RAPPORT FINAL PROJET FOUILLES DE DONNEES

MUSENGA Christ et ***ELKRIEFF Benjamin***

MASTER 2 INGÉNIERIE DES **ALGORITHMES** ET **DES PROGRAMMES**

1) Introduction

Ce Rapport fait suite au rapport préliminaire.

Nous tenons à préciser avant toute chose que tout ce qui sera dit est le fruit d'une étude approfondie de plusieurs ouvrages et de concepts sur lesquels nous avons beaucoup réfléchi, et n'est en aucun cas de la paraphrase.

Dans ce rapport, nous nous sommes inspirés des ouvrages intitulés « ***Data Science : fondamentaux et études de cas*** » (Édition Eyrolles) et « ***Big Data* et *Machine Learning*** » (Édition Dunod).

Ce premier rapport donnait une vue d'ensemble de notre projet, sans trop rentrer dans les détails.

Le but de notre projet est de prédire le résultat d'un match de football de la Ligue Professionnelle de football Française.

Nous nous chargerons ici de présenter notre projet de manière plus détaillée.

Nous commenceront tout d'abord par une étude sur les données. Cette étape est primordiale pour le data scientist. En effet, négliger cette étape pourrait amener à des incohérences et le dépourvoir de toute capacité à analyser et interpréter la situation (il est important de pouvoir s'apercevoir qu'on a un problème de sur-apprentissage par exemple, et de pouvoir l'expliquer. Le sur-apprentissage est souvent causé par les données même si d'autres aspect influent là-dessus).

Le but sera d'analyser nos données, et de les transformer afin d'obtenir des variables qui ont du sens pour nos algorithmes.

Notre projet est un problème d'apprentissage supervisé.
Nous utiliserons les forêts aléatoires et nous expliquerons les raisons de ce choix.

Nous présenterons ensuite les résultats de notre algorithme et ferons quelques essais sur des matchs qui se sont déjà déroulés mais qui ne sont pas dans nos données. Tout ceci sera présenté de manière transparente. Nous feront l'analyse et la critique de ces résultats dans la partie discussion.

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.

Nous marquerons chaque phrase inspirée de nos ouvrages 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).

(Nous avions entre temps, essayé d'extraire les données du site www.lfp.fr contenant aussi des données sur les joueurs pour chaque matchs, mais nous avons fait face à des difficultés dont nous ne pouvions pas borner le temps nécessaire pour les résoudre, nous nous devions d'avancer et de nous concentrer sur le sujet même du projet.)

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 ne se fait plus de toute manière car cela mène vite à du sur-apprentissage)

Expliquons ce choix :

Par exemple, nous aurions pu utiliser la régression linéaire, à priori plus simple et plus facile à entraîner. Cependant, nous aurions été obligés de travailler encore plus sur les variables afin d'obtenir de bons résultats. Par exemple, dès qu'une variable à deux sens différents en dessous et au-delà d'un certain seuil, le modèle linéaire est inadapté et nous devrions changer la variable pour augmenter la performance (c'est le ***feature engineering (*)***).

Nous aurions aussi pu utiliser le très puissant algorithme du Support Vector Machine.

La méthode utilisée par cette algorithme est très intuitive et facile à la compréhension. Il s'agit de transposer les données vers une plus grande dimension, séparant mieux les classes car le passage à une dimension supérieure permet de trouver un modèle linéaire pour les séparer. Cependant, l'utilisation de cette méthode requiert l'élaboration d'une fonction noyau **K** qui est difficile à trouver. Les machines à vecteur de support sont utiles lorsque l'on fait fasse au fléau de la dimension (*), mais nous ne faisons pas face à ce problème : notre nombre de variables est petit. Donc inutile d'utiliser cette méthode.

Les forêts aléatoires apparaissent pour nous comme étant la méthode la plus intuitive pour répondre au problème posé. En comprenant le principe d'un arbre de décision, la compréhension des forêts aléatoires découle directement. On se base sur plusieurs arbres de décision qui auront chacun une vision parcellaire du problème, pour à la fin procéder à un vote final de tous les arbres. Elles se basent sur un principe très puissant qui consiste à dire que la majorité à toujours raison.

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

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 = 1000 : on utilise 1000 arbres de décision
- criterion = gini
- 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. (*)

Nous n'avons pas besoin d'utiliser la validation croisée en utilisant cette méthode, RandomForestClassifier trouve lui même la profondeur adéquate pour chaque arbre.

Au début, nous testions 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 $\boldsymbol{t}$:

- la différence de but de chaque équipe
- la moyenne des tirs cadrés
- la moyenne des ratio tirs Cadrés / total tirs
- la moyenne des corners
- la moyenne des cartons rouges (expulsion d'un joueur)
- nombre de victoires
- nombre de défaites
- nombre de matchs nuls
- nombre de points en championnat

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.

Nous nous expliquons sur ces choix :

Avant, nous avions par exemple la moyenne des buts marqués pour chaque équipe.

Mais une équipe forte peut tout le temps gagner que par 1 but à 0 (moyenne de buts faible) et une équipe faible peut tout le temps perdre 2 – 3, 3 – 4 , etc … (moyenne de buts forte). La moyenne de buts marqués n'est donc pas un bon critère de séparation des classes et peut induire le modèle en erreur.

Nous avons la chance, en football, d'avoir à notre disposition d'un paramètre toujours constant : une équipe du haut du classement présente toujours une différence de buts positive et une équipe du bas de classement présente toujours une différence de buts négative. On peut donc considérer cela comme un meilleur critère de séparation.

Autre exemple : nous n'incluons pas les fautes commises / concédées.

Ce paramètre peut clairement induire le modèle en erreur. En effet, si la clé d'un match est l'impact physique, une équipe faisant beaucoup de fautes implique une certaine agressivité et à donc des chances de gagner le match. A l'inverse, si la clé du match repose sur la technique et la capacité à dribbler, on peut remarquer l'équipe qui fait beaucoup de fautes les commet car elle est submergées par les attaquants adverses, et n'arrive pas à bien défendre à la régulière. On peut en déduire qu'elle est susceptible d'encaisser beaucoup de buts et de perdre le match. Ce critère est donc à proscrire.

Le nombre de tirs cadrés est important, mais il est aussi important de le comparer au nombre de total tirs. Cela nous permet d'en déduire des choses sur le réalisme de l'équipe. Si une équipe fait 3 tirs cadrés sur 3, ce n'est pas pareil que si elle fait 3 tirs cadrés sur 20.

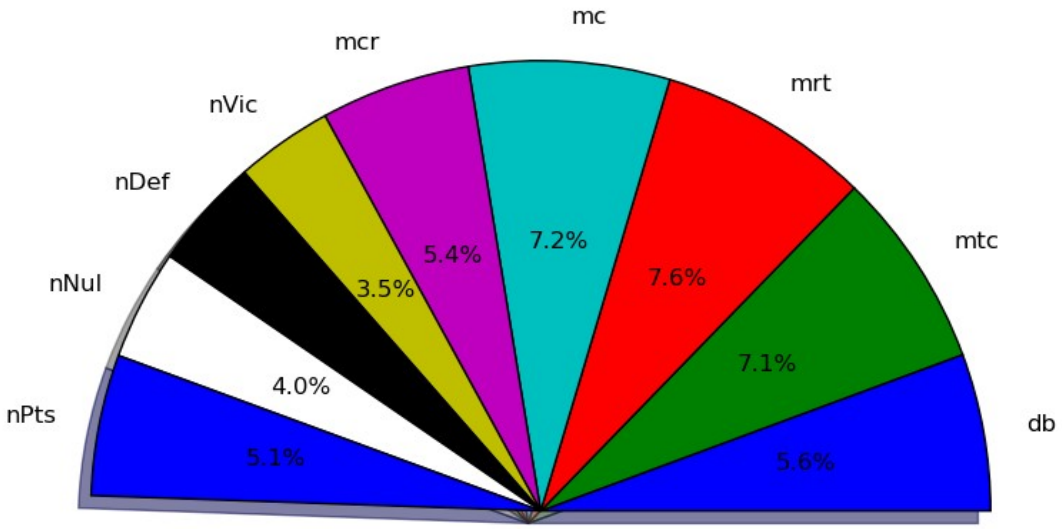
Nous avons choisis d'ignorer le nombre de cartons jaunes, qui en général dans le football ne révèlent rien sur le déroulement d'un match. Néanmoins, le nombre de cartons rouge est important. L'expulsion d'un joueur mettant en infériorité numérique son équipe, cela influe souvent sur le déroulement d'un match car cela perturbe les schémas tactiques des deux équipes.

Le nombre de victoires, de défaites, de nuls, de points et de corners apportent des informations supplémentaires qui peuvent affiner nos résultats.

3) Résultats

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

Voici un graphique montrant l'importance des variables dans la prédiction du résultat final. A noter que l'équipe domicile et l'équipe extérieure possèdent les mêmes variables.



Légende :

- db : différence de buts
- mtc : moyenne des tirs cadrés
- mrt : moyenne des ratios tirs Cadrés / total tirs
- mc : moyenne des corners
- mcr : moyenne des cartons rouges
- nVic : nombre de victoires
- nDef : nombre de défaites
- nNul : nombre de matchs nuls

- nPts : nombre de points de l'équipe

Comme prévu, on voit que la différence de but, la moyenne des tirs cadrés, la moyenne des ratios tirs Cadrés / total tirs, la moyenne des corners et des cartons rouges ont une plus grande influence sur le résultat.

4) Discussion

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

Une interprétation que l'on peut faire 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.

Les sports collectifs sont difficiles à prédire. Quand nous avions choisi ce sujet, nous connaissions peu de choses au sujet du datamining et des arbres de décisions. C'est au fur et à mesure que nous avons réussi à appréhender le fonctionnement d'une telle méthode. Il nous faudrait beaucoup plus de données comme :

- l'affluence dans le stade
- le volume sonore (plus les supporters sont bruyants plus l'équipe à domicile se galvanise et à de chances de l'emporter)
- l'opinion des médias sur l'équipe à l'instant t. Plus une équipe est critiquée par les médias et plus les joueurs vont vouloir prouver ce qu'ils valent. Moins une équipe est déclarée favorite et moins elle a à perdre, plus elle joue sans complexe. Une équipe étant présentée comme le gros favori à une plus grosse pression sur elle.
- Le fantasme absolu : des statistiques sur les joueurs à l'instant t. Il est clair que l'élaboration de telles données présente une certaine complexité et demande beaucoup de temps. Nous aurions aimé le faire mais le temps ne nous l'a pas permit. Nous pouvons même aller encore plus loin, si nous avions des données personnelles sur chaque joueur telles que l'état mental, la motivation, nous pourrions avoir de bien meilleurs résultats.

Nos variables sont de bonnes variables de coupure, mais ne permettent pas à elles seules d'élaborer de bonnes règles dans les arbres de décisions.

Nous avons pensé, afin d'obtenir de meilleurs résultats, à basculer complètement sur une autre méthode : faire des clusters de matchs. Chaque clusters correspondraient à un résultat. Nous aurions pu le faire si nous avions eu plus de temps.

5) Conclusion

Nous avons donc cherché des données (après de longues recherches) nous avons testé 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.