

GameSpy



ANALYSE DES VENTES DE JEUX VIDÉOS

PROJET DATA ANALYST

SESSION CONTINUE SEPT 2021 - AVRIL 2022

Constance MARTINA

Jean-Luc DJEKE

Jérémy PIRIS

DESCRIPTION DU PROJET

Contexte

Les jeux vidéo sont des jeux électroniques utilisés sur des PC, des consoles, des téléphones portables ou d'autres supports. L'introduction des jeux multi-joueurs en ligne a donné un énorme coup de fouet au secteur, car il s'agit d'un lieu de rencontre pour les groupes en ligne. Toute personne qui pratique le jeu vidéo est appelée un joueur.

Le secteur des jeux vidéo est divisé en trois grandes catégories : les jeux vidéo sur téléphone portable, les jeux vidéo sur console et les jeux vidéo sur PC.

Le potentiel de l'industrie du jeu vidéo implique plusieurs pipelines. Il est important de comprendre tous les processus impliqués dans l'industrie. Par exemple, le développement des jeux, la livraison des jeux vidéo aux clients, la distribution de chaque jeu vidéo, la propriété des jeux, ou l'accord d'un secteur spécifique avec les joueurs.

Problématique

Au cours des dernières années, l'industrie des jeux vidéo a connu une croissance rapide. Loin de stopper cette dynamique, la pandémie de Covid-19 l'a amplifiée. 2020 a, en effet, été une année record, puisque les mesures de confinement prises par la plupart des pays du monde ont stimulé les activités virtuelles. En France, par exemple, l'industrie des jeux vidéo a réalisé un chiffre d'affaires de 5,3 milliards d'euros, selon le Syndicat des éditeurs de logiciels de loisirs (SELL). Près de la moitié de ce chiffre d'affaires est lié à l'univers des consoles de jeux.

Vu cette croissance importante en termes de ventes, beaucoup d'entreprises investissent d'énormes sommes d'argent dans cette industrie. Comme tout autre entreprise, elles souhaitent maintenir une connaissance plus ou moins précise du risque lors de la sortie d'un nouveau produit, donc d'un nouveau jeu vidéo. Ce projet s'inscrit dans cette logique.

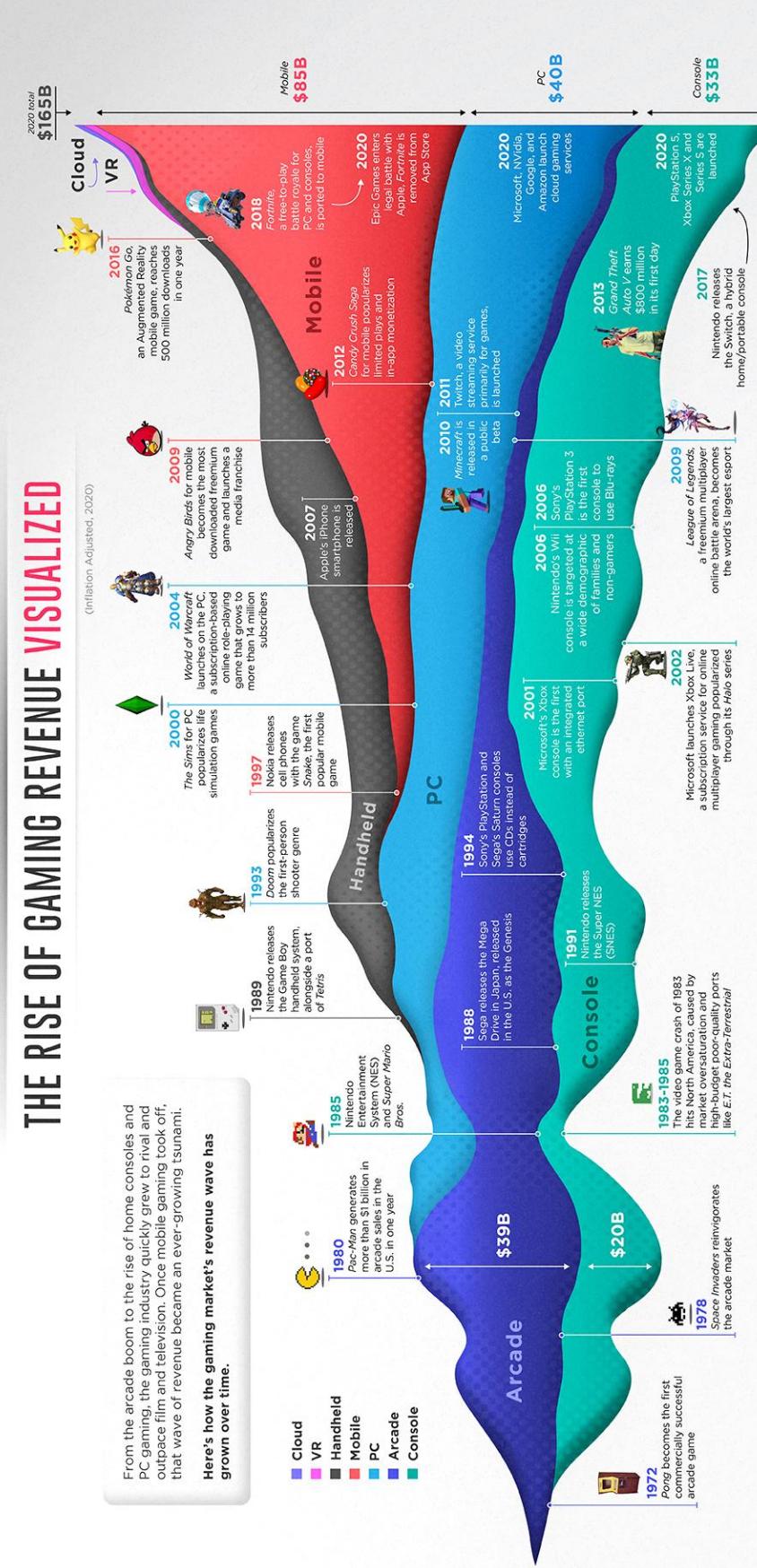
Ci-dessous, une illustration de l'évolution des revenus du marché de 1972 à 2020 :

THE RISE OF GAMING REVENUE VISUALIZED

From the arcade boom to the rise of home consoles and PC gaming, the gaming industry quickly grew to rival and outpace film and television. Once mobile gaming took off, that wave of revenue became an ever-growing tsunami.

Here's how the gaming market's revenue has grown over time.

Cloud
VR
Handheld
Mobile
PC
Arcade
Console



SOURCE: Pelham Smithers | RESEARCH + WRITING: Orrin Walsh | DESIGN + ART DIRECTION: Clayton Vidowson
 COLLABORATORS: visualcapitalist.com

visualcapitalist.com

VISUAL CAPITALIST

OBJECTIF DE NOTRE ÉTUDE

L'objectif de notre étude est d'estimer le nombre de ventes d'un jeu vidéo avant sa sortie.

Pour ce faire, nous partirons du dataset fourni par l'équipe de DataScientest composé d'une liste de jeux vidéos, de leur nombre de ventes et de quelques informations complémentaires concernant le jeu (année de sortie, éditeur, plateforme, genre).

Nous enrichirons ce dataset par d'autres informations complémentaires afin d'avoir suffisamment de matière pour pouvoir entraîner un modèle de machine learning et, ainsi, parvenir à estimer au mieux le nombre de ventes pour un nouveau jeu à venir.

Sommaire

1. Découverte du dataset initial	6
2. Enrichissement du dataset initial	7
3. Analyse et visualisation	9
4. 2ème phase d'enrichissement	26
4.1. Création des moyennes mobiles	26
4.2. Scraping d'informations avant lancement des jeux	28
5. Modélisation	32
5.1 Pré-processing	32
5.1.1. Gestion des valeurs manquantes	32
5.1.2. Encodage des données	32
5.2 Choix du modèle de machine learning	34
5.2.1. Régression	34
5.2.2. Classification	37
Conclusion	47

1. Découverte du dataset initial

Nous avons récupéré un dataset initial avec **16598 lignes**.

Dataset fourni : <https://www.kaggle.com/gregorut/videogamesales>

Ce dataset propose un classement des jeux vidéos en fonction de leur nombre de ventes, dont voici le détail et les informations globales qu'il contient :

#	Column	Non-Null Count	Dtype
0	Rank	16598	non-null int64
1	Name	16598	non-null object
2	Platform	16598	non-null object
3	Year	16327	non-null float64
4	Genre	16598	non-null object
5	Publisher	16540	non-null object
6	NA_Sales	16598	non-null float64
7	EU_Sales	16598	non-null float64
8	JP_Sales	16598	non-null float64
9	Other_Sales	16598	non-null float64
10	Global_Sales	16598	non-null float64

La **clé d'entrée** de chaque ligne est **le nom du jeu** et une **plateforme de jeu**. Si un jeu est disponible sur plusieurs plateformes, il peut donc y avoir plusieurs lignes pour un même jeu et la plateforme qui diffère.

Les jeux présentés dans le dataset ont une **date de lancement entre 1980 à 2020**. Ils couvrent donc une large période qui va nous aider à comprendre l'évolution du marché des jeux vidéo dans le monde. Néanmoins, nous voyons que la répartition du **nombre de jeux par an n'est pas homogène** (certaines années sont associées à un seul jeu). Il faudra bien prendre cela en compte dans nos analyses.

Pour chaque jeu, nous avons le **genre** qui y est associé, **l'entreprise qui l'a publié** et un volume de **ventes** pour plusieurs régions du monde : les Etats-Unis, le Japon, l'Europe et le reste du monde. Toutes les ventes de ces régions sont sommées dans une colonne: *Global_Sales*.

Rapidement, l'exécution de la fonction *describe* nous permet de voir que l'échelle des ventes est très basse (proche de 0) avec un maximum sur les ventes globales à 82,74. Nous comprenons donc que ces dernières sont exprimées en millions et qu'une grande **majorité des jeux présentés dans le dataset cumulent des ventes autour de 0,1 soit 100 000 unités**. La moyenne étant à 0.53 et la médiane à 0.17.

Le dataset est composé de **très peu de données** et notamment peu de données pouvant expliquer le niveau des ventes.

Nous nous rendons compte que pour comprendre le marché et la composition des premières données qui nous ont été partagées, **nous allons devoir les compléter**.

2. Enrichissement du dataset initial

Nous avons ciblé **3 sites populaires**, spécialisés dans les revues de jeux vidéo : **jeuxvideo.com, metacritic et gamekult**.

Notre objectif était de récupérer des données pouvant expliquer le volume des ventes d'un jeu par rapport à un autre afin d'enrichir le dataset initial avant l'analyse approfondie.

Nous avons donc scrapé les données suivantes via la **librairie BeautifulSoup** et **GoogleSearch** :

- **Date de lancement** : avec la précision jour, mois, année
- **Studio** : l'entreprise qui a développé le jeu vidéo
- **Licence** : la série liée au jeu vidéo (ex : licence Mario, GTA, FIFA, etc.)
- **Note** : la note donnée par un journaliste ou la moyenne de plusieurs notes journalistes
- **Rate** : la note moyenne donnée par les joueurs
- **Reviews** : le nombre d'avis laissés par les joueurs

Les nouvelles données récupérées sur les 3 sites n'avaient pas forcément les mêmes formats de date ou la même échelle de notations. Certaines données étaient de meilleures qualités en fonction des sites et de ce qui avait pu être récupéré au moment du scraping.

Voici un détail des démarches de scraping effectuées sur chacun des trois sites :

- Site Gamekult :



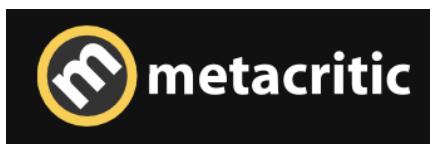
- **1er scraping** :

- Après avoir identifié les pages du site Gamekult sur lesquelles se trouvaient le plus d'informations utiles pour développer notre dataset, création d'une boucle pour récupérer tous les liens de ces pages
 - Dans chacune de ces pages, récupération des éléments html contenant les informations utiles
 - Cleaning et récupération pour chaque jeu des infos de ces éléments html : **nom du jeu, note des joueurs, nombre d'avis des joueurs, note attribuée par Gamekult, distributeur, développeur, éditeur et licence**.
 - Création du 1er dataset

- **2ème scraping :**

- Certains jeux du dataset original n'apparaissent pas le dataset issu du 1er scraping. Nous avons donc créé une liste de ces jeux manquants et nous sommes partis du nom du jeu dans le dataset original pour scraper cette fois-ci les résultats google à l'aide de la bibliothèque googlesearch.
- Pour chaque lien valide (c'est-à-dire les liens vers la page Gamekult de test), application de la même méthode faite au 1er scraping pour récupérer les informations utiles et augmenter le dataset.

- **Site Metacritic/Jeux :**



- Vérification des pages disponibles au scraping via <https://www.metacritic.com/robots.txt>
- Scraping de tous les liens des jeux pour chaque plateforme
- Pour chaque lien de jeux, récupération du **nom du jeu**, de la **plateforme**, du **studio**, de la **note Metacritic**, de la **note des utilisateurs** et de la **date de sortie** du jeu
- Constitution du dataset final et nettoyage des données scrapées

- **Site jeuxvideo.com :**



- Identification de la page <https://www.jeuxvideo.com/tests.htm> regroupant tous les jeux testés
- Scraping sur cette page de tous les liens amenant à chaque page jeu
- Pour chaque jeu dans la liste, récupération du **nom de jeu**, des **plateformes compatibles**, du **nombre d'avis**, de la **note des joueurs par plateforme**, de la **note journaliste** et de la **date de lancement** du jeu.
- Nettoyage des données et constitution d'un dataset à partir des données récupérées

Après avoir **mis en commun toutes ces données**, nous avons défini un **format unique**, ramené les notations sur une **même échelle** et appliqué des **règles de priorité** sur les données récupérées. Pour chaque nouvelle entrée, l'information était prise du site offrant les données les plus propres. Les informations manquantes étaient complétées par le deuxième

site le plus complet et enfin par le dernier site. Cela afin de maximiser le nombre de clés jeu/plateforme sur lesquelles nous allions pouvoir ajouter de la donnée et d'éviter les NaNs.

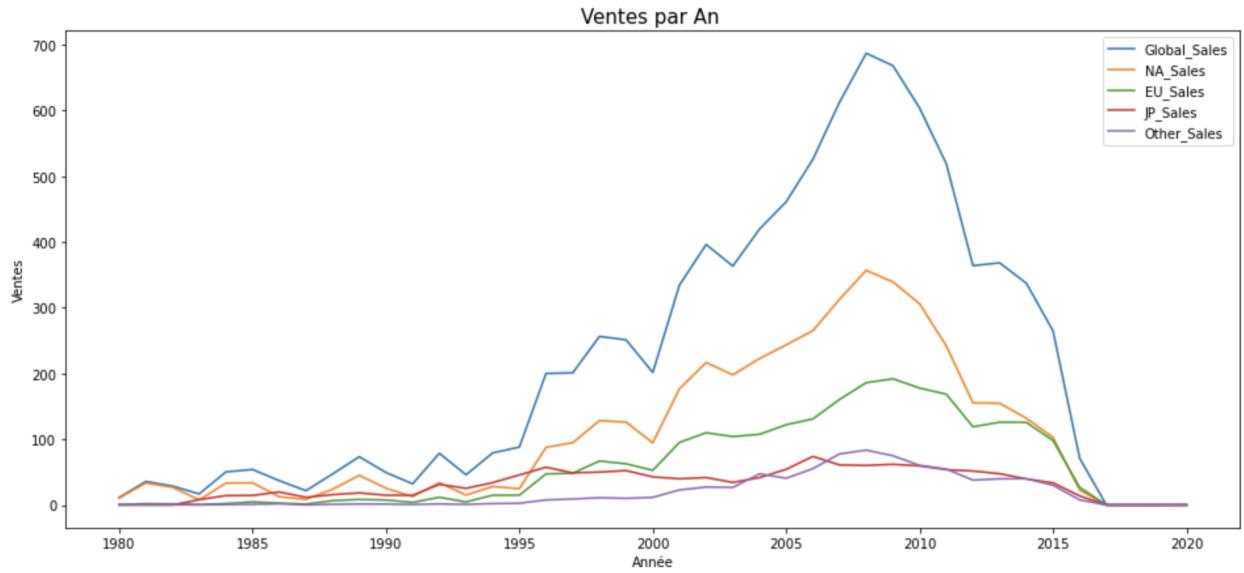
3. Analyse et visualisation

Suite à nos premières observations, à l'ajout de nouvelles données provenant du scraping et ayant en tête les caractéristiques du marché des jeux vidéo, nous avons posé les **15 hypothèses suivantes** :

1. L'arrivée de certaines consoles de jeux-vidéos a influé significativement les ventes de jeux video
2. Le mois de sortie du jeu influe sur l'évolution des ventes
3. La répartition des ventes par région évolue au fur et à mesure des décennies
4. Le genre des jeux les plus vendus n'est pas le même en fonction des régions du monde
5. Les plateformes qui font le plus de ventes sont celles pour lesquelles il y a le plus de jeux
6. Les genres qui font le plus de ventes sont ceux pour lesquels il y a le plus de jeux
7. Certains genres sont mieux notés que d'autres
8. Certaines variables sont corrélées entre elles et surtout avec des variables de ventes
9. Les jeux les mieux notés sont les plus vendus
10. Les jeux qui font partie d'une licence sont les plus vendus
11. Les jeux qui comptabilisent le plus de ventes ont été lancés par les meilleurs studios
12. Les éditeurs qui font le plus de ventes produisent le plus de jeux
13. Les éditeurs avec le plus de ventes sont ceux dont les jeux ont les meilleures notes moyennes
14. Les studios qui font le plus de ventes lancent le plus de jeux
15. Les studios avec le plus de ventes sont ceux dont les jeux ont les meilleures notes moyennes

Chacune de ces hypothèses a été vérifiée via la **data visualisation**.

3.1. L'arrivée de certaines consoles de jeux-vidéos a influé significativement sur les ventes de jeux

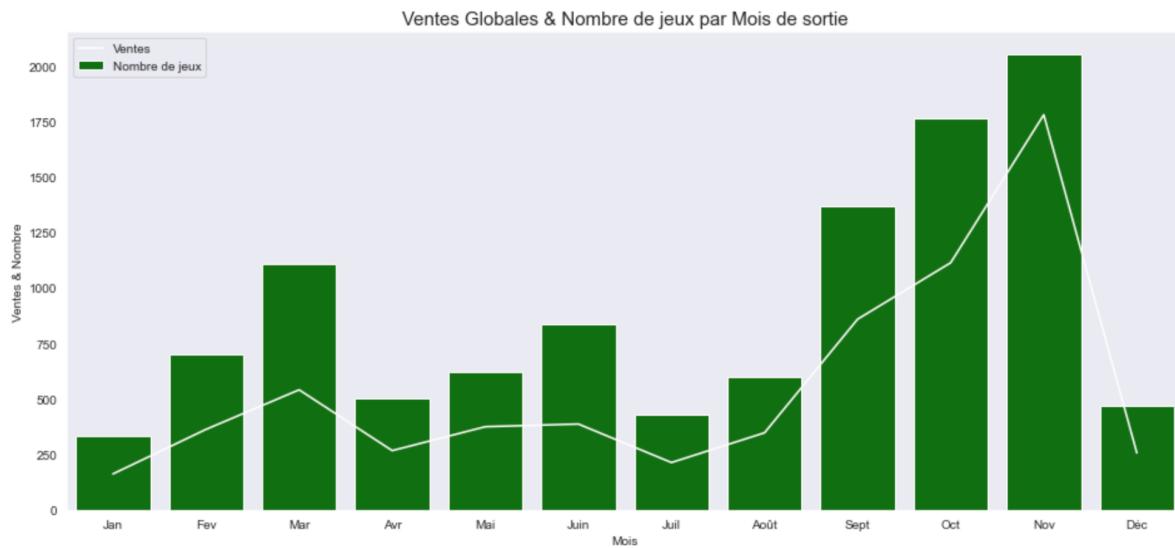


Nous remarquons que les ventes de jeux vidéo ont fait un bon significatif à partir des années **1990**. Cette période correspond à l'essor des consoles domestiques (par rapport aux jeux d'arcades) qui font leur entrée sur le marché à cette époque. C'est alors le début de l'âge d'or des jeux vidéo avec la sortie de consoles qui resteront plusieurs années sur le marché (Playstation, Xbox, Wii, etc.). Cf image introduction.

Dans notre dataset, les ventes de jeux vidéo **atteignent leur paroxysme en 2010** et décroissent par la suite. Mais cela s'explique par le fait qu'il y a beaucoup moins de jeux présents dans le dataset à partir de ces années-là.

Nous observons déjà grâce à cette première visualisation que **l'Amérique du Nord a une part importante des ventes mondiales**.

3.2. Le mois de sortie du jeu influe sur l'évolution des ventes

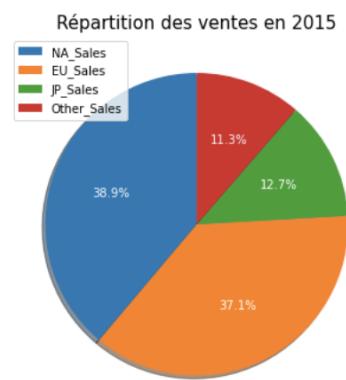
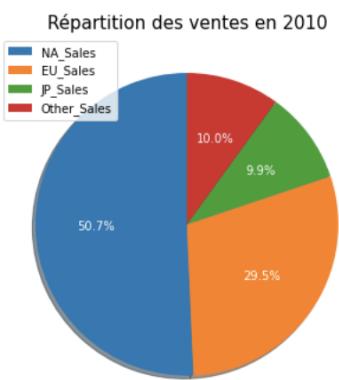
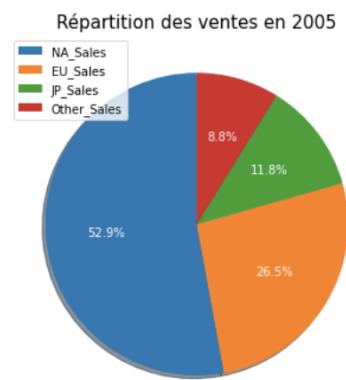
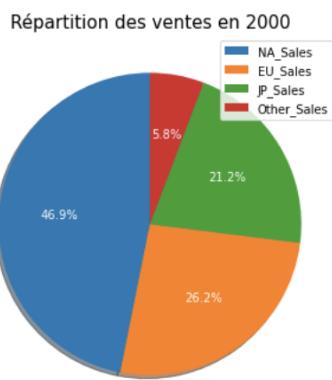
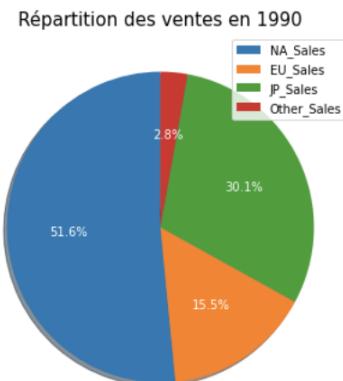
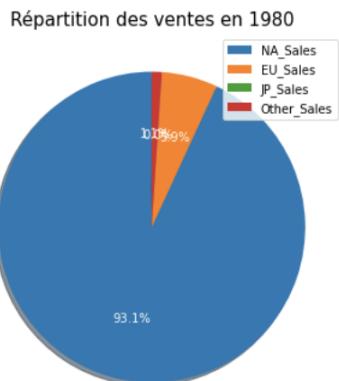


Nous voyons que la **fin d'année est propice à la sortie de nouveaux jeux vidéo** et cela est certainement lié aux fêtes de fin d'année.

En regardant la courbe des ventes seule, nous pourrions penser à une saisonnalité du marché sur la fin de l'année. Hors quand nous regardons les **ratios du nombre de jeux lancés par rapport aux ventes**, nous voyons que les derniers mois de l'année ne sont pas forcément les mieux placés.

Mars et Juin sont des mois intéressants pour le marché car les ventes sont élevées par rapport au nombre de jeux lancés.

3.3. La répartition des ventes par région évolue au fur et à mesure des décennies



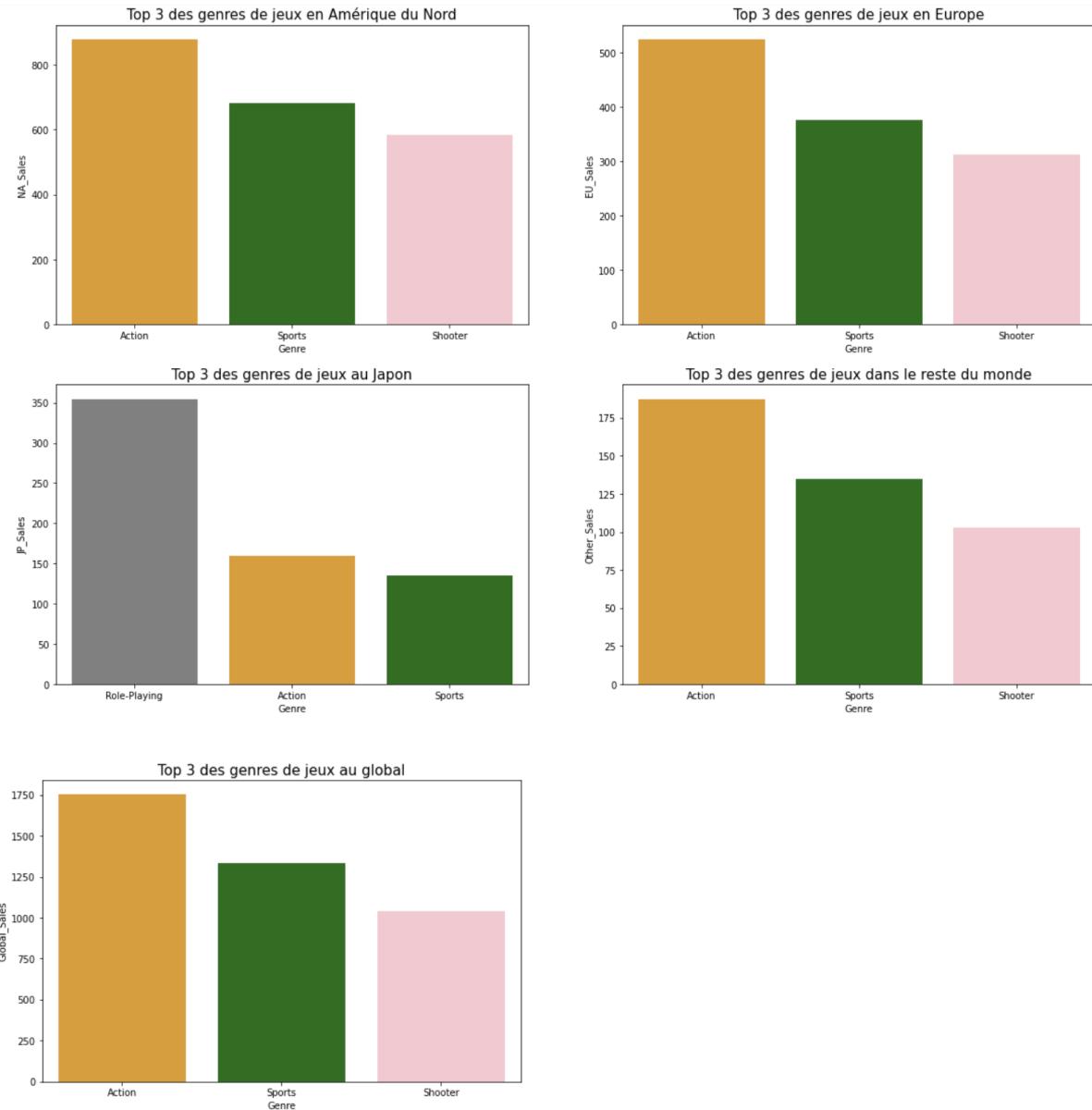
Ce que nous avons observé dans le premier graphique se confirme bien ici. **L'Amérique du Nord possède la majorité du marché mondial des jeux vidéo.**

Cependant, nous voyons bien ici qu'elle avait quasiment un **monopole** sur le marché du jeu vidéo en **1980** et qu'en **2015**, sa part de marché est **égale à celle de l'Europe**, qui a évolué au fur et à mesure des années.

Le **Japon** a connu un **essor** dans les **années 90** mais sa part de marché a eu tendance à diminuer depuis. Enfin, petit à petit, nous voyons aussi les **ventes d'autres régions émerger** sur le marché.

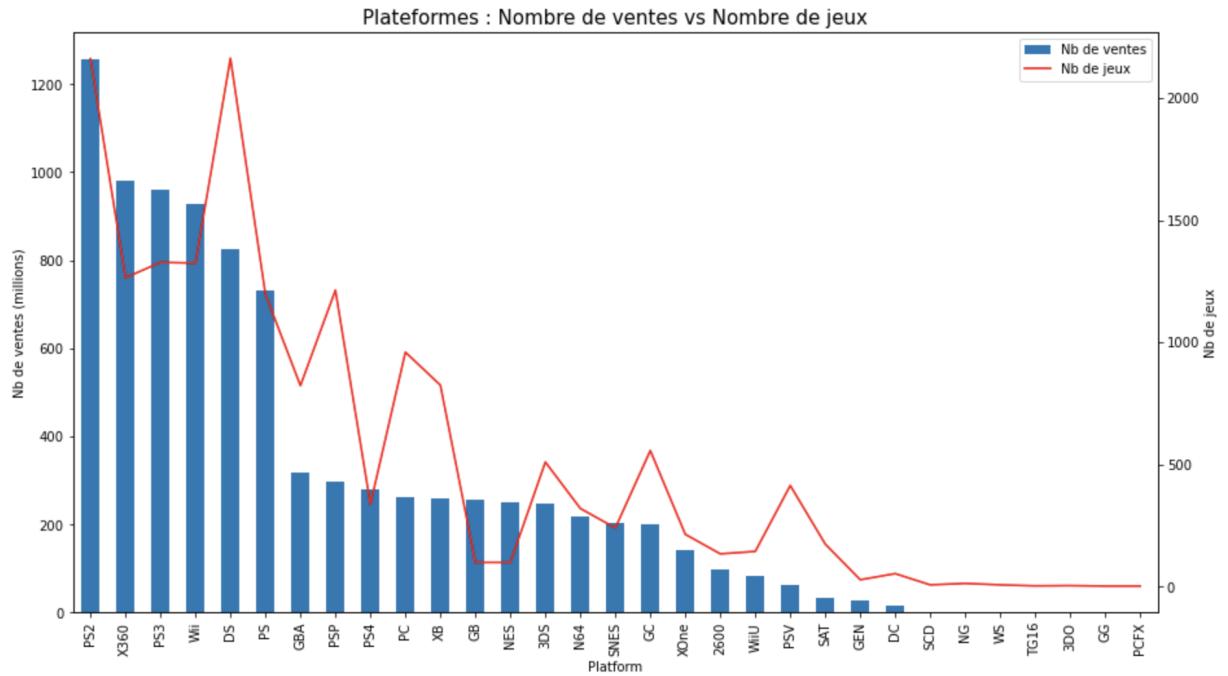
La répartition précédente a été faite que jusqu'en 2015 car au vu du faible nombre de jeux présents dans le dataset après cette période, les données nous semblaient moins représentatives.

3.4. Le genre des jeux les plus vendus n'est pas le même en fonction des régions du monde



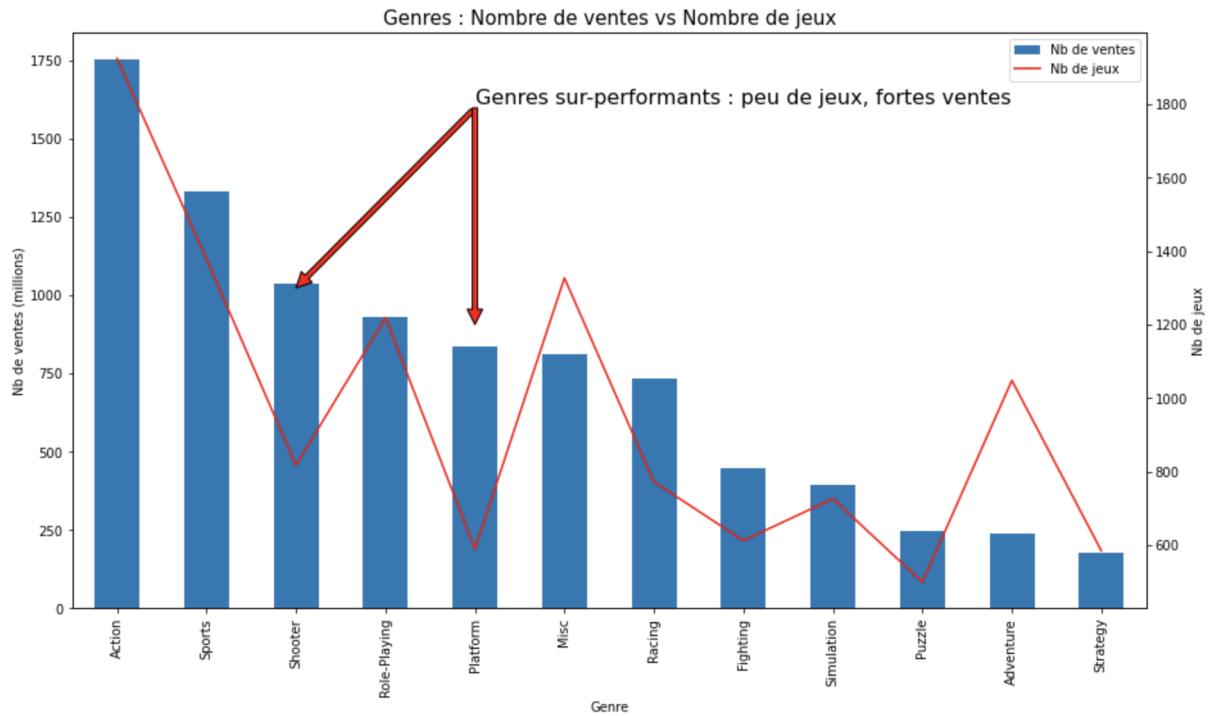
Nous retrouvons les **mêmes genres de jeux vidéo** les plus vendus entre les régions **Amérique du Nord, Europe et reste du monde**. Les jeux d'action, de sport et de tirs se retrouvent donc dans le top des ventes mondiales. Seul le **Japon** se démarque, en préférant largement les jeux de rôles parmi tous les autres genres (presque 2x plus de ventes que le genre action qui se retrouve en 2ème position).

3.5. Les plateformes qui font le plus de ventes sont celles pour lesquelles il y a le plus de jeux



Nous pouvons observer une courbe du nombre de jeux qui globalement va dans le sens des ventes de la plateforme. Mais dans le détail, nous voyons que la courbe est très irrégulière et que **l'hypothèse ne se vérifie pas**. Par exemple la Xbox et la PS3 (respectivement plateformes **top 2 et top 3** sur le niveau de ventes de jeux) ont eu bien moins de jeux compatibles (**1/3 de moins**) que la DS qui se retrouve en **5ème position** au regard des ventes de jeux par plateforme.

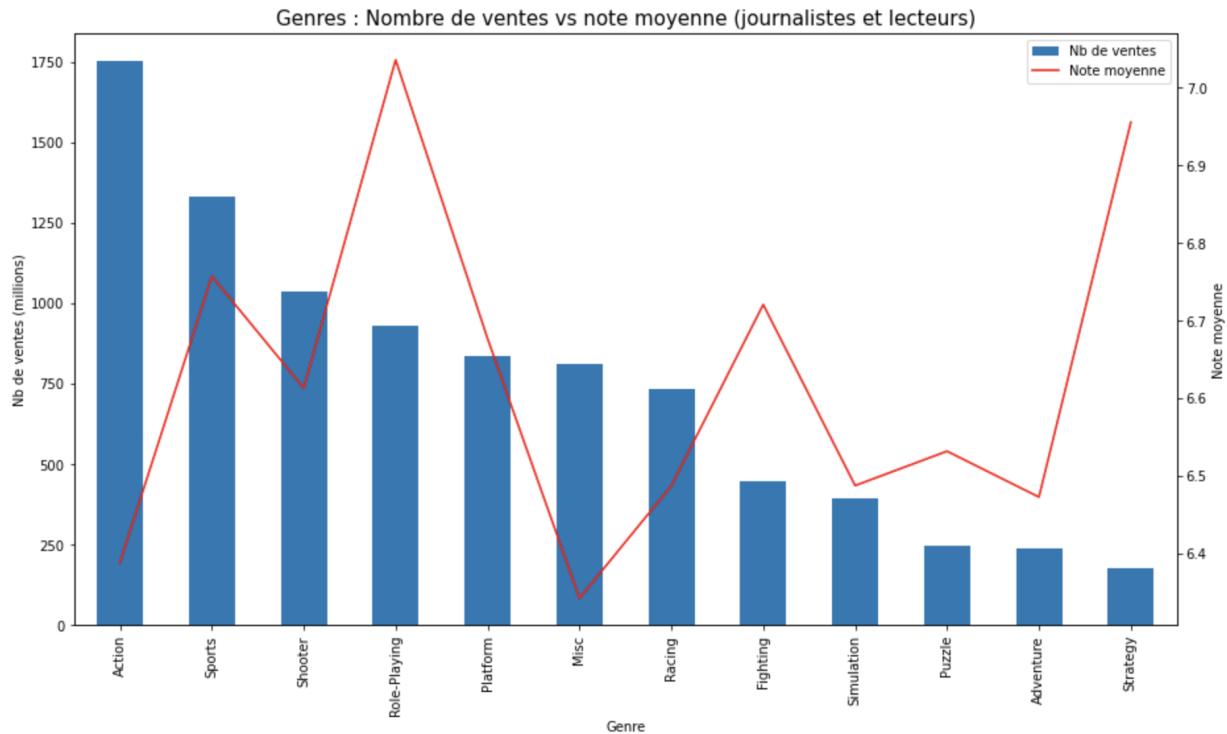
3.6. Les genres qui font le plus de ventes sont ceux pour lesquels il y a le plus de jeux



Sur l'échelle des genres de jeux, il n'y a **pas non plus de corrélation entre le nombre de ventes par genre et le nombre de jeux lancés**. Comme l'indique le graphique ci-dessus, certains genres se retrouvent dans les top ventes alors que peu de jeux associés ont été lancés. C'est le cas des jeux de tirs (top 3 des genres niveau ventes) et des jeux de plateformes.

Au contraire, le genre *Aventure* se retrouve dans les genres les moins performants niveau ventes mais avec un niveau de jeux lancés assez élevé.

3.7. Certains genres sont mieux notés que d'autres



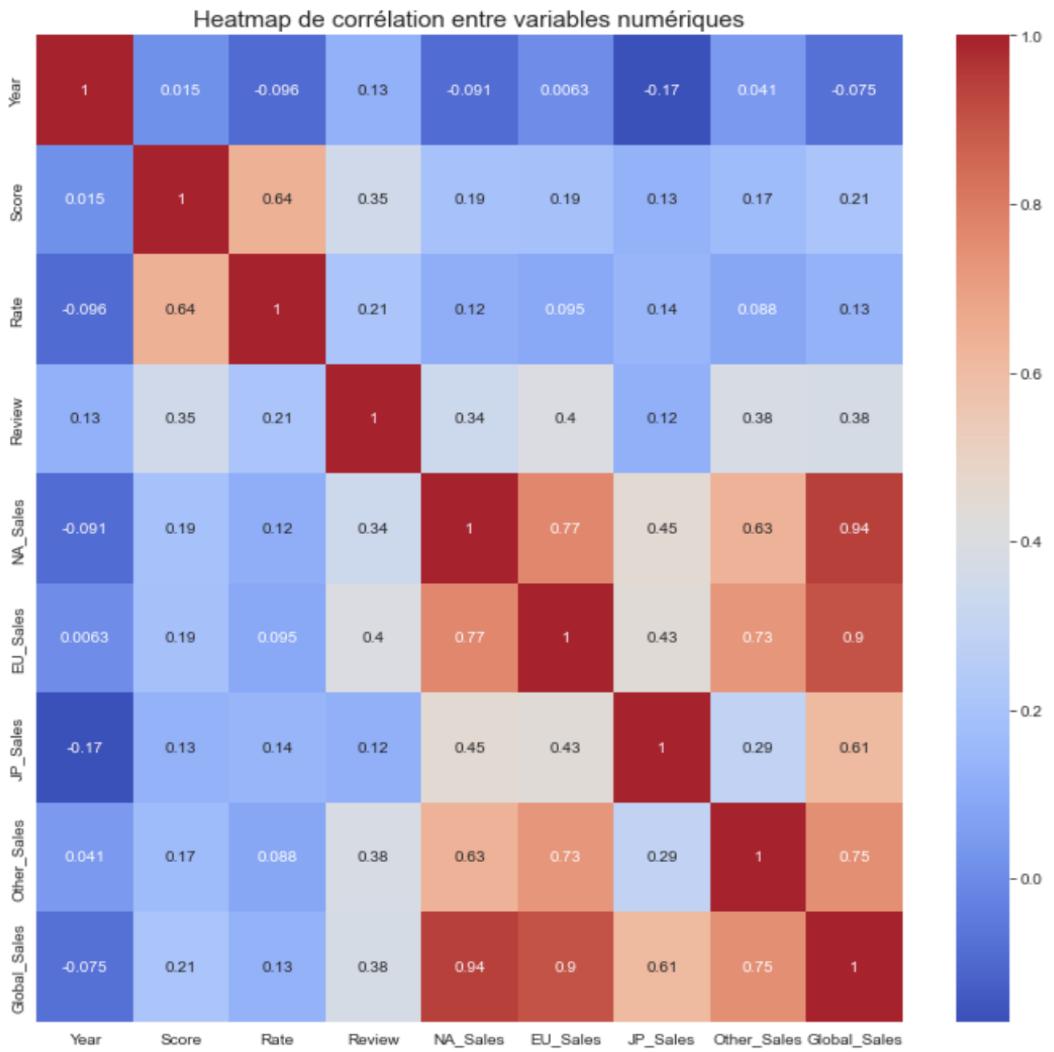
Après avoir mis en relation le niveau de ventes avec le nombre de jeux lancés par genre, il était intéressant de comparer également le niveau de ventes avec la moyenne des notes joueurs et journalistes.

L'échelle des **notes oscille entre 6.4 et 7**, ce qui montre qu'il y a **peu de variations** entre les notes moyennes et qu'il n'y a pas un grand écart entre les genres.

Cependant, nous voyons que certains genres **sont un peu mieux notés que d'autres et que cela n'a pas de lien avec le niveau de ventes** du genre en question.

Par exemple, le genre cumulant le plus de ventes fait partie des genres les moins bien notés de notre dataset alors que le genre *Strategy* avec le moins de ventes obtient la 2ème meilleure note moyenne.

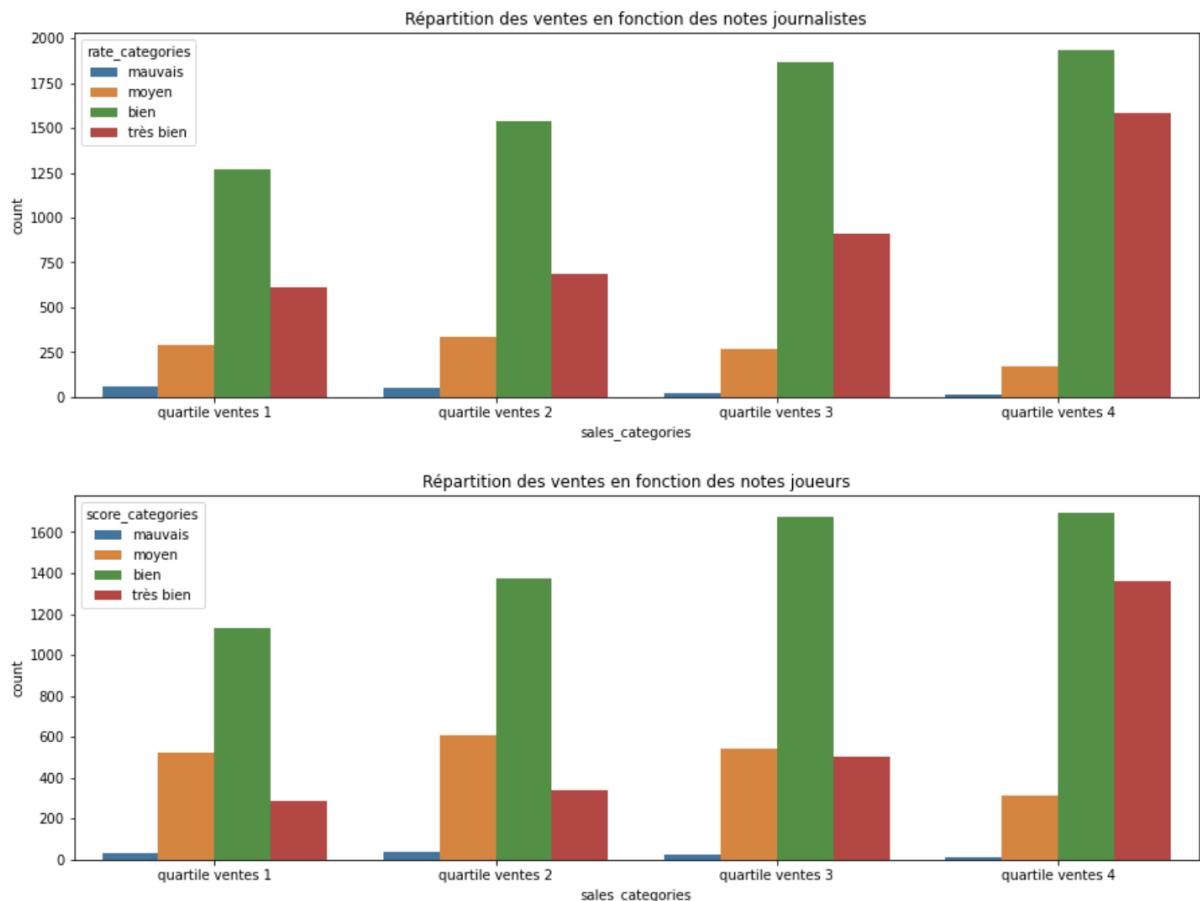
3.8. Certaines variables sont corrélées entre elles et surtout au niveau de ventes



La matrice de corrélation nous permet de voir que **peu de variables sont corrélées entre elles et surtout avec les ventes** (à part les variables liées aux ventes entre elles). Seul le nombre d'avis se démarque avec une corrélation à 0.4 sur les ventes de la région européenne et 0.38 avec les ventes globales. Cela **réfère assez bien les observations précédentes** sur les ventes des plateformes et des genres.

Nous allons voir sur les prochaines hypothèses si nous avons plus de chance en croisant les données de ventes avec celles des licences et des studios.

3.9. Les jeux les mieux notés sont les plus vendus



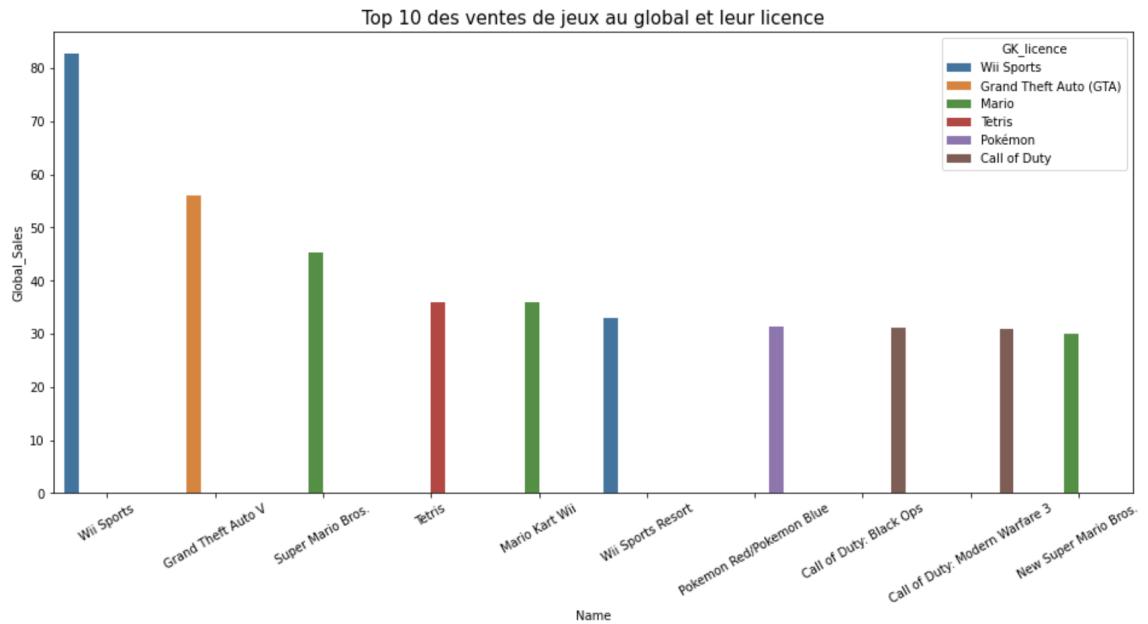
Nous observons ici que les notes les plus données par les joueurs et les journalistes sont des notes de **catégorie “bien”**, c'est à dire entre 5 et 7,5 sur 10. Cette dernière catégorie de notes **est majoritaire sur toutes les catégories de ventes**.

Nous pouvons voir que pour les notes journalistes et les notes de joueurs, plus la catégorie de ventes est haute, plus les mauvaises notes diminuent et les bonnes notes augmentent. Néanmoins, nous observons **parfois peu de différence sur les notes données entre deux catégories de ventes**. Par exemple, le nombre de jeux avec une note “moyenne” reste à peu près similaire sur les 3 premières catégories de ventes. Tout comme le nombre de jeux “bien” entre les catégories 3 et 4.

Les journalistes semblent mieux répartir le nombre de très bonnes notes contrairement aux joueurs qui les réservent plus pour les jeux les plus vendus.

Globalement, nous pouvons quand même dire que **les jeux les plus vendus ont plus de très bonnes notes et moins de mauvaises et moyennes notes**.

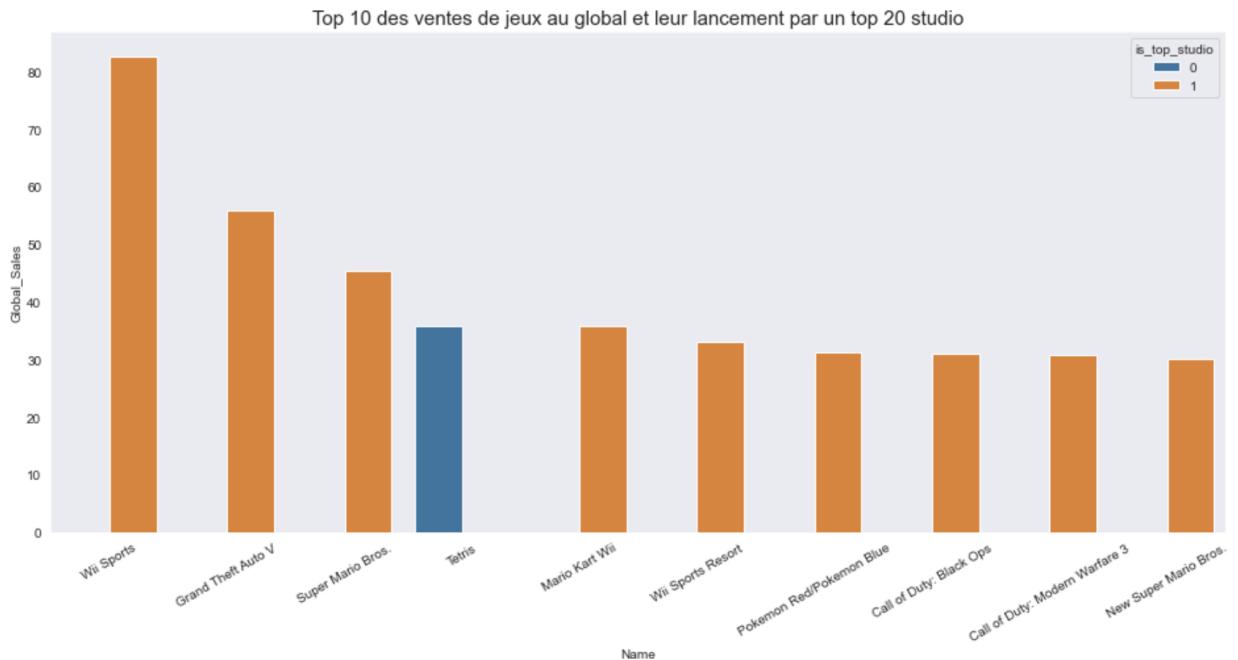
3.10. Les jeux qui font partie d'une licence sont les plus vendus



Nous pouvons clairement conclure ici que **les jeux rattachés à une licence (série de jeux) sont mieux vendus** que des jeux indépendants car parmi le **top 10** des jeux avec les meilleures ventes de notre dataset, tous **font partie d'une licence**.

Ainsi, nous pouvons garder en tête que **le fait d'être rattaché à une série pour un jeu est un bon attribut pour sa vente**.

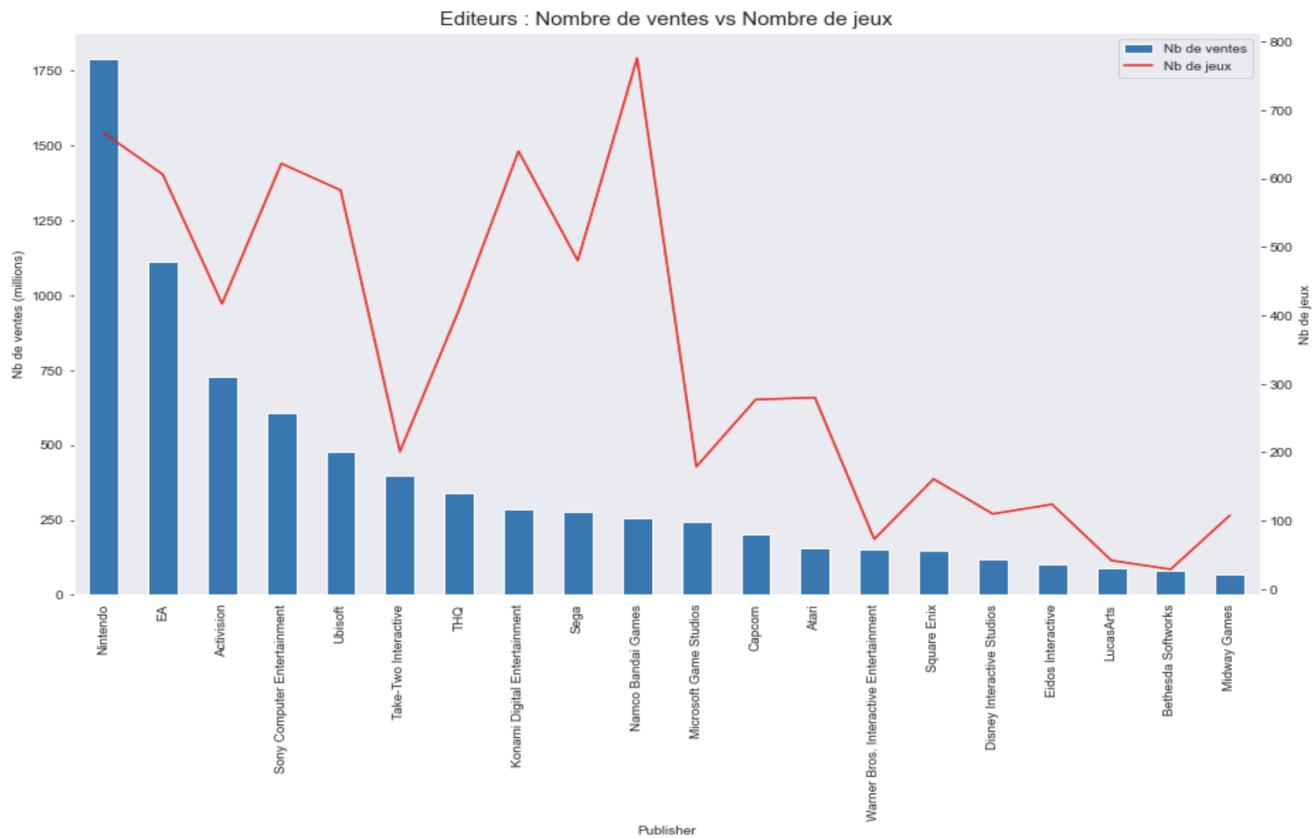
3.11. Les jeux qui comptabilisent le plus de ventes ont été lancés par les meilleurs studios



En reprenant le top 10 des jeux les plus vendus du dataset et en regardant s'ils sont rattachés au 20 studios comptabilisant les meilleures ventes (encore une fois du dataset), nous voyons que **9 jeux sur 10 valident l'hypothèse**. Seul le jeu *Tétris* a été lancé par un studio comptabilisant moins de ventes.

Nous pouvons en déduire que l'influence et la notoriété d'un studio a donc également des impacts positifs sur les ventes d'un jeu vidéo.

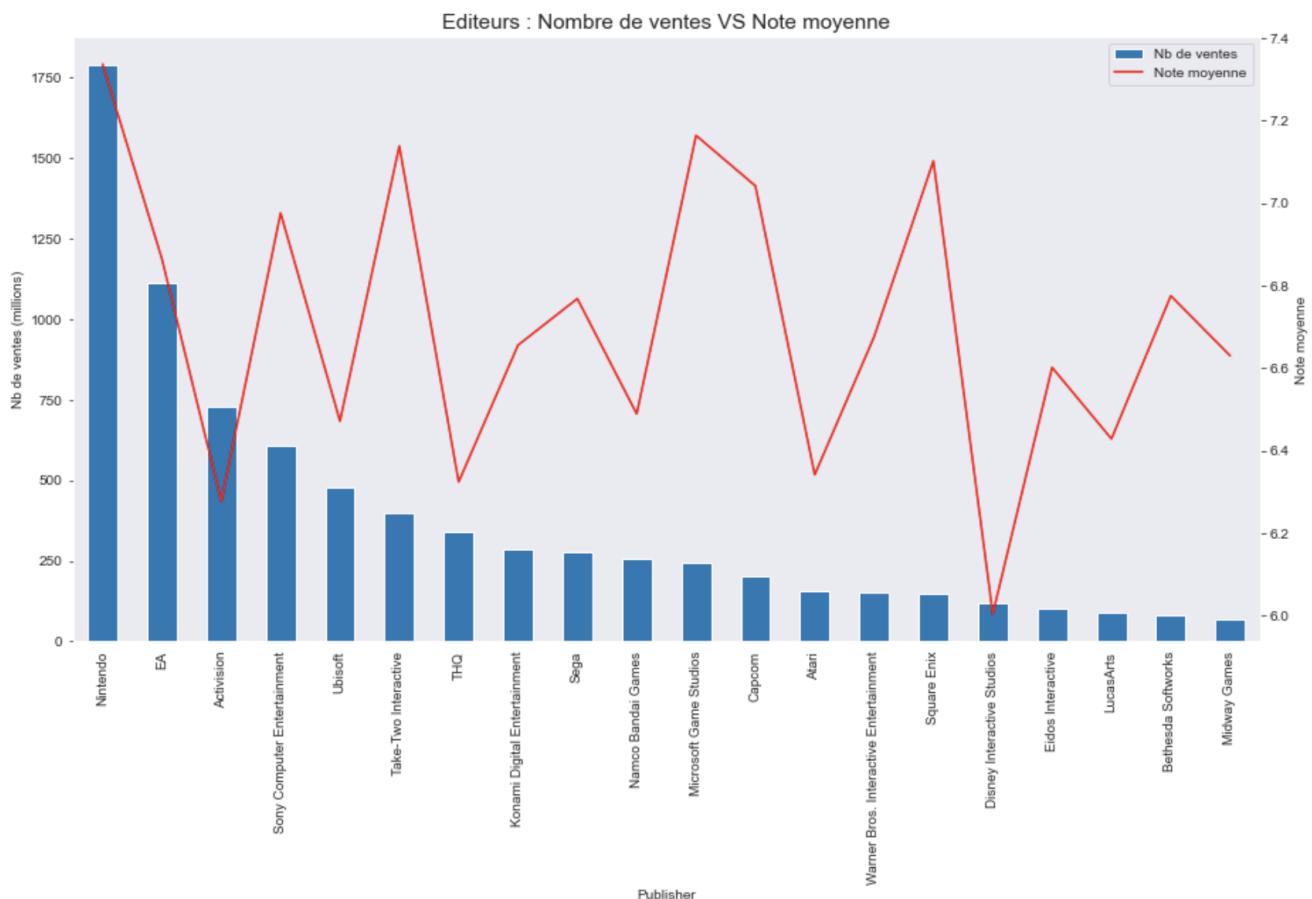
3.12. Les éditeurs qui font le plus de ventes produisent le plus de jeux



Nous voyons qu'il y a pour certains éditeurs de **grandes disparités entre le nombre de jeux produits et les ventes cumulées au globales**. Ce qui saute directement aux yeux, c'est par exemple l'éditeur **Namco Bandai Games**, qui rentre à peine dans le top 10 des éditeurs générant le plus de ventes et qui pourtant a **produit près de 800 jeux** (200 de plus que Nintendo qui a les meilleures ventes).

Nous voyons donc par ce graphique que **le nombre de jeux produit par un éditeur n'est pas représentatif du nombre de ventes générées**.

3.13. Les éditeurs avec le plus de ventes sont ceux dont les jeux ont les meilleures notes moyennes

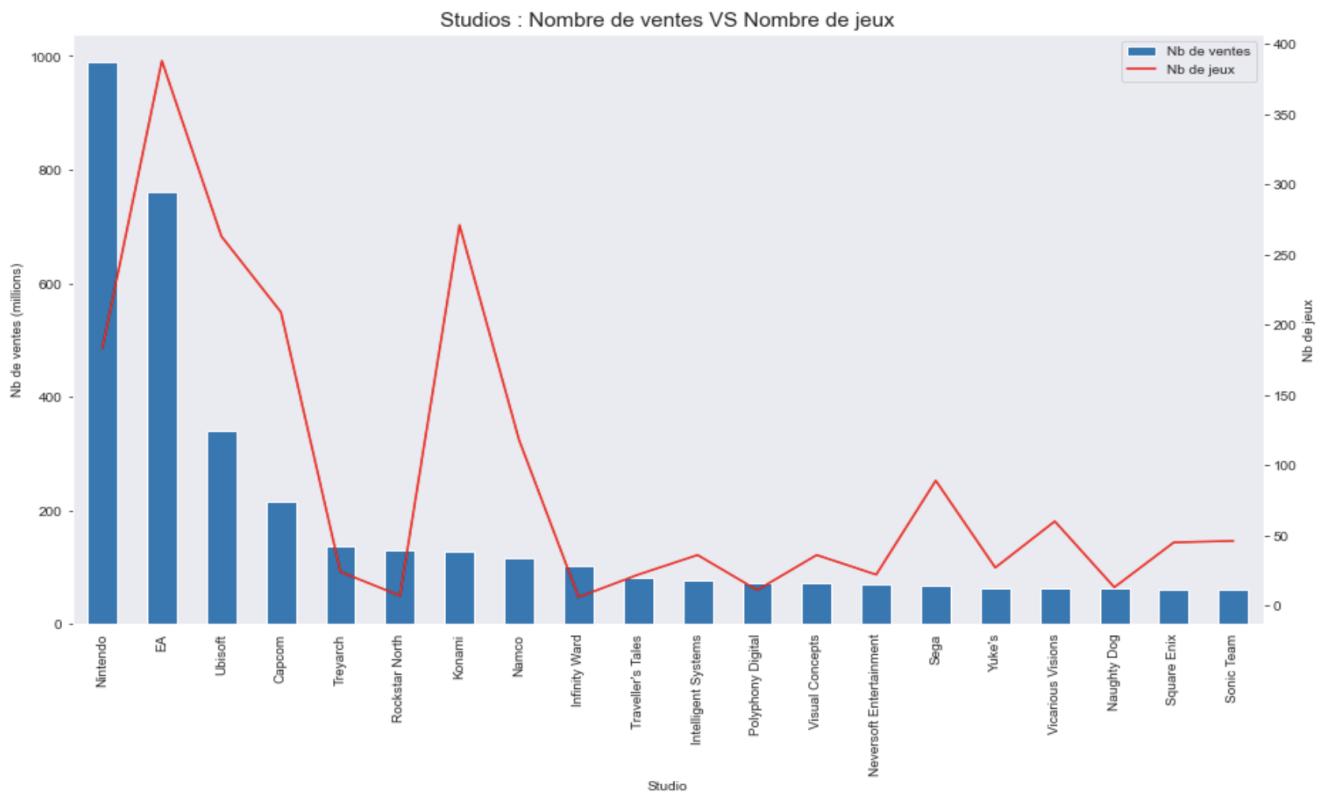


L'échelle des notes varie entre 6 et 7.4 sur 10. Ce qui veut dire qu'il n'y a pas de grandes différences sur les notes moyennes globales en fonction des éditeurs.

Dans le détail, nous pouvons quand même voir qu'**aucun lien est établi entre l'éditeur du jeu et la note moyenne donnée** par les joueurs et les journalistes vu le comportement de la courbe.

Certains éditeurs ont une note moyenne sur leurs jeux plus cohérente avec le niveau de ventes généré. C'est le cas pour *Nintendo*, meilleures ventes et meilleure note moyenne. Et moins positivement c'est le cas aussi pour *Disney Interactive Studios*, qui arrive dans les derniers niveaux ventes et avec la note moyenne la plus basse.

3.14. Les studios qui font le plus de ventes lancent le plus de jeux

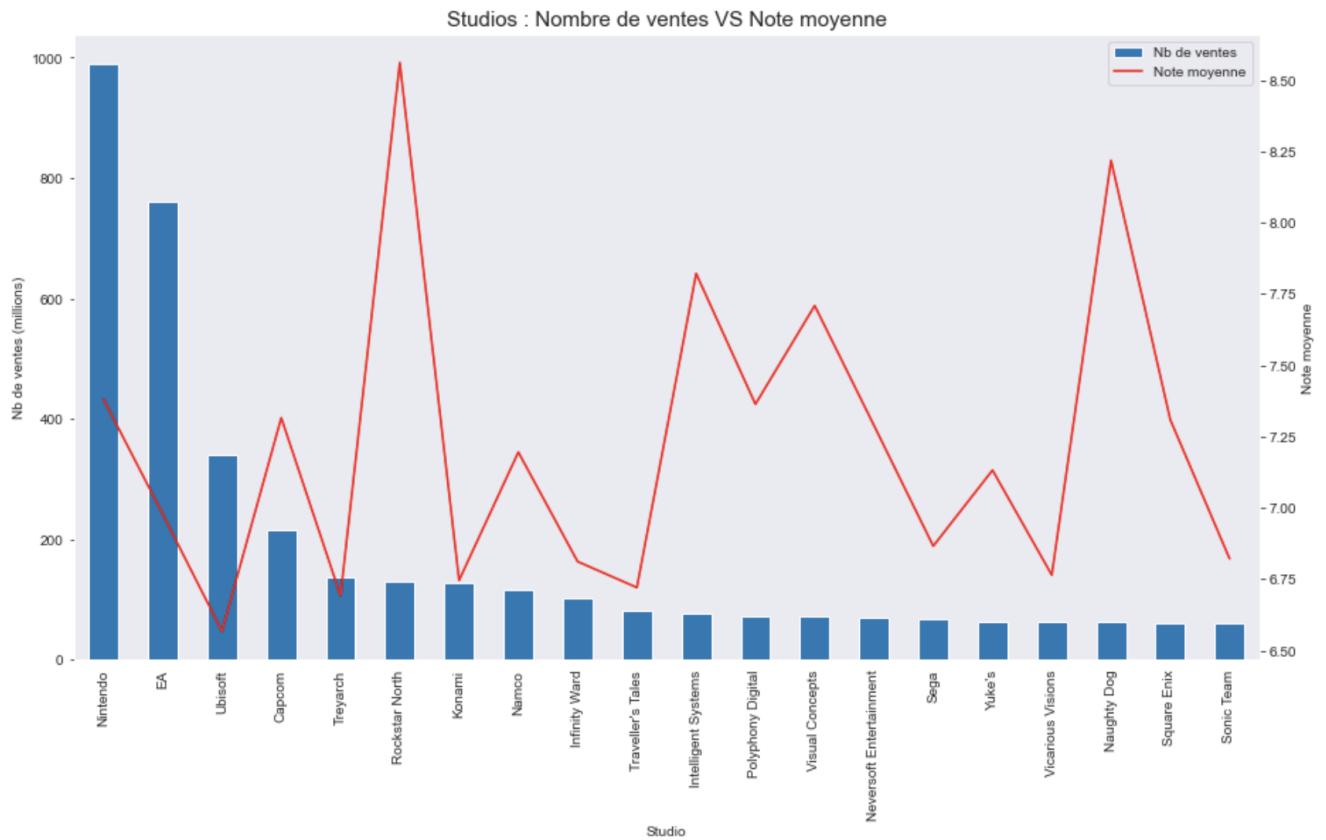


Sur les derniers graphiques construits, c'est celui-ci qui **semble montrer une meilleure relation entre les variables analysées**. A l'exception de *Nintendo* et des deux autres studios japonais, *Konami* et *Namco*, la courbe du nombre de jeux lancés pourrait suivre de près les barres du niveau de ventes.

La relation nombre de jeux lancés vs nombre de ventes semble donc **plus forte pour les studios** par rapport aux éditeurs comme nous l'avons vu précédemment.

Malgré cela, nous ne pouvons **pas affirmer qu'il existe une vraie corrélation** car nous voyons que **pour plusieurs studios, il existe un décalage** entre le nombre de jeux lancés et le niveau de ventes générées.

3.15. Les studios avec le plus de ventes sont ceux dont les jeux ont les meilleures notes moyennes



Comme pour les éditeurs, l'**échelle de notes est serrée** avec des notes entre 6.5 et 8.5 sur 10. Ici aussi, nous voyons donc qu'il n'y a pas une grande différence entre les studios et que **les notes moyennes sont plutôt bonnes peu importe le niveau de ventes.**

En analysant plus précisément la courbe, **nous voyons que la note moyenne et le niveau de ventes vivent de manière indépendante** et qu'aucune des deux variables peut nous aider à prédire l'autre.

Conclusion sur la data visualisation et les 15 hypothèses établies

- **Ventes concentrées entre 1995 et 2015** mais de moins en moins de jeux présents dans le dataset après 2010
- **Fausse saisonnalité** du marché sur la fin de l'année car c'est une période forte pour le lancement des jeux. Les mois avec un **meilleur ratio nombre de ventes vs nombre de lancements** semblent être **mars et juin**.
- **La majorité des ventes du dataset est drainée par la région Amérique du Nord** qui a un quasi monopole au début des années 90. Mais nous voyons que les autres régions prennent de plus en plus de parts de marché au fil des années
- Les régions du monde ont les **mêmes préférences sur les genres des jeux**. Sauf le **Japon** qui semble être un pays qui **raffole des jeux de rôles**.
- **En tête des genres** les plus vendus dans le monde, nous retrouvons : les **jeux d'action**, les **jeux de sport** et les **jeux de tirs**.
- Une **tendance se dessine grossièrement** sur le fait qu'une **plateforme avec plus de ventes connaît plus de lancement de jeux**. Néanmoins certaines plateformes ne répondent pas à cette tendance et la relation entre les deux variables est quand même fragile.
- **Pas de lien marqué** entre le **niveau de ventes d'un genre et le nombre de lancements de jeux dans ce genre**. Nous voyons que certains genres surperforment niveau ventes par rapport au nombre de jeux lancés. C'est le cas des jeux de tirs et de plateformes.
- Les notes moyennes données par les joueurs et les journalistes **par genre** sont très serrées mais **les notes ne sont pas croissantes ou décroissantes en fonction des ventes**.
- **Pas de corrélations** marquées sur les **variables numériques entre elles et surtout avec celles liées aux ventes**
- **Plus de bonnes notes** parmi les **jeux les mieux vendus** mais ici encore beaucoup d'exceptions qui ne nous permettent pas d'affirmer une relation entre les deux variables.
- Un **lien semble fort** entre le **niveau de ventes d'un jeu et le fait qu'il soit rattaché à une série de jeux**.
- Relation établie aussi entre **les ventes d'un jeu et le fait qu'il ait été lancé par un des 20 studios cumulant le plus de ventes**
- Que ce soit pour les éditeurs ou les studios, il ne semble **pas y avoir de vérité sur le fait qu'ils produisent/lancent des jeux proportionnellement à leurs ventes**. Les éditeurs et studios qui **réalisent le plus de ventes n'ont pas non plus une meilleure note moyenne** sur les jeux produits/lancés par rapport aux autres.

4. 2ème phase d'enrichissement

4.1. Création des moyennes mobiles :

Pour préparer nos données au machine learning, nous devons écarter certains biais qui peuvent avoir une influence sur l'algorithme d'apprentissage comme les notes (des joueurs et des testeurs) et le nombre d'avis des jeux. En effet, ces informations sont la plupart du temps données après la sortie d'un jeu et sont potentiellement influencées par le nombre de ventes de ce jeu.

D'autre part, pour entraîner un modèle dont la cible est de prédire le nombre de ventes des jeux, nous devons évidemment écarter les variables de ventes du dataset de test.

Pour éviter ces influences biaisées tout en conservant les informations que ces variables pouvaient nous apporter, nous les avons transformées en moyennes mobiles pour pouvoir les intégrer dans notre dataset de la manière suivante :

- classement du dataset par éditeur (*Publisher*) et par date de sortie
- création de 4 moyennes mobiles : nombre de ventes, note journaliste, note joueurs, nombre d'avis
- pour le 1er jeu d'un éditeur : les 4 moyennes mobiles sont mises à 0 (les moyennes mobiles nous servant à utiliser les données du passé pour voir l'influence sur le jeu qui va sortir, si le jeu est le 1er d'un éditeur, il n'y a donc aucune vente dans le passé)
- pour le 2ème jeu d'un éditeur : les 4 moyennes mobiles sont égales aux valeurs des 4 variables du 1er jeu (moyenne mobile du nombre de ventes du jeu 2 = nb de ventes du jeu 1 ; même principe pour les 3 autres moyennes mobiles)
- à partir du 3ème jeu : les moyennes mobiles sont égales à la moyenne des moyennes mobiles des jeux n-1 et n-2.

Nous avons procédé de la même manière pour calculer les moyennes mobiles par studio et par licence en changeant seulement la valeur de la moyenne mobile du 1er jeu :

Dans l'industrie du jeu vidéo, l'éditeur englobe un studio qui est lui-même englobe une licence. Donc la moyenne mobile du 1er jeu d'un studio prend la valeur de la moyenne mobile de l'éditeur correspondant; et celle du 1er jeu d'une licence, celle de la moyenne mobile du studio correspondant.

→ Gestion des valeurs manquantes des moyennes mobiles

À ce stade, voici l'état de notre dataset et de ses valeurs manquantes :

#	Column	Non-Null Count	Dtype
0	Name	16538 non-null	object
1	Platform	16538 non-null	object
2	Year	16538 non-null	object
3	Genre	16538 non-null	object
4	Publisher	16495 non-null	object
5	Studio	13917 non-null	object
6	Score	10441 non-null	float64
7	Rate	11556 non-null	float64
8	Review	9928 non-null	float64
9	NA_Sales	16538 non-null	float64
10	EU_Sales	16538 non-null	float64
11	JP_Sales	16538 non-null	float64
12	Other_Sales	16538 non-null	float64
13	Global_Sales	16538 non-null	float64
14	GK_licence	10863 non-null	object
15	GK_distributeur	2415 non-null	object
16	Mois	16538 non-null	object
17	Date_Sortie	16538 non-null	datetime64[ns]
18	RM_Publisher	16495 non-null	float64
19	RM_Publisher_score	10752 non-null	float64
20	RM_Publisher_rate	11839 non-null	float64
21	RM_Publisher_reviews	10258 non-null	float64
22	RM_Studio	16512 non-null	float64
23	RM_Studio_score	12311 non-null	float64
24	RM_Studio_rate	13493 non-null	float64
25	RM_Studio_reviews	12437 non-null	float64
26	RM_Licence	10857 non-null	float64
27	RM_Licence_score	7548 non-null	float64
28	RM_Licence_rate	8561 non-null	float64
29	RM_Licence_reviews	7738 non-null	float64

Avec la même logique que pour la création des rolling means (publisher > studio > licence), on remplace les valeurs manquantes des 4 rolling means des publishers par les valeurs correspondantes des rolling means des studios. Puis les valeurs des 4 rolling means des studios par les valeurs correspondantes des rolling means des licences.

On fait ensuite la même chose dans l'autre sens : les valeurs manquantes des rolling means des licences sont remplacées par les valeurs correspondantes des rolling means des studios; puis les valeurs manquantes des studios par les valeurs des rolling means des publishers.

Les derniers NaN restants des moyennes mobiles sont remplacés par les moyennes des valeurs de ces variables dont la date est strictement antérieure.

Exemple pour les moyennes mobiles des scores, les derniers NaN pour les moyennes mobiles des rates et des reviews sont traitées de la même manière.

```
1 rm_publisher_score=[]
2 for date,rm in zip(df_rm_tosplit["Date_Sortie"],df_rm_tosplit["RM_Publisher_score"]):
3     if str(rm)=="nan":
4         rm_publisher_score.append(df_rm_tosplit["RM_Publisher_score"] [df_rm_tosplit["Date_Sortie"]
5             <str(date)].mean())
6     else:
7         rm_publisher_score.append(rm)
8
9
10 rm_studio_score=[]
11 for date,rm in zip(df_rm_tosplit["Date_Sortie"],df_rm_tosplit["RM_Studio_score"]):
12     if str(rm)=="nan":
13         rm_studio_score.append(df_rm_tosplit["RM_Studio_score"] [df_rm_tosplit["Date_Sortie"]
14             <str(date)].mean())
15     else:
16         rm_studio_score.append(rm)
17
18 rm_licence_score=[]
19 for date,rm in zip(df_rm_tosplit["Date_Sortie"],df_rm_tosplit["RM_Licence_score"]):
20     if str(rm)=="nan":
21         rm_licence_score.append(df_rm_tosplit["RM_Licence_score"] [df_rm_tosplit["Date_Sortie"]
22             <str(date)].mean())
23     else:
24         rm_licence_score.append(rm)
25
26 df_rm_tosplit["RM_Publisher_score"]=rm_publisher_score
27 df_rm_tosplit["RM_Studio_score"]=rm_studio_score
28 df_rm_tosplit["RM_Licence_score"]=rm_licence_score
```

4.2 Scraping d'informations avant lancement des jeux :

L'analyse approfondie de notre dataset, nous a montré que peu de variables parmi celles présentes dans le dataset et celles que nous avons scrappées, avaient un lien fort avec les ventes nous permettant de les prédire.

Les modèles de régression que nous voulons développer peuvent être exigeants sur la corrélation qu'il doit y avoir entre les variables..

Nous sommes donc repartis à la recherche de données disponibles **avant le lancement d'un jeu vidéo**, nous permettant d'établir une relation avec le niveau de ventes des jeux.

4.2.1. Association à une série de jeux :

Le premier scraping sur Gamekult a déjà permis de récupérer la licence des jeux, ce qui nous a aidé à voir au moment de la visualisation que les jeux les plus vendus faisaient partie d'une série.

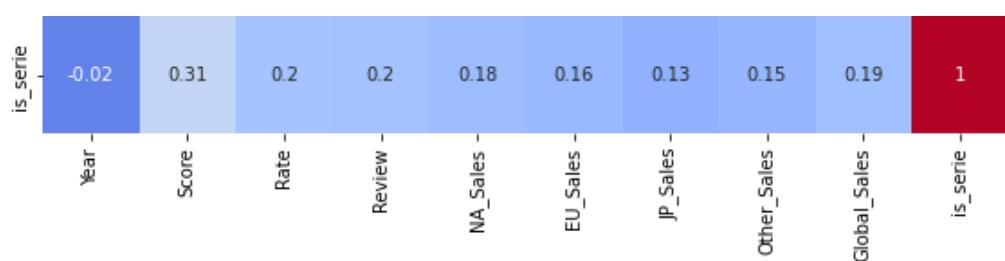
Néanmoins, nous avons trouvé une autre source de données, qui complètent cette première donnée sur les licences et qui enrichissent le dataset d'une autre façon : 1

si le jeu fait partie d'une série, 0 sinon. Cela contrairement à la donnée de licence scrapée sur Gamekult, qui donne le nom de la licence à laquelle le jeu est rattaché.

Une page de Wikipédia, nous a permis de sortir une liste de toutes les séries de jeux qui existent dans le monde et de comparer les noms des jeux du dataset afin de voir la correspondance avec les séries scrappées.

Nous avons identifié plus de 4700 lignes avec des jeux appartenant à une série.

Une nouvelle heatmap nous permet de voir que la corrélation avec les ventes n'est pas nulle mais assez éloignée de 1 quand même. Le faible volume de lignes avec une valeur 1 par rapport au nombre de lignes du dataset (16 617 lignes) n'aide pas à maximiser ce score.



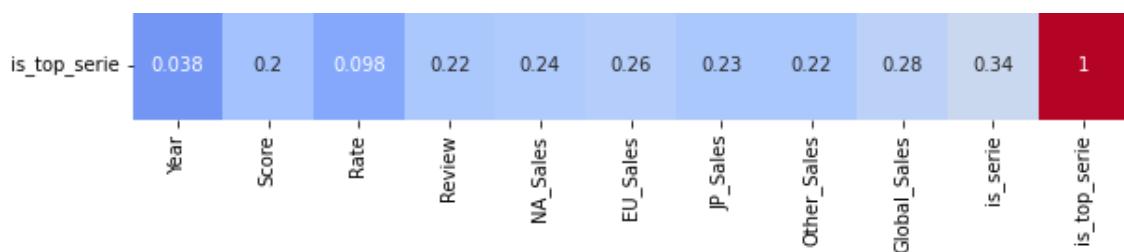
4.2.2. Séries les + vendues :

Nous avons trouvé une page sur Wikipédia répertoriant les franchises de jeux les plus vendues dans l'histoire du jeu vidéo à travers le monde.

Sur cette page, 43 franchises sont listées. Nous avons comparé ces franchises avec les noms des jeux vidéos disponibles dans notre dataset et nous avons créé une nouvelle colonne *is_top_series*. Cette colonne porte deux valeurs : 0 si le jeu n'est pas rattaché par son nom à une franchise citée sur la page et 1 si le jeu fait partie d'une série à succès listée sur la page.

Par exemple, Mario fait partie des franchises listées parmi les franchises les plus vendues dans le monde. Tous les jeux dont les noms contiennent Mario (Mario Kart, Mario Super Smash Bros, Mario Party, etc.) portent la valeur 1 dans la colonne *is_top_series*.

1230 lignes ont été labellisées à 1 sur cette nouvelle variable *is_top_series* et nous obtenons un score de corrélation plus élevé à 0.28.



4.2.3. Association aux meilleurs studios :

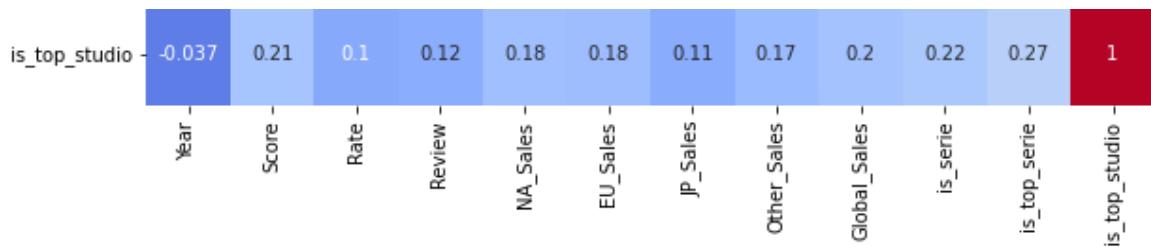
Pour l'ajout de cette nouvelle variable, nous n'avons pas fait de scraping mais une fonction *groupby* pour identifier les 20 studios réalisant le plus de ventes a suffit.

Nous avions fait l'exercice dans la data visualisation de voir si les jeux les plus vendus avaient été lancés par les studios réalisant le plus de ventes. Sur les 10 jeux les plus vendus, seul 1 n'a pas été lancé par un studio du top 20.

Le fait d'être lancé par un gros studio pouvant donc avoir un impact sur les ventes des jeux, nous avons donc ajouté cette variable à notre dataset en préparation du machine learning.

Nous avons donc créé une nouvelle colonne *is_top_studio* dans notre dataset et pour chaque ligne de notre dataset, cette colonne porte la valeur 1 si le studio est un studio du top 20 ou 0 sinon.

Presque 3200 lignes ont été identifiées avec des studios faisant partie du top 20 et nous obtenons un score de corrélation avec les ventes globales à 0.2.



4.2.4. Annonces au festival E3 :

De la même manière que précédemment, nous avons trouvé sur Wikipédia les jeux cités pour chaque année du festival E3 - Electronic Entertainment Expo.

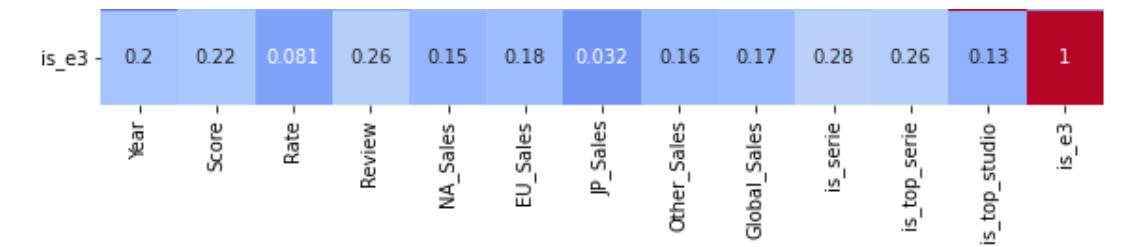
Ce festival a lieu tous les ans aux Etats-Unis et regroupe les plus grands studios de jeux vidéo. Chaque studio annonce à ce moment là ses gros lancements de jeux vidéo sur l'année à venir et des experts, qui ont pu tester les jeux en avant première récompensent nommément des jeux dans des catégories spécifiques (meilleur design, innovations, meilleur scénario, etc.).

Pour chaque année disponible, nous avons récupéré sur Wikipédia les jeux qui avaient été cités lors du festival E3, que ce soit l'annonce d'un lancement ou la nomination dans une certaine catégorie.

L'idée étant qu'un jeu cité d'une quelconque façon à ce festival, suivi par des millions de fans, allait faire le buzz au moment de sa sortie et potentiellement générer beaucoup de ventes.

Après comparaison de la liste de jeux ayant été cités au festival E3 avec la liste des jeux du dataset, 2284 lignes pour lesquelles les jeux avaient été nommés au festival ont été identifiées.

Ici encore, le volume de lignes avec une valeur 1 est faible pour obtenir une score de corrélation élevé. Avec la heatmap mise à jour, nous obtenons un score à 0.17.



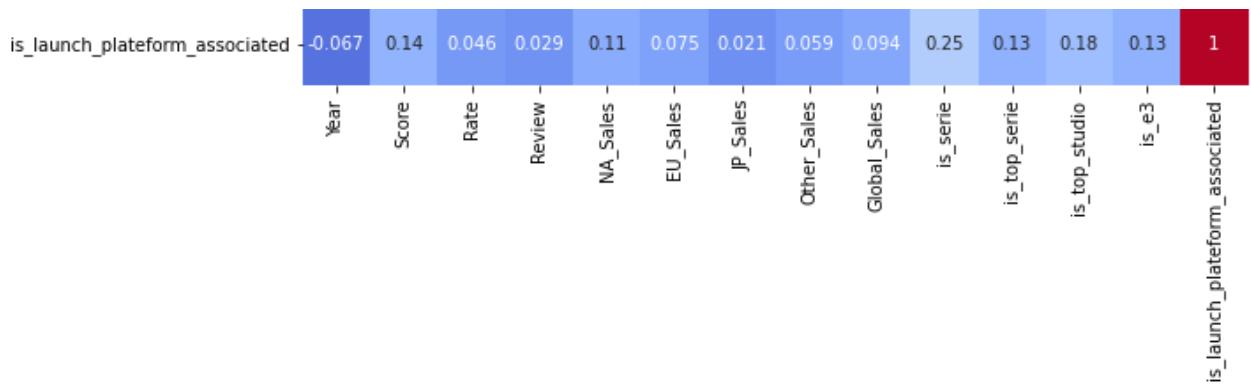
4.2.5. Lancement des jeux en même temps que le lancement d'une console :

Nous nous sommes posés la question de savoir si certaines ventes de jeux n'étaient pas influencées par le fait que les jeux étaient lancés en même temps qu'une nouvelle console.

Nous avons trouvé une autre page intéressante de Wikipédia répertoriant tous les jeux lancés en simultanés d'une console en fonction des régions.

Nous avons scrappé cette liste de jeux et nous l'avons fait correspondre avec les jeux de notre dataset. Nous avons créé une nouvelle colonne *is_launch_plateform_associated* dans laquelle nous avons mis la valeur 1 si le jeu était reconnu dans la liste des jeux associés au lancement d'une console et 0 sinon.

Nous avons eu plus de 1600 lignes dont la valeur de la colonne *is_launch_plateform_associated* était égale à 1 soit 10% de notre dataset environ.



Sur cette dernière variable le score est très bas et nous laisse penser que le lancement d'un jeu en simultané d'une console n'influe pas sur son niveau de ventes.

Toutes les pages Wikipedia scrapées sont répertoriées dans les sources ci-dessous.

5. Modélisation

5.1 Pré-processing :

L'objectif de ce projet est de mettre en place un prédicteur en se servant des données déjà en notre possession pour alimenter un algorithme de machine learning. Les données sont donc au cœur de notre démarche et constituent un élément important pour l'atteinte de notre objectif. La première étape de notre processus de modélisation a été de traiter les données pour qu'elles soient mieux interprétées par les algorithmes de Machine Learning. Ce qui est appelé le pré-processing des données. Cette étape comprend la gestion des dernières valeurs manquantes et la normalisation et la standardisation de nos données.

5.1.1. Gestion des valeurs manquantes :

Gestions des valeurs manquantes des variables *Publisher*, *Studio* et *GK_Licence*

- remplacement des Licences manquantes par le nom du jeu correspondant
- remplacement des Studios manquants par la licence correspondante
- remplacement des Publisher manquants par le studio correspondant

5.1.2. Encodage des données :

Pour optimiser le fonctionnement des algorithmes de machine learning sur nos données, il est nécessaire d'encoder les variables catégorielles afin d'obtenir un jeu de données avec des variables entièrement numériques. À chaque valeur unique des variables a été attribué un entier à partir de 1. Les occurrences de ces valeurs ont été remplacées dans le jeu de données.

```
cat_cols = df_rm_tosplit.select_dtypes(include=['object']).columns.to_list()
for col in cat_cols:
    i = 1
    occurrences = df_rm_tosplit[col].unique()
    for occ in occurrences:
        df_rm_tosplit.loc[df_rm_tosplit[col]==occ, col] = i
        i+=1

df_rm_tosplit[cat_cols] = df_rm_tosplit[cat_cols].astype('float')
df_rm_tosplit.info()
```

→ Visualisation de notre dataset final après pré-processing

```
Data columns (total 31 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   Name             16538 non-null  object  
 1   Platform         16538 non-null  object  
 2   Year             16538 non-null  int64   
 3   Genre            16538 non-null  object  
 4   Publisher        16538 non-null  object  
 5   Studio            16538 non-null  object  
 6   NA_Sales          16538 non-null  float64 
 7   EU_Sales          16538 non-null  float64 
 8   JP_Sales          16538 non-null  float64 
 9   Other_Sales       16538 non-null  float64 
 10  Global_Sales     16538 non-null  float64 
 11  GK_licence        16538 non-null  object  
 12  Mois              16538 non-null  int64   
 13  Date_Sortie       16538 non-null  object  
 14  RM_Publisher      16516 non-null  float64 
 15  RM_Publisher_score 14347 non-null  float64 
 16  RM_Publisher_rate 14347 non-null  float64 
 17  RM_Publisher_reviews 13076 non-null  float64 
 18  RM_Studio          16516 non-null  float64 
 19  RM_Studio_score    14347 non-null  float64 
 20  RM_Studio_rate     14347 non-null  float64 
 21  RM_Studio_reviews   13076 non-null  float64 
 22  RM_Licence         16516 non-null  float64 
 23  RM_Licence_score    14347 non-null  float64 
 24  RM_Licence_rate     14347 non-null  float64 
 25  RM_Licence_reviews   13076 non-null  float64 
 26  is_serie           16538 non-null  int64   
 27  is_top_serie       16538 non-null  int64   
 28  is_top_studio       16538 non-null  int64   
 29  is_e3               16538 non-null  int64   
 30  is_launch_plateform_associated 16538 non-null  int64 

dtypes: float64(17), int64(7), object(7)
memory usage: 4.0+ MB
```

5.2. Choix du modèle de machine learning

Notre jeu de données se prête plutôt à un modèle de régression car notre objectif est de prédire exactement le nombre d'exemplaires vendu d'un jeu vidéo avant sa sortie (donc une valeur continue).

Regardons donc les résultats que nous obtenons avec ce type de modèle.

5.2.1. Régression :

→ Séparation en jeux d'entraînement et de test

Notre variable cible dans ce projet est ***Global_Sales*** qui représente le nombre d'exemplaires vendus.

Nous supprimons ensuite les variables devenues inutiles (*Name*, *NA_Sales*, *EU_Sales*, *JP_Sales*, *Other_Sales* et *Date_Sortie*) pour obtenir notre échantillon X.

Nous séparons enfin notre dataset en jeux d'entraînement et de test pour que notre jeu d'entraînement représente 80% de notre dataset global.

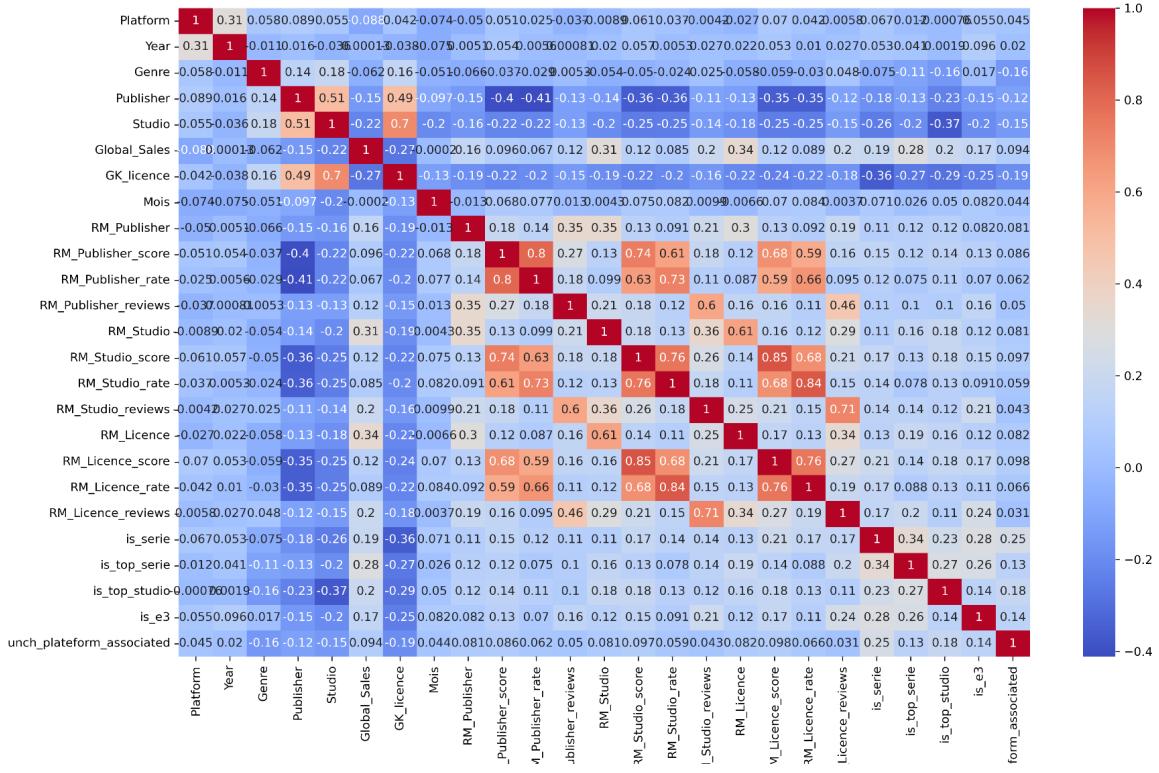
→ Gestion des dernières valeurs manquantes

Pour remplir nos dernières valeurs manquantes et pour ne pas faire fuiter de données dans l'échantillon de test, nous procédons de la sorte :

- Dans l'échantillon d'entraînement uniquement, calcul des moyennes des valeurs antérieures chronologiquement pour remplir les NaN de cet échantillon
- Une fois l'échantillon d'entraînement totalement comblé, calcul de la moyenne globale de toutes les rolling means
- Utilisation de ces moyennes pour combler les valeurs manquantes dans l'échantillon de test.

→ Évaluation du modèle

Dans le cadre de la régression, les corrélations entre les variables explicatives et notre variable cible est primordiale afin d'avoir un avant-goût dans la fonction de prédiction de la variable cible.



Cette visualisation montre que les variables explicatives ne sont pas vraiment très corrélées à notre variable cible. Ce qui n'augure pas des performances énormes avec les algorithmes de régression.

La seconde étape consiste à évaluer plusieurs algorithmes de régression afin de trouver le plus performant pour notre problématique. Les algorithmes que nous avons sélectionnés sont les suivants :

- LinearRegression
- DecisionTreeRegressor
- Lasso
- SVR
- RandomForestRegressor

```

from sklearn.linear_model import LinearRegression,Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn import linear_model
from sklearn.ensemble import RandomForestRegressor

lr=LinearRegression()
lr.fit(X_train,y_train)
y_pred_test=lr.predict(X_test)
print(" lr score :" , lr.score(X_test,y_test))

dt=DecisionTreeRegressor()
dt.fit(X_train,y_train)
y_pred_test=dt.predict(X_test)
print(" dt score :" , dt.score(X_test,y_test))

lm=Lasso(alpha=0.1)
lm.fit(X_train,y_train)
y_pred_test=lm.predict(X_test)
print(" lm score :" , lm.score(X_test,y_test))

svr=SVR()
svr.fit(X_train,y_train)
y_pred_test=svr.predict(X_test)
print(" svr score :" , svr.score(X_test,y_test))

rf=RandomForestRegressor()
rf.fit(X_train,y_train)
y_pred_test=rf.predict(X_test)
print(" rf score :" , rf.score(X_test,y_test))

```

```

lr score : 0.07676634286080508
dt score : -1.8371242850441543
lm score : 0.009193045108662146
svr score : 0.01669102380916354
rf score : -0.11192406364690921

```

Ces scores obtenus sont très mauvais (car très éloignés de 1) et ne permettent pas d'opter pour un modèle de régression afin d'arriver à atteindre notre objectif.

Nous décidons alors de repenser notre problématique et d'opter plutôt pour un modèle de classification.

5.2.2. Classification :

Etant donné les données dont nous disposons, il n'est pas possible d'obtenir de meilleurs résultats avec les algorithmes de régression.

À défaut, nous pouvons prédire qu'un jeu vidéo se vendra bien, moins bien ou très bien. Et dans ce cas, nous tendons plutôt vers un modèle de classification qui est plus performant dans la prédiction de variable qualitative.

Pour appliquer les algorithmes de classification, avant de séparer notre dataset en jeu d'entraînement et de test, nous devons catégoriser notre variable *Global_Sales* pour pouvoir ensuite utiliser un modèle de classification.

→ Catégorisation de la variable *Global_Sales*

Notre première réflexion a été de choisir le nombre de catégories.

Il nous a fallu choisir entre un nombre de catégories suffisamment important pour que le nombre de jeux par classe soit utilisable pour une entreprise qui souhaiterait estimer ses ventes mais pas non plus trop important pour ne pas retomber dans le biais de la régression.

Nous avons choisi 2 catégorisations (en millions d'exemplaires vendus) :

- 1ère catégorisation : 4 classes de ventes (
 - de 0 à 0.1,
 - de 0.1 à 0.249,
 - de 0.25 à 1,
 - et supérieur à 1))
- 2ème catégorisation : 6 classes de ventes (
 - de 0 à 0.1,
 - de 0.1 à 249,
 - de 0.25 à 0.499,
 - de 0.5 à 0.999,
 - de 1 à 5,
 - et supérieur à 5))

→ Séparation en jeux d'entraînement et de test

La variable ainsi obtenue est ***Sales_Cat***, qui devient notre variable cible.

Nous supprimons ensuite les variables devenues inutiles (*Name*, *NA_Sales*, *EU_Sales*, *JP_Sales*, *Other_Sales*, *Global_Sales* et *Date_Sortie*) pour obtenir notre échantillon X.

Nous séparons enfin notre dataset en jeux d'entraînement et de test pour que notre jeu d'entraînement représente 80% de notre dataset global.

Pour chacune des 2 catégorisations, notre échantillon d'entraînement est composé ainsi :

- 1ère catégorisation : 4 classes de ventes (
 - de 0 à 0.1 : 4861 lignes,
 - de 0.1 à 0.249 : 2917 lignes,
 - de 0.25 à 1 : 3795 lignes,
 - et supérieur à 1 : 1657 lignes))
- 2ème catégorisation : 6 classes de ventes (
 - de 0 à 0.1 : 4888 lignes,
 - de 0.1 à 249 : 2902 lignes,
 - de 0.25 à 0.499 : 2205 lignes,
 - de 0.5 à 0.999 : 1552 lignes,
 - de 1 à 5 : 1517 lignes,
 - et supérieur à 5 : 166 lignes))

→ Gestion des dernières valeurs manquantes

Pour remplir nos dernières valeurs manquantes et pour ne pas faire fuiter de données dans l'échantillon de test, nous procédons de la sorte :

- Dans l'échantillon d'entraînement uniquement, calcul des moyennes des valeurs antérieures chronologiquement pour remplir les NaN de cet échantillon
- Une fois l'échantillon d'entraînement totalement comblé, calcul de la moyenne globale de toutes les rolling means
- Utilisation de ces moyennes pour combler les valeurs manquantes dans l'échantillon de test.

→ Évaluation du modèle pour la 1ère catégorisation (4 classes de ventes)

Un modèle est dit performant lorsqu'il enregistre le taux le plus faible d'erreur. Dans le cadre de la classification, l'erreur peut être considérée comme étant la distance entre la valeur réelle et la valeur prédite par l'algorithme. Notre modèle est censé prédire exactement la classe de vente d'un jeu vidéo donné. Cependant il est tout à fait possible que notre modèle se trompe c'est-à-dire qu'il prédise une classe différente de la classe réelle de vente du jeu. Par exemple, il peut prédire la classe 4 alors que la classe réelle est la classe 2. L'erreur sera donc la distance entre les deux classes qui est de 2. La métrique qui nous servira à évaluer notre modèle, sera le quotient de la somme des distances entre les prédictions et les valeurs réelles sur la pire erreur possible.

À l'aide de cette métrique, nous avons évalué tous les modèles mentionnés plus haut afin d'en sélectionner le plus performant.

```

1 knn=KNeighborsClassifier()
2 knn.fit(X_train,y_train)
3 y_pred_test=knn.predict(X_test)
4 metric_eval_knn=compute_score(y_test,y_pred_test)
5 print(" KNN compute score :" , metric_eval_knn)
6 print(" KNN score :" , knn.score(X_test,y_test))
7
8 rfc=RandomForestClassifier()
9 rfc.fit(X_train,y_train)
10 y_pred_test=rfc.predict(X_test)
11 metric_eval_rfc=compute_score(y_test,y_pred_test)
12 print(" RF compute score" , metric_eval_rfc)
13 print(" RF score :" , rfc.score(X_test,y_test))
14
15 dt=DecisionTreeClassifier()
16 dt.fit(X_train,y_train)
17 y_pred_test=dt.predict(X_test)
18 metric_eval_dt=compute_score(y_test,y_pred_test)
19 print(" DT compute score" , metric_eval_dt)
20 print(" DT score :" , dt.score(X_test,y_test))
21
22 lg=LogisticRegression(max_iter=1000,class_weight="balanced")
23 lg.fit(X_train,y_train)
24 y_pred_test=lg.predict(X_test)
25 metric_eval_lg=compute_score(y_test,y_pred_test)
26 print(" LG compute score" , metric_eval_lg)
27 print(" LG score :" , lg.score(X_test,y_test))

```

```

KNN compute score : 0.18488243011403774
KNN score : 0.5909778988798062
RF compute score 0.11999192653143607
RF score : 0.7260066606115653
DT compute score 0.1557170249268342
DT score : 0.657281259461096
LG compute score 0.19477242910485418
LG score : 0.5395095367847411

```

De cette évaluation, il ressort clairement que le RandomForest est le plus performant pour résoudre notre problématique car il obtient le score le plus faible d'erreur : **12%**.

→ Évaluation du modèle pour la 2ème catégorisation (6 classes de ventes)

```
1 knn=KNeighborsClassifier()
2 knn.fit(X_train,y_train)
3 y_pred_test=knn.predict(X_test)
4 metric_eval_knn=compute_score(y_test,y_pred_test)
5 print(" KNN compute score :" , metric_eval_knn)
6 print(" KNN score :" , knn.score(X_test,y_test))
7
8 rfc=RandomForestClassifier()
9 rfc.fit(X_train,y_train)
10 y_pred_test=rfc.predict(X_test)
11 metric_eval_rfc=compute_score(y_test,y_pred_test)
12 print(" RF compute score" , metric_eval_rfc)
13 print(" RF score :" , rfc.score(X_test,y_test))
14
15 dt=DecisionTreeClassifier()
16 dt.fit(X_train,y_train)
17 y_pred_test=dt.predict(X_test)
18 metric_eval_dt=compute_score(y_test,y_pred_test)
19 print(" DT compute score" , metric_eval_dt)
20 print(" DT score :" , dt.score(X_test,y_test))
21
22 lg=LogisticRegression(max_iter=1000,class_weight="balanced")
23 lg.fit(X_train,y_train)
24 y_pred_test=lg.predict(X_test)
25 metric_eval_lg=compute_score(y_test,y_pred_test)
26 print(" LG compute score" , metric_eval_lg)
27 print(" LG score :" , lg.score(X_test,y_test))
```

```
KNN compute score : 0.2471490564133616
KNN score : 0.547683923705722
RF compute score 0.17317590069633668
RF score : 0.6657584014532243
DT compute score 0.2176808961550106
DT score : 0.5885558583106267
LG compute score 0.26410334039761835
LG score : 0.48077505298213746
```

Comme on pouvait s'y attendre, le nombre de classes de ventes impacte la fiabilité du résultat.

Mais nous pouvons constater clairement, au moins pour le RandomForest, que dans les 2 cas, la classification nous apporte de bien meilleurs résultats que la régression.

Nous allons donc optimiser ce modèle (dans le cas de la catégorisation en 4 classes).

→ Optimisation de la classification avec le RandomForest (cas des 4 classes)

L'évaluation nous a permis de retrouver le modèle le plus performant. Cependant, cette évaluation s'est faite avec les paramètres par défaut du modèle. Il est possible d'utiliser des paramètres supplémentaires afin de gagner encore plus en performance avec ce modèle. Plusieurs hyperparamètres du RandomForest nous permettent de l'optimiser, dans le cas de notre modèle, nous nous sommes contentés des hyperparamètres suivants :

- **Criterion** : il s'agit du critère utilisé pour construire les arbres et séparer les branches des arbres
- **max_depth** : il s'agit de la profondeur maximale des arbres utilisés (le nombre de niveaux dans l'arbre de décision)
- **max_features** : il s'agit du nombre de colonnes sélectionnées pour chaque arbre (par défaut on prend la racine carré du nombre de colonnes)

Nous fournissons à un algorithme (*gridsearch*), une liste d'exemples de ces paramètres. Le but sera de tester plusieurs combinaisons de ces hyperparamètres afin de sélectionner celle qui minimise au maximum l'erreur.

```

1 param_rf=[{'criterion': ['gini', 'entropy'],
2             'max_depth': [4,5,6,7,8,9,10,11,12,15,20,30,40,50,70,90,120,150],
3             'max_features': ['auto', 'sqrt', 'log2']}]
4
5 gridcv=GridSearchCV(rfc,param_grid=param_rf, cv=3, scoring=make_scorer(compute_score, greater_is_better=False))
6 gridcv.fit(X_train,y_train)
7
8 gridcv.best_params_
9
10
{'criterion': 'entropy', 'max_depth': 150, 'max_features': 'log2'}
```

Il ressort de cette optimisation les hyperparamètres optimaux suivants :

- **Criterion** : entropy
- **max_depth** : 150
- **max_features** : log2

Malheureusement avec ces hyperparamètres optimaux n'ont pas réellement influencé notre score qui est de 73%.

```

1 metric_eval_rfc_grid = compute_score(y_test,y_pred)
2 print(" RF compute score" , metric_eval_rfc_grid)
3 print(" RF_grid score :" , rfc_grid.score(X_test,y_test))

RF compute score 0.11777172267635483
RF_grid score : 0.7308507417499243
```

→ Résultats du modèle

Dans notre démarche, nous avons séparé notre jeu de données en deux parties : un échantillon d'entraînement et un autre de test. Le second servira à cet instant à tester notre modèle de machine learning. Nous allons demander à notre modèle de prédire les ventes des jeux vidéo contenus dans cet échantillon et de les comparer aux classes réelles.

Les résultats de ces prédictions sont exploitables grâce à une matrice de confusion qui nous présente les bonnes et mauvaises prédictions de notre modèle.

```

1 cm = pd.crosstab(y_test, y_pred, rownames=['Classe réelle'], colnames=['Classe prédictive'])
2 cm

```

Classe prédictive	1	2	3	4
Classe réelle				
1	1009	125	123	31
2	91	406	184	19
3	40	48	740	76
4	13	8	131	259

Exemple de lecture de la matrice:

Sur 1288 jeux vidéos de notre dataset de test, 1009 ont été bien prédits pour la classe 1, 125 ont été prédits en classe 2, 123 en classe 3 et 31 en classe 4

Un autre outil nous permet également d'évaluer les résultats de notre modèle. Il s'agit de rapport de classification :

```

1 from sklearn.metrics import classification_report
2 print(classification_report(y_test, y_pred))

```

	precision	recall	f1-score	support
1	0.88	0.78	0.83	1288
2	0.69	0.58	0.63	700
3	0.63	0.82	0.71	904
4	0.67	0.63	0.65	411
accuracy			0.73	3303
macro avg	0.72	0.70	0.70	3303
weighted avg	0.74	0.73	0.73	3303

Le rapport de classification ressort des résultats plutôt équilibrés vu le rappel, la précision et la f1-score au niveau de toutes les classes.

Notre modèle obtient globalement des résultats plutôt satisfaisants. Cependant, il est important de se pencher sur les mauvaises prédictions des jeux vidéo.

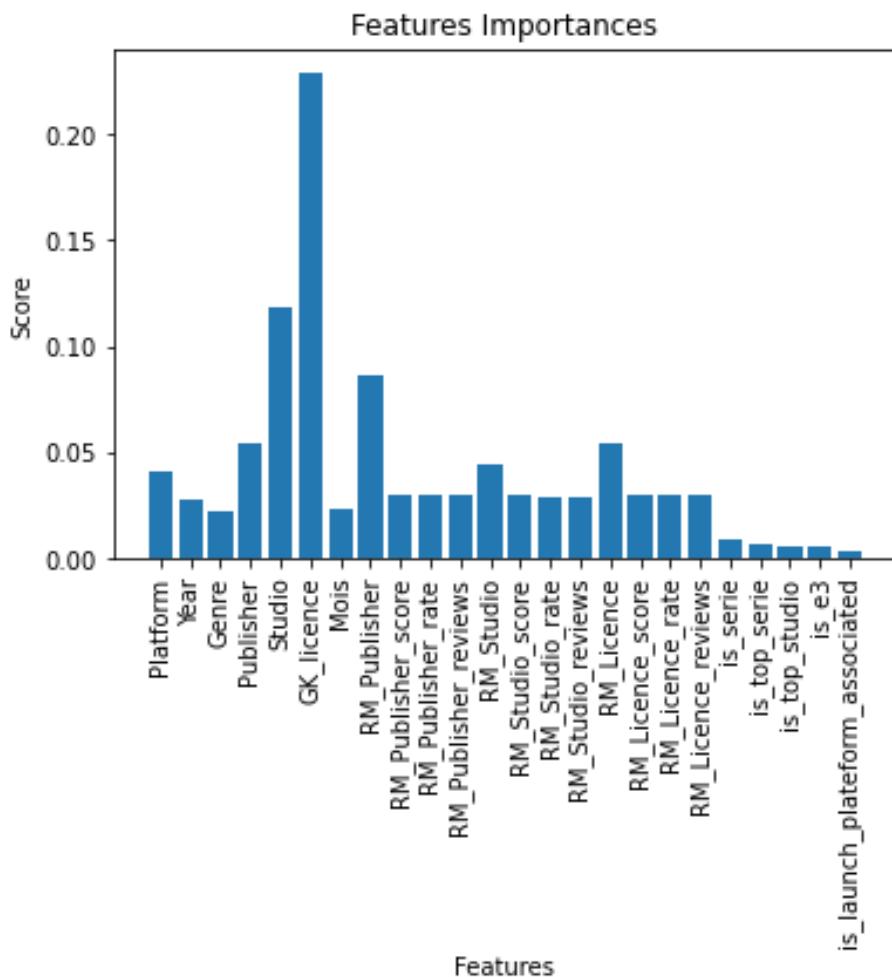
→ Variables ayant le plus influencé notre modèle

Grâce à la fonction feature_importances, nous pouvons voir que les variables qui ont eu le plus d'influences sur le résultat du modèle sont la licence, le studio et la moyenne mobile des ventes par éditeur.

```

1 import matplotlib.pyplot as plt
2 plt.bar(X_train.columns, rfc_grid.feature_importances_)
3 plt.xticks(rotation=90)
4 plt.xlabel("Features")
5 plt.ylabel("Score")
6 plt.title("Features Importances");

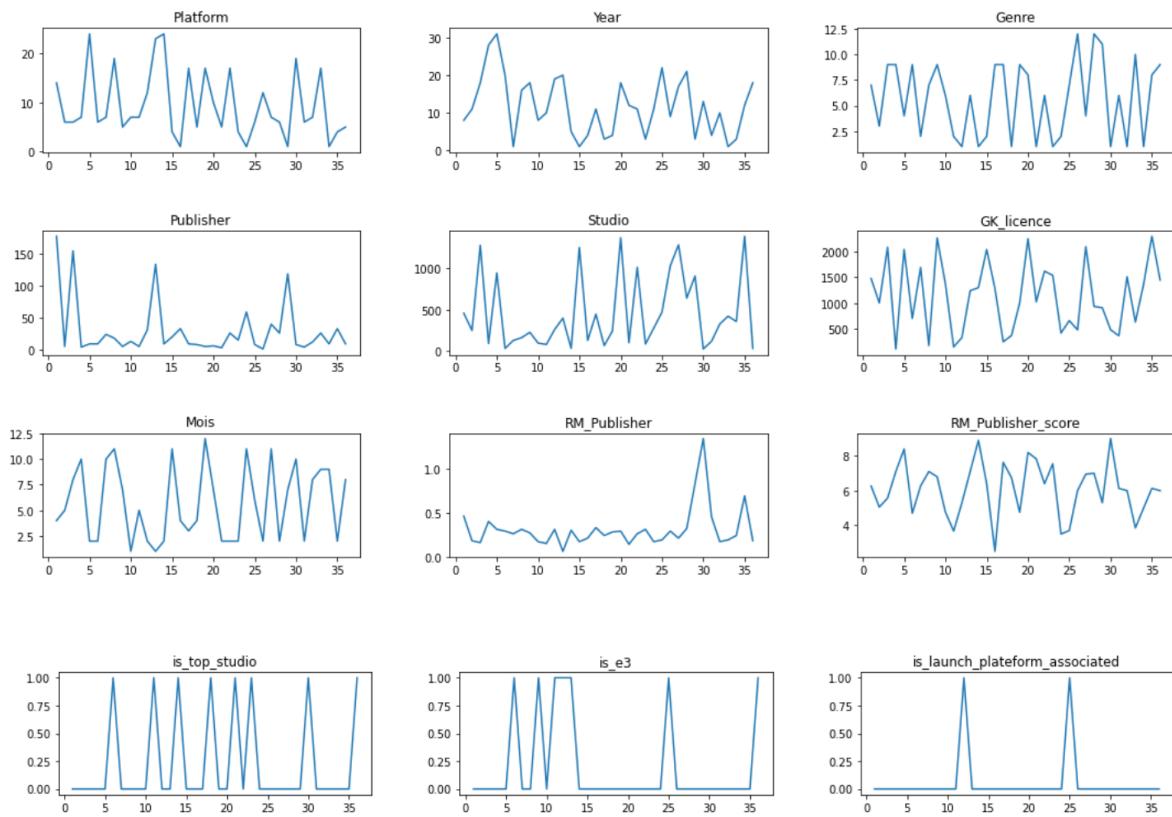
```



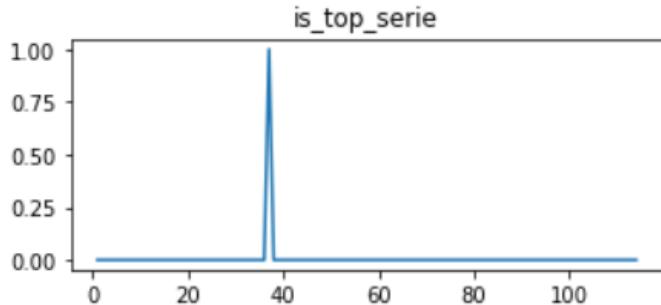
→ Analyse des jeux mals classés

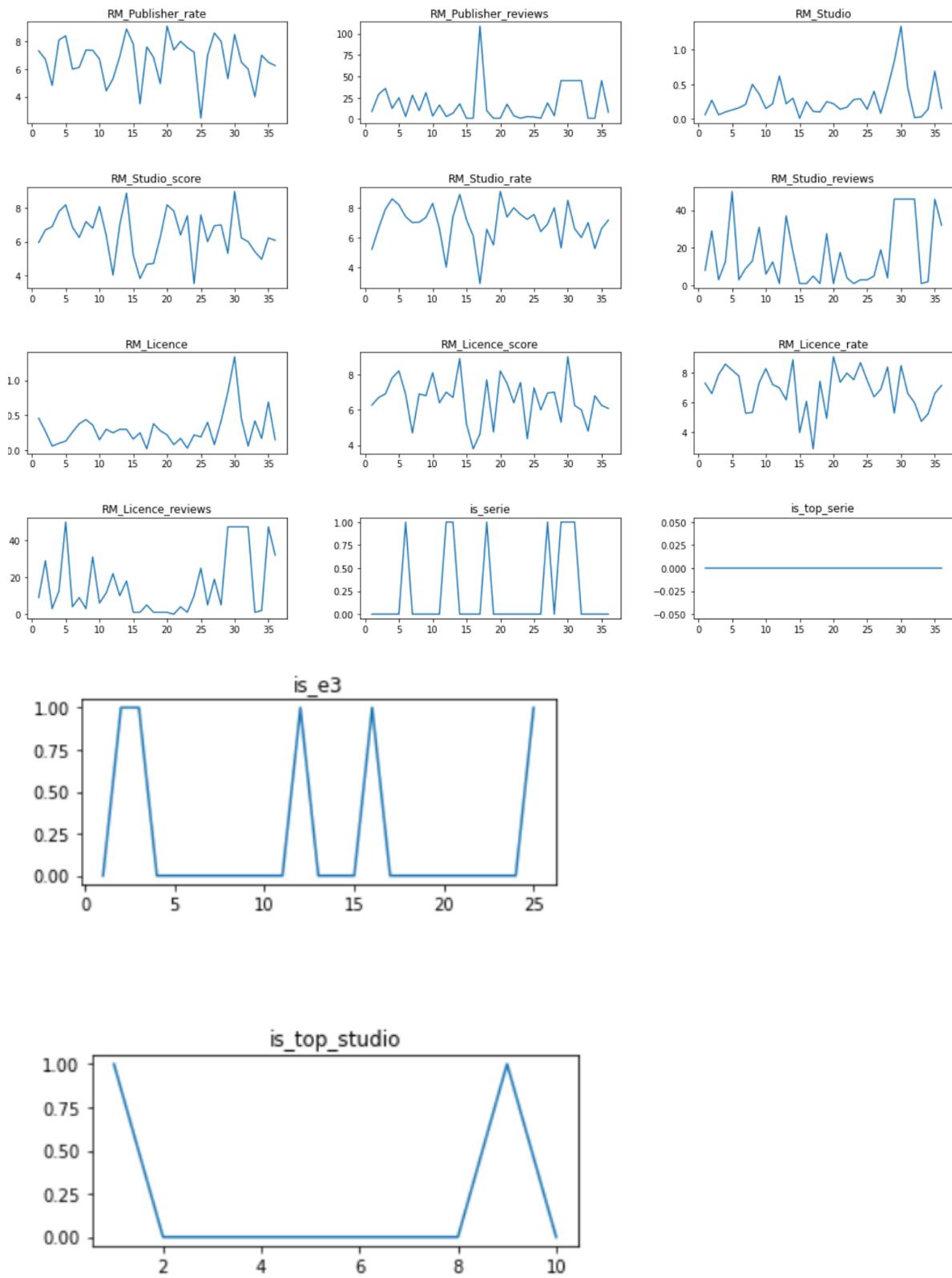
Nous avons fait un focus sur les mauvaises prédictions afin de découvrir de possibles tendances qui influeraient sur le comportement de notre modèle vis-à-vis de certains jeux. Pour chaque classe de jeux vidéo, nous avons visualisé les features pour déterminer ces tendances.

Exemple : Jeux vidéo de classe 3 qui ont été prédits en classe 2



Ces visualisations nous ont permis de détecter une certaine tendance sur des features en particuliers : *is_e3*, *is_launch_plateform_associated*, *is_top_studio* et *is_top_serie*





Pour affiner les performances de notre modèle, un approfondissement de recherche de tendances serait nécessaire. Nous n'avons malheureusement pas suffisamment de temps pour arriver au bout de cette démarche.

Conclusion

Le but de ce projet était de prédire avec exactitude le nombre d'exemplaires auquel serait vendu un nouveau jeu vidéo. Dans notre démarche, nous nous sommes rendus compte qu'avec le modèle de régression, les résultats obtenus ne convenaient pas à nos attentes. Nous nous sommes donc tournés vers un modèle de classification pour prédire plutôt la catégorie de vente d'un nouveau jeu vidéo.

Nous obtenons le score de **73% de bonnes prédictions sur 4 catégories de vente** et de **67% sur 6 catégories**.

Nous avons testé les résultats sur 4 et 6 catégories et nous avons observé les résultats obtenus. Le découpage en 4 classes nous semble être le meilleur compromis pour être utilisé par une entreprise. Mais le choix d'intervalles de nombre de ventes peut être adapté selon l'entreprise qui utilisera le modèle.

Lors de la data visualisation, nous avons observé que peu de données semblaient liées au niveau de ventes d'un jeu. Les seules relations que nous avions identifiées étaient que les jeux les plus vendus faisaient partie d'une licence (série de jeux) et étaient rattachés aux studios réalisant le plus de ventes.

L'affichage des *features* avec le plus d'influence sur notre modèle, à la fin du projet, montre une cohérence avec nos analyses. Comme observé, **le studio et la licence sont les principaux critères de notre modèle**.

Sources :

https://fr.wikipedia.org/wiki/Apprentissage_supervis%C3%A9

<https://www.lafinancepourtous.com/decryptages/entreprise/secteurs-dactivites/l-industrie-des-jeux-video-la-gratuite-peut-rapporter-gros/>

<https://www.jeuxvideo.com>

<https://www.metacritic.com/game>

<https://www.gamekult.com>

liste des séries de jeux vidéo :

https://fr.wikipedia.org/wiki/Lista_de_s%C3%A9ries_de_jeux_vid%C3%A9o

liste des séries de jeux vidéo les plus vendues :

https://fr.wikipedia.org/wiki/Lista_des_s%C3%A9ries_de_jeux_vid%C3%A9o_les_plus_vendues

liste des jeux cités lors du festival E3 :

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2000

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2001

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2002

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2003

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2004

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2005

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2006

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2009

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2010

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2011

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2012

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2013

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2014

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2015

https://fr.wikipedia.org/wiki/Electronic_Entertainment_Expo_2016

liste des jeux lancés en même temps qu'une console :

https://fr.wikipedia.org/wiki/Liste_de_jeux_au_lancement_des_consoles_de_jeux_vid%C3%A9o