cherche la recommandation.

Il est possible de distinguer trois types d'approches pour définir la similitude ou la similarité :

- Les approches Item-to-Item basées sur la similarité entre les éléments (items). Notons que cette approche s'adapte à un nombre très important d'utilisateurs ou d'éléments.
- Les approches User-to-User basées sur la similarité entre les utilisateurs (users). Notons que cette approche n'est pas adaptée à un nombre très important d'utilisateurs.
- Les approches par modèle de l'utilisateur c'est L'approche classique pour les systèmes de recommandation consiste à construire des modèles des utilisateurs en fonction d'informations les concernant, comme leur intervalle de revenus, leur niveau culturel, leur âge, leurs habitudes culturelles, etc.,

Exemple :

Marie et Elsa ont des avis similaires ; Marie apprécie aussi « Harry Potter », c'est donc une bonne recommandation pour Elsa.

Utilisons maintenant l'approche Item-to-Item sur ce même exemple.

Elsa et Marie aiment « L'Âge de glace » et « Bienvenue chez les Ch'tis ». Cela suggère que, en général, les personnes qui aiment « L'Âge de glace » aimeront aussi « Bienvenue chez les Ch'tis », donc « Bienvenue chez les Ch'tis » pourra être recommandé à Arnaud (qui aime « L'Âge de glace »).

Utilisons maintenant l'approche Item-to-Item sur ce même exemple.

Elsa et Marie aiment « L'Âge de glace » et « Bienvenue chez les Ch'tis ». Cela suggère que, en général, les personnes qui aiment « L'Âge de glace » aimeront aussi « Bienvenue chez les Ch'tis », donc « Bienvenue chez les Ch'tis » pourra être recommandé à Arnaud (qui aime « L'Âge de glace »).

Les systèmes de recommandation collaboratifs, par leur diversité, s'appuient donc sur de nombreuses techniques, qu'il s'agisse de :

- Similarité entre utilisateurs (coefficient de corrélation de Pearson [20], …) ou de sélection de voisinage (les algorithmes basés sur la recherche de voisinage) pour les approches User-to-User ;
- Similarité entre éléments (la mesure de similarité cosinus [21] où A et B sont des utilisateurs, et θ c'est l'angle entre A et B $sim(A, B) = \cos\theta = \frac{A.B}{\|A\|\|B\|}$, …) pour les approches Item-to-Item ;
- Techniques de prédiction de scores (analyse en composantes principales ou ACP, factorisation de matrices, analyse sémantique latente, règles d'association, approches bayésiennes, ...) pour les autres approches.

1.8.3 Filtrage hybride (hybridfiltering) :

Un système de recommandation hybride utilise des composants de différents types d'approches de recommandation ou s'appuie sur leur logique[07]. Par exemple, un tel système peut utiliser à la fois des connaissances extérieures et les caractéristiques des éléments, combinant ainsi des approches collaboratives et basées sur le contenu [18].

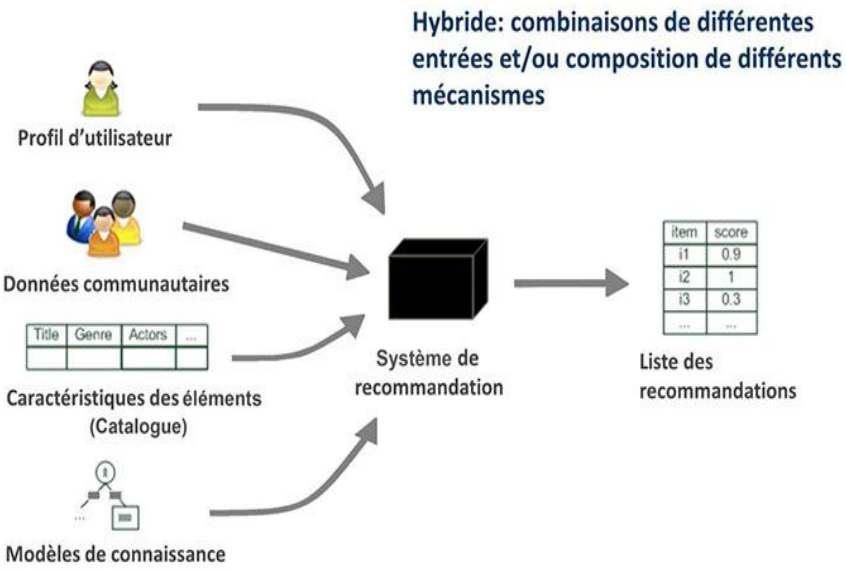


Figure 1.5 : Le système de recommandation hybride [18]

Il est à noter que le terme « hybride » est un artefact de l'évolution historique des systèmes de recommandation où certaines sources de connaissances ont été exploitées en premier lieu, conduisant à des techniques bien établies qui ont ensuite été combinées. L'objectif est alors de s'appuyer sur des sources de connaissances multiples, en choisissant les plus appropriées à une tâche donnée afin de les utiliser le plus efficacement possible, la figure 1.5 illustre ce processus.

Il existe trois grandes catégories de combinaisons de systèmes de recommandation pour concevoir un système de recommandation hybride [07, 18] : la combinaison monolithique (monolithichybridization design), la combinaison parallèle (parallelizedhybridization design) et la combinaison tubulaire (pipelinedhybridization design).

« Monolithique » décrit une conception d'hybridation qui intègre les aspects de différentes stratégies de recommandation en un seul algorithme. Comme illustré sur la figure 1.6 ci-dessous, différents systèmes de recommandation y contribuent puisque l'approche hybride utilise des données d'entrée additionnelles qui sont spécifiques à un autre algorithme de recommandation, ou bien les données d'entrée sont complétées par une technique et exploitées par une autre. Par exemple, un système de recommandation basé sur le contenu qui exploite également des données communautaires pour déterminer des similarités entre éléments relève de cette catégorie [07, 18].

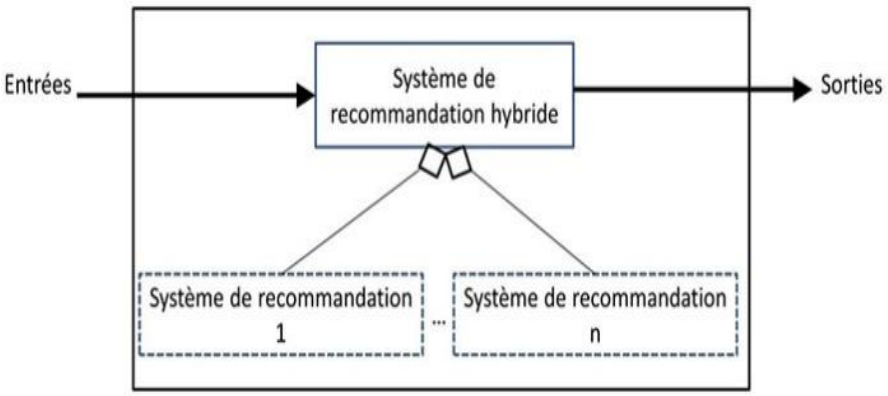


Figure 1.6 : Conception d'hybridation monolithique [18]

Les deux autres approches hybrides nécessitent au moins deux mises en œuvre de recommandations séparées qui sont combinées en conséquence. Sur la base de leurs données d'entrée, les systèmes hybrides de recommandation parallèles fonctionnent indépendamment l'un de l'autre et produisent des listes de recommandations distinctes, comme illustré sur la figure 1.7. Dans une étape ultérieure d'hybridation, leurs sorties sont combinées en un ensemble final de recommandations [07, 18].

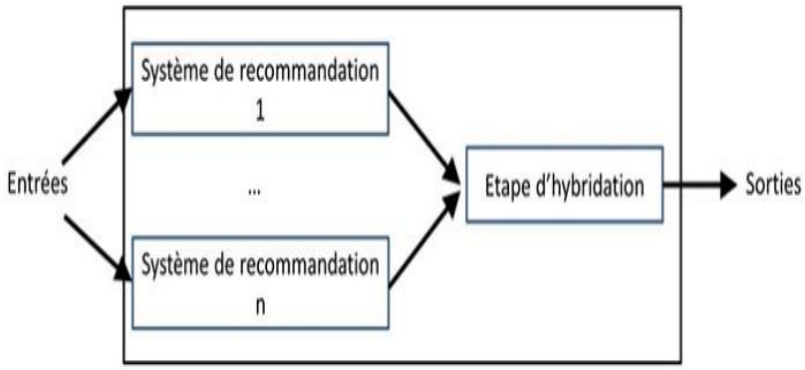


Figure 1.7 : Conception d'hybridation parallèle [18]

Lorsque plusieurs systèmes de recommandation sont joints dans une architecture tubulaire, comme illustré par la figure 1.8, la sortie de l'un des systèmes de recommandation devient une partie des données d'entrée du système suivant [07, 18].

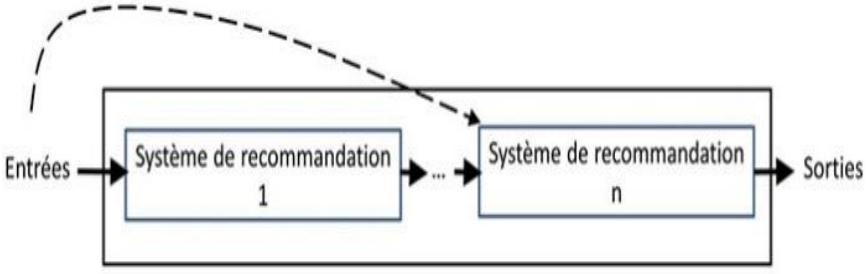


Figure 1.8 : Conception d'hybridation tubulaire [18]

1.9 Avantages et inconvénients des systèmes de recommandations :

Malgré leur popularité croissante, les systèmes de recommandation ont subi quelques ratés. Nous identifions quelques problèmes dont ils souffrent [15] :

- **Adaptabilité :** Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise.
- **Nouvel utilisateur** : Un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
- **Nouvel item :** Un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation.
- **Démarrage à froid :** Souvent, on se retrouve confronté au problème qu'un utilisateur ne soit comparable avec aucun autre. Ce problème est dû au fait que peu ou pas d'utilisateurs ont évalué un article donné, ou qu'un utilisateur donné a évalué très peu ou pas d'articles. Généralement, ce problème survient quand un nouvel utilisateur ou une nouvelle ressource est ajouté à la base de recommandation.
- **Le cas du système débutant** : provient lors du lancement d'un nouveau service de recommandation. Le système ne possède alors aucune information sur les utilisateurs et sur les items. Les méthodes de filtrage collaboratif ne peuvent pas fonctionner sur une matrice d'usages vide. La solution consiste en général à trouver des informations descriptives des items afin d'organiser le catalogue et inciter les utilisateurs à le parcourir jusqu'à ce que la matrice d'usages soit assez remplie et permette de passer en mode collaboratif. Le tableau 1.1 résume les forces et faiblesses des méthodes traditionnelles utilisées par les systèmes de recommandation, en l'occurrence le Filtrage Collaboratif (FC), le Filtrage à Base de Contenu (FBC)

Techniques	Avantages	Inconvénients
Filtrage à base de contenu	▪ Pas besoin d'une large communauté d'utilisateurs pour effectuer des recommandations. ▪ Une liste de recommandations peut être générée même s'il n'y qu'un seul utilisateur. ▪ La qualité croit avec le temps. ▪ Pas besoin d'information sur les autres utilisateurs. ▪ Prendre en considération les gouts uniques des utilisateurs.	▪ L'analyse du contenu est nécessaire pour faire une recommandation. ▪ Problème de recommandation des images et de vidéos en absence de métadonnées. ▪ Nécessité du profil d'utilisateur.
	▪ Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique. ▪ La qualité de la recommandation peut être	▪ Démarrage à froid. ▪ Nouvel Item. ▪ Nouvel utilisateur.

Filtrage collaboratif	évaluée. ▪ Plus les nombre d'utilisateurs est grand plus la recommandation est meilleure.	▪ Problème de confidentialité. ▪ La complexité dans les systèmes avec un grand nombre d'items et d'utilisateurs. ▪ Le calcul croit linéairement.

Tableau 1.1 : Les avantages et les inconvénients des techniques de recommandation

1.10 Evaluation des Systèmes de recommandation :

Evaluer un système de recommandation permet de mesurer ses performances vis-à-vis de ses objectifs. De ce fait, les dimensions à évaluer diffèrent selon les objectifs fixés (prédiction des notes des utilisateurs pour les items, augmentation des ventes, …)

Il existe une classification des méthodes d'évaluation des systèmes de recommandation. Comme expliqué dans [22], ces méthodes d'évaluation sont classées en trois types : expérimentations offline, études avec des utilisateurs (user studies) et tests réels (real life testing). Ce dernier type est nommé expérimentation en ligne (Online experiments) par [23].

1.10.1 Évaluation online :

L'évaluation online peut aussi recueillir le point de vue de l'utilisateur concernant le système de recommandation. Dans ce type d'évaluation, des utilisateurs réels utilisent le système dans des conditions réelles sur une longue période [08].

Selon Herlocker[24] ce type d'évaluation peut montrer les usages et les habitudes d'utilisation des utilisateurs, les problèmes et les besoins non satisfaits, et les problèmes que les chercheurs n'ont peut-être pas envisagés dans une étude utilisateurs. Avec ces tests réels sur le terrain, la plupart des objectifs centrés sur l'utilisateur peuvent être efficacement évalués, comme l'évaluation de l'expérience utilisateur, la satisfaction des utilisateurs ou la rétention des utilisateurs [23, 24].

1.10.2 Étude utilisateurs :

Un système de recommandation a un objectif qui dépasse les métriques de précision. En fait, les évaluations basées sur telles métriques de précision n'offrent pas la connaissance si les utilisateurs sont réellement satisfaits des recommandations proposées par le système, c.à.d. elles ne donnent pas un feedback sur leurs efficacités [25, 26],environ 50% de la réussite commercial d'un système de recommandation donné est directement lié aux interactions entre les utilisateurs et le système lui-même, tandis que l'algorithme de recommandation ne compte que pour 5% [27].

L'avis de l'utilisateur peut être adopté à partir une étude utilisateurs. Il s'agit d'une approche employée pour connaitre comment un système de recommandation agit sur l'expérience, la perception et les interactions des utilisateurs avec le système [26]. Une étude nommée utilisateurs est adoptée en recrutant un ensemble d'utilisateurs, et en leur demandant de

réaliser certaines tâches dans un milieu contrôlé pendant une courte période de temps. Cette interaction entre les utilisateurs et le système de recommandation est observée et des informations sont enregistrées comme le temps nécessaire pour terminer une tâche ou une qualité des résultats d'une tâche. En outre, il est aussi possible de faire passer certains questionnaires aux utilisateurs pour mettre en œuvre et exploiter des données qui ne sont pas directement observables comme l'appréciation de l'interface utilisateur ou de la pertinence des recommandations.

1.10.3 Evaluation offline :

Une grande partie du travail d'évaluation des algorithmes des systèmes de recommandation s'est concentrée sur l'analyse hors ligne de la précision des prédictions que peuvent faire ces systèmes [24].

Les évaluations hors lignes utilisent des ensembles de données (dataset) constitués d'actions des utilisateurs. Les évaluations offlines simulent le processus de recommandation où un sous ensemble des actions utilisateurs du dataset est caché et le système de recommandation prédit ces actions cachées. Le système de recommandation est évalué en fonction de sa capacité à prédire ces interactions cachées. Les résultats de ces prédictions sont analysés en utilisant une ou plusieurs métriques [30].

- **Protocoles d'évaluation :**

Il en existe trois selon [31] :

- **Retrait (Hold out) :** un pourcentage du dataset est utilisé comme données d'apprentissage tandis que le reste est utilisé comme données de test.
- **Tous sauf 1 (All but 1) :** Un seul item du dataset est utilisé comme données de test, ce qui maximise la quantité de données utilisées comme données d'apprentissage.
- **Validation croisée K-fold (K-fold cross validation) :** Les données d'évaluation sont divisées K fois en données d'apprentissage et en données de test.

- **Métriques d'évaluations :**

Les métriques d'évaluation permettent de mesurer la capacité de l'algorithme de recommandation. Ces métriques utilisent des évaluations correctes d'un dataset de test et les comparent avec les résultats prédictions proposées par l'algorithme de recommandation en se basant sur un dataset d'apprentissage.

La majorité de ces métriques proviennent du domaine de la recherche d'information. La précision et le rappel [30] font partie de ces métriques.

- **La précision :** est une mesure d'exactitude. Elle détermine la proportion des items pertinents recommandés parmi tous les items recommandes : autrement dit, la proportion des recommandations qui se sont avérées pertinentes

- **Le rappel :** est une mesure d'exhaustivité qui détermine la proportion des items pertinents recommandés parmi tous les items pertinents.

D'autres métriques sont utilisées pour évaluer les systèmes de recommandation telle l'erreur quadratique absolue (MAE) et l'erreur quadratique moyenne (RMSE).

- **MAE :** calcule l'écart entre les notes prédites et les vraies notes.
- **RMSE :** elle repose sur le même principe que la MAE, tout en mettant l'accent sur les écarts importants [31].

Soit $T \subset$ Iun ensemble test d'items, $p(u,i)$ une prédiction de note de l'utilisateur u pour l'item i et $n(u,i)$ la note réelle attribuée par l'utilisateur u pour l'item i.

$$MAE = \frac{1}{T}\sum_{i\in t,u\in U} |p(u,i) - n(u,i)| \quad (1.1)$$

$$RMSE = \frac{1}{T}\sqrt{\sum_{i\in T,u\in U}(p(u,i) - n(u,i))2} \quad (1.2)$$

Lors d'une évaluation hors-ligne, il s'agira donc de prédire, grâce à l'échantillon d'apprentissage et du système de recommandation, une note $p(u,i)$ pour chaque item i, de l'échantillon detest, noté $n(u,i)$ par un utilisateur u. La performance du système est ensuite calculée en utilisant l'une des mesures précédentes

1.11 Conclusion :

Ce chapitre avait pour objectif de faire un tour d'horizon, non exhaustif, sur les systèmes de recommandation. Nous avons évoqué leur historique, une définition théorique de ce qu'un système de recommandation, Nous avons présenté les trois principaux types de systèmes de recommandation : ceux basés sur le contenu, ceux basés sur un filtrage collaboratif, et finalement les approches hybrides. Ensuite, nous avons présenté les avantages et les inconvénients de chaque type de système de recommandation. Et en fin les algorithmes et les métriques d'évaluation.

Chapitre 2

L’apprentissage automatique, l’apprentissage profond et l’apprentissage profond pour la recommandation

2.1 Introduction :

Durant ces deux dernières décennies, l'Apprentissage Automatique (Machine Learning) a connu une explosion énorme surtout au niveau des résultats qu'il donne, ils ont attiré l'attention de plusieurs chercheurs dans le domaine. Cette explosion a permis le développement rapide du Machine Learning. Ce dernier a donné ce qu'on l'appelle le Deep Learning qui a donné de meilleurs résultats, et bien sûr avant de parler de ces deux concepts, on doit tout d'abord parler de l'intelligence artificielle (IA).

L'IA a venu avec un objectif de doter à une machine des capacités de réflexions similaires à celles des humains, il a été considéré comme l'un des sujets les plus populaires durant ces dernières années[32].

En fait, L'IA est devenue un sujet en vogue dans les médias et revues scientifiques en raison de ses nombreuses applications, dont beaucoup sont le fruit des progrès accomplis dans le domaine de l'apprentissage automatique. De grandes entreprises comme Google, Facebook, IBM, Microsoft mais aussi des constructeurs automobiles à l'instar de Toyota, Volvo et Renault, sont très actifs dans la recherche en IA et prévoient d'y investir davantage encore dans le futur. Plusieurs chercheurs spécialisés dans l'IA dirigent désormais les laboratoires de recherche de ces grandes entreprises et de nombreuses autres. La recherche en IA a permis de réaliser d'importants progrès dans la dernière décennie, et ce dans différents secteurs. Les avancées les plus connues sont celles réalisées dans l'apprentissage automatique, grâce notamment au développement d'architectures d'apprentissage profond, des réseaux de neurones convolutifs multicouche dont l'apprentissage s'opère à partir de gros volumes de données sur des architectures de calcul intensif. Parmi les réalisations de l'apprentissage automatique, il convient de citer la résolution de jeux Atari (Bricks, Spaceinvaders, …) par Google DeepMind, utilisant les pixels images affichés à l'écran comme données d'entrée afin de décider quelle action adopter pour atteindre le plus haut score possible à la fin de cette partie. [33, 34].

Dans ce chapitre, nous allons parler des différents types d'apprentissage automatique et présenter leur principe de façon très brève, nous passerons ensuite à l'apprentissage profond et nous passerons en revue ses différents types.

2.2 Apprentissage automatique :

Dans cette partie, nous essayons de donner une vue générale sur l'apprentissage machine et ses deux déclinaisons : l'apprentissage supervisé et l'apprentissage non supervisé. L'apprentissage machine est le domaine s'intéressant à comprendre et reproduire la faculté de l'apprentissage humain par des systèmes artificiels. Il s'agit, très schématiquement, de concevoir des algorithmes et des méthodes permettant d'extraire l'information pertinente de données, ou d'apprendre des comportements à partir d'exemples. Ainsi, le but essentiel de l'apprentissage machine est de déterminer la relation entre les objets et leurs catégories pour la

prédiction et la découverte des connaissances [35]. Nous distinguons ainsi l'apprentissage supervisé de l'apprentissage non supervisé.

2.2.1 Apprentissage supervisé :

L'apprentissage supervisé a pour but d'établir des règles de comportement à partir d'une base de données contenant des exemples de cas déjà étiquetés. La base de données est en principe un ensemble de couples entrées / sorties {(X, Y)}. Le but est d'apprendre à prédire pour toute nouvelle entrée X, la sortie Y [36].

La technique de la détection de page Web spam correspond à un problème de classification supervisé[37], Dans la classification supervisée, les pages précédemment classées forment un ensemble d'entraînement pour décider si la page est spam ou non.

Il existe plusieurs algorithmes et techniques utilisés pour la classification supervisée [38] :

2.2.1.1 Classification Bayésienne :

Les classificateurs bayésiens (appelés également classificateurs naïfs bayésiens) sont des classificateurs statistiques [39]. Leur particularité est de prédire la probabilité d'appartenance à une classe.

Les classificateurs naïfs bayésiens supposent que l'effet d'une valeur d'attribut sur une classe donnée est indépendant des valeurs des autres attributs. Cette hypothèse s'appelle l'indépendance conditionnelle de classe. C'est dans ce sens qu'ils sont considérés comme étant « naïf ». Cette classification porte le nom de Thomas Bayes [39], qui avait proposé le théorème de Bayes dont l'énoncé est :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.1)$$

Où :

- $P(A)\ et\ P(B)$Désignent les probabilités à priori.
- $P(A|B)$Désigne la probabilité conditionnelle de A sachant B.

Un classificateur bayésien fonctionne comme suit [39] :

a) Soit **D,** un ensemble d'apprentissage de tuples et leurs étiquettes de classe associées. Chaque tuple est représenté par un vecteur d'attributs à n dimensions X = (x_1, x_2, …, x_n), représentant **n** mesures effectuées sur le tuple à partir de **n** attributs, respectivement, A_1, A_2, …, A_n.

b) Supposons que nous avons **m** classes, C_1, C_2, ..., C_m. Étant donné un tuple X, le classifieur prédira que X appartient à la classe ayant la probabilité à posteriori la plus élevée, conditionnée par X. Autrement dit, le classifieur bayésien prédira que le tuple X appartiendra à la classe C_i si et seulement si :

$$P(C_i|X) > P(C_j|X) pour \quad 1 \leq j \leq m, j \neq i \tag{2.2}$$

Ainsi, nous maximisons P(C_i|X). La classe C_i pour laquelle P(C_i|X) est maximisée et appelée hypothèse maximum posteriori. Selon le théorème de Bayes (Équation 2.1) nous obtiendrons :

$$P(C_i|X) = \frac{P(X|C_i)\,P(C_i)}{P(X)} \tag{2.3}$$

c) Comme P(X) est constant pour toutes les classes, seule $P(X|C_i)P(C_i)$ doit être maximisée. Si les probabilités à priori de la classe ne sont pas connues, alors nous supposerons généralement que les classes sont équiprobables, c'est-à-dire que P (C_1) = $P(C_2)$ = ... = P (C_m), et nous maximisons P(X|C_i). Sinon, nous maximisons P(X|C_i) P (C_i).

d) Étant donné des ensembles de données comportant de nombreux attributs, il serait extrêmement coûteux de calculer P$(X|C_i)$. Pour réduire le calcul dans l'évaluation de P$(X|C_i)$, l'hypothèse naïve d'indépendance conditionnelle de classe sera prise en considération. Cela suppose que les valeurs des attributs soient conditionnellement indépendantes les unes des autres. De ce fait, nous aurons :

$$P(X|C_i) = \prod_{k=1}^{n} P(X_k|C_i) = P(x_1|C_i)P(x_2|C_i) \ldots\ldots P(x_n|C_i) \tag{2.4}$$

Nous pourrons facilement estimer les probabilités $P(x_1|C_i)P(x_2|C_i) \ldots\ldots P(x_n|C_i)$à partir de l'ensemble d'entrainement. Rappelons qu'ici $x_k$désigne la valeur de l'attribut $A_k$pour le tuple X.

e) Pour prédire la classe de X, $P(X|C_i)P(C_i)$est évaluée pour chaque classe C_i. Le classifieur prédit que la classe du tuple X est la classe C_i si et seulement si :

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ Pour } 1 \leq j \leq m, j \neq i \tag{2.5}$$

En d'autres termes, l'étiquette de classe prédite est la classe Ci pour laquelle la valeur de $P(X|C_i)P(C_i)$ est la plus élevée.

2.2.1.2 Machine à vecteurs de support (SVM) :

Dans un cas simple, il s'agit d'un tracé de caractéristiques (variables) dont les coordonnées sont appelées « vecteurs de support ». Dans le cas d'un problème à deux classes, le SVM sépare les deux groupes de points en définissant une ligne séparatrice à marge maximale, c'est-à-dire avec une distance maximale entre la ligne et les points les plus proches. La configuration du SVM définit le type de ligne - linéaire, polynôme, logarithmique.[W6]

Nous avons deux types de séparation :

- Séparation linéaire : séparé les données par un ensemble de données faciles à trouver.

- Séparation non linéaire : séparé les données par des fonctions multi-classe.

Exemple 1 : un problème de discrimination à deux classes, avec un séparateur linéaire la droite d'équation **y = x**.

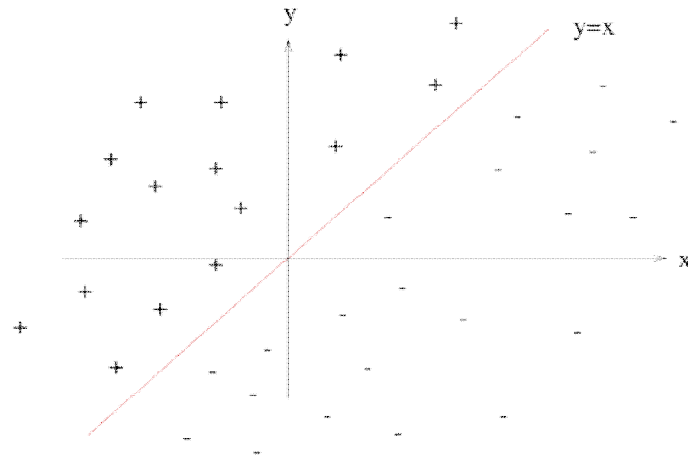


Figure 2.1 : Problème linéairement séparable.

Exemple 2 : problème de discrimination à deux classes, avec un séparateur non-linéaire

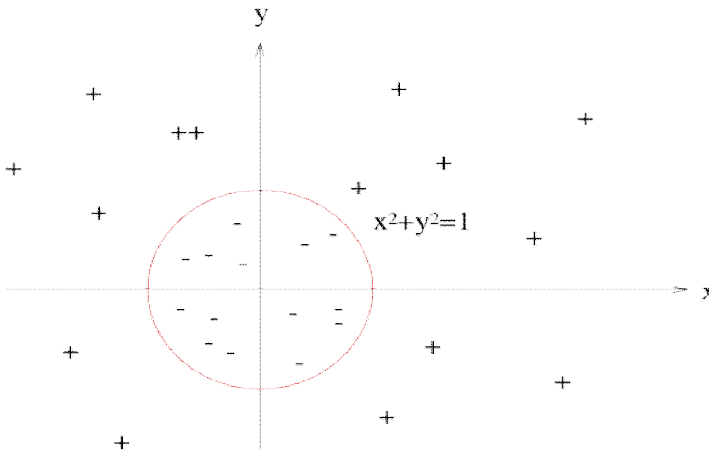


Figure 2.2 : Problèmenon linéairement séparable.

2.2.1.3 Réseau neuronal :

À l'inverse des algorithmes de déduction, ces derniers sont des algorithmes de type induction, c'est-à-dire que par le biais d'observations limitées, ils essayent de tirer des généralisations plausibles. C'est un système basé sur l'expérience qui se constitue une mémoire lors de sa phase d'apprentissage (qui peut être aussi non supervisée), appelée entrainement[40].

En général, un réseau de neurones est composé d'une succession de couches (entrée, caché et sortie) connectés entre eux dont chacune prend ses entrées sur les sorties de précédente, les neurones d'entrées possèdent un poids synaptique, tel que nous avons les neurones synchrones : tous les neurones calculent leur sortie simultanément, les neurones rétroaction : la sortie d'un neurone est raccordée à son entrée.

Afin de produire la sortie, nous calculons au début l'activation des couches cachées, par le produit des poids synaptique et les entrées, ensuite avec la même méthode nous activons les couches de sortie, la fonction d'activation la plus utilisée est la fonction sigmoïde, puis le dernier calcule est l'erreur de la sortie, si la sortie désirée est vraie nous arrêtons le calcule sinon nous calculons la rétropropagation avec la mise à jour des poids.

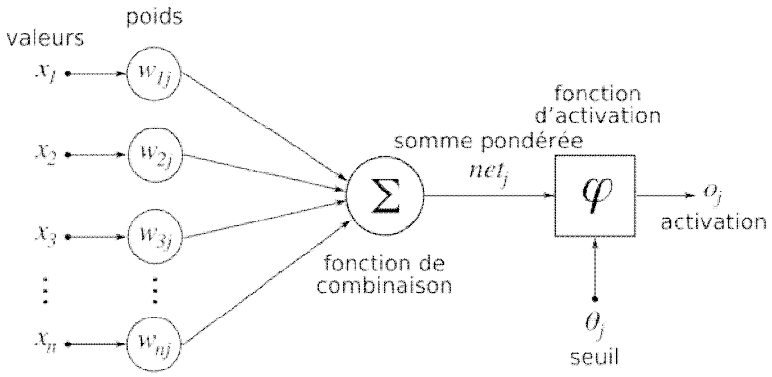


Figure 2.3 : Structure d'un réseau de neurone1.

- **Les formules pour faire ces calculs :**
- Activation de couches cachées et sorties :$\sigma_i = \sum w_i * x_i + b_i$, sigmoïde=$y_i = \frac{1}{1+e^{-\tau\sigma_i}}$
- Erreur de sortie : $\delta_j = (U_j - y_i)y_i(1 - y_i)$
- Mise à jour des poids : $\Delta w(i,j) = \alpha\delta_j y_i$, new (i,j)=$\Delta w(i,j) + oldw(i,j)$
- Bais : $\Delta b_j = \varphi * \delta_j * U_j$, $new\ b_j = old b_j + \Delta b_j$
- Erreur de caché : $\delta_j = y_j * (1 - y_j) * \sum(w_{j,k} * \delta_k)$

2.2.1.4 Arbre de décision :

Un arbre de décision est un enchaînement hiérarchique de règles logiques qui permettent de diviser la base d'exemples en sous-groupes, en fonction de la valeur des variables d'entrées. L'arbre est construit en recherchant à chaque niveau le paramètre le plus discriminant pour classifier un exemple [41]. Différents algorithmes peuvent être utilisés pour développer l'arborescence. Les arbres de décision suscitent un large engouement en fouille de données, car ils sont simples et rapides tout en restituant de manière compréhensible les relations existantes entre les variables d'entrée et le phénomène à modéliser.

Exemple : Un ensemble d'individus décrits par plusieurs variables catégorielles (Toux, Fièvre, Poids, Douleur).

	Toux	Fièvre	Poids	Douleurs	Symptôme
Marie	Non	Oui	Normal	Gorge	Rhume
Fred	Non	Oui	Normal	Abdomen	Appendicite
Julie	Oui	Oui	Maigre	Aucune	Rhume
Elvis	Oui	Non	Obèse	Poitrine	Infarctus

[1]Récupéré de https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels

Tableau 2.1 : Ensemble d'apprentissage sur les symptômes

En bas on voit l'arbre de décision obtenu. La variable cible se retrouve dans les feuilles

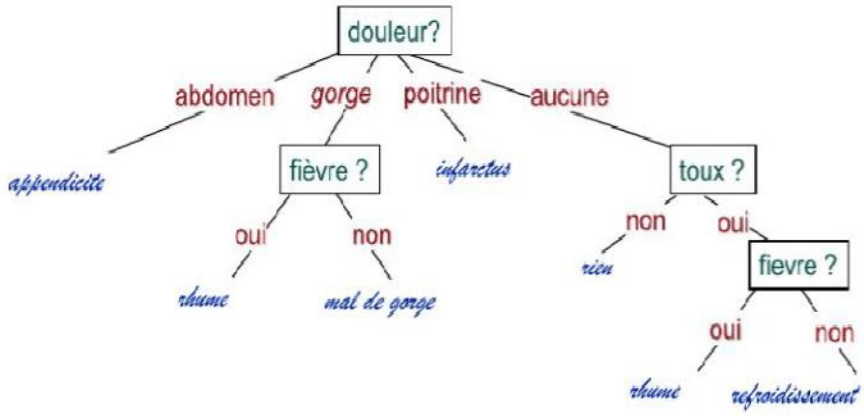


Figure 2.4 : Illustration d'un arbre de décision [W5]

Notons que la construction de l'arbre suit certaines directives dans le choix de l'attribut à segmenter en se basant sur certaines règles. Il existe plusieurs algorithmes de construction d'arbre, les plus populaires sont ID3, C4.5.

2.2.2 Apprentissage non supervisé :

Contrairement à l'apprentissage supervisé, le non supervisé traite le cas où on dispose seulement des entrées {X} sans avoir au préalable les sorties.

L'apprentissage non supervisé ou le « clustering » vise à construire des groupes (clusters) d'objets similaires à partir d'un ensemble hétérogène d'objets [35], Chaque cluster issu de ce processus doit vérifier les deux propriétés suivantes :

- La cohésion interne (les objets appartenant à ce cluster sont les plus similaires possible).
- L'isolation externe (les objets appartenant aux autres clusters sont les plus distincts possible).

En effet, le processus de « clustering » repose sur une mesure précise de la similarité des objets que l'on veut regrouper. Cette mesure est appelée distance ou métrique. On distingue plusieurs algorithmes de « clustering », comme [42]:

2.2.2.1 K-moyennes (K-Means) :

K-Meansest probablement l'algorithme le plus connu et le plus utilisé pour le clustering. Il s'ensuit une procédure simple consistant à classer un ensemble de données donné dans un certain nombre de groupes, définies par la lettre "K", qui est fixée au préalable. Les groupes sont ensuite positionnés en tant que points et tous les points de données sont associés au

groupe le plus proche en termes de distance par rapport au centroïde le plus proche, puis, le processus recommence à l'aide des nouveaux ajustements jusqu'à l'obtention du résultat souhaité[43].

Fonctionnement de l'algorithme K-Means:

L'Algorithme K-Means
DEBUT **Entrée :** D : Ensemble de données, k : nombre de clusters **1.** Choisir arbitrairement k centres initiaux $C = \{c1, \ldots, ck\}$ depuis l'ensemble de données. **2.** Pour chaque i ∈ {1, . . . k}, définissez le groupe C_i comme étant l'ensemble de points dans X qui sont plus proches de c_i qu'ils sont à Ci pour tous $j \neq i$. **3.** Pour chaque i ∈ {1, . . . k}, fixez c_i comme étant le centre de masse de tous les points :C_i : $C_i = \frac{1}{|Ci|}\sum_{x \epsilon Ci} x$ **4.** Répétez les étapes 2 et 3 jusqu'à ce que C_i ne change plus. **FIN.**

2.2.2.2 FuzzyC-Means (FCM) :

Est une méthode de clustering qui permet à un objet de données d'appartenir à deux ou plusieurs clusters. Cette méthode dérivée de l'algorithme C-Means[44], identique à l'algorithme K-Means décrit précédemment, On peut résumer la différence entre fuzzyC-Means et K-Means dans la fonction d'appartenance d'un nuage de points dans deux clusters dans l'exemple suivant :

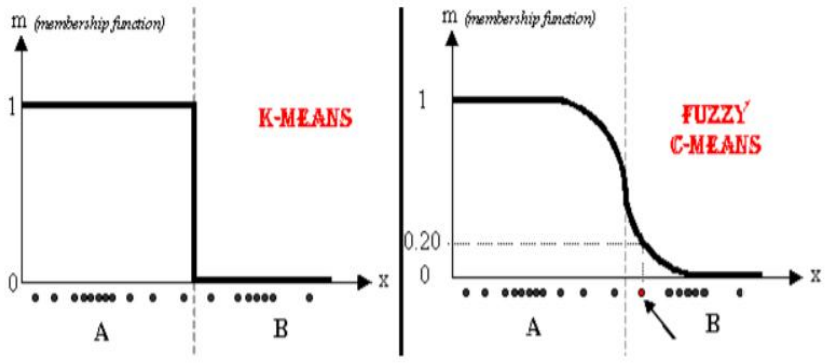


Figure 2.5 : Fonction d'appartenance dans K-Means/Fuzzy C-means

Dans le cas de K-Means un objet ne peut pas appartenir dans deux clusters Simultanément, ce qui explique la Discrimination binaire entre les clusters mais en FCM il est possible qu'un objet appartient à deux ou plusieurs clusters selon différents pourcentages c.-à-d. que les données sont liées à chaque groupe par le biais d'une fonction d'appartenance, ce qui représente le comportement floue de cet algorithme.

Fonctionnement de l'algorithme FuzzyC-Means :

Algorithme FuzzyC-Means
DEBUT
1. Initialiser U = $[u_{ij}]$ matrice $u_{(0)}$
2. A la k-étape : calculer les centres $c_{(k)}$= $[c_j]$ avec $U_{(k)}$
$$C_J = \sum_{i=1}^{n} U_{ij}^{m} . X_i \sum_{i=1}^{N} U_{ij}^{m}$$
3. Mise à jour de $U_{(k)}$, $U_{(k+1)} U_i^{j=1} C_{k=1} \|\|x_i - c_k\|\| 2m - 1$
4. Si $\left\|\left\|U_{(k+1)} - U_{(k)}\right\|\right\| < \varepsilon \quad (0 < \varepsilon < 1)$, Alors STOP, Sinon le retour à l'étape 2.
FIN.

Ce type d'algorithme est fort utilisé en traitement d'images [45, 46, 47] afin d'identifier des zones similaires (contours, Coins, région homogènes,. . .).

2.2.2.3 Regroupement hiérarchique :

Dans le clustering hiérarchique il n'est pas nécessaire de spécifier au préalable le nombre de cluster à générer car les données ne sont pas partitionnées dans un cluster particulier en une seule étape mais en une série de partitionnements pouvant aller d'un seul cluster contenant tous les objets à N clusters contenant chacun un seul objet. Une méthode hiérarchique peut être classée comme étant une méthode agglomérative(également appelée approche ascendante), qui commence par chaque objet formant un groupe séparé puis fusionne successivement les objets ou les groupes proches les uns aux autres jusqu'à ce que tous les groupes soient fusionnés en un seul. Ou bien comme étant une méthode par division (également appelée approche descendante), commence par tous les objets dans un même cluster et à chaque itération, un cluster est divisé en clusters plus petits, jusqu'à ce que chaque objet soit finalement dans un cluster, ou une condition de terminaison est atteinte [48].

Algorithme de l'approche agglomérative (CHA) :

Algorithme(CHA)
1. Initialisation :
- Chaque individu est placé dans son propre cluster,
- Calcul de la matrice de ressemblance M entre chaque couple de clusters (ici les points)
2. Répéter :
- Sélection dans M des deux clusters les plus proches $C_i$et C_j
- Fusion de $C_i$et $C_j$par un clusterC_G plus général
- Mise à jour de M en calculant la ressemblance entre C_G et les clusters existants

Jusqu'à fusionner les 2 derniers clusters **FIN.**

Dans la figure suivante, on représente une illustration du principe de CHA et la hiérarchie finale obtenue où les liens hiérarchiques apparaissent clairement :

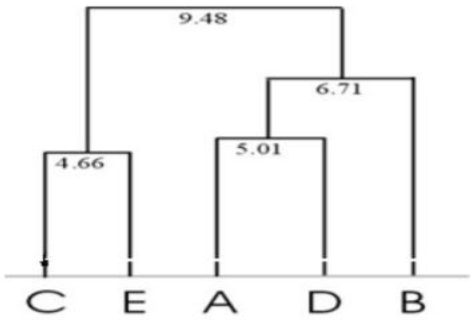


Figure 2.6 : Dendrogramme[2] **de la hiérarchie H de la suite de partitions d'un ensemble {A, B, C, D, E}**

Algorithmede l'approche par division :

Algorithme
DEBUT **1.** Mettre tous les objets dans un seul cluster ; **2.** Répéter jusqu'à atteindre la condition d'arrêt : - Choisir un cluster à diviser ; - Remplacer le cluster choisi par le sous cluster obtenu. **FIN**

2.3 L'apprentissage en profondeur (Deep Learning) :

Le Deep Learning est un nouveau domaine de recherche de la machine Learning (ML), qui a été introduit dans le but de rapprocher le ML de son objectif principal à savoir : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et du fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données [W3].

[2]Un dendrogramme = la représentation graphique d'une classification ascendante hiérarchique sous forme d'un arbre binaire.

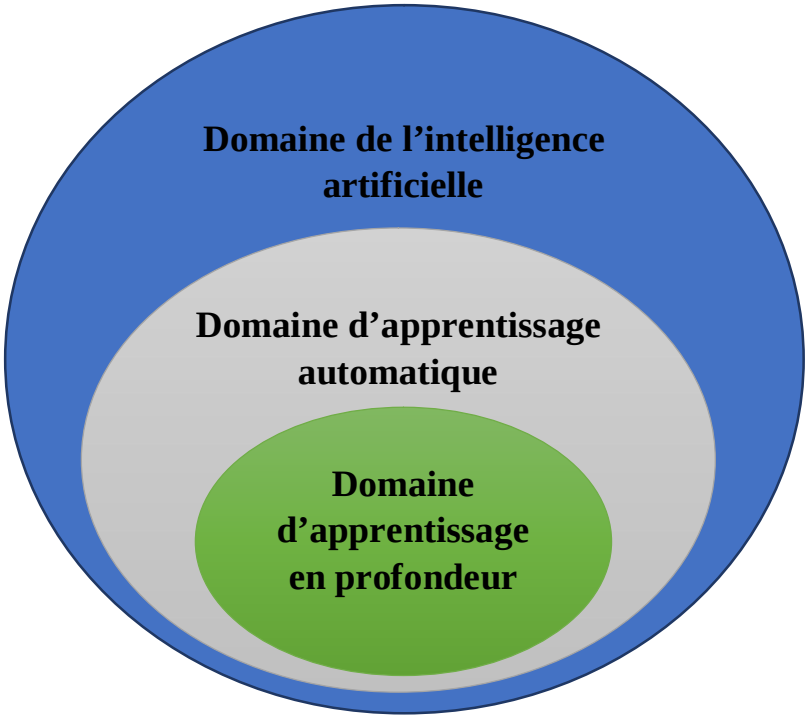


Figure 2.7 : Relation entre l'IA, le ML et le DL

2.3.1 Définition :

L'apprentissage en profondeur est un ensemble d'algorithmes d'apprentissage automatique qui tentent d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minime [49,50, 51, 52, 53, 54, 55].

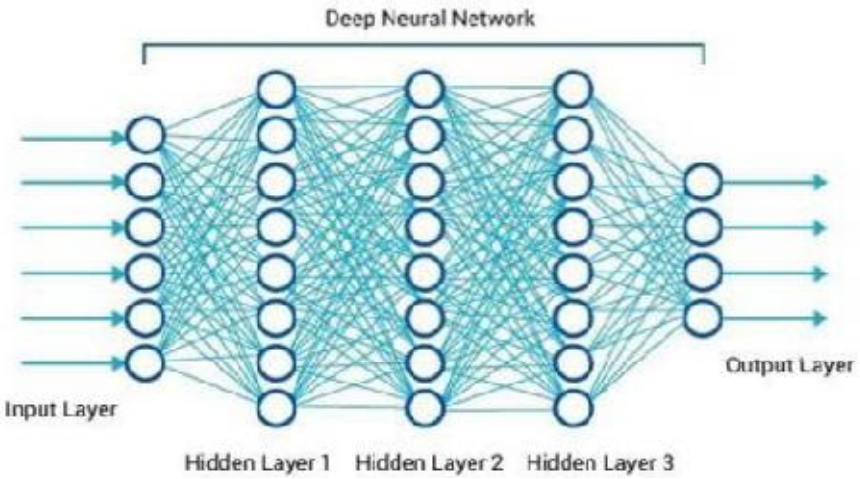


Figure 2.8: Schéma illustratif de DL avec plusieurs couches.

2.3.2 Historique :

L'idée du Deep Learning n'est pas une idée récente, mais elle date en réalité des années 1980, plus particulièrement suite aux travaux de réseaux de neurones multicouches et aux travaux de certains pionniers du machine Learning et du Deep Learning comme le français Yann Le Cun. En collaboration avec deux autres informaticiens, Kunihiko Fukushima et Geoffrey Hinton, ils mettent au point un type d'algorithme particulier appelé « Convolutional neural network » [W4].

Bien que cette approche donne des résultats, ses progrès et son évolution sont limités par les progrès technologiques en matière de micro-processeurs, de puissance de calculs, et du manque d'accessibilités à des données afin de pouvoir entraîner les neurones. Cependant certains chercheurs ont continué à travailler sur ce modèle pendant environ deux décennies et, avec l'aide des évolutions en matière de technologies, mais surtout avec la disponibilité toujours plus grande de données, ont pu améliorer cette technique [W4].

La seconde révolution des réseaux de neurone est due à l'apparition de nouveaux processeurs, et en particulier de processeurs "massivement parallèles". Ceux-ci ont été dans un premier temps été utilisés dans les circuits graphiques de nos micro-ordinateurs, calculateurs graphiques GPU et de circuits programmables FPGA, qui a rendu possible l'explosion récente du Deep Learning [W4].

2.3.3 Pour quoi le choix du Deep Learning :

Tout d'abord les différents algorithmes du Deep Learning ne sont apparus qu'à l'échec de l'apprentissage automatique tentant de résoudre une grande variété de problèmes de l'intelligence artificielle (IA) [56]:

- Afin d'améliorer le développement des algorithmes traditionnels dans de telles tâches de l'IA.
- De développer une grande quantité de données telle que les big data.
- De s'adapter à n'importe quel type de problème.
- D'extraire les caractéristiques de façon automatique.



Flux d'apprentissage automatique



Flux d'apprentissage profond

Figure 2.9 : Comparaison entre la machine Learning et le Deep Learning.

2.3.4 Les différentes Architectures du Deep Learning :

Bien qu'il existe un grand nombre de variantes d'architectures profondes. Il n'est pas toujours possible de comparer les performances de toutes les architectures, car elles ne sont pas toutes évaluées sur les mêmes ensembles de données. Le Deep Learning est un domaine à croissance rapide, et de nouvelles architectures, variantes ou algorithmes apparaissent toutes les semaines.

a. Les réseaux de neurones convolutifs :

Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Qui se sont avérés très efficaces dans des domaines tels que la reconnaissance et la classification d'images et vidéos. CNN a réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures [57]. Récemment, les CNN ont été efficaces dans plusieurs tâches de traitement du langage naturel (telles que la classification des phrases) [58, 59, 60].

Dans le ML, un réseau convolutif est un type de réseau de neurones feed-forward, il a été inspiré par des processus biologiques [61]. Il existe quatre (4) principales opérations illustrées dans le CNN à savoir :

- La couche convolution
- La couche RectifiedLinear Unit
- La couche Pooling
- La couche entièrement connectée

b. Réseau de neurones récurrents :

L'idée derrière les RNN est d'utiliser des informations séquentielles. Dans un réseau neuronal traditionnel, nous supposons que toutes les entrées (et les sorties) sont indépendantes les unes des autres. Mais pour de nombreuses tâches, c'est une très mauvaise idée. Si on veut prédire le prochain mot dans une phrase, il faut connaître les mots qui sont venus avant. Les RNN sont appelés récurrents, car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents[62].

Une autre façon de penserles RNN est qu'ils ont une « mémoire » qui capture l'information sur ce qui a été calculé jusqu'ici. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière [63, 64, 62, 65]. Il est utilisé pour :

- La modélisation du langage et génération de texte
- La traduction automatique
- La reconnaissance vocale
- Et la description des images

c. Modèle génératif :

Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l'entrée, tant disque le modèle génératif décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes [66].

Cependant les modèles génératifs sont capables de bien plus que la simple classification comme par exemple générer de nouvelles observations.

Voici quelques exemples de modèle génératif :

- Boltzmann Machines [67]
- Restricted Boltzmann Machines[68, 69]
- DeepBelief Networks [70]
- Deep Boltzmann Machines [71, [72]
- Generative Adversarial Networks, Generative Stochastic Networks [73]
- Adversarialautoencoders[74]

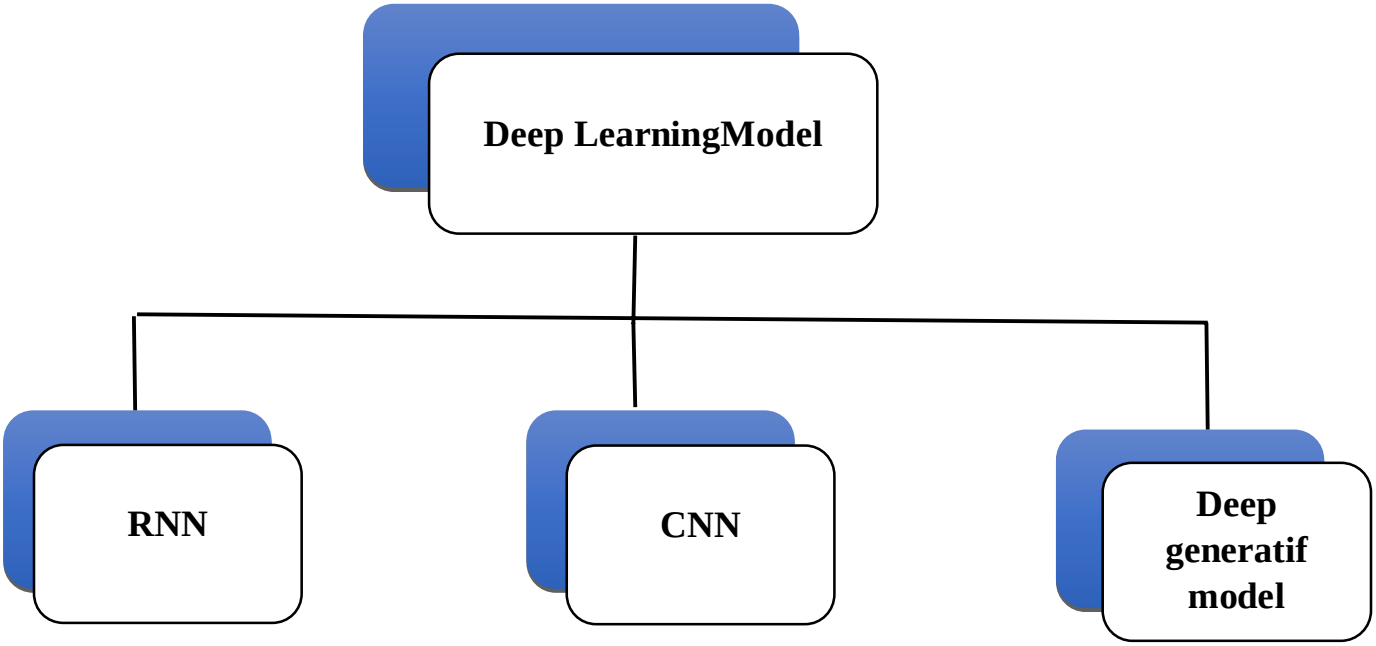


Figure 2.10 : Différents modèles du Deep Learning

2.4 L'apprentissage profond pour la recommandation :

Voici les quatre points forts clés des systèmes de recommandations axés sur l'apprentissage profond par rapport à ceux des approches traditionnelles de filtrage basée sur le contenu et en collaboration [75] :

- **L'apprentissage profond peut modéliser les interactions non linéaires dans les données avec des activations non linéaires telles que ReLU, Sigmoid, Tang, ...** Cette propriété permet de capturer les modèles d'interaction utilisateur-élément complexes. Les méthodes conventionnelles telles que les machines de factorisation matricielle et de factorisation sont essentiellement des modèles linéaires [76]. Cette hypothèse linéaire, agissant comme base de nombreuses recommandations traditionnelles, est trop simplifiée et limitera grandement leur expressivité de modélisation. Il est bien établi que les réseaux neuronaux sont capables d'approximer n'importe quelle fonction continue avec une précision arbitraire en variant les choix d'activation et les combinaisons [77]. Cette propriété permet de traiter des modèles d'interaction complexes et de refléter précisément la préférence de l'utilisateur.

- **L'apprentissage profond peut apprendre efficacement les facteurs explicatifs sous-jacents et les représentations utiles à partir des données d'entrée.** En général, une grande quantité d'informations descriptives sur les éléments et les utilisateurs est disponible dans des applications réelles. L'utilisation de ces informations fournit un moyen de faire progresser notre compréhension des articles et des utilisateurs, ce qui

entraîne une meilleure recommandation. En tant que tel, c'est un choix naturel d'appliquer des réseaux neuronaux profonds à l'apprentissage de la représentation dans les modèles de recommandation. Les avantages de l'utilisation de réseaux neuronaux profonds pour aider l'apprentissage de la représentation sont doubles : (1) il réduit les efforts dans la conception de fonctionnalités artisanales ; et (2) il permet aux modèles de recommandation d'inclure des informations de contenu hétérogènes telles que le texte, les images, l'audio et même la vidéo. [75]

- **L'apprentissage profond est puissant pour les tâches de modélisation séquentielle**. Dans des tâches telles que la traduction automatique, la compréhension du langage naturel, la reconnaissance vocale, ..., les RNN et les CNN jouent un rôle essentiel. Ils sont largement applicables et flexibles dans l'exploitation minière de la structure séquentielle dans les données. La modélisation des signaux séquentiels est un sujet important pour l'extraction de la dynamique temporelle du comportement de l'utilisateur et de l'évolution des éléments. Par exemple, les prévisions d'article suivant/panier et les recommandations basées sur la session sont des applications typiques. En tant que tel, les réseaux neuronaux profonds deviennent un ajustement parfait pour cette tâche d'extraction de modèle séquentiel. [75]
- **L'apprentissage profond possède une grande flexibilité**. Il existe de nombreux cadres populaires d'apprentissage profond de nos jours, y compris TensorFlow, Keras, Caffe, MXnet, DeepLearning4j, PyTorch, Theano, ... Ces outils sont développés de manière modulaire et disposent d'un soutien communautaire/professionnel actif. La bonne modularisation rend le développement et l'ingénierie beaucoup plus efficaces. Par exemple, il est facile de combiner différentes structures neuronales pour formuler de puissants modèles hybrides ou remplacer un module par d'autres. Ainsi, nous pourrions facilement construire des modèles de recommandation hybrides et composites pour capturer simultanément différentes caractéristiques et facteurs [75].

Pour fournir une vue d'ensemble de ce domaine, on peut classer les modèles existants en fonction des types de techniques d'apprentissage profond utilisées [75] :

2.4.1 Recommandation basée sur le perceptron multicouche :

MLP est un réseau neuronal de flux avant avec plusieurs couches cachées entre la couche d'entrée et la couche de sortie. MLP est interprété comme une couche empilée de transformations non linéaires, apprenant des représentations hiérarchiques de fonctionnalités. Il s'agit d'un réseau concis mais efficace qui peut rapprocher n'importe quelle fonction mesurable à n'importe quel degré désiré de précision. En tant que tel, il est la base de nombreuses approches avancées et est largement utilisé dans de nombreux domaines [78].

MLP peut ajouter la transformation non linéaire aux approches existantes du système de recommandation et les interpréter en extensions neuronales.

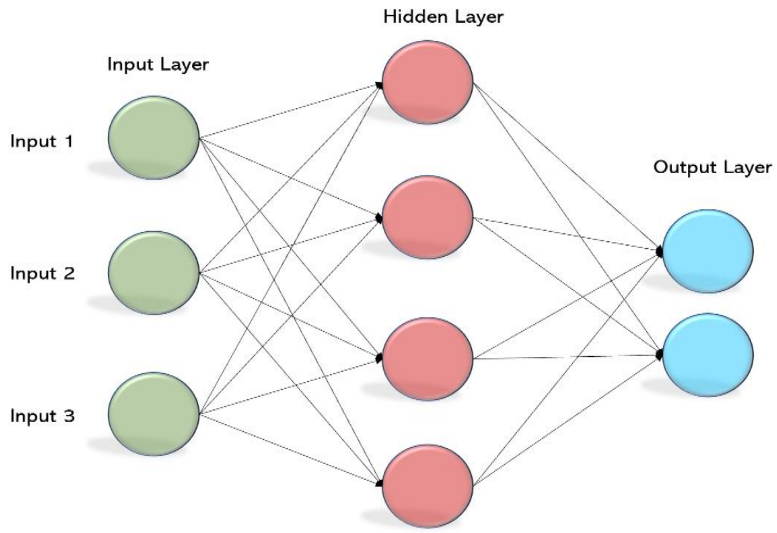


Figure 2.11: Multi-Layer Perceptron

- Une recommandation peut être considérée comme une interaction bidirectionnelle entre les préférences des utilisateurs et les fonctionnalités des éléments. Par exemple, la factorisation de la matrice décompose la matrice d'évaluation en facteurs latents utilisateur/élément faiblement dimensionnel. Neural Collaborative Filtering est un travail représentatif qui construit un double réseau neuronal pour modéliser cette interaction bidirectionnelle entre les utilisateurs et les éléments [76].

  Les Règles Mathématique de travail [76] :

  a_2 = Fonction d'activation ,

  $W_x = Matricedespoids$,

  b_x Vecteur de Bias

  $Z1 = \phi(pu, qi) = [pu, qi]$: Concaténation des deux vecteurs

  Dans ces couches on va multiplier chaque poids avec chaque donnée de couche précédente et on ajoute le bais, le résultat sera défini par la fonction d'activation Relu (= max (0, x)).

$$\phi_2(Z1) = a_2(W_2^T * z_1 + b_2)$$

$$\ldots\ldots\ldots$$

$$\phi_L(Z_{L-1}) = a_2(W_L^T * z_{L-1} + b_L)$$

$$\hat{y}_{ui} = \sigma(h^T \phi_L(z_{L-1}))$$

$$h^T = \text{poids de la couche de sortie}$$

- Deep Factorisation Machine est un modèle de bout en bout qui intègre parfaitement une machine de factorisation et un MLP. Il peut modéliser les interactions de fonction de haut niveau via un réseau neuronal profond et des interactions de faible ordre avec les machines de factorisation[79].

L'utilisation de MLP pour la représentation des fonctionnalités est très simple et très efficace, même si elle n'est peut-être pas aussi expressive que l'auto-encodeur, les CNN et les RNN.

- Wide and Deep Learning est un modèle agréable qui peut résoudre à la fois les problèmes de régression et de classification qui a été initialement introduit pour la recommandation de l'application dans Google Play. Le composant d'apprentissage large est un perceptron à couche unique qui peut également être considéré comme un modèle linéaire généralisé. Le volet apprentissage profond est un MLP. La combinaison de ces deux techniques d'apprentissage permet au recommandeur de capturer à la fois la mémorisation et la généralisation[80].

- Deep Neural Networks for YouTube Recommandations divise la tâche de recommandation en deux étapes : la génération de candidats et le classement des candidats. Le réseau de génération de candidats récupère un sous-ensemble de tout corpus vidéo. Le réseau de classement génère une liste Top-N basée sur les scores des voisins les plus proches des candidats[81].

- Collaborative Matrice Learning remplace le produit point de la factorisation de la matrice esprit Distance euclidien parce que le produit point ne satisfait pas l'inégalité triangle de la fonction de distance. Les intégrations de l'utilisateur et de l'article sont apprises en maximisant la distance entre les utilisateurs et leurs éléments détestés et en minimisant cela entre les utilisateurs et leurs articles préférés[82].

2.4.2 Recommandation basée sur Auto-Encoder :

L'Auto-Encoder(AE) est un modèle non supervisé qui tente de reconstituer ses données d'entrée dans la couche de sortie. En général, la couche de goulot d'étranglement est utilisée comme une représentation de caractéristiques saillantes des données d'entrée. Presque toutes ses variantes (dénotation de l'AE (DAE), AE variationnel (VAE), AE conjonctif et AE marginalisée) peuvent être appliquées à la tâche de recommandation[83, 84].

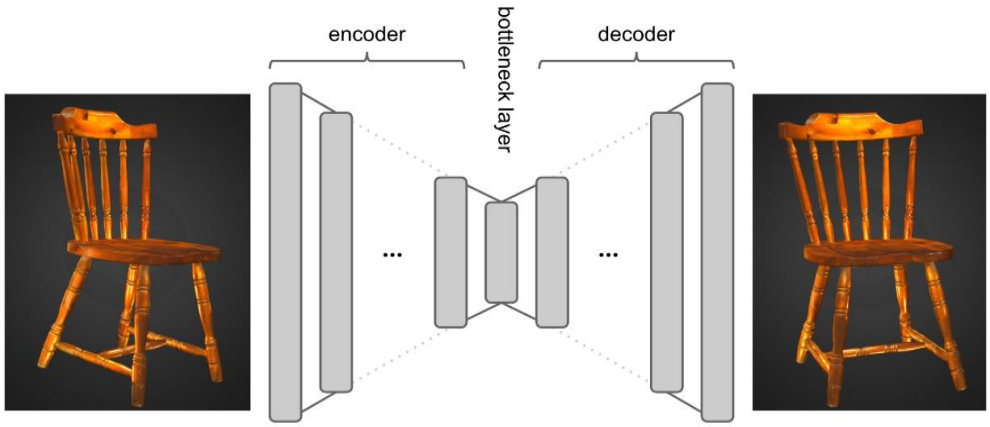


Figure 2.12 : Auto-Encoder

AE peut être utilisé pour apprendre les représentations de fonctionnalités à faible dimension à la couche de goulot d'étranglement.

- Collaborative Deep Learning est un modèle bayésien hiérarchique qui intègre l'auto-encodeur denoising empilé (SDAE) dans la factorisation de la matrice probabiliste (PMF). Pour combiner de façon transparente le modèle d'apprentissage profond et de recommandation, le document propose un cadre général bayésien d'apprentissage profond composé de deux composantes étroitement articulées : un composant de perception (SDAE) et un composant spécifique aux tâches (PMF). Cela permet au modèle d'équilibrer les influences de l'information latérale et de l'historique des interactions[85].
- Collaborative DeepRanking est conçu spécifiquement dans un cadre en couple pour la recommandation Top-N. Le document montre que le modèle en couple est plus adapté à la génération de listes de classement[86].
- Deep Collaborative Filtering est un cadre général pour unifier les approches d'apprentissage profond avec un modèle de filtrage collaboratif. Le cadre facilite l'utilisation de techniques d'apprentissage des fonctionnalités profondes pour construire des modèles hybrides collaboratifs [87].

AE peut être utilisé pour remplir les blancs de la matrice d'interaction utilisateur-élément directement dans la couche de reconstruction.

- AutoRec prend les vecteurs partiels utilisateur/élément comme entrée et vise à les reconstruire dans la couche de sortie. [88]
- Collaborative Denoising Auto-encoder (CDAE) est principalement utilisé pour la prédiction de classement. L'entrée de CDAE est une rétroaction implicite partiellement observée par l'utilisateur, qui peut être considérée comme un vecteur de préférence qui reflète les intérêts d'un utilisateur pour les éléments. L'auteur propose également une technique d'échantillonnage négatif pour échantillonner un petit sous-ensemble de l'ensemble négatif (éléments avec lesquels l'utilisateur n'a pas interagi), ce qui réduit considérablement la complexité du temps sans dégrader la qualité de classement[75].
- Multi-VAE et Multi-DAE proposent une variante d'un auto-encoder variationnel pour la recommandation avec des données implicites. L'article introduit une approche d'inférence bayésienne de principe pour l'estimation des paramètres et affiche des résultats favorables que les fonctions de probabilité couramment utilisées[89].

2.4.3 Recommandation basée sur les réseaux neuronauxconvolutionnels :

Le réseau neuronal convolutionnel (CNN) est essentiellement un réseau neuronal de flux avant avec des couches de convolution et des opérations de mise en commun. Il peut capturer

les caractéristiques mondiales et locales, améliorant ainsi de manière significative l'efficacité et la précision du modèle. Il est très puissant dans le traitement des données multimédias non structurées [75].

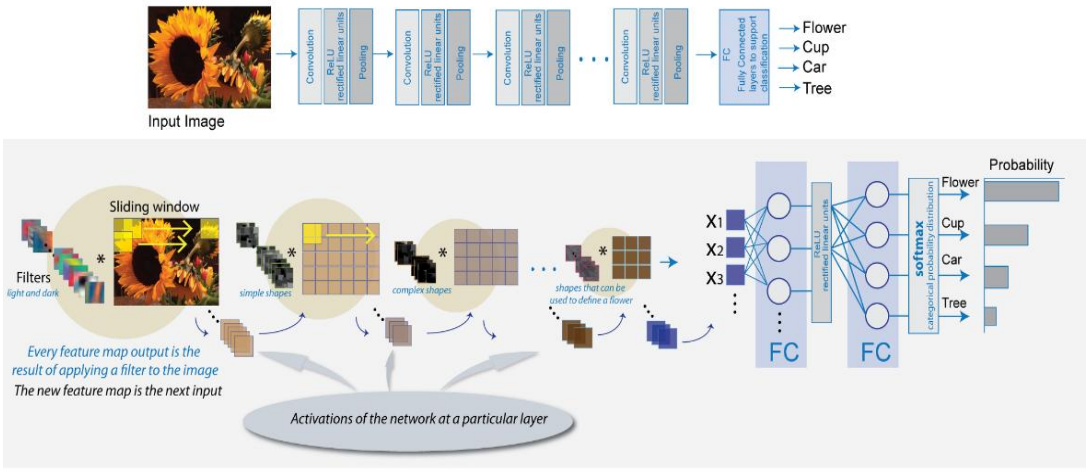


Figure 9.13 : Convolutional Neural Network

CNN peut être utilisé pour extraire des fonctionnalités à partir d'images.

- Filtrage collaboratif basé sur CNN : L'application directe des CNN au filtrage collaboratif de la vanille est également viable. Par exemple, He et al. [90] ont proposé d'utiliser les CNN pour améliorer NCF(Neural Collaborative Filtering) et ont présenté le ConvNCF. Il utilise le produit externe au lieu du produit point pour modéliser les modèles d'interaction utilisateur/élément. Les CNN sont appliqués sur le résultat du produit extérieur et pourraient capturer les corrélations de haut niveau entre les dimensions d'intégration. Tang et al. [91] ont présenté une recommandation séquentielle (avec identificateur d'utilisateur) avec les CNN, où deux CNN (hiérarchiques et verticales) sont utilisés pour utiliser les modèles séquentiels au niveau syndical et ignorer les comportements pour la recommandation consciente de la séquence .

- What Your Images Reveal étudie les influences des fonctionnalités visuelles sur la recommandation Point-of-Interest (POI) et propose un système de recommandation de POI amélioré en contenu visuel. Ce système adopte CNN pour extraire les fonctionnalités d'image, qui est construit sur probabilistic Matrix Factorization en explorant les interactions entre le contenu visuel et le facteur utilisateur/emplacement latent [92].

- Comparative Deep Learning of Hybrid Representations for Image Recommendations propose un modèle comparatif d'apprentissage profond avec les CNN pour la recommandation d'image. Le réseau se compose de deux CNN qui sont utilisés pour

l'apprentissage de la représentation d'image et d'un MLP pour la modélisation des préférences des utilisateurs[93].

CNN peut être utilisé pour extraire des fonctionnalités à partir de texte.

- DeepCoNN adopte deux CNN parallèles pour modéliser les comportements des utilisateurs et les propriétés des éléments à partir de textes d'examen[94]. Ce modèle atténue le problème de la sparsité et améliore l'interprétabilité du modèle en exploitant la riche représentation sémantique des textes de révision avec les CNN. Il utilise une technique d'incorporation de mots pour cartographier les textes d'examen dans un espace sémantique à faible dimension et conserver les informations sur les séquences de mots. Les représentations d'examen extraites passent ensuite par une couche convolutionnelle avec différents noyaux, une couche de mise en commun maximale et une couche entièrement connectée consécutivement [75].
- La technologie de recommandation automatique pour les ressources d'apprentissage avec le réseau neuronal convolutionnel établit un modèle de recommandation de ressources d'apprentissage en ligne, qui utilise les CNN pour extraire les éléments de l'information textuelle des ressources d'apprentissage telles que l'introduction et le contenu du matériel d'apprentissage [95].

CNN peut être utilisé pour extraire des fonctionnalités de l'audio et de la vidéo

- Deep Content-Based Music Recommendation utilise CNN pour extraire des fonctionnalités des signaux musicaux. Les grains convolutionnels et les calques de mise en commun permettent des opérations à plusieurs échelles de temps. Ce modèle basé sur le contenu peut atténuer le problème de démarrage à froid de la recommandation de musique [96].
- Collaborative Deep Metric Learning for Video Understanding extrait des fonctionnalités audio avec le modèle resNet basé sur CNN. La recommandation est effectuée dans le cadre d'apprentissage métrique collaboratif, semblable à la CML(collaborative metric learning) mentionnée précédemment [97].

2.4.4 Recommandation basée sur les réseaux neuronaux récurrents :

Le réseau neuronal récurrent (RNN) convient à la modélisation des données séquentielles. Il a des boucles et des souvenirs pour se souvenir des anciens calculs. Des variantes de RNNs, y compris le LSTM(Long Short Term Memory) et le GRU(Gated Recurrent Unit), sont déployées pour surmonter le problème de gradient de disparition [84, 75].

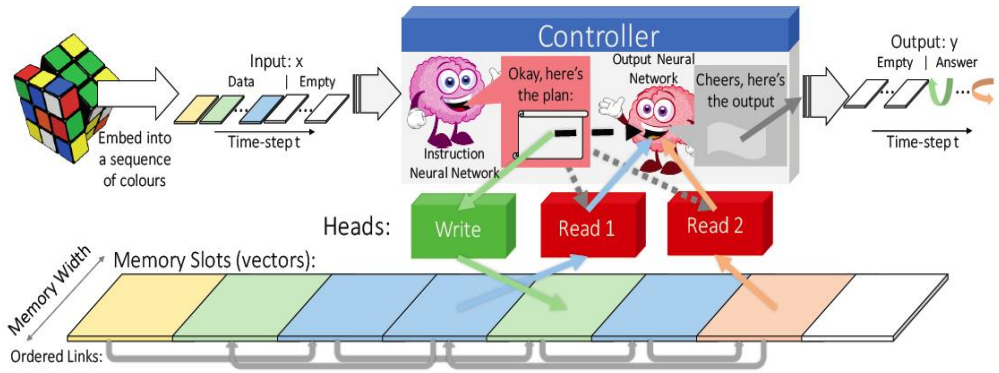


Figure 2.14 : Recurrent Neural Networks

RNN peut traiter la dynamique temporelle des interactions et des modèles séquentiels des comportements des utilisateurs dans les tâches de recommandation basées sur la session.

- GRU4Rec est un modèle de recommandation basé sur la session, où l'entrée est l'état réel d'une session avec un codage 1-of-N, où N est le nombre d'éléments. La coordonnée sera 1 si l'élément correspondant est actif dans cette session, sinon 0. La sortie est la probabilité d'être la suivante dans la session pour chaque élément [98].
- Recommandation personnelle utilisant Deep Recurrent Neural Networks dans NetEase est un modèle de recommandation basé sur la session pour un site Web de commerce électronique du monde réel. Il utilise les RNN de base pour prédire ce que l'utilisateur va acheter ensuite en fonction de l'historique des clics,Pour minimiser les coûts de calcul, il ne conserve qu'un nombre limité des derniers états tout en s'effondrant les anciens états dans un seul état historique. Cette méthode permet d'équilibrer le compromis entre les coûts de calcul et l'exactitude des prédictions [99].
- Recurrent Recommander Network est un modèle de recommandation non paramétrique construit sur des RNN. Il peut modéliser l'évolution saisonnière des éléments et les changements dans les préférences des utilisateurs au fil du temps. Il utilise deux réseaux LSTM comme bloc de construction pour modéliser les états utilisateur/élément dynamiques [100].

RNN est également un bon choix pour apprendre l'information latérale avec des modèles séquentiels.

- Les processus récurrents de fonctionnalité latente couvolution pour la recommandation de temps continu présentent un modèle latent co-évolutif pour capturer la nature co-évolution des caractéristiques latentes des utilisateurs et des éléments. Les interactions entre les utilisateurs et les éléments jouent un rôle important dans les changements dans les préférences des utilisateurs et l'état de l'élément. Pour modéliser les interactions historiques, l'auteur a proposé d'utiliser des RNN pour apprendre

automatiquement les représentations des influences de la dérive, de l'évolution et de la co-évolution des fonctionnalités de l'utilisateur et de l'élément[101].

- Embedding-based News Recommendation for Millions of Users propose d'utiliser des GRUs pour en apprendre davantage l'agrégation expressive pour l'historique de navigation des utilisateurs et de recommander des articles de nouvelles avec un modèle de facteur latent. Les résultats montrent une amélioration significative par rapport à l'approche traditionnelle fondée sur les mots. Le système a été entièrement déployé pour les services de production en ligne et desservant plus de 10 millions d'utilisateurs uniques chaque jour [102].

2.5 Conclusion :

Dans cette partie, nous avons présenté l'apprentissage automatique, à savoir : celui supervisé etcelui non supervisé ainsi que leurs algorithmes correspondants, tout en parlant sur l'apprentissage en profondeur ainsi que l'apprentissage profond pour la recommandation.

Partie 2 : Modélisation et expérimentation

Chapitre 3 : Approche proposée

Chapitre 4: Evaluation expérimentale de l'approche proposée

Chapitre3

Approche proposée

3.1 Introduction :

La prise en considération du contexte par les systèmes de recommandation permet une meilleure modélisation du comportement des utilisateurs, en fournissant des informations supplémentaires qui peuvent être exploitées afin d'effectuer de meilleures recommandations. Parmi les techniques utilisées, le modèle de filtrage collaboratif basé sur l'apprentissage en profondeur.

Ayant pour objectif d'apporter des améliorations à cette approche de recommandation, notre approche inclura plusieurs méthodes, à savoir : la méthode multi-layerperceptron et l'algorithmede clusteringK-Means.

Ce chapitre, est consacré à détailler cette approche.

3.2 Architecture générale de l'approche proposée :

Dans cette section, nous allons décrire l'architecture globale de notre système. Cette architecture est divisée en deux principales parties :

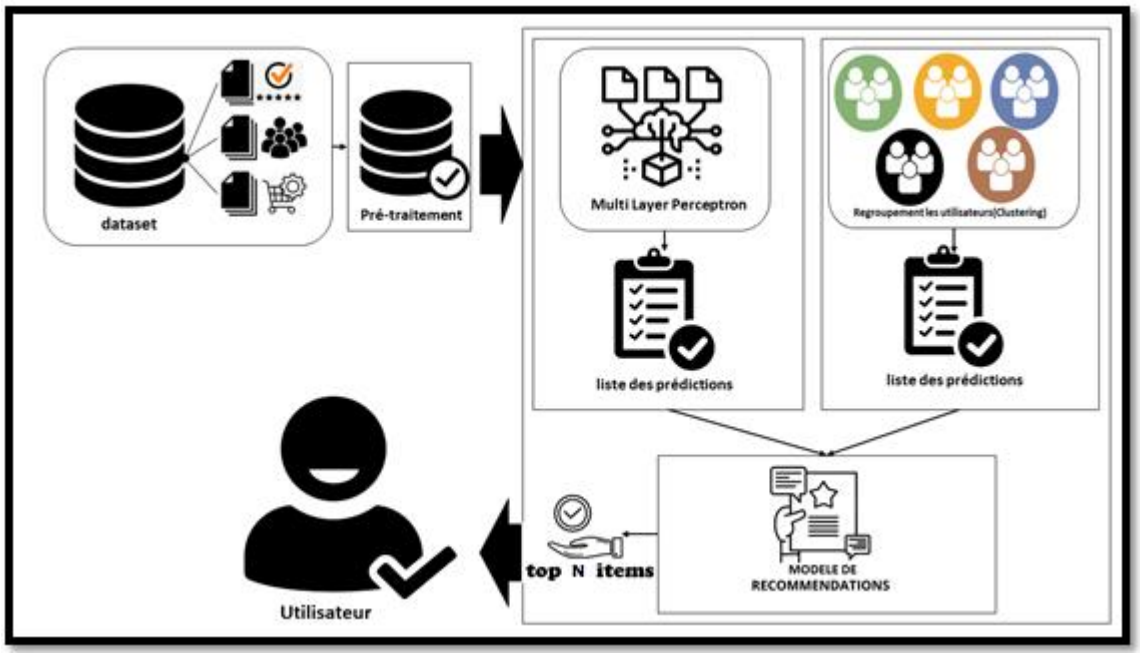


Figure 3.1: Architecture générale de l'approche proposée

Nous commençons tout d'abord par illustrer notre architecture du modèle que nous avons créé, nous allons expliquer tous ses étapes dans les prochaines lignes, nous avons trois étapes à suivre : prétraitement, apprentissage et recommandation. Nous dressons un tableau qui présente toutes les formes utilisées dans notre architecture.

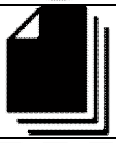
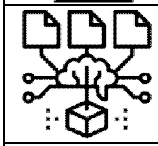
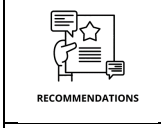
Schéma	Description
	Base de données
	Fichier de données
	Multi Layer Perceptron
	Regroupement des utilisateurs
	Liste des prédictions
	Modèle de recommandation
	Utilisateur
	Traitement

Tableau 3.1 : Une description des formes utilisées dans l'architecture.

3.3 Principe de fonctionnement :

Notre système de recommandation est un système qui est décomposé en trois principales parties : Pré-traitement**,** l'apprentissage et la recommandation.

3.3.1 Pré-traitement :

L'objectif de cette partie est l'analyse et la normalisation des données, elle consiste àsupprimer les données non exploitables et aussi éliminer le bruit.

Premièrement, il s'agit de traiter les données pour préparer une collection de données.

La deuxième étape consiste à extraire les attributs (features) essentiels et éliminer ceux non essentiels, Il est bien de noter que, parfois, il est nécessaire de séparer quelques attributs, pour

sélectionner uniquement l'essentiel (par exemple : nous avons un attribut composé de plusieurs informations et nous avons besoin d'une seule information pour notre système).

Ensuite, il faut convertir nos données aux valeurs numériques (valorisation en numérique).

Donc notre prétraitement sur la datasetMovieLens consiste a :

- transformer le sexe de l'utilisateur a des 0 si elle est une femme et 1 sinon.
- transformer les variables sexe et occupation a des variables qualitative.
- diviser la catégories des films a des colonnes ou chaque colonne est soit 1 ou 0.
- éliminer le titre du film de nos données.
- séparer la colonne des évaluations des films et les autres colonnespour l'entrainement de notre modèle MLP

Et en phase final, nous allant faire la jointure des fichiers de données que nous avons sélectionné en un seul fichier, dans le but de concevoir une nouvelle méthode qui prend en considération tous les attributs et non pas un seul attribut comme dans les systèmes de recommandation classiques.

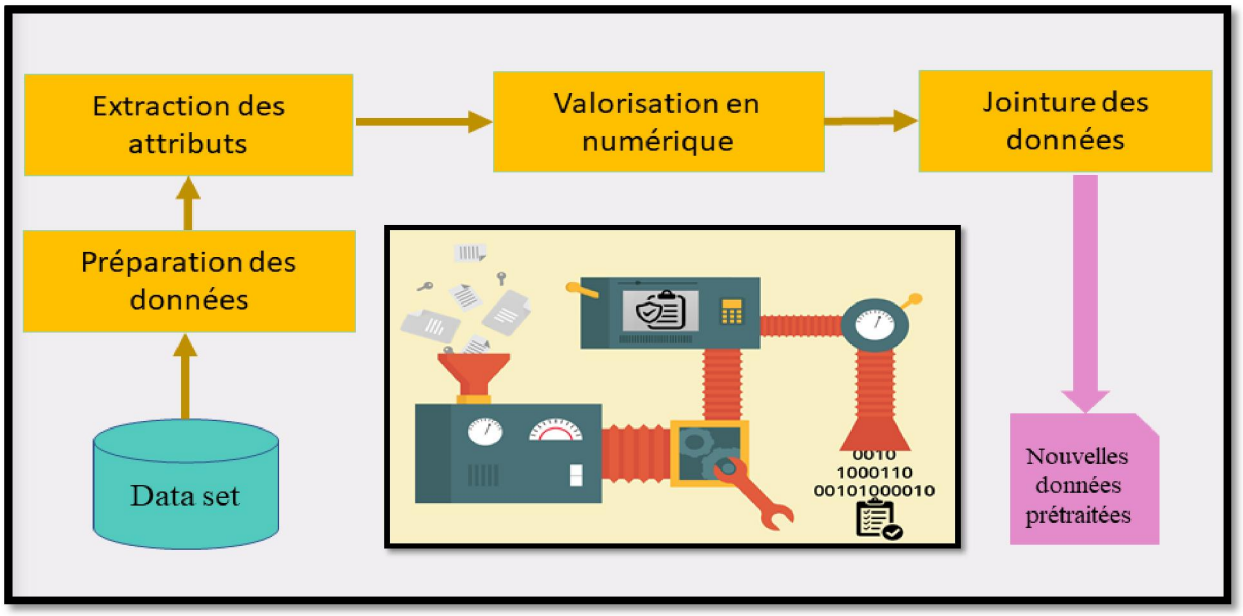


Figure 3.2 : Architecture de prétraitement.

3.3.2 L'apprentissage :

Comme mentionné précédemment, la deuxième partie de notre système de recommandation concerne l'apprentissage, elle est divisée en deux étapes : la construction de multi-layer perceptron, le regroupement des utilisateurs similaires dans le même cluster.

3.3.2.1 Multi-layer perceptron :

Cette partie est l'entrainement de notre modèle. Une fois le fichier de données est traité, nous allons utiliser un des algorithmes supervisés qui est le réseau de neurones, nous pouvons aussi

utiliser d'autres algorithmes, cependant, ce dernier donne de bons résultats mais avec un temps d'exclusion un peu lent.

La création de notre modèle d'entrainement se fait via l'algorithme de réseau de neurones profond. Nous avons créé un modèle avec une seule sortie et trois niveaux avec la fonction d'activation Linéaire Relu. La raison pour laquelle nous avons choisi cette fonction et non pas d'autre fonctions traditionnelle (tangente, sigmoïde et hyperbolique) est que la fonction d'activation Linéaire Relu est plus économique dans le calcul et capable d'accélérer la vitesse d'entrainements des réseaux de neurones, en outre, dans les fonctions traditionnelles telle que sigmoïde, sa valeur de sortie est limitée entre 0 et 1, alors que dans notre cas, la valeur de sortie doit être entre 1 et 5.

Il existe d'autres paramètres dans le modèle, à savoir, l'optimisateur Adam[3] qui fait la mise à jour des poids [103], de plus une fonction d'optimisation grâce à un apprentissage par gradient (Lossfunction) car le modèle doit être optimisé pour un maximum de vraisemblance. Mathématiquement, cela peut être appliqué en minimisant la perte d'entropie croisée entre les étiquettes de données et les prédictions du modèle. L'entropie croisée équivalente peut également être minimisée en minimisant l'erreur quadratique moyenne (MSE) de la prédictiondu modèle. L'entropie croisée catégorique s'est avéré la mieux adaptée aux réseaux de neurones profonds généraux [104].

Passons à l'entrainement du modèle, comme nous savons que l'apprentissage automatique facilite beaucoup de problèmes dans la vie informatique, dans ce travail nous avons utilisé l'apprentissage profond pour entrainer notre modèle.

Alors, nous avons décomposé notre jeu de données en deux parties : partie train (training_data) avec une proportion de 80 %, et partie test (testing_data) avec 20 %.

La partie train aura pour objectif d'entrainer le modèle sur ces données, ensuite nous pouvons faire une comparaison des résultats obtenus avec la partie test, où la partie train (training_data) est un ensemble des attributs (features=input) est ses réponses (label=output), c'est comme une relation de question / réponse, alors l'opération de training va mémoriser, analyser et essayera de trouver une relation entre input-output exprimé par des poids et des valeurs ajoutés.

Nous évaluons notre approche par le MAE, RMSE, la précision et à la fin de cette opération, nous pouvons encore évaluer l'efficacité du système par la mesure (test_accuracy) qui est le pourcentage des réponses vrai dans le modèle.

Les bonnes performances d'un modèle exigent que la mesure de la partie test soit élevée, parce ce que nous ne pouvons pas juger un modèle seulement par la mesure de traininig. La

[3] Un algorithme d'optimisation basée sur le gradient du premier ordre de fonctions objectives stochastiques, basé sur des estimations adaptatives de moments d'ordre inférieur.

qualité des résultats revient du choix des fonctions d'activation de réseau (tangente, sigmoïde, relu, …) et aussi à l'intervalle des poids synaptiques.

Finalement, le résultat de réseau neurone profond après chaque entrainement est une seule prédiction (vote), que nous allons l'utilisée pour la recommandation.

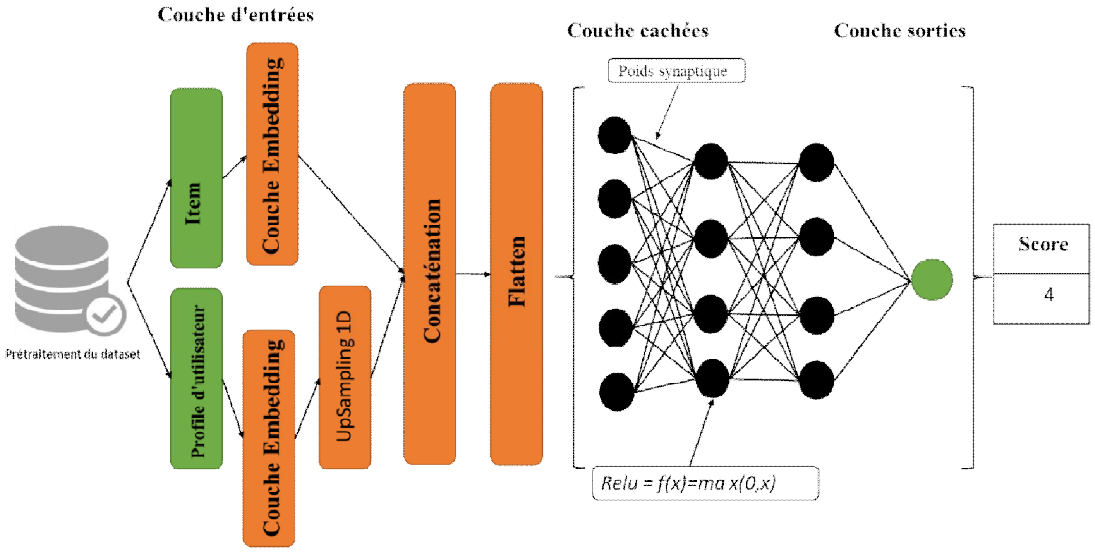


Figure 3.3 : Architecture de Multi Layer Perceptron (MLP)

3.3.2.2 Clustering :

Cette partie consiste à regrouper les utilisateurs par L'algorithme de clustering,K-Meansavec la similarité cosinus en utilisant l'identifiant du film avec ses catégories et l'évaluation ; cela va nous permettre de créer des groupes d'utilisateurs similaires (clusters) ou chaque cluster va contenir un groupe d'utilisateurs qui ont évaluer les films avec des évaluations proches.

Exemple :Etant donnée un ensemble des utilisateurs, afin de regrouper les utilisateurs selon leurs évaluations des films,nous allons appliquer l'algorithme K-Means.Dans cette exemple le nombre de clusters k=5, voir la figure ci-dessous :

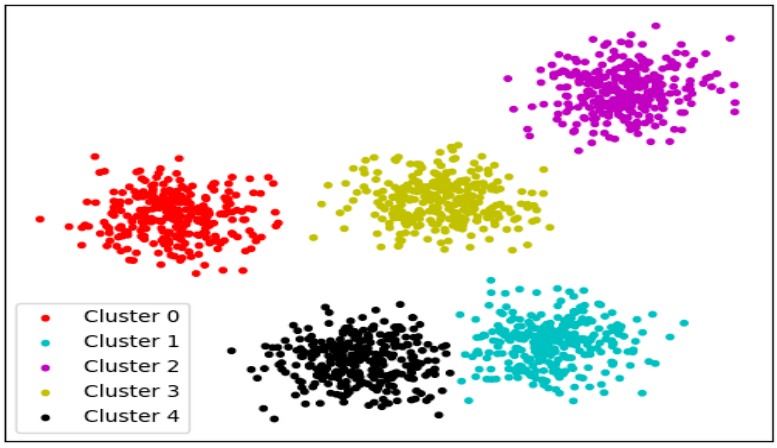


Figure 3.4 : Regroupement par l'algorithme K-Means

3.3.3 La recommandation :

Cette partie constitue l'objectif principale et l'étape finale de notre projet, c'est bien la phase de recommandation où nous allons planifier et suggérer les ressources pour les utilisateurs. Nous avons développé un nouveau modèle qui peut garantir des résultats acceptables, grâce à son bon fonctionnement surtout au niveau de la mémorisation des données déjà appris. Ce modèle est utilisé avec des entrées différentes, ces derniers sont constitués d'une part d'une liste des items et d'autre un utilisateur que nous souhaitons-luirecommander les ressources satisfaisantes. Il est bien de noter que la liste des items est issue du cluster cité ci-dessus auquel appartient l'utilisateur concerné, d'autre part, la sortie est une liste des top-N items classés par leurs scores, et par conséquent à partir de cette dernière liste, nous recommandons les items appropriés pour chaque utilisateur.

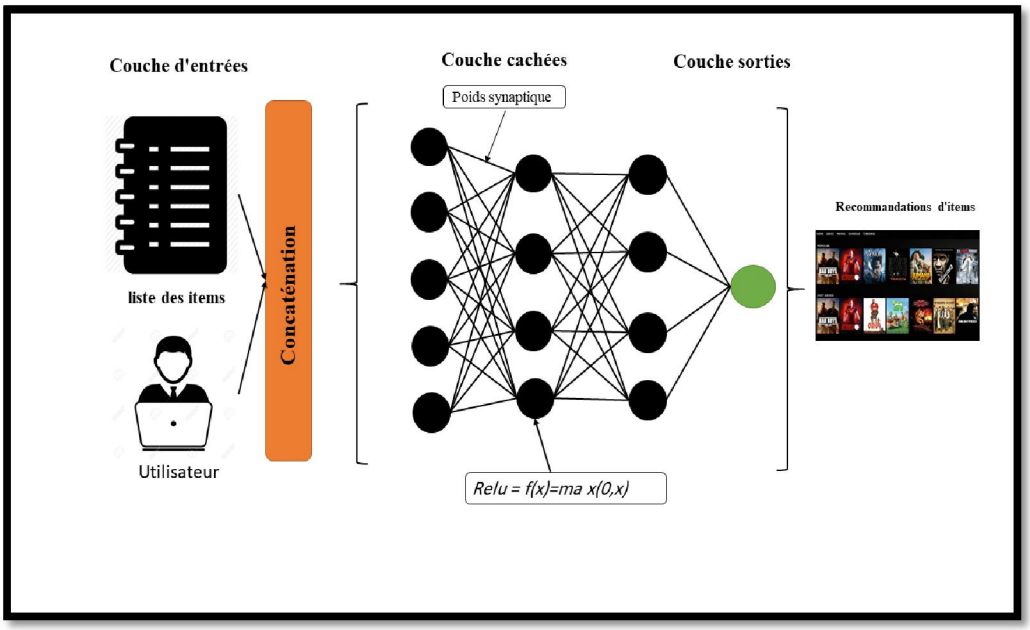


Figure 3.5 : L'architecture de la recommandation.

3.4 Conclusion :

Nous avons présenté dans ce chapitre notre méthodologie de travail et bien détaillé tous ses étapes, nous avons créé un nouveau modèle de système de recommandation sans utiliser une des techniques classiques, et à l'aide de l'apprentissage automatique avec l'algorithme de réseau de neurones et nous allons voir les résultats de l'expérimentation dans le chapitre suivant.

Chapitre 4

Evaluation expérimentale de l’approche proposée

4.1 Introduction :

Après avoir achevé l'étape de modélisation, nous consacrerons ce chapitre à l'implémentation et aux évaluations des différents algorithmes qui composent notre système de recommandation, ainsi son expérimentation sur un dataset du domaine. Nous finirons ce chapitre par des comparaisons entre le modèle et autres modèles étudiés, puis nous allons présenter évaluation de l'efficacité de l'approche.

4.2 Cadre d'évaluation :

Dans cette partie, nous allons mener une analyse offline, présenter le corpus de données et tous les outils pour la réalisation de notre système de recommandation.

4.2.1 Corpus de données (Collection de données)

Afin de pouvoir appliquer notre démarche, nous avons besoin d'un jeu de données contenant les utilisateurs, les items à recommander et les ratings donnés pour ces items. Pour cela nous allons utiliser la base lM de MovieLens[4]qui a été collectée par le projet de recherche GroupLens à l'Université de Minnesota. Cette base contient les informations de 3952 films, 6040 utilisateurs et 1000209 votes.
La valeur d'un vote est entre 1 et 5 et chaque utilisateur a noté au moins 20films.
Les fichiers principaux de la base MovieLens sont les suivants : users.csv,movies.csv etrating.csv.
Les informations des utilisateurs (Users) sont contenues dans le fichier users.csv comme suit :

Champs	Description
UserId	C'est l'identifiant du user (Il est unique pour chaque user)
Gender	Est le sexe est noté par un "M "pour les hommes et "F "pour les femmes
Âge	L'âge est choisi parmi des plages d'âges
Occupation	La fonction occupée

Tableau 4.1 : Description des champs du fichier users

[4]Voir Annexe A, récupérer de https://movielens.org/

Les informations des films (items) sont contenues dans le fichier movies.csv comme suit :

Champs	Description
MovieId	C'est l'identifiant du film (Il est unique pour chaque film)
Title	Contient le titre de chaque film.
Genres	Le genre du film

Tableau 4.2 : Description des champs du fichiermovies

Le fichier ratting.csv contient des votes créés par des utilisateurs sur des films (items). C'estune liste dont les champs sont :

Champs	Description
UserId	L'identifiant de chaque utilisateur.
MovieId	C'est l'identifiant du film.
Rating	La note donnée par un utilisateur à un film.

Tableau 4.3 : Description des champs du fichier ratings

Formats de données

CSV :

Pour « CommaSeparatedValues » est un type de format de données dans lequel chaque élément de données est séparé par une virgule. Les fichiers CSV sont utilisés lorsque le transfert de flux de produits est nécessaire.

4.2.2 Langage de programmation

Nous avons utilisé lors de la réalisation de notre système le langage de programmation**Python**avec différents bibliothèques que nous allons présenter ci-après.

Python est un langage de programmation de haut niveau, il supporte la programmation orientée objet, impérative séquentielle et fonctionnelle. Il se distingue notamment par une syntaxe simplifiée et concise permettant un gain de temps considérable aux programmeurs [60]. Il fournit une grande quantité de modules et de bibliothèques afin de faciliter les développements, sans oublier que c'est un langage multiplateformes, il fonctionne aussi bien sous Windows, Mac, Linux ainsi que sous de nombreuses variantes dérivées d'UNIX.

Librairies (bibliothèques)



Figure 4.1: Logos de quelques librairies utilisées

Keras :

Est une API de réseaux de neurones de haut niveau, écrite en Python et interfaçable avec TensorFlow, CNTK et Theano. Elle a été développée avec pour objectif de permettre des expérimentations rapides.

TensorFlow :

Est une bibliothèque open source de Machine Learning, créée par Google, permettant de développer et d'exécuter des applications de Machine Learning et de Deep Learning.

4.2.3 Métriques d'évaluation

Pour définir le nombre de cluster approprié nous avons utilisé le **coefficient de silhouette** qui nous permet de quantifier à quel point un clustering répond à les deux exigences (cohésionet séparation), sa valeur est entre [-1,1]. Pour chaque point, nous calculons le coefficient de silhouette qui est la différence entre la distance moyenne avec les points du même groupe que lui (cohésion) et la distance moyenne avec les points des autres groupes voisins (séparation). Si cette différence est inférieure àzéro, le point est en moyenne plus proche du groupe voisin que du sien : il est donc mal classé. A l'inverse, si cette différence est grand que 0, le point est en moyenne plus proche de son groupe que du groupe voisin : il est donc bien classé.

Le coefficient de silhouette proprement dit est la moyenne du coefficient de silhouette pour tous les points.

Comme nous avons cité précédemment dans le premier chapitre,il existe différents protocoles d'évaluations dont nous avons choisi parmi eux le protocole **Hold out**qui divise le dataset en deux partie (training, test) selon un pourcentage donné , nous avons fixé ce pourcentage à 80% c.-à-d. , il va mettre les 80% du data dans la partie training et les 20% restant dans la partie test. Et parmi les différentes métriques citées dans le premier chapitre, nous avons utilisé **MAE** qui calcule l'écart entre la valeur réel et la valeur prédite, et **RMSE**qui repose sur le même principe que la MAE, tout en mettant l'accent sur les écarts importants.

4.2.3.1 Métrique d'évaluation de la prédiction basée modèle (RMSE) :

Nous avons obtenu ces résultats concernant la métrique de RMSE pour le MLP avec et sans l'utilisation des catégories de films :

	Avec l'utilisation des catégories	Sans l'utilisation des catégories
Train RMSE	0.9270	0.7217
Test RMSE	0.9317	0.9376

Tableau 4.4 : Les résultats de RMSE pour le MLP

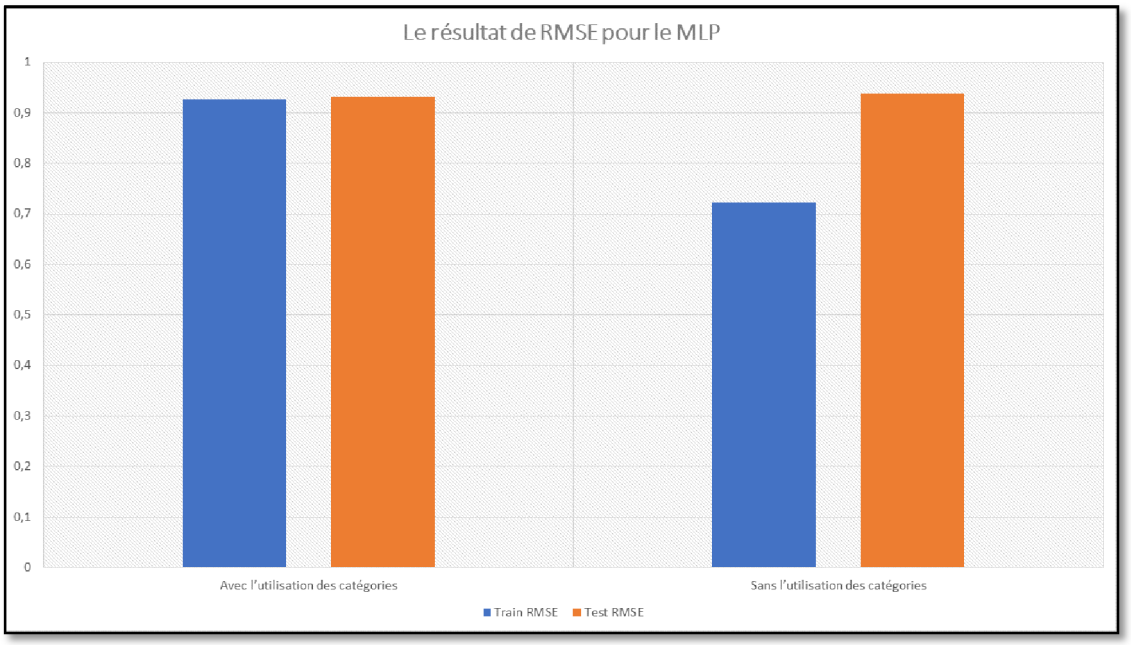


Figure 4.2 : Les résultats de RMSE pour le MLP

Analyse des résultats de RMSE :

Selon les résultats du tableau 4.4, nous constatons que l'application du ML contribue à une amélioration de la RMSE.Nous remarquons que les valeurs RMSE sont très proches, en revanche, le RMSE est légèrement meilleur en utilisant les catégories que celui sans utilisation de catégories (test).

4.2.3.2 Métrique d'évaluation de la prédiction (MAE) :

Nous avons testé une autre métrique MAE.Le tableau 4.5 présente les résultats avec l'utilisation ou non des catégories:

	Avec l'utilisation des catégories	Sans l'utilisation des catégories
Train MAE	0.7424	0.5522
Test MAE	0.7458	0.7342

Tableau 4.5 : Les résultats de MAE pour le MLP

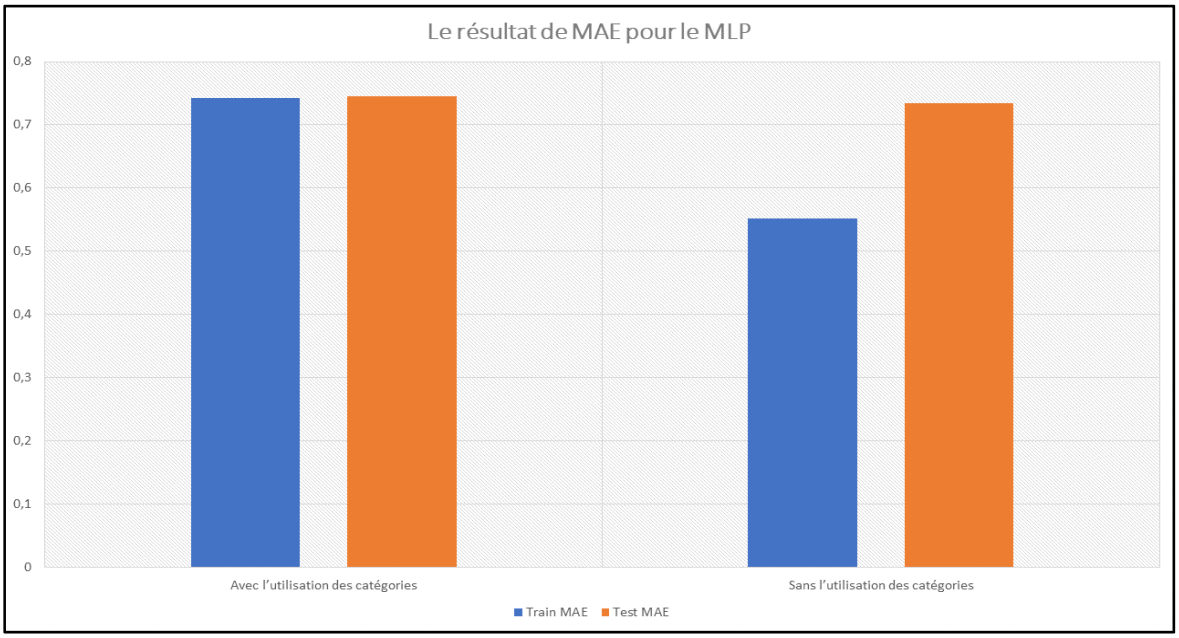


Figure 4.3 : Les résultats de MAE pour le MLP

Analyse des résultats de MAE :

En observant les résultats du tableau 4.5, nousremarquons que l'utilisation des catégories n'a pas amélioré les résultats de MAE.

4.2.3.3 Temps de calcul

Nous nous intéressons à l'évaluation du temps de calcul requis pour la phase de calcul des similarités pour la recommandation, avec ou sans clustering. Les résultats de cette évaluation ont montré que les modèles n'intégrant pas le clustering, requièrent en moyenne un temps de calcul plus élevé. Ce temps de calcul résulte du fait que les similarités ont été évaluées entre toutes les paires d'utilisateurs contenues dans le corpus d'apprentissage. Or, avec l'application du clustering, ces similarités sont calculées uniquement au sein des clusters, ce qui se répercute sur le nombre d'utilisateurs concernés par l'évaluation, qui tend bien évidemment à la baisse. En effet, ce temps est de **12.61 sec** pour notre modèle hybride (avec l'application du clustering.) et de **29.73 sec** pour le MLP (sans l'utilisation du clustering) ce qui a diminué le temps d'exécution jusqu'à **57,59%.**

4.2.3.4 Analyse des résultats généraux :

Malgré que notre modèle avec l'utilisation des catégories et sans catégories a donné des résultats un peu similaire, notre modèle a donné des résultats acceptable pour les deux métriques.

Les résultats relatifs à l'expérimentation du notre modèle MPLavec catégories correspondent à **0.9317** en terme de RMSE et à **0.7458** en MAE. Ainsi, au niveau de la MAE, aucune amélioration n'est atteinte. Or, en terme de RMSE, le modèle a permis d'atteindre une légère amélioration.

De plus, le modèle sans catégories contribue à une amélioration de RMSE=**0.9376** et MAE=**0.7342**comparés au modèle de filtrage collaboratif classique.

L'application du clustering reste bénéfique en terme de temps de calcul.

4.2.3.5 Amélioration des résultats :

Nous pouvons améliorer ce travail prochainement par les propositions suivantes :

- Construire un réseau plus complexe en ajoutant plus de couches cachées.
- Utiliser un jeu de données de MovieLens de plus de 1M.
- Ajouter d'autres attributs plus performants.

4.3 Partie application :

Nous avons développé une application Python qui calcule la précision des meilleures N recommandations pour chaque modèle (MLP, hybride). Cette application utilise un ensemble de données ml-latest-small de MovieLens. Comme illustré sur la figure 4.4, l'expérimentation est composée de trois phases. L'application d'abord divise les évaluations des utilisateurs de manière aléatoire dans un ensemble d'apprentissage (80%) et un ensemble de tests (20%). Dans une deuxième étape, l'application utilise l'ensemble d'apprentissage pour calculer les meilleures N ressources recommandées pour chaque utilisateur en utilisant les deux algorithmes, MLP et clustering. Enfin, l'application utilise ces recommandations et l'ensemble de test pour calculer et vérifier les scores.

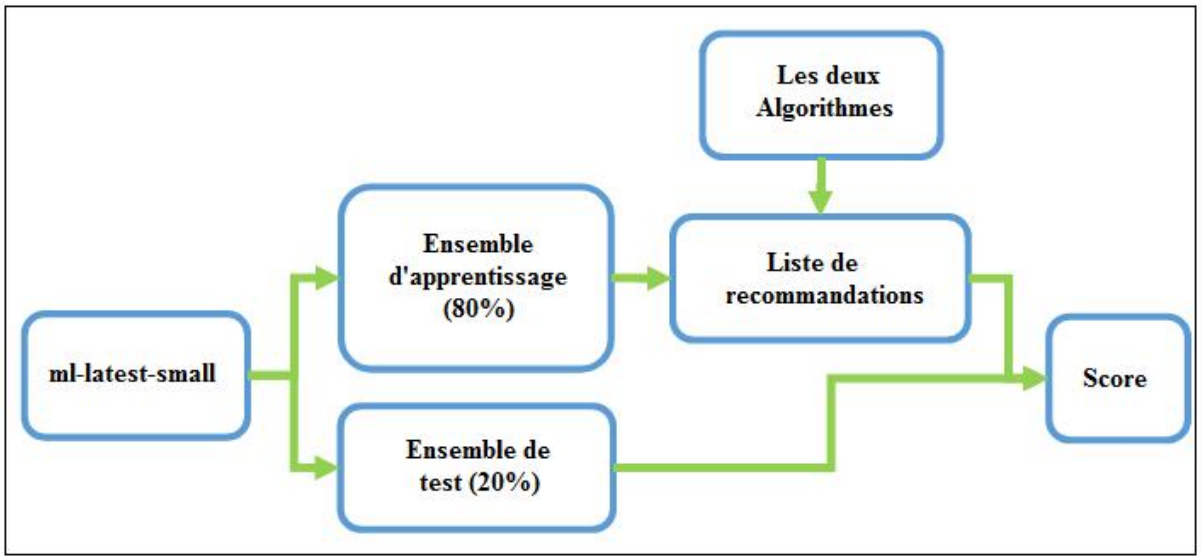


Figure 4.4 : Processus d'évaluation offline

4.3.1 Recommandation en ligne :

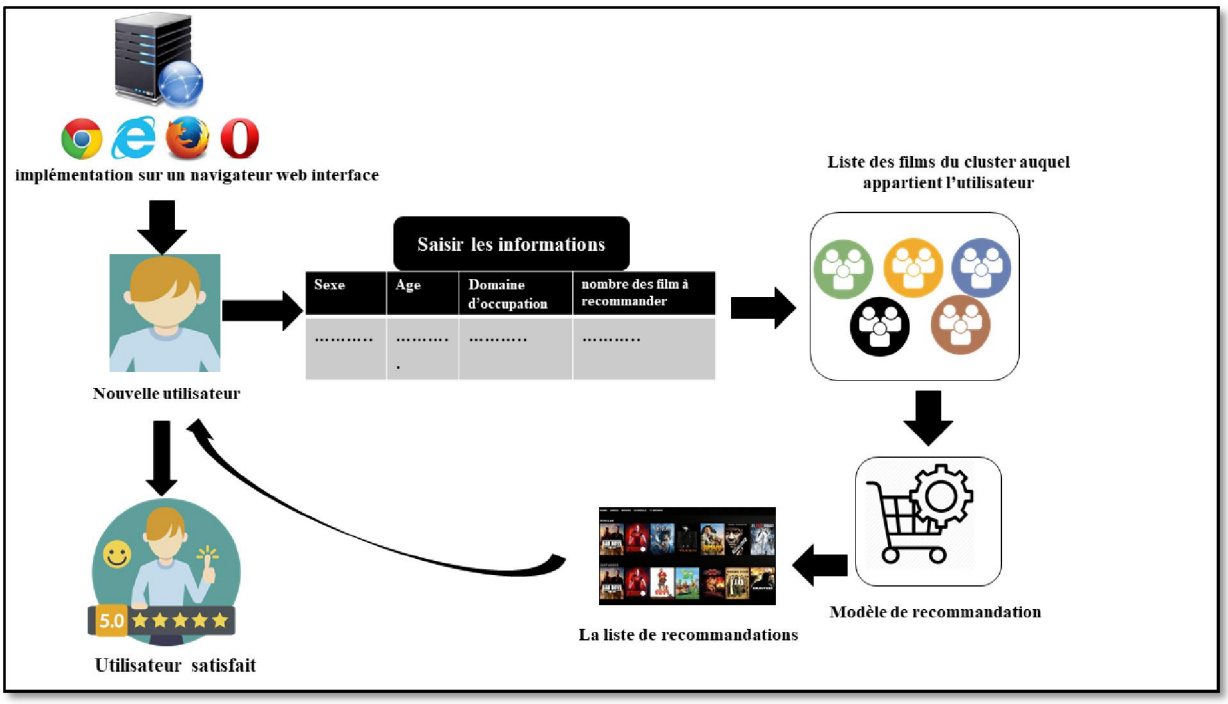


Figure 4.5 : Recommandation en ligne

1. La saisie des informations d'un nouvel utilisateur etle nombre des films à recommander(les top-N) :

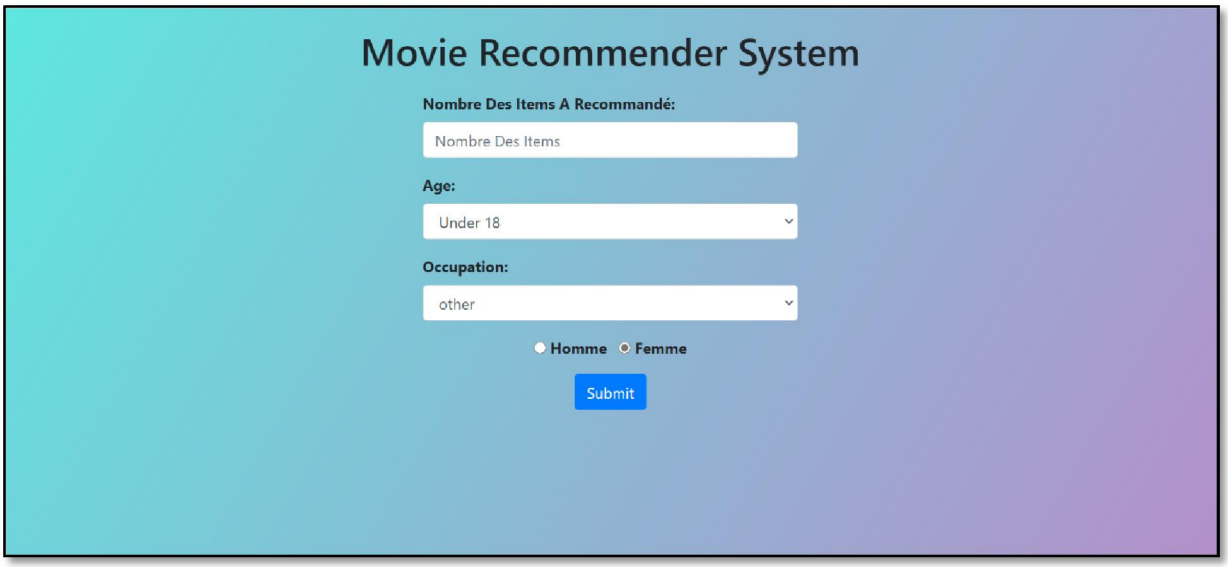
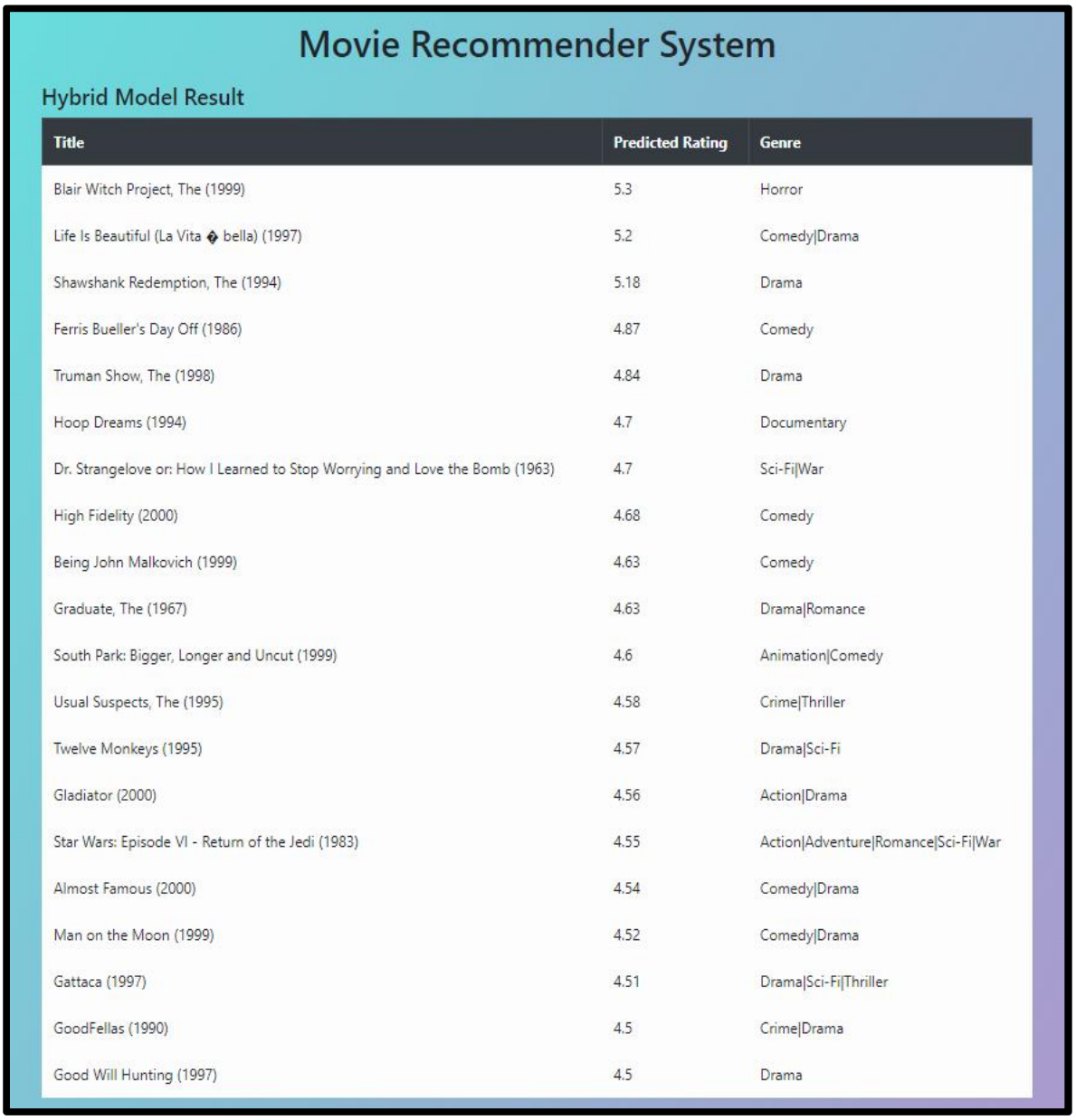


Figure 4.6 : Capture de la partie saisis des informations d'un nouvel utilisateur

2. Les résultats de recommandation sont sous forme d'une liste de top-N films avec leur notes prédites et le genre (action, comédie, romance, …).



Title	Predicted Rating	Genre
Blair Witch Project, The (1999)	5.3	Horror
Life Is Beautiful (La Vita bella) (1997)	5.2	Comedy|Drama
Shawshank Redemption, The (1994)	5.18	Drama
Ferris Bueller's Day Off (1986)	4.87	Comedy
Truman Show, The (1998)	4.84	Drama
Hoop Dreams (1994)	4.7	Documentary
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1963)	4.7	Sci-Fi|War
High Fidelity (2000)	4.68	Comedy
Being John Malkovich (1999)	4.63	Comedy
Graduate, The (1967)	4.63	Drama|Romance
South Park: Bigger, Longer and Uncut (1999)	4.6	Animation|Comedy
Usual Suspects, The (1995)	4.58	Crime|Thriller
Twelve Monkeys (1995)	4.57	Drama|Sci-Fi
Gladiator (2000)	4.56	Action|Drama
Star Wars: Episode VI - Return of the Jedi (1983)	4.55	Action|Adventure|Romance|Sci-Fi|War
Almost Famous (2000)	4.54	Comedy|Drama
Man on the Moon (1999)	4.52	Comedy|Drama
Gattaca (1997)	4.51	Drama|Sci-Fi|Thriller
GoodFellas (1990)	4.5	Crime|Drama
Good Will Hunting (1997)	4.5	Drama

Figure 4.7 : Capture des résultats de la recommandation de films à un utilisateur

4.3.2 Recommandation hors ligne :

Pour effectuer une évaluation, nous avons besoin de résultats à partir d'algorithme de recommandation. Or ils fournissent des items defaçon triée à l'aide d'une valeur de pertinence (entre 1 et 5). Donc, nous avons défini un seuilau-delà duquel un item est recommandé et en dessous duquel il ne l'est pas. Le seuil de 3,5 a étéchoisi pour l'évaluation de la pertinence. La mesure d'évaluation de la recommandation classique est le calcul de la précision et du rappel. Précisions et rappel sontdes fonctions qui produisent un résultat entre 0 et 1.

Hybrid Model Result

Title	Real Rating	Predicted Rating	Genre
Godfather, The (1972)	5	4.90	Action|Crime|Drama
Fargo (1996)	5	4.86	Crime|Drama|Thriller
Boat, The (Das Boot) (1981)	Inconnu	4.86	Action|Drama|War
Raising Arizona (1987)	5	4.84	Comedy
Annie Hall (1977)	5	4.84	Comedy|Romance
Blade Runner (1982)	4	4.82	Film-Noir|Sci-Fi
Pulp Fiction (1994)	5	4.81	Crime|Drama
North by Northwest (1959)	4	4.8	Drama|Thriller
Citizen Kane (1941)	5	4.79	Drama
This Is Spinal Tap (1984)	5	4.7	Comedy|Drama|Musical
Clockwork Orange, A (1971)	5	4.77	Sci-Fi
Usual Suspects, The (1995)	5	4.73	Crime|Thriller
Monty Python and the Holy Grail (1974)	4	4.72	Comedy
Rushmore (1998)	4	4.67	comedy
American Beauty (1999)	5	4.66	Comedy|Drama
Taxi Driver (1976)	5	4.64	Drama|Thriller
Boogie Nights (1997)	4	4.63	Drama
Bridge on the River Kwai, The (1957)	Inconnu	4.62	Drama|War
Rear Window (1954)	5	4.60	Mystery|Thriller

Figure 4.8 : Capture sur les premiers 20 films recommandés à l'utilisateur par notre modèle hybride

4.3.2.1 Les résultats de mesure de précision et rappel :

Nous avons défini un seuil au-delà duquel un item est recommandé qui est 3,5 ; mais, pour qu'il soit un élément pertinent il faut que le vote prédit soit plus grand que 3.5 et la différence entre le vote prédit et le vote réelle doit être moins que 1.

Après nouscalculons les vrais positifs (les éléments pertinents et recommandés), les vrais négatifs (élément non pertinent et non recommandé), les faux positifs (les éléments non pertinents et recommandés) et les faux négatifs (les éléments Pertinent et non recommandé). Ensuite, nous calculons la moyenne de la précision et le rappel sur N éléments depuis les résultats des films recommandés pour des utilisateurs.

$$Précision = TP / (TP + FP) \quad (4.1)$$

$$Rappel = TP / (TP + FN) \quad (4.2)$$

Cette stratégie consiste à sélectionner les Top N items recommandés. Nous avons considéré ces Top N, tel que N=10, 20, 30 et 40.

➢ Modèle MLP :

N	Précision	Rappel
10	**0.7595**	0.5269
20	0.6003	0.6649
30	0.4945	0.7244
40	0.3633	0.7748

Tableau 4.6 : Moyenne de Précision et Rappel à N éléments du modèle MLP

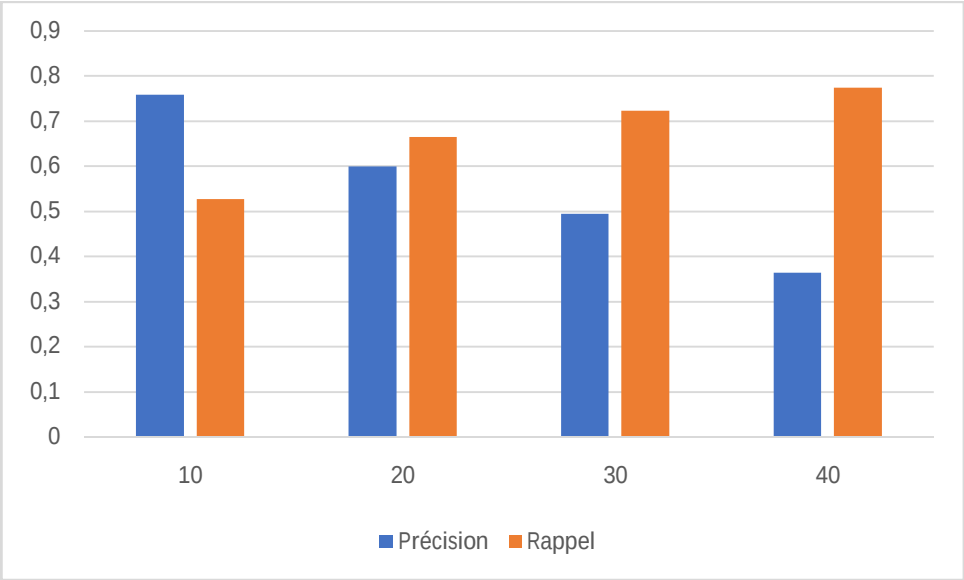


Figure 4.9 : Précision et Rappel du modèle MLP

Le meilleur résultat relatif à cette expérimentation correspondent au choix duN=10 pour la précision (=**0,7595)** mais avec un rappel de 0,5269.

Nous avons aussi calculé la précision de prédiction (accuracy) sur toute la partie test et nous avons obtenu une valeur de**86.64%**que nous concèderons qu'il est un résultat acceptable et qui prouve l'efficacité de notre modèle MLP.

➢ Modèle Hybride :

N	Précision	Rappel
10	**0.8970**	0.3443
20	0.8235	0. 5363
30	0.7860	0.5988
40	0.7860	0.8886

Tableau 4.7 : Moyenne de Précision et Rappel à N éléments du modèle hybride

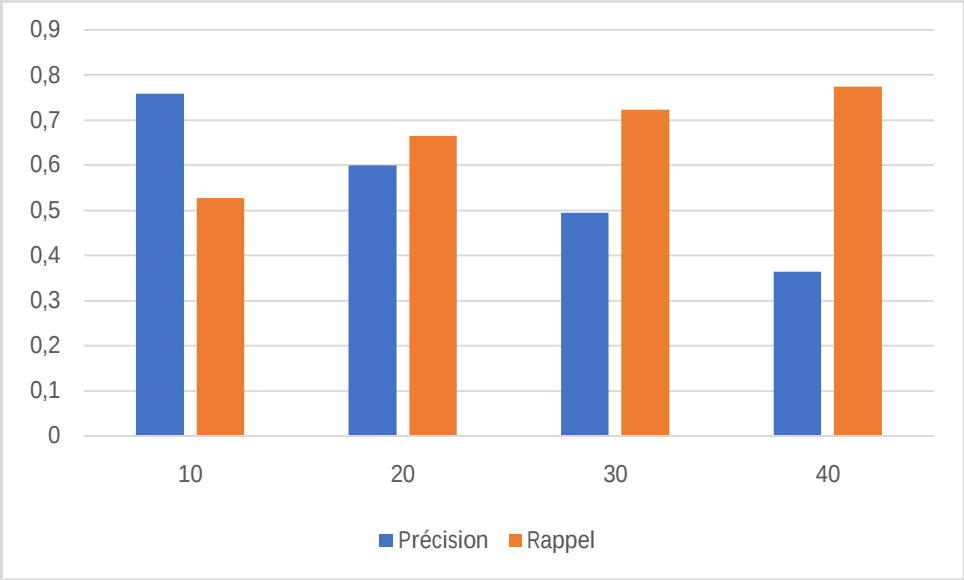


Figure 4.10 : Précision et Rappel du modèle hybride

Le résultat obtenu pour cette expérimentation correspond à 0.8970 en terme de précision et à 0.3443 en rappel pour N=10. Ainsi, en comparant avec le modèle non hybride, nous pouvons déduire que le clustering a amélioré la performance du modèle.

Notre modèle hybride a permis d'atteindre une amélioration de la précision jusqu'à 13,75%, comparé au meilleur score de précision obtenu par le modèle sans clustering. De plus, il donne toujours de bonnes précisions pour N=20, 30 et 40 comparés à ceux du MLP qui sont faibles.

La précision de la prédiction (accuracy) max d'un utilisateur est 88.55% sur tous ses films,

4.3.2.2Analyse des résultats de recommandation :

Globalement, l'expérience a donné des résultats positifs. Les expérimentations nous ont permis de mettre en évidence certaines limites,la précision pour N=30 et N=40 est inférieur à la moyenne (seulement 0.4945 et 0.3633) dans le modèle sans clustering.

4.4 Comparaison entre notre modèle et les autres modèles étudiés [105]:

Notre modèle qui utiliseune technologie d'apprentissage en profondeur est comparé avec d'autres algorithmes de recommandation :

Algo	IB	SVD	PMF	MCoc	DSRec	PMM-MF	Hern	SCC	TYCO	Zhang	Our MLP
RMSE	0.9642	0.9533	0.9512	0.9676	0.9424	0.9515	0.9473	0.9441	0.9522	0.9357	0.9317
MAE	0.7004	0.6977	0.6901	0.6834	0.6790	0.6767	0.6671	0.6922	0.6646	0.6586	0.7458

Tableau 4.8 : Résultats d'évaluations des techniques avec différents algorithmes avec dataset MovieLens1M.

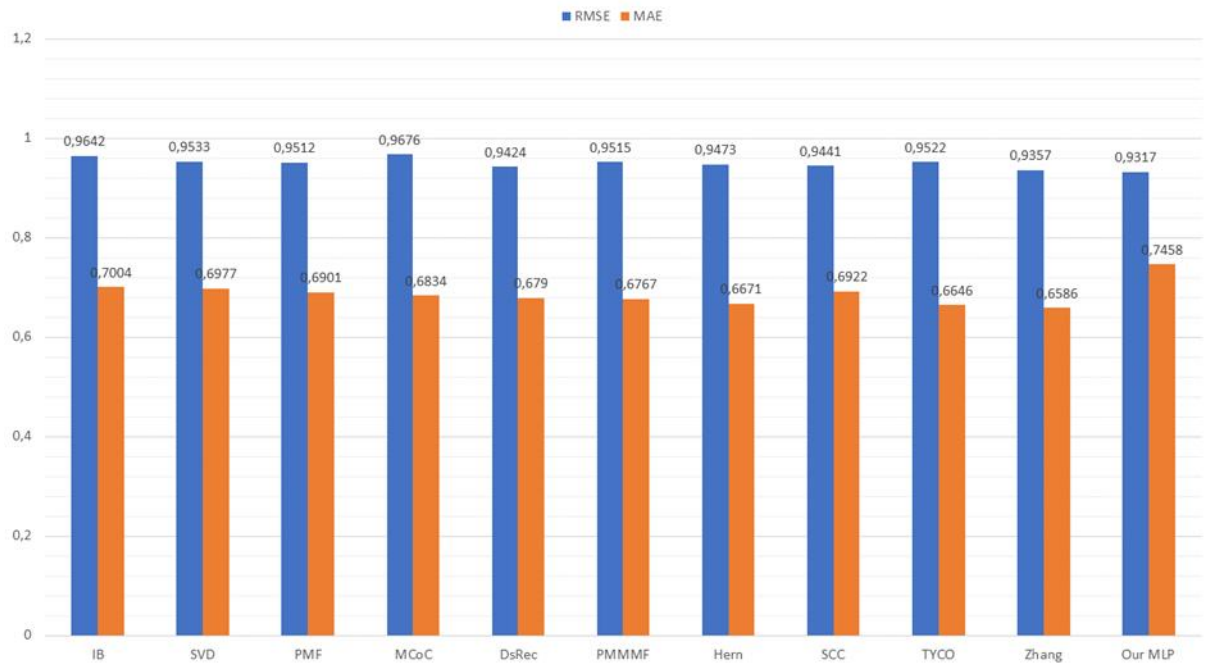


Figure 4.11 : Résultats d'évaluations des techniques avec différents algorithmes avec dataset MovieLens1M.

❖ **Méthodes comparées [105] :**

Les algorithmes de recommandation choisie dans la comparaison afin de vérifier l'efficacité et la précision du modèle de Zhang, sont listés comme suit :

- **Item-Based (IB) :** L'idée de base de cette méthode est de calculer la similarité entre les éléments, puis de recommander des éléments similaires à ceux que l'utilisateur actif préfère.
- **SVD :** Cet algorithme est basé sur la méthode de décomposition en valeurs singulières, dans laquelle la matrice d'évaluation est décomposée en deux matrices de faible dimension, qui seront ensuite utilisées pour effectuer la prédiction.
- **PMF :** Modèle de factorisation matricielle largement utilisé, basé sur le modèle gaussien de la recommandation.
- **MCoC :** L'idée principale de cette méthode est de regrouper les utilisateurs et les éléments en plusieurs sous-groupes, puis de faire des recommandations au sein de chaque sous-groupe en utilisant la méthode CF de base.
- **DsRec :** Un modèle hybride combinant le modèle de clustering de base et le modèle de factorisation matricielle pour améliorer la précision de la prédiction.
- **PMMMF :** cette méthode a été proposée par Kumar, qui a amélioré la méthode MMMF traditionnelle en utilisant une machine à vecteur de support proximal (PSVM).

- **Hern :** cet algorithme utilise la méthode de factorisation matricielle basée sur le modèle de probabilité bayésien pour décomposez la matrice de classement des utilisateurs en deux matrices non négatives, où les valeurs de tous les éléments sont comprises entre 0 et 1.
- **SCC :** cette méthode est identique à MCoC, qui est un algorithme de recommandation basé sur le clustering ; la différence est qu'il regroupe des éléments en utilisant une méthode de construction automatique.
- **TyCo :** cette méthode emprunte des idées de la typologie des objets à la psychologie cognitive et trouve les « voisins » des utilisateurs en fonction du degré de typicité des utilisateurs dans les groupes d'utilisateurs.

Nous constatons que les résultats obtenuspar notre modèle du système de recommandation en relation avec l'apprentissage profond(MAE=**0.7458** avec catégories, MAE=**0.7342**sans catégories& RMSE**=0.9317** avec catégories RMSE=**0.9376** sans catégories) sont proche à ceux proposées dans ce domaine d'étude par de nombreux auteurs dans d'autres travaux connexes.

4.5 Conclusion :

Ce chapitre a été consacré à la présentation de l'expérimentation et l'évaluation de notre modèle, dans lequel nous avonsprésenté les résultats obtenus en analysant et discutantd'unefaçon rigoureuse ces résultats. Cela nous abien amené à conclure que la recommandation avec l'utilisation de l'apprentissage profond et le clustering a de bonnes performances et réponds bien aux objectifs ciblés.

Conclusion générale et perspectives

Dans ce manuscrit nous avons proposéun système de recommandation avec l'utilisation des algorithmes d'apprentissage en profondeur pourobjectif de recommander une liste des films à l'utilisateur.

Nous avons étudiérigoureusement le processus de recommandation ainsi que ses techniques, de la même manière, nous avons focalisé notre étudesur l'apprentissage profond.Nous avons donnéaussi une illustration du modèle développé, ainsi qu'une évaluation que nous avons réaliséeafin de tester et valider le modèle via le jeu de données Movielens 1M, qui nous a permi de prouver les bonnes résultats du modèle proposé,d'après cela on conclut que les objectifs fixés au préalable ont été atteints.

Comme perspectives nous proposons d'améliorer notre système de recommandation en utilisant un jeu de données plus grand que celui adopté dans notre travail contenant plus d'attributs, en parallèle nous pouvons aussiaméliorer la pertinence des résultats obtenus en hybridant ce modèle avec d'autre approche de recommandation commele filtrage basé contenuou une approche utilisant la matrice de factorisation à titre d'exemple.

Nous proposons aussi comme perspective, une nouvelle approche sociale de recommandation afin d'identifier des voisins fiables et de d'améliorer la précision des recommandations.

Références Bibliographique

[01]Armelle Brun, MarharytaAleksandrova, and Anne Boyer. Can latent features be interpreted as users in matrix factorization-based recommender systems? In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02, pages 226233. IEEE Computer Society, 2014.

[02]Margaritis, K.G. &Vozalis, E. Analysis of Recommender Systems' Algorithms. 6th Hellenic European Conference on Computer Mathematics & its Applications (HERCMA), 2003

[03]Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6): pages 734-749.

[04]LemdanRoza. Thèse de Doctorat de l'Université Paris-Saclay préparée à Ecole doctorale n◦580 Sciences et technologies de l'information et de la communication, 2016.

[05] Goldberg, D., Nichols, D., Oki, M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry.In Commun. ACM, volume 35, pages 61-70.

[06] Michael J. Pazzani and Daniel Billsus.The adaptive web. Chapter Content-based Recommendation Systems, pages 325-341. Springer Verlag, Berlin, Heidelberg, 2007.

[07] Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) :331–370, 2002.

[08]Manouselis, N., Drachsler, H., Verbert, K., and Duval, E. Recommender Systems for Learning. Berlin, Springer, 2012.

[09] N. BECHET, Etat de l'art sur les Systèmes de Recommandation

[10]Belkin N.J., Croft W.B. (1992, December). Information filtering and information retrieval: two sides of the same colin? Communication of the ACM, 5(12), pp. 29-38.

[11] Croft W. B. (1993). Knowledge-based and Statistical approaches to Text Retrieval.IEEE EXPERT, 8(2), pp. 8-12.

[12] Clémente, R.P. (2010). Une architecture générique de Système de recommandation de combinaison d'items, Application au domaine du tourisme. Thèse de doctorat, Université de Bourgogne, France.

[13] O. Nouali, S. Benmezùine, M. Djaid, S. Sidi-Boumediène, Chapitre rédigé « Filtrage de l'Information », p 27-35, Laboratoire Intelligence Artificielle, CE.R.I.S. T, Rue des 3 frères Aïssou, Ben Aknoun, Alger, Algérie, RIST Vol.7 N°02 Année 1997.

[14] Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A. (2013). Recommendersystemssurvey. Knowledge-BasedSystems, 46, pp.109–132.

[15]Laouar, A. Korichi, K. Un système de recommandation pour l'assistance à la navigation sur Internet. Mémoire MASTER ACADEMIQUE Spécialité : Informatique industrielle. UNIVERSITE KASDI MERBAH OUARGLA. 2016.

[16]Basu, C., Hirsh, H., Cohen, W. and Manning, N. C. Technical paper recommendation: A study in combining multiple information sources, 2001.

[17]Margaritis, K. and Vozalis, E. Analysis of recommender systems' algorithms. In 6th Hellenic European Conference on Computer Mathematics its Applications (HERCMA), Athens, Greece, 2003.

[18]Jannach, Zanker, Felfernig and Friedrich, Recommender Systems –2010

[19]Bettini C., Brdiczka O., Henricksen K., Indulska J., Nicklas D., Ranganathan A., and Riboni D., A survey of context modelling and reasoning techniques. Journal of Pervasive and Mobile Computing, 2010.

[20] Pascoe, J. Adding generic contextual capabilities to wearable computers. In (ISWC, pages 92–99, 1998.

[21] Ali Mustafa QAMAR generalized cosine and similarity Matrice:A supervised : 7 août 2006

[22]Erdt, M., Fernandez, A., and Rensing, C. Evaluating recommender systems for technology enhanced learning: a quantitative survey. IEEE Transactions on Learning Technologies, 8(4), (pp. 326-344), 2015.

[23]Schilit, B., Hilbert, D., and Trevor, J. Context-aware communication. IEEE Wireless Commun, 9(5) :46–54, 2002.

[24]Herlocker, J. L., Konstan, J. A., Terveen, L. G., &Riedl, J. T. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), (pp. 5-53), 2004.

[25]McNee, S. M., Riedl, J., &Konstan, J. A. (2006, April). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In CHI'06 extended abstracts on Human factors in computing systems (pp.1097-1101). ACM.

[26]Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction, 22(4-5), (pp. 441-504).

[27]Fazeli, S., Drachsler, H., Bitter-Rijpkema, M., Brouns, F., Van der Vegt, W., &Sloep, P. B. (2017). User-centric Evaluation of Recommender Systems in Social Learning Platforms: Accuracy is Just the Tip of the Iceberg. IEEE Transactions on Learning Technologies.

[28]Chatti, M. A., Dakova, S., Thus, H., & Schroeder, U. (2013). Tag-based collaborative filtering recommendation in personal learning environments. IEEE Transactions on learning technologies, 6(4), (pp. 337-349).

[29]Erdt, M., Fernandez, A., &Rensing, C. (2015). Evaluating recommender systems for technology enhanced learning: a quantitative survey. IEEE Transactions on Learning Technologies, 8(4), (pp. 326-344).

[30]Zheng. Y, Mobasher. B and Burke. R. CSLIM: Contextual SLIM recommendation algorithms. In Proceedings of the 8th ACM Conference on Recommender Systems, pages 301– 304.ACM, 2014

[31]Gabrielsson, S. &Gabrielsson, S. The use of Self-Organizing Maps in Recommender Systems, A survey of the Recommender Systems field and a presentation of a State of the Art Highly Interactive Visual Movie Recommender System. Mémoire de master, Uppsala University, 2006

[32] C. Beaulac, “Intelligence artificielle avec apprentissage automatique pour l'estimation de la position d'un agent mobile en utilisant les modèles de markov cachés”, Mémoire. Montréal (Québec, Canada), Université du Québec à Montréal, Maîtrise en mathématique 2015.

[33] Kevin Kelly. The Three Breakthroughs That Have Finally Unleashed AI On The World. http://www.wired.com/2014/10/future-of-artificial-intelligence/, 2014

[34] Bertrand Braunschweig, “Intelligence artificielle - livre blanc,” pp. 1–21, 2016.

[35] Bouguessa, M. (2015b). Forage de données. Notes de cours. Université du Québec à Montréal, 88 pages.

[36] El Fouz, I. (2013). Clustering des News.Université de nicesophiaantipolis.

[37]Gadhvi, H., et Madhu, S. (2013). Comparative Study of Classification Algorithms for Web Spam Detection. International Journal of Engineering Research et Technology (IJERT), pp. 2497-2501.

[38] Silva, C. etRibeiro, B. (2009). Inductive infeTencefoT large scale text classification: kem el appr-oaches and techniq·ues, volume 255. Springer.

[39] J. HAN, M. KAMBER, J. PEI, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2012.

[40]https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-app-rn.pdf

[41]Naffakhi, N. (2004). Apprentissage supervisé pour la classification des images à l'aide de l'algèbre P-tree. Mémoire.

[42] L. ROKACH & O. MAIMON, Clustering methods, Department of Industrial Engineering, Tel-Aviv University, 2002.

[43]https://mrmint.fr/algorithme-k-means

[44] Ball, G. H. et Hall, D. J. ISODATA, an Iterative Method of Multivariate Analysis and Pattern Recognition. Behavior Science, 153, 1967.

[45] V. di Gesu. « Mathematical Morphology and Image Analysis: A Fuzzy

Approach ». Workshop on Knowledge-Based Systems and Models of Logical Reasoning, Reasoning, 1988

[46]Fairouz Hadi, KhierBenmahammed , Etude comparative entre la morphologie mathématique floue et le regroupement flou , Faculté des Sciences de l'Ingénieur, Université Ferhat Abbas-Sétif, Algérie.„ 3rd International Conference :SETIT 2005

[47] F. H¨oppner, F. Klawonn, R. Kruse, T. Runkler. Fuzzy Cluster Analysis, Methods for classification, data analysis and image recognition.Wiley, 2000.

[48] HAMIDOUCHE SADDEK & IDJERAOUI TAYEB, « Clustering : Approche par la théorie des jeux » mémoire de master, Université Mira Abderrahmane de Béjaa 2013

[49] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. Foundations and Trends® in Signal Processing, 7(3–4), 197-387.

[50]Bengio, Y. (2009). Learning deep architectures for AI.Foundations and trends in Machine Learning, 2(1), 1-127.

[51]Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

[52]Schmid Huber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

[53]LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning.nature, 521(7553), 436.

[54]Arel, I., Rose, D. C., &Karnowski, T. P. (2010). Deep machine learning-a new frontier in artificial intelligence research.IEEE computational intelligence magazine, 5(4), 13-18.

[55] J. Schmid Huber, "Deep Learning.," Scholarpedia, vol. 10, no. 11, p. 32832, 2015.

[56]Moualek, D. Y. (2017). Deep Learning pour la classification des images (Thèse de doctorat), Université Abou BakrBelkaid– Tlemcen.

[57] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives.IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI),38:1798–1828,2013

[58]Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2017). Quantized neural networks: Training neural networks with low precision weights and activations. The Journal of Machine Learning Research, 18(1), 6869-6898.

[59] Van den Oord, A., Dieleman, S., &Schrauwen, B. (2013).Deep content-based music recommendation.In Advances in neural information processing systems (pp. 2643-2651).

[60]Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitasks learning. In Proceedings of the 25th international conference on Machine learning (pp. 160-167).

[61]Matsugu, M., Mori, K., Mitari, Y., &Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks, 16(5-6), 555-559.

[62] Graves, A., &Jaitly, N. (2014, January). Towards end-to-end speech recognition with recurrent neural networks.In International conference on machine learning (pp. 1764-1772).

[63]Mikolov, T., Karafiát, M., Burget, L., Černocký, J., &Khudanpur, S. (2010). Recurrent neural network-based language model. In Eleventh annual conference of the international speech communication association

[64] Liu, S., Yang, N., Li, M., & Zhou, M. (2014, June). A recursive recurrent neural network for statistical machine translation. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1491-1500).

[39]Karpathy, A., &Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions.In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3128-3137).

[66] Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841-848).

[67] Ackley, D. H., Hinton, G. E., &Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. Cognitive science, 9(1), 147-169.

[68]Salakhutdinov, R., Mnih, A., & Hinton, G. (2007, June). Restricted Boltzmann machines for collaborative filtering.In Proceedings of the 24th international conference on Machine learning (pp. 791-798).

[69] Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In Neural networks: Tricks of the trade (pp. 599-619). Springer, Berlin, Heidelberg.

[70] Hinton, G. E., Osindero, S., &Teh, Y. W. (2006). A fast learning algorithm for deep belief nets.Neural computation, 18(7), 1527-1554.

[71]Salakhutdinov, R., & Hinton, G. (2009, April). Deep boltzmann machines. In Artificial intelligence and statistics (pp. 448-455).

[72]Srivastava, N., &Salakhutdinov, R. R. (2012). Multimodal learning with deep boltzmann machines. In Advances in neural information processing systems (pp. 2222-2230).

[73]Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ...&Bengio, Y. (2014). Generative adversarial nets.In Advances in neural information processing systems (pp. 2672-2680).

[74] Denton, E. L., Chintala, S., &Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in neural information processing systems (pp. 1486-1494).

[75]Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. ACM Comput.Surv.52, 1, Article 5 (February 2019), 38 pages. https://doi.org/10.1145/3285029

[76]Xiangnan He, Lizi Liao, Hanwang Zhang, LiqiangNie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the WWW.173–182. https://doi.org/10.1145/3038912.3052569

[77] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. Neural Networks 4, 2 (1991), 251–257.

[78] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 5 (1989), 359–366.

[79]Xiangnan He and Chua Tat-Seng. 2017. Neural factorization machines for sparse predictive analytics. In Proceedings of the SIGIR. ACM, Shinjuku, Tokyo, 355–364.

[80]Heng-Tze Cheng, LeventKoc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, HrishiAradhye, Glen Anderson,GregCorrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In Proceedings of the Recsys.7–10. http://doi.acm.org/10.1145/2988450.2988454

[81] Paul Covington, Jay Adams, and EmreSargin. 2016. Deep neural networks for YouTube recommendations. In Proceedings of the Recsys.191–198. http://doi.acm.org/10.1145/2959100.2959190

[82] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In Proceedings of the WWW.193–201. https://doi.org/10.1145/3038912.3052639a

[83]Minmin Chen, ZhixiangXu, KilianWeinberger, and FeiSha. 2012. Marginalized denoisingautoencoders for domain adaptation. arXiv preprint arXiv:1206.4683 (2012).

[84] Ian Goodfellow, YoshuaBengio, and Aaron Courville. 2016. Deep Learning. MIT Press.http://www.deeplearningbook.org.

[85]Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In Proceedings of the SIGKDD.1235–1244.

[86]Haochao Ying, Liang Chen, YuwenXiong, and Jian Wu. 2016. Collaborative deep ranking: A hybrid pair-wise recommendation algorithm with implicit feedback. In Proceedings of the PAKDD.555–567. http://doi.org/10.1007/ 978-3-319-31750-2_44

[87] Sheng Li, Jaya Kawale, and Yun Fu. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In Proceedings of the CIKM. ACM, Melbourne, 811–820.

[88]SuvashSedhain, Aditya Krishna Menon, Scott Sanner, and LexingXie. 2015. Autorec: Autoencoders meet collaborative filtering. In Proceedings of the WWW.111–112. http://doi.acm.org/10.1145/2740908.2742726

[89]Dawen Liang, Rahul G Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variationalautoencoders for collaborative filtering. arXiv preprint arXiv:1802.05814 (2018).

[90] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer product-based neural collaborative filtering. arXiv preprint arXiv:1808.03912 (2018).

[91]Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In Proceedings of the WSDM.565–573.http://doi.acm.org/10.1145/3159652.3159656

[92]Suhang Wang, Yilin Wang, Jiliang Tang, Kai Shu, SuhasRanganath, and Huan Liu. 2017. What your images reveal: Exploiting visual contents for point-of-interest recommendation. In Proceedings of the WWW.391–400. https://doi. org/10.1145/3038912.3052638

[93] Chenyi Lei, Dong Liu, Weiping Li, Zheng-Jun Zha, and Houqiang Li. 2016. Comparative deep learning of hybrid representations for image recommendations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2545–2553. http://dx.doi.org/10.1109/CVPR.2016.279

[94]Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the WSDM. 425–434. http://doi.acm.org/10.1145/3018661.3018665

[95]Xiaoxuan Shen, Baolin Yi, Zhaoli Zhang, Jiangbo Shu, and Hai Liu. 2016. Automatic recommendation technology for learning resources with convolutional neural network. In Proceedings of the International Symposium on Educational Technology. IEEE, Beijing, China, 30–34.

[96] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In Proceedings of the NIPS. Curran Associates, Inc., 2643–2651.

[97]Joonseok Lee, Sami Abu-El-Haija, BalakrishnanVaradarajan, and Apostol Paul Natsev. 2018. Collaborative deep metric learning for video understanding. In Proceedings of the KDD'18. ACM, 481–490.

[98]BalázsHidasi, AlexandrosKaratzoglou, LinasBaltrunas, and DomonkosTikk. 2015. Session-based recommendations with recurrent neural networks. Proceedings of the International Conference on Learning Representations. https://arxiv.org/abs/1511.06939

[99]Sai Wu, WeichaoRen, Chengchao Yu, Gang Chen, Dongxiang Zhang, and Jingbo Zhu. 2016. Personal recommendation using deep recurrent neural networks in NetEase. In Proceedings of the ICDE.1218–1229. http://doi.org/10.1109/ICDE.2016.7498326

[100] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent recommender networks. In Proceedings of the WSDM. 495–503. https://openreview.net/pdf?id=Bkv9FyHYx

[101] Hanjun Dai, YichenWang, Rakshit Trivedi, and Le Song. 2016. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In Proceedings of the Recsys. 29–34. http://doi.acm.org/10.1145/2988450.2988451

[102] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In Proceedings of the SIGKDD. ACM, Halifax, NS, 1933–1942.

[103]Diederik, P. Kingma. Et Jimmy, Lei Ba. (2015). Adam : A method for stochastic optimization. Published as a conference paper at ICLR.

[104] Bankman, Daniel. Et Moons, Bert. Verhelst, Marian. (2019). Embedded deep learning Algorithms, Architectures and Circuits for Always-on Neural Network Processing, Springer Nature Switzerland AG 2019, Library of Congress Control Number: 2018956280.

[105] Zhao, Jia. Et Sun, Gang. (2018). Detect User's Rating Characteristics by Separate Scores for Matrix Factorization Technique, School of Computer and Information Engineering, MDPI, Summetry, China.

Webographie

[W1] www.image-net.org consultée le : 02 avril 2020

[W2] www.fr.wikiversity.org consultée le : 02 avril 2020

[W3]http://datasciencetoday.net/index.php/fr/deep-learning/147-c-est-quoi-le-deep-learming consultée le : 13 juillet 2020

[W4]https://blogs.mediapart.fr/marc-tertre/blog/130318/intelligence-artificielle-comprendre-le-deep-et-le-machine-learningconsultée le 16 juillet 2020

[W5] http://cedric.cnam.fr/vertigo/Cours/ml2/coursArbresDecision.html 15 juillet 2020 consultée le 16 juillet 2020

[W6]https://www.stemmer-imaging.com/fr-fr/conseil-techniqu/apprentissage-automatique-et-apprentissage-profond/ consultée le 17 octobre 2020

Annexe A

Les fichiers principaux de la base MovieLens sont les suivants : users.csv, movies.csv etrating.csv.
Les informations des utilisateurs (Users) sont contenues dans le fichier movies.csv comme suit :

Champs	Description
UserID	C'est l'identifiant du user (Il est unique pour chaque user)
Gender	Est le sexe est noté par un "M "pour les hommes et "F "pour les femmes
Âge	L'âge est choisi parmi les plages suivantes : [Moins de 18], [18-24], [25-34], [35-44], [45-49], [50-55], [56+]

Occupation	• Student • Employed • Scientist • Executive/managerial • Writer • Homemaker • Programmer • Technician/engineer • Academic/educator • Clerical/admin • Sales/marketing • Self-employed • College/gradstudent • Lawyer • Farmer • Unemployed • Artist • Tradesman/cratsman • Customer service • Retired • Doctor/health care • Other or not specified

Tableau A.1 : Tableau de description des champs du fichier movies

Les informations des films (items) sont contenues dans le fichier movies.csv comme suit :

Champs	Description
MovieId	C'est l'identifiant du film (Il est unique pour chaque film)
Title	Contient le titre de chaque film.
Genres	Le genre du film : • Action • Adventure • Animation

	• Children's • Comedy • Crime • Documentary • Drama • Fantasy • Film-Noir • Horror • Musical • Mystery • Romance • Sci-Fi • Thriller • War • Western • (no genres listed)

Tableau A.2 : Tableau de description des champs du fichier movies

Le fichier ratting.csv contient des votes créés par des utilisateurs sur des films (items). C'est une liste dont les champs sont :

Champs	Description
UserId	L'identifiant de chaque utilisateur.
MovieId	C'est l'identifiant du film.
Rating	La note donnée par un utilisateur à un film.

Tableau A.3 : Tableau de description des champs du fichier ratting