

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie HOUARI BOUMEDIENE
Faculté d'Informatique



THÈSE DE MASTER

Présentée pour l'obtention du diplôme de **MASTER**

En : Informatique

Spécialité : Bioinformatique

Par : M. MOUHOU B Massinissa, M. ZIGADI Sadji Amine

Encadrés par : Pr. MEHDI Malika, Pr. HENNI Khadidja

Thème

**Développement d'un système de recommandation
thérapeutique de l'arthrose du genou**

Mme. BOULKRINAT Samia	Professeur à l'USTHB	Présidente
Mme. KERKAD Amira	Professeur à l'USTHB	Membre

On dédie ce travail à toutes les personnes qui ont cru en nous et nous ont soutenu tout au long de ce parcours.

À nos parents, pour leur amour inestimable, leur patience et leurs sacrifices quotidiens.

À nos famille, pour leur présence, leur encouragement et leur bienveillance.

À nos enseignants et encadrants, pour leur dévouement et leurs précieux conseils.

À nos amis, pour leur soutien moral et leurs encouragements constants.

À tous ceux qui, de près ou de loin, ont contribué à l'accomplissement de ce projet.

Merci.

Remerciements

Avant toute chose. Avant tout, on dit Dieu merci.

Nous souhaitons exprimer notre profonde gratitude à nos parents et à nos familles, dont le soutien inconditionnel, les encouragements constants et les sacrifices silencieux ont été essentiels tout au long de ce parcours. Nos pensées les plus affectueuses et reconnaissantes vont particulièrement à nos mères, dont l'amour et la patience nous ont portés dans les moments d'effort et de doute.

Nous tenons également à remercier sincèrement nos encadrantes, Madame Mehdi et Madame Henni, pour leur disponibilité, leurs conseils précieux, leur rigueur scientifique et leur bienveillance qui ont grandement contribué à la qualité et à l'aboutissement de ce travail.

Et enfin nos remerciements s'adressent aussi à Abdelhak Kaci Aissa et Manel Arfi pour leurs aide précieuses, leurs remarques constructives et leurs échanges enrichissants qui nous ont permis d'améliorer considérablement notre projet.

ملخص

تتناول هذه الدراسة الحاجة الأساسية إلى توصيات شخصية للتمارين العلاجية للأشخاص المصابين بالفُصال العظمي في الركبة. تُظهر التقييمات التجريبية الأداء المتفوق لنموذج الشبكات العصبية الرسومية، حيث حقق نموذج GNN درجة RPS بلغت 91.20%، مقارنةً بـ 64.70% لنموذج MF. وتُبرز هذه النتائج الإمكانيات الكبيرة لـ GNN في تحسين أنظمة التوصية في المجال الصحي.

الكلمات المفتاحية : أنظمة التوصية الطبية، الشبكات العصبية الرسومية، التصفية التعاونية.

Abstract

This study addresses the essential need for personalized therapeutic exercise recommendations for individuals with knee osteoarthritis. Experimental evaluations demonstrate the superior performance of the GNN model, achieving an RPS score of 91.20%, compared to 64.70% for the MF model. These results highlight the potential of GNNs to enhance health recommendation systems.

Keywords: Health-care Recommender Systems, Graph Neural Networks, Matrix Factorization (MF), Collaborative filtering.

Résumé

Cette étude répond au besoin essentiel de recommandations personnalisées d'exercices thérapeutiques pour les personnes souffrant d'arthrose du genou. Les évaluations expérimentales démontrent la performance supérieure du modèle GNN, avec un score RPS de 91,20%, contre 64,70% pour le modèle MF. Ces résultats soulignent le potentiel des GNN pour améliorer les systèmes de recommandation de santé.

Mots clés : Systèmes de recommandation médicale, Réseaux de neurones graphiques, Filtrage collaboratif.

Table des matières

1	Contexte et problématique	3
1.1	Introduction	3
1.2	Anatomie du genou	4
1.2.1	Les parties osseuses	4
1.2.2	Les muscles	7
1.2.3	Les tendons	8
1.2.4	Les ligaments	10
1.2.5	Les ménisques	11
1.2.6	Fonctionnement du genou	12
1.3	Diagnostic de l'arthrose et de la gonarthrose	13
1.4	KneeKG : la machine utilisée pour le diagnostic du genou	13
1.5	Les données cinématiques	14
1.6	Problématique	16
1.7	Conclusion	17
2	Systèmes de recommandations et GNN	18
2.1	Introduction	18
2.2	Contexte général des systèmes de recommandation	19
2.2.1	Définition d'un système de recommandations	19
2.2.2	Classification des approches de recommandation	19
2.3	Application des systèmes de recommandation en santé	22
2.4	Système de recommandation collaboratif	24
2.4.1	Types d'approches de filtrage collaboratif	25
2.4.2	Avantages et limites du filtrage collaboratif	29
2.5	Introduction aux Graph Neural Networks	30
2.5.1	Définition des GNNs	31
2.5.2	Architecture et fonctionnement des GNN	33
2.5.3	Utilisation des GNNs dans les Systèmes de recommandation	34
2.6	État de l'art des système de recommandation en santé	38

Table des matières

2.7	Conclusion	39
3	Conception du système de recommandation	43
3.1	Introduction	43
3.2	Définition des données disponible	44
3.3	Prétraitement et structuration des données graphiques	45
3.4	Conception du modèle GNN	47
3.4.1	Description des entités manipulées	47
3.4.2	Représentation des features	48
3.4.3	Définition des arêtes du graphe	49
3.4.4	Architecture du modèle GNN	50
3.4.5	Stratégie de décodage des scores	50
3.4.6	Fonction de perte	51
3.5	Conception du modèle Matrix Factorization	52
3.5.1	Principe de la Matrix Factorization	52
3.5.2	Fonction de perte	53
3.6	Problème du Cold Start	53
3.7	Comparaison des approches GNN et MF pour la recommandation	55
3.8	Conclusion	55
4	Implémentation et tests	57
4.1	Introduction	57
4.2	Environnement de développement	57
4.2.1	Matériels utilisés	58
4.2.2	Logiciels et bibliothèques	58
4.3	Décodage et prédiction des scores du GNN	58
4.4	Entraînement des modèles	59
4.4.1	Entraînement du modèle GNN	59
4.4.2	Entraînement du modèle Embedding Encoder	60
4.5	Évaluation de la performance des modèles GNN et MF	61
4.6	Recherche de la configuration optimale du modèle GNN	63
4.6.1	Objectif des expérimentations	63
4.6.2	Résultats expérimentaux et analyse	66
4.6.3	Configuration optimale	70
4.7	Recherche de la configuration optimale du modèle MF	74
4.8	Interface graphique	76
4.9	Comparaison des performances entre le modèle GNN et MF	76
4.10	Conclusion	79

Table des matières

A	Liste des champs et explication	100
1.1	Caractéristiques orthopédiques lors de la prescription	100
1.2	Tests fonctionnels liés aux effets des exercices	101
B	Résultats des Tests sur les Hyperparamètres d'Entraînement	103

Table des figures

1.1	Partie osseuse du genou.	4
1.2	Les os du genou sous différentes vues.	5
1.3	Anatomie du tibia.	6
1.4	Le péroné.	7
1.5	Le muscle quadriceps.	8
1.6	Les muscles ischio-jambiers.	9
1.7	Les tendons du genou.	9
1.8	Le tendon rotulien.	10
1.9	Les ménisques du genou	12
1.10	L'exosquelette du KneeKG	14
1.11	Le cycle de marche	15
1.12	Les trois composantes des données cinématiques	15
1.13	Les capteurs de la machine KneeKG	16
2.1	Les approches de systèmes de recommandation	20
2.2	Approche basée sur le contenu	21
2.3	Approche collaborative	21
2.4	Approche hybride	22
2.5	La matrice user-item	24
2.6	La recommandation basée sur l'utilisateur et celle basé sur l'élément	25
2.7	Le filtrage collaboratif basé sur l'utilisateur.	26
2.8	Le filtrage collaboratif basé sur l'élément	27
2.9	La factorisation matricielle	28
2.10	Clustering	29
2.11	Les réseaux de neurones profonds	29
2.12	les Graph Neural Networks (GNN)	30
2.13	Exemple de structure de graphe	31
2.14	Représentation graphique des types de graphes.	40
2.15	Exemple d'utilisation des graphes	40

Table des figures

2.16	Exemple d'embeddings de mots illustrant les relations sémantiques entre termes	41
2.17	Illustration de l'architecture d'un GNN avec deux couches	41
2.18	La processus de propagation de messages dans les GNNs	42
2.19	Un exemple de graphe bipartite Utilisateur-Item avec des notes	42
3.1	Visualisation d'un graphe biparti.	48
3.2	Illustration de l'impact des features initiales dans un GNN	48
3.3	La fonction sigmoïde	54
3.4	Schéma du traitement du problème de Cold Start.	54
4.1	L'algorithme de la descente de gradient	60
4.2	Évolution de la fonction de perte lors de l'entraînement de l'Embedding Encoder dans l'espace GNN.	62
4.3	Évolution de la fonction de perte lors de l'entraînement de l'Embedding Encoder dans l'espace MF.	62
4.4	La courbe AUC	67
4.5	Évolution du RPS en fonction de <code>hidden_dim</code> et de <code>embedding_dim</code>	67
4.6	Évolution du RPS en fonction du nombre de couches (<code>nb_layers</code>). . . .	69
4.7	Évolution du RPS en fonction du dropout.	69
4.8	Évolution du RPS en fonction de l'utilisation du mécanisme d'attention et du nombre de têtes d'attention.	71
4.9	Évolution du RPS selon l'utilisation des paramètres booléens.	71
4.10	Évolution des métriques d'entraînement pour la configuration optimale. .	73
4.11	Comparaison entre les dix exercices les plus recommandés par le modèle et les dix exercices les plus fréquents dans la base de données. . . .	75
4.12	Analyse de l'impact du nombre de facteurs latents sur les performances du modèle MF.	75
4.13	Génération d'exercices pour des patients présents dans la base de données.	77
4.14	Résultats des recommandations.	78
4.15	Génération de recommandations pour un nouveau patient (cas <i>cold-start</i>).	78
4.16	Comparaison des performances entre le modèle GNN et MF.	80
2.1	Évolution du score RPS en fonction du taux d'apprentissage.	104
2.2	Évolution du score RPS en fonction de patience.	104
2.3	L'entropie croisée binaire.	106
2.4	la Focal Loss	106

Table des figures

2.5	Évolution du score RPS en fonction du paramètre γ de la <i>Focal Loss</i> .	108
2.6	Évolution du score RPS en fonction du paramètre α de la <i>Focal Loss</i> .	108
2.7	Évolution du score RPS en fonction du ratio d'échantillonnage négatif.	109

Liste des tableaux

2.1	Notations principales utilisées dans cet article	35
3.1	Exemple de sous-matrice user-item binaire extraite des données . . .	54
3.2	Comparaison des approches GNN et Matrix Factorization.	56
4.1	Caractéristiques des postes de travail utilisés	62
4.2	Grille d'interprétation du score RPS	64
4.3	Hyperparamètres de la meilleure configuration du modèle	72
4.4	Performances finales du modèle GNN recommandé (ensemble de test)	72
4.5	Performances finales du modèle GNN après fine-tuning et entraîne- ment sur l'ensemble de la base de données	75
4.6	Hyperparamètres et performances de la meilleure configuration du modèle MF.	77

List of Algorithms

1	Prétraitement des données	47
2	Fine-tuning du Modèle GNN	74

Introduction général

La gonarthrose, une affection fréquente affecte considérablement la qualité de vie de nombreuses personnes dans le monde, en particulier les personnes âgées. Caractérisée par une usure progressive du cartilage recouvrant les os du genou, elle entraîne douleurs, raideur articulaire et perte de mobilité, gênant des mouvements quotidiens comme la marche ou la montée des escaliers. La complexité de l'articulation du genou, composée de plusieurs os, muscles, tendons et ligaments, exige une compréhension approfondie de son anatomie et de son fonctionnement pour développer des stratégies thérapeutiques efficaces.

Les approches traditionnelles de gestion de la gonarthrose impliquent souvent la prescription d'exercices physiques visant à renforcer les muscles périarticulaires, améliorer la mobilité et réduire la douleur. Cependant, l'efficacité de ces programmes d'exercices standardisés est souvent limitée en raison des besoins spécifiques de chaque patient, influencés par leur morphologie, leur âge, leur niveau de douleur et leurs caractéristiques biomécaniques. Cette variabilité souligne l'importance cruciale de personnaliser les recommandations d'exercices pour optimiser les bénéfices thérapeutiques et améliorer l'adhésion du patient au traitement.

Des avancées technologiques récentes, telles que la machine KneeKG, permettent de mesurer avec précision le mouvement dynamique du genou pendant le cycle de marche. Ces données cinématiques objectives, non invasives et sans irradiation, offrent une vision fidèle de la fonction articulaire, de sa mobilité et de ses limitations. L'exploitation de ces données dynamiques via des techniques d'intelligence artificielle, et plus particulièrement les systèmes de recommandation, représente une voie prometteuse pour la conception de programmes d'exercices thérapeutiques hautement personnalisés.

Les systèmes de recommandation, largement adoptés dans divers secteurs comme le commerce électronique et le divertissement, sont des algorithmes d'apprentissage automatique essentiels qui prédisent, filtrent et identifient les éléments susceptibles d'intéresser un utilisateur. Leur évolution a progressé des méthodes classiques de filtrage collaboratif vers des approches plus avancées basées sur l'apprentissage profond, notamment les réseaux de neurones graphiques (GNN). Les GNN se distinguent

Nomenclature

par leur capacité à modéliser des relations complexes au sein de données structurées en graphes, où chaque nœud peut avoir un nombre variable de connexions et des caractéristiques propres, ce qui les rend particulièrement adaptés aux applications de recommandation en santé.

La présente étude vise à développer un système capable d’identifier, pour chaque patient atteint de gonarthrose, les exercices les plus adaptés en fonction de ses données biomécaniques. Pour ce faire, nous explorons deux approches principales : les réseaux de neurones graphiques (GNN) et la factorisation de matrice (MF). L’approche GNN est conçue pour exploiter les structures relationnelles complexes entre les patients et les exercices dans un graphe biparti, tandis que la MF sert de référence comparative pour évaluer son efficacité dans la capture des préférences latentes à partir des données d’interaction. Un défi majeur, le problème du *cold start* est également abordé grâce à un modèle qui projette les caractéristiques des nouveaux patients dans l’espace latent appris par les modèles. Ce travail pose ainsi les bases d’une intervention thérapeutique optimisée par la personnalisation des exercices.

Chapitre 1

Contexte et problématique

1.1 Introduction

La gonarthrose, ou arthrose du genou, est une maladie fréquente qui touche de nombreuses personnes dans le monde, en particulier les personnes âgées [1, 2]. Elle se caractérise par une usure progressive du cartilage qui recouvre les os du genou [3]. Cette usure provoque des douleurs, une raideur articulaire, une perte de mobilité et une gêne dans les mouvements de tous les jours [2], comme marcher ou monter les escaliers.

L’arthrose est une pathologie répandue qui impacte considérablement la fonction du genou, rendant cruciale la compréhension de son anatomie pour élaborer des recommandations d’exercices efficaces. De nombreux travaux scientifiques [4] soulignent la prévalence et les conséquences de l’arthrose du genou, justifiant l’intérêt de cette étude pour fournir une base solide de connaissances sur l’articulation touchée.

Pour mieux comprendre cette maladie et ses conséquences, il est donc important de connaître la structure et le fonctionnement du genou. Le genou est une articulation très complexe, composée de plusieurs os, de muscles, de tendons, de ligaments et de ménisques. Tous ces éléments travaillent ensemble pour permettre des mouvements comme la flexion, l’extension et la rotation. Quand certains de ces éléments sont abîmés, comme c’est le cas dans la gonarthrose, cela peut entraîner des douleurs et une perte de stabilité.

Ce chapitre présente donc les bases anatomiques du genou, les différents tissus qui le composent et leur rôle. Il introduit ensuite la gonarthrose et ses effets sur l’articulation. Enfin, il décrit un outil moderne appelé KneeKG, qui permet de mesurer les mouvements du genou de manière dynamique. Ces données, appelées données cinématiques, seront très utiles pour la suite du travail, car elles permettent de mieux analyser la situation de chaque patient et de proposer un traitement adapté.

1.2 Anatomie du genou

L'articulation du genou, la plus grande articulation du corps humain [5], est le segment du membre inférieur situé entre la cuisse et la jambe. Il enferme l'articulation du genou. C'est une structure biomécanique complexe, et elle est souvent le siège de lésions traumatiques et de dérangements internes chroniques [6]. Pour comprendre les problèmes liés au genou, comme l'arthrose du genou, communément appelée gonarthrose, il faut connaître l'anatomie de cette articulation très complexe [7]. Dans ce qui suit, nous allons voir l'anatomie du genou en faisant une étude sur les différentes parties qui composent le genou : les parties osseuses, les muscles, les tendons, les ligaments et les ménisques.

1.2.1 Les parties osseuses

Le genou est généralement considéré comme constitué des articulations fémoro-tibiale et fémoro-patellaire. La partie osseuse du genou est composée de : le fémur, le tibia, la rotule et le péroné [7], voir [figure 1.1](#). Des fractures peuvent donc toucher ces 4 os, certaines fractures nécessitent une immobilisation avec une interdiction d'appui, d'autres fractures nécessitent une rééducation précoce et enfin certaines, une intervention chirurgicale [8].

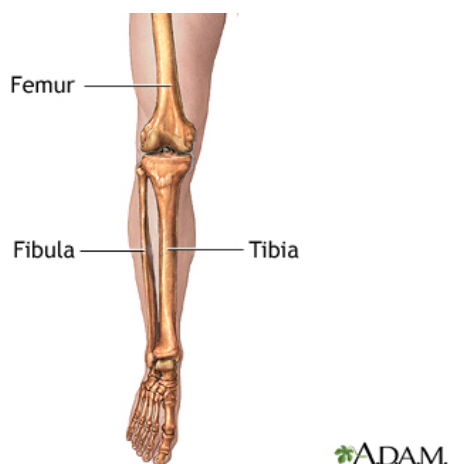


FIGURE 1.1. Partie osseuse du genou [9].

Le fémur

L'une des plus grandes parties qui sont attachées au genou est le fémur. Il est considéré comme le plus long et le plus fort os du corps humain, et il joue un rôle très important dans la posture et l'équilibre de l'humain [10, 11]. Le fémur s'articule

Chapitre 1. Contexte et problématique

en haut avec l'os iliaque et en bas avec le tibia comme illustré dans [figure 1.1](#) et [figure 1.2](#). C'est le lieu d'insertion des principaux muscles de la cuisse : le muscle quadriceps fémoral, le muscle artulaire du genou, etc. [12]. Son extrémité distale¹ présente des caractéristiques anatomiques spécifiques importantes pour l'articulation du genou. Les condyles médial et latéral sont deux surfaces articulaires arrondies recouvertes de cartilage hyalin, qui s'articulent avec les plateaux tibiaux² [5]. Le fémur ne se fracturera généralement qu'à la suite de traumatismes graves, comme les accidents de voiture, sauf si ce dernier est fragilisé par l'ostéoporose. L'anatomie complexe de l'extrémité distale du fémur souligne son rôle essentiel dans la biomécanique de l'articulation du genou et sa vulnérabilité aux changements arthrosiques. Les condyles et la trochlée fémorale sont directement impliqués dans l'articulation et le support du poids, rendant leur cartilage susceptible à la dégénérescence [13].

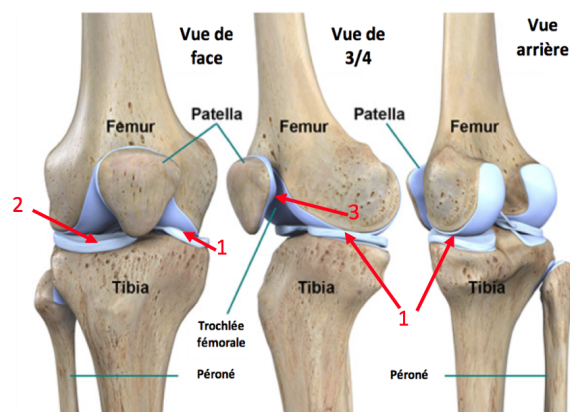


FIGURE 1.2. Les os du genou sous différentes vus [14].

Le tibia

Le tibia, également appelé os tibial comme on peut voir dans [figure 1.3](#) et [figure 1.2](#), est l'un des deux os les plus longs de la jambe, et constitue le principal os porteur de poids [15]. Il joue un rôle fondamental dans l'articulation du genou [6]. Son extrémité proximale³ comprend les condyles médial et latéral, qui sont des surfaces articulaires planes ou légèrement concaves s'articulant avec les condyles fémoraux pour former l'articulation tibio-fémorale [5]. Le tibia est un os robuste dont l'extrémité proximale, élargie, forme le plateau tibial. Ce plateau s'articule

1. L'extrémité distale du fémur est la partie inférieure de l'os de la cuisse (fémur), qui s'articule avec le tibia et la rotule pour former l'articulation du genou.

2. Le plateau tibial est la surface articulaire supérieure formée par ces condyles

3. L'extrémité proximale du tibia (ou épiphyse proximale) est la partie supérieure de l'os de la jambe (tibia), qui s'articule avec le fémur et la fibula (péroné) pour former l'articulation du genou.

Chapitre 1. Contexte et problématique

avec les condyles fémoraux pour constituer l'articulation tibio-fémorale du genou. Il est composé de deux parties distinctes. Ces plateaux sont séparés par les éminences intercondylaires, qui servent de points d'attache pour les ligaments croisés. Les surfaces articulaires du tibia sont recouvertes de cartilage hyalin, permettant un mouvement fluide avec le fémur [6].

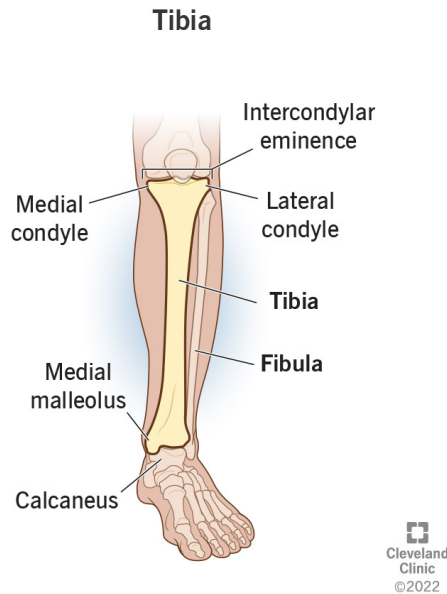


FIGURE 1.3. Anatomie du tibia [16].

Le péroné

Le péroné⁴ est un os long et mince, situé latéralement au tibia, et qui ne supporte pas de poids significatif. Son extrémité proximale est constituée de la tête, qui s'articule avec le condyle tibial latéral au niveau de l'articulation tibiofibulaire proximale, et du col, une région étroite située sous la tête, qui est un site fréquent de lésion du nerf fibulaire [15]. Il se distingue par sa nature grêle et sa taille plus réduite, qui est le plus mince os du corps humain [17]. Bien que souvent éclipsé par le tibia, principal support du poids corporel, le péroné joue un rôle essentiel dans la biomécanique du membre inférieur. Sa contribution à la stabilité de l'articulation de la cheville est significative, et il sert également de point d'attache à de nombreux muscles cruciaux pour les mouvements du pied et de la cheville [5].

4. Également appelée *fibula* en anglais.

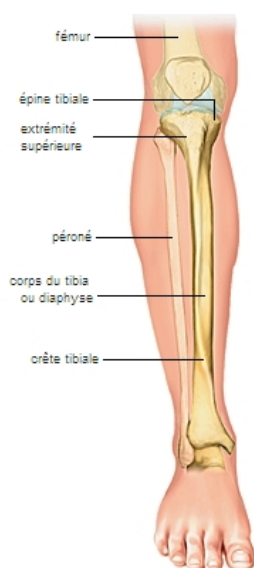


FIGURE 1.4. Le péroné [18].

La rotule

La rotule est un os sésamoïde de forme triangulaire, situé à l'avant de l'articulation du genou et enchâssé dans le tendon du quadriceps [5], comme illustré dans [figure 1.5](#). Elle possède une base supérieure, un apex inférieur, et des bords médial et latéral [19]. Sa surface postérieure est recouverte de cartilage articulaire et divisée en plusieurs facettes : une facette interne (petite), une facette externe (plus grande), et une facette odd (inconstante, située à l'extrémité interne). Ces facettes assurent un contact optimal avec la trochlée fémorale lors des mouvements de flexion-extension du genou [6]. La rotule protège la face antérieure de l'articulation du genou [20] et permet un mouvement fluide lors de la flexion et de l'extension du genou [21]. Sa position et sa fonction uniques en font une composante essentielle du mécanisme extenseur du genou et un site fréquent de changements arthrosiques affectant l'articulation fémoropatellaire.

1.2.2 Les muscles

Les muscles sont des tissus mous composés de fibres extensibles qui se contractent pour produire le mouvement [22]. Les mouvements du genou sont principalement contrôlés par deux groupes musculaires majeurs situés dans la cuisse : les quadriceps à l'avant et les ischio-jambiers à l'arrière [23].

Le **quadriceps fémoral**, situé à l'avant de la cuisse, voir [figure 1.5](#), est un groupe musculaire puissant composé de quatre chefs : le droit fémoral, le vaste latéral, le vaste intermédiaire et le vaste médial [22]. Les trois vastes prennent leur origine sur

Chapitre 1. Contexte et problématique

le fémur, tandis que le droit fémoral s'insère sur le bassin. La fonction principale du quadriceps est l'extension du genou, c'est-à-dire le redressement de la jambe. Il est également le muscle le plus important pour stabiliser l'articulation du genou. Le rôle du quadriceps dans l'extension du genou et sa fonction stabilisatrice sont fondamentaux pour les activités de port de poids et les exercices pertinents pour les patients atteints d'arthrose [23].

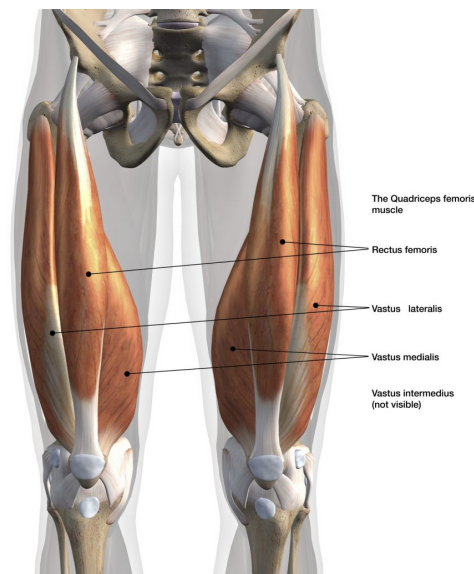


FIGURE 1.5. Le muscle quadriceps [24].

Les **muscles ischio-jambiers**, situés à l'arrière de la cuisse, voir [figure 1.6](#), comprennent le biceps fémoral, le semi-tendineux et le semi-membraneux [22]. À l'exception du chef court du biceps fémoral qui prend son origine sur le fémur, ces muscles proviennent de la tubérosité ischiatique et s'insèrent sur la partie proximale du tibia et du péroné. Leur fonction principale est la flexion du genou, c'est-à-dire le pliage de la jambe. Les ischio-jambiers sont les principaux antagonistes des quadriceps, et l'équilibre entre ces deux groupes musculaires est essentiel pour un mouvement sain du genou [23].

1.2.3 Les tendons

Les tendons sont des structures fibreuses très rigides qui permettent de relier les muscles aux os, transmettant la force générée par la contraction musculaire pour produire le mouvement [23, 8]. Plusieurs tendons importants sont associés aux muscles du genou.

Le **tendon rotulien** relie la rotule à la tubérosité tibiale⁵ [23], comme indi-

5. La tubérosité tibiale est une proéminence osseuse située sur la partie supérieure et antérieure

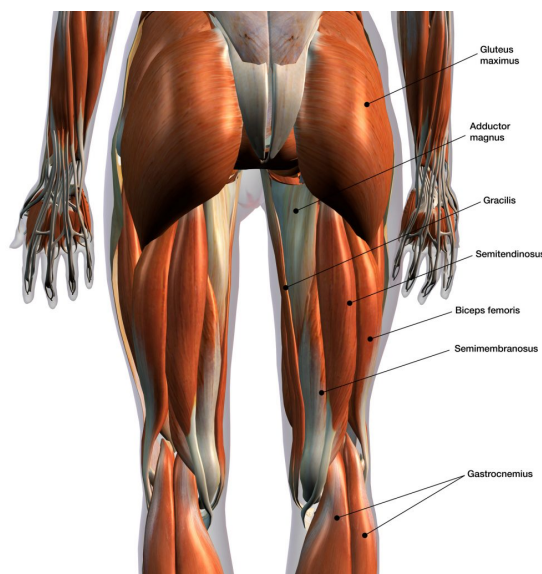


FIGURE 1.6. Les muscles ischio-jambiers [24].

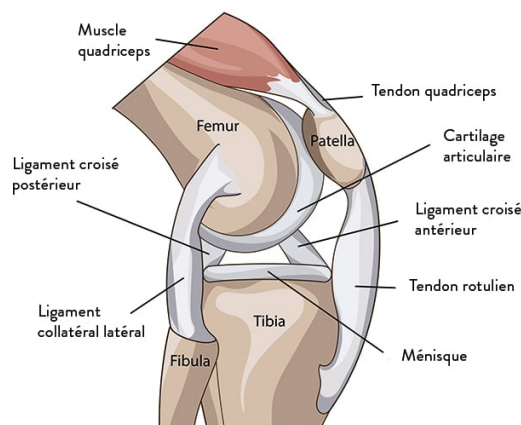


FIGURE 1.7. Les tendons du genou [25].

qué dans [figure 1.8](#). Il constitue la prolongation inférieure du tendon du quadriceps fémoral et joue un rôle essentiel dans le mécanisme d'extension du genou [26]. Le tendon rotulien est un composant essentiel du mécanisme d'extension du genou, transmettant directement la force des puissants muscles quadriceps pour redresser la jambe.

Le **tendon du quadriceps** relie les quatre muscles du quadriceps à la partie supérieure de la rotule [23], tel qu'illustré dans [figure 1.7](#). Il travaille en synergie avec le tendon rotulien pour permettre l'extension du genou [26]. Le tendon du quadriceps constitue le point d'attache proximal des muscles quadriceps à la rotule, et représente une composante intégrale du système de transmission de la force nécessaire à du tibia.

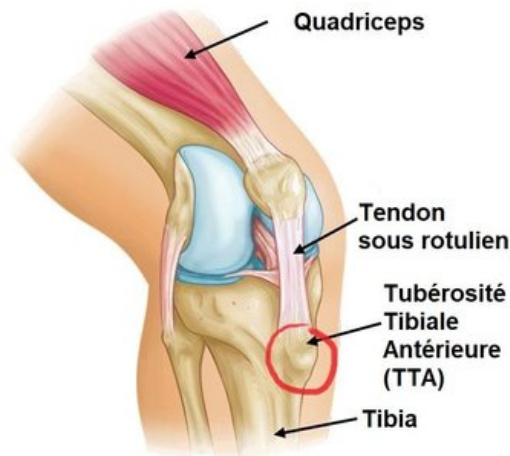


FIGURE 1.8. Le tendon rotulien [27].

l'extension du genou.

Les **tendons des ischio-jambiers** s'insèrent autour de l'articulation du genou, contribuant à la flexion et à la rotation [23], comme on peut le voir dans [figure 1.7](#). Les tendons des ischio-jambiers, s'insérant à l'arrière du genou, sont responsables de la flexion de la jambe et contribuent également aux mouvements de rotation.

Il est important de noter que les tendons autour du genou peuvent être affectés par l'arthrose, entraînant une tendinopathie⁶ (par exemple, la tendinopathie rotulienne) [28]. Cela peut constituer une source de douleur supplémentaire, s'ajoutant à celle provoquée par l'arthrose articulaire [29]. L'arthrose peut être associée à des problèmes tendineux autour du genou, ce qui peut contribuer à la douleur et aux limitations fonctionnelles, nécessitant une prise en compte dans les recommandations d'exercices.

1.2.4 Les ligaments

Les ligaments sont de fortes bandes fibreuses de tissu conjonctif qui relient les os entre eux, assurant la stabilité de l'articulation et limitant les mouvements excessifs ou anormaux. Les ligaments du genou sont divisés en ligaments collatéraux et en ligaments croisés [22].

Les ligaments collatéraux comprennent [22] :

- **Le ligament collatéral médial (LCM)** – situé sur la face interne (médiale) du genou, relie le fémur au tibia, comme illustré dans [figure 1.7](#). Il assure la stabilité de la face médiale du genou, empêchant un mouvement excessif

6. Une tendinopathie est un terme général qui désigne une affection douloureuse touchant un tendon.

valgus⁷ et la rotation externe.

- **Le ligament collatéral latéral (LCL)** – situé sur la face externe (latérale) du genou, relie le fémur au péroné, comme représenté dans [figure 1.7](#). Il assure la stabilité de la face latérale du genou, empêchant un mouvement excessif en varus⁸ et la rotation externe.

Les ligaments croisés, situés à l'intérieur de l'articulation du genou, se croisent en formant un « X » (voir [figure 1.7](#)) et contrôlent les mouvements antéro-postérieurs du genou. Ils comprennent [22] :

- **Le ligament croisé antérieur (LCA)** – situé à l'avant du genou, empêche le déplacement antérieur du tibia sur le fémur et le roulement postérieur du fémur lors de la flexion, et il résiste également au déplacement rotatoire [30].
- **Le ligament croisé postérieur (LCP)** – situé à l'arrière du genou, derrière le LCA, empêche le déplacement postérieur (vers l'arrière) du tibia sur le fémur et le mouvement antérieur excessif du fémur sur le tibia, et est souvent considéré comme le principal stabilisateur du genou [23, 31].

Dans l'arthrose, les ligaments autour du genou peuvent s'affaiblir ou s'étirer. Cela peut contribuer à l'instabilité de l'articulation [4]. Une lésion ligamentaire est considérée comme une cause potentielle d'arthrose [32]. L'arthrose peut provoquer des modifications ligamentaires, induisant une laxité et une instabilité accrues, ce qui peut contribuer davantage à la progression de la maladie et avoir un impact sur les recommandations d'exercices.

1.2.5 Les ménisques

Les ménisques sont deux coussinets de fibrocartilage en forme de croissant situés entre le fémur et le tibia dans chaque genou [23]. Ils jouent le rôle de coussinets amortisseurs, augmentant la congruence entre la surface articulaire tibiale et les condyles fémoraux arrondis [33].

Le **ménisque médial**, situé sur la face interne du genou [23], comme révélé dans [figure 1.9](#), a une forme de "C" et est moins mobile. Il recouvre 51% à 74% de la surface articulaire médiale [34]. En raison de sa forme et de ses attaches, le ménisque médial joue un rôle crucial dans la transmission des charges et la stabilité de la face interne du genou, et son lien avec le LCM le rend vulnérable aux lésions en cas de traumatisme du LCM.

7. Un mouvement vers l'intérieur.

8. Un mouvement vers l'extérieur.

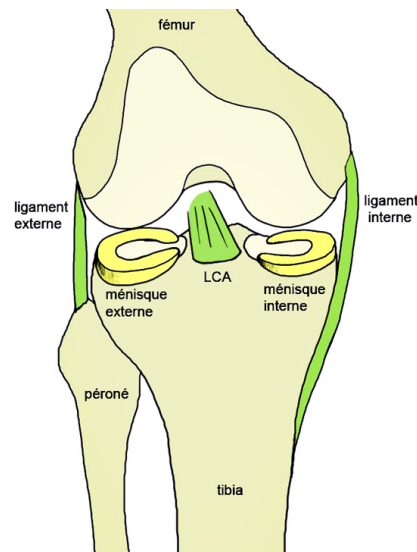


FIGURE 1.9. Les ménisques du genou [35].

Le **ménisque latéral**, situé sur la face externe du genou, comme illustré dans figure 1.9, est plus circulaire et plus mobile que le ménisque médial [23]. Il recouvre une plus grande partie de la surface articulaire tibiale (75% à 93%) [34]. Grâce à sa plus grande mobilité, le ménisque latéral contribue à la répartition des charges et à la stabilité de la face externe du genou, et ses attaches particulières lui confèrent une amplitude de mouvement supérieure à celle du ménisque médial.

Les déchirures méniscales sont fréquentes, surtout avec l'âge et chez les personnes souffrant d'arthrite du genou [23]. Les déchirures dégénératives peuvent survenir en l'absence de traumatisme spécifique et sont fréquemment associées à l'arthrose [36]. Les lésions méniscales peuvent entraîner une usure du cartilage et augmenter le risque de développer de l'arthrose ou d'accélérer sa progression [37].

1.2.6 Fonctionnement du genou

L'articulation du genou fonctionne principalement comme une articulation synoviale de type charnière, permettant la flexion et l'extension [22]. Elle autorise également un léger degré de rotation médiale et latérale, en particulier lorsque le genou est fléchi [23]. Les muscles, principalement le quadriceps et les ischio-jambiers, génèrent les mouvements principaux de flexion et d'extension. Les tendons transmettent ces forces musculaires aux os (fémur et tibia) via la rotule. Les ligaments, collatéraux et croisés, jouent un rôle essentiel en assurant la stabilité durant ces mouvements, en empêchant les déplacements excessifs dans différents plans. Les ménisques ont quant à eux une fonction cruciale d'absorption des chocs et de répartition des charges lorsque le genou bouge et supporte le poids, protégeant ainsi

le cartilage articulaire. L’arthrose perturbe la coordination des structures du genou, réduisant sa mobilité ainsi que la capacité à effectuer des mouvements complexes.

1.3 Diagnostic de l’arthrose et de la gonarthrose

L’arthrose (Osteoarthritis en anglais, OA) est la forme d’arthrite la plus fréquente dans le monde, touchant une large partie de la population et constituant une cause majeure de douleur chronique et de handicap [38, 39]. La gonarthrose, forme la plus courante d’arthrose, affecte spécifiquement l’articulation du genou, qui est la plus fréquemment atteinte [40, 41]. Sa prévalence augmente avec l’âge, affectant une proportion croissante de la population âgée [42], ce qui constitue un défi majeur pour les systèmes de santé. À l’échelle mondiale, environ 20% de la population est touchée, avec des variations régionales [43]. Au-delà des douleurs physiques, l’arthrose engendre également une détresse psychologique et des conséquences socioéconomiques importantes [44].

En clinique, la gonarthrose peut être diagnostiquée par deux approches principales. La première repose sur des examens radiologiques (radiographie, imagerie par résonance magnétique - IRM) ou arthroscopiques, qui permettent d’évaluer l’intégrité des structures du genou. Cependant, ces examens sont réalisés dans un contexte statique et ne renseignent pas sur la fonction dynamique de l’articulation [45]. Pour contourner cette limitation, la machine KneeKG permet une analyse dynamique du genou, en extrayant les composantes cinématiques pendant un cycle de marche d’un patient.

1.4 KneeKG : la machine utilisée pour le diagnostic du genou

KneeKG est une technologie développée par l’entreprise *Emovi* (Emotion • Movement • Vision [46]), qui constitue la première méthode permettant de mesurer avec précision l’alignement dynamique 3D du genou. Ce système fournit des données objectives, aidant les chirurgiens à prendre des décisions éclairées pour corriger l’alignement du genou et restaurer sa fonction tridimensionnelle [46].

L’exosquelette illustré dans la [figure 1.10](#) est un dispositif de fixation du genou conçu pour éliminer les artefacts dus aux mouvements cutanés et musculaires. Il est placé sur le membre inférieur du patient. Grâce à un système optique, la cinématique 3D du genou est capturée pendant la marche sur un tapis roulant à vitesse confortable et auto-déterminée, dans un contexte dynamique et en charge. Les données

Chapitre 1. Contexte et problématique

sont enregistrées selon les trois plans de mouvement.

Le système KneeKG peut être utilisé en clinique, à l'hôpital ou en centre d'imagerie. Après une phase d'échauffement sur tapis roulant d'environ cinq minutes, le clinicien fixe les composants du KneeKG sur la jambe du patient pour suivre le mouvement articulaire. Après un calibrage, le système enregistre deux essais de marche de 45 secondes chacun, permettant d'observer plusieurs cycles de marche. Les données recueillies sont ensuite analysées objectivement, fournissant des informations fiables sur la fonction du genou.



FIGURE 1.10. L'exosquelette du KneeKG [47].

1.5 Les données cinématiques

Les données cinématiques de l'articulation du genou correspondent à des mesures dynamiques du mouvement obtenues lors de l'analyse du cycle de marche, comme illustré dans la [figure 1.11](#), à l'aide du système KneeKG. Ces données sont de plus en plus utilisées pour quantifier la fonction du genou [48], comprendre les altérations pathologiques [49] et évaluer la progression des pathologies ainsi que leur impact sur la marche [50, 51, 52]. Elles offrent aussi des opportunités pour le diagnostic [53], la classification [54] et le traitement des pathologies musculo-squelettiques du genou, notamment la gonarthrose [55].

Plusieurs études [51, 54, 56, 57] ont utilisé ces données pour distinguer l'arthrose des genoux symptomatiques des sujets asymptomatiques [58] et classer les niveaux de gravité de la maladie.

Les données cinématiques incluent notamment les forces de réaction au sol (GRF

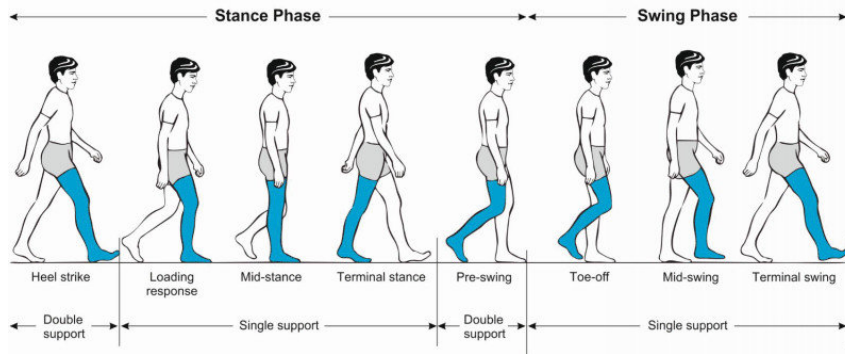


FIGURE 1.11. Le cycle de marche [59].

– Ground Reaction Force) décomposées en trois composantes orthogonales F_x , F_y , F_z (antéro-postérieure, médio-latérale et verticale), comme illustré dans la [figure 1.12](#), ainsi que les déplacements relatifs du tibia et du fémur dans les trois axes. Ces mesures caractérisent les mouvements du genou (flexion/extension, abduction/adduction, rotation interne/externe) en conditions dynamiques [56].

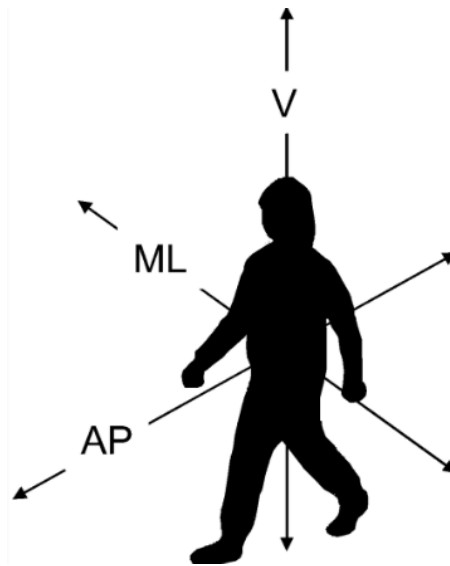


FIGURE 1.12. Les trois composantes des données cinématiques [60].

Les données cinématiques présentent plusieurs avantages :

- *Elles reflètent l'état fonctionnel du genou* : contrairement aux images statiques, ces données révèlent la dynamique réelle du genou en mouvement, sa mobilité, ses limitations ou compensations, offrant une vision fidèle de la fonction articulaire.
- *Ce sont des mesures objectives, non invasives et sans irradiation* : collectées par des capteurs externes (voir [figure 1.13](#)), elles évitent les biais d'interpréta-

Chapitre 1. Contexte et problématique

tion subjective et les risques liés aux radiations.

- *Elles permettent de personnaliser la prise en charge* : les professionnels de santé peuvent suivre l'évolution de la maladie, adapter les traitements et proposer des parcours de soins sur mesure basés sur les données dynamiques du genou.

En résumé, ces données permettent une prise en charge plus précise, sûre et adaptée des patients atteints d'arthrose du genou [61].



FIGURE 1.13. Les capteurs de la machine KneeKG [62].

1.6 Problématique

La gonarthrose est une maladie chronique touchant l'articulation du genou, provoquant des douleurs, raideur et une perte de mobilité, altérant significativement la qualité de vie. Pour soulager ces symptômes, des exercices physiques sont souvent prescrits afin de renforcer les muscles périarticulaires, améliorer la mobilité et réduire la douleur. Toutefois, chaque patient présente des besoins spécifiques liés à sa morphologie, son âge, son niveau de douleur et ses caractéristiques biomécaniques. Proposer un même programme d'exercices à tous peut donc s'avérer inefficace.

Il est dès lors essentiel de personnaliser ces exercices selon les données individuelles. Aujourd'hui, grâce à des technologies comme KneeKG, il est possible de mesurer avec précision le mouvement du genou en dynamique. Ces données peuvent être exploitées via des techniques d'intelligence artificielle, détaillées dans les chapitres suivants, afin de concevoir un système de recommandation d'exercices personnalisés. Ce travail vise ainsi à développer un système capable d'identifier, pour chaque patient, les exercices les plus adaptés en fonction de ses données biomécaniques. Cette approche, inspirée des systèmes de recommandation utilisés dans

Chapitre 1. Contexte et problématique

d'autres domaines, vise à optimiser les bénéfices thérapeutiques et l'adhésion au traitement.

1.7 Conclusion

Ce chapitre a présenté l'anatomie et le fonctionnement du genou, ainsi que la gonarthrose, maladie qui endommage l'articulation et engendre douleurs et difficultés motrices. Nous avons introduit la technologie KneeKG, qui analyse finement le mouvement articulaire afin de mieux comprendre les besoins spécifiques des patients. Enfin, nous avons mis en lumière la nécessité d'adapter les exercices physiques, à l'aide de données cinématiques dynamiques, pour améliorer la prise en charge des personnes atteintes d'arthrose.

Le chapitre suivant sera consacré à une revue des travaux existants sur les systèmes de recommandation en rééducation, les algorithmes d'intelligence artificielle pertinents, et les bases méthodologiques pour concevoir un système adapté à ce contexte médical.

Chapitre 2

Systèmes de recommandations et GNN

2.1 Introduction

Dans le paysage numérique contemporain, les individus sont fréquemment confrontés à une surabondance d'informations et d'options dans divers domaines, allant des plateformes de commerce électronique aux services de diffusion en continu, en passant par les ressources en ligne. Cette surcharge informationnelle complique la tâche des utilisateurs lorsqu'il s'agit d'identifier les contenus ou produits correspondant à leurs besoins et préférences spécifiques. Les systèmes de recommandation se sont ainsi imposés comme une classe essentielle d'algorithmes d'apprentissage automatique, visant à relever ce défi en prédisant, filtrant et identifiant les éléments susceptibles d'intéresser un utilisateur parmi un large éventail de choix possibles [63].

Ces systèmes permettent de personnaliser l'exploration des espaces informationnels en hiérarchisant les éléments pertinents pour un utilisateur donné, réduisant ainsi l'effort de recherche tout en améliorant l'expérience globale d'interaction. L'évolution des systèmes de recommandation a connu une progression significative, passant des méthodes classiques de filtrage collaboratif à des approches plus avancées reposant sur l'apprentissage profond. Parmi ces avancées, les réseaux de neurones graphiques (Graph Neural Networks, GNNs) se distinguent par leur capacité à modéliser des relations complexes au sein des données, constituant ainsi un changement de paradigme dans la manière dont les recommandations sont générées.

2.2 Contexte général des systèmes de recommandation

2.2.1 Définition d'un système de recommandations

Un système de recommandation, ou RS, a pour but de générer des recommandations pour un ensemble d'utilisateurs. Ces recommandations peuvent être sous forme de produits qui peuvent être des livres, des films, des chansons, etc. (Souvent les termes utilisés sont *users* pour désigner les utilisateurs, et *items* pour désigner les objets à recommander) qui peuvent intéresser l'utilisateur. La création d'un système de recommandation dépend du domaine et des caractéristiques particulières des données disponibles [64]. Le système de recommandation utilise les données historiques des utilisateurs afin de générer des recommandations, alors ceux-ci jouent un rôle important [65].

Ricci *et al* [66] ont défini un système de recommandation comme suit :

"Les systèmes de recommandation (RS) sont des outils logiciels et techniques qui proposent des suggestions d'articles utiles à un utilisateur [67, 68, 69]. Les suggestions concernent divers processus de prise de décision, tels que les articles à acheter, la musique à écouter ou les actualités en ligne à lire."

Les systèmes de recommandation se sont généralisés dans de nombreux domaines : e-commerce (Amazon), divertissement (Netflix, Spotify), réseaux sociaux (YouTube, Facebook), éducation (MOOCs), et santé (recommandations de traitements, suivi personnalisé).

2.2.2 Classification des approches de recommandation

Il existe plusieurs approches¹ de systèmes de recommandation. Parmi les plus connus, on retrouve les systèmes de recommandation basés sur le contenu, les systèmes collaboratifs, ainsi que les systèmes hybrides qui combinent plusieurs approches. Par ailleurs, il existe également des systèmes de recommandation plus spécifiques, adaptés à des domaines particuliers d'application comme illustrés dans [figure 2.1](#).

1. Une « approche de recommandation » est un modèle de mise en pratique d'une recommandation [70].

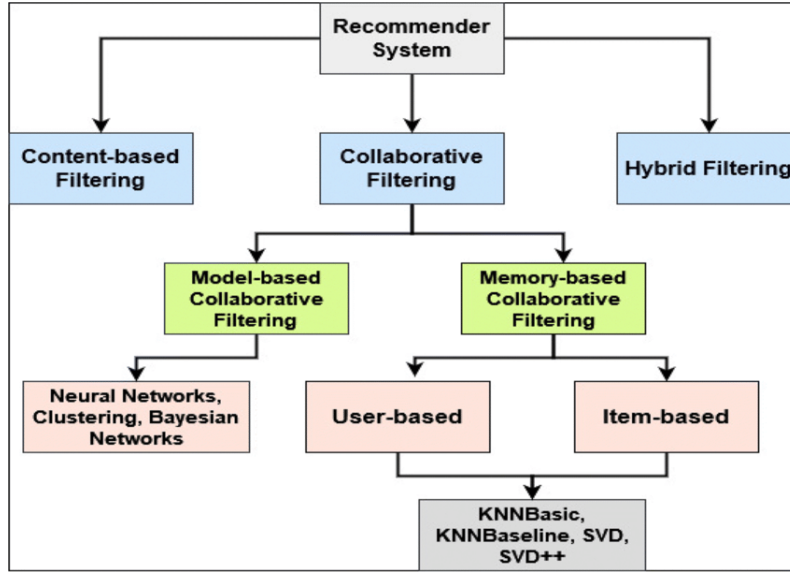


FIGURE 2.1. Les approches de systèmes de recommandation [71].

Approche basée sur le contenu

Le filtrage basé sur le contenu (Content-Based Filtering, CBF) est l’une des approches les plus courantes dans les systèmes de recommandation [66]. Il repose sur l’analyse des interactions passées d’un utilisateur afin de modéliser ses préférences — un processus appelé *user modeling* — en inférant ses intérêts à partir des éléments déjà consultés ou appréciés [70]. Le principe clé est que les utilisateurs ont tendance à aimer des contenus similaires à ceux qu’ils ont déjà aimés [72], comme illustré dans figure 2.2. Cette approche permet une personnalisation fine, sans dépendre des préférences d’autres utilisateurs [70]. Toutefois, elle présente certaines limites, telles qu’une faible sérendipité² et un risque de surspécialisation, où les recommandations deviennent trop similaires au contenu déjà consommé [66]. De plus, elle ne prend généralement pas en compte la qualité ou la popularité des éléments, ce qui peut nuire à la diversité des suggestions. Malgré ces limites, le CBF reste largement utilisé, notamment dans des domaines comme la recommandation d’articles scientifiques [70].

Approche collaborative

Le filtrage collaboratif (Collaborative Filtering, CF) repose sur l’idée que les utilisateurs apprécient généralement des éléments que d’autres, aux goûts similaires, ont aussi aimés, comme illustré dans figure 2.3. Deux utilisateurs sont considérés similaires lorsqu’ils donnent des évaluations proches aux mêmes items. Une fois

2. Capacité du système à recommander des éléments pertinents mais inattendus.

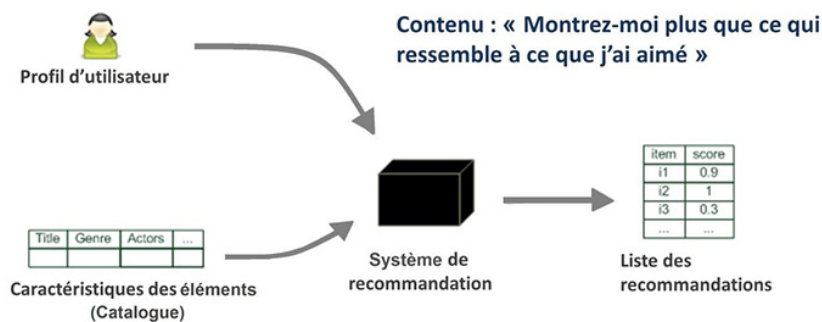


FIGURE 2.2. Approche basée sur le contenu [73].

ces profils détectés, les éléments appréciés par l'un peuvent être proposés à l'autre [70, 72].

Comparé au filtrage basé sur le contenu (CBF), le CF présente plusieurs avantages [70] :

- (i) Il ne dépend pas des caractéristiques des items [74].
- (ii) Il exploite des jugements subjectifs réels émis par les utilisateurs [75].
- (iii) Il favorise des recommandations plus surprenantes et personnalisées, grâce aux similarités entre utilisateurs [74].

Cependant, cette approche n'est pas exempte de limites. Elle est sensible au problème du *cold-start*³, et peut échouer face aux utilisateurs dits *gray sheep*⁴, dont les goûts ne sont partagés par aucun segment clair [72, 79].

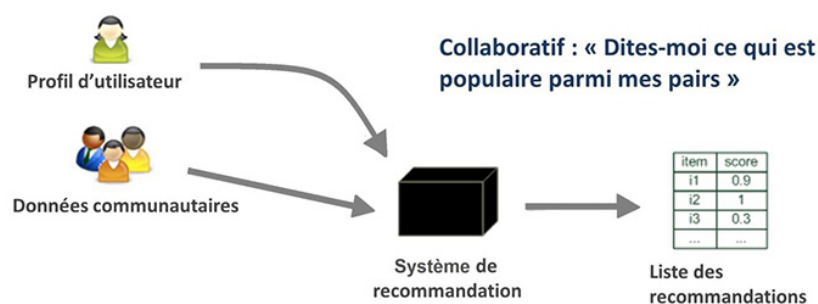


FIGURE 2.3. Approche collaborative [73].

3. Le cold-start désigne l'incapacité à faire des recommandations pertinentes en l'absence de données suffisantes sur les nouveaux utilisateurs ou éléments [76, 77].

4. Utilisateurs aux préférences atypiques, ne correspondant à aucun groupe identifiable [78].

Approche hybride

Pour combiner les avantages du filtrage basé sur le contenu et du filtrage collaboratif, plusieurs approches hybrides ont été développées, comme illustré dans [figure 2.4](#). Une méthode simple consiste à laisser chaque technique générer ses recommandations, puis à fusionner les résultats pour produire une liste finale [\[80\]](#). Les systèmes de recommandation hybrides (Hybrid RS) associent ainsi plusieurs stratégies pour renforcer leurs points forts et compenser leurs faiblesses.

Un des premiers exemples est *Fab*, un méta-système de recommandation de sites web combinant CF (pour repérer des utilisateurs aux goûts similaires) et CBF (pour identifier des contenus proches) [\[81\]](#).

Par exemple, la fonction « entreprises que vous souhaitez suivre » sur LinkedIn combine les deux approches : le CF évalue la similarité avec les entreprises déjà suivies, tandis que le CBF vérifie que le secteur ou la localisation correspondent aux intérêts de l'utilisateur [\[82\]](#).

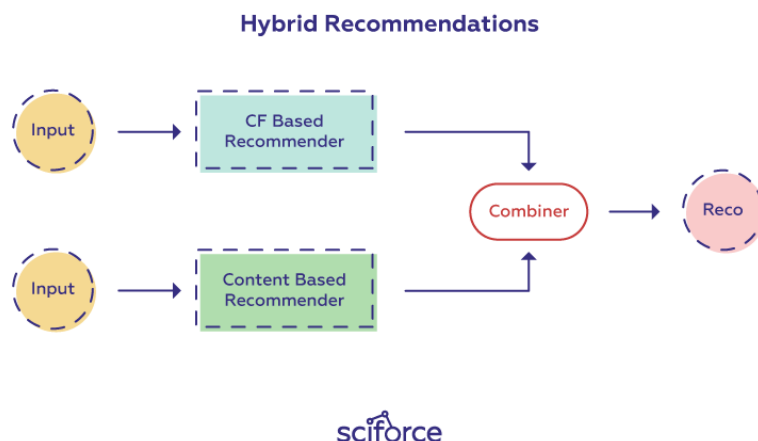


FIGURE 2.4. Approche hybride [\[83\]](#).

2.3 Application des systèmes de recommandation en santé

Les systèmes de recommandation en santé (Healthcare Recommender Systems, HRS) tirent profit des approches de systèmes de recommandation déjà citées, telles que le filtrage collaboratif (CF), le filtrage basé sur le contenu (CBF), ainsi que des méthodes hybrides, afin de proposer des traitements, des médicaments ou des plans de soins personnalisés [\[84\]](#). Ces systèmes s'appuient sur une variété de données

cliniques — antécédents médicaux, symptômes, diagnostics, prescriptions en cours, et même résultats d’examens — pour identifier des patients au profil similaire et recommander des options thérapeutiques ayant montré leur efficacité dans des cas analogues [85].

Un exemple d’un HRS est IBM Watson [86, 87], qui analyse de vastes ensembles de données cliniques et suggère des traitements pertinents en utilisant des méthodes d’apprentissage machine. Selon IBM, 81% des cadres du secteur médical familiers avec Watson reconnaissent son impact positif sur leur prise de décision, soulignant le rôle croissant de l’analyse des données dans la pratique médicale [72, 79].

Au-delà des techniques traditionnelles, les HRS intègrent également des algorithmes avancés, tels que les colonies de fourmis [81], le clustering, les arbres de décision, la régression logistique, les machines à vecteurs de support (SVM), ou encore des approches plus récentes comme matrix factorization, les ontologies médicales, le raisonnement sémantique, ou le traitement automatique du langage naturel (NLP) [84, 85, 88]. Ces techniques sont cruciales pour représenter la complexité des données médicales et modéliser les préférences ou les besoins spécifiques des patients.

Parmi ces méthodes, les approches basées sur la connaissance (knowledge-based)⁵ jouent un rôle central. Elles permettent d’intégrer des informations explicites sur les patients, comme les allergies ou les interactions médicamenteuses, dans le processus de recommandation [89]. Selon Sadasivam et al. [88], ces approches permettent également de contourner le problème du démarrage à froid — l’un des plus grands défis des systèmes de recommandation [76, 90] — en générant des suggestions même en l’absence d’historique d’évaluations ou de données similaires.

Ainsi, la distribution et l’intégration des algorithmes intelligents dans les systèmes de santé permettent non seulement d’améliorer la pertinence des recommandations, mais également de soutenir les professionnels de santé dans des prises de décision complexes, en offrant un appui objectif fondé sur des preuves cliniques et statistiques [91].

5. Les systèmes de recommandation basés sur la connaissance (Knowledge-Based Recommender Systems) reposent sur des règles explicites et des connaissances préalables concernant les utilisateurs et les éléments à recommander. Contrairement aux approches collaboratives ou basées sur le contenu, ils n’ont pas besoin d’historiques d’interaction. Ils utilisent plutôt des contraintes explicites (par exemple : « un patient diabétique ne doit pas recevoir de médicament X ») pour générer des recommandations précises, souvent dans des contextes critiques comme la santé ou les systèmes experts

2.4 Système de recommandation collaboratif

À la base, un système de recommandation collaboratif est conçu pour prédire l'intérêt d'un utilisateur pour de nouveaux éléments en se basant sur les préférences collectives d'une communauté d'utilisateurs [92, 93]. Les composantes fondamentales d'un tel système comprennent un ensemble d'utilisateurs, un ensemble d'éléments et un mécanisme pour représenter les préférences des utilisateurs pour ces éléments, souvent appelé fonction d'utilité [93, 94]. Cette fonction d'utilité peut prendre diverses formes, telles que des évaluations explicites sur une échelle numérique ou des indicateurs implicites de préférence, comme l'historique des achats ou le temps de visionnage. Généralement, ces préférences sont organisées dans une matrice *user-item* comme illustré dans figure 2.5, où chaque ligne représente un utilisateur, chaque colonne représente un élément et les entrées indiquent la préférence de l'utilisateur pour cet élément [95]. Une caractéristique importante de cette matrice dans les applications du monde réel est sa sparsité, ce qui signifie que la plupart des utilisateurs n'ont interagi qu'avec un petit sous-ensemble du total des éléments disponibles [96]. Cette rareté présente un défi majeur pour les algorithmes de filtrage collaboratif, car ils doivent inférer les préférences manquantes en se basant sur une quantité limitée de données observées.



				
John 	5	1	3	5
Tom 	?	?	?	2
Alice 	4	?	3	?

FIGURE 2.5. La matrice user-item [97].

Le filtrage collaboratif se distingue des autres approches de systèmes de recommandation, par son recours principal aux interactions user-item [98]. Les systèmes basés-contenu, en revanche, se concentrent sur les attributs ou les caractéristiques des éléments, recommandant des éléments similaires à ceux qu'un utilisateur a préférés dans le passé [99]. Cette approche nécessite des informations détaillées sur les items eux-mêmes pour déterminer la similarité [100]. Les systèmes hybrides visent à combiner les forces des méthodes collaboratives et basées sur le contenu, ce qui

conduit souvent à une amélioration de la précision et de la couverture des recommandations [93].

Dans le domaine du filtrage collaboratif, plusieurs formulations de problèmes clés existent [93]. La recommandation basée sur l'utilisateur implique l'identification des utilisateurs qui ont des schémas de préférences similaires à un utilisateur cible, puis la recommandation s'agit des éléments que ces utilisateurs similaires ont aimés mais que l'utilisateur cible n'a pas encore rencontrés [100], comme illustré dans figure 2.6. La recommandation basée sur l'élément, à l'inverse, se concentre sur l'identification des éléments similaires à ceux que l'utilisateur cible a aimés dans le passé et la recommandation s'agit de ces éléments similaires [93], comme illustré dans figure 2.6.

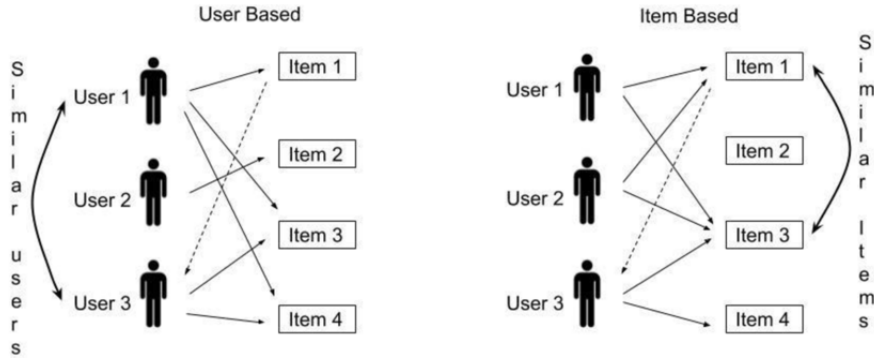


FIGURE 2.6. La recommandation basée sur l'utilisateur et celle basé sur l'élément [101].

Une autre formulation consiste à prédire les évaluations individuelles, où le système vise à estimer l'utilité ou l'évaluation spécifique qu'un utilisateur donnerait à un élément qu'il n'a pas encore évalué [93]. Enfin, de nombreux systèmes visent à générer des recommandations *top-N*, fournissant une liste classée des N éléments qu'un utilisateur est le plus susceptible de trouver intéressants [102]. Dans le contexte des recommandations d'exercices pour les patients atteints d'OA (Osteoarthritis, l'arthrose en français), les approches basées sur l'utilisateur (trouver des patients similaires et leurs exercices préférés) et basées sur l'élément (trouver des exercices similaires à ceux qu'un patient a appréciés) pourraient être pertinentes [100], et ce serait l'objet de notre conception.

2.4.1 Types d'approches de filtrage collaboratif

Les techniques de filtrage collaboratif peuvent être globalement classées en trois catégories : les méthodes basées sur la mémoire, les méthodes basées sur un modèle

et les approches hybrides [103]. Nous présentons ci-après ces méthodes et certaines des techniques qu'elles utilisent.

Méthodes basées sur la mémoire

Ces approches exploitent directement l'ensemble des données d'interaction utilisateur-élément pour générer des recommandations. Elles reposent sur l'identification d'utilisateurs ou d'éléments similaires à l'utilisateur ou à l'élément cible, puis sur l'utilisation des préférences de ces voisins afin de prédire l'utilité des éléments pour la cible [103].

- **Le filtrage collaboratif basé sur l'utilisateur** est une technique fondée sur la mémoire, qui recommande des éléments à un utilisateur cible en s'appuyant sur les préférences d'autres utilisateurs ayant manifesté un comportement similaire [96]. L'algorithme identifie les utilisateurs dont les goûts et préférences sont proches de ceux de l'utilisateur cible, puis prédit l'utilité d'un élément en fonction des évaluations attribuées à cet élément par ces utilisateurs similaires [93] comme illustré dans figure 2.7. La similarité entre les utilisateurs est généralement calculée en comparant les lignes de la matrice user-item à l'aide de mesures telles que la similarité cosinus ou le coefficient de corrélation de Pearson [96]. Pour notre cas on peut utiliser les données cinématiques pour identifier la similarité entre utilisateurs.

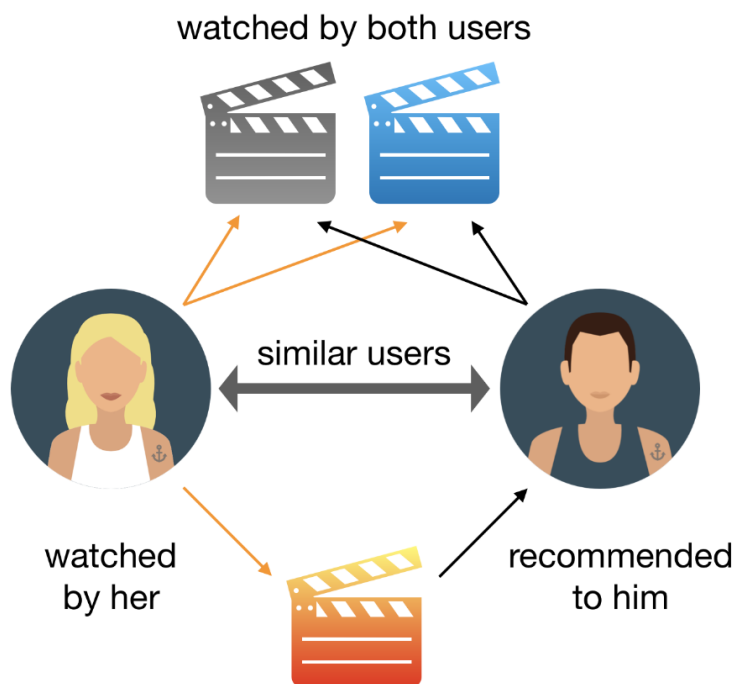


FIGURE 2.7. Le filtrage collaboratif basé sur l'utilisateur.

- **Le filtrage collaboratif basé sur l'élément** est une autre technique fondée sur la mémoire, qui recommande de nouveaux éléments à un utilisateur cible en se basant sur la manière dont cet utilisateur a interagi avec des éléments similaires [96]. L'algorithme identifie les éléments proches de ceux que l'utilisateur a appréciés par le passé, puis prédit l'utilité d'un nouvel élément en fonction du degré d'appréciation manifesté envers les éléments similaires [96] comme illustré dans figure 2.8. La similarité entre les éléments est déterminée en analysant les utilisateurs ayant interagi avec eux, et se calcule généralement en comparant les colonnes de la matrice user-item [93]. Dans notre cas, cette approche consisterait à suggérer des exercices présentant des caractéristiques similaires à ceux que le patient a déjà pratiqués ou évalués positivement, en tenant compte de critères comme l'intensité, le type d'exercice, ou les parties du corps concernées [104].

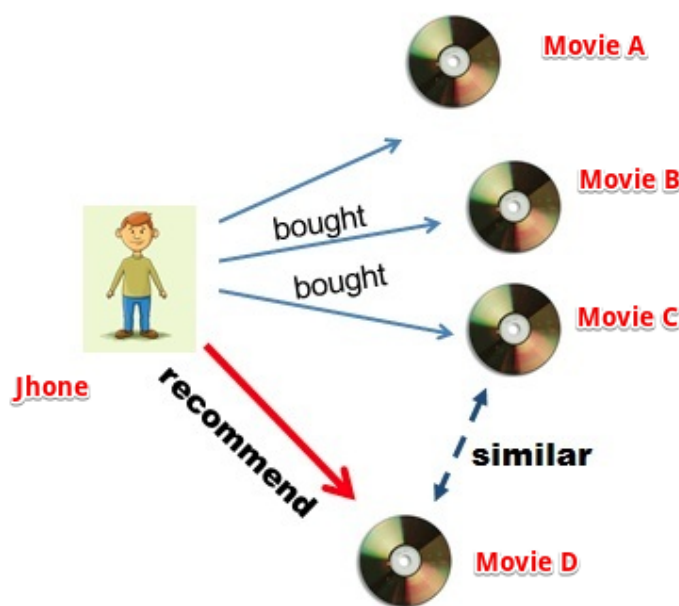


FIGURE 2.8. Le filtrage collaboratif basé sur l'élément [105].

Méthodes basées sur un modèle

Ces approches utilisent les données d'interaction user-item pour apprendre un modèle, qui est ensuite utilisé pour faire des prédictions [93]. L'objectif est de capturer les schémas et les relations sous-jacentes dans les données [106].

- **La factorisation matricielle** est une technique populaire qui consiste à dé-

composer la matrice user-item, en le produit de deux matrices de plus faible dimension représentant les facteurs latents des users et des items [93]. Des méthodes telles que la décomposition en valeurs singulières (SVD) et la factorisation matricielle non négative (NMF) sont couramment utilisées [107] comme le montre [figure 2.9](#). Ces facteurs latents peuvent représenter des attributs non observés qui influencent les préférences des utilisateurs ou les caractéristiques des éléments [108]. Dans notre cas, on peut l'utiliser pour d'identifier des préférences latentes pour certains types d'exercices ou des bénéfices spécifiques liés à certaines routines thérapeutiques.

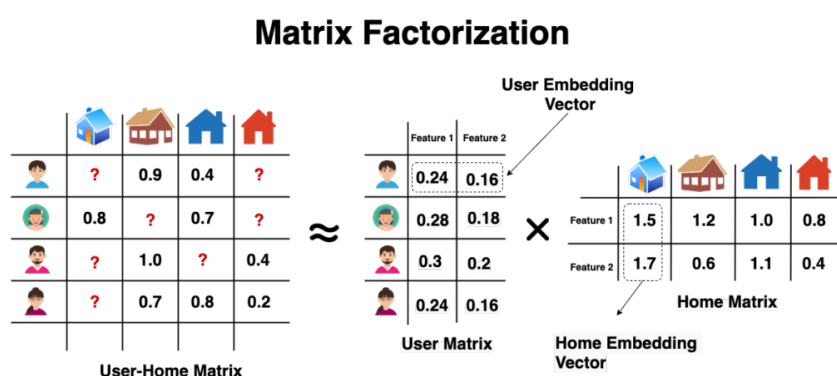


FIGURE 2.9. La factorisation matricielle [109].

- **Les techniques de clustering** consistent à regrouper les utilisateurs ou les éléments en clusters⁶ selon leur similarité [110], comme illustré dans [figure 2.10](#). Une fois les clusters formés, les prédictions pour un utilisateur peuvent être établies à partir des préférences des autres membres de son cluster [111]. Pour notre cas, le clustering pourrait permettre de regrouper les patients ayant des comportements ou des réponses similaires face à certains exercices, facilitant ainsi la recommandation d'exercices au sein du cluster auquel un nouveau patient est associé.
- **Les réseaux de neurones profonds** qui sont illustré dans [figure 2.11](#) ont été introduits pour le filtrage collaboratif afin de remplacer le produit scalaire linéaire de la factorisation matricielle par des architectures non linéaires capables de modéliser des interactions complexes entre utilisateurs et éléments [113]. Dans ce cadre, des variantes comme GMF, MLP ou NeuMF exploitent des perceptrons multicouches pour apprendre directement la fonction d'interaction utilisateur-élément à partir de données implicites [114].

6. Un cluster désigne un groupe d'éléments (par exemple, des utilisateurs ou des objets) qui partagent des caractéristiques similaires selon une certaine métrique de distance ou de similarité.

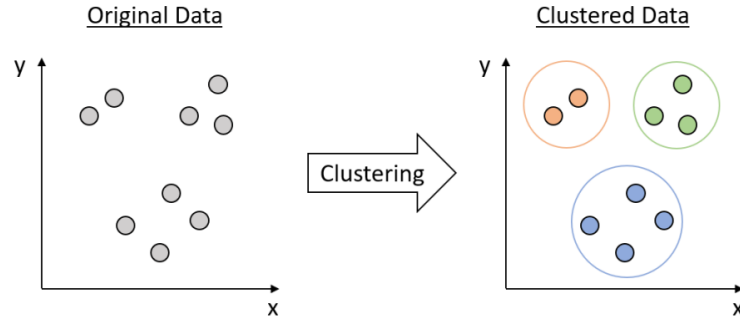


FIGURE 2.10. Clustering [112].

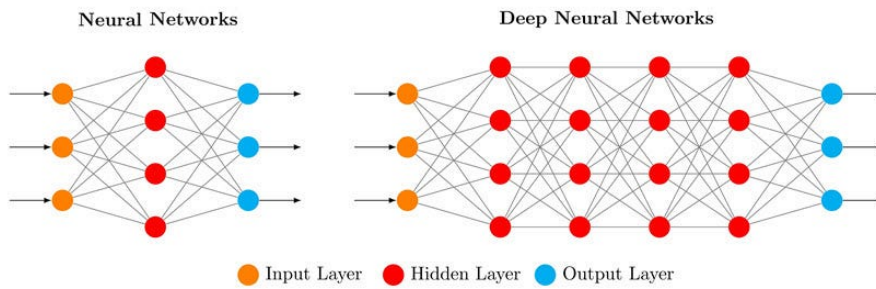


FIGURE 2.11. Les réseaux de neurones profonds [115].

Plus récemment, les Graph Neural Networks (GNN), voir [figure 2.12](#) ont été appliqués aux systèmes de recommandation collaboratifs : en considérant l'information utilisateur-élément comme un graphe, les GNN agréent les messages des voisins pour capturer les dépendances structurelles et relationnelles. Les études montrent que ces modèles, grâce à leur mécanisme de passage de messages, obtiennent des gains de performance significatifs par rapport aux approches traditionnelles telles que sur des jeux de données standards [116].

2.4.2 Avantages et limites du filtrage collaboratif

Le filtrage collaboratif constitue l'une des approches les plus éprouvées et largement adoptées dans le domaine des systèmes de recommandation, en raison de ses nombreux avantages fonctionnels et méthodologiques [118]. L'un de ses principaux atouts réside dans sa capacité à générer des recommandations inattendues ou non triviales, permettant ainsi aux utilisateurs de découvrir de nouveaux centres d'intérêt non explicitement exprimés dans leurs interactions antérieures. Contrairement aux méthodes fondées sur le contenu, qui tendent à suggérer des éléments similaires à ceux déjà appréciés, le filtrage collaboratif exploite les préférences d'utilisateurs partageant des comportements analogues, facilitant ainsi la recommandation d'élé-

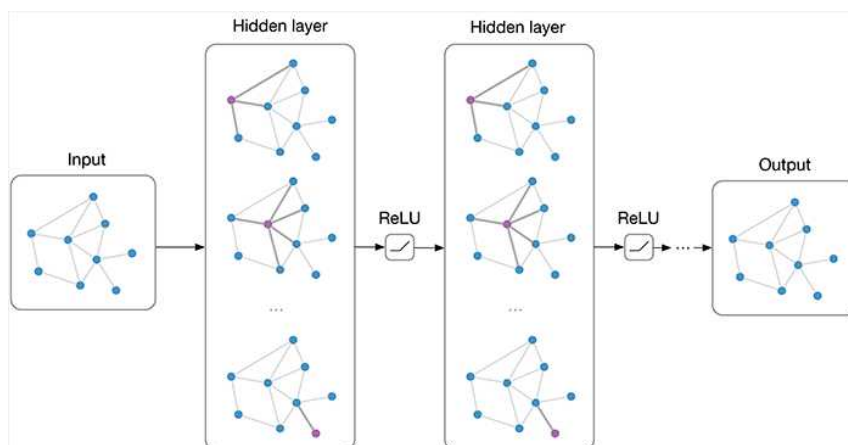


FIGURE 2.12. les Graph Neural Networks (GNN) [117].

ments populaires auprès de profils similaires, même en l’absence de caractéristiques communes [119].

Un avantage substantiel de cette approche est qu’elle ne repose pas sur une connaissance explicite des attributs ou du contenu des éléments recommandés, ce qui la rend particulièrement adaptée à des contextes où l’extraction de caractéristiques est imprécise, coûteuse ou subjectivement biaisée [100]. En s’appuyant exclusivement sur les interactions réelles des utilisateurs, le filtrage collaboratif permet de refléter de manière plus fidèle les préférences individuelles que ne le feraient des descriptions produites manuellement [120]. Il se distingue également par sa capacité à identifier des préférences latentes ou de niche au sein d’une communauté d’utilisateurs, souvent imperceptibles par d’autres techniques.

Par ailleurs, la performance de ces systèmes tend à s’améliorer avec l’accroissement du volume de données collectées, rendant les recommandations progressivement plus personnalisées et précises au fil du temps. Cette approche présente aussi une flexibilité remarquable, en identifiant des préférences croisées entre genres ou catégories, tout en restant sensible aux évolutions dynamiques des goûts des utilisateurs. Enfin, le filtrage collaboratif offre un rapport coût-efficacité favorable, en réduisant la nécessité d’une annotation manuelle des données, tout en assurant la diversité et la pertinence des recommandations, ce qui en fait une solution particulièrement adaptée à notre cas. [120].

2.5 Introduction aux Graph Neural Networks

L’évolution des systèmes de recommandation a connu une progression à travers diverses techniques, allant des méthodes de filtrage collaboratif de base jusqu’à des

approches plus sophistiquées reposant sur l'apprentissage profond. Récemment, les techniques de réseau neuronal graphique (Graph Neural Networks, GNNs) se sont imposées comme un outil puissant pour capturer des relations complexes au sein des données, et elles ont été largement utilisées dans les systèmes de recommandation [121], car la plupart des informations dans les systèmes de recommandation ont essentiellement une structure graphique [122, 123], et les GNNs ont une supériorité dans l'apprentissage de la représentation graphique, apportant ainsi un changement de paradigme dans la manière dont les recommandations sont générées [121]. À l'échelle de la recherche académique, de nombreuses études attestent que les modèles reposant sur les GNNs surpassent les approches antérieures et établissent de nouveaux résultats de référence sur les jeux de données publics [124, 125].

2.5.1 Définition des GNNs

Les Graph Neural Networks (GNNs), ou réseaux de neurones graphiques, constituent une classe d'algorithmes d'apprentissage profond spécialement conçus pour traiter des données pouvant être représentées sous forme de graphes [126]. Un graphe est une structure de données composée de nœuds⁷ (ou sommets) et d'arêtes⁸ reliant ces nœuds représentant respectivement des entités et leurs relations [126], comme illustré dans figure 2.13.

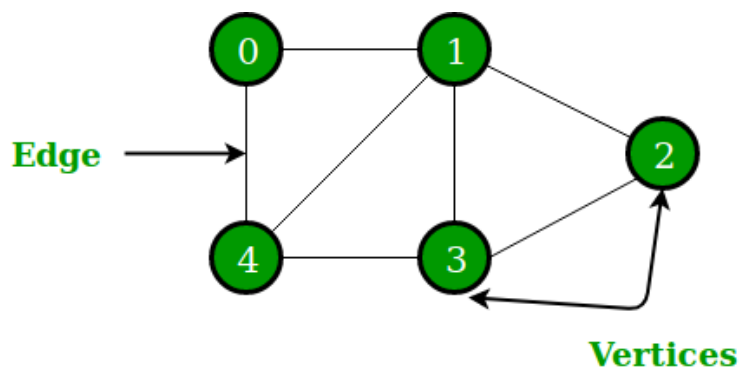


FIGURE 2.13. Exemple de structure de graphe [127].

De manière générale, les graphes peuvent être classés comme suit :

- **Graphe dirigé / non dirigé** : Un graphe dirigé est un graphe dont toutes les arêtes sont orientées d'un nœud vers un autre. Un graphe non dirigé peut être considéré comme un cas particulier de graphe dirigé, dans lequel chaque lien

7. *Vertices* en anglais.

8. *Edges* en anglais.

entre deux nœuds est représenté par une paire d'arêtes de directions opposées, voir [figure 2.14a](#).

- **Grphe homogène / hétérogène** : Un graphe homogène est constitué d'un seul type de nœuds et d'arêtes, tandis qu'un graphe hétérogène comporte plusieurs types de nœuds ou d'arêtes, voir [figure 2.14b](#).
- **Hypergraphe** : Un hypergraphe est une généralisation du graphe dans laquelle une arête (appelée hyperarête) peut relier un nombre quelconque de nœuds, voir [figure 2.14c](#).

Les GNNs sont particulièrement adaptés aux applications où les relations entre les points de données sont aussi importantes que les points eux-mêmes, comme les réseaux sociaux, les graphes de connaissances, les systèmes de recommandation et la bioinformatique [131], voir [figure 2.15](#). Contrairement aux réseaux neuronaux traditionnels, qui opèrent sur des données de type grille ou séquentielles, les GNNs peuvent gérer la complexité et l'hétérogénéité des données structurées en graphes, où chaque nœud peut avoir un nombre variable de connexions et des caractéristiques propres [132].

Le développement des GNNs s'inspire d'algorithmes d'apprentissage profond tels que les réseaux convolutifs (CNNs) et les réseaux récurrents (RNNs), mais ils sont adaptés à la nature non euclidienne⁹ des données en graphe [134]. L'idée centrale des GNNs est d'apprendre des représentations (ou *embeddings* en anglais) pour les nœuds, les arêtes ou l'ensemble du graphe en tenant compte à la fois des caractéristiques des nœuds et de la topologie du graphe [121].

Les *embeddings* sont des représentations vectorielles continues qui traduisent des entités complexes (comme des mots, des utilisateurs ou des nœuds) en vecteurs exploitables par des algorithmes d'apprentissage automatique, comme illustré dans [figure 2.16](#). Ils permettent de capturer la similarité sémantique¹⁰ ou structurelle¹¹ entre ces entités en les positionnant de manière significative dans un espace vectoriel. Les embeddings sont utilisés dans divers domaines comme le traitement du langage,

9. On parle de données euclidiennes lorsque les éléments sont organisés dans un espace régulier, comme : une image, un texte ou un tableau. À l'inverse, les données en graphe n'ont pas de structure régulière. Les relations entre les nœuds peuvent être arbitraires.

10. Une similarité sémantique signifie que deux entités ont des sens ou rôles proches, même si elles sont différentes en apparence. Par exemple, deux utilisateurs peuvent être considérés comme sémantiquement similaires s'ils aiment les mêmes genres de films ou ont des goûts comparables, même s'ils ne se connaissent pas.

11. Une similarité structurelle concerne la position ou le rôle des entités dans une structure comme un graphe. Par exemple, dans un réseau social représenté sous forme de graphe, deux utilisateurs peuvent être structurellement similaires s'ils occupent une place semblable dans le réseau, même s'ils ne sont pas connectés directement ou n'ont pas les mêmes intérêts.

la recommandation ou les graphes ; ils jouent un rôle crucial dans la transformation des données en caractéristiques apprenables et exploitables [135, 136].

La génération des embeddings dans les GNNs se fait par un processus d'échange de messages, où les nœuds échangent de l'information avec leurs voisins de façon itérative, en agrégeant et transformant ces messages pour mettre à jour leurs représentations. La capacité des GNNs à capturer ces relations et dépendances complexes explique leur popularité croissante dans diverses tâches d'apprentissage automatique [121].

2.5.2 Architecture et fonctionnement des GNN

À l'instar des réseaux de neurones profonds classiques, l'architecture d'un GNN s'appuie généralement sur plusieurs couches fondamentales, spécifiquement conçues pour traiter efficacement les données relationnelles, comme illustré dans la [figure 2.17](#).

L'un des composants essentiels est la couche de propagation de messages, dans laquelle chaque nœud agrège les informations provenant de ses voisins immédiats [138]. Ce processus implique le calcul de messages neuronaux entre nœuds connectés, en tenant compte de leurs embeddings actuels et des caractéristiques des arêtes qui les relient [139]. Pour chaque nœud, ces messages sont ensuite agrégés à l'aide d'une fonction d'agrégation, telle que l'opération de mean-pooling [140, 141], le mécanisme d'attention [142], etc. Le message agrégé est combiné à l'ancien embedding du nœud via une fonction de mise à jour, souvent non linéaire, afin de produire un nouvel embedding [139]. Ce mécanisme est itératif et peut être appliqué sur plusieurs couches, permettant ainsi la propagation de l'information à travers le graphe et l'apprentissage à partir de voisins de plus en plus éloignés [143]. Le nombre de couches définit la distance à laquelle l'information est propagée ; un nœud dans un modèle GNN à deux couches recevra des informations de ses voisins immédiats et des voisins de ses voisins.

Comme illustré dans [figure 2.18](#), le nœud 1 dans l'image reçoit les embeddings de ses voisins immédiats (ici les nœuds 2 et 3). Ces embeddings sont ensuite agrégés à l'aide d'une fonction d'agrégation afin de synthétiser les informations contextuelles locales. Cette étape est illustrée dans la section *AGGREGATE* de l'image. Une fois l'agrégation effectuée, le nœud met à jour son propre embedding en combinant son embedding actuel avec l'information agrégée via une fonction de mise à jour, comme montré dans la section *UPDATE*. Ce mécanisme est itératif et peut être appliqué sur plusieurs couches pour permettre à chaque nœud de capturer l'information de plus en plus éloignée dans le graphe.

Les GNNs sont particulièrement efficaces pour traiter des données structurées

Chapitre 2. Systèmes de recommandations et GNN

en graphe, en raison de leur capacité à exploiter à la fois les caractéristiques individuelles des nœuds et les informations encodées dans la topologie du graphe [139]. En agrégeant de manière itérative les informations provenant des nœuds voisins, les GNNs sont capables de capturer des relations et des dépendances complexes au sein des données [145]. Ce mécanisme de propagation de messages permet à l'information de circuler à travers le graphe, permettant ainsi à chaque nœud de collecter et d'agréger des informations issues de son contexte local — une propriété cruciale pour des tâches telles que la recommandation, où les relations entre utilisateurs et items sont primordiales [143]. Les GNNs peuvent réaliser différents types d'analyses sur les graphes, notamment des tâches au niveau des nœuds (comme la classification ou la prédiction de propriétés individuelles), au niveau des arêtes (comme la prédiction de l'existence ou des caractéristiques de liens entre les nœuds), et au niveau global du graphe (telles que la classification de graphes entiers). Cette flexibilité les rend particulièrement adaptés à une grande variété de scénarios de recommandation, allant de la prédiction des préférences d'un utilisateur pour un item spécifique (prédiction de liens dans un graphe user-item) à la recommandation de groupes d'items ou d'utilisateurs (tâches globales sur le graphe). De plus, les GNNs sont capables de reconstruire des caractéristiques latentes uniquement à partir de la structure du graphe, soulignant ainsi leur capacité à extraire de manière intrinsèque des informations à partir des relations entre entités [146].

2.5.3 Utilisation des GNNs dans les Systèmes de recommandation

Dans cette section, nous détaillons l'utilisation des GNNs dans le cadre des systèmes de recommandation, en nous appuyant principalement sur l'approche collaborative. L'analyse présentée s'inspire en grande partie de l'article intitulé *Graph Neural Networks in Recommender Systems : A Survey* [121]. Afin de mieux comprendre le fonctionnement et l'implémentation des GNNs dans les systèmes de recommandation, nous introduirons les formulations mathématiques associées. Pour faciliter la lecture et la clarté des notations, un résumé des symboles utilisés est présenté dans le tableau [tableau 2.1](#).

Dans le cadre des RS basés sur GNNs, on modélise généralement les utilisateurs $u \in \mathcal{U}$ et les items $i \in \mathcal{I}$ comme des nœuds dans un graphe biparti, où les interactions entre utilisateurs et items sont représentées par les arêtes, comme illustré dans [figure 2.19](#).

La tâche principale consiste à prédire le niveau de préférence d'un utilisateur u pour un item i , noté $y_{u,i}$. Cette variable peut représenter une valeur binaire (in-

Chapitre 2. Systèmes de recommandations et GNN

Notations	Descriptions
$\mathcal{U}/\mathcal{I}$	L'ensemble des utilisateurs/éléments
A	Matrice d'adjacence du graphe
$\mathcal{N}_v$	Ensemble de voisinage du nœud v
$\mathbf{h}_v^{(l)}$	Embedding du nœuds v à la couche l
$\mathbf{h}_i^{(l)}$	Embedding de l'élément i à la couche l
$\mathbf{n}_v^{(l)}$	Embedding agrégée des voisins du nœud v à la couche l
$\mathbf{h}_u^*$	Embedding finale de l'utilisateur u
$\mathbf{h}_i^*$	Embedding finale de l'élément i
$\mathbf{b}^{(l)}$	Biais à la couche l
$\oplus$	Concaténation de vecteur

TABLE 2.1. Notations principales utilisées dans cet article [121].

teraction ou non), une note explicite (par exemple de 1 à 5), ou une probabilité d'interaction. Dans [figure 2.19](#), le niveau de préférence de l'utilisateur 1 (User1) et l'item 2 (Item2) est $y_{User1,Item2} = 3$.

Pour cela, chaque utilisateur et chaque item sont associés à un embedding, noté respectivement $\mathbf{h}_u^*$ et $\mathbf{h}_i^*$, obtenu après l'agrégation de voisinage sur plusieurs couches du GNN.

La préférence $y_{u,i}$ est alors estimée à l'aide d'une fonction f prenant comme entrée ces deux embeddings :

$$y_{u,i} = f(\mathbf{h}_u^*, \mathbf{h}_i^*) \quad (2.1)$$

La fonction f peut prendre différentes formes selon le modèle utilisé. Elle peut être un produit scalaire entre les deux vecteurs, une distance, ou encore un réseau de neurones (MLP) appliqué à la concaténation des deux représentations [148].

Mathématiquement, un graphe $\mathcal{G}$ peut être formalisé par la paire $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, où $\mathcal{V}$ représente l'ensemble des nœuds et $\mathcal{E}$ l'ensemble des arêtes, voir [tableau 2.1](#). Soit $v_i \in \mathcal{V}$ un nœud et $e_{ij} = (v_i, v_j) \in \mathcal{E}$ une arête allant de v_j vers v_i . Le voisinage d'un nœud v est alors défini comme $\mathcal{N}(v) = \{u \in \mathcal{V} \mid (v, u) \in \mathcal{E}\}$.

Comme on l'a déjà mentionné, le principe fondamental des GNN repose sur un processus itératif d'agrégation des caractéristiques des nœuds voisins, suivi de leur intégration à l'embedding actuel du nœud central au cours de la phase de propagation [149, 150]. D'un point de vue architectural, un GNN est composé de plusieurs couches empilées de propagation, chacune étant constituée de deux opérations principales : l'agrégation des informations issues du voisinage et la mise à jour de l'état du nœud central ou nœud ciblé. Ce mécanisme de propagation peut être formalisé

Chapitre 2. Systèmes de recommandations et GNN

par une équation spécifique, appelée formule de propagation, mettant en œuvre ces deux étapes de manière récurrente à travers les couches du modèle. Voici la formule de propagation :

$$\begin{aligned} \text{Agrégation : } \quad \mathbf{n}_v^{(l)} &= \text{Agrégateur}_l \left(\{ \mathbf{h}_u^{(l)} \mid u \in \mathcal{N}_v \} \right), \\ \text{Mise à jour : } \quad \mathbf{h}_v^{(l+1)} &= \text{MiseÀJour}_l \left(\mathbf{h}_v^{(l)}, \mathbf{n}_v^{(l)} \right), \end{aligned} \quad (2.2)$$

où $\mathbf{h}_u^{(l)}$ désigne l'embedding du nœud u à la l -ième couche, et Agrégateur et MiseÀJour représentent respectivement la fonction d'agrégation et la fonction de mise à jour à la l -ième couche.

Nous résumons ici brièvement les opérations d'agrégation et de mise à jour de cinq architectures typiques de GNN largement utilisées dans le domaine de la recommandation.

- **Graph Convolutional Networks, GCN** [151] approxime la décomposition spectrale de premier ordre du Laplacien du graphe pour agréger itérativement les informations provenant des voisins.
- **GraphSAGE** [140] échantillonne un nombre fixe de voisins pour chaque nœud, propose des fonctions d'agrégation basées sur la moyenne, la somme, le max-pooling, et utilise une concaténation pour la mise à jour :

$$\begin{aligned} \text{Agrégation : } \quad \mathbf{n}_v^{(l)} &= \text{Aggregator}_l \left(\{ \mathbf{h}_u^{(l)}, \forall u \in \mathcal{N}_v \} \right) \\ \text{Mise à jour : } \quad \mathbf{h}_v^{(l+1)} &= \delta \left(\mathbf{W}^{(l)} \cdot [\mathbf{h}_v^{(l)} \oplus \mathbf{n}_v^{(l)}] \right) \end{aligned} \quad (2.3)$$

où $\oplus$ représente la concaténation, $\delta(\cdot)$ est une fonction d'activation, et $\mathbf{W}^{(l)}$ est une matrice de transformation apprenable.

- **Gated Graph Sequence Neural Networks, GGNN** [152] utilise une unité récurrente à barrière (Gated Recurrent Unit, GRU) dans l'étape de mise à jour. Cette approche effectue plusieurs passes du mécanisme récurrent sur l'ensemble des nœuds, ce qui peut poser des problèmes d'échelle pour les grands graphes.
- **Hypergraph Neural Networks, HGNN** [153] est un réseau de neurones de type hypergraphe, qui encode des corrélations d'ordre supérieur dans une structure d'hypergraphe.
- **Graph Attention Networks, GAT** [154] considère que l'influence des voisins n'est ni identique ni prédéfinie par la structure du graphe. Il différencie les contributions des voisins à l'aide d'un mécanisme d'attention, qui attribue un poids d'importance à chaque voisin en fonction de ses caractéristiques :

$$\text{Agrégation : } \mathbf{n}_v^{(l)} = \sum_{j \in \mathcal{N}_v} \alpha_{vj} \mathbf{h}_j^{(l)}, \quad \alpha_{vj} = \frac{\exp(\text{Att}(\mathbf{h}_v^{(l)}, \mathbf{h}_j^{(l)}))}{\sum_{k \in \mathcal{N}_v} \exp(\text{Att}(\mathbf{h}_v^{(l)}, \mathbf{h}_k^{(l)}))} \quad (2.4)$$

$$\text{Mise à jour : } \mathbf{h}_v^{(l+1)} = \delta(\mathbf{W}^{(l)} \mathbf{n}_v^{(l)})$$

où $\text{Att}(\cdot)$ est une fonction d'attention, généralement implémentée sous la forme d'une couche entièrement connectée suivie d'une non-linéarité (par exemple ReLU), α_{vj} représente le coefficient d'attention normalisé mesurant l'importance du nœud j pour la mise à jour de v , et $\mathbf{W}^{(l)}$ est une matrice de poids entraînable.

Le mécanisme d'attention permet au modèle de pondérer différemment chaque voisin en fonction de sa pertinence contextuelle, plutôt que d'appliquer une agrégation uniforme comme dans les modèles classiques tels que GCN. Cela rend le modèle plus expressif, notamment dans les graphes où les relations entre les nœuds ne sont pas toutes équivalentes en importance.

De plus, pour améliorer la stabilité et la capacité d'expression du modèle, GAT utilise des *multi-head attentions* : plusieurs têtes d'attention parallèles sont calculées, chacune apprenant à capturer un aspect différent du voisinage, avant d'être concaténées ou moyennées pour former la nouvelle représentation du nœud :

$$\mathbf{h}_v^{(l+1)} = \left\|_{k=1}^K \delta \left(\sum_{j \in \mathcal{N}_v} \alpha_{vj}^k \mathbf{W}^k \mathbf{h}_j^{(l)} \right) \right. \quad (2.5)$$

où K est le nombre de têtes d'attention, et $\|$ représente l'opération de concaténation.

Avantages de GAT

- Capacité à attribuer dynamiquement des poids d'importance aux voisins, améliorant la qualité des représentations dans les graphes hétérogènes ou bruités.
- Utilisation de l'attention multi-têtes pour capturer diverses relations ou dépendances locales.
- Indépendance vis-à-vis de la structure globale du graphe : seules les caractéristiques locales des voisins sont utilisées pour le calcul des poids d'attention.

2.6 État de l’art des système de recommandation en santé

La littérature scientifique témoigne d’un intérêt croissant pour l’application des systèmes de recommandation dans divers domaines de la santé [155]. Ces HRS (Health Recommender Systems, système de recommandation en santé en français) ont été explorés pour leur potentiel à personnaliser et à améliorer plusieurs aspects des soins aux patients, notamment les recommandations médicamenteuses, l’aide au diagnostic, la planification des traitements et la fourniture de conseils individualisés en matière de mode de vie [156]. Bien que le filtrage collaboratif ait été appliqué dans le domaine médical, il reste relativement moins répandu en tant qu’approche autonome, au profit de méthodes fondées sur le contenu ou sur la connaissance [91]. Néanmoins, la progression vers des approches hybrides, combinant filtrage collaboratif et autres techniques, constitue une réponse prometteuse aux limites inhérentes au filtrage collaboratif pur, telles que la rareté des données ou le problème du démarrage à froid [157]. En parallèle, l’intégration de méthodes d’apprentissage automatique avancées, notamment le filtrage collaboratif fondé sur l’apprentissage profond, suscite un intérêt croissant pour accroître la personnalisation et la précision des recommandations au sein des HRS [156, 107].

Après une analyse approfondie de plusieurs articles scientifiques traitant de l’arthrose et de l’accompagnement par l’exercice physique, il ressort que peu de travaux de recherche ont été consacrés au développement de systèmes de recommandation pour les activités d’exercice physique [158], et qu’aucune étude ne semble aborder spécifiquement la problématique de la recommandation personnalisée d’exercices pour les patients atteints d’arthrose du genou à l’aide de méthodes issues des systèmes de recommandation. Bien que la littérature explore en détail les bienfaits de l’exercice dans le traitement de l’arthrose et examine divers dispositifs numériques de suivi ou d’intervention, aucune approche ne cible directement la personnalisation des exercices basée sur les préférences ou les profils des patients à travers des techniques de filtrage collaboratif.

Bien que des approches de recommandation par filtrage collaboratif aient été testées pour améliorer l’engagement et les performances des patients [159, 157, 156, 160], ces travaux ne se focalisent pas spécifiquement sur l’arthrose du genou. Par ailleurs, les revues consacrées à l’application de l’apprentissage automatique au diagnostic et à la prédiction de l’évolution de l’arthrose du genou se limitent essentiellement à la classification d’images radiographiques et à la modélisation pronostique, sans aborder la recommandation d’exercices thérapeutiques [161, 162]. De même, les

études récentes évaluant l'efficacité d'interventions numériques d'exercices en libre-service (apps mHealth) pour l'arthrose ne mettent pas en œuvre de mécanismes de filtrage collaboratif ou de réseaux de neurones graphiques pour personnaliser les recommandations [163].

En l'absence d'exemple concret ciblant la recommandation d'exercices pour l'arthrose du genou via une approche de filtrage collaboratif, notre projet de master constitue une première recherche se basant sur un modèle de filtrage collaboratif enrichi par des Graph Neural Networks pour proposer des programmes d'exercices personnalisés.

2.7 Conclusion

Ce chapitre a exploré les systèmes de recommandation, leurs approches principales (filtrage basé sur le contenu, collaboratif et hybride) et leur importance croissante en santé pour personnaliser les soins. Il a mis l'accent sur le filtrage collaboratif, ses variantes et les défis liés à la sparsité des données. L'émergence des Graph Neural Networks (GNNs) a été présentée comme une avancée prometteuse pour améliorer la précision des recommandations, bien que peu de travaux portent sur leur application à la recommandation d'exercices thérapeutiques pour la gonarthrose. Ce constat justifie l'approche novatrice développée dans ce projet, détaillée dans le chapitre suivant.

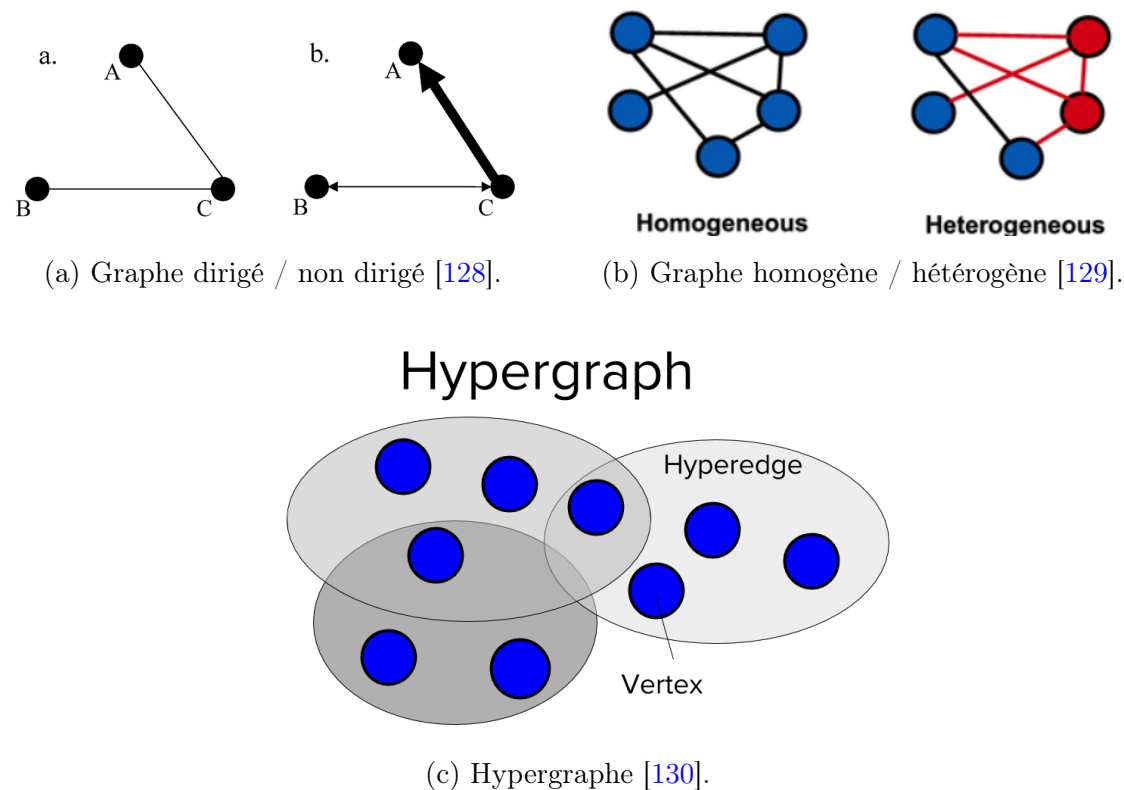


FIGURE 2.14. Représentation graphique des types de graphes.

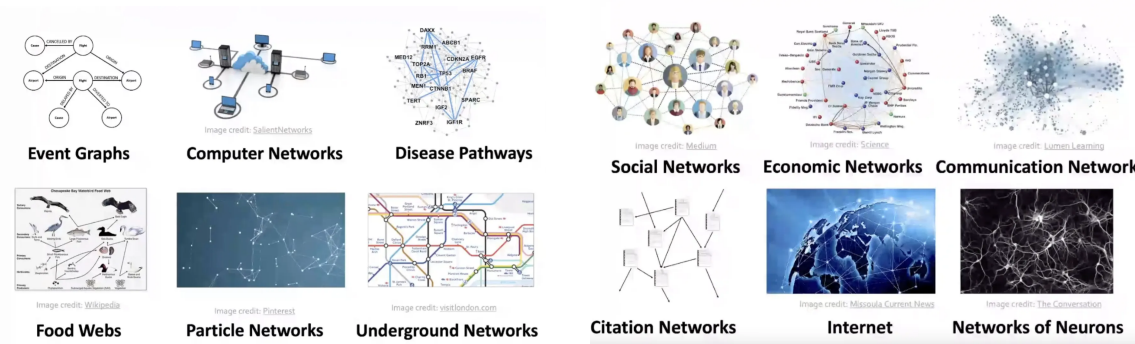


FIGURE 2.15. Exemple d'utilisation des graphes [133].

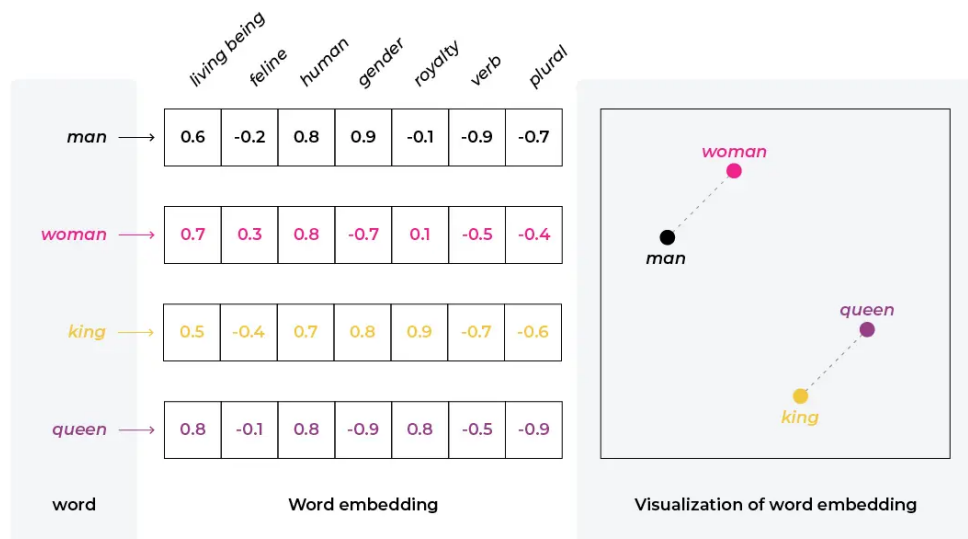


FIGURE 2.16. Exemple d’embeddings de mots illustrant les relations sémantiques entre termes [137].

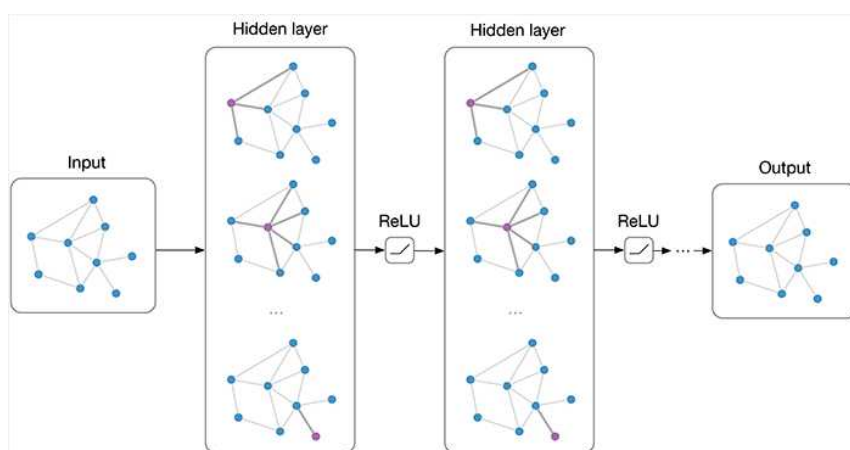


FIGURE 2.17. Illustration de l’architecture d’un GNN avec deux couches [117].

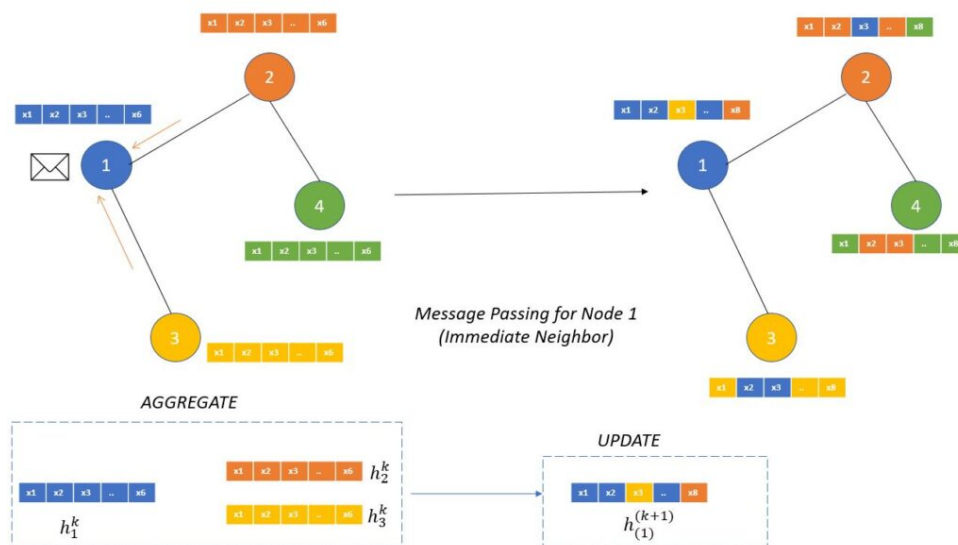


FIGURE 2.18. La processus de propagation de messages dans les GNNs [144].

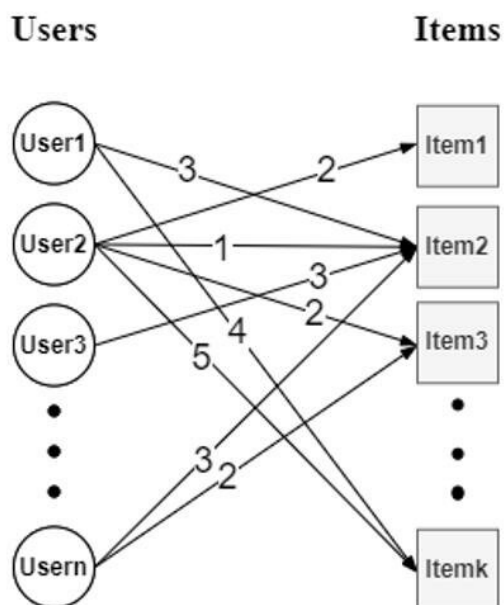


FIGURE 2.19. Un exemple de graphe bipartite Utilisateur-Item avec des notes [147].

Chapitre 3

Conception du système de recommandation

3.1 Introduction

Cette section présente la démarche adoptée pour concevoir un système de recommandation d'exercices thérapeutiques destiné aux patients atteints de gonarthrose. Cette conception repose sur les spécificités des données cinématiques biomécaniques disponibles ainsi que sur l'usage combiné d'une modélisation par graphes et d'une modélisation par factorisation de matrice (MF).

L'objectif principal est de développer une architecture capable de recommander des exercices physiques adaptés aux profils individuels des patients, en tenant compte de leurs caractéristiques cliniques, fonctionnelles et biomécaniques.

Pour modéliser cette problématique, nous avons formulé la tâche comme une prédiction de liens (*link prediction*) entre deux types d'entités : les patients et les exercices. Cela nous conduit à représenter le problème sous la forme d'un graphe biparti, dans lequel les nœuds correspondent aux patients et aux exercices, et les arêtes indiquent la prescription effective d'un exercice à un patient. L'enjeu consiste alors à prédire de nouvelles connexions dans ce graphe, c'est-à-dire à identifier quels exercices pourraient être recommandés à un patient donné.

La solution retenue s'appuie principalement sur les GNN, qui permettent d'exploiter conjointement les structures relationnelles entre les nœuds et les attributs individuels associés à chaque entité. De plus, nous avons modélisé la solution sous forme de MF afin d'évaluer laquelle des deux approches offre les meilleurs résultats. Notre objectif est également de contribuer à l'avancement des recherches scientifiques dans ce domaine.

Ainsi, cette phase de conception établit le cadre conceptuel et technique sur

lequel reposera l'implémentation des modèles présentée dans le chapitre suivant.

3.2 Définition des données disponible

La base de données utilisée dans ce travail regroupe un ensemble d'informations cliniques, fonctionnelles et biomécaniques collectées auprès de patients atteints de gonarthrose. Ces données sont essentielles pour la construction du graphe et la personnalisation des recommandations. La base de données est un fichier *Excel* composé de cinq feuilles, que nous avons séparées en cinq fichiers portant les mêmes noms afin d'en faciliter la manipulation :

- **CinematicData.csv** : Il s'agit de mesures cinématiques 3D du genou enregistrées pendant la phase de marche des patients. Chaque ligne représente un patient (au total 413 individus), et les colonnes correspondent aux variations articulaires du genou selon trois axes : la flexion-extension (*Flexion1* à *Flexion100*), l'abduction-adduction (*Abduction1* à *Abduction100*) et la rotation axiale (*Rotation1* à *Rotation100*). Chaque série de 100 colonnes représente l'évolution de l'angle articulaire tout au long du cycle de marche, découpé en 100 intervalles réguliers. Par exemple, *Flexion1* correspond à l'angle de flexion au tout début du cycle (1%), *Flexion2* au 2%, et ainsi de suite jusqu'à *Flexion100* (100% du cycle de marche).
- **EmoviFeatures.csv** : Il s'agit de données cinématiques ayant été prétraitées et interprétées par l'entreprise *Emovi*. Ces variables correspondent à des résultats cliniques dérivés des signaux bruts, reflétant la présence ou non d'anomalies biomécaniques.
- **70features.csv** : ce fichier regroupe un ensemble de 70 descripteurs numériques extraits à partir des deux séries de données cinématiques brutes. Les données ont été traitées par un organisme externe, et nous utilisons uniquement les résultats fournis. Ce fichier permet de réduire les données cinématiques brutes de dimension 300 (100 pour la flexion-extension, 100 pour l'abduction-adduction et 100 pour la rotation) et de mieux caractériser les propriétés cinématiques des mouvements articulaires.
- **Exercices + orthèses + MSK.csv** : Ce fichier contient les exercices recommandés aux patients (colonnes *ID_exo_1* à *ID_exo_14*), ainsi que certaines caractéristiques pertinentes au moment de la prescription, telles que le port ou non d'une genouillère. Pour la liste complète de ces attributs, se référer à l'annexe [section 1.1](#). Les autres champs de ce fichier regroupent les résultats de tests fonctionnels relatifs à la biomécanique du genou et des membres

inférieurs.

- **Liste détails 36 exercices.csv** : Ce fichier contient les exercices à recommander. Chaque ligne correspond à une catégorie d'exercices, identifiée par un identifiant unique. La colonne `NewID` représente l'identifiant principal de chaque catégorie d'exercices. Pour chaque catégorie, les exercices associés sont listés dans les colonnes `EX0_1` à `EX0_5`, qui correspondent aux identifiants spécifiques des exercices individuels appartenant à cette catégorie. Une brève description de chaque catégorie est également incluse afin de mieux caractériser les types d'exercices regroupés.

L'un des défis majeurs rencontrés dans ce travail réside dans la disponibilité partielle des données. En effet, la base de données contient un nombre significatif de valeurs manquantes. Cette situation s'explique en grande partie par la difficulté inhérente à la collecte de données biologiques et fonctionnelles, qui nécessitent des équipements spécialisés, des protocoles rigoureux et une implication active des patients. De plus, certaines mesures ne sont pas systématiquement effectuées pour tous les sujets, ce qui limite la richesse globale de la base et la représentativité des profils. Cette rareté des données impose des choix méthodologiques prudents en matière de traitement des valeurs manquantes, d'exclusion de cas ou de sélection de variables pour l'analyse.

3.3 Prétraitement et structuration des données graphiques

Le bon fonctionnement d'un GNN repose fortement sur la qualité et la structuration des données en entrée. Dans le cadre de ce travail, les données doivent être transformées en représentations numériques exploitables par le modèle.

Cette section présente les étapes nécessaires pour encoder les caractéristiques des nœuds (patients et exercices), gérer les valeurs manquantes, transformer les variables catégorielles, et structurer les relations entre entités sous forme de graphe. Ces opérations constituent la phase de prétraitement, essentielle pour assurer la compatibilité avec l'architecture du GNN et garantir la qualité de l'apprentissage.

Pour le prétraitement des données, voici les étapes :

1. Gestion des valeurs manquantes

- Pour les colonnes numériques, les valeurs manquantes sont remplacées par la moyenne de la colonne.

Chapitre 3. Conception du système de recommandation

- Pour les colonnes catégorielles (texte), elles sont remplacées par la valeur la plus fréquente (le mode).
2. **Gestion des outliers**¹ – Plutôt que de supprimer les valeurs aberrantes, celles-ci sont plafonnées : les observations inférieures à la borne inférieure sont ramenées à cette valeur, et celles supérieures à la borne supérieure sont réduites à celle-ci. Ici, les bornes sont définies à partir de l'écart interquartile (IQR) :

$$\begin{aligned} \text{IQR} &= Q_3 - Q_1 \\ \text{borne inférieure} &= Q_1 - 1,5 \cdot \text{IQR} \\ \text{borne supérieure} &= Q_3 + 1,5 \cdot \text{IQR}. \end{aligned} \tag{3.1}$$

3. **Suppression des doublons** – Les doublons sont supprimés garantissant que chaque instance du jeu de données n'apparaît qu'une seule fois.
4. **Standardisation des caractéristiques numériques** – Les caractéristiques numériques subissent un processus de standardisation pour les mettre à une échelle commune. Cette transformation ajuste les données pour qu'elles aient une moyenne de zéro et un écart-type de un. Pour chaque valeur x dans une caractéristique, la valeur standardisée z est calculée comme suit :

$$z = \frac{x - \mu}{\sigma} \tag{3.2}$$

où μ est la moyenne de la caractéristique et σ son écart-type. Cette étape est cruciale pour les algorithmes sensibles à l'échelle des caractéristiques d'entrée.

5. **Encodage des caractéristiques catégorielles** – Les variables catégorielles (comme le genou concerné ou le type d'orthèse) ont été transformées par un encodage *one-hot*².

L'[algorithme 1](#) résume le pipeline de prétraitement des données.

1. Les outliers, ou valeurs aberrantes, sont des observations qui s'écartent significativement de la majorité des données d'un ensemble. Ils peuvent résulter d'erreurs de mesure, de variations naturelles extrêmes ou d'événements rares. Leur détection est essentielle, car ils peuvent fausser les analyses statistiques et les modèles prédictifs si ils ne sont pas traités correctement.

2. L'encodage *one-hot* est une technique de transformation des variables catégorielles en vecteurs binaires. Chaque catégorie est représentée par un vecteur dont une seule composante est égale à 1 (indiquant la présence de la catégorie), tandis que les autres sont égales à 0. Par exemple, une variable *type d'orthèse* avec trois modalités différentes serait encodée en trois colonnes binaires.

Chapitre 3. Conception du système de recommandation

Algorithm 1: Prétraitement des données

Input: Dataset D avec variables numériques et catégorielles

Output: Dataset D prétraité

```
1 foreach colonne numérique  $col\_num$  dans  $D$  do
2   | Remplacer les valeurs manquantes par la moyenne;
3 foreach colonne catégorielle  $col\_cat$  dans  $D$  do
4   | Remplacer les valeurs manquantes par le mode;
5 foreach colonne numérique  $col\_num$  dans  $D$  do
6   | Calculer  $Q_1, Q_3, IQR$ ; Définir  $borne\_inf, borne\_sup$ ; foreach valeur
7   |   |  $x$  dans  $col\_num$  do
8   |   |   if  $x < borne\_inf$  then
9   |   |   |  $x \leftarrow borne\_inf$ ;
10  |   |   else
11  |   |   | if  $x > borne\_sup$  then
12  |   |   |   |  $x \leftarrow borne\_sup$ ;
13 foreach colonne numérique  $col\_num$  dans  $D$  do
14   | Calculer  $\mu, \sigma$ ; foreach valeur  $x$  dans  $col\_num$  do
15   |   |  $x \leftarrow \frac{x - \mu}{\sigma}$ ;
16 foreach colonne catégorielle  $col\_cat$  dans  $D$  do
17   | Appliquer l'encodage one-hot;
18 return  $D$  prétraité;
```

3.4 Conception du modèle GNN

3.4.1 Description des entités manipulées

Dans le cadre de la modélisation du système de recommandation par GNN, deux types principaux de nœuds sont définis : les nœuds *patient* et les nœuds *exercice*. Les caractéristiques des nœuds patient sont extraites à partir de l'ensemble des fichiers de la base de données, où chaque identifiant unique ($ID_patient$) correspond à un nœud distinct. Quant aux caractéristiques des nœuds exercice, elles sont issues du fichier `Liste détails 36 exercices`, qui recense un ensemble de 36 catégories d'exercices, identifiées par la variable `New_ID`. Chaque catégorie recense un ensemble d'exercices.

Chapitre 3. Conception du système de recommandation

Ce schéma permet la construction d'un graphe biparti, dans lequel les arêtes relient les patients aux exercices qui leur ont été prescrits. Autrement dit, une arête est ajoutée entre un patient et un exercice si ce dernier lui a été recommandé ; en l'absence de prescription, aucune arête n'est créée. Une illustration de cette structure est présentée dans la [figure 3.1](#).

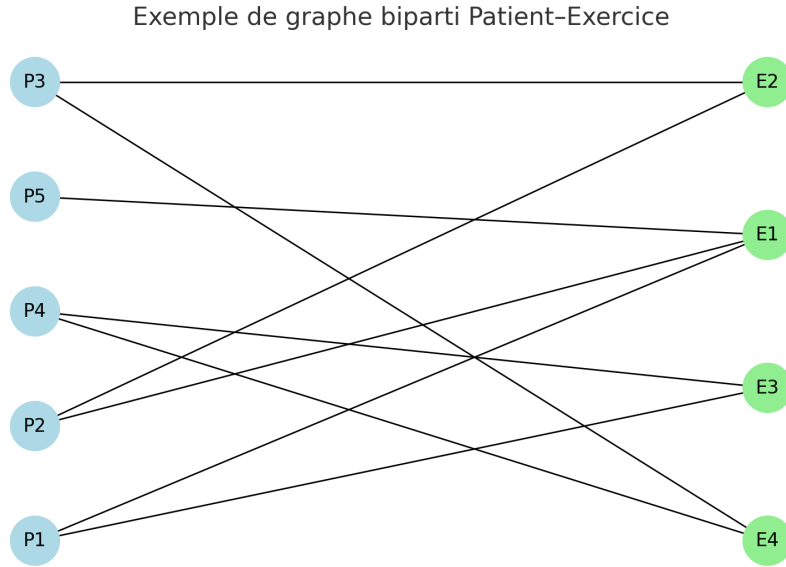


FIGURE 3.1. Visualisation d'un graphe biparti.

3.4.2 Représentation des features

Dans un GNN, chaque nœud est initialement représenté par un vecteur de *features*, ou caractéristiques, décrivant ses propriétés. Ces vecteurs servent de point de départ pour la génération des embeddings appris au fil des couches du GNN. L'image [figure 3.2](#) illustre cette notion fondamentale.

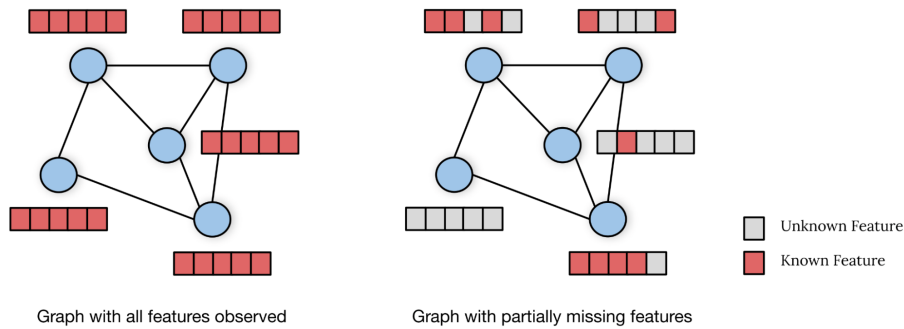


FIGURE 3.2. Illustration de l'impact des features initiales dans un GNN [164].

Chapitre 3. Conception du système de recommandation

Lorsque toutes les caractéristiques sont disponibles (partie gauche de [figure 3.2](#)), le GNN peut directement exploiter les propriétés de chaque nœud pour propager l'information à travers le graphe. Cependant, dans de nombreux cas réels, certaines features peuvent être manquantes (partie droite de [figure 3.2](#)), rendant l'apprentissage plus complexe. Dans ces cas, le GNN peut combler les lacunes en exploitant les connexions entre nœuds et en apprenant à partir des voisins, mettant en évidence l'importance de la structure du graphe pour enrichir ou reconstruire les embeddings. Ainsi, le choix des features initiales influence fortement les performances finales du modèle.

Features des nœuds « Patient »

Les nœuds représentant les patients dans le graphe sont décrits à l'aide des caractéristiques extraites du fichier *70Features.csv*. Ces caractéristiques résultent des modifications apportées aux données cinématiques brutes. Un fichier *user_features.csv* a été créé et contient les attributs des nœuds patients. Ce fichier possède une dimension de 41371, correspondant à 413 patients avec, pour chacun, un identifiant et 70 caractéristiques.

Features des nœuds « Exercice »

Les nœuds représentant les exercices sont décrits à partir des informations extraites de la colonne « Description » du fichier *Liste détails 36 exercices.csv*. L'approche repose sur l'identification de mots-clés pour générer des caractéristiques binaires. Par exemple, la détection du terme « contrôle moteur » crée la variable *Contrôle moteur* (valeur 1 si présent, 0 sinon), et la présence de « renforcement des fléchisseurs de la hanche » crée la variable *Renforcement des fléchisseurs de la hanche*. Chaque mot-clé correspond ainsi à une feature binaire.

Un fichier *item_features.csv* a été créé, avec 71 lignes (exercices) et 38 colonnes : l'identifiant de l'exercice et 37 caractéristiques extraites.

3.4.3 Définition des arêtes du graphe

Les arêtes du graphe représentent la relation de prescription entre un patient et un exercice. Plus précisément, un lien est créé lorsqu'un exercice est prescrit à un patient. Ces relations ont été extraites des colonnes *ID_exo_1* à *ID_exo_14* du fichier *Exercices + orthèses + MSK.csv*, comme illustré dans [figure 3.1](#).

Le graphe intègre deux types de relations bidirectionnelles :

0. **Relation "recommande"** : patient $\xrightarrow{\text{recommande}}$ exercice.

Chapitre 3. Conception du système de recommandation

0. **Relation "recommandé à"** : exercice $\xrightarrow{\text{recommandé à}}$ patient.

Cette structure bidirectionnelle permet la propagation d'information dans les deux sens, enrichissant la représentation des embeddings par l'agrégation de voisinages.

3.4.4 Architecture du modèle GNN

L'architecture GNN proposée suit une approche hiérarchique multi-couches, combinant les avantages des convolutions graphiques et des mécanismes d'attention pour capturer les motifs complexes d'interaction patient-exercice.

Avant la propagation, les caractéristiques d'entrée (features des utilisateurs et des items) sont projetées dans un espace latent de 256 dimensions à l'aide de couches linéaires. Après trois couches GAT successives, les embeddings finaux sont réduits à 128 dimensions.

Des connexions résiduelles stabilisent l'entraînement et, après chaque couche (sauf la dernière), on applique successivement :

1. la normalisation par lot ³,
2. la fonction d'activation ReLU pour introduire de la non-linéarité,
3. le dropout ⁴ configuré dynamiquement sur les features et les arêtes. Cette technique régularise le modèle et contribue à réduire le surapprentissage ⁵.

3.4.5 Stratégie de décodage des scores

Pour estimer la pertinence d'une interaction potentielle entre un patient u et un exercice i , notée $y_{u,i}$, le modèle combine trois approches complémentaires : un réseau de neurones multi-couches (MLP), un produit scalaire et une similarité cosinus. Cette

3. La *Batch Normalization* [165] vise à accélérer la convergence et de stabiliser l'entraînement en normalisant, pour chaque mini-batch, les activations de chaque neurone :

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \quad y_i = \gamma \hat{x}_i + \beta,$$

où γ et β sont des paramètres appris, et ε un terme de régularisation numérique.

4. Le *Dropout* consiste à désactiver aléatoirement, à chaque itération d'entraînement, une fraction p des neurones d'une couche. Pour chaque neurone, on échantillonne une variable de Bernoulli $d_i \sim \text{Bernoulli}(1 - p)$ et on remplace son activation x_i par $d_i x_i$. En phase d'inférence, aucune suppression n'est effectuée, mais les activations sont multipliées par $1 - p$ pour compenser l'effet du dropout.

5. Le surapprentissage (*overfitting*) est un phénomène où un modèle s'adapte trop fortement aux données d'entraînement, capturant même le bruit ou les particularités spécifiques, ce qui nuit à sa capacité à généraliser sur de nouvelles données.

Chapitre 3. Conception du système de recommandation

combinaison permet de capturer à la fois des relations non linéaires complexes (via le MLP) et des mesures directes de similarité vectorielle (via le produit scalaire et la similarité cosinus). Le score final est obtenu par une moyenne pondérée de ces trois méthodes, les poids ayant été ajustés empiriquement afin d’optimiser la performance globale du système sur les données de validation.

Le modèle retourne une liste d’exercices correspondant aux meilleures recommandations pour un patient donné, en fonction de ses données cinématiques. Le nombre d’exercices à recommander est spécifié par un paramètre : par exemple, si ce paramètre est fixé à 10, le modèle renverra les dix exercices jugés les plus pertinents.

Ensuite, une fonction sigmoïde, illustrée dans la [figure 3.3](#), est appliquée à la sortie finale du modèle :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$

Cette fonction permet de transformer le score brut (appartenant potentiellement à l’ensemble des réels $\mathbb{R}$) en une probabilité comprise entre 0 et 1.

Cette transformation est essentielle car la tâche de recommandation est formulée comme un problème de classification binaire : prédire s’il existe ou non une interaction pertinente entre un utilisateur et un exercice donné. Un seuil est ensuite appliqué afin de décider si la recommandation doit être effectuée ou non :

$$\hat{r}_{u,i} = \begin{cases} 1 & \text{si } y_{u,i} \geq \tau \\ 0 & \text{sinon} \end{cases} \quad (3.4)$$

3.4.6 Fonction de perte

La fonction de perte utilisée dans ce travail est la *Focal Loss*, définie par l’équation suivante :

$$\mathcal{L}_{\text{focal}} = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (3.5)$$

où α et γ sont des hyperparamètres de pondération permettant de moduler l’impact des exemples sur la perte totale. Le terme p_t représente la probabilité prédite pour la classe correcte et est défini par :

$$p_t = \begin{cases} p & \text{si la classe vraie est 1,} \\ 1 - p & \text{si la classe vraie est 0,} \end{cases} \quad (3.6)$$

avec p la probabilité estimée par le modèle pour la classe positive. Ainsi, p_t représente toujours la probabilité assignée par le modèle à la classe cible correcte,

Chapitre 3. Conception du système de recommandation

indépendamment de sa valeur réelle (0 ou 1).

La *Focal Loss* a été choisie pour atténuer l’impact du déséquilibre entre les classes, fréquent dans les systèmes de recommandation où certaines interactions user-item sont rares. Le terme $(1 - p_t)^\gamma$ réduit l’influence des exemples faciles à classer, focalisant l’apprentissage sur les cas difficiles. Ce mécanisme renforce la capacité du modèle à traiter les classes minoritaires, améliorant ainsi la qualité et la pertinence des recommandations produites.

3.5 Conception du modèle Matrix Factorization

Le jeu de données utilisé pour ce système de recommandation d’exercices thérapeutiques comprend les mêmes 413 patients uniques et 71 exercices différents que ceux utilisés dans le modèle GNN, générant un total de 9 376 interactions user-item. Les exercices recommandés à chaque patient ont été extraits du fichier `Liste détails 36 exercices.csv`, permettant de construire une matrice binaire utilisateur-exercice : une valeur de 1 indique qu’un exercice a été recommandé à un patient donné, tandis qu’une valeur de 0 signifie l’absence de recommandation.

Un extrait illustratif de cette matrice est présenté dans la [tableau 3.1](#) ci-dessous :

Dans cet exemple, l’exercice 1 a été recommandé aux patients 1 et 3, tandis que l’exercice 2 n’a été prescrit qu’au patient 2. Cette représentation binaire constitue l’entrée principale pour l’entraînement du modèle de factorisation matricielle. Pour l’évaluation, les données ont été divisées selon une partition stratifiée 80/20, permettant d’obtenir une mesure fiable de la performance de généralisation du modèle sur des données non vues.

3.5.1 Principe de la Matrix Factorization

La Matrix Factorization (ou matrice de factorisation) consiste à décomposer la matrice binaire d’interactions user-item $R \in \{0, 1\}^{nm}$ en deux matrices réelles à plus faible rang $W \in \mathbb{R}^{nk}$ et $H \in \mathbb{R}^{km}$ où $k \ll \min(n, m)$ est le nombre de facteurs latents.

Chaque utilisateur u est ainsi représenté par un vecteur $w_u \in \mathbb{R}^k$ (ligne de W), et chaque exercice i par un vecteur $h_i \in \mathbb{R}^k$ (colonne de H).

La prédiction de la probabilité qu’un utilisateur u se voie recommander l’exercice i s’effectue via un produit scalaire biaisé :

$$y_{u,i} = \sigma(w_u^\top h_i + b_u + b_i + b_0),$$

où b_u (resp. b_i) est le biais associé à l’utilisateur u (resp. à l’exercice i), b_0 un biais

Chapitre 3. Conception du système de recommandation

global, et $\sigma(x) = \frac{1}{1+e^{-x}}$ la fonction sigmoïde. La recommandation se fait selon un seuil défini.

3.5.2 Fonction de perte

Pour apprendre W , H et les biais, on minimise la *binary cross-entropy* entre les prédictions $\hat{r}_{u,i}$ et les valeurs observées $r_{u,i} \in \{0, 1\}$. En notant Ω l'ensemble des paires (u, i) utilisées (échantillonnage équilibré de positifs et négatifs), la perte s'écrit :

$$\mathcal{L} = -\frac{1}{|\Omega|} \sum_{(u,i) \in \Omega} \left[r_{u,i} \log \hat{r}_{u,i} + (1 - r_{u,i}) \log(1 - \hat{r}_{u,i}) \right]. \quad (3.7)$$

Cette formulation pénalise de façon symétrique les erreurs sur les exemples positifs et négatifs, et s'adapte naturellement à la tâche de classification binaire recommandation / non-recommandation.

En résumé, la MF apprend des *embeddings* utilisateurs et exercices de dimension k qui, combinés au biais, offrent une estimation de la probabilité de chaque interaction, optimisée par la minimisation de la binary cross-entropy.

3.6 Problème du Cold Start

Le *problème du cold start* désigne la difficulté rencontrée par les systèmes de recommandation lorsqu'ils doivent traiter de nouveaux utilisateurs ou de nouveaux items pour lesquels aucune donnée d'interaction préalable n'est disponible. Dans le cadre des approches collaboratives, notamment les modèles basés sur les graphes, cette absence d'historique empêche le calcul direct d'embeddings pertinents, ce qui limite la capacité du modèle à générer des recommandations adaptées.

Pour pallier cette limitation, un modèle de réseau de neurones profond, qu'on a nommé l'*Embedding Encoder* a été conçu afin de projeter les caractéristiques des utilisateurs dans l'espace des embeddings appris par le modèle GNN ou MF. Ce modèle est entraîné à reproduire les embeddings générés par le GNN ou MF à partir des données des patients déjà présentes dans le graphe, assurant ainsi la cohérence des embeddings. Lorsqu'un nouvel utilisateur est rencontré, ses caractéristiques (vecteur de dimension 70) sont transformées en un embedding compatible avec l'espace latent du GNN ou MF. Le système de recommandation peut alors calculer les scores de similarité entre cet embedding projeté et les embeddings des items déjà appris par le GNN ou MF, permettant de générer une liste des exercices les plus adaptés pour ce nouvel utilisateur.

Chapitre 3. Conception du système de recommandation

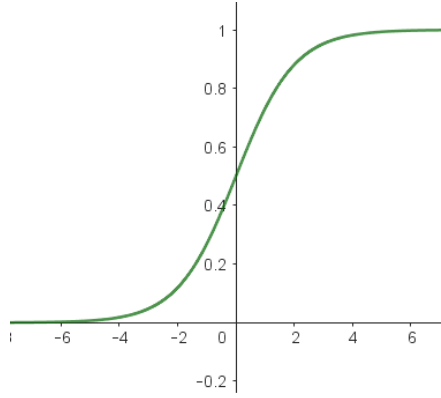


FIGURE 3.3. La fonction sigmoïde [166].

user-item	Item 1	Item 2	Item 3	Item 4
User 1	1	0	1	0
User 2	0	1	1	0
User 3	1	0	0	1

TABLE 3.1. Exemple de sous-matrice user-item binaire extraite des données

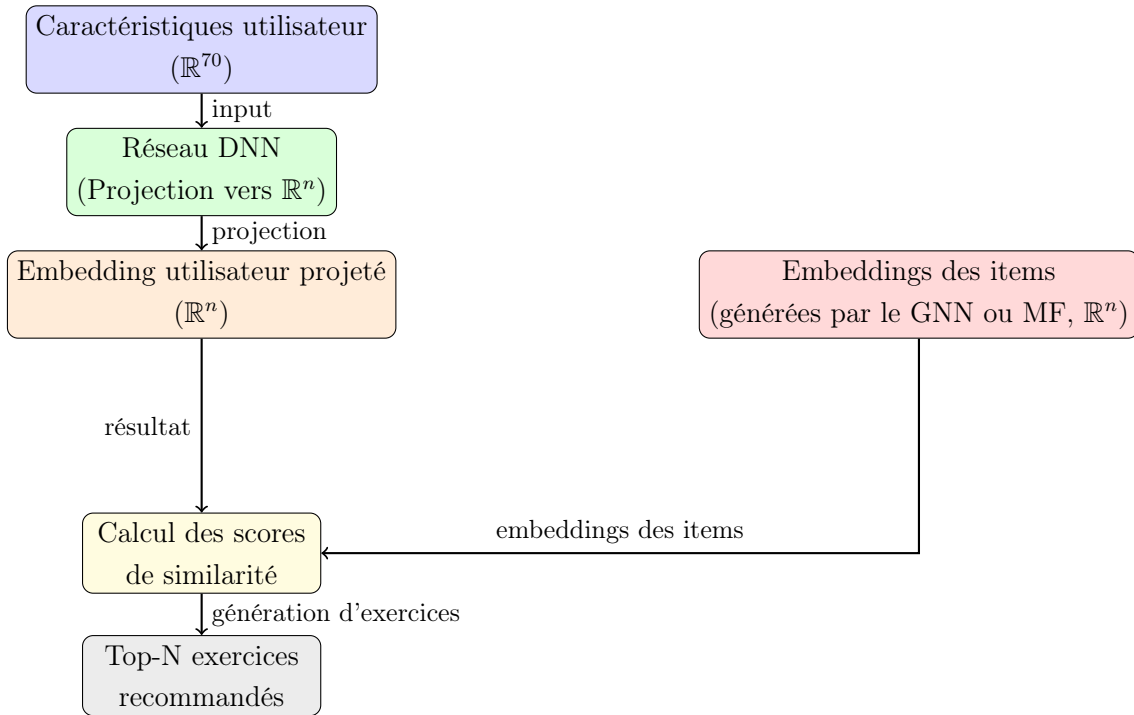


FIGURE 3.4. Schéma du traitement du problème de Cold Start.

Chapitre 3. Conception du système de recommandation

Le schéma illustre la stratégie adoptée pour résoudre le problème de cold start. Les caractéristiques des nouveaux utilisateurs sont projetées dans l'espace des embeddings appris par le GNN ou MF à l'Embedding Encoder. L'embedding obtenu est ensuite comparé aux embeddings des items existants pour générer les recommandations adaptées.

3.7 Comparaison des approches GNN et MF pour la recommandation

Afin de mieux comprendre les spécificités, avantages et limitations respectives des deux approches étudiées dans ce travail, à savoir les réseaux de neurones graphes (GNN) et la factorisation de matrice (MF), la [tableau 3.2](#) présente une synthèse comparative selon plusieurs critères clés influant sur la qualité, l'interprétabilité et la complexité des recommandations générées.

3.8 Conclusion

Ce chapitre a présenté l'ensemble des étapes de conception du système de recommandation d'exercices pour les patients atteints de gonarthrose, en s'appuyant sur deux approches : la modélisation par graphes via les réseaux de neurones à graphes (GNN) et la Matrix Factorization (MF).

Le GNN se distingue par sa capacité à exploiter à la fois les attributs individuels des nœuds et les relations entre eux, fournissant ainsi des représentations latentes riches et contextualisées. À l'inverse, la MF permet une modélisation efficace des interactions directes sous forme de facteurs latents, avec une complexité moindre mais au prix d'une moindre exploitation des caractéristiques intrinsèques des entités.

Afin de surmonter le problème critique du cold start, un modèle d'Embedding Encoder a été intégré pour projeter les caractéristiques des nouveaux patients dans l'espace latent appris par chaque modèle. Cette démarche garantit la généralisabilité du système, même pour des profils jamais observés durant l'entraînement.

Le chapitre suivant détaillera l'implémentation pratique de ces modèles, les choix techniques réalisés ainsi que l'évaluation comparative des performances obtenues sur le jeu de données clinique.

Critère	GNN	MF
Exploite les features	Oui	Non
Nécessite un graphe	Oui	Non
Sensible au <i>cold start</i>	Oui	Oui
Complexité computationnelle	Plus élevée	Moindre
Qualité des embeddings appris	Plus élevée	Moyen

TABLE 3.2. Comparaison des approches GNN et Matrix Factorization.

Chapitre 4

Implémentation et tests

4.1 Introduction

Ce chapitre présente la mise en œuvre pratique et l'évaluation expérimentale de notre système de recommandation d'exercices de rééducation basé sur les GNN et sur la MF. L'objectif principal consiste à développer, optimiser et valider un modèle capable de générer des recommandations personnalisées d'exercices thérapeutiques en exploitant les relations complexes entre les profils pathologiques des patients et les propriétés biomécaniques des exercices.

La phase d'implémentation s'articule autour de plusieurs axes fondamentaux ; l'environnement de développement, les mécanismes de décodage et de prédiction, les stratégies d'entraînement adoptées, incluant l'optimisation des hyperparamètres et les techniques de régularisation mises en œuvre pour maximiser les performances tout en préservant la capacité de généralisation. L'évaluation expérimentale repose sur une méthodologie d'exploration systématique de l'espace des hyperparamètres, permettant d'identifier les configurations optimales pour les architectures GNN et MF. Cette analyse comparative s'appuie sur le *Recommender Performance Score* (RPS), une métrique composite spécialement conçue pour évaluer la qualité globale des systèmes de recommandation en contexte médical.

4.2 Environnement de développement

Ce projet a nécessité un environnement de développement adapté à la complexité des graphes hétérogènes et à l'entraînement intensif des modèles de réseaux de neurones graphiques (GNN). Deux postes de travail distincts ont été utilisés pour le développement, l'entraînement et l'évaluation du modèle.

Chapitre 4. Implémentation et tests

4.2.1 Matériels utilisés

Nous avons utilisé principalement deux postes de travail pour effectuer les calculs et l’entraînement des modèles. La [tableau 4.1](#) présente les caractéristiques techniques de ces postes utilisés.

4.2.2 Logiciels et bibliothèques

Le développement a été réalisé en **Python 3.13.3**, avec l’appui des bibliothèques suivantes :

- **PyTorch** : pour la création des couches GNN et l’entraînement du modèle ;
- **PyTorch Geometric** : manipulation efficace des graphes hétérogènes ;
- **Scikit-learn** : prétraitement des données, calcul des métriques ;
- **Pickle** : sérialisation et sauvegarde des objets Python (scaler, embeddings, etc.) ;
- **Pandas** et **NumPy** : gestion des jeux de données tabulaires ;

4.3 Décodage et prédiction des scores du GNN

Le score de recommandation $y_{u,i}$ pour une paire (u, i) est calculé en combinant trois approches distinctes :

- **MLP** : un réseau à trois couches qui transforme la concaténation des embeddings de l’utilisateur et de l’exercice $[e_u \parallel e_i] \in \mathbb{R}^{2d}$ (avec $d = 128$) en un score unique S_{MLP} . Cette concaténation permet d’exploiter conjointement les caractéristiques des deux nœuds via une transformation non linéaire apprise par le réseau.
- **Produit scalaire** : $S_{\text{dot}} = e_u^\top e_i$, qui mesure la similarité directe des vecteurs dans l’espace latent. Un score élevé indique une forte affinité.
- **Similarité cosinus** : $S_{\text{cos}} = \frac{e_u}{\|e_u\|_2} \cdot \frac{e_i}{\|e_i\|_2}$, qui évalue l’alignement directionnel des vecteurs, indépendamment de leur norme.

Le modèle MLP suit l’architecture suivante :

0. **Première couche linéaire** ($2d \rightarrow d$) avec activation ReLU et Dropout :

$$h_1 = \text{ReLU}(W_1 x + b_1), \quad h'_1 = \text{Dropout}(h_1) \quad (4.1)$$

0. **Deuxième couche linéaire** ($d \rightarrow d/2$) avec ReLU :

$$h_2 = \text{ReLU}(W_2 h'_1 + b_2) \quad (4.2)$$

Chapitre 4. Implémentation et tests

0. **Troisième couche linéaire** ($d/2 \rightarrow 1$) produisant le score final :

$$S_{\text{MLP}} = W_3 h_2 + b_3 \quad (4.3)$$

Le score total de recommandation est une moyenne pondérée des trois composantes suivantes :

$$y_{u,i} = 0.5 S_{\text{MLP}} + 0.3 S_{\text{dot}} + 0.2 S_{\text{cos}} \quad (4.4)$$

Ces poids ont été déterminés de manière empirique afin de maximiser la performance sur le jeu de validation.

Le choix d’accorder un poids principal de 0.5 au score produit par le MLP repose sur sa capacité à capturer des relations non linéaires complexes entre les caractéristiques des patients et des exercices, offrant ainsi une meilleure expressivité et une plus grande capacité d’adaptation aux spécificités des données. Cette composante est considérée comme la plus déterminante dans la prédiction finale.

Le produit scalaire reçoit un poids intermédiaire de 0.3 car il constitue une méthode simple, rapide et efficace pour évaluer la similarité globale entre les représentations latentes des nœuds. Il agit comme un estimateur robuste de la compatibilité entre les entités, notamment lorsque les embeddings contiennent déjà des informations pertinentes extraites par le modèle GNN.

Enfin, la similarité cosinus est pondérée à 0.2, car elle fournit une mesure normalisée de l’alignement directionnel entre les vecteurs d’embedding, ce qui permet de compléter les deux autres scores en capturant des informations géométriques supplémentaires, notamment lorsque les amplitudes des vecteurs peuvent varier sans affecter la direction de similarité.

4.4 Entraînement des modèles

4.4.1 Entraînement du modèle GNN

L’entraînement du modèle GNN repose sur un algorithme d’optimisation basé sur l’algorithme de la descente de gradient, illustré dans [figure 4.1](#).

L’entraînement utilise l’optimiseur *Adam*, accompagné d’un scheduler adaptatif pour réduire le taux d’apprentissage en cas de stagnation de la métrique de validation. Les arêtes du graphe sont séparées aléatoirement en ensembles d’entraînement et de test via une fonction personnalisée.

Pour contrer le déséquilibre des classes, un *negative sampling*¹ est employé avec un ratio fixe de 2, produisant deux négatifs pour chaque positif.

1. Pour chaque interaction positive, on génère r exemples négatifs en échantillonnant aléatoirement des paires utilisateur–item absentes des interactions réelles.

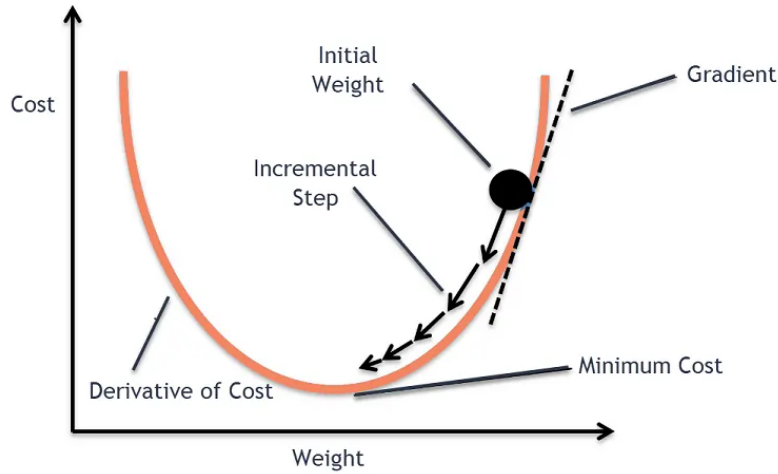


FIGURE 4.1. L'algorithme de la descente de gradient [167].

La fonction de perte utilisée est la *Focal Loss*. Cette formulation permet de moduler l'importance des exemples faciles et difficiles, en réduisant l'influence des exemples bien classés (p_t élevé) et en mettant l'accent sur ceux qui sont mal prédits (p_t faible). Un *gradient clipping*² est appliqué pour prévenir l'explosion des gradients.

4.4.2 Entraînement du modèle Embedding Encoder

Pour rappel, le modèle *Embedding Encoder* a pour objectif de projeter les données cliniques des nouveaux patients dans l'espace d'embedding appris par le modèle GNN ou MF. Étant donné que les espaces latents générés respectivement par le GNN et la MF diffèrent, deux modèles distincts d'*Embedding Encoder* sont entraînés séparément.

Espace GNN

Pour l'*Embedding Encoder* associé à l'espace GNN, l'évolution de la fonction de perte (*loss*) au cours de l'entraînement est illustrée dans la [figure 4.2](#).

L'erreur quadratique moyenne finale (RMSE) obtenue sur l'ensemble de validation est de 0,0314, témoignant d'une excellente capacité du modèle à approximer fidèlement les embeddings des patients. Cette performance garantit des recommandations pertinentes et fiables, même en situation de cold start, c'est-à-dire pour des patients n'ayant jamais interagi avec le système.

2. Le *gradient clipping* est une technique utilisée lors de l'entraînement des réseaux de neurones pour limiter la norme des gradients calculés pendant la rétropropagation. Lorsque les gradients dépassent une valeur seuil spécifiée, ils sont réduits (*clipped*) afin d'éviter les problèmes d'explosion de gradient, stabilisant ainsi l'apprentissage, notamment dans les modèles profonds ou récurrents.

Espace MF

De manière similaire, un second modèle *Embedding Encoder* est entraîné pour projeter les nouveaux patients dans l'espace latent issu de la MF. L'évolution de la fonction de perte associée à cet entraînement est présentée dans la [figure 4.3](#).

Dans ce cas, l'RMSE obtenue sur l'ensemble de validation est de 0,1755, une valeur légèrement plus élevée que celle observée pour l'espace GNN. Cette différence s'explique par la dimension plus importante de l'espace latent généré par la MF, rendant l'approximation des embeddings plus complexe.

Néanmoins, la qualité de cette approximation demeure suffisante pour assurer la cohérence des recommandations produites pour de nouveaux patients, en adéquation avec les représentations latentes apprises lors de l'entraînement du modèle MF.

Ainsi, l'utilisation conjointe de deux modèles *Embedding Encoder* distincts permet une adaptation spécifique à la structure de chaque espace latent (GNN ou MF), garantissant des recommandations robustes et cliniquement pertinentes même en situation de *cold start*.

4.5 Évaluation de la performance des modèles GNN et MF

La performance du modèle est mesurée par le *Recommender Performance Score (RPS)*, une métrique composite définie comme une combinaison pondérée de trois mesures essentielles d'évaluation :

- **AUC (Area Under the Curve)** (poids 0.5) : mesure la capacité du modèle à distinguer entre les classes positives (liens existants) et négatives (liens inexistantes), comme illustré dans [figure 4.4](#). Une valeur de 1.0 indique une séparation parfaite, tandis qu'une valeur de 0.5 correspond à un modèle aléatoire.
- **AP (Average Precision)** (poids 0.3) : reflète la précision moyenne des prédictions du modèle sur l'ensemble des seuils possibles, ce qui permet d'évaluer la capacité du modèle à bien classer les exemples positifs en tête de liste des recommandations.
- **Accuracy** (poids 0.2) : indique le taux global de bonnes classifications, c'est-à-dire la proportion d'exemples correctement prédits, toutes classes confondues.

Les valeurs entre parenthèses correspondent aux poids attribués à chaque métrique dans le calcul du RPS. La formule complète s'écrit :

$$\text{RPS} = 0.5\text{AUC} + 0.3\text{AP} + 0.2\text{Accuracy} \quad (4.5)$$

Chapitre 4. Implémentation et tests

Composant	Poste 1	Poste 2
Processeur	Intel Core i5-12400f	Apple M1 (ARM-based)
RAM	16 Go	16 Go
GPU	NVIDIA RTX 3050 (CUDA 11.8)	Apple M1 GPU
Système d'exploitation	Windows 11	macOS Sequoia 15.4.1
Outils	VS Code, Jupyter Notebook	VS Code, Jupyter Notebook

TABLE 4.1. Caractéristiques des postes de travail utilisés

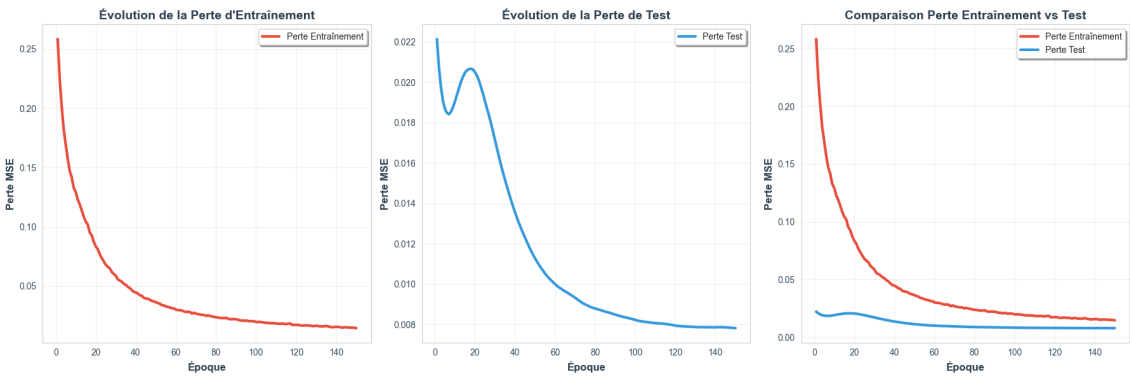


FIGURE 4.2. Évolution de la fonction de perte lors de l'entraînement de l'Embedding Encoder dans l'espace GNN.

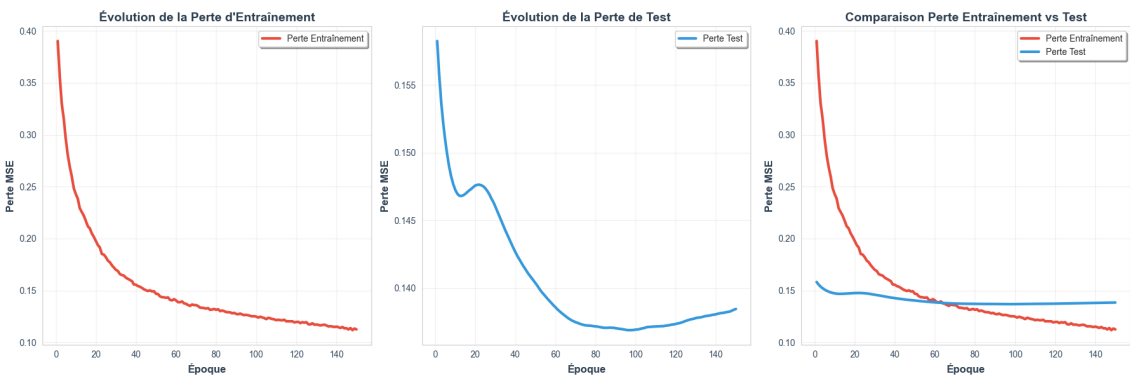


FIGURE 4.3. Évolution de la fonction de perte lors de l'entraînement de l'Embedding Encoder dans l'espace MF.

Chapitre 4. Implémentation et tests

Le RPS a été conçu pour fournir une mesure synthétique et équilibrée de la performance des modèles de recommandation en combinant plusieurs aspects complémentaires :

- **L’AUC** reçoit le poids le plus élevé (0.5) car elle évalue la capacité discriminante globale du modèle, indépendamment du choix de seuil.
- **L’Average Precision (AP)** est pondérée à 0.3 car elle se focalise sur la qualité des premières recommandations. De plus, l’AP est particulièrement adaptée aux contextes où les classes sont déséquilibrées, comme dans les tâches de recommandation où les interactions positives représentent une minorité par rapport à l’ensemble des paires possibles. Cette métrique permet ainsi d’évaluer la capacité du modèle à identifier efficacement les éléments pertinents sans être influencée par la majorité négative, contrairement à des mesures comme l’Accuracy.
- **L’Accuracy** est incluse avec un poids réduit (0.2) car, bien que représentative du taux global de bonnes prédictions, elle peut être trompeuse en présence de classes déséquilibrées, ce qui est fréquent dans les tâches de prédiction de liens ou de recommandation.

L’utilisation combinée de ces trois métriques permet d’éviter les biais liés à une seule mesure et d’obtenir une évaluation plus complète et représentative des performances réelles du modèle, tant sur sa capacité de classement global que sur la qualité des recommandations et la fiabilité des prédictions.

Le [tableau 4.2](#) permet d’interpréter la valeur du RPS et d’évaluer la qualité globale des performances du modèle.

4.6 Recherche de la configuration optimale du modèle GNN

4.6.1 Objectif des expérimentations

L’objectif de cette phase est de déterminer la combinaison optimale des hyperparamètres du modèle GNN afin de maximiser le RPS. Plusieurs séries de tests ont été réalisées en faisant varier une liste exhaustive des hyperparamètres des GNN, tels que les dimensions cachées, les dimensions des embeddings, ainsi que le nombre de têtes dans les couches d’attention. Les hyperparamètres se divisent en deux catégories : les hyperparamètres du modèle et ceux liés à l’entraînement.

Score RPS	Qualité	Interprétation	Justification
≥ 0.80	Excellente	Système hautement fiable, capable de produire des recommandations pertinentes et cohérentes.	Bon équilibre entre précision, capacité de classement et robustesse face aux variations des données.
≥ 0.75	Très bonne	Modèle performant avec un risque d'erreur de recommandation.	Bon compromis entre généralisation et spécialisation.
≥ 0.70	Bonne	Système fonctionnel ; toutefois, des ajustements pourraient renforcer sa robustesse.	Les performances sont correctes mais le modèle peut échouer sur certains sous-groupes de données.
≥ 0.60	Acceptable	Utilisable à des fins expérimentales ou de recherches.	Le modèle présente des lacunes notables en classification ou en précision de recommandation.
< 0.60	Insuffisante	Modèle inadapté aux objectifs de recommandation.	Mauvaise capacité de généralisation et de classement ; performances comparables à un modèle aléatoire ou naïf.

TABLE 4.2. Grille d'interprétation du score RPS

Hyperparamètres du modèle

- **Dimension cachée** (`hidden_dim`) – Correspond au nombre de neurones dans chaque couche intermédiaire du GNN. Ces neurones sont responsables du traitement et de la transformation des caractéristiques des utilisateurs et des items.
- **Dimension des embeddings** (`embedding_dim`) – Représente la taille de l'espace vectoriel latent dans lequel les utilisateurs et les items sont projetés. C'est la dimension du vecteur d'embedding utilisé pour encoder les caractéristiques des nœuds dans un espace commun.
- **Nombre de couches** (`num_layers`) – Indique le nombre total de couches de convolution du GNN.
- **Taux de dropout** (`dropout`) – Taux de désactivation aléatoire des neurones pendant l'entraînement, utilisé pour régulariser le modèle et limiter le surapprentissage.
- **Connexions résiduelles** (`use_residual`) – Les connexions résiduelles permettent d'ajouter l'entrée d'une couche à sa sortie, facilitant l'apprentissage des réseaux profonds en préservant l'information initiale. La sortie d'une couche avec connexion résiduelle est donnée par :

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x} \quad (4.6)$$

où $\mathbf{x}$ est l'entrée de la couche, $\mathcal{F}(\mathbf{x})$ la transformation réalisée (par exemple convolution suivie d'une activation) et $\mathbf{y}$ la sortie finale. Ce mécanisme améliore la stabilité de l'entraînement et prévient l'atténuation des gradients dans les architectures profondes, comme dans notre cas.

- **Normalisation par lots** (`use_batch_norm`) – La *Batch Normalization* est une technique d'entraînement introduite par [165] visant à stabiliser et accélérer l'apprentissage des réseaux de neurones profonds. Elle consiste à normaliser les activations d'une couche en leur appliquant une transformation statistique qui force leur moyenne à zéro et leur variance à un. Formellement, pour un mini-batch $\mathcal{B} = \{x_1, x_2, \dots, x_m\}$, la normalisation s'effectue ainsi :

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \varepsilon}}, \quad (4.7)$$

où $\mu_{\mathcal{B}}$ et $\sigma_{\mathcal{B}}^2$ sont respectivement la moyenne et la variance du mini-batch, et ε un terme de régularisation pour éviter la division par zéro. Deux paramètres appris, γ (facteur d'échelle) et β (décalage), permettent à la couche de retrouver si nécessaire l'identité :

$$y_i = \gamma \hat{x}_i + \beta. \quad (4.8)$$

Chapitre 4. Implémentation et tests

- **Mécanisme d'attention** (`use_attention` et `attention_heads`) – Le mécanisme d'attention dans les GNN permet au modèle d'accorder plus d'importance à certaines relations dans le graphe. Le paramètre `attention_heads` détermine le nombre de têtes utilisées dans l'attention multi-têtes, chaque tête apprenant à capturer un aspect spécifique des relations entre nœuds.
- **Décodeur MLP** (`simple_decoder`) – Indique si le décodeur du GNN utilise une architecture simplifiée (produit scalaire) ou une architecture plus expressive basée sur un perceptron multicouche (MLP).

Hyperparamètres d'entraînement

Les résultats des tests des hyperparamètres d'entraînement se trouvent dans [figure 1.2](#).

- **Taux d'apprentissage initial** (`lr = 0.005`) – Il contrôle la taille des pas effectués lors de la mise à jour des poids du modèle à chaque itération d'entraînement.
- **Scheduler de taux d'apprentissage** – Un `ReduceLROnPlateau` est utilisé pour ajuster dynamiquement le taux d'apprentissage en fonction des performances sur l'ensemble de validation, il accepte 3 paramètres :
 - **Mode** `mode = 'max'` – Le scheduler attend une maximisation de la métrique (ici le RPS).
 - **Facteur de réduction** (`factor = 0.75`) – En cas de stagnation du RPS, le taux d'apprentissage est multiplié par ce facteur.
 - **Patience** (`patience = 10`) – Si aucune amélioration du RPS n'est observée pendant 10 epochs consécutives, le taux d'apprentissage est réduit.
 - **Taux minimal** (`min_lr = 10-6`) – Le scheduler n'abaissera jamais le taux d'apprentissage en dessous de cette valeur.
- **Ratio d'échantillonnage négatif** (`neg_ratio = 2`) – Contrôle le ratio d'échantillonnage négatif lors de l'entraînement du modèle de recommandation.

4.6.2 Résultats expérimentaux et analyse

Ci-dessous sont présentés les résultats obtenus après l'entraînement du modèle selon différentes configurations d'hyperparamètres.

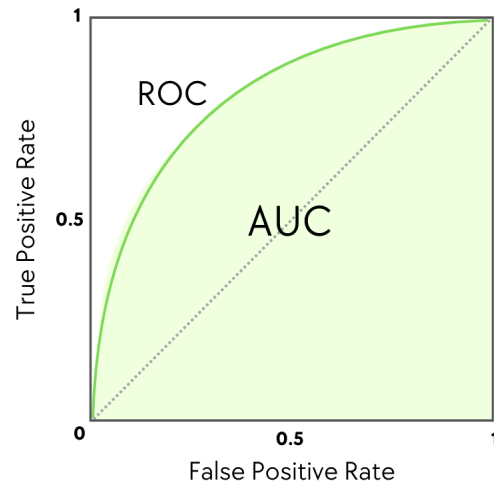


FIGURE 4.4. La courbe AUC [168].

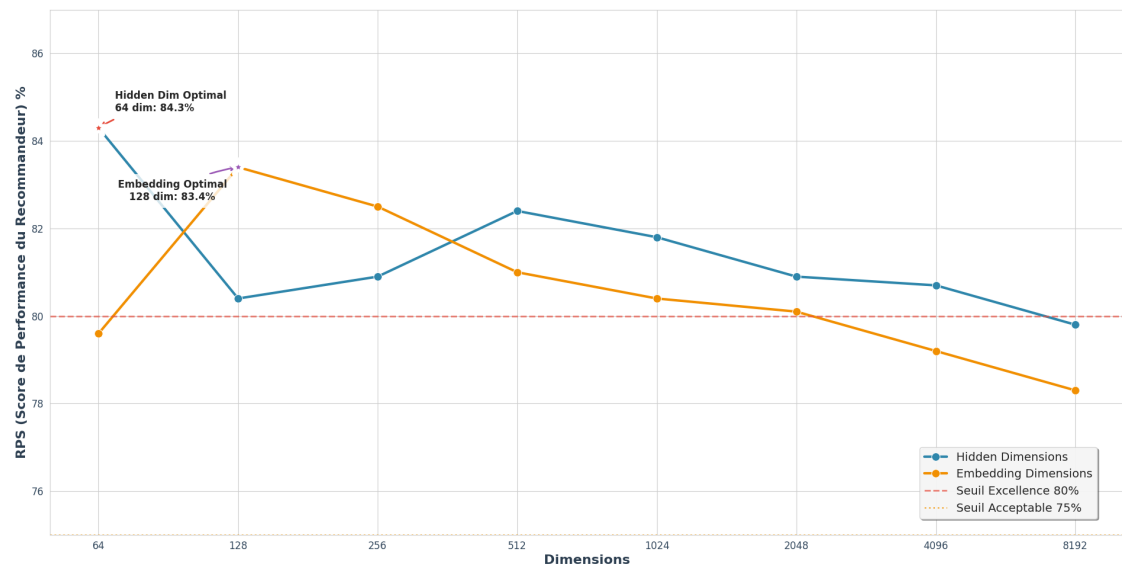


FIGURE 4.5. Évolution du RPS en fonction de `hidden_dim` et de `embedding_dim`.

Chapitre 4. Implémentation et tests

Dimension cachée et dimension des embeddings

Les expérimentations sur les hyperparamètres `hidden_dim` et `embedding_dim` (64 à 8192) montrent que le modèle atteint ses meilleures performances avec `hidden_dim` = 64 (84,3% RPS) et `embedding_dim` = 128 (83,4% RPS). Cette configuration optimale (64 dimensions cachées, 128 dimensions d’embedding) a été retenue pour le modèle final, offrant un bon équilibre entre performance et efficacité computationnelle.

Nombre de couches

L’analyse de la profondeur du modèle révèle une architecture optimale à quatre couches atteignant 83,3% RPS. Avec deux couches, les performances restent limitées (77,5%) par une extraction insuffisante des motifs structurels³ présents dans le graphe. Au-delà de cinq couches, une dégradation modérée apparaît (75,5% pour huit couches) due à l’homogénéisation progressive des représentations nodales qui diminue le pouvoir discriminant du modèle.

Dropout

L’étude du dropout révèle une configuration optimale à 0,2 atteignant 82,2% RPS. L’analyse montre trois zones distinctes : une zone optimale (0,1 à 0,3) avec des performances supérieures à 79%, une zone de transition (0,3 à 0,6) avec dégradation progressive, et une zone de sur-régularisation (au-delà de 0,6) culminant à 46% RPS pour un dropout de 1,0, soit une dégradation de 36,2% par rapport à la configuration optimale.

Mécanisme d’attention

L’intégration du mécanisme d’attention via les *Graph Attention Networks (GAT)* apporte un gain substantiel de +4,1% RPS, passant de 80,9% (configuration de base) à 85,0% avec l’attention multi-têtes. La configuration optimale est atteinte avec seulement 2 têtes d’attention, les performances diminuant graduellement au-delà (83,5% pour 4 têtes, puis stabilisation entre 81,8% et 82,9% pour 8 à 64 têtes), démontrant qu’une approche minimaliste est plus efficace dans notre contexte.

3. Les *motifs structurels* représentent des sous-structures caractéristiques récurrentes dans le graphe, telles que des hubs, des ponts ou des chemins, qui traduisent des relations importantes entre les nœuds. Dans le contexte de la recommandation, ils permettent de capturer des similarités d’usage ou des comportements partagés entre groupes d’utilisateurs ou d’items.

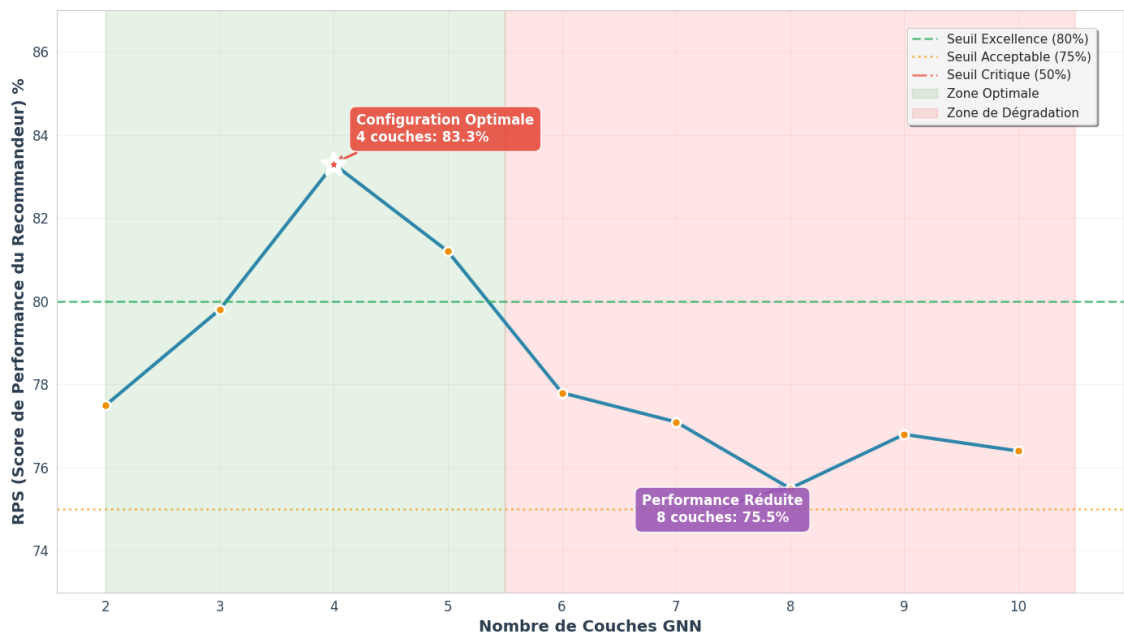


FIGURE 4.6. Évolution du RPS en fonction du nombre de couches (nb_layers).

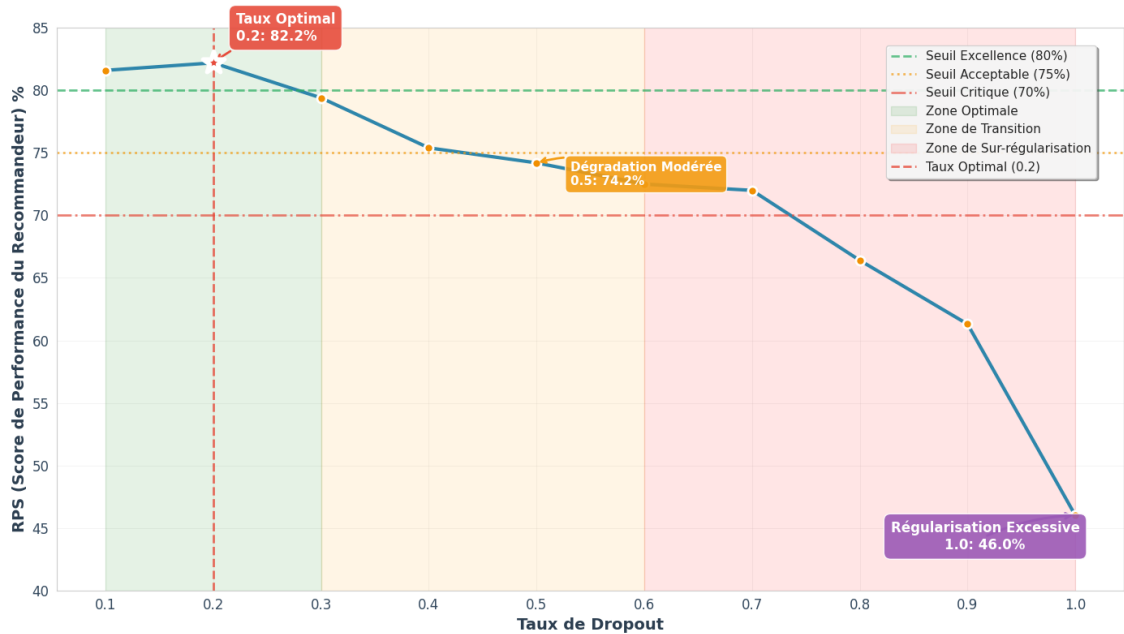


FIGURE 4.7. Évolution du RPS en fonction du dropout.

Chapitre 4. Implémentation et tests

Composants architecturaux

L’analyse des composants booléens révèle trois contributions positives cumulatives : les connexions résiduelles apportent le gain le plus significatif (+7,4%, de 74,9% à 82,3%) en atténuant le gradient évanescent, la normalisation par lots procure un gain modéré (+1,5%, de 82,3% à 83,8%) en stabilisant l’entraînement, et le décodeur MLP génère un gain significatif (+4,3%, de 79,1% à 83,4%) en capturant les interactions non linéaires complexes. La configuration optimale combine les trois composants activés.

4.6.3 Configuration optimale

La configuration optimale du modèle GNN, déterminée après l’exploration approfondie des hyperparamètres, est résumée dans le tableau [tableau 4.3](#), et le GNN généré est associé à un total de 132 545 paramètres entraînables.

L’évolution des métriques d’apprentissage du modèle selon cette configuration optimale est illustrée dans la [figure 4.10](#).

Performances finales

Les performances finales obtenues par le modèle GNN sur l’ensemble de test sont présentées dans la [tableau 4.4](#).

Phase de Fine-Tuning

Compte tenu de la taille relativement réduite de la base de données, une seconde phase d’entraînement a été réalisée sur l’ensemble complet des données disponibles (sans séparation en ensemble de test), précédée d’une étape de *fine-tuning*⁴. Cette stratégie a permis d’exploiter pleinement l’information présente, tout en bénéficiant des connaissances acquises lors de l’entraînement supervisé initial, afin de générer des embeddings plus représentatifs et robustes pour une utilisation finale en production [169].

4. Le *fine-tuning* désigne une phase d’ajustement des poids d’un modèle préalablement entraîné, sur l’ensemble complet des données ou sur un sous-ensemble spécifique, dans le but d’améliorer sa spécialisation et ses performances finales. Cette technique permet de conserver les embeddings apprises tout en les adaptant plus finement à la distribution globale des données.

Chapitre 4. Implémentation et tests

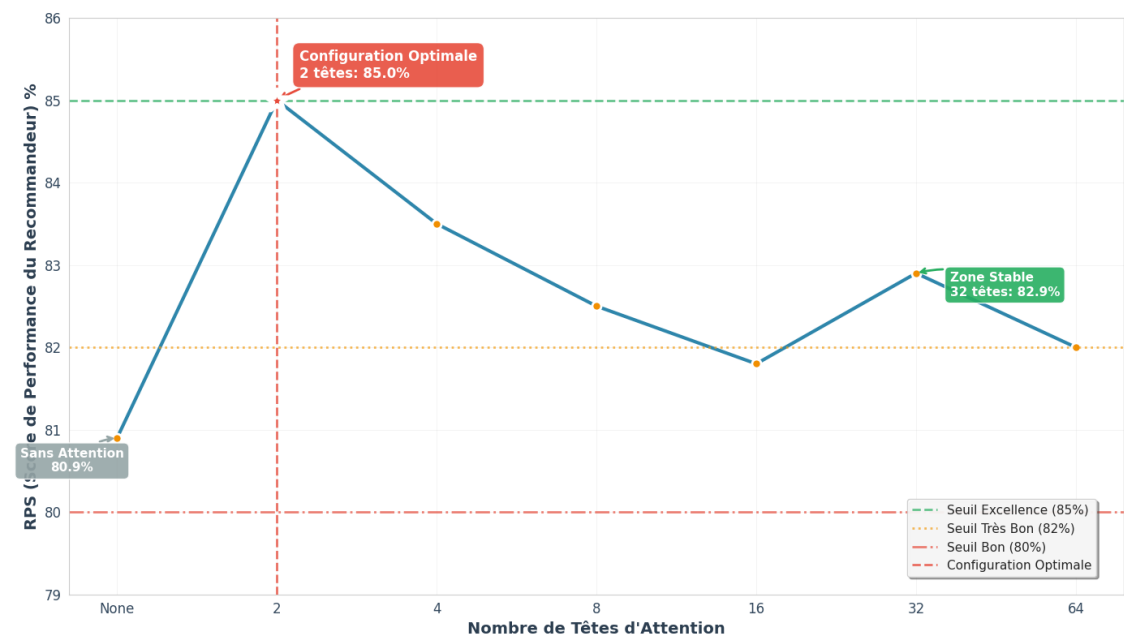


FIGURE 4.8. Évolution du RPS en fonction de l'utilisation du mécanisme d'attention et du nombre de têtes d'attention.

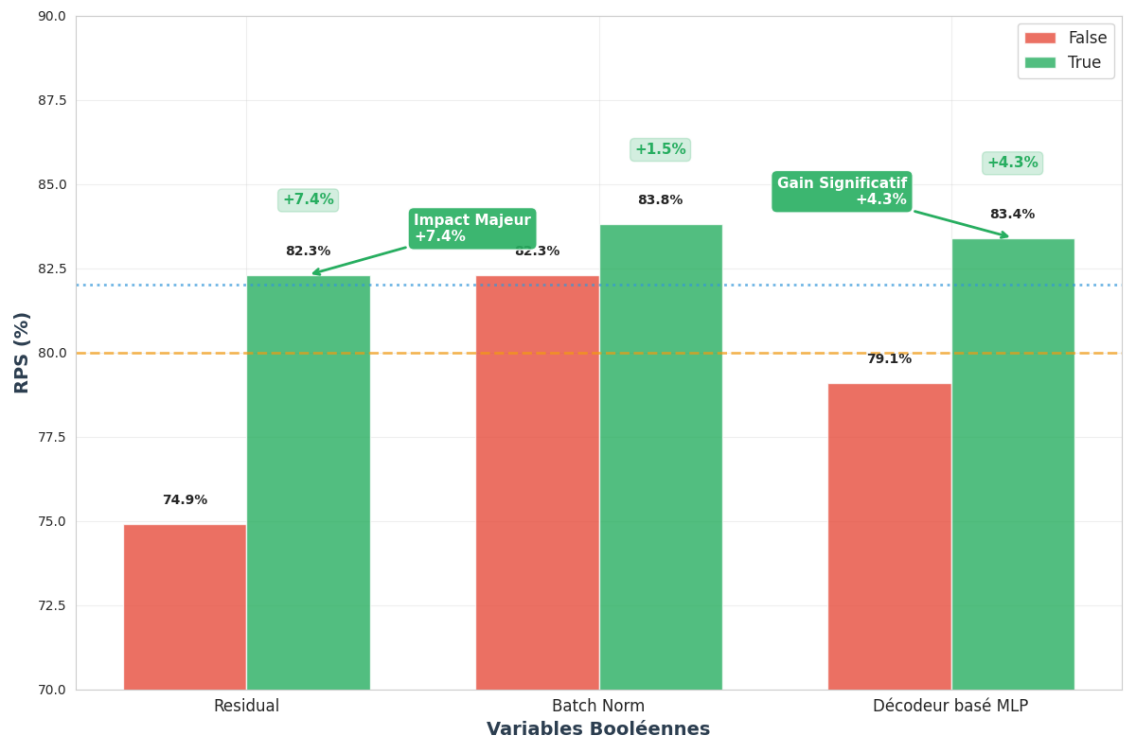


FIGURE 4.9. Évolution du RPS selon l'utilisation des paramètres booléens.

Chapitre 4. Implémentation et tests

Hyperparamètre	Valeur optimale
Dimension cachée (<code>hidden_dim</code>)	64
Dimension des embeddings (<code>embedding_dim</code>)	128
Nombre de couches GNN (<code>num_layers</code>)	4
Taux de dropout (<code>dropout</code>)	0.2
Connexions résiduelles (<code>use_residual</code>)	Oui
Batch Normalisation (<code>use_batch_norm</code>)	Oui
Mécanisme d'attention (<code>use_attention</code>)	Oui (2 têtes)
Type de décodeur (<code>simple_decoder</code>)	Décodeur basé MLP
Taux d'apprentissage (<code>learning_rate</code>)	0.005
Patience du scheduler (<code>patience</code>)	15
Facteur de réduction du LR (<code>factor</code>)	0.75
Taux d'apprentissage minimal (<code>min_lr</code>)	110^{-6}
Paramètre Focal Loss γ	3.0
Paramètre Focal Loss α	0.25
Ratio d'échantillonnage négatif (<code>neg_ratio</code>)	2

TABLE 4.3. Hyperparamètres de la meilleure configuration du modèle

Métrique	Valeur obtenue (%)
AUC	92.4
Average Precision (AP)	71.8
Accuracy	86.3
RPS (Recommender Performance Score)	85.0

TABLE 4.4. Performances finales du modèle GNN recommandé (ensemble de test)

Chapitre 4. Implémentation et tests

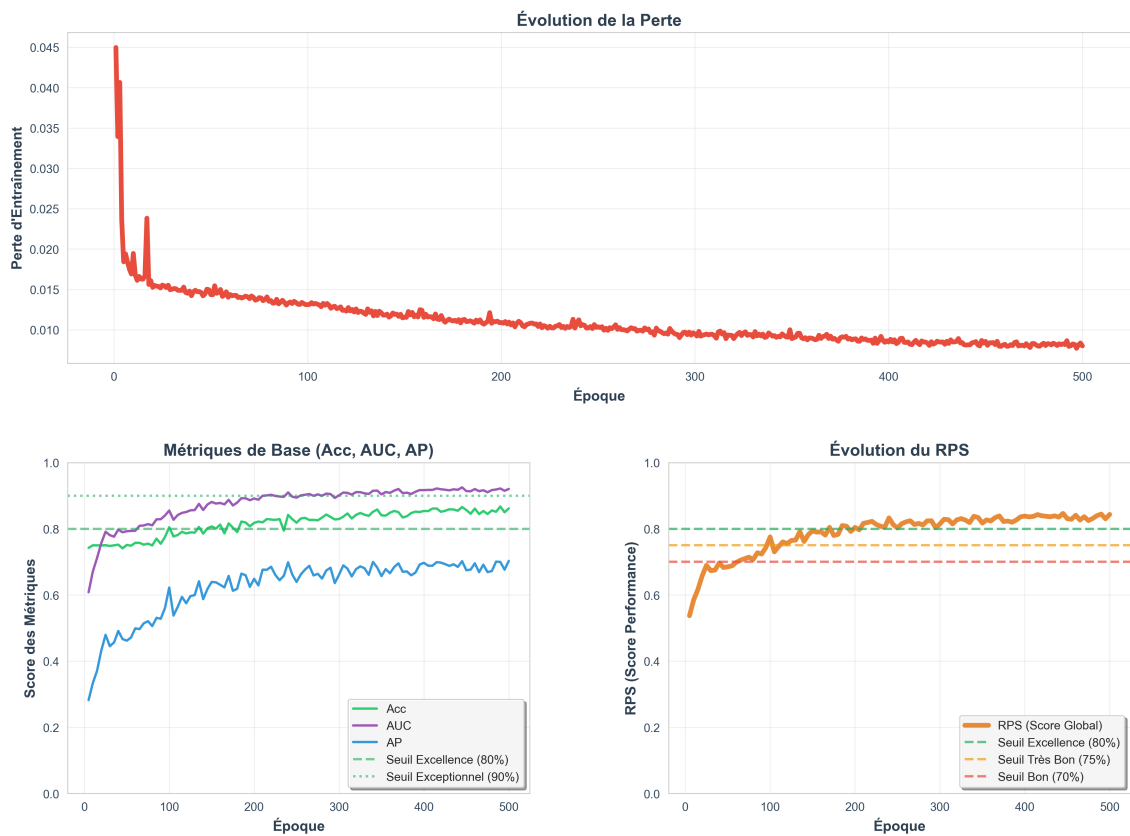


FIGURE 4.10. Évolution des métriques d'entraînement pour la configuration optimale.

Chapitre 4. Implémentation et tests

Algorithm 2: Fine-tuning du Modèle GNN

Require: Modèle pré-entraîné : `model`, dataset : `data`

Ensure: Modèle fine-tuné et métriques de performance associées

- 1: `lr` $\leftarrow$ 0.0005 ▷ Faible taux d'apprentissage pour ajustement fin des poids
 - 2: `optimizer` $\leftarrow$ ADAM(`model.parameters()`, `lr`)
 - 3: `epochs` $\leftarrow$ 20 ▷ Nombre limité d'époques pour éviter le surapprentissage
 - 4: `model, results` $\leftarrow$ TRAIN_MODEL(`model`, `data`, `optimizer`, `epochs`)
 - 5: **return** Modèle fine-tuné `model` et métriques de performance `results`
-

Les performances obtenues après cette étape de fine-tuning et l'entraînement complet sont synthétisées dans la [tableau 4.5](#).

Un script a été développé et constitue l'outil principal d'évaluation du modèle GNN de recommandation d'exercices de physiothérapie sur l'ensemble complet du jeu de données. Ce programme effectue une évaluation exhaustive des performances du système de recommandation. L'évaluation adopte une approche individualisée où, pour chaque patient, le modèle génère un nombre de recommandations K correspondant exactement au nombre d'exercices qui lui ont été effectivement prescrits dans les données de référence.

Enfin, la [figure 4.11](#) présente une comparaison visuelle entre les dix exercices les plus recommandés par le modèle et les dix exercices les plus fréquemment présents dans la base de données d'origine. Cette analyse permet d'évaluer la cohérence des suggestions générées par le système de recommandation par rapport à la distribution réelle des exercices prescrits. Un bon recouvrement entre ces deux listes indique que le modèle a bien capté les tendances globales des données, tandis que des divergences peuvent révéler des biais d'apprentissage ou des capacités de découverte d'exercices sous-représentés.

4.7 Recherche de la configuration optimale du modèle MF

Pour le modèle de MF, l'optimisation des hyperparamètres est considérablement simplifiée par rapport aux architectures de réseaux de neurones complexes, car il n'existe qu'un seul hyperparamètre critique à ajuster : le nombre de facteurs latents.

La [figure 4.12](#) présente une analyse de l'impact du nombre de facteurs latents sur les performances du modèle MF, testant 13 configurations différentes allant de 2 à 256 facteurs. Les métriques de base (panneau de gauche) révèlent des tendances distinctes : l'AUC (courbe bleue) progresse rapidement jusqu'à 32 facteurs puis se stabilise autour de 83-84% avec un optimum de 83.6% à 150 facteurs. L'accuracy

Chapitre 4. Implémentation et tests

Métrique	Valeur obtenue (%)
AUC	94.77
Average Precision (AP)	87.41
Accuracy	87.97
RPS (Recommender Performance Score)	91.20

TABLE 4.5. Performances finales du modèle GNN après fine-tuning et entraînement sur l'ensemble de la base de données

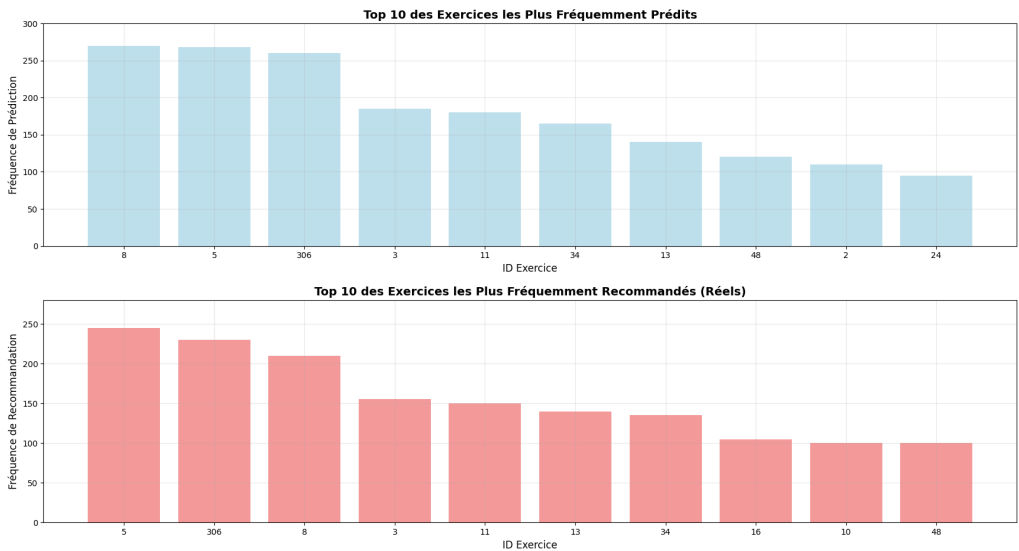


FIGURE 4.11. Comparaison entre les dix exercices les plus recommandés par le modèle et les dix exercices les plus fréquents dans la base de données.

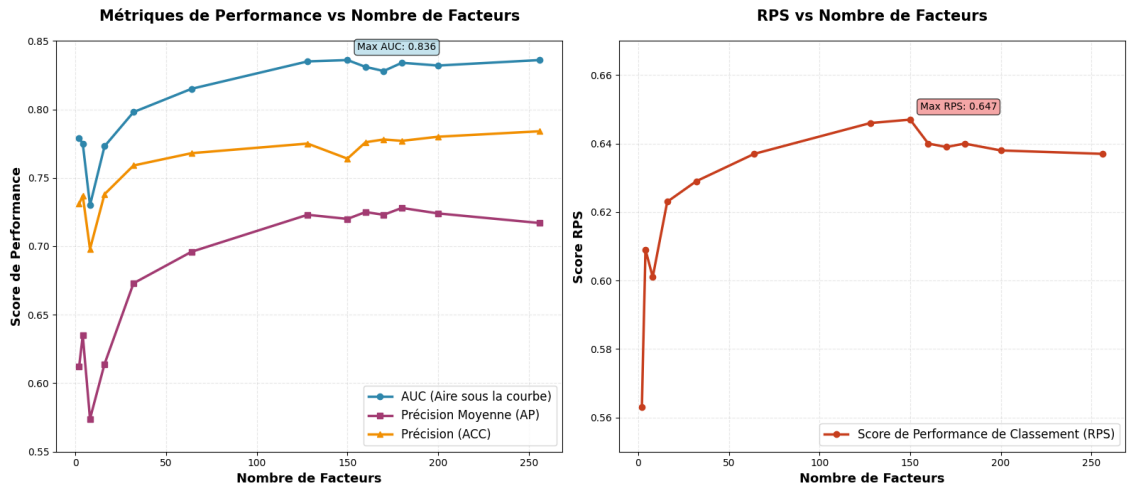


FIGURE 4.12. Analyse de l'impact du nombre de facteurs latents sur les performances du modèle MF.

(courbe orange) montre une amélioration constante, passant de 73.1% avec 2 facteurs à 78.4% avec 256 facteurs, avec une chute notable à 8 facteurs (69.8%). La précision moyenne (courbe violette) présente une évolution plus erratique, partant de 61.2% avec 2 facteurs pour atteindre son maximum de 72.8% à 180 facteurs. L'évolution du score RPS (panneau de droite) met en évidence la configuration optimale à 150 facteurs avec un score de 64.7%, montrant une progression rapide jusqu'à 128 facteurs puis une stabilisation. Cette analyse démontre que les performances du modèle MF atteignent leur optimum autour de 150 facteurs, avec des rendements décroissants ou une stagnation au-delà de cette valeur, suggérant un équilibre entre complexité du modèle et généralisation.

La configuration optimale du modèle de factorisation de matrice (MF) est résumée dans le tableau [tableau 4.6](#), accompagnée des métriques obtenues sur le jeu de test. Le modèle généré comporte un total de 73 085 paramètres entraînaibles.

4.8 Interface graphique

Pour faciliter l'utilisation des deux systèmes de recommandation, une application web a été développée. Relativement simple d'usage, elle permet, comme illustré dans la [figure 4.13](#), de générer des recommandations d'exercices pour les patients déjà présents dans la base de données, ainsi que pour les nouveaux patients absents lors de l'entraînement, tout en offrant la possibilité de choisir entre les approches MF ou GNN.

La [figure 4.14](#) présente les résultats des recommandations générées.

L'application permet également de générer des recommandations pour de nouveaux patients, comme illustré dans la [figure 4.15](#).

4.9 Comparaison des performances entre le modèle GNN et MF

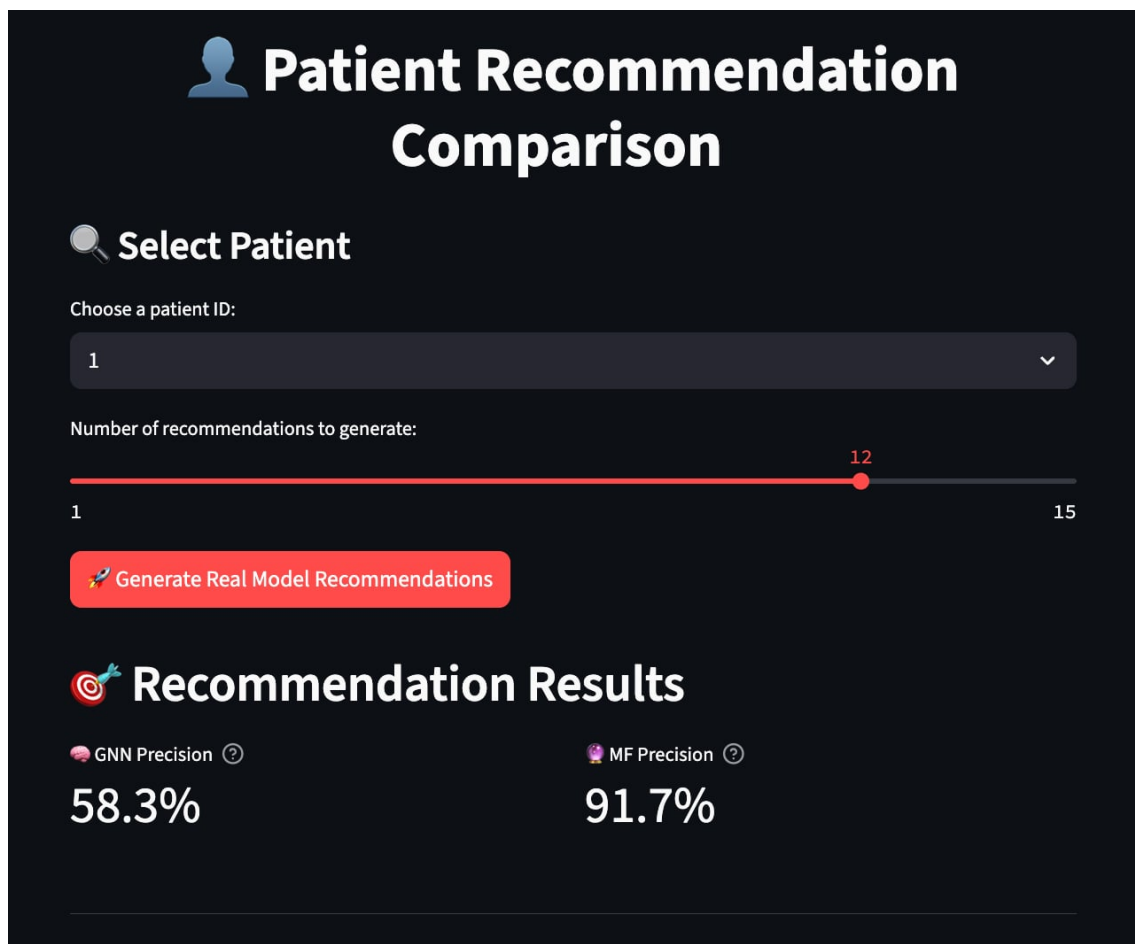
L'évaluation comparative entre nos modèles GNN et MF met en évidence la nette supériorité de l'architecture basée sur les graphes. Comme illustré dans la [figure 4.16](#), le modèle GNN surpasse significativement le modèle MF sur l'ensemble des métriques d'évaluation, avec une amélioration relative moyenne de +22,7 %. L'amélioration la plus marquante concerne le score RPS, qui affiche un gain de +26,5 % (91,20 % contre 64,70 %).

Bien que le GNN démontre globalement de meilleures performances, le modèle MF reste très performant pour les patients présents dans l'ensemble d'entraînement,

Chapitre 4. Implémentation et tests

Hyperparamètre	Valeur optimale	AUC (%)	AP (%)	ACC (%)	RPS (%)
Nombre de facteurs latents	150	83.6	72.0	76.4	64.7

TABLE 4.6. Hyperparamètres et performances de la meilleure configuration du modèle MF.



The image shows a web interface titled "Patient Recommendation Comparison". It features a "Select Patient" section with a search icon and a dropdown menu showing "1". Below this is a slider for "Number of recommendations to generate:" ranging from 1 to 15, with a red dot at 12. A red button labeled "Generate Real Model Recommendations" is positioned below the slider. The "Recommendation Results" section displays two metrics: "GNN Precision" at 58.3% and "MF Precision" at 91.7%, each with a question mark icon.

Patient Recommendation Comparison

 **Select Patient**

Choose a patient ID:

1

Number of recommendations to generate:

1 12 15

 **Generate Real Model Recommendations**

Recommendation Results

 **GNN Precision** ? **58.3%**

 **MF Precision** ? **91.7%**

FIGURE 4.13. Génération d'exercices pour des patients présents dans la base de données.

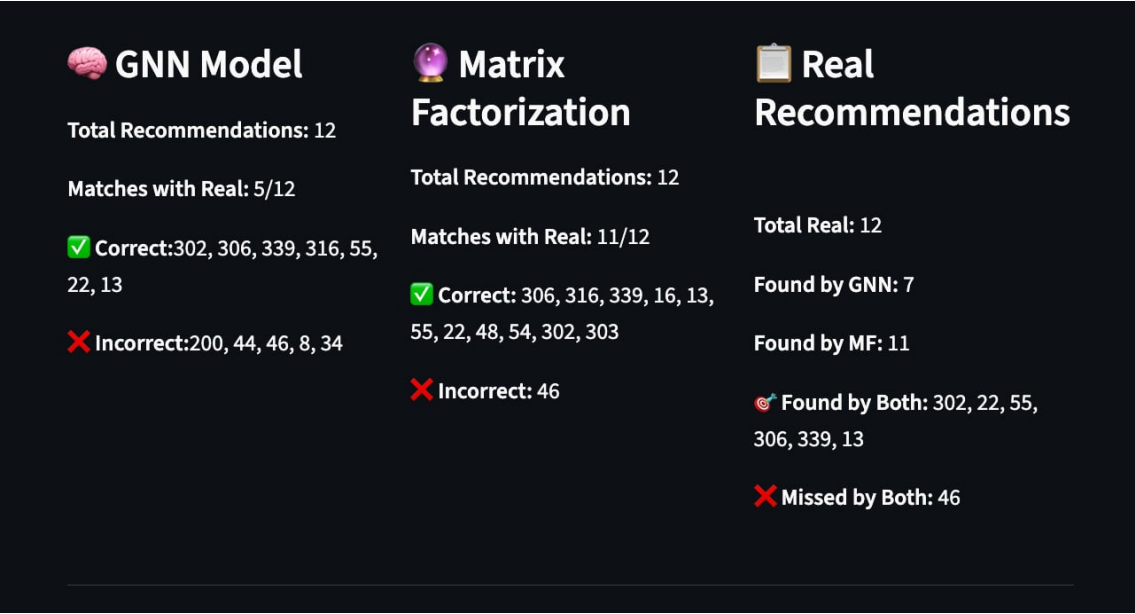


FIGURE 4.14. Résultats des recommandations.

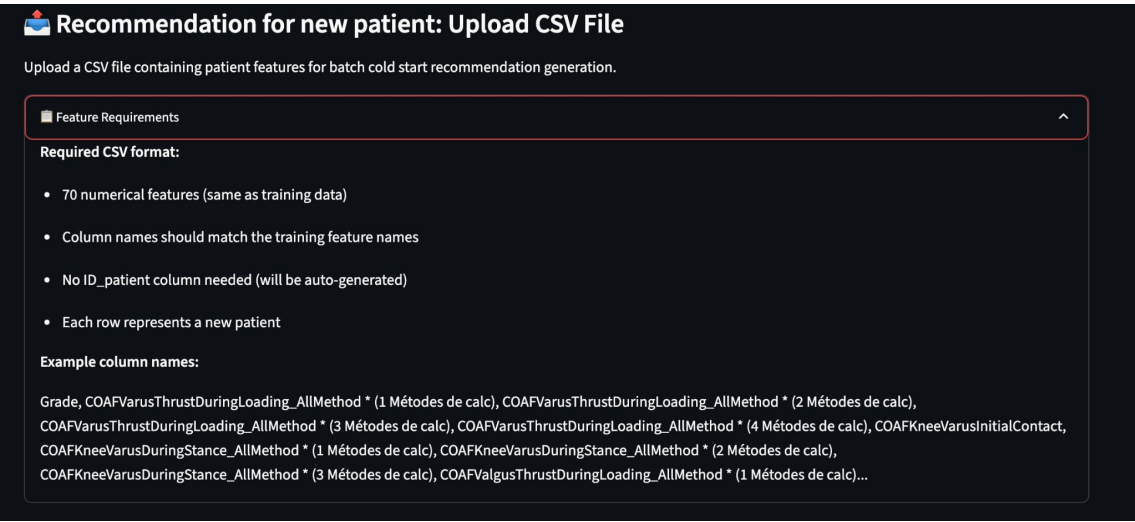


FIGURE 4.15. Génération de recommandations pour un nouveau patient (cas *cold-start*).

Chapitre 4. Implémentation et tests

comment il est illustre dans [figure 4.13](#). En effet, les embeddings générés par MF sont directement optimisés pour ces patients spécifiques, reflétant fidèlement les interactions observées durant l'apprentissage. Toutefois, ces représentations ne sont pertinentes que pour les données vues lors de l'entraînement et peinent à généraliser à de nouveaux patients, limitant ainsi leur capacité d'adaptation dans des scénarios de recommandation plus larges.

Cette supériorité s'explique par trois facteurs clés :

- Premièrement, contrairement à la MF qui se limite aux interactions historiques, le GNN intègre explicitement les caractéristiques des patients et des exercices, permettant de capturer des relations sémantiques complexes.
- Deuxièmement, le mécanisme d'attention via les GAT attribue dynamiquement des poids différenciés aux relations, alors que la MF traite toutes les interactions uniformément.
- Troisièmement, la MF souffre du problème de démarrage à froid et génère des embeddings qui sont propres aux données d'entraînement, tandis que le GNN apprend des relations logiques généralisables entre profils pathologiques et propriétés thérapeutiques.

Ces résultats démontrent l'avantage fondamental des GNN dans la modélisation des dépendances structurelles pour la recommandation d'exercices de rééducation, générant des embeddings plus expressifs et des prédictions cliniquement plus pertinentes.

4.10 Conclusion

Les expérimentations démontrent la supériorité de l'approche GNN par rapport à la MF pour la recommandation d'exercices de rééducation. L'architecture optimale identifiée (4 couches, 64 dimensions cachées, 128 embeddings, 2 têtes d'attention) atteint un excellent équilibre entre performance et efficacité computationnelle. L'analyse révèle l'importance cruciale des connexions résiduelles (+7,4% RPS) et du décodeur MLP (+4,3% RPS).

La supériorité du GNN s'explique par sa capacité à exploiter les caractéristiques intrinsèques des patients et exercices, à attribuer dynamiquement des poids via l'attention, et à généraliser vers de nouveaux patients, contrairement aux limitations de la MF. L'Embedding Encoder garantit des recommandations cohérentes pour les nouveaux patients, validant l'applicabilité clinique du système et ouvrant la voie vers une personnalisation thérapeutique optimisée.

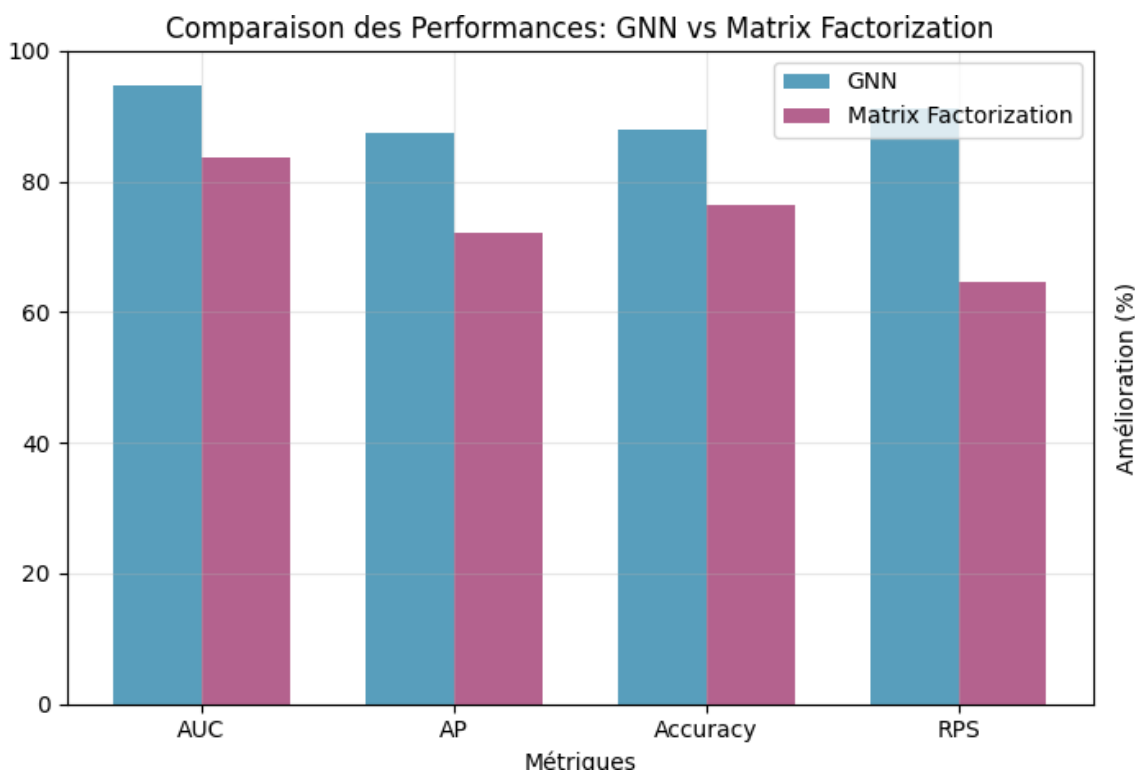


FIGURE 4.16. Comparaison des performances entre le modèle GNN et MF.

Conclusion Générale

Cette étude a permis de concevoir, d'implémenter et d'évaluer un système de recommandation d'exercices thérapeutiques spécifiquement adapté aux patients atteints de gonarthrose, en tirant parti des Graph Neural Networks (GNN) et de la Matrix Factorization (MF). Les résultats expérimentaux ont clairement démontré la supériorité significative du modèle basé sur les GNN par rapport à l'approche de MF. Le GNN a atteint un Recommender Performance Score (RPS) de 91,20%, soit une amélioration notable de +26,5% par rapport aux 64,70% obtenus par le modèle MF.

Cette performance accrue s'explique par la capacité intrinsèque du GNN à intégrer des caractéristiques riches des patients et des exercices, à capturer des relations non linéaires complexes via son architecture multicouche, et à pondérer dynamiquement l'importance des connexions grâce à son mécanisme d'attention. En revanche, la MF, bien que moins coûteuse en calcul, est limitée par son incapacité à incorporer directement des caractéristiques explicites et par son traitement uniforme des interactions, ce qui réduit sa capacité à modéliser les dynamiques nuancées patient-exercice.

Une contribution essentielle de ce travail réside dans la résolution efficace du problème du "cold start" grâce au développement et à l'intégration d'un Embedding Encoder. Ce composant garantit que le système peut générer des recommandations cohérentes et pertinentes pour les nouveaux patients, même en l'absence de données historiques d'interaction étendues. En projetant les caractéristiques des nouveaux patients dans l'espace latent appris par le modèle GNN ou MF, l'Embedding Encoder permet au système de fournir des suggestions thérapeutiques personnalisées dès le premier contact, améliorant ainsi son applicabilité clinique.

L'analyse a également mis en évidence le rôle crucial de composants architecturaux tels que les connexions résiduelles et le décodeur MLP, qui ont significativement amélioré les performances en résolvant des problèmes comme l'atténuation du gradient et en augmentant la capacité du modèle à apprendre des motifs non linéaires complexes.

En synthèse, cette recherche valide non seulement le potentiel des GNN pour

Chapitre 4. Implémentation et tests

développer des systèmes de recommandation de santé hautement efficaces, mais elle établit également un cadre robuste pour la prescription personnalisée d'exercices thérapeutiques dans le contexte de la gonarthrose. En dépassant les programmes d'exercices généralisés, le système proposé ouvre la voie à l'optimisation des résultats de rééducation, à l'amélioration de l'adhésion des patients et, in fine, à l'amélioration de la qualité de vie des personnes atteintes de cette affection débilante.

Les travaux futurs pourraient explorer l'intégration de mécanismes de rétroaction en temps réel et l'élargissement de l'ensemble de données pour affiner et valider davantage les capacités du système dans divers environnements cliniques.

Bibliographie

- [1] Arthrose du genou (gonarthrose), symptômes et traitements, . URL <https://www.ramsaysante.fr/vous-etes-patient-en-savoir-plus-sur-ma-pathologie/arthrose-du-genou>.
- [2] Henry Knipe. Osteoarthritis of the knee | Radiology Reference Article | Radiopaedia.org. <https://radiopaedia.org/articles/osteoarthritis-of-the-knee>.
- [3] What Is Osteoarthritis (OA) of the Knee? <https://my.clevelandclinic.org/health/diseases/21750-osteoarthritis-knee>, .
- [4] Overview : Osteoarthritis of the knee. In *InformedHealth.Org [Internet]*. Institute for Quality and Efficiency in Health Care (IQWiG), May 2024.
- [5] Marco Gupton, Onyebuchi Imonugo, Asa C. Black, Marjorie V. Launico, and Robert R. Terreberry. Anatomy, Bony Pelvis and Lower Limb, Knee. In *StatPearls*. StatPearls Publishing. URL <http://www.ncbi.nlm.nih.gov/books/NBK500017/>.
- [6] M. Shahabpour, N. DeMeyere, M. DeMaeseneer, P. David, F. DeRidder, and T. Stadnik. Anatomie normale du genou en imagerie par résonance magnétique. *EMC - Radiologie*, 2(2) :165–182, May 2005. ISSN 1762-4185. doi : 10.1016/j.emcrad.2004.12.001.
- [7] Turner A Blackburn and Emily Craig. Knee Anatomy : A Brief Review. *Physical Therapy*, 60(12) :1556–1560, December 1980. ISSN 1538-6724. doi : 10.1093/ptj/60.12.1556.
- [8] Anatomie du genou : os, cartilage, muscles, tendons, ménisques, nerfs, .
- [9] Leg skeletal anatomy : MedlinePlus Medical Encyclopedia Image. <https://medlineplus.gov/ency/imagepages/8844.htm>.

Bibliographie

- [10] Femur (Thighbone) : Anatomy, Function & Common Conditions. <https://my.clevelandclinic.org/health/body/22503-femur>, .
- [11] Femur. <https://www.kenhub.com/en/library/anatomy/femur>, .
- [12] Dr.AFOUTNI A. Muscles de la cuisse, 2022.
- [13] Articulation du genou. <https://www.kenhub.com/fr/library/anatomie/articulation-du-genou>, .
- [14] Docteur Philippe Paris Chirurgien Orthopédiste Marseille | Genou et sport, Hanche, Cheville et Pied. <https://www.chirortho13.fr/>.
- [15] The Knee Joint - Articulations - Movements - Injuries - TeachMeAnatomy, . URL <https://teachmeanatomy.info/lower-limb/joints/knee-joint/>.
- [16] Tibia (Shin Bone) : Location, Anatomy & Common Conditions. URL <https://my.clevelandclinic.org/health/body/23026-tibia>.
- [17] Perone anatomie artistique. <https://medecine-des-arts.com/fr/article/perone-anatomie-artistique-lecon-43.php>, .
- [18] Péroné - LAROUSSE. <https://www.larousse.fr/encyclopedie/medical/p%C3%A9ron%C3%A9>, .
- [19] The Patella - Surface Anatomy - Functions - Dislocation - TeachMeAnatomy. <https://teachmeanatomy.info/lower-limb/bones/patella/>, .
- [20] Chandler F. Cox, Margaret A. Sinkler, Asa C. Black, Marjorie V. Launico, and John B. Hubbard. Anatomy, Bony Pelvis and Lower Limb, Knee Patella. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), 2025.
- [21] Patella (Rotule). <https://www.kenhub.com/fr/library/anatomie/patella-fr>, .
- [22] Knee Joint : Function & Anatomy. <https://my.clevelandclinic.org/health/body/24777-knee-joint>, .
- [23] Knee Joint Anatomy : Overview, Gross Anatomy, Natural Variants. 2025-02-19-18 :09.
- [24] damien. Quadriceps Anatomy And Training - Blog Eric Favre UK, July 2021.
- [25] Symptômes et diagnostic de la rupture du tendon rotulien | Dr Paillard.

Bibliographie

- [26] Knee Anatomy : Muscles, Ligaments, and Cartilage. <https://www.joionline.net/library/knee-anatomy-muscles-ligaments-and-cartilage/>, .
- [27] Soigner la maladie d'Osgood-Schlatter - ORTHOREPASS. <https://www.orthorepass.com/soigner-la-maladie-dosgood-schlatter/>.
- [28] Patellar tendinitis - Symptoms & causes - Mayo Clinic. <https://www.mayoclinic.org/diseases-conditions/patellar-tendinitis/symptoms-causes/syc-20376113>, .
- [29] Syed Zahid Hussain Bokhari. Tendonitis : The major cause of pain in osteoarthritis knee joint. *Journal of Ayub Medical College, Abbottabad : JAMC*, 24 (3-4) :109–112, 2012. ISSN 1025-9589.
- [30] Jawad F. Abulhasan and Michael J. Grey. Anatomy and Physiology of Knee Stability. *Journal of Functional Morphology and Kinesiology*, 2(4) :34, December 2017. ISSN 2411-5142. doi : 10.3390/jfink2040034.
- [31] Dr. Justin Newman | Orthopedic Surgeon Parker | Sports Medicine Castle Rock CO. <https://www.justinnewmanmd.com/normal-anatomy-of-the-knee-joint/>.
- [32] Joint Instability and developing degenerative joint disease and progressive osteoarthritis.
- [33] Patient Education. <https://www.concordortho.com/patient-resources/patient-education/topic/8e9ebd3d2fd79fd452e155cb85d1bade>, .
- [34] James Kevin Bryceland, Andrew John Powell, and Thomas Nunn. Knee Menisci : Structure, Function, and Management of Pathology. 8(2) :99–104. ISSN 1947-6035, 1947-6043. doi : 10.1177/1947603516654945. URL <https://journals.sagepub.com/doi/10.1177/1947603516654945>.
- [35] Anatomie du ménisque, . URL <https://www.chirurgiedusport.com/nos-specialites/anatomie-du-menisque/>.
- [36] Knee – Degenerative Meniscal Tears - NHS Ayrshire & Arran, November 2023.
- [37] Knee meniscus tear with arthritis : Treatment, link, and more. <https://www.medicalnewstoday.com/articles/how-to-treat-knee-meniscus-tear-with-arthritis>, October 2022.

Bibliographie

- [38] Rouhin Sen and John A. Hurley. Osteoarthritis. In *StatPearls*. StatPearls Publishing, Treasure Island (FL), 2025.
- [39] Yuchen He, Zhong Li, Peter G. Alexander, Brian D. Ocasio-Nieves, Lauren Yocum, Hang Lin, and Rocky S. Tuan. Pathogenesis of Osteoarthritis : Risk Factors, Regulatory Pathways in Chondrocytes, and Experimental Models. *Biology*, 9(8) :194, August 2020. ISSN 2079-7737. doi : 10.3390/biology9080194.
- [40] Olivia Carmen Timnea, Ana Maria Stănescu, Larisa Potop, and Andreea Consuela Timnea. Kinetic Rehabilitation For Adults With Gonarthrosis. *European Proceedings of Social and Behavioural Sciences*, Education and Sports Science in the 21st Century, February 2019. ISSN 2357-1330. doi : 10.15405/epsbs.2019.02.46.
- [41] L. Sharma, J. Song, D. T. Felson, S. Cahue, E. Shamiyeh, and D. D. Dunlop. The role of knee alignment in disease progression and functional decline in knee osteoarthritis. *JAMA*, 286(2) :188–195, July 2001. ISSN 0098-7484. doi : 10.1001/jama.286.2.188.
- [42] Knee Osteoarthritis. https://www.physio-pedia.com/Knee_Osteoarthritis, .
- [43] Stefano Coaccioli, Piercarlo Sarzi-Puttini, Panagiotis Zis, Giuseppe Rinonapoli, and Giustino Varrassi. Osteoarthritis : New Insight on Its Pathophysiology. 11(20) :6013. ISSN 2077-0383. doi : 10.3390/jcm11206013. URL <https://www.mdpi.com/2077-0383/11/20/6013>.
- [44] Anna Litwic, Mark H. Edwards, Elaine M. Dennison, and Cyrus Cooper. Epidemiology and burden of osteoarthritis. *British Medical Bulletin*, 105(1) : 185–199, March 2013. ISSN 0007-1420. doi : 10.1093/bmb/lds038.
- [45] Youssef Ouakrim. Classification de sujets asymptomatiques et gonarthrosiques en fonction des données cinématiques comparaison de l’approche globale et de l’approche locale. Master’s thesis, École de technologie supérieure, Montréal, October 2011.
- [46] Knee Kinesiology, a Dynamic Knee Exam | KNEEKG®. <https://emovi.ca/>, .
- [47] What Is a Knee Kinesiology Exam ? | EMOVI® | KNEEKG®, .
- [48] Julien Clément, Panagiota Toliopoulos, Nicola Hagemeister, François Desmeules, Alexandre Fuentes, and Pascal-André Vendittoli. Healthy 3D knee

Bibliographie

- kinematics during gait : Differences between women and men, and correlation with x-ray alignment. *Gait & Posture*, 64 :198–204, July 2018. ISSN 1879-2219. doi : 10.1016/j.gaitpost.2018.06.024.
- [49] Dafina Bytyqi, Bujar Shabani, Sebastien Lustig, Laurence Cheze, Natyra Karahoda Gjurgjeala, and Philippe Neyret. Gait knee kinematic alterations in medial osteoarthritis : Three dimensional assessment. *International Orthopaedics*, 38(6) :1191–1198, June 2014. ISSN 1432-5195. doi : 10.1007/s00264-014-2312-3.
- [50] Neila Mezghani, Youssef Ouakrim, Alexandre Fuentes, Amar Mitiche, Nicola Hagemeister, Pascal-André Vendittoli, and Jacques A. de Guise. Mechanical biomarkers of medial compartment knee osteoarthritis diagnosis and severity grading : Discovery phase. *Journal of Biomechanics*, 52 :106–112, February 2017. ISSN 1873-2380. doi : 10.1016/j.jbiomech.2016.12.022.
- [51] Neila Mezghani, Karine Boivin, Katia Turcot, Rachid Aissaoui, Nicola Hagemeister, and Jacques A. De Guise. Hierarchical analysis and classification of asymptomatic and knee osteoarthritis gait patterns using a wavelet representation of kinetic data and the nearest neighbor classifier. *Journal of Mechanics in Medicine and Biology*, 08(01) :45–54, March 2008. ISSN 0219-5194. doi : 10.1142/S0219519408002474.
- [52] Xiaolong Zeng, Limin Ma, Zefeng Lin, Wenhan Huang, Zhiqiang Huang, Yu Zhang, and Chuanbin Mao. Relationship between Kellgren-Lawrence score and 3D kinematic gait analysis of patients with medial knee osteoarthritis using a new gait system. *Scientific Reports*, 7(1) :4080, June 2017. ISSN 2045-2322. doi : 10.1038/s41598-017-04390-5.
- [53] David J. Hunter. Focusing osteoarthritis management on modifiable risk factors and future therapeutic prospects. *Therapeutic Advances in Musculoskeletal Disease*, 1(1) :35–47, October 2009. ISSN 1759-720X. doi : 10.1177/1759720X09342132.
- [54] N. Mezghani, Y. Ouakrim, A. Fuentes, N. Hagemeister, R. Aissaoui, M. Pelletier, and J. De Guise. Knee osteoarthritis severity assessment using knee kinematic data classification. 20 :S97. ISSN 10634584. doi : 10.1016/j.joca.2012.02.102. URL <https://linkinghub.elsevier.com/retrieve/pii/S1063458412001744>.

Bibliographie

- [55] Nathaly Gaudreault, Neila Mezghani, Katia Turcot, Nicola Hagemeister, Karine Boivin, and Jacques A. de Guise. Effects of physiotherapy treatment on knee osteoarthritis gait data using principal component analysis. *Clinical Biomechanics (Bristol, Avon)*, 26(3) :284–291, March 2011. ISSN 1879-1271. doi : 10.1016/j.clinbiomech.2010.10.004.
- [56] Neila Mezghani, Sabine Husse, Karine Boivin, Katia Turcot, Rachid Aissaoui, Nicola Hagemeister, and Jacques A. de Guise. Automatic classification of asymptomatic and osteoarthritis knee gait patterns using kinematic data features and the nearest neighbor classifier. *IEEE transactions on biomedical engineering*, 55(3) :1230–1232, March 2008. ISSN 0018-9294. doi : 10.1109/TBME.2007.905388.
- [57] Automatic classification of 3D kinetic data of healthy and ACL deficient participants. In *ResearchGate*.
- [58] M.J. Beynon, L. Jones, and C.A. Holt. Classification of osteoarthritic and normal knee function using three-dimensional motion analysis and the Dempster-Shafer theory of evidence. *IEEE Transactions on Systems, Man, and Cybernetics - Part A : Systems and Humans*, 36(1) :173–186, January 2006. ISSN 1558-2426. doi : 10.1109/TSMCA.2006.859098.
- [59] Phases of the normal gait cycle | Download Scientific Diagram. https://www.researchgate.net/figure/Phases-of-the-normal-gait-cycle_fig3_309362425.
- [60] Figure 1. Directions of displacement. AP : Anterior-posterior direction... https://www.researchgate.net/figure/Directions-of-displacement-AP-Anterior-posterior-direction-forward-backward_fig6_225059668, .
- [61] Neila Mezghani, Imene Mechmeche, Amar Mitiche, Youssef Ouakrim, and Jacques A. de Guise. An analysis of 3D knee kinematic data complexity in knee osteoarthritis and asymptomatic controls. *PloS One*, 13(10) :e0202348, 2018. ISSN 1932-6203. doi : 10.1371/journal.pone.0202348.
- [62] Sébastien Lustig, Robert A. Magnussen, Laurence Cheze, and Philippe Neyret. The KneeKG system : A review of the literature. *Knee Surgery, Sports Traumatology, Arthroscopy*, 20(4) :633–638, April 2012. ISSN 1433-7347. doi : 10.1007/s00167-011-1867-4.
- [63] What is a Recommendation System? | Data Science | NVIDIA Glossary. <https://www.nvidia.com/en-us/glossary/recommendation-system/>, .

Bibliographie

- [64] Prem Melville and Vikas Sindhwani. Recommender Systems. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 829–838. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi : 10.1007/978-0-387-30164-8_705.
- [65] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports*, 519(1) :1–49, October 2012. ISSN 0370-1573. doi : 10.1016/j.physrep.2012.02.006.
- [66] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to Recommender Systems Handbook. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi : 10.1007/978-0-387-85820-3_1.
- [67] Peter Brusilovsky and Eva Millán. User Models for Adaptive Hypermedia and Adaptive Educational Systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web : Methods and Strategies of Web Personalization*, pages 3–53. Springer, Berlin, Heidelberg, 2007. ISBN 978-3-540-72079-9. doi : 10.1007/978-3-540-72079-9_1.
- [68] Tariq Mahmood and Francesco Ricci. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, pages 73–82, New York, NY, USA, June 2009. Association for Computing Machinery. ISBN 978-1-60558-486-7. doi : 10.1145/1557914.1557930.
- [69] Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3) :56–58, March 1997. ISSN 0001-0782. doi : 10.1145/245108.245121.
- [70] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems : A literature survey. *International Journal on Digital Libraries*, 17(4) :305–338, November 2016. ISSN 1432-1300. doi : 10.1007/s00799-015-0156-0.
- [71] Types of recommender system. https://www.researchgate.net/figure/Types-of-recommender-system_fig2_359335702.
- [72] Erion Çano and Maurizio Morisio. Hybrid Recommender Systems : A Systematic Literature Review. *Intelligent Data Analysis*, 21(6) :1487–1524, November 2017. ISSN 1088467X, 15714128. doi : 10.3233/IDA-163209.

Bibliographie

- [73] christine. Les systèmes de recommandation : une catégorisation, September 2018.
- [74] Luigi Palopoli, Domenico Rosaci, and Giuseppe M. L. Sarné. A Multi-tiered Recommender System Architecture for Supporting E-Commerce. In Giancarlo Fortino, Costin Badica, Michele Malgeri, and Rainer Unland, editors, *Intelligent Distributed Computing VI*, pages 71–81, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-32524-3. doi : 10.1007/978-3-642-32524-3_10.
- [75] R Dong and L Tokarchuk. Digging Friendship : Paper Recommendation in Social Network.
- [76] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. Facing the cold start problem in recommender systems. 41(4) :2065–2073. ISSN 09574174. doi : 10.1016/j.eswa.2013.09.005. URL <https://linkinghub.elsevier.com/retrieve/pii/S0957417413007240>.
- [77] Deepak Kumar Panda and Sanjog Ray. Approaches and algorithms to mitigate cold start problems in recommender systems : A systematic literature review. *Journal of Intelligent Information Systems*, 59(2) :341–366, October 2022. ISSN 1573-7675. doi : 10.1007/s10844-022-00698-5.
- [78] Rabaa Alabdulrahman and Herna Viktor. Catering for unique tastes : Targeting grey-sheep users recommender systems through one-class machine learning. *Expert Systems with Applications*, 166 :114061, March 2021. ISSN 0957-4174. doi : 10.1016/j.eswa.2020.114061.
- [79] Paul. Knowledge-Based Recommenders. URL <https://medium.com/@paul0/knowledge-based-rs-b4547b84108a>.
- [80] Rakhi Seth and Aakanksha Sharaff. A Comparative Overview of Hybrid Recommender Systems : Review, Challenges, and Prospects. In Rohit Raja, Kapil Kumar Nagwanshi, Sandeep Kumar, and K. Ramya Laxmi, editors, *Data Mining and Machine Learning Applications*, pages 57–98. Wiley, 1 edition, February 2022. ISBN 978-1-119-79178-2 978-1-119-79252-9. doi : 10.1002/9781119792529.ch3.
- [81] Marko Balabanović and Yoav Shoham. Fab : Content-based, collaborative recommendation. *Commun. ACM*, 40(3) :66–72, March 1997. ISSN 0001-0782. doi : 10.1145/245108.245124.

Bibliographie

- [82] Lili Wu, Sam Shah, Sean Choi, Mitul Tiwari, and C. Posse. The Browsemaps : Collaborative Filtering at LinkedIn. In *RSWeb@RecSys*, 2014.
- [83] Sciforce. Inside recommendations : How a recommender system recommends, February 2020.
- [84] Thi Ngoc Trang Tran, Alexander Felfernig, Christoph Trattner, and Andreas Holzinger. Recommender systems in the healthcare domain : State-of-the-art and research issues. *Journal of Intelligent Information Systems*, 57(1) : 171–201, August 2021. ISSN 1573-7675. doi : 10.1007/s10844-020-00633-6.
- [85] Benjamin Stark, Constanze Knahl, Mert Aydin, and Karim Elish. A Literature Review on Medicine Recommender Systems. 10(8). ISSN 21565570, 2158107X. doi : 10.14569/IJACSA.2019.0100802. URL <http://thesai.org/Publications/ViewPaper?Volume=10&Issue=8&Code=IJACSA&SerialNo=2>.
- [86] IBM Watson. <https://www.ibm.com/watson>, July 2024.
- [87] Sai kiran Adlluru. The Rise and Fall of IBM Watson : Lessons from AI’s Journey in Healthcare, August 2024.
- [88] Rajani Shankar Sadasivam, Sarah L. Cutrona, Rebecca L. Kinney, Benjamin M. Marlin, Kathleen M. Mazor, Stephenie C. Lemon, and Thomas K. Houston. Collective-Intelligence Recommender Systems : Advancing Computer Tailoring for Health Behavior Change Into the 21st Century. *Journal of Medical Internet Research*, 18(3) :e4448, March 2016. doi : 10.2196/jmir.4448.
- [89] Zheyu Wang, Haoce Huang, Liping Cui, Juan Chen, Jiye An, Huilong Duan, Huiqing Ge, and Ning Deng. Using Natural Language Processing Techniques to Provide Personalized Educational Materials for Chronic Disease Patients in China : Development and Assessment of a Knowledge-Based Health Recommender System. 8(4) :e17642. doi : 10.2196/17642. URL <https://medinform.jmir.org/2020/4/e17642>.
- [90] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. 17 (6) :734–749. ISSN 1041-4347. doi : 10.1109/TKDE.2005.99. URL <http://ieeexplore.ieee.org/document/1423975/>.
- [91] Yao Cai, Fei Yu, Manish Kumar, Roderick Gladney, and Javed Mostafa. Health Recommender Systems Development, Usage, and Evaluation from 2010 to 2022 : A Scoping Review. *International Journal of Environmental Research*

Bibliographie

- and Public Health*, 19(22) :15115, November 2022. ISSN 1660-4601. doi : 10.3390/ijerph192215115.
- [92] Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20 : 29–50, May 2016. ISSN 1574-0137. doi : 10.1016/j.cosrev.2016.05.002.
- [93] Michael D. Ekstrand. Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human-Computer Interaction*, 4(2) :81–173, 2011. ISSN 1551-3955, 1551-3963. doi : 10.1561/11000000009.
- [94] Charu C. Aggarwal. *Recommender Systems*. Springer International Publishing, Cham, 2016. ISBN 978-3-319-29657-9 978-3-319-29659-3. doi : 10.1007/978-3-319-29659-3.
- [95] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative Filtering Recommender Systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321, pages 291–324. Springer Berlin Heidelberg. ISBN 978-3-540-72078-2. doi : 10.1007/978-3-540-72079-9_9. URL http://link.springer.com/10.1007/978-3-540-72079-9_9.
- [96] What is collaborative filtering ? | IBM. <https://www.ibm.com/think/topics/collaborative-filtering>, March 2024.
- [97] Example of user-item ratings matrix in a movie recommendation scenario. | Download Scientific Diagram. https://www.researchgate.net/figure/Example-of-user-item-ratings-matrix-in-a-movie-recommendation-scenario_fig2_300646445.
- [98] Ms. Tejashri Sharad Phalle and Prof. Shivendu Bhushan. Content Based Filtering And Collaborative Filtering : A Comparative Study. pages 96–100. doi : 10.53555/jaz.v45iS4.4158. URL <https://jazindia.com/index.php/jaz/article/view/4158>.
- [99] The Significance of Collaborative Filtering in Recommender Systems. <https://www.cloudthat.com/resources/blog/the-significance-of-collaborative-filtering-in-recommender-systems>.
- [100] What Is Collaborative Filtering : A Simple Introduction | Built In. <https://builtin.com/data-science/collaborative-filtering-recommender-system>, .

Bibliographie

- [101] User-based and Item-based Collaborative Filtering | Download Scientific Diagram. https://www.researchgate.net/figure/User-based-and-Item-based-Collaborative-Filtering_fig2_366902172.
- [102] Mukund Deshpande and George Karypis. Item-based top- N recommendation algorithms. *ACM Transactions on Information Systems*, 22(1) :143–177, January 2004. ISSN 1046-8188, 1558-2868. doi : 10.1145/963770.963776.
- [103] Wenbin Yue, Zidong Wang, Bo Tian, Mark Pook, and Xiaohui Liu. A Hybrid Model- and Memory-Based Collaborative Filtering Algorithm for Baseline Data Prediction of Friedreich’s Ataxia Patients. *IEEE Transactions on Industrial Informatics*, 17(2) :1428–1437, February 2021. ISSN 1551-3203, 1941-0050. doi : 10.1109/TII.2020.2984540.
- [104] Hael Al-Bashiri, Mansoor Abdullateef Abdulgabber, Awanis Romli, and Fadhl Hujainah. Collaborative Filtering Recommender System : Overview and Challenges. *Advanced Science Letters*, 23(9) :9045–9049, September 2017. ISSN 1936-6612. doi : 10.1166/asl.2017.10020.
- [105] Muffaddal Qutbuddin. Comprehensive Guide on Item Based Recommendation Systems, February 2024.
- [106] Norm Matloff. Collaborative Filtering in Recommender Systems : A Short Introduction.
- [107] Abhaya Kumar Sahoo, Chittaranjan Pradhan, Rabindra Kumar Barik, and Harishchandra Dubey. DeepReco : Deep Learning Based Health Recommender System Using Collaborative Filtering. *Computation*, 7(2) :25, May 2019. ISSN 2079-3197. doi : 10.3390/computation7020025.
- [108] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Matrix Factorization Model in Collaborative Filtering Algorithms : A Survey. *Procedia Computer Science*, 49 :136–146, 2015. ISSN 18770509. doi : 10.1016/j.procs.2015.04.237.
- [109] Implementing Matrix Factorization Technique for Recommender Systems from scratch | by Sadak Vali | Medium. <https://medium.com/@rebirth4vali/implementing-matrix-factorization-technique-for-recommender-systems-from-scratch-7828c9166d3c>.
- [110] Rishabh Ahuja, Arun Solanki, and Anand Nayyar. Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor. In *2019 9th*

Bibliographie

- International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 263–268, Noida, India, January 2019. IEEE. ISBN 978-1-5386-5933-5. doi : 10.1109/CONFLUENCE.2019.8776969.
- [111] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295, Hong Kong Hong Kong, April 2001. ACM. ISBN 978-1-58113-348-6. doi : 10.1145/371920.372071.
- [112] Clustering basics and a demonstration in clustering infrastructure pathways – Water Programming : A Collaborative Research Blog. <https://waterprogramming.wordpress.com/2022/03/16/clustering-basics-and-a-demonstration-in-clustering-infrastructure-pathways/>.
- [113] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural Collaborative Filtering, August 2017.
- [114] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, Perth Australia, April 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4913-0. doi : 10.1145/3038912.3052569.
- [115] ZEMIM. Deep Neural Network (DNN) Explained, April 2024.
- [116] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph Neural Networks in Recommender Systems : A Survey, April 2022.
- [117] Sergios Karagiannakos. Graph Neural Networks - An overview. https://theai-summer.com/Graph_Neural_Networks/, February 2020.
- [118] Carmel Wenga, Majirus Fansi, Sébastien Chabrier, Jean-Martial Mari, and Alban Gabillon. A Comprehensive Review on Non-Neural Networks Collaborative Filtering Recommendation Systems, 2021.
- [119] Wenbin Wu, Zhanyong Qi, Jiawei Tian, Bixi Wang, Minyi Tang, and Xuan Liu. An Enhanced Latent Factor Recommendation Approach for Sparse Datasets of E-Commerce Platforms. *Systems*, 13(5) :372, May 2025. ISSN 2079-8954. doi : 10.3390/systems13050372.

Bibliographie

- [120] Aleksandra Pawlicka, Marek Pawlicki, Rafał Kozik, and Ryszard S. Choraś. A Systematic Review of Recommender Systems and Their Applications in Cybersecurity. 21(15) :5248. ISSN 1424-8220. doi : 10.3390/s21155248. URL <https://www.mdpi.com/1424-8220/21/15/5248>.
- [121] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph Neural Networks in Recommender Systems : A Survey. *ACM Computing Surveys*, 55 (5) :1–37, May 2023. ISSN 0360-0300, 1557-7341. doi : 10.1145/3535101.
- [122] Rianne van den Berg, Thomas N. Kipf, and Max Welling. Graph Convolutional Matrix Completion, October 2017.
- [123] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, July 2018. doi : 10.1145/3219819.3219890.
- [124] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. LightGCN : Simplifying and Powering Graph Convolution Network for Recommendation, July 2020.
- [125] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174, Paris France, July 2019. ACM. ISBN 978-1-4503-6172-9. doi : 10.1145/3331184.3331267.
- [126] Annielle Mendes Brito Da Silva, Natiele Carla Da Silva Ferreira, Luiza Amara Maciel Braga, Fabio Batista Mota, Victor Maricato, and Luiz Anastacio Alves. Graph Neural Networks : A Bibliometric Mapping of the Research Landscape and Applications. *Information*, 15(10) :626, October 2024. ISSN 2078-2489. doi : 10.3390/info15100626.
- [127] Applications of Graph Data Structure | GeeksforGeeks. <https://www.geeksforgeeks.org/applications-of-graph-data-structure/>.
- [128] Représentation graphique de (a.) un réseau non-dirigé et... https://www.researchgate.net/figure/representation-graphique-de-a-un-reseau-non-dirige-et-non-pondere-a-trois-noeuds-et_fig4_277587050.

Bibliographie

- [129] Variant of graphs found in nature and science. Directed VS Undirected... https://www.researchgate.net/figure/Variant-of-graphs-found-in-nature-and-science-Directed-VS-Undirected-Graphs-Static-VS_fig1_365692581.
- [130] What is a Hypergraph ? – Be on the Right Side of Change, April 2022.
- [131] What is a Graph Neural Network | IBM. <https://www.ibm.com/think/topics/graph-neural-network>, February 2025.
- [132] James H. Tanis, Chris Giannella, and Adrian V. Mariano. Introduction to Graph Neural Networks : A Starting Point for Machine Learning Engineers, 2024.
- [133] DiploDoc. Les réseaux de neurones en graphes (GNN) permettent d'étudier les interactions entre les..., May 2021.
- [134] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer : Generating Explanations for Graph Neural Networks. *Advances in Neural Information Processing Systems*, 32 :9240–9251, December 2019. ISSN 1049-5258.
- [135] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model. page 19, 2003.
- [136] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, 2013.
- [137] Embeddings : Meaning, Examples and How To Compute. <https://arize.com/blog-course/embeddings-meaning-examples-and-how-to-compute/>.
- [138] Graph neural network. *Wikipedia*, May 2025.
- [139] Nico Klingler. Graph Neural Networks (GNNs) - Comprehensive Guide, February 2024.
- [140] Will Hamilton, Zhitaoy Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [141] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks, September 2017.

Bibliographie

- [142] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks, February 2018.
- [143] Amal Menzli. Graph Neural Network and Some of GNN Applications. URL <https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications>.
- [144] Graph Neural Network – Message Passing (GCN) – 1.1 – Denken, November 2022.
- [145] What are Graph Neural Networks? <https://www.geeksforgeeks.org/what-are-graph-neural-networks/>, 2023.
- [146] Rudina Seseeri. AI Atlas #18 : Graph Neural Networks (GNNs). URL <https://glasswing.vc/blog/ai-atlas-18-graph-neural-networks-gnns/>.
- [147] Figure 1. An example User-Item Bipartite Graph with ratings. https://www.researchgate.net/figure/An-example-User-Item-Bipartite-Graph-with-ratings_fig1_361591389, .
- [148] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural Collaborative Filtering vs. Matrix Factorization Revisited, 2020.
- [149] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. 2019. doi : 10.48550/ARXIV.1901.00596.
- [150] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph Neural Networks : A Review of Methods and Applications, 2018.
- [151] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, 2016.
- [152] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks. URL <https://arxiv.org/abs/1511.05493>.
- [153] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph Neural Networks. URL <https://arxiv.org/abs/1809.09401>.
- [154] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. URL <https://arxiv.org/abs/1710.10903>.

Bibliographie

- [155] Robin De Croon, Leen Van Houdt, Nyi Nyi Htun, Gregor Štiglic, Vero Vanden Abeele, and Katrien Verbert. Health Recommender Systems : Systematic Review. *Journal of Medical Internet Research*, 23(6) :e18035, June 2021. ISSN 1438-8871. doi : 10.2196/18035.
- [156] P. Chinnasamy, Wing-Keung Wong, A. Ambeth Raja, Osamah Ibrahim Khalaf, Ajmeera Kiran, and J. Chinna Babu. Health Recommendation System using Deep Learning-based Collaborative Filtering. *Heliyon*, 9(12) :e22844, December 2023. ISSN 2405-8440. doi : 10.1016/j.heliyon.2023.e22844.
- [157] Su-Cheng Haw, Jayapradha Jayaram, Elham Abdulwahab Anaam, and Heru Agus Santoso. Exploring Recommender Systems in the Healthcare : A Review on Methods, Applications and Evaluations. *International Journal on Robotics, Automation and Sciences*, 6(2) :6–15, September 2024. ISSN 2682-860X. doi : 10.33093/ijoras.2024.6.2.2.
- [158] Arash Mahyari and Peter Pirolli. Physical Exercise Recommendation and Success Prediction Using Interconnected Recurrent Neural Networks. In *2021 IEEE International Conference on Digital Health (ICDH)*, pages 148–153, September 2021. doi : 10.1109/ICDH52753.2021.00027.
- [159] Rob Edge and Kelly Farrah. *Exercise for the Management of Knee Osteoarthritis : A Review of Clinical Effectiveness*. CADTH Rapid Response Reports. Canadian Agency for Drugs and Technologies in Health, Ottawa (ON), 2017.
- [160] Waidah Ismail, Ismail Ahmed Al-Qasem Al-Hadi, Crina Grosan, and Rimuljo Hendradi. Improving patient rehabilitation performance in exercise games using collaborative filtering approach. *PeerJ Computer Science*, 7 :e599, July 2021. ISSN 2376-5992. doi : 10.7717/peerj-cs.599.
- [161] Siyeon Kim, Michael R. Kosorok, Liubov Arbeeva, Todd A. Schwartz, Leigh F. Callahan, Yvonne M. Golightly, Amanda E. Nelson, and Kelli D. Allen. Precision Medicine–Based Machine Learning Analyses to Explore Optimal Exercise Therapies for Individuals With Knee Osteoarthritis : Random Forest-Informed Tree-Based Learning. *The Journal of Rheumatology*, 50(10) :1341–1345, October 2023. ISSN 0315-162X, 1499-2752. doi : 10.3899/jrheum.2022-1039.
- [162] C. Kokkotis, S. Moustakidis, E. Papageorgiou, G. Giakas, and D. E. Tsapopoulos. Machine learning in knee osteoarthritis : A review. *Osteoarthritis and Cartilage Open*, 2(3) :100069, September 2020. ISSN 2665-9131. doi : 10.1016/j.ocarto.2020.100069.

Bibliographie

- [163] Valerie Dieter, Pia Janssen, and Inga Krauss. Efficacy of the mHealth-Based Exercise Intervention re.flex for Patients With Knee Osteoarthritis : Pilot Randomized Controlled Trial. *JMIR mHealth and uHealth*, 12 :e54356, September 2024. ISSN 2291-5222. doi : 10.2196/54356.
- [164] Graph machine learning with missing node features. URL https://blog.x.com/engineering/en_us/topics/insights/2022/graph-machine-learning-with-missing-node-features.
- [165] Sergey Ioffe and Christian Szegedy. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. URL <https://arxiv.org/abs/1502.03167>.
- [166] Fonction sigmoïde - <https://www.bibmath.net/>. URL <https://www.bibmath.net/dico/index.php?action=affiche&quoi=./s/sigmoide.html>.
- [167] Crypto1. Gradient Descent Algorithm : How Does it Work in Machine Learning? URL <https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/>.
- [168] Priyansh Soni. ROC-AUC Analysis - A Deep Dive. URL <https://www.blog.trainindata.com/auc-roc-analysis/>.
- [169] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.

Annexe A

Liste des champs et explication

1.1 Caractéristiques orthopédiques lors de la prescription

Certaines variables de la base de données décrivent les dispositifs orthopédiques portés par les patients au moment des tests ou lors de la prescription d'exercices. Ces informations permettent de mieux contextualiser les recommandations formulées. Les variables concernées sont les suivantes :

- `orthese_stabilisatrice`
- `orthese_plantaire`
- `chaussures_orthopédiques`
- `1ere_ligne_orthese_stab`
- `2eme_ligne_orthese_stab`

Ces champs indiquent la présence ou l'absence d'un dispositif orthopédique prescrit dans le cadre de la rééducation fonctionnelle. Les valeurs possibles sont :

- 1 : dispositif utilisé,
- 0 : dispositif non utilisé,
- NaN (vide) : information non renseignée ou non documentée.

Une autre variable, `si_oui_genouillere`, indique l'identifiant ou le type de genouillère portée par le patient, si applicable.

Il est important de noter que l'imputation systématique de la valeur 0 pour les champs manquants pourrait induire une mauvaise interprétation. En effet, une valeur absente ne signifie pas nécessairement que le patient ne portait pas de dispositif. Il est donc recommandé :

Annexe A. Liste des champs et explication

- d'exclure les patients présentant des valeurs manquantes dans ces variables si celles-ci sont utilisées dans l'analyse,
- ou de les ignorer si ces champs ne sont pas exploités dans le modèle.

1.2 Tests fonctionnels liés aux effets des exercices

Certains champs de la base de données renseignent les résultats de tests fonctionnels évaluant l'état biomécanique du genou et des membres inférieurs. Ces tests permettent de mesurer l'effet potentiel des exercices prescrits, en fournissant des indicateurs de performance, de mobilité articulaire, de force musculaire, de souplesse et d'équilibre.

Les principales variables sont :

- `flow_test` : test de performance globale, souvent associé à l'endurance fonctionnelle.
- `diff_circ_interligne` : écart de circonférence articulaire, indicateur de présence d'œdème ou d'inflammation.
- `amp_flexion_passive` : amplitude passive de flexion du genou.
- `amp_extension_passive` : amplitude passive d'extension du genou.
- `force_flexion_genou` : force musculaire en flexion du genou.
- `force_extension_genou` : force musculaire en extension du genou.
- `force_flexion_hanche` : force musculaire en flexion de la hanche.
- `force_extension_hanche` : force musculaire en extension de la hanche.
- `force_abd_hanche` : force d'abduction de la hanche.
- `force_add_hanche` : force d'adduction de la hanche.
- `force_rot_INT_hanche` : force en rotation interne de la hanche.
- `force_rot_EXT_hanche` : force en rotation externe de la hanche.
- `force_plantaire_cheville` : force de flexion plantaire de la cheville.
- `force_dorsale_cheville` : force de flexion dorsale de la cheville.
- `flexibilite_ischio` : souplesse des ischio-jambiers.
- `flexibilite_quad_psoas` : souplesse du quadriceps et du psoas.
- `flexibilite_BIT` : souplesse du biceps fémoral (BIT).
- `flexibilite_fessiers_pyramidal` : souplesse des fessiers et du pyramidal.
- `equilibre_unipodal` : évaluation de l'équilibre sur un seul appui.

Annexe A. Liste des champs et explication

- 30s_CST : Chair Stand Test, mesurant le nombre de levées en 30 secondes, indicateur de force fonctionnelle des membres inférieurs.

Ces mesures offrent une évaluation multidimensionnelle de l'état fonctionnel du patient et peuvent être utilisées comme cibles cliniques, indicateurs de progression, ou encore comme variables explicatives dans le modèle de recommandation.

Annexe B

Résultats des Tests sur les Hyperparamètres d'Entraînement

Taux d'apprentissage

L'analyse de l'impact du taux d'apprentissage, illustrée par la [figure 2.1](#), révèle une courbe de performance en cloche mettant en évidence trois régimes d'optimisation distincts. L'optimum est atteint pour un taux d'apprentissage $lr = 0,005$, avec un score RPS de 81,4 %, soulignant l'importance critique de ce paramètre pour assurer la bonne convergence du modèle. Une zone de sous-apprentissage est observée pour des valeurs faibles ($lr \leq 0,001$), où la convergence devient trop lente et le modèle n'atteint pas son potentiel optimal. À l'inverse, une zone d'instabilité apparaît pour des valeurs élevées ($lr \geq 0,01$), caractérisée par une dégradation des performances. La zone optimale centrée autour de $lr = 0,005$ assure un équilibre entre vitesse de convergence et stabilité d'entraînement. Cette courbe illustre le compromis classique biais-variance : un taux d'apprentissage trop élevé entraîne une instabilité numérique, tandis qu'un taux trop faible limite l'exploration efficace de l'espace des solutions. L'optimiseur Adam couplé à $lr = 0,005$ constitue ainsi la configuration la plus adaptée pour notre problème de recommandation.

Patience

L'analyse de l'impact du paramètre de `patience`, illustrée dans la [figure 2.2](#), montre une stabilité autour de l'optimum, avec une valeur de patience fixée à 15 époques permettant d'atteindre un score RPS de 81,1%. Une patience trop faible (5 époques) conduit à un arrêt prématuré de l'entraînement avant convergence complète, limitant le RPS à 79%. En revanche, une zone optimale est identifiée pour des valeurs comprises entre 10 et 20 époques, où la performance reste stable au-dessus

Annexe B. Résultats des Tests sur les Hyperparamètres d'Entraînement

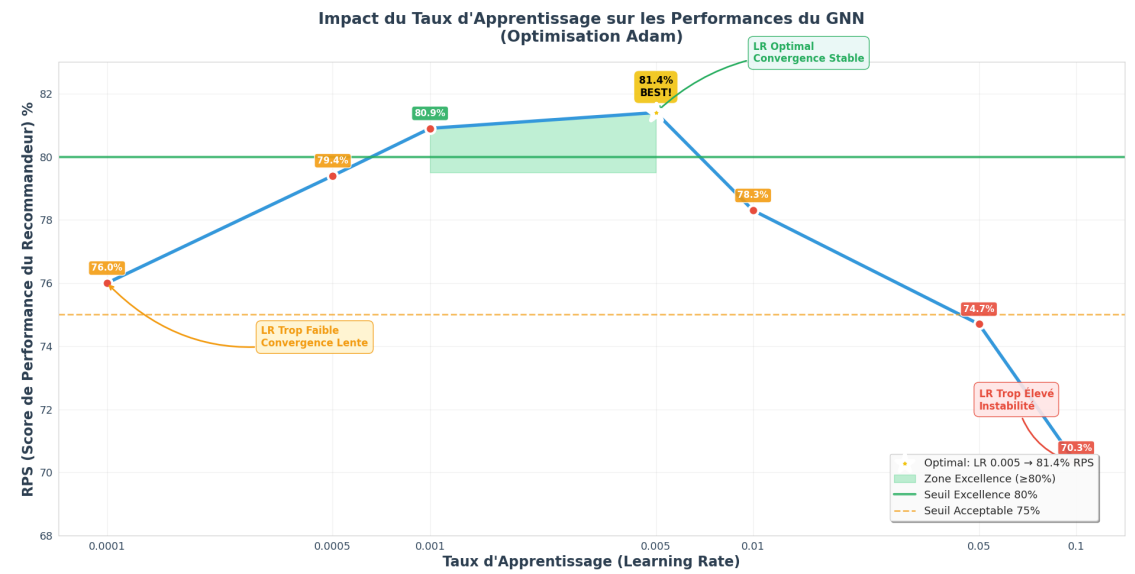


FIGURE 2.1. Évolution du score RPS en fonction du taux d'apprentissage.

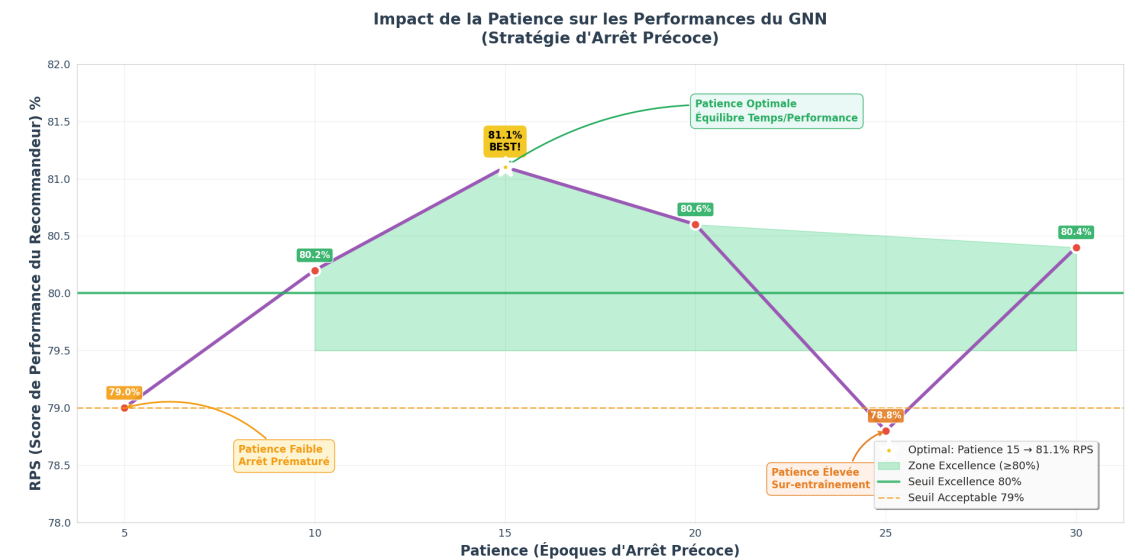


FIGURE 2.2. Évolution du score RPS en fonction de patience.

Annexe B. Résultats des Tests sur les Hyperparamètres d’Entraînement

de 80%. Pour des valeurs élevées (`patience` ≥ 25), le risque de sur-entraînement s’accroît sans gain significatif de performance. La variation limitée de 2,3% du RPS sur l’ensemble des tests indique une convergence cohérente et robuste du modèle. Ainsi, une patience fixée à 15 époques constitue le meilleur compromis entre temps d’entraînement raisonnable et performance finale, évitant le sur-apprentissage tout en garantissant une optimisation efficace.

Comparaison entre la Focal Loss et l’Entropie Croisée Binaire

L’analyse comparative des métriques d’entraînement met en évidence les avantages et limites de la Focal Loss par rapport à la fonction d’entropie croisée binaire standard dans le cadre de notre système de recommandation. Comme illustré dans la [figure 2.4](#) et la [figure 2.3](#), l’utilisation de la Focal Loss conduit à une convergence nettement plus rapide, avec une perte finale de 0,004 contre 0,062 pour la perte standard, soit une amélioration relative de 93 %. Cette convergence accélérée s’explique par la capacité intrinsèque de la Focal Loss à accorder une pondération plus importante aux exemples difficiles à classer, ce qui est particulièrement pertinent dans notre contexte où les interactions positives (user-item) sont rares par rapport à l’ensemble des combinaisons possibles. De plus, l’analyse des scores RPS finaux confirme l’intérêt de cette approche, la Focal Loss permettant d’atteindre un score RPS supérieur à celui obtenu avec l’entropie croisée binaire, justifiant son adoption dans la configuration finale du modèle.

Paramètre γ de la Focal Loss

L’analyse de l’impact du paramètre γ de la *Focal Loss*, présentée dans la [figure 2.5](#), révèle une courbe en cloche marquée, avec un optimum net atteint pour $\gamma = 3,0$, permettant d’obtenir un score RPS maximal de 82,5%. Ce paramètre joue un rôle essentiel en modulant l’intensité de focalisation du modèle sur les exemples difficiles à classer. Pour des valeurs faibles ($\gamma \leq 1,5$), la focalisation est insuffisante, amenant le modèle à traiter tous les exemples de manière quasi uniforme, ce qui limite sa capacité à corriger les erreurs sur les cas rares ou ambigus. La zone optimale est observée pour des valeurs de γ comprises entre 2,5 et 3,0, garantissant une attention équilibrée sur les échantillons complexes sans pour autant négliger les exemples faciles. En revanche, une valeur trop élevée ($\gamma \geq 4,0$) entraîne une sur-focalisation qui pousse le modèle à ignorer les interactions simples au détriment de sa capacité de généralisation. La variation notable de 5,2% du score RPS sur cet intervalle met en évidence l’importance cruciale d’un bon réglage de γ .

Annexe B. Résultats des Tests sur les Hyperparamètres d'Entraînement

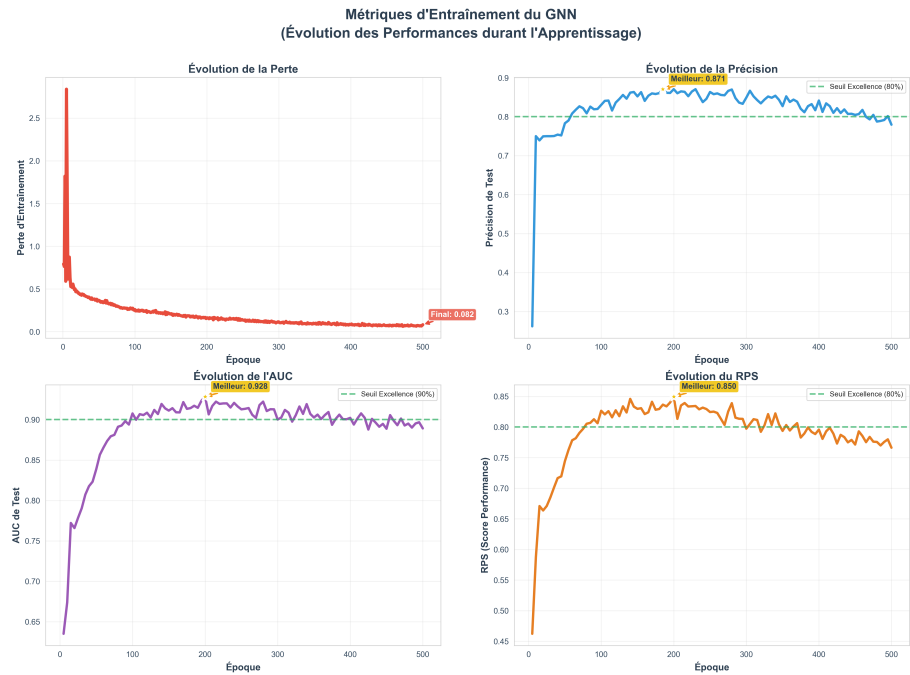


FIGURE 2.3. L'entropie croisée binaire.

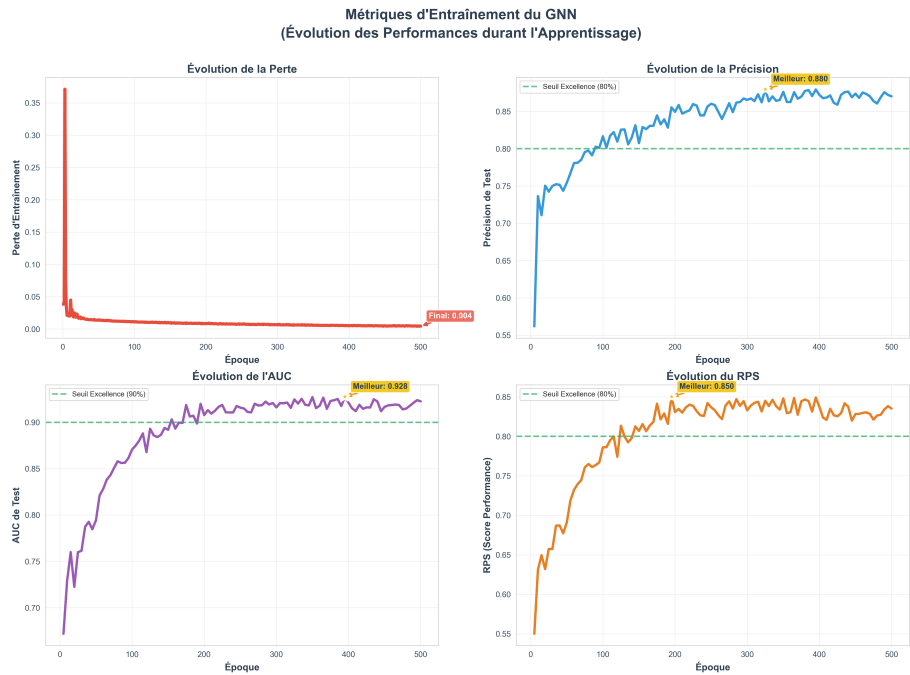


FIGURE 2.4. la Focal Loss

Annexe B. Résultats des Tests sur les Hyperparamètres d'Entraînement

Paramètre α de la Focal Loss

L'analyse du paramètre α de la *Focal Loss*, illustrée dans la [figure 2.6](#), met en évidence un pic de performance significatif pour $\alpha = 0,25$, atteignant un score RPS de 83,1%. Ce paramètre permet de contrôler l'équilibrage entre les classes positives et négatives dans un contexte de données déséquilibrées. Pour des valeurs faibles ($\alpha \leq 0,15$), on observe une sous-pondération des classes positives, conduisant à un biais du modèle en faveur des prédictions négatives. La zone optimale est atteinte entre $\alpha = 0,20$ et $\alpha = 0,25$, assurant un équilibrage adéquat qui reflète fidèlement la distribution naturelle des données. À l'inverse, des valeurs élevées ($\alpha \geq 0,35$) induisent une sur-pondération des interactions positives, ce qui dégrade la précision globale en surcompensant la rareté des exemples positifs. La configuration optimale $\alpha = 0,25$ indique que les échantillons positifs (relations patient-exercice existantes) doivent recevoir une pondération relative de 25 % contre 75 % pour les exemples négatifs, ce qui correspond à la nature déséquilibrée inhérente aux jeux de données de recommandation biomédicale.

Ratio d'Échantillonnage Négatif

L'analyse de l'impact du ratio d'échantillonnage négatif, présentée dans la [figure 2.7](#), met en évidence une courbe de performance asymétrique avec un optimum marqué pour un ratio de 2, aboutissant à un score RPS de 82,7%. Ce paramètre est essentiel car il détermine le nombre d'échantillons négatifs générés pour chaque exemple positif dans le cadre de la stratégie de *negative sampling*. Un ratio trop faible (ratio = 1) entraîne une sous-représentation des exemples négatifs, insuffisante pour apprendre efficacement la frontière entre interactions et non-interactions, limitant ainsi le RPS à 77,0%. À l'opposé, un ratio trop élevé (≥ 5) introduit un excès d'exemples négatifs, générant du bruit et diluant le signal d'apprentissage, ce qui dégrade progressivement la performance jusqu'à 71,6% pour un ratio de 10. La zone optimale identifiée entre les ratios 2 et 3 assure un équilibre contrastif idéal, permettant au modèle de capturer la structure des non-préférences sans surcharger l'espace de recherche. Ce résultat confirme qu'un ratio de 2 :1 entre négatifs et positifs constitue le compromis le plus efficace pour notre tâche.

Annexe B. Résultats des Tests sur les Hyperparamètres d'Entraînement

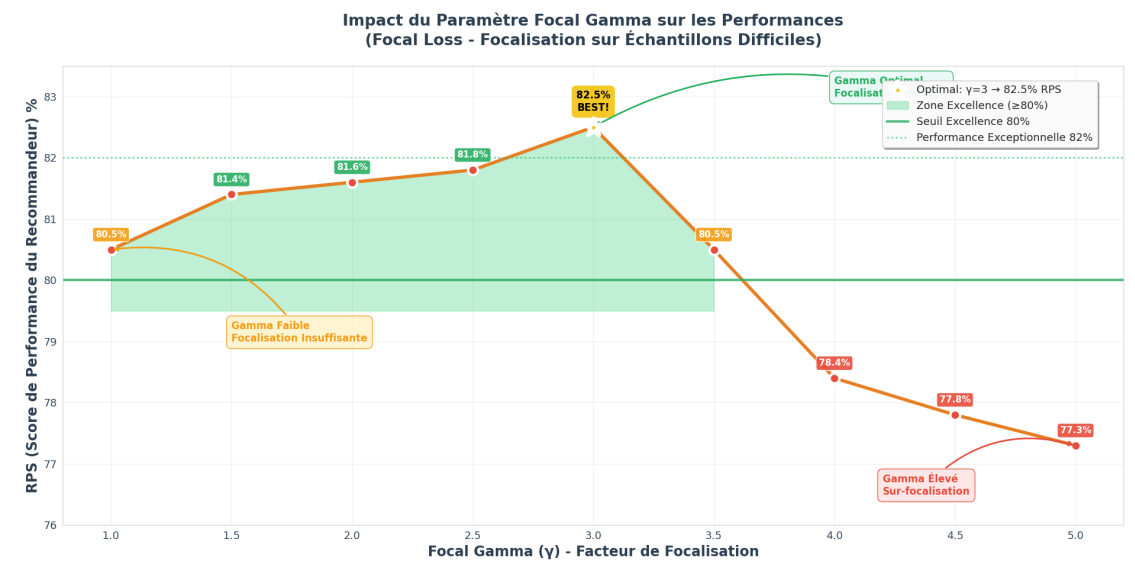


FIGURE 2.5. Évolution du score RPS en fonction du paramètre γ de la *Focal Loss*.

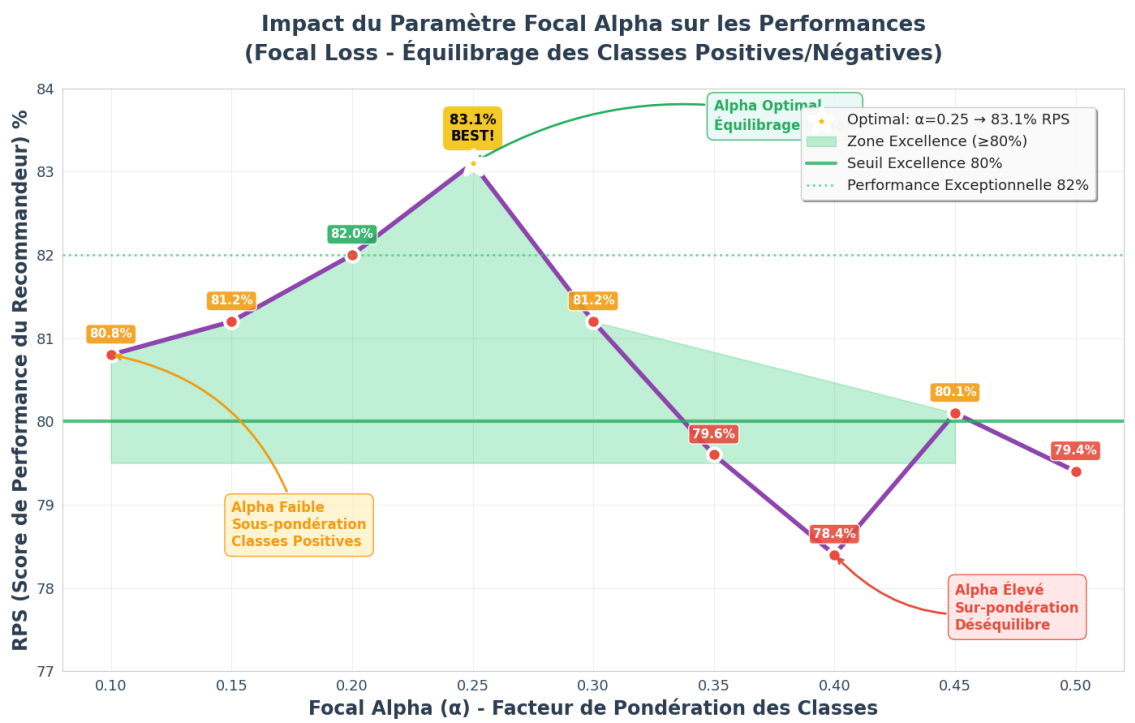


FIGURE 2.6. Évolution du score RPS en fonction du paramètre α de la *Focal Loss*.

Annexe B. Résultats des Tests sur les Hyperparamètres d'Entraînement

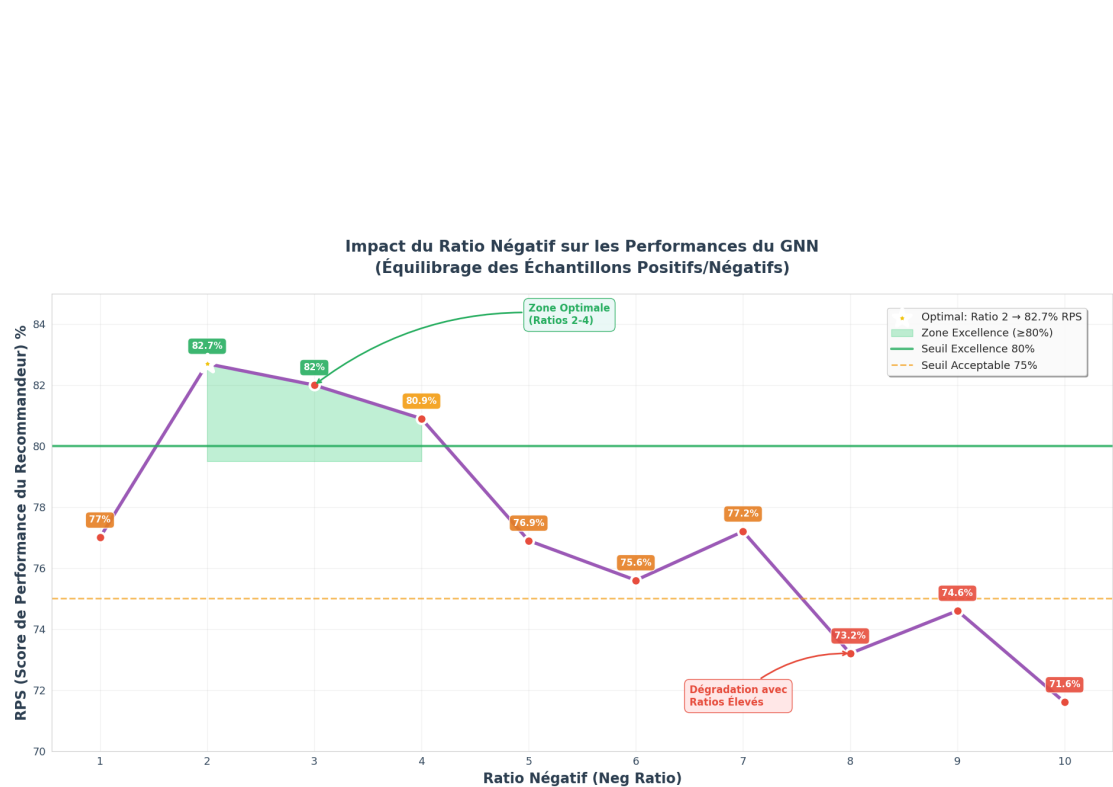


FIGURE 2.7. Évolution du score RPS en fonction du ratio d'échantillonnage négatif.