# Recommendation System

## Implémentation d'un algorithme de recommandation pour la plateforme de jeu Steam

Guillaume Charvolin

6 mars 2025

**Résumé executif**

Dans ce rapport, nous présentons la mise en place d'un système de recommandation de jeux vidéo, en utilisant des algorithmes tels que le filtrage collaboratif et une méthode basée sur le contenu. Le projet vise à personnaliser les recommandations pour les utilisateurs en se basant sur leurs achats.

# Table des matières

**1 Introduction** **3**
1.1 Contexte . . . . . 3
1.2 Objectifs du projet . . . . . 3

**2 Analyse du domaine et des besoins** **4**
2.1 Système de recommandation actuel . . . . . 4
2.2 Profil des utilisateurs . . . . . 4

**3 Choix des données, technologie et méthode** **5**
3.1 Jeux de données . . . . . 5
3.2 Exploration et preprocessing . . . . . 5
3.3 Métrics de test . . . . . 8
3.4 Séparation entraînement, test . . . . . 8
3.5 Méthodes de Recommandation . . . . . 8

**4 Analyse** **10**
4.1 Configurations et Hyperparamètres des Modèles Testés . . . . . 10
4.2 Résultats des Modèles Testés . . . . . 11
4.3 Recommandations des Modèles pour un Échantillon d'Utilisateurs . . . . . 13

**5 Diversification des recommandations** **15**
5.1 Méthode de Diversification : Maximal Marginal Relevance (MMR) . . . . . 15
5.2 Diversification des Recommandations appliqué . . . . . 16

**6 Conclusion** **19**

# 1 Introduction

## 1.1 Contexte

En tant que plateforme de distribution de jeux vidéo comptant des millions d'utilisateurs actifs, Steam fait face à un défi croissant : la découverte de nouveaux jeux pour les utilisateurs devient de plus en plus difficile en raison de l'augmentation exponentielle du nombre de titres disponibles. Avec des milliers de nouveaux jeux ajoutés chaque année, les utilisateurs se retrouvent souvent submergés par l'abondance de choix et peuvent avoir du mal à identifier les jeux qui correspondent à leurs préférences. Cela rend indispensable la mise en place d'un système de recommandation performant. Ce système doit permettre de filtrer et personnaliser les suggestions pour offrir une expérience utilisateur fluide et efficace, tout en augmentant l'engagement et les ventes sur la plateforme.

## 1.2 Objectifs du projet

L'objectif du projet est de développer un système de recommandation pour Steam qui remplisse deux fonctions essentielles. À court terme, le système doit maximiser les ventes en suggérant des jeux ayant une forte probabilité d'être achetés par l'utilisateur, en se basant sur son historique de jeu et ses préférences. À long terme, le système vise à fidéliser les utilisateurs en leur proposant des jeux adaptés à leurs goûts, tout en leur offrant une expérience plus personnalisée et diversifiée.

# 2 Analyse du domaine et des besoins

## 2.1 Système de recommandation actuel

Steam utilise principalement deux systèmes de recommandation pour améliorer l'expérience utilisateur. D'une part, le système basé sur les tags permet de suggérer des jeux en fonction des étiquettes attribuées à ceux que les utilisateurs ont déjà joués. Ces tags aident à catégoriser les jeux en fonction de leur genre, de leurs mécaniques et de leurs caractéristiques visuelles, facilitant ainsi la découverte de titres similaires [7], [2].

D'autre part, depuis 2020, Steam a intégré un système de recommandation interactif basé sur l'intelligence artificielle. Ce système analyse les habitudes de jeu des utilisateurs en utilisant des modèles d'apprentissage automatique pour offrir des recommandations personnalisées. Contrairement au système de tags, il ne repose pas sur les métadonnées, mais sur les données de jeu réelles des utilisateurs. Ce recommender est activé par défaut et accessible via l'onglet "Your Store", où les utilisateurs peuvent ajuster des paramètres pour affiner les suggestions selon la popularité ou la période de sortie des jeux, [3].

## 2.2 Profil des utilisateurs

Selon l'analyse de Christine Brownell, la base d'utilisateurs de Steam se divise en trois catégories principales. Les joueurs occasionnels jouent à des jeux non-core, souvent des titres légers et accessibles. Les light core gamers passent moins de cinq heures par semaine sur des jeux core, tandis que les heavy core gamers y consacrent plus de cinq heures par semaine. Les core games incluent des genres tels que les jeux de rôle, les FPS et les MMORPG, tandis que les jeux non-core incluent les jeux de puzzle ou de simulation plus légers [4].

# 3 Choix des données, technologie et méthode

## 3.1 Jeux de données

Le jeu de données principal utilisé dans ce projet est le **Steam Video Games**, disponible sur Kaggle [5]. Ce dataset, que nous appellerons `user_dataset`, contient 200 000 interactions entre utilisateurs et jeux. Chaque enregistrement indique si un jeu a été acheté ou joué, et, pour les interactions de type "joué", le temps de jeu est précisé. Cela représente environ 130 000 enregistrements d'achats.

Le second dataset est le **Steam Store Games (Clean dataset)**, que nous désignerons comme `item_dataset`. Ce dataset regroupe des informations sur 27 100 jeux : le nom, le publisher, le développeur, ainsi que des tags. Il comprend deux types de tags : les tags officiels fournis par Steam et les tags votés par la communauté, disponibles dans un fichier séparé du même dataset [6, 7]. Nous utiliserons ces deux ensembles de tags, et ferons référence au fichier des tags votés par la communauté sous le nom `tags_dataset` dans ce rapport.

## 3.2 Exploration et preprocessing

Dans la phase de prétraitement des données, nous avons créé un jeu de données nommé ratings_data qui répertorie les interactions d'achat entre utilisateurs et jeux. Ce jeu de données est structuré sous forme de paires userid-gameid, ne tenant pas compte du temps de jeu (playtime), tout les ratings ont finalement été mis à 1, bien qu'il soit disponible dans le dataset d'origine. Ce choix s'explique par la difficulté d'interpréter le temps de jeu comme une note de préférence.

Nous avons également filtré pour ne conserver que les jeux présents dans item_dataset et tags_dataset, de sorte que chaque interaction soit associée à des informations complètes, c'est à dire que pour chaque couple indiquant un achat userid-gameid ont peu à partir du gameid retrouvé ces tags et les informations du jeu.

Une analyse de la répartition des achats par utilisateur montre une concentration importante d'utilisateurs ayant acquis peu de jeux, rendant la distribution fortement déséquilibrée. En effet, le nombre d'utilisateurs possédant n jeux diminue rapidement au fur et à mesure que n augmente, créant une décroissance exponentielle typique.

Afin de garantir des données suffisantes pour l'évaluation des modèles de recommandation, les utilisateurs ayant acheté moins de quatre jeux ont été exclus, comme illustré dans les trois graphiques suivants :

La distribution complète des achats (voir Figure 1) représente tous les utilisateurs en fonction du nombre de jeux achetés, montrant la forte concentration sur les petites valeurs.

La distribution limitée à 60 jeux (voir Figure 2) affine la visualisation pour les valeurs plus faibles, là où se trouve la majorité des utilisateurs.

La distribution des utilisateurs ayant $\bar{4}$ achats (voir Figure 3) présente le dataset final filtré, où la décroissance exponentielle reste marquée mais plus homogène pour la recommandation.

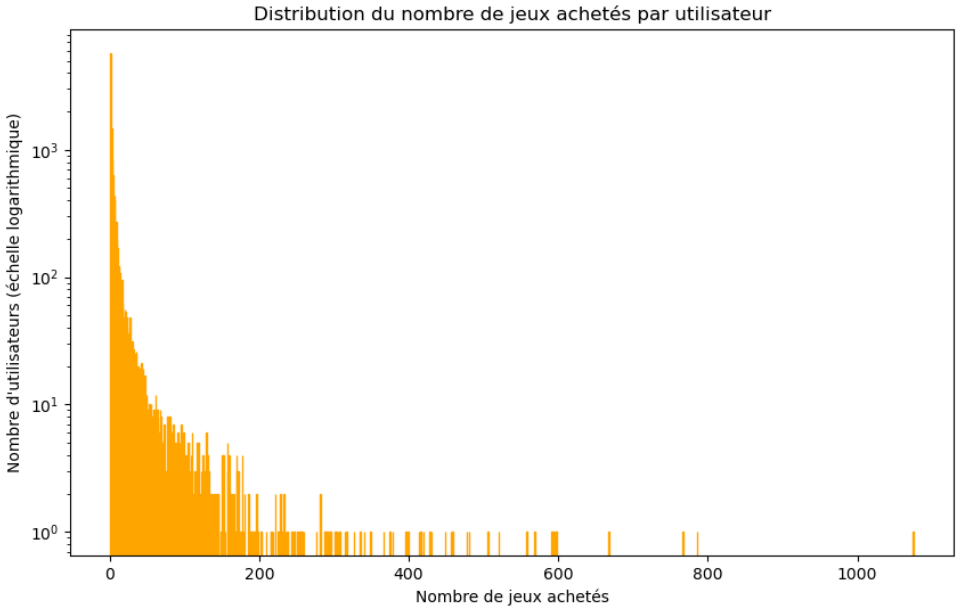


FIGURE 1 – Distribution du nombre de jeux achetés par utilisateur

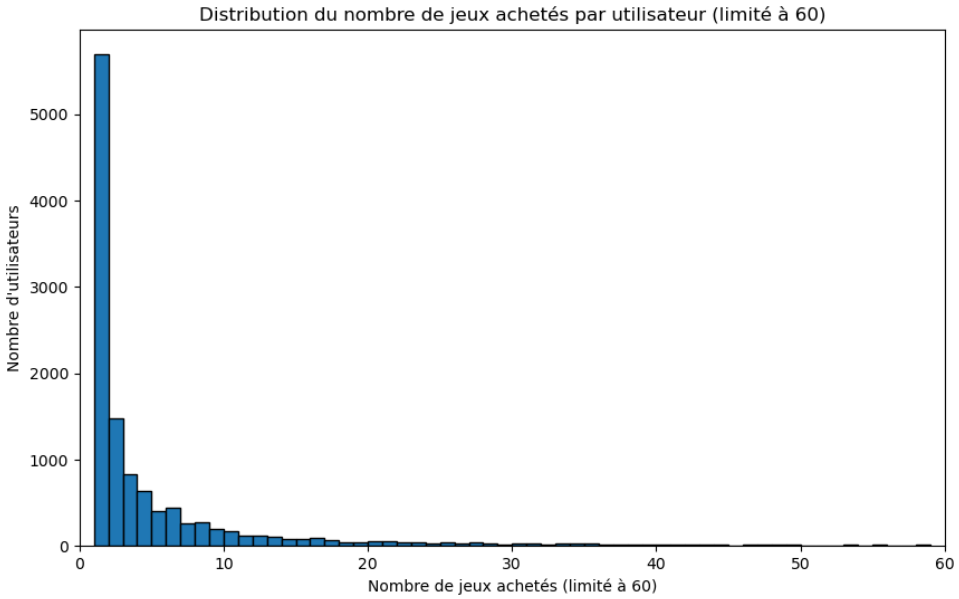


FIGURE 2 – Distribution du nombre de jeux achetés par utilisateur (limité à 60)

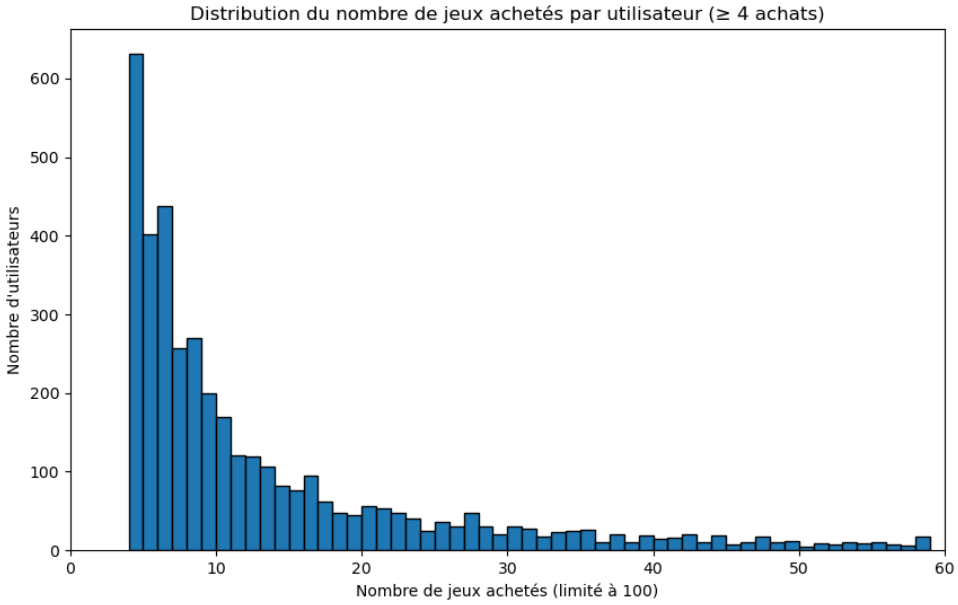


FIGURE 3 – Distribution du nombre de jeux achetés par utilisateur (>= 4 achats)

En parallèle, nous avons enrichi les informations des jeux en y intégrant les tags communautaires issus de tags_dataset. Un seuil de 10 % des votes totaux pour chaque jeu a été appliqué : seuls les tags atteignant ce seuil de popularité ont été retenus. Cette sélection privilégie les tags réellement associés au jeu, réduisant ainsi le bruit que pourraient introduire des tags votés trop rarement par les utilisateurs. Ce choix garantit une meilleure précision des informations descriptives, bien qu'il puisse parfois réduire la variété des tags pour certains jeux dont les tags ont été voté de manière très éparse.

À l'issue de cette phase de prétraitement, nous disposons de 3708 utilisateurs ayant une médiane de 9 jeux achetés chacun. Le dataset final rassemble des informations complètes pour chaque jeu, comprenant les tags communautaires et ceux fournis par la plateforme Steam, offrant ainsi une base solide pour la création et l'évaluation des modèles de recommandation.

Nous disposons des interactions d'utilisateurs avec 3205 jeux uniques. Cependant, l'ensemble des informations disponibles sur les jeux comprend 27 033 titres au total. Cette distinction est essentielle, car les modèles de recommandation basés sur le contenu pourront exploiter les 27 033 jeux de la base complète, même si seuls 3205 jeux sont représentés dans les interactions utilisateurs. Cela permet d'étendre le champ des recommandations à des jeux potentiellement inconnus des utilisateurs, tout en s'appuyant sur les similarités de contenu.

## 3.3 Métrics de test

Pour évaluer les performances des modèles de recommandation, la métrique MAP (Mean Average Precision) a été utilisée. MAP calcule la moyenne des précisions obtenues pour chaque utilisateur à des positions de classement successives, en attribuant un score plus élevé lorsque les items effectivement achetés par l'utilisateur apparaissent en haut des recommandations.

La MAP est particulièrement pertinente dans ce contexte, car elle reflète l'efficacité du modèle pour recommander les jeux qu'un utilisateur serait susceptible d'acheter en priorité.

## 3.4 Séparation entraînement, test

Une technique de séparation des données a été spécialement customisé pour ce projet. Pour chaque utilisateur, 75 % des jeux achetés ont été assignés au jeu d'entraînement. Ce dataset d'entraînement est utilisé par les modèles pour estimer les recommandations.

Ensuite, les 25 % restants des jeux de chaque utilisateur sont réservés au jeu de test. Ce jeu de test permet d'évaluer la pertinence des recommandations en calculant la métrique MAP. En effectuant cette évaluation sur les jeux exclus du jeu d'entraînement, il est possible de mesurer la capacité du modèle à recommander des jeux non observés par l'utilisateur dans le jeu d'entraînement, assurant ainsi une évaluation réaliste de ses performances.

Cela implique un jeu de données contenant 59758 interactions d'entraînement, un jeu de 18186 interactions de test et donc un ratio de 30% de test sur l'ensemble des données.

## 3.5 Méthodes de Recommandation

Pour les tests de modèles de recommandation, plusieurs techniques de collaborative filtering ont été explorées : itemKNN, userKNN, SVD, et BPR. Ces algorithmes offrent une diversité de méthodes pour exploiter les données d'interactions, et permettent de comparer leur efficacité dans notre contexte. En complément, une approche content-based a également été mise en place via TF-IDF.

itemKNN : Ce modèle repose sur les similarités entre jeux pour générer des recommandations. Il recommande des jeux qui sont similaires à ceux que l'utilisateur a déjà achetés, en se basant sur les interactions d'achat de jeu entre utilisateurs.

userKNN : Ce modèle identifie les jeux qu'aiment les utilisateurs proches du profil de l'utilisateur cible. Avec un petit nombre de voisins (k faible), userKNN fournit des recommandations personnalisées, mais si les performances sont insuffisantes, on peut aussi augmenter le nombre de voisins pour tendre vers des recommandations basées sur les jeux les plus populaires de la base. En effet, en augmentant le k de manière importante, userKNN va de plus en plus recommander les jeux populaires, ce qui donne un modèle de type "popularity-based" avec un léger biais personnalisé. Étant donné la nature de nos données, ce modèle servira de baseline, bien qu'on ne s'attende pas à des performances exceptionnelles.

SVD : La factorisation en valeurs singulières (SVD) décompose la matrice utilisateur-item en facteurs latents, identifiant des similarités implicites non observables dans les interactions. SVD est souvent efficace dans le filtrage collaboratif classique, mais dans notre cas, les données sont limitées en quantité d'interactions par utilisateur, ce qui pourrait réduire son efficacité.

BPR (Bayesian Personalized Ranking) : BPR est une technique de pairwise learning, adaptée aux systèmes de recommandation avec des interactions uniquement positives, comme les achats de jeux dans notre dataset. Contrairement aux modèles précédents, BPR maximise la probabilité que les jeux achetés par l'utilisateur soient classés avant ceux non explorés. Cette méthode est particulièrement adaptée dans notre contexte, car elle exploite le fait que chaque interaction entre un utilisateur et un jeu reflète une préférence implicite.

TF-IDF (content-based) : Cette approche content-based a également été testée pour tirer parti des 27 033 jeux disponibles dans item_dataset, même si tous ne sont pas associés à des interactions utilisateurs. TF-IDF (Term Frequency-Inverse Document Frequency) attribue un poids aux tags en fonction de leur fréquence dans un jeu donné et de leur rareté dans l'ensemble des jeux. En utilisant la matrice de similarité générée par TF-IDF, on peut également ajouter facilement de la diversité aux recommandations en exploitant les similarités de contenu entre jeux, cela sera utilisé dans la partie diversification sur les modèles performants.

TF-IDF permet de recommander des jeux peu connus ou inexplorés par les utilisateurs tout en restant pertinent, étendant ainsi le champ des recommandations possibles.

Dans l'ensemble, bien que les modèles userKNN, itemKNN, et SVD puissent ne pas donner des performances exceptionnelles en raison de la nature de nos données (faible nombre d'interactions par utilisateur et forte disparité dans les achats), ils fournissent des bases utiles pour la comparaison. L'approche BPR, quant à elle, est bien adaptée à nos données d'interactions positives, et TF-IDF permet d'enrichir les recommandations en ajoutant de la diversité et en exploitant le catalogue complet de jeux.

# 4 Analyse

## 4.1 Configurations et Hyperparamètres des Modèles Testés

Les modèles de recommandation ont été évalués avec les configurations et hyperparamètres suivants :

- itemKNN : similarité cosinus, avec des valeurs de voisinage k de
  [5, 20, 60, 100, 300]
- userKNN : similarité cosinus, avec des valeurs de voisinage k de
  [5, 20, 40, 50, 60, 70, 80, 300, 500]
- SVD : avec des valeurs de dimension latente de
  [3, 6, 9, 14, 20, 27, 40, 100]
  et 10 itérations pour chaque configuration, suivies d'une moyenne des résultats
- BPR : avec des valeurs de dimension latente de
  [3, 6, 9, 14, 20, 27, 40, 100],
  un maximum de 100 itérations, un taux d'apprentissage de 0.01, et une moyenne des résultats sur 10 itérations pour chaque configuration

- TF-IDF (content-based) :

  - Représentation par moyenne des vecteurs des jeux d'entraînement : cette méthode calcule la moyenne des vecteurs TF-IDF des jeux possédés par l'utilisateur pour obtenir une représentation globale de ses préférences, en tenant compte de l'ensemble des attributs de chaque jeu. Elle favorise les recommandations qui correspondent globalement aux jeux déjà achetés.

  - Représentation par maximum des vecteurs des jeux d'entraînement : cette approche prend pour chaque dimension le maximum des valeurs TF-IDF parmi les jeux de l'utilisateur. Cela met en avant les caractéristiques les plus marquées dans le profil de l'utilisateur, accentuant les recommandations qui possèdent des attributs forts similaires aux jeux déjà acquis.

  - Représentation par agrégation des vecteurs des jeux d'entraînement : cette technique génère les recommandations séparément pour chaque jeu du jeu d'entraînement de l'utilisateur. Ensuite, les recommandations finales sont déterminées en sélectionnant les jeux qui apparaissent le plus souvent dans les différentes listes. Cette approche favorise les jeux les plus recommandés dans l'ensemble des profils associés à chaque jeu d'entraînement, renforçant les choix qui sont pertinents pour plusieurs préférences individuelles.

Pour les méthodes de filtrage collaboratif (itemKNN, userKNN, SVD, BPR), le MAP est calculé sur un classement de tous les jeux présents dans les interactions utilisateurs. Pour le modèle TF-IDF, en revanche, le MAP est basé sur les 100 meilleures recommandations sélectionnées dans l'ensemble du catalogue global, ce qui inclut les jeux sans interactions utilisateurs. Cette approche rend l'évaluation plus stricte pour TF-IDF, et son MAP est donc sous-évalué par rapport aux autres modèles. Cependant, un score élevé pour TF-IDF indiquerait sa capacité à identifier des recommandations pertinentes parmi un choix beaucoup plus vaste, démontrant une bonne généralisation.

## 4.2 Résultats des Modèles Testés

Les résultats des différents modèles de recommandation sont résumés dans les tableaux ci-dessous. Ces résultats montrent des performances variées selon les méthodes et les paramètres utilisés, ce qui permet de tirer quelques conclusions sur les configurations les plus adaptées au contexte de recommandation de jeux.

| **Configuration** | **MAP** |
|---|---|
| ItemKNN - k=5 | 0.3% |
| ItemKNN - k=20 | 0.7% |
| ItemKNN - k=60 | 1.4% |
| ItemKNN - k=100 | 1.6% |
| ItemKNN - k=300 | 1.8% |

Table 1 – Résultats pour ItemKNN

Le modèle ItemKNN affiche des performances limitées, avec un MAP qui plafonne à 1.8%. La similarité entre jeux ne semble pas apporter des recommandations efficaces dans ce contexte. Cela pourrait être lié au fait que les jeux sont trop variés pour obtenir une similarité pertinente dans les interactions utilisateur-jeu. En conséquence, ItemKNN ne sera pas retenu pour la suite des analyses.

| **Configuration** | **MAP** |
|---|---|
| UserKNN - k=5 | 0.4% |
| UserKNN - k=20 | 1.0% |
| UserKNN - k=40 | 1.8% |
| UserKNN - k=50 | 2.2% |
| UserKNN - k=60 | 2.6% |
| UserKNN - k=70 | 2.9% |
| UserKNN - k=80 | 3.3% |
| UserKNN - k=300 | 15.9% |
| UserKNN - k=380 | 21.9% |
| UserKNN - k=500 | 26.4% |
| UserKNN - k=580 | 27.6% |
| UserKNN - k=600 | 27.8% |
| UserKNN - k=640 | 28.1% |
| UserKNN - k=700 | 28.2% |

Table 2 – Résultats pour UserKNN

Le modèle UserKNN montre une tendance croissante avec l'augmentation de k, le MAP atteignant un pic de 28.2% pour des valeurs très élevées. En effet, avec un grand nombre de voisins, UserKNN se rapproche de recommandations basées sur la popularité, ce qui s'avère particulièrement efficace dans ce cas. Pour maintenir une personnalisation suffisante sans tomber dans une approche purement basée sur les jeux populaires, une valeur de k = 300 sera retenue pour la suite.

| Configuration | MAP |
|---|---|
| SVD - k=3 | 0.2% |
| SVD - k=6 | 0.2% |
| SVD - k=9 | 0.2% |
| SVD - k=14 | 0.2% |
| SVD - k=20 | 0.2% |
| SVD - k=27 | 0.2% |
| SVD - k=40 | 0.2% |
| SVD - k=100 | 0.2% |

Table 3 – Résultats pour SVD

Les résultats du modèle SVD sont très faibles, le MAP restant à 0.2% quelle que soit la valeur de k. Cette stagnation pourrait s'expliquer par le fait que le modèle SVD repose sur la présence de notes de préférence (ratings), qui sont absentes dans notre dataset. Ainsi, SVD ne sera pas retenu pour la suite des tests.

| Configuration | MAP |
|---|---|
| BPR - k=3 | 15.4% |
| BPR - k=6 | 15.6% |
| BPR - k=9 | 15.4% |
| BPR - k=14 | 15.4% |
| BPR - k=20 | 15.4% |
| BPR - k=27 | 15.2% |
| BPR - k=40 | 15.1% |
| BPR - k=100 | 14.7% |

Table 4 – Résultats pour BPR

Le modèle BPR présente des performances relativement stables malgré les variations de facteurs latents, le MAP fluctuant autour de 15%. Étant donné que l'optimisation pour le ranking semble peu influencée par la complexité du modèle dans ce contexte, une configuration avec 6 dimensions latentes sera retenue pour la suite.

| Configuration TF-IDF | MAP |
|---|---|
| Vecteur moyen des jeux d'entraînement | 12.77% |
| Vecteur maximum des jeux d'entraînement | 10.05% |
| Agrégation pondérée des jeux d'entraînement | 10.11% |

Table 5 – Résultats pour TF-IDF

Les scores MAP pour les différentes configurations TF-IDF sont relativement faibles, mais ils restent significatifs compte tenu de la difficulté de la tâche. Contrairement aux modèles de filtrage collaboratif, TF-IDF est évalué sur tout le catalogue de jeux, incluant ceux sans interactions. Ce cadre d'évaluation élargi rend la tâche plus complexe, mais TF-IDF parvient tout de même à obtenir un MAP de 12.77% en moyenne des vecteurs d'entraînement.

Les modèles sélectionnés pour la suite sont donc :

— UserKNN avec k = 300 pour bénéficier d'une personnalisation modérée tout en exploitant les jeux populaires
— BPR avec une dimension latente de 6
— TF-IDF avec la moyenne des vecteurs des jeux d'entraînement

## 4.3 Recommandations des Modèles pour un Échantillon d'Utilisateurs

Dans cette section, nous présentons les recommandations des trois modèles retenus pour un échantillon d'utilisateurs aux profils divers.

Les tests ont été réalisés sur six utilisateurs : deux utilisateurs ayant 4 achats, deux ayant 10 achats, et deux ayant 20 achats.

Pour chacun de ces utilisateurs, ont affiche les jeux du jeu d'entraînement (jeux connus par le modèle) et ceux du jeu de test (jeux à recommander).

Pour chaque modèle, la position de chaque jeu du jeu de test dans les recommandations est indiquée, ainsi que les 10 jeux principaux recommandés par le modèle pour cet utilisateur.

Par souci de concision, seuls les résultats d'un utilisateur ayant 10 achats sont présentés ici. Les résultats complets pour tous les utilisateurs sont disponibles dans le notebook associé.

— **Utilisateur 24858005 :**
  — Jeux du jeu d'entraînement :
  Alpha Prime, Call of Duty® : Modern Warfare® 2, Counter-Strike : Source, Half-Life 2, Half-Life 2 : Deathmatch, Half-Life 2 : Lost Coast, Half-Life Deathmatch : Source, Left 4 Dead
  — Jeux du jeu de test :
  Call of Duty® : Black Ops, Half-Life : Source

**Résultats des Modèles pour l'Utilisateur 82174211 :**

**Pour le modèle UserKNN :**

— Position des jeux de test dans les recommandations :
  — **PAYDAY™ The Heist** : 46
  — **The Binding of Isaac** : 72
  — **Dota 2** : 20
  — **Serious Sam 3 : BFE** : 137
  — **Survarium** : 82
— Top 10 des jeux recommandés :
  1. Counter-Strike : Condition Zero
  2. Heroes & Generals
  3. Portal 2
  4. Portal
  5. Left 4 Dead 2
  6. Warframe
  7. War Thunder
  8. PAYDAY 2

9. Garry's Mod
10. Robocraft

**Pour le modèle BPR :**

— Position des jeux de test dans les recommandations :
PAYDAY™ The Heist : 45
The Binding of Isaac : 75
Dota 2 : 3
Serious Sam 3 : BFE : 152
Survarium : 109

— Top 10 des jeux recommandés :

1. Dota 2
2. Half-Life 2 : Lost Coast
3. Half-Life 2 : Deathmatch
4. Left 4 Dead 2
5. Half-Life 2
6. Counter-Strike
7. Garry's Mod
8. The Elder Scrolls V : Skyrim
9. Portal 2
10. Portal

**Pour le modèle TF-IDF :**

— Position des jeux de test dans les recommandations :
PAYDAY™ The Heist : 2345
The Binding of Isaac : Non recommandé dans le top 3000
Dota 2 : 9
Serious Sam 3 : BFE : 552
Survarium : 58

— Top 10 des jeux recommandés :

1. Alien Swarm
2. The Lab
3. Counter-Strike
4. Counter-Strike : Condition Zero
5. Team Fortress Classic
6. Half-Life 2 : Deathmatch
7. Ricochet
8. OrcCraft
9. Dota 2
10. Dungeon Defenders II

# 5 Diversification des recommandations

## 5.1 Méthode de Diversification : Maximal Marginal Relevance (MMR)

Pour enrichir les recommandations et éviter la redondance, une diversification par Maximal Marginal Relevance (MMR) est appliquée à chaque modèle (UserKNN, BPR, et TF-IDF), en se basant sur les tags TF-IDF des jeux pour mesurer la diversité.

Fonctionnement de l'algorithme MMR :

— **Pertinence et diversité**. Chaque jeu candidat est évalué selon un score MMR :

$$\text{Score MMR} = \lambda \times \text{Pertinence} - (1 - \lambda) \times \text{Diversité}$$

La pertinence est obtenue via les scores de recommandation du modèle. La diversité est calculée par la similarité cosinus entre les représentations TF-IDF des jeux candidats et ceux déjà sélectionnés, sur la base des tags.

— **Processus itératif de sélection**. À chaque itération, le jeu avec le score MMR le plus élevé est ajouté aux recommandations et retiré des candidats. Ce processus se répète jusqu'à atteindre le nombre de recommandations diversifiées souhaité (top_k).

Cette approche garantit que les recommandations fournies par UserKNN, BPR et TF-IDF sont variées, sans redondance dans les thèmes et genres, grâce à l'usage des tags TF-IDF pour quantifier la diversité.

## 5.2 Diversification des Recommandations appliqué

Dans cette section, nous présentons l'effet de la diversification sur les recommandations, en comparant les recommandations initiales des modèles avec leurs versions diversifiées pour un utilisateur à 10 achats. Les résultats complets pour d'autres utilisateurs (2 utilisateurs avec 4 achats, 2 avec 10 achats et 2 avec 20 achats) sont disponibles dans le notebook associé.

Pour chaque modèle, une fonction de diversification MMR (Maximal Marginal Relevance) a été utilisée avec des valeurs de $\lambda$ adaptées. Pour les modèles UserKNN et BPR, $\lambda$ est fixé à 0.01 pour minimiser la perte de pertinence sur les premiers résultats, qui affichent une forte différence de scores entre les premières positions. Pour TF-IDF, $\lambda$ est de 0.005, car ce modèle produit des recommandations plus similaires entre elles, nécessitant une pondération plus élevée pour favoriser la diversité.

Dans cette présentation, nous affichons également les jeux du jeu d'entraînement pour chaque utilisateur afin de mettre en évidence la proximité ou non de ces recommandations avec les jeux connues et l'impact de la diversification.

**Jeux du jeu d'entraînement** :
Left 4 Dead, Call of Duty® : Modern Warfare® 2, Alpha Prime, Counter-Strike : Source, Half-Life 2, Half-Life 2 : Deathmatch, Half-Life 2 : Lost Coast, Half-Life Deathmatch : Source

**Pour le modèle UserKNN** :

— **Top 10 des jeux recommandés (avant diversification)** :

1. Team Fortress 2
2. Counter-Strike : Global Offensive
3. Left 4 Dead 2
4. Portal
5. Half-Life 2 : Episode One
6. Counter-Strike
7. Day of Defeat : Source
8. Half-Life 2 : Episode Two
9. Counter-Strike : Condition Zero
10. Dota 2

— **Top 10 des jeux diversifiés** :

1. Team Fortress 2
2. The Elder Scrolls V : Skyrim
3. Sid Meier's Civilization® V
4. Terraria
5. Arma 2
6. PAYDAY 2
7. Portal
8. Chivalry : Medieval Warfare
9. The Witcher 2 : Assassins of Kings Enhanced Edition

10. Garry's Mod

**Pour le modèle BPR** :
— **Top 10 des jeux recommandés (avant diversification)** :
1. Team Fortress 2
2. Dota 2
3. Counter-Strike : Global Offensive
4. Left 4 Dead 2
5. Unturned
6. Counter-Strike
7. Garry's Mod
8. The Elder Scrolls V : Skyrim
9. Portal
10. Portal 2

— **Top 10 des jeux diversifiés** :
1. Team Fortress 2
2. The Elder Scrolls V : Skyrim
3. Sid Meier's Civilization® V
4. Terraria
5. Unturned
6. PAYDAY 2
7. Arma 2
8. Portal
9. Garry's Mod
10. Warframe

**Pour le modèle TF-IDF** :
— **Top 10 des jeux recommandés (avant diversification)** :
1. Counter-Strike
2. Ricochet
3. Counter-Strike : Condition Zero
4. Team Fortress Classic
5. Half-Life 2 : Episode One
6. Half-Life 2 : Episode Two
7. Half-Life : Source
8. Deathmatch Classic
9. Half-Life
10. Team Fortress 2

— **Top 10 des jeux diversifiés** :

1. Counter-Strike
2. Unreal 2 : The Awakening
3. Call of Duty® 4 : Modern Warfare®
4. Monument
5. AXYOS
6. Earthfall
7. Alien Swarm
8. FARHOME
9. Call of Duty : Black Ops - Mac Edition
10. Portal

# 6 Conclusion

En conclusion, le projet a mis en lumière plusieurs limitations liées au jeu de données et aux défis inhérents aux systèmes de recommandation. La base de données utilisée contenait de nombreux utilisateurs ayant acheté très peu de jeux, rendant difficile l'évaluation précise des modèles. En conséquence, les scores de Mean Average Precision (MAP) observés, situés entre 10 et 25%, sont relativement faibles. Sans grande expérience préalable en systèmes de recommandation, il est difficile de conclure s'ils sont acceptables dans ce contexte, mais ils restent inférieurs aux attentes. Les modèles obtenus remplissent différents rôles. Le modèle UserKNN, bien qu'offrant une personnalisation modérée, se repose au final en grande partie sur la popularité des jeux, ce qui peut être un atout pour des recommandations généralistes mais limite la profondeur de personnalisation. Le modèle BPR, quant à lui, se révèle intéressant pour exploiter au mieux les interactions positives entre utilisateurs et jeux, tout en restant robuste aux données binaires (achat ou non). Enfin, le modèle TF-IDF ajoute de la diversité en exploitant les similarités de contenu et, combiné à une diversification MMR, devient un outil puissant pour apporter une variété de recommandations.

Les scores de MAP obtenus sont possiblement influencés par la nature complexe et imprévisible des comportements d'achat de jeux vidéo. Par ailleurs, l'approche utilisée ici a été exigeante, avec un ratio de 30% de jeux en test, rendant l'évaluation des modèles plus sévère. Une approche future consisterait à essayer de fusionner ces modèles pour optimiser le compromis entre pertinence des recommandations et diversité. Il aurait également été pertinent d'évaluer TF-IDF sur les seuls jeux avec interactions pour obtenir un MAP plus significatif et ainsi comparer de manière plus équitable les modèles.

L'utilisation de Cornac a présenté certains défis, car cette bibliothèque est avant tout orientée vers des approches collaboratives, avec peu de solutions prêtes pour le content-based sans passer par des réseaux de neurones. Ce manque de flexibilité a nécessité un effort de personnalisation, notamment dans la création d'un module de séparation des données d'entraînement et de test plus modulaire et explicite.

Pour un utilisateur n'ayant que peu d'interactions (problème de départ à froid), UserKNN pourra offrir des recommandations basées sur les jeux les plus populaires, tandis que TF-IDF peut diriger rapidement l'utilisateur vers des jeux similaires dès son premier achat. Cependant, les modèles actuellement implémentés restent peu performants pour le problème de départ à froid, soulignant la nécessité d'outils plus adaptés pour des utilisateurs sans historique d'achat.

# Références


[1] Steamworks Documentation, *Tags and Store Visibility on Steam*
https://partner.steamgames.com/doc/store/tags?l=french

[2] Gamepressure, *Full Version of Steamś Interactive Recommender Goes Live*
https://www.gamepressure.com/newsroom/full-version-of-steams-interactive-advisor-goes-live/z6199c

[3] Steam Community, *Introducing the Interactive Recommender*
https://steamcommunity.com/games/593110/announcements/detail/1612767708821405787

[4] Christine Brownell, *Who is Playing Games on Steam ?*, Medium, 2018.
https://cbrownell.medium.com/who-is-playing-games-on-steam-e765142cd983

[5] Tamber. *Steam Video Games Dataset*. 2021. Disponible sur Kaggle : https://www.kaggle.com/datasets/tamber/steam-video-games.

[6] Nik Davis. *Steam Store Games (Clean dataset)*. 2021. Disponible sur Kaggle : https://www.kaggle.com/datasets/nikdavis/steam-store-games?select=steam.csv.

[7] Nik Davis. *Steam Store Games - SteamSpy Tag Data*. 2021. Disponible sur Kaggle : https://www.kaggle.com/datasets/nikdavis/steam-store-games?select=steamspy_tag_data.csv.