

Python for data analysis

Skillcraft dataset

Johnny LIN - Jérémy Goldschild



Contexte

StarCraft II est un jeu vidéo de stratégie en temps réel de science-fiction développé et publié par Blizzard Entertainment;

Le jeu vidéo StarCraft II est un **jeu multijoueur** où plusieurs joueurs s'affrontent dans **une arène** pour collecter des ressources, capturer des points stratégiques et détruire les bases adverses



Présentation du dataset

Le dataset regroupe **un ensemble de données des différentes statistiques** de matchs de Starcraft II d'un joueur selon son niveau de jeu défini par son classement dans la ligue.

On explique certaines données techniques.

- 1 seconde = 88,1 timestamp (système de temps du jeu)
- PAC signifie Perception-Action Cycle : L'ensemble d'action prise par le joueur sur une zone de la carte
- HotsKeys : Les touches utilisées pour effectuer des actions spécifiques sur le jeu








Feature	Description	Type
GameID	Unique ID number for each game	integer
LeagueIndex	Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional leagues Ordinal 1-8	integer
Age	Age of each player	integer
HoursPerWeek	Reported hours spent playing per week	integer
TotalHours	Reported total hours spent playing	integer
APM	Action per minute	float
SelectByHotkeys	Number of unit or building selections made using hotkeys per timestamp	float
AssignToHotkeys	Number of units or buildings assigned to hotkeys per timestamp	float
UniqueHotkeys	Number of unique hotkeys used per timestamp	int
MinimapAttacks	Number of attack actions on minimap per timestamp	float
MinimapRightClicks	number of right-clicks on minimap per timestamp	float
NumberOfPACs	Number of PACs per timestamp	float
GapBetweenPACs	Mean duration in milliseconds between PACs	float
ActionLatency	Mean latency from the onset of a PACs to their first action in milliseconds	float
ActionsInPAC	Mean number of actions within each PAC	float
TotalMapExplored	The number of 24x24 game coordinate grids viewed by the player per timestamp	int
WorkersMade	Number of SCVs, drones, and probes trained per timestamp	float
UniqueUnitsMade	Unique units made per timestamp	int
ComplexUnitsMade	Number of ghosts, infestors, and high templars trained per timestamp	float
ComplexAbilitiesUsed	Abilities requiring specific targeting instructions used per timestamp	float

Présentation des données

Le jeu vidéo comporte un **système de classement** qui classe les joueurs en fonction de leurs performances et de leurs victoires dans les matchs de compétition. Le système de classement est caractérisé par différentes ligue, de la ligue de bronze (le niveau le plus bas) à la ligue professionnelle (un niveau professionnel).

Son mode multijoueur a été particulièrement populaire, où de nombreux joueurs ont atteint le statut de professionnels.

Ce projet peut être utilisé comme **un outil pour évaluer la performance du joueur** selon sa performance durant une partie.

Icon	Name Rank	Level
	Grandmaster	Professional
	Master	Competitive
	Diamond	Good
	Platinum	Medium+
	Gold	Medium
	Silver	Medium-
	Bronze	Low

Travail effectué



- Data Exploration : Voir l'état des données
- Data Cleaning : Nettoyer et préparer les données
- Data Visualisation : Observer les liens entre les données



Modélisation : Avec du Machine Learning, chercher le meilleur algorithme prédictif



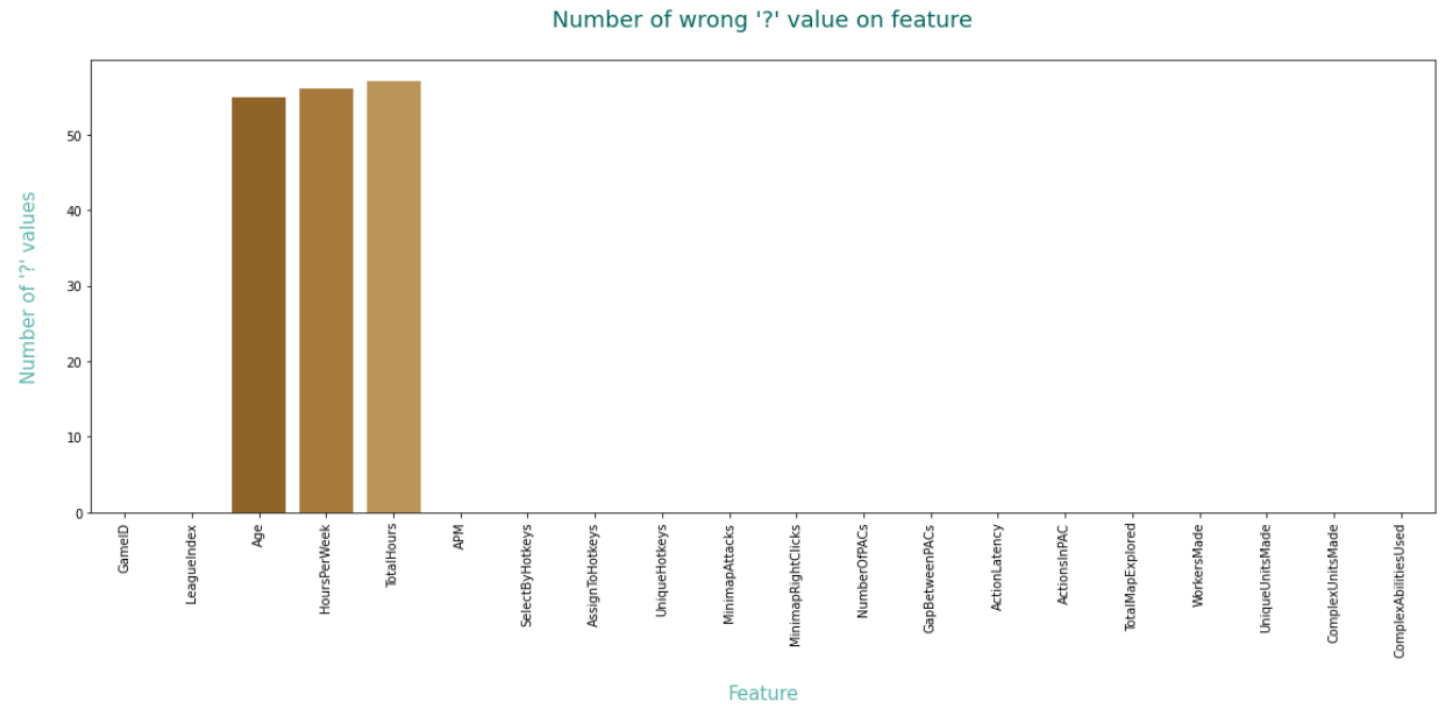
API Django : Une interface pour obtenir les résultats du projet

Data Cleaning

Nous avons rencontré des difficultés à plusieurs niveaux avec ce dataset notamment lors du Data Cleaning.

Les données fournies possèdent des données erronées ou manquantes sous la forme « ? » dans 3 colonnes.

Nous avons décidé de supprimer la ligne concernant **ces données manquantes**

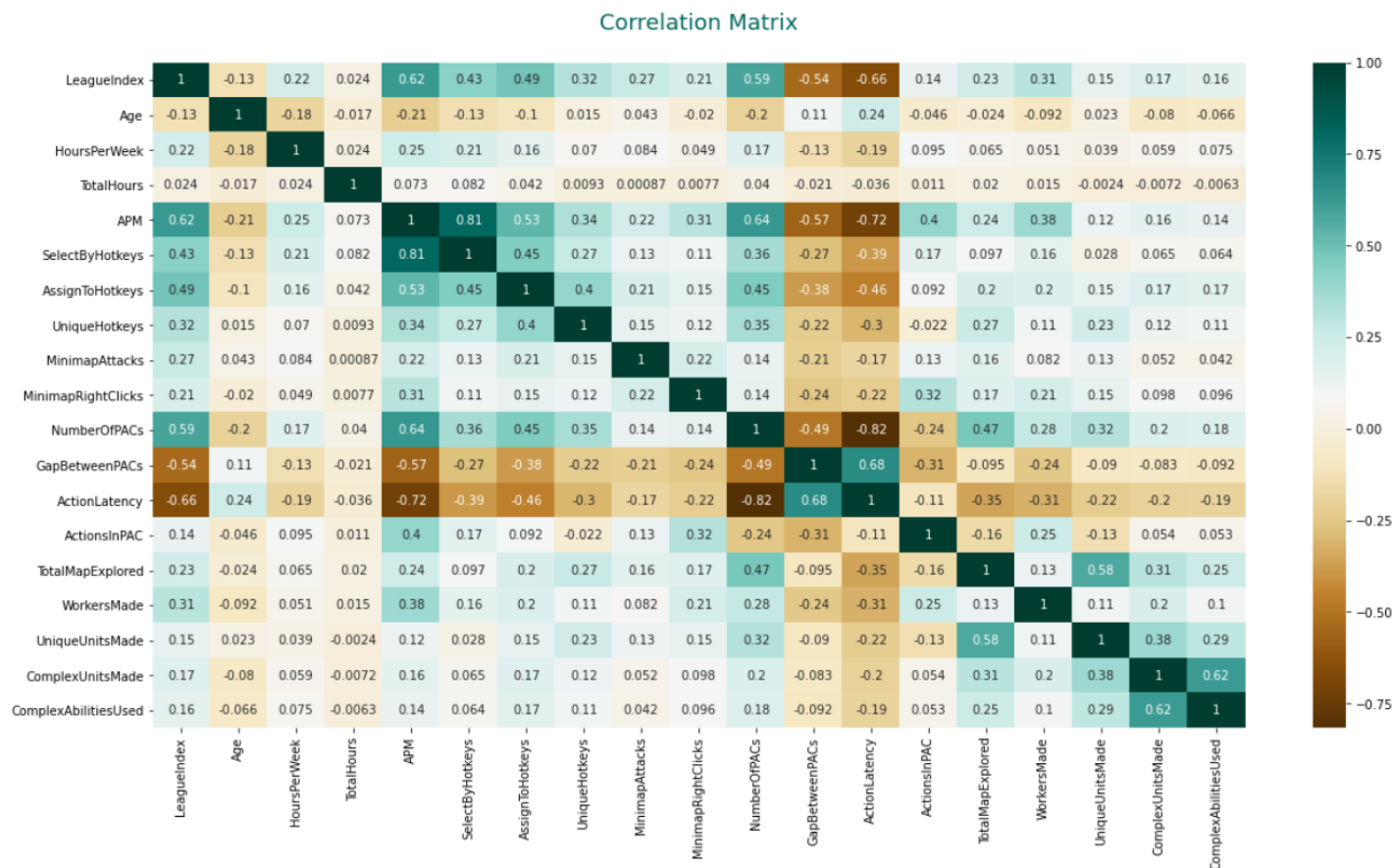


Data Visualisation

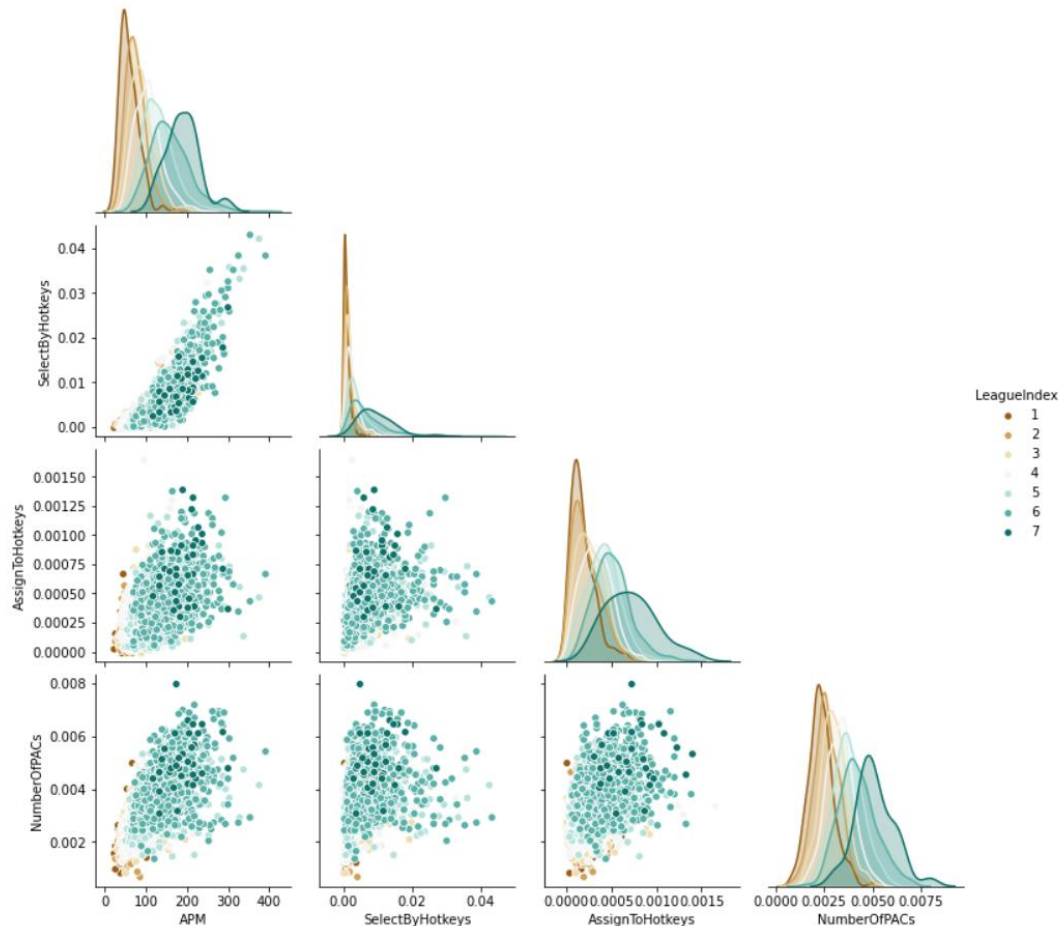
Matrice de corrélation :

On remarque 3 groupes de données :

- **Les données corrélées positivement avec LeagueIndex :**
APM, SelectByHotKeys, AssignToHotkeys, NumberOfPACs
- **Les données corrélées négativement avec LeagueIndex :**
GapBetweenPACs, ActionLatency
- **Les données peu corrélées**



Data Visualisation



Nous construisons un pairplot pour visualiser dans un premier temps les **features fortement corrélées positivement** avec la ligue du joueur **LeagueIndex**.

Un joueur d'un niveau bas aura un **NumberOfPACs faible** (peu cycle d'action) **donc moins d'action par un faible APM et des HotKeys** (des raccourcis clavier moins sollicités).

On visualise donc le profil bon joueur par **un grand nombre d'action APM, des cycles Perception Action** bien plus nombreux et une sélection plus faible du nombre d'unités.

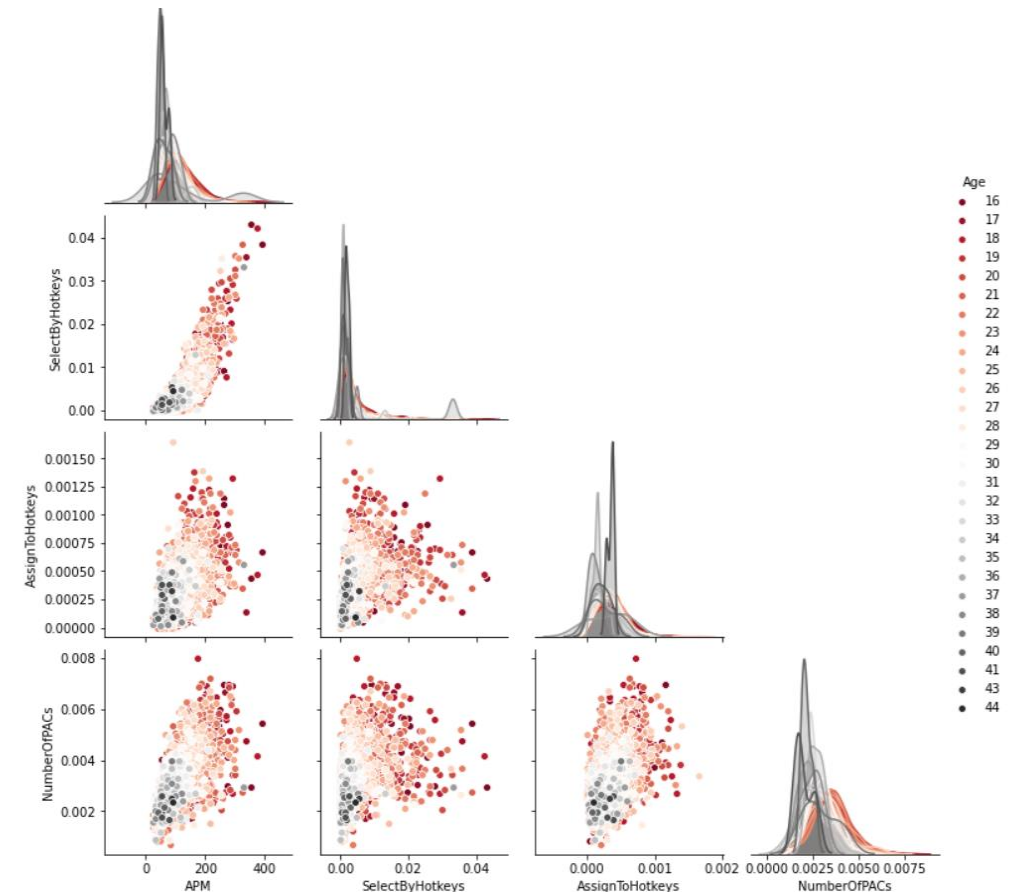
Data Visualisation

D'après la matrice de corrélation, il existe des faibles corrélations négatives de l'âge sur le nombre d'action APM et le nombre de Cycle Perspection Action (NumberOfPACs).

Un âge plus élevé entraîne des actions moins rapides ?

On remarque seulement qu'un écart d'âge élevé de plus de 30 ans a un **impact significatif sur la performance et la rapidité**.

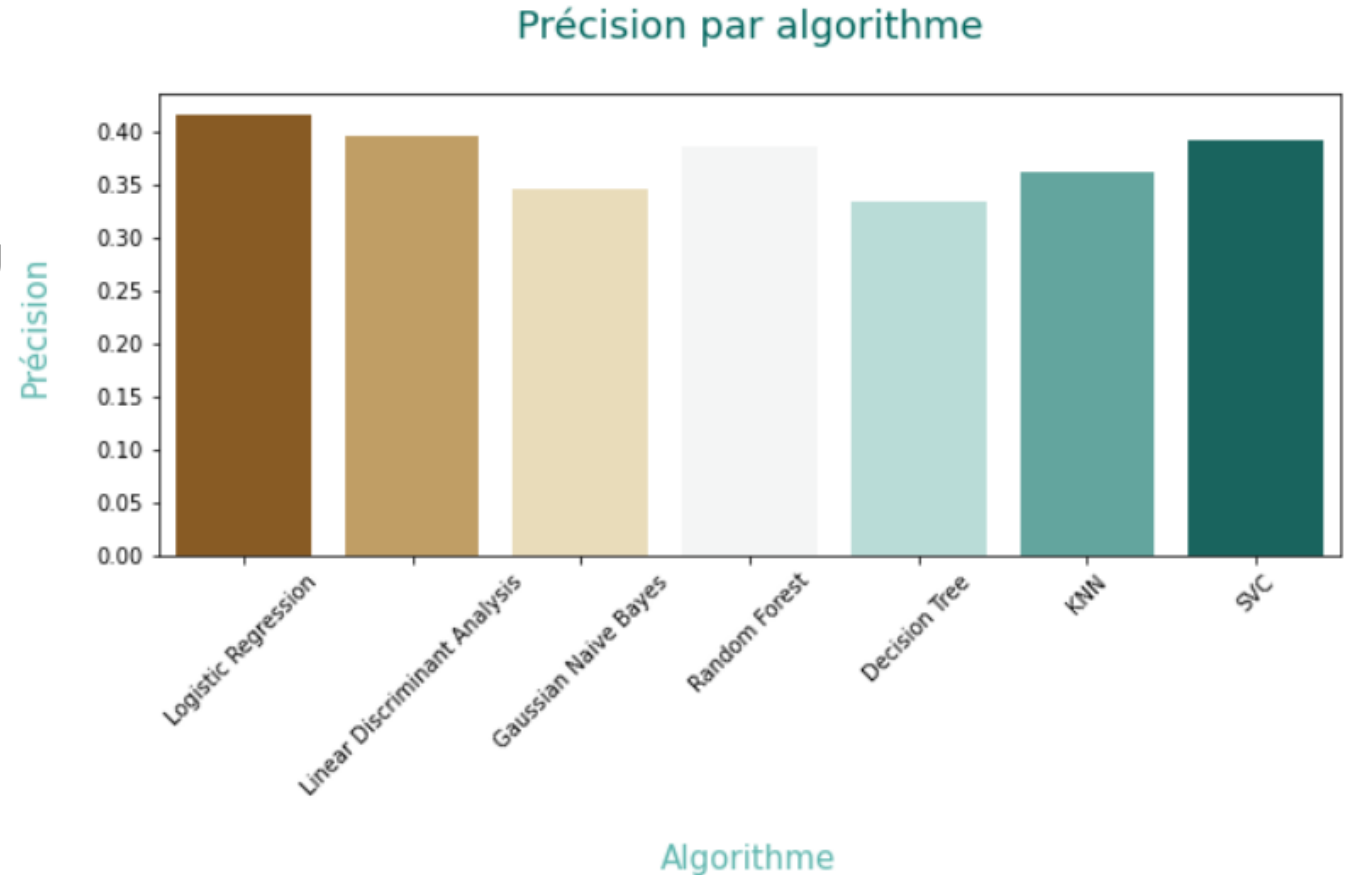
Les jeunes de moins de 30 ans possèdent **des performances très similaires** cependant.



Machine Learning

Nous avons testé différents algorithmes de machine learning afin de choisir celui qui aura le meilleur taux de prédiction sur le rang d'un joueur.

Après avoir optimisé les paramètres des algorithmes avec des méthodes de grid-search et cross-validation, le meilleur algorithme que nous avons obtenu était la Régression Logistique avec un taux de prédiction de 41,3%.



API

Nous avons transformé notre modèle en API Flask afin de pouvoir prédire le rang d'un joueur avec des données entrées à travers une interface en ligne.

Predict Starcraft 2 rank

Age
Hours Per Week
Total Hours
APM
Select By Hotkeys
Assign To Hotkeys
Unique Hotkeys
Minimap Attacks
Minimap Right Clicks
Number Of PACs
Gap Between PACs
Action Latency
Actions In PAC
Total Map Explored
Workers Made
Unique Units Made
Complex Units Made
Complex Abilities Used
Predict