Faculté des Sciences et Ingénierie - Sorbonne université

Master Informatique - parcours DAC

DALAS – Datascience, Learning et ApplicationS

Rapport de Projet

Exploration, Analyse et Recommandation de Jeux Vidéos

Réalisé par :

Raphaël Renard
Luc Salvon

Supervisé par :

Laure Soulier

Mai 2024

CONTENTS

CHAPTER 1

INTRODUCTION

Le secteur du jeu vidéo connaît une croissance exponentielle, tant en termes de popularité que de diversité. Avec des milliers de jeux disponibles sur différentes plateformes, il est essentiel pour les entreprises de comprendre les tendances du marché, les préférences des joueurs et les facteurs qui contribuent au succès d'un jeu. Dans ce contexte, notre projet vise à analyser un ensemble de données provenant de trois sources principales : HowLongToBeat (HLTB), Steam et Steam Charts.

HowLongToBeat[1] est une plateforme en ligne dont le but principal est de permettre aux joueurs de rentrer leur temps mis pour compléter des jeux vidéos. Cela sert notamment à renseigner les autres utilisateurs sur le temps estimé de jeu auquel s'attendre avant un potentiel achat. D'autres attributs intéressants des jeux sont présents sur cette plateforme.
Steam[2] est l'une des plus grandes plateformes de distribution de jeux vidéo sur PC, offrant des informations sur les jeux, les critiques des utilisateurs et les statistiques de jeu.
Steam Charts [3] est un site recensant le nombre de joueurs concurrents sur des jeux Steam.

Notre jeu de données combine ces trois sources d'informations pour fournir une vue d'ensemble des jeux vidéo, incluant des détails tels que les évaluations des utilisateurs, les genres, les plateformes prises en charge, les prix et bien plus encore.

En examinant ce jeu de données, nous cherchons à répondre à des questions clés telles que : Quels sont les genres de jeux les plus populaires ? Quels sont les caractéristiques qui font qu'un jeu est bien noté ? En comprenant ces aspects, nous espérons apporter des éclairages précieux aux développeurs de jeux, aux éditeurs et aux chercheurs intéressés par l'industrie du jeu vidéo. Puis, dans une seconde partie, nous créerons un système de recommandation de jeux pour les utilisateurs.

CHAPTER 2

PRÉSENTATION DES DONNÉES

2.1 Description des attributs collectés

Pour notre analyse nous avons utilisé des données provenant d'un scraping des plateformes Steam[2], HowLongToBeat[1] et SteamCharts[3].

Chaque ligne du fichier correspond à un jeu vidéo, avec diverses informations recueillies des trois sources. Les colonnes représentent les informations suivantes :

- `hltb_id` : Un identifiant unique pour chaque jeu sur HowLongToBeat.
- `title` : Le titre du jeu.
- `rating` : La note globale du jeu donnée par les utilisateurs de HowLongToBeat.
- `retirement` : Le taux de joueurs ayant abandonné le jeu.
- `platform` : La ou les plateformes sur lesquelles le jeu est disponible (PC, PS4, Xbox, etc).
- `genre` : Le genre du jeu (Action, Aventure, RPG, etc) d'après la plateforme HowLongToBeat.
- `date` : La date de sortie du jeu.
- `time` : Le temps de jeu moyen pour terminer le jeu.
- `price` : Le prix du jeu sur Steam.
- `memoire_vive` : La quantité de mémoire vive recommandée pour jouer au jeu.
- `espace_disque` : L'espace disque requis pour installer le jeu.
- `pourcentage_pos` : Le pourcentage de critiques positives sur Steam.
- `review_count` : Le nombre total de critiques sur Steam.
- `rating_value` : La note donnée au jeu par les utilisateurs de Steam.
- `description` : Une brève description du jeu.
- `twenty_four_hours` : Le nombre de joueurs ayant joué au jeu au cours des dernières 24 heures sur Steam.
- `all_time` : Le nombre de joueurs actifs depuis la sortie du jeu sur Steam.
- `steam_id` : Un identifiant unique pour chaque jeu sur Steam.
- `steam_tags` : Les tags associés au jeu sur Steam (Multijoueur, RPG, Indépendant, etc).

- `steam_genres` : Les genres associés au jeu sur Steam (similaires à la colonne `steam_genre` mais spécifiques à Steam).
- `players_by_time` : Le nombre de joueurs en fonction des mois.

2.2 Scrapping

Une particularité de la plateforme HowLongToBeat[1] est que les identifiants attribués à ses entrées sont séquentiels. Nous avons donc utilisé cette plateforme pour parcourir les différents jeux y étant recensés.

Les entrées HowLongToBeat concernant les jeux disponibles sur Steam comportent systématiquement un lien vers la page Steam du jeu. Nous pouvons alors nous servir de ce lien pour récupérer l'identifiant utilisé par Steam pour ce jeu, nous donnant accès à la fois à la page Steam de ce jeu ainsi que sa page SteamCharts, qui utilise les mêmes identifiants que Steam.

2.2.1 Récupération des identifiants valides

Une première étape pour la collecte est de trouver et filtrer les entrées de la plateforme HowLongToBeat nous intéressant, en effectuant un crawling. L'URL d'une telle entrée se présente sous la forme `https://howlongtobeat.com/game/<id_du_jeu>`, nous permettant donc de facilement accéder à la page d'un jeu pour un identifiant défini. Les identifiants étant séquentiels, nous pouvons alors créer un script opérant de la manière suivante :

- On crée un fichier `how-long-to-beat-ids.txt` qui contiendra les identifiants des jeux intéressants.
- On parcourt les identifiants séquentiellement (de 1 à 1000).

  Pour chacun de ces identifiants :

  - On récupère le contenu de la page associée via la bibliothèque urllib, que l'on traitera via la bibliothèque BeautifulSoup.
  - Si la page demandée n'existe pas, on continue avec le prochain identifiant.
  - Si la page existe, on essaie de récupérer un lien vers la page Steam du jeu. Si un tel lien existe, on note l'identifiant du jeu dans le fichier de sortie. Sinon, le jeu ne nous intéresse pas et on ne le retient alors pas.

Ce script nous permet donc de récupérer les identifiants utilisés par la plateforme HowLongToBeat pour les associer aux jeux disponibles sur Steam. On obtient 42116 identifiants différents, que l'on peut alors utiliser pour récupérer les données des jeux associés.

2.2.2 Récupération des données

Certaines données, notamment le prix du jeu, ne sont pas disponibles directement dans la source HTML des pages Steam, nécessitant alors l'utilisation d'une bibliothèque prenant en charge JavaScript. Nous utilisons pour cette tâche la bibliothèque Selenium.

Pour chaque information souhaitée, on identifie son emplacement dans la page HTML du site associé. On peut alors créer un script qui parcourt les identifiants HowLongToBeat précédemment crawlés, récupère tous les attributs souhaités, et les enregistre dans un fichier CSV, appelé `game_data.csv`.

Selenium n'étant pas exécutable sur la PPTI, nous avons dû effectuer ce scrapping sur nos machines personnelles. Le temps de scrapping étant conséquent (quelques secondes par entrée), nous n'avons pas pu traiter tous les identifiants de la liste. Nous avons alors après cette étape pu retenir 25259 entrées sur les 42116 identifiants pertinents.

2.3 Nettoyage des données

Les données récupérées nécessitent un traitement de nettoyage, que ce soit pour harmoniser les valeurs, ou pour filtrer les entrées possédant des valeurs manquantes (par exemple un jeu dont la page steam n'existe pas, qui n'a donc aucune données pour les informations tirées de Steam). Afin de ne pas modifier les données initiales pour pouvoir réviser notre stratégie si besoin, les données après traitement seront stockées dans un deuxième fichier CSV `cleaned_data.csv`.

Les traitements suivants ont été effectués :

- Prix : Les prix sont normalisés en valeurs numériques et convertis en dollars.
- Description : Les descriptions sont nettoyées en supprimant les premiers mots inutiles (comme "jeu", "contenu", etc.) pour ne garder que la partie descriptive pertinente.
- Conversion en nombres : Certaines colonnes contenant des valeurs numériques qui ont été récupérées sous forme de chaînes de caractères sont converties en nombres.
- Listes : Les colonnes contenant les listes `genre`, `steam_genres` et `steam_tags` sont fusionnées, triées et nettoyées pour éliminer les doublons et les caractères spéciaux.
- Suppression de colonnes : Les colonnes `espace_disque` et `memoire_vive` sont supprimées car elles contiennent trop de valeurs manquantes.
- Suppression des valeurs manquantes : Les lignes contenant des valeurs manquantes sont supprimées du jeu de données.
- Suppression des doublons : Les doublons dans le jeu de données sont supprimés pour éviter toute redondance dans les analyses ultérieures.

Ainsi, après nettoyage, il reste les colonnes `hltb_id`, `title`, `rating`, `retirement`, `platform`, `date`, `time`, `price`, `pourcentage_pos`, `review_count`, `rating_value`, `description`, `twenty_four_hours`, `all_time`, `steam_id`, `players_by_time`, `genres`. Sur les 25259 entrées scrappées, il nous reste 2991 entrées après traitement.

CHAPTER 3

VISUALISATION ET ANALYSE EXPLORATOIRE

Pour bien comprendre nos données et leurs tendances sous-jacentes, nous utilisons plusieurs techniques de visualisation et d'analyse exploratoire.

3.1 Données numériques

Les données numériques dont nous disposons sont les suivantes : `rating`, `retirement`, `time`, `price`, `review_count`, `rating_value`, `twenty_four_hours` et `all_time`.

Avant toute visualisation sur ces données, nous isolons les outliers avec une `IsolationForest()`, et les enlevons du jeu de données. Nous normalisons aussi nos données.

En faisant un premier pairplot pour avoir une idée globale de la répartition des données, nous obtenons la figure 3.1. Pour une meilleure lisibilité, nous calculons aussi une matrice de corrélation (figure 3.2). On observe que les deux ratings, celui de Steam et celui de HLTB, sont corrélés, ce qui indique que les notes ne dépendent pas de la plateforme. Nous pouvons aussi noter que les colonnes exprimant un nombre de joueurs (`review_count`, `twenty_four_hours`, `all_time`) sont aussi, logiquement, corrélées entre elles.

Pour avoir des descriptions plus précises des données, nous avons affiché un scatterplot des notes des jeux en fonction de leur prix (figure 3.3). On peut observer que les notes les plus basses sont attribuées aux jeux les moins chers, et que plus le prix augmente, moins il y a de mauvaises notes. Nous pouvons aussi noter un biais dans le données, du fait qu'il y a plus de notes positives que négatives, bias qui se voit clairement sur la figure 3.4 : l'écrasante majorité des jeux ont une note entre 60 et 80 sur 100.

D'autre part, les figures 3.5 et 3.6 montrent que ces dernières années, la moyenne des notes des jeux a fortement augmenté, tout comme le prix.

3.2 Données textuelles

La seule donnée textuelle que nous avons est la description des jeux. Le défaut de cette donnée est que lorsque nous l'avons scrapée, certaines descriptions ont été récupérées en français et d'autres en anglais. La première chose à faire est donc de séparer les jeux en deux : ceux avec une description en français et ceux avec une description en anglais. Pour se faire, nous utilisons le module `langdetect` de python.

Une fois cette séparation faite, nous pouvons faire de l'analyse de sentiment sur les descriptions anglaise (le faire à côté sur les descriptions françaises serait redondant) pour déterminer si elles sont positives ou négatives. Nous obtenons la figure 3.7. On observe que les descriptions sont plutôt neutres, légèrement positives. On pourrait s'attendre à des descriptions en moyenne assez positives pour donner aux gens envie de jouer au jeu, mais ce n'est pas le cas. On peut donc supposer que les sentiments exprimés dans les descriptions sont des indicateurs

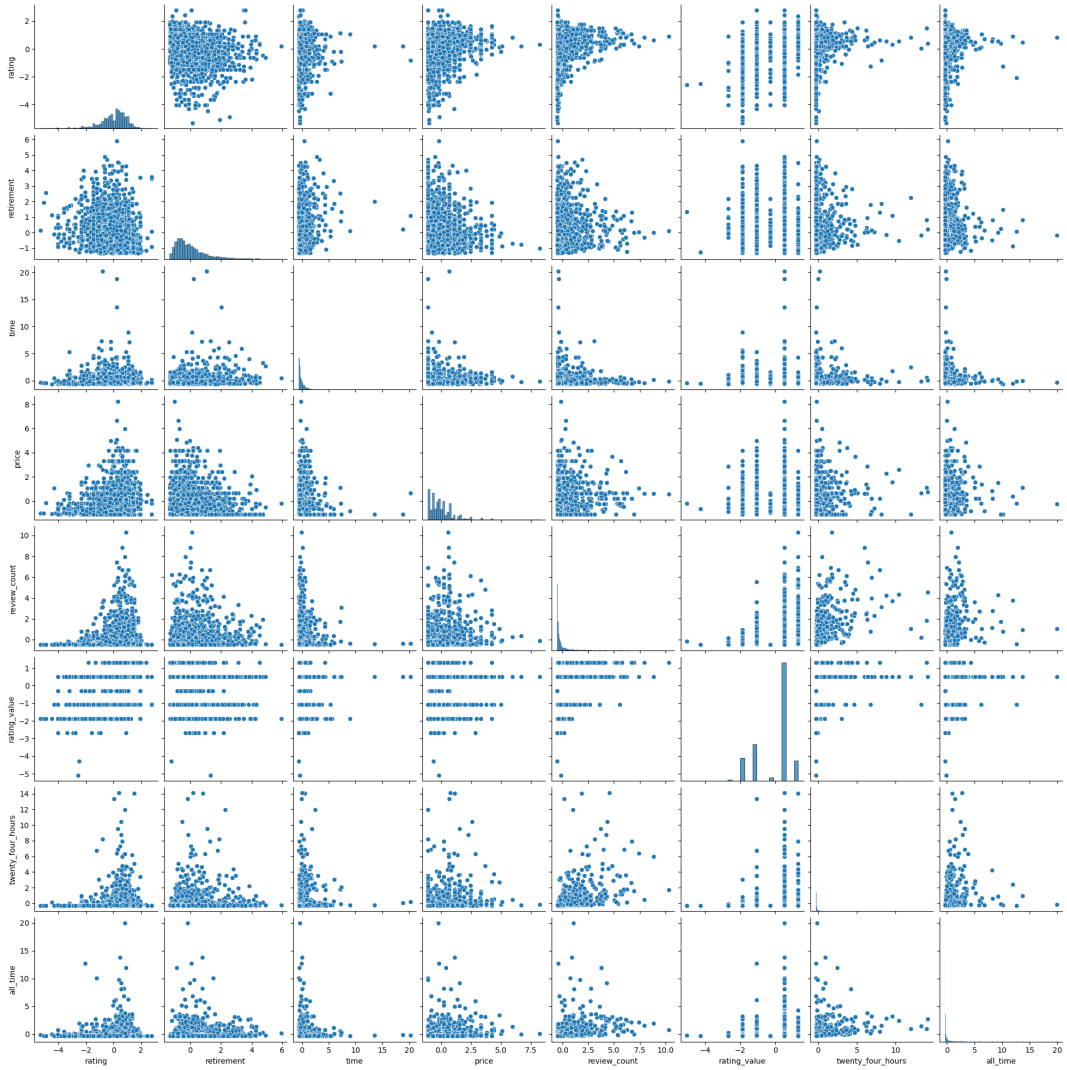

Figure 3.1: Pairplot des données numériques du dataset

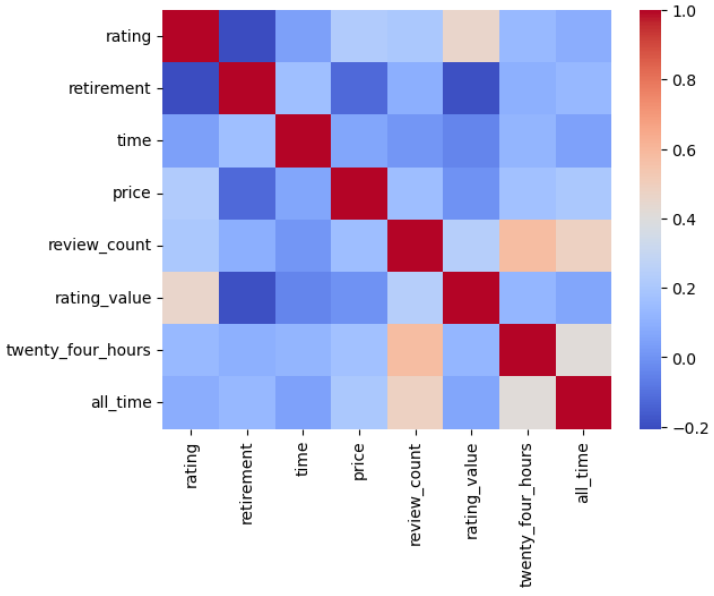

Figure 3.2: Matrice de corrélation entre les données numériques

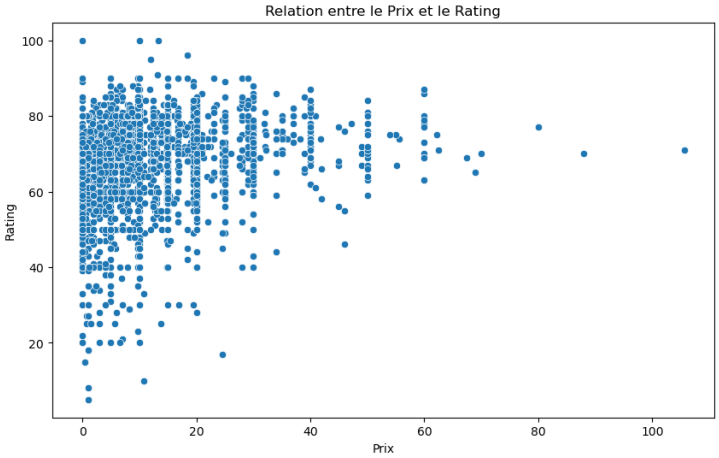

Figure 3.3: Notes des jeux en fonction de leur prix

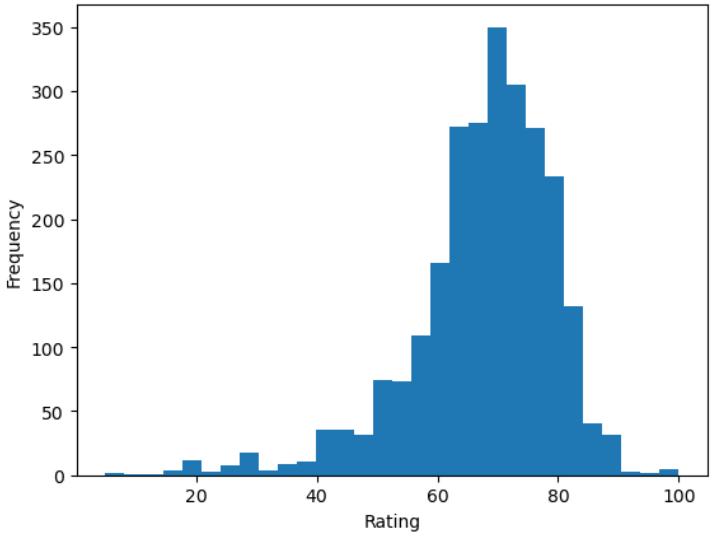

Figure 3.4: Répartition des notes des jeux

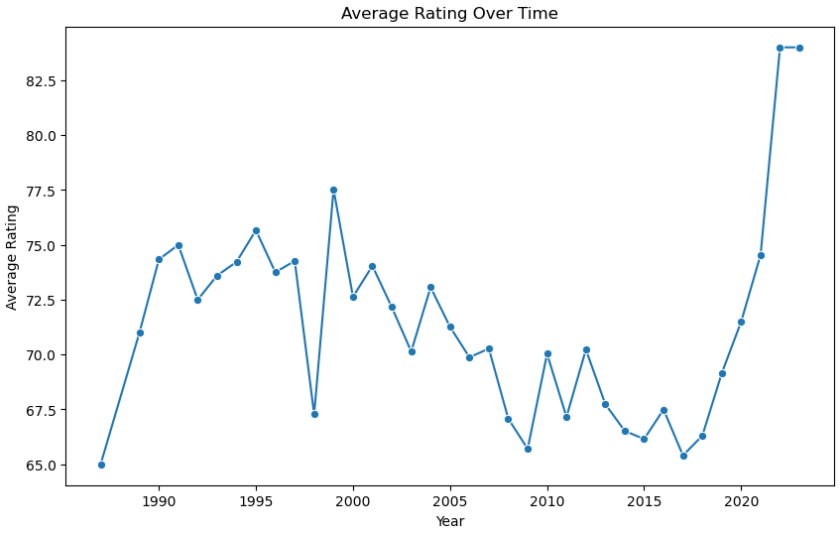

Figure 3.5: Moyenne des notes des jeux en fonction des années

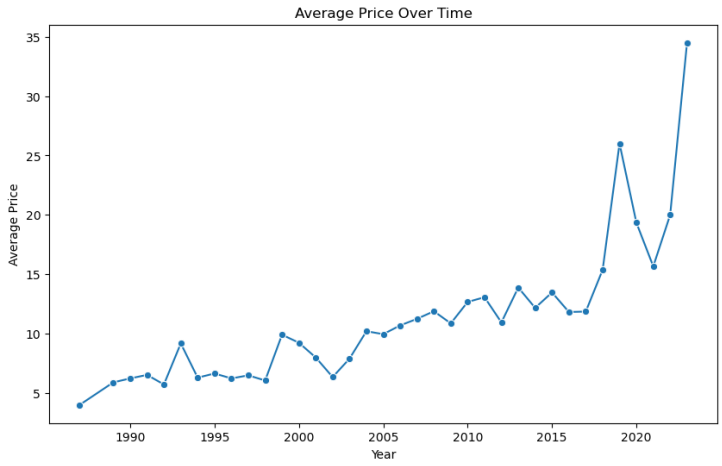

Figure 3.6: Moyenne des prix des jeux en fonction des années

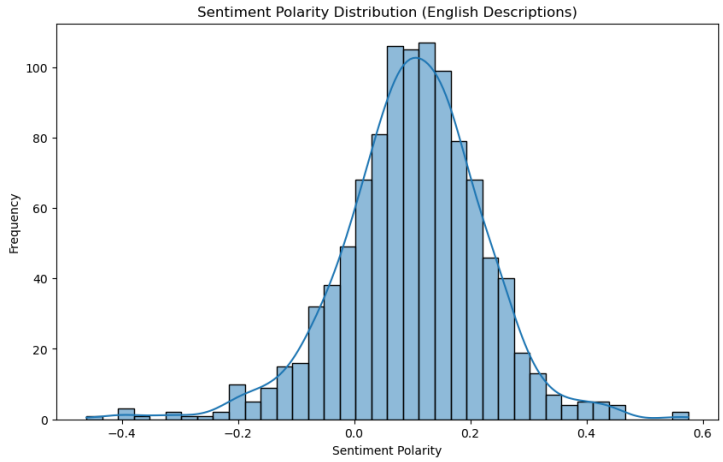

Figure 3.7: Polarité des sentiments dans les descriptions des jeux

relativement réalistes de l'ambiance du jeu.

Nous en profitons pour regarder si la positivité de la description (et donc possiblement du jeu) influe sur les notes. La figure 3.8 nous montre que ce n'est pas le cas.

Afin de mieux comprendre quel genre de description les jeux ont, nous pouvons faire une LDA à 5 composantes pour extraire des thèmes sous-jacents, puis regarder les mots les plus fréquents pour chaque thème. Nous obtenons les résultats suivant :

- Thème 1 : friends, games, puzzles, fun, level, use, vr, new, levels, time
- Thème 2 : ancient, battle, space, ship, story, explore, time, combat, new, world
- Thème 3 : characters, features, player, different, character, items, unique, mode, players, new
- Thème 4 : units, based, unique, players, strategy, time, battle, war, new, world
- Thème 5 : just, adventure, like, ll, characters, time, life, new, world, story

Pour résumer, les mots (hors stopwords communs, *play* et *game*) qui reviennent le plus dans les descriptions sont ceux présentés en figure 3.9.

3.3 Données catégorielles

Pour les données catégorielles (genres et plateformes), nous avons d'abord dû transformer les colonnes contenant des listes en plusieurs colonnes booléennes, chaque correspondant à un genre

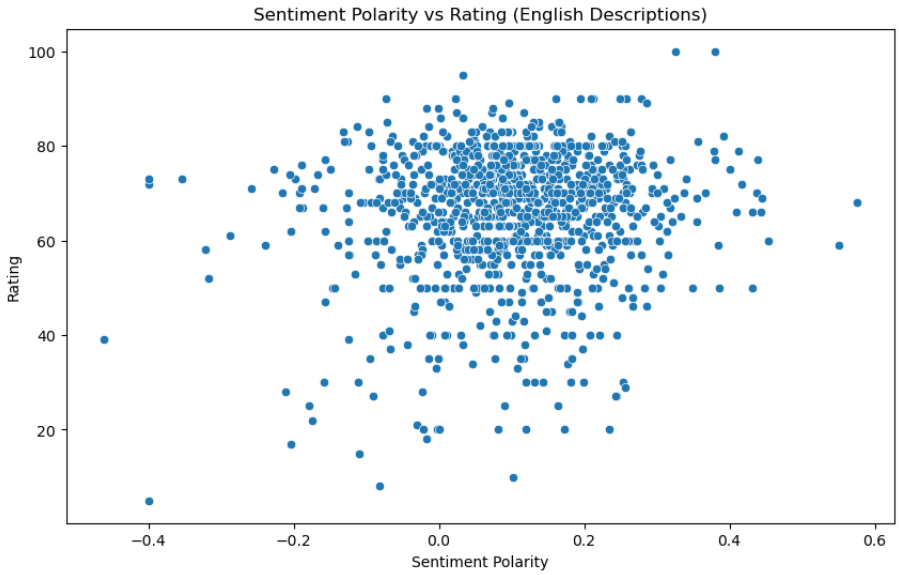

Figure 3.8: Notes des jeux en fonction de la polarité de leur description

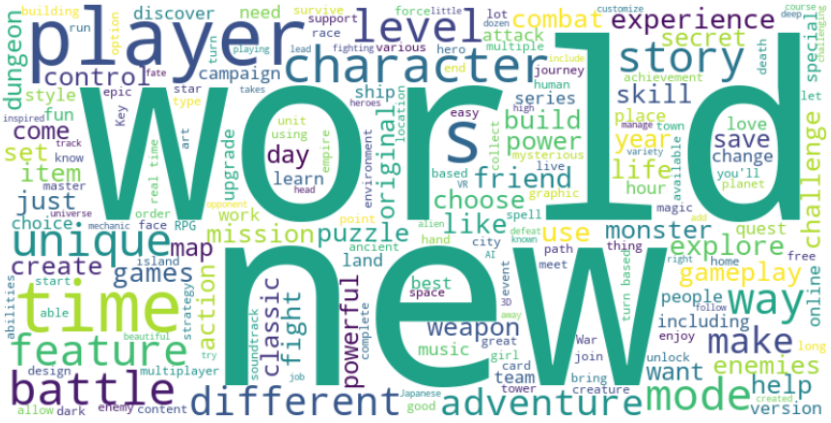

Figure 3.9: Nuage de mots des descriptions des jeux

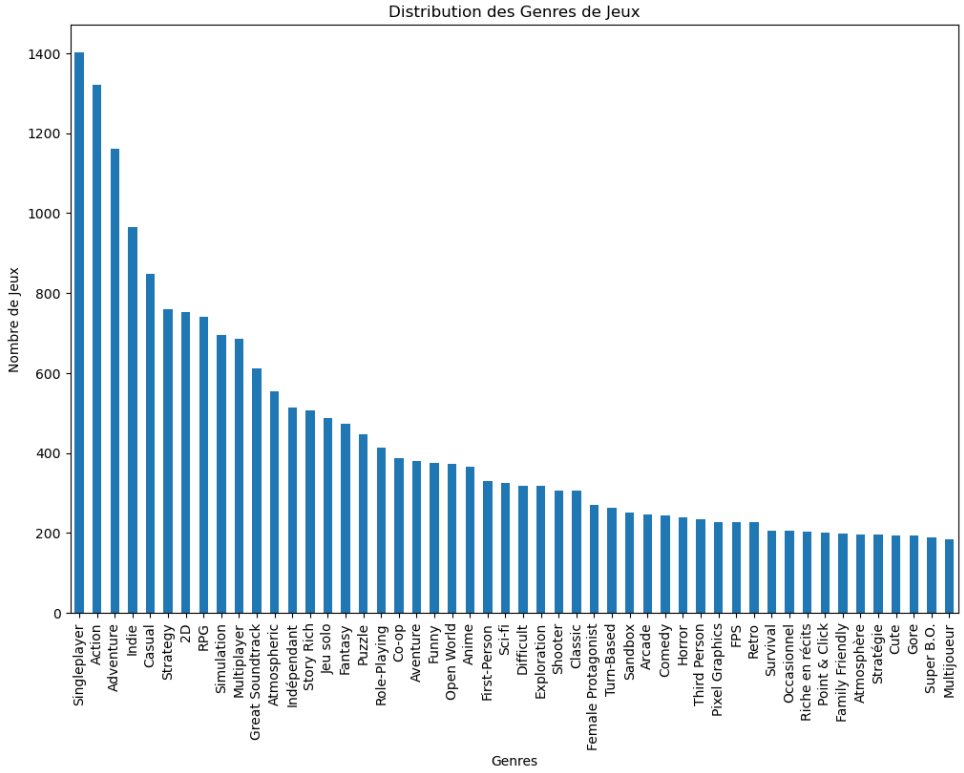

Figure 3.10: Nombre de jeux pour chaque genre

ou une plateforme en particulier.

Ensuite, nous pouvons regarder la répartition des genres. La figure 3.10 montre le nombre de jeux pour chaque genre (on n'affiche que les 50 premiers genres pour plus de lisibilité). On voit notamment que beaucoup de jeux sont des jeux solo, avec de l'action ou de l'aventure.

Nous comparons ensuite ce nombre de jeux avec le nombre de joueurs par genre (figure 3.11). On voit que l'ordre des genres est différent. Par exemple, les jeux multiplayers arrivent en deuxième position en terme de nombre de joueurs mais seulement en dixième position en terme de nombre de jeux.

Toujours sur les genres, nous pouvons observer sur la figure 3.12 que la moyenne des notes par genre est plus ou moins constante.

Nous nous penchons ensuite sur les plateformes. Comme pour les genres, nous regardons d'abord leur distribution (figure 3.13). Puis nous affichons leur note moyenne (figure 3.14). Encore une fois, les notes ne changent pas beaucoup en fonction des plateformes.

3.4 Analyse en Composantes Principales (PCA)

En appliquant une PCA à nos données numériques pour réduire la dimensionnalité et visualiser les corrélations, nous obtenons les ratio cumulatifs de variances présentés en figure 3.15. Nous décidons de garder 80% de l'information, soit 4 colonnes.

Nous traçons ensuite le cercle de corrélation (figure 3.16), et en superposant les données issues de la PCA avec les vecteurs représentant chaque colonne initiale, nous obtenons la figure 3.17.

Nous pouvons enfin faire un K-means sur les données provenant de la PCA pour observer les clusters (figure 3.18) et la répartition des variables dans chaque cluster (figure 3.19).

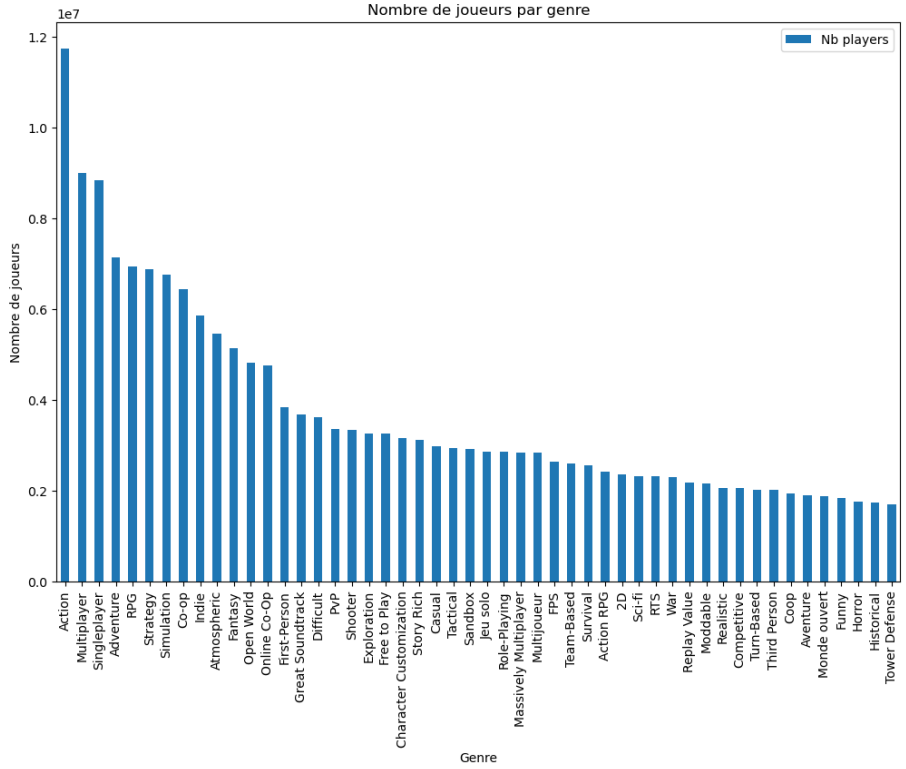

Figure 3.11: Nombre de joueurs pour chaque genre

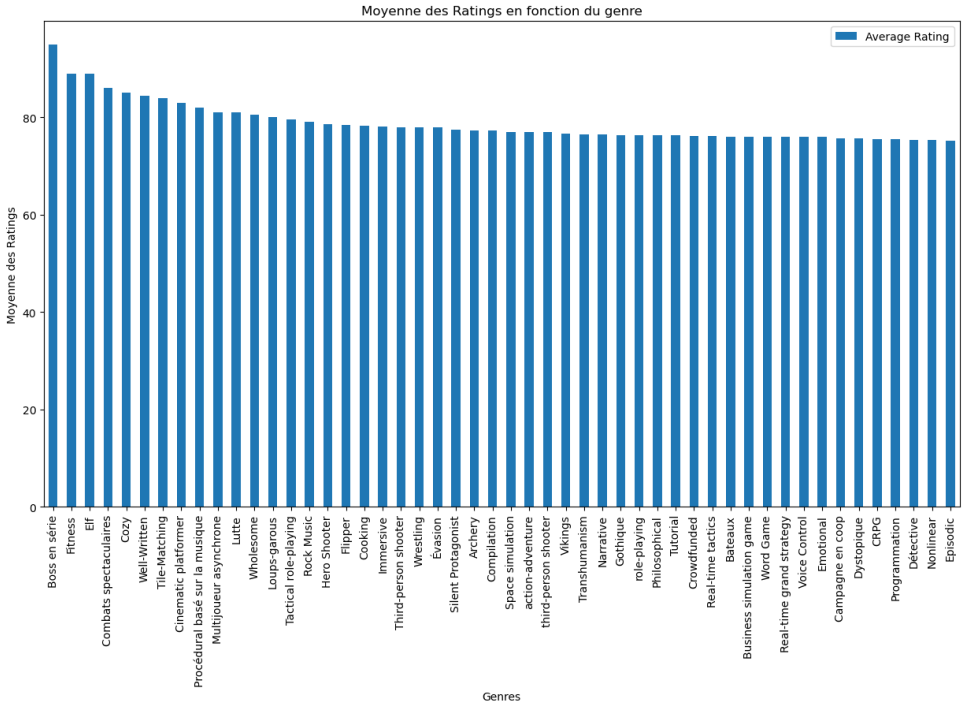

Figure 3.12: Moyenne des notes en fonction du genre

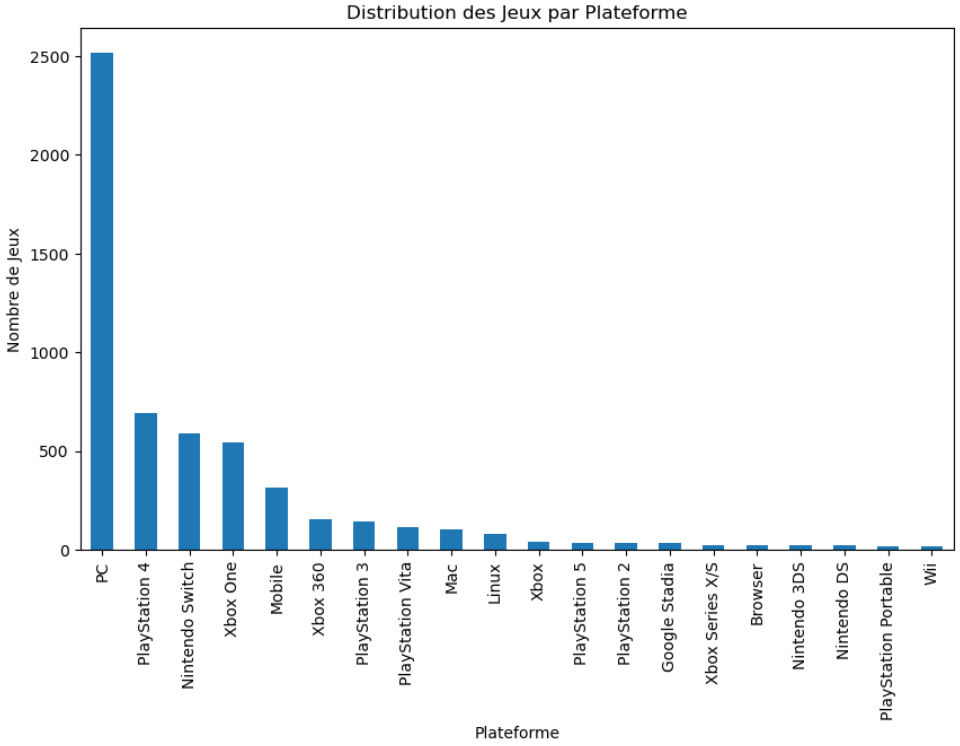

Figure 3.13: Nombre de jeux pour chaque plateforme

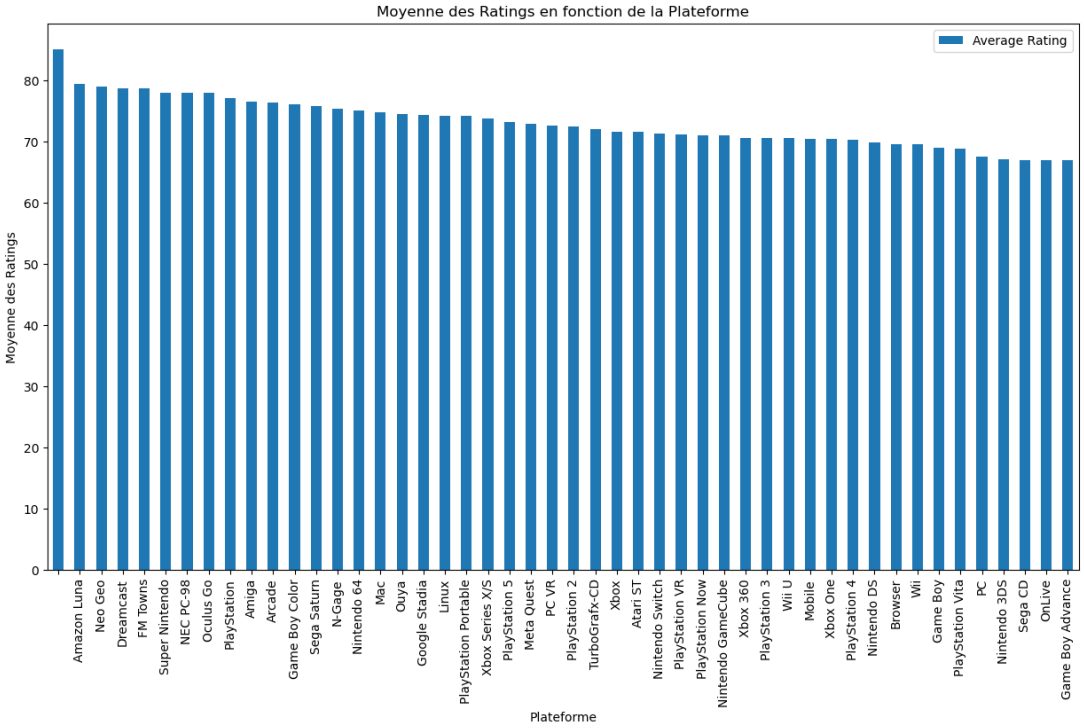

Figure 3.14: Moyenne des notes en fonction des plateformes

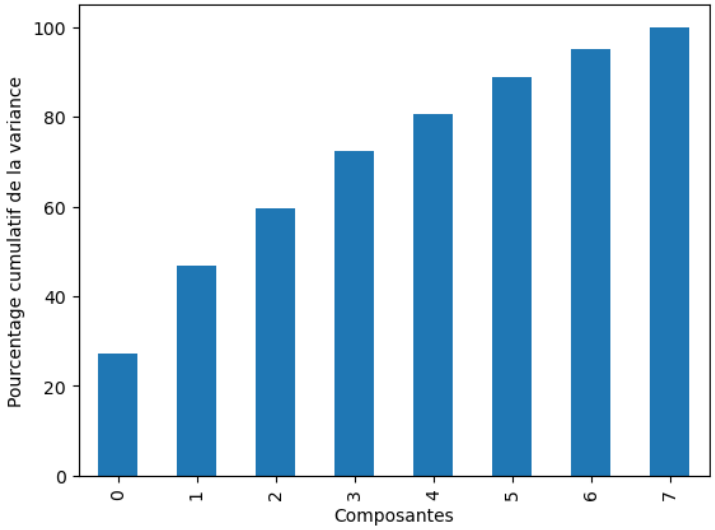

Figure 3.15: Pourcentage cumulatif de la variance pour chaque composante de la PCA

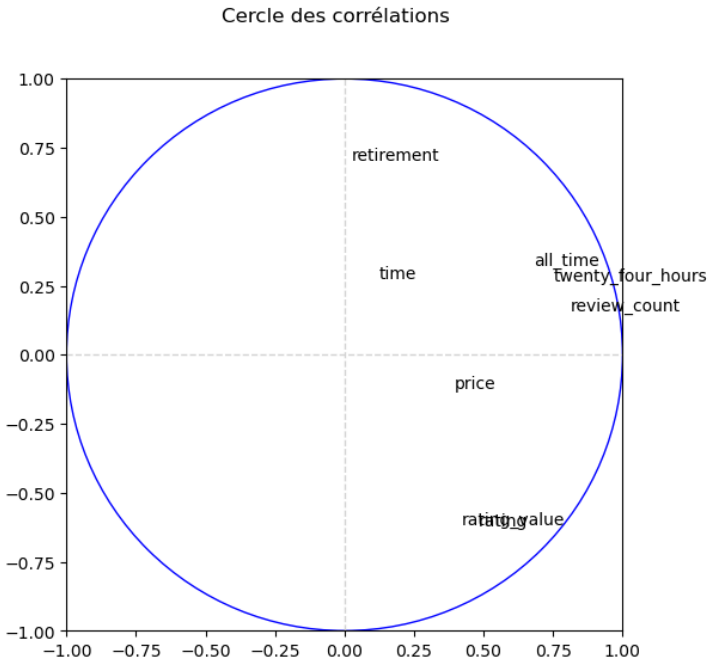

Figure 3.16: Cercle de corrélation des données numériques

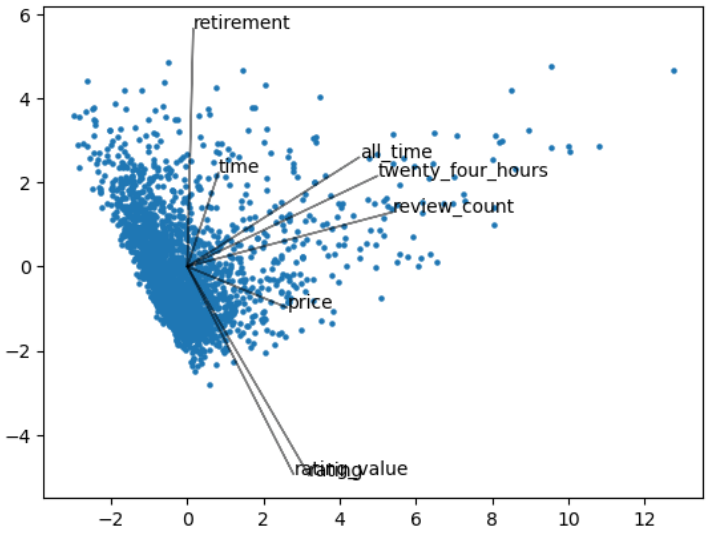

Figure 3.17: Projection des jeux sur les 4 premiers axes de la PCA, ainsi que les vecteurs représentant les axes précédents

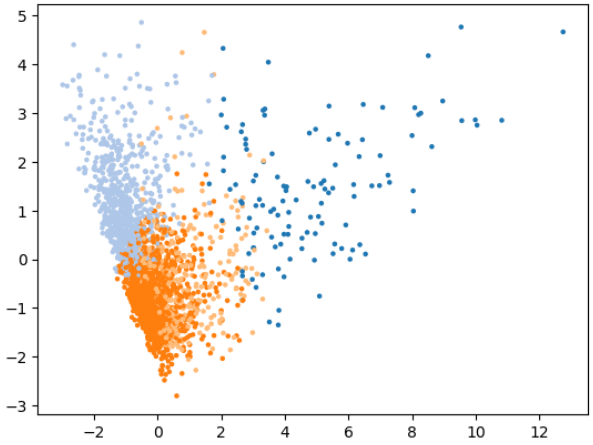

Figure 3.18: Clusterisation des données après la PCA avec K-means

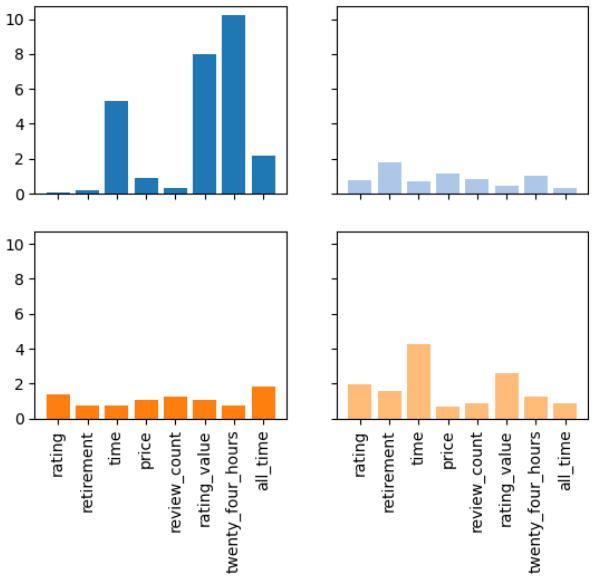

Figure 3.19: Répartition des variables pour chaque cluster

CHAPTER 4

MODÈLES

Maintenant que nous avons bien compris la répartition de nos données et les relations entre nos colonnes, nous décidons de faire deux types de modèles : des modèles de prédiction et des modèles de recommandation.

4.1 Prédiction

Nous avons construit des modèles de régression linéaire et de forêts aléatoires pour prédire des variables cibles telles que le *rating* et le *retirement*.

Nous essayons d'abord de prédire le futur rating d'un jeu. Pour cela on ne peut prendre en entrée que les plateformes, les genres, la date, le prix, et la durée du jeu. En effet, utiliser d'autres données telles que le nombre de joueurs n'aurait pas de sens car si une entreprise a la donnée du nombre de joueurs, elle a sûrement aussi celle du rating.

En faisant tourner une régression linéaire sur l'ensemble des données mentionnées ci-dessus, le résultat est excessivement mauvais, avec un score R-2 de -8×10^{17} et une MSE de 9×10^{19}.
En utilisant XGBoost, on peut regarder l'importance de chaque *feature*. Les résultats obtenus (figure 4.1) montrent que ce sont les genres qui sont le plus pris en compte.

Nous décidons donc de faire un pré-traitement sur le genres. Après avoir transformé la liste des genres en plusieurs colonnes, nous nous sommes retrouvés avec environ 700 colonnes de genres. Nous décidons donc d'utiliser une PCA pour réduire drastiquement ce nombre. En faisant varier le nombre de colonnes gardées, on remarque une amélioration des résultats (figure 4.2). Ainsi, en prenant 47 colonnes, on obtient un score R2 de 0.1619.
En réutilisant XGBoost après réduction du nombre de colonnes, on obtient la figure 4.3. On peut aussi afficher les coefficients de la régression linéaire pour voir quelles caractéristiques sont bonnes et mauvaises (figure 4.4).

Prédire le retirement d'un jeu pré-existant peut aussi être intéressant. En utilisant un modèle `RandomForestRegressor` à 90 estimateurs et 50 random states qui prend en entrée `rating`, `pourcentage_pos`, `platform`, `genres`, `date`, `price`, `time`, `twenty_four_hours`, `all_time`, on obtient un score R-2 de 0.449 en test.

4.2 Recommandation

Pour recommander des jeux, nous créons notre propre algorithme.
Le système de recommandation se base sur la similarité cosinus sur les descriptions des jeux et les caractéristiques numériques. Les descriptions des jeux, ainsi que le titre et les listes de plateformes et de genres sont converties en vecteurs à l'aide de `TfidfVectorizer()`, et les

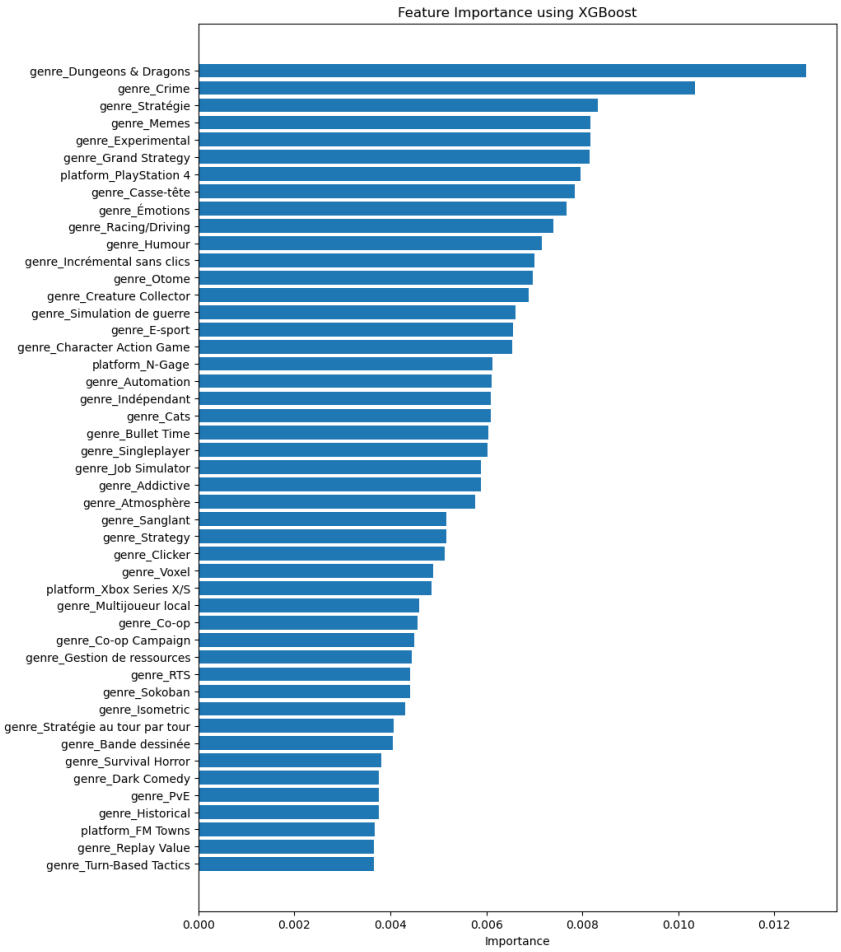

Figure 4.1: Importance des caractéristiques dans la prédiction du rating

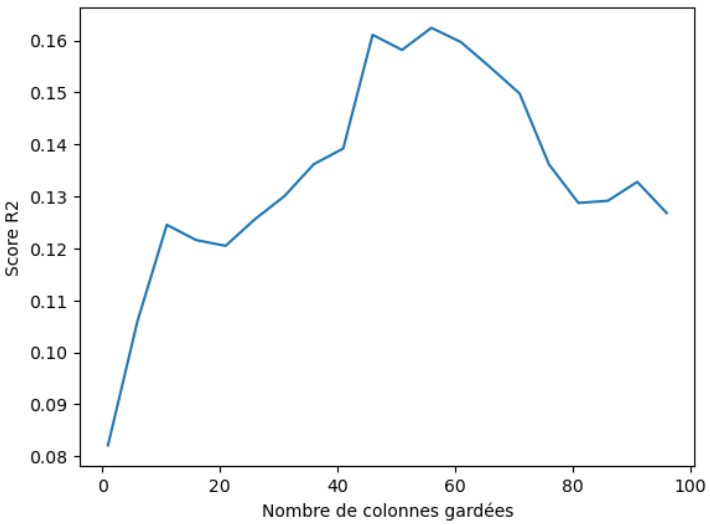

Figure 4.2: Résultats de prédiction du rating en fonction du nombre de colonnes représentant les genres gardées après la PCA

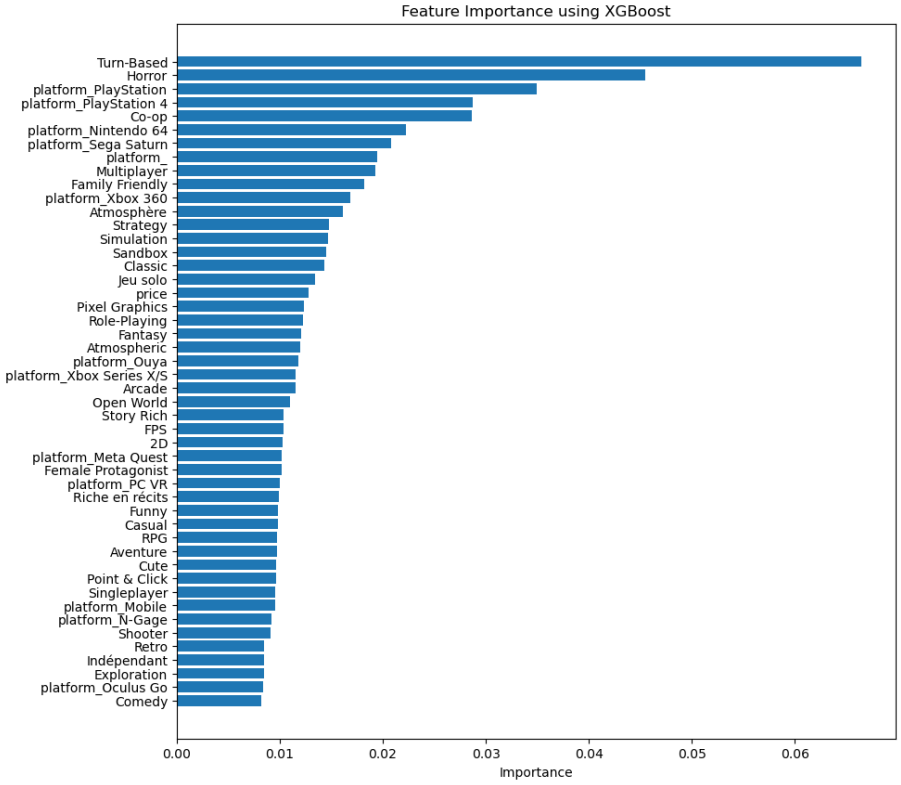

Figure 4.3: Importance des caractéristiques dans la prédiction du rating après avoir réduit le nombre de colonnes représentant les genres à 47

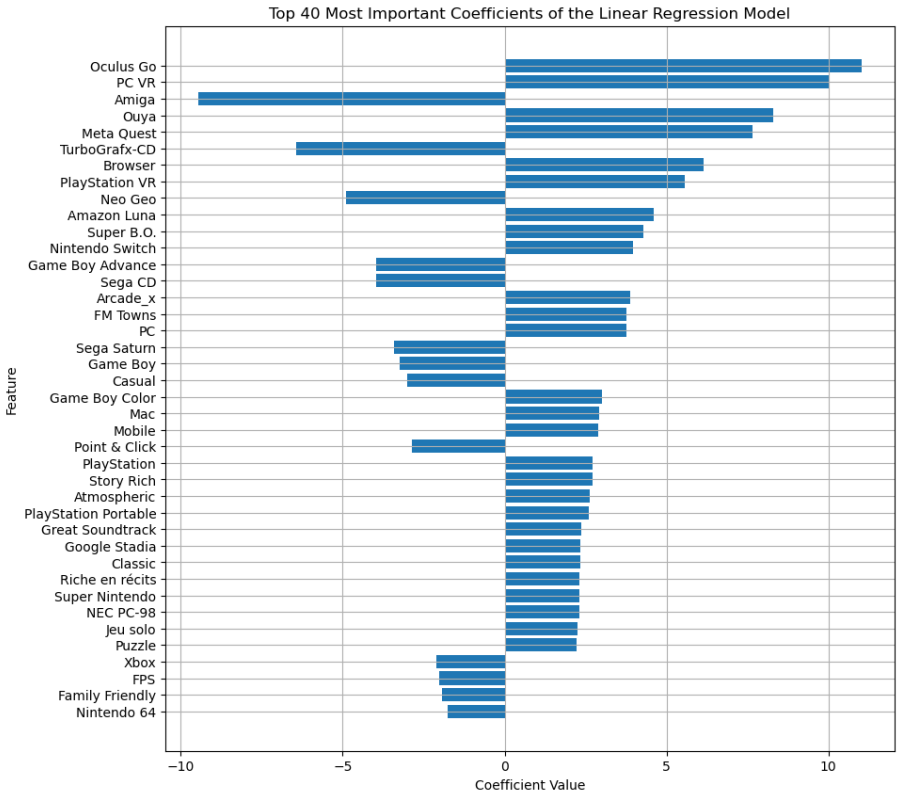

Figure 4.4: Coefficients de la régression linéaire pour prédire le rating

donnéees numériques sont normalisées. On calcule ensuite la similarité cosinus sur les vecteurs résultants.

L'utilisateur donne en entrée une liste de jeux qu'il a bien aimé. On cherche dans la matrice de similarité les lignes correspondantes, et on fait une moyenne sur les jeux que l'utilisateur a donné pour se retrouver avec une liste de similarité entre chaque autre jeux et les jeux d'entrée. Il ne reste ensuite plus qu'à renvoyer les jeux correspondant aux valeurs les plus élevées.

Par exemple, si l'utilisateur rentre les jeux *7 Wonders II*, *Clue/Cluedo: The Classic Mystery Game* et *A Hat in Time*, le système de recommandation renvoie *Paint the Town Red*, *Grim Legends 2: Song of the Dark Swan*, *Ding Dong XL* et *D'LIRIUM*.

Nous avons aussi codé une variante de ce système en pondérant les similarités cosinus par les notes des jeux afin de recommander en priorité des jeux bien notés. Avec cette version de l'algorithme, pour les mêmes jeux nous obtenons les résultats suivants : *Streets of Rogue*, *Paint the Town Red*, *Supraland* et *Bejeweled 3*.

CHAPTER 5

CONCLUSION

En conclusion, notre analyse des données provenant de HowLongToBeat, Steam, et Steam Charts a permis de mettre en lumière des tendances significatives de l'industrie du jeu vidéo telles que l'augmentation des notes et des prix des jeux au fil des ans, ainsi que la corrélation entre les évaluations des utilisateurs sur différentes plateformes. Notre exploration des genres et des plateformes a révélé des préférences distinctes des joueurs, montrant notamment que certains genres attirent plus de joueurs malgré un nombre plus faible de titres.

L'analyse textuelle des descriptions de jeux a montré que les sentiments exprimés sont généralement neutres, indiquant que ces descriptions visent plus à décrire l'ambiance du jeu qu'à influencer directement les notes.

Nos modèles de prédiction, bien que confrontés à des défis, ont montré que certaines caractéristiques, notamment les genres des jeux, jouent un rôle crucial dans la prédiction des évaluations.

Enfin, notre projet met en évidence l'importance d'un nettoyage et d'une normalisation rigoureuse des données pour obtenir des résultats précis et exploitables.

Les visualisations et les analyses exploratoires effectuées offrent une vue d'ensemble pouvant guider les développeurs et éditeurs de jeux dans leurs décisions stratégiques, pour comprendre les facteurs de succès dans l'industrie du jeu vidéo et ouvrent la voie à des recommandations personnalisées pour les joueurs.

BIBLIOGRAPHY

[1] *HowLongToBeat*. https://howlongtobeat.com/.

[2] *Steam*. https://store.steampowered.com/?.

[3] *Steam Charts*. https://steamcharts.com/.