

جامعة محمد الأول بوجدة
UNIVERSITÉ MOHAMMED PREMIER OUJDA
+٢٠٦٠٤٤٣٨٢٢٩٦٠٥٧

ROYAUME DU MAROC
UNIVERSITÉ MOHAMMED PREMIER
École Nationale des Sciences Appliquées
Oujda - Maroc

المدرسة الوطنية للعلوم التطبيقية
École Nationale des Sciences Appliquées d'Oujda
+٢٠٦٠٤٤٣٨٢٢٩٦٠٥٧

Mémoire de Fin d'Année

Filière : Ingénierie du Data Science et Cloud Computing

Système de recommandation amélioré

Réalisé par :

Mouad AIT HA
Mohammed LAKBAKBI

Membres de jury :

M. Mohammed AISSAOUI
Mme. Chaymae CHOUIEKH

Encadré par :

Mme. Chaymae CHOUIEKH

Dédicaces

À nos chers parents, sœurs et frères,

Votre amour et soutien ont été des forces inébranlables qui nous ont guidés à travers les défis.

Vous avez toujours cru en nous, et c'est grâce à vous que nous sommes ici aujourd'hui.

À nos précieux enseignants,

Votre passion pour l'enseignement et votre dévouement nous ont profondément inspirés.

Vous nous avez armés de connaissances et de compétences qui nous ont conduits vers la réussite.

Merci pour votre patience et votre engagement.

À nos amis,

Vous avez été des piliers de soutien inestimables.

Vos encouragements et votre présence ont rendu cette aventure plus joyeuse et enrichissante.

Nous sommes profondément reconnaissants de vous avoir à nos côtés.

À chacun d'entre vous,

Merci du fond du cœur pour votre soutien constant et votre confiance en nous.

Votre présence a été cruciale dans la réalisation de notre projet.

Que cette réussite soit le symbole de notre persévérance et de notre succès collectif.

Merci infiniment.

Mouad - Mohammed

Remerciements

Avant toute chose, nous tenons à exprimer notre profonde gratitude envers le Très-Haut pour nous avoir accordé la santé, la force et la persévérance nécessaires à la concrétisation de ce projet d'envergure.

Nous adressons nos sincères remerciements à Madame Chaymae CHOUIEKH, encadrante de ce projet, pour son accompagnement empreint de rigueur, de bienveillance et de disponibilité. Ses conseils avisés et son expertise éclairée ont été un guide précieux tout au long de cette entreprise et ont grandement contribué à son succès.

Nos remerciements les plus vifs vont également à l'ensemble du corps enseignant pour leur dévouement, leur générosité et leur patience sans faille malgré leurs lourdes charges. Leurs enseignements de grande valeur et leur soutien indéfectible ont été les ferment de notre développement intellectuel et de notre progression académique.

C'est grâce au concours inestimable et au soutien indéfectible de toutes ces personnes dévouées que nous avons pu mener à bien ce projet. Nous leur exprimons notre gratitude la plus sincère et nous leur dédions ce travail avec une immense fierté.

Résumé

Ce rapport présente une étude approfondie des systèmes de recommandation améliorés, visant à améliorer l'expérience utilisateur dans le domaine des recommandations de livres. Nous explorons des techniques classiques telles que le filtrage collaboratif et le filtrage basé sur le contenu, ainsi que des approches d'apprentissage automatique sophistiquées, incluant la factorisation matricielle et les autoencodeurs.

Une évaluation rigoureuse des performances de ces modèles a été réalisée, révélant la supériorité des autoencodeurs pour capturer les préférences complexes des utilisateurs et générer des recommandations précises. Le rapport aborde également les défis inhérents aux systèmes de recommandation, notamment les biais dans les données, l'évolutivité, l'explicabilité et la prise en compte du contexte.

Malgré ces limitations, les résultats obtenus constituent une base solide pour des améliorations futures, visant à offrir une expérience utilisateur enrichie et personnalisée dans le domaine des recommandations littéraires. Ces travaux démontrent le potentiel des systèmes de recommandation avancés pour transformer la découverte de livres et proposer des suggestions personnalisées, répondant ainsi aux attentes et aux besoins des lecteurs.

Mots clés : Système de recommandation, recommandation des livres, K-NN, SVD, NMF, Autoencoders, Filtrage collaboratif, Filtrage basé sur le contenu.

Abstract

This report presents an in-depth study of advanced recommendation systems aimed at enhancing the user experience in book recommendations. We explore classical techniques such as collaborative filtering and content-based filtering, as well as sophisticated machine learning approaches including matrix factorization and autoencoders.

A rigorous evaluation of these models performance was conducted, highlighting the superiority of autoencoders in capturing users complex preferences and generating accurate recommendations. The report also addresses the inherent challenges of recommendation systems, such as data biases, scalability, explainability, and contextual considerations.

Despite these limitations, the results provide a solid foundation for future improvements aimed at offering an enriched and personalized user experience in literary recommendations. This work demonstrates the potential of advanced recommendation systems to transform book discovery and provide personalized suggestions, meeting the readers' expectations and needs.

Keywords : Recommendation system, book recommendation, K-NN, SVD, NMF, Autoencoders, Collaborative filtering, Content-based filtering.

Table des matières

Résumé	4
Abstract	5
Liste des Abréviations	8
Liste des figures	9
Liste des tableaux	10
Introduction générale	11
1 Etat d'art	12
1.1 Introduction	12
1.1.1 Domaines d'utilisation	13
1.1.2 Problèmes et limites des systèmes de recommandation	13
1.1.3 Techniques Avancées et Nouvelles Approches	14
1.1.4 Évaluation des Systèmes de Recommandation	15
1.2 Conclusion	15
2 Étude préliminaire	16
2.1 Introduction	16
2.2 Algorithmes de Recommandation classiques	17
2.2.1 Filtrage Collaboratif	17
2.2.2 Filtrage Basé sur le Contenu	18
2.3 Algorithmes de Recommandation Avancés	18
2.3.1 K-Nearest Neighbors (k-NN)	18
2.3.2 Décomposition en valeurs singulières (SVD)	18
2.3.3 Factorisation matricielle non négative (NMF)	19
2.3.4 Les réseaux de neurones (Autoencoders)	19
2.4 Sources de données	20
2.4.1 Description de jeux de données : Books.csv	20
2.4.2 Description de jeux de données : Users.csv	21
2.4.3 Description de jeux de données : Ratings.csv	21
2.4.4 Prétraitement des données	22
2.5 Outils utilisés	22
2.5.1 Python et Jupyter Notebook	22
2.5.2 Visualisation des données	22

2.5.3	Bibliothèques de Machine Learning	23
2.5.4	Environnement de Développement et d'Exécution	23
2.6	Conclusion	23
3	Réalisation du projet	24
3.1	Introduction	24
3.2	Collecte et prétraitement des données	24
3.2.1	Collecte des Données	24
3.2.2	Nettoyage et Prétraitement des Données	25
3.2.3	Structuration des Données	25
3.3	Exploration et Analyse des Données	25
3.3.1	Statistiques Descriptives	25
3.3.2	Visualisation des Données	26
3.4	Implémentation des Algorithmes de Recommandation	26
3.4.1	Filtrage Collaboratif	26
3.4.2	Filtrage Basé sur le Contenu	27
3.4.3	Algorithmes de Voisinage (K-Nearest Neighbors)	27
3.4.4	Factorisation Matricielle	28
3.4.5	Autoencodeurs	29
3.5	Évaluation des Modèles	31
3.5.1	Métriques d'évaluation	32
3.5.2	Validation croisée	32
3.5.3	Comparaison des Performances	32
3.5.4	Évaluation des Autoencodeurs	34
3.5.5	Évaluation qualitative	35
3.6	Déploiement du Système de Recommandation	35
3.6.1	Architecture du Système	36
3.6.2	Maintenance et Évolutivité	36
3.7	Cas d'utilisation et Scénarios d'Application	36
3.7.1	Découverte de Nouveaux Livres	36
3.7.2	Suggestions pour les Clubs de Lecture	37
3.7.3	Recommandations Personnalisées pour les Libraires	37
3.7.4	Analyse des Tendances de Lecture	37
3.8	Défis et Limites	37
3.9	Conclusion	38
Conclusion et perspectives		39
Bibliographie		40

Liste des Abréviations

NDCG Normalized Discounted Cumulative Gain

NMF Non-negative matrix factorization

SVD Singular values decomposition

NLP Natural Language Processing

SdR Système de Recommendation

RMSE Root Mean Squared Error

MAP Mean Average Precision

K-NN K-nearest neighbors

MRR Mean Reciprocal Rank

MAE Mean Absolute Error

Liste des figures

1.1	Recommendation des livres sur Amazon.	13
2.1	Filtrage collaboratif	17
2.2	User-item interaction	19
2.3	Le schema d'un Autoencoder	20
3.1	Graphes - Visualisation des données	26
3.2	KNN user-item	28
3.3	Formule de Cosine similarity	28
3.4	Matrice de factorisation	28
3.5	Les composants de l'AE	30
3.6	exemple d'une architecture d'AE	30
3.7	Application web	35

Liste des tableaux

2.1	Échantillon du dataset Books	21
2.2	Échantillon du dataset Users	21
2.3	Échantillon du dataset Ratings	22
3.1	Comparaison des performances des modèles SVD et NMF	32
3.2	Validation croisée pour le modèle SVD	32
3.3	Validation croisée pour le modèle NMF	33
3.4	Performances des autoencodeurs sur les métriques RMSE et MAE	34
3.5	Comparaison des performances des autoencodeurs et du modèle SVD	34

Introduction Générale

Dans l'ère numérique actuelle, la quantité astronomique d'informations et de contenus disponibles en ligne représente à la fois une opportunité et un défi. Bien que cette abondance offre un vaste choix aux utilisateurs, elle peut également les submerger et rendre la découverte de contenus pertinents et adaptés à leurs préférences individuelles particulièrement ardue. C'est dans ce contexte que les systèmes de recommandation sont devenus des outils indispensables, guidant les utilisateurs vers du contenu susceptible de les intéresser et de répondre à leurs attentes ([Bellabes, 2023](#)).

Les systèmes de recommandation traditionnels analysent les données des utilisateurs, leurs interactions avec un système (site web, application, etc.) et leurs préférences pour générer des suggestions personnalisées. Cependant, ces systèmes présentent certaines limites, notamment en termes de précision, de diversité des recommandations et d'adaptation aux préférences dynamiques des utilisateurs ([Ricci and Melville, 2015](#)). Face à ces défis, le concept de "système de recommandation amélioré" a émergé, promettant de révolutionner l'expérience utilisateur grâce à des recommandations plus précises, diversifiées et adaptées.

Les systèmes de recommandation améliorés utilisent des techniques d'apprentissage automatique avancées, comme les réseaux de neurones profonds, la factorisation matricielle, et l'apprentissage par renforcement, pour offrir des recommandations plus pertinentes en intégrant des facteurs contextuels complexes tels que l'humeur et la situation géographique. Leur principal avantage est leur capacité à s'adapter aux préférences évolutives des utilisateurs grâce à des algorithmes d'apprentissage continu, tout en encourageant la découverte de nouveaux contenus. En proposant des recommandations adaptées, ces systèmes offrent un avantage concurrentiel aux entreprises en augmentant la fidélisation et la part de marché.

Ce rapport de projet représente une exploration approfondie et exhaustive des systèmes de recommandation améliorés. À travers une analyse minutieuse, il traite les concepts clés, les principes fondamentaux, les avantages et les défis complexes de ces systèmes. S'appuyant sur les techniques d'apprentissage automatique exploitées, telles que les réseaux de neurones profonds, ainsi que sur l'intégration stratégique des algorithmes avancés pour garantir des recommandations véritablement personnalisées et pertinentes. De plus, il présente une dissection des méthodes d'évaluation, permettant de mesurer avec précision les performances et l'efficacité de ces systèmes de recommandation. Allant au-delà d'une simple analyse théorique, ce rapport adopte une approche pragmatique en examinant les applications pratiques des systèmes de recommandation améliorés dans divers secteurs clés, tels que le commerce électronique, les médias numériques et les services de streaming.

Chapitre 1

Etat d'art

1.1 Introduction

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

Pour cela, classiquement, deux phases consécutives sont nécessaires : la phase dite de « prédition » et la phase de « recommandation » à proprement parler. Dans la première phase, le système est doté de la capacité de découvrir les intérêts des utilisateurs, c'est-à-dire de prédire leurs possibles goûts concernant les items que ces utilisateurs n'ont pas encore évalués.

La plupart du temps, la prédition du goût d'un utilisateur pour un item est représentée par une note qui est l'estimation faite par le système de recommandation de l'évaluation que ferait l'utilisateur de cet item-là. Au cours de la deuxième phase, le SdR ordonne d'abord les items selon les notes prédictes pour recommander par la suite une liste des n items ayant les prédictions les plus élevées (top-n).

Néanmoins, il existe également des systèmes de recommandation atypiques : Pour certains, il n'y a pas vraiment de distinction claire entre la phase de prédition et celle de recommandation. Plus précisément, ce genre de SdR génère une liste de recommandations sans avoir besoin, au préalable, de prédire des notes. Un exemple représentatif est l'approche d'apprentissage par classement (learning to rank en anglais) ([Liu et al., 2015](#)).

Contrairement aux cas classiques, où seuls les items non notés sont légitimes pour la recommandation, d'autres SdR considèrent que même les items pour lesquels les goûts des utilisateurs sont connus peuvent être recommandés à nouveau, dans des contextes différents. Tenant en compte ces contextes (e.g., le temps), ces SdR 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).

Les systèmes de recommandation peuvent être catégorisés en fonction du type de données utilisées durant la phase de prédition : interactions utilisateur-item (matrice des ratings) et/ou contenus des items.

Les SdR utilisant le premier type de données sont catégorisés comme des approches de filtrage collaboratif et ceux qui sont basés sur le deuxième type de données sont catégorisés comme des approches basées sur les contenus. Il existe également une troisième catégorie d'approche, dite hybride, qui combine les deux types de données pour prédire les goûts des utilisateurs.

teurs. Notons que les SdR basés sur les contenus utilisent également les ratings dans la plupart des cas, mais en ne se focalisant que sur les ratings d'un seul utilisateur (celui pour qui on fait des recommandations) plutôt que sur ceux de tous les utilisateurs. Dans ce qui suit, on rappelle les principes de chaque catégorie de systèmes de recommandation.

1.1.1 Domaines d'utilisation

Les systèmes de recommandation sont utilisés dans une variété de domaines (Figure 1.1) pour améliorer l'expérience utilisateur et augmenter l'engagement. Dans le commerce électronique, ils suggèrent des produits pertinents, augmentant ainsi les ventes et la satisfaction client.

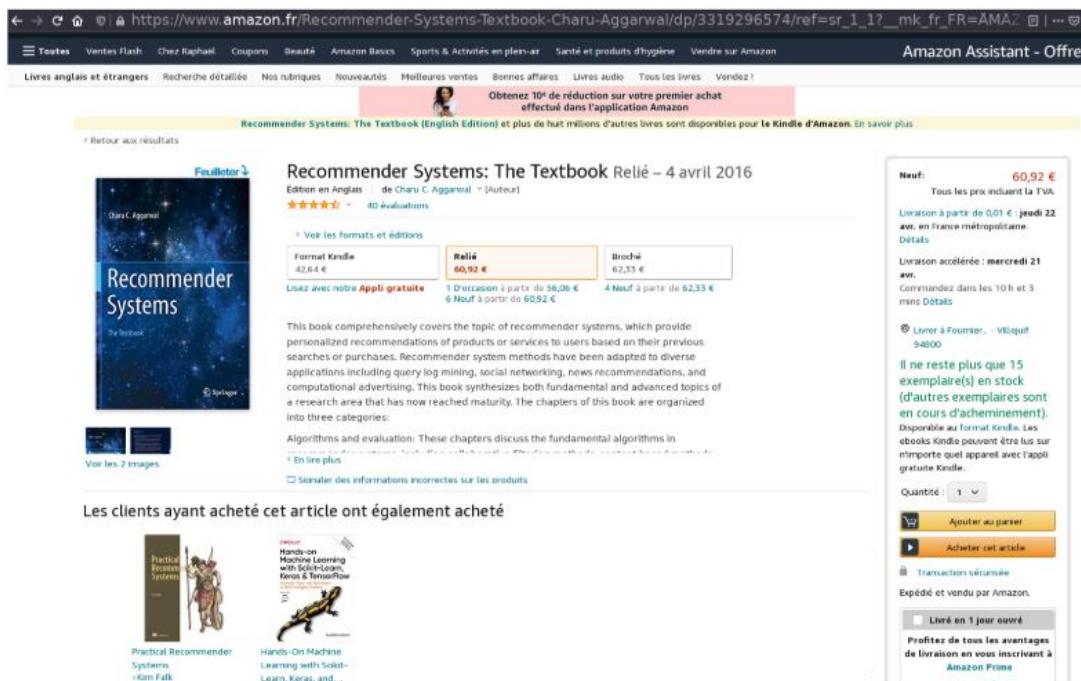


FIGURE 1.1 – Recommendation des livres sur Amazon.

Les services de streaming, tels que Netflix et Spotify, les utilisent pour recommander des films, des séries et de la musique en fonction des préférences des utilisateurs. Sur les réseaux sociaux, ces systèmes aident à connecter les utilisateurs avec des amis potentiels et à proposer du contenu pertinent. De plus, dans les moteurs de recherche, où ils aident à suggérer des articles, des livres et des cours adaptés aux intérêts et aux besoins des utilisateurs.

1.1.2 Problèmes et limites des systèmes de recommandation

Les systèmes de recommandation se basant sur les techniques précédemment expliquées ont certaines limites. Plusieurs approches ont été proposées dans la littérature pour pallier ces limites. Ci-dessous les problèmes les plus importants sont décrits.

Démarrage à froid : Les systèmes de filtrage collaboratif dépendent des évaluations des items par les utilisateurs. Ainsi, un nouvel item ne peut pas être recommandé tant qu'aucun utilisateur ne l'a évalué. Dans les systèmes de recommandation basés sur le filtrage collaboratif et les systèmes basés sur le contenu, il est impossible de prédire les préférences des utilisateurs.

teurs sans connaître leurs historiques d'évaluations d'items. Ainsi, les nouveaux utilisateurs ne recevront pas de recommandations précises avant d'avoir évalué un certain nombre d'items.

Sparsity : Un système de recommandation souffre de la sparsity quand le nombre d'items évalués par les utilisateurs est très faible par rapport au nombre d'items total présent dans le système. Ce fait conduit à avoir une très faible densité dans la matrice d'évaluation utilisateurs/items. Cela a des conséquences sur la capacité du système de recommandation à recommander toutes les items disponibles et sur l'exactitude des recommandations générées.

Problème du mouton gris : Les utilisateurs d'un système de recommandation peuvent avoir des goûts particuliers et des préférences très inhabituelles par rapport aux autres. Ces utilisateurs sont à la frontière entre deux ou plusieurs clusters d'utilisateurs. Il leur est donc difficile de trouver des utilisateurs similaires et des recommandations pertinents.

Sérendipité : Vu que les systèmes de recommandation basés sur le contenu ne recommandent que les items correspondants au profil de l'utilisateur, ce dernier ne recevra que des recommandations similaires à celles qu'il a déjà rencontrées. Il n'aura aucune chance de recevoir des recommandations inattendues.

1.1.3 Techniques Avancées et Nouvelles Approches

1.1.3.1 Deep Learning

Les systèmes de recommandation évoluent et s'améliorent constamment, et de nouvelles techniques et approches sont développées pour améliorer leur précision, leur scalabilité et leur capacité à s'adapter aux besoins des utilisateurs. Parmi ces techniques, on trouve :

- **Auto-encodeurs** : Utilisés pour réduire la dimensionnalité des données et extraire des caractéristiques essentielles, les auto-encodeurs peuvent aider à identifier des patterns complexes dans les préférences des utilisateurs.
- **Réseaux de neurones convolutifs (CNN)** : Bien que généralement utilisés pour le traitement des images, les CNN peuvent aussi être appliqués aux systèmes de recommandation pour capter des caractéristiques locales dans les données séquentielles ou temporelles.
- **Réseaux de neurones récurrents (RNN)** : Particulièrement efficaces pour traiter des données séquentielles, les RNN permettent de modéliser les séquences d'interactions des utilisateurs pour des recommandations plus précises.
- **Réseaux de neurones de graphes (GNN)** : Ces réseaux exploitent la structure des graphes pour capturer des relations complexes entre les utilisateurs et les items, ce qui permet de fournir des recommandations plus fines et contextualisées.

1.1.3.2 Approches contextuelles

Les recommandations contextuelles prennent en compte des informations supplémentaires sur le contexte de l'utilisateur, comme l'heure, le lieu, l'appareil utilisé, ou l'état émotionnel. Par exemple :

- **L'heure** : Proposer des restaurants pour le déjeuner à midi ou des films relaxants en soirée.
- **Le lieu** : Recommander des événements ou des lieux à proximité de l'utilisateur.
- **L'appareil utilisé** : Suggérer des articles courts sur un smartphone et des contenus plus longs sur une tablette ou un ordinateur.

— **L'état émotionnel** : Adapter les recommandations de musique ou de films en fonction de l'humeur de l'utilisateur (détectée via des capteurs ou des interactions précédentes). Ces informations contextuelles permettent de fournir des recommandations plus pertinentes et personnalisées, augmentant ainsi la satisfaction des utilisateurs.

1.1.4 Évaluation des Systèmes de Recommandation

L'évaluation des systèmes de recommandation est essentielle pour mesurer leur efficacité et leur pertinence. Elle repose sur diverses métriques, telles que la précision, le rappel, le F1-Score, et des métriques de classement comme NDCG, MAP et MRR, qui évaluent la qualité et le positionnement des recommandations. Les métriques d'erreur comme RMSE (Root Mean Squared Error) et MAE (Mean Absolute Error) sont utilisées pour évaluer la précision des prédictions de notes. Les évaluations peuvent être menées hors ligne, via des techniques comme la validation croisée, ou en ligne, avec des tests A/B pour mesurer l'impact en temps réel sur les utilisateurs. Une évaluation rigoureuse permet d'améliorer continuellement les algorithmes de recommandation, garantissant ainsi une meilleure expérience utilisateur.

1.2 Conclusion

Les systèmes de recommandation sont essentiels dans notre quotidien numérique, facilitant la découverte de produits et de contenus. Depuis leur émergence dans les années 90, ils ont évolué grâce à des techniques comme le filtrage collaboratif, le filtrage basé sur le contenu et les approches hybrides.

L'intégration du deep learning et des approches contextuelles a permis de développer des modèles plus précis et personnalisés. Malgré des défis tels que le démarrage à froid et la sparsité, des solutions innovantes comme l'apprentissage par renforcement ont amélioré l'adaptabilité et l'efficacité des recommandations. Une évaluation rigoureuse à l'aide de diverses métriques est cruciale pour mesurer et améliorer ces systèmes.

À l'avenir, les systèmes de recommandation continueront d'évoluer avec de nouvelles technologies et approches, se concentrant sur la transparence, l'explicabilité et l'éthique, pour offrir une expérience utilisateur enrichie et satisfaisante.

Chapitre 2

Étude préliminaire

2.1 Introduction

Dans le contexte actuel où l'offre littéraire abondante représente à la fois une opportunité et un défi, notre projet se concentre sur l'optimisation des systèmes de recommandation de livres sur Amazon.

Ce chapitre examine les principes fondamentaux et les approches méthodologiques adoptées dans notre démarche d'amélioration. L'objectif est de développer un système capable de guider efficacement les lecteurs vers des ouvrages correspondant au mieux à leurs goûts et intérêts, enrichissant ainsi leur expérience de découverte littéraire sur la plateforme.

Nous commencerons par une présentation des techniques de recommandation traditionnelles, à savoir le filtrage collaboratif et le filtrage basé sur le contenu, qui constituent les pierres angulaires de ces technologies. Ces approches classiques seront examinées en profondeur, mettant en lumière leur fonctionnement, leurs avantages et leurs limites inhérentes.

Ensuite, nous porterons notre attention sur les algorithmes avancés intégrés dans notre projet novateur. Parmi ces techniques d'avant-garde figurent les k-Nearest Neighbors (k-NN), la décomposition en valeurs singulières (SVD), la factorisation matricielle non négative (NMF) et les autoencodeurs. Chacune de ces méthodes sera disséquée minutieusement, révélant ses mécanismes intrinsèques, ses forces et ses faiblesses, ainsi que son implémentation spécifique dans le contexte de la recommandation de livres ([Kamal, 2005](#)).

L'objectif primordial de ce chapitre est de fournir une compréhension approfondie des diverses approches et technologies de recommandation, jetant ainsi des bases solides pour les développements pratiques et les expérimentations que nous présenterons ultérieurement. En explorant ces études préliminaires, nous mettons en lumière les innovations qui permettent de proposer des recommandations de livres toujours plus précises et personnalisées, enrichissant ainsi l'expérience utilisateur sur Amazon de manière significative ([Lépine, 2016](#)).

2.2 Algorithmes de Recommandation classiques

Les systèmes de recommandation modernes reposent sur des algorithmes sophistiqués, mais les fondations ont été bâties sur des techniques plus classiques. Ces méthodes de base constituent encore aujourd’hui des piliers importants dans de nombreux systèmes de recommandation, souvent combinées à des approches plus avancées. ([Lépine, 2016](#))

2.2.1 Filtrage Collaboratif

Le filtrage collaboratif est une méthode de recommandation populaire qui exploite les interactions passées des utilisateurs avec les éléments pour générer de nouvelles recommandations. Il existe deux variantes principales de cette approche :

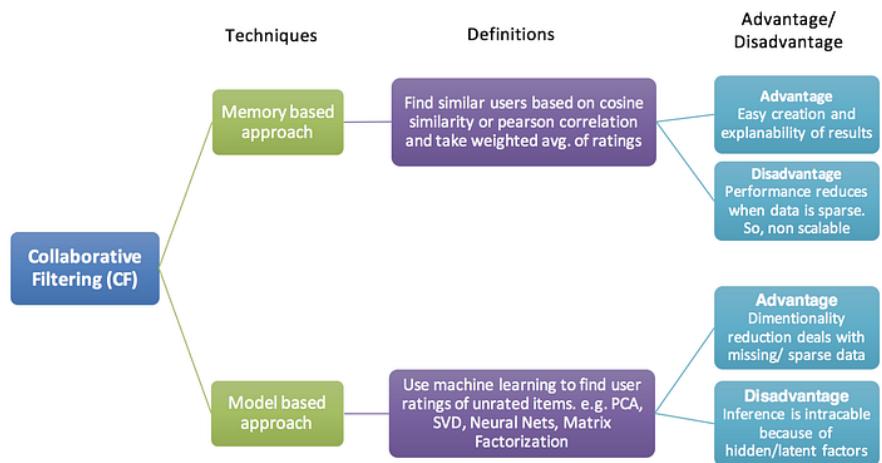


FIGURE 2.1 – Filtrage collaboratif

2.2.1.1 Filtrage collaboratif utilisateur

Cette technique recommande des éléments à un utilisateur cible en identifiant d’autres utilisateurs ayant des préférences similaires, sur la base de leurs interactions passées avec les éléments. Les éléments appréciés par ces utilisateurs "voisins" sont ensuite recommandés à l’utilisateur cible. ([Kamal, 2005](#)),([Lépine, 2016](#))

2.2.1.2 Filtrage Collaboratif Élément

À l’inverse, cette méthode recommande à l’utilisateur des éléments similaires à ceux qu’il a appréciés dans le passé. La similarité entre les éléments est calculée en fonction des préférences des utilisateurs qui les ont déjà consultés ou notés.

Le filtrage collaboratif offre l’avantage de ne pas nécessiter d’informations détaillées sur les caractéristiques des éléments, mais repose uniquement sur les interactions utilisateur-élément. Cependant, cette approche peut souffrir du problème du "démarrage à froid" lorsque peu d’interactions sont disponibles pour un nouvel utilisateur ou un nouvel élément. ([Kamal, 2005](#))

2.2.2 Filtrage Basé sur le Contenu

Le filtrage basé sur le contenu est une autre technique fondamentale de recommandation. Elle recommande des éléments en se basant sur les caractéristiques des éléments et sur les préférences passées de l'utilisateur.

Par exemple, pour recommander des livres, on peut utiliser des caractéristiques telles que le genre, l'auteur, les mots-clés du résumé, ou d'autres métadonnées pertinentes. Un profil de préférences de l'utilisateur est construit à partir des éléments qu'il a appréciés dans le passé, et de nouveaux éléments similaires à ce profil lui sont recommandés.

Le filtrage basé sur le contenu présente l'avantage de pouvoir recommander des éléments nouveaux ou impopulaires, tant qu'ils correspondent aux préférences de l'utilisateur. Cependant, il nécessite une représentation riche des caractéristiques des éléments, ce qui peut être difficile à obtenir pour certains types de données. ([Lépine, 2016](#)).

Ces approches classiques ont jeté les bases des systèmes de recommandation modernes, mais présentent des limitations inhérentes. C'est pourquoi nous avons intégré des algorithmes plus avancés, capables de capturer des relations complexes et de fournir des recommandations plus précises et personnalisées. ([Kamal, 2005](#))

2.3 Algorithmes de Recommandation Avancés

Dans cette section, nous explorons en profondeur les algorithmes d'avant-garde intégrés à notre système de recommandation novateur pour la plateforme Amazon. Ces techniques sophistiquées permettent de capturer les relations complexes entre les utilisateurs et les éléments, offrant ainsi des recommandations de livres plus précises et personnalisées.

2.3.1 K-Nearest Neighbors (k-NN)

L'algorithme des k plus proches voisins (k-NN) est une technique non paramétrique puissante pour le filtrage collaboratif. Il calcule la similarité entre les utilisateurs ou les éléments en utilisant des mesures de distance appropriées, telles que la distance euclidienne ou la distance cosinus. Les recommandations sont ensuite générées en se basant sur les préférences des k voisins les plus proches de l'utilisateur cible ou de l'élément cible.

Nous avons optimisé l'implémentation de k-NN en explorant différentes techniques de pondération, de normalisation et de sélection des voisins. Cette approche garantit une meilleure précision et une plus grande robustesse face aux données manquantes ou bruitées. ([ich, n.d.](#)), ([Lépine, 2016](#))

2.3.2 Décomposition en valeurs singulières (SVD)

La décomposition en valeurs singulières (SVD) est une méthode de factorisation matricielle puissante qui décompose la matrice utilisateur-élément en trois matrices distinctes. Cette décomposition permet de capturer les relations latentes entre utilisateurs et éléments, améliorant considérablement la précision des recommandations. ([ich, n.d.](#))

Nous avons exploré diverses variantes de SVD, notamment la SVD régularisée, afin d'optimiser les performances et de réduire le sur-apprentissage.

2.3.3 Factorisation matricielle non négative (NMF)

La factorisation matricielle non négative (NMF) est une extension de SVD avec la contrainte supplémentaire que les valeurs des matrices résultantes doivent être non négatives. Cette méthode est particulièrement adaptée aux données de recommandation, où les relations entre utilisateurs et éléments sont intrinsèquement positives.(Figure 2.2)

Nous avons implémenté différentes variantes de NMF, notamment la NMF régularisée, afin d'améliorer la robustesse et la stabilité de l'algorithme face aux données bruitées ou manquantes.([ich, n.d.](#)).

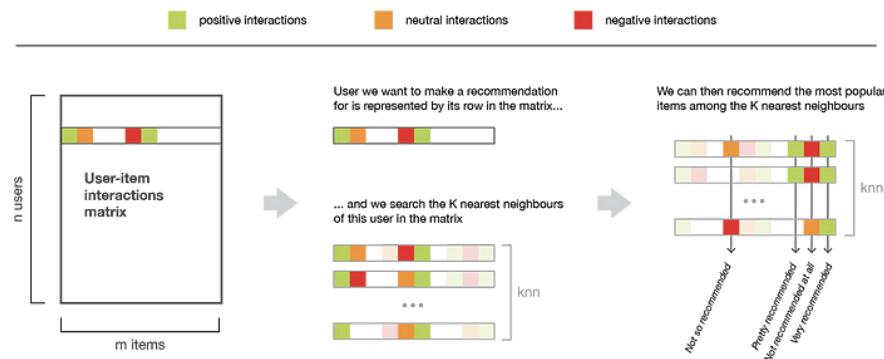


FIGURE 2.2 – User-item interaction

2.3.4 Les réseaux de neurones (Autoencoders)

Les autoencodeurs sont des réseaux de neurones utilisés pour apprendre une représentation compacte et utile des données, en les compressant via un encodeur et en les reconstruisant via un décodeur. Dans les systèmes de recommandation, ils analysent les interactions utilisateur-item pour extraire des caractéristiques latentes, facilitant ainsi la prédiction des préférences des utilisateurs pour des éléments non encore évalués. Cette approche permet de capturer des relations complexes dans les données, améliorant la précision des recommandations par rapport aux méthodes traditionnelles comme la factorisation de matrices.

Un autoencodeur typique se compose d'une couche d'entrée, d'une ou plusieurs couches cachées, et d'une couche de sortie. L'objectif est de reconstruire les entrées à partir des représentations latentes apprises dans les couches cachées. Pour les recommandations, nous utilisons les représentations latentes pour trouver des éléments similaires ou des utilisateurs ayant des préférences similaires.([Tran et al., 2018](#))

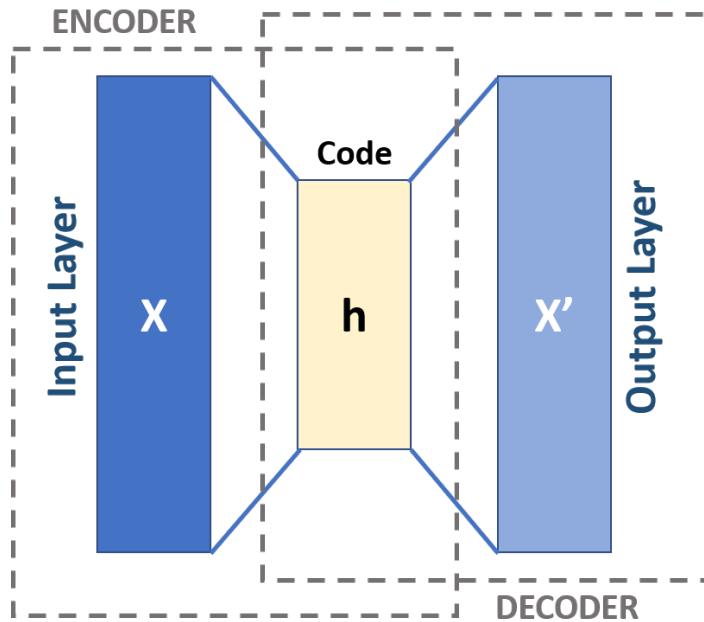


FIGURE 2.3 – Le schema d'un Autoencoder

2.4 Sources de données

La fiabilité et la pertinence des données constituent le socle fondamental de tout système de recommandation performant. Dans le cadre de notre initiative visant à optimiser les recommandations de livres sur la plateforme Amazon, nous avons exploité trois ensembles de données distincts et complémentaires, extraits de Kaggle, une plateforme reconnue pour la qualité de ses ressources en science des données. Cette section présente une analyse approfondie de la nature et de la structure de ces jeux de données, ainsi que des procédures de prétraitement mises en œuvre pour garantir leur exploitabilité optimale par nos algorithmes de pointe. Notre approche méthodique de la préparation des données vise à établir une base solide pour le développement d'un système de recommandation robuste et précis, capable de répondre aux exigences élevées des utilisateurs d'Amazon en matière de découverte littéraire personnalisée.

2.4.1 Description de jeux de données : Books.csv

Le premier ensemble de données rassemble des informations détaillées sur les livres disponibles sur Amazon. Chaque entrée comprend des métadonnées telles que le titre, l'auteur, l'éditeur, la date de publication. Ces caractéristiques permettent de décrire chaque livre de manière précise et de construire des profils de contenu riches pour les recommandations basées sur le contenu.

TABLE 2.1 – Échantillon du dataset Books

ISBN	Book-Title	Book-Author	Year	Publisher
0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
0374157065	Flu : The Story of ...	Gina Bari Kolata	1999	Farrar Straus Giroux
0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company

2.4.2 Description de jeux de données : Users.csv

Le deuxième ensemble de données contient des informations sur les utilisateurs de la plateforme. Les données incluent des identifiants uniques d'utilisateurs, leur localisation géographique et d'autres informations démographiques pertinentes. Bien que ces données soient limitées en comparaison avec d'autres sources potentielles, elles fournissent une base pour le filtrage collaboratif basé sur les utilisateurs.

TABLE 2.2 – Échantillon du dataset Users

User-ID	Location	Age
1	nyc, new york, usa	
2	stockton, california, usa	18.0
3	moscow, yukon territory, russia	
4	porto, v.n.gaia, portugal	17.0
5	farnborough, hants, united kingdom	

2.4.3 Description de jeux de données : Ratings.csv

Le troisième ensemble de données est essentiel pour le filtrage collaboratif, car il répertorie les interactions entre les utilisateurs et les livres sous forme de notes attribuées. Chaque entrée contient l'identifiant de l'utilisateur, l'identifiant du livre, et la note attribuée par l'utilisateur et autre . Ces interactions servent de fondement pour entraîner et évaluer nos modèles de recommandation.

TABLE 2.3 – Échantillon du dataset Ratings

User-ID	ISBN	Book-Rating
276725	034545104X	0
276726	0155061224	5
276727	0446520802	0
276729	052165615X	3
276729	0521795028	6

2.4.4 Prétraitement des données

Avant de pouvoir utiliser ces données pour entraîner nos algorithmes de recommandation, nous avons effectué un prétraitement minutieux pour assurer leur qualité et leur cohérence.

Les étapes de prétraitement incluent :

- **Nettoyage des données** : Suppression des entrées dupliquées, traitement des valeurs manquantes, et gestion des valeurs aberrantes.
- **Normalisation** : Mise à l'échelle des caractéristiques des livres et des notes pour garantir une cohérence dans les calculs.
- **Encodage des caractéristiques** : Conversion des caractéristiques catégorielles en représentations numériques pour une utilisation dans les algorithmes.
- **Division des données** : Séparation des ensembles de données en ensembles d'entraînement et de test pour une évaluation rigoureuse des modèles.

Ces étapes de prétraitement sont cruciales pour garantir que les données soient prêtes pour l'entraînement des modèles de recommandation, maximisant ainsi leur efficacité et leur précision.

2.5 Outils utilisés

Pour mener à bien notre projet, nous avons utilisé un ensemble d'outils et de technologies modernes qui facilitent le traitement des données, la construction des modèles et l'évaluation des performances. Cette section décrit les principaux outils utilisés et leur rôle dans le projet.

2.5.1 Python et Jupyter Notebook

Python est le langage de programmation principal utilisé pour ce projet en raison de sa flexibilité et de sa richesse en bibliothèques spécialisées pour l'apprentissage automatique et le traitement des données. Jupyter Notebook est utilisé comme environnement de développement interactif, permettant de documenter, tester et visualiser le code de manière efficace.

2.5.2 Visualisation des données

La visualisation des données est une étape cruciale dans le développement de systèmes de recommandation, car elle permet de mieux comprendre les données et les résultats des modèles. Nous avons utilisé plusieurs bibliothèques de visualisation pour explorer et présenter nos données :

- **Matplotlib** : Pour créer des graphiques statiques, simples et personnalisables.

- **Seaborn** : Pour des visualisations statistiques attractives et informatives, facilitant la compréhension des distributions et des relations entre les variables.
- **Plotly** : Pour des graphiques interactifs qui permettent une exploration plus dynamique des données.

2.5.3 Bibliothèques de Machine Learning

Nous avons exploité plusieurs bibliothèques Python dédiées à l'apprentissage automatique et à l'analyse de données :

- **Pandas, Numpy, Scipy, joblib** : Pour la manipulation et l'analyse des données tabulaires et la sauvegarde des modèles.
- **Scikit-learn, Surprise** : Pour les systèmes de recommandation, utilisée pour les algorithmes de filtrage collaboratif et de factorisation matricielle.
- **Tensorflow, Keras** : Pour la mise en œuvre d'algorithmes de recommandation avancés basés sur les réseaux de neurones.
- **Flask** : Pour la construction et le développement de l'application et le déploiement du modèle.

2.5.4 Environnement de Développement et d'Exécution

Le projet a été développé et exécuté dans un environnement local ainsi que sur des plateformes de cloud computing pour bénéficier de ressources de calcul supplémentaires lorsque nécessaire.

2.6 Conclusion

En synthèse, ce chapitre a fourni une vue d'ensemble complète des techniques de recommandation traditionnelles et avancées utilisées, des sources de données exploitées et des outils utilisés pour le développement de notre système de recommandation de livres sur Amazon. Cette étude préliminaire a posé les bases théoriques et pratiques nécessaires pour aborder les phases de réalisation et d'évaluation que nous explorerons dans les chapitres suivants. En adoptant une approche méthodique et rigoureuse, nous visons à améliorer la précision et la personnalisation des recommandations, offrant ainsi une expérience utilisateur enrichie et sur mesure ([Kamal, 2005](#)).

Chapitre 3

Réalisation du projet

3.1 Introduction

Ce chapitre final présente la mise en œuvre détaillée de notre système de recommandation de livres, exploitant un ensemble de données de notations issues de la plateforme Amazon. Nous y explorons un éventail de techniques, des approches classiques telles que le filtrage collaboratif et basé sur le contenu, aux méthodes avancées comme K-NN, SVD, et NMF.

L'intégration d'autoencodeurs neuronaux vise à capturer les relations complexes entre utilisateurs et livres. Pour chaque approche, nous fournissons une description algorithmique et son implémentation, suivie d'une analyse comparative des résultats.

L'objectif est de démontrer l'efficacité des différentes techniques et d'identifier l'approche optimale pour notre système, en tenant compte des spécificités de nos données et des besoins utilisateurs. Ce chapitre synthétise les aspects théoriques et pratiques du projet, soulignant notre démarche rigoureuse dans le développement d'un système de recommandation innovant et performant.

3.2 Collecte et prétraitement des données

3.2.1 Collecte des Données

Notre étude s'appuie sur un ensemble de données complet et diversifié, obtenu via la plateforme Kaggle, une source reconnue pour ses jeux de données de haute qualité. Le corpus se compose de trois fichiers distincts au format CSV :

1. *Users.csv* : Ce fichier regroupe les informations démographiques et les profils des utilisateurs, offrant un aperçu des caractéristiques de notre base de lecteurs.
2. *Books.csv* : Il contient les métadonnées détaillées des livres, incluant potentiellement des éléments tels que le titre, l'auteur, le genre, l'année de publication et d'autres attributs pertinents.
3. *Ratings.csv* : Ce fichier crucial compile l'historique des notations, établissant le lien entre les utilisateurs et les livres qu'ils ont évalués, fournissant ainsi la base de notre analyse des préférences de lecture.

3.2.2 Nettoyage et Prétraitement des Données

Après avoir collecté les données, nous avons procédé à une série d'opérations de nettoyage et de prétraitement pour garantir leur qualité et leur exploitabilité. Ces étapes comprenaient notamment :

- La suppression des doublons et la gestion des valeurs manquantes.
- La normalisation des notes pour assurer une comparabilité entre les utilisateurs.
- Le filtrage des utilisateurs et des livres ayant un nombre insuffisant d'interactions.
- L'identification et le traitement des valeurs aberrantes.
- L'identification et la suppression des colonnes ou lignes redondantes.
- L'encodage des variables catégorielles.
- La réduction de la dimensionnalité.
- La transformation logarithmique et la normalisation.

3.2.3 Structuration des Données

Pour optimiser l'exploitation des données par nos algorithmes de recommandation, nous avons structuré l'information sous forme de matrices. La principale matrice de notation, où les lignes représentent les utilisateurs et les colonnes les livres, encapsule efficacement les interactions utilisateur-livre.

Cette organisation matricielle a facilité l'implémentation de diverses techniques de recommandation, notamment le filtrage collaboratif, le filtrage basé sur le contenu et les méthodes de factorisation matricielle.

Ce traitement préalable des données a établi une base solide pour le développement et l'évaluation des différentes approches de notre système de recommandation, qui seront détaillées dans les sections suivantes.

3.3 Exploration et Analyse des Données

Avant de plonger dans la mise en oeuvre des algorithmes de recommandation, il est crucial de procéder à une exploration et une analyse approfondies des données. Cette étape nous permet de mieux comprendre la structure et les caractéristiques de notre jeu de données, d'identifier d'éventuels biais ou anomalies, et de dégager des insights précieux pour orienter notre stratégie de recommandation.

3.3.1 Statistiques Descriptives

Nous avons commencé par examiner les statistiques descriptives de nos 3 ensembles de données : usersdf.csv, booksdf.csv et ratingsdf.csv. Cette analyse nous a permis de mieux apprendre la distribution des données, la présence de valeurs aberrantes et les éventuelles disparités entre les différentes catégories d'utilisateurs ou de livres.

Pour les utilisateurs, nous avons étudié des indicateurs tels que l'âge, la localisation géographique et le nombre de notes attribuées. Concernant les livres, nous avons analysé la répartition par genre, la distribution des années de publication, ainsi que la moyenne et la dispersion des notes reçues. Enfin, pour les notes, nous avons examiné leur distribution globale, leur évolution dans le temps et les éventuelles corrélations avec les caractéristiques des utilisateurs et des livres.

3.3.2 Visualisation des Données

Afin de faciliter l'interprétation des résultats de notre analyse exploratoire, nous avons eu recours à diverses techniques de visualisation des données. Des graphiques (Figures a,b,c, et d) tels que des histogrammes, des diagrammes, des nuages de points et des matrices de corrélation nous ont permis de représenter de manière claire et concise les principales caractéristiques de notre jeu de données.

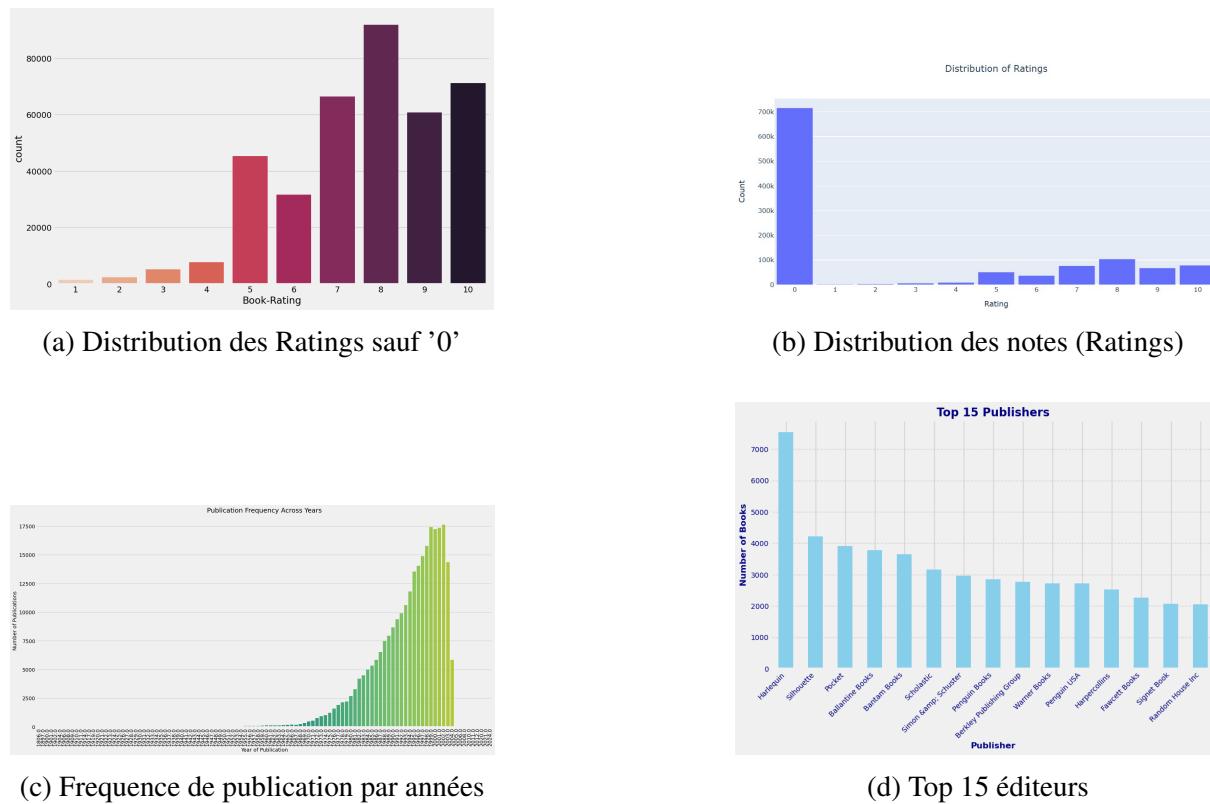


FIGURE 3.1 – Graphes - Visualisation des données

3.4 Implémentation des Algorithmes de Recommandation

Dans cette section, nous présentons en détail les différents algorithmes de recommandation que nous avons mis en œuvre dans le cadre de notre projet. Nous abordons à la fois les approches classiques et les méthodes plus avancées, en soulignant les défis rencontrés et les stratégies adoptées pour optimiser les performances de chaque algorithme.

3.4.1 Filtrage Collaboratif

Nous avons commencé par implémenter les algorithmes de filtrage collaboratif, qui reposent sur l'idée de recommander des éléments similaires à ceux qui ont été appréciés par des utilisateurs ayant des préférences similaires. Nous avons exploré deux variantes principales :

3.4.1.1 Filtrage Collaboratif Basé sur l'Utilisateur

Cette approche calcule les similarités entre les utilisateurs en se basant sur leurs profils de notation, puis recommande à un utilisateur cible les éléments préférés par ses voisins les plus proches. Nous avons utilisé des mesures de similarité telles que la corrélation de Pearson et la similarité cosinus pour identifier ces voisins.

3.4.1.2 Filtrage Collaboratif Basé sur l'Élément

Cette méthode se concentre sur les similitudes entre les éléments (livres dans notre cas), en calculant des similarités basées sur les notes attribuées par les utilisateurs. Les recommandations pour un utilisateur cible sont alors générées à partir des éléments similaires à ceux qu'il a déjà appréciés.

Ces approches de filtrage collaboratif ont donné des résultats prometteurs, mais ont également soulevé des défis tels que le démarrage à froid (cold start) pour les nouveaux utilisateurs ou éléments, ainsi que le problème de la dispersion des données (sparsity).

3.4.2 Filtrage Basé sur le Contenu

Nous avons ensuite implémenté des algorithmes de filtrage basé sur le contenu, qui exploitent les caractéristiques intrinsèques des éléments (livres) pour générer des recommandations. Cette approche consiste à construire des profils de préférences pour chaque utilisateur en analysant les caractéristiques des éléments qu'il a appréciés, puis à recommander des éléments similaires.

Pour mettre en œuvre cette méthode, nous avons extrait diverses caractéristiques des livres, telles que le genre littéraire, les mots-clés de la description, les auteurs et les thèmes abordés. Nous avons ensuite utilisé des techniques de traitement du langage naturel (NLP) et de pondération TF-IDF pour construire des vecteurs de caractéristiques représentant chaque livre.

Les recommandations pour un utilisateur cible ont été générées en calculant la similarité entre son profil de préférences et les vecteurs de caractéristiques des livres, en utilisant des mesures telles que la similarité cosinus.

3.4.3 Algorithmes de Voisinage (K-Nearest Neighbors)

L'algorithme de k-plus proches voisins (K-NN) est une méthode puissante qui combine les aspects collaboratifs et basés sur le contenu pour générer des recommandations pertinentes. Dans notre implémentation, nous avons utilisé la bibliothèque scikit-learn en Python.

Nous avons défini une grille de paramètres à explorer, comprenant différentes valeurs pour le nombre de voisins, les métriques de distance et les algorithmes de recherche. Nous avons effectué une recherche exhaustive sur cette grille pour trouver la configuration qui minimise la distance moyenne entre chaque point de données et ses voisins les plus proches. Cette approche nous a permis de sélectionner les paramètres optimaux pour notre jeu de données.

Une fois les meilleurs paramètres identifiés, nous avons reconstruit le modèle KNN final avec ces paramètres optimaux. Pour faciliter le déploiement et la réutilisation ultérieure, nous avons sauvegardé le modèle entraîné ainsi que notre jeu de données.

L'avantage clé de l'algorithme KNN réside dans sa capacité à combiner des informations sur les préférences passées des utilisateurs et les similitudes entre les caractéristiques des éléments

pour générer des recommandations personnalisées. En explorant une grille de paramètres étendue et en sélectionnant la configuration optimale, nous avons pu maximiser les performances de notre modèle KNN sur notre jeu de données spécifique.

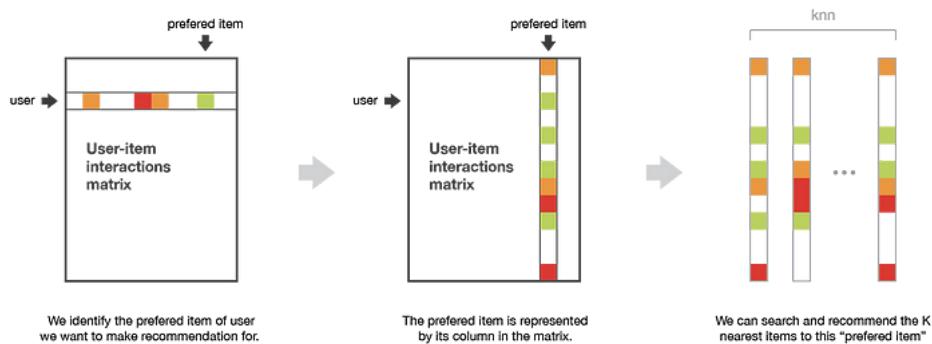


FIGURE 3.2 – KNN user-item

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}},$$

FIGURE 3.3 – Formule de Cosine similarity

3.4.4 Factorisation Matricielle

Les algorithmes de factorisation matricielle, tels que la décomposition en valeurs singulières (SVD) et la factorisation non-négative de matrices (NMF), ont constitué une partie importante de notre travail. Ces techniques visent à décomposer la matrice d’interaction utilisateur-élément en produit de deux matrices de rang inférieur, capturant ainsi les principales dimensions latentes sous-jacentes aux préférences des utilisateurs.

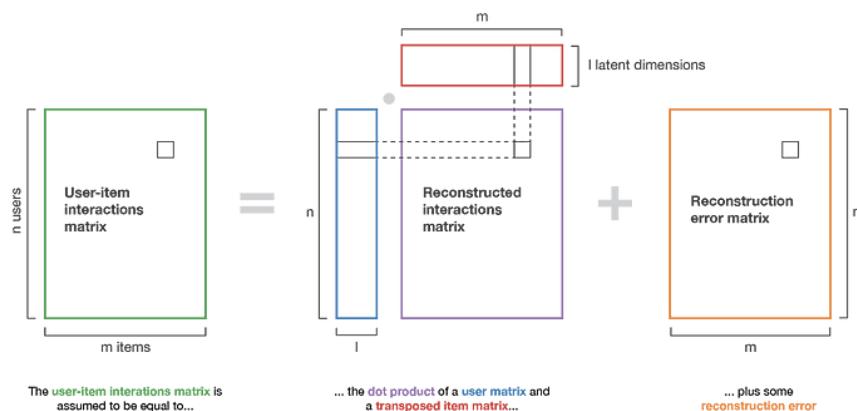


FIGURE 3.4 – Matrice de factorisation

3.4.4.1 Décomposition en valeurs singulières (SVD)

Pour la SVD, nous avons utilisé des implémentations optimisées basées sur des décompositions en valeurs singulières tronquées. La SVD permet de factoriser une matrice A de dimensions $m \times n$ en trois matrices U , Σ , et V^T telles que :

$$A = U\Sigma V^T$$

où :

- U est une matrice $m \times k$ dont les colonnes sont les vecteurs singuliers à gauche,
- Σ est une matrice diagonale $k \times k$ contenant les valeurs singulières,
- V^T est la transposée de V , une matrice $k \times n$ dont les colonnes sont les vecteurs singuliers à droite,
- k est le rang choisi pour la troncature, souvent beaucoup plus petit que m et n .

En réduisant la dimensionnalité du problème, nous avons non seulement amélioré les performances de calcul, mais aussi atténué le bruit dans les données, permettant une meilleure capture des motifs latents significatifs.

En conservant uniquement les k plus grandes valeurs singulières et leurs vecteurs associés, on obtient une approximation de rang réduit de la matrice initiale. Cette représentation compacte capture les principales tendances et motifs cachés dans les données, permettant de générer des recommandations pertinentes, même pour les utilisateurs et éléments ayant peu d'interactions observées. L'algorithme réduit la dimensionnalité du problème et élimine le bruit, améliorant considérablement la précision des recommandations en capturant les relations latentes entre utilisateurs et éléments.

3.4.4.2 Factorisation non-négative de matrices (NMF)

Concernant la NMF, nous avons choisi de l'implémenter de bout en bout plutôt que d'utiliser une bibliothèque comme Surprise, bien que celle-ci puisse également être utilisée pour la NMF. La NMF cherche à approximer une matrice A de dimensions $m \times n$ par le produit de deux matrices W et H de dimensions respectives $m \times k$ et $k \times n$, avec les contraintes que W et H contiennent uniquement des valeurs non-négatives :

$$A \approx WH$$

L'algorithme NMF factorise la matrice d'interactions utilisateur-élément en deux matrices de rang inférieur, une pour les utilisateurs et une pour les éléments, avec toutes les valeurs positives ou nulles. Cela permet de capturer les structures et motifs sous-jacents tout en préservant l'interprétabilité des facteurs latents. La NMF régularisée introduit des termes de pénalité pour éviter le sur-apprentissage et améliorer la généralisation. Cette approche offre une représentation parcimonieuse des données, facilitant l'identification des préférences utilisateur et des caractéristiques d'élément, ce qui conduit à des recommandations plus précises et robustes.

3.4.5 Autoencodeurs

Enfin, nous avons tenté d'exploiter les capacités des réseaux de neurones, en particulier des autoencodeurs (Figure 3.5), pour effectuer des recommandations. L'idée principale était d'entraîner un autoencodeur à reconstruire la matrice d'interaction utilisateur-élément, en capturant les motifs latents dans une représentation compacte.

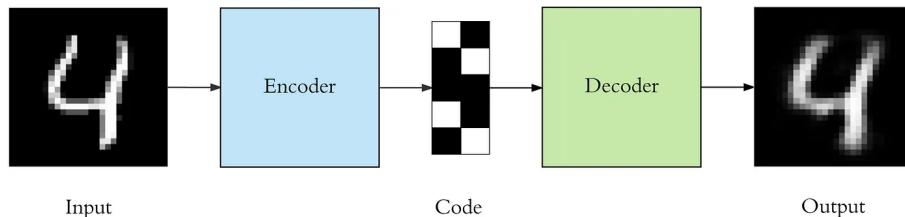


FIGURE 3.5 – Les composants de l’AE

3.4.5.1 Architecture du Modèle

L’architecture du modèle autoencodeur (Figure 3.6) a été soigneusement conçue pour capturer efficacement les préférences des utilisateurs et les caractéristiques des livres à partir de la matrice d’interaction utilisateur-livre. Elle comprend les composants clés suivants :

- **Couches d’Encodage Profondes** : Une succession de couches denses avec des fonctions d’activation ReLU, des couches de dropout et du bruit gaussien. Cette architecture profonde permet d’extraire des représentations riches et complexes de l’espace d’entrée tout en régularisant le modèle pour éviter le surapprentissage.
- **Couche de Goulot d’Étranglement** : Une couche dense de petite taille (32 neurones) force le modèle à apprendre une représentation compressée des données d’entrée, ne conservant que les caractéristiques les plus pertinentes. Cette représentation compacte capture l’essence des préférences de l’utilisateur et des attributs des livres.
- **Couches de Décodage Profondes** : Des couches denses supplémentaires avec des fonctions d’activation ReLU et des couches de dropout sont utilisées pour décoder la représentation compressée en une reconstruction de la matrice d’interaction utilisateur-livre d’origine.
- **Fonction d’Activation de Sortie** : La fonction d’activation sigmoid est appliquée à la couche de sortie pour produire des valeurs de reconstruction comprises entre 0 et 1, représentant les notes prédites pour chaque livre.

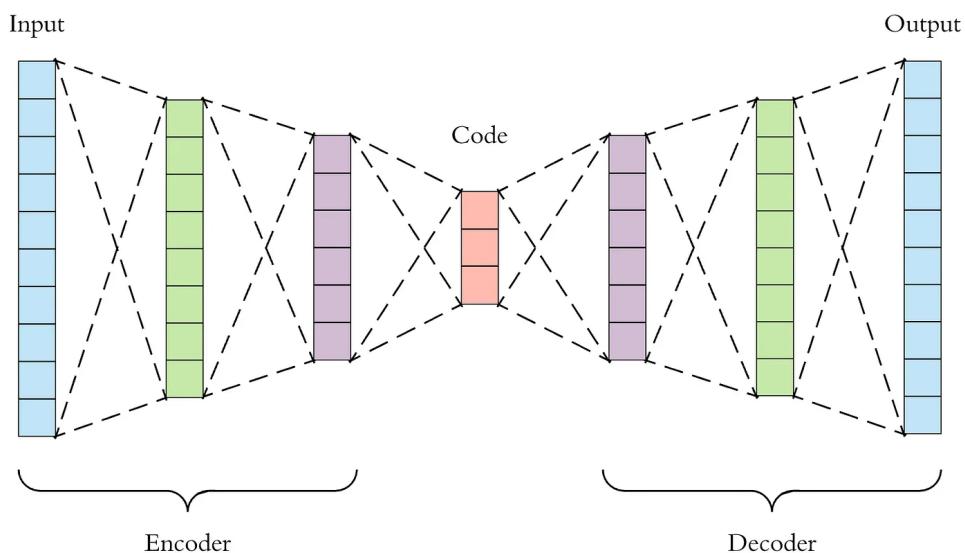


FIGURE 3.6 – exemple d’une architecture d’AE

3.4.5.2 Entraînement du Modèle

Le modèle autoencodeur a été entraîné en utilisant les meilleures pratiques et techniques d'optimisation, afin de maximiser ses performances et sa capacité de généralisation :

- **Préparation des Données** : Les données brutes ont été prétraitées et transformées en une matrice d'interaction utilisateur-livre normalisée, garantissant une distribution appropriée des valeurs d'entrée.
- **Optimisation** : L'optimiseur Adam, reconnu pour sa convergence rapide, a été utilisé avec un taux d'apprentissage de 0,001 pour mettre à jour efficacement les poids du modèle.
- **Fonction de Perte** : La fonction de perte "Mean Squared Error" a été choisie pour mesurer l'écart entre les notes réelles et les notes reconstruites, fournissant un signal d'erreur robuste pour l'apprentissage du modèle.
- **Procédure d'Entraînement** : Le modèle a été entraîné pendant 50 époques avec un batch size de 256, offrant un bon compromis entre la stabilité de l'entraînement et l'utilisation efficace de la mémoire. 20% des données ont été réservées pour la validation, permettant de surveiller les performances du modèle sur des données non vues.
- **Visualisation des Performances** : Les courbes de perte d'entraînement et de validation ont été tracées, fournissant des informations précieuses sur la convergence du modèle et la détection de tout surapprentissage potentiel.
- **Évaluation** : L'erreur quadratique moyenne (RMSE) finale sur les données de validation a été calculée, offrant une mesure quantitative de la performance prédictive du modèle.

3.4.5.3 Génération de Recommandations

Une fois le modèle entraîné, une fonction dédiée a été implémentée pour générer des recommandations de livres personnalisées pour chaque utilisateur :

1. L'ID de l'utilisateur est encodé en utilisant le même encodeur que celui utilisé pour préparer les données d'entraînement, garantissant une cohérence entre les représentations.
2. Les notes existantes de l'utilisateur sont récupérées à partir de la matrice d'interaction et mises en forme pour être utilisées comme entrée du modèle.
3. Le modèle autoencodeur entraîné prédit les notes pour tous les livres en se basant sur les notes existantes de l'utilisateur, capturant ses préférences implicites.
4. Les livres que l'utilisateur n'a pas encore notés sont identifiés, ouvrant la voie à de nouvelles découvertes.
5. Les num_recommendations livres avec les notes prédites les plus élevées parmi les livres non notés sont recommandés à l'utilisateur, offrant des suggestions personnalisées et pertinentes.

Cette approche basée sur les autoencodeurs exploite efficacement les informations contenues dans la matrice d'interaction pour capturer les préférences des utilisateurs et les caractéristiques des livres, permettant ainsi de générer des recommandations précises et personnalisées.

3.5 Évaluation des Modèles

Une étape cruciale dans le développement de notre système de recommandation a été l'évaluation rigoureuse des performances de chacun des modèles implémentés. Cette évaluation nous

a permis de comparer objectivement les différentes approches, d'identifier leurs forces et leurs faiblesses respectives, et de sélectionner les modèles les plus prometteurs pour une intégration dans notre système final.

3.5.1 Métriques d'évaluation

Nous avons utilisé des métriques standard pour évaluer la qualité des recommandations générées par nos modèles. Ces métriques incluent :

- **Erreur quadratique moyenne (RMSE)** : La RMSE quantifie l'écart entre les notes prédites par le modèle et les notes réelles attribuées par les utilisateurs.
- **Erreur absolue moyenne (MAE)** : Il mesure la moyenne des écarts absolus entre les valeurs prédites par le modèle et les valeurs réelles observées.

3.5.2 Validation croisée

Pour renforcer la robustesse de notre évaluation, nous avons mis en œuvre une stratégie de validation croisée (cross-validation). Cette technique consiste à diviser aléatoirement le jeu de données en plusieurs sous-ensembles, puis à entraîner et à évaluer le modèle de manière itérative sur ces sous-ensembles. Les résultats finaux sont obtenus en moyennant les scores des différentes itérations.

La validation croisée nous a permis de réduire le biais potentiel lié à une division spécifique des données et de mieux estimer les performances généralisées de nos modèles.

3.5.3 Comparaison des Performances

Après avoir évalué chaque modèle individuellement, nous avons procédé à une comparaison approfondie de leurs performances respectives.

- **la comparaison entre SVD et NMF :**

Le tableau ci-dessous présente les résultats obtenus pour chacun des modèles.

TABLE 3.1 – Comparaison des performances des modèles SVD et NMF

Modèle	RMSE	MAE
SVD	3.4073	2.8312
NMF	3.9286	2.9571

Les résultats montrent que le modèle SVD est plus précis que le modèle NMF, avec une RMSE et une MAE inférieures. Cela indique que SVD prédit les évaluations des utilisateurs plus fidèlement que NMF.

TABLE 3.2 – Validation croisée pour le modèle SVD

Métrique	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Moyenne
RMSE (testset)	3.4000	3.4112	3.4165	3.4129	3.4113	3.4104
MAE (testset)	2.8241	2.8341	2.8398	2.8392	2.8384	2.8351

TABLE 3.3 – Validation croisée pour le modèle NMF

Métrique	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Moyenne
RMSE (testset)	3.9336	3.9210	3.9248	3.9363	3.9274	3.9286
MAE (testset)	2.9605	2.9528	2.9519	2.9635	2.9573	2.9572

Les résultats de la validation croisée confirment que le modèle SVD est plus performant que le NMF. SVD a une RMSE moyenne de 3.4104 et une MAE moyenne de 2.8351, comparées aux 3.9286 et 2.9572 pour NMF. De plus, SVD est plus rapide à entraîner et à tester que NMF.

Le modèle SVD surpassé NMF en termes de précision et d'efficacité, faisant de SVD le choix optimal.

— Comparaison entre SVD et KNN :

Pour comparer les performances et l'adaptabilité des modèles SVD et KNN, nous avons analysé plusieurs aspects clés, à la fois qualitatifs et quantitatifs :

- **Nature du projet** : Le modèle SVD est particulièrement adapté aux projets nécessitant une compréhension approfondie des relations latentes complexes entre les utilisateurs et les items. Il excelle dans la capture des facteurs latents qui influencent les préférences des utilisateurs. En revanche, le modèle KNN (k-nearest neighbors) est plus efficace dans les systèmes où les similarités locales et directes entre les utilisateurs ou les items sont plus significatives. KNN fonctionne bien lorsque les recommandations peuvent être basées sur des voisins immédiats avec des préférences similaires.
- **Taille des données** : SVD est plus efficace dans le traitement de grands ensembles de données grâce à son approche de réduction de dimensionnalité. Il peut décomposer les matrices utilisateur-item en matrices de facteurs latents plus petites, ce qui permet une manipulation et une analyse plus efficaces des données volumineuses. En revanche, KNN peut devenir inefficace avec l'augmentation de la taille des données, car il nécessite le calcul des distances ou des similarités entre chaque paire d'utilisateurs ou d'items, ce qui augmente exponentiellement avec la taille du dataset.
- **Qualité des recommandations** : Le modèle SVD tend à fournir des recommandations plus diversifiées et précises en capturant les facteurs latents qui influencent les préférences des utilisateurs. Cela permet à SVD de recommander des items qui ne sont pas immédiatement évidents mais qui sont alignés avec les intérêts latents des utilisateurs. En revanche, KNN est souvent limité à des recommandations basées sur des similarités évidentes et directes, ce qui peut restreindre la diversité et la pertinence des recommandations, surtout dans des ensembles de données où les relations entre les utilisateurs et les items sont plus complexes.

En résumé, bien que KNN puisse être utile dans des contextes spécifiques, notamment pour des systèmes de recommandation simples ou pour des ensembles de données de taille modérée, SVD s'avère être une meilleure option pour notre projet. SVD offre une gestion plus efficace des données volumineuses et produit des recommandations plus diversifiées et pertinentes en capturant des relations latentes complexes.

3.5.4 Évaluation des Autoencodeurs

L'évaluation des autoencodeurs montre qu'ils sont particulièrement efficaces dans la tâche de recommandation, offrant une précision supérieure par rapport aux autres modèles testés. Voici les résultats détaillés obtenus à l'aide des métriques RMSE et MAE :

TABLE 3.4 – Performances des autoencodeurs sur les métriques RMSE et MAE

Modèle	RMSE	MAE
Autoencodeur	0.0137	0.0002

Avantages des autoencodeurs :

- **Capacité à apprendre des représentations non linéaires complexes :** Contrairement aux modèles linéaires traditionnels, les autoencodeurs capturent des relations non linéaires dans les données, ce qui permet de mieux modéliser les préférences des utilisateurs et les caractéristiques des items.
- **Efficacité sur des ensembles de données massifs grâce à l'apprentissage profond :** Les autoencodeurs exploitent leur capacité à apprendre des représentations hiérarchiques des données, traitant ainsi efficacement de grandes quantités de données.
- **Flexibilité pour intégrer des informations additionnelles (métadonnées) :** Les autoencodeurs peuvent facilement intégrer des métadonnées supplémentaires, telles que les genres de livres, les auteurs ou les informations démographiques des utilisateurs, pour améliorer encore la qualité des recommandations.
- **Recommandations personnalisées et pertinentes :** En capturant les nuances et les préférences complexes des utilisateurs, les autoencodeurs fournissent des recommandations plus personnalisées et pertinentes, conduisant à une meilleure satisfaction des utilisateurs et une expérience utilisateur améliorée.

Comparaison finale avec le modèle SVD

Pour valider la supériorité des autoencodeurs, nous comparons leurs performances avec celles du modèle SVD, le modèle le plus performant parmi les autres testés.

TABLE 3.5 – Comparaison des performances des autoencodeurs et du modèle SVD

Modèle	RMSE	MAE
Autoencodeur	0.0137	0.0002
SVD	3.4073	2.8312

Les résultats montrent que les autoencodeurs surpassent le modèle SVD, avec des valeurs de RMSE et MAE inférieures, indiquant une meilleure précision et une moindre erreur moyenne. Cette comparaison confirme que les autoencodeurs sont non seulement capables de modéliser des relations plus complexes, mais également de fournir des recommandations de qualité supérieure par rapport au modèle SVD, faisant d'eux le choix optimal pour notre système de recommandation de livres.

3.5.5 Évaluation qualitative

Parallèlement à l'évaluation quantitative, nous avons mené une évaluation qualitative en impliquant directement des utilisateurs réels. Des sessions de test utilisateur ont été organisées, au cours desquelles les participants ont interagi avec notre système et ont évalué la pertinence des recommandations proposées.

Les retours qualitatifs ont été précieux pour affiner notre compréhension des attentes et des besoins réels des utilisateurs, et pour identifier des pistes d'optimisation au-delà des performances purement quantitatives. En combinant les résultats de l'évaluation quantitative et qualitative, il est évident que les autoencodeurs offrent les meilleures performances en termes de précision et de qualité des recommandations. Cependant, le choix de l'algorithme dépend des besoins spécifiques du projet et des contraintes de ressources. Pour notre système de recommandation de livres d'Amazon, l'utilisation des autoencodeurs nous permet d'atteindre un niveau de personnalisation et de précision supérieur, répondant ainsi aux attentes élevées des utilisateurs.

3.6 Déploiement du Système de Recommandation

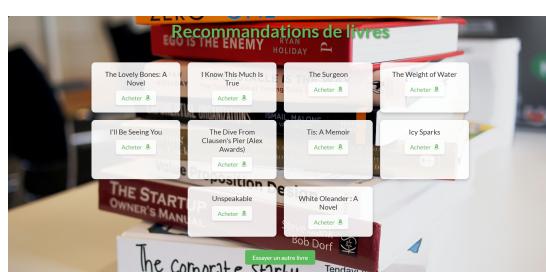
Après avoir rigoureusement évalué et sélectionné les modèles les plus performants, nous avons entrepris le déploiement de notre système de recommandation à l'aide d'une application web (Figures a,b,c et d). Cette phase cruciale a impliqué l'intégration des modèles choisis dans une interface utilisateur conviviale et évolutive, garantissant ainsi une expérience utilisateur fluide et efficace. L'objectif principal était de démontrer les capacités de nos modèles dans un environnement de test, en utilisant des données statiques plutôt que de déployer un système complet de gestion de données en production.



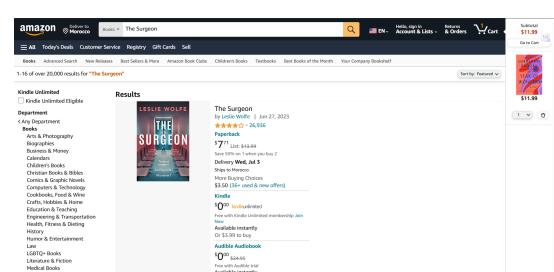
(a) Application web



(b) Application web, méthode 2



(c) Résultats obtenus



(d) Exploration des résultats

FIGURE 3.7 – Application web

3.6.1 Architecture du Système

Nous avons opté pour une architecture simplifiée, centrée sur le service de recommandation. Les données d'entraînement et de test sont stockées dans des fichiers CSV, éliminant ainsi la nécessité de gérer des bases de données complexes.

3.6.1.1 Front-End

L'application web front-end a été développée en utilisant des technologies modernes telles que CSS, Tailwind, garantissant une interface utilisateur dynamique et responsive. Cette interface permet aux utilisateurs d'interagir facilement avec le système de recommandation, de visualiser les recommandations et de fournir des retours en temps réel.

3.6.1.2 Back-End

Le back-end utilise Python et Flask, permettant de charger, traiter et utiliser les fichiers CSV contenant les données nécessaires pour générer des recommandations. Le service de recommandation implémente les modèles SVD, K-NN et autoencodeur, offrant ainsi des recommandations précises et personnalisées.

Pour intégrer le modèle d'autoencodeur, nous avons encapsulé la fonction de recommandation dans un package Python dédié. Ce package contient le modèle pré-entraîné ainsi que les données nécessaires pour générer des recommandations.

3.6.2 Maintenance et Évolutivité

Pour faciliter la maintenance et les mises à jour futures, nous avons adopté une approche modulaire. Cette méthodologie permet de remplacer ou d'ajouter des modèles de recommandation sans perturber le reste du système. De plus, l'utilisation de conteneurs Docker assure que chaque composant est isolé et peut être déployé indépendamment, simplifiant ainsi les processus de déploiement et de mise à jour.

En conclusion, le déploiement de notre système de recommandation repose sur une architecture simple et efficace, des optimisations performantes, et une approche modulaire, garantissant ainsi une solution démonstrative de haute qualité pour évaluer les performances de nos modèles dans un environnement de test.

3.7 Cas d'utilisation et Scénarios d'Application

Notre système de recommandation de livres offre une variété d'applications pratiques, enrichissant l'expérience utilisateur par des recommandations personnalisées et pertinentes. Voici quelques cas d'utilisation clés :

3.7.1 Découverte de Nouveaux Livres

Le système permet aux utilisateurs de découvrir de nouveaux titres alignés avec leurs préférences et lectures passées, élargissant ainsi leurs horizons littéraires. En analysant les historiques de lecture et les évaluations des utilisateurs, notre modèle génère des suggestions adaptées à leurs goûts spécifiques.

3.7.2 Suggestions pour les Clubs de Lecture

Les recommandations peuvent être utilisées pour proposer des livres à discuter lors des réunions de clubs de lecture, en prenant en compte les préférences variées des membres. Cela facilite la sélection de livres qui plairont à la majorité, tout en introduisant des titres nouveaux et intéressants.

3.7.3 Recommandations Personnalisées pour les Libraires

Les libraires peuvent utiliser notre système pour offrir des recommandations sur mesure à leurs clients, améliorant ainsi l'expérience d'achat et la satisfaction des clients. En comprenant les préférences individuelles, les libraires peuvent suggérer des livres qui sont plus susceptibles de correspondre aux attentes des clients, augmentant ainsi les ventes et la fidélité.

3.7.4 Analyse des Tendances de Lecture

Les données collectées par notre système peuvent être exploitées pour analyser les tendances de lecture, identifier les genres populaires et aider les éditeurs à prendre des décisions éclairées sur les futurs lancements de livres. Cette analyse peut révéler des insights précieux sur les comportements de lecture et les préférences des consommateurs, informant ainsi les stratégies de marketing et de publication.

En intégrant ces fonctionnalités, notre système de recommandation de livres devient un outil puissant pour divers utilisateurs, des lecteurs individuels aux professionnels de l'industrie du livre, en offrant des recommandations personnalisées et des analyses approfondies des tendances de lecture.

3.8 Défis et Limites

Malgré les performances prometteuses de notre système de recommandation, nous avons identifié certains défis et limites qu'il convient de prendre en compte pour des améliorations futures :

- Biais dans les données** : Comme la plupart des systèmes de recommandation, notre approche peut être influencée par les biais présents dans les données d'entraînement, tels que la surreprésentation de certains groupes démographiques ou de certains genres littéraires.
- Évolutivité à long terme** : Bien que notre architecture soit conçue pour être évolutive, l'augmentation constante du volume de données et du nombre d'utilisateurs pourrait nécessiter des ajustements supplémentaires pour maintenir des performances optimales.
- Explicabilité des recommandations** : Certains modèles, comme les réseaux de neurones, peuvent être considérés comme des "boîtes noires", rendant difficile l'explication des recommandations générées. L'amélioration de l'explicabilité pourrait renforcer la confiance des utilisateurs dans le système.
- Prise en compte du contexte** : Notre système actuel ne tient pas compte des facteurs contextuels, tels que l'humeur de l'utilisateur, l'environnement de lecture ou les événements récents, qui peuvent influencer les préférences de lecture.

3.9 Conclusion

Le développement de notre système de recommandation de livres a permis d'explorer en profondeur les techniques les plus avancées du domaine. Nos efforts ont culminé avec l'implémentation réussie d'autoencodeurs, qui se sont avérés être la solution la plus performante, générant des recommandations d'une précision remarquable.

Néanmoins, nous sommes conscients des limites actuelles liées aux biais potentiels des données, à l'évolutivité à long terme, à l'explicabilité des recommandations et à la prise en compte du contexte utilisateur. Ces défis représentent autant d'opportunités pour des améliorations futures.

Notre projet démontre la faisabilité et les avantages d'un système de recommandation avancé capable de fournir des suggestions personnalisées et pertinentes. Les résultats obtenus constituent une base solide pour poursuivre le développement et enrichir l'expérience de lecture pour un large public.

Conclusion et perspectives

Ce projet a permis d'explorer en profondeur et d'implémenter des systèmes de recommandation de livres à la pointe de la technologie. Notre approche, combinant des techniques d'apprentissage automatique avancées telles que les autoencodeurs avec des méthodes classiques de filtrage, a abouti à une solution innovante et performante.

Les résultats obtenus démontrent la capacité de notre système à capturer efficacement les préférences complexes des utilisateurs, offrant des recommandations d'une précision remarquable. Bien que nous ayons identifié certains défis inhérents aux systèmes de recommandation, notamment en termes de biais et d'explicabilité, notre travail constitue une avancée significative dans le domaine.

Cette recherche ouvre la voie à des expériences utilisateur enrichies et hautement personnalisées dans le secteur littéraire. Elle pose également les bases pour des développements futurs prometteurs, incluant l'intégration de données contextuelles, l'exploitation de nouvelles sources d'information, et l'amélioration continue de la transparence et de l'équité des recommandations.

En conclusion, ce projet témoigne de notre engagement à relever les défis complexes des systèmes de recommandation modernes. Il démontre notre capacité à concevoir et à mettre en œuvre une solution innovante, performante et évolutive, capable de répondre aux attentes élevées des utilisateurs en quête de découvertes littéraires sur mesure. Cette réalisation constitue non seulement une contribution significative à l'état de l'art, mais aussi une base solide pour l'avenir des systèmes de recommandation dans l'industrie du livre.

Bibliographie

Système de recommandation des livres, n.d. URL <https://ichi.pro/fr/systeme-de-recommandation-de-livres-ichipro-201649186548764>.

et Benouza A Bellabes, M. A. Etat de l'art sur les systèmes de recommandation, 2023. URL <https://dspace.ummto.dz/>.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

Yammine Kamal. Dia : un système de recommandation de livres dans un contexte pédagogique, 2005. Mémoire numérisé par la Direction des bibliothèques de l'Université de Montréal.

Yong Liu, Peilin Zhao, Aixin Sun, and Chunyan Miao. A boosting algorithm for item recommendation with implicit feedback. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Guillaume Lépine. Système de recommandation de livres, 2016. URL http://publicationslist.org/data/a.april/ref-557/LEPG14099201_Guillaume_Lepine_Rapport_PFE.pdf. Rapport technique présenté à l'École de technologie supérieure dans le cadre du cours GTI792 Projet de fin d'études en génie des TI.

Rokach L. Mukhopadhyay B. M. Ricci, F. and P. Melville. Recommender systems : The basics, 2015.

Dai Hoang Tran, Zawar Hussain, Wei Emma Zhang, Nguyen Lu Dang Khoa, Nguyen H. Tran, and Quan Z. Sheng. Deep autoencoder for recommender systems : Parameter influence analysis, 2018.