Université Dr.Yahia Fares Médéa

جامعة الدكتور يحي فارس

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Médéa / Faculté des Sciences

Département des mathématiques et informatique

Projet de fin d'études

Présenté pour l'obtention du Diplôme de **Master en Informatique**

Option : Ingénierie des Systèmes et Technologie du Web

Thème

Une nouvelle approche pour un système de recommandation utilisant une technique d'apprentissage automatique

Préparé par :

M. Sakhri Yahia

Proposé par :

Mme. SADOUKI F.

2023/2024

Remerciement

Je tiens tout d’abord à remercier Dieu le tout, qui nous a donné la force et la patience d’accomplir ce Modeste travail.

Ensuite, je tiens à remercier tout particulièrement mon encadreur Mme Sadouki Fatiha pour ses conseils et l'intérêt qu'elle a porté à mon travail.

Je remercie également les membres du jury pour l'intérêt qu'ils ont porté à ma mémoire en acceptant d'étudier nom travail et en l'enrichissant de leurs suggestions.

Dédicaces

Je dédie ce mémoire à mes très chers parents.

A mes sœurs et leurs filles Mariam, Rital et Rahil

A mon frère et mes amis

Résume :

Les systèmes de recommandation (RS) sont utilisés dans de nombreux produits et services tels que Google, Amazon, YouTube, Facebook, Netflix... Dans ce mémoire, nous visons à créer un système de recommandation basé sur les relations sémantiques des genres et des tags de films en appliquant des algorithmes d'apprentissage profond pour répondre aux préférences des utilisateurs.

Notre travail est subdivisé en deux parties, la première partie présente l'état de l'art de projet qui explique les systèmes de recommandation en détail et l'apprentissage automatique. La deuxième partie présente l'approche proposée y compris les étapes de création du modèle ainsi que l'évaluation des résultats en utilisant deux métriques différentes et à la fin, la comparaison de nos résultats avec ceux d'autres modèles..

Mots-clés : Système de recommandation, Apprentissage en profondeur, DNN, Auto-encodeur, prédiction.

Abstract

Recommender systems (RS) are used in many products and services such as Google, Amazon, YouTube, Facebook and Netflix... In this dissertation, we aim to create a recommendation system based on the semantic relationships of movie genres and tags by applying deep learning algorithms to match user preferences.

Our work is subdivided into two parts: the first part presents the state-of-the-art of the project, explaining recommendation systems in detail and machine learning. The second part presents the

proposed approach, including the steps involved in creating the model, evaluating the results using two different metrics and, finally, comparing our results with those of other models.

Keywords: Recommender system, Deep learning, DNN, Auto-encoder, prediction.

ملخص

تُستخدم أنظمة التوصية (RS) في العديد من المنتجات والخدمات مثل Google وAmazon وYouTube وFacebook وNetflix... في هذه الأطروحة، نهدف إلى إنشاء نظام توصية يعتمد على العلاقات الدلالية لأنواع الأفلام والعلامات من خلال تطبيق خوارزميات التعلم العميق للاستجابة لتفضيلات المستخدم.

ينقسم عملنا إلى جزأين، يعرض الجزء الأول حالة المشروع الفني الذي يشرح أنظمة التوصية بالتفصيل والتعلم الآلي. ويعرض الجزء الثاني النهج المقترح بما في ذلك خطوات إنشاء النموذج بالإضافة إلى تقييم النتائج باستخدام مقياسين مختلفين وفي النهاية، مقارنة نتائجنا بنتائج النماذج الأخرى.

الكلمات المفتاحية: نظام التوصية، التعلم العميق، شبكة DNN، Auto-encoder، التنبؤ.

Table des matières

Remerciement.....II

Dédicaces III

Résumé.....IV

Abstract IV

ملخص V

Introduction générale..... 01

Chapitre 1: **Systèmes de recommandation .....** 04

1.1. Introduction 04

1.2. Définition 04

1.3. Classifications des systèmes de recommandation 06

1.3.1 Classification classique 06

1.3.2 Classification de Su 2009..... 07

1.3.3 Classification de Rao et Talwar 2008 07

1.4 Types de recommandation..... 08

1.4.1 Système de recommandation basé sur le contenu 08

1.4.2 Système de recommandation basé sur filtrage collaboratif 11

1.4.3 Système de recommandation filtrage hybride..... 16

1.4.4 Autre type..... 18

1.5 Évaluation de system de recommandation 19

1.6 Méthodes d'évaluations..... 21

7.1 conclusion 22

Chapitre 2 : **Apprentissage automatique** 23

2.1. Introduction 24

2.2. Apprentissage automatique 24

2.2.1 L'apprentissage automatique selon la nature des données en entrée 24

2.2.2 L'apprentissage automatique selon le type de résultat 25

2.3 Réseaux de neurones et Apprentissage profond 28

2.3.1 Historique 28

2.3.2 Réseaux de neurones 28

2.3.3 Réseaux de neurones profonds 23

2.3.4 Les principales architectures des réseaux de neurones 34

a. Les réseaux de neurones convolutifs (CNN) 35

b. Les réseaux de neurones récurrents (RNN) 36

c. Les models LSTM (Long Short Term Memory) 37

d. Les modeles Multi-Layer Perceptron (MLP) 37

2.3.5 Avantages et inconvénients 39

2.4 Présentation de quelques travaux de systèmes de recommandation basés sur le Deep Learning 39

2.4.1. Système recommandation utilisant RNN 39

2.4.2. Système de recommandation utilisant Auto-Encoder 40

2.4.3. Système de recommandation hybride utilisant Deep Sémantique Model 40

2.4.4. Les systèmes de recommandation contextuels 41

2.5 Bibliothèques et plateformes de développement 41

2.5.1 Tensorflow 41

2.5.2 Keras 42
2.5.3 Sklearn..... 42
2.5.4 Anaconda..... 43
2.5.5 Google colab 43
2.6 Conclusion..... 43
Chapitre 3 : Modélisation 44
3.1 Introduction 45
3.2 Dataset (les données)..... 45
3.3 Architecture du Modèle..... 46
3.4 Conclusion..... 51
Chapitre4 : Expérimentations 52
4.1 Introduction 53
4.2 Métriques de l'évaluation 53
4.3 Evaluation Expérimentale 54
4.3.1 Evaluation des mesures 54
4.3.2 Evaluation des mesures basées sur la perte..... 56
4.4 Evaluation de l'Approche de Recommandation 59
4.4.1 Comparaison..... 59
4.4.2 Analyse des résultats 60
4.5 Conclusion..... 61
Conclusion générale & Perspectives 62
Références 63

Liste des figures

Figure 1.1 : CLASSIFICATION DES SYSTEMES DE RECOMMANDATION ISINKAYE..... 06

Figure 1.2 : CLASSIFICATION PRINCIPALE DES SYSTEMES DE RECOMMANDATIONS7.1 08

Figure 1.3 : Un système de recommandation basé sur le contenu 10

Figure 1.4 : UN SYSTEME DE RECOMMANDATION COLLABORATIF 12

Figure 1.5 : Exemple de recommandation basé sur le filtrage collaboratif..... 13

Figure 1.6 : METHODES DE FILTRAGE COLLABORATIF.....14

Figure 1.7 : Le système de recommandation hybride 16

Figure 2.1 : Illustratif de deep learning avec plusieurs couches 29

Figure 2.2 : Illustration d'un Réseau de Neurones..... 30

Figure 2.3 : Structure simplifié d'un neurone artificiel..... 31

Figure 2.4 : Architecture standard d'un réseau CNN 35

Figure 2.5: Architecture of Multi-Layer Perceptron 38

Figure 2.6: l'algorithme de MLP..... 38

Figure 3.1 : Architecture du modèle 46

Figure 3.2 : Prétraitement..... 48

Figure 3.3 : modèle de Traitement sans Auto-encodeur 49

Figure 3.4: Schéma du modèle avec Auto-encodeur..... 50

Figure 4.1 : Résultats des critères d'évaluation sans auto-encodeur 54

Figure 4.2 : les Résultats des critères d'évaluation avec auto-encodeur 55

Figure 4.3 : Analyse d'entrainement et validation des courbes de perte Modèle1..... 56

Figure 4.4: Analyse d'entrainement et validation des courbes de perte Modèle2..... 56

Figure 4.5: Analyse d'entrainement et validation des courbes de perte Modèle3..... 57

Figure 4.6 : Analyse d'entrainement et validation des courbes de perte Modèle4..... 57

Figure 4.7 : Analyse d'entrainement et validation des courbes de perte Modèle5..... 58

Figure 4.8 : Analyse d'entrainement et validation des courbes de perte Modèle6..... 58

Figure 4.9 : Comparaison entre nos modèles avec différents autres algorithmes 60

Liste des Tableaux

Tableau 1.1 : Exemples de quelques systèmes de recommandation existants. 05

Tableau 2.1 : Les principales fonctions d'activation existantes 32

Tableau 3.1 : Ensemble de données (ml-latest-small) 46

Tableau 4.1 : Résultats des critères d'évaluation sans auto-encodeur 54

Tableau 4.2 : les Résultats des critères d'évaluation avec auto-encodeur..... 55

Tableau 4.3 : Comparaison entre nos modèles avec d'autres algorithmes..... 60

Liste des acronymes

AE Auto-encoder.
ANN Artificial Neural Networks.
BP Back-Propagation
CNN Conventional Neural Network.
DNN Deep Neural Network.
DL Deep Learning.
FM Factorisation Matricielle.
HMAE High Mean Absolute Error.
IMDb Internet Movie Data base.
IB Items-Based.
KNN K-Nearest Neighbors.
MAE Mean Absolute Error.
ML Machine Learning.
MSE Mean Squared Error.
ReLu Rectified Linear Unit.
RN Réseau de neurone.
RNN Récurrent Neural Network.
IA Intelligence Artificielle
RMSE Root Mean Squared Error.
SVD Singular Value Decompositi
SVM Support Vector Machine.
MLP Perceptron Multicouche
GMF Generalized Matrix Factorization
SR System Recommendation

Introduction générale

Introduction générale

L'homme est naturellement enclin à se méfier rapidement, en particulier avec l'émergence d'Internet et la multitude de sources et d'informations. Par conséquent, il est devenu nécessaire de classer ces informations liées les unes aux autres pour faciliter l'accès et les fournir spécifiquement à l'utilisateur en fonction de ses préférences. Cela se fait en prévoyant ce que l'utilisateur préfère, ce qui rend les systèmes de recommandation extrêmement importants pour les propriétaires de sites Web pour attirer les utilisateurs le plus longtemps possible [1].

Le système de recommandation (SR) est un moyen de fournir du contenu qui pourrait être utile à l'utilisateur pour réduire le temps de recherche. Les recommandations sont personnalisées pour chaque utilisateur en fonction de son sexe, de l'âge, du pays et de son interaction avec le système utilisateur, y compris les commentaires sur des plateformes telles que Facebook ou les évaluations sur les plateformes de cinéma [2].

Plusieurs systèmes ont été proposés au fil du temps pour relever ce défi, dont la méthode approuvée sur le modèle, et cela inclut également l'utilisation de l'apprentissage profond pour construire ces systèmes. La méthode de numérisation collaborative s'appuie sur l'analyse du comportement et des préférences d'un groupe d'utilisateurs pour fournir des recommandations précises et pertinentes.

L'apprentissage profond est une branche spécialisée de l'intelligence artificielle qui provient du machine Learning et qui peut souvent être utilisé. Il s'agit d'un ensemble de méthodes d'apprentissage automatique qui ont permis de faire des progrès considérables dans l'intelligence artificielle au cours des dernières années [3].

Les algorithmes traditionnels n'atteignent plus les résultats souhaités, ce qui nous amène à nous poser la question suivante : l'utilisation de l'apprentissage profond pour modéliser de nouveaux systèmes de recommandation peut-elle améliorer les performances et obtenir de meilleurs résultats ? Pour tenter de

répondre à cette question, nous avons tout d'abord introduit un modèle de réseau neuronal en utilisant les données dont nous disposons. Cela nous permet de trouver les relations entre les entrées. La deuxième étape consiste à prédire de nouveaux résultats à partir des données brutes. Le modèle créé pour faire des recommandations à partir de ces données est le réseau neuronal DNN.

Notre mémoire est divisé en quatre (4) chapitres. Le premier et le second présentent une explication détaillée des algorithmes des systèmes de recommandation et d'apprentissage automatique. Le troisième chapitre présente la méthode proposée, y compris les étapes pour créer des modèles. Le dernier présente l'évaluation des résultats en utilisant différentes mesures, et enfin comparer nos résultats avec d'autres modèles de référence.

Chapitre 1 :

Les systèmes de recommandations

1.1 Introduction

Depuis le début des années 1990, le développement des technologies du web et des communications a facilité la génération d'initiatives visant à créer des opportunités pour la communication et le partage de l'information. Cette évolution nécessite de nouvelles technologies pour aider les utilisateurs à trouver ce qu'ils recherchent.

L'émergence des systèmes de recommandation vise à résoudre le problème de la surcharge d'informations et de choix et à fournir aux utilisateurs des ressources pertinentes en fonction de leurs préférences [4].

1.2 Définition des systèmes de recommandation :

Les systèmes de recommandation sont 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 et sont une forme spécifique de filtrage d'information [5].

Le système comprend généralement quatre composants : le modèle utilisateur, le modèle d'objet de recommandation, l'algorithme de recommandation et l'évaluation des recommandations [6].

Quelques exemples de systèmes de recommandation existants :

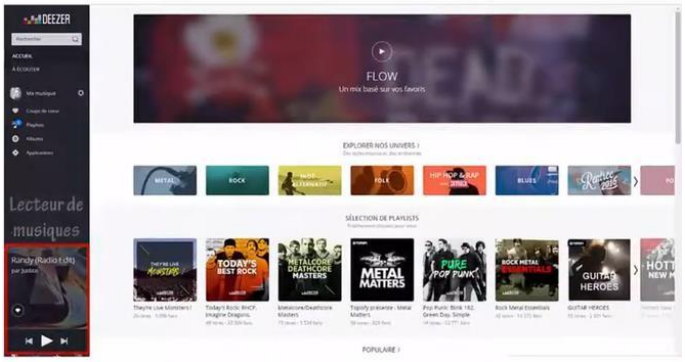
Système	Site web
Un système de recommandation de film. Exemple : Netflix	
Un système de recommandation De music. Exemple : Deezer	
Un système de recommandation de livre Exemple : Amazon	

Tableau 1.1 : Exemples de quelques systèmes de recommandation existants.

1.3 Classification des systèmes de recommandation

Nous pouvons classer les techniques de recommandation de plusieurs manières.

1.3.1 Classification classique

La classification classique est divisée en 3 types : filtrage basé sur le contenu, filtrage collaboratif (CF) et filtrage hybride [5]. Robin Burke [4] a proposé de considérer trois autres approches : la recommandation basée sur les données démographiques, la recommandation basée sur la connaissance et la recommandation basée sur l'utilité. Nous présentons, dans la figure 1.1 les approches basées sur le contenu, le filtrage collaboratif et les approches hybrides

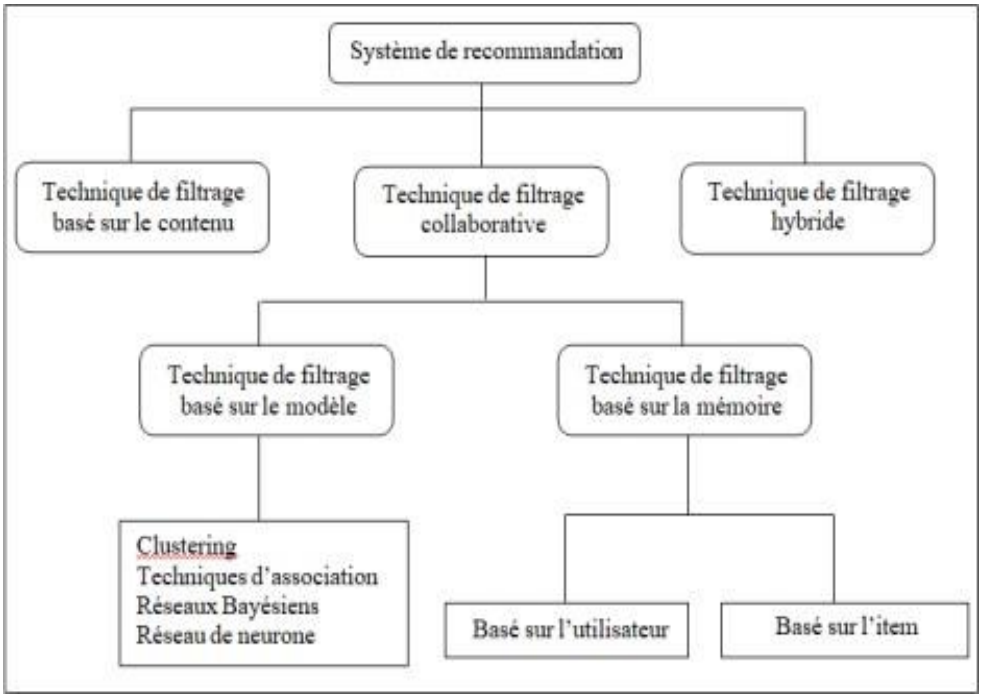


FIGURE 1.1 : CLASSIFICATION DES SYSTEMES DE RECOMMANDATION ISINKAYE [5].

1.3.2 Classification de Su 2009

Elle est utilisée dans les systèmes collaboratifs [7]. Ils proposent une sous-catégorie qui comprend les techniques hybrides, les classant comme des méthodes collaboratives hybrides. Divisent le filtrage collaboratif en trois catégories [7] :

• Méthode CF basée sur la mémoire : comme K-Nearest Neighbours.

• Les méthodes CF basées sur des modèles incluent des techniques telles que : le clustering, les réseaux bayésiens, la factorisation matricielle, les processus décisionnels de Markov...

• Hybride CF, qui combine les techniques de recommandation CF avec une ou plusieurs autres méthodes

1.3.3 La classification de Rao et Talwar 2008

C'est une classification en fonction de la source d'information utilisée [7].

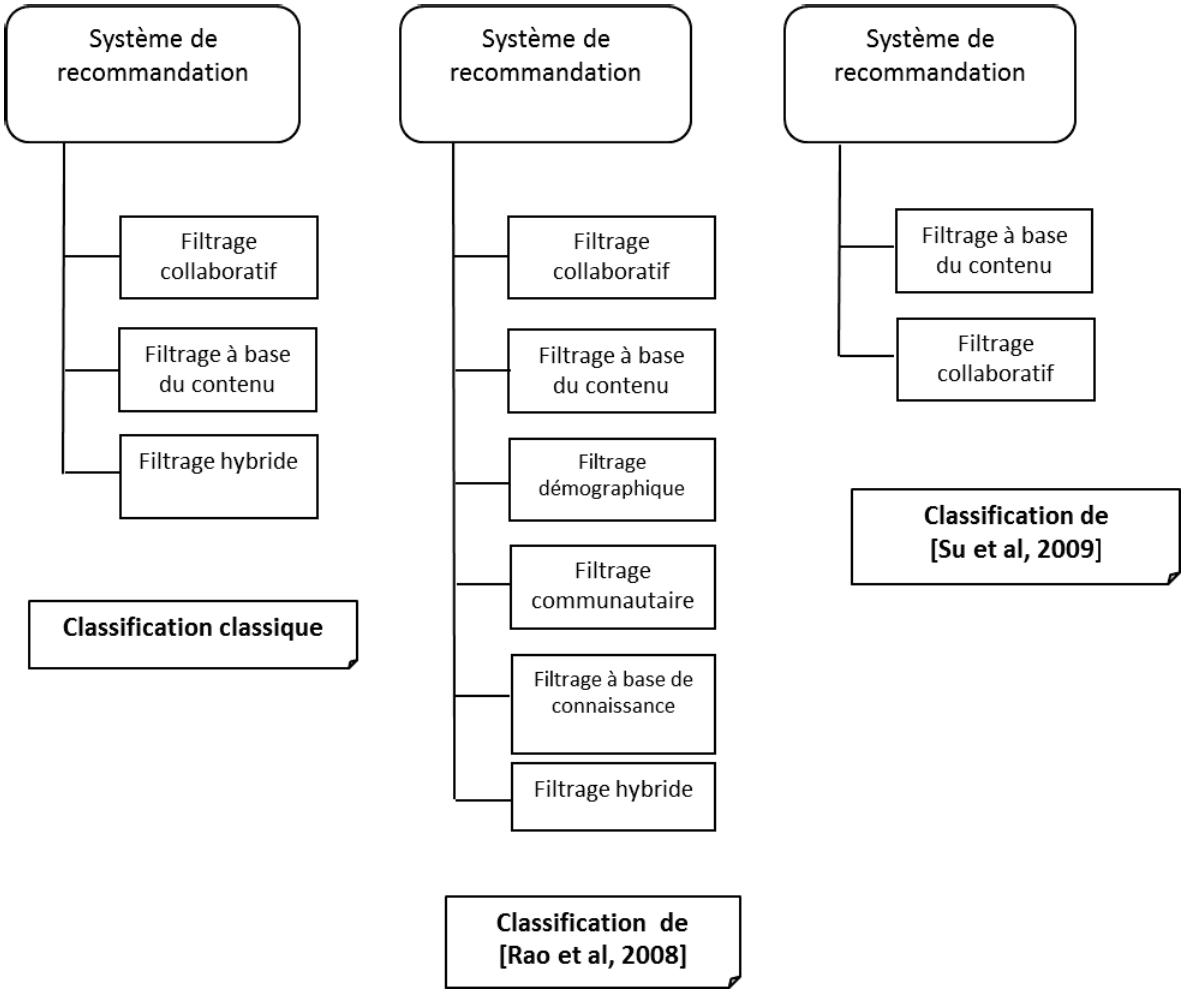


FIGURE 1.2 : CLASSIFICATION PRINCIPALE DES SYSTEMES DE RECOMMANDATIONS.

1.4 Types des systèmes de recommandation

1.4.1 Systèmes de recommandation basés sur le contenu (CBF)

Des recherches antérieures sur l'accès personnalisé à l'information s'étaient concentrées sur la combinaison des connaissances sur les éléments et des informations sur les préférences de l'utilisateur afin de localiser les éléments appropriés. Les premiers travaux cités dans [9] sur Grundy (recommandation des livres) et la méthode citée dans [10] (recherche d'informations) peuvent maintenant

être considérés comme les premiers exemples des systèmes de recommandation bien que le terme n'ait pas encore été inventé. Cette approche, en raison de sa dépendance à la source de connaissances sur le contenu, en particulier aux fonctionnalités des éléments, est devenue la recommandation basée sur le contenu.

La recommandation basée sur le contenu est étroitement liée à l'apprentissage automatique supervisé.

La correspondance entre l'ensemble de fonctionnalités et la fonction utilitaire de l'utilisateur doit également être bonne. Etant donné les informations sur le genre de film et sachant qu'un utilisateur aime « Spiderman » et « Black Panther », on peut en déduire une prédilection pour la science- fiction et donc recommander « Avengers : infinitywar ». Les recommandeurs basés sur le contenu font référence à de telles approches, qui fournissent des recommandations en comparant les représentations de contenu décrivant un élément à des représentations de contenu qui intéressent l'utilisateur. Ces approches sont parfois également appelées filtrage par contenu [11].

Le CBF utilise différents types de modèles pour trouver des similitudes entre les items afin de générer des recommandations significatives. Il pourrait utiliser des modèles probabilistes tels que le classifier de Naïve Bayes, les arbres décisionnels ou les réseaux de neurone [12] pour modéliser la relation entre différents items au sein d'un corpus. Ces techniques font des recommandations en apprenant le modèle sous-jacent à l'aide de techniques d'analyse statistique ou d'apprentissage automatique.

Pour les recommandations basées sur le contenu, 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.

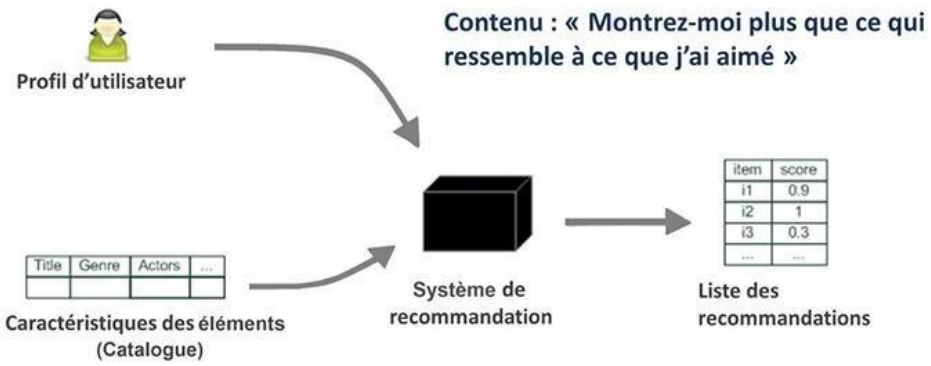


Figure 1.3 : Un système de recommandation basé sur le contenu [8]

Par exemple (cas de Netflix), on construit un profil d'un utilisateur « A » qui préfère les séries dont les genres préférés sont actions et romances. Et, on essaie de recommander des produits qui sont de la même section que « A » préfère. (On ne se base plus sur les opinions des autres utilisateurs) [13].

1.4.1.1 Avantages et inconvénients du filtrage basé sur le contenu

Les avantages :

- Recommander des items similaires à ceux que les utilisateurs ont aimés dans le passé.
- Le matching entre les préférences de l'utilisateur et les caractéristiques des items fonctionne aussi pour les données textuelles.
- Pas besoin de données sur les autres utilisateurs.
- Possibilité de faire des recommandations à des utilisateurs avec des goûts « uniques ».
- Possibilité de recommander de nouveaux items ou même des items qui ne sont pas populaires [14].

Les inconvénients :

- Tous les contenus ne peuvent pas être représentés avec des mots-clés (exemple : les images, …).

- ❖ Des items représentés par le même ensemble de mots-clés ne peuvent pas être distingués.
- ❖ Les utilisateurs avec des milliers d'achats/items sont un problème.
- ❖ Nouvel utilisateur : pas d'historique. Problème de création de profil [13].
- ❖ "Over-specialization" : limitation aux items similaires.
- ❖ Pour produire des recommandations précises, l'utilisateur doit fournir un « feedback » sur les suggestions retournées
- ❖ Entièrement basé sur les scores d'articles et de sujets d'intérêt : moins il y a de scores, plus l'ensemble de recommandations possibles est limité.
- ❖ Il faut faire attention à l'évolution des intérêts de l'utilisateur et les prendre en compte.
- ❖ Impossible d'exploiter les jugements des autres utilisateurs.
- ❖ Trouver les bons "features" n'est pas toujours facile.

1.4.2 Systèmes de recommandation basés sur le filtrage collaboratif

Le filtrage collaboratif (de l'anglais : collaborative filtering) regroupe l'ensemble des méthodes qui visent à construire des systèmes de recommandation utilisant les opinions et évaluations d'un groupe pour aider l'individu [15]. Ce modèle s'appuie sur les appréciations données par un ensemble d'utilisateurs sur un ensemble d'articles. Ces appréciations, traduites en valeurs numériques peuvent être des notes, des comptes d'achats effectués, des nombres de visites, etc. On distingue deux grandes approches de filtrage collaboratif. L'approche se référant aux utilisateurs Resnick and Varia [63] consiste à comparer les utilisateurs entre eux et à retrouver ceux ayant des goûts en commun, les notes d'un utilisateur étant ensuite prédites selon son voisinage. L'approche se référant aux articles Sarwar [64] consiste à rapprocher les articles appréciés par les mêmes personnes et à prédire les notes des utilisateurs en fonction des articles les plus proches de ceux qu'ils ont déjà notés. Dans un système du filtrage collaboratif, il faut que les utilisateurs fournissent des évaluations des items qu'ils ont déjà utilisés, sous forme des notes, pour constituer leurs profils. Il n'y a aucune analyse du sujet ou du contenu des objets à recommander. Ce type de système est très efficace au cas où le contenu des objets est complexe, compliqué ou impossible à analyser, l'utilisateur peut apercevoir divers domaines intéressants, car le

principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils. Parmi les avantages du filtrage collaboratif, les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des items tels que la diversité, la nouveauté, l'adéquation du public visé, ... Un problème du système collaboratif est que sa performance dépend beaucoup de la distribution des évaluations (notes) données par utilisateurs. Dans le cas où il y a plusieurs items qui ont été utilisés et évalués par très peu d'utilisateurs, ces items seraient recommandés très rarement, même si ces utilisateurs ont donné des notes très hautes pour ces items. Ce problème est connu comme le problème de parcimonie (sparsity problem). De la même façon, si dans le système il existe des utilisateurs qui ont des goûts très différents en comparaison avec les autres, le système ne peut pas trouver des similarités entre utilisateurs et donc ne peut pas donner des bonnes recommandations (voir figure 1.4).

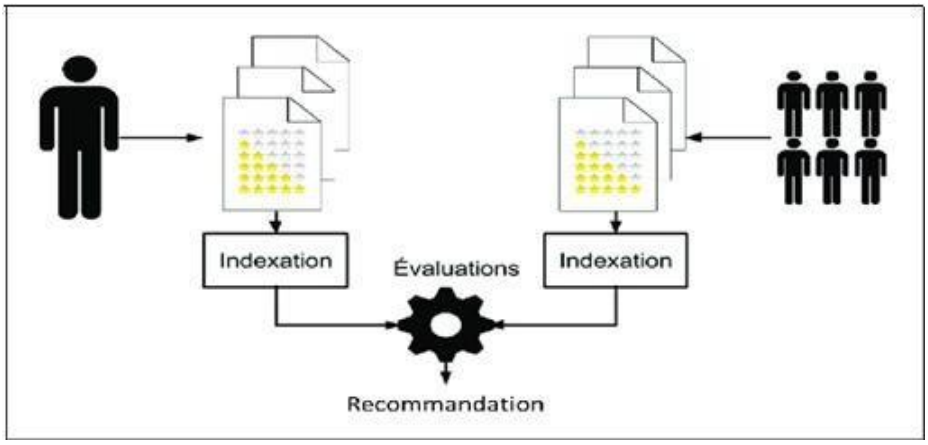


FIGURE 1.4 : UN SYSTEME DE RECOMMANDATION COLLABORATIF [11]

La figure 1.5 représente un tableau de films avec sur un axe les utilisateurs (ex : un groupe d'amis sur MovieLens) et sur un autre les films. Chaque cellule de la matrice contient l'avis donné par un utilisateur pour un film, la cellule vide signifie qu'il n'a pas d'avis particulier sur le film. Afin de prédire si Illyes apprécierait le film "Harry Potter" et probablement lui recommander ce film, on compare les Votes d'Illyes à ceux des autres utilisateurs choisis. On peut alors voir qu'Illyes et Imen ont des Votes identiques, et que Imen n'a pas aimé le film « Harry Potter », on pourrait alors prédire qu'Illyes n'aimera pas aussi ce film et de ne lui pas faire cette suggestion.

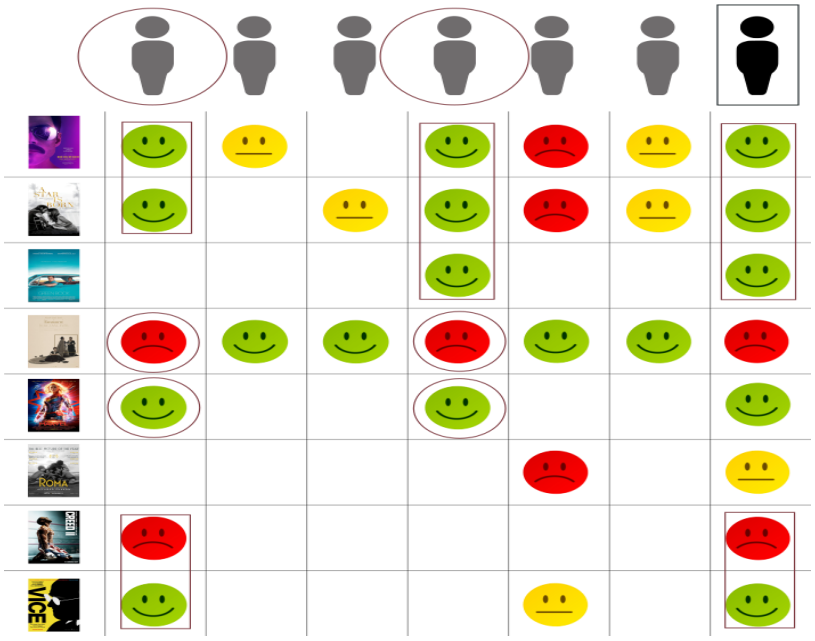


Figure 1.5 : Exemple de recommandation basé sur le filtrage collaboratif [11]

Le filtrage collaboratif est très utilisé par les systèmes vu ses avantages, parmi ces systèmes on peut citer : Amazon, Netflix, MovieLens, Jester, Citeseer, Ta-pestry, Phoaks, etc. L'exploitation des données disponibles dans un système de filtrage peut se faire de plusieurs manières. Ces méthodes sont classées en deux familles principales : les algorithmes basés mémoire et les algorithmes basés modèle [11].

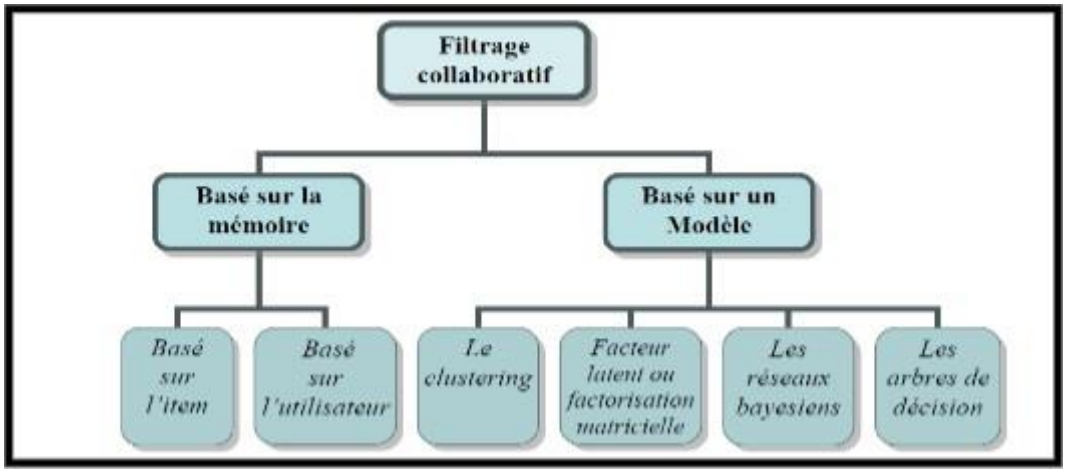


FIGURE 1.6 METHODES DE FILTRAGE COLLABORATIF

1.4.2.1 Méthode basée sur la mémoire

L'approche basée sur la mémoire exploite les appréciations des utilisateurs sur les items dans le but de générer les prédictions et identifier des utilisateurs voisins ayant, sur un même ensemble d'items, des appréciations similaires à celles de l'utilisateur actif.

Les données sont représentées sous forme d'une matrice « Utilisateur × Item » où les lignes représentent les utilisateurs U= {U_1, ...Um} et les colonnes constituent les items I= {i_1, ...i_n}.

1.4.2.2 Méthode basée sur un modèle

Ces méthodes ont été intégrées aux systèmes de recommandation pour améliorer et remédier aux problèmes des méthodes basées sur la mémoire. Les algorithmes basés sur le modèle se basent aussi sur les évaluations précédentes des utilisateurs, mais cette méthode ne calcule pas directement les prédictions, elle classifie les utilisateurs suivant des groupes ou d'apprendre les modèles à partir de leurs données. Généralement, les méthodes basées sur le modèle utilisent les techniques d'apprentissage automatiques pour la construction du modèle plusieurs [16].

- **Matrice de factorisation**

La Factorisation Matricielle (FM) est une méthode très utilisée en Analyse en Composantes Principales (ACP) qui consiste à décomposer une matrice en valeurs singulières (Singular Value

Décomposition : SVD) pour construire des facteurs orthogonaux deux à deux, ou bien en facteurs avec contraintes de non négativité sur leurs contenus [17].

Il existe deux principales méthodes de la factorisation matricielle.

- **Décomposition en valeurs singulières (Singular Value Décomposition (SVD)) :**

C'est la méthode la plus puissante de réduction de la dimensionnalité d'une matrice, avant d'utiliser le résultat de ce processus pour rapprocher les scores originaux. Cette décomposition est motivée par deux opérations faites dans l'analyse des données : découplage (séparation dans les composants indépendants pour faciliter l'analyse) et triage (Ordonnancements de contributions par leur importance ou capacité d'explication).

- **Analyse des composants principaux PCA** :

Une réalisation de l'approche de la factorisation (MF), définie comme une procédure statistique qui utilise une transformation orthogonale pour convertir un ensemble d'observation de variables éventuellement corrélées en un ensemble de valeur de variables linéairement non corrélées appelées composantes principales, tel que le nombre de variables d'origine est supérieur ou égal aux composants principaux.

- **Factorisation matricielle PMF :**

Un modèle linéaire probabiliste avec un bruit d'observation gaussienne. La matrice de préférence de l'utilisateur est représentée comme le produit des matrices d'objets et d'éléments de base inférieurs de filature dans la factorisation probabiliste de la matrice.

1.4.2.3 Avantages et inconvénients du filtrage collaboratif [71]

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

- ❖ Effet de surprise (trouver autre chose que ce que l'on cherchait).
- ❖ Non nécessité de la connaissance du domaine.
- ❖ Possibilité d'indexation de tout genre d'items.
- ❖ L'élimination du problème de là sur-spécialisation.
- ❖ La qualité des suggestions s'améliore avec le temps.

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

- ❖ Le démarrage à froid (un nouvel utilisateur qui n'a aimé aucun item ne peut pas recevoir de recommandation puisque le système ne connait pas ses goûts).
- ❖ La parcimonie (sparsity, le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous-ensemble des items disponibles).
- ❖ Le problème du mouton gris (gray sheep, n'auront pas beaucoup d'utilisateurs voisins. Il sera donc difficile de faire des recommandations pertinentes pour ce genre d'utilisateurs).

1.4.3 Systèmes recommandation le filtrage hybride [65][66]

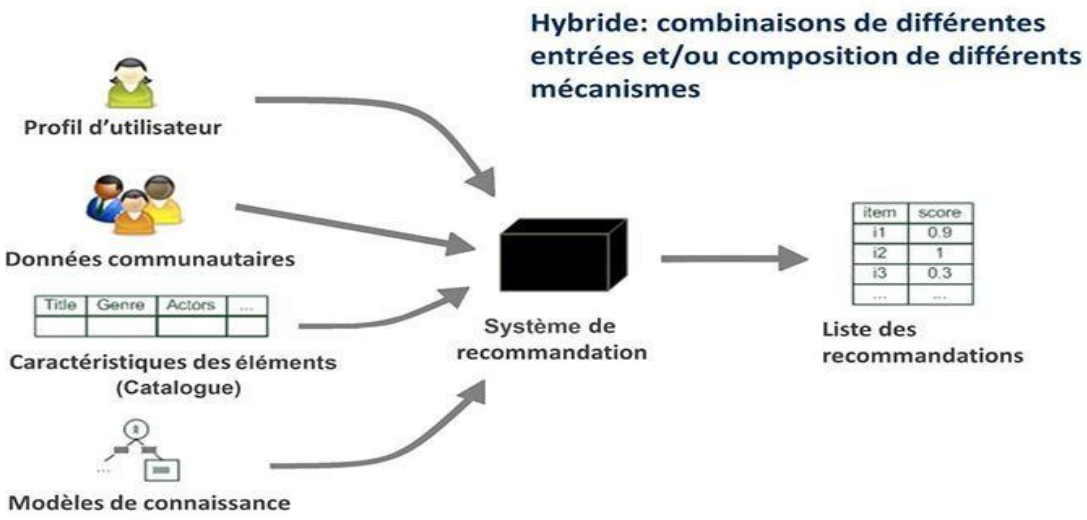


Figure 1.7 : Le système de recommandation hybride [11]

Constatant les avantages et les inconvénients de chacune des deux approches ci-dessus, on comprend que de nombreux systèmes reposent sur leur combinaison, ce qui en fait des systèmes de filtrage dits « hybrides ». En général, l'hybridation s'effectue en deux phases : (i) appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates, et (ii) combiner ces ensembles de recommandations préliminaires selon certaines méthodes telles que la pondération, la mixtion, la cascade, la commutation, ..., afin de produire les recommandations finales pour les utilisateurs.

Plus généralement, les systèmes hybrides gèrent des profils d'utilisateurs orientés contenu, et la comparaison entre ces profils donne lieu à la formation de communautés d'utilisateurs permettant le

filtrage collaboratif. La meilleure description des méthodes hybrides a été faite par Burke [5]. Alors, selon Burke on peut distinguer sept façons de combiner les méthodes traditionnelles :

- **Pondération (Weighted)**

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

- **Commutation (Switching)**

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

- **Technique mixte (Mixed)**

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

- **Combinaison de caractéristiques (Features combination)**

Dans un hybride basé sur la combinaison de caractéristiques, les données provenant de techniques collaboratives sont traitées comme une caractéristique, et une approche basée sur le contenu est utilisé sur ces données.

- **Cascade**

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

- **Augmentation de caractéristiques (Feature augmentation)**

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

- **Méta niveau (Meta-level)**

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

1.4.4 Autres types

1.4.4.1 Recommendation démographique

C'est une recommandation simple qui propose des items par rapport au profil démographique d'usager, Elle comporte à partager les usagers en plusieurs classes ou groupes par rapport aux informations démographiques telles que le sexe, l'âge, la profession, la localisation, la langue, le pays, ... Le principe de cette approche est que deux utilisateurs ayant évolué dans un environnement similaire partagent des goûts communs que deux utilisateurs ayant évolué dans des environnements différents et ne partageant donc pas les mêmes codes. De nombreux sites utilisent cette solution simple pour proposer une offre de contenu "personnalisé", par exemple, les utilisateurs sont redirigés vers un site Web particulier en fonction de leur langue ou de pays. Ces approches ont été très populaires dans la littérature de marketing, mais ont reçu peu d'attention dans le domaine des algorithmes de recommandation [67].

1.4.4.2 Recommandations à base de connaissances

Les recommandations sont générées en utilisant des connaissances spécifiques dont certaines caractéristiques d'items répondent aux préférences de l'usager. Généralement les systèmes à base de connaissances sont plus fiables par rapport à d'autres types de recommandation silles données limitées sont disponibles, à savoir, si le système ne peut pas compter sur l'existence d'UN historique de l'utilisateur. Mais si le système de la connaissance n'est pas conçu pour apprendre des notes ou des actions de l'utilisateur, le raisonnement à base des cas tire parti de la régularité du monde réel afin de

résoudre des problèmes en recherchant la solution d'un cas semblable rencontré et résolu dans le passé [18].

1.4.4.3 Le raisonnement à base de contraintes

Une recommandation à base de contraintes est un autre type de systèmes à base de connaissances. Elle exploite des bases de connaissances prédéfinies qui contiennent des règles explicites sur la façon de relier les exigences des clients avec des fonctionnalités d'item. Par exemple, un utilisateur peut être intéressé à acheter des produits avec un certain ensemble de caractéristiques et dans une gamme de prix spécifique [68].

1.4.4.4 Recommandation communautaire

L'idée de base est donc de dire que si des utilisateurs ont partagés des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans le futur. Le système propose des recommandations à partir des relations de l'usager avec ces amis dans le réseau social, et parfois cette recommandation dépend aussi de la valeur de confiance d'usager dans chacun de ses amis. L'exemple le plus connu de cette recommandation est la section des pages et des groupes qu'apparait dans la partie droite d'une page Facebook. L'importance décisionnelle du bouton « I Like » de Facebook a donné un succès croissant dont 55% des utilisateurs sont influencés par leurs amis [69].

1.5 Evaluation des systèmes de recommandation

1.5.1 Métriques d'évaluation du système recommandation

La qualité d'un algorithme de recommandation peut être évaluée à l'aide de différents types de mesure qui peuvent être la précision ou la couverture. Le type de métrique utilisé dépend du type de technique de filtrage. La précision est la fraction de recommandations correctes sur le total des recommandations possibles, tandis que la couverture mesure la fraction d'objets dans l'espace de recherche pour lesquels le système est en mesure de fournir des recommandations. Les métriques pour mesurer la précision des systèmes de filtrage des recommandations sont divisées en métriques de précision statistiques et d'aide à la décision. L'adéquation de chaque métrique dépend des caractéristiques de l'ensemble de données et du type de tâches que le système de recommandation effectuera [70].

1.5.1.1 Les mesures de précision statistique

Les mesures de précision statistique évaluent la précision d'une technique de filtrage en comparant les évaluations prévues directement avec l'évaluation réelle de l'utilisateur. L'erreur absolue moyenne (MAE) [19], l'erreur quadratique moyenne (RMSE) et la corrélation sont généralement utilisées comme mesures de précision statistique. MAE est le plus populaire et le plus couramment utilisé ; il s'agit d'une mesure de l'écart de la recommandation par rapport à la valeur spécifique de l'utilisateur. Elle est calculée comme suite [20] :

$$\mathrm{MAE} = \frac{1}{N} \sum_{u,i} |p_{u,i} - r_{u,i}| \tag{7}$$

Où $P_{u,i}$ est la note prédite pour l'utilisateur « u » sur l'élément « i », $r_{u,i}$ est la note réelle et N est le nombre total de notes sur l'ensemble d'articles. Plus le MAE est bas, plus le moteur de recommandation prédit avec précision les évaluations des utilisateurs. En outre, l'erreur quadratique moyenne (RMSE) est donnée par Cotter et al [21] comme suite :

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{u,i} (p_{u,i} - r_{u,i})^2} \tag{8}$$

L'erreur quadratique moyenne (RMSE) met davantage l'accent sur une erreur absolue plus grande, et plus le RMSE est bas, meilleure est la précision de la recommandation.

1.5.1.2 Les mesures de précision et d'aide à la décision

Les mesures de précision d'aide à la décision couramment utilisées sont le taux d'inversion, les erreurs pondérées, les caractéristiques de fonctionnement du récepteur (ROC) et la courbe de rappel de précision (PRC), la précision, le rappel et la mesure F. Ces métriques aident les utilisateurs à sélectionner des éléments de très haute qualité parmi l'ensemble d'éléments disponibles [21]. Les métriques considèrent la procédure de prédiction comme une opération binaire qui distingue les bons éléments de ceux qui ne le sont pas. Les courbes ROC sont très efficaces lors de l'exécution d'évaluations complètes des performances de certains algorithmes spécifiques. La précision est la fraction des éléments recommandés qui est réellement pertinente pour l'utilisateur, tandis que le rappel peut être défini comme

la fraction des éléments pertinents qui font également partie de l'ensemble des éléments recommandés [22]. Elles sont calculées comme suite :

$$\text{Precision} = \frac{\text{Correctly recommended items}}{\text{Total recommended items}} \tag{9}$$

$$\text{Recall} = \frac{\text{Correctly recommended items}}{\text{Total useful recommended items}} \tag{10}$$

La mesure F définie ci-dessous permet de simplifier la précision et le rappel en une seule métrique. La valeur résultante rend la comparaison entre les algorithmes et entre les ensembles de données très simple et directe.

$$F\text{-measure} = \frac{2\,PR}{P+R} \tag{11}$$

1.5.1.3 La couverture

Proportion d'éléments dans la base de données que le système est capable de recommander. Une couverture élevée indique que le système peut proposer des recommandations pour un large éventail d'éléments.

1.6 Méthodes d'évaluation

1.6.1 Evaluation online :

Les évaluations en ligne tirent parti des études d'utilisateurs, sauf que les utilisateurs sont souvent des utilisateurs réels dans un système pleinement déployé ou commercial. Cette approche est parfois moins susceptible à des préjugés du processus de recrutement, car les utilisateurs utilisent souvent directement le système dans le cours naturel des affaires. Et est le plus précis pour tester les performances à long terme du système. En outre, les évaluations en ligne peuvent recueillir les commentaires des utilisateurs concernant le système de recommandation. Selon Herlocker, ce type d'évaluation peut révéler les habitudes des utilisateurs, les besoins non satisfaits et les problèmes inattendus, évaluant efficacement les

objectifs centrés sur l'utilisateur tels que l'expérience utilisateur, la satisfaction et la conservation sur une longue période dans des conditions réelles [26][27].

1.6.2 Evaluation offline :

Les évaluations hors lignes utilisent des ensembles de données constitués d'actions des utilisateurs. Le processus simule des recommandations en cachant un sous-ensemble des actions des utilisateurs du dataset et en demandant au système de prédire ces actions cachées. Les systèmes de recommandation sont évalués en fonction de leur capacité à prédire ces interactions cachées, et les résultats de ces prédictions sont analysés en utilisant une ou plusieurs métriques pour mesurer la précision et l'efficacité [28][29].

1.7 Conclusion

Dans ce chapitre, nous avons présenté le problème de la recommandation et les composants principaux d'un système de recommandation. Nous avons étudié les différents types de systèmes de recommandation et les différentes techniques de filtrage en relevant les points forts et les points faibles de chacun.

Chapitre2 :

L’apprentissage automatique

2.1. Introduction :

L'apprentissage automatique est une discipline qui se concentre sur la construction de systèmes informatiques qui s'améliorent grâce à l'expérience et à la compréhension des lois fondamentales de la statistique, de l'informatique et de la théorie de l'information. Au cours des deux dernières décennies, l'apprentissage automatique est passé du statut de curiosité de laboratoire à celui de technologie pratique de l'intelligence artificielle, utilisée dans la vision artificielle, la reconnaissance vocale, le traitement du langage naturel et le contrôle des robots. Il a été appliqué dans divers secteurs, notamment les services aux consommateurs, le diagnostic des pannes et le contrôle de la chaîne logistique. Des méthodes d'apprentissage automatique ont également été mises au point pour analyser de manière novatrice les données expérimentales à haut débit [30].

2.2. L’apprentissage automatique :

Les algorithmes d'apprentissage automatique sont des processus informatiques qui utilisent les données d'entrée pour réaliser les tâches souhaitées sans être codés en dur. Ils sont "codés en douceur" et s'adaptent grâce à la formation, où des échantillons de données d'entrée sont fournis avec les résultats souhaités. L'algorithme s'optimise pour produire le résultat souhaité lorsqu'on lui présente des données d'entrée et se généralise à de nouvelles données. Cette formation est la partie "apprentissage" de l'apprentissage automatique et peut durer toute la vie. L'algorithme peut sélectionner et pondérer les données d'entrée, ajuster les paramètres numériques, organiser les voies de calcul et prédire les résultats [31].

L’apprentissage automatique est subdivisé selon la nature des données en entrée puis selon la nature des données en sortie :

2.2.1 L’apprentissage automatique selon la nature des données en entrée

- **Apprentissage supervisé :** L'apprentissage supervisé associe des données d'entrée à des étiquettes de classification connues, ce qui permet aux apprenants d'identifier des objets présentant des similitudes et des différences. Les apprenants qui réussissent peuvent reconnaître des objets non familiers, ce qui garantit qu'ils peuvent reconnaître des objets présentant des qualités fondamentales [31].

La Machine Learning est généralement utilisée pour l'apprentissage supervisé, où les algorithmes statistiques apprennent des modèles à partir de données historiques, qui comprennent des variables d'entrée et de sortie et des exemples d'apprentissage [32].

- **Apprentissage non supervisé :** C'est une technique d'apprentissage automatique qui utilise une structure des données non étiquetées.

  L'apprentissage non supervisé est classé en deux catégories d'algorithmes :

  - **Clustering** : Un problème de clustering est où nous voulons découvrir les regroupements inhérents dans les données, comme le regroupement des clients par le comportement d'achat [35].
  - **Association** : Un problème d'apprentissage de la règle d'association est où nous voulons découvrir des règles qui décrivent de grandes portions de vos données, comme les gens qui achètent X ont également tendance à acheter Y [35].

- **Apprentissage par renforcement :** L'algorithme apprend un comportement étant donné une observation. L'action de l'algorithme sur l'environnement produit une valeur de retour qui guide l'algorithme d'apprentissage.

2.2.2 L'apprentissage automatique selon le type de résultat [33][34]

- **Régression**

  La régression est utilisée pour comprendre la relation entre les variables fiables et indépendantes. En outre, il s'agit d'un type d'apprentissage supervisé qui apprend des ensembles de données étiquetées pour prédire la sortie continue pour différentes données dans un algorithme. On pense qu'il est largement utilisé dans les scénarios où la sortie doit être une valeur limitée, par exemple, la hauteur ou le poids, etc.

  Il existe deux types de régression qui sont :

- **Régression linéaire (Linear regression) :** Elle est utilisée pour identifier la relation entre deux variables, généralement utilisée pour faire des prédictions futures. En outre, la régression linéaire est subdivisée en fonction du nombre de variables indépendantes et dépendantes. Par exemple, s'il y a une variable indépendante ou dépendante, elle est appelée régression linéaire simple. Pendant ce temps, s'il y a deux ou plusieurs variables indépendantes et dépendantes, on l'appelle une régression linéaire multiple.
- **Régression logistique (Logistic regression) :** Elle est utilisée lorsque la variable dépendante est catégorique ou à des sorties binaires comme « oui » ou « non ». De plus, elle est utilisée pour résoudre des problèmes de classification binaire ; c'est pourquoi elle prédit des valeurs discrètes pour les variables.

➢ **Naïve Bayes**

Un algorithme naïf de Bayes est utilisé pour les grands ensembles de données. L'approche fonctionne sur le principe fondamental que chaque programme dans l'algorithme fonctionne indépendamment. Cela signifie que la présence d'une fonctionnalité n'aura pas d'impact sur l'autre. Généralement, il est utilisé dans la classification de texte, les systèmes de recommandation et autres.

➢ **Arbre de décision**

Un arbre de décision est un algorithme d'apprentissage supervisé unique qui ressemble structurellement à un diagramme de flux. Cependant, ils remplissent des rôles et des responsabilités fondamentalement différents.

Un arbre de décision se compose de déclarations de contrôle contenant des décisions et leurs conséquences. La sortie dans un arbre de décision concerne l'étiquetage des données imprévues. ID3 et CART sont quelques-uns des algorithmes d'arbre de décision populaires largement utilisés dans diverses industries.

- **Classification**

Il s'agit d'un type d'algorithme d'apprentissage supervisé qui assigne avec précision les données dans différentes catégories ou classes. Il reconnaît des entités spécifiques et les analyse afin de déterminer où ces entités doivent être classées. Certains des algorithms de classification sont les suivants :

- K-nearest neighbor
- Random forest
- Support vector machines
- Decision tree
- Linear classifiers

- **Neutral networks**

Ce type d'algorithme d'apprentissage supervisé est utilisé pour regrouper ou catégoriser les données brutes. En outre, il est utilisé pour trouver un motif ou interpréter des données sensorielles. Cependant, l'algorithme nécessite de nombreuses ressources de calcul.

- **Random Forest**

Un algorithme forestier aléatoire est souvent appelé une méthode d'ensemble parce qu'il combine différentes méthodes d'apprentissage supervisées pour conclure. En outre, il utilise de nombreux arbres de décision pour produire la classification des arbres individuels. En conséquence, il est largement utilisé dans les industries.

- **Clustering**

Un problème de clustering est où vous voulez découvrir les regroupements inhérents dans les données, comme le regroupement des clients par le comportement d'achat[35].

- **Association**

Un problème d'apprentissage de la règle d'association est où vous voulez découvrir des règles qui décrivent de grandes portions de vos données, comme les gens qui achètent X ont également tendance à acheter Y [35].

2.3 Réseaux de neurones et Apprentissage profond

2.3.1 Historique

En 1943 apparait le premier modèle de « neurone formel » proposé par Warren McCulloch et Walter Pitts. Il s'agit d'un neurone binaire, la sortie étant équivalente à 0 ou 1. C'est la première représentation informatique de cerveau humain. Puis c'est avec le "perceptron" en 1957 que l'on commence à parler de « réseau de neurones artificiels » [36].

Ensuite vient l'hiver de l'IA et il faudra attendre les années 1980 pour observer des progrès remarquables comme le « perceptron multicouche ». Le terme "deep learning" voit le jour grâce aux travaux du chercheur français Yann LeCun et à sa technique des réseaux de neurones convolutifs (réseaux de neurones inspirés par le cortex visuel des animaux). Mais le problème de ces nouveaux réseaux de neurones est la trop grande puissance de calcul nécessaire pour entrainer les modèles. La technologie sera donc délaissée [36].

C'est en 2012, à l'occasion du concours ImageNet Large Scale Visual Recognition Challenge, un défi de recherches informatiques organisé par l'université de Stanford, que le deep learning fait son grand retour, grâce au programme vainqueur qui s'appuie sur des techniques d'apprentissages profonds.

Aujourd'hui, la majorité des entreprises utilisant le machine learning s'appuie sur des techniques de deep Learning, quel que soit le domaine : images, texte, audio, … [37].

2.3.2 Réseaux de neurones [38]

Il s'agit d'un ensemble de cellules neuronales artificielles disposées en couches superposées les unes aux autres, avec une couche initiale et une couche finale. La couche initiale reçoit les données brutes et les traites avant de les transmettre ultérieurement à la couche suivante, et ainsi de suite jusqu'à ce que

nous obtenions la sortie de la couche finale. Certaines de ses applications pratiques les plus célèbres utilisées actuellement sont :

- La détection d'objets dans les images et les vidéos.
- La reconnaissance de la parole et des structures linguistiques.
- Le traitement d'images et la conversion de formats.
- La traduction automatique.

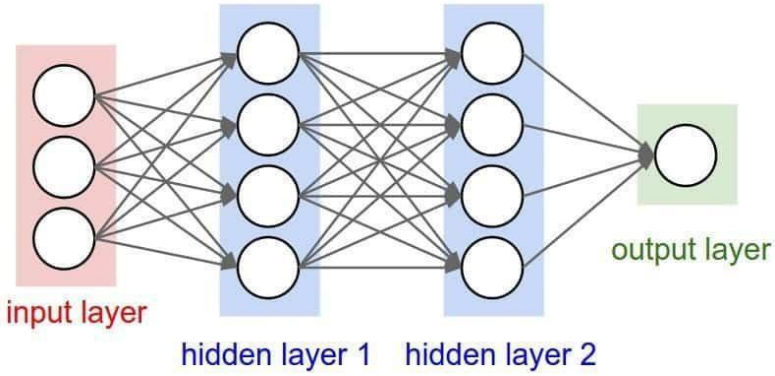


Figure 2.1 : Illustration d'un Réseau de Neurones

- **Embedding :**

L'intégration (Embedding) dans les réseaux de neurones permet de représenter les données d'entrée dans des dimensions réduites tout en préservant la valeur des données. Cette méthode a fait l'objet de développements importants et est essentielle pour l'apprentissage des représentations des sommets d'un réseau.

Les encastrements de réseaux neuronaux sont utiles pour surmonter les limites du codage traditionnel, pour permettre des tâches telles que la recherche des voisins les plus proches, pour servir d'entrée à d'autres modèles et pour faciliter les visualisations. Ils sont intuitifs, relativement simples à mettre en œuvre et constituent un outil efficace pour traiter les variables discrètes [42].

2.3.3 Réseaux de neurones profonds :

Il s'agit d'un ensemble d'algorithmes qui apprennent par couches, c'est-à-dire qui permettent à l'algorithme de créer une hiérarchie de concepts complexes à partir de concepts plus simples, c'est-à-dire qui permettent à l'algorithme de créer une hiérarchie de concepts complexes à partir de concepts plus simples [36][37]

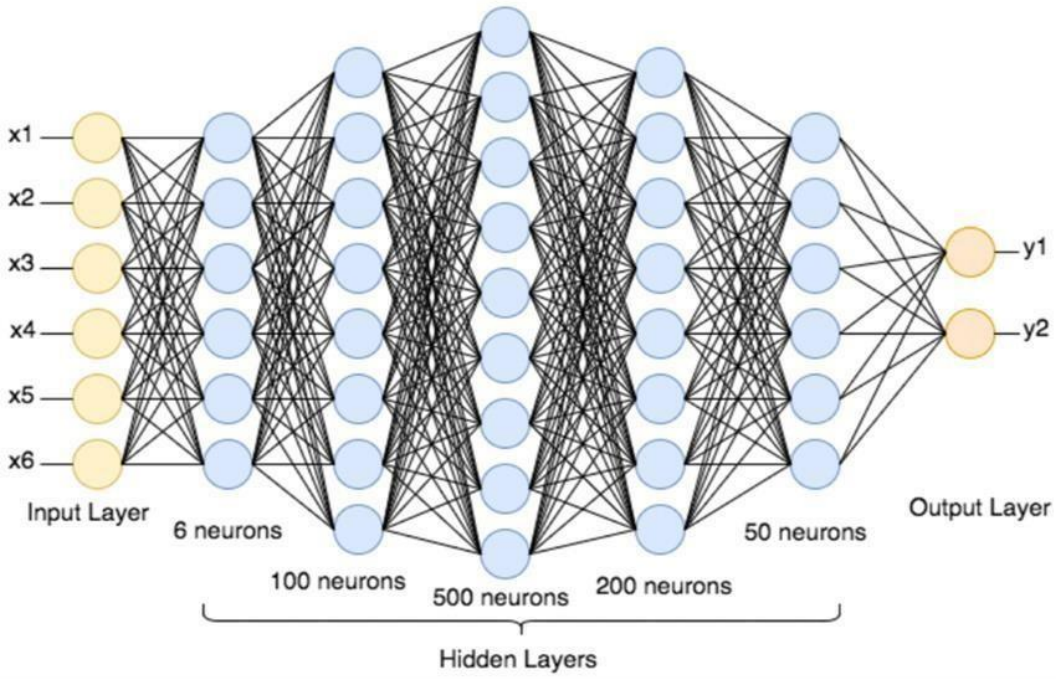


Figure 2.2 : Illustratif de deep learning avec plusieurs couches

Tout réseau neuronal artificiel est essentiellement constitué d'un ensemble de neurones artificiels et des connexions entre eux. Un neurone artificiel est simplement une entité possédant un ensemble d'entrées et une seule sortie. La tâche d'un neurone artificiel est de prendre toutes les valeurs de ses entrées, exécuter la fonction qui lui est assignée, puis transmettre le résultat en sortie.

Les connexions sont quelque peu similaires aux synapses entre les vrais neurones, reliant les sorties d'un neurone à l'entrée d'un autre, leur permettant de transmettre des valeurs et des résultats les uns aux autres. Chaque connexion possède un seul médiateur, qui est le poids. Ce poids est similaire à la force de connexion pour le signal ; ainsi, lorsque le nombre 10 passe à travers une connexion d'un poids de 0,5, il devient 5. Ces poids demandent aux neurones artificiels de répondre davantage à l'entrée ayant un poids

plus élevé et moins à celle ayant un poids plus faible. Ces poids sont ajustés lors de l'entraînement, permettant ainsi à l'ensemble du réseau neuronal de s'adapter et d'apprendre.

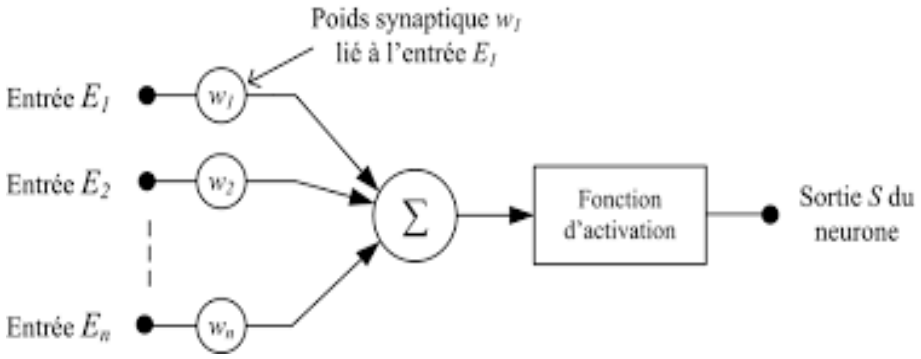


Figure 2.3 : Structure simplifié d'un neurone artificiel [23]

- **Fonctions d'activation [39] :**

Une fonction d'activation dans le contexte des réseaux de neurones est une fonction mathématique appliquée à la sortie d'un neurone, qui détermine si ce neurone doit être activé (c'est-à-dire s'il doit transmettre son signal de sortie à d'autres neurones) en fonction de son entrée pondérée.

En d'autres termes, la fonction d'activation introduit de la non-linéarité dans le réseau de neurones, ce qui lui permet d'apprendre des modèles complexes à partir des données en modifiant les poids et les biais des neurones lors de l'entraînement.

Les fonctions d'activation sont cruciales car elles permettent au réseau de neurones de modéliser des relations non linéaires entre les entrées et les sorties, ce qui est essentiel pour résoudre des problèmes complexes tels que la classification, la régression, la détection d'objets, ...

Les différentes fonctions d'activation telles que la sigmoïde, la tangente hyperbolique (tanh), ReLU (Rectified Linear Unit), Leaky ReLU, ELU, softmax, swish, ..., ont des propriétés différentes qui peuvent être adaptées en fonction du problème à résoudre et des caractéristiques des données.

Le tableau ci-dessous décrit les principales fonctions d'activation existantes :

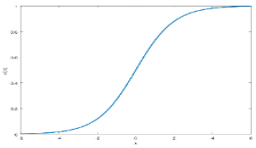
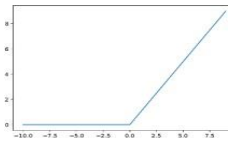
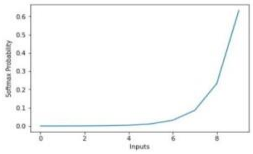
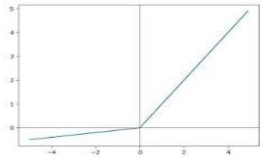
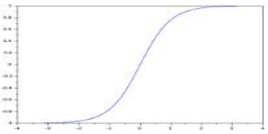
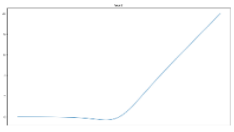
Nom	Description	Equation : f(x)	Graphe
Sigmoïde	C'est la fonction d'activation la plus largement utilisée car c'est une fonction non linéaire. La fonction sigmoïde transforme les valeurs dans la plage de 0 à 1.	$1 / (1 + e^{(-x)})$	
ReLu	Une fonction d'activation non linéaire que tous les neurones ne sont pas activés en même temps, efficace en termes de calcul mais n'est pas en mesure de traiter des entrées qui approchent de zéro ou négatives.	Max(0, x)	
Softmax	Utilisée pour les neurones de sortie. Il normalise les sorties pour chaque classe entre 0 et 1 et renvoie la probabilité que l'entrée appartient à une classe spécifique.	$e_{xi} \ / \ \Sigma k_{\ i=1}\ e^{xk}$ k : nombre de classes	
leaky relu	Petite pente positive dans sa zone négative, lui permettant de traiter des valeurs nulles ou négatives	0.01x, x < 0 x, x >= 0	
TanH	Centrée sur zéro, ce qui facilite la modélisation d'entrées fortement négatives, fortement positives ou neutres.	$2/(1 + e^{-2x^{-1}})$	
Swish	Nouvelle fonction d'activation découverte par les chercheurs de Google. Il fonctionne mieux que ReLu avec un niveau similaire d'efficacité de calcul.	x * sigmoid(x)	

Tableau 2.1 : Les principales fonctions d'activation existantes

- **Hyperparamètre de réseau neuronal**

Un hyperparamètre est un paramètre constant dont la valeur est définie avant le début du processus d'apprentissage. Les valeurs des paramètres sont dérivées par apprentissage. Des exemples d'hyperparamètres incluent le taux d'apprentissage, le nombre de couches cachées et la taille du lot. Les valeurs de certains hyperparamètres peuvent dépendre de celles d'autres hyperparamètres. Par exemple, la taille de certaines couches peut dépendre du nombre total de couches.

a. La différence entre les paramètres et les hyperparamètres [40]

- **Un paramètre de modèle :** Un paramètre est interne au réseau de neurones. Il va évoluer durant l'ensemble du processus d'entrainement, lors de la backpropagation.
- **Un hyperparamètre :** Un hyper-paramètre est à l'inverse d'un paramètre, externe au processus d'entrainement, il en définit des propriétés. Il reste statique durant l'entrainement (ex : batch size, epoch, learning rate, …).

b. Liste des hyperparamètres [41] :

- **Le batch size** est un hyperparamètre de la descente de gradient qui contrôle le nombre d'échantillons d'entraînement sur lesquels travailler avant que les paramètres internes du modèle ne soient mis à jour.
- **Le nombre d'époques** est un hyperparamètre de la descente de gradient qui contrôle le nombre de passes complètes à travers l'ensemble de données d'entraînement.
- **Taux d'apprentissage** : Facteur multiplicatif appliqué au gradient afin de faire varier le gain du gradient.
- **Abandon :** Améliore les performances des réseaux de neurones sur les tâches d'apprentissage supervisé en obtenant des résultats de pointe sur de nombreux ensembles de données de référence
- **Fonction d'activation :** Une fonction d'activation est une équation mathématique qui détermine la sortie de chaque élément dans le réseau neuronal.
- **Algorithme d'optimisation :** Une procédure mathématique qui permet à une fonction d'obtenir la valeur maximale ou minimale. Par exemple la descente de gradient.
- **Élan** : Calcule une moyenne glissante pondérée exponentiellement des gradients au fil du temps.

c. Les méthodes de réglage des hyperparamètres [43]

Dans une expérience de réseau de neurones, on essaye généralement de nombreuses valeurs possibles des hyperparamètres et on choisit ce qui fonctionne le mieux. Afin d'évaluer le succès de différentes valeurs, on convertit le réseau à l'aide de chaque ensemble d'hyperparamètres et le test par rapport à notre ensemble de validation. Si notre ensemble d'apprentissage est petit, on peut utiliser la validation croisée en divisant l'ensemble d'apprentissage en plusieurs groupes, en entraînant le modèle sur chacun des groupes puis en le validant sur les autres groupes. Voici les méthodes courantes utilisées pour régler les hyperparamètres :

- **Recherche dans la grille** :

La stratégie d'optimisation des hyperparamètres la plus couramment utilisée est une combinaison de recherche sur la grille (GS) et de réglage manuel. Cela est dû à sa simplicité, à l'absence de frais techniques et à sa fiabilité dans les espaces de faible dimension. La GS souffre cependant de la malédiction de la dimensionnalité, ce qui la rend déconseillée pour l'optimisation de nombreux hyperparamètres en raison de la croissance exponentielle des valeurs conjointes.

- **Recherche aléatoire :**

Recherche aléatoire est une méthode qui tire des échantillons de l'espace des paramètres, donne de meilleurs résultats que l'optimisation GS ou hyperparamétrique, nécessite moins de ressources de calcul et peut être arrêtée à tout moment.

- **Optimisation bayésienne** : L'optimisation bayésienne est un processus itératif qui calcule une distribution postérieure de fonctions pour décrire une fonction objective, en évaluant des hyperparamètres à partir de résultats antérieurs, en construisant un modèle de probabilité de substitution et en l'actualisant.

2.3.4 Les principales architectures des réseaux de neurones

Un réseau de neurones peut prendre des formes différentes selon l'objet de la donnée qu'il traite et selon sa complexité et la méthode de traitement de la donnée.

Les architectures ont leurs forces et faiblesses ce qui les rend plus en adéquation pour une tâche donnée, elles peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial, déterminer l'objectif à atteindre est le meilleur moyen de choisir l'approche la plus adéquate en fonction de ses avantages et surtout de ses contraintes.

Nous présentons dans ce qui suit les principales architectures des réseaux de neurones.

a. Les réseaux de neurones convolutifs (CNN) [44]

Le Réseau de Neurones Convolutif (CNN), également appelé ConvNet, est un type de Réseau de Neurones Artificiels (ANN), qui possède une architecture profonde à propagation avant et une capacité de généralisation incroyable par rapport à d'autres réseaux avec des couches Fully-Connected (FC). Il peut apprendre des caractéristiques hautement abstraites des objets, en particulier des données spatiales, et les identifier de manière plus efficace.

Un modèle CNN profond se compose d'un ensemble fini de couches de traitement qui peuvent apprendre différentes caractéristiques des données d'entrée (par exemple, une image) avec plusieurs niveaux d'abstraction. Les couches initiales apprennent et extraient les caractéristiques de haut niveau (avec une abstraction plus faible), et les couches plus profondes apprennent et extraient les caractéristiques de bas niveau (avec une abstraction plus élevée). Le modèle conceptuel de base de CNN est illustré dans la figure 2, avec différents types de couches décrits.

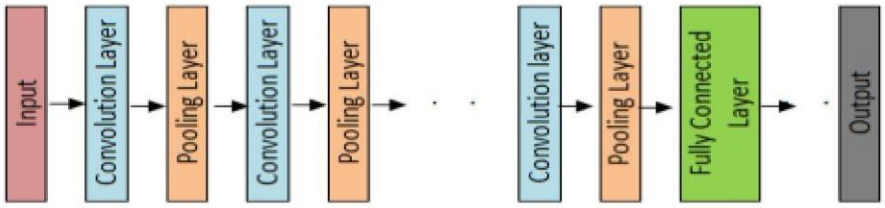


Figure 2.4 : Différents types de couches

- **Couche Convolution** : Il s'agit de l'élément le plus important de toute architecture de réseau neuronal convolutionnel. Il effectue une série de calculs sur les images d'entrée afin d'en extraire les caractéristiques.
- **Couche Pooling** : Il est utilisé pour agréger les caractéristiques résultant des opérations de convolution, car il réduit leur dimensionnalité et retient les caractéristiques les plus dominantes.
- **Couche Flatting :** Convertit les données en données unidimensionnelles (plates)

- **Couche Fully-Connected :** Dernière couche du réseau neuronal, Les couches FC reçoivent l'entrée de la dernière couche convolutive ou de mise en commun, qui se présente sous la forme d'un ensemble de métriques (cartes de caractéristiques). Ces métriques sont traitées pour créer un vecteur qui est ensuite introduit dans la couche FC pour générer la sortie finale du CNN.

b. Les réseaux de neurones récurrents (RNN) [45]

Un réseau neuronal récurrent est utile pour reconnaître la prochaine séquence de données (prédire la prochaine valeur d'une série) et est similaire à un réseau neuronal traditionnel, à l'exception du cas de la mémoire, et la plupart de ses utilisations concernent les données séquentielles.

Les RNN sont notamment utilisés pour :

- **La modélisation linguistique et génération de texte** : les modèles fonctionnent en prédisant le caractère suivant, le mot suivant ou la phrase suivante les plus appropriés dans une chaîne de texte.
- **La traduction automatique** : le texte dans la langue source est saisi par lots et le modèle tente de générer le texte correspondant dans la langue cible.
- **La reconnaissance vocale** : l'entrée est constituée de signaux acoustiques analysés à partir d'un enregistrement audio ; le modèle produit l'élément phonétique de langue le plus probable pour chaque partie de l'enregistrement.
- **La génération de légendes d'image** : l'entrée est une image et le modèle identifie les éléments de l'image et génère du texte qui la décrit.
- **La détection d'anomalies de séries chronologiques** : l'entrée est une série de données séquentielles, comme une série d'événements dans un incident de cyber sécurité potentiel. Le modèle prédit si les données sont une anomalie par rapport au comportement normal.
- **Balisage vidéo** : l'entrée est une série d'images vidéo et le modèle génère une description textuelle de chaque image de la vidéo.

c. Les models LSTM (Long Short Term Memory) [46]:

Pour les architectures RNN standard, l'éventail des contextes accessibles est limité.

Le problème est que l'influence d'une entrée donnée sur la couche cachée, et donc sur la sortie du réseau, diminue ou augmente de façon exponentielle au fur et à mesure qu'elle passe par les connexions récurrentes du réseau.

La solution la plus efficace à ce jour est l'architecture LSTM Hochreiter et Schmidhuber, 1997[46].

L'architecture LSTM consiste en un ensemble de sous-réseaux connectés de manière récurrente, appelés blocs de mémoire. Ces blocs peuvent être considérés comme une version différenciable des puces de mémoire d'un ordinateur numérique. Chaque bloc contient une ou plusieurs cellules de mémoire autoconnectées et trois unités multiplicatives qui fournissent des analogues continus des opérations d'écriture, de lecture et de réinitialisation pour les cellules.

d. Les modeles Multi-Layer Perceptron (MLP) [47]

Il s'agit d'un réseau neuronal couramment utilisé, basé sur l'apprentissage supervisé, dans lequel l'information circule dans une seule direction et ne comporte pas de boucles. L'objectif principal est de trouver la fonction optimisée f qui associe l'entrée à la sortie souhaitée et d'apprendre la valeur de biais optimisée (θ) pour cette fonction. L'apprentissage s'effectue dans le MLP à l'aide d'un algorithme de rétropropagation en ajustant les poids de connexion lorsqu'il y a un écart entre la sortie attendue et la sortie réelle. Leurs principales applications sont la résolution de problèmes d'optimisation dans les domaines de la finance, du transport, de la facilité et de l'énergie.

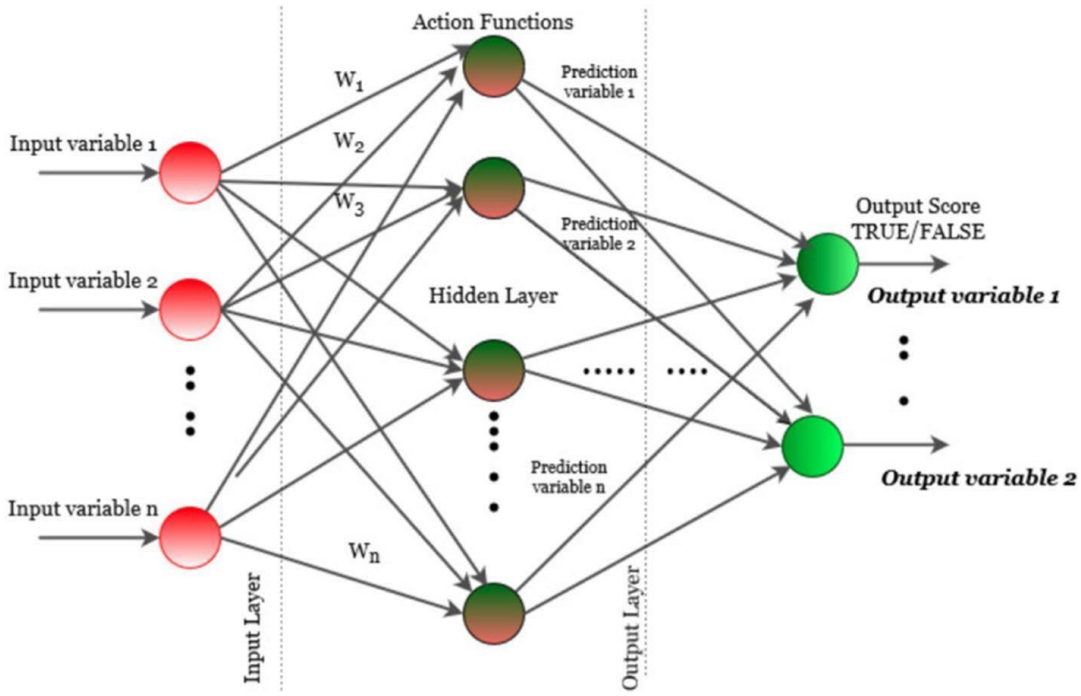


Figure 2.5 : Architecture of Multi-Layer Perceptron

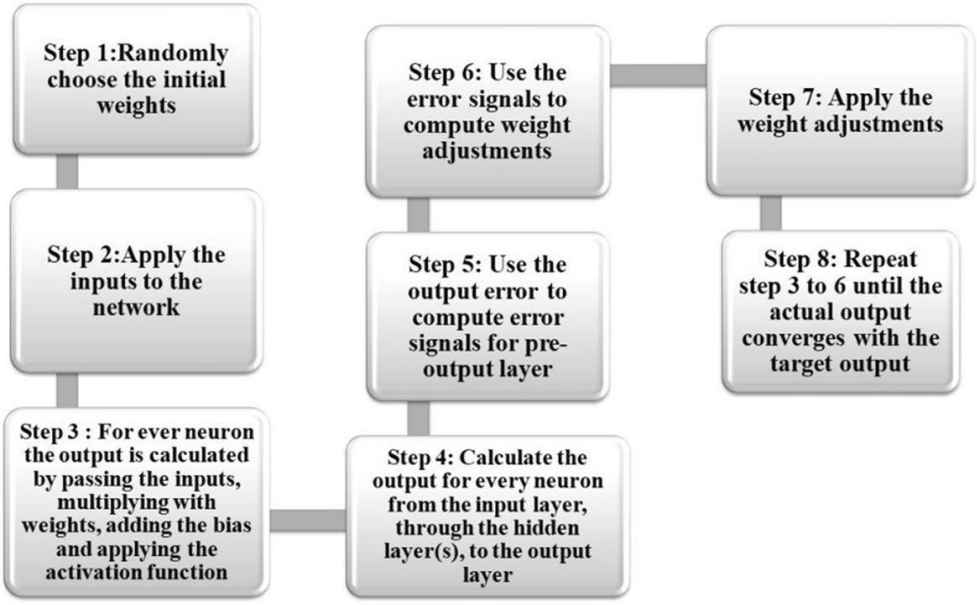


Figure 2.6 : Etapes de l'algorithme MLP

2.3.5 Avantages et inconvénients [48][49]

Les réseaux de neurones sont largement utilisés ces dernières années dans beaucoup de domaines car ils présentent plus d'avantages que d'inconvénients. Nous citons ses avantages et inconvénients comme suit :

- **Avantage :**

- Précision souvent élevée.
- Robuste, marche lorsque le jeu d'apprentissage contient des erreurs.
- Permet de résoudre des problèmes complexes par un apprentissage automatique (pas besoin de définir manuellement les règles de l'IA comme dans la méthode classique).
- La sortie peut être une valeur discrète, continue, ou un vecteur de plusieurs valeurs discrètes ou continues.
- S'adapte bien aux problèmes qui nécessitent des niveaux d'abstractions élevée qui sont difficiles à décrire explicitement avec la méthode classique (coder tous les traitements et les conditions).

- **Inconvénients :**

- Apprentissage long (dû au nombre de paramètres souvent élevés).
- Difficile de comprendre le modèle.
- Non évidence du choix des hyperparamètres.

2.4 Présentation de quelques travaux de systèmes de recommandation basés sur le Deep Learning :

Nous présentons dans cette partie quelques travaux de l'état de l'art basé sur l'apprentissage profond.

2.4.1. Système recommandation utilisant RNN :

Les utilisateurs et les items sont représentés à l'aide de vecteurs intégrés (Embeddings) qui capturent les caractéristiques de base de chacun d'entre eux et la séquence d'interactions entre les utilisateurs et les items est introduite dans le réseau, en l'occurrence les évaluations, et le réseau neuronal

maintient un état caché qui change à chaque pas de temps en fonction de la nouvelle interaction introduite et, à chaque repli, l'état est mis à jour à l'aide d'un algorithme de rétropropagation (BPTT- Bac Propagation Though Time), ce qui signifie que l'état actuel dépend des interactions précédentes ainsi que de l'interaction actuelle, En bref, RNN est utilisé pour capturer la coévolution entre les préférences des utilisateurs et les items. L'idée est que les préférences des utilisateurs évoluent dans le temps et peuvent être influencées par le matériel avec lequel ils ont interagi précédemment [50].

2.4.2. Système de recommandation utilisant Auto-Encoder :

C'est un type de réseaux neuronaux conçus pour apprendre une représentation de basse dimension des données, puis reconstruire les données d'origine. Il se compose de deux parties principales :

Encodeur : convertit les données originales en une représentation à faible dimension.

Décodeur : tente de reconstruire les données originales à partir de la représentation comprimée.

Dans ce cas, une matrice d'interaction R est créée, qui contient les interactions de chaque utilisateur avec différents items, soit au niveau de l'utilisateur, soit au niveau de l'item, où l'encodeur prend les interactions de l'utilisateur avec différents items ou vice versa et les convertit en une représentation comprimée, puis le décodeur reconstruit les données originales, et ce modèle est formé à l'aide de différents algorithmes. Après l'entraînement, le modèle peut prédire les interactions inconnues (comme les évaluations futures des utilisateurs) [51].

2.4.3. Système de recommandation hybride utilisant Deep Sémantique Model :

Le modèle sémantique profond, composé de deux branches de réseaux neuronaux profonds ou de leurs versions convolutées, identifie et prédit les documents cibles susceptibles d'intéresser les utilisateurs qui lisent les documents sources. Le modèle appris observe, identifie et détecte les signaux naturels d'intérêt dans les transitions de clics entre les documents source et cible dérivés des journaux des navigateurs web. L'intérêt est modélisé à l'aide de réseaux neuronaux profonds qui cartographient les paires de documents source-cible en vecteurs de caractéristiques dans un espace latent, formés sur les transitions de documents en vue d'un « contexte » et d'un « centre d'intérêt » facultatif des documents source et cible. Les paramètres du réseau sont appris pour minimiser les distances entre les documents

sources et leurs cibles « intéressantes » correspondantes dans cet espace. Le modèle d'intérêt qui en résulte peut-être utiliser, notamment pour la recherche contextuelle d'entités, la mise en évidence automatique de textes, l'extraction préalable de documents susceptibles de présenter un intérêt, la recommandation automatisée de contenu et le placement automatisé de publicités [52].

2.4.4. Les systèmes de recommandation contextuels :

Les systèmes de recommandation contextuels (Context Aware Recommendation Systems) fonctionnent en prenant en compte les facteurs contextuels entourant l'utilisateur à un moment donné, tels que la localisation géographique, le temps, la météo, l'humeur, ... Ces facteurs contextuels aident à déterminer les préférences de l'utilisateur de manière plus précise et à fournir des recommandations adaptées au contexte actuel de l'utilisateur.

User × Item × Context → Rating

Lorsque le contexte est intégré dans le processus de recommandation, le modèle de prédiction est étendu pour inclure le contexte en plus de l'utilisateur et de l'élément. Ainsi, la fonction de recommandation est représentée de manière multidimensionnelle où le système est orienté vers la recommandation d'éléments adaptés au contexte actuel.

Les défis des systèmes de recommandation contextuels incluent la compréhension du sens du contexte, l'identification des caractéristiques contextuelles importantes, et l'apprentissage et la prédiction des contextes manquants en fonction des connaissances disponibles. En intégrant le contexte de manière efficace et précise, les systèmes de recommandation contextuels peuvent améliorer leur précision et leur efficacité dans la fourniture de recommandations adaptées aux utilisateurs dans différents contextes [53].

2.5 Bibliothèques et plateformes de développement

2.5.1 Tensorflow

TensorFlow est une bibliothèque open source développée par l'équipe de Google en 2011. Elle implémente des méthodes d'apprentissage automatique basées sur les principes des réseaux neuronaux profonds [54]. Initialement, il s'appelait DistBelief, puis le code source de DistBelief a été modifié et l'outil est devenu une bibliothèque basée sur des applications. En 2015, il a été renommé TensorFlow et

Google l'a rendu open source. Depuis, plus de 21000 modifications ont été apportées par communication, pour atteindre la version 1.0 en février 2017. TensorFlow est une boîte à outils permettant de résoudre facilement des problèmes mathématiques extrêmement complexes. Il permet aux chercheurs de développer des architectures d'apprentissage expérimental et de les traduire en logiciels, où les calculs sont représentés sous forme de graphiques de flux de données, les nœuds du graphique représentant des opérations mathématiques. Chaque connexion entre ces nœuds est une flèche pour les données multidimensionnelles communiquées entre eux : les tenseurs. TensorFlow est multiplateforme. Il fonctionne sur presque tous les GPU et CPU, y compris les plates-formes mobiles et embarquées, et même sur les unités de traitement tensoriel (TPU), qui sont du matériel spécialisé utilisé pour effectuer des calculs tensoriels.

2.5.2 Keras

Keras est une bibliothèque open source écrite en Python qui vous permet d'interagir avec des réseaux de neurones profonds et des algorithmes d'apprentissage automatique [55]. Il s'agit d'une bibliothèque Python qui encapsule l'accès aux fonctionnalités fournies par plusieurs bibliothèques d'apprentissage automatique. Elle s'intègre à un niveau bas avec des outils d'apprentissage profond (notamment TensorFlow), ce qui nous permet d'implémenter toutes les fonctionnalités souhaitées.

2.5.3 Sklearn

Ce projet a été lancé en 2007 sous la forme d'un projet d'été de Google pour la création de sites Web par David Coronaboo. Plus tard cette année, Matthew Brothers a commencé à travailler sur ce projet dans le cadre de sa thèse.

Scikit-Learn est un projet social, développé par un grand groupe de personnes, dans le monde entier. Il s'agit d'une bibliothèque open source de la langue de programmation Python qui utilise dans l'apprentissage de ces caractéristiques et qui contient de nombreux algorithmes pour l'enseignement quotidien (SVM, Random Forest, Linear Regression, PCA, K-Means), demandes de traitement des données (fonction transformer), la séparation, la vérification et bien plus encore [56].

2.5.4 Anaconda

Anaconda Python est une plate-forme libre et open-source qui permet d'écrire et d'exécuter du code dans le langage de programmation Python. Il est réalisé par continuum.io, une entreprise spécialisée dans le développement de Python. La plate-forme Anaconda est le moyen le plus populaire d'apprendre et d'utiliser Python pour l'informatique scientifique, la science des données et l'enseignement automatique. Il est utilisé par plus de trente millions de personnes dans le monde entier et est disponible pour Windows, Mac-OS et Linux [57].

Anaconda simplifie le déploiement et la gestion des paquets. Il est également livré avec un grand nombre de bibliothèques / paquets utilisés dans les projets. Il est libre et open-source, n'importe qui peut contribuer à son développement [57].

2.5.5 Google colab

Colab (Google Colaboratory) est un service gratuit de notebooks Jupyter qui permet d'exécuter du code dans le cloud sans aucune configuration et qui permet d'accéder gratuitement à des ressources informatiques, y compris des GPU et des TPU. Colab est particulièrement populaire pour l'apprentissage automatique, à la data science et à l'enseignement [58].

2.6 Conclusion

L'apprentissage automatique et l'apprentissage profond sont les fondements de l'intelligence artificielle, l'un étant basé sur des données structurées et guidé par elles, l'autre ne l'étant pas et étant efficace pour travailler avec des données de haute dimension, ainsi que les réseaux neuronaux artificiels qui sont d'une grande importance pour le développement de ce domaine.

Nous avons vu dans ce chapitre qu’est-ce que le deep learning et ses domaines d’application, avec les notions importantes qui sont en relation avec (définition, Architectures, etc.), les réseaux de neurones artificiels et réseaux de neurones profonds.

Chapitre 3 :

Modélisation

3.1 Introduction

Les mécanismes de modélisation permettent de représenter les systèmes informatiques pour obtenir des solutions adaptées. Le principe est de faire abstraction de la réalité afin de bien comprendre le système à créer et ainsi mieux comprendre la notion de travail, son approche et les méthodes utilisées.

L'objectif de ce système de recommandation est de prédire les évaluations des utilisateurs pour divers items (films) en utilisant un modèle basé sur un Deep Neural Network (DNN). Les entrées du modèle incluent les utilisateurs, les items, les évaluations, les tags et les genres.

Dans ce chapitre, nous présentons l'architecture du modèle et détaillons chaque partie. Nous préparerons (prétraiterons) les données puis nous essayons d'extraire une sémantique entre ces données, et par la suite nous utilisons un DNN pour prédire les évaluations (ratings). La dernière étape consiste en une recommandation utilisant une méthode nouvellement proposée et différente des méthodes traditionnelles afin de recommander une liste de ressources avec des scores à l'utilisateur.

3.2 Dataset (les données)

L'ensemble de données ml-l-latest-small est un ensemble de données de films souvent utilisé pour tester et évaluer les systèmes de recommandation de films. Il s'agit d'un sous-ensemble du jeu de données MovieLens, qui contient des informations sur les évaluations des films par les utilisateurs. Cet ensemble de données décrit l'activité de notation 5 étoiles et de marquage de texte libre de MovieLens , un service de recommandation de films. Il contient 100 836 évaluations (Ratings) et 3 683 applications de balises (Tags) sur 9 742 films (Movies). Ces données ont été créées par 610 utilisateurs (Users) entre le 29 mars 1996 et le 24 septembre 2018. Cet ensemble de données a été généré le 26 septembre 2018.

Les utilisateurs ont été sélectionnés au hasard pour être inclus. Tous les utilisateurs sélectionnés avaient noté au moins 20 films. Aucune information démographique n'est incluse. Chaque utilisateur est représenté par un identifiant et aucune autre information n'est fournie.

Les données sont contenues dans les fichiers links.csv, movies.csv, ratings.csv et tags.csv.

Nom de la donnée	Nombre
User	610
Movie	9742
Tag	3 683
Genre	20
Rating	100836

Tableau 3.1 : Ensemble de données (ml-latest-small)

3.3 Architecture du Modèle

Le modèle de recommandation que nous proposons est basé sur deux étapes principales, chaque étape est constituée de sous-étapes, nous commençons d'abord par expliquer l'architecture du modèle que nous avons créé, nous expliquerons toutes ses étapes par la suite

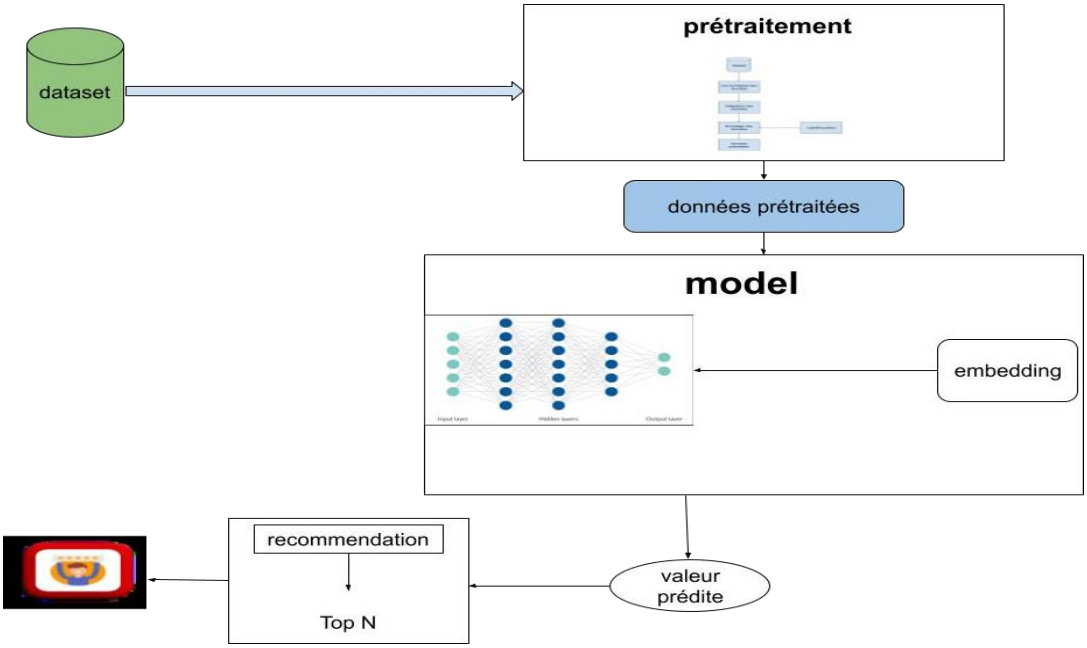


Figure 3.1 : Architecture du modèle

a. Entrées du modèle

- **Utilisateurs** : Chaque utilisateur est représenté par un identifiant unique.
- **Items** : Chaque item (film) est représenté par un identifiant unique
- **Évaluations** : Les évaluations sont des notes données par les utilisateurs aux items. Elles servent à l'entraînement du modèle pour apprendre les préférences des utilisateurs.
- **Tags** : Les tags sont des mots-clés ou des étiquettes associés aux items, aidant à capturer des aspects spécifiques ou des thèmes des items.
- **Genres** : Les genres sont des catégories ou des types auxquels appartiennent les items, comme l'action, la comédie, le drame pour les films.

b. Prétraitement :

L'objectif de cette étape est de comprendre, de nettoyer et de préparer les données, ce qui inclut la suppression de toutes les choses inutiles ainsi que la correction des contradictions dans l'ensemble de données.

Tout d'abord, nous avons supprimé toutes les propriétés que nous n'avons pas utilisées, nous ne conservons que celles qui étaient nécessaires à notre modèle, avec la résolution de certains problèmes avec les ensembles de données.

Par la suite nous avons encodé les utilisateurs et les items en vecteurs denses (embeddings) pour capturer les relations et les similarités. Les tags et les genres sont également encodés en vecteurs pour capturer leurs influences respectives sur les évaluations.

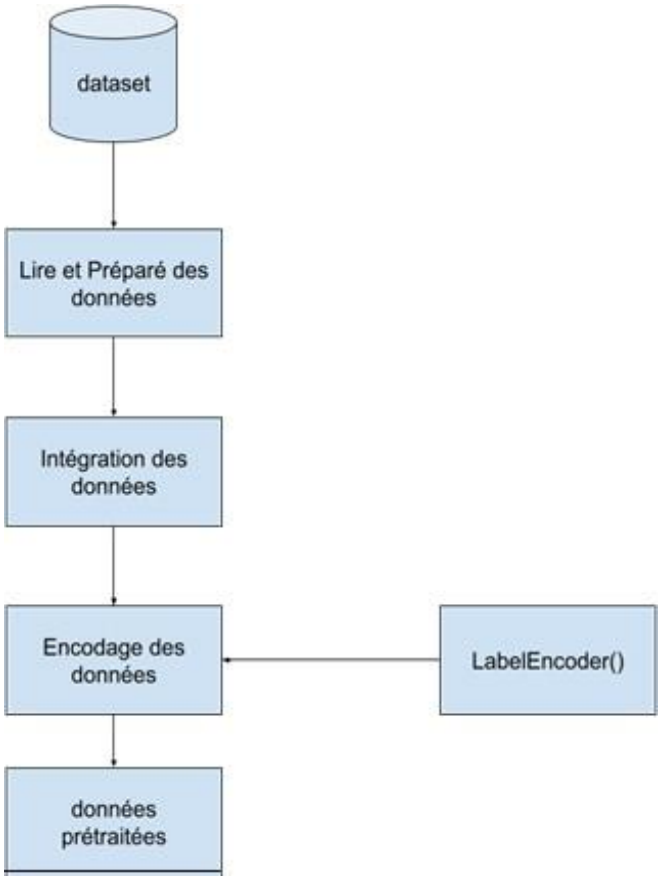


Figure 3.2 : Prétraitement

c. Modèle (Traitement) :

La deuxième étape consiste à affiner les dernières données prétraitées à l'aide d'une méthode d'encodeur automatique, utilisée pour éliminer tous les attributs inutiles susceptibles de gêner ou d'interférer avec nos résultats.

Un auto-encodeur est un réseau de neurones artificiels qui apprend une représentation (codage) d'un ensemble de données d'entrée. Il possède le même nombre de couches d'entrée et de sortie afin de reconstruire ses propres données d’entrée.

Nos modèles sont des réseaux neuronaux profonds (DNN), l'un avec auto-encodeur et l'autre sans. Un modèle prend en entrée les vecteurs d'intégrations (embeddings) pour les utilisateurs, les items, les tags et les genres combinés pour former la couche d'entrée du réseau.

Plusieurs couches entièrement connectées (connected layers) sont utilisées, avec des fonctions d'activation non linéaires ReLU pour modéliser les interactions complexes entre les utilisateurs et les items.

La couche de sortie produit une prédiction de l'évaluation pour un utilisateur donné et un item donné. La fonction d'activation de la couche de sortie est linéaire pour prédire une note continue.

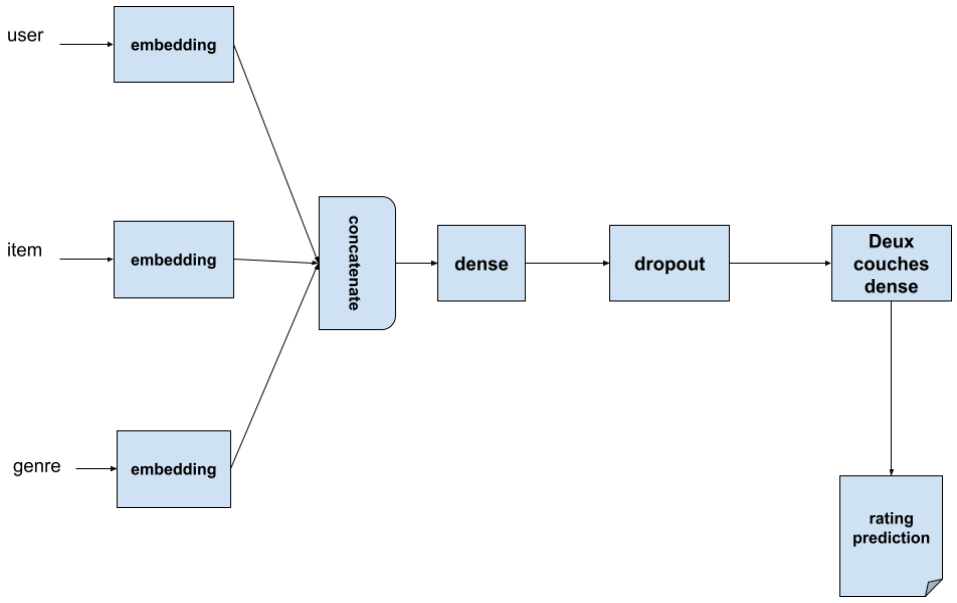


Figure 3.3 : modèle de Traitement sans Auto-encodeur

d. Entraînement du Modèle

Nous avons utilisé la fonction de perte l'erreur quadratique moyenne (**Mean Squared Error, MSE**) qui est couramment utilisée entre les évaluations prédites et les évaluations réelles.

L'optimisation est réalisée en utilisant d'algorithme **Adam** pour minimiser la fonction de perte.

La technique de régularisation **Dropout** est utilisée pour éviter le surapprentissage (Overfitting).

e. Évaluation du Modèle

Les métriques telles que RMSE (Root Mean Squared Error) et MAE (Mean Absolute Error) ont été utilisées pour évaluer la performance du modèle sur un ensemble de validation. De même, une validation croisée a été utilisée pour évaluer la généralisation du modèle.

f. Recommandation

Dans la dernière étape de notre projet, nous recommanderons les éléments qui pourraient intéresser ou satisfaire un utilisateur en utilisant les prédictions de notre modèle. Pour chaque utilisateur, il y aura une liste de tous les éléments triés par note en l'ordre décroissant. A partir de la liste, nous choisirons les N films les mieux notés et les recommanderons à l'utilisateur.

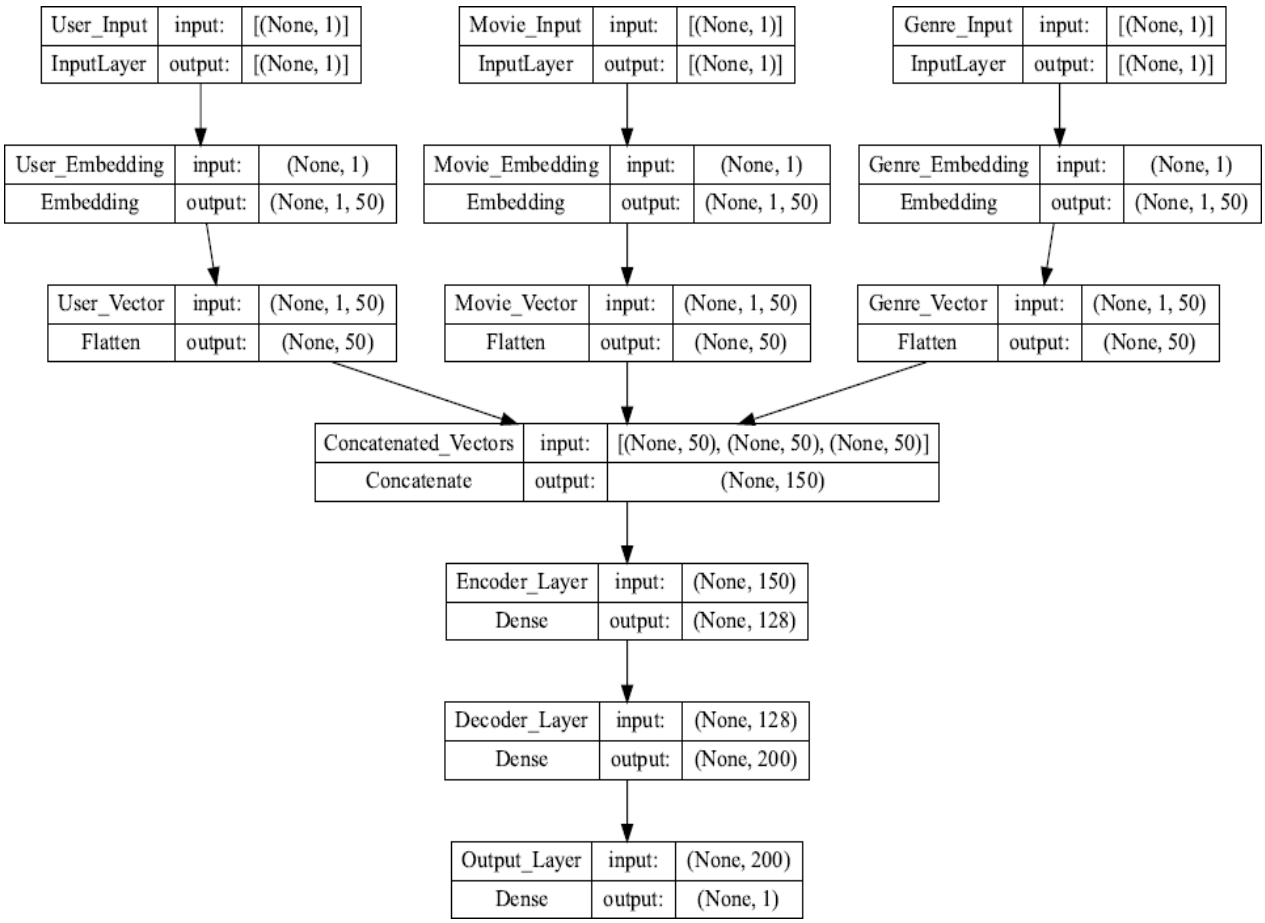


Figure 3.4: Schéma du modèle avec Auto-encodeur

En résumé, voici les étapes de construction de notre modèle

- Charger les fichiers ratings.csv, movies.csv et tags.csv.
- Fusionner les données en un seul DataFrame.
- Encoder user, movie, tags et genres en indices numériques.
- Séparer les caractéristiques (user, movie, genres, tags) et la cible (rating).
- Diviser les données en ensembles d'entraînement et de test (80/20).

// Définition du Modèle

  - **Entrées** : Définir les entrées pour utilisateur, film, tag et genre.
  - **Embeddings** : Créer des embeddings pour chaque entrée.
  - **Aplatissement**: Aplatir les embeddings.
  - **Concaténation** : Concaténer les vecteurs aplatis.
  - **Couches Denses** : Ajouter deux couches denses avec activation ReLU.
  - **Sortie** : Ajouter une couche de sortie avec activation linéaire.

- Compiler le modèle avec l'optimiseur Adam, la perte Mean_Squared_Error et le régularisateur Dropout
- Entraîner le modèle avec les données d'entraînement, en utilisant une taille de batch de 32 et 40 époques.
- Évaluer les performances du modèle sur l'ensemble de test.

3.4 Conclusion

Nous avons présenté dans ce chapitre notre méthodologie de travail et 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 profond avec l'algorithme de réseau de neurones et nous allons voir les résultats de l'expérimentation dans le chapitre suivant.

Chapitre 4 :

Expérimentations

4.1 Introduction

Ce chapitre présente l'expérimentation et l'évaluation menées pour valider l'approche proposée pour notre système de recommandation de films. Nous avons testé notre système en suivant différentes étapes et analysé les résultats obtenus, dans l'optique de développer une nouvelle méthode de recommandation basée sur l'apprentissage automatique.

4.2 Métriques de l’évaluation

L'évaluation des performances de notre système de recommandation de films s'appuie sur une comparaison de différents modèles d'apprentissage automatique. Nous exploiterons les capacités des réseaux neuronaux profonds en variant les données et les modèles pour identifier l'approche la plus efficace. Deux types de modèles seront testés, avec et sans auto-encodeur, entraînés et évalués sur trois ensembles de données distincts : (user, movie, tag, genre), (user, movie, genre) et (user, movie). Les mesures Erreur moyenne absolue (MAE), Erreur quadratique moyenne (RMSE) et la précision sont utilisées pour quantifier et comparer les performances des modèles.

- **Erreur moyenne absolue (MAE) :** Moyenne de la différence absolue entre les notes prédites et les notes réelles attribuées par les utilisateurs. Une valeur de MAE faible indique une meilleure précision des recommandations.
- **Erreur quadratique moyenne (RMSE) :** Racine carrée de la moyenne des carrés des différences entre les notes prédites et les notes réelles. Comme la MAE, une valeur de RMSE faible est synonyme de meilleures recommandations.
- **Précision :** Proportion de prédictions correctes pour un modèle. Une précision élevée indique que le modèle est capable de prédire correctement les évaluations.

4.3 Evaluation Expérimentale

4.3.1 Evaluation des mesures

- **Evaluation sans auto-encodeur :**

Modèle	RMSE	MAE	Précision
Modèle1	0.55	0.368	0.767
Modèle2	**0.49**	**0.297**	**0.839**
Modèle3	0.529	0.324	0.806

Tableau 4.1 : Résultats des critères d'évaluation sans auto-encodeur

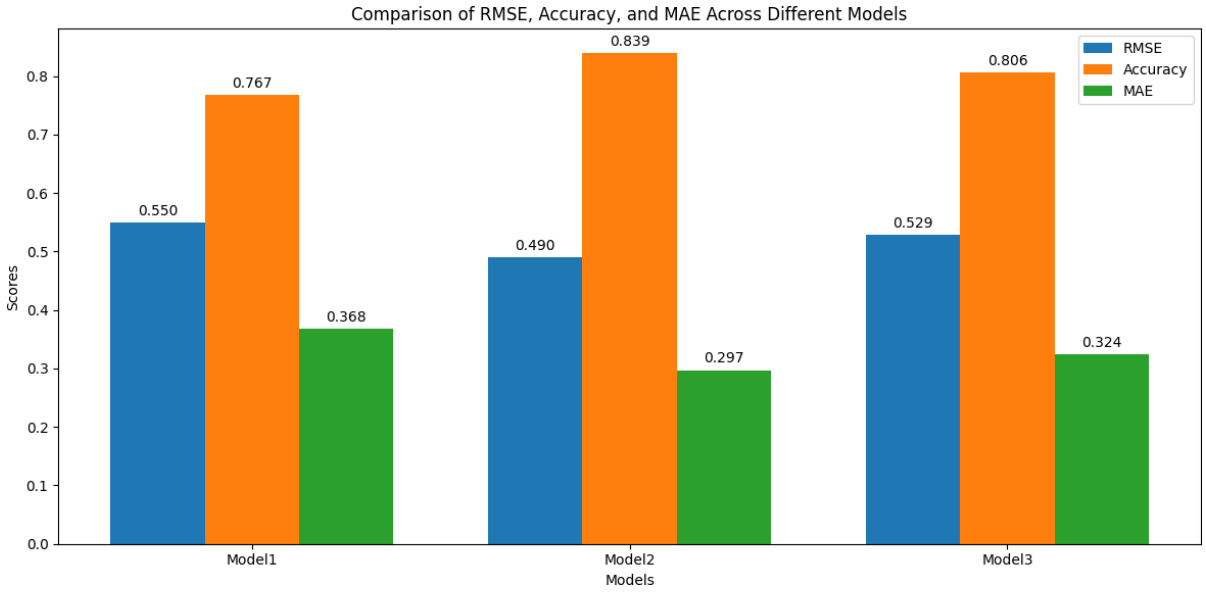


Figure 4.1 : Résultats des critères d'évaluation sans auto-encodeur

➢ **Evaluation avec auto-encodeur :**

Modèle	RMSE	MAE	Précision
Modèle4	0.526	0.325	0.769
Modèle5	**0.509**	**0.262**	**0.802**
Modèle6	0.52	0.262	0.795

Tableau 4.2 : les Résultats des critères d'évaluation avec auto-encodeur

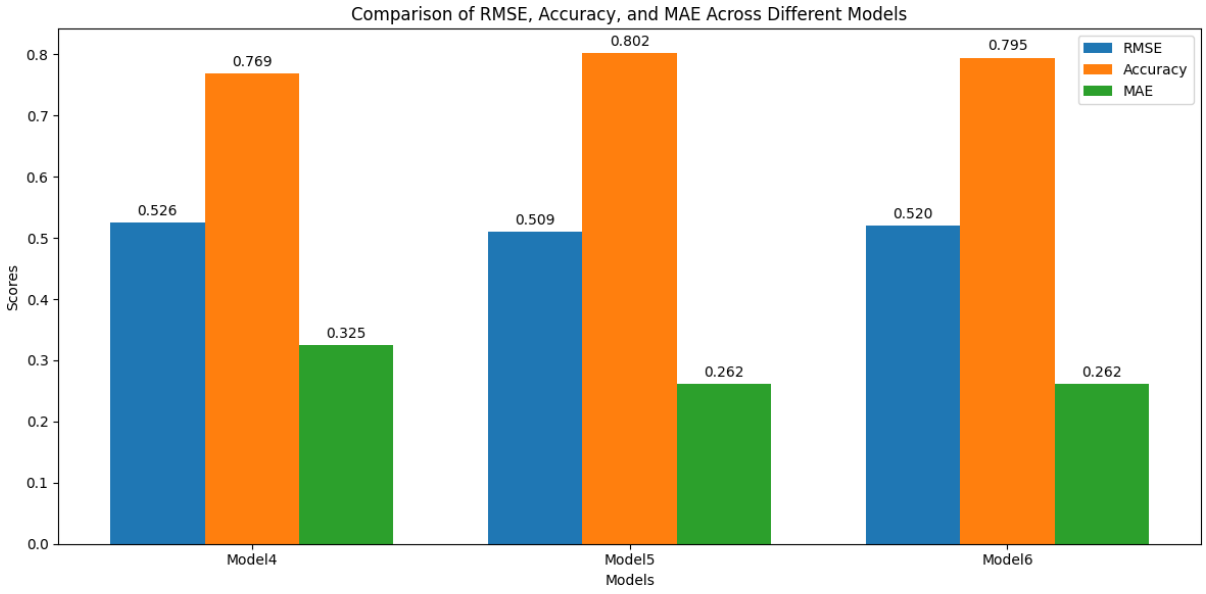


Figure 4.2 : les Résultats des critères d'évaluation avec auto-encodeur

Nous remarquons que dans les deux cas, le deuxième modèle est meilleur, suivi du troisième puis le premier, cela se justifie par le fait que les données (tags) ne sont pas bien structurées, ce qui complique la tâche du modèle, et qu'elles peuvent également se chevaucher avec les données (genres).

Quant à l'auto-encodeur, nous remarquons que son utilisation donne de meilleur résultat en MAE.

4.3.2 Evaluation des mesures basées sur la perte

- **Evaluation sans auto-encodeur :**

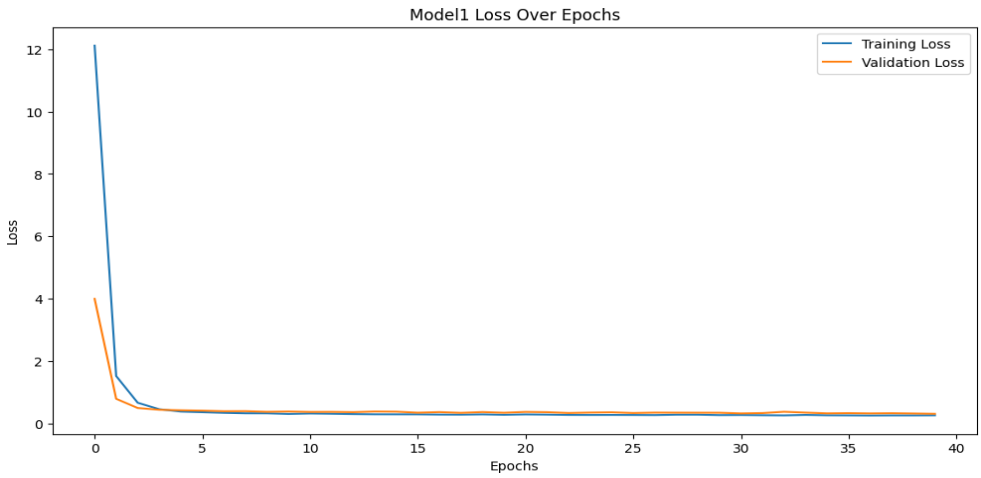


Figure 4.3 : **Analyse d'entrainement et validation des courbes de perte *Modèle1***

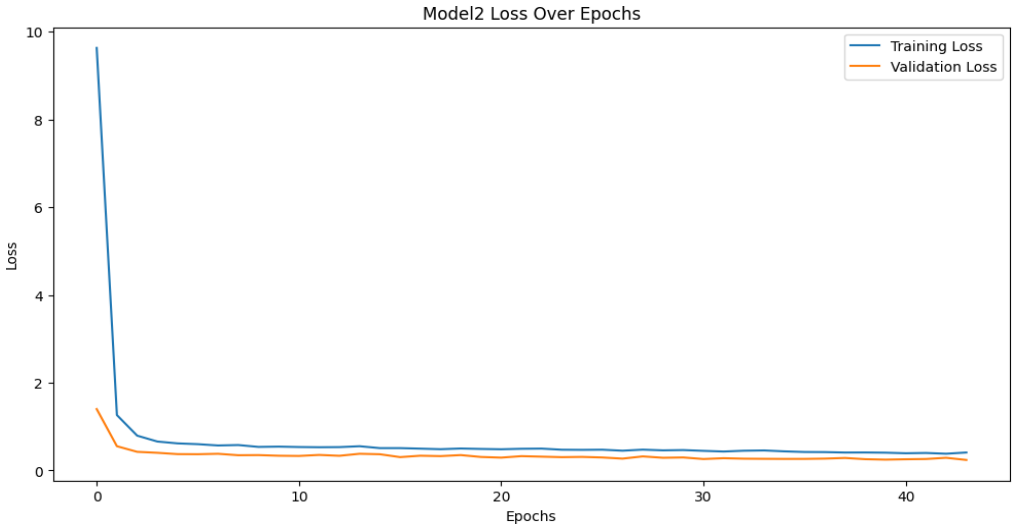


Figure 4.4: **Analyse d'entrainement et validation des courbes de perte *Modèle2***

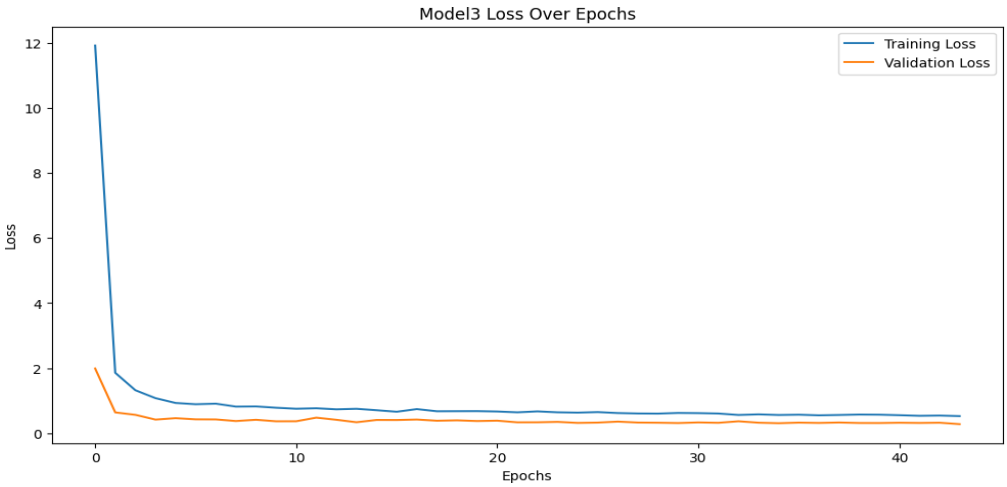


Figure 4.5: **Analyse d'entrainement et validation des courbes de perte *Modèle3***

- **Evaluation avec auto-encodeur :**

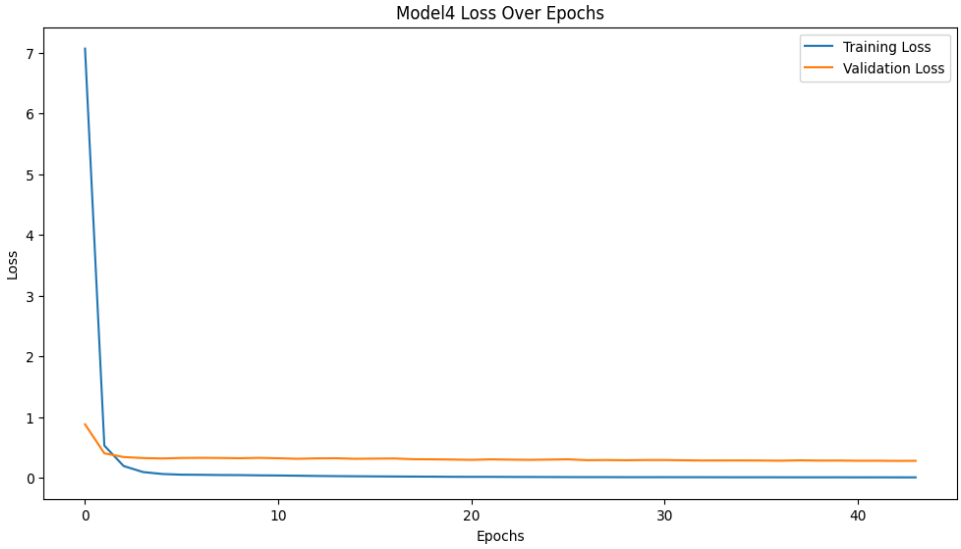


Figure 4.6 : **Analyse d'entrainement et validation des courbes de perte *Modèle4***

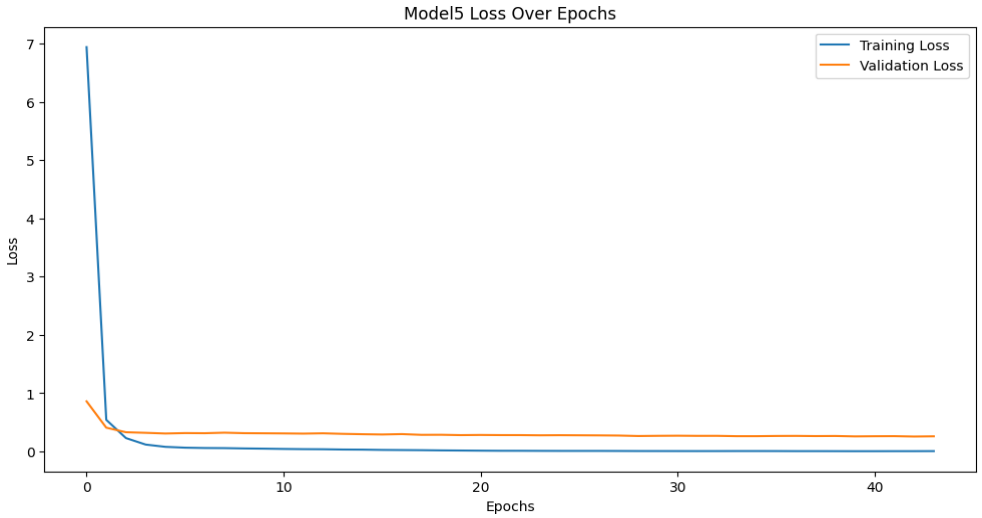


Figure 4.7 : **Analyse d'entrainement et validation des courbes de perte *Modèle5***

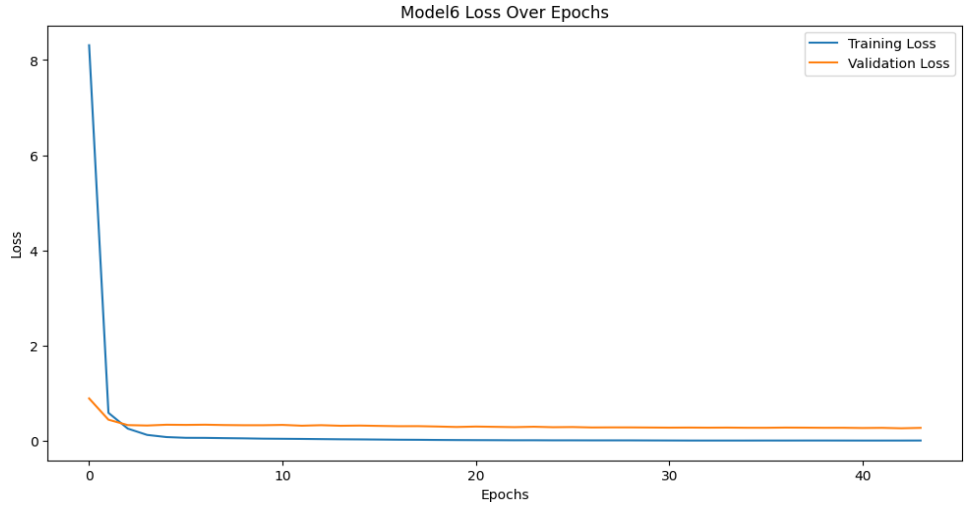


Figure 4.8 : **Analyse d'entrainement et validation des courbes de perte *Modèle6***

En regardant ces six distinctions, nous observons que les valeurs d'entraînement et de validation des pertes diminuent au cours des premières valeurs des époques, puis se réduisent à des valeurs basses et qu'il n'y a pas de grande différence entre elles après la stabilisation, ce qui suggère que les modèles n'ont pas de déficit d'apprentissage (Overfitting). Cela signifie que le modèle est stable et fonctionne bien.

4.4 Evaluation de l'Approche de Recommandation

Nous comparons nos modèles avec différents algorithmes utilisés dans le domaine du système de recommandation.

KNN : est une méthode d'apprentissage paresseux non paramétrique. Il utilise une base de données dans laquelle les points de données sont séparés en plusieurs groupes pour faire des inférences pour de nouveaux échantillons [59]

SVD : une technique de factorisation matricielle qui décompose notre matrice de factorisation en 3 matrices génériques et familières, qui sont ensuite utilisées pour faire nos prédictions [60]

Item-based collaborative filtering (Filtrage collaboratif basé sur les éléments) : Dans cet algorithme, les similitudes entre les différents éléments de l'ensemble de données sont calculées à l'aide d'un certain nombre de mesures de similarité, puis ces valeurs de similarité sont utilisées pour prédire les évaluations [61].

User-Based Collaborative Filtering (Filtrage collaboratif basé sur l'utilisateur) : il s'agit d'une technique utilisée pour prédire les articles qu'un utilisateur pourrait aimer sur la base des évaluations données à cet article par d'autres utilisateurs ayant des goûts similaires à ceux de l'utilisateur cible [62].

4.4.1 Comparaison

Modèle	RMSE	MAE	Precision
Model1	0.55	0.368	0.767
Model2	**0.49**	0.297	**0.839**
Model3	0.529	0.324	0.806
KNN	0.981	0.755	0.711
SVD	**0.882**	**0.679**	**0.716**

User-based	1.245	0.98	0.442
Item-based	0.931	0.719	0.34
Model4	0.526	0.325	0.769
Model5	0.509	**0.262**	0.802
Model6	0.52	0.262	0.795

Tableau 4.3 : Comparaison entre nos modèles avec d'autres algorithmes

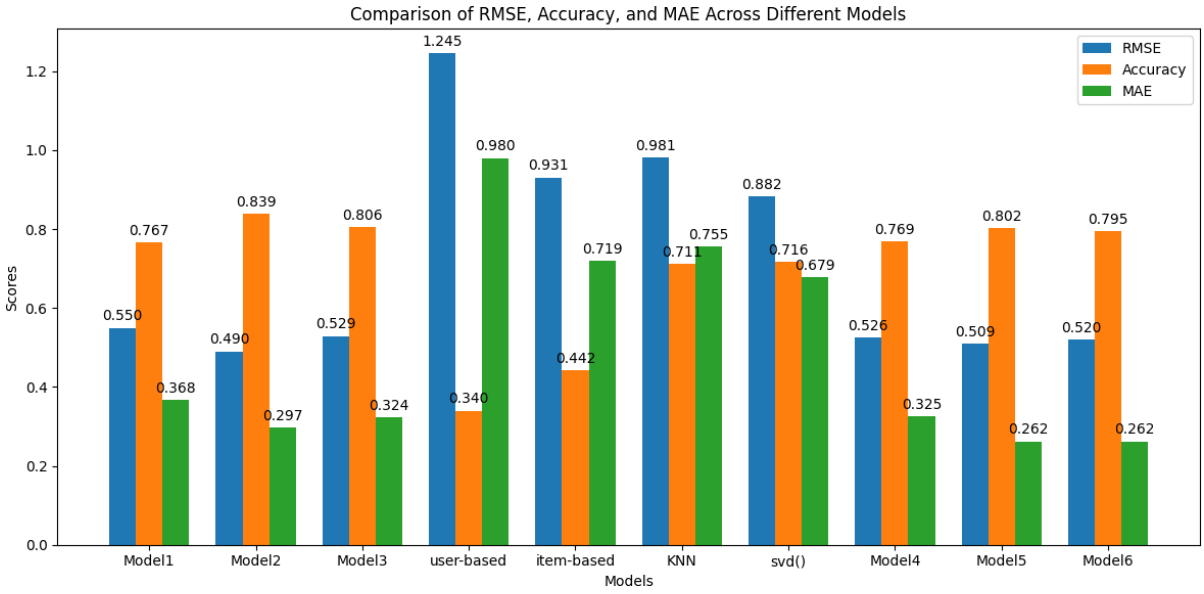


Figure 4.9 : Comparaison entre nos modèles avec différents autres algorithmes

4.4.2 Analyse des résultats

Nous avons constaté que l'algorithme SVD offre de meilleurs résultats que les autres systèmes traditionnels. Nous avons également trouvé que notre modèle (Model2), qui utilise (userId, movieId, genre), améliore la précision de 12 % et réduit les valeurs de RMSE et MAE d'environ 40 % par rapport au modèle de référence (SVD). Par conséquent, nous concluons que le modèle 2 fournit de meilleurs résultats.

En fin de compte, les modèles d'apprentissage profond et les réseaux neuronaux surpassent les systèmes traditionnels.

4.5 Conclusion :

Dans ce dernier chapitre, nous parlons de l'expérience et l'évaluation de notre modèle, nous présentons les résultats écrits avec une analyse et une discussion plus approfondie, et une comparaison entre les résultats. Nous avons conclu qu'avec l'utilisation de l'apprentissage automatique, de meilleurs résultats sont offerts par rapport à ce que notre modèle n'a pas offert d'excellents résultats, mais qu'il est meilleur que les autres algorithmes.

Conclusion générale & Perspectives

Dans ce mémoire, nous avons abordé le développement d'un système de recommandation basé sur l'apprentissage en profondeur, dans lequel nous avons ajouté des améliorations et testé sur trois ensembles de données différents. Les résultats ont montré que les performances pouvaient être améliorées en ajoutant des ensembles de données diversifiés avec des propriétés plus variées. En outre, nous avons fourni une brève description des systèmes de recommandation, de leur fonctionnement et de leurs applications, tout en expliquant la relation entre eux et les techniques d'apprentissage automatique et de profondeur. La recherche dans ce domaine se poursuit pour répondre aux exigences croissantes des utilisateurs et améliorer les performances des systèmes de recommandation.

Les résultats peuvent être améliorés en ajoutant plus de couches cachées, entraîner le modèle pour plus de nombre d'époques ou utiliser un jeu de données plus grand.

Des améliorations peuvent être suggérées comme : Intégration de données supplémentaires **(**les historiques d'interactions, attributs plus performants ou les réseaux sociaux) pour améliorer les prédictions. Combiner le DNN avec des techniques de filtrage collaboratif ou de filtrage basé sur le contenu pour améliorer la précision des recommandations. Et utiliser des techniques d'apprentissage par renforcement pour affiner les recommandations en fonction des retours d'utilisateurs en temps réel.

Références :

[1] : Mémoire : Conception et Réalisation d'un Système de Recommandation à l'aide d'un GAN Sujet Proposé par : Mme BENSAOU Nacera et M. DJOUADA Moussa l'année 2021

[2]: N. Vaidya and A. R. Khachane, "Recommender systems-the need of the ecommerce ERA," 2017 International Conference on Computing Methodologies and Communication (ICCMC), 2017, pp. 100-104, doi: 10.1109/ICCMC.2017.8282616.

[3]: Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context awareness. In Handheld and ubiquitous computing, pages 304–307. Springe

[4]: Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. In Proceedings of the 1st ACM conference on Electronic commerce (pp. 158-166). ACM.

[5]: BURKE, R. (2002). HYBRID RECOMMENDER SYSTEMS: SURVEY ANDEXPERIMENTS. USER MODELING AND USER- ADAPTED INTERACTION, 12(4), 331–370.

[6]: G. ADOMAVICIUS ET A. TUZHILIN, "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, T. 17, NO 6, P. 734-749, JUIN 2005, ISSN: 1041-4347. DOI : 10.1109/TKDE.2005.99. ADRESSE : HTTP://IEEEXPLORE.IEEE. ORG/DOCUMENT/1423975/ (VISITE LE 19/08/2020).

[7]: XIAOYUAN SU AND TAGHI M. KHOSHGOFTAAR. A SURVEY OF COLLABORATIVE FILTERING TECHNIQUES. ADVANCES IN ARTIFICIAL INTELLIGENCE, 2009 :1–19, 2009.

[8]: ISINKAYE, F. O., FOLAJIMI, Y. O., & OJOKOH, B., A.: RECOMMENDATION SYSTEMS: PRINCIPLES

METHODS AND EVALUATION, EGYPTIAN INFORMATICS JOURNAL, 16, 2015, PP 261-273.

[9]: RICH, E. 1979. USER MODELING VIA STEREOTYPES. COGNITIVE SCIENCE 3(4):329–354.

[10]: ROCCHIO, J. J. 1971. RELEVANCE FEEDBACK IN INFORMATION RETRIEVAL. IN THE SMARTRETRIEVALSYSTEM:EXPERIMENTSINAUTOMATICDOCUMENTPROCESSING,ED.G.SALTON,313–323. ENGLEWOOD CLIFFS, NJ: PRENTICE-HALL.

[11] : MERYEM, M. K. (14 JUIN 2015). SYSTEME DE RECOMMANDATION DESSERVICES WEB SEMANTIQUES.UNIVERSITE ABOU BAKR BELKAID– TLEMCEN.

[12] : LES SYSTEMES DE RECOMMANDATION : UNE CATEGORISATION. (S.D.). RECUPERE SUR INTERSTICES.INFO:HTTPS://INTERSTICES.INFO/LES-SYSTEMES-DE-RECOMMANDATION-CATEGORISATION/

[13] : A LA DECOUVERTE DES SYSTEMES DE RECOMMANDATION. (S.D.). RECUPERE SUR INVIVOO: HTTPS://WWW.INVIVOO.COM/SYSTEME-DE-RECOMMANDATION/

[14] : ELSA.NEGRE. (25/06/2012). SYSTEME DE RECOMMANDATION. UNIVERSITE PARIS-DAUPHINE.

[15] : FILTRAGE_COLLABORATIF. (S.D.). RECUPERE SUR WIKIPEDIA: https://fr.wikipedia.org/wiki/Filtrage_collaboratif

[16] : AMRAOUI YASMINE ET AGGOUN DJAMEL, 2019, APPRENTISSAGE AUTOMATIQUE : APPLICATION AUX SYSTEMES DE RECOMMANDATION, UNIVERSITE DE MEDEA.

[17] : PAATERO P. AND TAPPER U., (1994). POSITIVE MATRIX FACTORIZATION: A NON-NEGATIVE FACTOR MODEL WITH OPTIMAL UTILIZATION OF ERROR ESTIMATES OF DATA VALUES. ENVIRONMETRICS, VOL. 5, PP. 111–126.

[18] : HTTP://DSPACE.UNIV-TLEMCEN.DZ/BITSTREAM/112/8177/1/SYSTEME-DE-RECOMMANDATION-DES-SERVICES-WEB-SEMANTIQUES.PDF

[19] : K. GOLDBERG, T. ROEDER, D. GUPTA, C. PERKINS EIGENTASTE: A CONSTANT TIME COLLABORATIVE FILTERING ALGORITHM INFORM RETRIEVAL J, 4 (2) (2001)

[20] : CLAYPOOL M, GOKHALE A, MIRANDA T, MURNIKOV P, NETES D, SARTIN M. COMBINING CONTENT- BASED AND COLLABORATIVE FILTERS IN AN ONLINE NEWSPAPER. IN: PROCEEDINGS OF ACM PAGE 56

SIGIR WORKSHOP ON RECOMMENDER SYSTEMS: ALGORITHMS AND EVALUATION, BERKELEY, CALIFORNIA; 1999.

[21] : B.M. SARWAR, J.A. KONSTAN, J.L. HERLOCKER, B. MILLER, J.T. RIEDL USING FILTERING AGENTS TO IMPROVE PREDICTION QUALITY IN THE GROUPLENS RESEARCH, COLLABORATIVE FILTERING SYSTEM PROCEEDINGS OF THE ACM CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, ACM, NEW YORK (NY, USA) (1998), PP. 345-354

[22] : M. DROSOU, E. PITOURA SEARCH RESULT DIVERSIFICATION SIGMOD REC, 39 (1) (2010), PP.41-47

[23] : M. PAPAGELIS, D. PLEXOUSAKIS QUALITATIVE ANALYSIS OF USER-BASED AND ITEM-BASED PREDICTION ALGORITHMS FOR RECOMMENDATION AGENTS INT J ENG APPL ARTIF INTELL, 18 (4) (2005), PP.781-789

[24] : NGUYEN, A.-T. (N 17 JAN 2009). UN NOUVEAU SYSTEME DE FILTRAGE COLLABORATIF. HTTPS://TEL.ARCHIVES-OUVERTES.FR/TEL-00353945/DOCUMENT: /TEL.ARCHIVES-OUVERTES.FR

[25] : Systèmes de recommandation – Olcya

[26] : Recommender System evaluation Sergey Vasilyev Senior Machine Learning engineer / Data Scientist at SOFTSWISS Publié le 12 oct. 2023

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

[28] : Tadlaoui. M. 2018, Système de recommandation de ressources pédagogiques fondé sur les liens sociaux : formalisation et évaluation. THESE de DOCTORAT DE L'UNIVERSITE DE LYON opérée au sein de L'INSA Lyon et délivrée en partenariat international avec L'université de Tlemcen.

[29] : https://learn.microsoft.com/fr-fr/azure/ai-services/personalizer/concepts-offline-evaluation

[30]: Machine learning: Trends, perspectives, and prospects M. I. Jordan1 * and T. M. Mitchell2 *

[31] : What Is Machine Learning? Issam El Naqa and Martin J. Murphy

[32] : Supervised Learning with Python Vaibhav Verdhan Foreword by Dr. Eli Yechezkiel Kling (PhD)

[33] : https://www.varonis.com/fr/blog/classification-des-donnees-kesako

[34] : Types of Supervised Learning You Must Know By Varun Samarth https://emeritus.org/in/learn/types-of-supervised-learning/

[35] : https://www.geeksforgeeks.org/supervised-unsupervised-learning/

[36] : livre : Apprentissage profond : Principes, concepts et méthodes

Auteur: Milad Vazan Milad Vazan

[37] : https://www.sicara.fr/fr/parlons-data/deep-learning

[38] : livre : Introduction à l'intelligence artificielle et à l'apprentissage automatique

Auteur : Mohamed Lahleh Mohamed Lahleh Edité par : Jamil Belloni

Maison d'édition: Académie Hassoub

[39] : International Journal of Engineering Applied Sciences and Technology, 2020 Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 310-316 Published Online April 2020 in IJEAST (http://www.ijeast.com) 310 ACTIVATION FUNCTIONS IN NEURAL NETWORKS

[40] : Réglages des hyper-paramètres 4 juin 20195 juillet 2019 par Bastien Maurice

[41] : Difference Between a Batch and an Epoch in a Neural Network by Jason Brownlee on August 15, 2022 in Deep Learning

[42] : Neural Network Embeddings Explained How deep learning can represent *War and Peace* as vector Will Koehrsen Published in Towards Data Science Oct 2,2018

[43] : Hyperparameter optimization in learning systems Răzvan Andonie1 Received: 16 August 2019 / Accepted: 1 October 2019

[44] : Fundamental Concepts of Convolutional Neural Network Anirudha Ghosh A. Sufian Farhana Sultana Chapter · January 2020

[45] : Recurrent Neural Network TINGWU WANG, MACHINE LEARNING GROUP,

UNIVERSITY OF TORONTO FOR CSC 2541, SPORT ANALYTICS

[46] : Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735.

[47] : A Study on Diferent Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN J. Naskath1 · G. Sivakamasundari1 · A. Alif Siddiqua Begum1, 2

[48] : LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." Nature, 521(7553), 436-444. doi: 10.1038/nature14539.

[49] : Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning." MIT Press. Chapitre 6 : Networks.

[50] : Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep coevolutionary network: Embedding user and item features for recommendation. arXiv preprint. *arXiv preprint arXiv:1609.03675* (2016).

[51] : Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the WWW*. 111–112. http://doi.acm.org/10.1145/2740908.2742726

[52] : Jianfeng Gao, Li Deng, Michael Gamon, Xiaodong He, and Patrick Pantel. 2014. Modeling interestingness with deep neural networks. (June 13 2014). US Patent App. 14/304,863.

[53] : Context Aware Recommendation Systems: A review of the state of the art techniques

Author links open overlay panel Saurabh Kulkarni [a], Sunil F. Rodd

[54] : https://towardsdatascience.com/neural-collaborative-filtering-96cef1009401

[55] : https://www.anaconda.fr/

[56] : https://scikit-learn.org/stable/about

[57] : https://www.anaconda.com/

[58] : google colab

[59] : https://machinelearningmastery.com/using-singular-value-decomposition-to-build-arecommender-system/

[60] : https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-1-knn-item-based-collaborative-filtering-637969614ea

[61] : https://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/itembased.html

[62] : https://www.geeksforgeeks.org/user-based-collaborative-filtering/

[63] : Resnick, P., & Varian, H. R. (1997). "Recommender Systems". *Communications of the ACM*, 40(3), 56-58.

[64] : Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). "Item-Based Collaborative Filtering Recommendation Algorithms". *Proceedings of the 10th International Conference on World Wide Web*, 285-295.

[65] : A Survey of Collaborative Filtering Techniques Gediminas Adomavicius, Alexander Tuzhilin

[66] : Hybrid Recommender Systems: Survey and Experiments Robin Burke

[67] : Demographic-based Collaborative Filtering Recommender Systems Michael Pazzani Intelligent Systems, IEEE

[68] : Constraint-based Recommender Systems Francesco Ricci, Lior Rokach, Bracha Shapira
Recommender Systems Handbook 2011

[69] : Trust-aware Recommender Systems Paolo Massa Paolo Avesani

[70] : Evaluation Metrics for Recommender Systems Paolo Cremonesi, Yehuda Koren, Roberto Turrin

[71] : Recommender Systems Handbook Francesco Ricci, Lior Rokach, Bracha Shapira