EFREI - Master

PARIS PANTHÉON-ASSAS UNIVERSITÉ

Mr. Thierno MANSOUR

Suivi d'avancé Groupe 1

DATACAMP

Réalisé la semaine de 09 décembre 2024

De Elliot FESQUET, Ethan TOMASO,

Antoine VANDEPLANQUE

Table des Matières

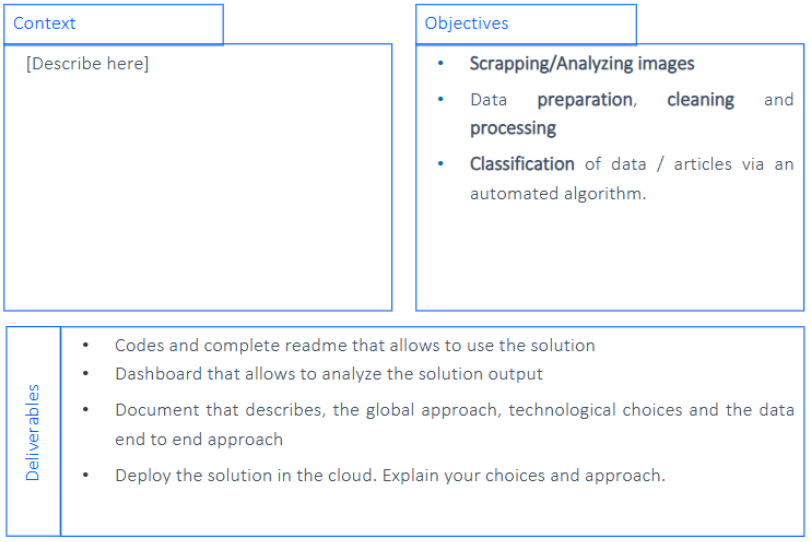
Table des Matières **2**
Projet DataCamp – Analyse et Prédiction du Succès des Jeux Steam **3**
Introduction 3
Choix du sujet 4
Répartition du travail (RACI) 4
Architecture et Plan du Projet 5
Collecte des données 5
Stockage brut sur Amazon S3 6
Préparation, Nettoyage et Classification (AWS Lambda) 6
Entraînement des Modèles et Prédiction (Local) 6
Base de Données MySQL 7
Datavisualisation avec Tableau 7
Choix Technologiques et Outils 8
Divergence & recontextualisation du projet: 10
Conclusion **11**

Projet DataCamp – Analyse et Prédiction du Succès des Jeux Steam

Introduction

Dans le cadre de ce projet, nous avons choisi d'explorer les données de la plateforme de jeux en ligne Steam, avec pour objectif d'identifier les facteurs déterminants du succès d'un jeu et de prédire sa popularité future. Pour ce faire, nous articulons notre approche autour d'une pipeline complète, allant de la collecte des données à leur visualisation, en passant par le prétraitement, l'entraînement de modèles prédictifs et le stockage des résultats. Ce projet s'appuie à la fois sur des données issues de notre propre processus de scrapping (informations journalières sur les jeux Steam) et sur un jeu de données externalisé (reviews disponibles sur Kaggle, datées de 2017), afin d'enrichir notre base d'analyse et d'améliorer la qualité de nos prédictions.

Nous avons donc suivi le sujet 1 :



Choix du sujet

Le marché du jeu vidéo sur Steam est caractérisé par une très grande diversité de titres et de profils de joueurs. Notre objectif est de procéder à une analyse de sentiments pour des jeux données, avec des paramètres sur le jeu en lui même et surtout les commentaires liés, et comprendre de leur contenu l'avis et le sentiment que l'utilisateur à voulu transcrire dans son avis, pour affiner la compréhension d'un modèle hypothétique sur une analyse poussé en langage naturel.

Nous avons décidé d'emettre la problématique suivante :

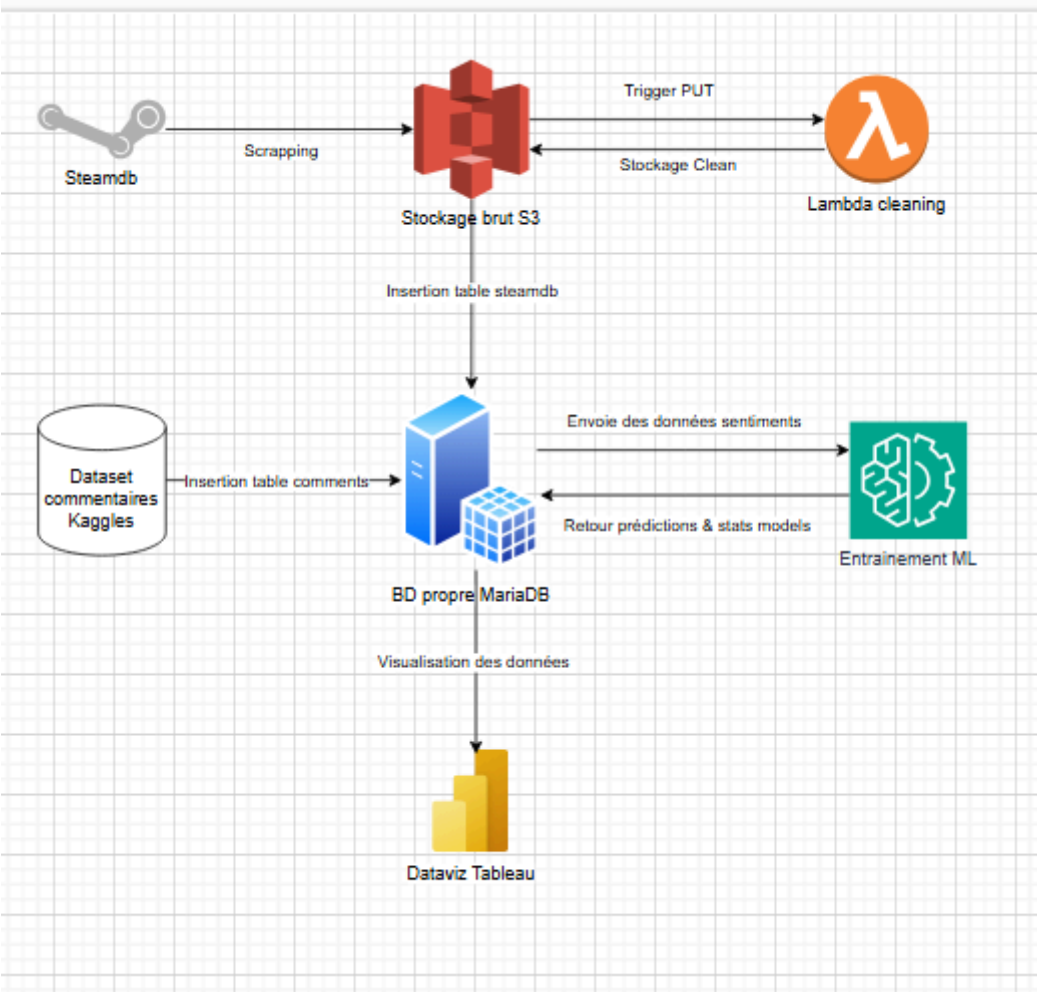
Peut-on, à partir d'une analyse sentimentale des commentaires des utilisateurs de Steam, prédire efficacement le ressenti global et identifier les caractéristiques des jeux susceptibles d'entraîner un succès ou un échec auprès de la communauté des joueurs ?

Répartition du travail (RACI)

Tâche	Réalisateur	Approbateur	Consulté	Informé
Tâche	réalisateur	approbateur	Consulté	Informé
scrapping	Antoine	Ethan	Ethan	Tout le monde
Pipeline Aws	Ethan	Elliot	Antoine	Tout le monde
BD Mysql & Tableau	Elliot	Ethan/Antoine	Antoine	Tout le monde

Architecture et Plan du Projet

Notre architecture se décompose en plusieurs étapes, intégrant des services Cloud AWS, un traitement local pour l'entraînement de modèles, et une solution de visualisation des données.



Collecte des données

Scrapping SteamDB : À l'aide de Selenium, nous extrayons régulièrement des informations clés sur les jeux (nombre de joueurs actifs, pics, éditeur, développeur, date de sortie, évaluation générale, support manette, genre, prix).

Données complémentaires : Nous enrichissons notre corpus avec des reviews issues d'un dataset Kaggle (2017), fournissant un point d'ancrage pour des analyses textuelles, notamment de sentiment.

Stockage brut sur Amazon S3

Les données brutes issues du scrapping et le dataset externe (reviews Kaggle) sont stockés sur Amazon S3. Ce référentiel centralisé facilite le déclenchement de processus automatisés (triggers) lors de l'arrivée de nouvelles données.

Préparation, Nettoyage et Classification (AWS Lambda)

Une fonction AWS Lambda, déclenchée par l'événement d'écriture sur S3, prend en charge la préparation et le nettoyage des données :

- Uniformisation des formats
- Traitement de valeurs manquantes et suppression de doublons
- Filtrage et prétraitement des reviews (text mining : tokenization, stopwords, stemming, etc.)

Cette étape garantit la constitution d'un jeu de données propre et structuré, prêt pour l'entraînement des modèles.

Entraînement des Modèles et Prédiction (Local)

Nous entraînons nos modèles prédictifs en local. Trois types de prédictions sont envisagés :

- Prédiction du succès global d'un jeu (analyse par sentiment, mesures d'engagement)
- Prédiction du pic de joueurs au mois M+1
- Prédiction du gain ou de la croissance du nombre de joueurs sur la période suivante

Les modèles sont comparés (par exemple via PyCaret ou Scikit-Learn) afin d'identifier ceux offrant les meilleures performances. Les paramètres sont optimisés pour maximiser la précision des prédictions. Les modèles retenus peuvent ensuite être stockés sur S3 pour faciliter leur réutilisation.

Base de Données MySQL

Une base MySQL (initialement envisagée sous Aurora, mais finalement retenue sur MySQL pour des raisons de coût et de simplicité) servira de stockage des données prédites. On y retrouvera :

- Les prédictions générées par nos modèles
- Les métriques de performance (RMSE, MAE, précision, etc.)

Ce stockage structuré offre une source fiable pour l'accés aux données avec l'outil de dataviz.

Datavisualisation avec Tableau

En connectant Tableau à la base MySQL, nous sommes en mesure d'élaborer des dashboards de visualisations. Ces visualisations permettent d'explorer les résultats, de mettre en évidence les tendances clés et d'obtenir une vision plus concrète des prédictions modélisées, qu'il s'agisse de l'évolution d'un ressenti général, du type de jeux, de ces thèmes, des commentaires les plus impactant et d'autres métadonnées intéressantes.

Cette visualisation a pour but de mettre à disposition chaque KPI's des données exploitées. A travers un dashboard dynamique, nous répondrons à la problématique suivante:

Peut-on, à partir d'une analyse sentimentale des commentaires des utilisateurs de Steam, prédire efficacement le ressenti global et identifier les caractéristiques des jeux susceptibles d'entraîner un succès ou un échec auprès de la communauté des joueurs ?

Dashboard avant réadaptation du sujet

Choix Technologiques et Outils

La réussite de ce projet repose sur une combinaison cohérente de technologies et de librairies, couvrant l'ensemble de la chaîne de valeur des données, du scraping à la visualisation finale. Les choix effectués découlent de considérations pragmatiques (performance, coût, compatibilité, facilité d'utilisation) et de standards de l'industrie.

1. **Environnement Cloud (AWS)**
    - **Amazon S3** : Service de stockage d'objets scalable, utilisé pour entreposer les données brutes issues du scraping (fichiers JSON, CSV, etc.) ainsi que les éventuels modèles de Machine Learning. Choisi pour sa capacité à gérer de larges volumes de données, sa durabilité et son intégration native avec d'autres services AWS.
    - **AWS Lambda** : Service de fonctions serverless déclenchées par des événements sur S3 (e.g. l'arrivée d'un nouveau fichier). Lambda permet de transformer, nettoyer et prétraiter les données sans gérer d'infrastructure, de sorte que les opérations d'ETL sont déclenchées automatiquement. Ce service facilite l'intégration continue des données et leur préparation.
2. L'ensemble S3 + Lambda offre un pipeline automatisé, élastique et sans maintenance serveur, adapté à la nature intermittente et variable du flux de données.
3. **Stockage Structuré (MySQL)**
    - **MySQL** : Base de données relationnelle open-source, largement adoptée dans l'industrie. Elle sert ici à conserver les données finales, les résultats des prédictions, les métriques de performance des modèles, ainsi que certaines informations normalisées. Le choix de MySQL (au lieu d'Aurora) repose sur la maîtrise des coûts et la simplicité de mise en place, tout en offrant une intégration aisée avec des outils de data visualization.
4. MySQL s'intègre facilement aux workflows analytiques, est compatible avec la majorité des outils BI et permet des requêtes SQL efficaces pour le reporting.
5. **Scraping Web & Acquisition de Données**
    - **Selenium** : Outil de test et d'automatisation de navigateurs web. Dans ce projet, il permet d'effectuer du web scraping sur SteamDB, un site protégeant activement ses données contre certaines méthodes de scraping plus simples (comme BeautifulSoup). Selenium offre la possibilité de simuler un vrai navigateur et de contourner des mesures anti-bot élémentaires, garantissant ainsi l'extraction fiable des informations nécessaires (nombre de joueurs, pics, développeurs, éditeurs, genres, prix, etc.).

- **Dataset externe Kaggle (Reviews de 2017)** : En complément du scraping, nous utilisons un dataset de reviews provenant de Kaggle. Ce corpus de critiques textuelles enrichit les données disponibles et fournit un support pour des analyses sémantiques (sentiment analysis) et l'entraînement de modèles prédictifs basés sur le langage naturel.

6. **Data Engineering & Prétraitement**
   - **Python** : Langage de programmation généraliste, populaire en data science et data engineering. Python est privilégié pour sa riche écosystème de librairies dédiées à la manipulation, au traitement et à l'analyse de données, ainsi qu'à l'entraînement de modèles de Machine Learning.
   - **Pandas** : Librairie Python standard pour le traitement et la manipulation de données tabulaires. Pandas permet un nettoyage efficace des données, la gestion des valeurs manquantes, le filtrage, le groupement et la préparation structurée avant l'étape d'entraînement des modèles.
   - **Numpy** : Librairie Python pour la gestion de tableaux multidimensionnels et l'algèbre linéaire. Numpy est le socle sur lequel reposent de nombreuses autres librairies et garantit des opérations efficaces et vectorisées sur les données, améliorant ainsi les performances dans les étapes de feature engineering.
   - **NLTK** (Natural Language Toolkit) : Librairie dédiée au traitement du langage naturel. Elle est utilisée pour le tokenization, la suppression des stopwords, le stemming/lemmatisation, et d'autres tâches de préparation textuelle. L'analyse des reviews nécessite ces transformations pour obtenir des features textuelles prêtes à l'emploi dans les modèles de classification ou de régression.
   - **String, Re et autres librairies standard Python** : Pour le nettoyage de chaînes de caractères, la manipulation de textes, et les opérations de base sur les données.
7. **Machine Learning & Modélisation**
   - **Scikit-Learn** : Librairie centrale en Python pour l'apprentissage automatique. Elle offre une large panoplie de modèles supervisés (régression, classification), non supervisés (clustering), ainsi que des outils pour la sélection de features, la validation croisée, la recherche paramètres (GridSearch, RandomizedSearch), etc. Scikit-Learn facilite les expérimentations rapides et robustes.
   - **PyCaret** (optionnel selon nos tests) : Outil d'auto-ML qui permet de tester rapidement différents algorithmes, comparer leurs performances et

sélectionner le meilleur modèle. PyCaret simplifie et accélère le processus d'expérimentation, déchargeant l'équipe de l'implémentation et de la comparaison manuelle de plusieurs techniques d'apprentissage (Naive Bayes, SVM, Random Forest, XGBoost, etc.). Le choix entre scikit-learn et PyCaret dépendra des exigences de flexibilité et du niveau de contrôle souhaité.

8. **Visualisation et Reporting**
    - **Tableau** : Outil professionnel de Business Intelligence et de data visualization. En se connectant directement à la base MySQL, Tableau permet de créer des tableaux de bord interactifs, des graphiques, des cartes et des visualisations avancées. Ce choix facilite la présentation de résultats à des acteurs métiers non techniques, permettant une prise de décision éclairée basée sur des insights clairs et lisibles.
    - **Matplotlib, Seaborn** : Librairies Python pour la visualisation de données. Avant la mise en place du reporting final sur Tableau, ces outils permettent l'exploration visuelle rapide, la création de graphiques exploratoires pour comprendre la distribution des données, la relation entre variables et pour valider certains choix de preprocessing.
9. **Gestion de la Qualité & du Cycle de Vie**
    - **Git/GitHub** : Outil de contrôle de version et de collaboration essentiel. Le code du projet, les scripts de scraping, les notebooks d'analyse, les pipelines ETL et les configurations d'infrastructure sont versionnés, assurant traçabilité, collaboration et rollbacks aisés en cas de problèmes.
    - **Gestion d'Environnements Virtuels (venv, Conda)** : Isolation de l'environnement de développement pour garantir la reproductibilité. L'utilisation d'environnements virtuels permet d'éviter les conflits de dépendances entre librairies et de s'assurer que les mêmes versions sont utilisées tout au long du projet.

Divergence & recontextualisation du projet:

Ce projet a vu plusieurs problèmes et réhabilitation dû à des besoins spécifiques.

Notre première divergence à été au sujet des technologies cloud et de l'architecture générale, le plan d'origine avait été d'utiliser les services AWS à leur maximum,

malheureusement, les services Sagemaker, aurora, et redshift, était tout trois payant, ce qui nous à forcé la main sur une architecture plus local, tout en utilisant le s3 & la lambda pour l'ingestion du scrapping.

Le deuxième plus gros point à été sur le remaniement de la problématique. nous étions de base, parti sur une problématique de volonté de prédiction de succès d'un jeu par la fluctuation mensuel de nombre de joueurs, cette problématique à été, assez tard dans le développement, remise en question, car s'écartant du sujet choisi à l'origine, qui était plutôt centré sur une analyse de sentiment. Grâce à une réhabilitation rapide du projet, une grande partie de l'architecture à été gardé, pour pouvoir insérer un nouveau flux de données de commentaires qui se liait aux données déjà extraites, en définissant la problématique, nous avons pu, dans les temps, remplir l'objectif du sujet tout en répondant à une problématique cohérente en son rapport.

Conclusion

À l'issue de ce projet, nous avons réussi à construire un pipeline complet allant de la collecte des données au déploiement de résultats dans un tableau de bord interactif. Ce pipeline repose sur une infrastructure hybride combinant des services cloud AWS (S3, Lambda) et des traitements locaux pour l'entraînement des modèles.

Notre modèle final, un Multinomial Naive Bayes, atteint une précision de **88%**, un F1-score de **93%**, un recall de **94%**, et une précision de **88%**. Ces résultats indiquent une excellente capacité du modèle à identifier les sentiments exprimés dans les commentaires des utilisateurs. Le recall élevé montre que le modèle détecte efficacement la majorité des sentiments exprimés, tandis que l'équilibre global entre précision et F1-score démontre une robustesse dans les prédictions.

L'analyse sentimentale mise en place a permis de tirer plusieurs enseignements :

- Les jeux aux commentaires positifs ont souvent des points communs dans leur vocabulaire, avec des mots-clés liés à des expériences de jeu agréables, comme "fun", "great", ou "amazing".
- Les jeux aux commentaires négatifs présentent une structure linguistique plus détaillée, souvent avec des critiques constructives ou des frustrations marquées par des mots comme "bug", "crash", ou "boring".

Ces observations offrent des opportunités pour les studios de développement de jeux vidéo de mieux comprendre les attentes des joueurs et d'améliorer la qualité de leurs titres en répondant aux besoins identifiés dans les analyses sentimentales.

Limites et perspectives :

- Bien que le modèle ait obtenu des performances élevées, il pourrait être amélioré par l'intégration de données supplémentaires (comme des commentaires récents ou des évaluations croisées avec des plateformes concurrentes).
- L'analyse sentimentale pourrait être étendue à d'autres langues afin de mieux capturer les avis de la communauté internationale des joueurs.
- Enfin, en intégrant davantage de métadonnées des jeux (genre, prix, etc.), il serait possible de construire des modèles encore plus précis pour anticiper les succès commerciaux et l'engagement des joueurs.

En conclusion, cette étude démontre que l'analyse des commentaires des utilisateurs constitue une méthode fiable pour comprendre et prédire les sentiments globaux envers les jeux, et pourrait être utilisée par les développeurs et éditeurs comme un outil stratégique pour orienter leurs décisions et maximiser l'impact de leurs futurs jeux. Il faut donc retenir qu'un jeu avec un bon ressenti relève les mots clés "fun", "great", "amazing", et un mauvais "bug", "crash", et "boring", avec ce worldcloud qui par fréquence des mots représente bien ces sentiments: