RAPPORT PRELIMINAIRE PROJET FOUILLES DE DONNEES

MUSENGA Christ et ELKRIEFF Benjamin

MASTER 2 INGENIERIE DES ALGORITHMES ET DES PROGRAMMES

1) Introduction

Le but de notre projet est de prédire le résultat d'un ou plusieurs match de football (Ligue 1).

Le résultat d'un match est défini par 3 constantes :

- H : victoire de l'équipe à domicile
- D : égalité entre les 2 équipes
- A : victoire de l'équipe à l'extérieure

Le but final est déjà de voir quelle est le pouvoir de prédiction de nos algorithmes à partir des données que l'on possède, de voir le taux de bonne prédiction de ces algorithmes.

Dans ce rapport, nous nous sommes inspirés d'un ouvrage intitulé « ***Data Science : fondamentaux et études de cas*** » (Auteurs : *Eric Biernat* et *Michel Lutz*).

Nous marquerons chaque phrase inspirée de cet ouvrage par le symbole (*). Nous tenons à préciser que nous avons bien compris les notions dont nous nous inspirons.

2) Méthodes

Pour la réalisation de ce projet, nous avons récupéré sur le web (http://www.football-data.co.uk/) plusieurs fichiers d'extension '.csv', plus exactement un csv pour chaque saison (nous avons récupéré les fichiers des saisons depuis 1993/1994 jusqu'à la saison 2016/2017).

Néanmoins les fichiers des saisons avant 2007/2008 sont incomplets par rapport à ceux des saisons suivantes. Nous les avons donc exclus afin d'éviter d'avoir un trop grand nombre de valeurs Nulles, ce qui serait contraignant pour le bon déroulement de l'algorithme. Ainsi nous avons appliqué les algorithmes sur le total de 3520 matches (tous les matchs à partir de la saison 2007-2008 jusqu'à cette saison actuelle, c'est à dire 2016-2017).

Dans les fichiers qu'on a récupérés se trouvent les données des matches :

- l'id du match
- le numéro de la journée (allant de 1 à 38 pour une saison de Ligue 1)
- la date
- les deux équipes s'affrontant

- le nombre de buts marqués par chaque équipe à la fin du match
- le nombre de buts marqués par chaque équipe à la mi-temps
- le nombre total de tirs pour chaque équipe
- le nombre total de tirs cadrés pour chaque équipe
- le nombre de corners pour chaque équipe
- le nombre de fautes commises par chaque équipe
- le nombre de cartons jaunes concédés par chaque équipe
- le nombre de cartons rouges concédés par chaque équipe

Nous en venons à présent à la partie traitant des méthodes algorithmiques utilisées.

Nous avons choisi d'utiliser le langage **Python** pour la partie programmation de notre projet car nous avions déjà travaillé avec en TP's, et avions déjà vu plusieurs exemples d'implémentation avec vous dans ce langage. De plus, il existe dans ce langage différents algorithmes de Data Mining qui sont simples d'utilisation.

Nous avons choisi d'opter pour des algorithmes d'apprentissage supervisés (pour entraîner notre algorithme à prédire les bons résultats, que l'on connaît à l'avance).

Nous avons en particulier opté pour l'utilisation des forêts aléatoires, algorithme très puissants se basant sur les arbres de décisions. L'utilisation d'un seul arbre de décision peut poser problème dans le cas ou l'on change des exemples(*). On risque donc de faire du sur-apprentissage. L'intérêt de la forêt aléatoire est que l'on se base maintenant sur plusieurs arbres de décisions, dont la décision à chaque nœud est basé sur un sous-ensemble de variables tirés au hasard. Cet algorithme est facilement parallélisable et est basé sur un vote majoritaire des arbres ce qui le rend très robuste. (*)

C'est ainsi que notre choix s'est porté sur les forêts aléatoires, qui sont plus proches d'un raisonnement humain et donc beaucoup plus intuitifs.

Cependant, nous verrons dans la suite que cette méthode est très gourmande en données et que les résultats en dépendent beaucoup.

Dans python : **RandomForestClassifier()** de la librairie ***sklearn.ensemble*** avec les paramètres suivants :

- n_estimators = 100 : on utilise 100 arbres de décision
- criterion = gini
- max_depth = best_depth : la meilleure profondeur d'arbre

- min_sample_split : le nombre minimum d'observation à chaque nœud avant séparation (*)
- max_features = sqrt : le nombre de variables qu'on tire aléatoirement pour chaque arbre (*)
- n_jobs = 4 : le nombre de CPU utilisés pour paralléliser le calcul

On utilise comme critère de split l'indice de Gini. En effet, le but est de séparer le plus vite possible, c'est-à-dire le plus haut dans l'arbre de décision, la classe a plus représentée dans nos données ce qui aura pour effet de nous donner de très bons arbres de décision. (*)

Aussi, nous utilisons la validation croisée, c'est-à-dire que l'on découpe nos données en plusieurs sous ensemble, et que l'on effectue le travail apprentissage-test sur chacun de ces sous-ensembles, et en testant différentes valeurs de profondeur d'arbre, afin de trouver celle qui nous donne les meilleurs résultats. Ainsi, on pourra effectuer un travail apprentissage-test sur l'ensemble total avec la meilleur profondeur d'arbre.

Dans python : **StratifiedKFold()** de ***sklearn.model_selection***

Au début, on testait directement notre forêt sur les données que l'on a obtenues. On arrivait avec un taux de prédiction de 94 %. Mais nous nous sommes rendu compte que l'on faisait du sur-apprentissage. En effet, en donnant des données de fin de match, en particulier le nombre de buts qu'il y a eu pour chaque équipe en fin de rencontre, il est évident que l'algorithme prédira toujours le bon résultat (Si l'équipe **A** marque $\boldsymbol{x}$ buts et l'équipe **B** marque $\boldsymbol{y}$ buts avec $\boldsymbol{x > y}$, l'algorithme détectera à chaque fois que si une équipe **A** a un nombre de buts supérieur à celui de l'autre équipe **B** alors forcément la victoire sera attribué à l'équipe **A**).

Nous avons donc conclu qu'il fallait travailler sur nos données afin d'en produire des nouvelles, plus exactement des statistiques pour chaque équipe rendant compte de plusieurs paramètres avant le déroulement d'un match à l'instant $\boldsymbol{t}$.

On a ainsi pu produire les données suivantes, pour chaque équipe, à l'instant $\boldsymbol{t}$:

- nombre de buts marqués et concédés à la fin et à la mi-temps d'un match, en moyenne
- nombre moyen de tirs (cadrés et non cadrés)
- nombre moyen de fautes commises et concédées
- nombre moyen de corners obtenus et concédés
- nombre moyen de cartons jaunes et cartons rouges concédés

- nombre de points de l'équipe
- nombre de victoires à domicile et à l'extérieur
- nombre de défaites à domicile et à l'extérieur
- nombre de matchs nuls concédés à domicile et à l'extérieur
- nombre de victoires, défaites, matchs nuls consécutifs
- nombre de victoires consécutives à domicile et à l'extérieur
- nombre de défaites consécutives à domicile et à l'extérieur
- nombre de matchs nuls consécutifs à domicile et à l'extérieur

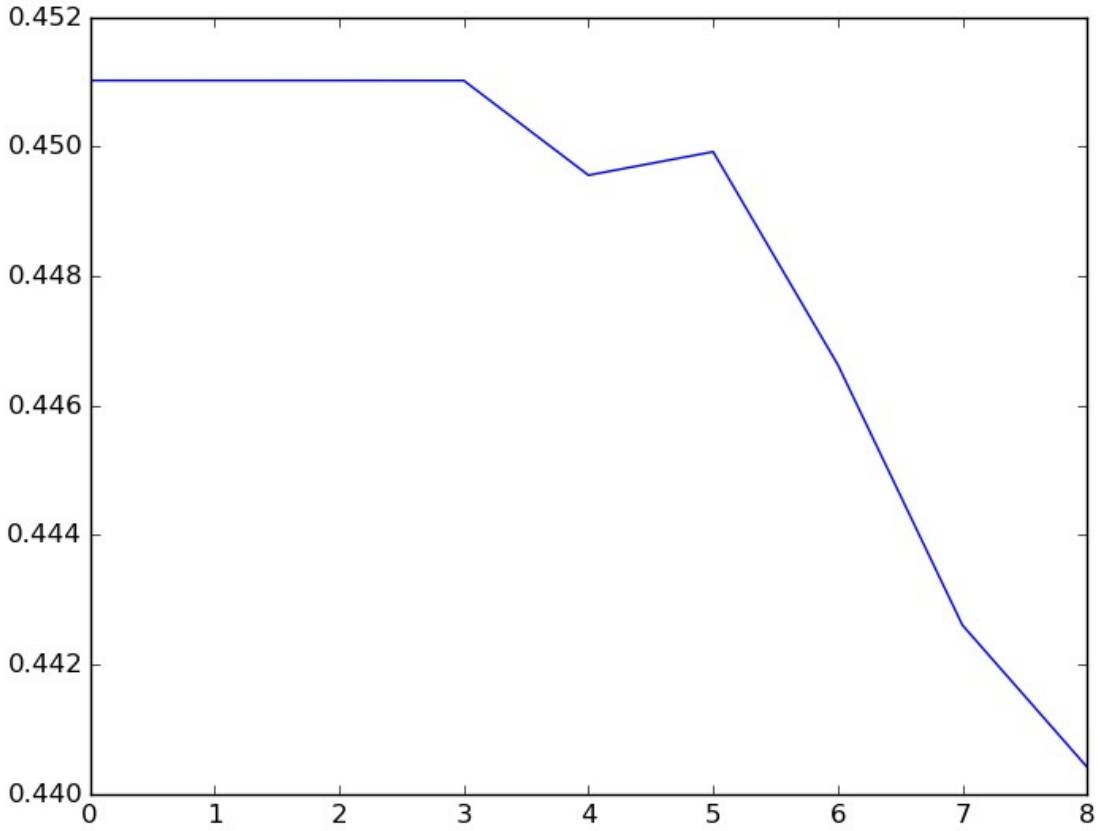
Nous avons ensuite construit un grand fichier csv (en programmant des algorithmes) avec chaque ligne correspondant à un match avec les statistiques de l'équipe A et celle de l'équipe B.

Les résultats en sortie sont bien moins bons. Nous allons en discuter dans la partie suivante.

3) Résultats

Après Exécution de la forêt aléatoire, on obtient une précision finale de 0.45 %

Voici un graphique montrant la variation des résultats en fonction de la profondeur d'arbre (de 1 à 10)



On en déduit que plus les arbres sont profonds, moins les résultats sont bons.

Nous mettrons plus d'informations et de graphiques dans le rapport final.

4) Discussion

Après une réflexion et une documentation un peu plus approfondie, on a pu réfléchir à ce qui a pu donner de tels résultats.

La première interprétation que l'on a fait est de se dire que l'on ne dispose pas d'assez de données, que se soit en terme de variables descriptives ou en terme de nombre de matchs, les forêts aléatoires étant performantes pour un grand nombre de lignes et de colonnes.

La deuxième interprétation que l'on a faite après documentation dans notre ouvrage, est que l'on dispose de plusieurs variables explicatives mais qui sont colinéaires entre elles. Par exemple, le nombre de victoires à domiciles, le nombre de victoires consécutives, le nombre de victoires consécutives à l'extérieur, le nombre de victoires total, etc. Rien que ces variables ont une forte corrélation entre elles, ce qui peut rendre l'interprétation du modèle difficile.

Une solution serait de faire une sélection de variables en prenant celles qui expliquent

le mieux le modèle.

Ainsi allons-nous sûrement tester d'autres méthodes de machine-learning (nous avons aussi beaucoup de mal à trouver de bonnes données sur internet mais nous allons continuer à chercher jusqu'à la fin)

5) Conclusion

Nous avons donc cherché des données (après de longues recherches) nous avons tester notre forêt dessus mais cela nous menait à du sur-apprentissage donc nous avons établit des statistiques à partir de ces données, puis relancé la forêt et obtenus de moins bons résultats. Nous pensons que tout ceci est normal. Les sports collectifs dépendent de beaucoup de paramètres dont nous ne disposons malheureusement pas (ne serait-ce que les joueurs).

6) Equipe

Nous n'avons pas vraiment séparé le travail, car nous avons toujours été physiquement ensemble pour travailler sur ce projet. Nous avons toujours tout fait à deux, en se déléguant à tour de rôle la partie programmation.