



League of Legends est un jeu multijoueur où une équipe, composée de 5 joueurs, affronte une autre équipe de 5 joueurs. Le but : détruire la base de l'équipe ennemie. Chaque joueur est assigné à un poste bien précis, et va devoir assumer une fonction durant la partie (soigner les alliés, faire des dégâts, protéger les alliés, etc.). Pour ce faire, chacun des joueurs incarne un champion parmi les quelque 150 disponibles, et la partie peut commencer. Ces champions sont séparés en plusieurs catégories, League of Legends en identifie 6 majeurs : assassin, combattant, mage, tireur, support et tank. Une fois la partie lancée, il est nécessaire pour chaque joueur d'améliorer son champion dans l'optique de combattre les champions ennemis. Cette amélioration passe par l'achat d'objets variés au fur et à mesure de la partie. Ces objets octroient des statistiques de combat bonus, comme des dégâts, de la résistance, des points de vie, de la vitesse de déplacement ou encore du mana (et bien d'autres). Ainsi, chacun de ces objets est naturellement plus adapté à une ou plusieurs catégories de champion. Notre projet consiste donc dans un premier temps à analyser un grand nombre de matchs pour y extraire les données concernant les différents objets achetés par chaque champion. Nous souhaitons ensuite observer des clusters qui représenteront les différentes catégories de champion à l'aide de méthodes de clustering non supervisées. Certains champions appartiennent à plusieurs catégories à la fois (et ont donc plusieurs choix d'items très différents mais tous viables), et nous espérons pouvoir observer et détecter sur nos clusters ces champions.

Plusieurs étapes de la recherche de données:

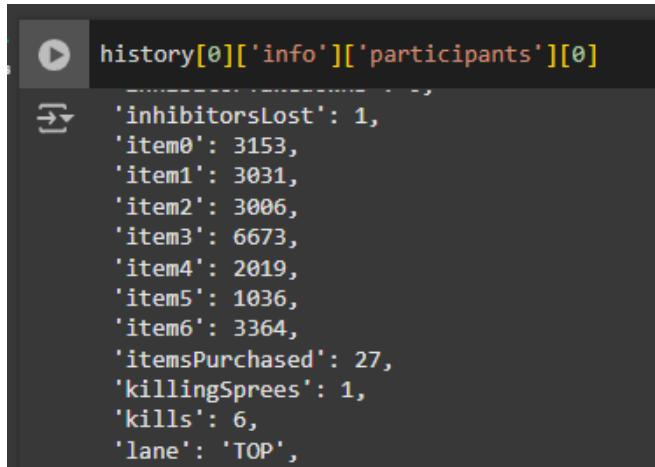
Il y a d'une part la récupération des données

- Le API de Riot nous permet d'avoir l'historique des parties d'un seul joueur à la fois, on va donc récupérer l'historique de plusieurs joueurs afin d'avoir un dataset plus conséquent.
- On fait une requête au travers du site pour récupérer les données des parties de ce joueur
- On observe la structure des données pour en récupérer celles qui nous intéressent (parmis des milliers d'infos pour 1 game)

D'autre part on organise notre dataset pour avoir les données sous la forme souhaitée

- On les range dans un premier dataset qui contient autant de lignes que d'instances de parties relevées x les 10 joueurs
- On retouche le dataset pour ne pas avoir de doublons, en effet il se peut que l'on ait récupéré les historiques de deux joueurs qui se sont rencontrés et donc, qu'on ait deux fois la même partie. On supprime donc toutes les lignes qui ont le même tuple (MatchID, champion) car un champion ne peut pas être choisi deux fois dans un même match.
- On peut drop la colonne match id. Il nous reste alors un dataset entier avec le noms de champions et l'achat ou non des items
- (On appliquera l'ACP sur les colonnes des items pour n'en avoir plus que 2)

Lorsqu'on envoie la requête au Riot API, on obtient un très grand nombre de données. Afin d'avoir accès aux items, on va dans la première partie de l'historique [0] puis on accède aux 'info' ensuite aux 'participants' qui sont les joueurs et dans le cas suivant le premier [0] qui est le Toplaner (joueur du haut).



```
history[0]['info']['participants'][0]
{'inhibitorsLost': 1,
 'item0': 3153,
 'item1': 3031,
 'item2': 3006,
 'item3': 6673,
 'item4': 2019,
 'item5': 1036,
 'item6': 3364,
 'itemsPurchased': 27,
 'killingSprees': 1,
 'kills': 6,
 'lane': 'TOP',}
```

Ensuite on ajoute [item0] pour avoir l'id associé aux premier item acheté par le joueur. Et ainsi de suite pour les autres items.

Site du lien entre items et les id: [ici](#)

Pour la première partie du projet, nous avons pour objectif de classer les champions selon leur style de jeu, nous avons ainsi décidé de ne garder que les items. Une observation du dataframe ressemble donc à ceci :

[ 'Champ Name', 'item1', 'item2', ... , 'item119 ']

où

'Champ Name' est le nom du champion.

'item1' vaut 1 si l'item1 a été pris lors de cette partie, 0 sinon.

Deux possibilités s'offrent à nous :

- considérer qu'une observation est définie par les items qui ont été pris par un champion dans une seule partie: une observation est définie par le tuple "(partie ,champion)" et nous aurons (10\* nombre de match) observations  
un exemple d'observation serait : ["warwick", 1,0,0,0,1, ... ]
- regrouper toutes les lignes concernant un même champion ainsi, une observation correspond à un champion. Nous aurons donc 168 observations.  
un exemple d'observation serait : ["warwick", 0.3 ,0.01 ,0 ,0.04 ,0.2, ... ]

Pour la deuxième partie du projet, nous allons déterminer si tous les champions sont bien équilibrés. Nous allons donc prendre tous les champions d'une même catégorie et les comparer. Nous prendrons donc cette fois toutes leur statistique de jeu comme les dégâts infligés, dégâts totaux subis, or gagné... Nous effectuerons des tests statistiques pour ne garder que les données pertinentes.