Projet Cloud Computing

Introduction

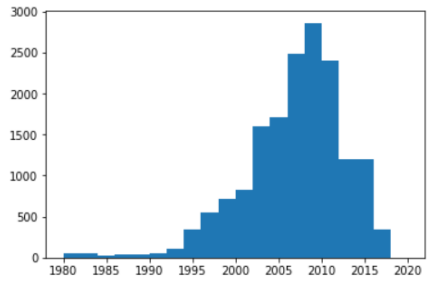
Dataset

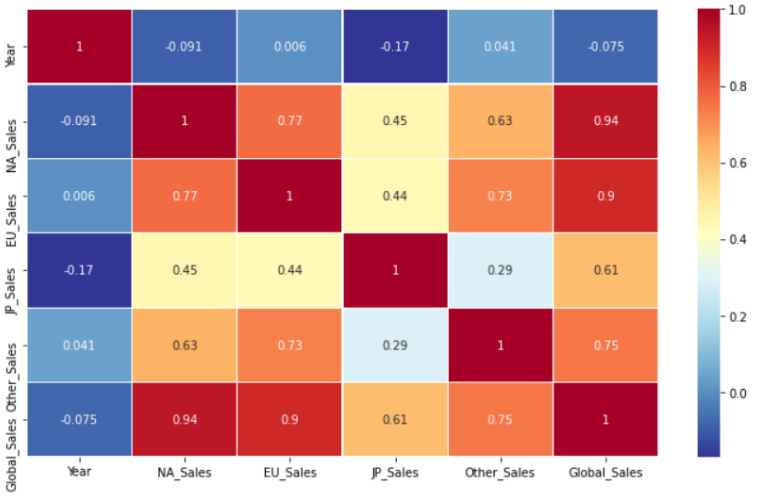
Le dataset est sur les jeux vidéo. Il regroupe leurs dates de sortie, le fabriquant ainsi que les ventes dans plusieurs parties du monde comme l'Europe ou le japon, avec un regroupement des ventes global.

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

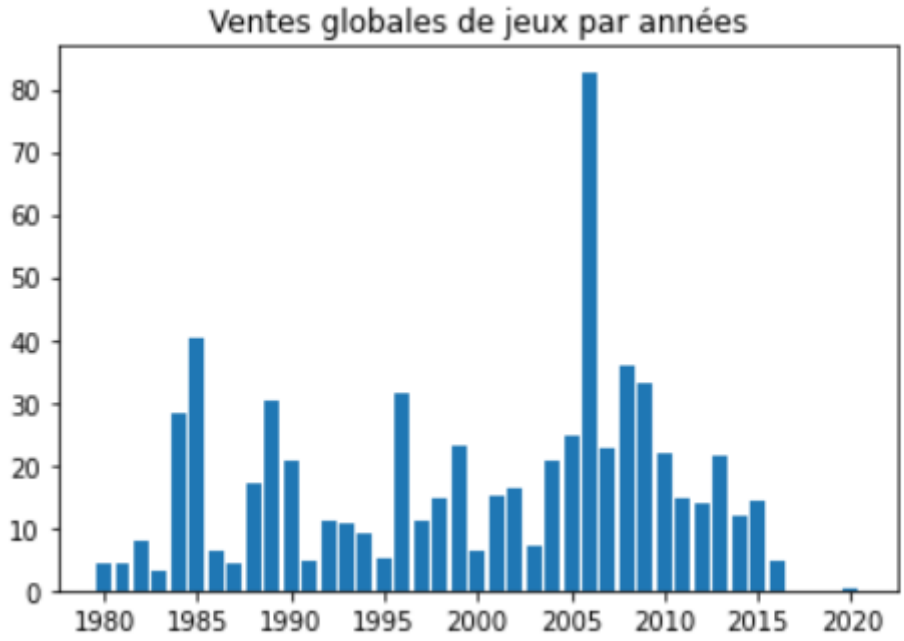
Diagrammes

Diagramme affichant le nombre de jeux vidéo par rapport a l'année





Heatmap



Pipieline

La pipeline est composé de 4 classes :

- Data Handler : qui charge le fichier .csv
- Feature Recipe : qui va traiter les données du tableau afin de les rendre exploitables
- Feature Extractor : dans cette classe sont définis les colonnes utilisées pour le model
- Model Builder : dans cette classe, un modèle de régression linéaire est utilisé pour déterminer la corrélation en les colonnes choisie dans Feature Extractor .

Cross validation

Une validation croisée est une méthode d'estimation de la fiabilité d'un model via une technique d'échantillonnage.

Résultats :

```
CrossValidation :
0.96 precision avec une deviation de 0.05
```

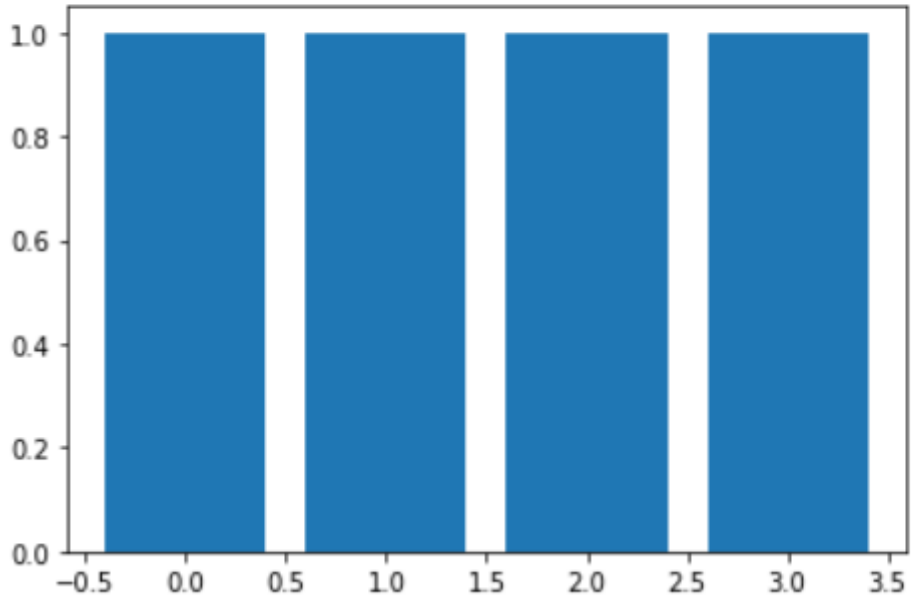
Feature Importance

C'est une technique utilisé pour attribuer un score sur les features passé en paramètres à un modèle de prédiction, ce qui permet d'indiquer l'impact de chaque features sur les prédictions

Cet exemple n'est pas probant puisque dans le tri effectué dans la classe FeaturExtractor a permis d'extraire du dataset les données qui aurai pu avoir un impact négatif sur les prédictions.

Diagramme & valeurs des features :

```
Feature importance :
Feature: 0, Score: 0.99985
Feature: 1, Score: 1.00004
Feature: 2, Score: 0.99986
Feature: 3, Score: 0.99981
```



Dans le cas de mon dataset les toutes les features sélectionnés ont un impact sur les prédictions

Model.py

La classe principale du pipeline qui instancie un objet de chaques classes afin de construire un model et le sauvegarder dans le schéma suivant : ''model_<date>.joblib.z''. Le fichier peut etre appelé avec 2 parametre :

- --model : pour spécifier le model de prédictions utilisé
- --split : pour spécifier le pourcentage de données utilisé pour l'entrainement et les tests afin de valider la fiabilité du model de prédictions

L'api

Dans l'api construite avec FastAPI est chargé le model sauvegardé dans la classe ModelBuilder afin de pouvoir afficher les coefficients de corrélations.

Docker

Le fichier DockerFile est présent dans le dossier Projet-Clourd-Computing avec le fichier requirements.txt qui liste toutes les bibliothèques nécessaires afin que l'api et le pipeline puise être fonctionnel.

Suite a une erreur a lancement de l'api, je n'ai pas pu déployer l'api via un docker ni sur heroku.