INF2251 - Projet Apprentissage Automatique

Emilio Raimond & Louis Francois–Downey

Notre projet portera sur le jeu League of Legends. C'est un jeu d'équipe en ligne en 5 contre 5. Vous pouvez si vous le souhaitez regarder un tutoriel de 3 minutes si vous avez envie de vous familiariser avec le jeu pour avoir une meilleure compréhension globale sur le thème de notre projet.

Sommaire :

Jalon 1 : Preprocessing

1. Comment le jeu de données a-t-il été collecté ?

Pour collecter nos données, nous sommes directement allé les chercher à la source en se branchant à l'api du jeu. On y récupère les noms des meilleurs joueurs du serveur européen. On fait ce choix de choisir les meilleurs joueurs car les stratégies de jeu changent moins et les parties sont moins hasardeuses qu'à bas niveau. Il faut aussi prendre en compte les restrictions de l'api concernant le nombre de requêtes, on a pour ça pu utiliser plusieurs clefs api privées pour faire les requêtes en simultanées.

2. Quel est le domaine d'application ?

Notre domaine d'application est la prédiction de matchs compétitifs.

3. Quelle est la tâche d'apprentissage ?

Notre but ici sera de prédire l'issue d'une partie du jeu League of legends.

4. Quelles sont les variables explicatives ?

L'api fournit cependant une quantité élevée de données pour notre projet (environ 300 par joueur). Ca ferait donc 300 x 10 = 3000 paramètres car on a 10 joueurs dans un match. On procède donc à supprimer tous les paramètres que nous, "experts" en League of legends pensons inutile à la prédiction d'une victoire .

Une fois cela fait on a toujours une dimension trop élevée alors on a fait le choix de cumuler les statistiques numériques des joueurs par équipe ce qui divise le nombre de variables explicatives par 5. On y perd en précision mais compte tenu des restrictions de taille de notre dataset pour le projet et la complexité de League of Legends cela est la meilleure solution que nous avons trouvé.

On obtient donc notre dataset pour lequel une ligne représente un match, une première série de colonne représente les statistiques de l'équipe bleue (paramètres finissant par _0) et à la suite de celle-ci une autre série de colonne se terminant par "_1" représentant l'équipe rouge.

On a x variables explicatives par équipe donc x*2 au total, elles représentent pour chaque équipe des informations concernant :

- L'économie
- Les éliminations et morts
- Les dégâts donnés et reçus
- Les bâtiments (permet de finir une partie)
- Les objectifs neutres (monstres donnant des bonus pour gagner la partie)
- La communication (pings)
- Les sorts (esquives de sorts, sorts touchés, sorts lancés etc…)

5. Quelle est la variable de réponse (type) ?

Une partie peut se conclure de deux manières, win = 0 si l'équipe bleue à perdue et win = 1 si l'équipe bleue à gagner.

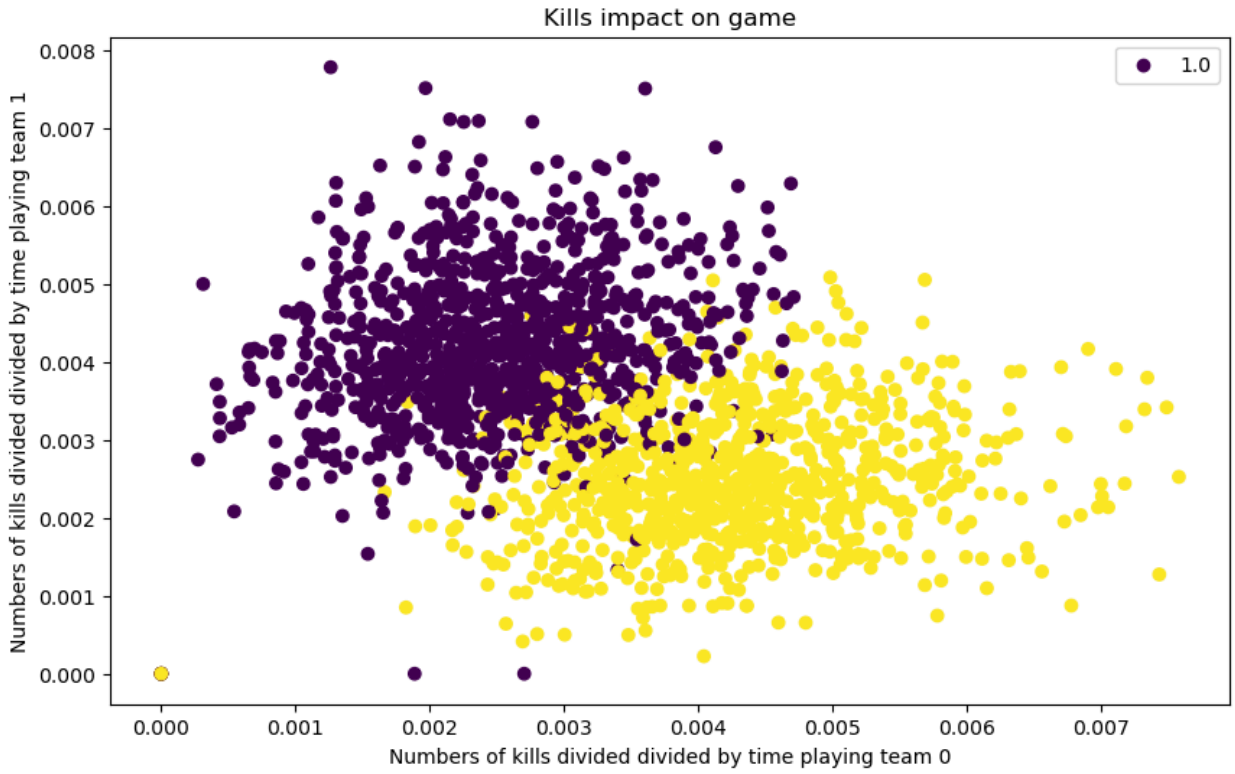
6. Description visuelle du jeu de données

Nous allons visualiser les principaux axes du jeu cités ci dessus pour avoir une intuition. Sur le notebook, commencer les cellules à partir du markdown "Visualiser les variables explicatives" qui importera directement le dataset sans passer par l'api. On a pris un échantillon de 2000 matchs sur notre dataset de ~3900.

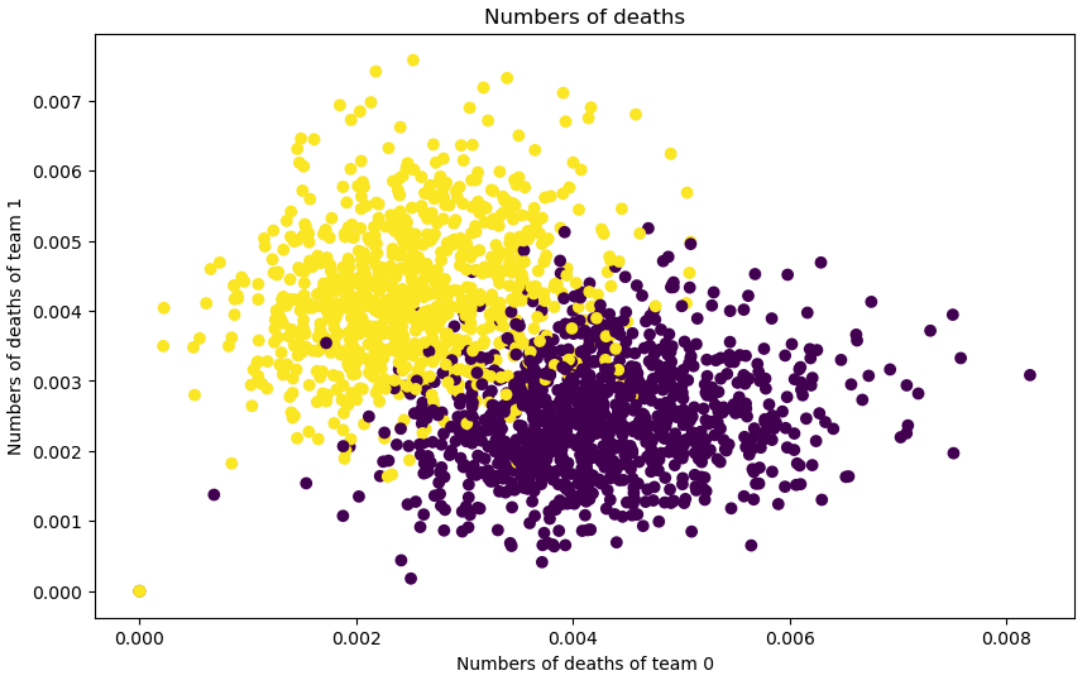
Sur les graphiques, un point jaune signifie qu'il y a eu une victoire de l'équipe bleue, un violet sera une victoire de l'équipe violette.

Pour certains paramètres exemple (Golds earned) , il sera important de diviser ces nombres par le temps de la partie pour être plus représentatifs.

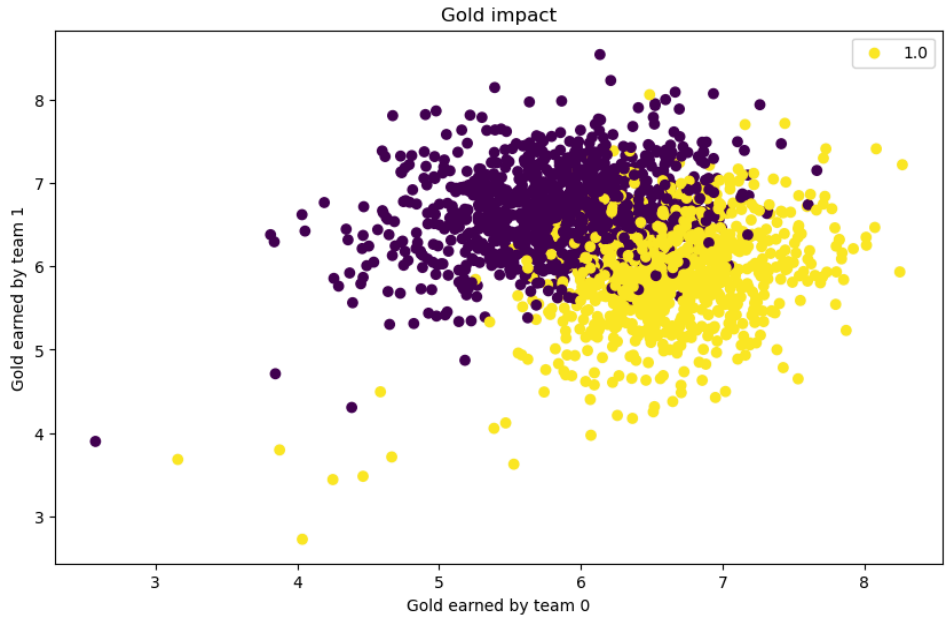
Graphe 1 : Impact des kills d'une équipe sur la partie



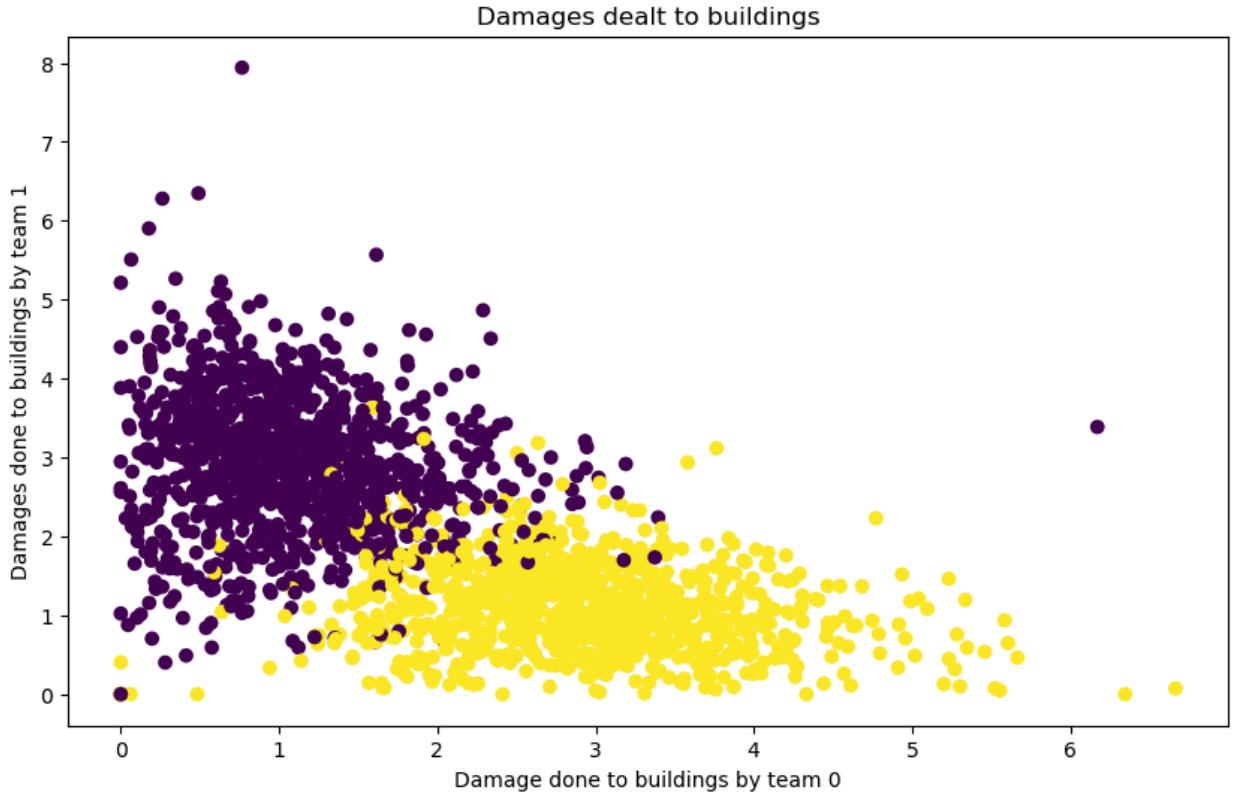
Graphe 1 bis : Impact des morts d'une équipe sur la partie (complémentaire à nombre de kills)



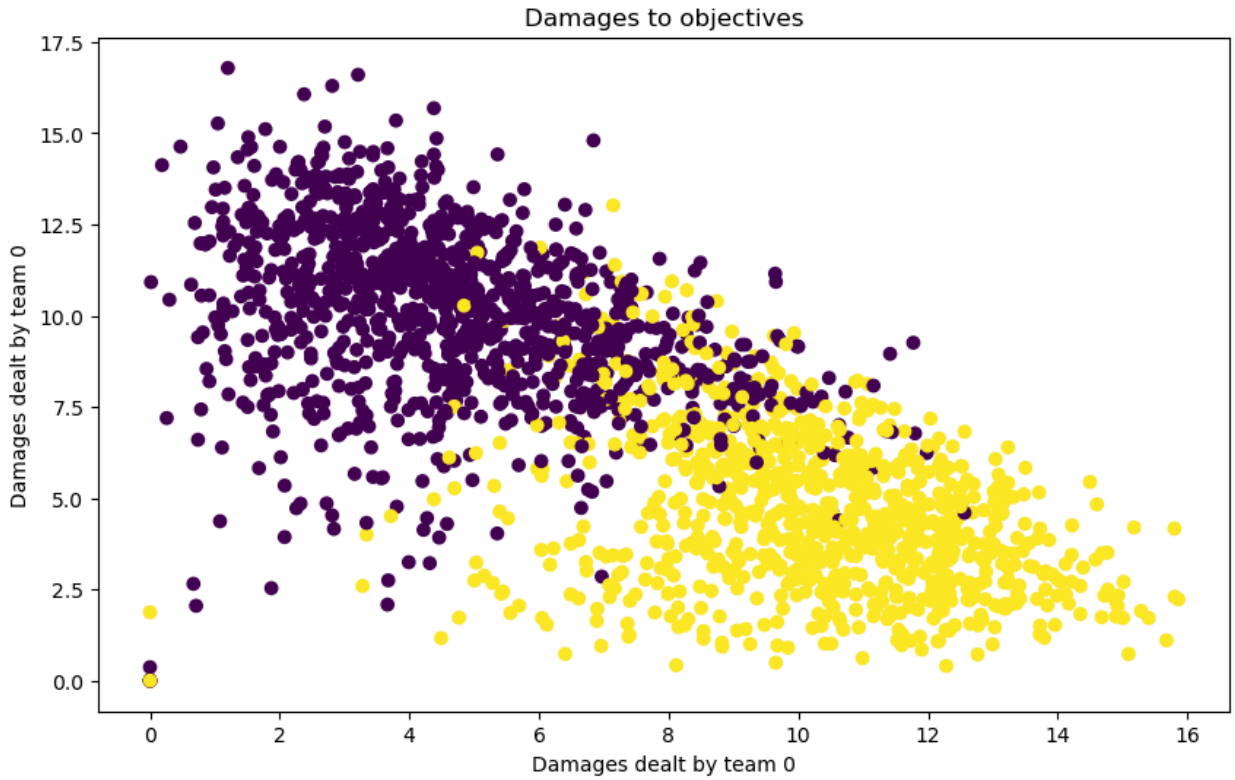
Graphe 2 : Impact des golds dépensés par une équipe sur la partie.



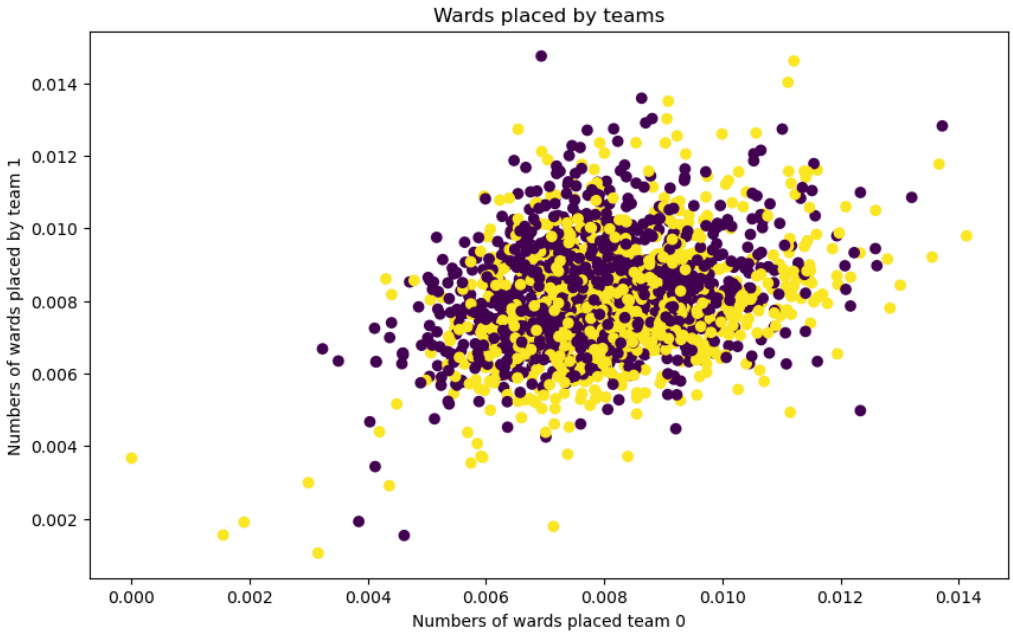
Graphe 3 : Impact des dégats réalisés sur les bâtiments par une équipe sur la partie.



Graphe 4 : Impact des dégats réalisés sur les objectives par une équipe sur la partie.



Graphe 5 : Impact des balises de vision (objet dans le jeu permettant de donner des informations sur la position ennemie) sur la partie



On voit que poser des balises de vision rend l'issue d'une partie assez incertaine.

Voici donc les principaux axes sur lesquels s'oriente le jeu et leurs impacts. Il faut cependant être prudent car league of legends est un jeu où l'effet boule de neige est très présent. Au plus on a de golds, au plus on en gagne. C'est donc normal que ce soit le graphique le plus révélateur de l'issue de la partie.

7. Description statistique du jeu de données

Voir la cellule "Histogrammes", la grande majorité de nos variables suivent une loi normale. On a très peu de valeurs aberrantes, dû notamment au choix d'avoir pris des parties de très haut niveau. On a par contre certains challenges ou pings n'arrivant que très rarement, il sera peut être utile de les supprimer. On ne peut pas faire de matrice de corrélation dû à notre nombre trop élevé de features. Mais étant connaisseurs du jeu, on est en droit de se demander si autant de features sont utiles car la plupart comme "Nombre de monstres tués" par exemple qui est une sous feature de du nombre de golds gagnés. On pourrait aussi diminuer par 2 les features en faisant un gap entre l'équipe bleue et rouge au lieu d'avoir leurs deux statistiques séparément. Pourrait on aussi toutes les statistiques par le temps joué pour n'avoir aucune différence entre les statistiques sur des parties courtes ou longues.

8. Métrique d'évaluation

Nous pourrons utiliser le F1 score pour évaluer notre modèle. On cherchera le meilleur compromis entre la précision et le Recall.