Prédire l'issue d'un match de football

Annabelle FILIP
Florine PRITZY

Aix-Marseille School of Economics
18 janvier 2021

Table des matières

Résumé

Dans ce rapport nous allons appliquer différents algorithmes de Machine Learning pour prédire l'issue d'un match de football (victoire, défaite ou match nul) et déterminer quelle est la meilleure méthode. Nous présenterons les résultats obtenus avec une régression logistique et une régression logistique avec une pénalité, puis ceux obtenus avec la méthode de machine à vecteurs de support (plus communément appelé Support vector classification), ceux d'une forêt aléatoire (Random Forest Classification), ceux utilisant la méthode des K-voisins les plus proches (K Nearest Neighbors classification) et enfin ceux obtenus avec un réseau de neurones (Neural Network Classification).

Notre but sera également d'analyser les matchs de Ligue 1 et les variables à notre disposition afin de conseiller l'équipe de l'Olympique de Marseille sur les stratégies à adopter pour augmenter ses chances de remporter un match.

Nous remarquons que la meilleure modélisation possible est le Support Vector Machine, une méthode de Machine Learning permettant de classifier nos matchs en 3 catégories. Grâce à cette technique nous arrivons à prédire 72% des matchs remportés par l'équipe extérieur. Quant aux matchs remportés par l'équipe à domicile, ils sont bien prédits une fois sur 2. L'algorithme a cependant quelques difficultés à prédire les matchs nuls. Mais, nous avons constaté dans la littérature, que les modèles de prédiction des scores (pour les paris notamment) ou de l'issue du match ont en moyenne une qualité de prédiction de l'ordre de 55%.

Ainsi, à la fin de ce rapport, nous avons essayé de prédire avec notre meilleur modèle, l'issue de 3 matchs, qui n'ont pas encore été joués. Nous vous garantissons pas que ces prédictions seront la vraie issue des matchs, mais nous espérons que vous aurez nos résultats en tête si vous en êtes spectateurs.

1. Introduction

Le football est l'un des sports les plus populaires de la planète réunissant à chaque événement nationaux et internationaux, des millions de supporters. Comme le disait le joueur anglais Gary Lineker[1]: *"Le football est un jeu simple; 22 hommes courent après un ballon pendant 90 minutes et, à la fin, ce sont les Allemands qui gagnent"*. C'est également la seule fédération sportive à avoir dépassé le seuil symbolique des deux millions de licenciés[2]. D'un point de vue économique, en France, les 43 clubs professionnels ont généré 35000 emplois et plus de 7,5 milliards d'euros de chiffre d'affaires, pour la saison 2015-2016. C'est aussi l'un des sports qui développe de plus en plus l'analyse des données tant dans un but marketing, que sur celui des performances des joueurs et des équipes et des prédictions autour des matchs.

En 2014, lors de la Coupe du Monde de football organisée au Brésil, Bing a réussi à prédire le bon résultat pour 15 des 16 matchs de poule. Il a également pronostiqué la victoire du FC Barcelone lors de la Ligue des Champions en 2015. Ce moteur de recherche de Microsoft a donc pu fournir des prévisions précises sur l'issue d'un match de football en utilisant des modèles de Machine Learning avancés.

Ces dernières années, de nouveaux types de données ont été recueillis sur les matchs, dans divers pays et de multiples compétitions, qui comprennent par exemple des informations sur chaque tir ou passe effectué pendant le jeu. La collecte de ces données a placé la Data Science au premier plan de l'industrie du football, avec de nombreuses utilisations et applications possibles : analyser la stratégie et les tactiques de match, identifier le style de jeu des joueurs, évaluer les joueurs et les dépenses de l'équipe afin de prévoir de nouvelle acquisition de joueurs où encore concevoir et programmer les différentes compétitions (ordre de sélection pour les matchs et différentes poules pour les compétitions internationales). En particulier, le marché des paris a connu une croissance très rapide au cours de la dernière décennie, avec une couverture accrue des matchs de football en direct ainsi qu'une meilleure accessibilité des sites de paris grâce au développement des appareils mobiles et des tablettes. En 2020, le volume total des mises du marché mondial des paris sportifs (légaux et illégaux) peut être estimé entre 500 et 1.000 milliards d'euros.

Sur appel d'offre du club de football français, l'Olympique de Marseille, nous allons essayer, de part nos modèles, de mieux guider le club dans ses décisions. Pour cela, nous allons nous concentrer en grande partie sur le Championnat de Ligue 1. Nous apporterons tout de même une comparaison entre certaines Ligues européennes. Ainsi, pour ce projet, nous avons décidé d'utiliser une base de données disponible sur Kaggle : "*European Soccer Database*", que nous avons fortement modifié, afin d'avoir les variables les plus prédictives possibles. Nous utiliserons des variables telles que la moyenne de poids, la taille ou l'âge des joueurs au sein de l'équipe. En utilisant les informations sur les matchs passés, les caractéristiques des équipes et des joueurs nous présenterons les techniques utilisées et les résultats obtenus. Nous testerons différents modèles afin de maximiser leurs performances prédictives.

[1] Gary Lineker, avant-centre anglais, à la fin du match de demi-finale de la coupe du monde 1990, que l'équipe allemande a gagné
[2] FFF: Fédération française de football, chiffre clé

Nous nous sommes également appuyés sur des modèles déjà existants, qui ont fait leurs preuves pour prédire l'issue d'un match de football. Comme nous allons le voir, une approche dite "indirecte", plus largement utilisée dans la littérature, consiste à prédire le nombre de buts pendant un match en s'appuyant sur des distributions statistiques telles que la distribution de Poisson ou encore la distribution binomiale négative. Une autre approche, plus récemment utilisée dans la littérature, dite "direct", consiste à ne plus modéliser le nombre de buts mais l'issue du match (gagner, perdre, nulle). Nous allons largement nous inspirer de ces méthodes "directes", en nous basant sur des modèles d'apprentissage supervisé. Nous présenterons donc les résultats obtenus avec une régression logistique, puis ceux obtenus avec le Support vector classification (SVM), ceux d'une forêt aléatoire (Random Forest Classification), ceux utilisant la méthode des K-voisins les plus proches (K Nearest Neighbors classification) et enfin ceux obtenus avec un réseau de neurones (Artificial Neural Network).

A notre échelle, nous allons tenter de prédire si lors d'une rencontre sportive l'équipe à domicile gagne, perd ou si l'issue du match se solde par un match nul. Pour cela nous allons appliquer plusieurs algorithmes de Machine Learning et définir lequel apporte la meilleure précision. L'enjeu est important à la fois pour améliorer les performances des joueurs et la compréhension globale de facteurs menant à la victoire. L'analyse des données est essentielle pour comprendre toutes les interactions et valoriser l'utilisation de celles-ci afin d'optimiser les décisions et développer de nouvelles stratégies efficaces. Nous essayerons également de prédire l'issue des matchs de Ligue 1 des semaines à venir, qui n'ont pas encore été joués comme notamment le match [3]Marseille-Lens; Paris-Montpellier ou encore Marseille-Monaco.

Notre dossier sera organisé de la façon suivante : dans un premier temps nous allons décrire les règles d'un match de football, afin d'expliquer tous les termes techniques et principes du jeu. Ensuite nous proposerons, dans la section suivante, une revue de littérature relative à la prédiction de résultats de matchs. Nous présenterons les modèles que nous avons choisi de réaliser dans une quatrième partie. La cinquième section décrit les données utilisées et une des étapes les plus importantes durant la réalisation de ce projet : le prétraitement des données. Nous partagerons ensuite les résultats de nos prédictions et la qualité prédictive de nos modèles dans une dernière section.

[3] Marseille-Lens - Ligue 1 - Mer.20/01, 21:00 ; Paris-Montpellier - Ligue 1 - Ven..22/01, 21:00; Marseille-Monaco - Ligue 1 - Sam..23/01, 21:00

2. Présentation

Dans cette section, nous voulons juste rappeler les règles d'un match de football, les différentes compétitions existantes et l'issue possible d'un match.

- **Règles simplifiées d'un match de football :**
  - Le football est un sport collectif qui se joue avec un ballon sphérique opposant 2 équipes de 11 joueurs et 5 remplaçants, durant 90 minutes.
  - De chaque côté du terrain, une équipe à une cible appelée but dans lequel elle tente de mettre le ballon.
  - Marquer un but donne un point à l'équipe.
  - L'équipe ayant le plus grand nombre de points à la fin du match gagne le match.
  - Si les deux équipes ont marqué le même nombre de buts, le match se termine par un match nul.
  - L'issue du match est donc soit : **gagné / perdu / nul**

- **Format de compétitions des ligues nationales :**
  - Chaque pays européen possède généralement une ligue nationale où les clubs s'affrontent. *En France, il existe la Ligue 1 où des équipes comme l'Olympique de Marseille, le Paris Saint Germain ou l'Olympique Lyonnais s'affrontent. En Allemagne c'est la Bundesliga avec le Bayern de Munich, Borussia Dortmund ou SC Fribourg qui se défient.*
  - Il y a généralement 20 équipes dans chaque ligue, chaque équipe jouant deux fois contre les autres, une fois dans son stade (match "à domicile") et une fois dans le stade de l'équipe adverse (match "à l'extérieur").
  - Gagner un match donne 3 points à une équipe, un match nul donne 1 point à chaque équipe.
  - Si deux équipes sont à égalité au cumul des buts marqués dans un match aller-retour, la qualification va à l'équipe ayant marqué le plus de buts à l'extérieur
  - L'équipe qui obtient le plus grand nombre de points à la fin de la saison remporte le championnat.
  - Il existe également des compétitions d'ordre européenne (tous les clubs européens s'affrontent notamment lors de la *Champions League* et *l'Europa League*).
  - Il faut également bien distinguer le football de club, du football de pays. Ce dernier comporte 2 compétitions principales (la Coupe du Monde et l'Euro).

- **Principales actions d'un match de football :**
  - **But** : un but est marqué lorsque le ballon entre dans le but de l'équipe adverse.
  - **Possession** : la possession représente la fraction du temps pendant lequel une équipe contrôle le ballon dans le match.

- **Penalty et coups francs** : les coups francs ont lieu lorsqu'une faute est commise par l'équipe adverse sur le terrain. Dans ce cas, l'équipe qui a concédé la faute peut jouer le ballon à l'endroit où la faute a été commise. Si la faute se produit à l'intérieur de la surface de réparation (la zone proche du but), un penalty est accordé : l'équipe qui a concédé la faute peut tirer au but à bout portant sans la présence de l'équipe adverse.
- **Carton** : des cartons sont attribués chaque fois que l'arbitre juge qu'une faute est faite. Les cartons jaunes sont attribuées pour des fautes de moindre importance et n'ont pas de conséquence directe. Toutefois, deux cartons jaunes reçus par le même joueur entraînent un carton rouge. Si un joueur reçoit un carton rouge, il doit quitter le terrain, laissant son équipe avec un joueur de moins. Les cartons rouges peuvent également être obtenus directement en cas de faute dangereuse ou dans d'autres circonstances spécifiques.
- **Corner** : les corners sont attribués à l'équipe adverse lorsqu'une équipe frappe le ballon en dehors du terrain derrière son but. Dans ce cas, la balle est placée sur le coin du terrain et peut être frappée sans qu'aucun autre joueur ne soit présent.

3. Revue de littérature

La prévision des résultats d'un match de football (score ou issue du match) est un thème de recherche important depuis le milieu du 20ème siècle. Les premières approches et idées de modélisation statistiques ont été réalisés par Moroney (1956) et Reep (1971) , qui ont utilisé à la fois la distribution de Poisson et la distribution binomiale négative pour modéliser le nombre de buts marqués lors d'un match de football, en se basant sur les résultats passés de l'équipe. Cependant, n'est qu'en 1974 que Hill prouve que les résultats d'un match de football sont dans une certaine mesure prévisible et non pas simplement une question de chance. La première grande avancée provient de Maher en 1982, qui a utilisé les distributions de Poisson pour modéliser les capacités offensives et défensives des équipes à domicile et à l'extérieur et pour prédire le nombre moyen de buts pour chaque équipe. Dixon et Coles (1997) sont les pionniers et créent un modèle capable de produire des probabilités pour l'issue et les scores correspondant aux matchs, toujours selon une distribution de Poisson. Ce modèle est encore considéré comme un modèle traditionnel réussi que nous pouvons utiliser comme référence, par rapport aux autres modélisations que nous avons créées. Il est notamment basé sur un modèle de régression de Poisson, ce qui signifie qu'un nombre attendu de buts pour chaque équipe est transformé en probabilités de buts suivant la distribution de Poisson (**Figure 1**)

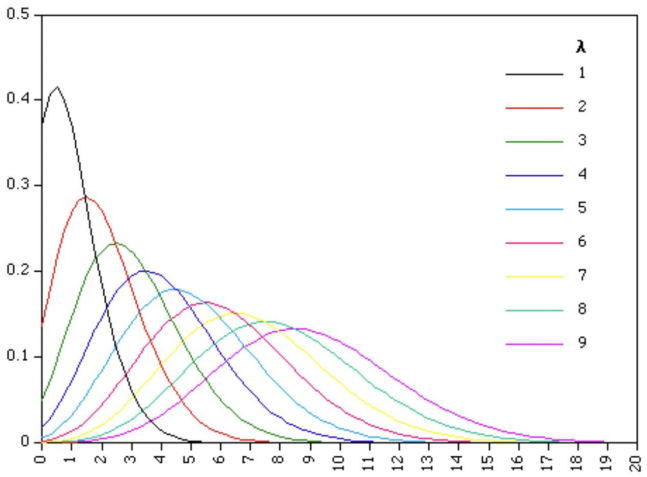


Figure 1 : Poisson distribution for values λ and k

$P(k\ buts\ dans\ le\ match) = e^{-\lambda}\frac{\lambda^k}{k!}$ ou λ représente le nombre de buts attendus dans le match. La distribution de Poisson permet donc de calculer la probabilité de marquer un certain nombre de buts pour l'équipe, qui peut donc être convertie en probabilités de gagner, perdre ou avoir un match nul.
Au début du 21ème siècle, les chercheurs ont commencé à modéliser l'issue du match, plutôt que le nombre de buts marqués. Par exemple, Forrest et Simmons (2000) ont utilisé un modèle de classification pour prédire directement le résultat du match au lieu de prédire les buts marqués à chaque fois. Cette technique leur a permis d'éviter le problème de l'interdépendance entre les scores des deux équipes. Kuypers à lui, utilisé des features tirées des résultats d'une saison de matchs pour générer un modèle capable de prédire les résultats de futurs matchs. Il a également été l'un des premiers à se pencher sur le marché des paris et à essayer de générer des stratégies de paris rentables en suivant le modèle qu'il a développé. D'autres études ont également été élaborées,

notamment celle de Palomino, Rigotti et Rustichini, 1999, qui ont mis en avant le rôle des émotions dans les performances. Ils ont estimé, sur la base de 2885 matches de football professionnels, la probabilité de marquer un but aux différents moments du match. Ils étudient comment cette probabilité est liée aux trois déterminants fondamentaux de la performance d'une équipe de football :

- **les aptitudes** (mesurées par des indicateurs tels que le nombre de buts marqués ou encaissés sur l'ensemble de la saison)
- **la stratégie** (définie comme le choix d'attaquer ou de défendre en réaction au score du match et en fonction du moment de la partie, et mesurée par la manière dont, pour une équipe, la probabilité de marquer dépend du score et du temps qui reste à jouer)
- **les émotions** (qui regroupent l'ensemble des facteurs émotionnels et psychologiques autour du match, et ramenées à l'avantage d'évoluer à domicile).

Les résultats montrent clairement que les trois facteurs interviennent simultanément et interagissent dans la détermination de la performance, soit la probabilité de marquer.

Maintenant, nous voulons examiner des modèles où des projets plus récents effectués sur le sujet, avec l'utilisation d'algorithmes modernes d'apprentissage automatique (Machine Learning) qui seront intéressants à étudier lorsque nous essaierons différents modèles prédictifs.
Goddard & Asimakopoulos (2004) utilisent un modèle probit ordonné. Le résultat du match entre les équipes peut prendre trois valeurs: 0 si l'équipe à l'extérieur gagne, $\frac{1}{2}$ s'il y a match nul, 1 si l'équipe à domicile gagne. Adam (2016) a utilisé un modèle linéaire généralisé simple, entraîné par descente de gradient, pour obtenir des prédictions de matchs et simuler le résultat d'un tournoi. Il a obtenu de bons résultats, même avec un ensemble limité de variables, et recommande d'ajouter d'autres fonctionnalités et d'utiliser un processus de sélection de variables, ce qui serait intéressant pour nous dans ce projet, compte tenu du nombre de fonctionnalités différentes qui sont à notre disposition. Pour faire face à ce problème de dimensionnalité, Tax et al. ont combiné des techniques de réduction de la dimensionnalité avec des algorithmes d'apprentissage automatique pour prédire une compétition de football néerlandaise.

Concernant les projets similaires, Enora Belz, Ewen Gallic, Romain Gatéa, Vincent Malardé, Jimmy Merlet, Arthur Charpentier ont réalisé, à l'occasion de la Coupe du Monde 2018, la prédiction des résultats à venir des rencontres. Ils ont utilisé des données sur les résultats de rencontres de Coupe du Monde, compétitions intercontinentales et coupes mondiales. Huit méthodes de Machine Learning en apprentissage supervisé ont été utilisées (les k plus proches voisins, la classification naïve bayésienne, les arbres de classification, les forêts aléatoires, le gradient boosting stochastique, la régression logistique par boosting, les machines à vecteurs de support et les réseaux de neurones artificiels). Concernant leurs résultats, des simulations de la compétition ont été réalisées. Ils ont trouvé que le Brésil avait 19,124% de chances de victoire (les premiers du classement), contre 14,52% pour l'Allemagne et 9,708% pour la France (qui se place en 4ème position), derrière le Brésil, l'Allemagne et l'Espagne.

4. Les modèles

4.1 Régression logistique multimodale

La régression logistique est utilisée lorsqu'on a un problème de classification binaire ou multiple. C'est un concept clé en Machine Learning, que nous allons utiliser pour la prédiction de nos matchs. C'est un modèle linéaire généralisé utilisant une fonction logistique comme fonction de liaison. L'objectif est de prédire la probabilité que l'événement se produise si la valeur prédite est supérieure à un seuil, ou ne se produise pas si elle est inférieur au même seuil, dans le cadre binaire, en se basant sur l'optimisation des coefficients de régression. L'objectif de la régression logistique est de trouver une fonction h = $\theta_0 + \theta_1 x$ telle que l'on puisse calculer :

$y = 1$ si $\theta_0 + \theta_1 x \geq 0.5$
$y = 0$ si $\theta_0 + \theta_1 x < 0.5$

La fonction h doit être une probabilité comprise entre 0 et 1 et dans la majorité des cas, le seuil est fixé à 0,5. La fonction sigmoid est utilisée et égale à $\sigma(x) = \frac{1}{1+e^{-x}}$.

On peut réécrire h : $h_\theta(x) := g(\theta_0 + \theta_1 x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$

Dans un cadre multimodal (gagner/perdre/nul) nous allons découper le problème de classification multi-classes en une multitude de problèmes de classification binaires, grâce à un algorithme **Un VS Tous** (One VS All)**.** Cet algorithme permets de prédire plusieurs régression logistiques sur des configurations possibles binaires soit $H_1(x), H_2(x), H_3(x)$ et chacunes de ces fonctions de prédiction nous donnera la probabilité que x soit de la classe y_i . La bonne classe de l'observation x est celle pour laquelle on a obtenu la plus grande probabilité soit $maxH^i(x)$ ou i=1,2,3

Notre problème de classification apparaît alors comme un simple problème d'optimisation, ou à partir des données de nos matchs, on essaie d'obtenir le meilleur ensemble de paramètres Θ permettant à notre courbe sigmoïde de coller au mieux aux données. Dans les régressions logistiques ainsi que les autres modèles, nous utiliserons la matrice de confusion, l'AUC (Area under the ROC curve) ou la validation croisée pour évaluer les performances de nos modèles.

4.2. Régression logistique avec un terme de pénalisation : Lasso

La régression logistique est sujette aux sur-ajustements du modèle aux données. En effet, le modèle peut ne pas être applicable à de nouvelles données (nouveaux matchs) à cause du sur-ajustement. Cela va se traduire par une variance élevée du modèle, car il accorde beaucoup d'attention aux données d'apprentissage et ne généralisent pas sur les données qu'il n'a pas vu auparavant. A l'inverse, lorsqu'il y a du biais, le modèle est trop simpliste et ne retranscrit pas les spécificités des données. Nous sommes également contraint par la dimension de notre database qui à un grand nombre de variables explicatives (52 features). On va alors utiliser des techniques de régularisation

(en pénalisant nos paramètres), pour éviter le sur-ajustement, qui est **LASSO** (*Least Absolute Shrinkage and Selection Operator*)

Lasso répond à ces deux contraintes qui sont que :

- Les paramètres Θ ne sont pas contraints, ils peuvent prendre de très grandes valeurs et avoir une grande variance
- Pour contrôler la variance, il faut contrôler la taille des paramètres Θ. Cette approche pourrait réduire alors les erreurs de prédictions.

Cette pénalisation, permettant d'éliminer de façon automatique les variables considérées non pertinentes, est particulièrement adaptée aux problèmes où le nombre de variables explicatives est élevé ou en cas de colinéarité. Un modèle de régression logistique classique peut suffire à éliminer toute confusion seulement s'il est correctement spécifié. Néanmoins, dans certaines situations, le poids des facteurs de confusion est tellement important, qu'un simple ajustement ne suffit pas pour garantir une interprétation simple des résultats.

D'un point de vue mathématiques, nous avons simplifier les formules et n'avons pas décrit la régression de **RIDGE** (une autre technique de pénalisation) mais si on se rapporte à une régression ou on doit minimiser la somme des carrés des résidus, soit RSS on a :

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_p x_{ip}\right)^2.$$

Les coefficients du modèle de régression sont obtenus en minimisant le critère suivant :

$$\frac{\mathbf{RSS}(\boldsymbol{\beta})}{2} + \lambda \sum_{j=1}^{p} \left|\beta_j\right|$$ ou on remarque un terme de pénalité qui est $\lambda \sum_{j=1}^{p} |\beta_j|$.

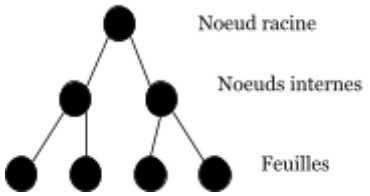
La présence de ce terme de pénalité nous permet de sélectionner des variables pour notre modèle car certains coefficients sont mis à zéro. La pénalité effectue une sélection des variables continues.

4.3. Machine à vecteurs de support (Support vector machine)

Machine à vecteurs de support (Support vector machine) a pour but de trouver le meilleur hyperplan, également appelé frontière de décision, séparant les classes. La marge ici correspond à la distance entre l'hyperplan et le point de données le plus proche. L'algorithme choisira un hyperplan avec la plus grande marge possible entre l'hyperplan et n'importe quel point du jeu de données d'entraînement. La maximisation augmente les chances qu'une observation soit bien classée. Il y a plusieurs façons de calculer la distance et nous avons opté pour celle qui maximise nos prévisions.

4.4. Forêt aléatoire

Un moyen d'améliorer les performances d'un modèle est l'utilisation d'algorithmes d'ensemble réalisant une prédiction plus précise. Ainsi, pour améliorer les arbres de décision classiques nous utilisons des forêts aléatoires. Un arbre de décision est une technique de modélisation non-paramétrique d'apprentissage supervisé. Un arbre est composé de 3 parties : les nœuds internes, les feuilles (à la fin de l'arbre), et le nœud racine (au début). À chaque nœud, l'algorithme doit faire le choix optimal pour classer les individus qui minimisent ce que l'on appelle l'impureté (mesure la fréquence à laquelle un élément choisi au hasard est mal étiqueté). Les mesures les plus utilisées dans les problèmes de classification sont l'entropie et le Gini impurity index. La variable avec la plus forte diminution de cet indice est choisie pour le nœud interne. Voici un schéma représentant la structure d'un arbre de décision:



Une forêt aléatoire crée un ensemble d'arbres de décision à partir d'un sous-ensemble d'observations sélectionnées au hasard dans l'échantillon. C'est une méthode supervisée utilisant des méthodes de bagging faisant la moyenne des prévisions de plusieurs modèles indépendants pour réduire la variance et donc l'erreur.
Le prétraitement des données requis est faible et la classification des nouvelles données est rapide. La forêt aléatoire peut traiter à la fois des caractéristiques catégorielles et continues, ainsi que des problèmes de classification multi-classes (ce qui est le cas ici).

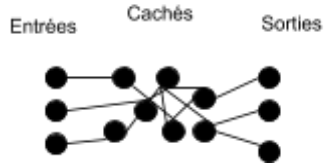
4.5. K-Voisins les plus proches.

La méthode de K-voisins les plus proches est une technique non paramétrique qui classifie les individus en fonction de leurs voisins. Le modèle mémorise les observations de l'ensemble d'apprentissage pour la classification des données de l'ensemble de test. Par exemple, si une nouvelle donnée est entourée de données dont la classe correspond à 'Victoire de l'équipe extérieure' alors l'algorithme conclura que la nouvelle donnée appartient également à la classe 'Victoire de l'équipe extérieure'. Il faut pour cela choisir les K voisins les plus proches à examiner. Ensuite, l'algorithme calcule la distance entre le nouveau point de donnée et les K autres. Il existe plusieurs mesures de distance qu'il peut utiliser (Euclidienne distance, Manhattan distance, Canberra distance). Puis il conserve toutes ces distances et les trie (par ordre croissant) afin de prendre les K premiers éléments, les K voisins qui sont les plus proches de nouvelle donnée.

Le choix du nombre de K est arbitraire et dépend du sujet. Il est possible de déterminer cette valeur en utilisant le taux d'erreur de prévision en fonction de la valeur de K. Celle qui minimise la valeur du taux d'erreur pourra être considérée comme le K optimal.

4.6. Réseau de neurones artificiels (ANN)

Notre cerveau a des milliards de neurones. Tous ces neurones sont connectés les uns avec les autres et échangent grâce à des impulsions électriques : les synapses. Des ingénieurs se sont inspirés du cerveau humain pour créer ce qu'on appelle des réseaux de neurones artificiels. Nous n'allons pas rentrer dans les détails techniques de cette méthode de Machine Learning, nous voulons l'expliquer le plus simplement possible. Pour que ce ANN puisse être utilisé, on déclenche un noeud avec une donnée d'entrée et ce noeud va déclencher d'autres noeuds auxquels il est connecté. On organise alors notre ANN en couche (d'entrées et de sorties). On définit également



des liens directs avec les noeuds, pour savoir comment se propage l'information. On assigne également différents poids à nos connexions, pour que certaines soient plus fortes que d'autres. Entre la couche d'entrées et de sorties, se trouvent une couche de neurones cachés (hidden layer). Voici un exemple plus complexe de Réseau de Neurones Artificielles **(Figure 2**).

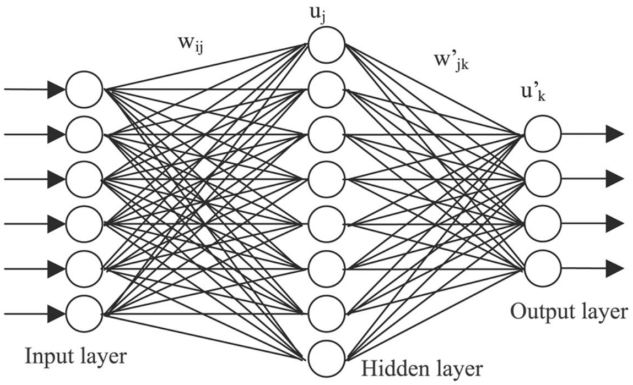
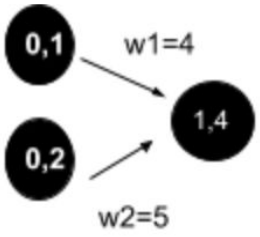


Figure 2 : Réseau de neurones artificiels

Pour utiliser un réseau de neurones, il faut une phase d'apprentissage. Nous sommes dans un cas de classification supervisé, nous connaissons l'issue de nos matchs. Pour que notre ANN soit efficace, il faut donc l'entraîner sur notre ensemble de données classifiées. Mais nous devons normaliser nos données (features), ce qui permet de traduire nos variables par des chiffres, afin qu'un réseau neural puisse les comprendre. En ce qui concerne l'output layer (la sortie), on remarque 4 valeurs possibles dans **la Figure 2.** Ce sont des probabilités de probabilités. Dans notre cas, on aura 3 possibilités en sortie (gagné/perdu/nul). Voici le fonctionnement global d'un ANN. Maintenant, concentrons nous sur les éléments qui composent l'ANN. Si nous nous intéressons à un seul neurone (cercle blanc), appelé Perceptron on remarque qu'il doit retenir un chiffre pour pouvoir se mutliplier aux autres. .

Le Perceptron a également un "faux neurone", une constante qui est importante pour l'apprentissage. *(1,4+2=3,4)*
On utilise une fonction d'activation pour avoir un seuil sur le résultat



que nous venons de calculer. Pour notre modèle nous utilisons la fonction ***"Sotfmax"***, mais il est également possible d'utiliser la fonction ***"Sigmoid"***. Cette fonction établit un seuil d'une valeur ou le neurone est stimulé ou pas. Cette fonction d'activation stimule le potentiel d'action d'un neurone, qui se propagera que s' il est au-dessus du seuil. La dernière étape concerne l'amélioration du modèle. Nous avons besoin d'un mécanisme qui est capable de corriger les erreurs de prédictions du modèle : qui apprend de ces erreurs. On utilise alors la rétropropagation du gradient qui démarre de la couche sorties à celle d'entrées. Dans notre ANN, nous utilisons **"Adam".** Son objectif est de modifier le poids de l'ensemble des connexions entre les neurones. Pour l'ANN, durant la phase d'apprentissage, on va vérifier si l'issue du match est correct. Si elle ne l'est pas, alors on va effectuer un calcul pour modifier le poids des connexions et la valeur des neurones précédents. Ainsi, le neurone qui est le plus responsable de l'erreur sera susceptible d'être modifié davantage. A mesure qu'on fait l'apprentissage, le nombre d'erreurs diminue, on peut alors utiliser notre ANN pour réaliser des prédictions.

5. Les données

5.1. Présentation des données

Les données utilisées dans ce rapport proviennent du site internet Kaggle[4]. Elles sont divisées en plusieurs tables. La première et la plus importante regroupe plus de 25.000 matchs nationaux entre 2008 et 2016 de 11 pays européens différents. Des détails des matchs sont présents comme les types de buts, les possessions, les corners, les fautes etc... Les ligues sont les ligues principales de chaque pays. Nous avons également plus de 10.000 joueurs et leurs attributs (comme leurs notes globales, leurs forces de frappe, leur précision de tir) ainsi que ceux de chaque équipe. Ces attributs proviennent du site FIFA qui les utilise pour calibrer ses jeux vidéos (se rapprochant donc au maximum des qualifications des joueurs).

Nous disposions uniquement des buts marqués par chaque équipe. Afin de pouvoir prédire si un match sera gagné ou non, il a été nécessaire de créer une variable (qui sera notre variable dépendante) prenant 3 valeurs possible :

Winner	Description
2	L'équipe à domicile GAGNE
1	L'équipe extérieure GAGNE
0	Match nul

[4] https://www.kaggle.com/hugomathien/soccer

Les différents variables de notre modèle sont les suivantes :

Nom de la variable	Description
Winner	Issue de la rencontre. Prend 3 valeurs (2 : l'équipe à domicile l'emporte, 1: l'équipe extérieure l'emporte, 0 : match nul).
season	Saison du match
home_team	Equipe à domicile représentée par son classement FIFA
away_team	Equipe extérieure représentée par son classement FIFA
league	La ligue à laquelle appartient le match
mean_birth_home	Age moyen des joueurs de l'équipe qui reçoit (en kg)
mean_birth_away	Age moyen des joueurs de l'équipe extérieure (en kg)
mean_height_home	Taille moyenne des joueurs de l'équipe qui reçoit (en cm)
mean_height_away	Taille moyenne des joueurs de l'équipe extérieure (en cm)
mean_weight_home	Moyenne de poids des joueurs de l'équipe à domicile
mean_weight_away	Moyenne de poids des joueurs de l'équipe à l'extérieur
mean_overall_rating_home	Moyenne de jeu des 3 meilleurs joueurs de l'équipe à domicile
overall_rating_away_x	Moyenne de jeu des 3 meilleurs joueurs de l'équipe à l'extérieur
penalties_away	Moyenne des penaltys des 3 meilleurs joueurs de l'équipe à l'extérieur
interceptions_away	Moyenne des interceptions des 3 meilleurs joueurs de l'équipe à l'extérieur
shot_power_away	Moyenne de la puissance de tirs des 3 meilleurs joueurs de l'équipe à l'extérieur
reactions_away	Moyenne du temps de réactions des 3 meilleurs joueurs de l'équipe à l'extérieur
dribbling_home	Moyenne des dribble des 3 meilleurs joueurs de l'équipe à domicile
free_kick_accuracy_home	Précision du coup franc moyenne des trois meilleurs joueurs de l'équipe à domicile
long_passing_home	Précision moyenne des passes longues des trois meilleurs joueurs de l'équipe à domicile
ball_control_home	Moyenne du taux de contrôle des 3 meilleurs joueurs de l'équipe à domicile
acceleration_home	Moyenne des accélérations des 3 meilleurs joueurs de l'équipe à domicile
reactions_home	Moyenne du temps de réactions des 3 meilleurs joueurs de l'équipe à domicile
shot_power_home	Puissance de tir moyenne des trois meilleurs joueurs de l'équipe à domicile
interceptions_home	Capacité d'intercepter les balles moyenne des trois meilleurs joueurs de l'équipe à domicile
penalties_home	Précision moyenne des penalty des trois meilleurs joueurs de l'équipe à domicile
overall_rating_away	Moyenne de la note globale des 3 meilleurs joueurs de l'équipe extérieure
heading_accuracy_away	Précision du jeu de tête moyenne des trois meilleurs joueurs de l'équipe extérieure
short_passing_away	Moyenne des passes courtes des 3 meilleurs joueurs de l'équipe à l'extérieur
dribbling_away	Moyenne des dribble des 3 meilleurs joueurs de l'équipe à l'extérieur

dibbling_away	Moyenne des dribble des 3 meilleurs joueurs de l'équipe à l'extérieur
free_kick_accuracy_away	Moyenne des coups francs des 3 meilleurs joueurs de l'équipe à l'extérieur
long_passing_away	Moyenne des longues passes des 3 meilleurs joueurs de l'équipe à l'extérieur
ball_control_away	Moyenne du taux de contrôle des 3 meilleurs joueurs de l'équipe à l'extérieur
acceleration_away	Moyenne des accélérations des 3 meilleurs joueurs de l'équipe à l'extérieur
buildUpPlaySpeed	Moyenne de la rapidité de jeu de l'équipe à domicile
buildUpPlayPassing	Moyenne de la rapidité des passes de l'équipe à domicile
chanceCreationPassing	Moyenne des chances de création de passe de l'équipe à domicile
chanceCreationCrossing	Moyenne de création de jeu de l'équipe à domicile
chanceCreationShooting	Moyenne des chances de création de tirs de l'équipe à domicile
defencePressure	Moyenne de la pression de la défense de l'équipe à domicile
defenceAggression	Moyenne de l'agression de la défense de l'équipe à domicile
defenceTeamWidth	Largeur de la défense de l'équipe à domicile
bUpPlaySpeed_away	Moyenne de la rapidité de jeu de l'équipe à l'extérieur
bUpPlayPassing_away	Moyenne de la rapidité des passes de l'équipe à l'extérieur
cCreationPassing_away	Moyenne des chances de création de passe de l'équipe à l'extérieur
cCreationCrossing_away	Moyenne de création de jeu de l'équipe à l'extérieur
cCreationShooting_away	Moyenne des chances de création de tirs de l'équipe à l'extérieur
defencePressure_away	Moyenne de la pression de la défense de l'équipe à l'extérieur
defenceAggression_away	Moyenne de l'agression de la défense de l'équipe à l'extérieur
defenceTeamWidth_away	Largeur de la défense de l'équipe à l'extérieur
overall_rating_home	Moyenne de jeu des 3 meilleurs joueurs de l'équipe à domicile
heading_accuracy_home	Moyenne des précisions des têtes des 3 meilleurs joueurs de l'équipe à domicile
short_passing_home	Moyenne des passes courtes des 3 meilleurs joueurs de l'équipe à domicile
free_kick_accuracy_home	Moyenne des coups francs des 3 meilleurs joueurs de l'équipe à domicile
long_passing_home	Moyenne des longues passes des 3 meilleurs joueurs de l'équipe à domicile
reactions_home	Moyenne du temps de réactions des 3 meilleurs joueurs de l'équipe à domicile
shot_power_home	Moyenne de la puissance de tirs des 3 meilleurs joueurs de l'équipe à domicile
interceptions_home	Moyenne des interceptions des 3 meilleurs joueurs de l'équipe à domicile
penalties_home	Moyenne des penaltys des 3 meilleurs joueurs de l'équipe à domicile
overall_rating_away_y	Moyenne de jeu des 3 meilleurs joueurs de l'équipe à l'extérieur

Comme nous l'avons précisé précédemment, les buts de football sont distribués selon une loi de Poisson. Cela signifie que se sont des processus sans mémoire où les événements passés n'ont pas d'impact sur les événements futurs. Contrairement à la croyance selon laquelle le moment le plus opportun pour marquer un but se situe juste après en avoir marqué un, est contredit par ceci. C'est pourquoi la meilleure équipe ne gagnera pas toujours. Nous pourrons donc considérer que les données dont nous disposons sont des données en coupe transversale.

5.2. Prétraitement des données

Le prétraitement des données a été une étape nécessaire. Dans un premier temps nous avons supprimé toutes les valeurs manquantes et les variables qualitatives intraitables. Ensuite notre base de données était composé en réalité de 7 tables différentes :

- **country** : le pays dans auquel appartient l'équipe
- **league** : la ligue à laquelle appartient l'équipe
- **team** : l'équipe
- **team attributes** : les statistiques générales de l'équipe telles que le taux de défense, de penalty ou encore l'agilité
- **player** : la liste de tous les joueurs, avec leurs âges, leurs poids et leurs tailles.
- **player attributs** : la liste de tous les attributs des joueurs, avec leurs taux d'accélération, de penalty ou encore d'interceptions.
- **match :** la liste de tous les matchs avec les 22 joueurs, les saisons et dates de match

Nous avons décidé de prédire l'issue d'un match. Nous devons donc regrouper toutes ces tables pour avoir une seule base, qui retransmet le plus d'informations possible de ces 7 tables et dont une ligne correspond à un match. De plus, nous avons décidé de nous concentrer que sur la Ligue 1 pour les modèles et donc les matchs et joueurs correspondants, ce qui a réduit notre base de données à 2,194 observations.

- **Pour la table player attributes :**

Sachant qu'un match est joué avec 22 joueurs, utiliser une variable dite "technique" pour chaque joueur (défense, attaque, taux d'interception de la balle) aurait été trop compliqué. Ainsi, pour pouvoir utiliser les attributs des joueurs sans créer un nombre trop important de variables (le nombre d'attributs multiplié par le nombre total de joueurs sur le terrain) nous avons calculé pour chaque match la moyenne pour chaque attribut des trois meilleurs joueurs de l'équipe.

- **Pour la table player :**

Concernant la taille, l'âge et le poids des joueurs, nous avons calculé une moyenne de tous les joueurs de l'équipe sur chaque match. Nous avons pu le réaliser en collant la table match à la table player (pour chaque joueur du match). Sachant qu'un joueur peut changer de club à chaque saison, la difficulté était de retrouver, pour chaque saison, les joueurs qui ont joué pour cette équipe.

- **Pour la table team attributes :**

Nous avons gardé toutes les variables qui décrivent la puissance en attaque, en défense, en rapidité et en agilité des équipes.

Une autre étape a été de désigner chaque équipe par son classement en Ligue 1 plutôt que de créer des variables binaires pour chaque équipe afin de connaître l'identité des équipes qui s'affrontent. Nous avons également contrôlé le nombre de matchs gagnés au total sur toutes les saisons par l'équipe, en plus de son classement en Ligue 1 depuis les saisons 2008, jusqu'à 2016. Ainsi, le Paris Saint Germain est recodé en 1, l'Olympique lyonnais en 2 et l'Olympique de Marseille en 3, par exemple.

Au final, après cette étape de pré-traitement des données, notre base de données est composée de 2194 matchs et de 53 variables. Notre base de données étant maintenant exploitable, il est important de détecter si nos features sont liées. Dans certains cas c'est évident (comme par exemple le lien de dépendance dans une hiérarchie) mais bien souvent ces liens ou corrélations sont presque invisibles. Il va falloir détecter et mesurer ces liens potentiels. Nous regardons donc la corrélation entre nos features (**Figure 3**). Nous décidons de supprimer les variables qui sont corrélées à plus de 80%. Nous supprimons donc 11 features (*reactions away, dribbling away, etc..)*

	Equipe à domicile	Equipe extérieure	Moyenne d'âge à domicile	Moyenne d'âge à l'extérieur
Equipe à domicile	1.0	-0.051	0.072	-0.032
Equipe extérieure	-0.051	1.0	-0.013	0.073
Moyenne d'âge à domicile	0.072	-0.013	1.0	0.59
Moyenne d'âge à l'extérieur	-0.032	0.073	0.59	1.0

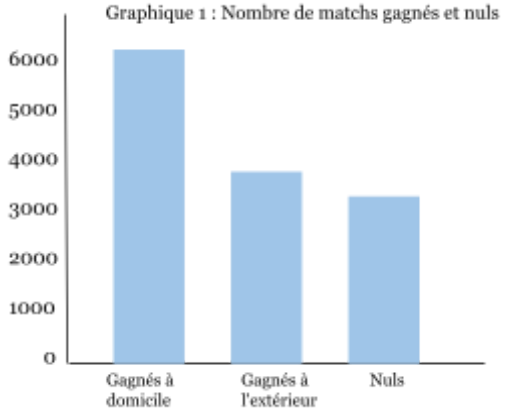
Figure 3 : Matrice de corrélation

La dernière étape de pré-traitement des données, consiste à rééquilibrer notre base de données. En effet, on peut constater 44% de matchs gagnés, contre 29% de matchs perdus et 27% de matchs nuls. Pour l'élaboration de nos modèles nous avons coupé notre base de données en échantillon d'apprentissage et de test (la partie d'apprentissage permets d'entraîner le modèle, pour s'approprier les paramètres au mieux et les ajuster; la partie test est utilisée pour évaluer le modèle et mesurer sa capacité prédictive). Nous allons donc réajuster la base de données que sur la partie d'apprentissage de notre modèle et laisser la partie test brut. Le rééchantillonnage de la base de données se fait par **la méthode SMOTE**. Nous n'allons pas rentrer dans les détails de cet algorithme mais son principe de base est de générer de nouveaux échantillons en combinant les données de la classe minoritaire avec celles de leurs voisins proches. Par conséquent, cette méthode génère synthétiquement de nouveaux matchs de la classe minoritaires en utilisant les matchs déjà existants. Après cette transformation, notre échantillon d'apprentissage contient 2286 matchs et 3 catégories équilibrées : 762 matchs gagnés, perdus et nuls. Notre partie test à 439 matchs dont 213 gagnés, 127 nuls et 99 perdus.

5.3. Statistiques descriptives basiques

Lors de l'étape d'exploration des données, nous avons pu affirmer le fait selon lequel une équipe qui joue à domicile à plus tendance à gagner que l'équipe adverse. Parmi tous les matchs enregistrés dans notre base de données, plus de 6200 matchs sont remportés à domicile tandis que moins de 4000 matchs ont été remportés à l'extérieur.

Graphique 1 : Nombre de matchs gagnés et nuls



Le fait d'être à domicile peut donc être déterminant pour l'issue du match. Nous verrons par la suite qu'en utilisant une méthode de sélection de variables, celles concernant les équipes et notamment les équipes à domicile seront sélectionnées.

En nous concentrant sur les matchs de l'Olympique de Marseille nous avons cherché à savoir quelle équipe était son adversaire le plus féroce. Bien que cela puisse en décevoir certains, il s'avère que l'Olympique de Marseille a le plus de fois perdu contre le Paris-Saint Germain. Aussi l'écart avec les autres équipes est assez grand.

Equipes	Nombre de matchs perdus par l'OM
PSG	10
AS Monaco	5
MHSC	5
OGC Nice	5
OL	5

Nous sommes donc aller regarder les performances de l'équipe parisienne et notamment les différents attributs d'équipe mais également ceux de ses 3 meilleurs joueurs (en termes de note globale). Nous avons donc analysé ces matchs et déterminé les points faibles de l'OM qui pourraient influencer l'issue de la rencontre face au PSG. Tout d'abord, le Paris Saint Germain a un score moyen d'accélération de 84, note élevée comparée aux autres équipes de la Ligue 1. Il est vrai qu'une bonne accélération au bon moment permet de faire la différence. L'élément le plus important qui doit faire la différence pour l'équipe parisienne est la force de frappe qui est bien plus élevée que celle de Marseille (83 comparé à 74 en moyenne). Le PSG a un peu plus de mal à mettre la pression en défense bien que leur moyenne de score d'interception soit elle aussi élevée par rapport aux autres équipes. Ce dans quoi le Paris Saint Germain domine c'est bien la création d'opportunités de tir. Par exemple, la moyenne de score des trois meilleurs joueurs est de 69 alors qu'elle est de 46 pour l'OM. Nous recommandons donc à l'entraîneur de l'équipe marseillaise de travailler les interceptions, l'agressivité en défense, les accélérations et créer lors des matchs plus d'opportunités de tir.

6. Les résultats et prédictions

6.1. Les indicateurs de performance utilisés

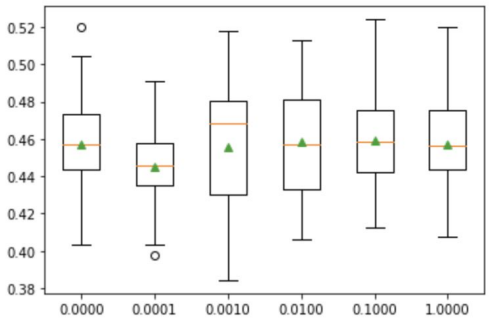
Pour prédire la performance des modèles nous utilisons deux principaux indicateurs : la performance de la partie test de notre base de données (celle qui n'a pas été modifiée par la technique SMOTE, de sur échantillonnage) et nous utilisons également l'air sous la courbe ROC.

- **performance de la partie test** : on estime la précision du modèle et ses propriétés telles que les erreurs de classification. On obtient alors un coefficient qui est la performance du modèle sur la partie test
- **l'air sous la courbe ROC** : pour un ensemble de données déséquilibrée, on utilise l'air sous la courbe ROC. Cet indicateur est une mesure unique de la performance d'un classificateur pour l'évaluation du modèle qui est meilleur en moyenne (Lopez et al. (2013)). Cette mesure montre à quel point le modèle distingue bien deux, ou trois classes.

6.2. Régression Logistique et terme de pénalisation

En appliquant la méthode Lasso pour sélectionner des variables, on découvre les résultats suivants. Nous avons appliqué des points de pénalités λ équivalent à [0.0, 0.0001, 0.001, 0.01, 0.1, 1.0]. On fait ensuite tourner nos régressions logistiques multimodales sur chacunes de ces valeurs λ. Pour tester la performance du meilleur modèle on utilise la technique de **validation croisée en K parties**. Nous n'allons pas décrire l'algorithme de cette méthode, qui permet d'évaluer la capacité de généralisation du modèle, nous allons juste résumer son principe. On prend toute la base de données, qu'on divise en k parties égales sur lesquelles on entraîne et teste un modèle pendant k itérations. A chaque itération, le modèle est entraîné sur k-1 parties et est testé sur la partie

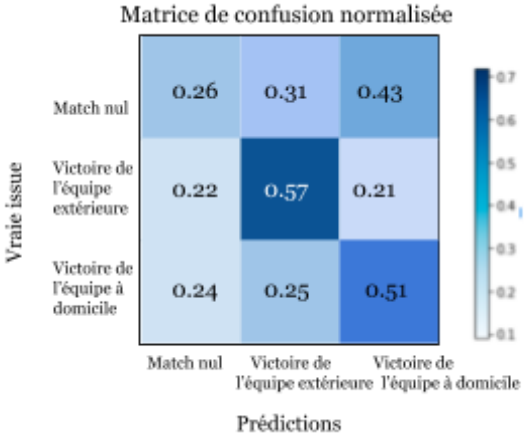
restante. Au final pour chacune des régressions logistiques prédites, on calcule un score qui évalue le modèle.



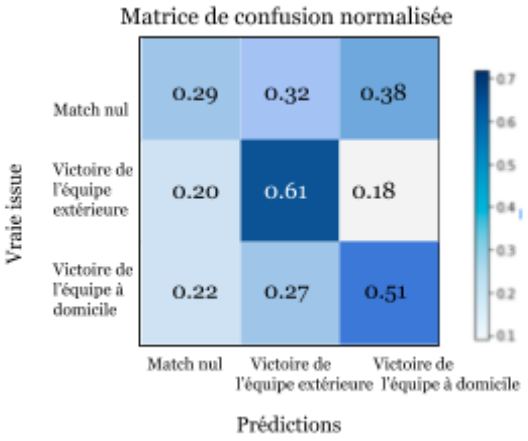
Score :

```
0.0000 0.457 (0.027)
0.0001 0.445 (0.023)
0.0010 0.456 (0.033)
0.0100 0.459 (0.031)
0.1000 0.459 (0.028)
1.0000 0.457 (0.028)
```

On remarque alors, que le meilleure modèle est celui avec une pénalité égale à 0,01 **(0,459)**. La pénalité est un hyper paramètre important à régler pour la régression logistique multinomiale car il impose une pression sur le modèle. Toutefois, notre régression a également un bon score sans pénalité **(0,457)**. Lorsqu'on élabore notre régression logistique multimodale avec un terme de pénalité égal à 0,01 on trouve les résultats suivants : La précision de notre pédiction est égale à 0,46 pour la partie d'apprentissage des données et égale à 0,42 pour la partie test. Nous arrivons presque à prédire correctement 1 match sur 2.



D'après la matrice de confusion, nous avons correctement prédit 51% des matchs gagnés, 26% des matchs nuls et 57% des matchs perdus (la diagonale correspond aux bonnes prédictions). Or, il faut faire attention aux matchs prédits gagnés, qui ont en réalité été perdus (21% ici) et aux matchs prédits perdus qui en réalité ont été gagnés (25% ici). La prédiction des matchs nuls est ici mitigée. Nous pouvons constater des résultats moyens, sauf pour la prédiction des matchs perdus. Nous avons également modélisé la régression logistique sans pénalité, on découvre les résultats suivants: la précision de notre prédiction est égale à 0,473 pour la partie d'apprentissage des données et égale à 0.415 pour la partie test. La matrice de confusion montre de meilleurs résultats que les prédictions avec pénalités.



Nous avons réussi à diminuer les faux matchs gagnés (18% contre 21% auparavant) mais nous n'avons pas diminué les faux matchs perdus (27%, contre 25% auparavant). De plus les prédictions sur la diagonale sont meilleures sauf pour les matchs gagnés ou on trouve le même taux. La prédiction des matchs nuls est nettement meilleure dans la régression logistique sans pénalité.

Concernant les features sélectionnées grâce à la méthode LASSO on a :

[*"home_team', 'away_team', 'mean_birth_home', 'mean_birth_away', 'interceptions_away', 'shot_power_away', 'heading_accuracy_away','buildUpPlayPassing', 'chanceCreationCrossing', 'bUpPlayPassing_away',' cCreationPassing_away', 'defenceAggression_away','defenceTeamWidth_away', 'overall_rating_home', 'heading_accuracy_home','dribbling_home', 'acceleration_home', 'shot_power_home','overall_rating_away_y*]

Ces variables traduisent pour la plupart les capacités sportives des trois meilleurs joueurs des équipes telles que la rapidité des tirs, la qualité des dribbles, la capacité d'accélération ou encore de création de passes. On a également les équipes, la moyenne d'âge des joueurs de l'équipe et parfois des moyennes de l'équipe comme sa défense moyenne. Maintenant si on fait la distinction équipe jouant à domicile, équipe jouant à l'extérieur, on se rend compte que :

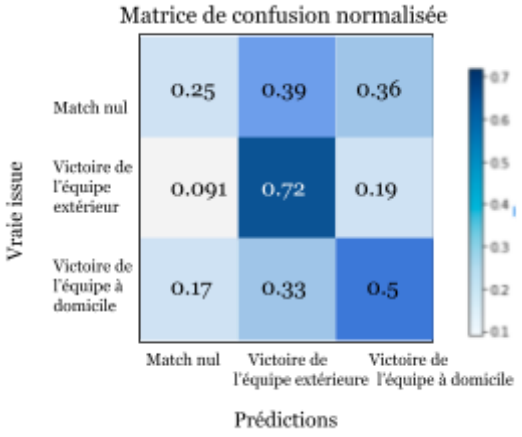
- **Pour l'équipe jouant à domicile,** les variables les plus importantes pour nos modèles sont l'accélération, la puissance de tir, l'occasion de croisement ou de tirs, la construction des passes de jeu
- **Pour l'équipe jouant à l'extérieur**, les variables sont la longueur de la défense, le taux d'agressivité de la défense ou encore la puissance de tir.

Cela rejoint bien notre théorie, selon laquelle jouer à domicile procure un regain d'espérance et un surplus de motivation pour favoriser les victoires. Les joueurs connaissent l'environnement (stade, vestiaire, etc.) et ont tous leurs repères. En effet, on retrouve des variables qui traduisent l'attaque pour l'équipe à domicile et des variables de défense pour l'équipe jouant à l'extérieur.

6.3. Machine à vecteurs de support

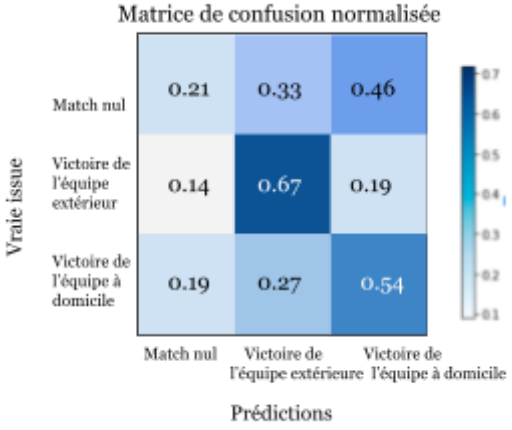
En utilisant cette méthode d'apprentissage supervisée, nous avons adapté les paramètres de sorte qu'ils maximisent l'air sous la courbe ROC. Grâce à cette technique nous obtenons une aire de 0,61 ce qui est pour l'instant notre meilleur résultat. Pour les matchs remportés par l'équipe extérieure,

nous arrivons à prédire 72% des matchs effectivement remportés par ces équipes. Quant aux matchs remportés par l'équipe à domicile, ils sont bien prédits une fois sur 2. L'algorithme a cependant quelques difficultés à prédire les matchs nuls.
Comme nous pouvons le voir dans la matrice de confusion ci-dessous, pour les matchs nuls, cette méthode utilisant les distances à l'hyperplan permet de plutôt bien classer et prédire l'issue d'une rencontre de football.



6.4. Forêts aléatoires

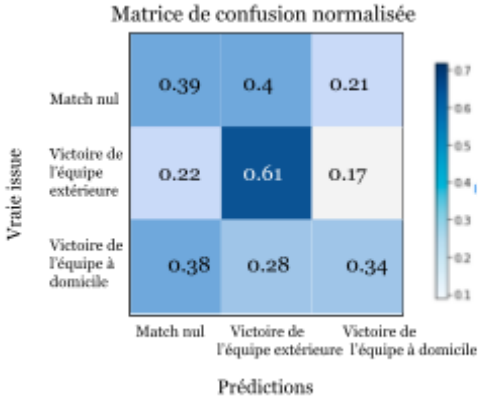
En ce qui concerne les résultats de la forêt aléatoire. Nous l'avons paramétrée de façon à ce que nous ayons les meilleurs résultats possibles. La forêt se compose de 500 arbres. L'air sous la courbe ROC est de 0,60.
La prédiction des matchs remportés par l'équipe à domicile est plutôt bonne puisque 67% des matchs gagnés par l'équipe extérieure ont été prédit comme remportés par l'équipe extérieure. Pour les matchs remportés par l'équipe à domicile ils sont un peu plus de 54% des fois bien prédits comme étant remportés par l'équipe à domicile. Pour ces matchs-là, il y a une amélioration par rapport à la méthode précédente, mais pour ce qui est des matchs gagnés par l'équipe adverse (ou perdus par l'équipe à domicile) le pourcentage de bonnes prédictions passe de 72 à 67%.



Comme nous pouvons le voir dans la matrice ci présente, la forêt aléatoire n'a pas vraiment réussi à prédire les autres catégories et se trompe 46% des fois lorsqu'elle rencontre les données d'un match nul mais le considère comme remportée par l'équipe à domicile.

6.5. K-Voisins les plus proches

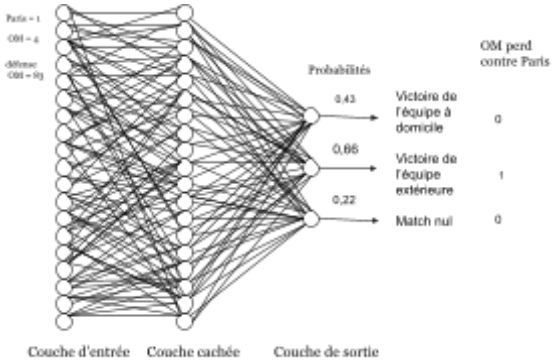
Avec l'algorithme des K-voisins les plus proches, et en testant ici aussi les paramètres, le K optimal obtenu est 20. Pour attribuer une classe (une issue de match) a une nouvelle observation, l'algorithme regardera donc les 20 voisins les plus proches de cette observations
Dans le cas de l'algorithme des K-voisins les plus proches, la valeur optimale de la courbe ROC est de 0.57. Nous pouvons voir également dans la matrice de confusion ci-dessus que les proportions de bonnes prédictions et celles des erreurs sont assez proches les unes des autres.



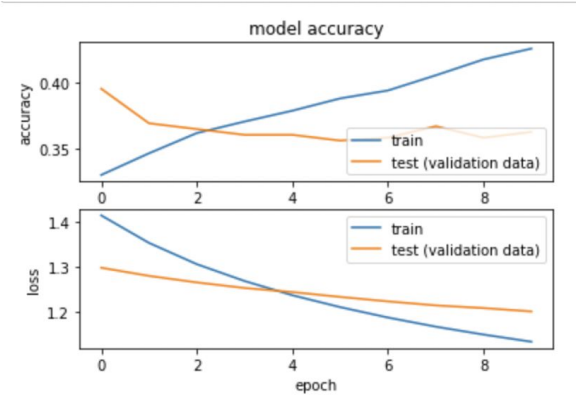
L'algorithme arrive cependant à prédire plus de 60% des matchs gagnés par l'équipe extérieure (donc perdus par l'équipe à domicile) et du côté des matchs nuls, réussit mieux à les classer dans la bonne classe que les autres méthodes de classification vues précédemment. En ce qui concerne les prédictions des matchs gagnés, l'algorithme est plus indécis, puisque ces observations sont réparties de façon presque égale entre les trois issues possibles. Si le but pour le club est de prédire au mieux possible une victoire alors cette méthode n'est pas la mieux adaptée pour une telle analyse.

6.6. Réseau de neurones artificiels

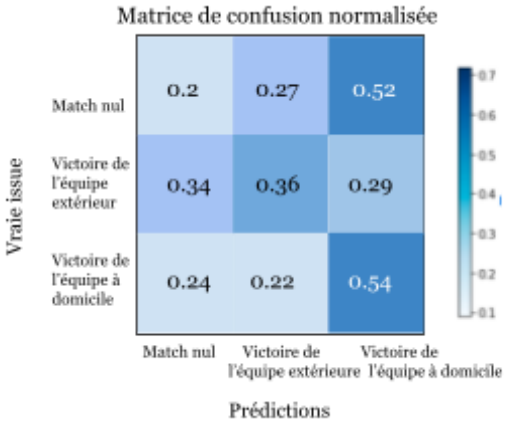
Concernant notre réseau de neurones artificiels, voici la structure de notre modèle :



Nous avons pour cette méthode coupé notre base de données en une partie d'apprentissage (train) et une partie test. Voici les graphiques de leurs performances :



On remarque que le partie test et la partie train se coupe à partir du troisième epoch. Finalement, la performance de la partie d'apprentissage est égale à 0,49 et celle de la partie test à 0,48, en réglant quelques paramètres de notre modèle. Lorsque que nous évaluons le réseau de neurones sur les prédictions réalisées, nous trouvons une performance de prédictions de 0,40 et une aire sous la courbe de 0,57. D'après la matrice de confusion, nous avons réussi à prédire 114 (54%) matchs gagnés, 36 matchs nuls (36%) et 26 matchs perdus (20%) .



Nous n'avons pas réussi à diminuer le taux des faux matchs gagnés (29% contre 17% pour la méthode KNN) mais pour celui des faux matchs perdus (22% contre 30 et 27% pour les autres méthodes), nous avons amélioré nos résultats. Les résultats du réseau de neurones artificiels ne sont pas les meilleurs, malgré le fait que ce soit un modèle très puissant en Machine Learning. Nous avons essayé un modèle assez simple, qui peut-être n'arrivait pas à capter la complexité de nos variables.

6.7. Prédiction des prochains matchs :

Grâce à ces modèles, nous avons essayé de prédire des matchs, dont on ne connaissait pas l'issue. Nous passons donc à un problème d'apprentissage non supervisé. Nous avons pris le meilleur modèle : **la Machine à vecteur de support (SVM),** qui arrive à prédire **50% des victoires, 72% des défaites et 25% des matchs nuls**. Nous avons aussi ajouté à notre partie test, 3 matchs qui vont prochainement être joués. Nous avons réussi à actualiser nos données pour ces équipes grâce au site FIFA, qui a toutes les statistiques des joueurs. Cette procédure nous a demandé beaucoup de temps,

car comme nous l'avons expliqué dans la section sur la présentation des données, chaque variable de jeu décrit soit la moyenne des 3 meilleurs joueurs, ou de tous les joueurs de l'équipe.

Match 1 : Olympique de Marseille - Lens ; Ligue 1 - Mer.20/01

Pour pouvoir réaliser cette prédiction, nous avons trouvé que les 3 meilleurs joueurs de chaque équipe sont :

- OM : Thauvin, Payet et Rongier
- Lens : Faufana, Médina et Badé

Pour ce match, nous prédisons **la victoire de l'Olympique de Marseille** au Stade Vélodrome.

Match 2 : Paris Saint Germain - Montpellier ; Ligue 1 - Ven.22/01

Les 3 meilleurs joueurs de chaque équipe sont :

- PSG: Mbappé, Neymar et DiMaria
- Montpellier : Delort, Laborde et Mollet

Pour ce match, nous prédisons **la victoire du Paris Saint Germain** au Parc des Princes.

Match 3 : Monaco - Olympique de Marseille ; Ligue 1 - Sam.23/01

Les 3 meilleurs joueurs de chaque équipe sont :

- Monaco : Yedder, Volland et Aguillar
- OM : Thauvin, Payet et Rongier

Pour ce match, nous prédisons **une défaite de l'Olympique de Marseille** au stade Louis-II.

7. Conclusion

Notre premier objectif a été d'analyser les matchs de l'Olympique de Marseille et de pouvoir trouver des recommandations pour améliorer les performances de jeu de l'équipe en nous appuyant sur la comparaison avec l'équipe du Paris Saint Germain. Il faut donc que lors des prochains entraînements de l'équipe, l'entraîneur de marseille veille à améliorer la capacité à accélérer, et aussi la puissance de tir. De plus il faudrait, pour améliorer le jeu de l'équipe, travailler les interception de balle et la précision des penalties.

Notre principal objectif, qui était de construire un modèle de prédictions, en explorant les différentes techniques d'apprentissage de Machine Learning, a été atteint. En effet, nous avons utilisé des algorithmes modernes d'apprentissage comme les réseaux de neurones, les forêts aléatoires et les machines à vecteurs de classification pour générer des prédictions sur l'issue des matchs de football

Nous avons réussi à trouver et à améliorer une base de données contenant suffisamment d'informations pour générer des modèles. Nous espérons que nos modèles et nos prédictions vous aideront dans votre quête de victoire de la Ligue des Champions.

8. Bibliographie

Adam, A. (2016). Generalised linear model for football matches prediction. KULeuven, 19.

Belz, E., Gallic, E., Gatéa, R., Malardé, V., Merlet, J., Charpentier, A. (2018). Coupe du monde 2018: Paul the octopus is back.

Dixon, M. J., & Coles, S. G. (1997). Modelling Association Football Scores and Inefficiencies in the Football Betting Market. Journal of the Royal Statistical Society: Series C (Applied Statistics), 46(2), 265‑280

Forrest, D., & Simmons, R. (2000). Forecasting sport: the behaviour and performance of football tipsters. International Journal of Forecasting, 16(3), 317‑331.

Goddard, J., & Asimakopoulos, I. (2004). Forecasting football results and the efficiency of fixed-odds betting. Journal of Forecasting, 23(1), 51‑66

Hill, I.D (1974).Association football and statistical inference. Applied Statistics, 23, 203- 208

Kuypers, T. (2000). Information and efficiency: An empirical study of a fixed odds betting market. Applied Economics

López, M.G. ,García-González, Franco-Robles, S.E. (2017).Carbohydrate analysis by NIRS-chemometrics, Intech, pp. 1-16, 10.5772/67208

Maher, M. J. (1982). Modelling association football scores. Statistica Neerlandica, 36(3), 109‑118

Moroney, M. (1956). Facts from Figures, 472 pp. Penguin: London.

Palomino, F. & Rigotti, L. & Rustichini, A. (1999). Skill, Strategy, and Passion: an Empirical Analysis of Soccer

Reep, C., Pollard, R., & Benjamin, B. (1971). Skill and Chance in Ball Games. Journal of the Royal Statistical Society. Series A (General), 134(4), 623.

Tax, N., & Joustra, Y. (2015). Predicting the Dutch football competition using public data: A machine learning approach. Transactions on Knowledge and Data Engineering, 10(10), 1‑13.