



**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche**  
**Scientifique**



**Université Ibn Khaldoun - Tiaret**  
**Faculté Mathématiques et Informatique**  
**Département d'Informatique**

**MEMOIRE DE FIN D'ETUDE**  
**Pour l'Obtention du Diplôme de Master en Informatique**  
**Spécialité : Génie Logiciel**

**Présenté par :**  
**BOUTELDJA ABDERRAZAK**  
**MOUSSAOUI HOUSSAM**

## **Sujet**

**Techniques de factorisation dans un**  
**systeme SFC**

**Devant le jury composé de :**

<b>Mr. ABID Khaled</b>	<b>President</b>
<b>Mme. BENATHMANE Laila</b>	<b>Examinatrice</b>
<b>Mr. KHARROUBI Sahraoui</b>	<b>Encadreur</b>

**Année Universitaire : 2022-2023**

## Dédicaces

*Tous les mots ne sauraient exprimer la gratitude, l'amour,  
Le respect, la reconnaissance, c'est tout simplement  
que je dédie ce mémoire.*

*À mes parents car ils m'ont inculqué un esprit de persévérance  
et qu'ils m'ont toujours poussé et motivé dans mes études.*

*Sans eux, certainement je ne serais pas à ce niveau.*

*Ils ont tout fait pour mon bonheur et ma réussite.*

*Nulle dédicace ne puisse exprimer ce que je leur dois.*

*Que dieu leur réserve la bonne santé et une longue vie.*

*À mes frères et sœurs, qu'ils restent pour moi des exemples de  
persévérance, de courage et de générosité.*

*Et à tous ceux qui me sont chers et qui ont contribué à mes  
encouragements..*

## **Remerciements**

*Nous tenons tout d'abord à remercier Dieu le tout puissant  
 , qui nous a donné la force et la patience  
 d'accomplir ce travail.*

*Nous tenons à remercier notre encadreur Mr KHARROUBI  
 SAHRAOUI pour son encadrement son écoute,  
 ses élucidations, ses conseils, ses directives et  
 ses encouragements.*

*Nos vifs remerciements vont également aux membres du  
 jury qui ont accepté de juger notre travail.*

*Nous tenons également à remercier tous ceux qui ont  
 contribué de près ou de loin à la réalisation de ce travail.*

## Résumé

La technique de factorisation matricielle est primordiale pour remédier le problème de manque d'évaluation. Ainsi, appliquer cette technique a pour objectif d'améliorer la prédiction dans un système de filtrage collaboratif. Dans ce travail, nous avons intégré des systèmes de recommandation dans une application web pour faire des recommandations des items en utilisant les techniques de factorisation matricielle en filtrage collaboratif qui se basent sur les méthodes réductions dimensionnelles et plus précisément la décomposition de la valeur singulière (SVD).

**Mots clés :** Factorisation matricielle, système de filtrage collaboratif, prédiction, pertinence.

## Abstract

The matrix factorization technique is paramount to address the problem of lack of evaluation. Thus, applying this technique aims to improve prediction in a collaborative filtering system. In this work, we integrated recommendation systems into a web application to make item recommendations using matrix factorization techniques in collaborative filtering that are based on dimensional reduction methods and more precisely singular value decomposition (SVD).

**Keywords:** Matrix factorization, collaborative filtering system, prediction, relevance.

## Liste des figures

Figure 1.1 : Classification des systèmes de recommandation.....	4
Figure 1.2 : Illustration de la recommandation basée sur le contenu.....	6
Figure 1.3 : Recommandation basée collaboratif.....	7
Figure 3.1 : Illustration de la factorisation de matrices.....	24
Figure 3.2 : Décomposition en valeur singulière SVD.....	25
Figure 4.1 : Diagramme de cas d'utilisation.....	34
Figure 4.2 : Diagramme de séquence «Inscription».....	37
Figure 4.3 : Diagramme de séquence «Connexion».....	38
Figure 4.4 : Diagramme d'activité Authentification.....	39
Figure 4.5 : Diagramme de classe.....	40
Figure 4.6 : Traitement des demandes dans Spring Web MVC.....	42
Figure 4.7 : la décomposition en valeur singulière.....	43
Figure 4.8 : Interface d'identification.....	47
Figure 4.9 : Interface d'inscription.....	47
Figure 4.10 : Interface de profil.....	48
Figure 4.11 : Interface de recommandation.....	49
Figure 4.12 : Interface d'évaluation de produit.....	50
Figure 4.13 : Interface de remplissage de matrice.....	50
Figure 4.14 : Interface de l'applique SVD.....	51
Figure 4.15 : Graphe pour les valeurs de MAE .....	52

## Liste des tableaux

Table 2.1 – Matrice des évaluations pour illustrer l’approche <b>NBCF</b> .....	20
Table 3.1 – Exemple des évaluations Utilisateurs / item.....	27
Table 3.2 – matrice <b>U</b> factorisé.....	27
Table 3.3 – matrice <b>V<sup>T</sup></b> factorisé.....	27
Table 3.4 – matrice des prédictions de NMF.....	28
Table 3.5 – analyse statistique des items.....	29
Table 3.6 – matrice des données centrées réduite <b>X<sub>c</sub></b> . ....	30
Table 3.7 – matrice de corrélation linéaire <b>C</b> .....	30
Table 3.8 – matrice des prédictions de PCA .....	31
Table 4.1 – Description textuelle du cas d'utilisation "inscription" .....	35
Table 4.2 – Description textuelle du cas d'utilisation "connexion".....	36
Table 4.3 – Matrice des évaluations utilisateurs / items .....	44
Table 4.4 – Matrice <b>U</b> orthogonal .....	44
Table 4.5 – Matrice <b>V<sup>T</sup></b> orthogonal.....	45
Table 4.6 – Matrice <b>Σ</b> diagonale.....	45
Table 4.7 – Matrice <b>U<sub>2</sub></b> factorisé.....	46
Table 4.8 – Matrice <b>V<sup>T</sup><sub>2</sub></b> factorisé.....	46
Table 4.9 – Matrice <b>Σ<sub>2</sub></b> factorisé.....	46
Table 4.10 – Matrice prédictions utilisateurs / items.....	46
Table 4.11 – les valeurs de MAE de chaque technique.....	51

# Glossaire

**CBF** filtrage basé sur le contenu – ou Content-Based Filtering.

**CF** filtrage collaboratif – ou Collaborative Filtering.

**COS** similarité cosinus.

**RMSE** La métrique d'erreur quadratique moyenne – ou Root Mean Squared Error

**MAE** La métrique d'erreur absolue moyenne – ou Mean Absolute Error)

**MSD** distance quadratique moyenne – ou Mean Squared Distance.

**NBCF** filtrage collaboratif basé sur le voisinage – ou Neighborhood-Based Collaborative Filtering.

**PCC** coefficient de corrélation de Pearson – ou Pearson Correlation Coefficient.

**SR** système de recommandation.

**SVM** machines à vecteurs de support – ou Support Vector Machine.

**SVD** la décomposition en valeur singulière – Singular value decomposition.

**NMF** Factorisation en matrices non-négatives.

**PCA** Analyse des composants principaux.

## Liste des notations

$\mathbf{U}$  ensemble des utilisateurs  $\mathbf{u}$ .

$\mathbf{I}$  ensemble des items  $\mathbf{i}$ .

$\mathbf{I}_{\mathbf{u}}$  ensemble des items notés par l'utilisateur  $\mathbf{u}$ .

$\mathbf{I}_{\mathbf{u},\mathbf{v}}$  ensemble des items co-notés par les utilisateurs  $\mathbf{u}$  et  $\mathbf{v}$ .

$\mathbf{R}$  matrice des évaluations (ratings).

$\mathbf{r}$  (rating) –  $\mathbf{r}_{\mathbf{u},\mathbf{i}}$  représente la note que l'utilisateur  $\mathbf{u}$  a assigné explicitement à l'item  $\mathbf{i}$ .

$\hat{\mathbf{r}}$  prédiction d'un (rating) –  $\hat{\mathbf{r}}_{\mathbf{u},\mathbf{i}}$  représente la prédiction de la note que l'utilisateur  $\mathbf{u}$  assignerait à l'item  $\mathbf{i}$ .

# SOMMAIRE

## Sommaire

---

Introduction générale .....	1
<b>CHAPITRE I : GENERALISATION SUR LES SYSTEMES DE RECOMMANDATION</b>	
1.1. Introduction.....	2
1.2. Systèmes de recommandation.....	2
1.2.1. Définitions et concepts de base.....	2
1.2.2. Classification des systèmes de recommandation.....	3
1.2.2.1. Filtrage basé sur le contenu.....	4
1.2.2.2. Filtrage collaboratif .....	6
1.2.2.3. Filtrage hybride .....	7
1.2.3. Évaluation des systèmes de recommandation.....	9
1.2.3.1. Évaluation basée sur prédictions .....	10
1.2.3.2. Évaluation basée sur la liste de top-N recommandation.....	10
1.2.4. Limites des systèmes de recommandation.....	11
1.3. Conclusion .....	12
<b>CHAPITRE II : SYSTEMES DE FILTRAGE COLLABORATIF</b>	
2.1. Introduction.....	13
2.2. Système de filtrage collaboratif.....	13
2.2.1. Approches basées mémoire .....	13
2.2.1.1. Identification du voisinage.....	14
2.2.1.1.1. Similarité cosinus (COS).....	14
2.2.1.1.2. Coefficient de corrélation Pearson (PCC) .....	15
2.2.1.1.3. Distance quadratique moyenne (MSD).....	16
2.2.1.2. Prédiction des évaluations (ratings) .....	17
2.2.1.3. Un exemple concret .....	19
2.2.2. Approches basées sur les modèles.....	21
2.3. Conclusion .....	22

### CHAPITRE III : FACTORISATION MATRICIELLE

3.1. Introduction.....	23
3.2. Factorisation de matrices FM.....	23
3.2.1. Définition La matrice de factorisation.....	23
3.2.2. Complétion de matrices.....	24
3.2.3. Principe de fonctionnement du FM.....	24
3.2.4. Les modèles de la factorisation matricielle.....	25
3.2.4.1. La décomposition de la valeur singulière SVD.....	25
3.2.4.2 Factorisation matricielle non négative NMF.....	26
3.2.4.3. Analyse des composants principaux PCA.....	28
3.3. Le rôle de la factorisation matricielle dans les SR.....	31
3.4. Conclusion.....	31

### CHAPITRE IV : IMPLEMENTATION

4.1. Introduction.....	32
4.2. Analyse et conception.....	32
4.2.1. Modélisation UML.....	32
4.2.2. Spécification des besoins du système.....	32
4.2.2.1. Identification des acteurs.....	32
4.2.2.2. Identification des besoins.....	33
4.3. Les diagrammes UML.....	34
4.3.1. Diagramme de cas d'utilisation.....	34
4.3.2. Description textuelle .....	35
4.3.3. Diagramme de séquence .....	37
4.3.4. Diagramme d'activité.....	39
4.3.5. Diagramme de classe.....	40
4.4. Environnement de développement.....	41
4.4.1. IntelliJ IDEA.....	41

## Sommaire

---

<b>4.4.2. XAMPP.....</b>	<b>41</b>
<b>4.4.3. Spring MVC.....</b>	<b>41</b>
<b>4.5. Langages de programmation.....</b>	<b>42</b>
<b>4.5.1. Java.....</b>	<b>42</b>
<b>4.5.2. XML.....</b>	<b>43</b>
<b>4.6. Algorithme de recommandation utilisé.....</b>	<b>43</b>
<b>4.7. Expérimentations et mise en œuvre.....</b>	<b>47</b>
<b>4.8. Comparaison des différents techniques .....</b>	<b>51</b>
<b>4.9. Conclusion.....</b>	<b>52</b>
<b>Conclusion général .....</b>	<b>53</b>
<b>Références.....</b>	<b>54</b>

# **INTRODUCTION**

## **GENERALE**

# **Introduction générale**

Le web contient une très grande variété de ressources telles que les films, les livres, les news, de la musique, etc. L'énorme masse d'informations existante et la rapidité de sa production instantanée conduit au problème de surcharge informationnelle.

Nous sommes en face d'une situation paradoxale, une nécessité d'accès à l'information et une difficulté de l'avoir. Alors, accéder à l'information pertinente est devenue une tâche ardue car elle est perdue dans la masse. C'est pourquoi les travaux s'orientent actuellement vers la révision de la chaîne d'accès à l'information dans la perspective d'intégrer l'utilisateur comme composante du modèle global de recherche et ce, dans le but de lui délivrer une information pertinente adaptée à ses besoins précis, son contexte et ses préférences.

Dès lors, la recherche d'information (RI) est une discipline de recherche qui s'intéresse à l'organisation, au stockage et à la sélection d'informations répondant aux besoins des utilisateurs. Les efforts continus des chercheurs en RI ont permis jusqu'à présent d'améliorer sans cesse les performances et la qualité des services d'accès à l'information. La deuxième facette alternative qui s'intéresse à l'accès à l'information, c'est le filtrage d'information (FI). Cette évolution peut être vue comme la sélection d'informations pertinentes sur un flux entrant. Le noyau d'un système de recommandation est généralement basé sur un processus de filtrage. Un système de recommandation permet à partir d'une source dynamique d'information de sélectionner et de présenter les seules ressources intéressantes à un utilisateur.

Le problème général des systèmes de recommandation est qu'ils disposent souvent de très peu de données. Dans le cas de recommandation collaborative, ce manque de données dit (sparsity) à une influence majeure en calcul de prédictions et souvent ce taux de (sparsity) dépasse les 95%. L'introduction des techniques de factorisation matricielle s'avère nécessaire et utile pour surmonter cette lacune. Dans ce travail, nous nous sommes focalisés sur l'application des techniques de factorisation matricielles pour améliorer la qualité de prédiction dans le filtrage collaboratif.

Ce mémoire est divisé en quatre chapitres, le premier chapitre présente les concepts de base des systèmes de recommandations, le deuxième se concentre sur le système de filtrage collaboratif, le troisième se détaille sur les techniques de factorisation matricielles et le dernier chapitre consacré à l'implémentation d'un noyau de système de recommandation par différentes techniques de factorisation matricielles.

**CHAPITRE I**  
**GENERALISATION SUR**  
**LES SYSTEMES DE**  
**RECOMMANDATION**

## **1. Introduction**

Les systèmes de recommandations sont devenus un domaine de recherche fertile depuis l'apparition des premiers documents sur le filtrage d'information en milieu des années 1990. Dans ce chapitre, nous focalisons sur la présentation des concepts de base liés aux systèmes de recommandations. En premier lieu, nous décrivons les différentes disciplines d'accès passives à l'information et/ou la ressource pertinente. Ainsi, nous exposons l'architecture de base de chaque type de système tout en montrant son mode de fonctionnement. En dernier lieu, cibler les limites de chaque catégorie de ces systèmes pour mieux viser la problématique et par conséquent chercher les solutions appropriées.

## **2. Systèmes de recommandation**

### **2.1. Définition et concepts de base**

Un système de recommandation (SR) est un environnement logiciel qui permet de fournir à chaque utilisateur les éléments (items) les plus susceptibles de l'intéresser [Ricci et al., 2015].

L'élaboration de systèmes de recommandation a pour objectif de réduire la surcharge d'informations en récupérant les informations et les services les plus pertinents à partir d'une quantité énorme de données, fournissant ainsi des services personnalisés. La caractéristique la plus importante d'un système de recommandation est sa capacité à « deviner » les préférences et les intérêts d'un utilisateur en analysant le comportement de cet utilisateur et / ou celui d'autres utilisateurs pour générer des recommandations [Resnick & Varian, 1997].

Néanmoins, il existe également des systèmes de recommandation atypiques :

- Pour certains, il n'y a pas vraiment de distinction claire entre la phase de prédiction et celle de recommandation. Plus précisément, ce genre de SR génère une liste de recommandations sans avoir besoin, au préalable, de prédire des notes. Un exemple représentatif est d'apprentissage par classement (Learning to rank en anglais) [Liu, 2009 ; Karatzoglou et al., 2013].

- Contrairement aux cas classiques, où seuls les items non notés sont légitimes pour la recommandation, d'autres SR considère que même les items pour lesquels les goûts des utilisateurs sont connus peuvent être recommandés à nouveau, dans des contextes différents. Prenant en compte ces contextes, ces SR visent à augmenter au maximum la personnalisation pour l'utilisateur cible. Par exemple, le système de recommandation de YouTube [Covington et al., 2016] tient compte de l'aspect temporel, permettant de recommander à nouveau des vidéos déjà regardées par l'utilisateur (e.g., une vidéo que l'utilisateur regarde de façon récurrente).

Considérant ces diverses approches de SR et afin d'éviter toute ambiguïté, dans la suite de ce mémoire on notera SR uniquement des systèmes de recommandation classiques qui :

1. disposent de deux phases consécutives, i.e. les phases de « prédiction » et de « recommandation » ;
2. ne prédisent les goûts des utilisateurs que pour les items non notés;
3. ne prennent pas en compte le contexte, ni le temps.

### **2.2. Classification des systèmes de recommandation**

Les techniques de recommandation peuvent être classées de différentes manières. Parfois plusieurs termes sont utilisés pour désigner une même méthode ou approche. La classification la plus utilisée repose sur trois types : filtrage basé sur le contenu, filtrage collaboratif et filtrage hybride [Adomavicius & Tuzhilin, 2005]. En plus de ces deux approches, Robin Burke [Burke, 2002] propose de considérer les trois autres approches : la recommandation basée sur les données démographiques, la recommandation basée sur la connaissance (knowledge-based) et la recommandation basée sur l'utilité (utility-based). Mais il a souligné que ces trois méthodes sont des cas particuliers de la méthode classique. Le but ici est de s'appuyer sur les classifications les plus connues sur lesquelles s'appuient nos recherches. Nous présentons ci-dessous les méthodes basées sur le contenu et le filtrage collaboratif, suivies des méthodes hybrides.

(Figure 1.1). Nous détaillerons les techniques de filtrage collaboratif que nous utiliserons dans notre travail.

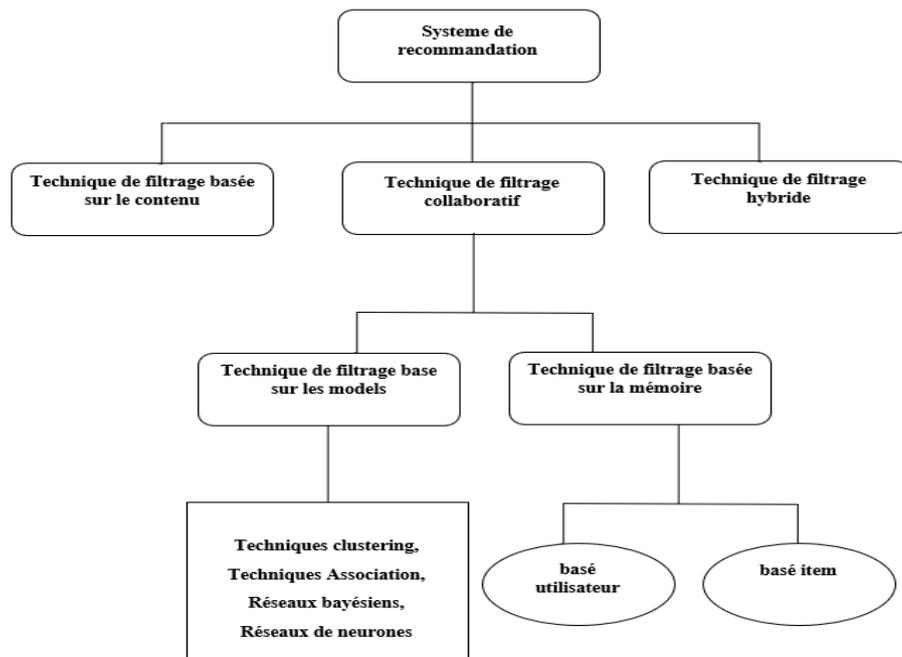


Figure 1.1 – Classification des systèmes de recommandation [Isinkaye et al., 2015]

### 2.2.1. Filtrage basé sur le contenu

« Si Mohamed aime regarder le film Harry Potter, il semble pertinent de lui recommander le film Harry Potter 2 ou un autre genre fantastique semble approprié ». L’hypothèse sous-jacente est que les utilisateurs ont une forte probabilité d’aimer les items qui sont similaires à ceux qu’ils ont aimés auparavant, en termes de contenu. Cette section s’intéresse à un autre type d’approche de recommandation, les approches de filtrage basé sur le contenu.

Approches de CBF se focalisent sur les feedbacks individuels d’un seul utilisateur, i.e. celui pour qui on va faire des recommandations [Lops et al.,2011 ; De Gemmis et al., 2015; Aggarwal, 2016a].

En plus des retours de cet utilisateur cible, pour que les approches de CBF fonctionnent, il est indispensable de disposer de données relatives aux contenus des items. Ces dernières peuvent être des données de différents types, structurées ou non-structurées selon la source des données. Par exemple, il peut s'agir de descriptions des items sous forme de textes (dans la plupart des SR basés sur le contenu) ou de manière plus structurée, sous la forme d'un ensemble de mots-clés (éventuellement hiérarchisés). Il peut s'agir également d'attributs (caractéristiques) des items stockés dans des bases de données relationnelles, e.g., le prix d'un produit, sa couleur, sa taille, etc.

De manière générale, les approches de recommandation basées sur le contenu des items procèdent en trois phases. Premièrement, disposant d'un ensemble de données relatives aux caractéristiques des items (attributs ou contenus) et quels que soient leurs sources ou leurs types, les SR basés sur le CBF analysent ces données afin de construire une représentation structurée (généralement sous la forme d'une modélisation vectorielle) de chaque item présent dans le catalogue du SR. Il s'agit souvent d'une phase de pré-traitement qui vise à extraire des informations pertinentes permettant de décrire les items. Ensuite, en se focalisant sur les (feedbacks) individuels de l'utilisateur cible, la deuxième phase vise à construire un profil pour l'utilisateur, qui révèle ses intérêts relatifs à chacune des caractéristiques des items. Enfin, la troisième phase consiste à comparer le profil de cet utilisateur avec les caractéristiques correspondant aux items du catalogue afin d'identifier ceux qui sont les plus pertinents pour lui. Il s'agit donc d'une phase de filtrage des items pour exclure ceux dont les attributs (contenus) ne correspondent pas au profil utilisateur.

L'avantage des systèmes basés contenu est qu'ils permettent d'associer des documents à un profil utilisateur. Notamment, en utilisant des techniques d'indexation et d'intelligence artificielle. L'utilisateur est indépendant des autres ce qui lui permet d'avoir des recommandations même s'il est le seul utilisateur du système pour éviter certaines limitations et inconvénients des systèmes basés sur le contenu et collaboratifs [Belloui, 2008].

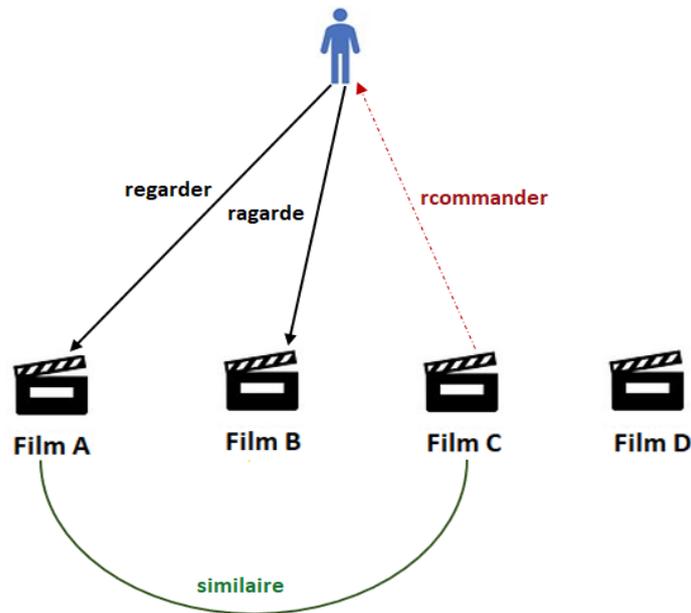


Figure 1.2 – Illustration de la recommandation basée sur le contenu.

### 2.2.2. Filtrage collaboratif

Le filtrage collaboratif ou « Collaborative Filtering (CF) », est une technique largement employée dans le domaine (à la fois scientifique et industriel) des systèmes de recommandation. Pour recommander des items à l'utilisateur  $u$ , le CF s'appuie sur les opinions des autres utilisateurs, d'où le terme « collaboratif ». GroupLens [Resnick et al., 1994; Konstan et al., 1997] est l'un des premiers systèmes de recommandation s'appuyant sur le filtrage collaboratif. Dans le domaine du commerce électronique, le SR d'Amazon est également représentatif de cette technique : « Les clients ayant acheté ce produit ont également acheté... ». Notons que dans un SR, il est très rare que les utilisateurs donnent un avis (rating) pour chaque item du catalogue, autrement dit, la matrice des évaluations (ratings)  $R$  est incomplète. De ce fait, l'objectif du CF est de prédire les valeurs des évaluations manquantes, en prenant appui sur les évaluations (ratings) disponibles, i.e.  $R$  initial.

L'approche du filtrage collaboratif est divisée en deux sous catégories qui dépendent du procédé de prédiction : l'approche basée sur la mémoire et l'approche basée sur les modèles.

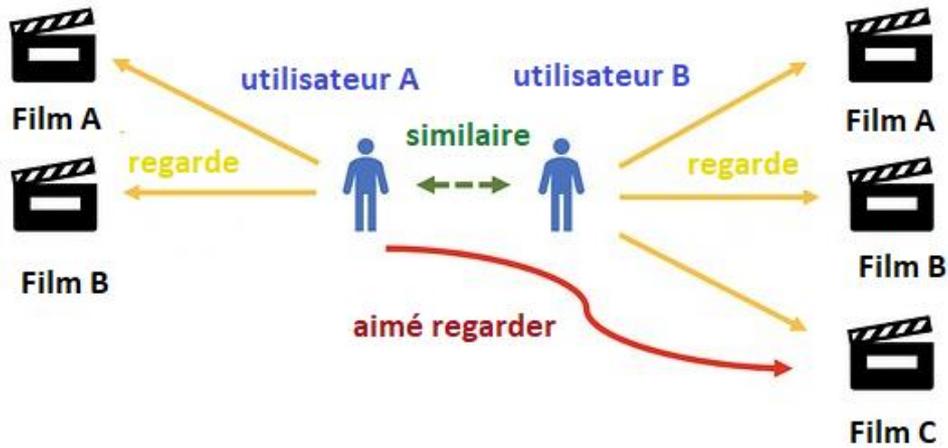


Figure 1.3 –Recommandation basée collaboratif.

Tout comme les approches basées sur le contenu, le filtrage collaboratif souffre du problème du démarrage à froid : avant que le système puisse fournir des recommandations pertinentes à un utilisateur, il faut que ce dernier ait fourni, implicitement ou explicitement, des appréciations pour un nombre suffisant de ressources. Un problème supplémentaire par rapport aux recommandations basées sur le contenu est que cette limitation s'applique également aux nouvelles ressources introduites dans le système. Des solutions à ces problèmes se trouvent dans les approches hybrides. Il est à noter que le filtrage collaboratif sera détaillé en chapitre 2.

### **2.2.3. Filtrage hybride**

Les systèmes de filtrage hybrides combinent le système de filtrage collaboratif avec d'autres techniques de recommandation (généralement avec des systèmes basés sur le contenu) pour faire des prédictions ou des recommandations pour éviter certaines limitations et inconvénients des systèmes basés sur le contenu et les systèmes filtrage collaboratifs [Rao & Talwar, 2008].

Même si de nombreuses méthodes d'hybridation ont été proposées, l'idée principale derrière ces méthodes reste la même : la combinaison de différentes techniques de recommandation et de différents types de données (ratings, connaissances liées aux items, données démographiques des utilisateurs, etc.) afin d'améliorer la performance des SR et de les rendre plus robustes. Dans ce qui suit, quelques méthodes d'hybridation largement utilisées dans la littérature sont brièvement présentées :

- **hybridation pondérée** — L'idée du hybridation pondéré est de combiner les scores obtenus par différents systèmes de recommandation (généralement de manière linéaire). En d'autres termes, la prédiction finale de la note de l'utilisateur  $u$  pour l'élément  $i$ , c'est-à-dire  $\widehat{r}_{u,i}$ , est une combinaison de prédictions faites par différents systèmes de recommandation. Par exemple, les auteurs [Miranda et al., 1999] ont introduit une méthode pour combiner linéairement les prédictions de CBF et CF est proposée. Les auteurs attribuent initialement des poids égaux aux scores des deux SR, puis proposent d'ajuster ces poids en fonction des commentaires des utilisateurs en ligne.
- **hybridation commutée** — Dans un SR hybride de manière commutée, le système bascule entre différentes techniques de recommandation selon certains critères. Les auteurs [Billsus et al., 2000] proposent une approche d'hybridation commutée dans laquelle une approche CBF est utilisée lorsque le nombre des évaluations (ratings) est petit, tandis qu'une approche CF est utilisée lorsque le système dispose de suffisamment de retours. Suivant le même principe, [Benouaret, 2017] propose un SR hybride pour la visite personnalisée de sites culturels. Dans ses travaux, il combine trois approches de recommandation, qui sont utilisées de manière commutée selon le contexte. Plus précisément, une approche démographique est initialement utilisée pour faire des recommandations à un nouvel utilisateur et ainsi pallier le problème du démarrage à froid. Ensuite, une approche CBF se basant sur la similarité sémantique est utilisée lorsque l'utilisateur a noté un certain nombre d'items. Enfin, une approche collaborative est utilisée lorsque le SR dispose de plus de retours de l'utilisateur.

- **hybridation mixte** — Il s’agit de présenter une liste d’items qui contient des recommandations de chaque approche utilisée. Les items résultants de chaque approche de recommandation sont présentés simultanément à l’utilisateur [Smyth and Cotter, 2000].
- **hybridation en cascade** — Les hybridations en cascade sont des exemples de processus de recommandation par étapes. Une première technique de recommandation est utilisée pour générer un classement grossier des items candidats, puis une seconde technique permet d’affiner cette première liste de candidats. Notons que cette hybridation est sensible à l’ordre des étapes : un CF-CBF produirait certainement des résultats différents d’un CBF-CF. Par exemple, les auteurs de [Lampropoulos et al., 2012] proposent un SR hybride (CBF-CF) en cascade dans le domaine musical. Un premier filtrage consiste à extraire les titres musicaux du même genre que ceux aimés par l’utilisateur, ces titres candidats sont ensuite affinés par un filtrage collaboratif avant de les proposer à l’utilisateur.

En général, le filtrage hybride utilise certaines méthodes pour combiner les ensembles de recommandations telles que la pondération, la cascade, la commutation, etc. afin de produire des recommandations finales pour les utilisateurs [Burke, 2002].

### **2.3. Évaluation des systèmes de recommandation**

L’objectif d’un système de recommandation est de proposer des items jugés « pertinents » pour un utilisateur cible, c’est-à-dire lui fournir une liste des  $N$  items ayant les scores prédits (pour cet utilisateur) les plus élevés. Rappelons ici qu’un système de recommandation procède en deux phases consécutives la prédiction et la recommandation. L’évaluation d’un système de recommandation peut donc porter sur l’une et/ou l’autre de ces phases et deux scénarios sont envisagés : l’évaluation basée sur les prédictions et l’évaluation basée sur la liste des top- $N$  items recommandés [Herlocker et al., 2004; Cremonesiet al., 2010; Steck, 2013; Gunawardana and Shani, 2015].

### 2.3.1. Évaluation basée sur la prédiction des évaluations (ratings)

Dans ce cas, on considère que la comparaison de la pertinence (accuracy) des SR est effectuée en fonction de leur capacité à prédire correctement (précisément) les valeurs des évaluations (ratings) des utilisateurs. Comme les évaluations (ratings) sont généralement représentés par des valeurs numériques, les métriques d'évaluation permettant de mesurer les erreurs de prédiction dans le cas des modèles de régression sont éligibles.

1. La métrique d'erreur quadratique moyenne **Root Mean Squared Error (RMSE)** est fréquemment utilisée pour évaluer l'erreur de prédiction globale, et elle est définie comme suit :

$$\text{RMSE} = \sqrt{\frac{\sum_{(u,i) \in K} (r_{u,i} - \widehat{r}_{u,i})^2}{K}} \quad (1.1)$$

$r_{u,i}$  est la vraie note donnée par  $u$  à  $i$ .

$\widehat{r}_{u,i}$  est la note prédite par le SR.

$K$  est l'ensemble des couples (**utilisateur, item**) pour lesquels la confrontation est effectuée.

2. La métrique d'erreur absolue moyenne (**EAM ou MAE pour Mean Absolute Error**) est la métrique la plus largement utilisée dans les systèmes de recommandation, qui estime la différence absolue moyenne entre l'évaluation et la prédiction. Lorsque la valeur de MAE est faible, le système de recommandation collaboratif est considéré comme efficace, et elle est définie comme suit :

$$\text{MAE} = \frac{\sum_{(u,i) \in K} |r_{u,i} - \widehat{r}_{u,i}|}{K} \quad (1.2)$$

### 2.3.2. Évaluation basée sur la liste de top-N recommandations

Contrairement aux évaluations basées sur des prédictions de notation, où tous paires  $(u,i) \in S_{\text{test}}$  sont considérés. L'évaluation discutée dans cette section ne considère pas Seuls les éléments qui apparaissent dans la liste des éléments recommandés mais il faut la prise en compte de toute la liste.

Formellement, laissez  $\mathbf{S}_{\text{test}}(\mathbf{u})$  désigner l'ensemble des éléments liés dans le jeu teste de l'utilisateur  $\mathbf{u}$ . Le projet associé d'un utilisateur peut être un projet Articles notés **npar** les utilisateurs (commentaires implicites) ou avec des valeurs de notation qui s'écartent Seuil dépassé (retour explicite). Soit  $\mathbf{L}_{\text{rec}}(\mathbf{u})$  la liste des éléments recommandés à l'utilisateur  $\mathbf{u}$ , où  $|\mathbf{L}_{\text{rec}}(\mathbf{u})| = \mathbf{N}$  est le nombre d'éléments recommandés. Mesurer la précision d'un système de recommandation basé sur l'évaluation de  $\mathbf{L}_{\text{rec}}(\mathbf{u})$ , par exemple : Précision, Rappel, Mesure F1, précision moyenne (MAP), gain cumulé actualisé normalisé (NDCG) et le taux de succès (HR).

Étant donné un utilisateur  $\mathbf{u} \in \mathbf{U}$ , la précision de la liste de recommandations de  $\mathbf{u}$  est définie par la formule (1.3). Il s'agit de la proportion d'éléments liés à l'utilisateur  $\mathbf{u}$  dans tous les articles recommandés pour l'utilisateur. D'autre part, Rappel par la formule (1.4) correspond au pourcentage d'items connexes recommandés,  $\mathbf{S}_{\text{test}}(\mathbf{u})$ .

$$\text{Précision}(\mathbf{L}_{\text{rec}}(\mathbf{u})) = \frac{|\mathbf{L}_{\text{rec}}(\mathbf{u}) \cap \mathbf{S}_{\text{test}}(\mathbf{u})|}{|\mathbf{L}_{\text{rec}}(\mathbf{u})|} \quad (1.3)$$

$$\text{Rappel}(\mathbf{L}_{\text{rec}}(\mathbf{u})) = \frac{|\mathbf{L}_{\text{rec}}(\mathbf{u}) \cap \mathbf{S}_{\text{test}}(\mathbf{u})|}{|\mathbf{S}_{\text{test}}(\mathbf{u})|} \quad (1.4)$$

F1-metric est un indicateur qui permet de combiner Précision et Rappel et Prendre la moyenne harmonique de ces deux mesures par la formule :

$$\mathbf{F1 - Mesure}(\mathbf{L}_{\text{rec}}(\mathbf{u})) = \frac{2 \cdot \text{Précision}(\mathbf{L}_{\text{rec}}(\mathbf{u})) \cdot \text{Rappel}(\mathbf{L}_{\text{rec}}(\mathbf{u}))}{\text{Précision}(\mathbf{L}_{\text{rec}}(\mathbf{u})) + \text{Rappel}(\mathbf{L}_{\text{rec}}(\mathbf{u}))} \quad (1.5)$$

## 2.4. Limites des systèmes de recommandation

Avec l'avènement du web et la masse importante d'information que contient, les systèmes de recommandation présentent un outil très important avec beaucoup d'avantages afin de faire face au problème de surcharge d'information et ainsi trouver l'information nécessaire pour les utilisateurs. Cependant, ces systèmes présentent un certain nombre de limites, on va citer dans ce qui suit deux importants problèmes de ces systèmes.

- **Démarrage à froid**, c'est un problème qui se pose avec l'arrivée d'un nouvel élément dans le système de recommandation soit un utilisateur soit un item. Le système doit le gérer comme un cas spécial, le problème de démarrage à froid c'est qu'on n'a pas assez d'informations pour émettre des recommandations [Rashid et al., 2002].
- **Parcimonie (Sparsity – rareté des données)**, le problème de rareté est lié à l'indisponibilité d'un grand nombre d'items notés pour chaque utilisateur actif. Le nombre d'items candidats à la recommandation est souvent énorme et les utilisateurs ne notent qu'un petit sous ensemble des items disponibles. De ce fait, la matrice des notes est une matrice creuse avec un taux de valeurs manquantes pouvant atteindre 95% du total des valeurs [Papagelis et al., 2005]. Par exemple, sur Amazon, si les utilisateurs actifs peuvent avoir acheté 1% des articles et que le montant total des articles est d'environ 2 millions de livres, cela signifie qu'il n'y a que 20 000 livres évalués. Par conséquent, une telle rareté dans les évaluations dégrade la sélection précise des voisins dans l'étape de formation du communauté et conduit à de mauvais résultats de recommandation [Adomavicius & Tuzhilin, 2005; Linden et al., 2003]

### 3. Conclusion

Dans ce chapitre, nous avons discuté des principes des systèmes de recommandation. Un SR est un outil efficace pour aider à identifier d'éventuels projets potentiels Intérêt, dans le contexte de la surcharge d'information à laquelle nous devons faire face. Un SR est capable de prédire les préférences de l'utilisateur en fonction de l'historique des interactions système de l'utilisateur. Les prédictions peuvent être basées sur les préférences des utilisateurs ciblés (méthode CBF) ou d'utilisateurs similaires (méthode CF). Une approche hybride est également proposée afin de bénéficier des avantages de chacune des méthodes et surmonter leurs lacunes.

Nous allons présenter dans le chapitre suivant les systèmes de filtrage collaboratif.

**CHAPITRE II**  
**SYSTEMES DE FILTRAGE**  
**COLLABORATIF**

## 1. Introduction

Dans ce chapitre nous allons décrire le système de filtrage collaboratif. Nous commençons d'abord par une présentation système et l'objectif de ses approches. Par la suite, nous allons présenter l'approches basées sur la mémoire ainsi qu'une description détaillée de chaque phase (identification du voisinage, prédiction des ratings). Le chapitre présente également l'approche basées sur les modèles.

## 2. Système de filtrage collaboratif (FC)

L'approche de filtrage collaboratif FC est basée sur les évaluations des items par un ensemble d'utilisateurs (voisins) dont les profils de notation sont les plus similaires à ceux de l'utilisateur cible. Ces méthodes supposent que si des utilisateurs ont les mêmes préférences sur un ensemble d'items, alors ils auront probablement les mêmes préférences sur un autre ensemble d'items qu'ils n'ont pas encore notés. Méthode de filtrage collaborative peuvent être divisées en deux catégories : basées sur la mémoire et basé sur un modèle.

### 2.1. Approches basées mémoire

L'approche de filtrage collaboratif basée sur la mémoire, également désignée par filtrage collaboratif basé sur le voisinage ou **Neighborhood- Collaborative Filtering (NBCF)**, fait partie des premiers algorithmes de filtrage [Resnick et al., 1994; Sarwar et al., 2001; Herlocker et al., 2002]. La stratégie principale de l'approche repose sur les hypothèses suivantes :

1. les utilisateurs similaires notent les items de façon analogue, autrement dit, leurs systèmes de notation sont similaires.
2. les items similaires reçoivent des évaluations (ratings) similaires.

On peut ainsi distinguer deux types d'approches :

- **NBCF** basé sur les utilisateurs, dans ce cas, les évaluations (ratings) fournis par les utilisateurs similaires à l'utilisateur  $\mathbf{u}$  sont utilisés pour faire des recommandations pour  $\mathbf{u}$ . La prédiction de  $\widehat{r_{\mathbf{u},i}}$  est calculée comme la

moyenne pondérée des évaluations (ratings) pour l'item  $i$  fournis par le voisinage de l'utilisateur  $u$ .

- **NBCF** basé sur les items. Dans ce cas, on identifie d'abord un ensemble d'items similaires à l'item  $i$ . Puis on prédit  $\widehat{r_{u,i}}$  en fonction des évaluations (ratings) spécifiques fournis par  $u$  pour ces items.

Qu'elles reposent sur les utilisateurs ou les items, les approches basées sur le voisinage (NBCF) sont constituées de deux étapes consécutives : \* l'identification du voisinage, \* la prédiction des évaluations (ratings) à partir du voisinage déterminé dans la première étape.

### 2.1.1. Identification du voisinage

A partir de la matrice des évaluations (ratings)  $\mathbf{R}$ , l'idée de base des approches de **NBCF** est de faire la recommandation en fonction de la similarité entre utilisateurs ou entre items. Cela implique que l'on doit déterminer soit des utilisateurs similaires, soit des items similaires, i.e. identifier le voisinage. Pour ce faire, l'approche des  $k$  plus proches voisins (**KNN** pour  $k$  nearest neighbors) est utilisée. Généralement, comme les utilisateurs et les items sont représentés par des vecteurs des évaluations (ratings) dans la matrice  $\mathbf{R}$ , les métriques standardisées permettant de calculer la proximité entre des vecteurs peuvent être utilisées. Les représentations vectorielles des utilisateurs  $u \in U$  et des items  $i \in I$  sont obtenues à partir de la matrice  $\mathbf{R}$ . Par exemple, le vecteur pour l'utilisateur  $u$ , est défini comme  $\vec{u} = (r_{u,i}, i = 1, 2, \dots, |I|)$ . De manière similaire, le vecteur d'un item  $i$  est défini comme  $\vec{i} = (r_{u,i}, u = 1, 2, \dots, |U|)$ . Nous introduisons par la suite quelques métriques courantes pour le calcul de similarité entre vecteurs dans le contexte du **NBCF**.

**2.1.1.1. Similarité cosinus (COS) :** est une métrique largement utilisée dans le domaine de la recherche d'information et de la fouille de textes pour calculer la similarité entre deux documents représentés par la fréquence des mots qu'ils contiennent [Crnic, 2011]. Elle représente le cosinus de l'angle entre les deux vecteurs représentant ces documents dans l'espace à  $n$  dimensions, avec  $n$  étant la longueur des vecteurs. Dans le contexte du **NBCF**, elle peut être employée pour mesurer la similarité entre deux utilisateurs ou entre deux items.

Formellement, la similarité cosinus entre deux utilisateurs est définie par l'équation (2.1) :

$$\text{sim}_{\text{cos}}(u_1, u_2) = \frac{\sum_{i \in I_{u_1, u_2}} r_{u_1, i} * r_{u_2, i}}{\sqrt{\sum_{i \in I_{u_1, u_2}} r_{u_1, i}^2} \sqrt{\sum_{i \in I_{u_1, u_2}} r_{u_2, i}^2}} \quad (2.1)$$

$I_{u_1, u_2}$  représente l'ensemble des items notés par l'utilisateur  $u_1$  et l'utilisateur  $u_2$ . De même, la similarité entre deux éléments est définie par l'équation (2.2), où  $U_{i, j}$  désigne l'ensemble des utilisateurs qui ont évalué à la fois l'élément  $i$  et l'élément  $j$ .

$$\text{sim}_{\text{cos}}(i, j) = \frac{\sum_{u \in U_{i, j}} r_{u, i} * r_{u, j}}{\sqrt{\sum_{u \in U_{i, j}} r_{u, i}^2} \sqrt{\sum_{u \in U_{i, j}} r_{u, j}^2}} \quad (2.2)$$

### 2.1.1.2. Coefficient de corrélation Pearson (PCC)

Une autre métrique de similarité largement utilisée dans ce contexte est le coefficient de corrélation de Pearson ou **Pearson Correlation Coefficient (PCC)** [Resnick et al., 1994]. PCC est généralement employé en statistique pour mesurer une corrélation entre deux variables, ses valeurs varient entre -1 et 1. La valeur de 1 traduit une corrélation parfaite et positive alors que la valeur de -1 signifie que les deux variables sont totalement et négativement corrélées (anti-corrélées). La valeur de 0 signifie qu'il n'y a pas de corrélation linéaire entre les deux variables. Formellement, pour mesurer la similarité entre deux utilisateurs  $u_1$  et  $u_2$ , PCC est défini par l'équation (2.3) :

$$\text{sim}_{\text{PCC}}(u_1, u_2) = \frac{\sum_{i \in I_{u_1, u_2}} [(r_{u_1, i} - \bar{r}_{u_1})(r_{u_2, i} - \bar{r}_{u_2})]}{\sqrt{\sum_{i \in I_{u_1, u_2}} (r_{u_1, i} - \bar{r}_{u_1})^2} \sqrt{\sum_{i \in I_{u_1, u_2}} (r_{u_2, i} - \bar{r}_{u_2})^2}} \quad (2.3)$$

où  $\bar{r}_{u_1}$  et  $\bar{r}_{u_2}$  représentent respectivement la valeur moyenne des évaluations (ratings) de l'utilisateur  $u_1$  et de l'utilisateur  $u_2$ .

La similarité entre deux items est, quant à elle, définie par l'équation (2.4), avec  $\bar{r}_i$  et  $\bar{r}_j$  représentant respectivement la valeur moyenne des évaluations (ratings) fournis par tous les utilisateurs pour l'item  $i$  et l'item  $j$ .

$$\text{sim}_{\text{PCC}}(i, j) = \frac{\sum_{u \in U_{i,j}} [(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)]}{\sqrt{\sum_{u \in U_{i,j}} (r_{u,i} - \bar{r}_i)^2 \sum_{u \in U_{i,j}} (r_{u,j} - \bar{r}_j)^2}} \quad (2.4)$$

Contrairement à la mesure directe de la similarité entre les vecteurs, tels que COS et PCC, d'autres méthodes calculent d'abord la distance entre les vecteurs ensuite, la convertir en similarité. Pour cela, la distance dans l'espace euclidien est généralement utilisée. Formellement, étant donné deux vecteurs  $\vec{a}$  et  $\vec{b}$  de dimension  $n$  la distance entre eux est définie comme suit :

$$d(\vec{a}, \vec{b}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2} \quad (2.5)$$

### 2.1.1.3. Distance quadratique moyenne (MSD)

Dans des nombreuses applications, et en particulier lors de la comparaison des distances, il peut être plus pratique d'inclure la racine carrée dans le calcul des distances euclidiennes (2.6). La distance résultante est alors appelée distance euclidienne quadratique [Spencer, 2013]. Dans le contexte du NBCF, on peut considérer deux utilisateurs (resp. deux items) comme deux points positionnés dans l'espace Euclidien de dimension  $I_{u_1, u_2}$  (resp.  $U_{i,j}$ ). Ainsi, la distance peut être calculée par la distance euclidienne quadratique. Pourtant, la distance calculée de cette manière a tendance d'être biaisée car les utilisateurs qui ont noté beaucoup d'items sembleront plus éloignés que ceux pour lesquels il n'y a que peu d'items notés en commun. De ce fait, il est préférable d'utiliser la distance quadratique moyenne **Mean Squared Distance (MSD)**. Formellement, pour calculer la distance entre deux utilisateurs  $u_1$  et  $u_2$  (resp. deux items  $i$  et  $j$ ), **MSD** est définie par l'équation (2.7) (resp. l'équation (2.9)).

$$d(\vec{a}, \vec{b}) = (a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2 \quad (2.6)$$

1) pour deux utilisateurs  $u_1$  et  $u_2$

$$\text{MSD}(u_1, u_2) = \frac{\sum_{i \in I_{u_1, u_2}} (r_{u_1, i} - r_{u_2, i})^2}{I_{u_1, u_2}} \quad (2.7)$$

$$\text{sim}_{\text{MSD}}(u_1, u_2) = \frac{1}{\text{MSD}(u_1, u_2) + 1} \quad (2.8)$$

2) pour deux items  $i$  et  $j$

$$\text{MSD}(i, j) = \frac{\sum_{u \in U_{i, j}} (r_{u, i} - r_{u, j})^2}{U_{i, j}} \quad (2.9)$$

$$\text{sim}_{\text{MSD}}(i, j) = \frac{1}{\text{MSD}(i, j) + 1} \quad (2.10)$$

Afin de convertir une distance  $\mathbf{d}$  en similarité  $\mathbf{sim}$ , différentes stratégies peuvent être appliquées, comme la mesure de son inverse, i.e.  $\mathbf{d}^{-1}$  ou la formule  $\mathbf{1} - \mathbf{d}$  lorsque la distance  $\mathbf{d}$  est normalisée dans l'intervalle  $[0, 1]$ . L'équation (2.8) représente, par exemple, la similarité entre deux utilisateurs  $\mathbf{u}$  et  $\mathbf{v}$  convertie à partir de la distance de  $\mathbf{MSD}(\mathbf{u}, \mathbf{v})$ . Notons que dans ce cas, 1 est ajouté au dénominateur pour éviter une éventuelle division par 0.

### 2.1.2. Prédiction des évaluations (ratings)

La deuxième étape de l'approche du **NBCF** est la prédiction des évaluations (ratings) à partir du voisinage déterminé lors de la première étape. Concrètement, il s'agit de calculer soit la moyenne pondérée des évaluations fournis par les individus du voisinage d'un utilisateur dans le cas du NBCF basé sur les utilisateurs, soit la moyenne pondérée des évaluations associés aux éléments du voisinage d'un item donné dans le cas du NBCF basé sur les items. Considérons, par exemple, la prédiction de  $\widehat{r}_{u, i}$ , dans le premier cas  $\widehat{r}_{u, i}$  est alors obtenu en prenant la moyenne pondérée des évaluations de l'item  $i$  donnés par chacun des  $\mathbf{k}$  plus proches voisins de l'utilisateur  $\mathbf{u}$  ayant évalué  $\mathbf{i}$ .

La pondération des évaluations est fonction de similarité entre l'utilisateur  $\mathbf{u}$  et ses voisins, autrement dit, plus un voisin est similaire à  $\mathbf{u}$ , plus le poids associé à l'évaluation de ce voisin est grand. Formellement, la prédiction de  $\widehat{r_{\mathbf{u},\mathbf{i}}}$  est obtenue par l'équation (2.11) pour l'approche de NBCF basée sur les utilisateurs, où le terme  $\mathbf{N}_{\mathbf{u}}(\mathbf{i})$  représente les plus proches voisins de l'utilisateur  $\mathbf{u}$  ayant noté l'item  $\mathbf{i}$ . De façon similaire, dans le cas où la prédiction est basée sur les items, la valeur de  $\widehat{r_{\mathbf{u},\mathbf{i}}}$  sera obtenue par l'équation (2.12) où le terme  $\mathbf{N}_{\mathbf{i}}(\mathbf{u})$  représente l'ensemble des  $\mathbf{k}$  plus proches voisins de l'item  $\mathbf{i}$  ayant été notés par  $\mathbf{u}$ .

$$\widehat{r_{\mathbf{u},\mathbf{i}}} = \frac{\sum_{\mathbf{u}_2 \in \mathbf{N}_{\mathbf{u}}(\mathbf{i})} r_{\mathbf{u}_2,\mathbf{i}} * \text{sim}(\mathbf{u}_1, \mathbf{u}_2)}{\sum_{\mathbf{u}_2 \in \mathbf{N}_{\mathbf{u}}(\mathbf{i})} \text{sim}(\mathbf{u}_1, \mathbf{u}_2)} \quad (2.11)$$

$$\widehat{r_{\mathbf{u},\mathbf{i}}} = \frac{\sum_{\mathbf{j} \in \mathbf{N}_{\mathbf{i}}(\mathbf{u})} r_{\mathbf{u},\mathbf{j}} * \text{sim}(\mathbf{i}, \mathbf{j})}{\sum_{\mathbf{j} \in \mathbf{N}_{\mathbf{i}}(\mathbf{u})} \text{sim}(\mathbf{i}, \mathbf{j})} \quad (2.12)$$

En général, la façon dont les utilisateurs évaluent les items est influencée par leur personnalité, leur humeur et aussi le contexte. Ainsi, sur une échelle d'évaluation de 1 à 5, un utilisateur optimiste (resp. Pessimiste) donnera rarement une évaluation de 1 (resp. 5) à l'item même s'il ne l'aime pas (resp. S'il l'aime). Les distributions des évaluations des utilisateurs peuvent donc être décalées ou comprimées les unes par rapport aux autres. Une distribution de évaluations  $r_x$  est notamment caractérisée par la moyenne ( $\bar{r}_x$ ) de ces évaluations et leur variance ( $\sigma_x$ ). Pour prendre en compte la disparité des systèmes de notation des utilisateurs, la normalisation des évaluations est généralement employée. Les auteurs [Herlocker et al., 2002] ont proposé de normaliser les évaluations de chaque utilisateur. Deux normalisations sont envisagées : la normalisation par la moyenne (centré) et la normalisation par le z-score (centré et réduit).

Formellement, dans la première forme de normalisation par la moyenne, les équations (2.11) et (2.12) sont remplacées par les équations (2.13) et (2.14).

$$\widehat{r_{u_1,1}} = \overline{r_{u_1}} + \frac{\sum_{u_2 \in N_{u(i)}} [(r_{u_2,i} - \overline{r_{u_2}}) * \text{sim}(u_1, u_2)]}{\sum_{u_2 \in N_{u(i)}} \text{sim}(u_1, u_2)} \quad (2.13)$$

$$\widehat{r_{u_1,1}} = \overline{r_1} + \frac{\sum_{j \in N_{i(u)}} [(r_{u,j} - \overline{r_j}) * \text{sim}(i,j)]}{\sum_{j \in N_{i(u)}} \text{sim}(i,j)} \quad (2.14)$$

Pour la deuxième forme de normalisation par les scores  $z$ , l'équation (2.11) et (2.12) sont remplacées par les équations (2.15) et (2.16).

$$\widehat{r_{u_1,1}} = \overline{r_{u_1}} + \sigma_x \frac{\sum_{u_2 \in N_{u(i)}} [(\frac{r_{u_2,i} - \overline{r_{u_2}}}{\sigma_x}) * \text{sim}(u_1, u_2)]}{\sum_{u_2 \in N_{u(i)}} \text{sim}(u_1, u_2)} \quad (2.15)$$

$$\widehat{r_{u_1,1}} = \overline{r_1} + \sigma_x \frac{\sum_{j \in N_{i(u)}} [(\frac{r_{u,j} - \overline{r_j}}{\sigma_x}) * \text{sim}(i,j)]}{\sum_{j \in N_{i(u)}} \text{sim}(i,j)} \quad (2.16)$$

### 2.1.3. Un exemple concret

Dans cette section, nous illustrons la procédure des approches du NBCF par un exemple concret de la prédiction des évaluations. On considère la matrice des évaluations représentée par table 2.1, où nous observons 7 utilisateurs évaluant 7 éléments. Notons que cette matrice est incomplète, étant donné qu'il y a des cases où les évaluations sont inconnues. L'objectif ici est de prédire la valeur de l'évaluations de l'utilisateur  $u_1$  à l'élément  $i_2$  est prédite par la méthode de filtrage collaboratif basée sur le voisinage, à savoir  $\widehat{r_{u_1, i_2}}$  (boîte bleu). Notons que différentes configurations de l'algorithme sont possibles selon par exemple le type de voisins choisi, leur nombre, la métrique de similarité, la méthode de normalisation, etc. Pour l'illustration, nous considérons : l'approche de **NBCF** basée sur les utilisateurs, avec la métrique **PCC** (2.3) pour l'identification du

voisinage et la normalisation par la moyenne (2.13) pour la prédiction des évaluations. Pour prédire  $\widehat{r_{u_1, i_2}}$ , la première étape consiste à identifier le voisinage de l'utilisateur  $u_1$  par rapport à  $i_2$ , c'est-à-dire  $N_{u_1(i_2)}$ . Dans ce cas,  $u_6$  ne fait pas partie du voisinage car il n'a pas évalué l'item  $i_2$ . Il suffit donc de calculer la similarité entre Utilisateur  $u_1$  et utilisateurs  $u_2, u_3, u_4, u_5$  et  $u_7$ . Par application on (2.3).

	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$U_1$	1		3	1			2
$U_2$	2	1		5	5		
$U_3$		2	3				
$U_4$	1	2					
$U_5$	2	5	3	1			1
$U_6$	1		3	1		5	2
$U_7$	3	4	1		4		2

Table 2.1 – Exemple d'une matrice des évaluations pour illustrer l'approche NBCF.

on obtient :

$$\text{sim}_{\text{pcc}}(u_1, u_2) = \frac{(1-1.75)(2-3.25)+(1-1.75)(5-3.25)}{\sqrt{[(1-1.75)^2+(1-1.75)^2][(2-3.25)^2+(5-3.25)^2]}}$$

$$\text{sim}_{\text{PCC}}(u_1, u_2) = -0.16$$

De même, nous obtenons d'autres valeurs de similarité :  $\text{sim}_{\text{PCC}}(u_1, u_3) = 1.00$ ,  $\text{sim}_{\text{PCC}}(u_1, u_4) = 1.00$ ,  $\text{sim}_{\text{PCC}}(u_1, u_5) = 0.50$ ,  $\text{sim}_{\text{PCC}}(u_1, u_7) = -1.00$ .

Une fois ces similarités calculées, la deuxième étape consiste à prédire la valeur de  $\widehat{r_{u_1, i_2}}$  en s'appuyant sur les évaluations de l'item  $i_2$  fournis par les utilisateurs appartenant au voisinage de l'utilisateur  $u_1$ . Notons que pour cet exemple, et étant donné le faible nombre d'utilisateurs, on considérera un voisinage de taille 3, mais dans la pratique,  $k$  est généralement plus grand. De ce fait, les 3 voisins de  $u_1$

considérés pour obtenir  $\widehat{r}_{u_1, i_2}$  sont  $u_3, u_4, u_5$ , c'est-à-dire  $N_{u_1(i_2)} = \{u_3, u_4, u_5\}$ . En appliquant la formule de prédiction de l'équation (2.13), on obtient :

$$\begin{aligned}\widehat{r}_{u_1, i_2} &= \overline{r}_{u_1} + \frac{\sum_{v \in N_{u_1(i_2)}} [(r_{v, i_2} - \overline{r}_v) * \text{sim}(u_1, v)]}{\sum_{v \in N_{u_1(i_2)}} \text{sim}(u_1, v)} \\ &= 1.75 + \frac{(2-2.5)*1.00 + (2-1.5)*1.00 + (5-2.4)*0.50}{1.00+1.00+0.50} \\ \widehat{r}_{u_1, i_2} &= 2.26\end{aligned}$$

Notons ici que l'évaluation prédit n'est pas un entier comme les autres valeurs de la matrice (Table 2.1), qui correspondent à des évaluations réels et non prédits. Comme nous le présenterons dans le chapitre 1, ces prédictions servent généralement à évaluer l'accuracy d'un SR donné ou à ordonner les items candidats pour un utilisateur donné.

## 2.2. Approches basées sur les modèles :

Les approches basées sur le voisinage, présentées dans la section précédente, ont été parmi les premières méthodes de filtrage collaboratif et ont également été parmi les plus populaires. Si elles gardent l'avantage de leur simplicité, elles ne sont pas nécessairement les méthodes les plus pertinentes pour la prédiction des évaluations.

Contrairement aux approches **NBCF** qui peuvent être considérées comme des méthodes spécifiques aux instances, une instance étant la valeur de  $\widehat{r}_{u, i}$  à prédire, il existe dans la littérature des approches de CF basées sur les modèles, dans lesquelles il y a une claire distinction entre la phase d'entraînement et la phase de prédiction. Plus précisément, dans le cas des approches **NBCF**, un modèle n'est pas spécifiquement créé à l'avance pour la prédiction autre qu'une phase de pré-traitement (e.g. calculs des similarités, calculs des prédictions basées sur des équations prédéfinies etc.). A l'inverse, pour les approches basées sur les modèles, un modèle d'apprentissage automatique (supervisé en général) est d'abord créé. Les paramètres de ce modèle sont ajustés sur la base des évaluations disponibles

dans le jeu de données, en optimisant une fonction objective. On parle d'entraînement du modèle.

Les modèles classiques d'apprentissage automatique tels que les arbres de décision, les machines à vecteurs de support ou **Support Vector Machine (SVM)**, ou les réseaux de neurones ont été largement utilisés pour résoudre des problèmes pratiques tels que la classification, la régression, etc. Ces modèles d'apprentissage automatique peuvent tous être adaptés pour résoudre le problème de la prédiction des évaluations, i.e. la complétion de la matrice de (ratings). En effet, on peut considérer la prédiction des (ratings) comme une généralisation des autres problématiques évoquées. Dans un cadre de classification, par exemple, il s'agit également de compléter une matrice, dans laquelle tous les champs sauf ceux de la dernière colonne (la variable de classe) sont remplis par des valeurs spécifiques (i.e. les caractéristiques des jeux de données). Plus généralement, dans le cas du filtrage collaboratif, chaque case de la matrice dont on dispose peut-être vide. Par conséquent, le problème de la prédiction des (ratings) peut être considéré comme une généralisation de la classification ou de la régression.

De nombreux modèles ont été appliqués dans le contexte du filtrage collaboratif pour prédire les (ratings). On peut citer les arbres de décision, les modèles basés sur les règles d'association, le modèle dit bayésien naïf, les modèles basés sur la factorisation des matrices (également connu sous la dénomination de facteur latent) et les modèles basés sur les réseaux de neurones [Aggarwal, 2016f].

### 3. Conclusion

Dans ce chapitre, nous avons discuté des approches de système de filtrage collaboratif. (SFC), comme étant des outils efficaces pour aider à identifier des voisins et capable de prédire des évaluations aux utilisateurs. En effet, nous sommes concentrés sur la façon de calculer la similarité et à l'aide de ce dernier on peut le calcule la prédiction de note a compagne avec un exemple concret, le chapitre suivant sera consacré la factorisation matricielle à laquelle nous allons factorisés notre proposition.

**CHAPITRE III**  
**FACTORISATION**  
**MATRICIELLE**

## 1. Introduction

Afin d'entrer dans le vif sujet, ce chapitre présente les techniques adoptées en factorisation matricielle. Un tel système de recommandation manipule des millions voire des billions d'objets (utilisateurs, items, évaluations, commentaires, données personnelles et démographiques, ...) et fait recours à des calculs matriciels avec des dimensions ultra élevées. Alors, le passage à des facteurs latents s'avère nécessaire pour assimiler et réduire les dimensions pour mieux avoir des prédictions pertinentes en temps raisonnable. Des notions mathématiques de base sont détaillées à savoir la factorisation en matrices non négatives, la décomposition en valeurs singulières et l'analyse des composants principaux.

## 2. Factorisation de matrices FM

Parmi les méthodes à facteurs latents, la factorisation de matrices (FM, matrix factorization) est devenue très populaire en filtrage collaboratif à la suite du concours « Netflix Prize » [J. Bennett, S. Lanning et al., 2007; R. M. Bell et Y. Koren., 2007]. La FM consiste à chercher une approximation de rang faible  $\hat{\mathbf{R}}$  de la matrice de (rating)  $\mathbf{R}$ . Autrement dit, on s'intéresse à l'approximation suivante :

$$\mathbf{R} \approx \mathbf{UV}^T = \hat{\mathbf{R}} \quad (3.1)$$

### 2.1 Définition La matrice de factorisation

Est une matrice d'évaluation  $\mathbf{R}$  qui permet d'identifier les utilisateurs et ses interactions sous forme des (ratings) par rapport aux items dans une matrice de taille  $(\mathbf{m} \times \mathbf{n})$  tel que  $\mathbf{m}$  les (ratings) des utilisateurs et  $\mathbf{n}$  la liste des items. Son principe est de décomposer une matrice en deux ou plusieurs autres matrices, et de calculer par la suite le produit de ces matrices pour obtenir la matrice des prédictions.

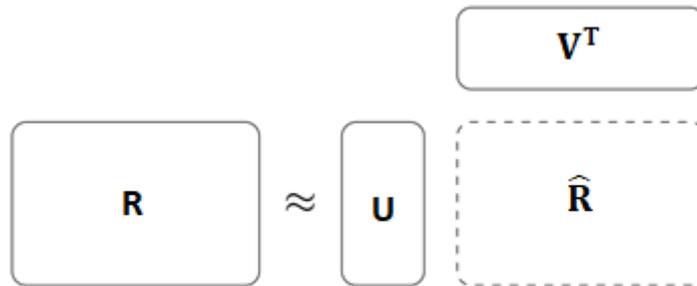


Figure 3.1 – Illustration de la factorisation de matrices.

## 2.2. Complétion de matrices

Les premiers travaux de **CF** se concentraient sur le traitement de données explicites. exister dans ce cas, le but du système de recommandation est de remplir les valeurs manquantes Matrice  $\mathbf{R}$  [E. J. Candès et B. Recht.,2009; E. J. Candès et Y. Plan.,2010]. Les prédictions de font alors à l'aide de l'approximation de rang faible  $\hat{\mathbf{R}}$ .

Les auteurs [B.Sarwar, G. Karypis, J. Konstan et J. Rieddl , 2000] proposent de compléter artificiellement les valeurs manquantes de la matrice  $\mathbf{R}$  avant d'appliquer la décomposition, mais ces méthodes sont coûteuses et sujettes au sur-apprentissage.

## 2.3. Principe de fonctionnement du FM

La plupart des modèles de factorisation matricielle sont basés sur le modèle du facteur latent. C'est l'approche la plus utilisé pour réduire le problème de sparsity et le manque de données dans la matrice d'un système de recommandation [Ain Picot-Clemente, 2011].

Tout d'abord, on a un ensemble d'utilisateurs  $\mathbf{U}$  et un ensemble des items  $\mathbf{I}$ .

Soit  $\mathbf{R}$  la matrice de taille  $(\mathbf{m} \times \mathbf{n})$  qui contient tous les (ratings) que les utilisateurs ont attribuées aux items (rating matrix).

La méthode consiste à trouver deux matrices  $\mathbf{U}$   $(\mathbf{m} \times \mathbf{k})$  et  $\mathbf{V}$   $(\mathbf{k} \times \mathbf{n})$ , tel que  $\mathbf{k}$  est le nombre facteur latent (un entier positif) et leur produit qui se rapproche étroitement de la matrice originale  $\mathbf{R}$ .

L'objectif de cette méthode est d'apprendre les vecteurs  $\mathbf{U}$  et  $\mathbf{V}$  de la matrice de (rating)  $\mathbf{R}$ , où  $\mathbf{U}$  exprime l'évaluation d'un élément en termes de facteurs  $\mathbf{k}$  et  $\mathbf{V}$  exprime l'intérêt des utilisateurs pour les facteurs.

L'identification de facteurs latents peut se faire à travers plusieurs modèles de factorisation matricielle, dont certains sont : la décomposition en valeur singulière (**SVD**), l'analyse des composantes principales (**PCA**), factorisation matricielle non négative (**NMF**) etc.

## 2.4. Les modèles de la factorisation matricielle

### 2.4.1. La décomposition de la valeur singulière SVD

La décomposition en valeur singulière **SVD** (Singular Value Decomposition) est la technique la plus puissante de réduction de la dimensionnalité [Ilham Esslmani, 2010].

Le modèle **SVD** est parmi les modèles de factorisation matricielle les plus populaires dans le contexte du filtrage collaboratif et la technique la plus puissante de réduction de la dimensionnalité. L'idée principale d'un modèle de factorisation matricielle est de considérer la matrice des (ratings)  $\mathbf{R}$  comme une approximation de la multiplication de deux matrices. **SVD** d'une matrice  $\mathbf{R}$  de taille  $\mathbf{m} \times \mathbf{n}$  est de la forme :  $\mathbf{R} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^T$  où  $\mathbf{U}$  et  $\mathbf{V}$  sont deux matrices orthogonales de taille  $(\mathbf{m} \times \mathbf{k})$  et  $(\mathbf{k} \times \mathbf{n})$  où  $\mathbf{k}$  est le rang de la matrice  $\mathbf{R}$ .  $\mathbf{\Sigma}$  est une matrice  $\mathbf{k} \times \mathbf{k}$  et contient toutes les valeurs singulières de  $\mathbf{R}$  comme ses entrées diagonales.

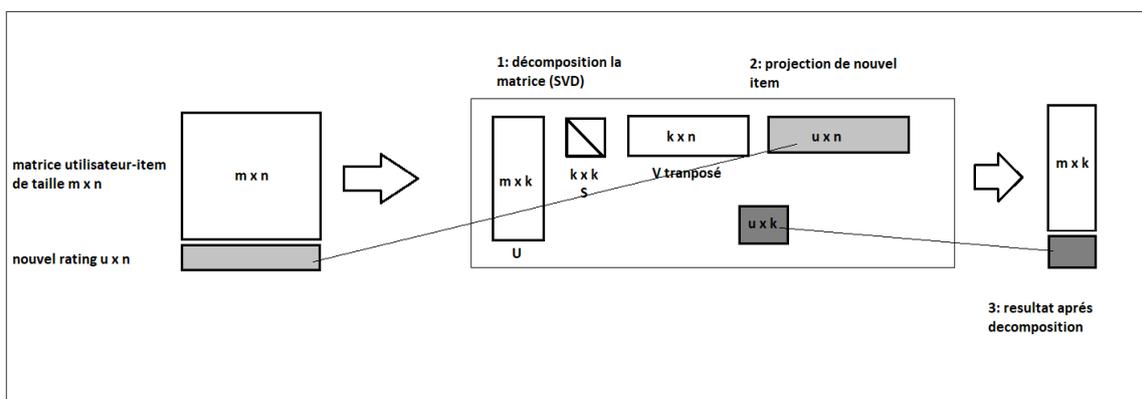


Figure 3.2 – La Décomposition en valeur singulière SVD [Bokde D., Girase S., Mukhopadhyay D.,2014].

La dimensionnalité des matrices peut être réduite en choisissant  $\Sigma$  pour ne contenir que les  $k$  plus grandes valeurs singulières de  $\mathbf{R}$ . Cela fait de  $\mathbf{S}_k$  une matrice  $k \times k$ , et puisque la réduction peut également être appliquée aux matrices  $\mathbf{U}$  et  $\mathbf{V}$  en supprimant les colonnes  $(\mathbf{r}_k)$  des lignes  $\mathbf{U}$  et  $(\mathbf{r}_k)$  de  $\mathbf{V}$ , la dimensionnalité de  $\mathbf{U}\Sigma\mathbf{V}^T$  peut être considérablement réduite en  $\mathbf{U}_k\Sigma_k\mathbf{V}_k^T$ , ce qui permet d'économiser de l'espace et de la puissance de calcul en plus de supprimer les données de bruit de  $\mathbf{R}$  [Ain Picot-Clemente, 2011]. Le résultat de l'exécution de **SVD** sur  $\mathbf{R}$  est donc  $\mathbf{R}_k$ , une version de dimension réduite de  $\mathbf{R}$  [Ilham Esslimani, 2010].

Il est à noter que l'exemple de le modèle SVD sera détaillée en chapitre 4.

#### 2.4.2. Factorisation matricielle non négative NMF

La **NMF** est une technique de réduction de dimension adaptée aux matrices creuses contenant des données positives, par exemple des occurrences ou dénombrements de mots, la méthode est donc plus adaptée à certaines situations que la **SVD**.

La factorisation non-négative de la matrice ( $\mathbf{R}_{m \times n}$ ) est la recherche de deux matrices ( $\mathbf{U}_{m \times k}$ ) et ( $\mathbf{V}_{k \times n}$ ) [D. D. Lee et H. S. Seung, 1999].ne contenant que des valeurs positives ou nulles et dont le produit approche  $\mathbf{R}$ .

$$\mathbf{R} \approx \mathbf{UV} \quad (3.2)$$

Le choix du rang de factorisation  $\mathbf{K} \ll \min(\mathbf{m}, \mathbf{n})$  assure une réduction de dimension et donc des représentations parcimonieuses. Évidemment, la qualité d'approximation dépend de la parcimonie de la matrice initiale.

$$\min_{\mathbf{U}, \mathbf{V}} = D(\mathbf{R}|\mathbf{UV}^T); \mathbf{U}, \mathbf{V} \geq 0 \quad (3.3)$$

où  $\mathbf{u} \geq 0$  exprime le fait que chaque coefficient de la matrice  $\mathbf{U}$  est positif, et  $\mathbf{D}$  est une divergence mesurant la dissemblance entre  $\mathbf{R}$  et  $\mathbf{UV}$ , telle que  $D(\mathbf{R}|\mathbf{UV}^T) = \sum_{ui} d(\mathbf{r}_{ui} | [\mathbf{UV}^T]_{ui})$ . Contrairement à une distance, une divergence

peut ne pas être symétrique. La divergence entre deux scalaires est toujours positive, i.e.,  $d(y|x) \geq 0$  et est nulle uniquement en  $x = y$ .

Une fois la factorisation construite, il est ensuite facile d'utiliser ces matrices  $U$  et  $V$  pour faire des prédictions à l'utilisateur cible.

**Exemple** on a la matrice de (rating)  $R$  de taille (6x4) suivant

	<b>I1</b>	<b>I2</b>	<b>I3</b>	<b>I4</b>
<b>U1</b>	5.0	0.0	3.0	5.0
<b>U2</b>	4.0	5.0	0.0	0.0
<b>U3</b>	0.0	5.0	0.0	3.0
<b>U4</b>	4.0	0.0	3.0	1.0
<b>U5</b>	0.0	3.0	5.0	5.0
<b>U6</b>	4.0	4.0	5.0	0.0

Table 3.1 – Exemple des évaluations Utilisateurs / item.

1. on applique l'algorithme de **NMF** on trouve

	<b>X1</b>	<b>X2</b>	<b>X3</b>
<b>Y1</b>	10.06	3.38	1.61
<b>Y2</b>	1.23	7.35	2.25
<b>Y3</b>	9.25	0.41	3.50
<b>Y4</b>	4.42	5.91	2.12
<b>Y5</b>	3.67	5.33	2.08
<b>Y6</b>	3.70	4.91	3.03

Table 3.2 – matrice  $U$ .

	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>
<b>U1</b>	0.29	0.46	0.10	0.38
<b>U2</b>	0.37	0.27	0.29	0.42
<b>U3</b>	0.34	0.14	0.27	0.06

Table 3.3 – matrice  $V^T$ .

2. La matrice finale avec les prédictions sera

	<b>I1</b>	<b>I2</b>	<b>I3</b>	<b>I4</b>
<b>U1</b>	<b>5.0</b>	<b>5.87</b>	<b>3.0</b>	<b>5.0</b>
<b>U2</b>	<b>4.0</b>	<b>5.0</b>	<b>2.95</b>	<b>3.79</b>
<b>U3</b>	<b>4.12</b>	<b>5.0</b>	<b>2.05</b>	<b>3.0</b>
<b>U4</b>	<b>4.0</b>	<b>3.98</b>	<b>3.0</b>	<b>1.0</b>
<b>U5</b>	<b>3.79</b>	<b>3.0</b>	<b>5.0</b>	<b>5.0</b>
<b>U6</b>	<b>4.0</b>	<b>4.0</b>	<b>5.0</b>	<b>3.73</b>

Table 3.4 – matrice des prédictions de NMF.

### 2.4.3. Analyse des composants principaux PCA

L'analyse des composantes principales (Principal Component Analysis PCA) est une réalisation particulière de l'approche de la factorisation de la matrice (FM).

L'PCA est une procédure statistique qui utilise une transformation orthogonale pour convertir un ensemble d'observations de variables éventuellement corrélées en un ensemble de valeurs de variables linéairement non corrélées appelées composantes principales. Le nombre de variables d'origine est supérieur ou égal aux composants principaux.

PCA est sensible à la mise à l'échelle relative des variables d'origine. PCA permet d'obtenir une liste ordonnée de composants qui représentent la plus grande quantité de variance à partir des données en termes d'erreurs de moindres carrés [S.Taouli, W.Benachenhou.,2017].

La quantité de variance saisie par le premier composant est supérieure à la quantité de variance sur le deuxième composant et ainsi de suite. Nous pouvons réduire la dimensionnalité des données en négligeant ces composants [Bokde D., Girase S., Mukhopadhyay D.,2014].

**L'analyse en composantes principales**, proposée initialement par (Pearson, 1901), considère les lignes d'une matrice de données comme des observations d'un vecteur aléatoire et les colonnes comme les variables. On cherche à les données par des variables non-observées qu'on appelle composantes principales. La PCA crée un lien entre les vecteurs des matrices ( $\mathbf{R}_{m \times n}$ ) et ( $\mathbf{U}_{m \times k}$ ) à l'aide de la matrice ( $\mathbf{V}_{k \times n}$ ). On cherche alors les  $K$  composantes principales qui maximisent la variance. Ces composantes sont des nouveaux axes. On peut effectuer cette tâche de plusieurs manières entre autres à l'aide de la décomposition en valeurs singulières ou de la décomposition en valeurs propres. Pour plus de détails [Wold et al., 1987].

Pour mieux comprendre on applique PCA sur matrice (Table 3.1)

1. On fait l'analyse statistique des items

1.1. On calcule la moyenne de chaque item par exemple **I1**

$$moyenne_{I_1} = \frac{5+4+4+4}{4} = 4.25$$

1.2. On calcule l'écart type de chaque item par exemple **I1**

$$Ecart\_Type_{I_1} = \sqrt{\frac{(5-4.25)^2 - (4-4.25)^2 - (4-4.25)^2 - (4-4.25)^2}{4}} = 0.5$$

Items	Moyenne	Ecart_type	Maximum de l'évaluations	Minimum de l'évaluations
<b>I1</b>	<b>4.25</b>	<b>0.5</b>	<b>5.0</b>	<b>4.0</b>
<b>I2</b>	<b>4.24</b>	<b>0.95</b>	<b>5.0</b>	<b>3.0</b>
<b>I3</b>	<b>4.00</b>	<b>1.15</b>	<b>5.0</b>	<b>3.0</b>
<b>I4</b>	<b>3.50</b>	<b>1.91</b>	<b>5.0</b>	<b>1.0</b>

Table 3.5 – analyse statistique des items.

2. On calcule la matrice des données centrées réduite  $X_c$  suivant cette formule

$$I_i = \frac{I_i - \text{Moyenne}}{\text{Ecart\_type}}$$

	<b>I1</b>	<b>I2</b>	<b>I3</b>	<b>I4</b>
<b>U1</b>	<b>1.5</b>	<b>-4.43</b>	<b>-0.86</b>	<b>0.78</b>
<b>U2</b>	<b>-0.5</b>	<b>0.78</b>	<b>-3.46</b>	<b>-1.82</b>
<b>U3</b>	<b>-8.5</b>	<b>0.78</b>	<b>-3.46</b>	<b>-0.26</b>
<b>U4</b>	<b>-0.5</b>	<b>-4.43</b>	<b>-0.86</b>	<b>-1.30</b>
<b>U5</b>	<b>-8.5</b>	<b>-1.30</b>	<b>0.86</b>	<b>0.78</b>
<b>U6</b>	<b>-0.5</b>	<b>-0.26</b>	<b>0.86</b>	<b>-1.82</b>

Table 3.6 – matrice des données centrées réduite  $X_c$ .

3. On déduit la matrice de corrélation linéaire  $C$  suivant cette formule

$$C = \frac{1}{I} \cdot X_c \cdot X_c^T$$

	<b>I1</b>	<b>I2</b>	<b>I3</b>	<b>I4</b>
<b>I1</b>	<b>0.9</b>	<b>-0.043</b>	<b>0.064</b>	<b>0.32</b>
<b>I2</b>	<b>-0.043</b>	<b>0.80</b>	<b>-0.30</b>	<b>-0.31</b>
<b>I3</b>	<b>0.064</b>	<b>-0.30</b>	<b>1.06</b>	<b>-0.76</b>
<b>I4</b>	<b>0.32</b>	<b>-0.31</b>	<b>-0.76</b>	<b>2.72</b>

Table 3.7 – matrice de corrélation linéaire  $C$ .

4. La matrice finale avec les prédictions sera

	<b>I1</b>	<b>I2</b>	<b>I3</b>	<b>I4</b>
<b>U1</b>	<b>5.0</b>	<b>4.49</b>	<b>3.0</b>	<b>5.0</b>
<b>U2</b>	<b>4.0</b>	<b>5.0</b>	<b>3.93</b>	<b>2.98</b>
<b>U3</b>	<b>4.21</b>	<b>5.0</b>	<b>3.93</b>	<b>3.0</b>
<b>U4</b>	<b>4.0</b>	<b>4.66</b>	<b>3.0</b>	<b>1.0</b>
<b>U5</b>	<b>4.45</b>	<b>3.0</b>	<b>5.0</b>	<b>5.0</b>
<b>U6</b>	<b>4.0</b>	<b>4.0</b>	<b>5.0</b>	<b>3.81</b>

Table 3.8 – matrice des prédictions de PCA.

### 3. Le rôle de la factorisation matricielle dans les SR

Nous avons d'abord analysé la matrice de (rating) de l'item utilisateur pour identifier la relation entre différents éléments, puis nous avons utilisé ces relations pour calculer directement les recommandations pour les utilisateurs en utilisant la technique basée sur l'objet. Nous avons appliqué le modèle de factorisation matricielle SVD pour réduire la dimensionnalité d'une matrice d'évaluation. À l'aide de l'ensemble de données MovieLens. Nous remplissons d'abord les valeurs manquantes en utilisant les moyennes d'évaluation des utilisateurs, puis postulez SVD. Pour cette grande base de données, la technique basée sur l'objet fournit une précision optimale avec des recommandations beaucoup plus rapides et de haute qualité.

### 4. Conclusion

Dans ce chapitre, nous avons passé en revue les techniques de factorisation matricielles utilisées dans le contexte des systèmes de recommandation. Ces techniques visent à décomposer une matrice de données en deux matrices (utilisateur-factor, item-factor) pour prédire les notes manquantes. Plusieurs techniques sont pallier notamment ACP, SVD, NMF, la complétion d'une matrice pour capturer des informations cachées et des relations complexes entre les utilisateurs et les items, ce qui peut conduire à des recommandations plus précises et personnalisées.

# CHAPITRE IV

## IMPLEMENTATION

## 1. Introduction

Une théorie sans pratique peut être qualifiée d'aveugle. Ce chapitre se rapporte à la mise en pratique des théories que nous avons étudiées précédemment. Il a pour vocation de présenter des illustrations pratiques de l'application réalisée à l'issue de notre mémoire. Nous présentons l'ensemble d'outils et d'environnements et nous terminons par quelques interfaces qui présentent notre application web.

## 2. Analyse et conception

### 2.1. Modélisation UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer les divers aspects d'un système d'information [Roques, P.2008].

Ce langage est certes issu du développement logiciel, mais pourrait être appliqué à toute science fondée sur la description d'un système [Roques, P., & Vallée, F.2000].

### 2.2. Spécification des besoins du système

Nous allons identifier dans cette section les fonctionnalités de notre système, en déterminant les acteurs et les besoins fonctionnels de l'application web.

#### 2.2.1. Identification des acteurs

- **L'utilisateur** : C'est l'acteur qui utilise le système et qui peut accéder à tous les fonctionnalités de l'application web.

### 2.2.2. Identification des besoins

- **Les besoins fonctionnels**

Le système doit permettre à l'utilisateur de :

1. S'authentifier
2. Ajouter des items
3. Avoir des recommandations personnalisées
4. Rechercher des items
5. Faire des évaluations sur les items

Le système doit permettre au visiteur du premier bien de s'inscrire.

- **Les besoins non fonctionnels**

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- a. Fiabilité** : le site doit fonctionner de façon cohérente sans erreurs et doit être satisfaisant.
- b. Erreurs** : Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre site web.
- c. Ergonomie et bonne Interface** : le site doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile).
- d. Sécurité** : Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans les sites web.

### 3. Les diagrammes UML

#### 3.1. Diagramme de cas d'utilisation

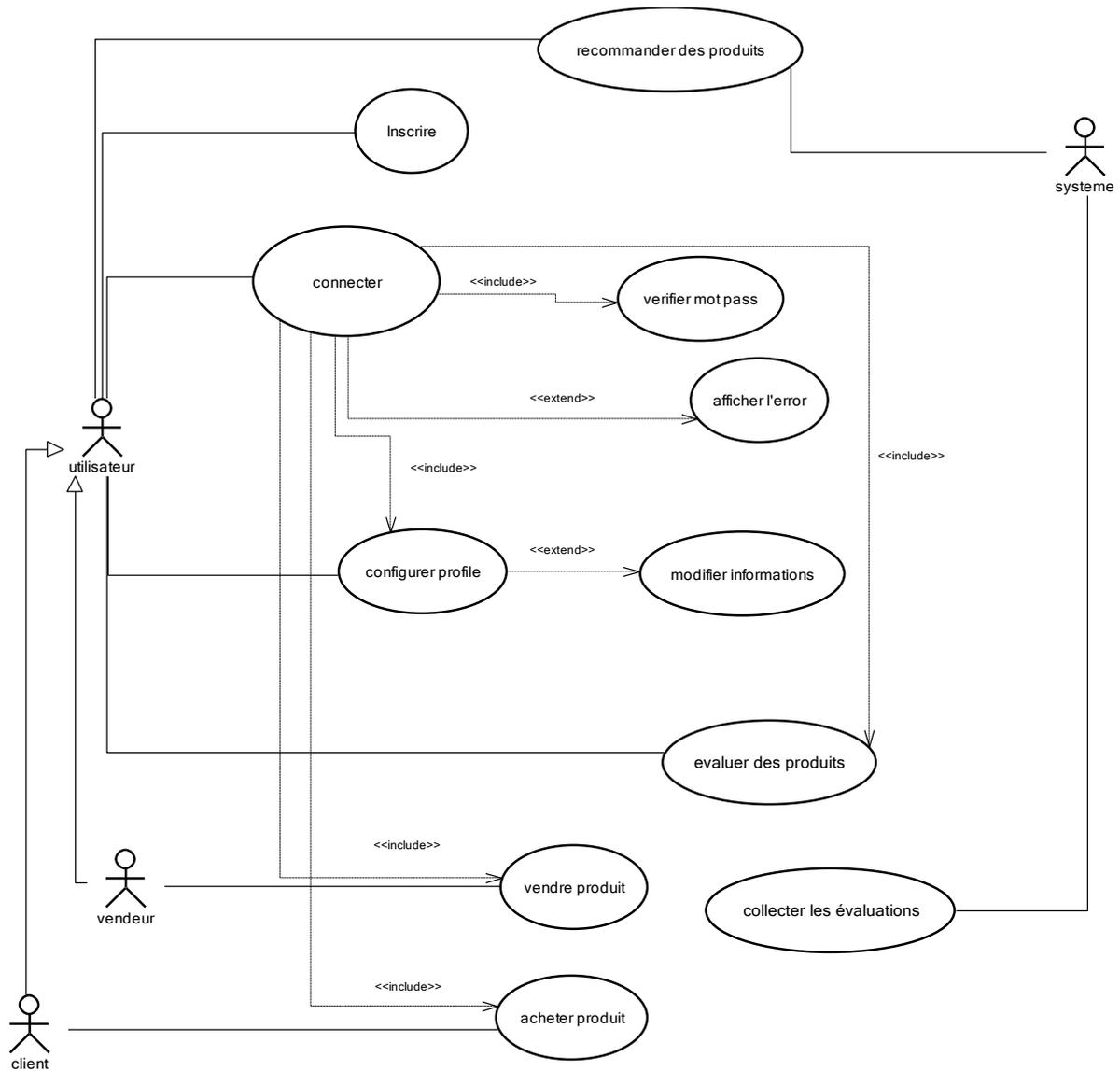


Figure 4.1 – Diagramme de cas d'utilisation.

### 3.2. Description textuelle

#### 3.2.1. Description textuelle du cas d'utilisation « Inscription »

<b>Scénario</b>	Inscription
<b>Inscription Objectif</b>	Créer un compte dans notre site
<b>Acteurs principaux</b>	Utilisateur
<b>Pré conditions</b>	Disposition d'une connexion Internet
<b>Post conditions</b>	Création d'un compte
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le visiteur demande l'interface de création de compte</li> <li>2. Le système affiche l'interface de création de compte</li> <li>3. Le visiteur remplit les champs requis</li> <li>4. Le système vérifie les champs introduit par le visiteur</li> <li>5. Le visiteur accède à la page d'accueil</li> </ol>
<b>Scénario alternatif</b>	<p>A1 : Champs obligatoires vide et/ou format incorrect</p> <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le système réaffiche l'interface de création de compte</li> </ol> <p>A2 : Email déjà existant dans le système</p> <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur « Utilisateur existe déjà »</li> <li>2. Le système réaffiche l'interface de création de compte.</li> </ol>

Table 4.1 – Description textuelle du cas d'utilisation "inscription".

## 3.2.2. Description textuelle du cas d'utilisation « Connexion »

<b>Scénario</b>	S'authentifier
<b>Objectif</b>	Authentification dans le site
<b>Acteurs principaux</b>	Utilisateur
<b>Pré conditions</b>	Avoir un compte
<b>Post conditions</b>	Accès à son compte
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède au site</li> <li>2. Le système affiche l'interface d'authentification</li> <li>3. L'utilisateur introduit son email et son mot de passe</li> <li>4. Le système vérifie les champs introduit par l'utilisateur</li> <li>5. Le système vérifie l'existence de l'utilisateur</li> </ol> <p>Si les informations introduites sont correctes, le système affiche l'accueil du site</p>
<b>Scénario alternatif</b>	<p>A1 : Champs obligatoires vide et/ou non valide</p> <ol style="list-style-type: none"> <li>1. Le système affiche un message d'erreur</li> <li>2. Le système reprend à l'étape 2 du scénario nominal</li> </ol> <p>A2 : Email et/ ou mot de passe incorrect</p> <ol style="list-style-type: none"> <li>4. Le système affiche un message d'erreur « Vérifier vos informations et réessayer »</li> <li>4. Le système réaffiche l'interface de connexion.</li> </ol>

Table 4.2 – Description textuelle du cas d'utilisation "connexion".

### 3.3. Diagramme de séquence

#### 3.3.1. Diagramme de séquence « Inscription »

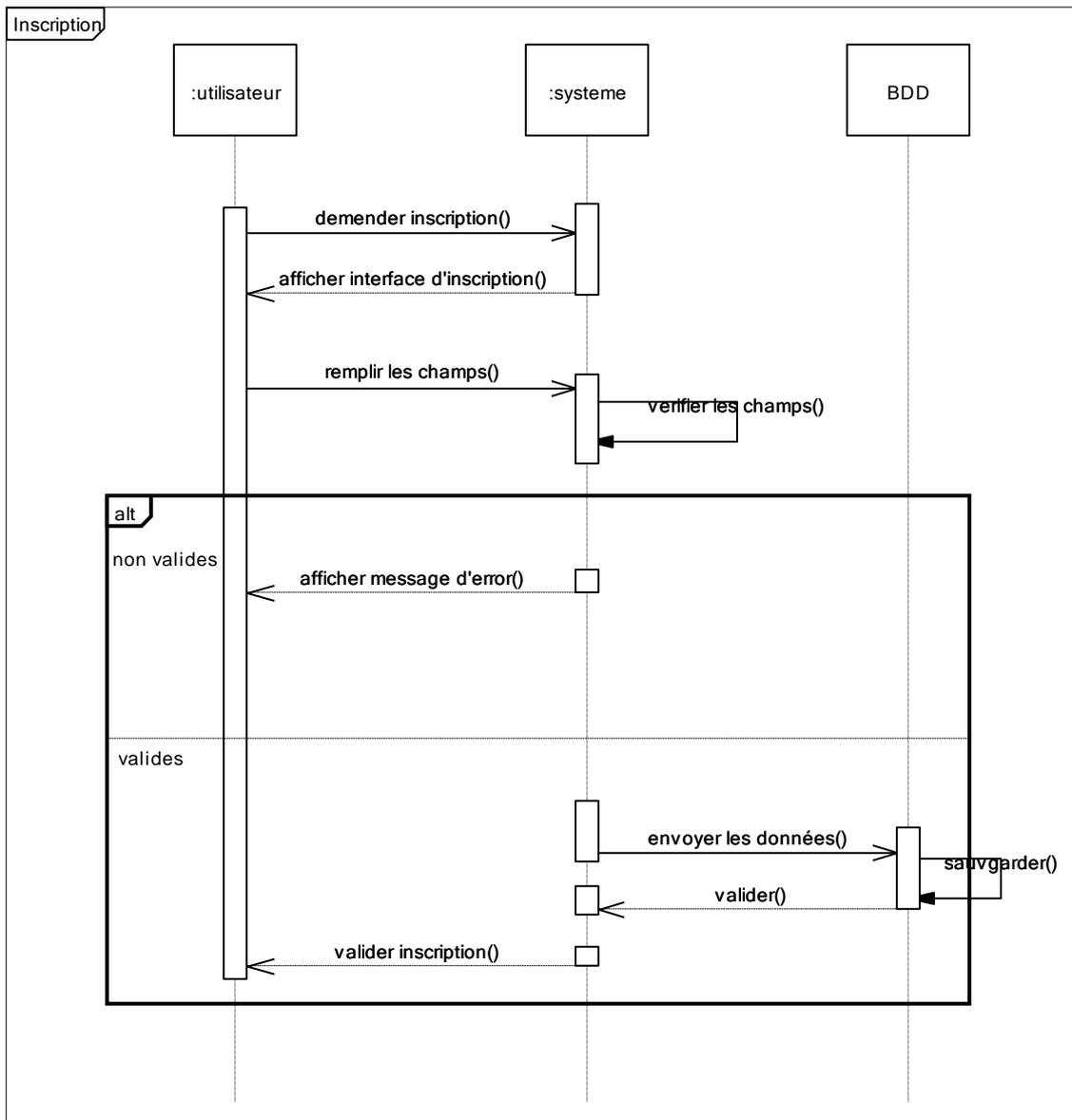


Figure 4.2 – Diagramme de séquence « Inscription ».

3.3.2. Diagramme de séquence « Connexion »

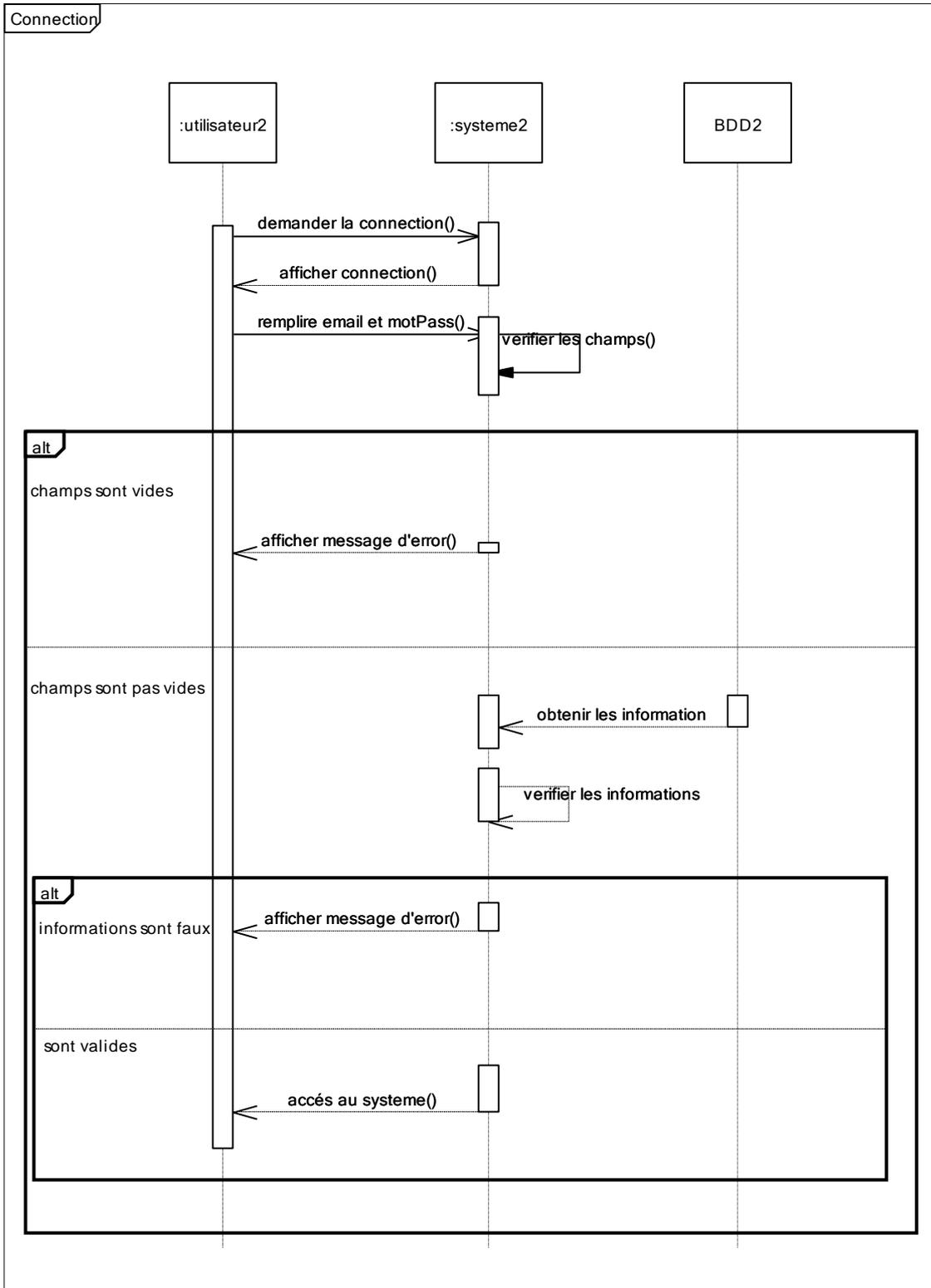


Figure 4.3 – Diagramme de séquence « Connexion ».

### 3.4. Diagramme d'activité

C'est un Diagramme associé à un objet particulier ou à un ensemble d'objets, qui illustre les flux entre les activités et les actions. Il permet de représenter graphiquement le déroulement d'un cas d'utilisation. [A.RAISSI,2013]. (La Figure 4.6) illustre le diagramme d'activité de notre application web.

#### Diagramme d'activité d'authentification

1. L'utilisateur demande l'authentification en cliquant sur le bouton connecter.
2. Le formulaire d'authentification s'affiche sur l'écran.
3. L'utilisateur entre son email et son mot de passe.
4. Le système vérifie les données d'utilisateur sur la base des données.
5. La confirmation du succès ou échec est envoyée à l'utilisateur.

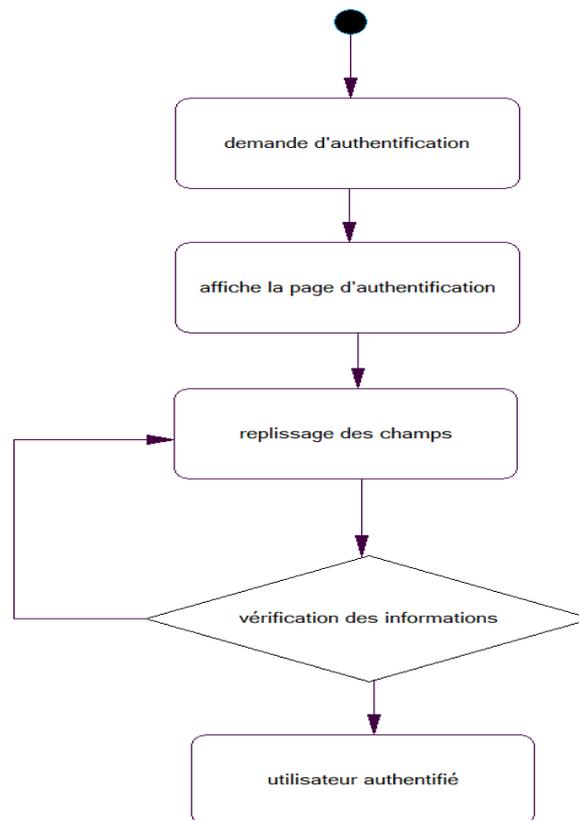


Figure 4.4 – Diagramme d'activité Authentification.

### 3.5. Diagramme de classe

La Figure 4.5 montre le diagramme de classes de notre système.

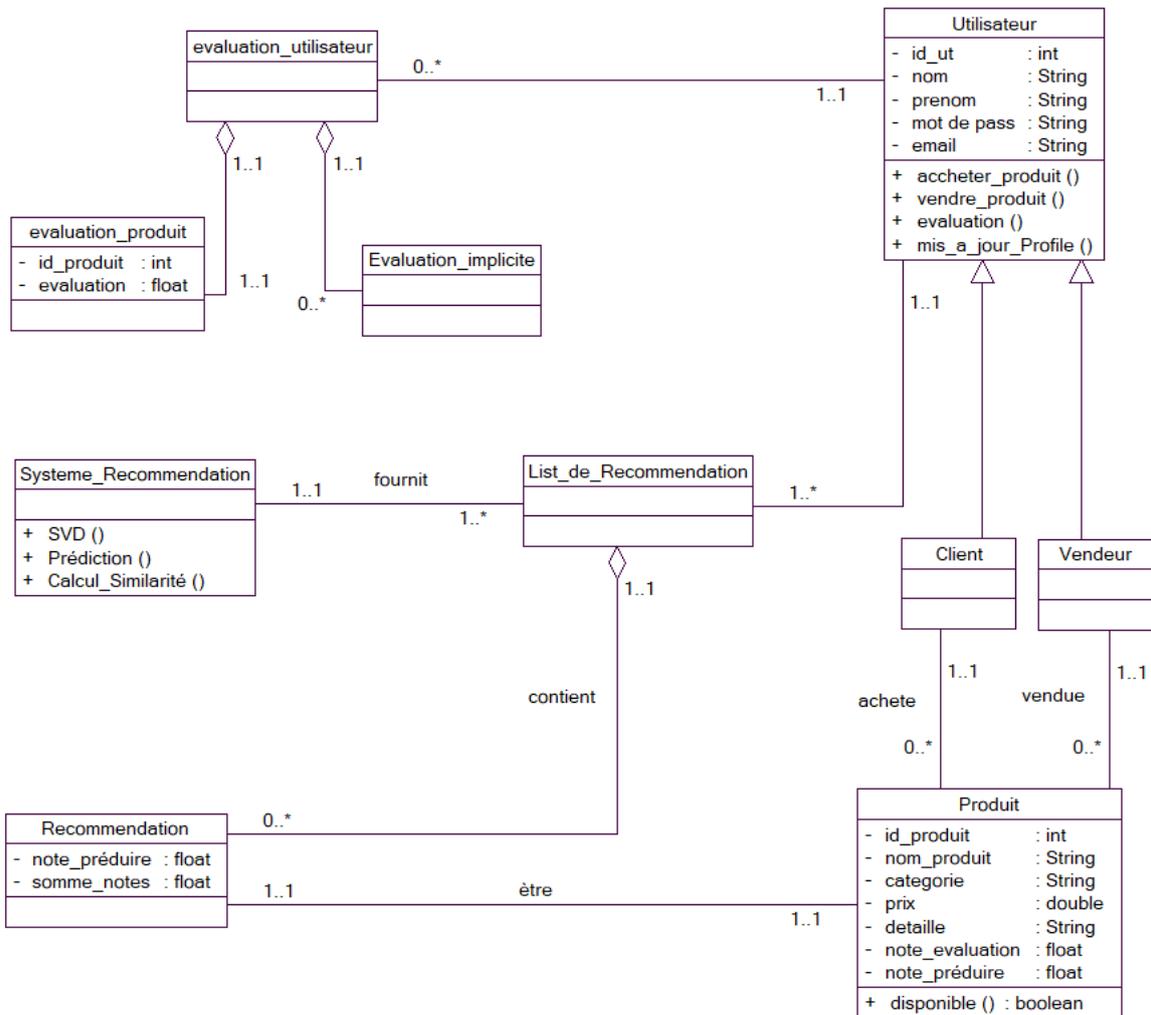


Figure 4.5 – Diagramme de classe.

## 4. Environnement de développement

### 4.1. IntelliJ IDEA

IntelliJ IDEA également appelé «IntelliJ », « IDEA » est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache2 et l'autre propriétaire. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala[[blog.jetbrains.com](http://blog.jetbrains.com)].

### 4.2. XAMPP

XAMPP est l'environnement de développement PHP le plus populaire, est une distribution Apache entièrement gratuite et facile à installer contenant MariaDB, PHP et Perl. Le package open source XAMPP a été configuré pour être incroyablement facile à installer et à utiliser[[apachefriends.org](http://apachefriends.org)].

### 4.3. Spring MVC

Le Framework Spring Web model-view-controller (MVC) est conçu autour d'un DispatcherServlet qui distribue les requêtes aux gestionnaires, avec des mappages de gestionnaires configurables, une résolution de vue, une résolution de paramètres régionaux et de thème ainsi qu'une prise en charge du téléchargement de fichiers. Le gestionnaire par défaut est basé sur les annotations `@Controller` et `@RequestMapping`, offrant un large éventail de méthodes de gestion flexibles. Avec l'introduction de Spring 3.0, le `@Controller` mécanisme vous permet également de créer des sites Web et des applications RESTful, via l'`@PathVariable` annotation et d'autres fonctionnalités [[docs.spring.io](http://docs.spring.io)].

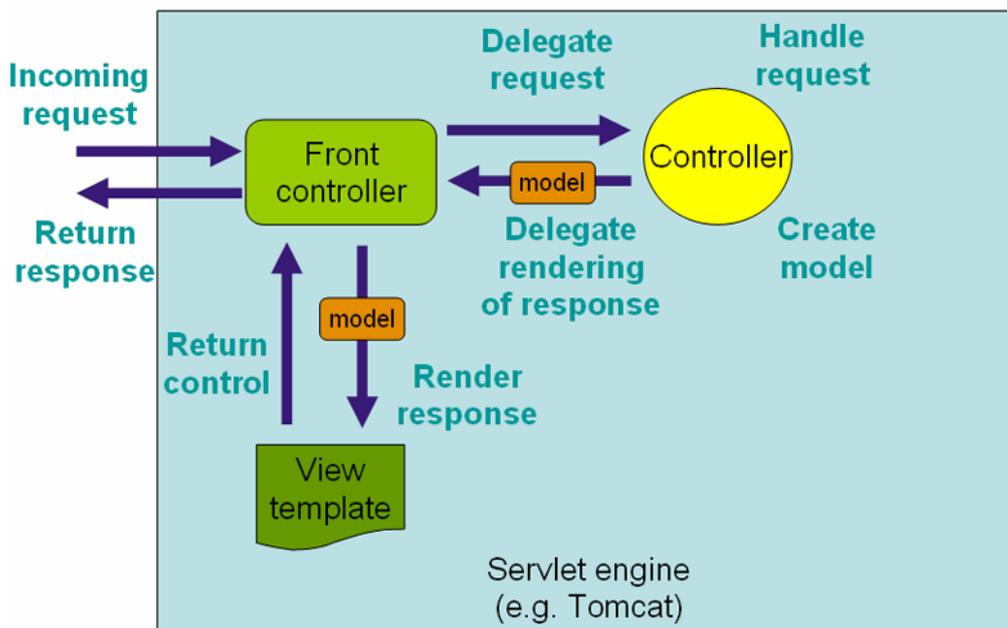


Figure 4.6 – Le workflow de traitement des demandes dans Spring Web MVC.

## 5. Langages de programmation

### 5.1. Java

Java est un langage de programmation de haut niveau orienté objet .

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers un code intermédiaire formé de byte codes qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (Java Server Pages) qui peuvent se substituer à PHP, ASP et ASP.NET[[www.oracle.com](http://www.oracle.com)].

## 5.2. XML

XML (Extensible Markup Language) est utilisé pour décrire les données. La norme XML est un moyen flexible de créer des formats d'information et de partager électroniquement des données structurées via l'Internet public, ainsi que via les réseaux d'entreprise.

XML est un langage de balisage basé sur le langage de balisage généralisé standard ( SGML ) utilisé pour définir les langages de balisage.

La fonction principale de XML est de créer des formats pour les données qui sont utilisées pour encoder des informations pour la documentation, les enregistrements de base de données, les transactions et de nombreux autres types de données[[www.techtarget.com](http://www.techtarget.com)].

## 6. Algorithme de recommandation utilisé

Nous avons utilisé la décomposition en valeur singulière, qui est l'un des algorithmes de la factorisation matricielle.

Le mot décomposition signifie qu'il exprime une matrice donnée comme un produit d'autres matrices de propriétés utiles. Dans le cas de SVD, la matrice originale,  $\mathbf{A}$  de taille  $(m, n)$  est exprimée par  $\mathbf{A}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  où  $\mathbf{U}$  et  $\mathbf{V}$  sont des matrices orthonormales de tailles  $m$  et  $n$ , respectivement, et  $\mathbf{\Sigma}$  est une matrice  $(m, n)$ , dont les entrées diagonales sont toutes non négatives et les entrées hors diagonale sont toutes nulles. Ces entrées diagonales non négatives sont appelées les valeurs singulières de la matrice  $\mathbf{A}$ , d'où le nom de décomposition en valeurs singulières [[yeunun-choo.medium.com](http://yeunun-choo.medium.com)].

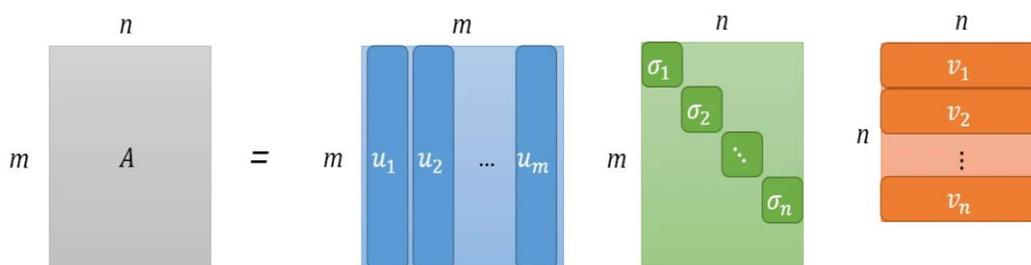


Figure 4.7 – La décomposition en valeurs singulières.

Nous construisons une matrice (utilisateurs et items) à partir des enregistrements des évaluations  $M_{u,p}$ .

	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>
<b>U1</b>	5.0	5.0	4.0	1.0	1.0
<b>U2</b>	4.0	5.0	5.0	0.0	1.0
<b>U3</b>	5.0	4.0	5.0	1.0	0.0
<b>U4</b>	4.0	5.0	4.0	1.0	1.0
<b>U5</b>	0.0	1.0	2.0	5.0	5.0
<b>U6</b>	1.0	2.0	1.0	4.0	5.0

Table 4.3 – Exemple des évaluations Utilisateurs / Item.

Nous appliquons l’algorithme de SVD à la matrice obtenue, Les résultats donnés sont deux matrices orthogonales et une matrice diagonale.

	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>	<b>X5</b>
<b>Y1</b>	0.48	-0.14	0.27	0.44	0.06
<b>Y2</b>	0.47	-0.20	0.10	-0.76	0.39
<b>Y3</b>	0.47	-0.22	-0.65	-0.36	0.22
<b>Y4</b>	0.45	-0.11	0.26	-0.06	-0.78
<b>Y5</b>	0.23	0.71	-0.47	-0.22	-0.25
<b>Y6</b>	0.25	0.61	0.46	0.23	0.37

Table 4.4 – Matrice U Orthogonal.

	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>	<b>X5</b>
<b>Y1</b>	0.52	0.58	0.55	0.21	0.23
<b>Y2</b>	-0.28	-0.14	-0.12	0.63	0.70
<b>Y3</b>	0.01	0.60	-0.63	-0.37	0.34
<b>Y4</b>	0.68	-0.18	-0.52	0.42	-0.23
<b>Y5</b>	0.43	-0.52	0.10	-0.50	0.53

Table 4.5 – Matrice  $\mathbf{V}^T$  Orthogonal.

	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>	<b>X5</b>
<b>Y1</b>	16.76	0.0	0.0	0.0	0.0
<b>Y2</b>	0.0	8.78	0.0	0.0	0.0
<b>Y3</b>	0.0	0.0	1.72	0.0	0.0
<b>Y4</b>	0.0	0.0	0.0	1.36	0.0
<b>Y5</b>	0.0	0.0	0.0	0.0	0.52

Table 4.6 – Matrice  $\mathbf{\Sigma}$  Diagonale.

On trouve que la matrice  $\mathbf{\Sigma}$  avec des valeurs triées dans l'ordre décroissant,  $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_4 \geq 0$  et ( $\sigma_1 = 16.76$ ,  $\sigma_2 = 8.78$ ) très grand sur les autres valeurs donc on prend le nombre des lignes et les colonnes égale à 2.

$$\mathbf{A}_2 = \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^T = (\mathbf{u}_1 \quad \mathbf{u}_2) \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} (\mathbf{v}_1 \quad \mathbf{v}_2)^T = \mathbf{u}_1 \sigma_1 \mathbf{v}_1^T + \mathbf{u}_2 \sigma_2 \mathbf{v}_2^T \quad (4.1)$$

On obtient :

	<b>X1</b>	<b>X2</b>
<b>Y1</b>	0.48	-0.14
<b>Y2</b>	0.47	-0.20
<b>Y3</b>	0.47	-0.22
<b>Y4</b>	0.45	-0.11
<b>Y5</b>	0.23	0.71
<b>Y6</b>	0.25	0.61

Table 4.7 – Matrice  $U_2$  factorisé.

	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>	<b>X5</b>
<b>Y1</b>	0.52	0.58	0.55	0.21	0.23
<b>Y2</b>	-0.28	-0.14	-0.12	0.63	0.70

Table 4.8 – Matrice  $V^T_2$  factorisé.

	<b>X1</b>	<b>X2</b>
<b>Y1</b>	16.76	0.0
<b>Y2</b>	0.0	8.78

Table 4.9 – Matrice  $\Sigma_2$  factorisé.

Nous appliquons l'équation (4.1) pour trouver une matrice des évaluations prévisible

	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>
<b>U1</b>	4.58	4.85	4.60	0.94	0.97
<b>U2</b>	4.60	4.81	4.55	0.60	0.59
<b>U3</b>	4.63	4.80	4.54	0.45	0.43
<b>U4</b>	4.23	4.51	4.28	1.0	1.04
<b>U5</b>	0.26	1.36	1.35	4.76	5.27
<b>U6</b>	0.7	1.68	1.65	4.25	4.70

Table 4.10 – Matrice des évaluations prévisible Utilisateurs / Item.

## 7. Expérimentations et mise en œuvre

### 7.1. Interface d'identification

Cette interface permet au visiteur déjà inscrit dans le site d'accéder à leur espace personnel.

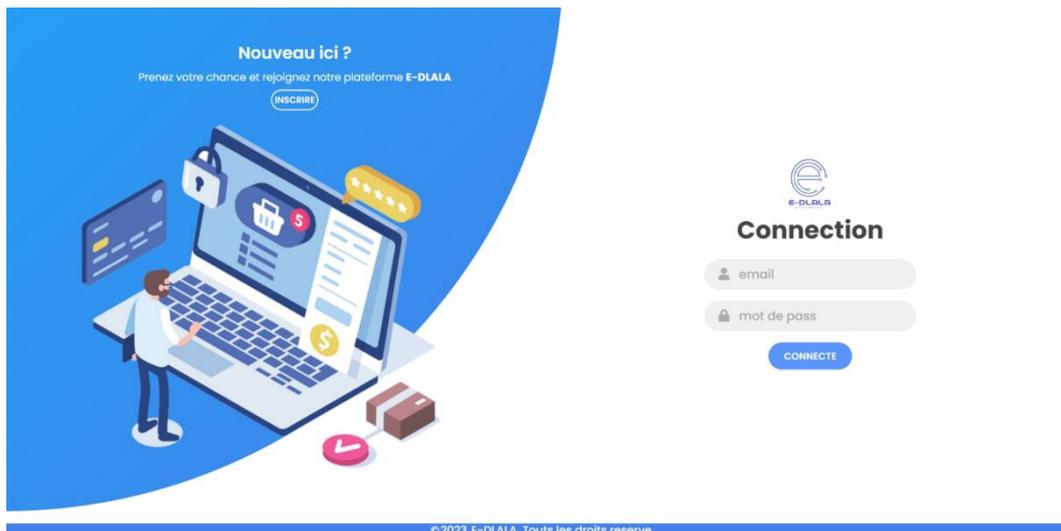


Figure 4.8 – Interface d'identification.

### 7.2. Interface d'inscription

Sur cette interface l'utilisateur doit remplir le formulaire d'inscription pour créer un compte (nom d'utilisateur, Email et mot de passe).

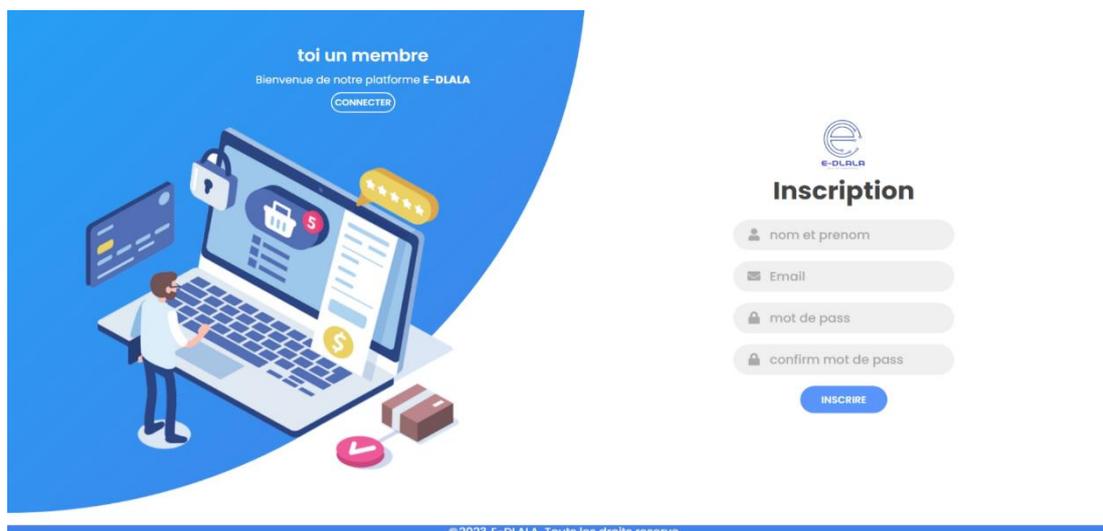


Figure 4.9 – Interface d'inscription.

### 7.3. Interface de profile

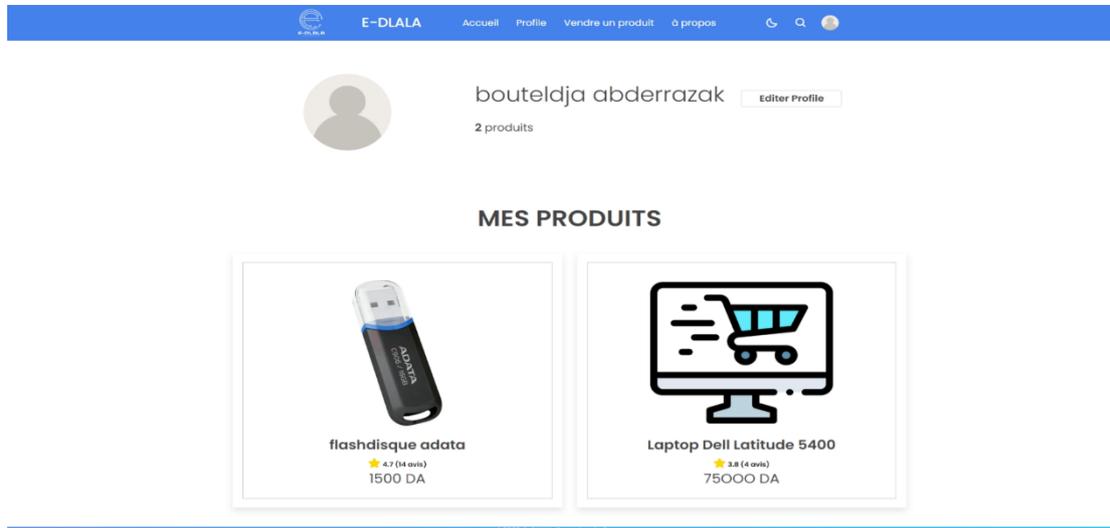


Figure 4.10 – Interface de profil.

### 7.4. Interface de recommandation

Après l'inscription, l'interface de recommandation s'affichée avec plusieurs fonctionnalités. Pour elle affiche les items recommandés et le plus similaires qui peuvent à intéresser pour l'utilisateur et qui sont disponibles.



Figure 4.11 – Interface de recommandation.

### 7.8. Interface d'évaluation de produit

Cette interface représente la page de détail de produit dans laquelle nous avons fournis les informations nécessaires. L'utilisateur peut évaluer le produit en cliquant sur les étoiles en milieu de l'image.

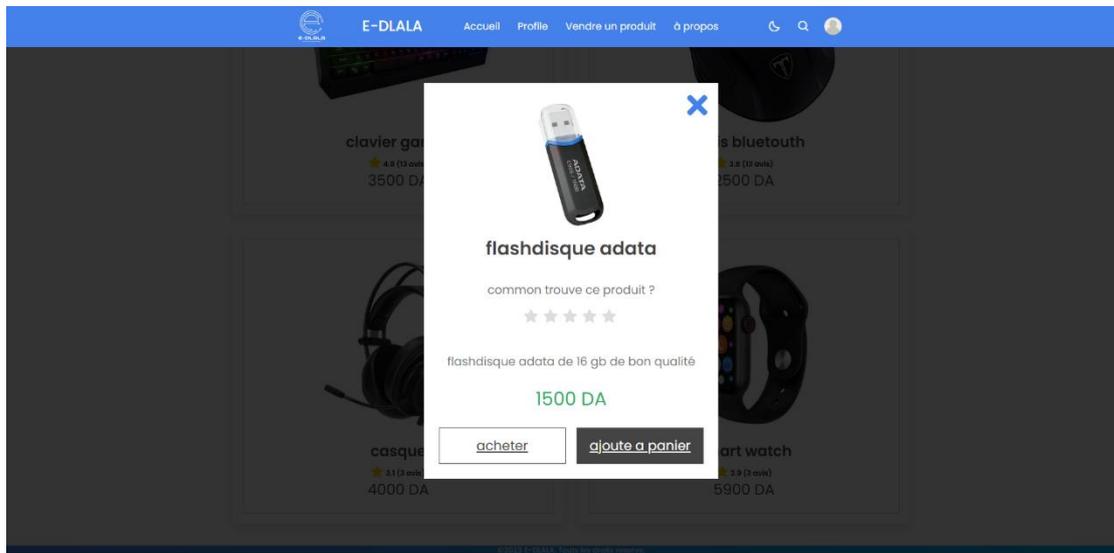


Figure 4.12 – Interface d’évaluation de produit.

### 7.9. Interface graphique de l’algorithme SVD

Sur cette interface on doit remplir la taille de notre matrice

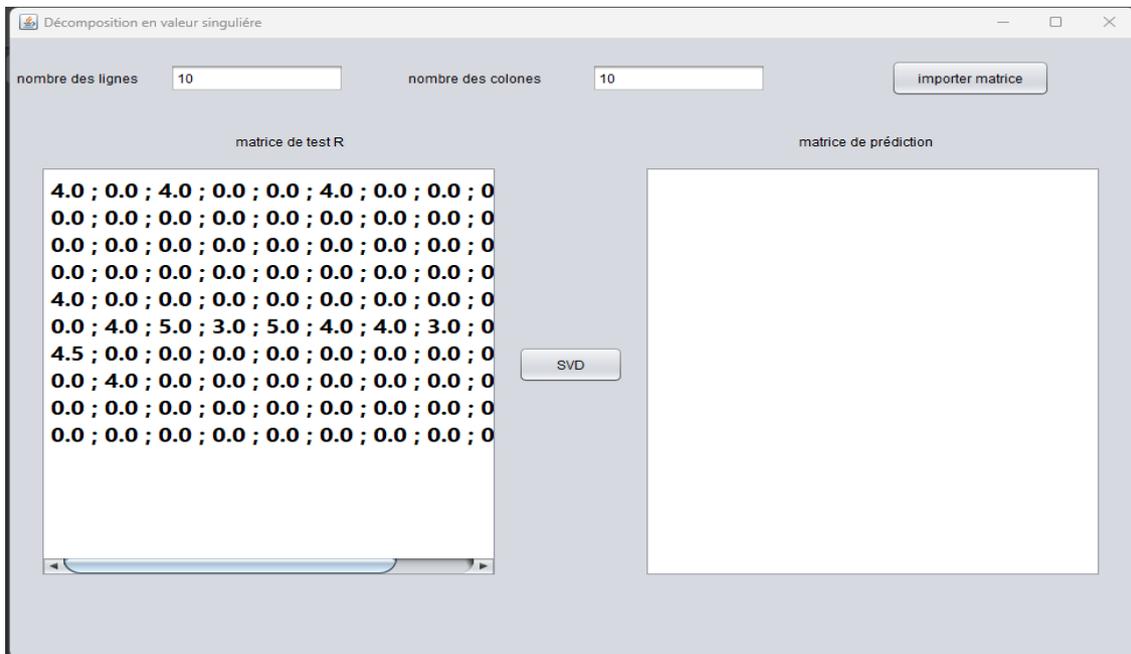


Figure 4.13 – Interface de remplissage de matrice.

Après l'identifier de la matrice on applique l'algorithme de SVD

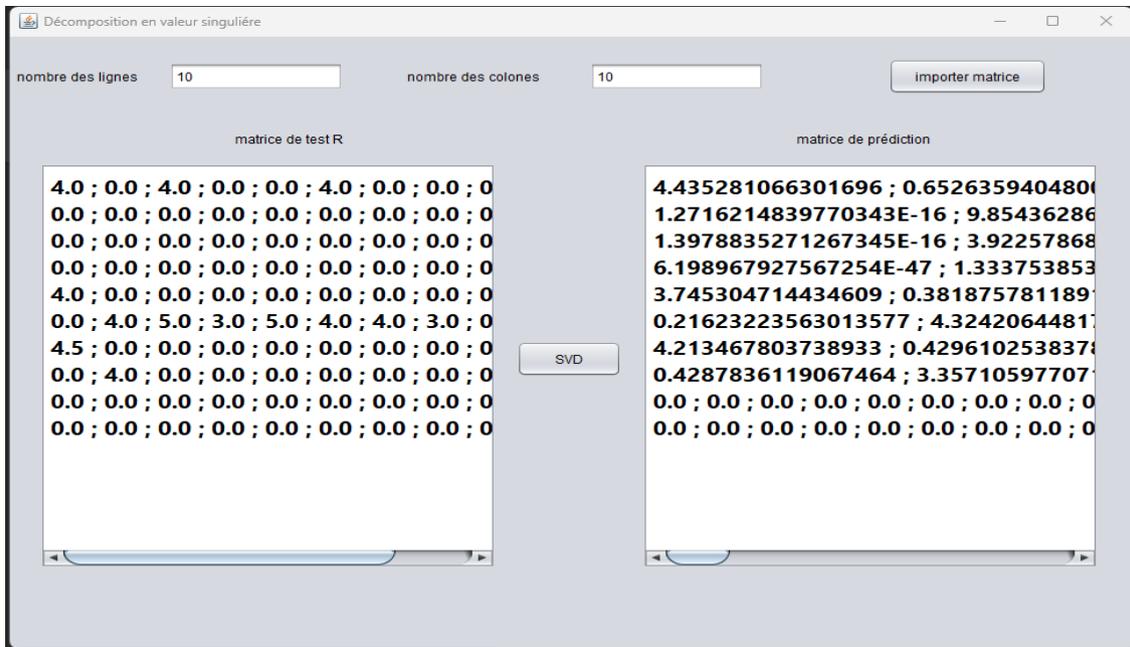


Figure 4.14 – Interface de l'applique SVD.

### 8. Comparaison des différentes techniques

On calcule la différence entre l'évaluation et la prédiction de chaque technique avec la métrique d'erreur absolue moyenne (MAE), avec des intervalles des utilisateurs, on trouve que plus le MAE se converge vers le zéro, la qualité de prédiction augmenter.

utilisateur	[1;9]	[40;49]	[60;69]	[80;89]	[100;149]	[200;249]	[300;349]	[400;450]
SVD	1.03	0.88	0.86	0.84	0.82	0.79	0.79	0.77
NMF	1.11	0.93	0.91	0.9	0.89	0.87	0.87	0.86
PCA	1.09	0.92	0.91	0.89	0.88	0.86	0.86	0.84

Table 4.11 – les valeurs de MAE de chaque technique.

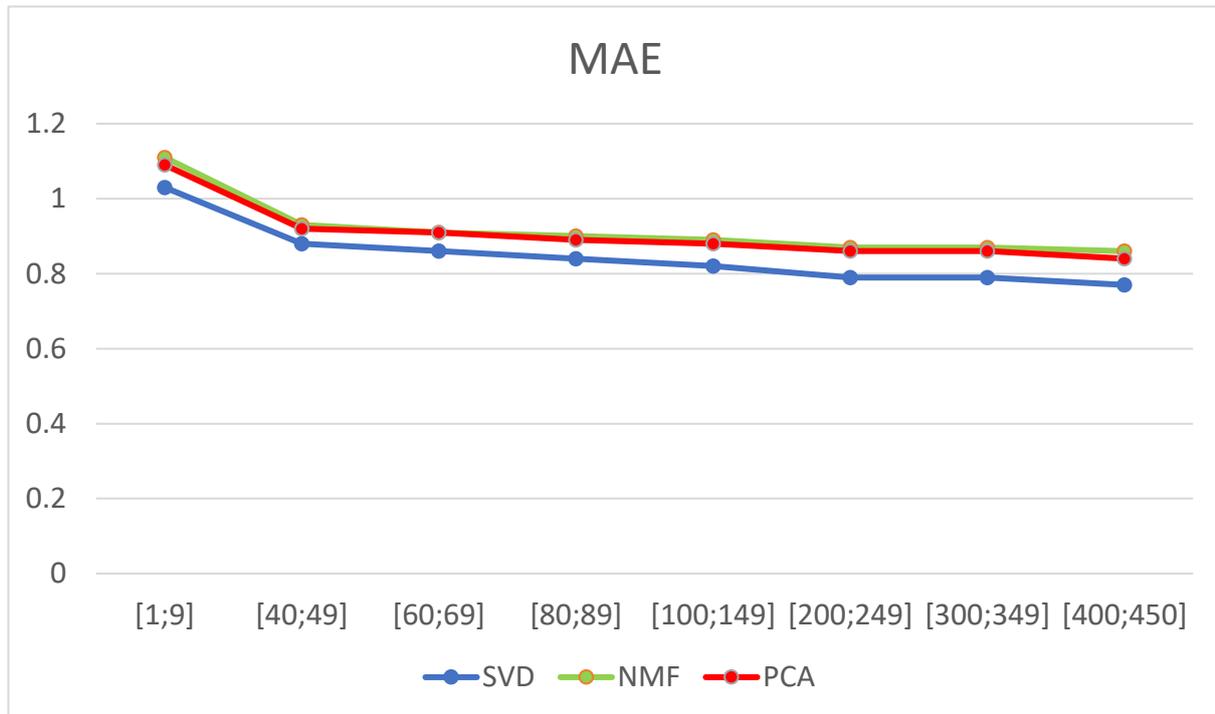


Figure 4.15 – Graphe pour les valeurs de MAE.

Enfin, après la représentation graphique, on trouve que la meilleure technique de factorisation matricielle est la décomposition de la valeur singulière (SVD), car elle fait de bons résultats des prédictions.

## 9. Conclusion

Dans ce chapitre, nous avons exposé et illustré les différentes phases pour la conception et la réalisation de notre système de recommandation, qui consiste à recommander les produits à un utilisateur donné, en utilisant le filtrage collaboratif et la factorisation matricielle « la décomposition de la valeur singulière (SVD) ». En commençant par modéliser les besoins du système à l'aide de langage UML. Ou nous avons présenté les technologies et les différents langages et logiciels utilisés dans l'implémentation de notre application. Nous avons présenté un exemple démonstratif sur SVD et les calculs qui s'effectuent. Enfin nous avons présenté les différentes interfaces de notre application web.

# **CONCLUSION GENERALE**

### Conclusion Générale

Les systèmes de recommandation sont une forme spécifique de filtrage d'information qui permet de fournir à chaque utilisateur les items les plus susceptibles de l'intéresser. Il existe plusieurs techniques de recommandation, nous nous sommes intéressés dans ce mémoire à la technique de filtrage collaboratif.

La problématique de manque des données est persistante dans le système de filtrage collaboratif abaissant la performance des recommandations. Néanmoins, les jeux de données enregistrent des taux très élevés de manque de données (Sparsity) comme Netflix (99%), MovieLens (95%) de même pour Flixster (97%) et Douban (95%) ce qui permet de brouter le processus de similarité et, par conséquent, réduire la qualité de prédiction.

Par la suite, nous sommes focalisés sur l'implication de la technique de factorisation matricielle pour booster les SR ce qui améliore la performance et la précision des prédictions.

Plus particulièrement, nous avons mis en œuvre l'approche basée items tout en exploitant une variété de techniques de factorisation telles que NMF, SVD et ACP,

Les techniques de factorisation non seulement améliorent la performance mais aussi réduisent la complexité combinatoire par la réduction des dimensions des matrices initiales.

Ainsi, on aboutit à alléger le problème de (sparsity) dans les SR qui est considéré comme un problème majeur persistant et omniprésent dans le processus de recommandation. Il existe d'autres techniques de factorisation qui restent à titre de perspectives telles que les facteurs latents, Choleski, etc.

# REFERENCES

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of Recommender Systems: A survey of the state of the art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.
- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of Recommender Systems: A survey of the state of the art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.
- Aggarwal, C. C. (2016a). Content-based recommender systems. In *Recommender Systems*, pages 139–166. Springer.
- Aggarwal, C. C. (2016e). An introduction to recommender systems. In *Recommender systems*, pages 1–28. Springer.
- Aggarwal, C. C. (2016f). Model-based collaborative filtering. In *Recommender systems*, pages 71–138. Springer.
- Ain Picot-Clemente. Une architecture générique de Systèmes de recommandation de combinaison d’items. Application au domaine du tourisme. Recherche d’information [cs.IR]. Université de Bourgogne, 2011.
- B. M. Marlin et R. S. Zemel. “Collaborative Prediction and Ranking with Non-Random Missing Data”. In: *Proc. ACM Conference on Recommender Systems (RecSys)*. 2009, p. 5-12.
- B. Marlin, R. S. Zemel, S. Roweis et M. Slaney. “Collaborative Filtering and the Missing at Random Assumption”. In: *Proc. Conference on Uncertainty*
- B. Sarwar, G. Karypis, J. Konstan et J. Riedl. Application of Dimensionality Reduction in Recommender System – a Case Study. 2000
- Belloui, A. (2008). L’usage des concepts du web sémantique dans le filtrage d’information collaboratif. Thèse de Doctorat, Ecole supérieur d’informatique, Alger.
- Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD Cup and Workshop*, volume 2007, page 35. New York.
- Benouaret, I. (2017). Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels. Theses, Université de Technologie de Compiègne.

- Billsus, D., Pazzani, M. J., and Chen, J. (2000). A learning agent for wireless news access. In Proceedings of the 5th international conference on Intelligent user interfaces, IUI '00, pages 33–36.
- Bokde D., Girase S., Mukhopadhyay D., Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey. IJAFRC, 2014.
- Burke, R. (2002). Hybrid Recommender Systems: survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331–370.
- C. Févotte et J. Idier. “Algorithms for Nonnegative Matrix Factorization with the Beta-Divergence”. In : Neural computation 23.9 (2011), p. 2421.
- Covington, P., Adams, J., and Sargin, E. (2016). Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems, RecSys '16, pages 191–198.
- Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the fourth ACM conference on Recommender systems, RecSys '10, pages 39–46.
- Crnic, J. (2011). Introduction to modern information retrieval. Library Management, 2(4).
- D. D. Lee et H. S. Seung. “Algorithms for Non-Negative Matrix Factorization”. In : Advances in Neural Information Processing Systems (NIPS). 2001, p. 556-562.
- D. D. Lee et H. S. Seung. “Learning the Parts of Objects by Non-Negative Matrix Factorization”. In: Nature 401.6755 (1999), p. 788-791.
- D. D. Lee et H. S. Seung. “Learning the Parts of Objects by Non-Negative Matrix Factorization”. In: Nature 401.6755 (1999), p. 788-791.
- De Gemmis, M., Lops, P., Musto, C., Narducci, F., and Semeraro, G. (2015). Semantics-aware content-based recommender systems. In Recommender systems handbook, pages 119–159. Springer.
- E. J. Candès et B. Recht. “Exact Matrix Completion via Convex Optimization”. In: Foundations of Computational mathematics 9.6 (2009), p. 717
- E. J. Candès et Y. Plan. “Matrix Completion with Noise”. In : Proceedings of the IEEE 98.6 (2010), p. 925-936.
- Gunawardana, A. and Shani, G. (2015). Evaluating recommender systems. In Recommender systems handbook, pages 265–308. Springer.

- Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1) :5–53
- Herlocker, J., Konstan, J. A., and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4) :287–310.
- Ilham Esslimani. Vers une approche comportementale de recommandation : apport de l'analyse des usages dans un processus de personnalisation. *Interface homme-machine [cs.HC]*. Université Nancy II, 2010.
- in *Artificial Intelligence (UAI)*. 2007.
- Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B., A. (2015). Recommendation Systems : Principles methods and evaluation. *Egyptian Informatics Journal*, 16, 261.
- J. Bennett, S. Lanning et al. “The Netflix Prize”. In: *Proc. KDD Cup and Workshop. T. 2007*. 2007, p. 35.
- J. M. Hernandez-Lobato, N. Houlsby et Z. Ghahramani. “Probabilistic Matrix Factorization with Non-Random Missing Data”. In: *Proc. International Conference on Machine Learning (ICML)*. 2014, p. 1512-1520.
- Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3) :77–87.
- Lampropoulos, A. S., Lampropoulou, P. S., and Tsihrintzis, G. A. (2012). A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis. *Multimedia Tools and Applications*, 59(1) :241–258.
- Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76–80.
- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3) :225.
- Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, page 73. Springer.
-

- M. C. Tweedie. “An Index Which Distinguishes between Some Important Exponential Families”. In: *Statistics: Applications and New Directions: Proc. Indian Statistical Institute Golden Jubilee International Conference*. 1984, p. 579-604.
- Matrix Factorization”. In : *Nature* 401.6755 (1999), p. 788-791.
- Miranda, T., Claypool, M., Gokhale, A., Mir, T., Murnikov, P., Netes, D., and Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*.
- Papagelis, M., Plexousakis, D., & Kutsuras, T. (2005). Alleviating the sparsity problem of collaborative filtering using trust inferences. *International Conference on Trust Management*, 224–239.
- R. J. A. Little et D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2014.
- R. M. Bell et Y. Koren. “Lessons from the Netflix Prize Challenge.” In: *SiGKDD Explorations* 9.2 (2007), p. 75-79.
- RAISSI Conception et développement d’un site web de e-commerce pour le compte de LSAT\_Nokia. Mémoire master, université virtuelle de Tunis, 2013.
- Rao, N., & Talwar, V. (2008). Application domain and functional classification of recommender systems: a survey. *Desidoc Journal of Library and Information Technology*, 28(3), 17–36.
- Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., & Riedl, J. (2002).
- Resnick, P., & Varian, H. R. (1997). *Recommender Systems*. *Communication of the ACM*, 40, 56–58.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens : an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work, CSCW '94*, pages 175–186.
- Ricci, F., Rokach, L., and Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. Springer.
- Roques, P. *UML 2: Modéliser une application web*. Editions Eyrolles. (2008).

- Roques, P., & Vallée, F. UML en action : de l'analyse des besoins à la conception en Java. Eyrolles. (2000).
- S.Taouli, W.Benachenhou. Utilisation de factorisation matricielle sans les systèmes de recommandation sensible au contexte. Mémoiremaster, université Abou Bakr Belkaid –Tlemcen, 2017
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, WWW '01, pages 285–295.
- Smyth, B. and Cotter, P. (2000). A personalised tv listings service for the digital tv age. *Knowledge-Based Systems*, 13(2-3) :53–59.
- Spencer, N. H. (2013). *Essentials of multivariate data analysis*. CRC Press.
- Steck, H. (2013). Evaluation of recommendations: Rating-prediction and ranking. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 213–220.
- Wold, S., Esbensen, K. et Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3), 37-52.
- <https://blog.jetbrains.com/idea/2023/03/intellij-idea-2023-1>
- <https://www.apachefriends.org/fr/index.html>
- <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html>
- <https://www.oracle.com/java/>
- <https://www.techtarget.com/whatis/definition/XML-Extensible-Markup-Language>
- <https://yeunun-choo.medium.com/singular-value-decomposition-in-a-movie-recommender-system-e3565ed42066>