

Louvain School of Management

L'utilisation d'indices moyens pour évaluer la performance des systèmes de recommandation

Analyses et perspectives

Auteur : Achtergael Margot
Promoteur : Fouss François
Co-promotrice : Cotton Louise
Année académique 2022-2023
Master [120] Ingénieur de gestion à finalité spécialisé

Résumé

Les systèmes de recommandation étant de plus en plus présents dans notre société actuelle, il est davantage important qu'ils soient le plus performants en produisant des recommandations pertinentes à nos utilisateurs. Or, comme nous avons pu le voir dans la littérature, l'évaluation de leurs performances se base essentiellement sur des mesures utilisant des moyennes. Ainsi, tous les utilisateurs présents dans notre base de données sont traités de la même manière dans l'évaluation des modèles.

Ce mémoire, intitulé : « *L'utilisation d'indices moyens pour évaluer la performance des systèmes de recommandation : Analyses et perspectives* », va nous permettre de voir si le fait de privilégier certains utilisateurs possédant des critères spécifiques aura une quelconque influence sur le modèle à proposer à nos utilisateurs. En effet, nous sommes partis du postulat que certains utilisateurs, par exemple, les plus actifs, sont nos clients principaux et doivent donc avoir plus d'importance dans l'évaluation de nos systèmes.

Nous avons, dans ce but, exploré plus en profondeur le sujet des systèmes de recommandation dans la littérature. C'est dans celle-ci que la constatation des mesures d'évaluation basées sur les moyennes précédemment évoquée est survenue.

Nous avons ensuite expliqué la méthodologie utilisée afin de voir l'impact de l'intégration de l'importance des utilisateurs. Nous avons décidé d'utiliser trois types de systèmes de recommandations différents : un système de filtrage collaboratif user based, un système de filtrage collaboratif item based et un système de recommandation basé sur le contenu. Nous avons également utilisé trois cas de figure dans l'évaluation des modèles en plus de la version classique. La première accordait une importance différente en fonction du nombre de notations fourni par l'utilisateur. La deuxième prenait en compte la date de la dernière notation de l'utilisateur. Enfin, la dernière utilisait la fréquence de notation. Ces trois listes d'importance ont été intégrées dans les mesures d'évaluation, d'abord linéairement et ensuite quadratiquement afin d'accentuer davantage la différence entre les utilisateurs d'intérêts et les autres.

Nos analyses ont alors révélé que le classement des systèmes de recommandation en fonction de leurs performances n'était pas influencé par l'intégration de l'importance des utilisateurs. Il n'est donc pas nécessaire d'introduire une telle configuration dans l'évaluation de système de

recommandation si notre objectif est uniquement de déterminer quel système de recommandation proposer à nos utilisateurs. Néanmoins, cette méthodologie nous permettait d'obtenir des informations sur le profil des clients recevant les meilleures recommandations et donc d'obtenir des pistes de solution afin d'améliorer la pertinence de celles-ci.

Remerciements

Je tiens à exprimer ma sincère gratitude à toutes les personnes qui m'ont soutenue tout au long de l'élaboration de ce mémoire. Leur contribution a été inestimable et a grandement contribué à la réussite de ce travail.

Tout d'abord, je tiens à exprimer ma profonde reconnaissance envers Monsieur Fouss, mon promoteur, pour ses conseils éclairés et sa patience tout au long de ce mémoire. Son expertise a été d'une aide précieuse, et je suis reconnaissante d'avoir pu bénéficier de son encadrement.

Je souhaite également exprimer ma gratitude à Louise Cotton, ma co-promotrice, pour ses suggestions pertinentes et son implication dans mon travail de recherche. Sa collaboration et ses commentaires constructifs ont été d'une grande valeur pour moi.

Un merci spécial à mes parents pour leur amour, leur soutien indéfectible et leur encouragement tout au long de mes études. Leur confiance en moi a été un moteur essentiel dans la réalisation de ce mémoire, et je leur suis profondément reconnaissante pour tout ce qu'ils ont fait pour moi.

Je souhaite adresser mes remerciements à mes amis et à mon groupe de travail, Juliette, Clément, Théo, Lorenzo et Benjamin. Merci à Mathilde, pour son soutien, même à 1100 km. Leur amitié, leurs échanges enrichissants et leur soutien moral ont contribué à rendre cette expérience de recherche plus agréable et fructueuse.

Enfin, je tiens à exprimer ma reconnaissance envers toute autre personne qui, de près ou de loin, a contribué à la réalisation de ce mémoire.

Table des matières

1. Introduction	1
2. Revue de littérature	2
2.1. Introduction.....	2
2.2. Système de recommandation	2
2.2.1. Définition et objectifs d'un système de recommandation	2
2.2.2. Filtrage collaboratif	3
2.1.1. Système de recommandation basé sur le contenu	6
2.2.3. Mesures de similarité.....	7
2.3. Évaluation des systèmes de recommandation	10
2.3.1 Mesures basées sur la précision.....	11
2.3.2. Mesures d'évaluation « au-delà de la précision ».....	14
2.4. Bénéfices d'un système de recommandation.....	17
2.5. Conclusion	18
3. Méthodologie	21
3.1. Introduction.....	21
3.2. Logiciels et outils.....	21
3.3. Présentation de la base de données.....	22
3.4. Types de systèmes de recommandation utilisés	24
3.4.1. Extraction des données et création des matrices	25
3.4.2. Système de recommandation de filtrage collaboratif basé sur les utilisateurs	26
3.4.3. Système de recommandation de filtrage collaboratif basé sur les articles	29
3.4.4. Système de recommandation basé sur le contenu	31
3.5. Moyens d'évaluation	33
3.5.1. Évaluation « classique »	33
3.5.2. Évaluation utilisant le nombre de ratings	36
3.5.3. Évaluation utilisant la date du dernier rating.....	38
3.5.4. Évaluation utilisant la fréquence des ratings	40
3.6. Conclusion	42
4. Analyse des résultats.....	43

4.1.	Introduction.....	43
4.2.	Prises de décisions concernant les algorithmes	43
4.3.	Présentation des résultats.....	45
4.3.1.	nDCG.....	45
4.3.2.	Recall.....	47
4.3.3.	Précision	48
4.3.4.	F-measure	50
4.3.5.	Dissimilarité moyenne ou distance intra-liste (ILD).....	52
4.3.6.	« Unexpectancy » ou Distance utilisateur	53
4.4.	Conclusion des analyses	55
4.5.	Conclusion managériale	57
5.	Conclusion.....	59
	Limites et perspectives de recherche	60
6.	Bibliographie	62
6.1.	Articles périodiques	62
6.2.	Livres.....	65
6.3.	Sites internet	65
6.4.	Mémoires	66

Table des équations

(2.1) Prédiction basée sur les utilisateurs	4
(2.2) Prédiction basée sur les articles	5
(2.3) Distance euclidienne	7
(2.4) Similarité cosin sur l'ensemble des articles connus.....	8
(2.5) Similarité cosin sur les articles évalués par les deux utilisateurs	8
(2.6) Similarité Pearson.....	9
(2.7) Moyenne des notations	9
(2.8) Similarité Spearman	10
(2.9) Erreur absolue moyenne	11
(2.10) Racine de l'erreur quadratique moyenne.....	11
(2.11) Précision	12
(2.12) Recall.....	13
(2.13) F-measure	13
(2.14) Gain cumulé actualisé.....	13
(2.15) Gain cumulé actualisé normalisé	14
(2.16) Nouveauté grâce à la popularité : méthode 1	15
(2.17) Nouveauté grâce à la popularité : méthode 2	15
(2.18) Nouveauté grâce à l'unexpectancy	15
(2.19) Dissimilarité moyenne	16
(2.20) Diversité agrégée	17
(3.1) Rappel de la similarité cosin basée sur les utilisateurs	26
(3.2) Rappel de la prédiction basé sur les utilisateurs	27
(3.3) Rappel de la similarité cosin basée sur les articles	29
(3.4) Rappel de la prédiction basé sur les articles	30
(3.5) Rappel de la précision.....	34
(3.6) Rappel du recall	34
(3.7) Rappel du gain cumulé actualisé	34
(3.8) Rappel du gain cumulé actualisé normalisé	34
(3.9) Rappel de la nouveauté grâce à l'unexpectancy	34
(3.10) Rappel de la dissimilarité moyenne.....	35

Table des figures

Figure 2.1 Résultats possible lors de recommandation	12
Figure 3.1 Quelques lignes du fichier movies.csv	22
Figure 3.2 Quelques lignes du fichier ratings.csv	23
Figure 3.3 Quelques lignes du fichier tags.csv.....	23
Figure 3.4 Quelques lignes du fichier links.csv	23
Figure 3.5 Quelques lignes du fichier ratings_small.csv	24
Figure 3.6 Quelques lignes du fichier movies_metadata.csv	24
Figure 3.7 Matrice de notation explicite non binaire (à gauche) et matrice de notation implicite binaire (à droite).....	25
Figure 3.8 Exemples de films dans la première base de données	25
Figure 3.9 Matrice des genres et des années correspondants	26
Figure 3.10 Matrice de similarité cosinus entre utilisateurs.....	27
Figure 3.11 Matrice des prédictions basée sur les utilisateurs	28
Figure 3.12 Liste de recommandations sur base des utilisateurs	28
Figure 3.13 Matrice de similarité cosinus entre articles.....	29
Figure 3.14 Matrice des prédictions basée sur les articles	30
Figure 3.15 Liste de recommandations sur base des articles	31
Figure 3.16 Matrice de similarité entre films	31
Figure 3.17 Matrice des scores par film et utilisateur basés sur le contenu	32
Figure 3.18 Lignes de la matrice de similarité utilisées pour le calcul des scores de l'utilisateur 2.....	32
Figure 3.19 Liste de recommandations sur base du contenu.....	32
Figure 3.20 Matrice d'entrainement sur base de la figure 3.7	35
Figure 3.21 Matrice de test sur base de la figure 3.5	35
Figure 3.22 Recommandation pour la matrice d'entrainement	36
Figure 3.23 Liste d'importance de l'exemple selon le nombre de notations.....	37
Figure 3.24 Liste d'importance de l'exemple selon la date de leur dernière notation	39
Figure 3.25 Liste d'importance de l'exemple selon la fréquence de leurs notations	41
Figure 4.1 Test de Student de comparaison de moyennes de Recall	48
Figure 4.2 Test de Student de comparaison de moyennes de Précision.....	50
Figure 4.3 Test de Student de comparaison de moyennes de F-measure.....	52

Table des tableaux

Tableau 3.1 Calcul des mesures d'évaluation classiques.....	36
Tableau 3.2 Calcul des mesures d'évaluation incorporant l'importance des utilisateurs en fonction de leur nombre de notations	38
Tableau 3.3 Calcul des mesures d'évaluation incorporant l'importance des utilisateurs en fonction de la date de leur dernière notation	40
Tableau 3.4 Calcul des mesures d'évaluation incorporant l'importance des utilisateurs en fonction de la fréquence de notation	41
Tableau 4.1 Variation de k pour le système basé sur les utilisateurs	44
Tableau 4.2 Variation de k pour le système basé sur les articles	44
Tableau 4.3 Résultats pour les nDCGs calculés linéairement.....	45
Tableau 4.4 Résultats pour les nDCGs calculés quadratiquement.....	46
Tableau 4.5 Résultats pour les Recalls calculés linéairement	47
Tableau 4.6 Résultats pour les Recalls calculés quadratiquement	48
Tableau 4.7 Résultats pour les Précisions calculées linéairement	49
Tableau 4.8 Résultats pour les Précisions calculées quadratiquement.....	50
Tableau 4.9 Résultats pour les F-mesures calculées linéairement	51
Tableau 4.10 Résultats pour les F-mesures calculés quadratiquement	51
Tableau 4.11 Résultats pour les dissimilarités moyennes calculées linéairement	52
Tableau 4.12 Résultats pour les dissimilarités moyennes calculées quadratiquement.....	53
Tableau 4.13 Résultats pour les distances utilisateur calculées linéairement	54
Tableau 4.14 Résultats pour les distances utilisateurs calculées quadratiquement.....	55

Liste des annexes

Annexe 1 : Codes utilisés

Annexe 2 : Variation des métriques en fonction de la longueur de la liste de recommandation

Annexe 3 : Test de Student de comparaison de moyenne

Annexe 4 : Distribution des critères de conception des listes d'importance

1. Introduction

Dans un monde qui a connu une explosion des données, les systèmes de recommandation se sont imposés comme une solution incontournable pour faciliter la découverte et l'accès à des contenus pertinents et personnalisés. Ceux-ci ont alors connu une croissance fulgurante avec des applications dans de nombreux domaines tels que le commerce électronique, les réseaux sociaux, les musiques...

Il est alors devenu primordial pour les entreprises de fournir un système de recommandation de qualité. Or, nous allons voir que l'évaluation des différents systèmes de recommandation utilise des indices moyens considérant alors chaque utilisateur de la même manière. Découlant de cette constatation, ce mémoire tentera de répondre à la question de recherche suivante :

« L'utilisation d'indices moyens pour évaluer la performance des systèmes de recommandation : Analyses et perspectives ».

Nous allons dans un premier temps découvrir les grands concepts de la littérature concernant les systèmes de recommandation ainsi que les moyens de les évaluer.

Ensuite, nous détaillerons la méthodologie que nous utiliserons dans le but de répondre à notre question de recherche. Pour cela, nous essayerons d'attribuer un poids différent à certains utilisateurs afin que ceux-ci soient privilégiés dans le calcul de l'évaluation des systèmes de recommandation. Notre hypothèse est qu'avec cette discrimination, le classement de nos systèmes de recommandation évalués, même si produisant les mêmes recommandations, pourrait changer. Cela nous aiderait donc à déterminer quel système produit de meilleures recommandations à nos utilisateurs d'intérêt par rapport aux autres.

Enfin, nous analyserons les résultats obtenus et tenterons de les mettre en contexte afin d'en tirer des conclusions managériales.

2. Revue de littérature

2.1. Introduction

Dans cette revue de littérature, nous allons définir à l'aide de la littérature existante les concepts importants dont nous allons parler tout au long de ce mémoire. Nous commencerons par clarifier ce qu'est un système de recommandation ainsi que ses objectifs. Ensuite, nous nous pencherons sur les types de système de recommandation que nous allons utiliser dans ce travail, à savoir, le filtrage collaboratif et le système basé sur le contenu ainsi que les différentes mesures de similarité possibles. Une fois ces notions expliquées, nous nous concentrerons sur le moyen d'évaluer les systèmes de recommandation en les distinguant en deux catégories : les mesures basées sur la précision avec des mesures telles que le RMSE, le MAE ou le Recall et les autres mesures d'évaluation allant « au-delà de la précision » comme la nouveauté, la sérendipité ou la diversité. Nous finirons cette partie par les bénéfices que peut nous apporter l'utilisation d'un système de recommandation et une brève conclusion.

2.2. Système de recommandation

2.2.1. Définition et objectifs d'un système de recommandation

Dans un monde où les systèmes de recommandation n'existaient pas, plusieurs stratégies étaient utilisées pour prendre des décisions face à des choix différents. Peut-être que la solution la plus facile était de simplement suivre son instinct, alors que d'autres préféraient demander l'avis à des amis ou un expert. Mais ces processus sont énergivores et prennent du temps (Jannach et al., 2010). L'utilisation d'Internet comme outil de renseignement sur un produit ou service constituait également une option valable et répandue. Néanmoins, le nombre exponentiel d'informations disponibles sur Internet ainsi que le nombre d'utilisateurs grandissant rendent la tâche de trouver des informations pertinentes à un problème de plus en plus difficile (Isinkaye et al., 2015).

L'absence de système de recommandation pourrait résulter en une mauvaise prise de décision de la part d'une personne manquant d'expérience ou de connaissance sur un sujet (Khachane & Vaidya, 2017). Les systèmes de recommandations ont donc été conçus dans l'optique d'offrir des recommandations faciles d'accès et de grande qualité aux utilisateurs afin de pallier ce problème (Jannach et al., 2010). La diversité des informations et des utilisateurs du Web va venir souligner la potentielle nécessité d'un *service d'information personnalisé* qui sera alors

capable de s'adapter aux besoins et aux priorités d'individu ou de groupe d'individus (Ma & Uchyigit, 2008).

Un système de recommandation permet aux utilisateurs de fournir des recommandations qui sont ensuite agrégées et envoyées aux personnes concernées. La valeur de ce système peut être liée à sa capacité à agréger de nombreuses recommandations ou à sa capacité à mettre en correspondance efficacement les utilisateurs qui donnent des recommandations et ceux qui en cherchent (Burke, Felfernig, & Göker, 2011). En observant notre comportement en ligne, par exemple, il est possible d'établir un profil d'utilisateur qui prend en compte nos préférences. Ce profil peut ensuite être utilisé pour personnaliser un service d'information particulier en fonction de ces préférences (Ma et Uchyigit, 2008) ou procurer des recommandations basées sur nos préférences connues (Melville & Sindhvani, 2010).

Plusieurs types de système de recommandation existent. Parmi ceux-ci, nous pouvons retrouver les systèmes de recommandation basés sur le filtrage collaboratif, les systèmes de recommandation basés sur le contenu, sur la connaissance ou encore des systèmes hybrides. (Aggarwal, 2016). Nous nous attarderons sur les deux premiers cités ci-dessus dans le cadre de ce mémoire.

2.2.2. Filtrage collaboratif

Le filtrage collaboratif est le processus de filtrage ou d'évaluation d'éléments par le biais des opinions d'autres personnes (Schafer et al., 2007). Plus généralement, cette technique recommande des articles (ou *items*) que d'autres utilisateurs avec des goûts similaires ont aimés auparavant. La similarité des goûts des utilisateurs est calculée sur base de la similarité des évaluations précédentes des utilisateurs (Ricci, Rokach, & Shapira, 2022). Nous nous appuyons donc sur l'hypothèse que les utilisateurs qui avaient des préférences similaires dans le passé auront vraisemblablement des préférences similaires dans le futur (Sinha & Dhanalakshmi, 2019).

Cette catégorie de système de recommandation utilise les interactions *user-item*, telles que les évaluations ou le comportement d'achat (Aggarwal, 2016). Le terme *user* désigne l'utilisateur pour lequel la recommandation est faite tandis que *item* caractérise ce que le système de recommandation propose à l'utilisateur (Shah et al., 2017). Ces notations sont reprises dans une matrice appelée « matrice de notations » qui sera de longueur $m * n$ où m est le nombre de *users* et n le nombre d'*items* (Aggarwal, 2016).

Memory-based

Nous pouvons distinguer deux méthodes de filtrage collaboratif : Le memory-based et le model-based. Développons tout d'abord la première qui est donc basée sur la mémoire. Dans celle-ci les évaluations des combinaisons *users-items* sont prédites sur la base de leurs voisinages (Aggarwal, 2016). Les évaluations *user-item* stockées dans la mémoire sont utilisées directement pour prédire les classifications des nouveaux *items*. Cette classification peut également se faire de deux manières, la première est la recommandation basée sur l'utilisateur (le *user*) et la seconde sur l'article (l'*item*) (Shah et al., 2017).

User based

L'approche basée sur les *users* utilise des utilisateurs similaires à l'utilisateur cible pour lui faire des recommandations. Le principe de base est donc de déterminer les utilisateurs qui sont similaires à l'utilisateur cible (Aggarwal, 2016). Les recommandations sont données à un utilisateur sur la base de l'évaluation des *items* par d'autres utilisateurs d'un groupe partageant des préférences communes (Al-Turjman, 2020). Plusieurs techniques existent afin de nous aider à sélectionner les voisins d'intérêt. La technique top k sélectionne les k utilisateurs les plus similaires. La technique du seuil de corrélation définit un seuil et ne conserve que les utilisateurs dont les similarités dépassent ce seuil (Fkih, 2021). Les mesures de similarités nous permettant de sélectionner les voisins d'intérêt seront développées dans la suite de ce mémoire.

L'une des méthodes nous permettant d'obtenir des prédictions est la somme pondérée (Fkih, 2021).

$$Prediction(u, i) = \tilde{r}_{ui} = \frac{\sum_{v \in N_u^i} Sim_{uv} * r_{vi}}{\sum_{v \in N_u^i} |Sim_{uv}|} \quad (2.1)$$

où $\tilde{r}_{ui}$ est la prédiction pour le *user* u et pour l'*item* i ;
 r_{vi} est l'évaluation du *user* v et pour l'*item* i ;
 N_u^i est l'ensemble des voisins qui sont le plus similaire au *user* u et qui ont évalué l'*item* i ;
 Sim_{uv} est la similarité entre le *user* u et le *user* v .

Item based

La seconde approche se base sur les *items*, qui plutôt que de faire correspondre des utilisateurs similaires, fait correspondre les *items* évalués par un utilisateur à des *items* similaires (Melville & Sindhvani, 2010). L'algorithme utilise la matrice *user-item* pour calculer la similarité entre l'ensemble des *items* et l'*item* qui nous intéresse. Si nous utilisons le principe des tops k *items*, les k *items* ayant obtenu les scores de similarités les plus élevés seront sélectionnés. Les recommandations sont proposées à l'utilisateur en fonction de ceux-ci (Patel, Desai & Panchal, 2017).

De manière similaire au modèle basé sur les utilisateurs, nous pouvons sélectionner les plus proches voisins grâce aux techniques des tops k et du seuil de corrélation en nous concentrant sur les *items* et ensuite calculer les prédictions grâce à la somme pondérée de la manière suivante (Fkih, 2021).

$$Prediction(u, i) = \tilde{r}_{ui} = \frac{\sum_{j \in N_i^u} Sim_{ij} * r_{uj}}{\sum_{j \in N_i^u} |Sim_{ij}|} \quad (2.2)$$

où $\tilde{r}_{ui}$ est la prédiction pour le *user* u et pour l'*item* i ;

r_{uj} est l'évaluation du *user* u et pour l'*item* j ;

N_i^u est l'ensemble des *items* qui sont le plus similaire à l'*item* i et qui ont été évalué par le *user* u ;

Sim_{ij} est la similarité entre le *item* i et le *item* j .

Model based

La deuxième méthode est quant à elle basée sur le modèle. Ce genre de système de recommandation va utiliser les évaluations fournies par les utilisateurs afin d'entraîner le modèle qui sera ensuite utilisé pour prédire les évaluations des utilisateurs pour de nouveaux articles. En d'autres termes, ce modèle utilise les évaluations connues pour acquérir des connaissances et créer un modèle prédictif (Shah et al., 2017). La technique de filtrage basée sur un modèle utilise généralement le clustering et la classification (Al-Turjman, 2020).

Avantages et inconvénients d'un filtrage collaboratif

La technique de filtrage collaboratif basé sur les voisins (ou sur la mémoire) comporte certains avantages. Les méthodes basées sur les voisinages sont intuitives et relativement simples à

mettre en œuvre (Shah et al., 2017). Il est également très simple de justifier les recommandations faites grâce à un système de recommandation de filtrage collaboratif puisqu'il nous fournit des résultats assez intuitifs. Nous pourrions, par exemple, fournir la liste des plus proches voisins afin de justifier la recommandation. Les filtrages collaboratifs ne nécessitent pas de phases d'entraînement qui s'avèrent généralement coûteuses et une partie des calculs peut se faire en hors-ligne. De plus, le stockage des plus proches voisins nécessite peu de mémoire. Cela rend ce type de système de recommandation très efficace. Enfin, une autre propriété utile des systèmes de recommandation basés sur cette approche est qu'ils sont peu affectés par l'ajout d'utilisateurs, d'articles et d'évaluations lorsque nous avons des informations suffisantes concernant les évaluations des nouveaux utilisateurs (Ricci, Rokach, & Shapira, 2022). Il n'est également pas obligatoire pour notre algorithme d'avoir de quelconques connaissances sur le sujet traité par le système de recommandation (Kembellec, Chartron, & Saleh, 2014).

La difficulté principale lors de la conception d'un système de recommandation basé sur du filtrage collaboratif se pose lorsque nous faisons face à des matrices d'évaluations clairsemées et peu denses (Aggarwal, 2016). Cela arrive quand nous manquons d'information. Ce genre de méthode souffre également de problème de démarrage à froid lors de l'ajout d'un nouvel utilisateur. Dans ce cas-ci, il s'agit encore une fois d'un problème d'information. Nous ne disposons pas de la quantité d'information adéquate pour permettre à notre système de prédire correctement. La scalabilité est d'une importance capitale et peut poser problème dans un filtrage collaboratif. En effet, un modèle peut être efficace pour un petit nombre de données, mais être incapable de générer des recommandations satisfaisantes lorsque le volume des données augmente. Enfin, la synonymie apparaît lorsque des *items* ont des noms différents alors qu'ils se trouvent être très similaires (Isinkaye et al., 2015).

2.1.1.1. Système de recommandation basé sur le contenu

Certaines bases de données comportent non seulement des informations sur les évaluations des utilisateurs sur des articles, mais également des informations propres à chaque article. Ceux-ci sont appelés « attributs » ou encore « caractéristiques » (Pazzani & Billsus, 2007). Cette catégorie de système de recommandation utilise les informations sur les attributs des utilisateurs et des articles, telles que les profils textuels ou les mots-clés pertinents (Aggarwal, 2016). Il va ensuite analyser ceux-ci afin de recommander des articles pouvant correspondre à l'utilisateur (Pazzani & Billsus, 2007).

Afin de pouvoir exploiter les informations mises à notre disposition, il est d'usage de les transformer en une représentation structurée. Ainsi, chaque mot ou caractéristique est représenté par la valeur 1 s'il est applicable à l'article en question. Pour des valeurs textuelles, il existe la solution dite du « stemming » pour ensuite calculer un tf-idf (Pazzani & Billsus, 2007). Néanmoins, cette technique ne sera pas nécessaire dans le cadre de ce mémoire et ne sera donc pas développée davantage.

L'étape suivant l'extraction et la vectorisation des attributs de nos articles est l'apprentissage des profils d'utilisateurs. En effet, une fois que nous avons les informations concernant nos articles, il nous faut comprendre les préférences de l'utilisateur. C'est uniquement après cette étape que les recommandations sont possibles (Aggarwal, 2016).

2.2.3. Mesures de similarité

Dans la conception d'un système de recommandation de filtrage collaboratif ou basé sur le contenu, tout repose sur le fait de trouver des similarités entre *items* et *users*. Une fois cela fait, notre système pourra alors déterminer quels groupes d'utilisateurs et d'articles sont les plus similaires à notre cible (Sondur, Nayak, & Chigadani, 2016). Pour ce faire, plusieurs techniques sont possibles. Nous allons en présenter quelques-unes en nous concentrant uniquement sur le calcul de similarité entre utilisateurs. Néanmoins, toutes ces techniques sont également applicables dans le calcul de similarité entre articles.

Distance euclidienne

La distance euclidienne est une notion fondamentale dans de nombreuses mesures de similarité et de dissimilarité. Elle est définie comme suit (Sondur, Nayak, & Chigadani, 2016).

$$Distance\ euclidienne(u, v) = d(u, v) = \sqrt{\sum_i^n (r_{ui} - r_{vi})^2} \quad (2.3)$$

où $d(u, v)$ est la distance entre les vecteurs U et V ;

r_{ui} et r_{vi} sont les notes d'un *item* i données par deux utilisateurs u et v différents pour le même élément ;

n est le nombre d'éléments évalués en commun.

Similarité Cosinus

Cette technique consiste à représenter des objets (dans notre cas, des *users* ou des *items*) sous la forme de vecteurs et d'ensuite calculer leur similarité grâce à leur position respective dans l'espace vectoriel (Sondur, Nayak, & Chigadani, 2016). Nous utiliserons alors la valeur de l'angle cosinus entre ces deux vecteurs qui représentent, pour rappel, deux utilisateurs. Cette valeur nous indique la similarité entre ceux-ci. En cas de valeur proche de 1, nous pouvons conclure une forte corrélation entre nos deux objets tandis qu'au contraire, une valeur proche de 0 reflète une faible voir absence de corrélation. Nous pouvons formuler cette similarité comme suit (Fkih, 2021). Gardons en tête que la similarité cosinus est calculée grâce aux évaluations brutes et non pas grâce aux évaluations centrées sur la moyenne (Aggarwal, 2016).

$$\text{Similarité Cosine}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{u \in I_u} r_{ui}^2} \sqrt{\sum_{v \in I_v} r_{vi}^2}} \quad (2.4)$$

où I_u est l'ensemble des *items* évalués par le *user* u ;
 I_{uv} est l'ensemble des *items* évalués par le *user* u et le *user* v , représentant ainsi l'intersection entre I_u et I_v ($I_u \cap I_v$) ;
 r_{uj} est l'évaluation du *user* u et pour l'*item* j .

Dans la formule précédente, les facteurs de normalisation du dénominateur se basent sur l'ensemble des *items* connus, mais nous pourrions n'utiliser que les *items* ayant été évalués par les deux utilisateurs. Ce qui nous donnerait la formule suivante (Aggarwal, 2016).

$$\text{Similarité Cosine}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (2.5)$$

où I_u est l'ensemble des *items* évalués par le *user* u ;
 I_{uv} est l'intersection entre I_u et I_v représentant ainsi l'ensemble des *items* évalués par le *user* u et le *user* v ($I_u \cap I_v$) ;
 r_{uj} est l'évaluation du *user* u et pour l'*item* j .

Cette méthode comporte néanmoins un désavantage. En effet, les différences de moyennes et de variances entre les évaluations des *user* u et *user* v ne sont pas prises en considération (Ricci,

Rokach, & Shapira, 2022). C'est pourquoi il est généralement préférable d'utiliser la similarité Pearson qui va tenir compte de ce biais (Aggarwal, 2016).

Similarité Pearson

Le coefficient de corrélation de Pearson est utilisé pour évaluer dans quelle mesure deux variables sont linéairement corrélées l'une à l'autre (Isinkaye et al., 2015). Le coefficient se situe entre -1 révélant que les objets ne sont pas corrélés et +1 indiquant une corrélation parfaite entre les deux objets d'intérêt (Sondur, Nayak, & Chigadani, 2016). La formule de cette similarité se présente comme ceci (Hassanieh, et al., 2018).

$$Similarité\ Pearson(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (2.6)$$

où I_{uv} est l'ensemble des *items* évalués par le *user* u et le *user* v , représentant ainsi l'intersection entre I_u et I_v ($I_u \cap I_v$) ;

$\bar{r}_u$ est la moyenne des évaluations fournies par le *user* u ;

r_{uj} est l'évaluation du *user* u et pour l'*item* j .

L'un des éléments nécessaires au calcul de cette similarité sont les moyennes des évaluations fournies par le *user* u et v , représentées dans notre formule par $\bar{r}_u$ et $\bar{r}_v$. Celles-ci peuvent être calculées de la manière suivante (Aggarwal, 2016).

$$Moyenne\ des\ notations(u) = \mu_u = \frac{\sum_{i \in I_u} r_{ui}}{|I_u|} \quad \forall u \in \{1 \dots U\} \quad (2.7)$$

où I_u est l'ensemble des *items* évalués par le *user* u ;

r_{ui} est l'évaluation du *user* u et pour l'*item* i .

Notons que cette moyenne inclut l'ensemble des *items* évalués par le *user* u alors que la définition de Pearson implique qu'elle soit uniquement calculée avec les éléments évalués à la fois par le *user* u et le *user* v . Néanmoins, il n'est pas prouvé que l'une façon de procéder donne de meilleurs résultats que l'autre. Il est donc plus courant et plus simple de calculer chaque μ_u une seule fois pour chaque utilisateur u (Aggarwal, 2016).

Cette mesure de similarité de Pearson ne parvient pas à fournir des résultats corrects lorsque les deux utilisateurs ont une évaluation en commun ou lorsqu'un utilisateur n'a évalué qu'un seul *item* (Hassanieh, et al., 2018).

Similarité Spearman

La méthode de Spearman utilise un système de ranking. Elle attribue un rang à chaque évaluation, ce qui va régler le problème de leur normalisation. En cas de note similaire, un rang moyen est attribué aux *items* en fonction de leur position dans le classement (Hassanieh, et al., 2018).

$$\text{Similarité Spearman } (u, v) = \frac{\sum_{i \in I_{uv}} (k_{ui} - \bar{k}_u)(k_{vi} - \bar{k}_v)}{\sqrt{\sum_{i \in I_{uv}} (k_{ui} - \bar{k}_u)^2 \sum_{i \in I_{uv}} (k_{vi} - \bar{k}_v)^2}} \quad (2.8)$$

où I_{uv} est l'ensemble des *items* évalués par le *user* u et le *user* v , représentant ainsi l'intersection entre I_u et I_v ($I_u \cap I_v$);

$\bar{k}_u$ est le rang moyen des éléments évalués par le *user* u ;

k_{uj} est le rang attribué à l'*item* j sur l'ensemble des *items* évalués par le *user* u
l'évaluation du *user* u et pour l'*item* j .

Cette mesure produit néanmoins un grand nombre de rangs égaux en cas de faible étendue d'évaluation possible, par exemple, si les utilisateurs ne peuvent évaluer leur *item* sur une échelle n'allant que de 1 à 5 rendant cette mesure mauvaise pour ce genre de cas. De plus, la nécessité de garder en mémoire les différentes évaluations afin de calculer les rangs correspondants rend cette technique plus coûteuse (Hassanieh, et al., 2018).

2.3. Évaluation des systèmes de recommandation

Plusieurs objectifs peuvent être recherchés lors de l'évaluation d'un système de recommandation. Les plus connus sont basés sur un objectif de précision avec notamment des mesures tel que le Recall, la racine de l'erreur quadratique moyenne (RMSE) ou encore l'erreur absolue moyenne (MAE) (Aggarwal, 2016). Les mesures basées sur la précision ont longtemps été un moyen d'évaluation de système de recommandation capital. L'emphase était mise sur ces mesures et nous permettait de définir si un système était bon ou non. Néanmoins, des qualités supplémentaires à la précision se sont imposées comme élément significatif impactant la qualité globale d'un système. C'est pour cela que nous voyons apparaître des mesures telles

que la diversité ou la nouveauté, nous permettant ainsi d'aller « *au-delà de la précision* » (Kaminskas, & Bridge, 2016). Ces différentes mesures peuvent pour certaines être calculées de manière précise avec des formules mathématiques. Tandis que d'autres demandent l'intervention des utilisateurs eux-mêmes qui devront répondre à des questionnaires afin de pouvoir évaluer les objectifs plus subjectifs (Aggarwal, 2016). Un bon système de recommandation est un système alliant la précision, la diversité et la nouveauté (Chen, & Liu, 2017).

2.3.1 Mesures basées sur la précision

Lorsque l'on parle de mesures basées sur la précision, nous mesurons l'écart entre le classement ou les notations des *items* prédits pour un utilisateur par un système de recommandation et son classement ou notations réelles. En d'autres mots, nous évaluons soit les estimations des notations, soit les *items* recommandés par notre système (Herlocker, & al., 2004). Il existe deux types de métriques de précision. Nous nous concentrerons sur l'erreur absolue moyenne (MAE) et la racine de l'erreur quadratique moyenne (RMSE) qui font partie des mesures de précision statistiques et qui sont exploitées dans le but de comparer les prédictions de notre modèle avec les prédictions réelles de l'utilisateur. La précision, le recall et la F-mesure sont alors des mesures de précision d'aide à la décision. Elles sont utiles afin de déterminer si les éléments proposés sont de bonne qualité ou non en fonction des éléments disponibles (Isinkaye et al., 2015). Il s'agit donc d'une mesure de précision de classification. Une troisième catégorie peut venir s'ajouter à ces deux dernières : les mesures de précision de classement. Celles-ci déterminent la qualité du classement, et donc, d'une liste ordonnée que fournit un système de recommandation, par rapport au classement réel des éléments par l'utilisateur (Herlocker, & al., 2004). Nous pouvons, par exemple, citer le gain cumulé actualisé (DCG).

Mesures pour évaluer les prédictions

$$\text{Erreur absolue moyenne} = MAE = \frac{1}{|Q|} \sum_{(u,i) \in Q} |r_{ui} - \hat{r}_{ui}| \quad (2.9)$$

$$\text{Racine de l'erreur quadratique moyenne} = RMSE = \sqrt{\frac{1}{|Q|} \sum_{(u,i) \in Q} (r_{ui} - \hat{r}_{ui})^2} \quad (2.10)$$

où Q est l'ensemble de test ;

r_{ui} est l'évaluation du *user* u pour l'*item* i ;

$\tilde{r}_{ui}$ est la prédiction de l'évaluation du *user* u pour l'*item* i .a

L'erreur absolue moyenne ne tient pas compte du signe de l'erreur. La racine de l'erreur quadratique règle alors ce problème du fait qu'un exposant carré est appliqué aux erreurs. De plus, le fait de prendre la racine nous évite un problème d'interprétation contrairement à la simple erreur quadratique (Chen, & Liu, 2017). Néanmoins, le fait de mettre au carré va pénaliser fortement la mesure de précision en cas d'une erreur de valeur importante. De plus, l'erreur moyenne n'est pas réellement représentée par la valeur que nous procure le RMSE, ce qui peut nous induire en erreur. La réponse à la question : « Quelle est la meilleure mesure ? » va alors dépendre du problème devant lequel nous nous trouvons. Nous aurons tendance à utiliser le RMSE lorsque nos prédictions sont robustes tandis que nous privilégierions plus le MAE si peu de valeurs extrêmes sont présentes dans nos données (Aggarwal, 2016).

Mesures pour évaluer la liste des items recommandés

La précision et le recall sont grandement utilisés dans le domaine de recherche d'information. Il est donc naturel de les retrouver dans l'évaluation de système de recommandation (Herlocker, & al., 2004). Afin de calculer ces métriques, il nous faut d'abord déterminer la matrice de confusion (figure 2.1) découlant de notre modèle (Hernández del Olmo, & Gaudioso, 2008).

	Pertinent	Non-Pertinent
<i>Recommandé</i>	a	b
<i>Non recommandé</i>	c	d

Figure 2.1 Résultats possible lors de recommandation

Une bonne décision de recommandation ou non-recommandation se trouvera dans les cases de diagonales. En effet, le « a » représente les éléments qui sont pertinents à l'utilisateur et qui lui ont effectivement été recommandés. Tandis que le « d » reprend les éléments non pertinents et qui non donc pas été recommandé par le système de recommandation (Hernández del Olmo, & Gaudioso, 2008).

En partant de cette matrice de confusion, nous pouvons alors déterminer les valeurs de précision et de recall de notre modèle.

$$Précision = \frac{a}{a + b} = \frac{\text{Items correctement recommandés}}{\text{Ensemble des items recommandés}}$$

(2.11)

$$Recall = \frac{a}{a + c} = \frac{\text{Items correctement recommandés}}{\text{Ensemble des items pertinents pour l'utilisateur}} \quad (2.12)$$

Selon les formules, nous pouvons constater que la probabilité qu'un élément recommandé soit pertinent est exprimée par la précision. Le recall révèle la probabilité qu'un élément pertinent soit recommandé (Herlocker, & al., 2004).

La mesure F utilise ces deux métriques afin de fournir un compromis entre la proportion des éléments préférés des utilisateurs représentée par la précision et la proportion des éléments préférés qui n'ont pas été oubliés dans la recommandation représentée par le recall. Cette valeur nous permet donc d'intégrer le comportement de ces deux métriques en une seule (Chen, & Liu, 2017).

$$F - measure = \frac{2 * Précision * Recall}{Précision + Recall} \quad (2.13)$$

Mesures pour évaluer le classement des items recommandés

L'apparition de mesure évaluant le classement des items recommandés dans l'évaluation d'un système de recommandation vient de l'importance pour les utilisateurs de trouver les meilleures recommandations en début de liste. En effet, nos utilisateurs ne veulent pas passer une période de temps interminable à inspecter chaque élément de la liste pour déterminer quelle solution est la meilleure. C'est pour cette raison que des mesures prenant en compte la position de chaque élément ont été introduites. Celles-ci attribuent alors un poids à chaque *item* en fonction de sa position (Lü, et al., 2012).

Contrairement aux moyens d'évaluation précédents qui ne prennent pas en considération la position des items dans le calcul des pénalités des évaluations, le gain cumulé actualisé actualise les positions de manière logarithmique (Shani, & Gunawardana, 2011). Il peut être formulé de la manière suivante (Aggarwal, 2016).

$$Gain\ cumulé\ actualisé = DCG = \frac{1}{U} \sum_{u=1}^m \sum_{j \in I_u} \frac{g_{uj}}{\log_2(k_j + 1)} \quad (2.14)$$

où U est l'ensemble des utilisateurs ;
 I_u l'ensemble de test ;
 r_{uj} est l'utilité (ou l'évaluation) du *user* u pour l'*item* j ;
 k_j est la position de l'*item* j dans l'ensemble de test.

La base logarithme de valeur 2 assure que toutes les positions soient actualisées et est donc généralement utilisée malgré le fait que la base reste un paramètre libre (Shani, & Gunawardana, 2011).

Il est ensuite possible de normaliser le gain cumulé actualisé en le divisant par sa valeur idéale (IDCG) (Aggarwal, 2016).

$$\text{Gain cumulé actualisé normalisé} = NDCG = \frac{DCG}{IDCG}$$

(2.15)

2.3.2. Mesures d'évaluation « au-delà de la précision »

Novelty and Serendipity

Le principe de nouveauté ou novelty provient du fait que notre modèle recommande des *items* à notre utilisateur que celui-ci ne connaissait pas avant (Schafer et al., 2007). La littérature nous offre également une autre définition prenant alors en compte l'historique des utilisateurs, définissant ainsi un nouvel *item* comme un article non présent dans son historique de consommation (Silveira, & al., 2019).

Plusieurs méthodes sont possibles afin d'évaluer la nouveauté d'un système. La plus simple consiste en un filtrage des *items* que les utilisateurs connaissent déjà et ont donc déjà évalués ou utilisés (Wu, He, & Yang, 2012). Une autre méthode communément exploitée utilise la popularité des *items* dans le mécanisme d'évaluation. L'idée se cachant derrière ce processus est que les *items* populaires ont plus de chance d'être connus et donc moins de chance d'apporter de la nouveauté dans l'éventail de choix proposé à l'utilisateur. Dans la pratique, des poids différents seront attribués aux *items* en fonction de leur popularité dans le calcul des mesures de précision (Shani, & Gunawardana, 2011).

Voici deux métriques proposées afin de calculer la nouveauté qui utilise la popularité qui, quant à elle, peut être calculée par le nombre d'utilisateurs qui ont évalué l'*item* en question (Silveira, & al., 2019).

$$nov(R_u) = \sum_{i \in R_u} \frac{\log_2 pop(i)}{|R_u|}$$

(2.16)

$$nov(R_u) = 1 - \frac{|pop(i)|}{|U|}$$

(2.17)

où R_u est la liste des recommandations pour le *user* u ;
 U est l'ensemble des *users* ;
 $pop(i)$ est la popularité de l'*item* i .

Le calcul de la nouveauté, en la définissant comme étant l'inverse de la popularité, assume une certaine indépendance quant à l'utilisateur concerné. En d'autres mots, nous considérons la nouveauté d'une manière globale. Néanmoins, la nouveauté peut, en pratique, grandement varier d'un utilisateur à un autre. C'est pourquoi une technique déterminant le caractère inattendu d'une recommandation d'un point de vue d'un utilisateur spécifique sert à mesurer la nouveauté d'un modèle (Castells, Hurley, & Vargas, 2015).

$$Unexp = \frac{1}{|R||I_u|} \sum_{i \in R} \sum_{j \in I_u} d(i, j)$$

(2.18)

où R est la liste de recommandations à un utilisateur ;
 I_u est l'ensemble des *items* avec lesquels le *user* u a interagi ;
 $d(i, j)$ est la distance entre l'*item* i et l'*item* j .

Cette formule traduit la nouveauté d'un *item* en fonction de sa distance sur ses attributs par rapport aux éléments déjà rencontrés par l'utilisateur cible. Nous nous retrouvons alors à évaluer la deuxième définition proposée par la littérature précédemment évoquée, à savoir, celle où l'on se base sur l'historique de notre utilisateur (Castells, Hurley, & Vargas, 2015).

Une idée très liée au concept de nouveauté est la sérendipité ou serendipity qui procure des recommandations que les utilisateurs ne connaissent pas, mais qu'ils n'auraient pas été capables de découvrir étant donné leur moyen de recherche et de découverte de nouveaux *items* (Schafer et al., 2007). En d'autres mots, elle mesure la surprise de l'utilisateur quant aux

recommandations fournies (Wu, He, & Yang, 2012). C'est cette notion de surprise qui la distingue de la nouveauté. En effet, le simple fait d'être inconnu à l'utilisateur suffit à rentrer dans la catégorie d'un nouvel *item* (Kaminskas, & Bridge, 2016). Il est possible de mesurer la sérendipité en jugeant si un élément est inattendu et utile pour l'utilisateur (Ge, Delgado, & Jannach, 2010). Cela peut se faire à l'aide des utilisateurs eux-mêmes qui nous font part de leur point de vue quant aux deux caractéristiques de la sérendipité précédemment évoquées. Les *items* rentrant dans les deux catégories sont utilisés comme mesure de la sérendipité. Ce processus se fait alors en ligne. Une technique hors ligne et automatisée est également possible. Dans ce cas, l'utilisation d'un système de recommandation basé sur le contenu est nécessaire afin de comparer ses recommandations et celles présentes dans la liste des tops k articles. Les articles corrects, mais qui ne sont pas recommandés par le modèle de base, sont exploités de manière à fournir le degré de sérendipité (Aggarwal, 2016).

Diversity

La diversité d'une liste de recommandations peut être définie comme l'opposé de la similarité (Chen, & Liu, 2017). Cette notion implique donc que les recommandations présentes dans la liste pour un seul utilisateur soient le plus variées possible (Aggarwal, 2016). Initialement, un seul ensemble d'*items* est nécessaire pour déterminer la mesure de diversité au sein d'une liste, correspondant donc à la diversité pour un seul *user*. Cette mesure étant ensuite généralement agrégée en moyenne pour l'ensemble des utilisateurs. Mais l'apparition d'une diversité globale peut également être utilisée avec plusieurs ensembles d'*items*, délivrant ainsi des résultats dans 3 types de situations : pour tous les utilisateurs, pour différents modèles traitant un seul utilisateur ou encore pour un seul utilisateur et modèle à des temporalités différentes (Castells, Hurley, & Vargas, 2015).

La dissimilarité moyenne nous permet de mesurer la diversité individuelle d'un utilisateur (Karakaya, & Aytekin, 2018). Celle-ci est également connue sous le nom de distance intra-liste moyenne dans d'autres travaux de la littérature. Afin de calculer la distance entre deux articles, nous utilisons 1- la similarité de ceux-ci (Castells, Hurley, & Vargas, 2015).

$$\text{Dissimilarité moyenne} = \frac{1}{|R_u| (|R_u| - 1)} \sum_{i \in L_u} \sum_{j \in R_u, j \neq i} d(i, j)$$

(2.19)

où R_u est la liste de recommandation proposée au *user* u ;
 $d(i, j)$ est la distance entre les *items* i et j .

La diversité agrégée est représentée par le nombre d'*items* distincts présents dans l'ensemble des recommandations faites aux utilisateurs (Niemann, & Wolpers, 2013).

$$Diversité\ agrégée = \left| \bigcup_{u \in U} (P_H(u) \cap R(u)) \right| \quad (2.20)$$

où $P_H(u)$ est l'ensemble des *items* bien évalués par le *user* u ;
 $R(u)$ est l'ensemble des *items* recommandé au *user* u .

Nous pouvons alors définir $P_H(u) \cap Rec(u)$ comme étant les meilleurs *items* recommandés au *user* (Castells, Hurley, & Vargas, 2015).

2.4. Bénéfices d'un système de recommandation

Au vu de l'utilisation grandissante des systèmes de recommandation dans notre société actuelle, il est intéressant de se demander comment ceux-ci nous affectent possiblement. Cette dernière impacte tant les consommateurs que les entreprises recourant à cette pratique. Nous allons dans cette section développer les bénéfices que peuvent tirer ces deux partis lors de l'emploi de ce type de technologie.

Commençons par les bénéfices des entreprises. L'un des bénéfices le plus direct et le plus important est l'augmentation des ventes de produits existants (Sarwar, & al., 2000). Cette augmentation est possible grâce à la recommandation d'articles susceptibles d'intéresser notre consommateur augmentant par ailleurs le taux de conversion. Ce dernier peut être évalué en prenant le nombre de fois qu'un article recommandé sera acheté ou non (Ricci, Rokach, & Shapira, 2022).

Une aide à la gestion de la surcharge des informations présentes sur Internet ainsi que l'étendue des choix proposés par les systèmes de recommandation permet une plus grande satisfaction des consommateurs (Kuzelewska, 2014). De plus, les utilisateurs tendent à être plus satisfaits face à une interface agréable et des recommandations pertinentes (Ricci, Rokach, & Shapira, 2022). Certains utilisateurs auront également tendance à valoriser davantage des produits similaires à leurs produits habituels. Ils seront alors plus satisfaits si le site conserve leurs

habitudes de consommation leur permettant donc d'être guidés lors de la prochaine visite (Mohamed, Khafagy, & Ibrahim, 2019). Un concept lié à ce dernier point est l'augmentation de la diversité des articles vendus résultant d'une facilité accrue à connaître de nouveaux produits difficiles à découvrir sans l'aide d'une telle technologie (Ricci, Rokach, & Shapira, 2022).

En s'appuyant sur la reconnaissance des utilisateurs et des analyses de sentiments, les systèmes de recommandations peuvent être utilisés afin d'augmenter la loyauté et donc de fidéliser sa clientèle (Tarnowska, Ras, & Daniel, 2019).

L'activité des utilisateurs peut également être impactée. Elle se manifeste par une augmentation des visites, de leurs durées, du nombre de clics lors d'un passage sur le site ou autres mesures. L'augmentation du taux d'engagement pourrait alors s'accompagner d'une utilisation répétée grandissante (Jannach, Zanker, 2022).

Un aspect important facilité par l'utilisation des systèmes de recommandation est la compréhension des besoins de l'utilisateur. Ceci est dû à la création de profils grâce aux préférences collectées ou prédites par le modèle. Les informations que nous avons à notre disposition peuvent alors être réutilisées pour d'autres objectifs (Ricci, Rokach, & Shapira, 2022).

Maintenant que nous avons vu les différents bénéfices que retirent les entreprises dans l'implémentation de recommandeurs sur leur site, il est légitime de se demander si les bénéfices de ce genre de pratique sont également applicables aux utilisateurs.

En plus de l'augmentation de la satisfaction des utilisateurs déjà mentionnée auparavant, plusieurs autres grands bénéfices sont à souligner. Le premier est l'aide à la prise de décision. En effet, en recommandant des articles correspondants au profil de l'utilisateur, les systèmes de recommandations permettent de faciliter les choix des consommateurs. Nous pouvons également noter un gain de temps considérable par rapport à une utilisation sans système de recommandation (Tintarev, & Masthoff, 2007).

2.5. Conclusion

Dans cette revue de littérature, nous avons pu clarifier les grands concepts de la littérature concernant les systèmes de recommandation. Nous avons dans un premier temps défini ce qu'est un système de recommandation ainsi que les raisons de son apparition. Nous nous

sommes ensuite concentrés sur deux types de système de recommandation, le filtrage collaboratif et le système de recommandation basé sur le contenu. Nous avons découvert deux méthodes de filtrage collaboratif : le système basé sur la mémoire ou, autrement dit, sur le voisinage avec ses déclinaisons se concentrant soit sur les articles, soit sur les utilisateurs et le système basé sur le modèle.

Pour rappel, ces derniers utilisent les similarités entre les utilisateurs ou les objets afin de fournir ses prédictions. C'est pour cela que nous nous sommes attardés sur quelques mesures de similarité possible ainsi que la manière de les calculer. Nous avons vu la distance euclidienne, la similarité cosinus, la similarité Pearson et la similarité Spearman. La similarité Pearson s'était alors révélée meilleure que la distance euclidienne étant donné qu'elle tient en compte les différences de moyennes et de variances des notations. La similarité Spearman, quant à elle, réglait le problème de normalisation, mais posait problème en cas d'échelle de notation trop faible.

La troisième partie de cette revue de littérature portait sur les moyens d'évaluations des systèmes de recommandation. Nous avons d'abord développé le principe de précision ainsi que différentes mesures pour la mesurer en les distinguant entre les mesures pour évaluer les prédictions, les listes des items recommandés ou le classement des items recommandés. Nous avons ensuite pu constater que de plus en plus de mesures allaient au-delà de la précision et nous permettaient donc d'évaluer notre modèle en utilisant une approche différente. C'est d'ailleurs ce que nous permet de faire la nouveauté, la sérendipité et la diversité. Mesures, qui ont été développées dans cette partie.

Afin, nous avons pu découvrir les bénéfices que les utilisateurs et les entreprises pouvaient retirer de l'utilisation de système de recommandation. Les principaux étant une facilité accrue dans la prise de décision des utilisateurs et l'augmentation des ventes pour les entreprises.

Comme nous l'avons laissé entendre dans l'introduction, nous constatons que les mesures d'évaluation de système de recommandation présentes dans la littérature se basent essentiellement sur une moyenne. En d'autres termes, nous considérons qu'un système est performant s'il, en moyenne, fournit de bons résultats pour l'ensemble des utilisateurs. Or, il est légitime de se demander s'il ne serait pas pertinent d'introduire le profil des utilisateurs dans ce calcul. Nous pourrions alors nous concentrer sur un certain groupe d'utilisateur qui nous intéresse davantage comme, par exemple, les plus actifs sur notre plateforme. C'est ce que nous

essayerons de mettre en place dans la suite de ce mémoire. La partie suivante explique comment nous procéderons afin d'obtenir des résultats afin de tenter de répondre à notre question de recherche.

3. Méthodologie

3.1. Introduction

Dans cette partie, nous allons développer la méthodologie avec laquelle ce travail sera réalisé. L'objectif de celle-ci est de comprendre le cheminement de notre raisonnement du début jusqu'à la fin.

Nous commencerons par la découverte des différents logiciels et outils utilisés lors de la conception de ce mémoire. Ceux-ci comprendront le langage de programmation ainsi que le logiciel utilisé et également les librairies présentes dans notre code.

Une présentation des bases de données sera effectuée afin de la comprendre en profondeur. Il s'agit d'une étape importante afin d'avoir une bonne compréhension de la provenance de nos résultats mais également lors de la conception des différents codes.

Ensuite, les différents systèmes de recommandation que nous utiliserons seront expliqués. Nous partirons de trois systèmes de base dont deux provenant du mémoire de Louise Cotton précédemment publié et pour lesquels certaines modifications ont été nécessaires afin de correspondre à notre objectif.

Enfin, nous présenterons les mesures d'évaluation nous permettant de réaliser notre analyse. Et nous adapterons ensuite celles-ci en fonction de différents critères. En effet, nous attribuerons des poids différents en fonction du profil d'utilisateurs ou de groupe d'utilisateurs. Le but de cette manœuvre est de privilégier et d'accorder plus d'importance à certains utilisateurs plutôt qu'à d'autres dans l'évaluation de la performance. Trois critères seront pris en considération : la fréquence de notation d'un utilisateur, le nombre de notations fournies par un utilisateur et la temporalité des notations d'un utilisateur.

3.2. Logiciels et outils

Nos systèmes de recommandation seront codés dans le langage de programmation Python. Nous utiliserons alors l'environnement Jupyter Lab qui est une interface utilisateur qui fonctionne à travers un navigateur web.

Lors de l'exécution de nos codes, certaines librairies python seront utilisées. La librairie Csv nous servira pour la lecture des fichiers de notre base de données. En effet, les fichiers sont disponibles sous ce format et sont détaillés dans la section « Présentation de la base de

données ». Une autre librairie traitant des fichiers qui nous sera utile est Json. Celle-ci sera utilisée pour stocker nos matrices d'adjacence, de genres et d'années ou encore nos listes d'importances créées. Cela nous permet de ne pas avoir à faire tourner le code à chaque fois que nous avons besoin d'un de ces éléments. De plus, il s'agit d'un format de fichier très léger et facilement exploitable en langage python. L'extraction de certaines données a besoin d'être faite grâce aux librairies Datetime et Re. En effet, Datetime nous servira à transformer les timestamps des notations en des dates compréhensibles. Tandis que Re, viendra extraire les années des titres des films afin de les exploiter. La présence de nombreuses listes et de matrices dans nos codes nécessite l'utilisation de librairies telles que Numpy et Scipy. Elles nous permettront de faciliter la manipulation de celles-ci et également la création de matrices creuses nous donnant la possibilité d'économiser beaucoup de mémoire. La librairie Random nous sera utile lors de la création de la validation croisée. Enfin, la dernière librairie utilisée sera Sklearn en nous fournissant la similarité cosinus de la matrice entrée en paramètre.

3.3. Présentation de la base de données

La première base de données que nous allons utiliser dans ce mémoire provient de « MovieLens » et a été générée en 2018. Celle-ci reprend les notations de 610 utilisateurs étalés sur 22 ans (de 1996 à 2018). Elle comptabilise alors 100836 notations sur 9724 films. Les utilisateurs ont été sélectionnés de manière aléatoire. Nous avons néanmoins décidé de ne travailler qu'avec les films ayant plus de 10 notations réduisant ce nombre à 2269 films.

Quatre fichiers sont à notre disposition : *links.csv*, *movies.csv*, *ratings.csv* et *tags.csv*. *movies.csv* reprend la liste des films présents dans la base de données. Chaque film comprend un « movieId » unique, une colonne « title » contenant le titre du film suivi de son année de sortie entre parenthèses et une colonne « genres » stockant tous les genres du film séparés par « | ». La figure 3.1 en reprend quelques lignes.

```
movieId, title, genres
3, Grumpier Old Men (1995), Comedy|Romance
15, Cutthroat Island (1995), Action|Adventure|Romance
34, Babe (1995), Children|Drama
```

Figure 3.1 Quelques lignes du fichier *movies.csv*

ratings.csv est le fichier reprenant les notations de films par les utilisateurs. Chaque ligne correspond la notation d'un film par un utilisateur comme vu dans la figure 3.2. Nous retrouvons d'abord l'identifiant unique de l'utilisateur dans « userId », ensuite le « movieId » identique à celui que l'on peut retrouver dans le fichier *movies.csv*. La troisième colonne,

appelée « rating », est celle de la notation en elle-même. Celle-ci est comprise entre 0.5 et 5.0 étoiles. Enfin la dernière colonne de ce fichier est une information temporelle de la notation. En effet, « timestamp » représente le nombre de secondes s'étant écoulées depuis le 1er janvier 1970 à minuit lorsque la notation a été réalisée. Cet élément pourra donc nous servir à calculer la date et l'heure exacte à laquelle un utilisateur a évalué un film.

```
userId, movieId, rating, timestamp
15, 34, 3.0, 997938310
15, 95, 1.5, 1093028331
15, 101, 4.0, 1134522072
```

Figure 3.2 Quelques lignes du fichier ratings.csv

Un tag fourni par un utilisateur sur un film est repris dans une ligne du fichier *tags.csv*. Nous retrouvons les colonnes « userId », « movieId » et « timestamp » respectivement en première, deuxième et quatrième position. Ceux-ci suivent le même principe que dans les fichiers précédemment expliqués. Les tags en eux-mêmes se retrouvent alors dans la troisième colonne intitulée « tag » et correspondent à un mot ou une courte phrase générée par l'utilisateur afin de donner une information supplémentaire sur le film, en outre, une métadonnée. Le sens de ce tag est alors propre à chaque utilisateur et peut donc diverger d'un utilisateur à l'autre.

```
userId, movieId, tag, timestamp
15, 32892, Russian, 1170626366
77, 4467, Trilogy of the Imagination, 1163220065
77, 5909, Takashi Miike, 1163219591
```

Figure 3.3 Quelques lignes du fichier tags.csv

Enfin, le dernier fichier est *links.csv* dont nous pouvons observer la structure en figure 3.4. Celui-ci permet de retrouver des informations supplémentaires d'un film en fournissant pour chaque identifiant de film présent dans notre base de données (identifiable grâce à son « movieId »), son équivalent sur imdb dans la colonne « imdbId » et tmdb dans « tmdbId » qui sont toutes les deux des bases de données en ligne. Pratiquement, nous pourrions retrouver les pages correspondants à un film sur les différentes plateformes grâce aux identifiants fournis.

```
movieId, imdbId, tmdbId
3, 113228, 15602
15, 112760, 1408
34, 112431, 9598
```

Figure 3.4 Quelques lignes du fichier links.csv

Dans l'objectif de confirmer ou infirmer les analyses et éventuelles tendances observées pour cette première base de données, nous en utiliserons une deuxième afin de comparer les différents résultats et de voir s'ils sont contradictoires ou non. Ici encore, nous utiliserons une

base de données provenant de MovieLens trouvée sur la plateforme Kaggle. Nous n'utiliserons dans ce cas que deux fichiers mis à notre disposition : *ratings_small.csv* et *movies_metadata.csv*. Ces fichiers comportent des informations sur les notations de films de 671 utilisateurs représentant 100 004 notations. De la même manière que pour la première base de données, nous ne prendrons en compte uniquement que les films ayant reçu plus de 10 notations. Ils s'élèvent alors au nombre de 2245. Une comparaison des deux fichiers *ratings* a été effectuée pour s'assurer qu'ils étaient assez différents. La similarité s'élevait à 0.00099%.

La structure du fichier *ratings_small.csv* est semblable à celle de la première base de données. Elle se présente, dans la figure 3.5, comme suit :

```
userId, movieId, rating, timestamp
1, 31, 2.5, 1260759144
1, 1061, 3.0, 1260759182
1, 1172, 4.0, 1260759205
```

Figure 3.5 Quelques lignes du fichier *ratings_small.csv*

Le fichier *movies_metadata.csv* est quant à lui quelque peu différent de son homologue de la première base de données. En effet, il se compose de beaucoup plus d'informations concernant les films présents dans la base de données comme par exemple, le budget, le réalisateur, la langue d'origine... Nous nous concentrerons néanmoins que sur les colonnes 4, 6 et 15 correspondant aux genres, à l'identifiant unique et à la date de parution du film. Celles-ci sont structurées comme dans la figure 3.6 suivante :

```
genres, id, release_date
[{'id': 10749, 'name': 'Romance'}, {'id': 35, 'name': 'Comedy'}] , 15602 , 1995-12-22
[{'id': 35, 'name': 'Comedy'}] , 11862 , 1995-02-10
[{'id': 35, 'name': 'Comedy'}, {'id': 10749, 'name': 'Romance'}] , 11860 , 1995-12-15
```

Figure 3.6 Quelques lignes du fichier *movies_metadata.csv*

3.4. Types de systèmes de recommandation utilisés

Dans le cadre de ce mémoire, nous nous concentrerons sur trois systèmes de base : un système de recommandation de filtrage collaboratif basé sur les utilisateurs (user based), un filtrage collaboratif basé sur les articles (item based) et un système de recommandation basé sur le contenu qui prendra en compte l'année de parution du film ainsi que ses genres. Le choix de l'utilisation de plusieurs types de recommandation repose sur la nécessité de pouvoir comparer ceux-ci entre eux en fonction des mesures d'évaluation classique et les mesures d'évaluation prenant en compte le profil des utilisateurs. Ainsi, nous pourrions observer si le classement de ces différents systèmes est influencé par la manière de les évaluer ou non.

3.4.1. Extraction des données et création des matrices

En partant de notre base de données, nous allons devoir créer une matrice afin de pouvoir calculer nos recommandations. La structure de notre base de données et le fait que nous ayons des données explicites, puisque les utilisateurs attribuent des notes aux films visionnés, nous permet de créer soit une matrice de notation explicite non binaire ou une matrice de notation implicite binaire. La première est composée des notations explicites des utilisateurs et prend donc en compte le degré d'appréciation de celui-ci pour chaque film noté. Tandis que la deuxième ne prend en compte que l'information que l'utilisateur a visionné le film ou non. La figure 3.7 reprend les deux matrices possibles.

4	2	0	0	0	1	1	0	0	0
0	1	5	0	0	0	1	1	0	0
1	0	4	0	1	1	0	1	0	1
0	0	3	0	4	0	0	1	0	1
1	4	0	3	0	1	1	0	1	0

Figure 3.7 Matrice de notation explicite non binaire (à gauche) et matrice de notation implicite binaire (à droite)

Les matrices créées à partir de cette extraction de données vont ensuite être stockées sous format JSON afin de faciliter leur utilisation par la suite. Notons néanmoins que nous considérons uniquement les films ayant obtenu plus de 10 notations. Étant donné que notre objectif est d'évaluer la qualité des recommandations et non pas la qualité de prédiction des notations par l'utilisateur, nous utiliserons la matrice de notation implicite binaire.

Afin de pouvoir construire un système de recommandation basé sur le contenu, nous avons besoin d'extraire les informations des films. Pour ce faire, nous avons repris la liste des films ayant plus de 10 notations, la liste des genres et la liste des années de sortie de tous les films. Nous avons ensuite créé une matrice creuse de dimension (nombre de films x nombre de genres + nombre d'années de sortie). Celle-ci sera une matrice binaire. Chaque film aura la valeur 1 dans les colonnes des genres qui lui correspond et dans la colonne adéquate suivant son année de parution.

Ainsi, si pour les films présents dans la figure 3.8,

Babe (1995)	ChildrenlDrama
Confessional, The (Confessionnal, Le) (1995)	Drama Mystery
Two if by Sea (1996)	Comedy Romance
Bio-Dome (1996)	Comedy
Broken Arrow (1996)	Action Adventure Thriller

Figure 3.8 Exemples de films dans la première base de données

Nous aurons la matrice 3.9 suivante :

1	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	0
0	0	0	1	1	0	0	0	0	1
0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	1	1	1	0	1

Figure 3.9 Matrice des genres et des années correspondants

Il est important de noter que lorsque l'année ou les genres sont absents, nous nous baserons uniquement sur l'information à notre disposition.

3.4.2. Système de recommandation de filtrage collaboratif basé sur les utilisateurs (user based)

Le premier système de recommandation sera un filtrage collaboratif basé sur les utilisateurs. Comme vu dans la revue de littérature, ce type de système de recommandation utilise les plus proches voisins de l'utilisateur en question pour calculer la prédiction. Pour rappel, la technique basée sur les utilisateurs va alors utiliser les utilisateurs ressemblant le plus à l'utilisateur cible et recommander en fonction.

Afin de déterminer les plus proches voisins de notre utilisateur, nous devons calculer la similarité entre ceux-ci. Plusieurs méthodes de calcul de similarité sont possibles et certaines d'entre elles ont été développées dans la revue de littérature. Au vu des différents avantages et inconvénients évoqués dans celle-ci et la considération de notre cas de figure (utilisation d'une matrice de notation implicite binaire), nous avons conclu que la méthode du cosinus serait la plus adaptée.

Nous rappelons donc brièvement sa formule :

$$Similarité\ Cosine(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (3.1)$$

où I_u est l'ensemble des *items* évalués par le *user* u ;

I_{uv} est l'intersection entre I_u et I_v représentant ainsi l'ensemble des *items* évalués par le *user* u et le *user* v ($I_u \cap I_v$) ;

r_{uj} est l'évaluation du *user* u et pour l'*item* j .

Cette similarité sera alors calculée pour chaque paire d'utilisateurs. La diagonale de notre matrice de similarité cosinus sera naturellement égale à 1, score le plus élevé possible, puisqu'il s'agit de la similarité entre l'utilisateur et lui-même. Notons également que la matrice est symétrique. Quant aux autres valeurs, nous trouvons la matrice 3.10 suivante lorsque nous utilisons les données de la figure 3.7 :

1	0.5	0.408	0	0.816
0.5	1	0.408	0.5	0.408
0.408	0.408	1	0.816	0.333
0	0.5	0.816	1	0
0.816	0.408	0.333	0	1

Figure 3.10 Matrice de similarité cosinus entre utilisateurs

Si nous jetons un œil à la similarité entre l'utilisateur 1 et 3, nous voyons qu'elle est égale à 0.408. Cela découle du fait que les deux utilisateurs ont une notation en commune, l'utilisateur 1 a noté 2 films en tout, tandis que l'utilisateur 3 en a noté 3. Nous avons donc :

$$\sum_{i \in I_{13}} r_{1i} r_{3i} = 1 * 1,$$

$$\sqrt{\sum_{i \in I_1} r_{1i}^2} = \sqrt{1^2 + 1^2}$$

$$\sqrt{\sum_{i \in I_3} r_{3i}^2} = \sqrt{1^2 + 1^2 + 1^2}.$$

$$\text{Nous donnons, } \frac{1*1}{\sqrt{1^2+1^2} \sqrt{1^2+1^2+1^2}} = \frac{1}{\sqrt{6}} = 0.408$$

Une fois cette matrice en notre possession, nous pouvons déterminer les plus proches voisins de chaque utilisateur et donc ceux nous permettant de faire nos prédictions. Nous sélectionnons alors les voisins ayant la similarité la plus élevée. Le nombre de voisins à considérer varie en fonction de la configuration du système de recommandation. Par exemple, pour un k égal à 2, nous voyons que les utilisateurs 2 et 5 ont la similarité la plus élevée par rapport à notre utilisateur 1. Ce seront donc ces deux utilisateurs que nous choisirons afin de prédire nos scores. Nous utilisons la formule suivante, comme vue dans la revue de littérature :

$$Prediction(u, i) = \tilde{r}_{ui} = \frac{\sum_{v \in N_u^i} Sim_{uv} * r_{vi}}{\sum_{v \in N_u^i} |Sim_{uv}|}$$

(3.2)

où $\tilde{r}_{ui}$ est la prédiction pour le user u et pour l'item i ;
 r_{vi} est l'évaluation du user v et pour l'item i ;
 N_u^i est l'ensemble des voisins les plus similaires au user u et qui ont évalué l'item i ;
 Sim_{uv} est la similarité entre le user u et le user v .

Nous obtenons la figure 3.11.

-1	-1	0.379	0.620	0
0.5	-1	-1	0	0.5
-1	0.333	-1	0	-1
0.620	0.379	-1	0	-1
-1	-1	0.333	-1	0

Figure 3.11 Matrice des prédictions basée sur les utilisateurs

Nous voyons que le score le plus élevé pour l'utilisateur 1 est celui du 4^e film. Nous avons obtenu ce score de la manière suivante :

$$\sum_{v \in N_1^4} Sim_{1v} * r_{v4} = 0.5 * 0 + 0.816 * 1 = 0.816$$

$$\sum_{v \in N_1^4} |Sim_{1v}| = 0.5 + 0.816 = 1.316$$

$$\widetilde{r}_{14} = 0.816 / 1.316 = 0.620$$

Notons que les films déjà vus par l'utilisateur se voient attribuer un score de -1 afin de ne pas être recommandés. Les autres films seront quant à eux sélectionnés en fonction de leur score et du nombre de films à recommander. Évidemment, les films ayant les plus gros scores seront proposés en priorité. C'est en suivant cette logique que nous obtenons les listes de recommandations de films de la figure 3.12.

Utilisateurs	Films vus	Films recommandés
1	1, 2	4, 3
2	2, 3	1, 5
3	1, 3, 5	2, 4
4	3, 5	1, 2
5	1, 2, 4	3, 5

Figure 3.12 Liste de recommandations sur base des utilisateurs

3.4.3. Système de recommandation de filtrage collaboratif basé sur les articles (item based)

Le deuxième système de recommandation utilisé sera un système de filtrage collaboratif basé sur les articles. Dans ce cas, et comme son nom l'indique, nous allons évaluer la similarité entre articles. Ensuite, les articles les plus similaires aux articles déjà consommés par l'utilisateur cible seront recommandés.

Nous procéderons de la même manière que pour le précédent système. À savoir, le calcul de la similarité cosinus entre articles et ensuite le calcul de la prédiction de notation de l'utilisateur en fonction de cette similarité. Une fois ces scores connus, nous recommanderons encore une fois les films ayant obtenu le score d'intérêt le plus élevé. Voyons ensemble le cheminement vers la liste de recommandation à l'aide d'un exemple. Nous repartons de notre matrice de notation implicite binaire de la figure 3.7.

La similarité cosinus entre articles est similaire à celle entre utilisateurs. Elle a néanmoins subi quelques modifications afin de pouvoir être calculée en fonction des articles. Elle s'exprime de la manière suivante :

$$Similarité\ Cosine(i, j) = \frac{\sum_{u \in U} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U} r_{ui}^2} \sqrt{\sum_{u \in U} r_{uj}^2}} \quad (3.3)$$

où U est l'ensemble des *users* ;

r_{ui} est l'évaluation du *user* u et pour l'*item* i .

La matrice 3.13 suivante reprend l'ensemble des scores de similarité entre articles.

1	0.666	0.333	0.577	0.408
0.666	1	0.333	0.577	0
0.333	0.333	1	0	0.816
0.577	0.577	0	1	0
0.408	0	0.816	0	1

Figure 3.13 Matrice de similarité cosinus entre articles

Pour rappel, nous ne prenons pas en compte la diagonale puisqu'il s'agit de la similarité d'un article avec lui-même. Nous pouvons remarquer que le deuxième film semble être le plus similaire au premier. Nous avons obtenu ce score de la manière suivante :

$$\sum_{u \in U} r_{u1} r_{u2} = 1 * 1 + 0 * 1 + 1 * 0 + 0 * 0 + 1 * 1$$

$$\sqrt{\sum_{u \in U} r_{u1}^2} = \sqrt{1^2 + 0^2 + 1^2 + 0^2 + 1^2}$$

$$\sqrt{\sum_{u \in U} r_{u2}^2} = \sqrt{1^2 + 1^2 + 0^2 + 0^2 + 1^2}.$$

$$\text{Nous donnant, } \frac{2}{\sqrt{3}\sqrt{3}} = 0.666$$

La matrice de similarité connue, nous sélectionnons, de la même manière que la méthode basée sur les utilisateurs, le nombre de plus proches voisins que nous désirons. Les deux plus proches voisins du premier film seront le film 2 et 4 puisqu'ils fournissent un score plus élevé que le reste des films. Cela signifie donc que nous utiliserons ces deux articles dans le calcul de nos prédictions pour le film 1.

La formule de calcul de prédiction pour un système de recommandation basé sur les articles est la suivante :

$$Prediction(u, i) = \tilde{r}_{ui} = \frac{\sum_{j \in N_i^u} Sim_{ij} * r_{uj}}{\sum_{j \in N_i^u} |Sim_{ij}|} \quad (3.4)$$

où $\tilde{r}_{ui}$ est la prédiction pour le user u et pour l'item i ;

r_{uj} est l'évaluation du user u et pour l'item j ;

N_i^u est l'ensemble des items les plus similaires à l'item i et qui ont été évalué par le user u ;

Sim_{ij} est la similarité entre l'item i et l'item j .

En appliquant cette formule à notre matrice d'exemple (figure 3.7) et aux résultats obtenus lors des calculs de similarité à la figure 3.13, nous obtenons les résultats présents à la figure 3.14.

-1	-1	0	1	0
0.535	-1	-1	0.5	0.667
-1	0.535	-1	0.5	-1
0	0	-1	0	-1
-1	-1	0	-1	0

Figure 3.14 Matrice des prédictions basée sur les articles

Nous remarquons une fois de plus que les films déjà visionnés par l'utilisateur se sont vu attribuer le score de -1 afin qu'ils ne soient plus recommandés. Décortiquons le score de l'utilisateur 2 pour le premier film. Celui-ci a obtenu un score de 0.535 de la manière suivante :

$$\sum_{j \in N_1^2} Sim_{1j} * r_{2j} = 0.666 * 1 + 0.577 * 0 = 0.666$$

$$\sum_{j \in N_1^2} |Sim_{1j}| = 0.666 + 0.577 = 1.243$$

$$r_{21} = 0.666/1.243 = 0.535$$

Les films ayant le score le plus élevé seront recommandés à l'utilisateur. Les listes de recommandations lorsque nous limitons le nombre de films à fournir à 2 se constituent comme présenté dans la figure 3.15.

<i>Utilisateurs</i>	<i>Films vus</i>	<i>Films recommandés</i>
1	1, 2	4, 3
2	2, 3	5, 1
3	1, 3, 5	2, 4
4	3, 5	1, 2
5	1, 2, 4	3, 5

Figure 3.15 Liste de recommandations sur base des articles

3.4.4. Système de recommandation basé sur le contenu

Enfin, le dernier type de système de recommandation sera basé sur le contenu. Les attributs pris en compte seront la date de parution des films ainsi que leurs genres. Ceux-ci seront alors utilisés afin de calculer la similarité entre les différentes productions.

Dans un premier temps, nous allons récupérer les données nécessaires extraites précédemment. Une fois que nous avons cette matrice à notre disposition. Nous calculons la mesure de similarité cosinus afin qu'elle nous fournisse un score de similarité d'un film par rapport à un autre film.

Si nous reprenons les données de la figure 3.7, nous obtenons alors la matrice 3.16 suivante :

1	0.667	0	0	0
0.667	1	0	0	0
0	0	1	0.816	0.288
0	0	0.816	1	0.353
0	0	0.288	0.353	1

Figure 3.16 Matrice de similarité entre films

Nous partons ensuite de cette matrice et de la matrice d'adjacence de la figure 3.7. Nous calculons alors un score pour chaque film et chaque utilisateur en fonction des films déjà vus par l'utilisateur. Comme pour les autres systèmes, ceux-ci auront une note de -1 afin de ne pas être recommandés. Notre matrice est présentée à la figure 3.17.

	-1	-1	0	0	0
0.333	-1	-1	0.408	0.144	
-1	0.222	-1	0.39	-1	
0	0	-1	0.585	-1	
-1	-1	0.272	-1	0.117	

Figure 3.17 Matrice des scores par film et utilisateur basés sur le contenu

Essayons de comprendre la détermination des scores. Si nous prenons le score de l'utilisateur 2 pour le film 4, nous voyons qu'il a obtenu le score de 0.408. Lorsque nous regardons dans la matrice d'adjacence, nous constatons que l'utilisateur 2 a visionné les films 2 et 3. Les lignes correspondantes à ces films dans la matrice de similarité sont exprimées en figure 3.18.

0.667	1	0	0	0
0	0	1	0.816	0.288

Figure 3.18 Lignes de la matrice de similarité utilisées pour le calcul des scores de l'utilisateur 2

Le 4^e film se trouve avoir un score de similarité de 0 par rapport au second film tandis qu'il obtient la note de 0.816 pour le troisième. Nous prenons alors la somme de la colonne correspondant au film d'intérêt. Ce qui fait donc $0 + 0.816 = 0.816$. Ensuite, nous divisons cette somme par le nombre de films que l'utilisateur a visionné. Nous donnant alors $\frac{0.816}{2} = 0.408$

Naturellement les films au score les plus élevés seront recommandés à l'utilisateur. Ceux-ci sont repris en figure 3.19.

Utilisateurs	Films vus	Films recommandés
1	1, 2	5, 4
2	2, 3	4, 1
3	1, 3, 5	4, 2
4	3, 5	4, 2
5	1, 2, 4	3, 5

Figure 3.19 Liste de recommandations sur base du contenu

3.5. Moyens d'évaluation

Une fois nos systèmes de recommandation opérationnels, ceux-ci vont nous fournir des recommandations de film pour nos différents utilisateurs. Étant donné que nous utilisons trois manières différentes pour le calcul de ces recommandations, il est fort probable que nos systèmes ne fournissent pas les mêmes résultats. C'est pour cela que nous devons les évaluer. Or, nous avons pu voir dans la revue de littérature que les métriques d'évaluation ne prennent pas en compte le « profil » des utilisateurs et se contentent uniquement de calculer la moyenne des métriques. L'objectif de ce mémoire est de voir l'impact de l'évaluation des systèmes de recommandation si nous prenons en compte certains éléments. C'est pour cela que nous utiliserons 4 méthodes d'évaluation.

La première se basera sur les métriques d'évaluation classique comme nous pouvons les retrouver dans la revue de littérature. Les suivantes attribueront un poids différent en fonction d'un critère en particulier afin de donner plus ou moins d'importance à la pertinence de la recommandation fournie.

Le nombre de notations globales sera un premier critère. Pour cette évaluation, nous accorderons un poids en fonction du nombre de notations des utilisateurs. L'importance de l'utilisateur variera alors avec celui-ci.

Nous ajouterons une dimension temporelle à notre évaluation en avantageant les utilisateurs encore actifs et dont les notations sont récentes. Les intervalles de dates « avant 2001 », « 2001-2005 », « 2005-2009 », « 2009-2013 », « plus de 2013 » ont été choisies afin de constituer nos groupes. Les utilisateurs de chaque groupe se verront attribuer un poids différent en fonction de la date de leur dernière notation.

Enfin, le dernier moyen d'évaluation prendra en compte la fréquence de notation d'un utilisateur. Nous partirons du postulat que la recommandation faite à un utilisateur ayant une fréquence de notation élevée sera plus importante que les autres.

3.5.1. Évaluation « classique »

Dans un premier temps et comme expliqué auparavant, nous allons évaluer nos systèmes de recommandation de manière classique en utilisant les métriques évoquées dans la revue de littérature. En l'occurrence, nous allons utiliser le Recall, la Précision, le nDCG, la distance utilisateur (« unexpectancy ») et l'ILD (distance intra-liste).

Rappelons-nous, tout d'abord, des formules des différentes métriques. Nous avons tout d'abord les métriques de précision avec la Précision et le Recall.

$$Précision = \frac{a}{a + b} = \frac{\text{Items correctement recommandés}}{\text{Ensemble des items recommandés}} \quad (3.5)$$

$$Recall = \frac{a}{a + c} = \frac{\text{Items correctement recommandés}}{\text{Ensemble des items pertinents pour l'utilisateur}} \quad (3.6)$$

Ces deux métriques étant des probabilités, au plus elles sont élevées, au plus notre système de recommandation est de qualité.

Nous utilisons le nDCG lorsque nous voulons intégrer l'information de la position de l'article recommandé. Cette fois encore, nous cherchons des valeurs élevées de nDCG pour traduire un bon système de recommandation. Voici les formules nécessaires pour le calculer :

$$\text{Gain cumulé actualisé} = DCG = \frac{1}{U} \sum_{u=1}^m \sum_{j \in I_u} \frac{g_{uj}}{\log_2(k_j + 1)} \quad (3.7)$$

où U est l'ensemble des utilisateurs ;

I_u l'ensemble de test ;

g_{uj} est l'utilité (ou l'évaluation) du *user* u pour l'*item* j ;

k_j est la position de l'*item* j dans l'ensemble de test.

$$\text{Gain cumulé actualisé normalisé} = NDCG = \frac{DCG}{IDCG} \quad (3.8)$$

Avec un $IDCG$ correspondant à la valeur idéale du DCG .

Afin de mesurer la notion de nouveauté, nous utilisons la formule suivante dans le but de calculer la distance des items proposés avec ceux déjà consommés par l'utilisateur. Une valeur élevée traduira une nouveauté élevée également.

$$Unexp = \frac{1}{|R||I_u|} \sum_{i \in R} \sum_{j \in I_u} d(i, j) \quad (3.9)$$

où R est une recommandation à un utilisateur ;
 I_u est l'ensemble des *items* avec lesquels le *user* u a interagi ;
 $d(i, j)$ est la distance entre l'*item* i et l'*item* j .

Enfin, nous calculons la diversité grâce à la dissimilarité moyenne ou distance intra liste moyenne (ILD). Ici encore, il est désirable d'obtenir une valeur élevée.

$$\text{Dissimilarité moyenne} = \frac{1}{|R_u| (|R_u| - 1)} \sum_{i \in R_u} \sum_{j \in R_u, j \neq i} d(i, j) \quad (3.10)$$

où R_u est la liste de recommandation proposée au *user* u ;
 $d(i, j)$ est la distance entre les *items* i et j .

Afin de pouvoir tester nos modèles et de voir s'ils sont capables de prédire pour des données non connues, nous utiliserons la validation croisée. Elle consiste à échantillonner notre base de données en deux sous-ensembles de façon aléatoire. Ceux-ci constitueront l'ensemble d'apprentissages, utilisé pour la conception des modèles, et l'ensemble test, nous permettant d'évaluer leur efficacité. Dans notre cas, nous utiliserons la technique du « sous-échantillonnage aléatoire k-fold ». Méthode qui répète l'opération un nombre k de fois et de prendre la moyenne des performances (Berrar, 2018). Nous avons choisi un k égal à 10.

Voyons avec un exemple en partant de notre matrice en figure 3.7. Celle-ci est alors divisée en une matrice d'entraînement et une matrice de test que l'on peut retrouver respectivement en figure 3.20 et 3.21.

1	0	0	0	0
0	1	1	0	0
1	0	0	0	1
0	0	1	0	1
0	1	0	1	0

Figure 3.20 Matrice d'entraînement sur base de la figure 3.7

0	1	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	0

Figure 3.21 Matrice de test sur base de la figure 3.5

Nous allons ensuite entrainer notre modèle de référence basé sur les utilisateurs avec la matrice d'entraînement et le tester avec la matrice de test. Les recommandations produites se trouvent en figure 3.22.

Utilisateurs	Films vus	Films recommandés
1	1	5, 2
2	2, 3	4, 5
3	1, 5	3, 2
4	3, 5	1, 2
5	1, 4	3, 1

Figure 3.22 Recommandation pour la matrice d'entraînement

Sur base de ces recommandations, nous pouvons calculer les métriques précédemment expliquées dans le tableau 3.1 suivant.

Métriques	Calculs	Résultats
Recall	$\frac{1 + 1 + 1}{3}$	1
Précision	$\frac{1 + 1 + 1}{10}$	0.3
nDCG	$\frac{1}{5} * (\frac{0}{\log_2(1+1)} + \frac{1}{\log_2(2+1)} + \frac{0}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{1}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{0}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{1}{\log_2(1+1)} + \frac{0}{\log_2(2+1)})$	0.4523
Unexp	$\frac{1}{5} * (\frac{0.5 + 1}{2 * 1} + \frac{0.3 + 1 + 1 + 0.5}{2 * 2} + \frac{1 + 1 + 0.5 + 1}{2 * 2} + \frac{1 + 0.5 + 1 + 0.5}{2 * 2} + \frac{1 + 1 + 0 + 1}{2 * 2})$	0.765
ILD	$\frac{1}{5} * (\frac{1 + 1}{2 * (2 - 1)} + \frac{1 + 1}{2 * (2 - 1)} + \frac{0.5 + 0.5}{2 * (2 - 1)} + \frac{1 + 1}{2 * (2 - 1)} + \frac{1 + 1}{2 * (2 - 1)})$	0.9

Tableau 3.1 Calcul des mesures d'évaluation classiques

3.5.2. Évaluation utilisant le nombre de ratings

La deuxième évaluation se fera en fonction du nombre de notations total qui a été effectué par les utilisateurs. Nous avons alors calculé le nombre de ratings par utilisateurs et créé une liste « importance_nb_ratings » que nous avons remplie avec les poids accordés à ceux-ci en fonction de leur profil.

Pour l'attribution des scores d'importance, nous avons, dans un premier temps, gardé le nombre de ratings comme score afin de conserver une proportionnalité dans les scores. Ainsi, un utilisateur ayant noté 10 fois plus de films qu'un autre aura 10 fois plus d'importance que ce dernier. Ensuite, nous avons créé une deuxième liste d'importance dans laquelle ces scores ont été mis au carré afin de donner beaucoup plus d'importance aux utilisateurs que nous considérons comme les plus pertinents.

Une fois cette information en notre possession, nous avons pu calculer nos métriques en l'intégrant et ajustant nos calculs. De ce fait, pour le Recall, le nombre d'articles noté et recommandé, ainsi que le nombre d'articles noté et non recommandé ont été multipliés par l'importance donnée à l'utilisateur en question et ensuite divisée par la somme de tous les scores d'importance attribués aux users. Avec cette méthode, cela nous permet d'avantager les personnes ayant noté un grand nombre de films et de garder un Recall identique à la méthode classique si nous accordons la même importance à chaque utilisateur. Une méthode similaire a été appliquée pour la Précision.

Pour le nDCG, la distance utilisateur (unexpectancy) et l'ILD, leur score respectif a été calculé pour chaque utilisateur et a été pondéré par la valeur de l'importance de l'utilisateur avant d'être incorporé dans la métrique globale. Ensuite, celle-ci a été divisée par la somme de l'ensemble des scores d'importance.

Dans notre exemple, la liste d'importance se présente comme en figure 3.23.

1 2 2 2 2

Figure 3.23 Liste d'importance de l'exemple selon le nombre de notations

Notre calcul des métriques d'évaluation sera adapté de la façon présenté dans le tableau 3.2 dans le cas de la méthode linéaire.

Métriques	Calculs	Résultats
Recall	$\frac{1 * \frac{1}{9} + 1 * \frac{2}{9} + 1 * \frac{2}{9}}{1 * \frac{1}{9} + 1 * \frac{2}{9} + 1 * \frac{2}{9}}$	1
Précision	$\frac{1 * \frac{1}{9} + 1 * \frac{2}{9} + 1 * \frac{2}{9}}{2 * \frac{1}{9} + 2 * \frac{2}{9} + 2 * \frac{2}{9} + 2 * \frac{2}{9} + 2 * \frac{2}{9}}$	0.277
nDCG	$\begin{aligned} &\frac{1}{9} * (\frac{0}{\log_2(1+1)} * 1 + \frac{1}{\log_2(2+1)} * 1 + \frac{0}{\log_2(1+1)} * 2 \\ &+ \frac{0}{\log_2(2+1)} * 2 + \frac{1}{\log_2(1+1)} * 2 + \frac{0}{\log_2(2+1)} * 2 \\ &+ \frac{0}{\log_2(1+1)} * 2 + \frac{0}{\log_2(2+1)} * 2 + \frac{0}{\log_2(1+1)} * 2 \\ &+ \frac{1}{\log_2(2+1)} * 2) \end{aligned}$	0.432
Unexp	$\begin{aligned} &\frac{1}{9} * (\frac{0.5+1}{2*1} * 1 + \frac{0.3+1+1+0.5}{2*2} * 2 + \frac{1+1+0.5+1}{2*2} * 2 \\ &+ \frac{1+0.5+1+0.5}{2*2} * 2 + \frac{1+1+0+1}{2*2} * 2) \end{aligned}$	0.766
ILD	$\begin{aligned} &\frac{1}{9} * (\frac{1+1}{2*(2-1)} * 1 + \frac{1+1}{2*(2-1)} * 2 + \frac{0.5+0.5}{2*(2-1)} * 2 + \frac{1+1}{2*(2-1)} * 2 \\ &+ \frac{1+1}{2*(2-1)} * 2 \end{aligned}$	0.888

Tableau 3.2 Calcul des mesures d'évaluation incorporant l'importance des utilisateurs en fonction de leur nombre de notations

Si nous avons procédé à la méthode quadratique, les scores de la figure 3.23. auraient été mis au carré, impactant à leur tour les métriques d'évaluations.

3.5.3. Évaluation utilisant la date du dernier rating

L'évaluation utilisant la date de la dernière notation va venir inclure une information temporelle à nos calculs. En effet, jusqu'ici, l'ensemble des notations étaient considérées comme ayant la même pertinence qu'elles aient été faites il y a 4 ans ou 20 ans. Or, il serait judicieux d'avantager les utilisateurs ayant fourni des notations récentes, puisque plus représentatifs des goûts actuels. De plus, cela permet d'accorder plus d'importance aux utilisateurs encore actifs.

Pour ce faire, nous avons extrait l'information « timestamp » de notre base de données. Nous avons pu alors déterminer la date et l'heure exacte de chaque notation. Pour des soucis de facilité, nous ne considérerons dans ce mémoire que les années des notations. Une liste d'importance intitulée « importance_timestamp » a été créée reprenant les scores attribués à chaque utilisateur.

Pour l'attribution des scores d'importance, nous avons décidé de diviser nos utilisateurs en 5 groupes. Chaque groupe se voyant attribuer un score de 1 à 5 dans la version linéaire d'attribution des scores. Le groupe d'utilisateurs ayant évalué des films récemment se verra attribuer un score de 5 tandis que les utilisateurs avec des notations anciennes auront un score de 1. En pratique, le groupe « avant 2001 » aura un score de 1, celui de « 2001-2005 » un score de 2, « 2005-2009 » un score de 3, « 2009-2013 » un score de 4 et enfin les « plus de 2013 » auront le score maximal de 5. La seconde méthode de conception des listes d'importance transformera les scores de 1 à 5 en score de 1 à 25. Une fois encore, l'objectif est de pénaliser grandement les utilisateurs non actifs et de favoriser les utilisateurs encore actifs récemment.

Une fois la liste d'importance en notre possession, nous procédons de la même manière que précédemment pour le calcul du Recall. À savoir, nous multiplions le coefficient d'importance d'un utilisateur avec son nombre d'articles noté et recommandé que nous divisons par la somme de tous les scores attribués. Nous faisons de même avec le nombre d'articles noté et non recommandé. La Précision utilise la même logique.

De la même manière que précédemment, les métriques nDCG, distance utilisateur et ILD sont déterminées pour chaque utilisateur en prenant en compte l'importance de celui-ci. Le résultat final est ensuite remis sur la somme de l'ensemble des scores d'importance.

Imaginons que les dates de dernières notations de nos utilisateurs soient 2008, 2012, 2004, 2014 et 1999. Nous donnons, dans le cas de la méthode linéaire, une liste d'importance ressemblant à celle de la figure 3.24.

3 4 2 5 1

Figure 3.24 Liste d'importance de l'exemple selon la date de leur dernière notation

Le tableau des métriques d'évaluation correspondant est le tableau 3.3 suivant.

Métriques	Calculs	Résultats
Recall	$\frac{1 * \frac{3}{15} + 1 * \frac{2}{15} + 1 * \frac{1}{15}}{1 * \frac{3}{15} + 1 * \frac{2}{15} + 1 * \frac{1}{15}}$	1
Précision	$\frac{1 * \frac{3}{15} + 1 * \frac{2}{15} + 1 * \frac{1}{15}}{2 * \frac{3}{15} + 2 * \frac{4}{15} + 2 * \frac{2}{15} + 2 * \frac{5}{15} + 2 * \frac{1}{15}}$	0.2
nDCG	$\begin{aligned} & \frac{1}{15} * (\frac{0}{\log_2(1+1)} * 3 + \frac{1}{\log_2(2+1)} * 3 + \frac{0}{\log_2(1+1)} * 4 \\ & + \frac{0}{\log_2(2+1)} * 4 + \frac{1}{\log_2(1+1)} * 2 + \frac{0}{\log_2(2+1)} * 2 \\ & + \frac{0}{\log_2(1+1)} * 5 + \frac{0}{\log_2(2+1)} * 5 + \frac{0}{\log_2(1+1)} * 1 \\ & + \frac{1}{\log_2(2+1)} * 1) \end{aligned}$	0.302
Unexp	$\begin{aligned} & \frac{1}{15} * (\frac{0.5+1}{2*1} * 3 + \frac{0.3+1+1+0.5}{2*2} * 4 + \frac{1+1+0.5+1}{2*2} * 2 \\ & + \frac{1+0.5+1+0.5}{2*2} * 5 + \frac{1+1+0+1}{2*2} * 1) \end{aligned}$	0.753
ILD	$\begin{aligned} & \frac{1}{15} * (\frac{1+1}{2*(2-1)} * 3 + \frac{1+1}{2*(2-1)} * 4 + \frac{0.5+0.5}{2*(2-1)} * 2 + \frac{1+1}{2*(2-1)} \\ & * 5 + \frac{1+1}{2*(2-1)} * 1) \end{aligned}$	0.933

Tableau 3.3 Calcul des mesures d'évaluation incorporant l'importance des utilisateurs en fonction de la date de leur dernière notation

3.5.4. Évaluation utilisant la fréquence des ratings

Pour la dernière évaluation, nous combinons en quelques sortes les deux moyens utilisés précédemment. En effet, nous utilisons l'information du nombre de ratings fournis par les utilisateurs mais également la temporalité de ces notations. Afin de calculer la fréquence de notation, nous avons utilisé, comme pour déterminer la date du dernier rating, la colonne « timestamp ». En plus de la date du dernier rating, nous avons alors déterminé la date de la première notation. Grâce à ces deux dates, nous pouvons alors déduire la période d'activité de chaque utilisateur. La fréquence sera calculée en divisant le nombre de notations par cette période en jours. Nous donnant alors le nombre de films notés par l'utilisateur par jour.

L'attribution des scores se fera en fonction de la valeur de la fréquence. Nous attribuerons une importance accrue en cas de fréquence élevée et inversement en cas de fréquence faible. C'est pour cela que la version linéaire de conception des listes d'importance garde la fréquence de

ratings comme score afin de conserver une proportionnalité dans les scores. Une fonction quadratique sera utilisée dans la deuxième version.

Lorsque la liste d'importance est générée, nous pouvons alors calculer le Recall, la Précision, le nDCG, la distance utilisateur et le ILD de la même manière que pour les deux autres méthodes.

Si les fréquences de notation de nos utilisateurs sont respectivement 3.5, 7, 1.2, 4.8 et 5.1. La liste d'importance linéaire se présentera comme en figure 3.25.

3.5 7 1.2 4.8 5.1

Figure 3.25 Liste d'importance de l'exemple selon la fréquence de leurs notations

Nous donnant le tableau 3.4 de métriques suivant :

Métriques	Calculs	Résultats
Recall	$\frac{1 * \frac{3.5}{21.6} + 1 * \frac{1.2}{21.6} + 1 * \frac{5.1}{21.6}}{1 * \frac{3.5}{21.6} + 1 * \frac{1.2}{21.6} + 1 * \frac{5.1}{21.6}}$	1
Précision	$\frac{1 * \frac{3.5}{21.6} + 1 * \frac{1.2}{21.6} + 1 * \frac{5.1}{21.6}}{2 * \frac{3.5}{21.6} + 2 * \frac{7}{21.6} + 2 * \frac{1.2}{21.6} + 2 * \frac{4.8}{21.6} + 2 * \frac{5.1}{21.6}}$	0.226
nDCG	$\begin{aligned} & \frac{1}{21.6} * (\frac{0}{\log_2(1+1)} * 3.5 + \frac{1}{\log_2(2+1)} * 3.5 + \frac{0}{\log_2(1+1)} * 7 \\ & + \frac{0}{\log_2(2+1)} * 7 + \frac{1}{\log_2(1+1)} * 1.2 + \frac{0}{\log_2(2+1)} \\ & * 1.2 + \frac{0}{\log_2(1+1)} * 4.8 + \frac{0}{\log_2(2+1)} * 4.8 \\ & + \frac{0}{\log_2(1+1)} * 5.1 + \frac{1}{\log_2(2+1)} * 5.1) \end{aligned}$	0.306
Unexp	$\begin{aligned} & \frac{1}{21.6} * (\frac{0.5+1}{2*1} * 3.5 + \frac{0.3+1+1+0.5}{2*2} * 7 + \frac{1+1+0.5+1}{2*2} * 1.2 \\ & + \frac{1+0.5+1+0.5}{2*2} * 4.8 + \frac{1+1+0+1}{2*2} * 5.1) \end{aligned}$	0.741
ILD	$\begin{aligned} & \frac{1}{21.6} * (\frac{1+1}{2*(2-1)} * 3.5 + \frac{1+1}{2*(2-1)} * 7 + \frac{0.5+0.5}{2*(2-1)} * 1.2 \\ & + \frac{1+1}{2*(2-1)} * 4.8 + \frac{1+1}{2*(2-1)} * 5.1 \end{aligned}$	0.972

Tableau 3.4 Calcul des mesures d'évaluation incorporant l'importance des utilisateurs en fonction de la fréquence de notation

3.6. Conclusion

Dans cette partie du mémoire, nous avons pu voir comment les analyses de la partie suivante ont pu être réalisées. Nous avons vu quelles bases de données ont été utilisées ainsi que les outils et logiciels nécessaires à la création des différents modèles. Nous avons également pu comprendre plus en profondeur comment les algorithmes des différents systèmes de recommandation fonctionnaient pour nous donner une liste de recommandation finale. Enfin, nous avons détaillé comment nous comptons évaluer nos modèles tant de façon classique que par l'attribution de poids lié à l'importance d'un profil d'utilisateur. Cette dernière partie est cruciale pour la bonne compréhension des analyses puisqu'elle nous permettra de comparer les classements d'efficacité des systèmes de recommandation entre les méthodes classiques et celles introduites dans le cadre de ce mémoire. Nous verrons dans la partie suivante les résultats produits par nos systèmes de recommandation, ainsi que les analyses et conclusions que nous pouvons en retirer.

4. Analyse des résultats

4.1. Introduction

Cette partie du mémoire va présenter les résultats produits par nos systèmes de recommandations conçus de la manière décrite dans la méthodologie. Nous pourrons ensuite voir les différentes métriques d'évaluation et ainsi voir le classement de nos systèmes du moins bon au meilleur.

Nous pourrons ensuite tenter de répondre à notre question de recherche qui est, pour rappel, l'analyse de l'utilisation des indices moyens pour évaluer un système de recommandation. De ce fait, nous serons dans la possibilité de voir l'impact de l'introduction d'un système de pondération dans l'évaluation des systèmes de recommandation et donc de la prise en compte du profil des utilisateurs dans notre évaluation. Nous utiliserons deux sortes de pondération différentes. La première sera une pondération linéaire, c'est-à-dire qu'un utilisateur aura une importance proportionnelle au critère d'intérêt. Par exemple, s'il a évalué 10 fois plus de films qu'un autre, il aura alors un poids 10 fois plus important. La deuxième pondération sera quadratique. Dans ce cas, nous accordons une importance beaucoup plus importante aux utilisateurs que nous considérons comme intéressants puisque nous mettrons au carré les poids présentés dans la version linéaire.

4.2. Prises de décisions concernant les algorithmes

Les résultats présentés dans la partie 4 de ce mémoire sont tous le fruit des algorithmes présents en annexe 1. Ceux-ci exécutent et appliquent les méthodes comme expliquées dans la méthodologie. Cependant certaines décisions restent à être prises afin de pouvoir faire tourner nos codes. En effet, deux paramètres présents dans certaines de nos fonctions doivent encore être déterminés.

Le premier concerne le nombre de recommandations à fournir. Nous avons alors décidé de créer des listes de recommandations de longueur 10, 20 et 50. Nous voulions, en outre, voir l'effet qu'une variation du nombre d'articles proposés aurait sur nos métriques mais également prendre en compte le fait que certains utilisateurs ne regarderont que très peu de recommandations. Tandis que d'autres parcourront une liste un peu plus longue. Nous avons finalement choisi une liste de recommandation de 20 articles puisqu'elle constitue un bon

compromis entre les mesures de précision et au-delà de la précision. L'ensemble des résultats qui nous ont permis d'arriver à cette conclusion se trouvent en annexe 2.

Le deuxième paramètre concerne le nombre de voisins que nos systèmes de filtrages collaboratifs devront prendre en compte dans le calcul des prédictions. Afin de déterminer ce nombre, nous avons procédé à plusieurs expérimentations. Notons que nous avons travaillé avec une base de données plus petite dans un souci de facilité et de gain de temps. Notre base de données était alors composée de 107 utilisateurs et 143 films, tous ayant plus de 10 notations. La valeur du nombre de voisins, k , a subi des variations ce qui nous permettra de voir pour quelle valeur nos algorithmes sont les plus performants. Voyons ce qu'il en est pour le système user based dans le tableau 4.1.

<i>Métriques</i>	K=5	K=10	K=15	K=20	K=25
<i>nDCG</i>	0.32	0.352	0.365	0.37	0.368
<i>Recall</i>	0.57	0.6	0.612	0.62	0.618
<i>Précision</i>	0.085	0.091	0.094	0.095	0.095
<i>ILD</i>	0.647	0.625	0.612	0.605	0.598

Tableau 4.1 Variation de k pour le système basé sur les utilisateurs

Nous voyons avec le tableau ci-dessus que les mesures de précisions sont à leur maximum si nous utilisons un voisinage de 20 utilisateurs.

Analysons la situation pour le système basé sur les articles grâce aux résultats du tableau 4.2.

<i>Métriques</i>	K=5	K=10	K=15	K=20	K=25
<i>nDCG</i>	0.28	0.3	0.3	0.3	0.28
<i>Recall</i>	0.483	0.505	0.512	0.515	0.513
<i>Précision</i>	0.075	0.078	0.079	0.079	0.078
<i>ILD</i>	0.664	0.654	0.656	0.659	0.663

Tableau 4.2 Variation de k pour le système basé sur les articles

De manière équivalente, nous constatons qu'un nombre de 20 voisins fait en sorte que notre modèle performe mieux. Nous pouvons donc conclure qu'il est pertinent de choisir un k égal à 20, ce que nous appliquerons dans l'implémentation de nos algorithmes.

4.3. Présentation des résultats

Voyons ensemble les résultats obtenus pour nos différents systèmes de recommandation. Ceux-ci sont divisés en 2 catégories : les résultats obtenus lorsque la liste d'importance est calculée de manière linéaire et lorsqu'elle est déterminée de façon quadratique. Il est important de noter que les valeurs présentées dans l'ensemble de nos tableaux correspondent à la moyenne des valeurs obtenues lors des 10 validations croisées. Pour rappel, les analyses ont été réalisées sur 2 bases de données différentes. De plus, trois critères différents ont été utilisés pour la conception des listes d'importance.

4.3.1. nDCG

Commençons par la métrique de nDCG. Pour la métrique classique, nous avons utilisé la formule 2.15 présentée dans la revue de littérature. Nous avons ensuite adapté le calcul en incluant les listes d'importance. Cette métrique est plus désirable lorsqu'elle est élevée puisqu'elle nous permet de déterminer la pertinence des articles recommandés tout en prenant en compte la position de ceux-ci dans la liste.

Linéaire

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>nDCG classique</i>	0.224	0.179	0.040	0.214	0.175	0.017
<i>nDCG importance fréquence</i>	0.213	0.185	0.039	0.219	0.185	0.020
<i>nDCG importance nb ratings</i>	0.209	0.176	0.040	0.211	0.178	0.020
<i>nDCG importance dernier rating</i>	0.200	0.142	0.032	0.192	0.144	0.012

Tableau 4.3 Résultats pour les nDCGs calculés linéairement

Nous constatons dans le tableau 4.3 que peu importe la base de données ou la manière de prendre en compte les différents individus, le système de filtrage collaboratif user based reste le meilleur. Le nDCG classique est le plus élevé pour notre première base de données tandis qu'il s'agit du nDCG incluant l'importance des fréquences de notations dans la deuxième. Cela implique donc que pour notre deuxième base de données, les utilisateurs ayant une plus grande

fréquence se voient proposer des recommandations plus performantes en termes de nDCG que les utilisateurs lambdas. Pour la première base de données, les utilisateurs d'importance produisent de moins bons résultats en termes de nDCG que les autres utilisateurs, faisant ainsi baisser les scores de la métrique. La seule exception se produit pour l'item based qui suit la logique de la deuxième base de données.

Quadratique

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>nDCG classique</i>	0.224	0.179	0.040	0.214	0.175	0.017
<i>nDCG importance fréquence</i>	0.187	0.177	0.034	0.204	0.189	0.025
<i>nDCG importance nb ratings</i>	0.202	0.176	0.051	0.203	0.177	0.029
<i>nDCG importance dernier rating</i>	0.194	0.127	0.030	0.186	0.130	0.010

Tableau 4.4 Résultats pour les nDCGs calculés quadratiquement

Le passage à la méthode quadratique (tableau 4.4) ne semble pas influencer les observations quant au classement de nos systèmes de recommandation. En effet, le système basé sur les utilisateurs (user based) conserve sa première place dans tous les cas de figure. La valeur la plus élevée pour notre première base de données se trouve encore être notre nDCG classique.

Cela traduit que les utilisateurs d'intérêts moindres faussent légèrement les scores en surévaluant la valeur du nDCG. Néanmoins, même si la version utilisant la fréquence de notation reste meilleure que la métrique classique pour les deux derniers systèmes de notre deuxième base de données, ce n'est pas le cas pour le user based qui produit de meilleurs résultats avec la méthode classique.

Remarquons également que le passage de la méthode linéaire à la méthode quadratique fait baisser presque l'intégralité de nos résultats. Accorder plus d'importance aux utilisateurs nous intéressant fait donc baisser le nDCG de nos recommandations. Nous pouvons donc déduire que les utilisateurs lambdas reçoivent de meilleures recommandations en termes de nDCG que les utilisateurs que nous souhaitons privilégier.

4.3.2. Recall

Voyons maintenant ce qu'il en est pour la mesure de Recall. Pour rappel, nous utilisons comme base la formule de Recall classique 2.12 de la revue de littérature. Nous cherchons, ici encore, des valeurs de métrique supérieures puisque le Recall représente la probabilité qu'un élément pertinent soit recommandé.

Linéaire

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>Recall classique</i>	0.188	0.151	0.039	0.195	0.161	0.020
<i>Recall importance fréquence</i>	0.166	0.145	0.036	0.166	0.148	0.022
<i>Recall importance nb ratings</i>	0.131	0.108	0.032	0.141	0.118	0.020
<i>Recall importance dernier rating</i>	0.166	0.120	0.033	0.178	0.133	0.016

Tableau 4.5 Résultats pour les Recalls calculés linéairement

Notre méthode de calcul de performance linéaire (tableau 4.5) nous place le système de recommandation user based en première position. La métrique classique fournit de meilleurs résultats quant à la performance des différents modèles. Tout comme pour le nDCG, cela veut dire que les utilisateurs peu importants faussent les résultats de Recall en le surévaluant dans la méthode classique.

Quadratique

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>Recall classique</i>	0.188	0.151	0.039	0.195	0.161	0.020
<i>Recall importance fréquence</i>	0.132	0.121	0.028	0.121	0.124	0.020
<i>Recall importance nb ratings</i>	0.112	0.096	0.033	0.119	0.101	0.022

<i>Recall</i>	<i>importance</i>	0.158	0.108	0.032	0.172	0.119	0.013
<i>dernier rating</i>							

Tableau 4.6 Résultats pour les Recalls calculés quadratiquement

En observant le tableau 4.6, nous pouvons tirer les mêmes conclusions pour notre première base de données pour la méthode quadratique que pour la méthode linéaire. À savoir, le système user based conserve sa première place au classement avec une performance plus élevée pour la métrique classique. Cependant, nous constatons un changement dans les résultats basés sur notre deuxième base de données. En effet, si le user based reste supérieur aux deux autres modèles pour la version classique, la version utilisant l'importance du nombre de notations et l'importance de la dernière notation, ce n'est pas le cas pour la métrique incluant l'importance de la fréquence dans son calcul. Nous constatons que celle-ci privilégie le système de recommandation de filtrage collaboratif item based.

Nous nous sommes donc attardés sur cette situation afin de voir si le changement entre un système basé sur les utilisateurs pour un système basé sur les articles était réellement pertinent. Nous avons alors réalisé un test de Student avec un alpha à 5% afin de voir si les moyennes de Recall de nos deux systèmes étaient significativement différentes. Ce test nous a alors révélé que ce n'était pas le cas. En effet, comme le montre la figure 4.1 suivante, nous ne parvenons pas à rejeter l'hypothèse nulle stipulant que nos moyennes sont égales. Les détails de ce test de Student sont disponibles en annexe 3.

<i>Métrique</i>	<i>p-value</i>
<i>Recall importance fréquence</i>	0.4061

Figure 4.1 Test de Student de comparaison de moyennes de Recall

Une fois encore, l'utilisation d'une fonction quadratique dans l'élaboration de nos listes d'importance a pour effet de faire baisser la quasi-totalité des résultats. Les utilisateurs d'importance reçoivent donc des recommandations moins performantes en termes de Recall que les utilisateurs lambdas ce qui s'exprime dans une valeur de métrique plus élevée dans sa version classique.

4.3.3. Précision

Observons les résultats pour notre métrique de précision. La formulation classique de cette métrique est présentée à l'équation 2.11 de la revue de littérature. De la même manière que pour

les deux métriques précédentes, nous recherchons des valeurs élevées puisqu'elle représente la probabilité qu'un élément recommandé soit pertinent.

Linéaire

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	UB	IB	CB	UB	IB	CB
<i>Précision classique</i>	0.125	0.100	0.026	0.119	0.098	0.012
<i>Précision importance fréquence</i>	0.146	0.127	0.031	0.164	0.146	0.022
<i>Précision importance nb ratings</i>	0.277	0.227	0.067	0.255	0.214	0.036
<i>Précision importance dernier rating</i>	0.127	0.092	0.026	0.122	0.091	0.011

Tableau 4.7 Résultats pour les Précisions calculées linéairement

Le système de recommandation user based se trouve une fois encore au premier rang du classement que ce soit pour la première base de données ou pour la deuxième comme le montre le tableau 4.7. Une observation assez importante est que, contrairement à ce que nous avons pu observer jusqu'ici, les valeurs des métriques ne sont plus les plus élevées pour la version classique. En effet, la version classique se trouve être l'une des plus basses lorsqu'on la compare avec ses homologues privilégiant les individus d'intérêts. Notons même que la précision privilégiant les individus ayant un grand nombre de notations est dans tous les cas plus de deux fois supérieure à la métrique classique. Ce phénomène traduit une précision nettement supérieure pour les individus ayant fourni un grand nombre de notations.

Quadratique

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	UB	IB	CB	UB	IB	CB
<i>Précision classique</i>	0.125	0.100	0.026	0.119	0.098	0.012
<i>Précision importance fréquence</i>	0.200	0.183	0.042	0.274	0.283	0.046
<i>Précision importance nb ratings</i>	0.463	0.398	0.136	0.415	0.352	0.078

<i>Précision dernier rating</i>	<i>importance</i>	0.129	0.088	0.026	0.124	0.086	0.010
-------------------------------------	-------------------	--------------	-------	-------	--------------	-------	-------

Tableau 4.8 Résultats pour les Précisions calculées quadratiquement

Les résultats produits par la version quadratique de la précision (tableau 4.8) sont similaires à ceux du Recall pour la même version. Nous observons toujours de meilleurs résultats pour le user based à l'exception de la précision prenant en compte l'importance de la fréquence qui produit un résultat plus élevé pour l'item based pour notre deuxième base de données.

De la même manière que pour le Recall, attardons-nous sur cette situation en figure 4.2. Nous avons réalisé un nouveau test de Student, cette fois-ci sur la comparaison des moyennes des précisions des systèmes user based et item based. L'hypothèse nulle était alors que les moyennes n'étaient pas significativement différentes à un seuil de 5%. La conclusion du test est que nous ne rejetons pas cette hypothèse nulle et donc que celles-ci ne sont pas significativement différentes. Les détails de ce test de Student sont disponibles en annexe 3.

<i>Métrique</i>	<i>p-value</i>
<i>Précision importance fréquence</i>	0.4644

Figure 4.2 Test de Student de comparaison de moyennes de Précision

Observons les valeurs de la Précision lorsque nous prenons en compte le nombre de notations laissé par les utilisateurs. Nous constatons que le facteur de multiplication par rapport à la version classique est encore plus grand que pour la version linéaire. En effet, nous observons une multiplication par près de quatre pour presque tous les systèmes et jusqu'à six pour le système basé sur le contenu de la deuxième base de données. Cela confirme donc nos dires lorsque nous affirmions que la précision était grandement meilleure pour nos utilisateurs d'intérêt en termes de nombre de notation. La méthode utilisant la fréquence de notation fournit également de meilleurs résultats que la version classique. L'utilisation d'un poids identique pour tout le monde fausse donc quelque peu la probabilité qu'un élément recommandé soit pertinent (représenté par la précision) en la sous-évaluant.

4.3.4. F-measure

La mesure suivante vient incorporer les deux métriques précédemment analysées, le Recall et la Précision, afin de créer une nouvelle mesure décrite par la formule 2.13. Il s'agit ici d'une sorte de compromis entre les deux mesures. Comme pour ces dernières, une valeur élevée de F-measure est désirable.

Linéaire

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>F-measure classique</i>	0.150	0.120	0.031	0.148	0.122	0.015
<i>F-measure importance fréquence</i>	0.155	0.136	0.033	0.165	0.147	0.022
<i>F-measure importance nb ratings</i>	0.178	0.147	0.043	0.181	0.152	0.025
<i>F-measure importance dernier rating</i>	0.144	0.104	0.029	0.145	0.108	0.012

Tableau 4.9 Résultats pour les F-measures calculées linéairement

Les résultats des F-measures de la version linéaire se trouvent dans le tableau 4.9. L'observation habituelle pour nos tableaux contenant les mesures linéaires semble être encore d'actualité puisque nous retrouvons le système user based comme étant le plus performant pour l'ensemble de nos configurations. Les résultats obtenus par les métriques classiques se trouvent une nouvelle fois être plus bas que lorsque nous introduisons les fréquences ou le nombre de notations des utilisateurs. Les utilisateurs ayant une plus grande importance dans ces critères produisent donc une F-measure supérieur aux autres utilisateurs.

Quadratique

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>F-measure classique</i>	0.150	0.120	0.031	0.148	0.122	0.015
<i>F-measure importance fréquence</i>	0.159	0.146	0.034	0.167	0.172	0.028
<i>F-measure importance nb ratings</i>	0.181	0.156	0.053	0.185	0.157	0.035
<i>F-measure importance dernier rating</i>	0.142	0.097	0.028	0.144	0.108	0.011

Tableau 4.10 Résultats pour les F-measures calculés quadratiquement

Sans grande surprise, nous constatons au tableau 4.10 un comportement identique que pour le Recall et la Précision quadratique. Le système à privilégier continue d'être le user based dans le cas de notre première ainsi que notre deuxième base de données. Une seule exception apparaît

lorsque nous évaluons nos systèmes avec les fréquences de notation pour notre deuxième base de données. Dans ce cas, le système item based semble mieux performer.

Un test de Student a de nouveau été effectué afin de voir la pertinence de ce changement. Nous échouons à rejeter l'hypothèse nulle qui stipulait que les moyennes des F-measures sont identiques puisque nous obtenons une p-value largement supérieur à notre seuil de 5% (figure 4.3). Les détails de ce test sont disponibles en annexe 3.

<i>Métrique</i>	<i>p-value</i>
<i>F-measure importance fréquence</i>	0.4094

Figure 4.3 Test de Student de comparaison de moyennes de F-measure

La même constatation est observable que pour le modèle linéaire, à savoir, une augmentation des métriques lorsque nous attribuons plus de poids en fonction de la fréquence ou du nombre de notations. Les mêmes conclusions sont alors à tirer, à savoir que la version classique considérant les utilisateurs comme égaux, sous-évalue la valeur de F-measure. Cela implique que les utilisateurs lambdas produisent une F-measure moindre que celle des utilisateurs d'intérêts pour ces deux critères.

4.3.5. Dissimilarité moyenne ou distance intra-liste (ILD)

La dissimilarité moyenne, ou distance intra-liste, comme exprimée par l'équation 2.19, nous permet d'estimer la diversité présente dans les recommandations de nos modèles. Une diversité élevée est attendue afin de proposer à nos utilisateurs des articles variés.

Linéaire

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>ILD classique</i>	0.577	0.639	0.805	0.593	0.644	0.83
<i>ILD importance fréquence</i>	0.579	0.646	0.805	0.602	0.648	0.831
<i>ILD importance nb ratings</i>	0.591	0.657	0.815	0.611	0.658	0.838
<i>ILD importance dernier rating</i>	0.573	0.631	0.799	0.591	0.639	0.828

Tableau 4.11 Résultats pour les dissimilarités moyennes calculées linéairement

Contrairement à nos trois métriques de précision, la valeur de notre mesure de diversité se trouve être plus haute pour notre système de recommandation basé sur le contenu comme observé au tableau 4.11. Une similitude avec nos précédentes analyses est, cependant, à noter. En effet, la diversité en prenant en compte la fréquence ou le nombre de notations est plus élevée que lors du calcul de la métrique classique traduisant donc une meilleure performance de cette métrique pour les utilisateurs ayant une grande importance dans ces critères par rapport aux utilisateurs ayant un petit score d'importance.

Quadratique

<i>Métriques</i>	<i>Base de données 1</i>			<i>Base de données 2</i>		
	<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>ILD classique</i>	0.577	0.639	0.805	0.593	0.644	0.830
<i>ILD importance fréquence</i>	0.583	0.658	0.808	0.619	0.669	0.836
<i>ILD importance nb ratings</i>	0.609	0.691	0.832	0.629	0.688	0.845
<i>ILD importance dernier rating</i>	0.573	0.625	0.795	0.591	0.635	0.827

Tableau 4.12 Résultats pour les dissimilarités moyennes calculées quadratiquement

Nous pouvons tirer exactement les mêmes conclusions pour la diversité calculée de manière quadratique (figure 4.12) que pour la diversité linéaire. Nous voyons également que l'utilisation de la fonction quadratique fait augmenter les valeurs des métriques pour la fréquence et le nombre de notations par rapport à son calcul linéaire tandis qu'elle baisse légèrement pour la date de la dernière notation. Cette observation prouve que les listes de recommandations proposées aux utilisateurs que l'on considère comme importants sont plus diversifiées lorsque le critère pris en compte est la fréquence ou le nombre de notations. Néanmoins, les utilisateurs ayant été actifs récemment, et donc avec un score plus élevé dans notre dernière version du calcul des métriques, semblent obtenir des évaluations plus basses en termes de diversité.

4.3.6. « Unexpectancy » ou Distance utilisateur

Continuons l'analyse des mesures allant « au-delà » de la précision en nous intéressant à une mesure nous permettant d'estimer la nouveauté présente dans les recommandations d'un système. Pour cela, nous calculons, comme décrites à l'équation 2.18, les métriques de distance

utilisateur ou « unexpectancy » en anglais. Un système procurant des propositions d'articles inédits à ces utilisateurs est grandement recherché, ce qui nous pousse à privilégier les systèmes ayant une grande valeur de métrique de distance utilisateur.

Linéaire

<i>Métriques</i>		<i>Base de données 1</i>			<i>Base de données 2</i>		
		<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>Distance classique</i>	<i>utilisateur</i>	0.682	0.715	0.797	0.701	0.726	0.829
<i>Distance importance fréquence</i>	<i>utilisateur</i>	0.692	0.727	0.803	0.712	0.737	0.833
<i>Distance importance nb ratings</i>	<i>utilisateur</i>	0.710	0.749	0.813	0.728	0.756	0.844
<i>Distance importance dernier rating</i>	<i>utilisateur</i>	0.683	0.724	0.794	0.706	0.739	0.84

Tableau 4.13 Résultats pour les distances utilisateur calculées linéairement

Les résultats pour les distances utilisateur calculées linéairement se trouvent dans le tableau 4.13. De même que pour la mesure de diversité, le système de recommandation basé sur le contenu procure des recommandations plus nouvelles à nos utilisateurs. Les métriques sont d'autant plus élevées si nous considérons les utilisateurs inégaux en termes de fréquence, de nombre de notations fournies ou de la date de leur dernière notation. Nos utilisateurs d'intérêts reçoivent donc des recommandations plus nouvelles que les utilisateurs lambdas. Ceux-ci font baisser le score de la métrique lorsque tous les utilisateurs reçoivent un poids identique.

Quadratique

<i>Métriques</i>		<i>Base de données 1</i>			<i>Base de données 2</i>		
		<i>UB</i>	<i>IB</i>	<i>CB</i>	<i>UB</i>	<i>IB</i>	<i>CB</i>
<i>Distance classique</i>	<i>utilisateur</i>	0.682	0.715	0.797	0.701	0.726	0.829
<i>Distance importance fréquence</i>	<i>utilisateur</i>	0.708	0.746	0.811	0.740	0.767	0.844
<i>Distance importance nb ratings</i>	<i>utilisateur</i>	0.738	0.783	0.831	0.756	0.789	0.858

<i>Distance importance rating</i>	<i>utilisateur dernier</i>	0.683	0.727	0.791	0.709	0.745	0.847
---	--------------------------------	-------	-------	--------------	-------	-------	--------------

Tableau 4.14 Résultats pour les distances utilisateurs calculées quadratiquement

Nous tirons nos conclusions grâce au tableau 4.14 reprenant les résultats pour les distances utilisateurs calculées quadratiquement. Les deux conclusions de l'analyse des résultats de la distance utilisateur calculée de façon linéaire sont également d'application lorsque la métrique est calculée quadratiquement. L'augmentation des scores lorsque nous passons de la méthode linéaire à la méthode quadratique ne fait que confirmer nos analyses quant à l'impact des utilisateurs lambdas sur les scores de la métrique.

4.4. Conclusion des analyses

Dans cette section du mémoire, nous avons pu analyser les performances de nos trois modèles grâce à des méthodes d'évaluations différentes : une méthode classique, une méthode prenant en compte la fréquence de notations des utilisateurs, une accordant plus d'importance en fonction du nombre de notations produite par chaque utilisateur et enfin, la dernière privilégiant les utilisateurs ayant noté un ou plusieurs films récemment.

Nous avons analysé l'impact de l'incorporation des critères dans notre calcul de façon linéaire. Nous avons alors deux cas de figure possibles, lorsqu'une métrique utilisant une liste d'importance fournissait des résultats inférieurs à la valeur classique ou lorsqu'elle fournissait des résultats supérieurs. Nous avons conclu que, dans le premier cas, cela se traduisait par une performance moindre pour les utilisateurs d'importance, ce qui faisait donc baisser les métriques lorsque nous leur accordons plus d'importance. Nous pouvons alors dire que la méthode classique surévalue la métrique si nous sommes dans une optique de privilégier les utilisateurs d'intérêts. C'est le cas notamment de la plupart des Recalls des deux bases de données, du nDCG de la première base de données (sauf pour le modèle item based utilisant la fréquence et le modèle basé sur le contenu utilisant le nombre de notations) et celui de la deuxième base de données de la version utilisant la date de la dernière notation. Nous pouvons voir que ce critère est également inférieur à la version classique pour les précisions des modèles item based et basé sur le contenu, des F-measures et des distances intra-liste. La deuxième situation est le cas contraire, c'est-à-dire, que les utilisateurs importants dans le critère spécifique produisent une meilleure performance dans la métrique. La valeur de la mesure est donc sous-évaluée dans la méthode classique. Nous retrouvons cette situation pour la plupart

des nDCGs de la deuxième base de données lorsque nous utilisons les importances en fonction de la fréquence et du nombre de notations, pour la Précision, la F-mesure, la distance intra-liste des deux bases de données pour ces mêmes critères et enfin, pour tous les critères des deux bases de données pour la distance utilisateur. Nous pouvons observer les mêmes tendances à quelques exceptions près lorsque nous utilisons des listes d'importances calculées à l'aide d'une fonction au carré.

De plus, nous avons étudié la variation des métriques homologues si nous passons d'un modèle linéaire à un modèle quadratique. Ici encore, nous avons alors deux situations. La première se procurait lorsque nous observions une diminution dans les scores lorsque nous passons d'un modèle linéaire à quadratique. En accordant encore plus d'importance à nos utilisateurs d'intérêts, nous nuisons à la valeur de la métrique traduisant alors le fait que ceux-ci reçoivent de moins bonnes recommandations que les utilisateurs lambdas. Inversement, une augmentation des scores lorsque nous passons à la version quadratique nous indique que le modèle performe mieux sur les utilisateurs d'intérêts que sur les autres. La première situation est observable pour les métriques de nDCG et de Recall, avec quelques exceptions pour les résultats du système basé sur le contenu. C'est également le cas pour la F-mesure et la distance intra-liste avec le critère de date de la dernière notation. La deuxième observation se produit le reste du temps, c'est-à-dire, pour la distance utilisateur et la Précision, la F-mesure et distance intra-liste utilisant les fréquences et nombre de notations.

Soulevons néanmoins une autre observation : lorsque nous avons une mesure modifiée qui se trouve être plus faible que la version classique, sa valeur diminuera dans la plupart des cas lors du passage à la version quadratique et inversement avec une mesure supérieure.

Malgré ces différentes variations, lorsque nous avons voulu établir un classement des modèles en fonction de leurs performances, nous avons constaté que ce dernier n'était ni influencé par la pondération utilisée (linéaire ou quadratique), ni par les critères utilisés pour la création des listes d'importance puisque chacune d'elle fournissait les mêmes résultats. La seule exception à cette règle est observée pour trois métriques de précision (Recall, Précision et F-mesure) dans le classement des systèmes lorsque nous basons la liste d'importance sur la fréquence de notations des utilisateurs de façon quadratique. Néanmoins, en effectuant un test de Student de comparaison des moyennes pour chacune de ces mesures, nous constatons que cette différence n'est pas significative.

Enfin, nous avons remarqué une différence assez importante pour la Précision de la méthode utilisant le nombre de notations des utilisateurs dans son calcul par rapport à la mesure classique. En effet, nous avons pu voir des valeurs deux à six fois supérieures pour les métriques favorisant certains utilisateurs. Nous avons alors conclu que la précision des individus ayant fourni beaucoup d'information quant à leur historique de films visionnés était amplement meilleure.

4.5. Conclusion managériale

Nous avons pu analyser les résultats de manière détaillée et avons pu tirer la conclusion que l'attribution de poids différents aux utilisateurs en fonction de leurs profils dans le but de discriminer leurs importances dans l'évaluation des différents modèles n'influençait pas leur classement. Mais quelles sont les implications de cette conclusion dans la réalité ? C'est ce que nous allons tenter de voir.

Les conclusions évoquées au point précédent nous poussent à affirmer que l'utilisation de listes d'importance dans le calcul des métriques d'évaluation n'a pas de sens si l'objectif du propriétaire du système de recommandation est de déterminer quel système privilégier. Nous avons pu voir que les métriques utilisées actuellement dans la littérature étaient efficaces pour un tel but. Nous pouvons donc nous fier directement aux résultats procurés par l'utilisation des métriques classiques et proposer directement le système de recommandation ayant obtenu les meilleurs résultats à nos utilisateurs. Gardons néanmoins en tête qu'il s'agit ici d'un travail universitaire qui comporte certaines limites notamment au niveau des bases de données et des systèmes de recommandation utilisés. Celles-ci seront développées plus amplement dans la conclusion. Il faut donc prendre un certain recul critique sur les conclusions que nous pouvons effectuer.

Néanmoins, nous avons constaté que malgré le fait que cette technique n'avait aucun impact sur l'ordre des performances des systèmes, elle nous permettait d'obtenir des informations supplémentaires quant à la performance des modèles. En effet, l'utilisation de ce genre de liste va nous permettre de mieux comprendre qui obtient de meilleures recommandations entre les utilisateurs que nous considérons comme importants et ceux que nous ne souhaitons pas prendre en compte. Si nous analysons la variation des métriques entre les versions classiques et les nouvelles versions introduites dans ce mémoire, il est alors possible de déterminer s'il s'agit de nos utilisateurs privilégiés qui contribuent le plus à la performance de notre système en

regardant si les métriques augmentent. Cela nous permettrait alors de dresser le portrait du « profil idéal » d'un utilisateur nécessaire afin que notre modèle lui fournisse des résultats optimaux. Des actions marketing pourraient être mises en place pour atteindre cet objectif.

Par exemple, le fait que nos valeurs de Précision augmentent grandement lorsque nous attribuons de l'importance aux utilisateurs ayant un grand nombre de notations nous prouve que de meilleures recommandations leur sont proposées. En pratique, un responsable souhaitant vérifier que ses meilleurs clients reçoivent les meilleures recommandations pourrait utiliser cette méthodologie pour s'en assurer. Il pourrait alors inciter ses clients à augmenter leur score d'importance afin que de meilleures recommandations leur soient également proposées. Dans notre cas, une possibilité serait de demander aux clients de fournir plus de notations de films ou d'envoyer un e-mail hebdomadaire afin de leur demander d'introduire de nouvelles informations ce qui lui permettra d'augmenter leur fréquence ainsi que d'améliorer leurs scores quant à la date de dernière notation.

5. Conclusion

Tout au long de ce mémoire, nous nous sommes interrogés sur la pertinence de l'utilisation d'indices moyens dans l'évaluation de la performance des systèmes de recommandation. L'idée sous-jacente était qu'il serait plus pertinent d'attribuer plus de poids à un certain groupe d'utilisateurs afin de privilégier leurs recommandations plutôt que d'utiliser un système de moyenne considérant chaque utilisateur comme égaux.

La première étape de la réalisation de ce travail était de s'imprégner du sujet et de la littérature existante. Nous avons donc rédigé une revue de littérature reprenant les grands concepts des systèmes de recommandation. Nous avons rappelé ce qu'était un système de recommandation, ses objectifs et avantages. Nous avons expliqué les trois modèles de base que nous allions utiliser par la suite ainsi que les différentes manières de calculer la similarité. Enfin, les moyens d'évaluations classiques et au-delà de la précision ont été introduits.

Ensuite, nous avons développé ensemble la méthodologie qui nous a permis d'obtenir nos résultats. Nous avons découvert nos bases de données, les modèles utilisés et comment ils étaient construits. Une partie importante était de comprendre comment nous comptions privilégier certains profils plutôt que d'autres. Nous avons alors expliqué ce raisonnement pour chacun des critères choisis : le nombre de notations, la date de la dernière notation et la fréquence de notation. Les mesures d'évaluation ont été présentées tant de la manière classique qu'avec l'intégration des listes d'importances dans leur calcul. Nous avons fait le choix d'utiliser le nDCG, le Recall, la Précision, la dissimilarité moyenne (ILD) et la distance utilisateur (Unexpectancy).

Enfin, nous avons pu collecter nos données et les analyser. Pour chaque modèle et chaque base de données, nous avons deux valeurs : les résultats avec une liste d'importance linéaire et ceux avec une liste d'importance quadratique. Nous avons, pour chaque métrique d'évaluation, regardé l'impact de l'introduction du système de pondération en fonction de l'importance de nos différents critères sur ses valeurs. Nous avons cherché à voir ce que les variations dans les scores d'évaluations traduisaient en réalité. Nous avons alors compris qu'une hausse d'un score impliquait que les utilisateurs d'importance produisaient de meilleurs résultats que les autres utilisateurs dans cette métrique et inversement lors d'une baisse de la valeur de la métrique. Ensuite, nous avons analysé si le changement dans le calcul des métriques influençait le classement de nos systèmes de recommandation.

La conclusion de cette partie était que l'introduction de la discrimination des utilisateurs n'a pas d'impact sur la hiérarchie des modèles évalués. Il n'est donc pas nécessaire d'introduire une telle configuration dans l'évaluation de système de recommandation si notre objectif est uniquement de déterminer quel système de recommandation proposer à nos utilisateurs. Nous avons, pour finir, souligné que malgré le fait que l'introduction de ce système d'évaluation ne produisait pas de changement quant au classement des modèles, il nous permettait néanmoins d'obtenir des informations sur le profil des clients recevant les meilleures recommandations et donc d'obtenir des pistes de solution afin d'améliorer la pertinence de celles-ci grâce à la mise en place de différentes actions.

Limites et perspectives de recherche

Dans cette partie, nous examinerons les limites de notre étude et les perspectives de recherche futures.

Une des principales limites que nous avons rencontrées est liée à nos données de base. En effet, nous avons utilisé deux bases de données provenant de MovieLens et caractérisant les notations d'utilisateurs sur les films. Il s'agit alors de base de données de même type. De plus, lorsque nous avons observé la distribution des attributs utilisés dans la conception de nos listes d'importance, nous nous rendons compte que, encore une fois, nous nous retrouvons avec des résultats assez similaires comme nous pouvons le voir en annexe 4. Il serait alors intéressant d'utiliser une base de données différente avec une distribution plus variée et de voir si nos résultats sont influencés ou non par le changement de type de base de données.

Une autre limite importante concerne la méthodologie utilisée et les choix que nous avons dû effectuer dans la réalisation de ce mémoire. Nous nous sommes, par exemple, concentrés sur trois systèmes de recommandation de base : un système de filtrage collaboratif basé sur les utilisateurs, un système de filtrage collaboratif basé sur les articles et un système basé sur le contenu avec pour attributs la date de parution des films ainsi que leurs genres. Une future recherche pourrait procéder à la même étude mais en utilisant des systèmes de recommandation différents.

De la même manière, nous avons fait le choix de deux métriques afin de calculer la nouveauté et la diversité de nos modèles. Cependant, nous avons vu dans la littérature qu'il en existe bien d'autres. L'utilisation de nouvelles métriques pourrait modifier nos résultats et donc nos conclusions.

Enfin, les choix effectués quant à la création des listes d'importances en fonction de l'utilisateur auraient pu être tout autre. Nous avons pris la décision de nous concentrer sur trois critères : le nombre de notations, la date de la dernière notation et la fréquence de notation. Néanmoins, il serait peut-être pertinent de tester avec d'autres critères. De plus, lorsque nous avons décidé de discriminer les utilisateurs de façon plus importante, notre choix s'est porté sur la fonction quadratique. De nombreuses autres fonctions pourraient être utilisées et fourniraient potentiellement des résultats différents.

6. Bibliographie

6.1. Articles périodiques

- Burke, R., Felfernig, A. & Göker, M.H. (2011). Recommender Systems: An Overview. *AI Magazine*, 32(3), 13-18. doi: 10.1609/aimag.v32i3.2361
- Castells, P., Hurley, N.J., Vargas, S. (2015). Novelty and Diversity in Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B. (eds) *Recommender Systems Handbook*. Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7637-6_26
- Chen, M., & Liu, P. (2017). Performance Evaluation of Recommender Systems. *International journal of performability engineering*, 13, 1246-1256.
- Fkih, F. (2021). Similarity measures for collaborative filtering-based recommender systems: Review and experimental comparison. *Journal of King Saud University – Computer and Information Sciences*, 7645-7669.
- Ge, M., Delgado, C., & Jannach, D. (2010). Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems* (pp. 257-260). <https://doi.org/10.1145/1864708.1864761>
- Harper, F. M., & Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4), 19:1–19:19. <https://doi.org/10.1145/2827872>
- Hassanieh, L.A., Jaoudeh, C.A., Abdo, J.B., & Demerjian, J. (2018). Similarity measures for collaborative filtering recommender systems. *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 1-5.
- Herlocker, J.L., Konstan, J.A., Terveen, L.G., & Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22, 5-53.
- Hernández del Olmo, F., & Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 35, 790-804. <https://doi.org/10.1016/j.eswa.2007.07.047>

- Isinkaye, F.O., Folajimi, Y.O. & Ojokoh, B.A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3), 261-273. doi: 10.1016/j.eij.2015.06.005
- Jannach, D., Zanker, M. (2022). Value and Impact of Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B. (eds) *Recommender Systems Handbook*. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-2197-4_14
- Kaminskas, M., & Bridge, D.G. (2016). Diversity, Serendipity, Novelty, and Coverage. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7, 1 - 42.
- Karakaya, M. Ö., & Aytekin, T. (2018). Effective methods for increasing aggregate diversity in recommender systems. *Knowledge and Information Systems*, 56, 355-372. <https://doi.org/10.1007/s10115-017-1135-0>
- Khachane, A.R. & Vaidya N. (2017). Recommender Systems-The need of the Ecommerce Era. *IEEE 2017 International Conference on Computing Methodologies and Communication*, 100- 114.
- Kuzelewska, U. (2014). Clustering Algorithms in Hybrid Recommender System on MovieLens Data. *Studies in Logic, Grammar and Rhetoric*, 37(1) 125-139. <https://doi.org/10.2478/slgr-2014-0021>
- Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1), 1-49. <https://doi.org/10.1016/j.physrep.2012.02.006>
- Ma, M. Y., & Uchyigit, G. (2008). Personalization techniques and recommender systems. *Series in machine perception and artificial intelligence* (Vol. 70). World Scientific.
- Melville, P., & Sindhvani, V. (2010). Recommender Systems. *Encyclopedia of Machine Learning and Data Mining*.
- Mohamed, M., Khafagy, M., & Ibrahim, M. (2019). Recommender systems challenges and solutions survey. *IEEE Transactions on Consumer Electronics*, 65(3), 299-309. doi: 10.1109/ITCE.2019.8646645
- Niemann, K., & Wolpers, M. (2013). A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 22nd international*

- conference on World Wide Web* (pp. 955-963). International World Wide Web Conferences Steering Committee. doi: 10.1145/2487575.2487656.
- Patel, B., Desai, P., & Panchal, U. (2017). Methods of recommender system: A review. *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-4). <https://doi.org/10.1109/ICIIECS.2017.8275856>
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web* (pp. 325-341). Lecture Notes in Computer Science, Vol. 4321. Springer. doi: 10.1007/978-3-540-72079-9_10
- Ricci, F., Rokach, L., Shapira, B. (2022). Recommender Systems: Techniques, Applications, and Challenges. In: Ricci, F., Rokach, L., Shapira, B. (eds) *Recommender Systems Handbook*. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-2197-4_1
- Schafer, J.B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds) *The Adaptive Web*. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg.
- Shah, K., Salunke, A., Dongare, S., & Antala, K. (2017). Recommender systems: An overview of different approaches to recommendations. *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-4). doi: 10.1109/ICIIECS.2017.8276172
- Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender Systems Handbook* (pp. 257-297). https://doi.org/10.1007/978-0-387-85820-3_8
- Silveira, T., Zhang, M., Lin, X., & al. (2019). How good your recommender system is? A survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics*, 10, 813-831. <https://doi.org/10.1007/s13042-017-0762-9>
- Sinha, B. B., & Dhanalakshmi, R. (2019). Evolution of recommender system over the time. *Soft Computing*, 23, 12169-12188. <https://doi.org/10.1007/s00500-019-04143-8>
- Sondur, S.D., Nayak, S., & Chigadani, A.P. (2016). Similarity Measures for Recommender Systems: A Comparative Study. *International Journal for Scientific Research and Development*, 2, 76-80.

Tintarev, N., & Masthoff, J. (2007). A survey of explanations in recommender systems. In *2007 IEEE 23rd International Conference on Data Engineering Workshop* (pp. 801-810). IEEE. doi:10.1109/ICDEW.2007.4401070

Tarnowska, K.A., Ras, Z.W., & Daniel, L. (2019). Recommender System for Improving Customer Loyalty.

Wu, W., He, L., & Yang, J. (2012). Evaluating recommender systems. In *Seventh International Conference on Digital Information Management (ICDIM 2012)* (pp. 56-61). <https://doi.org/10.1109/ICDIM.2012.6360092>

6.2. Livres

Aggarwal, C. C. (2016). *Recommender systems: The textbook (illustrated ed.)*. Springer.

Al-Turjman, F. (Ed.). (2020). *Trends in Cloud-based IoT*. Springer Cham.

Berrar, D. (2018). Cross-Validation. In *Reference Module in Life Sciences*. Elsevier. doi: 10.1016/B978-0-12-809633-8.20349-X.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (Eds.). (2010). *Recommender systems: An introduction*. Cambridge University Press.

Kembellec, G., Chartron, G., & Saleh, I. (2014). *Recommender systems (illustrated, reprinted ed.)*. John Wiley & Sons. (Originalmente publicado en la serie Information systems, web and pervasive computing series de Wiley Online Library)

Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2022). *Recommender systems handbook*. Springer New York, NY. <https://doi.org/10.1007/978-1-0716-2197-4>

6.3. Sites internet

GroupLens. (2018). Movielens. Récupéré le 15 juin, de <https://grouplens.org/datasets/movielens/>

JupyterLab. (s.d.). JupyterLab Documentation. Récupéré le 6 juin 2023, de <https://jupyterlab.readthedocs.io/en/stable/>

Kaggle. The Movies Dataset [Dataset]. Récupéré le 29 juin 2023, de
<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset>

6.4. Mémoires

Cotton, L. (2021). Intégration des mesures de centralité sur graphe dans les systèmes de recommandation pour augmenter la nouveauté et la diversité de leurs recommandations [Mémoire, UCLouvain FUCaM Mons].

