Modèles de prédiction d'événements rares en suivi longitudinal. Application au risque de blessure chez les sportifs professionnels

Mathieu Berthe

UNIVERSITÉ CLERMONT AUVERGNE

Doctorat

THÈSE

pour obtenir le grade de docteur délivré par

l'École doctorale des sciences fondamentales

Spécialité doctorale "Mathématiques appliquées et applications des mathématiques"

présentée et soutenue publiquement par

Mathieu BERTHE

le 25/02/2021

Modèles de prédiction d'événements rares en suivi longitudinal. Application au risque de blessure chez les sportifs professionnels

Directeur de thèse : **Pierre DRUILHET**
Co-encadrant de thèse : **Stéphanie LEGER**

Jury

Pierre Druilhet,	Professeur	Directeur
Stéphanie Léger,	Maître de conférences	Co-encadrante
Denys Pommeret,	Professeur	Rapporteur
Natacha Heutte,	Professeure	Rapporteure
Delphine Blanke,	Professeure	Rapporteure
Sergeï Dachian,	Professeur	Examinateur
Anne Françoise Yao,	Professeure	Examinatrice
Olivier Brachet,		Membre invité

THESE

Remerciements

Je tiens à adresser mes premiers remerciements à mes directeurs de thèses, Pierre (Druilhet) et Stéphanie. Ils m'ont encouragé, soutenu et conseillé tout au long de cette thèse. Ce fut un réel plaisir et un honneur de travailler avec vous. Je n'aurais pu souhaiter un meilleur duo pour m'accompagner. Merci de m'avoir éclairé de vos lumières. A l'avenir, je recevrai vos nouvelles avec une immense joie.

J'adresse mes plus sincères remerciements à Natacha Heutte, Delphine Blanke et Denys Pommeret pour m'avoir fait l'honneur de rapporter ma thèse. Je les remercie pour leur travail de relecture, leur rapport éclairé et leur présence à la soutenance. C'est avec un grand plaisir que remercie Anne Françoise Yao et Sergeï Dachian, pour m'avoir enseigner les statistiques en master et pour avoir accepté de faire partie de mon Jury de thèse.

Je remercie Olivier Brachet, directeur général d'IPA, pour m'avoir accueilli au sein de son entreprise et nous avoir fourni les données sans lesquelles cette thèse n'aurait pu voir le jour. Je remercie toute l'équipe d'IPA, ce fut toujours un plaisir de vous retrouver.

Je remercie les membres du laboratoire de mathématiques pour leur accueil. Je remercie également l'ensemble des doctorants avec qui j'ai eu la chance de partager de très bons moments, que ce soit dans notre bureau clermontois ou lors des différents séminaires.

Je remercie mes amis dijonnais pour tous les moments passés ensemble depuis des années. Je pense plus particulièrement à Simon, pour l'ensemble des dessert gratuits! A Valentine, pour toutes ces histoires croustillantes et autres potins! A Martin, avec qui l'on peut regarder par la fenêtre des heures durant en discutant de tout! A Cyril, pour ses précieux conseils sportifs! Je remercie également Fabien pour son hospitalité et nos longues soirées aux débats animés.

Je remercie avec amour ma mère et ma sœur, qui ont toujours cru en moi. Elles m'ont toujours tout donné et sans elles, je ne serais jamais arrivé jusqu'ici. Je remercie également toute ma famille et plus particulièrement ma grand-mère, je chérie chaque moment passé avec elle. J'adresse également un remerciement chaleureux à toute ma belle-famille pour leur accueil, leur soutien et leur conseils.

Enfin, c'est avec un plaisir non dissimulé que je remercie ma chérie, le « Docteur Chacha » (colonel Chacha pour d'autres!...), qui a toujours su trouver les mots pour me faire avancer, qui m'apporte conseils et soutien dans chaque décision que je prends. Et c'est toujours avec la même joie, immense, que je regarde fleurir un sourire sur son visage...

Table des matières

Liste des figures

Liste des tableaux

Chapitre 1

Introduction

Le football est le sport le plus populaire et le plus médiatisé au monde, regroupant des millions de joueurs. Les enjeux souvent importants, parfois de l'ordre du politique, sont disputés avec engagement par les joueurs ce qui peut entrainer des blessures. Ces blessures ont des répercutions, sur le joueur qui peut voir son niveau de jeu diminuer, mais aussi sur l'équipe qui peut perdre un élément important dans un moment crucial de la saison. Le football, sport d'engagement, peut entrainer des blessures après contact entre deux joueurs ou entre un joueur et le sol. Ces blessures imprévisibles ne peuvent être évitées. Au contraire, les blessures musculaires sans contact, qui surviennent du fait de causes multiples, par exemple la fatigue d'un joueur ou un mauvais entrainement, sont prévisibles et peuvent être prévenues. Les clubs et leurs équipes accordent une grande importance à ces blessures et de nombreuses publications médicales et sportives sont réalisées dans ce domaine. Si les blessures et leur traitement sont bien connus des staffs médicaux et des clubs, leur mécanisme d'apparition l'est encore insuffisamment. Il existe de multiples facteurs, certains connus mais encore non étudiés, le plus important étant le caractère personnel et individuel de la blessure. Ainsi, des facteurs pourtant identiques n'entraineront pas les mêmes conséquences sur deux joueurs différents. La prévalence de ces blessures se situe entre 20 à 40 pour 1000 heures de jeu en match ([M. et collab., 2005] [JAN et collab., 2011] et [DANIEL et JAVIER, 2017]), ce qui représente moins d'une blessure par match, donnant un caractère d'évènement rare au phénomène.

La prévision des évènements rares (crack boursier, catastrophe naturelle) est un domaine des mathématiques encore peu étudié, qui complique la prédiction. En effet, considérée pendant plusieurs années comme exceptionnelle et hasardeuse, leur étude a été plus ou moins mise de coté. Par conséquent, les outils habituels de prédiction donnent des résultats de qualité insuffisante, qui ont tendance à sous-évaluer le risque d'apparition d'un évènement rare. De plus, les métriques de comparaison de méthodes utilisées habituellement ne sont pas adaptées. Ces difficultés font que la recherche de modèles prédictifs de blessures sans contact n'en est qu'à son début et qu'aucun outil fiable n'a été présenté jusqu'alors dans la littérature [JOÃO GUSTAVO et collab., 2019] . Cependant, certaines études portant sur des blessures particulières et ciblées, (par exemple blessure à la cuisse [AYALA et collab., 2019]) ont montré de bons résultats. Ceci est encourageant pour l'avenir.

L'objectif au cours de cette thèse a été de déterminer une méthode de modélisation du risque de blessure musculaire sans contact chez les joueurs de football professionnels.

Dans la première partie de cette thèse, nous présenterons les méthodes de ré-échantillonnage, qui seront détaillées et expliquées. Ce sont des outils très pratiques pour rééquilibrer les données, donc pour obtenir de l'information supplémentaire sur les évènements rares. Que ce soit en augmentant le nombre de données : l'oversampling, en le diminuant : l'undersampling, ou bien en combinant les deux. Aujourd'hui, les méthodes d'échantillonnage constituent l'une des branches de la recherche des données déséquilibrées la plus active ; il existe de nombreuses méthodes que nous présenterons dans ce travail. Ensuite, nous présenterons les méthodes classiques de modélisation. Certaines méthodes sont plus performantes que d'autres pour prédire les évènements rares. D'autres qui présentaient des biais en présence de données déséquilibrées ont subit des mo-

difications pour palier ce problème. Enfin, quelques méthodes seront présentées pour leur faible niveau prédictif (les weak learners), mais qui peuvent donner tout de même de très bons résultats associées à d'autres stratégies, comme les méthodes d'agrégation.

Ces différentes méthodes, que nous présenterons, permettent de construire un modèle fiable à partir de plusieurs modèles faibles. Souvent chronophages, ces méthodes ont prouvé leur efficacité dans le domaine des évènements rares, mais aussi dans le domaine du big Data

Dans la deuxième partie, nous présenterons les problématiques des évènements rares et de la blessure. Puis, nous définirons et utiliserons les données réelles de joueurs de football d'une équipe professionnelle suivie pendant 5 saisons. À travers une analyse des variables, nous montrerons la difficulté à prédire les blessures. Ensuite, nous modéliserons la blessure à l'aide des méthodes présentées dans la partie 1. Nous montrerons l'effet des méthodes de ré-échantillonnage et d'agrégation sur notre jeu de données avec évènements rares. Enfin, nous présenterons les meilleurs résultats obtenus.

Dans le dernier chapitre, le chapitre 3, nous présenterons les indices de Sobol et leur application sur nos données. En effet, les modèles que nous allons construire seront souvent "boite noire" ; pourtant, si la bonne prédiction de la blessure est importante, les causes qui l'entraine le sont également. Pour améliorer notre connaissance des blessures nous utiliserons donc les méthodes d'analyse de la sensibilité de Sobol.

1.1 Références

AYALA, F., A. LÓPEZ-VALENCIANO, J. GÁMEZ MARTÍN, M. DE STE CROIX, F. VERA-GARCIA, M. GARCÍA-VAQUERO, I. RUIZ-PÉREZ et G. MYER. 2019, «A preventive model for hamstring injuries in professional soccer : Learning algorithms.», *Int J Sports Med.* 1

DANIEL, C. et R.-G. JAVIER. 2017, «The prevalence of injuries in professional soccer players», *Journal of Orthopedic Research and Therapy.* 1

JAN, E., H. MARTIN et W. MARKUS. 2011, «Epidemiology of muscle injuries in professional football (soccer)», *The American Journal of Sports Medicine.* 1

JOÃO GUSTAVO, C., D. O. C. DANIEL, V. D. S. THIAGO, S. JULIO CERCA, M. P. ADRIANO C. et N. GEORGE P. 2019, «Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports : a systematic review», *Sports Medicine - Open.* 1

M., H., W. M. et E. J. 2005, «Injury incidence and distribution in elite football–a prospective study of the danish and the swedish top divisions», *Scan J Med Sci Sports.* 1

Chapitre 2

Méthodes de prédiction des évènements rares

Sommaire

Lors de la réalisation d'un modèle statistique, plusieurs étapes sont effectuées :

1. pré-traitement des données,
2. sélection et utilisation d'une méthode de modélisation adaptée à la problématique,
3. évaluation et mesure de la performance de la modélisation effectuée.

Au cours de ce premier chapitre nous présenterons, des méthodologies adaptées à ces trois points dans le cas des évènements rares, en commençant par les méthodes de pré-traitement des données. Dans le cas de données déséquilibrées, cette étape consiste à utiliser des méthodes de ré-échantillonnage pour les équilibrer.

2.1 Méthodes de ré-échantillonnage des données

La première stratégie présentée est le ré-échantillonnage de données. L'idée principale est d'équilibrer les données afin d'améliorer l'apprentissage de la classe minoritaire. Plusieurs méthodologies existent, certaines méthodes vont simplement supprimer des données aléatoirement, d'autres vont synthétiser des observations non existantes dans la base et, d'autres vont chercher les points les moins importants de chaque classe pour les supprimer et rendre l'apprentissage le plus efficacement possible.

Plus formellement, prenons un jeu de données S avec m observations (i.e., le cardinal de S : $|S| = m$). Définissons $S = \{(x_i, y_i), i = 1...m\}$, avec $x_i \in \mathscr{X}$, où $\mathscr{X}$ est l'espace de dimension n des variables explicatives et $y_i \in \{0,1\}$ la variable réponse. Le nouveau jeu de données obtenu par ré-échantillonnage sera noté S'. Finalement la classe majoritaire sera noté S_{maj}, la classe minoritaire S_{min} tel que $S_{maj} \subset S$, $S_{min} \subset S$, $S_{min} \cap S_{maj} = \{\emptyset\}$, $S_{min} \cup S_{maj} = S$. Nous parlerons d'oversampling lorsque l'on ajoutera des données ($|S'_{min}| > |S_{min}|$) et d'undersampling lorsque la taille des données sera réduite ($|S'_{maj}| < |S_{maj}|$).

Il existe différentes méthodes d'oversampling et d'undersampling. Certaine méthodes peuvent même combiner les deux. Nous commencerons par exposer les méthodes d'undersampling puis nous étudierons celles d'oversampling.

2.1.1 Undersampling

Les méthodes d'undersampling consistent à supprimer des observations afin de réduire le déséquilibre entre la classe minoritaire et la classe majoritaire. Différentes méthodes existent pour choisir les données à ne pas sélectionner. L'une des plus simples est la méthode appelée Random Undersampling.

Random undersampling

Le "Random undersampling" est une méthode qui consiste à équilibrer les données en supprimant par tirage aléatoire sans remise un nombre d'observations de la classe majoritaire. On obtient donc un nouveau jeu de données S' tel que, $S'_{min} = S_{min}$ et $S'_{maj} < S_{maj}$.

Cette méthode nécessite un unique paramètre : le pourcentage d'observations de la classe majoritaire supprimées. Choisir un pourcentage trop important peut être néfaste sur la performance du classifieur du fait de la perte d'information de la classe majoritaire. Le plus fréquemment, on choisit un pourcentage permettant d'obtenir un jeu de données équilibré, c'est à dire tel que $S'_{min} = S_{maj}$. Cette méthode est la plus simple à mettre en place. Elle permet d'améliorer la connaissance de la classe minoritaire au dépend de la classe majoritaire [DRUMMOND et HOLTE, 2003]. Cependant, dans le cas de données avec évènements rares, l'équilibrage nécessite la suppression d'une partie importante des observations de la classe majoritaire (entre 15 à 30%), ce qui ajoute de l'aléa. Les modèles construits sont donc moins stables et nécessitent l'utilisation d'une méthode d'agrégation pour les stabiliser. La figure 2.1 montre un exemple d'undersampling lorsque 50% de la classe majoritaire est supprimée.

FIGURE 2.1 – Random undersampling (50%)

Tomek links

La méthode de "Tomek links" est plus complexe que le Random undersampling. Elle a été développée par Tomek [TOMEK, 1976] et repose sur un calcul de distance. Elle consiste à rechercher des paires de points de chaque classe en bordure de zone séparant la classe majoritaire et la classe minoritaire. Cette paire de points est appelée "Tomek links". Cette méthode nécessite de définir la distance entre deux individus :

Définition 1 ***Distance mathématique***

Soit E un ensemble. Toute application $d : \mathrm{E} \times \mathrm{E} \rightarrow \mathbb{R}$ *est appelée distance si et seulement si elle vérifie les propriétés suivantes :*

$$\textit{Symétrie} : \forall (a,b) \in \mathrm{E}^2,\ d(a,b) = d(b,a).$$
$$\textit{Séparation} : \forall (a,b) \in \mathrm{E}^2,\ d(a,b) = 0 \Leftrightarrow a = b.$$
$$\textit{Inégalité triangulaire} : \forall (a,b,c) \in \mathrm{E}^2,\ d(a,c) \leqslant d(a,b) + d(b,c)$$

Il existe de nombreuses distances. Nous définirons les plus classiques. Prenons $x = (x_1, x_2 ... x_n)$ et $y = (y_1, y_2 ... y_n)$ deux points de $\mathbb{R}^n$. Les différentes distance entre x et y se définissent comme suit :

— La distance de Manhattan :

$$d(x,y) = \sum_{i=1}^{n} \mid x_i - y_i \mid \tag{2.1}$$

— La distance de euclidienne :

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.2}$$

— La distance de Minkowski (généralisation de la distance euclidienne) :

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p} \tag{2.3}$$

— La distance de Techbychev :

$$d(x,y) = \lim_{p \to +\infty} \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p} = \max_i (\mid x_i - y_i \mid). \tag{2.4}$$

Le choix de la mesure est très important et peut modifier les résultats obtenus. Plusieurs autres distances existent, un comparatif de celles-ci dans une modélisation knn peut être trouvé dans l'article de Chomboon & al. [CHOMBOON et collab., 2015]. Nous pouvons maintenant donner une définition formelle des liens de Tomek :

Définition 2 ***Tomek links***

Deux points $x_i \in \mathrm{S}_{min}$ *et* $x_j \in \mathrm{S}_{maj}$ *forment un lien de Tomek si et seulement si, :*

$$\forall x_k \in \mathrm{S}_{min} \text{ et } \forall x_{k'} \in \mathrm{S}_{maj} : d(x_i, x_j) < min(d(x_i, x_{k'}), d(x_j, x_k)) \tag{2.5}$$

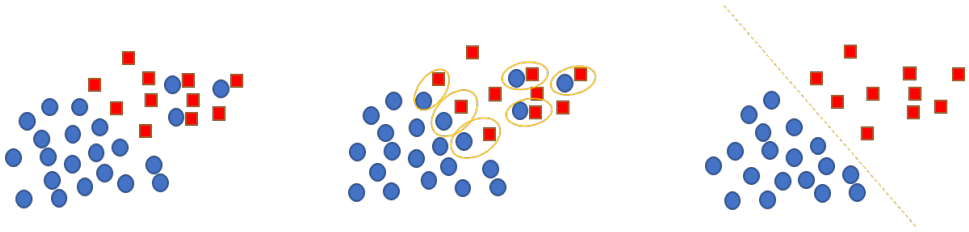


FIGURE 2.2 – Tomeks links

Une fois l'ensemble de ces paires trouvées, les observations de la classe majoritaire appartenant à une paire "Tomek links" sont supprimées pour rééquilibrer les données. Cette méthode permet de créer de l'espace et de faciliter la séparation entre les 2 classes. La figure 2.2 illustre cette méthode d'undersampling. Les paires de Tomek sont entourées en orange. Une étude comparative peut être trouver dans l'article de Kubat & Matwin [KUBAT, 2000]. Ils montrent que la suppression de certains points de la classe majoritaire en utilisant les liens de Tomek permet d'améliorer la prévision de la classe minoritaire mais, dans certains cas en diminuant la prévision de la classe majoritaire et donc la précision globale. Dans le cas où les classes ne sont pas déséquilibrées, cette méthode peut être également utilisée comme méthode de nettoyage des données. Dans ce cas les observations de la classe minoritaire et de la classe majoritaire sont supprimées.

NearMiss

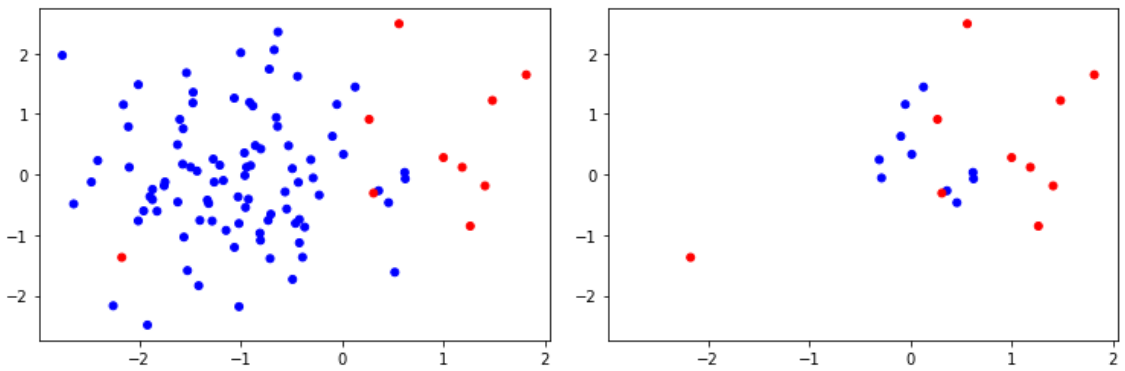
NearMiss est une méthode d'undersampling proposée par [ZIANG, 2003]. Elle a pour but de se concentrer sur la frontière entre la classe majoritaire et la classe minoritaire et donc de sélectionner uniquement les observations de la classe majoritaire proches de la classe minoritaire. NearMiss possède plusieurs variantes.

— NearMiss-1 : seules les observations de la classe majoritaire les plus proches d'une observation de la classe minoritaire sont conservées. Les données sont totalement équilibrées.

— NearMiss-2 : les observations les plus proches en moyenne de deux observations de la classe minoritaire sont conservées.

— Plus généralement, NearMiss-k : les observations les plus proche en moyenne de k observations de la classe minoritaire sont conservées. Souvent k est choisi inférieur à 5.

Dans [FIX et HODGES, 1989] les meilleurs résultats ont été obtenus en utilisant NearMiss-2. Avec cette méthode, l'undersampling effectué est donc important puisque la plupart des observations de la classe majoritaire sont supprimées. La figure 2.3 présente l'application de la méthode NearMiss-3



(a) Données déséquilibrées (b) Application de NearMiss

FIGURE 2.3 – Undersampling avec NearMiss

Condensed Nearest Neighbor Rule (CNN)

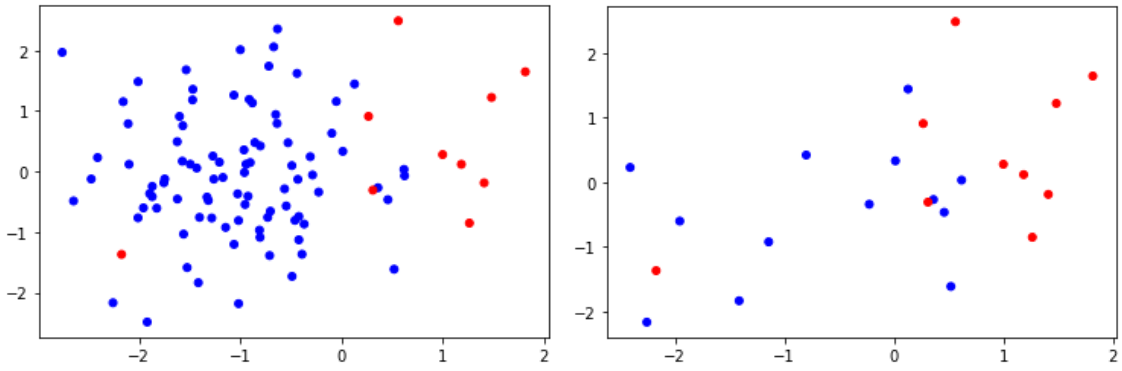
CNN [HART, 1968; HILBORN et LAINIOTIS, 1967] est utilisé pour réduire le nombre d'observations de la classe majoritaire en sélectionnant des sous-ensembles $\hat{S}$, dit consistant de l'ensemble de référence de S.

Un sous-ensemble $\hat{S}$ est consistant, s'il classe correctement l'intégralité de l'ensemble de référence S en utilisant la méthode des plus proches voisins à un seul voisin (1-NN, défini en Section 2.2.5). Il est toujours possible de trouver un sous-ensemble consistant, puisque l'ensemble S est consistant pour lui-même. Le plus intéressant est de trouver le sous-ensemble consistant minimal, c'est-à-dire qui contient le moins d'observations. Souvent, il existe plusieurs sous-ensembles minimaux. Dans ce cas, il est possible de choisir le sous-ensemble qui convient le mieux.

Le procédé pour trouver un sous-ensemble consistant est le suivant :

1. prenons S l'ensemble d'apprentissage de départ.
2. Au départ, l'ensemble $\hat{S}$ est constitué de l'ensemble des observations de la classe minoritaire.
3. En utilisant 1-NN, les observations de S_{maj} sont classées en utilisant $\hat{S}$ avec la méthode des 1-NN, les mal classées sont alors intégrées à $\hat{S}$. Les bien classées sont intégrées dans un ensemble appelé $\bar{S}$.
4. Une fois l'ensemble S parcouru intégralement, le procédé recommence, en reparcourant les observations mises dans $\bar{S}$ qui ont été correctement classées. Deux possibilités peuvent terminer le processus :
   (a) l'ensemble $\bar{S}$ est vidé et l'ensemble des observations sont dans $\hat{S}$, dans ce cas le sous-ensemble consistant est l'ensemble S d'origine.
   (b) Si après un passage $\bar{S}$ aucune observation n'est placée dans $\hat{S}$.
5. Finalement le sous-ensemble consistant est $\hat{S}$.

Le sous-ensemble consistant minimal (celui avec le moins d'observations) n'est pas toujours obtenu avec cette méthode. Mais un autre est trouvé permettant de réduire la taille de l'échantillon. Un exemple est donné dans la figure 2.4. Il est à noter que des variantes existent en utilisant la méthodes des plus proches voisins avec plus de voisins (2-NN, 3-NN etc.).

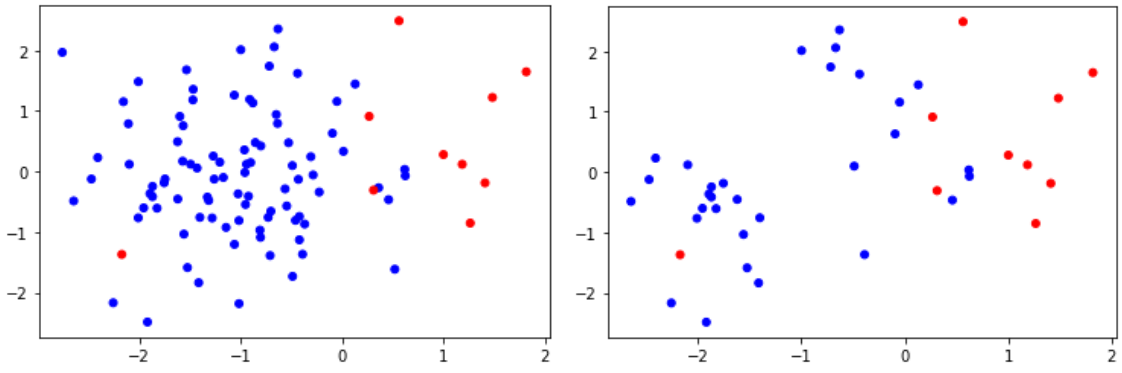


(a) Données déséquilibrées (b) Application de CNN rule

FIGURE 2.4 – Undersampling avec CNN rule

One-sided selection

One-sided selection KUBAT [2000] est une méthode dont l'objectif est d'améliorer l'information apportée par les "bords" de chaque classe en supprimant les observations de la classe majoritaire qui en sont éloignées. Pour cela, elle combine la méthode des liens de Tomek (section 2.1.1) et de CNN rules (section 2.1.1). Dans un premier temps un sous-ensemble consistant $\hat{S}$ est construit.



(a) Données déséquilibrées (b) Application de One-sided selection

FIGURE 2.5 – Undersampling avec One-sided selection

Ensuite, les points considérés comme du bruit, formant les liens de Tomek, sont supprimés. Cette méthode permet d'un côté en utilisant CNN-rules de supprimer les observations de la classe majoritaire éloignées de la bordure et d'un autre côté, la méthode de Tomek, permet de supprimer les observations de la classe majoritaire proches de la bordure qui sont considérées comme du bruit [FERNÁNDEZ et collab., 2018]. One-side selection peut se décrire comme :

1. prenons S l'ensemble d'apprentissage de départ.
2. Au départ, l'ensemble C est constitué de l'ensemble des observations de la classe minoritaire.
3. En utilisant 1-NN, les observations de S_{maj} sont classées en utilisant C avec la méthode des 1-NN, les mal classées sont alors intégrées à C.
4. Une fois C construit. Les observations de la classe majoritaire de C formant un lien de Tomek sont supprimées.
5. C peut alors servir de base d'apprentissage.

La figure 2.5 représente la base de données obtenue en utilisant cette méthode. Sur l'exemple que nous présentons, nous voyons clairement la bordure se former. Le principal problème de One-sided selection et Condensed Nearest Neighbor Rule est qu'elles sont particulièrement sensibles au bruit (observations présentant un évènement par "accident") AHA et collab. [1991]; WILSON et MARTINEZ [2000]. En effet, les bruits seront mal classés et donc utilisés dans la base d'apprentissage.

Neighborhood Cleaning Rule

La dernière méthode que nous présentons est Neighborhood Cleaning Rule LAURIKKALA [2001] (figure 2.6). Elle est similaire à One-sided selection, mais diffère sur certains points :

— la méthode utilisée pour supprimer les bruits est Edited Nearest Neighbor rule (ENN) WILSON [1972]. Elle consiste à supprimer les observations de la classe majoritaire mal classées par la méthode 3-NN.

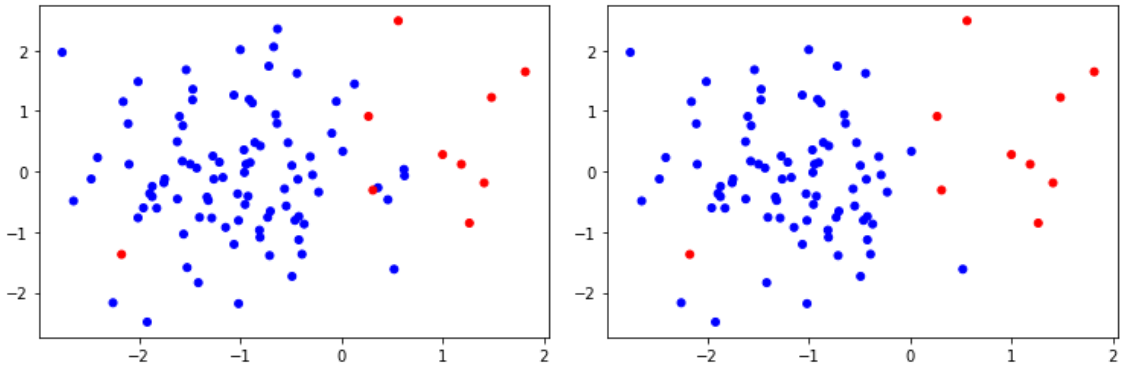
— Si une observation de la classe minoritaire est mal classée, ce sont les observations de la classe majoritaire formant ses 3 plus proches voisins qui sont supprimées.

— L'undersampling est limité à 50% de la classe majoritaire.

Cette méthode a montré de meilleurs résultats que one-side selection LAURIKKALA [2001].

2.1.2 Oversampling

Les méthodes d'undersampling cherchent à équilibrer les deux classes en supprimant des observations de la classe majoritaire afin d'améliorer la connaissance de la classe minoritaire. Cela



(a) Données équilibrées (b) Application de Neighborhood Cleaning Rule

FIGURE 2.6 – Undersampling avec Neighborhood Cleaning Rule

entraine une perte d'information sur la classe majoritaire. Une autre possibilité est d'utiliser les méthodes d'oversampling. Elles cherchent à augmenter l'information apportée par la classe minoritaire en dupliquant ou synthétisant des observations de cette classe.

Random Oversampling

Le Random Oversampling consiste à augmenter le nombre d'observations de la classe minoritaire en les dupliquant par tirage aléatoire avec remise. On obtient $|S'_{min}| > |S_{min}|$. L'information de la classe minoritaire est donc augmentée sans perte d'information sur la classe majoritaire. Le seul paramètre à prendre en compte est le ratio souhaité β entre la classe majoritaire et minoritaire, $\beta = \frac{|S'_{min}|}{|S_{maj}|}$:

— $\beta = \frac{|S_{min}|}{|S_{maj}|}$. Aucune donnée n'est générée ($|\hat{S}_{min}| = |S_{min}|$)

— $\beta = 1 \Rightarrow |\hat{S}'_{min}| = |S_{maj}|$. Des données sont générées pour que les classes minoritaire et majoritaire soient équilibrées.

Le nombre d'observations de la classe minoritaire à générer est donc donné par :

$$G = |S_{maj}| * \beta - |S_{min}| \tag{2.6}$$

Habituellement, on choisit d'équilibrer les données de sorte que $|S'_{min}| = |S_{maj}|$. Lorsque l'on parle d'oversampling une notation couramment utilisée également est '$a:b$', signifiant a observation de la classe minoritaire pour b observation de la classe majoritaire. Ainsi pour un jeu de données équilibré, la notation devient $1:1$. Nous utiliserons cette notation pour toutes les méthodes d'oversampling. Avec la méthode random oversampling, il est important de noter que :

— le nombre de duplications de chaque observation peut être différent.

— Une observation peut ne pas être sélectionnée ce qui peut entrainer une perte de stabilité dans la modélisation.

Ce problème peut être pallier en utilisant les méthodes agrégatives que l'on présentera par la suite.

Un autre problème peut venir des observations de la classe minoritaire accidentelles (non prévisibles). L'information erronée qu'elles apportent peut être dupliquée et donc entrainer une perte de performance de la modélisation.

Synthetic Minority Oversampling Technique (SMOTE)

SMOTE est une technique puissante basée sur les k-nn, où k est fixé qui a rencontré un fort succès dans de nombreuses applications [CHAWLA et collab., 2002]. La méthode consiste à créer

FIGURE 2.7 – Exemple de SMOTE en considérant 2 voisins

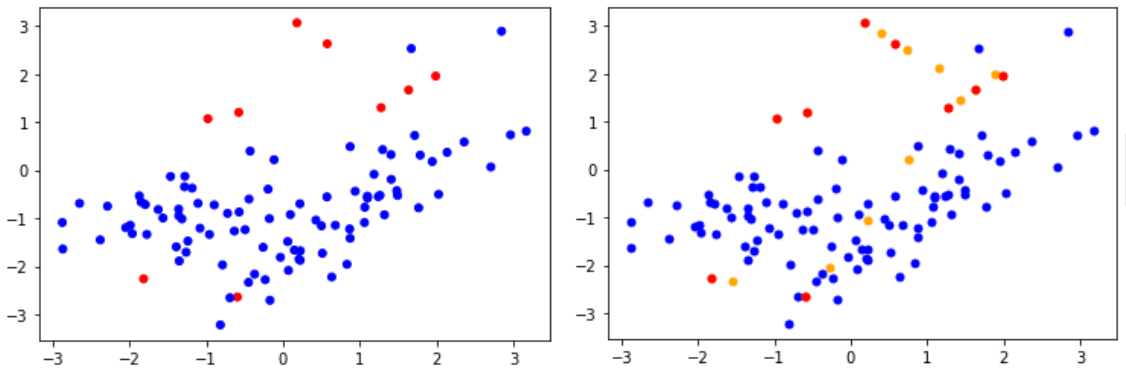
des données de la classe minoritaire synthétiques qui tiennent compte des caractéristiques dans l'espace de cette classe. Plus précisément, pour chaque observation de la classe minoritaire $x_i \in S_{min}$ une observation $\tilde{x_i}$ est créée sur le segment $[x_i, x_j]$, j étant l'un de ses k-plus proches voisins de la classe minoritaire sélectionné aléatoirement tel que

$$\tilde{x_i} = x_i + (x_j - x_i)\gamma \tag{2.7}$$

avec $\gamma \in [0,1]$ un nombre aléatoire. La figure 2.7 illustre le fonctionnement de la méthode basée sur les 2-NN.

Les deux seuls paramètres à régler en utilisant SMOTE sont le nombre de voisins sélectionnés et le taux d'oversampling souhaité. Le nombre de voisins est particulièrement important, puisque les observations synthétiques sont créées entre deux points de la classe minoritaire sans considérer la classe majoritaire. Certains nouveaux exemples peuvent être ambigus, i.e. situés dans des zones de non-évènement et donc augmenter les erreurs de classification. Ce phénomène est présenté dans la figure 2.8, cinq plus proches voisins ont été sélectionnés pour améliorer l'apprentissage de la classe minoritaire (rouge). Certains points synthétisés par SMOTE (en orange) se retrouvent au centre de la classe majoritaire. SMOTE peut donc être particulièrement sensible aux évènements qui se produisent sans cause, i.e. par accident.

Pour améliorer l'effet de la méthode, il est préconisé d'utiliser une méthode d'undersampling en complément afin d'équilibrer plus facilement les classes [HE et MA, 2013]. Dans leur article [VAN HULSE et KHOSHGOFTAAR, 2009] montrent que, lorsque les données sont bruitées, la méthode SMOTE n'obtient pas de bons résultats au niveau de l'AUC alors que la méthode de random undersampling est l'une des meilleures. Combiner les deux méthodes peut donc être une bonne solution pour améliorer les résultats



(a) Données brutes (b) Application de SMOTE

FIGURE 2.8 – Application de SMOTE lorsque des évènements sont accidentels

Méthodes SMOTE dérivées

Il existe de nombreuses méthodes qui sont des améliorations ou des dérivations de SMOTE. Nous proposons d'en présenter quelques-unes.

Borderline-SMOTE

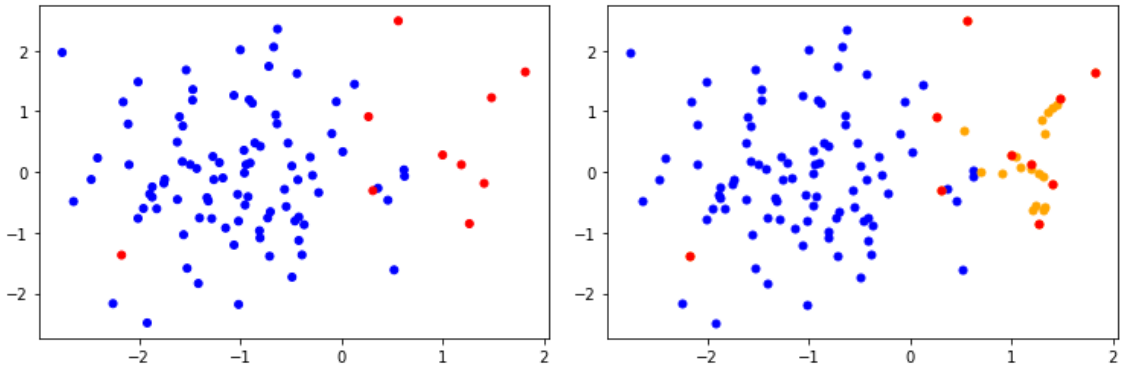
L'un des problèmes de la méthode SMOTE est qu'elle génère des cas pour chaque observation de la classe minoritaire S_{min} sans regarder si l'observation est un bruit. Borderline-SMOTE (HAN et collab. [2005]) pallie ce problème. En effet, pour qu'un point synthétique soit généré, certaines conditions doivent être remplies. Pour commencer, la méthode détermine l'ensemble des m-plus proches voisins pour chaque $x_i \in S_{min}$. Ces ensembles seront notés $S_{i,m-nn} \subset S$. Ensuite, pour chaque x_i il faut calculer le nombre de m-plus proches voisins appartenant à la classe majoritaire S_{maj}, c'est-à-dire $|S_{i,m-nn} \cap S_{maj}|$. Les x_i sont alors classés en trois catégories :

— **Sûr :** $|S_{i,m-nn} \cap S_{maj}| \leq \frac{m}{2}$. La majorité des m-plus proches voisins appartiennent à la classe minoritaire S_{min}. La zone est considérée comme sure et ne **nécessite pas** de création de données minoritaires synthétiques.

— **Danger :** $\frac{m}{2} \leq |S_{i,m-nn} \cap S_{maj}| \leq m$. C'est une zone à risque qui **nécessite** la création de données minoritaires synthétiques.

— **Bruit :** $m \leq |S_{i,m-nn} \cap S_{maj}|$. Les m plus proches voisins appartiennent à la classe majoritaire S_{maj}. La donnée est donc considérée comme du bruit et ne **nécessite pas** de création de données minoritaires synthétiques.

Le résultat de cette méthode est présenté dans la figure 2.9.



(a) Données brutes (b) Application de BorderlineSMOTE

FIGURE 2.9 – Effet de Borderline-SMOTE

SMOTE + Tomek links

Cette méthode effectue un oversampling et un undersampling. C'est une combinaison des méthodes SMOTE pour l'oversampling et les liens de Tomek pour supprimer les points en bordure de classe. Elle a notamment été comparée à SMOTE et au random Over/undersampling et a montré les deuxièmes meilleurs résultats, juste après SMOTE BATISTA et collab. [2003].

SMOTE + ENN

Comme pour le méthode précédente, c'est une combinaison de SMOTE et de la méthode d'undersampling Edited Nearest Neighbor rule, qui consiste à supprimer les observations mal classées par la méthode des 3-NN. On trouve une comparaison des méthodes d'échantillonnage dans [BATISTA et collab., 2004a].

ADASYN

ADASYN HE et collab. [2008] reprend les généralités de SMOTE. En effet, la méthode génère des observations synthétiques, mais cette fois en tenant compte des k plus proches voisins de

chaque observation de la classe minoritaire. Le seul paramètre est le taux d'équilibrage désiré β. On note G le nombre d'observations de la classe minoritaire à générer (équation 2.6)

Pour chaque observation de la classe minoritaire x_i, on défini r_i le taux de K plus proches voisin appartenant à la classe minoritaire tel que :

$$r_i = \frac{\Delta_i}{\mathrm{K}}, \tag{2.8}$$

avec Δ_i le nombre des K plus proches voisins appartenant à la classe majoritaire. Ainsi $r_i \in [0,1]$ Pour obtenir une densité de distribution, r_i est normalisé de telle sorte que $\sum_i \hat{r}_i = 1$

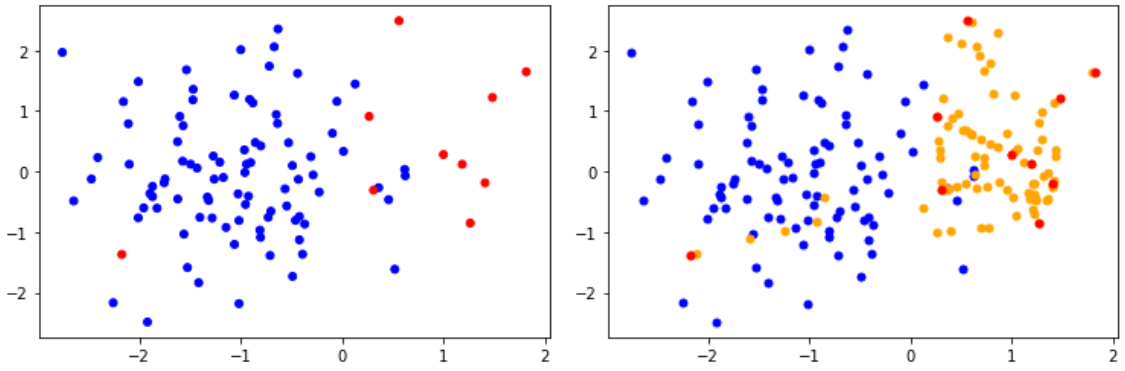
$$\hat{r}_i = \frac{r_i}{\sum_i r_i}. \tag{2.9}$$

À l'aide de cette densité, il est alors possible de calculer le nombre d'observations synthétiques g_i à générer par observation :

$$g_i = \hat{r}_i * \mathrm{G}. \tag{2.10}$$

Les observations synthétiques sont ensuite générées comme pour la méthode SMOTE. Avec Adasyn, plus une observation de la classe minoritaire est proche de la classe majoritaire, plus elle va générer de cas synthétiques. À l'inverse une observation minoritaire au "centre" de sa classe ne génère aucun point. Cela force les méthodes d'apprentissage à se concentrer sur les observations compliquées à classer a priori. Ce qui induit également un risque de sur-interprétation des informations apportées par le "bruit".

La figure 2.10 illustre l'utilisation d'adasyn lorsque β = 1. Les points synthétiques apparaissent en orange. On observe distinctement les deux classes, mais aussi le problème du bruit qui ajoute des observations de la mauvaise classe dans la classe majoritaire (bleu)



(a) Données brute

(b) Application de Adasyn

FIGURE 2.10 – Effet de Adasyn

Dans l'article de Haibo He (HE et collab. [2008]) la méthode ADASYN est comparée à SMOTE et "sans méthode de sampling", en utilisant les arbres de classification sur 5 jeux de données différents. Sur ces cinq jeux de données, ADASYN est la méthode qui a la plus souvent le meilleur classement. Il est également intéressant de noter que SMOTE obtient de moins bons résultats qu'en utilisant la table initiale (i.e sans méthode de sampling).

En résumé

Dans ce chapitre, nous avons présenté des méthodes permettant l'amélioration de l'apprentissage de la classe minoritaire par undersampling, oversampling ou par combinaison des deux. Il existe de nombreuses autres méthodes qui sont souvent des dérivations des méthodes présentées ci-dessus. Les possibilités de combinaisons sont nombreuses et l'efficacité dépend des données. Le choix de la bonne méthode est difficile. Plusieurs articles présentant et comparant les

différentes méthodes peuvent être trouvés dans la littérature [BATUWITA et PALADE, 2010], [BATISTA et collab., 2004b], [LIU et collab., 2009] et [HE et GARCIA, 2009] [ABD RANI et collab., 2013]. Ces articles n'obtiennent pas toujours la même "meilleures méthodes". Le choix est d'autant plus compliqué que l'effet de la méthode de sampling est modifié par la méthode de modélisation choisie. Par exemple dans l'article de Bing Zhu ZHU et collab. [2017] pour la régression logistique le meilleur choix (au niveau de l'AUC) est de ne pas utiliser de méthode de sampling, alors qu'avec C4.5 et les SVM, c'est le random Oversampling. En revanche dans l'article de SEIFFERT et collab. [2008], pour la méthode C4.5, c'est l'undersampling qui obtient les meilleurs résultats.

En bref, il n'existe pas une meilleure méthode qui fonctionne à tous les coups. Le bon choix est donc laissé à l'expert qui doit sélectionner la méthode correspondant le mieux à ses données. A travers les articles parcourus, globalement SMOTE donne de bons résultats. On peut noter cependant que les méthodes plus simples telles que random-oversampling obtiennent de très bons résultats également.

2.2 Méthodes de modélisation

Les méthodes de ré-échantillonnage permettent de faire un traitement des données. Il faut ensuite apprendre de ces données, afin de proposer un risque de blessure. Ce sont les méthodes de modélisation statistique qui le permettent. Dans notre cas, la réponse à prédire Y étant binaire, nous présenterons uniquement les modélisations dans ce cadre. Certaines modélisations permettent d'estimer une probabilité qu'un évènement se produise. C'est le cas de la régression logistique par exemple. D'autres méthodes, se contentent de classer les observations. Il est alors impossible d'obtenir une probabilité.

Les méthodes classiques de prédiction montrent des biais lorsque les jeux de données sont déséquilibrés, notamment une sous-évaluation du risque qu'un évènement se produise. Ces biais sont différents d'une méthode à une autre. Sur certaines, des corrections existent. Dans ce chapitre nous présenterons les méthodes les plus utilisées pour la prédiction de réponses binaires, ainsi que les améliorations qu'il est possible d'effectuer pour la prédiction d'évènements rares. Nous commencerons par la régression logistique, puis la régression de Poisson. Ensuite nous présenterons les arbres de classification et les random Forest. Nous terminerons par les modèles de SVM et de réseaux de neurones.

2.2.1 Régression logistique

La régression logistique ou modèle logit est un modèle de régression binomial HASTIE et collab. [2004]; HILBE [2009]; MCCULLAGH et NELDER [1989]. Elle présente de nombreuses similitudes avec la régression linéaire classique utilisée pour les variables réponses continues. La régression linéaire a pour but d'estimer une variable réponse quantitative $Y \in \mathbb{R}$ en fonction du vecteur des covariables (ou des variables explicatives) $X = (x_1, x_2, ...x_k)$, selon l'équation suivante :

$$Y = \beta_0 + \sum_{j=1}^{k} \beta_j * x_j \tag{2.11}$$

avec $\beta_0, \beta_1, \beta_2, ..., \beta_k$ des paramètres inconnus à estimer (figure 2.11).

La figure 2.12 illustre l'utilisation de régression linéaire lorsque la variables réponse Y est binaire, les observations présentant un évènement sont en rouge, les autres en bleu. Le modèle linéaire n'est pas adaptée à ce type de données.

La régression logistique, quand à elle, a pour but d'estimer une variable réponse qualitative binaire Y prenant par exemple les valeurs 0 ou 1 en fonction du vecteur des variables explicatives $X = (x_1, x_2, ...x_k)$. La probabilité $P(Y = 1|X)$ n'est pas obtenue de manière directe mais en estimant le logit de cette probabilité $logit(P(Y = 1|X))$.

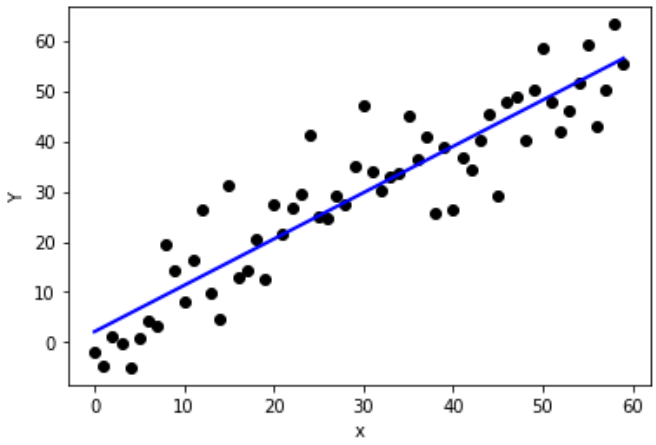


FIGURE 2.11 – Régression linéaire avec réponse continue

Pour rappel la fonction logit pour $p \in [0,1[$ est définie par :

$$logit(p) = ln\left(\frac{p}{1-p}\right) \tag{2.12}$$

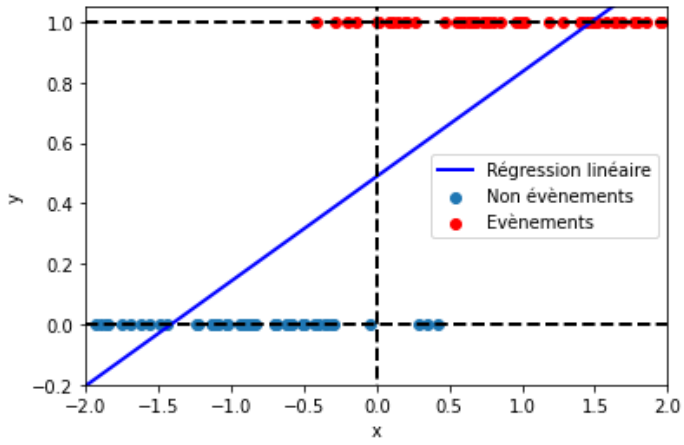


FIGURE 2.12 – Régression linéaire avec réponse binaire

La fonction logit ajuste mieux les données comme le montre la figure 2.13a.

On souhaite alors estimer.

$$logit(p(\mathrm{Y}=1|\mathrm{X})) = ln\left(\frac{p(\mathrm{Y}=1|\mathrm{X})}{p(\mathrm{Y}=0|\mathrm{X})}\right) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k \tag{2.13}$$

Il s'agit bien d'un régression puisque l'on cherche à montrer une dépendance entre la variable réponse et les variables explicatives. En modifiant simplement l'équation (2.13) on obtient la probabilité que l'évènement se produise sachant les variables explicatives :

$$p(\mathrm{Y}=1|\mathrm{X}) = \frac{e^{\beta_0+\beta_1 x_1+...+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+...+\beta_k x_k}} = \frac{1}{1+e^{-\beta_0-\beta_1 x_1-...-\beta_k x_k}} \tag{2.14}$$

Les paramètres $\beta_0, \beta_1, ..., \beta_k$ sont inconnus. Ils sont obtenus par maximisation de la vraisemblance. Prenons un échantillon $(\mathrm{X}_1, \mathrm{Y}_1), ..., (\mathrm{X}_n, \mathrm{Y}_n)$. Notons $p(\mathrm{X}_i) = \mathrm{P}(\mathrm{Y}_i = 1|\mathrm{X}_i)$ la probabilité que l'individu i présente l'évènement. Alors $\mathrm{Y}_i \mid \mathrm{X}_i$ suit une loi de Bernoulli de paramètre $p(\mathrm{X}_i)$. La

probabilité d'appartenance d'un individu i à un des deux groupes peut donc être décrite de la manière suivante

$$P(Y_i = y_i) = P(Y_i = 1|X_i)^{y_i} * [1 - P(Y_i = 1|X_i)]^{1-y_i} = p(X_i)^{y_i} * (1 - p(X_i))^{1-y_i} \tag{2.15}$$

La vraisemblance d'un échantillon s'écrit alors :

$$\begin{aligned} L_n(\beta; y) &= \prod_{i=1}^{n} p(X_i)^{y_i} * (1 - p(X_i))^{1-y_i} \\ &= \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-\beta^T X_i}}\right)^{y_i} * \left(1 - \frac{1}{1 + e^{-\beta^T X_i}}\right)^{1-y_i} \\ &= \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-\beta^T X_i}}\right)^{y_i} * \left(\frac{e^{-\beta^T X_i}}{1 + e^{-\beta^T X_i}}\right)^{1-y_i} \\ &= \prod_{i=1}^{n} \frac{e^{y_i \beta^T X_i}}{1 + e^{\beta^T X_i}} \end{aligned} \tag{2.16}$$

Les paramètres $\widehat{\beta} = \widehat{\beta_0}, \widehat{\beta_1}, ..., \widehat{\beta_k}$ qui maximisent cette quantité sont les estimateurs du maximum de vraisemblance de la régression logistique LEVERSHA [2003]. Souvent il est plus simple de maximiser le logarithme de la vraisemblance GREENE [2008] :

$$log(L_n(\beta; y)) = \sum_{i=1}^{n} y_i \beta^T X_i - log(1 + e^{\beta^T X_i}) \tag{2.17}$$

Dans la pratique l'équation 2.17 a rarement une solution explicite, mais $\widehat{\beta}$ peut être approché à l'aide d'algorithme d'optimisation tel que la méthode itérative du gradient de Newton Raphson [YPMA, 1995] ou plus récemment l'algorithme de Givens et Hoeting [GIVENS et HOETING, 2012]

La figure 2.13 montre la modélisation par la régression logistique sur deux jeux de données simulés. Sur le premier (2.13a), les données sont équilibrées (1 évènement pour 1 "non évènement") alors que sur la seconde (2.13b) les données sont déséquilibrées (1 évènement pour 9 "non évènements"). On observe bien dans la seconde situation que la pente de la fonction logistique est plus douce et que la probabilité qu'un évènement se produise est sous-évaluée. Pour pallier ces problèmes, King KING et ZENG [2002] préconise d'utiliser des méthodes de sampling de type cas-contrôle BRESLOW [1996],LANCASTER et IMBENS [1996] qui consiste à sélectionner une observation de la classe majoritaire pour chaque observation de la classe minoritaire, i.e. de l'undersampling. Plusieurs articles font référence à l'utilité de la régression logistique pour prédire les évènements rares [CALABRESE et OSMETTI, 2015],[VISA et RALESCU, 2005] [MAALOUF, 2011] et montrent son efficacité, notamment en utilisant les méthodes de sampling et/ou certaines corrections MAALOUF [2011]que nous allons présenter.

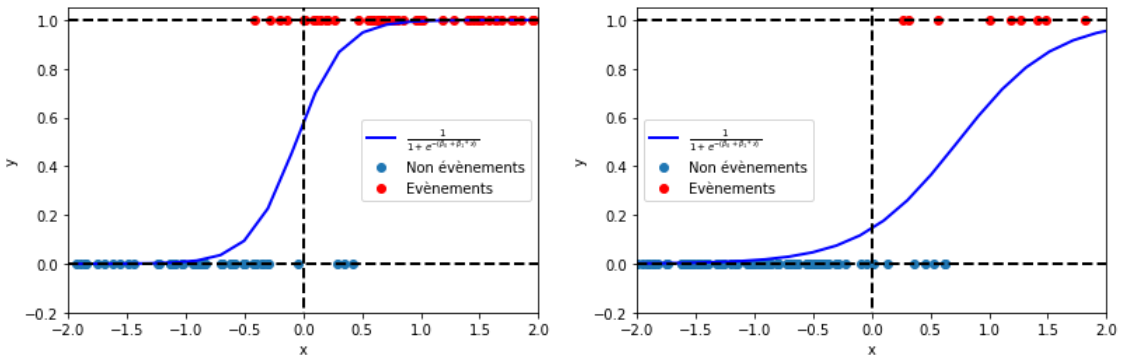
Prior correction

La "prior correction" a pour objectif de mieux rendre compte de la probabilité qu'une évènement se produise en corrigeant la constante β_0. Prenons une population de $\mathbb{N}$, avec τ la proportion d'évènements et $1 - \tau$ la proportion de non évènements. Posons également $\bar{y}$ la proportion d'évènements dans l'échantillon d'apprentissage, obtenu par exemple après une méthode d'échantillonnage et $1 - \bar{y}$ la proportion de non évènements, on peut alors corriger la constante :

$$\tilde{\beta}_0 = \widehat{\beta}_0 - \ln\left[\left(\frac{1-\tau}{\tau}\right)\left(\frac{\bar{y}}{1-\bar{y}}\right)\right] \tag{2.18}$$

Régression logistique pondérée

La prior correction corrige uniquement la constante β_0. Pour les autres paramètres, la proposition faite par G.King KING et ZENG [2002] est de pondérer différemment les classes afin d'augmenter l'importance de la classe minoritaire lors de l'apprentissage du modèle. Cela correspond



(a) Données équilibrées (1 :1)

(b) Données déséquilibrées (1:9)

FIGURE 2.13 – Régression logistique

à mettre un plus gros coût sur le mauvais classement d'une observation de la classe minoritaire. Prenons comme poids pour les observations de la classe minoritaire w_{min} et w_{maj} celui associé à la classe majoritaire. La log-vraisemblance s'écrit alors

$$log(\mathrm{L}_n(\beta;\mathbf{y})) = \sum_{i:y_i=1} w_{min} \ln\big(p(\mathrm{X}_i)\big) + \sum_{i:y_i=0} w_{maj} \ln\big(1-p(\mathrm{X}_i)\big) \tag{2.19}$$

Le choix des poids accordés à chaque classe reste à définir. Souvent, on essaiera d'équilibrer le poids des classes. Par exemple pour un jeu de données déséquilibré 1 : 25 (une observation de la classe minoritaire pour 25 de la classe majoritaire), on prendra $w_{min} = 25$ et $w_{maj} = 1$. Cette stratégie a montré des résultats supérieurs à la régression logistique simple dans le cas de données déséquilibrées MAALOUF et collab. [2017]; MAALOUF et SIDDIQI [2014].

Il est intéressant de noter le rapport direct entre la méthode de la régression logistique pondérée et l'effet des méthodes de sampling comme l'over et undersampling. En effet appliquer ces méthodes de sampling avec la régression correspond à faire exactement une régression logistique pondérée avec un poids $w_i \in \mathbb{R}\ (i = 1,..,.n)$ différent pour chaque observation. Pour le cas de l'oversampling, posons $\mathrm{X}_i^j\ (j = 1,...,\mathrm{J}_i)$, la $j$ème duplication de l'observation x_i et J_i le nombre de fois où x_i a été sélectionnée(par convention nous noterons x_i^0 l'observation originale). La log vraisemblance s'exprime alors :

$$\begin{aligned} log(\mathrm{L}_n(\beta;\mathbf{y})) &= \sum_{i:y_i=1} \sum_{j=1}^{\mathrm{J}_i} \ln\big(p(\mathrm{X}_i)\big) + \sum_{i:y_i=0} \sum_{j=1}^{\mathrm{J}_i} \ln\big(1-p(\mathrm{X}_i)\big) \\ &= \sum_{y_i=1} w_i \ln\big(p(\mathrm{X}_i)\big) + \sum_{y_i=0} w_i \ln\big(1-p(\mathrm{X}_i)\big) \end{aligned}$$

où w_i représente le nombre de duplication. On retrouve bien la log vraisemblance d'un modèle de régression logistique pondérée, où chaque observation a un poids w différent. On retrouve le même phénomène pour l'undersampling, les poids w_i prenant les valeurs 0 ou 1. La figure 2.14 montre le résultat de la régression lorsque les données sont pondérées de manière inversement proportionnelle à leur fréquence. Avec la pondération, les probabilités des évènements sont plus hautes, comme dans le cas des données équilibrées.

2.2.2 Régression de Poisson

La régression de poisson est un modèle de régression qui est surtout utilisé pour obtenir les probabilité qu'un évènement se produise k fois. La variable réponse Y est donc quantitative discrète. On rappelle qu'une variable aléatoire Y suit une loi de poisson POISSON [1887] de paramètre

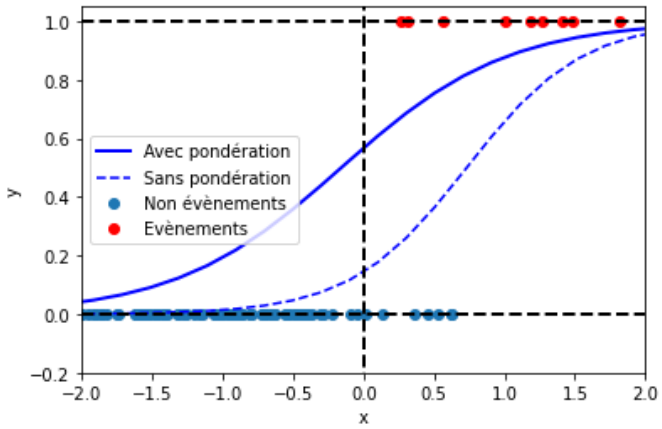


FIGURE 2.14 – Régression logistique pondérée

λ réel si et seulement si

$$P(Y = k) = e^{-\lambda}\frac{\lambda^k}{k!}, \text{ avec k un entier} \tag{2.20}$$

Nous avons alors :

$$E(Y) = V(Y) = \lambda. \tag{2.21}$$

Comme pour la régression logistique la régression de poisson ne cherche pas à estimer directement P(Y = 1|X) mais $log(E(Y|X))$. Nous avons donc

$$log(E(Y|X)) = \lambda = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k. \tag{2.22}$$

et finalement

$$\lambda = e^{\beta_0+\beta_1 x_1+...+\beta_k x_k}. \tag{2.23}$$

On peut, si on le souhaite, se ramener à une variable réponse dichotomique. En effet Y représente le nombre d'occurrence d'un évènement. Donc pour obtenir la probabilité qu'un évènement se produise, il suffit de regarder la probabilité d'avoir au moins une occurrence. Prenons Y une variable réponse binaire (0,1) et Z une variable réponse suivant une loi de poisson. On a alors : $P(Y = 1|x) = P(Z \geq 1|x) = 1 - P(Z = 0|x) = 1 - e^{-\lambda} = 1 - e^{-e^{\beta_0+\beta_1 x_1+...+\beta_k x_k}}$

Comme pour la régression logistique, l'estimation des $\widehat{\beta_0}, \widehat{\beta_1}, ..., \widehat{\beta_k}$ se fait par maximisation de la vraisemblance. En utilisant l'équation de la fonction de densité (2.20) d'une variable aléatoire suivant une loi de poisson, nous pouvons, pour n observations, calculer la vraisemblance donnée par :

$$\prod_i^n P(Y_i = k_i) = \prod_i^n e^{-\lambda_i}\frac{\lambda_i^{k_i}}{k_i!} \tag{2.24}$$

Avec $\lambda_i = e^{\beta_0+\beta x_i}$, $x_i \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ij} \end{pmatrix}$ et $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_j \end{pmatrix}$

Pour maximiser la vraisemblance, il est plus simple de maximiser le logarithme soit :

$$\sum_i^n [k_i ln(\lambda_i) - \lambda_i] - constante \tag{2.25}$$

Comme pour la régression logistique les paramètres qui maximisent cette vraisemblance peuvent être obtenus avec l'algorithme de Newton-Raphson.

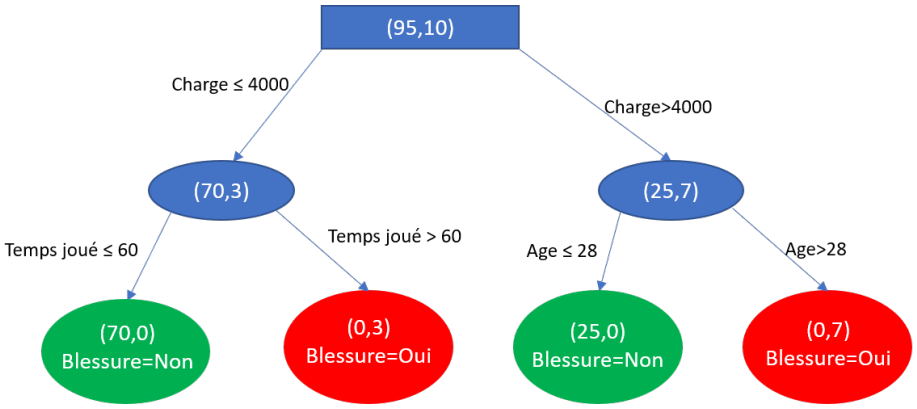


FIGURE 2.15 – Exemple d'arbre de classification

2.2.3 Arbre de classification

Un arbre de classification ([BREIMAN et collab., 1984], [DUDA et collab., 2001]) est un outil d'aide à la décision et à l'exploration de données. Il permet de modéliser simplement, graphiquement et rapidement un phénomène mesuré, plus ou moins complexe, grâce à la création de règles de décision. Sa lisibilité, sa rapidité d'exécution et le peu d'hypothèses nécessaires à priori, expliquent sa popularité actuelle. Un arbre de classification est composé de nœuds et d'arêtes. Le nœud d'origine de l'arbre est appelé "racine". Les nœuds intermédiaires sont appelés "feuilles internes". Enfin les derniers nœuds sont appelées "feuilles terminales". Ce sont, dans les feuilles terminales, que les résultats de la prédiction sont donnés. La figure 2.15 suivante présente un arbre de classification obtenu sur les données de blessure des joueurs (présenté dans la section). Il est composé d'un nœud d'origine, de 2 feuilles internes et de 4 feuilles terminales.

Principe de construction d'un arbre

Les algorithmes d'arbre de décisions utilisent le schéma suivant :

1. Sélection d'une variable d'entrée (parmi les variables candidates) suivant un certain critère de segmentation et de détermination des divisions (prémisses) correspondantes à cette variable d'entrée.
2. Si les critères d'arrêt ne sont pas vérifiés, on réitère l'étape 1 sur les divisions déterminées. L'algorithme se termine une fois les critères d'arrêt vérifiés.

Les algorithmes de création d'arbres de décision diffèrent par le critère de sélection des variables d'entrées , la méthode de division (ou segmentation) des variables et le critère d'arrêt.

Les critères de segmentation

Les critères de segmentation consistent à trouver la variable d'entrée explicative $X = (x_1, x_2, ..., x_k)$ qui va segmenter les données étudiées de façon à séparer au mieux la variable réponse Y qui est dans notre cas, $(Y \in \{0,1\})$. On recherche donc le meilleur classifieur à chaque étape. Il existe deux courants pour le faire :

- Le choix par maximisation des indicateurs de qualité d'une règle. Pour choisir la variable d'entrée de segmentation, les algorithmes s'appuient sur cette technique. Ils testent toutes les variables d'entrées potentielles et choisissent celle qui maximise ou minimise un critère donné tel que l'indice de Gini, cette indice appelé "impureté", est souvent utilisé. Il représente la vraisemblance qu'un élément du nœud soit mal classé par un tirage aléatoire selon la loi statistique de la cible estimée dans le nœud. Pour un nœud donné N, nous définissons p_j, $j = 0,1$ la probabilité qu'un élément du nœud appartienne à la classe j (i.e. $Y = j$),

$$p_j = \frac{\text{Nombre d'élements de la classe j}}{\text{Nombre d'élements du noeud}}$$

Nous pouvons alors définir l'indice de Gini par :

$$I_g(N) = \sum_{j=0}^{1} p_j(1-p_j) = \sum_{j=0}^{1} p_j - \sum_{j=0}^{1} p_j^2 = 1 - \sum_{j=0}^{1} p_j^2 \tag{2.26}$$

Si $I_g(n) = 0$ alors l'ensemble des éléments de N sont de la même classe. L'impureté du groupe est donc nul. Nous pouvons citer comme autres critères : le pourcentage d'une classe dans un nœud, le support absolu ou le pourcentage de points concernés par la règle.

- Le choix par utilisation de tests d'hypothèses : on choisit la variable qui a la plus grande influence sur la variable de sortie. Une manière de caractériser la segmentation est de mesurer le lien ou la causalité entre la variable candidate et la variable à prédire. On choisira la variable ayant le plus fort lien avec la variable à prédire. Dans ce cas, le critère le plus utilisé est le lien du Chi2 et ses dérivés.

Le traitement des variables continues doit être en accord avec l'utilisation du critère de segmentation. Dans la grande majorité des cas, le principe de segmentation des variables continues est très simple : trier les données selon la variable à traiter, tester tous les points de coupure possibles situés entre deux points successifs et évaluer la valeur du critère soit par test d'hypothèse (Anova ou Chi2) soit par maximisation des indicateurs de qualité dans chaque cas. Le point de coupure optimal correspond tout simplement à celui qui maximise le critère de segmentation.

Remarque : chaque modalité de la variable d'entrée permet de produire une prémisse. Les algorithmes peuvent différer sur ce point. Certains, tels que CART ([BREIMAN et collab., 1984] produisent systématiquement des arbres binaires. Ils cherchent donc lors de la segmentation, le regroupement binaire qui optimise le critère de segmentation. D'autres, tels que CHAID, [KASS, 1980] cherchent à effectuer les regroupements les plus pertinents en s'appuyant sur des critères statistiques.

Selon la technique, nous obtiendrons des arbres plus ou moins larges ou plus ou moins profonds. Il faudra cependant éviter de fractionner exagérément les données afin de ne pas produire des groupes d'effectifs trop faibles, ne correspondant à aucune réalité physique.

Les critères d'arrêt

Les objectifs des critères d'arrêt sont multiples :

- Économiser du temps de calcul en arrêtant le développement de certaines branches.
- Diminuer le nombre de règles produites.
- Augmenter les capacités de généralisation de l'arbre en ne faisant pas de sur-apprentissage.
- Maximiser la qualité et la performance de l'arbre.

L'objectif d'un arbre de décision est de produire des groupes généralisables les plus purs possibles. Il paraît donc naturel de fixer comme règle d'arrêt de construction de l'arbre, la constitution de groupes purs du point de vue de la variable à prédire. Souhaitable en théorie, cette attitude n'est pas tenable dans la pratique. En effet, nous travaillons souvent sur un échantillon que l'on espère représentatif d'une population. Tout l'enjeu est donc de trouver la bonne mesure entre capter l'information utile, correspondant réellement à une relation dans la population, et supprimer les spécificités du fichier sur lequel nous sommes en train de travailler (l'échantillon). Autrement dit, il faut éviter le sur-ajustement. En effet, dans un cas extrême, nous pouvons obtenir un arbre composé d'autant de règles que d'individus dans la base d'apprentissage. Il va donc falloir trouver l'arbre le plus petit possible, ayant la meilleure performance possible. Dans le cas des arbres de décision, plusieurs types de solutions algorithmiques ont été envisagées pour tenter d'éviter autant que possible un problème de sur-ajustement des modèles. Il s'agit des techniques dites de pré ou de post pruning des arbres. Le « pruning » a pour objectif d'éviter le sur-apprentissage et de simplifier les arbres de décision. Ainsi on va sélectionner des sous arbres qui ont certaines propriétés

et on espère, de la sorte, obtenir une meilleure capacité en généralisation tout en ayant moins de règles.

A partir d'un arbre donné, l'idée est de sélectionner, comme arbre final, un sous-arbre ayant certaines caractéristiques optimales.

La première stratégie, pour éviter un sur-ajustement massif des arbres de décision, consiste à proposer des critères d'arrêt lors de la phase d'expansion (réitération de la phase de segmentation). C'est le principe du pré-pruning. Nous considérons, par exemple, qu'une segmentation n'est plus nécessaire lorsque le support relatif ou absolu est trop faible ; et/ou encore, lorsque la pureté a atteint un niveau suffisant. Autre critère souvent rencontré dans ce cadre est l'utilisation d'un test statistique pour évaluer si la segmentation introduit un apport d'information significatif pour à la prédiction des valeurs de la variable à prédire.

Pruning (élagage)

Le pruning consiste à :

1. construire l'arbre en deux temps : produire l'arbre le plus pur possible dans une phase d'expansion en utilisant la base d'apprentissage,
2. effectuer une marche arrière pour réduire l'arbre. Certaines feuilles sont donc élaguées et ne serviront pas à la prédiction.

Cet élagage peut se faire sur la base de tests afin d'optimiser les performances de l'arbre.

Les algorithmes les plus couramment utilisés

Algorithme CHAID

CHAID ([KASS, 1980]) est une méthode d'arbre de décision reposant sur un critère de segmentation statistique : la mesure du Chi-2. Il possède deux particularités :

- Le critère d'arrêt permettant la détermination de la bonne taille de l'arbre s'effectue par pré-pruning. Lors de l'expansion de l'arbre, la décision de continuer la division dépend d'un test d'indépendance du chi-2 effectué sur le tableau de contingence associé aux prémisses qui seront produites par la segmentation suivante. Si ce test n'est pas significatif, les variables d'entrées ne seront pas segmentées.
- La méthode procède, éventuellement, à un regroupement des modalités de la variable de segmentation.

La méthode CHAID est particulièrement appropriée si le temps de calcul est un critère important pour l'utilisateur. Elle est indiquée lorsque l'on veut procéder à une première exploration des données.

C&RT

C&RT ([BREIMAN et collab., 1984]) (ou CART) propose une approche unifiée pour traiter les problèmes de discrimination (la variable à prédire est qualitative) à l'aide d'un arbre. Dans le cadre de la discrimination, le critère utilisé repose sur la notion de « pureté » en utilisant l'indice de Gini présenté plus tôt. Il est également possible de l'interpréter comme une analyse de variance sur données catégorielles. L'arbre, est dans un premier temps, complètement développé avec le critère de Gini sur une base d'apprentissage. Puis, dans un second temps, il est réduit de manière à optimiser le taux de mauvais classements calculé sur la base de test. Lors de cette seconde phase, il est possible d'introduire une matrice de coût de mauvais classements. Ce coût peut être utilisé pour améliorer la prédiction des évènements rares comme pour la régression logistique pondérée. C&RT possède deux particularités :

- Sur chaque variable d'entrée retenue, la méthode procède à un regroupement de manière à ce que l'arbre soit systématiquement binaire, c'est à dire chaque variable d'entrée segmentée ne possède que deux conditions.
- C&RT construit généralement des arbres très « compacts », ayant de bonnes capacités de prédiction.

Le calcul est coûteux en temps à cause du dispositif de pruning. De plus, cette méthode n'est pas très appropriée lorsque la taille de la base de données est faible.

2.2.4 Random Forest

La méthode des random forest BREIMAN [2001] CUTLER et collab. [2011] (ou forêts d'arbres décisionnels ou encore forêts aléatoires) est une méthode d'apprentissage automatique. Cette méthode consiste à construire plusieurs arbres (avec la méthode CART [BREIMAN et collab., 1984]) dit aléatoires. L'algorithme effectue un apprentissage sur plusieurs arbres de décision entrainés sur des sous-ensembles de données légèrement différents. Pour prédire la classe d'un individu, chaque arbre classe l'individu, puis un vote à partir de ces classements est exécuté pour savoir dans qu'elle classe sera l'individu. On compte le nombre de fois que l'individu a été classé dans une classe, puis il est classé dans la classe la plus représentée.

Construction de la forêt

Dans la méthode random forest, la construction des arbres qui vont constituer la forêt, se fait avec la méthode C&RT décrite précédemment (subsection 2.2.3).

Aléatoire

L'arbre est dit aléatoire du fait de sa construction particulière. On commence par construire plusieurs échantillons d'individus bootstrap $\theta_1, \theta_2, ..., \theta_q$ à partir des données initiales. Sur chaque échantillon θ_i un arbre de décision C&RT est construit en sélectionnant aléatoirement les variables utilisées FRIEDMAN et POPESCU [2003]. La construction des arbres est décrite ci-dessous :

- m variables sont tirées aléatoirement (sans remise).
- L'arbre est développé (arbre maximal) .
- L'arbre n'est pas élagué.

Avec cet aléa, on perturbe à la fois les individus et les variables. De plus cette méthode permet d'accélérer les calculs sur chaque arbre et de se passer de validation croisée qui est déjà «intégrée».

La forêt

Une fois les arbres construits, ils sont mis en relation pour former la forêt d'arbres aléatoires. A la fin de l'algorithme, le classement de futurs individus se fait par vote. Pour illustrer ce vote nous allons donner un exemple. Nous voulons savoir si un individu A est dans la catégorie "blessé" ou "non blessé". Cet individu, va alors parcourir chaque arbre de la forêt, ces arbres vont le classer "blessé" ou "non blessé". Une fois que l'individu a parcouru l'ensemble des arbres, il sera affecté dans la classe où il a été le plus souvent classé. Par exemple, si sur 100 arbres il a été classé 70 fois "blessé" et 30 fois "non blessé", il sera prédit "blessé". Nous pouvons même associer une probabilité à cette prédiction qui sera de 0.7. La figure suivante 2.16 illustre cette agrégation de vote.

Indice

La méthode random forest permet de calculer deux indices importants. Le premier concernant l'erreur de généralisation (Erreur out-of-bag ou OOB) et le second l'importance de chaque variable dans le modèle.

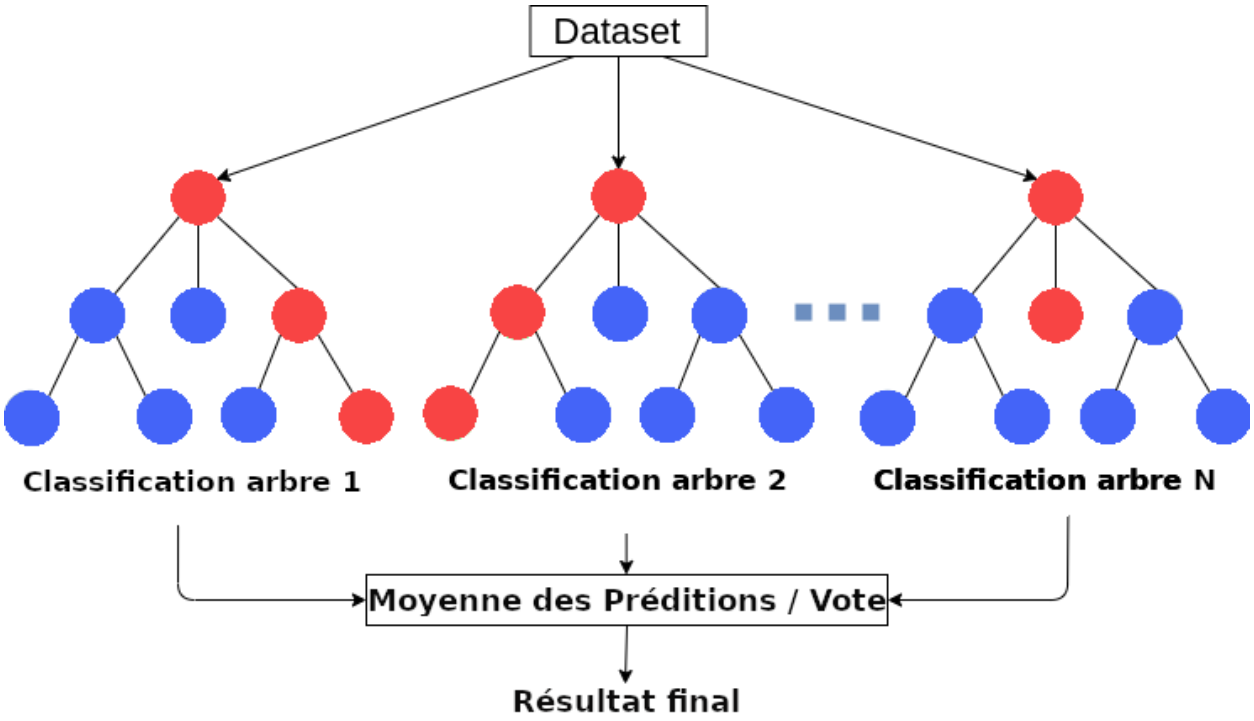


FIGURE 2.16 – Random Forest

Erreur Out-Of-Bag (OOB) qui signifie « en dehors du bootstrap » estime l'erreur de généralisation d'une forêt. Cette erreur est souvent considérée optimiste. Néanmoins, il existe une version corrigée de cette quantité introduite par [EFRON et TIBSHIRANI, 1993]. L'avantage de OOB est qu'elle ne nécessite pas le découpage de l'échantillon d'apprentissage puisque ce dernier est inclus dans les différents échantillons bootstrap. De plus, cette erreur impose les mêmes contraintes que les estimateurs classiques (échantillon test, validation croisée) au sens où les données prédites sont les données qui n'ont pas été rencontrées au préalable par le prédicteur.

Cette erreur est calculée de la manière suivante. Fixons une observation (X_i, Y_i) d'un individu i de l'échantillon d'apprentissage Ψ_n. Considérons alors uniquement les arbres construits sur les échantillons bootstrap ne contenant pas cette observation. On dit que cette observation est « out-of-bag ». Nous agrégeons alors la prédiction de ces arbres pour fabriquer notre prédiction $\widehat{Y_i}$ de Y_i.

Une fois cette opération effectuée pour l'ensemble des observations de Ψ_n, nous calculons l'erreur commise par nos prédictions, i.e. la proportion de mal classées en classification appelée erreur OOB :

$$\text{Erreur}_{\text{OOB}} = \frac{1}{n}\sum_{i=1}^{n} 1_{\widehat{Y}_i \neq Y_i} \tag{2.27}$$

Importance des variables est un indice très intéressant qui permet de mesurer l'impact de chaque variable dans la classification des individus. Elle se calcule de la manière suivante : fixons $j \in 1, ..., p$ et détaillons le calcul de l'indice pour la variable explicative X_j. On considère un échantillon bootstrap θ_l et l'échantillon OOB_l associé, c'est-à-dire l'ensemble des observations qui n'apparaissent pas dans θ_l. On calcul alors $\text{Erreur}_{\text{OOB}}$ sur OOB_l. Ensuite, nous permutons aléatoirement les valeurs de la j-ème variable dans l'échantillon OOB_l pour obtenir un échantillon perturbé, noté $\widetilde{\text{OOB}}_l^j$. Nous effectuons ces opérations sur tous les échantillons bootstrap. L'impor-

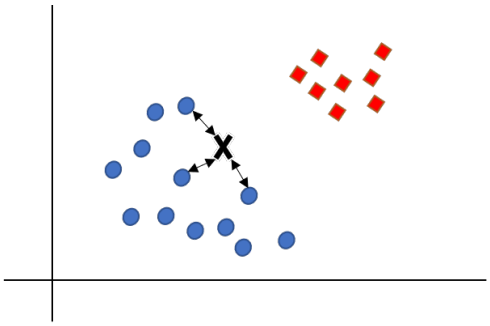


FIGURE 2.17 – Exemple d'une classification par 3-NN

tance de la variable X_j donnée par $VI(X_j)$ est définie par la différence entre l'erreur moyenne d'un arbre sur l'échantillon OOB perturbé ($\widetilde{OOB}_l^j$) et celle sur l'échantillon OOB :

$$VI(X_j) = \frac{1}{q}\sum_{i=1}^{q}(\text{Erreur}_{\widetilde{OOB}_i} - \text{Erreur}_{OOB_i}) \tag{2.28}$$

Ainsi, plus la permutation aléatoire de la j-ème variable engendre une forte augmentation de l'erreur, plus la variable est importante. Au contraire si les permutations n'ont quasiment aucun effet sur l'erreur, la variable est considérée comme peu importante.

2.2.5 K plus proches voisins (knn)

La technique des k-plus proches [FIX et HODGES, 1989] voisins ou k-NN pour k Nearest Neighbors en anglais est un algorithme qui peut servir autant pour la classification que la régression. Il est surnommé « nearest neighbors » (plus proches voisins en français). En effet, le principe de ce modèle consiste à choisir les k données les plus proches du point étudié afin d'en prédire sa valeur en considérant la classe majoritaire de ses voisins. C'est une méthode intuitive qui utilise la proximité entre les observations. Supposons, comme dans les méthodes précédentes Y une variable aléatoire binaire (0 ou 1) et X les variables explicatives. $d(i,j)$ est la distance entre les observations i et j. Cela implique donc que l'espace soit métrique. La distance la plus couramment utilisée est la distance euclidienne. Cependant, il est possible d'utiliser d'autres distances comme le montre le tableau ci-dessous. Soient deux points $X = (x_1, ..., x_n)$ et $Y = (y_1, ..., y_n)$.

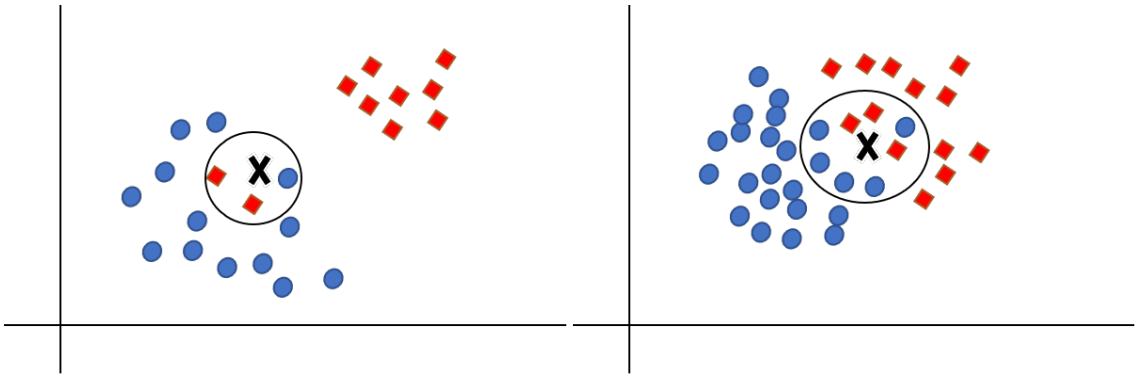
nom	Paramètre	fonction
distance de Manhattan	1-distance	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$
distance euclidienne	2-distance	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
distance de Minkowski	p-distance	$\sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}$
distance de Tchebychev	∞-distance	$\lim_{p\to+\infty} \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p} = \sup_{1\le i\le n} \lvert x_i - y_i \rvert$

On définit les k plus proches voisins de l'observation x_i sur l'espace E par l'ensemble K_i des observations $x_{i1}, ...x_{ik}$ tel que $\forall x_{ij} \in K_i, \forall x_t \in E \setminus \{K_i, x_i\}$ on a $d(x_i, x_{ij}) < d(x_i, x_t)$. Notons Y_i (respectivement Y_{ij}), la valeur de la variable Y pour l'observation x_i (respectivement x_{ij}), alors la probabilité que $Y_i = 1$ est donnée par

$$P(Y_i = 1) = \frac{|x_{ij} \in K_i, Y_{ij} = 1|}{k} \tag{2.29}$$

Un exemple de classification par 3-NN est donné en figure 2.17

Le seul paramètre de cette méthode est le nombre de voisins sélectionnés k. A priori, le choix est compliqué. Prendre un k grand permettra de diminuer l'effet des bruits, mais nécessite une plus grande base d'apprentissage. Un k plus petit rendra mieux compte des structures fines. Dans le cas de données déséquilibrées le choix du nombre de voisins est encore plus compliqué. S'il est pris trop grand la classification sera mauvaise en partie à cause du déséquilibre des données. La figure 2.18 présente la problématique du bon choix de k.



(a) Classe par 3NN, k trop petit

(b) Classe par 8NN, k trop grand

FIGURE 2.18 – Importance du nombre de voisins pour la classification avec les knn

Sur la figure 2.18a, on voit que l'observation classifiée par 3-nn va être classée dans la classe rouge alors qu'elle devrait être dans la classe bleue. En effet si l'on regarde les 3 plus proches voisins, il y a 2 rouges contre 1 bleu. La modélisation est trop fidèle au maillage fin et fait des erreurs. Alors que choisir un k plus grand (≥ 5) aurait corrigé ce problème. Sur la seconde figure 2.18b, le point appartenant à la classe rouge a été classé comme bleu, puisque parmi ses 8 plus proches voisins 5 sont bleus. Ici un k plus petit aurait évité l'erreur. La recherche du nombre k de voisins est donc très important. Une solution pour trouver k est d'utiliser une méthode de validation croisée :

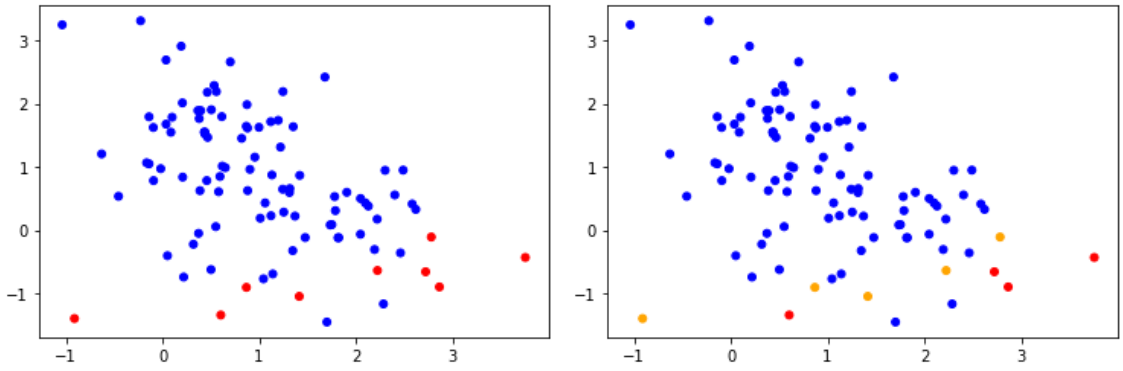
— Le jeu de données initial est séparé en 3 partie : un jeu d'entrainement, un jeu de validation et un jeu de test

— En faisant varier k le nombre de voisins, plusieurs modèles de k-nn sont construits en utilisant le jeu d'entrainement.

— Les modèles sont comparés entre eux à l'aide d'indicateurs, comme par exemple le nombre d'erreurs commises. On sélectionne alors la valeur de k associée au meilleur modèle.

Dans le cas de jeu de données déséquilibrées le choix du nombre de plus proches voisins est encore plus important. En effet, les observations de la classe majoritaire tendent à dominer la classification d'une observation, puisque la classe est statistiquement plus fréquente. Même si les classes sont séparables, il n'est pas rare que les observations de la classe minoritaire en bordure de classe aient plus d'observations de la classe majoritaire en plus proches voisins que de leur propre classe et seront donc mal classées. La figure 2.19 illustre ce phénomène. La seconde figure 2.19b représente les prédictions faites en utilisant les 2 plus proches voisis. Les points bleus sont les observations correctement classées de la classe majoritaire, en rouge celles de la classe minoritaire et en orange les observations de la classe minoritaire mal classées. Du fait du déséquilibrage les observations de la classe minoritaire en bordure de classe sont mal classées. En augmentant le nombre de voisins k, la classe minoritaire serait de moins en moins bien prédite.

Dans ce cas, il existe plusieurs solutions. La première consiste à pondérer la classification par l'inverse de la distance entre les points à classer et les plus proches voisins. Ainsi les points rapprochés de l'observation à classer ont une importance plus grande que ceux éloignés. Cependant, cette méthode n'est pas toujours efficace. Lorsque une classe (ou les deux) est trop dense, alors tous les k plus proches voisins sont proches de l'observation test. La différence de distance n'est alors plus discriminante YANG et collab. [2010].



(a) Données initiales déséquilibrées

(b) Prédiction par la méthode 2-NN

FIGURE 2.19 – Illustration de l'effet des données déséquilibrées sur 2-NN

2.2.6 Support Vector Machine (SVM)

Les Support Vector Machine, machines à vecteurs de support ou séparateurs à vaste marge en français, sont un ensemble de techniques d'apprentissage supervisées destinées à résoudre des problèmes de discrimination et de régression. Elles ont été introduites en 1963 par [VAPNIK et LERNER, 1963]. Dans le cas de la discrimination d'une variable dichotomique, elles sont basées sur la recherche de l'hyperplan de marge optimale qui, lorsque c'est possible, classe ou sépare correctement les données tout en étant le plus éloigné possible de toutes les observations.

Le principe de base des SVM consiste à ramener le problème de la discrimination à celui, linéaire, de la recherche d'un hyperplan optimal. Deux idées ou astuces permettent d'atteindre cet objectif.

- La première consiste à définir l'hyperplan comme solution d'un problème d'optimisation sous contraintes dont la fonction objectif ne s'exprime qu'à l'aide de produits scalaires entre vecteurs et dans lequel le nombre de contraintes "actives" ou vecteurs supports contrôle la complexité du modèle.
- Le passage à la recherche de surfaces séparatrices non linéaires est obtenu par l'introduction d'une fonction noyau (kernel) dans le produit scalaire induisant implicitement une transformation non linéaire des données vers un espace intermédiaire (feature space) de plus grande dimension. D'où l'appellation, couramment rencontrée, de "machine à noyau" ou "kernel machine". Sur le plan théorique, la fonction noyau définit un espace hilbertien, dit auto-reproduisant, et isométrique par la transformation non linéaire de l'espace initial et dans lequel est résolu le problème linéaire.

On distinguera le cas des classes linéairement séparables et non linéairement séparables.

Cas de classes linéairement séparables

On parle de classes linéairement séparables s'il existe un hyperplan permettant de séparer les deux classes. L'ajustement d'un SVM est la recherche de l'hyperplan, le plus éloigné possible des deux ensembles. La marge, distance entre l'hyperplan et les deux classes, permet de mesurer cette caractéristique. Elle garantit l'optimalité du SVM.

Équation du séparateur linéaire et calcul de la marge

Notons $\vec{x}_i$ les valeurs des variables d'entrées pour un individu i et y_i la variable réponse associée (pouvant prendre les valeurs -1 ou 1). Il existe un hyperplan h(x) séparateur des deux classes donné par l'équation :

$$\langle \vec{w}, \vec{x}_i \rangle + w_0 = 0 \tag{2.30}$$

avec $\langle .,. \rangle$ le produit scalaire, $\vec{w}$ un vecteur orthogonal à l'hyperplan et w_0 une constante comme illustrée sur la figure 2.20. Il est possible de montrer que la marge est égale à : $\frac{1}{\|\vec{w}\|}$

Cette hyperplan permet de classer les observations :

$$\begin{cases} h(x) \geq 0 \Longrightarrow \hat{y} = +1 \\ h(x) < 0 \Longrightarrow \hat{y} = -1 \end{cases}$$

Autrement dit, pour chaque individu $k \in \{1, \ldots, p\}$ si le SVM est correctement entrainé, on a

$$y_k h(x_k) = y_k(\langle \vec{w}, \overrightarrow{x_k} \rangle + b)) \geq 0 \qquad \forall k \in \{1, \ldots, p\} \tag{2.31}$$

Solution optimale dans le cas linéairement séparable

Il existe plusieurs plans séparateurs (figure 2.20). Celui qui sépare le mieux les données, est appelé hyperplan optimal. Il est celui qui maximise la marge en respectant les contraintes de séparation des classes, c'est-à-dire $y_k(\langle \vec{w}, \overrightarrow{x_k} \rangle + b) \geq 0$. La distance d'un point $x_k \in \mathbb{R}^n$ à l'hyperplan de vecteur support w et de biais w_0 est donnée par :

$$\frac{y_k \left(w^\top \cdot x_k + w_0 \right)}{\| w \|} \tag{2.32}$$

Puisque l'on cherche l'hyperplan qui maximise la marge, on cherche l'unique hyperplan dont les paramètres (w, w_0) sont donnés par la formule :

$$\arg\max_{w,b} \min_k \frac{y_k \left(w^\top \cdot x_k + w_0 \right)}{\| w \|} \tag{2.33}$$

Pour simplifier les calculs on choisit de normaliser les vecteurs w et w_0 tels que les vecteurs supports x_s vérifient $y_s \left(w^\top \cdot x_s + w_0 \right) = 1$. Le problème d'optimisation 2.33 devient alors $\arg\max_{w,w_0} \frac{1}{\|w\|}$ sous la contrainte $\forall k, y_k \left(w^\top \cdot x_k + w_0 \right) \geq 1$ Le problème se reformule pour des raisons pratiques de calculs :

$$\text{Minimiser } \frac{1}{2} \| w \|^2 \text{ sous les contraintes } y_k \left(w^\mathrm{T} x_k + w_0 \right) \geq 1 \tag{2.34}$$

C'est donc un problème d'optimisation non linéaire avec contraintes. Il peut se résoudre par une méthode des multiplicateurs de Lagrange WILLIAM P. [1989] ou le lagrangien, à minimiser par rapport à w et w_0, et maximiser par rapport à α, est donné par :

$$\mathrm{L}(w, w_0, \alpha) = \frac{1}{2} \| w \|^2 - \sum_{k=1}^{p} \alpha_k \left\{ y_k \left(w^\mathrm{T} x_k + w_0 \right) - 1 \right\} \tag{2.35}$$

Puis, en utilisant les dérivés partielles selon les conditions de Kuhn-TuckerKUHN et TUCKER [1951], on obtient :

$$\begin{cases} \sum_{k=1}^{p} \alpha_k y_k x_k & = w^* \\ \sum_{k=1}^{p} \alpha_k y_k & = 0 \end{cases} \tag{2.36}$$

Ce qui, une fois réinjecté dans l'équation 2.35, donne la formulation duale :

$$\text{Maximiser } \tilde{\mathrm{L}}(\alpha) = \sum_{k=1}^{p} \alpha_k - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^\mathrm{T} x_j \qquad \text{sous les contraintes } \alpha_k \geq 0, \text{ et } \sum_{k=1}^{p} \alpha_k y_k = 0 \tag{2.37}$$

Cela nous permet d'obtenir les multiplicateurs de Lagrange optimaux α_k^* et l'hyperplan en remplaçant w par sa valeur optimale $w*$ dans l'équation de $h(x)$

$$h(x) = \sum_{k=1}^{p} \alpha_k^* y_k (x \cdot x_k) + w_0 \tag{2.38}$$

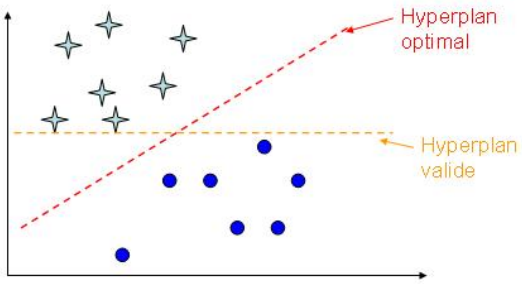


FIGURE 2.20 – Plans séparateurs

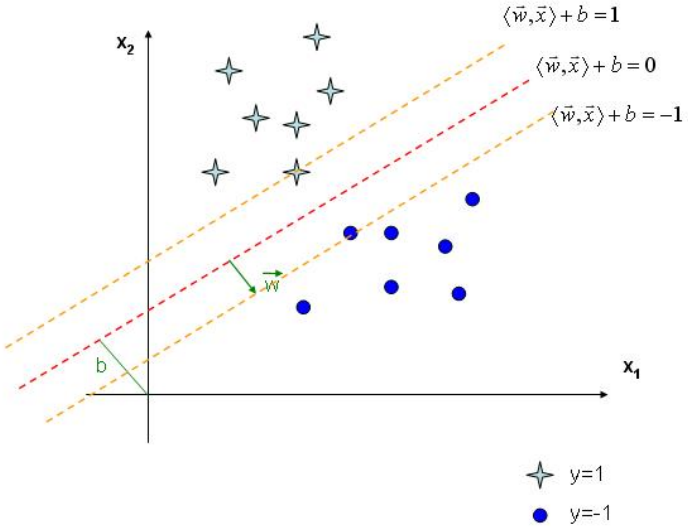


FIGURE 2.21 – Hyperplan optimal et marge

Un point à noter est que l'une des conditions de Kuhn-Tucker donne :

$$\alpha_k \left[y_k h(x_k) - 1 \right] = 0 \quad 1 \leq k \leq p \tag{2.39}$$

Cela implique que les seuls points où les contraintes du lagrangien sont actives sont les points tels que $y_k h(x_k) = 1$. Ce sont ceux sur la marge maximale, à la "frontière" de l'hyperplan. Ces points sont appelés "vecteurs supports". Ce sont les seuls qui participent à la définition de l'hyperplan optimal. La figure 2.21 présente l'hyperplan optimal (en rouge), la marge (en orange) et les deux vecteurs supports (en vert).

Enfin la classification d'un individu k se fait par la fonction de décision :

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \mathbf{x}_i \cdot \mathbf{x} - b \right) \tag{2.40}$$

Cas de classes non linéairement séparables

Dans le cas où les classes ne sont pas linéairement séparables, la notion de marge maximale et la recherche de l'hyperplan séparateur optimal décrit précédemment ne fonctionne pas. Il est alors nécessaire de passer par d'autres méthodes et notamment la méthode de l'astuce des noyaux (en anglais Kernel trick). Cette dernière consiste à transformer les données et reconsidérer le problème dans un espace de dimension supérieur (qui peut être de dimension infini) dans lequel une séparation linéaire existe (figure 2.22).

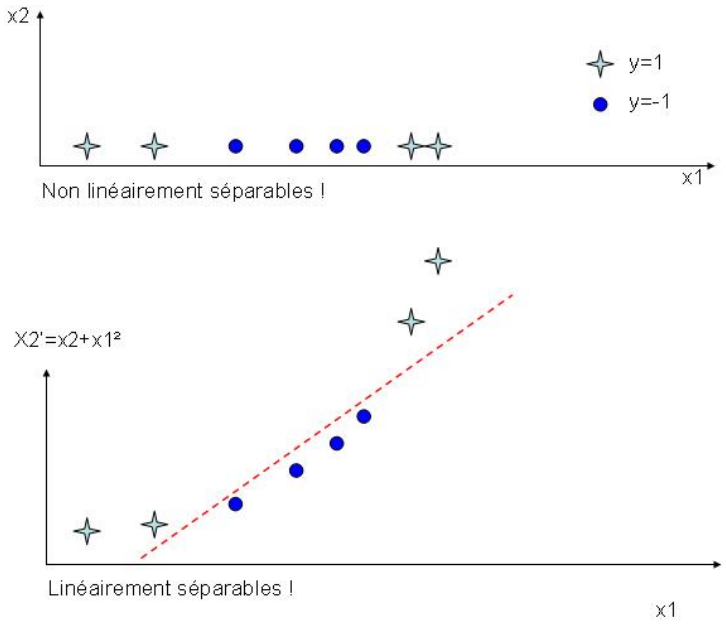


FIGURE 2.22 – Classe non séparable et utilisation d'un noyaux

Transformation des données

En conservant les notations précédentes (x_i les variables explicatives, y_i la variables réponse de l'individu i), on note ϕ la fonction de transformation non linéaire appliquée à x_i. L'espace d'arrivée $\phi(X)$ est appelé espace de description. Dans ce nouvel espace, on cherche l'hyperplan d'équation :

$$h(x) = w^{\mathrm{T}}\phi(\mathrm{X}) + w_0 = 0 \tag{2.41}$$

vérifiant $y_k * h(x_k) > 0$ pour tous les pointsx_k de l'ensemble d'apprentissage, c'est-à-dire l'hyperplan séparateur dans l'espace de redescription.

En utilisant la même procédure que dans le cas linéairement séparable, on se retrouve avec le problème d'optimisation suivant :

$$\text{Maximiser } \mathrm{L}(\alpha) = \sum_{k=1}^{p} \alpha_k - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i)^{\mathrm{T}}\phi(x_j) \tag{2.42}$$

sous les contraintes : $\alpha_i > 0$ et $\sum_{k=1}^{p} \alpha_k y_k = 0$

Le problème d'optimisation implique alors un produit scalaire des vecteurs de l'espace de redescription qui peut être de dimension élevée, donc couteux en temps. Pour pallier ce problème l'astuce utilisée, appelée Kernel trick, consiste à appliquer des fonctions noyaux qui vérifient

$$\mathrm{K}(x_i, x_j) = \phi(x_i)^{\mathrm{T}}\phi(x_j) \tag{2.43}$$

L'utilisation de 2.43 permet de faire les calculs dans l'espace d'origine, moins couteux que dans l'espace de redescription. De plus la transformation $\phi(x_i)$ n'a pas besoin d'être connue puisque seule la fonction noyau intervient dans le calcul.

Choix de la fonction noyau

Dans la pratique il n'existe pas de méthode permettant de choisir le bon noyau à utiliser. Le choix est donc laissé à l'expérience de l'utilisateur. Il existe une multitude de noyaux. Nous présenterons les plus utilisés :

— **Le noyaux linéaire**

$$k(x,y) = x^{\mathrm{T}} y + c \tag{2.44}$$

Il permet de se ramener au cas d'un classifieur linéaire, et donc de généraliser l'approche linéaire.

— **Le noyau polynomial**

$$k(x,y) = (\alpha x^{\mathrm{T}} y + c)^d \tag{2.45}$$

Dans ce noyau 3 paramètres interviennent : la constante c, le degré d du polynôme et la pente α. Le plus souvent on l'utilise avec $\alpha = c = 1$. Il est particulièrement efficace lorsque les données sont normalisées

— **Le noyau RBF (Radial Basis Function)**

$$k(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma}\right) \tag{2.46}$$

Pour ce noyau un seul paramètre est à calibrer, mais il joue un rôle majeur dans l'efficacité du classifieur. S'il est surestimé le noyau aura un comportement presque linéaire et la projection dans des espaces de dimensions supérieures perdra sa puissance non linéaire. S'il est sous estimé le noyau manquera de stabilité et sera plus sensible aux bruits et aux artefacts

— **Le noyau Sigmoïde**

$$k(x,y) = \tanh(\alpha x^{\mathrm{T}} y + c) \tag{2.47}$$

Ces noyaux permettent de répondre presque toujours aux problématiques des jeux de données même si le choix et la calibration des paramètres peut être laborieux. S'ils ne suffisent pas, il est possible d'en construire d'autres par combinaison de noyaux existants. Prenons deux noyaux k_1 et k_2 sur $\mathscr{X}^2$ ($\mathscr{X} \subset \mathbb{R}^n$) une fonction $f : \mathscr{X} \to \mathbb{R}$, une fonction $\Phi : \mathscr{X} \to \mathbb{R}^{\mathrm{N}}$, un noyau $k_3 : \mathbb{R}^{\mathrm{N}} \times \mathbb{R}^{\mathrm{N}} \to \mathbb{R}$, $a \in \mathbb{R}^+$ et B une matrice N × N symétrique semi-définie positive, les fonctions suivantes sont des noyaux [SHAWE-TAYLOR et CRISTIANINI, 2004] :

1. $k(\mathbf{x},\mathbf{y}) = k_1(\mathbf{x},\mathbf{y}) + k_2(\mathbf{x},\mathbf{y})$
2. $k(\mathbf{x},\mathbf{y}) = k_1(\mathbf{x},\mathbf{y})\, k_2(\mathbf{x},\mathbf{y})$
3. $k(\mathbf{x},\mathbf{y}) = a k_1(\mathbf{x},\mathbf{y})$
4. $k(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) f(\mathbf{y})$
5. $k(\mathbf{x},\mathbf{y}) = k_3(\Phi(\mathbf{x}),\Phi(\mathbf{y}))$
6. $k(\mathbf{x},\mathbf{y}) = \mathbf{x}\mathrm{B}\mathbf{y}^{\mathrm{T}}$

Marge souple

Il arrive souvent qu'il soit impossible de trouver l'hyperplan séparateur optimal. Dans ce cas, il est nécessaire d'utiliser la méthode de la marge souple CORTES et VAPNIK [2009]. Cette technique consiste à chercher un hyperplan séparateur qui minimise le nombre d'erreurs de classement en introduisant le ressort ξ_k (slack variables en anglais) :

$$l_k\left(w^{\mathrm{T}} x_k + w_0\right) \geq 1 - \xi_k \quad \xi_k \geq 0, \quad 1 \leq k \leq p \tag{2.48}$$

Le problème d'optimisation est alors modifié par un terme C, pénalisant les variables de ressorts élevés :

$$\text{Minimiser } \frac{1}{2}\|w\|^2 + \mathrm{C}\sum_{k=1}^{p} \xi_k \quad , \quad \mathrm{C} > 0 \tag{2.49}$$

Le paramètre C est un paramètre à définir et doit être optimisé à l'aide de la validation croisée par exemple. Il représente la balance entre le nombre d'erreurs de classification et la taille de la marge. La figure 2.23 schématise la marge souple.

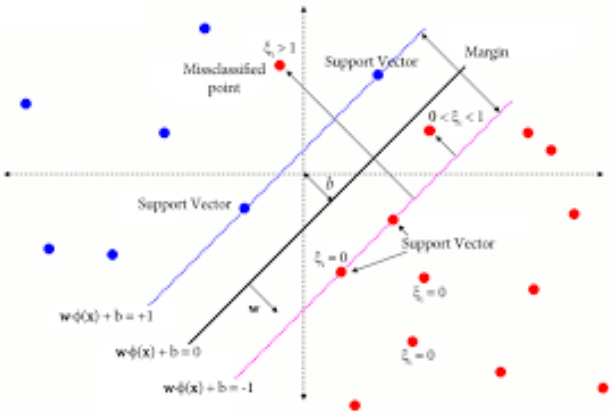


FIGURE 2.23 – Marge souple

Probabilité avec les SVM

La fonction principale des SVM est la classification. Cependant il est possible d'y associer des probabilités. La méthode a été proposée par Platt en 1999 PLATT [1999]. Elle repose sur la valeur de l'hyperplan h(x) obtenue avec la méthode de classification originale. La probabilité proposée est :

$$\mathrm{P}(\mathrm{Y}=1 \mid h) = \frac{1}{1+\exp(\mathrm{A}h+\mathrm{B})}, \tag{2.50}$$

où A et B sont deux paramètres estimés en utilisant le maximum de vraisemblance de la base d'apprentissage. Posons $t_i = \frac{y_i+1}{2}$, on obtient A et B en minimisant la log vraisemblance, avec une fonction d'erreur cross-entropie :

$$\min -\sum_i t_i \log\left(p_i\right) + (1-t_i)\log\left(1-p_i\right), \tag{2.51}$$

avec

$$p_i = \frac{1}{1+\exp\left(\mathrm{A}f_i+\mathrm{B}\right)} \text{ et } f_i(x) = h(x_i)+b, \tag{2.52}$$

Les probabilités nous permettent notamment de trouver la probabilité discriminant le mieux les évènement des non évènements. Elles nous permettent également de comparer les méthodes entre elles.

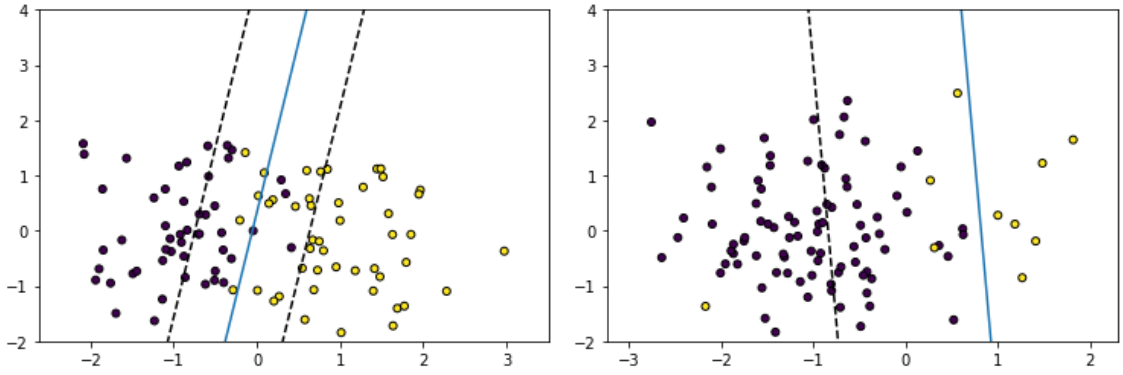
Données déséquilibrées

Lorsque les jeux de données sont déséquilibrés, l'astuce des noyaux ne fonctionne pas toujours. Il arrive alors que la modélisation ne soit pas performante ou que la probabilité des évènements rares soit sous évaluée. La figure 2.24 présente la séparation construite à l'aide de l'hyperplan optimal (en bleu) avec la marge maximale (en noir). La première figure (2.24a) présente des résultats pour des données équilibrées et la seconde (2.24b)pour des données déséquilibrées. L'hyperplan optimal obtenu dans le cas des données déséquilibrées est trop proche de la classe minoritaire. De plus les marges sont bien plus éloignées de l'hyperplan que dans le jeu de données balancées.

Afin d'améliorer le modèle plusieurs solutions ont été proposées :

— Appliquer une méthode d'échantillonnage telle que random oversampling RAMIREZ et ALLENDE [2012], VEROPOULOS et collab. [1999] ou SMOTE KÖKNAR-TEZEL et LATECKI [2009] ou utiliser des méthodes d'agrégation de modèles WU et collab. [2003].

— Corriger la constante b de l'hyperplan [NUÑEZ et collab., 2017] en utilisant deux biais différents pour améliorer la modélisation de survenance d'un événement rare.



(a) Données équilibrées (1:1)

(b) Données déséquilibrées (1:9)

FIGURE 2.24 – Effet des données déséquilibrées sur la classification avec les SVM

Prenons $\mathcal{Z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ un échantillon de données avec $\mathbf{x}_i \subseteq \Re^m$ et $y_i \in \{+1, -1\}$. Enfin prenons $\mathcal{Z}_1$ (resp. $\mathcal{Z}_2$) l'échantillon de données des cas positifs (+) (respectivement des cas négatifs (-)). Le biais standard des SVM peut être obtenu par,

$$b_s = \frac{\alpha + \beta}{2} \tag{2.53}$$

avec α la valeur maximale de l'hyperplan sans biais appliqué aux observations de la classe négative et β la valeur minimale de l'hyperplan sans biais appliqué aux observations de la classe positive, i.e. :

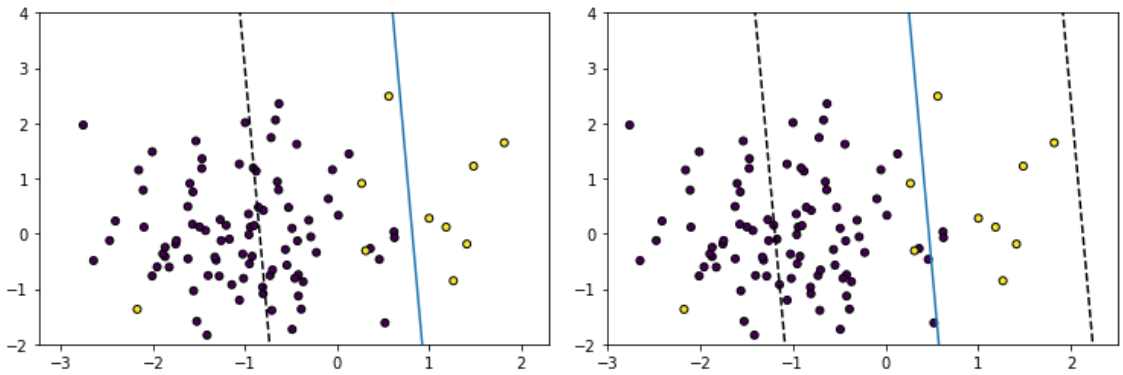
$$\alpha = \max_{\mathbf{x}_k \in \mathcal{Z}_2} \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}_k), \quad \text{et} \quad \beta = \min_{\mathbf{x}_k \in \mathcal{Z}_1} \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}_k) \tag{2.54}$$

Le biais influe directement sur la position de l'hyperplan. Si le biais est égal à β, alors toutes les observations positives seront classées correctement. C'est d'ailleurs la plus petite valeur qui assure que 100% des observations positives sont correctement classées ABRIL et collab. [2014].

La première correction du biais tient compte du nombre d'observations dans chaque classe (N_1 pour la classe positive et N_2 pour la classe négative). Le nouveau biais proportionnel peut alors se définir par :

$$b_p = \frac{N_1\alpha + N_2\beta}{N_1 + N_2} \tag{2.55}$$

Dans les cas des évènements rares ce biais va déplacer l'hyperplan en direction de la classe majoritaire permettant l'amélioration des performances prédictives sur la classe minoritaire, comme nous pouvons le voir sur la figure 2.25.



(a) Sans correction

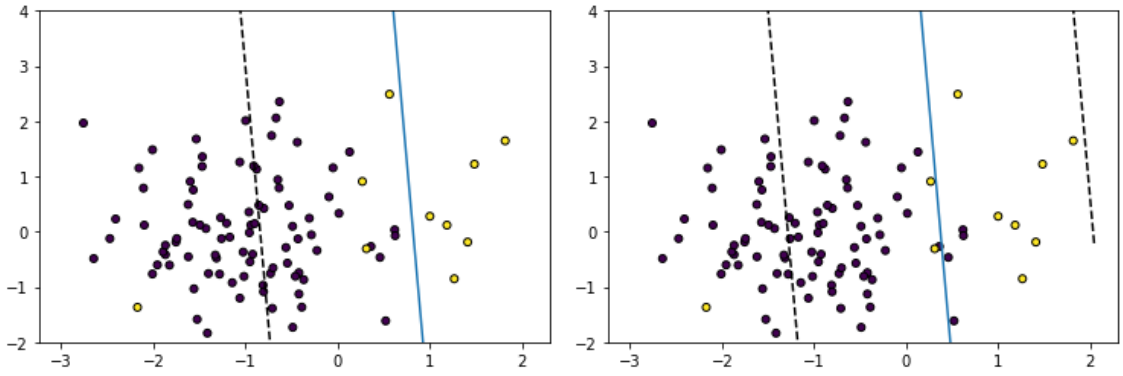
(b) Avec application de b_{p1}

FIGURE 2.25 – Effet de la première correction du biais : b_p

Le second biais proposé se base sur les vecteurs supports , qui sont les observations les plus informatives de chaque classe pour la génération de l'hyperplan optimal. Soit N_{Sv1} (resp. N_{Sv2}) le nombre de vecteurs supports de la classe positive (resp. négative,) le second biais est défini par :

$$b_{p1} = \frac{N_{sv1}\alpha + N_{sv2}\beta}{N_{sv1} + N_{sv2}} \tag{2.56}$$

Le résultat obtenu en utilisant ce biais est présenté dans la figure 2.26



(a) Sans correction

(b) Avec application de la correction b_{p1}

FIGURE 2.26 – Effet de la seconde correction du biais : b_{p_1}

Enfin, il est possible de modifier la marge souple en pondérant les classes. Dans l'équation 2.48, C est divisé en deux constantes, C^+ et C− qui représentent le coût de mauvais classement de la classe positive et de la classe négative. Cela permet de donner plus d'importance à la classe minoritaire, en prenant $C^+ \gg C^-$. En effet cela donne d'avantage d'importance aux mauvais classement de la classe minoritaire. La figure 2.27, montre l'action de cette pondération lorsque le poids est égal à l'inverse de la fréquence de la classe.

2.2.7 Réseau de neurones artificiels : perceptron multicouche

Les réseaux de neurones artificiels sont des modèles mathématiques inspirés du fonctionnement des neurones biologiques. Il existe une multitude de réseaux de neurones (par exemple les réseaux de Kohonen KOHONEN [2001] ou les réseaux RBF (Radial Basis Function) BROOMHEAD et LOWE [1988]), spécialisés ou non dans des domaines variés (reconnaissance de caractères, etc.).

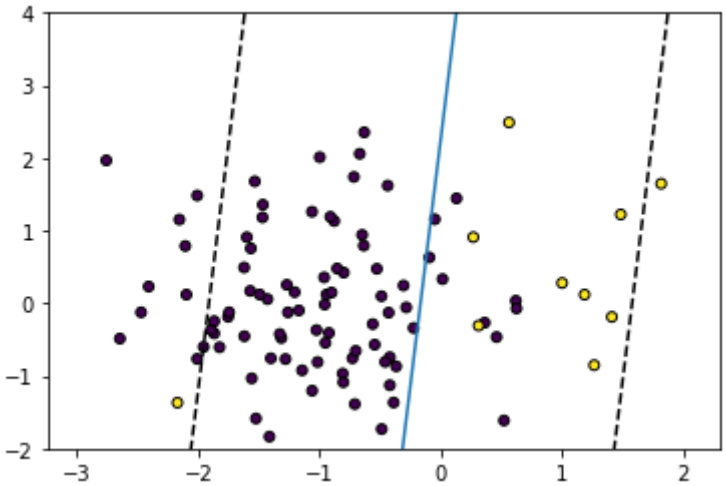


FIGURE 2.27 – Effet de la pondération

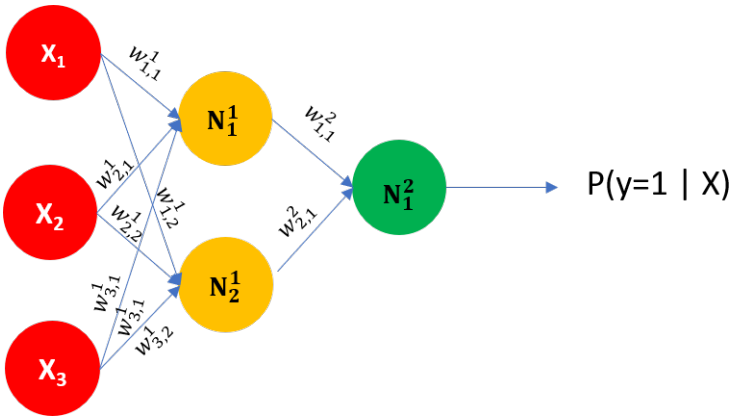


FIGURE 2.28 – Réseau de neurones à deux couches

Nous nous intéresserons à la famille des perceptrons multicouches RUMELHART et collab. [1986] pour la prédiction d'événements dans un cadre d'apprentissage supervisé. Le domaine des réseaux de neurones est vaste. Les lecteurs souhaitant avoir une vision globale peuvent s'intéresser aux ouvrages de HAYKIN [1999] et BISHOP [2006]

Les réseaux de neurones sont constitués de plusieurs couches de neurones. C'est dans les neurones que les transformations mathématiques (linéaire ou non) sont effectuées. Chaque neurone d'une couche est relié à tous les autres neurones des couches adjacentes par des synapses. Chaque synapse a un "poids" différent qui définit l'importance de la liaison entre deux neurones. La figure 2.28 présente un réseau de neurones à deux couches. Il est formé d'une couche d'entrée constituée de trois neurones, d'une couche cachée (hidden layer) et d'une couche de sortie (output layer). La couche cachée est constituée de deux neurones, la dernière couche, la couche de sortie d'un seul.

Sur l'illustration :

- en rouge, la couche d'entrée (Input layer) représente les variables explicative $X = (X_1, X_2, X_3)$ servant à l'apprentissage du modèle,
- en orange, la couche cachée (hidden layer)est formée de deux neurones,
- en vert, la couche de sortie (Output layer), formée d'un neurone, fournit le résultat final $P(Y_i = 1 \mid X)$,
- en bleu, les synapses et leur poids relient les neurones entre deux couches.

La couche cachée doit être formée au minimum d'un neurone mais peut en contenir plusieurs centaines. La couche de sortie doit être formée, dans les problèmes de prédiction d'une variable binaire, d'un seul et unique neurone permettant d'obtenir une probabilité $P(Y_i = 1 \mid X)$

Dans la suite, pour clarifier les notations, i représentera la $i$ème couche, j le jème niveau d'une couche. Ainsi, le $j$ème neurone de la $i$ème couche sera noté N^i_j, le poids des synapses entre le neurone j de la couche i et neurone k de la couche $i-1$ sera noté $w^i_{k,j}$. De manière générale un élément du réseau sera noté $a^{\text{couche}}_{\text{place dans la couche}}$

Les Neurones

Le neurone est l'élément central du réseau. Il est le lieu des opérations mathématiques. La valeur d'entrée d'un neurone que nous noterons a^i_j est la somme des valeurs des neurones de la couche précédente pondérée selon le poids des synapse $w^i_{k,j}$ plus un biais b^i_j . Pour la première couche cachée, c'est la somme pondérée des variables d'entrées x_i. Dans le cas de notre réseau de neurone, la valeur d'entrée (a^1_1) reçue par le premier neurone de la couche 1 (N^1_1) s'exprime par :

$$a^1_1 = \sum_{k=1}^{n=3} w^1_{k,1} * x_i + b_{1,1} \tag{2.57}$$

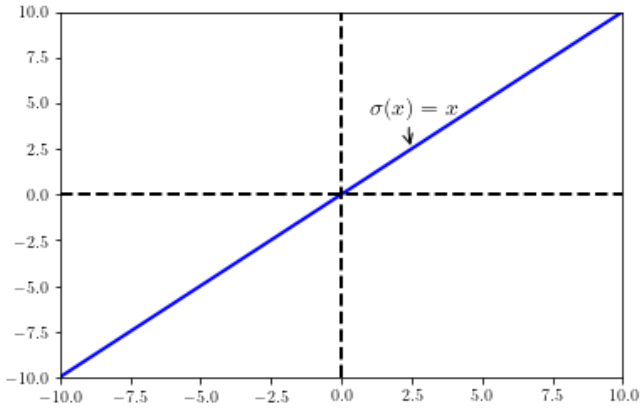


FIGURE 2.29 – Fonction d'activation identité

Une fois dans le neurone, cette entrée est ensuite modifiée par une fonction appelée la fonction d'activation ou de transformation σ_j^i afin d'obtenir la valeur de sortie du neurone s_j^i. La fonction d'activation peut être linéaire ou non. Chaque neurone a une fonction qui lui est propre. Dans la pratique nous choisirons une fonction d'activation pour l'ensemble des neurones d'une couche. La sortie s_1^1 du premier neurone a_1^1 de la première couche est donnée par.

$$s_1^1 = \sigma_1^1(a_1^1) = \sigma(\sum_{k=1}^{n=3} w_{k,1}^1 * x_i + b_1^1) \tag{2.58}$$

s_1^1 est la sortie du neurone N_1^1 qui sera utilisée par les neurones des couches suivantes. Le choix de la fonction d'activation a un rôle primordial dans l'efficacité d'un réseau. Elle n'est pas unique et peut être différente d'une couche à une autre.

Fonctions d'activation

Il existe un grand nombres de fonctions d'activation. Dans cette partie nous ne présenterons que les plus importantes et celles que nous utiliserons.

La fonction identité :

$$\sigma(x) = x \tag{2.59}$$

La sortie du neurone est la même que l'entrée

$$s_1^1 = \sigma_1^1(a_1^1) = b_1^1 + \sum_{i=1}^{n=3} w_1^i * x_i$$

La transformation neuronale n'est alors qu'une somme pondérée des variables. Un réseau de neurones formé de plusieurs couches de neurones utilisant le fonction identité aura pour résultat final une somme pondérée des variables. Le réseau de neurones trivial avec une seule couche de sortie (sans couche cachée) utilisant cette fonction d'activation peut être apparenté à une régression linéaire.

La fonction Marche

$$\sigma(x) = \begin{cases} 0 & \text{si} \quad x < 0 \\ 1 & \text{si} \quad x \geq 0 \end{cases} \tag{2.60}$$

La transformation neuronale modifie la somme pondérée reçue en entrée par la valeur 0 ou 1. Cette fonction d'activation peut être utilisée dans des problématiques de classification binaire. Ou dans une seconde couche de neurones de sortie, précédée d'un neurone avec une fonction d'activation donnant un probabilité. Le biais b ($b * w$, si le poids w entre les deux sorties est différent de 1) sera alors le point de séparation des deux classes.

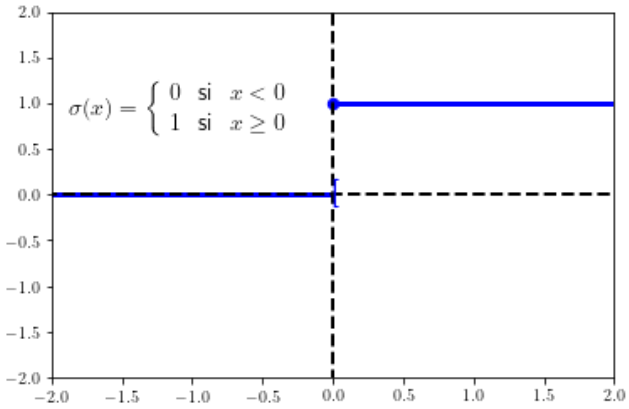


FIGURE 2.30 – Fonction d'activation Marche

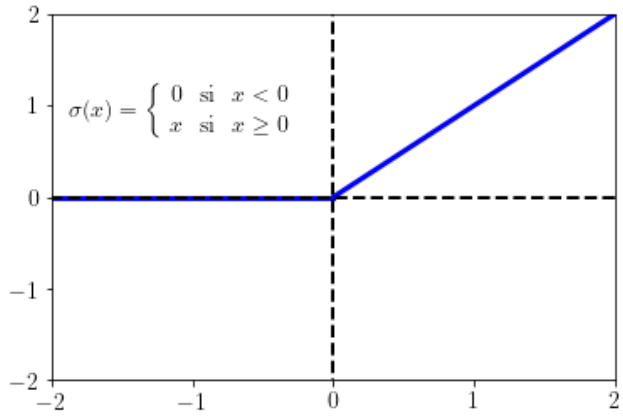


FIGURE 2.31 – Fonction d'activation Rectified Linear Unit (Relu)

La fonction Rectified Linear Unit (Relu) : C'est une fonction linéaire par morceau. L'un de ces avantages est que les valeurs négatives sont mises à 0 ce qui facilite l'entrainement du modèle.

$$\sigma(x) = \begin{cases} 0 & \text{si} \quad x < 0 \\ x & \text{si} \quad x > 1 \end{cases} \tag{2.61}$$

La fonction d'activation logistique ou sigmoïde : La fonction d'activation sigmoïde est utilisée afin d'obtenir les probabilités d'appartenance à une classe dans le cas de classification binaire. Un réseau de neurones trivial avec une seule couche de sortie utilisant cette fonction d'activation, s'apparente à une régression logistique où les poids des synapses sont égaux aux paramètres β obtenus par la régression logistique.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.62}$$

La sortie du neurone est donnée par : $\dfrac{1}{1+e^{-(b_1^1+\sum_{i=1}^{n=3} w_{1,i}*x_i)}}$

La fonction tangente hyperbolique : Moins utilisée que les autres fonctions d'activation, elle a cependant l'avantage de borner les données et d'être anti-symétrique.

$$\sigma(x) = \frac{2}{1+e^{-2x}} - 1 \tag{2.63}$$

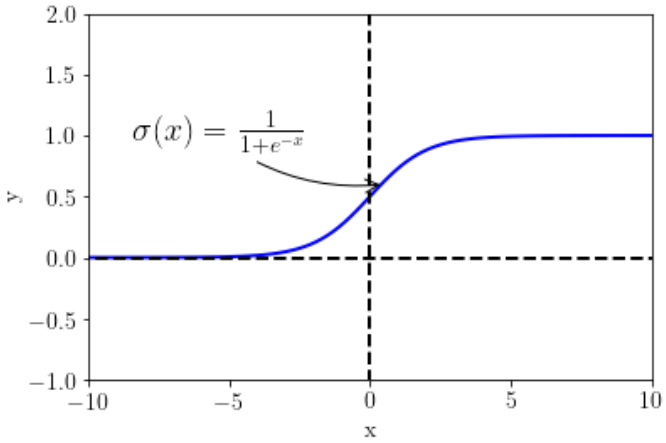


FIGURE 2.32 – Fonction d'activation sigmoïde

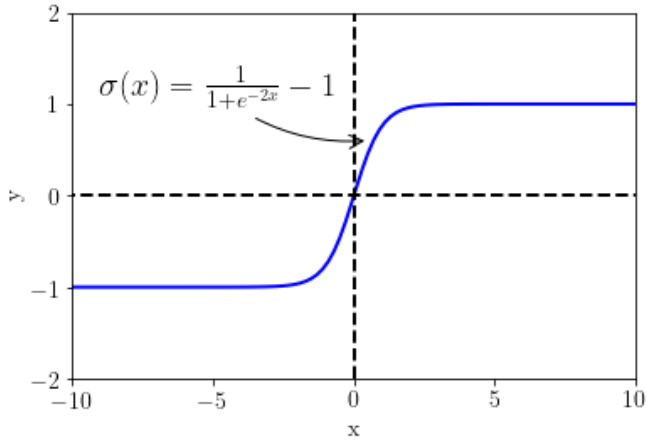


FIGURE 2.33 – Fonction d'activation tangente hyperbolique

Le résultat final (output layer)

La sortie du réseau est donnée par le résultat du dernier neurone : le output layer. La sortie peut être de plusieurs formes selon les problèmes à traiter (classification ou prédiction). Dans le cas des problèmes de prédiction, comme pour la figure 1, le résultat est une probabilité. Il peut être dans certain cas un score. Si nous reprenons le réseau de neurone de la figure 1, l'entrée de l'output layer est la moyenne pondérée des sorties des neurones de la couche précédente donnée par :

$$a_1^2 = w_{1,1}^2 * s_1^2 + w_{2,1}^2 * s_2^2$$

La sortie final du neurone est donné via la transformation par la fonction d'activation :

$$s_1^3 = \mathrm{P}(\mathrm{Y} = 1 \mid \mathrm{X} = x) = \sigma_1^2(a_1^2)$$

L'apprentissage

L'apprentissage des réseaux de neurones consiste à estimer les poids $w_{k,j}^i$ des connexions entre les neurones en fonction des résultats obtenus à chaque prédiction d'une nouvelle instance. Cet apprentissage se fait le plus souvent en utilisant la méthode de rétro-propagation du gradient (backpropagation) (Rumelhart et al., 1988). Dans un premier temps les poids des synapses sont initialisés aléatoirement. Il est possible d'imposer des poids déterminés au préalable.

Ensuite, vient l'étape du "forward pass" qui consiste à faire passer les données d'entrées initiales dans le réseau, de produire une sortie $\widehat{y}$ (probabilité par exemple) et de comparer ce résultat

au résultat attendu y à l'aide d'une fonction de perte L. Cela permet de calculer l'erreur $E = L(\hat{y}, y)$ effectuée par le modèle. Il existe une multitude de fonction de perte. Nous définirons les deux plus courantes.

— La fonction de perte quadratique :

$$L(\hat{y}, y) = (\hat{y} - y)^2$$

Il est possible de trouver cette fonction avec un facteur $\frac{1}{2}$ supplémentaire pour simplifier les calculs des dérivés.

— La fonction de perte cross-Entropy :

$$L(\hat{y}, y) = y log(\hat{y}) + (1 - y) log(1 - \hat{y})$$

La seconde étape le "backward pass", consiste à rétro-propager l'erreur afin de mettre à jour les poids du réseau. Pour ce faire, chaque poids synaptique $w^i_{k,j}$ est mis à jour en calculant la quantité suivant :

$$\Delta w^i_{k,j} = -\eta \frac{\partial E}{\partial w^i_{k,j}} \tag{2.64}$$

avec η le paramètre de la vitesse d'apprentissage. Ainsi à l'itération t, les poids sont mis à jour par la formule suivante :

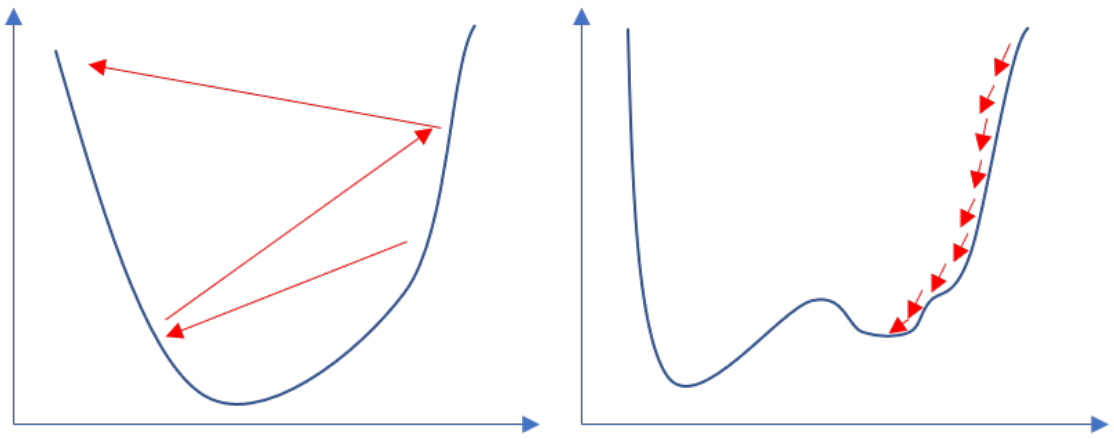
$$w^{i(t)}_{k,j} = w^{i(t-1)}_{k,j} + \Delta w^{i(t-1)}_{k,j}$$

Le choix du bon paramètre de vitesse de convergence η permet d'éviter deux problèmes.

— Si η est choisi trop petit, le réseau peut converger vers un minimum local et donc ne pas trouver la solution optimale 2.34b.

— Si η est choisi trop grand, le réseau peut tourner autour de la solution optimale sans jamais l'atteindre (2.34a).

En pratique on choisira un η entre 0.001 à 0.5.



(a) Vitesse d'apprentissage élevée

(b) Vitesse d'apprentissage faible

FIGURE 2.34 – Fonction de perte

2.3 Méthodes d'agrégation (Ensemble based methods)

Bien souvent, dans les problématiques d'évènements rares, l'utilisation de la bonne méthode prédictive et d'échantillonnage ne suffisent pas à obtenir des modélisations satisfaisantes. L'undersampling aura tendance à diminuer l'apprentissage de la classe majoritaire et l'oversampling

peut avoir des effets de sur-apprentissage de la classe minoritaire. La faiblesse des méthodes de modélisation peut également venir de la dispersion des évènements ou du fait que différentes causes peuvent entrainer un événement. Pour pallier ces problèmes nous pouvons utiliser les méthodes d'agrégation. Elles consistent en la construction de plusieurs modèles différents que l'on regroupe ensuite pour obtenir une classification. Dans la bibliographie, ces différents modèles utilisés qui seront agrégés sont appelés weak-classifier. Dans cette section, nous présenterons les méthodes que nous avons utilisées en commençant par la méthode la plus intuitive : le Bagging. Ensuite, dans la section suivante nous présenterons la méthode de boosting qui est une méthode d'agrégation itérative. La dernière section sera consacrée aux méthodes d'agrégation les plus évolués de boosting de gradient.

2.3.1 Bagging

Bagging ou Bootstrap agregagging est une méthode d'agrégation de modèles statistiques utilisée pour la classification ou la régression, développé par Breiman [BREIMAN, 1996]. Elle consiste à construire plusieurs modèles à l'aide de tirages aléatoires et à regrouper ces modèles par vote.

En reprenant les notations précédentes, soit S un échantillon de données de taille m, $S = \{(x_i, y_i), i = 1...m\}$, avec $x_i \in X$ espace de dimension n des variables explicatives et $y_i \in \{0,1\}$ la variable réponse. En considèrent la fonction aléatoire $U : \{1,...,m\} \rightarrow \{1,...,m\}$, la méthode bagging génère K nouveaux échantillons $S_{U^k} = (X_{U^k(1)},...,X_{U^k(m')})$ de taille m' par tirage aléatoire avec remise dans S (bootsrap [EFRON et TIBSHIRANI, 1993]). Il est donc possible d'avoir des répétitions d'observations dans chaque S_k ou que certaines ne soient jamais sélectionnées. Les variables aléatoires U sont définies sur $\mathbb{U} = \{(u_k)_k = 1,...,m^K\}$ l'ensemble des tirages de taille K avec remise dont le cardinal est m^K. On définit également $B = (U_1,...U_K)$ l'ensemble des fonctions aléatoires générées. Sur ces m échantillons, m modèles vont être construits et combinés par vote pour la classification ou par moyenne pour la régression. En admettant que $\varphi(x,S)$ est notre prédicateur, c'est à dire qu'en utilisant les données de S et x_i les variables explicatives de l'individu i, on a $y_i = \varphi(x_i,S)$.

On cherche à obtenir un meilleur prédicateur φ en utilisant les échantillons S_i quand utilisant simplement S. Pour les problème de régression les classifieurs peuvent être regroupés par moyenne :

$$\varphi_A(x,B) = \frac{1}{K}\sum_{i=1}^{K}\Phi(x,S_{U^i}) \tag{2.65}$$

Nous allons montrer la précision de l'estimateur $\varphi_A(x)$ obtenu par bagging en calculant l'erreur quadratique moyenne BREIMAN [1996].

Theorem 1 *Pour une statistique* θ*, il existe F et G positifs, indépendants tel que l'erreur quadratique moyenne (MSE) d'un estimateur "bagging"* $\tilde{\theta}$ *défini par 2.65, satisfait :*

$$MSE(\tilde{\theta}) = \frac{1}{K}F + G \tag{2.66}$$

avec,

$$\begin{aligned} F &= \mathbb{E}_S(Var_U(\varphi(S_U))) \\ G &= Var_S(\mathbb{E}_U(\varphi(S_U))) + \mathbb{E}_S(\mathbb{E}_U(\varphi(S_U) - \theta)^2 \end{aligned}$$

Où U *est la variable aléatoire distribuée uniformément sur* $\mathbb{U}$*. Plus généralement, on a*

$$\begin{aligned} &1.\mathbb{E}_{(L,B)}(\tilde{\theta}(L,B)) = \mathbb{E}_L(\mathbb{E}_U(\tilde{\theta}(L_U))) \\ &2.Var_{(L,B)}(\tilde{\theta}(L,B)) = \frac{1}{K}\mathbb{E}_L(Var_U(\tilde{\theta}(L_U)) + Var_L(\mathbb{E}_U(\tilde{\theta}(L_U))) \end{aligned}$$

On remarque que plus le nombre d'échantillons bootstrap K augmente plus la MSE est baisse. Et lorsque K tend vers ∞ :

$$\lim_{K\to+\infty} \mathrm{MSE}(\tilde{\theta}) = (Var_U(\tilde{\theta}(L_U)) + \mathbb{E}_S(\mathbb{E}_U(\varphi(S_U) - \theta)^2 \tag{2.67}$$

Bien souvent, il suffira de prendre un nombre d'échantillons entre 10 et 100 pour avoir un estimateur par bagging le plus précis.

Les modèles ou "weak learner" peuvent être choisis selon les besoins : régression logistique, régression de poisson, SVM ou arbre de classification. Concernant le nombre d'observations dans chaque échantillon bootstrap aucun consensus n'est trouvé dans la littérature. Il est cependant conseillé de le choisir entre 80 et 90% de la base pour éviter une trop grosse perte d'information. Il est également possible de prendre le même nombre d'observations que dans la base de départ. Dans ce cas, les cas uniques représenteront 63.2%, le reste sera des duplicatas.

L'algorithme du Bagging est donné ci-dessous

Algorithm 1 Algorithme de la régression bagging

Indiquer K le nombre de sous-échantillons à construire par tirage aléatoire dans S, n la taille des échantillons et I le classifieur faible utilisé.
$i = 0$
while $i \neq K$ **do**
  $i = i + 1$
  Construire S_i par tirage aléatoire avec remise dans S
  Construire le classifieur φ_{A_i} en utilisant le modèle I et l'échantillon S_i
end while
Finalement : $\varphi_A(x) = \dfrac{\sum_{i=1}^{K} \varphi_{A_i(x)}}{T}$

2.3.2 Généralité sur le Boosting

Le boosting, comme le bagging est une méthode d'agrégation de modèles qui consiste à transformer un ensemble de classifieurs faibles en un plus puissant. Au contraire du bagging où tous des classifieurs sont indépendants et construits en même temps, le boosting est une méthode itérative : les classifieurs sont construits les uns après les autres. Il y a donc une dépendance entre chaque classifieur. En effet, les données sont pondérées selon leur classement par le classifieur précédant. Finalement, l'agrégation des classifieurs se fait en utilisant des poids selon leur efficacité. C'est donc une méthode d'agrégation pondérée de modèles statistiques.

La méthode a dans un premier temps été développée pour les problèmes de classification [FREUND et SCHAPIRE, 1996] mais marche tout aussi bien sur des problèmes de régression. Les trois points importants du boosting sont les suivants :

— L'utilisation d'un comité d'experts spécialisés qui vote pour atteindre une décision : les classifieurs.

— La modification de la distribution des exemples disponibles pour entraîner chaque expert, en sur-pondérant au fur et à mesure les exemples mal classés aux étapes précédentes.

— La pondération adaptative des votes par une technique de mise à jour multiplicative, permettant d'augmenter le poids des meilleurs experts, (i.e. les classifieurs classant le mieux les données).

Le boosting est une méthode très flexible. Il en existe de nombreuses variantes. Nous présenterons les plus utilisées. Elles reposent toutes sur les même fondements que nous pouvons décrire ici étape par étape. Cela nous permettra de décrire le fonctionnement général de la méthode et de définir les notations utilisées dans cette partie.

Reprenons un échantillon S de taille m, $S = \{(x_i, y_i), i = 1...m\}$, φ_i défini le classifieur faible i. Par facilité de notation l'espace de Y est $\{-1, 1\}$. Pour commencer, $D^0 = (w_1^0, .., w_m^0)$ le vecteur des poids initiaux de chaque individu est initialisé. Le plus souvent on choisi $w_i^0 = \dfrac{1}{m}$, $i = 1, ..., m$ de

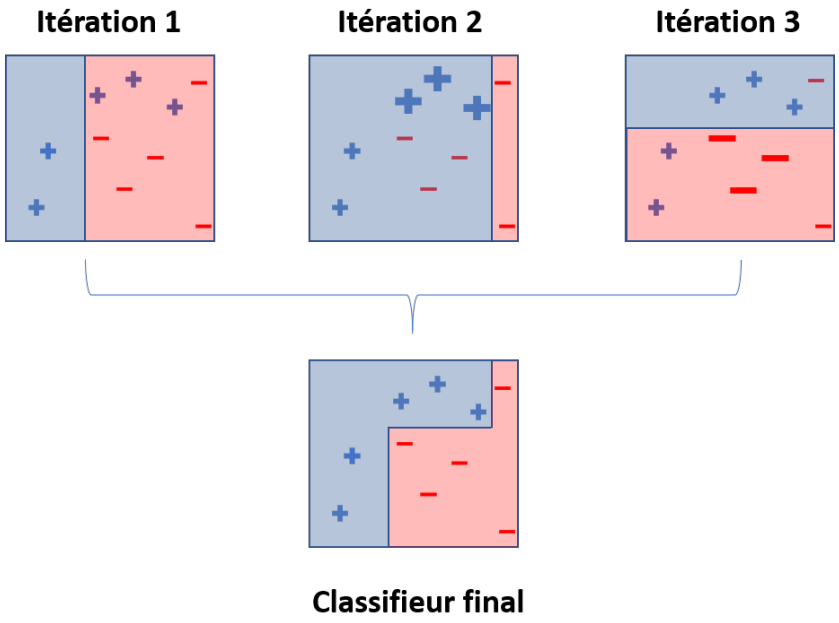


FIGURE 2.35 – Exemple d'un boosting à 3 itérations

sorte que chaque individu ait le même poids. Cependant dans des problème d'évènements rares, il est possible de donner plus d'importance (i.e. un poids plus élevé) aux individus de la classe minoritaire. Une fois D^1 choisi et T le nombre d'itérations (ou de classifieurs faibles) choisi, il suffit alors de répéter les étapes suivantes pour $t = 1, ..., T$:

1. Construction du classifieur $\varphi_t(x)$ en utilisant les poids D^t
2. Calcul de l'erreur ϵ_t de prédiction de $\varphi_t(x)$, mise à jour du poids du classifieur α_t
3. Mise à jour des poids D^{t+1} : le poids des individus bien classés diminue, celui des mal classés augmente.
4. Agrégation : $\varphi_A(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t \varphi_t(x))$

La figure 2.35 présente l'évolution simplifiée d'une classification avec boosting. L'augmentation des poids des observations mal classées est indiquée par l'augmentation de la taille de ces observations. La classification finale est obtenu en agrégeant les trois classifieurs faibles :

— Le premier classifieur classe correctement deux (+) et l'ensemble des (-). Les poids des mal classés vont alors augmenter pour le second classifieur.

— Le second classifieur tient compte des nouveaux poids. Il classe correctement l'ensemble des (+), mais trois (-) sont mal classés. Les poids sont alors mis à jour pour le dernier classifieur.

— Le troisième classifieur, en tenant compte des nouveaux poids, construit une nouvelle classification.

— Au final, ils sont regroupés dans un classifieur final qui classe correctement toutes les observations.

Ce graphique illustre bien le passage de classifieurs faibles à un bon classifieur. Si l'on regarde de plus près, les deux premiers classifieurs classent correctement 80% des observations, le troisième 70% et le final 100%.

Dans cette description du boosting, nous pouvons voir les multiples possibilités et l'importance du choix des α_t, ϵ_t et $\varphi_t(x)$. Contrairement au bagging, le boosting utilise l'ensemble des observations de S et à, chaque itération de l'algorithme, le poids des observations mal classées est

augmenté ce qui entraine un meilleur apprentissage de ces points, mais il en résulte souvent un sur-apprentissage.

Dans la prochaine partie, nous allons présenter les méthodes de modélisation utilisables ou les "weak learners".

Weak learners (classifieurs faibles)

Le choix de la méthode de modélisation n'est soumis à aucune règle. Il est possible de booster quasiment toutes les méthodes prédictives à la seule condition qu'elles doivent pouvoir utiliser les pondérations : SVM, régression logistique, etc (FRIEDMAN et collab. [2000]). On évitera cependant d'utiliser des méthodes qui utilisent déjà une agrégation telle que les Random Forest ou un boosting de boosting. Cela n'améliora pas la prédiction et aura pour effet de rallonger lourdement le temps de calcul. Dans la pratique, la méthode la plus utilisée sont les arbres de classification 2.2.3. En effet, ils sont simples à mettre en placent et ne possèdent que peu de paramètres personnalisables. Souvent, les arbres utilisés pour le boosting sont sous-développés avec seulement 2 couches : une racine et 2 feuilles terminales. Pourtant le boosting avec ces classifieurs faibles sont très performants. Ils sont même définis comme "best off-the-shelf classifier » en 1998 [BREIMAN, 1998]. Les arbres à deux couches séparent l'espace en deux à chaque itération (figure 2.35) jusqu'à produire un classifieur non linéaire performant. Finalement, l'interprétation du modèle construit est facile du fait de la simplicité des règles de décisions générées.

Il existe de nombreux algorithmes. L'un des premiers proposés à été AdaBoost FREUND et SCHAPIRE [1997]. Les classifieurs faibles utilisés sont les arbres de classification. Plusieurs papiers montrent l'efficacité de cette méthode KARAKOULAS et SHAWE-TAYLOR [1998]; SCHAPIRE et SINGER [1998] GALAR et collab. [2012]. La méthode étant efficace, elle a été modifiée pour les évènements rares ADAcost FAN et collab. [1999].

Adacost

Adacost permet d'améliorer l'apprentissage des observations de la classe minoritaire. Pour ce faire, la mise à jour des poids dépend de la classe de l'observation. Pour définir Adacoost, prenons $\mathscr{S} = \big((x_1, c_1, y_1), \ldots, (x_m, c_m, y_m)\big)$ Avec $x_i, i = 1, ..., n$ dans l'espace des variables $\mathscr{X}$, $\mathscr{Y}$ $in\{-1, +1\}$ et c_i le coût dans $\mathbb{R}^+$. Les classifieurs faibles sont définis par $h : \mathscr{X} \to \mathbb{R}$. Soit t l'index de l'itération du boosting et $D_t(i)$ le poids à l'instant t de l'observation (x_i, c_i, y_i). A chaque étape, les poids sont normalisés pour avoir $\sum D_t(i) = 1$. Enfin, α_t est le poids du weak learner à l'étape t. La principale différence avec les méthodes classiques est dans le paramètre de mise à jour des poids. Dans les méthodes de boosting le poids des mal classés est le même quelque soit l'observation. Dans adacost il diffère puisqu'il est donné par : $\beta\big(\text{sign}\big(y_i h_t(x_i)\big), c_i\big)$ qui dépend de la classification et du coût c_i. Ainsi une observation mal classée avec un gros coût initial aura un point encore plus augmenté. S'il est bien classé le poids ne sera que peu diminué ce qui permet de garder de l'importance dans l'apprentissage du modèle. L'algorithme est le suivant :

— Prendre $\mathscr{S} = \{(x_1, c_1, y_1), \ldots, (x_m, c_m, y_m)\}$ $x_i \in \mathscr{X}, c_i \in \mathbb{R}^+, y_i \in \{-1, +1\}$ et $D_1(i)$ ($D_1(i) = c_i / \sum_j^m c_j$).

— Pour $t = 1, \ldots, T$:

1. Entrainer un classifieur faible en utilisant D_t
2. Calculer $h_t : \mathscr{X} \to \mathbb{R}$.
3. Choisir $\alpha_t \in \mathbb{R}$ et $\beta(i) \in \mathbb{R}^+$.
4. Mettre à jour les poids

$$D_{t+1}(i) = \frac{D_t(i) \exp\big(-\alpha_t y_i h_t(x_i) \beta(i)\big)}{Z_t}$$

avec $\beta(i) = \beta\big(\text{sign}\big(y_i h_t(x_i)\big), c_i\big)$ la fonction d'ajustement des coût et Z_t un facteur de normalisation pour que D_{t+1} soit une distribution.

Enfin, le classifieur final est calculé avec $H(x) = \text{sign}(f(x))$, $f(x) = \left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$.

Adacost a été comparée à Adaboost sur plusieurs jeux de données et a montré une réduction des erreurs de classification FAN et collab. [1999].

Variante des algorithmes de boosting

Dans cette section, nous présentons deux variantes des méthodes classiques de boosting : EasyEnsemble et BalanceCascade LIU et collab. [2009]. Les deux méthodes sont construites pour les problèmes de classe déséquilibrées. Il en existe d'autres et les possibilités sont quasiment infinies

EasyEnsemble

EasyEnsemble parcourt aléatoirement la classe majoritaire S_{maj} et construit plusieurs classifieurs H_i par la méthode ADaboost afin de les regrouper et d'obtenir un unique classifieur performant. Plus précisément, la méthode construit T sous échantillons $S^1_{maj}, S^2_{maj}..., S^T_{maj}$ provenant d'un tirage aléatoire avec remise dans la classe majoritaire de telle sorte que les échantillons soient équilibrés ($|S^i_{maj}| = |S_{min}|$). Ensuite, pour chaque itération j, un classifieur H_j provenant de la méthode ADAboost est construit un utilisant S^j_{maj} et l'ensemble des observations de la classe minoritaire S_{min}. Finalement ,les H_i sont regroupés dans un classifieur H final. L'algorithme est présenté ci-dessous. Une comparaison sur les des jeux de données déséquilibrées peut être trouvée dans [LIU, 2009]

Algorithm 2 EasyEnsemble algorithm

Indiquer T le nombre de sous-échantillon S^i_{maj} à construire par tirage aléatoire dans S_{maj} et s_i le nombre d'itérations pour constuire le classifieur H_i avec adaboost.
$i = 0$
while $i \neq T$ **do**
  $i = i + 1$
  Construire S^i_{maj} par tirage aléatoire tel que $|S^i_{maj}| = |S_{min}|$
  Constuire le classifieur H_i en utilisant S^i_{maj} et S_{min}. H_i est un classifieur construit par adaboost avec s_i "weak classifiers" $h_{i,j}, \alpha_{i,j}$ les poids correspondants. i.e $H_i(x) = sgn(\sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x)$
end while
Finalement : $H(x) = \sum_{i=1}^{T} H_i(x)$

BalanceCascade

BalanceCascade est quasiment identique à EasyEnsemble. Elle diffère sur un point : les observations de la classe majoritaire bien classées à l'étape i par le classifieur ADaboost H_i sont supprimées et ne peuvent plus être utilisées dans l'apprentissage pour les autres classifieurs.

2.3.3 Gradient tree Boosting

Dans la partie précédente, nous avons exposé le principe général des algorithmes de boosting. De manière plus formelle, les méthodes de boosting sont des méthodes dites de descente de gradient. Au départ appelées Arcing (Adaptive Reweighting and Combining)BREIMAN [1999] par Breiman puis ensuite Gradient Boosting Machines par Friedman FRIEDMAN [2001], leur appellation s'est transformée en Gradient tree Boosting ou plus généralement Gradient Boosting. Les procédures de descente de gradient sont habituellement utilisées pour minimiser un ensemble de paramètres comme les coefficients dans une régression ou les poids dans les réseaux de neurones. Après le calcul d'une fonction de perte, les poids sont mis à jour pour minimiser cette erreur. Le boosting de gradient, lui, repose sur trois points

Algorithm 3 BalanceCascade algorithm

Indiquer T le nombre de sous-échantillon S^i_{maj} à construire par tirage aléatoire dans S_{maj} et s_i le nombre d'itération pour constuire le classifieur h_i avec adaboost.
$i = 0$
while $i \neq T$ **do**
  $i = i + 1$
  Construire S^i_{maj} par tirage aléatoire tel que $|S^i_{maj}| = |S_{min}|$
  Constuire le classifieur H_i en utilisant S^i_{maj} et S_{min}. H_i est un classifieur construit par adaboost avec s_i "weak classifiers" $h_{i,j}, \alpha_{i,j}$ les poids correspondants. i.e $H_i(x) = \sum_{j=1}^{s_i} \alpha_{i,j} h_{i,j}(x)$

  Retirer les observations bien classées de $\hat{Sm}_{aj}$ de l'ensemble des observations de la classe majoritaire S_{maj}
end while
Finalement : $H(x) = \sum_{i=1}^{T} H_i(x)$

1. La fonction de perte à optimiser (loss function).
2. Les classifieurs ou prédicteurs faibles (weak learner).
3. Le modèle additif.

Les fonctions de perte dépendent du problème à résoudre. Cependant, le choix est très varié. La seule condition de la fonction de perte est qu'elle soit différentiable. Il est possible par exemple de choisir la fonction "mean squared error" (MSE). Concernant les classifieurs faibles, comme exposé précédemment, le choix est grand mais dans cette partie, nous décidons de nous intéresser principalement aux arbres de classification à travers deux méthodes EXtreme Gradient Boosting (Xgboost) et LightGBM. Finalement, dans la modélisation additive les arbres sont construits l'un après l'autre et ne sont jamais modifiés au cours de la procédure. A chaque étape, on cherche à minimiser la fonction de perte en ajoutant un arbre. Les deux prochaines méthodes présentées sont de type Gradient tree Boosting mais la construction des arbres est différente.

EXtreme Gradient Boosting (Xgboost)

Xgboost proposée par Chen et Guestrin CHEN et GUESTRIN [2016], est une variante, très performante des méthodes dites de gradient boosting FRIEDMAN [2001]. C'est actuellement l'une des méthodes les plus concurrentielles dans les compétitions de machine learning. Le procédé repose sur la méthode du boosting combinée à la prédiction de K "weak learners" pour construire un prédicateur fort à l'aide d'une stratégie d'apprentissage particulière. Les "weak learners" utilisés sont les arbres de décision (CART). Le but principal est de prévenir le sur-apprentissage mais aussi d'optimiser les ressources matérielles, en proposant une méthode peu coûteuse en temps de calcul. Cela peut être fait en simplifiant les fonctions de coût et en utilisant des termes de régularisation.

Le fonctionnement de XGboost se définit comme suit. Prenons X_i le vecteur des variables explicatives pour l'observation i ($X_i = \{x_1, x_2, \ldots, x_n\}, i = 1, 2, \ldots, N$) et y_i la variable cible à prédire $\left(y_i = \{-1, +1\}, i = 1, 2, \ldots, N\right)$. La phase d'apprentissage est faite en utilisant une stratégie additive (identique au boosting), c'est à dire que chaque classifieur est construit l'un après l'autre. Le premier classifieur est construit sur la base d'apprentissage originale. Les poids sont redéfinis de sorte que le second donnera plus d'importance aux observations mal classées par le premier. La fonction de prédiction finale est donnée par :

$$\widehat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathscr{F}, \tag{2.68}$$

avec $\mathscr{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}\left(q : \mathbb{R}^m \to \mathrm{T}, w \in \mathbb{R}^{\mathrm{T}}\right)$ l'espace des arbres de régression, q la structure de chaque arbre et T le nombre de feuilles. Chaque f_k représente un arbre de structure avec comme poids des feuilles w. L'apprentissage des fonctions se fait en minimisant la fonction régularisée suivante :

$$\begin{aligned} &\mathscr{L}(\phi) = \sum_i l\left(\hat{y}_i, y_i\right) + \sum_k \Omega\left(f_k\right), \\ &\text{avec } \Omega(f) = \gamma \mathrm{T} + \tfrac{1}{2}\lambda \|w\|^2, \end{aligned} \tag{2.69}$$

où l est une fonction de perte différentiable et convexe qui mesure la différence entre la prédiction $\hat{y}_i$ et la valeur réelle y_i. Le second terme Ω pénalise la complexité du modèle. Il permet d'éviter le sur-apprentissage.

Intuitivement, l'objectif de la régularisation est de sélectionner des modèles performants et simples, en comparant le gain du modèle et sa complexité. Finalement après chaque étape de boosting une pondération η est ajoutée à chaque modèle (selon les prédictions), ce qui réduit l'influence de chaque arbre et améliore la performance. La méthode montre des résultats au moins aussi bon que ceux obtenus avec randomforest ou adaboost FAUZAN et MURFI [2018] La complexité de la mise en place de cette méthode est son paramétrage. Il existe de nombreux paramètres : la taille des arbres, le nombre de tirage aléatoire des observations ou des variables par exemple. Un mauvais paramétrage peut conduire à un sur-apprentissage. Il faut d'une part gérer les paramétrages des arbres tels que la taille maximum, le nombre d'observations maximum par feuille, les paramètres du boosting comme le nombre d'itérations, la pondération des mal classées et ceux propres à Xgboost comme la vitesse de descente du gradient et les termes de régularisation.

LightGBM

LightGBM est un algorithme de type boosting de gradient basé sur les arbres de classification. Il permet de résoudre des problèmes de classification ou de prédiction. Il est particulièrement rapide. La principale différence avec les autres algorithmes est la manière de construire les arbres de classification. Il divise l'arbre en fonction des feuilles avec le meilleur ajustement, tandis que d'autres algorithmes d'amplification divisent l'arbre en profondeur ou en niveau plutôt qu'en feuille. Cette modification apporte une meilleure précision de l'algorithme. La figure 2.36 montre la différence entre XGBoost et lightGBM. Avec lightGBM, lors de la construction des arbres, ce sont les feuilles avec le meilleur ajustement qui sont développées.

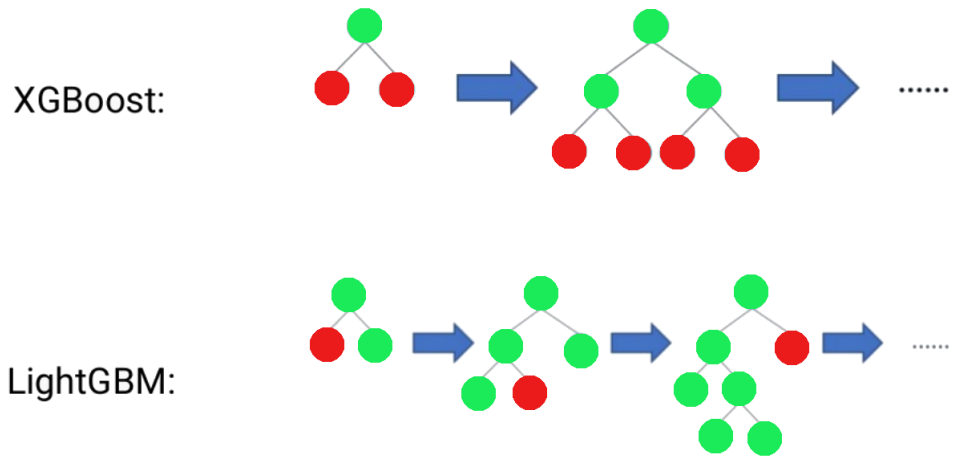


FIGURE 2.36 – Construction des arbres avec XGBoost et LightGBM

2.4 Validation des modèles avec données longitudinales

La performance des modèles peut être très sensible aux échantillons utilisés pour les valider. Pour éviter cette dépendance et le sur-apprentissage, le modèle doit être validé avec des observations tests qui ne sont pas utilisées pour le construire. Habituellement, ce sont des méthodes de validation croisée STONE [1974] qui sont utilisées. La base de données est divisée aléatoirement en k jeux de données, le modèle est construit sur $k-1$ jeux (les bases d'apprentissage) et testé sur le jeu de données restant (base test) qui servira à calculer une statistique s_1 évaluant le modèle. Ensuite, par permutation, les bases d'apprentissage deviennent tour à tour base de test et les bases tests, bases d'apprentissage, jusqu'à l'obtention des k statistiques s_i, $i = 1...k$. Finalement le modèle est évalué par la statistique de test :

$$\mathrm{S} = \sum_{i=1}^{k} s_i. \tag{2.70}$$

Dans le cas des données longitudinales, il existe deux raisons qui nous empêchent d'utiliser cette méthode ou une de ses variantes :

— la première est le biais engendré par les données déséquilibrées. En effet, lorsque les classes sont déséquilibrées, la répartition dans les bases d'apprentissage et de test peut être changeante, biaisant la performance de validation.

— La seconde, le caractère longitudinal de nos données. En effet ces méthodes n'en tiennent pas compte, et peuvent amener à prédire le passé en utilisant le futur comme base d'apprentissage.

2.4.1 Mesure de la qualité de prédiction

La performance d'un modèle d'apprentissage s'évalue par un risque ou une erreur de prévision. La mesure de cette performance a plusieurs utilités. D'une part, elle permet de sélectionner les modèles qui semblent les plus performants. Dans un second temps, elle guide le choix du paramétrage des modèles sélectionnés et permet donc de les optimiser. Finalement, une fois le meilleur modèle trouvé, elle mesure la qualité ou la confiance que l'on peut accorder à la prévision.

Le choix de la bonne mesure est donc primordial, d'autant plus lorsque les données sont déséquilibrées. Par exemple, si l'on considère le taux d'erreurs de classement, un modèle trivial qui ne prédit jamais la classe minoritaire commet un taux d'erreurs égal au pourcentage de cette classe (VALVERDE-ALBACETE et PELÁEZ-MORENO [2014]). Dans notre cas, il serait alors seulement de 4%, ce qui est un résultat très bon alors qu'aucune blessure n'est correctement prédite. Il faut donc s'intéresser à d'autres mesures. Dans le cas binomial, le modèle fournit une probabilité $\widehat{\pi_i}$ qu'un individu i présente l'évènement $(\mathrm{Y}_i = 1)$. Pour mesurer l'efficacité de nos modèles, il est possible de comparer cette valeur à un seuil γ tel que :

$$\text{si } \widehat{\pi}_i > \gamma, \quad \widehat{y}_i = 1 \quad \text{sinon} \quad \widehat{y}_i = 0. \tag{2.71}$$

Il suffit ensuite de confronter les résultats obtenus aux observations réelles et de calculer différentes mesures de performance.

Matrice de confusion

Il est possible de construire un tableau appelé "matrice de confusion", en croisant les modalités de la variable prédite pour un seuil s fixé avec celles de la variable observée. Pour un échantillon de taille n dont les observations Y_i sont connues elle peut être représentée par le tableau 2.1.

où :

Réel \ Prédit	$\tilde{y}$ = Non-blessure	$\tilde{y}$ = Blessure
y = Non-blessure	vrais négatifs	faux positifs
y = Blessure	faux négatifs	vrais positifs

TABLEAU 2.1 – Matrice de confusion

— Les vrais positifs (TP) représentent les observations (Y=1) correctement classées.
— Les vrais négatifs (TN) représentent les observations (Y=0) correctement classées.
— Les faux négatifs (FN) représentent les observations (Y=1) mal classées.
— Les faux positifs (FP) représentent les observations (Y=0) mal classées

Nous pouvons alors définir plusieurs mesures de performance du modèle :

- La précision (ou accuracy) :
la précision globale du modèle(taux de bien classés) :

$$\mathrm{ACC}(\gamma) = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}. \tag{2.72}$$

Comme explicité plus haut, lorsque les données sont déséquilibrées, il faut accompagner la précision d'autres mesures de performances.

- Taux de vrais positifs (ou sensibilité, recall) :
taux de bien prédit parmi les observations présentant l'évènement (Y=1)

$$\mathrm{TPR}(\gamma) = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{2.73}$$

- Taux de vrais négatifs (ou spécificité) :
taux de bien prédit parmi les observations ne présentant pas l'évènement (Y=1)

$$\mathrm{TNR}(\gamma) = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}. \tag{2.74}$$

Ces trois mesures permettent d'obtenir une évaluation plus complète puisque l'on connait la performance globale du modèle, son taux d'évènements correctement prédits et son taux d'évènements mal prédits. L'objectif est de trouver le seuil qui offre le meilleur compromis entre la sensibilité et la spécificité, pour cela il est possible de faire varier le seuil γ, bien souvent une balance s'installe entre ces deux paramètres : le gain obtenu sur l'un est perdu sur l'autre.

Courbe ROC

Le lien entre spécificité et sensibilité peut être représenté graphiquement par la courbe ROC (Receiver Operating Caracteristic) qui croise sensibilité en fonction de 1 – spécificité pour chaque valeur γ du seuil. Le point optimal de la courbe se trouve en $(0,1)$, qui correspond à un modèle sans erreur. La courbe ROC permet :

— de comparer visuellement les modèles entre eux. Plus la courbe se rapproche du point $(0,1)$ plus le modèle est considéré comme bon. La figure 2.37 présente la courbe ROC d'un modèle. La première bissectrice correspond au modèle classant les observations aléatoirement. Si la courbe ROC d'un modèle est toujours supérieure à la courbe d'un autre modèle alors il est considéré comme plus performant. Si les deux courbes se croisent, il est possible de les départager à l'aide de l'AUC (Aera Under the Curve). Ce paramètre correspond à l'aire sous la courbe ROC et mesure la qualité globale de discrimination du modèle.

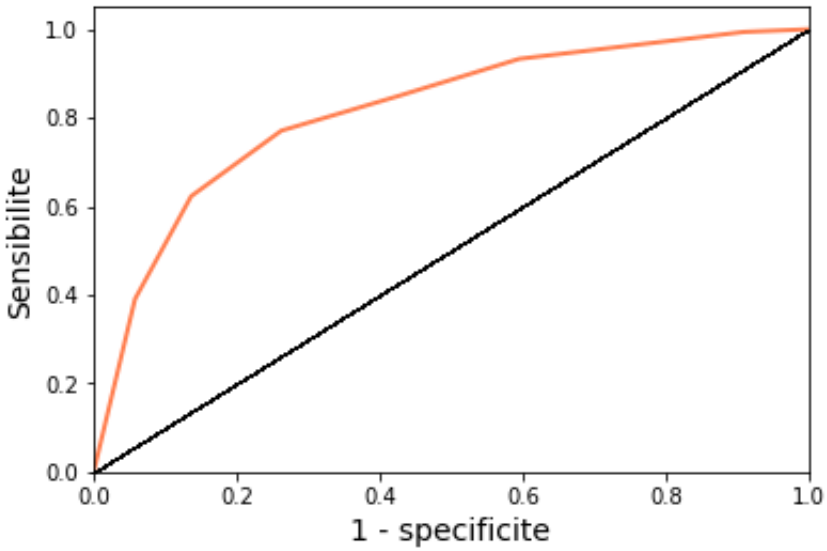


FIGURE 2.37 – Courbe ROC

— de trouver le seuil qui correspond aux attentes recherchées. En effet, il est possible dans certain cas de privilégier un seuil qui favorise la sensibilité au dépend de la spécificité ou inversement. Dans le cas de données déséquilibrées, nous seront particulièrement attentifs à ce que la sensibilité soit bonne. En effet, puisque la blessure est un évènement grave, il est préférable de pouvoir en détecter le plus possible, même si cela augmente le nombre de faux positifs.

.

Un indicateur important que nous présentons ici est le score de Peirce (PS) [PEIRCE, 1884]. Il a été conçu pour la prévision d'événements rares afin de pénaliser les modèles ne prévoyant jamais ces événements ou générant trop de fausses alertes. Il est défini par :

$$\mathrm{PS}(\gamma) = \text{sensitivity} + \text{specificity} - 1. \tag{2.75}$$

Il est compris entre -1 et 1. Si ce score est supérieur à 0, le taux de bonnes prévisions est supérieur à celui des fausses alertes. Plus il est proche de 1, meilleur est le modèle. Il faut donc faire en sorte de maximiser cet indicateur. C'est pour cela que nous définissons l'index de Peirce (PI) qui est score de Peirce le plus élevé :

$$\mathrm{PI} = \max_{\gamma \in [0,1]} \mathrm{PS}(\gamma). \tag{2.76}$$

Il représente un bon compromis entre la sensibilité et la spécificité. Il peut être vu comme $\mathrm{PI} = 1 - d^*$, avec d^* la distance de Manhattan entre le point $(0,1)$ et le point le plus proche sur la courbe ROC. C'est également la distance euclidienne entre le point de la courbe ROC le plus éloigné de la diagonale, à un facteur $\sqrt{2}$ près. Un bon modèle doit avoir le plus haut AUC et le plus haut PI possible.

Courbe Précision-rappel (PRC)

Le dernier outil d'évaluation que nous présentons est la courbe précision-rappel (PRC). La précision décrit la quantité de cas réellement positifs parmi l'ensemble des cas prédits positifs par le modèle. Elle est définie par

$$\text{précision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.77}$$

La courbe précision-rappel représente la précision en fonction de la sensibilité (recall) pour des seuils γ différents. Un exemple est donné dans la figure 2.38. Un modèle idéal est égal à la fonction $f(x) = 1$, alors qu'un modèle non informatif est égal à la fonction $f(x) = C$ avec C le taux d'observations présentant l'évènement (Y=1) dans la base :

$$C = \frac{TP + FN}{TP + FN + TN + FP} \tag{2.78}$$

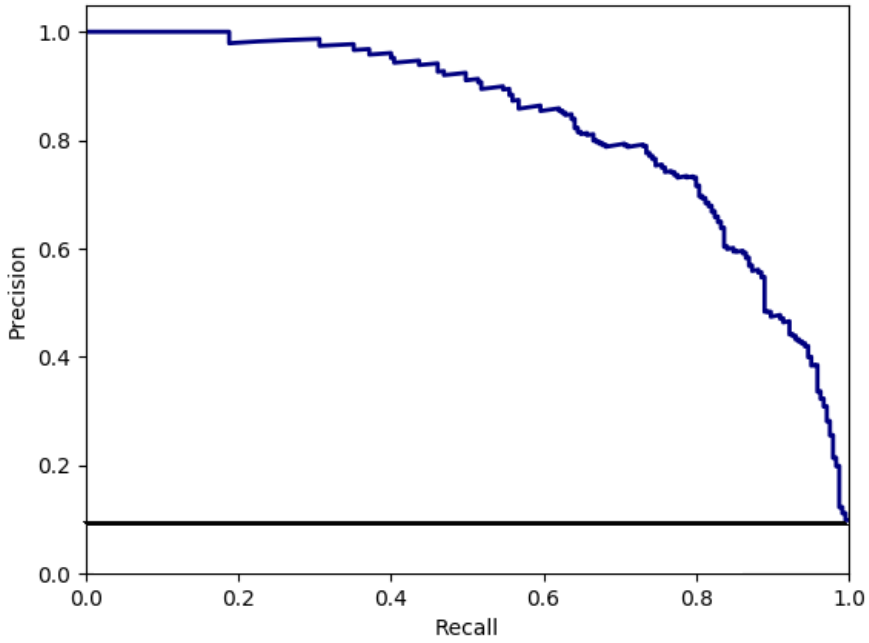


FIGURE 2.38 – Courbe précision rappel

La courbe précision rappel permet de visualiser la perte de performance prédictive de la classe majoritaire par rapport au gain sur la classe minoritaire. C'est un critère important dans les jeux de données déséquilibrées puisqu'il arrive fréquemment que le gain d'une observation de la classe minoritaire bien classée entraine le mauvais classement de nombreuses observations de la classe majoritaire. Comme pour la courbe ROC, il est possible de calculer l'aire sous la courbe, ce qui donne un indicateur de comparaison de modèles appelé Average Precision ou plus simplement AP. La courbe PRC est beaucoup moins utilisée que la courbe ROC alors qu'elle apporte un complement d'information important sur les modèles SAITO et REHMSMEIER [2015]

Les indicateurs utilisés dans la suite du manuscrit sont résumés dans la figure 2.39

2.4.2 Validation du modèle avec évènements rares

Le plus souvent pour valider un modèle la validation croisée ([REFAEILZADEH et collab., 2009] est utilisée. En présence d'évènements rares, elle peut être biaisée. En effet avec n_i^b (res.p n_j^b) le nombre d'évènements du $i$ème jeu (res. du $j$ème jeu), il est possible que $n_i \neq n_j$ ce qui peut entrainer un biais de la statistique d'évaluation S. Pour illustrer ce phénomène, prenons un échantillon constitué de 100 non évènements et 15 évènements ainsi qu'un modèle dont la précision est 95.7% classant correctement l'intégralité des non évènements et 2/3 des évènements (10 des 15). Considérons une validation croisée à 5 blocs présentant les caractéristiques suivantes :

1. 1 évènement bien prédit et 10 non évènements bien prédits
2. 1 évènement bien prédit et 10 non évènements bien prédits

Mesure	Descriptif
Sencibilité	Taux d'observastion présentant l'évènement correcement classé
Specificité	Taux d'observastion ne présentant pas l'évènement correcement classé
Index de Pierce	Plus courte distance entre la courbe ROC et le point optimal (0,1)
ROC	Courbe Roc croisant la sencibilité et la specificité
PRC	Courbe précision rappel
AUC	Aire sous la courbe ROC
AP	Aire sous la courbe Precision-Rappel

FIGURE 2.39 – Résumé des mesures utilisées

3. 4 évènements (3 bien prédits et 1 mal prédit) et 10 non évènements bien prédits
4. 4 évènements (3 bien prédits et 1 mal prédit) et 10 non évènements bien prédits
5. 5 évènements (2 bien prédits et 3 mal prédits) et 10 non évènements bien prédits

Pour chaque bloc le taux de non évènements bien classés est 100% mais la précision et le taux d'évènements bien classés sont les suivants :

1. Taux évènements bien classés : 100%, précision : 100%
2. Taux évènements bien classés : 100%, précision : 100%
3. Taux évènements bien classés : 75%, précision : 92.9%
4. Taux évènements bien classés : 75%, précision : 92.9%
5. Taux évènements bien classés : 40%, précision : 80.0%

Finalement après moyenne, la précision globale sera de 93.16%, i.e. en dessous de la précision initiale et le taux d'évènements bien classés de 78% soit au dessus des 75% du modèle initial.

Il est possible de corriger ce biais en utilisant d'autres techniques comme par exemple la validation croisée stratifiée [LÓPEZ et collab., 2014]. Elle consiste à s'assurer que dans chaque bloc d'apprentissage et de validation, la répartition des classes soit la même que dans la base de données initiale.

Le suivi longitudinal

Lors d'un suivi longitudinal, les individus étudiés évoluent dans le temps et sont observés à différents instants. La survenue d'un évènement (blessure par exemple) est très dépendant des données passées et change le futur de l'individu. Il est donc primordial de tenir compte de l'évolution temporelle des données. Dans ce contexte il est impossible de séparer aléatoirement la base en apprentissage et validation. Ce procédé pourrait nous conduire à prendre des conséquences pour des causes et inversement.

Nous avons donc décidé d'utiliser une méthode de validation longitudinale qui correspond à l'utilisation de l'algorithme en vie réelle. Supposons K instants à prédire ordonnés $t_1 < t_i < ... < t_K$ (correspondant à un match pour nous). La technique de validation consiste à prédire dans un premier temps t_1 en utilisant comme données d'apprentissage les données disponibles avant t_1, puis prédire t_2 en utilisant comme données d'apprentissage les données disponibles avant t_2 (en englobant les données à t_1) et ainsi de suite jusqu'à t_K. Au fur et à mesure, les bases tests deviennent base d'apprentissage. Au final, nous avons K bases d'apprentissage et K bases de validation différentes. Une fois les K prédictions faites, nous les comparons aux valeurs observées pour obtenir la précision de nos modèles. Un exemple graphique est donné dans la figure 2.40, pour la prédiction des 10 derniers matchs d'une série de 20.

Cette méthodologie nous permet d'évaluer, en utilisant plusieurs bases de validation et d'apprentissage, nos méthodes dans des conditions réelles. De plus, cela nous permet de tenir compte du maximum d'informations puisque la base d'apprentissage augmente au fur et à mesure.

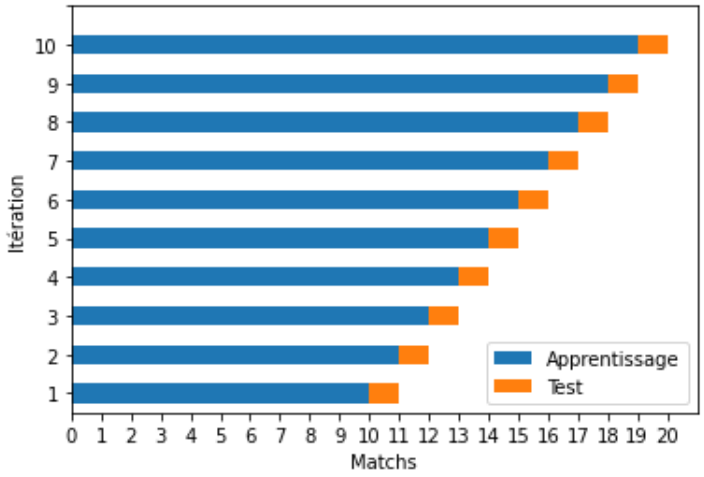


FIGURE 2.40 – Validation longitudinal

Comparaison des méthodes de validations

Pour montrer l'apport de l'enrichissement de la base d'apprentissage à chaque étape, nous avons comparé différentes méthodes de validation :

- la validation longitudinale, comme décrite dans la section 2.4.2.
- La validation par une base de validation obtenu en séparant aléatoirement la base en base d'apprentissage et de base validation.
- La validation croisée par block, en $k = 10$ block.
- La validation longitudinale par blocs : elle fonctionne comme la validation longitudinale, mais le "pas temporel" est différent. En effet, l'avancement est plus rapide, la prédiction ne se fait pas match par match mais K/10 matchs par K/10 matchs. Comme pour la validation croisée la base de données est divisée en 10 blocs mais en utilisant la temporalité des données. Les bases de données ne sont plus créées aléatoirement :
  - la base de données initiale est divisée en 10 bases de données ordonnées, notées base 1, base2,... base 10. L'ensemble des observations de la base 1 ont été observées avant celles de la base 2, celles de la base 2 avant celles de la base 3 et ainsi de suite.
  - Pour la première itération la base 1 est utilisée en apprentissage et la base 2 en base test.
  - Pour la seconde itération les bases 1 et 2 sont utilisées en apprentissage et la base 3 en base test et ainsi de suite.
  - Pour la dernière itération les bases 1 à 9 sont utilisées en base apprentissage et la base 10 en base de validation.
  - Les mesures de qualité obtenues à chaque itération sont moyennées pour obtenir une mesure finale.

Nous utiliserons la régression logistique comme méthode de modélisation. La statique ou mesure utilisée pour comparer sera l'AUC, puisqu'elle permet d'évaluer l'efficacité moyenne du modèle. Cette statistique sera calculée sur la prévision des matchs de 2017 à 2019. Les résultats sont présentés dans le tableau suivant.

	validation longitudinale	validation croisée	validation croisée temporelle	Validation par base test
AUC	0.679	0.617	0.652	0.634

La validation croisée simple donne le résultat le plus faible comparé aux résultats obtenus avec les validations considérant la temporalité des données. La meilleure est la validation longitudinale que nous proposons d'utiliser.Elle correspond à l'utilisation des données dans la vie réelle . De plus, elle prend en considération les données les plus proches du match à prédire. Il faut noter que les résultats restent néanmoins proches pour l'ensemble des méthodes.

2.5 Références

ABD RANI, K., H. A. ABD RAHMAN, S. FONG, K. ZURAIDA et N. ABDULLAH. 2013, «An application of oversampling, undersampling, bagging and boosting in handling imbalanced dataset», . 13

ABRIL, L. G., H. NÚÑEZ, C. ANGULO et F. V. MORENTE. 2014, «Gsvm : An svm for handling imbalanced accuracy between classes inbi-classification problems», *Appl. Soft Comput.*, vol. 17, p. 23–31. 31

AHA, W., D. KIBLER et M. ALBERT. 1991, «Instance-based learning algorithms», *Machine Learning*, vol. 6, p. 37–66. 8

BATISTA, G., A. BAZZAN et M.-C. MONARD. 2003, «Balancing training data for automated annotation of keywords : a case study.», *the Proc. Of Workshop on Bioinformatics*, p. 10–18. 11

BATISTA, G., R. PRATI et M.-C. MONARD. 2004a, «A study of the behavior of several methods for balancing machine learning training data», *SIGKDD Explorations*, vol. 6, doi :10.1145/1007730.1007735, p. 20–29. 11

BATISTA, G., R. PRATI et M.-C. MONARD. 2004b, «A study of the behavior of several methods for balancing machine learning training data», *SIGKDD Explorations*, vol. 6, p. 20–29. 13

BATUWITA, R. et V. PALADE. 2010, «Efficient resampling methods for training support vector machines with imbalanced datasets», *The 2010 International Joint Conference on Neural Networks (IJCNN)*, p. pp. 1–8. 13

BISHOP, C. 2006, *Pattern Recognition and Machine Learning*. 33

BREIMAN, L. 1996, «Bagging predictors», *Springer*. 38

BREIMAN, L. 1998, «Arcing classifiers», *The Annals of Statistics*, vol. 26, n° 3, p. 801–824. 41

BREIMAN, L. 1999, «Prediction games and arcing algorithms», *Neural Computation*, vol. 11, p. 1493–1517. 42

BREIMAN, L., J. FRIEDMAN, C. STONE et R. OLSHEN. 1984, «Classification and regression trees», . 18, 19, 20, 21

BREIMAN, M. 2001, «Random forests», *Machine Learning*, vol. 45(1), p. 5–32. 21

BRESLOW, N. E. 1996, «Statistics in epidemiology : The case-control study», *Journal of the American Statistical Association*, vol. 91, n° 433, p. 14–28. 15

BROOMHEAD, D. et D. LOWE. 1988, «Radial basis functions, multi-variable functional interpolation and adaptive networks», *ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM)*, vol. RSRE-MEMO-4148. 32

CALABRESE, R. et S. OSMETTI. 2015, «Improving forecast of binary rare events data : A gam-based approach», *Journal of Forecasting*, vol. 34, doi :10.1002/for.2335. 15

CHAWLA, N., K. BOWYER, L. HALL et W. KEGELMEYER. 2002, «Smote : Synthetic minority oversampling technique», *J. Artif. Intell. Res. (JAIR)*, vol. 16, p. 321–357. 9

CHEN, T. et C. GUESTRIN. 2016, «Xgboost : A scalable tree boosting system», p. 785–794, doi :10.1145/2939672.2939785. 43

CHOMBOON, K., P. CHUJAI, P. TEERARASSAMMEE, K. KERDPRASOP et N. KERDPRASOP. 2015, «An empirical study of distance metrics for k-nearest neighbor algorithm», , p. 280–285. 5

CORTES, C. et V. VAPNIK. 2009, «Support-vector networks», *Chem. Biol. Drug Des.*, vol. 297, p. 273–297. 29

CUTLER, A., D. CUTLER et J. STEVENS. 2011, «Random forests», *Machine Learning - ML*, vol. 45, doi :10.1007/978-1-4419-9326-7_5, p. 157–176. 21

DRUMMOND, C. et R. HOLTE. 2003, «C4.5, class imbalance, and cost sensitivity : Why under-sampling beats oversampling», *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets*. 4

DUDA, R., P. HART et D. G.STORK. 2001, *Pattern Classification*. 18

EFRON, B. et R. . TIBSHIRANI. 1993, «An introduction to the bootstrap», *Chapman and Hall*. 22, 38

FAN, W., S. STOLFO, J. ZHANG et P. CHAN. 1999, «Adacost : Misclassification cost-sensitive boosting», *Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*. 41, 42

FAUZAN, M. et H. MURFI. 2018, «The accuracy of xgboost for insurance claim prediction», *International Journal of Advances in Soft Computing and its Applications*, vol. 10, p. 159–171. 44

FERNÁNDEZ, A., S. GARCÍA, M. GALAR, R. PRATI, B. KRAWCZYK et F. HERRERA. 2018, «Learning from imbalanced data sets», . 8

FIX, E. et J. L. HODGES. 1989, «Discriminatory analysis. nonparametric discrimination : Consistency properties», *International Statistical Review / Revue Internationale de Statistique*, vol. 57, n° 3, p. 238–247. 6, 23

FREUND, Y. et R. SCHAPIRE. 1997, «A decision-theoretic generalization of on-line learning and an application to boosting», *J. Comput. Syst. Sci.*, vol. 55, p. 119–139. 41

FREUND, Y. et R. E. SCHAPIRE. 1996, «Experiments with a new boosting algorithm», . 39

FRIEDMAN, J., T. HASTIE et R. TIBSHIRANI. 2000, «Additive logistic regression : a statistical view of boosting (with discussion and a rejoinder by the authors)», *Ann. Statist.*, vol. 28, n° 2, doi : 10.1214/aos/1016218223, p. 337–407. URL https://doi.org/10.1214/aos/1016218223. 41

FRIEDMAN, J. et B. POPESCU. 2003, «Importance sampled learning ensembles», . 21

FRIEDMAN, J. H. 2001, «Greedy function approximation : A gradient boosting machine», *The Annals of Statistics*, vol. 29, n° 5, p. 1189–1232. 42, 43

GALAR, M., A. FERNÁNDEZ, E. BARRENECHEA, H. SOLA et F. HERRERA. 2012, «A review on ensembles for the class imbalance problem : Bagging-, boosting-, and hybrid-based approaches», *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, vol. 42, p. 463 – 484. 41

GIVENS, G. H. et J. A. HOETING. 2012, «Computational statistics», *John Wiley and Sons*. 15

GREENE, W. 2008, «Econometric analysis», *Macmillan Publishing Company*, vol. 7. 15

HAN, H., W.-Y. WANG et B.-H. MAO. 2005, «Borderline-smote : A new over-sampling method in imbalanced data sets learning», p. 878–887. 11

HART, P. 1968, «The condensed nearest neighbour rule», *IEEE Transactions on Information Theory*, vol. 14, p. 515–516. 7

HASTIE, T., R. TIBSHIRANI, J. FRIEDMAN et J. FRANKLIN. 2004, «The elements of statistical learning : Data mining, inference, and prediction», *Math. Intell.*, vol. 27, p. 83–85. 13

HAYKIN, S. 1999, *Neural Networks : A Comprehensive Foundation*. 33

HE, H., Y. BAI, E. GARCIA et S. LI. 2008, «Adasyn : Adaptive synthetic sampling approach for imbalanced learning», *Proceedings of the International Joint Conference on Neural Networks*, p. 1322 – 1328. 11, 12

HE, H. et E. GARCIA. 2009, «Learning from imbalanced data», *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, p. 1263 – 1284. 13

HE, H. et Y. MA. 2013, «Imbalanced learning : Foundations, algorithms, and applications», . 10

HILBE, J. M. 2009, *Logistic Regression Models*. 13

HILBORN, C. G. et D. LAINIOTIS. 1967, «The condensed nearest neighbor rule», . 7

KARAKOULAS, G. I. et J. SHAWE-TAYLOR. 1998, «Optimizing classifers for imbalanced training sets», dans *NIPS*. 41

KASS, G. V. 1980, «An exploratory technique for investigating large quantities of categorical data», *Journal of Applied Statistics*, vol. 2, p. pp. 119–127. 19, 20

KING, G. et L. ZENG. 2002, «Logistic regression in rare events data», *Political Analysis*, vol. 9, doi : 10.1093/oxfordjournals.pan.a004868. 15

KÖKNAR-TEZEL, S. et L. J. LATECKI. 2009, «Improving svm classification on imbalanced data sets in distance spaces», , p. 259–267. 30

KOHONEN, T. 2001, *Self-Organizing Maps*. 32

KUBAT, M. 2000, «Addressing the curse of imbalanced training sets : One-sided selection», *Fourteenth International Conference on Machine Learning*. 6, 7

KUHN, H. W. et A. W. TUCKER. 1951, «Nonlinear programming», , p. 481–492. 26

LANCASTER, T. et G. IMBENS. 1996, «Case-control studies with contaminated controls», *Journal of Econometrics*, vol. 71, n° 1, p. 145 – 160. 15

LAURIKKALA, J. 2001, «Improving identification of difficult small classes by balancing class distribution», p. 63–66, doi :10.1007/3-540-48229-6_9. 8

LEVERSHA, G. 2003, «Statistical inference (2nd edn)», *The Mathematical Gazette*, vol. 87, n° 509, p. 401–403. 15

LIU, T.-Y. 2009, «Easyensemble and feature selection for imbalance data sets», p. 517–520. 42

LIU, X.-Y., J. WU et Z.-H. ZHOU. 2009, «Exploratory undersampling for class-imbalance learning», *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 39, p. 539 – 550. 13, 42

LÓPEZ, V., A. FERNÁNDEZ et F. HERRERA. 2014, «On the importance of the validation technique for classification with imbalanced datasets : Addressing covariate shift when data is skewed», *Information Sciences*, vol. 257, p. 1 – 13. 49

MAALOUF, M. 2011, «Logistic regression in data analysis : An overview», *International Journal of Data Analysis Techniques and Strategies*, vol. 3, doi :10.1504/IJDATS.2011.041335, p. 281–299. 15

MAALOUF, M., D. HOMOUZ et T. TRAFALIS. 2017, «Logistic regression in large rare events and imbalanced data : A performance comparison of prior correction and weighting methods : Maalouf et al .», *Computational Intelligence*, vol. 34. 16

MAALOUF, M. et M. SIDDIQI. 2014, «Weighted logistic regression for large-scale imbalanced and rare events data», *Knowledge-Based Systems*, vol. 59. 16

MCCULLAGH, P. et J. NELDER. 1989, *Generalized Linear Model*. 13

NUÑEZ, H., L. GONZALEZ-ABRIL et C. ANGULO. 2017, «Improving svm classification on imbalanced datasets by introducing a new bias», *Journal of Classification*, doi :10.1007/s00357-017-9242-x. 30

PEIRCE, C. S. 1884, «The numerical measure of the success of predictions.», *Science*, vol. ns-4, p. 453–454. 47

PLATT, J. C. 1999, «Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods», , p. 61–74. 30

POISSON, S. D. 1887, *Recherches sur la probabilité des jugements en matière criminelle et en matière civile*, Elibron Classics. 16

RAMIREZ, F. et H. ALLENDE. 2012, «Dual support vector domain description for imbalanced classification», , p. 710–717. 30

REFAEILZADEH, P., L. TANG et H. LIU. 2009, *Cross-Validation*, Springer US, Boston, MA, ISBN 978-0-387-39940-9, p. 532–538. 48

RUMELHART, D., G. E. HINTON et R. J. WILLIAMS. 1986, «Learning representations by back-propagating errors», *Nature*, vol. 323, p. 533–536. 33

SAITO, T. et M. REHMSMEIER. 2015, «The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets», *PloS one*, vol. 10, p. e0118 432. 48

SCHAPIRE, R. et Y. SINGER. 1998, «Boostexter : A system for multiclass multi-label text categorization», . 41

SEIFFERT, C., T. KHOSHGOFTAAR, J. VAN HULSE et A. NAPOLITANO. 2008, «Building useful models from imbalanced data with sampling and boosting.», p. 306–311. 13

SHAWE-TAYLOR, J. et N. CRISTIANINI. 2004, *Kernel Methods for Pattern Analysis*, Cambridge University Press. 29

STONE, M. 1974, «Cross-validatory choice and assessment of statistical predictions», *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 36, n° 2, p. 111–133. 45

TOMEK, I. 1976, «Two modifications of cnn», *EEE Transactions on Systems Man and Communications*, p. 769–772. 5

VALVERDE-ALBACETE, F. J. et C. PELÁEZ-MORENO. 2014, «100normalized information transfer factor explains the accuracy paradox», *PLoS ONE*, vol. 9. 45

VAN HULSE, J. et T. KHOSHGOFTAAR. 2009, «Knowledge discovery from imbalanced and noisy data», *Data and Knowledge Engineering*, vol. 68, n° 12, p. 1513 – 1542. 10

VAPNIK, V. et A. LERNER. 1963, «Pattern recognition using generalized portrait method», *Automation and Remote Control*, vol. 24, p. 774–780. 25

VEROPOULOS, K., C. CAMPBELL et N. CRISTIANINI. 1999, «Controlling the sensitivity of support vector machines», *Proceedings of International Joint Conference Artificial Intelligence*. 30

VISA, S. et A. RALESCU. 2005, «Issues in mining imbalanced data sets - a review paper», *Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conference*. 15

WILLIAM P., Z. 1989, «Weakly differentiable functions», . 26

WILSON, D. 1972, «Asymptotic properties of nearest neighbor rules using edited data», *IEEE Trans. Syst. Man Cybern.*, vol. 2, p. 408–421. 8

WILSON, D. et T. MARTINEZ. 2000, «Reduction techniques for instance-based learning algorithms», *Machine Learning*, vol. 38, p. 257–286. 8

WU, J., J. REHG et M. MULLIN. 2003, «Learning a rare event detection cascade by direct feature selection», . 30

YANG, T., L. CAO et C. ZHANG. 2010, «A novel prototype reduction method for the k-nearest neighbor algorithm with k>1», p. 89–100. 24

YPMA, T. J. 1995, «Historical development of the newton-raphson method», *SIAM review*, vol. 21, p. 531–551. 15

ZHU, B., B. BAESENS, A. BACKIEL et S. VANDEN BROUCKE. 2017, «Benchmarking sampling techniques for imbalance learning in churn prediction», *Journal of the Operational Research Society*, vol. 69, p. 1–17. 13

ZIANG, J. 2003, «Knn approach to unbalanced data distributions : a case study involving information extraction», *Proc. Int'l. Conf. Machine Learning1 (ICML'03), Workshop Learning from Imbalanced Data Sets*. 6

Chapitre 3

Problématiques de la blessure et des évènements rares

Sommaire

3.1 Contexte général

La maitrise de l'état de forme des joueurs et de leur santé, plus particulièrement la blessure, est aujourd'hui un enjeu majeur pour les clubs de football professionnel. En effet, la blessure musculaire peut avoir un impact important pour l'athlète, qui peut voir ses capacités physiques diminuer, mais également pour le club, puisqu'elle peut entrainer une perte financière mais surtout une perte de puissance compétitive. Plusieurs études ont montré que les blessures affectaient les résultats et donc le classement des équipes de football [CRISTIANO et collab., 2012],[MARTIN et collab., 2013]. La blessure musculaire fait partie des évènements considérés comme rares, c'est à dire dont la probabilité d'apparition p est inférieure à 0.05. Des études épidémiologiques ont montré que pour 1000 heures d'exposition, entre 8-13 blessures survenaient JAN et collab. [2011], DANIEL et JAVIER [2017]. Cette rareté pose une difficulté statistique immédiate : comment étudier et donc prévoir un évènement lorsqu'il est rarement observé? Augmenter le nombre d'observations permet-il de répondre à cette problématique des évènements rares? Ou encore, faut-il faire en sorte que la base de données soit équilibrée pour ne plus avoir ce déséquilibre entre les classes? A la complexité statistique s'ajoute une problématique médicale et métier. En effet, le mécanisme de la blessure, même s'il a été largement étudié BAHR et KROSSHAUG [2005],ARNI et collab. [2004] est encore très méconnu.

3.1.1 Généralités sur la blessure

Commençons par définir la blessure dans le football. D'après la littérature, on définit une blessure musculaire sans contact comme une blessure survenant sans contact avec une autre joueur ou avec le sol et conduisant à un arrêt d'activité sportive d'au moins une journée. Durant la saison elle est définie comme telle par le staff médical du club. Le diagnostic de blessure est basé sur différentes caractéristiques comme le type (entorse par exemple), la localisation et le muscle atteint. La blessure peut être par exemple une simple crampe, qui permet un retour à l'entrainement rapide ou la déchirure musculaire qui est plus longue à guérir (jusqu'à plusieurs mois).

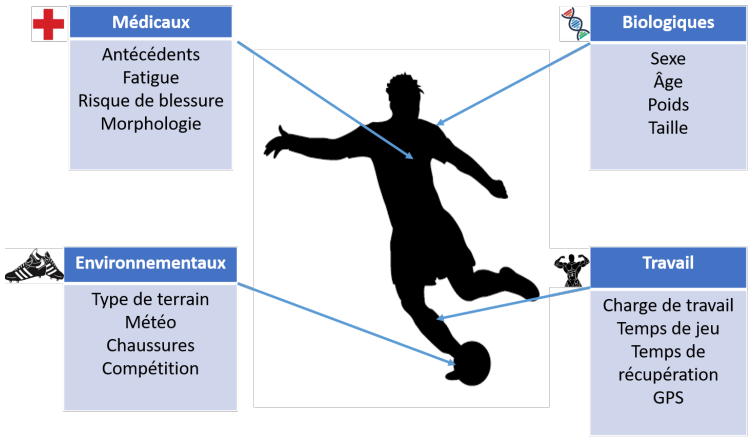


FIGURE 3.1 – Facteurs de risque

Les causes entrainant une blessure ne sont pas toujours bien connues et résultent souvent d'une combinaison de phénomènes ou facteurs de risque. Ces facteurs peuvent être de différents types. La figure 3.1 les résume :

— Les facteurs de risque environnementaux : ils regroupent l'ensemble des facteurs de risque externes au joueur, comme par exemple les conditions météorologiques durant un match, les chaussures utilisées, ou encore l'effet du type de terrain sur lequel évolue le joueur[EKSTRAND et collab., 2006]. Bien que certains de ces facteurs soient connus, dans la pratique, ils ne sont pas utilisables. En effet, il est souvent très difficile de connaitre la météo ou la qualité du terrain avant de débuter un match et ces facteurs peuvent évoluer au fur et mesure du match.

— Les facteurs de risque médicaux : ils sont constitués des antécédents médicaux (blessures antérieures, pathologie pré-existante éventuelle) ou la fatigue ou encore la morphologie du joueur. Certains de ces facteurs peuvent être connus et utilisés dans la prédiction et la prévention de blessure. C'est notamment le cas des blessures antérieures qui permettent de prendre en compte la récidive, comme l'a étudié Hägglund en 2006 MARTIN et collab. [2016]. Ce dernier a montré que les joueurs précédemment blessés avaient un risque plus élevé de se blesser au même endroit.

— Les facteurs de risque biologiques : Ils sont constitués des facteurs propres aux joueurs, comme le sexe, l'âge ou la taille. Ce sont des facteurs facilement accessibles, mais non modifiables et qui permettent une personnalisation du risque de blessure. Par exemple, il a été démontré que la différence de renforcement entre deux groupes musculaires (travail isométrique ou travail isocinétique) augmentait le risque de se blesser KONSTANTINOS et collab. [2010]

— Les facteurs de risque liés au travail : ce sont les facteurs liés à l'activité sportive du joueur comme le temps joué en match, le nombre de jours de récupération entre deux matchs, les performances sportives (vitesse moyenne, distance parcourue) et la charge de travail. Cette dernière est un paramètre d'importance, elle correspond au volume d'effort du joueur.

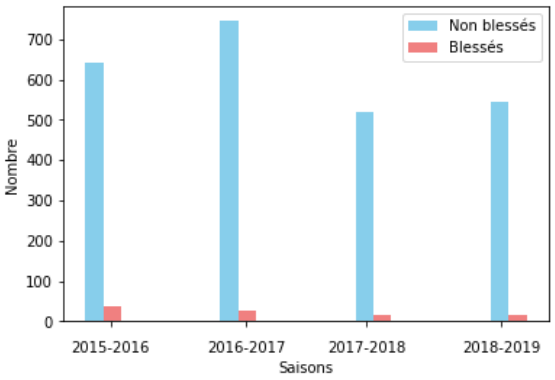
C'est sur ces derniers facteurs que les recherches sont les plus actives actuellement et notamment sur la charge de travail. Celle-ci peut être calculée en utilisant la méthode de Foster FOSTER [1998]. Elle repose sur un produit entre le temps d'exercice et la difficulté à l'effort du joueur, obtenue à l'aide d'un questionnaire. La méthode de Foster est la plus utilisée aujourd'hui et son impact a été plusieurs fois étudié dans l'apparition de la blessure. Nous pouvons notamment cité le travail de Hulin & Gabbett [BILLY et collab., 2015], qui ont montré que l'impact de la charge de travail sur l'apparition de la blessure n'était pas linéaire et qu'un sous-entrainement ou un surentrainement augmentaient les risques de blessure BILLY et collab. [2015], TIM J [2016] et JOHANN et TIM J [2016]. Il est également intéressant de noter que l'intensité et l'engagement des joueurs sont des facteurs de risque notables, puisque les blessures surviennent 5 fois plus en match qu'à l'entrainement.

Ces facteurs de risque montrent que pour prédire les blessures, il faut connaitre les joueurs et pouvoir les suivre au quotidien pour évaluer leur risque de blessure en permanence. Il est inévitable que la prédiction soit moins efficace sur les nouveaux joueurs arrivant dans le club, a fortiori en cours de saison. En effet, les informations sur ces joueurs dans le club sont inexistantes dans la base de données, il faut donc commencer l'apprentissage sur ce nouvel élément.

Enfin, il est important de signaler que toutes les blessures sans contact ne sont pas prévisibles. En effet, les causes peuvent être accidentelles (chute ou contact non signalé). Ces blessures non prévisibles sont l'une des problématiques que nous rencontrons. Elles apportent une information erronée et donc perturbent nos prédictions. En effet, ces blessures seront mal prédites. Elles diminueront donc l'efficacité factuelle de notre modélisation. Si dans d'autres domaines, il existe un moyen de déterminer ces données aberrantes, il n'y a ici aucun moyen de les détecter. Nous devons donc composer avec elles et tenir compte des problématiques décrites plus haut.

Un autre biais peut parfois provenir de l'équipe médicale. Puisque les joueurs sont encadrés par des professionnels médicaux, il est possible qu'après examen clinique ou selon d'après le ressenti personnel des médecins, certains joueurs jugés à risque ne jouent pas un match et donc ne se blessent pas. La base de données est donc constituée des blessures non détectables humainement et donc sont les plus compliquées à prédire. Un derniers biais peut venir de la méthode elle-même. Celle-ci étant utilisée par le club, il est possible que certains joueurs déclarés à risque élevé de blessure soient plus observés et protégés par le staff médical. Ces joueurs ne se blessant pas prennent en défaut la classification globale.

C'est à partir de ces connaissances que nos modèles de prédiction de blessure sans contact en match seront construits. A ce stade il est important de noter que rien n'a été précisé sur la personnalisation du risque de blessure. Nous présenterons certains points montrant l'importance d'étudier la blessure de façon individuelle, joueur pas joueur.



(a) Histogramme des blessures

Saison	Nombre de joueurs non blessés en match	Nombre de joueurs blessés en match	Taux de joueurs blessés en match (%)
2015-2016	745	38	5.60
2016-2017	641	28	3.62
2017-2018	543	16	3.00
2018-2019	520	16	2.86
Global	**2449**	**98**	**4.00**

(b) Description des blessures

FIGURE 3.2 – Nombre et incidence des blessures par saison

3.1.2 Le jeu de données

Le jeu de données que nous utilisons concerne une équipe de joueurs professionnels évoluant en ligue 1 du championnat français. Les données ont été récoltées durant 5 saisons de 2015 à 2019. Au total 52 joueurs différents ont évolué dans le club durant cette période (âge 27.1 ans ± 4.9). Les joueurs sont suivis quotidiennement. Leurs performances, efforts et blessures sont enregistrées pendant l'entrainement et également pendant les matchs.

Une observation de la base de données correspond à un match pour un joueur. Les joueurs ne participant pas au match ne sont pas renseignés. Les blessures sans contact durant les différentes saisons sont reportées dans la figure 3.2. La différence du nombre d'observations par saison est due aux participations et aux nombres de matchs dans les différentes compétitions. Le club a notamment participé aux compétitions européennes durant les saisons 2015-2016 et 2016-2017.

La première chose que l'on peut confirmer avec ce tableau, est le caractère rare de la blessure puisque sur l'ensemble des matchs on observe 4% de blessures. Une seconde chose intéressante à noter est la diminution du taux de blessures par saison. Cela peut être dû à un meilleur contrôle de l'effectif en utilisant le recueil des données et l'utilisation des prédictions de risque données par nos modélisations. Pour les deux dernières saisons, sans participation aux compétitions européennes, cela peut être la conséquence d'une intensité moindre et/ou à un nombre moins élevé de matchs joués.

Pour chaque blessure, l'équipe médicale renseigne le diagnostic de la blessure qui repose sur différentes caractéristiques tel que : le type, la localisation, le muscle touché et le temps d'absence du joueur. Ces informations sont résumées pour 1000 heures d'expositions dans le tableau 3.3, au global et pour la saison 2018-2019 sur lesquelles les méthodes seront testées. La première chose à noter est que l'incidence des blessures observées dans l'équipe correspond à celle reportée dans les articles parus sur l'incidence des blessures dans le football professionnel DANIEL et JAVIER [2017]; JAN et collab. [2011]. En effet, cette dernière a été évaluée entre 13 et 40 blessures pour 1000 heures d'exposition en match (30.5 pour le club), et entre 1.9 et 5.9 à l'entrainement (3.5 pour le club). Chez les joueurs de foot les principaux muscles touchés sont surtout les cuisses.

On remarque que les blessures à l'entrainement sont moins sévères (0.3 blessures mineures et 0.4 importantes pour 1000 heures d'exposition), que les blessures en match (0 mineures et 9.4 importantes). Les blessures en match entrainent donc plus de jours d'inactivité pour le joueur. Sur les 4 saisons, en moyenne, les blessures sans contact ont entrainé 3766 jours d'absence (24.14 ± 26.6 jours d'absence moyen par blessure)

Globalement, on observe une différence des blessures à l'entrainement et en match ce qui explique notre volonté de se concentrer sur les blessures en match qui sont plus importantes, car plus sévères. Après avoir présenté la blessure et les facteurs de risques associé, nous allons présenter les variables explicatives disponibles dans la base de données.

	Total (2015-2019)			Saison 2018-2019		
(Pour 1000 heures d'exposition)	Total (9.0)	Entrainement (3.5)	Match (30.5)	Total (6.1)	Entrainement (2.6)	Match (21.6)
Site						
Cheville	0.3	0.1	1.1	0.2	0.0	1.3
Cuisse	4.9	1.8	17.1	1.9	0.0	10.1
Genou	0.1	0.0	0.6	1.2	0.0	1.3
Hanche/Bassin/Aine	1.7	1.0	4.3	1.7	1.5	2.5
Jambe	1.9	0.6	7.4	2.1	1.6	6.3
Type de blessure						
Courbature/Fatigue musculaire	1.4	0.8	4.0	0.7	0.3	2.5
Entorse	0.4	0.1	1.7	0.5	0.0	2.5
Lésion musculaire et tendineuse	7.1	2.7	24.8	5.0	2.3	16.4
Gravité *						
Mineure (1 à 3 jours)	0.2	0.3	0.0	0.0	0.0	0.0
Légère (4 à 7 jours)	1.4	0.4	5.4	0.9	0.3	3.8
Modérée (8 à 28 jours)	4.9	2.4	14.8	3.3	2.0	8.9
Importante (plus de 28 jours)	2.2	0.4	9.4	1.7	0.3	7.6

* (Nombre de jours d'activité physique manqué)

FIGURE 3.3 – Classification des blessures sans contact pour 1000h d'exposition

Les variables explicatives renseignées dans la base de données permettant de prédire les blessures sans contact, elle se regroupent en 3 types. Nous présenterons dans un premier temps les variables d'effort. Elles caractérisent l'effort physique et le temps de repos des joueurs.

La charge de travail cumulée lors des 21 derniers jours avant le match

La charge de travail effectuée durant chaque activité physique comme les entraînements (collectifs ou individuels) ou durant les matchs est renseignée pour quantifier l'effort. Les activités thérapeutiques, comme les séances de kinésithérapie, de remises à niveau après une blessure sont également utilisées. La charge de travail en match est définie comme suit :

— si le temps de jeu du joueur est strictement inférieur à 65 minutes

$$\text{Charge} = (\text{temps de jeu} * 5.75) + 20.$$

— sinon

$$\text{Charge} = (\text{temps de jeu} * 5.75) + 48.75.$$

Pour les autres activités, elle est définie comme le produit de la durée, en minute, de l'activité et de sa difficulté (coefficient d'intensité entre 1 et 10), évaluée par le staff de l'équipe :

$$\text{Charge} = \text{Durée} * \text{Difficulté}.$$

Le tableau 3.1 suivant donne la charge pour une minutes de différent exercices :

Une séance complète d'entrainement est constituée de plusieurs activités. La charge d'une séance est donc égale à la somme des charges arbitraires de chaque activité. Ensuite il suffit de sommer les différentes charges durant les 21 jours précédant un match.

Le temps de jeu en match cumulé lors des 21 derniers jours précédant le match :

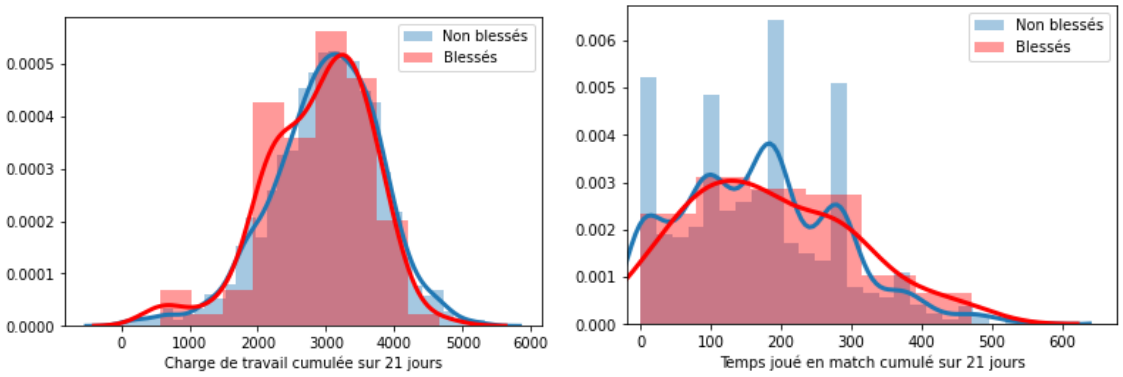
Pour chaque joueur, correspond aux nombres de minutes jouées par ce dernier, au cours d'un match, lors des 21 derniers jours.

Le temps de repos entre deux matchs

C'est le nombre de jours entre deux matchs joués par un joueur. Cette variable est qualitative. Les modalités représentent le nombre de jours de repos. Les modalités de cette variable sont :

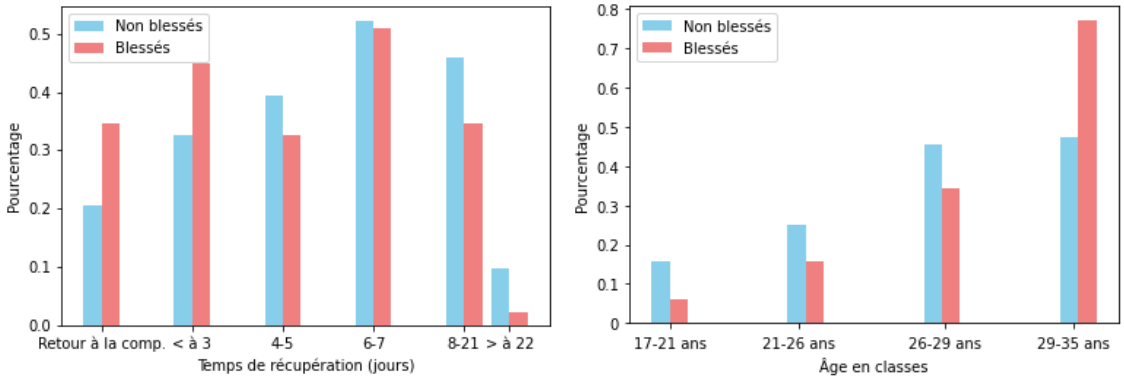
Exercices	Charge par minute
Course (échauffement)	1
Travail technique léger	2
Duels	9
Kiné : Musculation du haut du corps	2

TABLEAU 3.1 – Exemple de charge de travail par exercice



(a) Charge de travail cumulée sur les 21 derniers jours (b) Temps de jeu cumulée sur les 21 derniers jours

FIGURE 3.4 – Histogrammes et densités des variables d'effort



(a) Temps de récupération entre deux matchs

(b) Âge des joueurs en classes

FIGURE 3.5 – Histogrammes des variables qualitatives

— Moins de 3 jours
— Entre 4 et 5 jours
— Entre 6 et 7 jours
— Entre 8 et 21 jours
— Plus de 22 jours
— Retour à la compétition

La modalité "Retour à la compétition" indique que le joueur est de retour dans l'effectif après un arrêt des compétitions (reprise d'une nouvelle saison, trêve hivernale) ou suite à la perte de vue du joueur dans le club (période en équipe national).

L'histogramme et la densité des variables charge de travail et temps de jeu sont présentés dans le figure 3.4a et 3.4b.

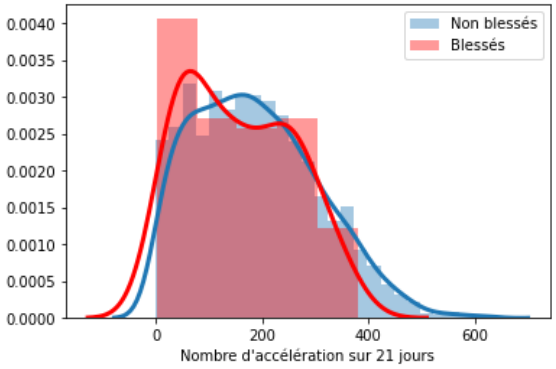
Sur le graphique du temps de jeu on remarque 4 pics :

— Le premier correspond au remplaçant "joker" jouant peu (mois de 30 minutes sur 21 jours).
— Le second aux joueurs jouant un match sur les 21 jours. Ce sont des remplaçant "habituel" qui rentrent en fin de matchs, ou des joueurs permettant de mettre au repos les titulaires pour un match.
— Le troisième pic, est constitué des titulaires qui jouent quasiment touts les matchs (plus de deux matchs en 3 semaines)
— Le quatrième pic correspond aux titulaires indiscutables.

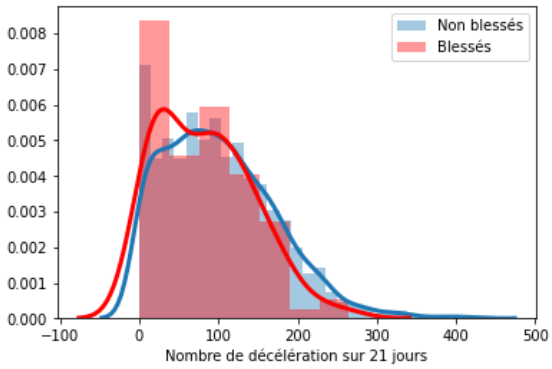
Pour la charge de travail et le temps de jeu la différence en densité entre les blessés et les non-blessés est quasi inexistante.

Concernant le temps de récupération (figure 3.5a), on note que les blessures surviennent plus dans les modalités "moins de 3 jours" et "6-7 jours". C'est à dire, lorsque les matchs s'enchainent rapidement et que le joueur a peu de temps pour se reposer, mais également pour la modalité "retour à la compétition", donc après une rupture avec la compétition en club, et donc un changement d'habitude. Que ce soit pour blessure ou pour une reprise de saison, le joueur a besoin de retrouver son état de forme. Suite à une période en équipe nationale, l'entraineur et les conditions d'entrainement sont modifiés, ce qui peut perturber l'état de forme d'un joueur.

Le seconde catégorie de variables sont les données obtenues à l'aide de GPS. A l'entrainement, chaque joueur est équipé de GPS ce qui permet d'obtenir leurs performances comme leur vitesse moyenne, le nombre d'accélération ou de décélération.



(a) Nombre d'accélération cumulé sur 21 jours



(b) Nombre de décélération cumulé sur 21 jours

FIGURE 3.6 – Histogrammes et densités des variables GPS

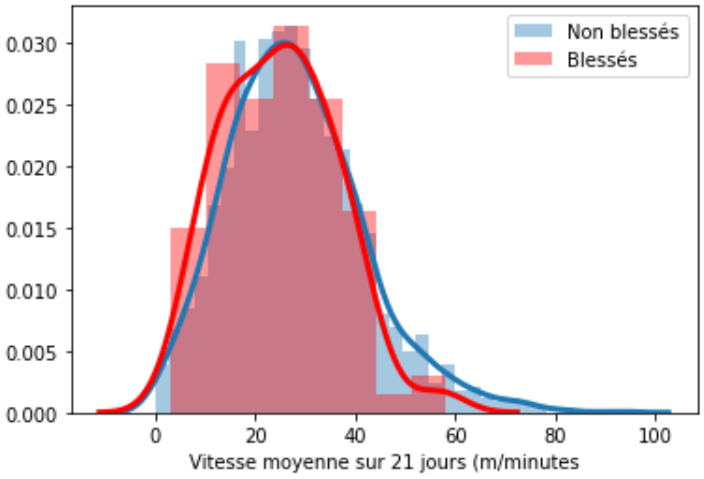


FIGURE 3.7 – Histogramme et densité de la vitesse moyenne lors des 21 derniers jours (m/minutes)

Le nombre d'accélérations cumulé sur 21 jours : nombre d'accélérations effectuées à une vitesse supérieure à 21 km/h cumulé lors les 21 derniers jours

Le nombre de décélérations cumulé sur 21 jours

La vitesse moyenne sur 21 jours : vitesse moyenne mesurée à l'entrainement sur les 21 derniers jours.

Les figure 3.6 et 3.7 présentent l'histogramme et la densité des 3 variables GPS pour les joueurs blessés et non-blessés. Ces graphiques montrent un léger décalage entre les deux catégories. Plus les joueurs réalisent des accélérations et décélérations moins les blessures surviennent. C'est d'autant plus marqué que lorsque des accélérations/décélérations ne sont presque pas pratiquées à l'entrainement (première barre) les blessures sont bien plus fréquentes.

Enfin la dernière catégorie représentent les facteurs de risque personnel au joueur. La première variable est l'indice de récidive, qui mesure la fragilité du joueur à la suite d'une blessure. La seconde est l'âge du joueur. Finalement la troisième est l'identifiant du joueur qui permet de personnaliser le risque joueur par joueur. Nous en verrons l'importance et l'utilité dans la suite.

Indice de rechute : L'indice de rechute mesure le risque de rechute du joueur en fonction du

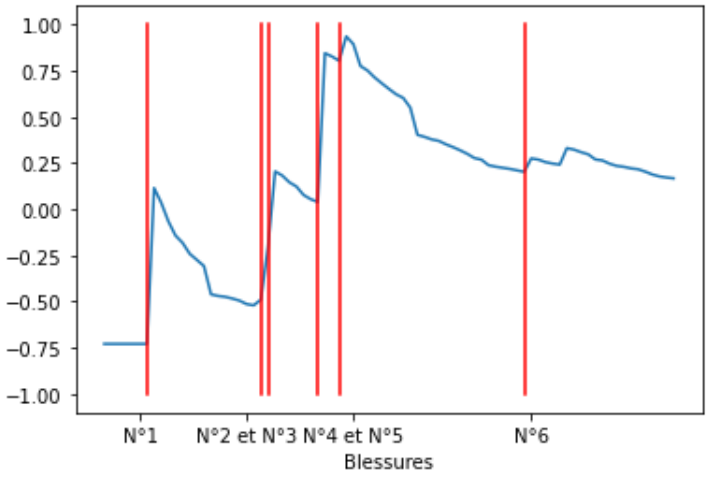


FIGURE 3.8 – Évolution de l'indice de rechute normalisé au cours du temps chez un joueur

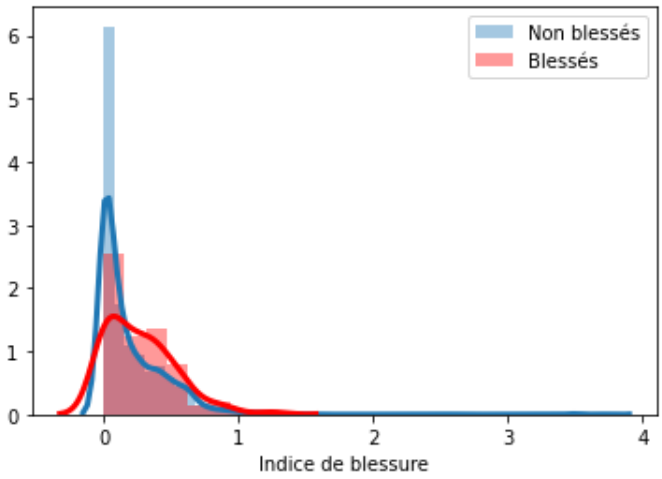


FIGURE 3.9 – Histogramme et densité l'indice de rechute

risque global de l'équipe. Il est calculé de la manière suivante :

$$\text{Indice de rechute} = \frac{\text{Nombre de jours indisponibles pour cause de blessures}}{\text{Nombre de jours disponibles dans l'effectif}}$$

Plus le joueur a été absent pour cause de blessures plus l'indice de rechute sera haut. Après une blessure, l'indice est élevé, indiquant que le joueur est à risque. Ensuite, au fur et à mesure du temps, l'indice diminue, indiquant un risque de rechute d'une blessure plus faible. Pour construire cet indicateur, les blessures à l'entrainement comme les blessures en matchs sont utilisées. Cela signifie que sur cet indicateur, les blessures à l'entrainement entrainent les mêmes variations de l'indicateur que les blessures en match. Cela est justifié par le fait qu'une rechute d'une blessure à l'entrainement peut se produire en match. La figure 3.8 représente, pour un joueur donné, l'évolution temporelle de l'indice de rechute, les blessures en match étant indiquées par une ligne rouge. On observe bien une hausse de l'indicateur après chacune de ces blessures. Il faut noter que pour les nouveaux joueurs arrivant dans le club, une blessure survenant rapidement entraine une hausse importante de l'indice puisque que son nombre de jours de disponibilité (dénominateur de l'indice) est faible. On note sur la figure 3.9 que les joueurs qui se blessent ont un indice légèrement plus élevé que les joueur ne se blessent pas.

L'âge des joueurs en classe représente l'âge des joueurs regroupé en 4 classes : [17-21],]21,26],]26-29] et]29-35] ans. L'histogramme et densité est présenté dans la figure 3.5b. Il est intéressant de noter que les blessures surviennent principalement chez les joueurs ayant entre 29 et 35 ans. Alors que le risque est bien plus faible chez les jeunes joueurs (17 à 21 ans).

Identifiant du joueur correspond à un numéro du joueur. Ils ne changent pas pendant toute la durée de son contrat au sein du club.

La prise en compte seul d'une variable n'est pas discriminante. Quelque soit la variable, la différence entre blessés et non-blessés est quasiment inexistante, comme le montre la diagonal de la figure 3.10. Maintenant que l'ensemble des variables ont été présentées, nous leur allons expliquer le traitement que nous avons appliqué pour les utiliser.

Traitement des variables

Dans la suite les variables seront utilisées dans différentes méthodes de modélisation. Il est important d'introduire les modifications que nous avons apportées ici. Pour commencer la base ne contient aucune donnée manquante pour les joueurs ayant participé à un match. Pour faciliter l'interprétation, les variables continues ont été centrées et réduites.

Cela nous assure que chaque variable ait la même moyenne et le même écart-type, nous permettant ainsi de pas tenir compte de l'échelle ce qui facilitera l'interprétation. Concernant les variables qualitatives, elles ont été transformées en variables indicatrices POLISSAR et DIEHR [1982]. A partir d'une variable X prenant les modalités $m_1, m_2, ..., m_k$, nous construisons k variables X^d de tel sorte, que pour tout individu j, $X_j^d = 1$ si $X_j = m_d$, 0 sinon. Aucune autre modification de la base de données n'a été effectuée. La base, une fois modifiée, restera donc la même pour l'ensemble des modélisations effectuées dans ce travail.

Analyse bidimensionnelle

Nous l'avons vu dans la section précédente, il est impossible d'expliquer la survenue d'une blessure par une seule variable. Aucune n'étant discriminante. Nous avons donc cherché à déterminer si un groupe de variables pouvait l'expliquer. Pour cela, nous avons, dans un premier temps, regardé graphiquement les variables deux à deux (figure 3.10). Sur cette "matrice" chaque graphique représente le nuage de points du croisement de deux variables, les blessures en orange, les non-blessures en bleu. Pour donner un exemple le graphique ligne 2 et colonne 1 représente le nuage de points croisant la charge de travail et le temps de jeu. Sur la diagonale, les densités des blessés et non-blessés sont tracées pour la variable concernée. Ces graphiques ont déjà été étudiés plus haut (section 3.1.2.

Sur l'ensemble de ces croisements aucun couple de variables ne permet de séparer les joueurs blessés des non-blessés. On voit clairement que nos données ne sont pas séparables et que seule la combinaison de plusieurs variables pourrait expliquer les blessures. Globalement la distribution des joueurs blessés est équivalente à celle des joueurs non-blessés. Nous avons donc en plus de la problématique des évènements rares un jeu de données relativement compliqué à prédire.

Nous voyons qu'il est nécessaire d'augmenter le nombre de dimensions pour essayer de comprendre le mécanisme et améliorer la prédiction des blessures. Nous avons donc utilisé des méthodes de modélisation multidimensionnelle, permettant d'améliorer la connaissance de nos données, comme les arbres de classification, générant des règles de décision, ou la régression logistique permettant d'obtenir les coefficients β_i associé à chaque variable. Nous verrons dans la prochaine section les méthodes de modélisation que nous avons utilisées pour prédire le risque de blessure.

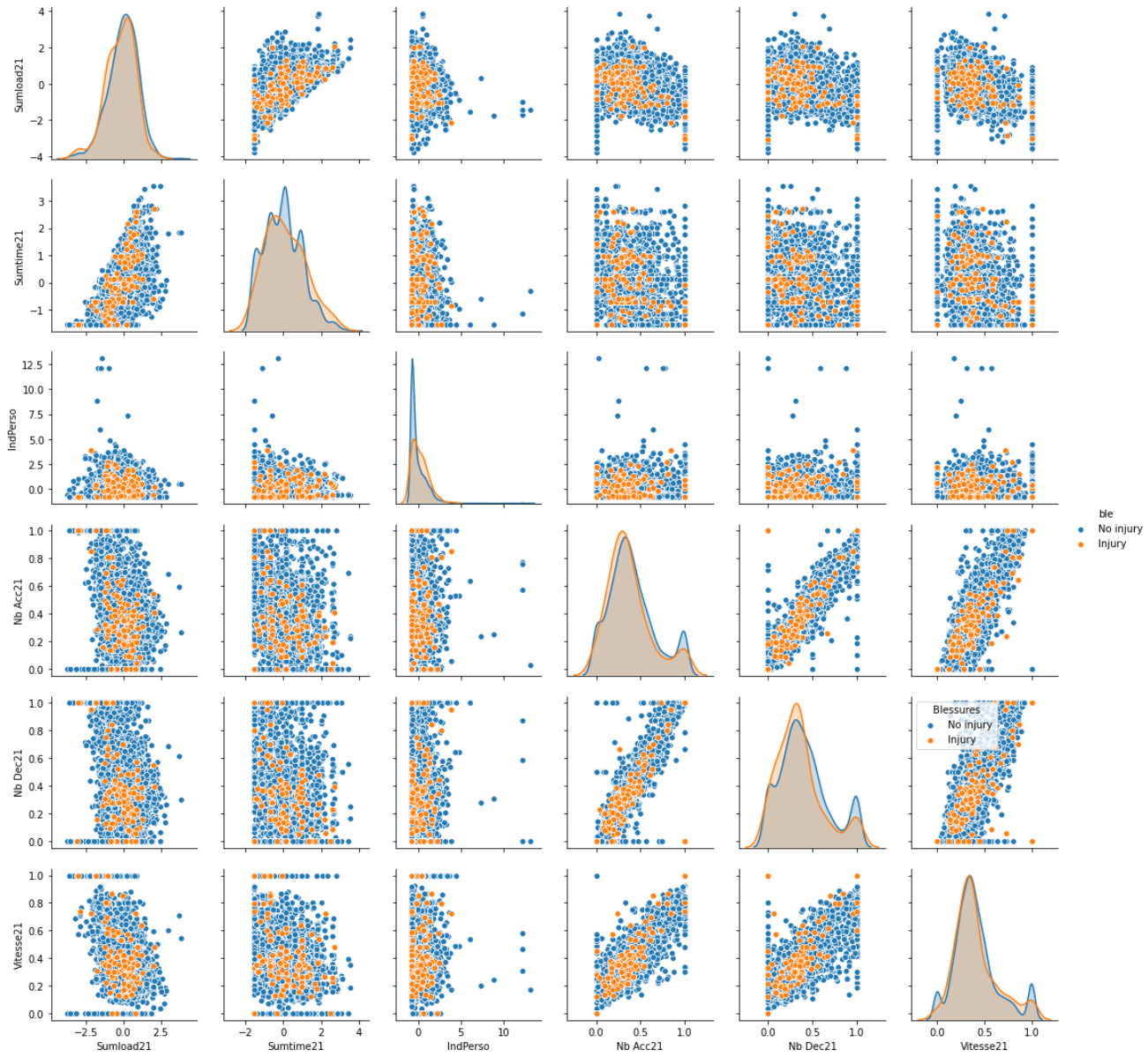


FIGURE 3.10 – Analyse bivariée des facteurs de risque

3.2 Modélisation de la blessure chez le joueur de football professionnel

Dans ce paragraphe, nous présenterons les effets des méthodes présentées dans le Chapitre 2 sur le jeu de données réelles et déséquilibrées des blessures sans contact de joueurs de foot professionnels. Chaque modèle sera évalué sur la prédiction de la saison 2018-2019, en utilisant la méthode de validation longitudinale présentée dans la section 2.4.2.

Nous présenterons les méthodes utilisées et les modélisations montrant les meilleures résultats. Pour les comparer, dans un premier temps nous utiliserons la courbe ROC et l'AUC, qui permettent d'avoir une vue générale du modèle. Ensuite, nous compléterons avec le meilleur indice de Peirce, la sensibilité et la spécificité. Pour commencer cette section nous présentons le premier modèle construit, qui permettait de prédire un risque de blessure.

3.2.1 Modélisation par la régression logistique (modèle de référence)

Dans cette partie nous présentons les deux premiers modèles que nous avons construis. Ils nous serviront de référence pour la comparaison des modélisations.

Modèle initial

Le modèle initialement proposé était construit à partir d'une régression logistique en considérant les variables explicatives suivantes :

— Âge en classe
— Charge de travail
— Temps de jeu
— Indice de rechute
— Temps de récupération entre deux matchs

Il permettait d'obtenir le risque qu'un joueur se blesse. Cette probabilité était communiquée aux encadrants de l'équipe afin de les aider à gérer le risque de blessures des joueurs. Il permettait dans certains cas de lever des interrogations sur l'état physique d'un joueur et dans d'autres de confirmer le doute de l'équipe médical.

La courbe ROC de ce modèle est présentée dans la figure 3.11.

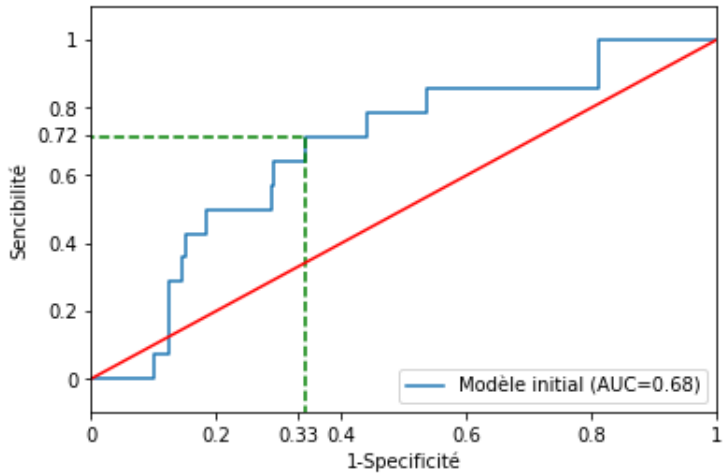


FIGURE 3.11 – Courbe ROC du modèle initial

L'AUC obtenue est de 0.68, pour un indice de Peirce (point en verte sur le graphique) de 0.39. Les sensibilité et spécificité correspondantes à cet indice sont 0.72 et 0.68. Le γ permettant cette classification étant à 0.0314. Ce modèle permet donc de classer correctement 72% des blessures et 0.68% des non-blessures. Il est intéressant de noter que les probabilités sont faibles. En effet, en moyenne, pour les joueurs blessés, elles sont de 0.046 et pour les non-blessés de 0.035.

A partir de ce modèle nous avons souhaité améliorer notre connaissance des facteurs de risque. L'un des avantages de la régression logistique est la possibilité de connaitre les coefficients β_i affectés à chaque variable. Ces coefficients ou les odds-ratio associés peuvent nous renseigner sur l'effet "protecteur", ou "aggravant" du risque de blessure.

Définition : l'odds-ratio est le rapport des cotes des probabilités de se blesser pour ceux qui présentent une variable X d'une part et ceux qui ne l'ont pas d'autre part. Il existe alors trois possibilités :

1. Odd-ration < 1 : l'exposition diminue la probabilité de blessure → effet protecteur.
2. Odd-ration = 1 : la survenue de blessure est indépendante de l'exposition
3. Odd-ration > 1 : l'exposition augmente la probabilité de blessure → effet aggravant.

Les odds ratio obtenus ainsi que leur intervalle de confiance à 95% sont présentés sur la figure 3.12. Commençons par nous intéresser aux variables quantitatives.

Le temps de jeu est un facteur de risque pour le joueur. Plus l'athlète a cumulé du temps de jeu sur les 3 dernières semaines plus son risque de blessure est élevé. Pour chaque unité supérieure de cette variable, le risque est multiplié par 1.35. Dans cette situation une unité correspond à 100 minutes de jeu soit un peu plus d'un match. L'indice de récidive semble également engendrer une hausse de risque (Odds-ratio à 1.62) mais son intervalle chevauchant la valeur 1, nous ne pouvons pas l'affirmer. Au contraire, la charge de travail a un rôle protecteur. En effet l'augmentation d'une unité de cette variable diminue le risque de 33% (Odss-ratio à 0.67).

Concernant les variables qualitatives, commençons par la variable temps de récupération entre deux matchs, la modalité de référence est "moins de 3 jours". On observe que globalement un temps de repos supérieur à 3 jours est protecteur pour le joueur. Même si la seule modalité pour laquelle nous pouvons l'affirmer est la modalité "plus de 22 jours de récupération" puisque son intervalle de confiance ne chevauche pas 1. Enfin pour la variable âge, dont la modalité de référence est "17-21 ans", plus les joueurs vieillissent plus l'odds-ratio augmente, la modalité la plus à risque étant "29-35 ans". Le risque de blessure est 7 fois plus important pour les joueurs de cette catégorie que pour les joueur ayant entre 17 et 21 ans.

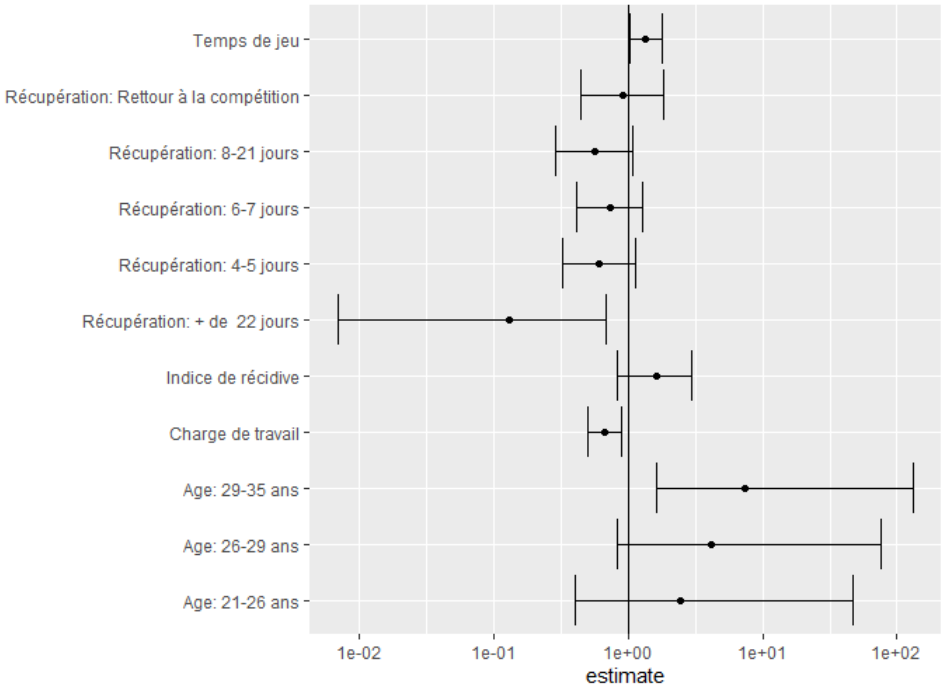


FIGURE 3.12 – Odds-ratio du modèle du modèle initial

Second modèle (modèle de référence)

Avec l'acquisition de ces connaissances, nous avons décidé d'améliorer le modèle initial par l'ajout et le remplacement de certaines variables. En effet les résultats du premier modèle étaient faibles. Il fallait donc l'améliorer. Pour cela nous avons ajouté des variables GPS permettant de tenir compte de la performance du joueur lors des entrainements et de remplacer la variable âge par le numéro de joueur. Ce changement nous permet de tenir compte des particularités de chaque athlète et d'obtenir un modèle plus individualisé. Ce dernier est donc toujours construit en utilisant la régression logistique mais avec les variables :

— Identifiant du joueur

— Charge de travail

— Temps de jeu

— Indice de rechute
— Temps de récupération entre deux matchs
— Vitesse moyenne
— Nombre de décélérations
— Nombre d'accélérations

La figure 3.13 présente la courbe ROC de ce modèle. L'AUC est meilleure que celle du modèle initial puisqu'il est de 0.733. L'indice de Peirce (0.51), la sensibilité (0.78) et la spécificité (0.71) sont également améliorées. Avec ces nouvelles variables nous obtenons donc un modèle capable de mieux prédire les blessures, mais permettant également de diminuer le nombre d'erreurs de classification global. Il faut noter que le γ permettant d'obtenir ces résultats est de 0.020, que les probabilités moyennes pour les blessures est de 0.046 comme pour le premier modèle . En revanche la moyenne des probabilités des non-blessures est de 0.020, ce qui est bien inférieur au premier modèle (0.035). La seconde modélisation permet donc de mieux séparer les données. C'est donc à partir de ces variables explicatives que nous essayerons d'améliorer la modélisation.

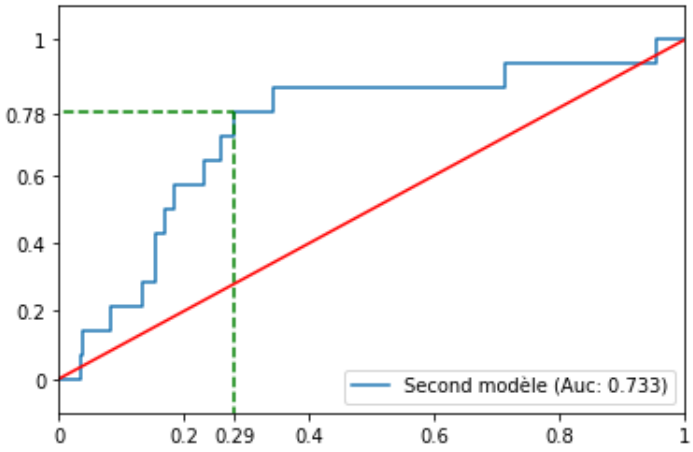


FIGURE 3.13 – Courbe ROC du deuxième modèle

Nous avons analysé les odds-ratio obtenus par ce nouveau modèle (figure 3.14) de la même manière que pour le premier modèle. Nous avons décidé de ne présenter uniquement les l'odds ratios de 4 joueurs pour ne pas surcharger la figure. Les variables utilisées dans le premier modèle présentent les mêmes caractéristiques.

Pour les variables GPS, l'impact du nombre d'accélérations et de décélérations n'est pas simple à évaluer. En revanche la vitesse moyenne est une variable aggravante puisque sa valeur est à 1.47. Pour la variable joueur, on observe qu'il existe une différence de risque entre les joueurs. En effet le joueur 2 a presque 7 fois plus de risque de se blesser que le joueur 1, qui est la modalité de référence. Cela confirme l'importance de la personnalisation du risque et donc de l'ajout de cette variable.

Cette analyse nous donne des informations essentielles sur les variables. Il est important de noter qu'une des limites des odds-ratio est leur effet linéaire. En effet, la charge de travail est protectrice, ce qui conduirait à augmenter la charge des joueurs pour éviter les blessures. Cependant il est naturel de penser qu'une charge trop importante conduirait à une blessure. Cette non linéarité de la charge a été démontrée dans BILLY et collab. [2015], TIM J [2016]. Il y a donc des zones à risque (faible ou trop haute charge), et des zones de sécurité. Pour mettre en évidence ces zones de sécurité, il faudra utiliser d'autres méthodes. Comme par exemple les arbres de classification. Ces derniers permettent de générer des règles de décision, en segmentant les variables et donc de pouvoir retrouver ces zones de risque.

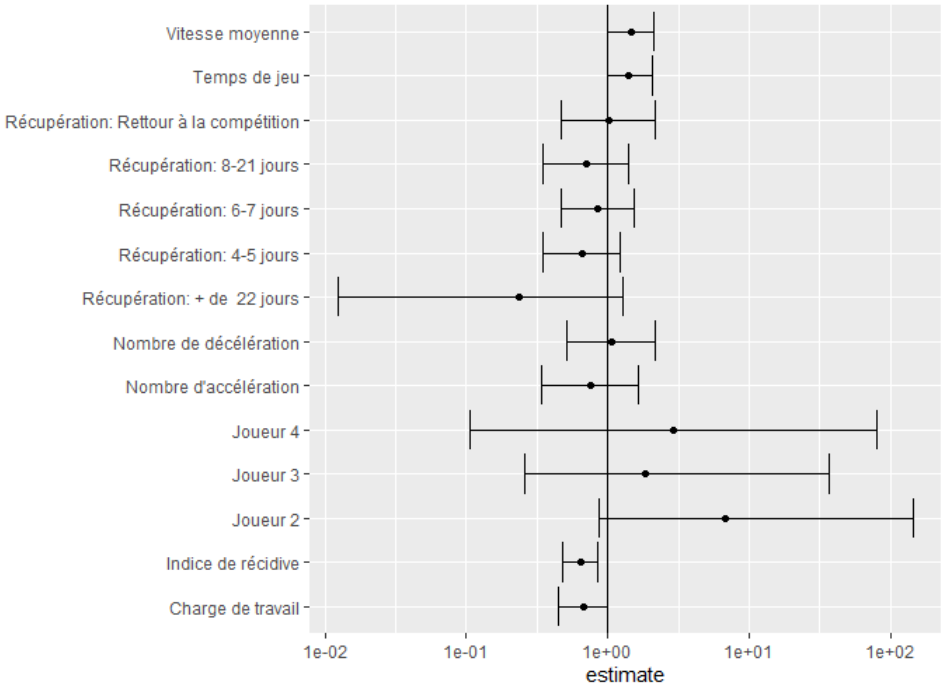


FIGURE 3.14 – Odds-ratio du deuxième modèle

3.2.2 Modélisation par les arbres de classification

Les arbres de classification utilisés comme méthode de modélisation présentent l'avantage d'avoir peu de paramètres à calibrer. Les deux plus importants sont le nombre maximal d'observations dans les feuilles terminales, et la profondeur de l'arbre final souhaité. Ces deux paramètres permettent notamment d'éviter d'avoir des arbres trop complexes qui ne soit pas généralisables.

Les arbres de classifications, en présence d'évènements rares, montrent souvent des résultats faibles. Ils sont souvent choisis en tant que "weak learner" dans des méthodes d'assemblage. Pour évaluer les arbres, nous avons cherché à trouver les paramètres les plus optimaux. La figure 3.15 présente l'AUC en fonction du nombre d'observations maximales dans les feuilles finales. On observe un plateau entre les profondeurs 5 et 8 où l'AUC est meilleur avec une valeur maximale pour la profondeur de 5, donnant un AUC à 0.684. Nous utiliserons donc cette profondeur. Il est intéressant de noter qu'une augmentation de la profondeur n'implique pas systématiquement une amélioration des résultats.

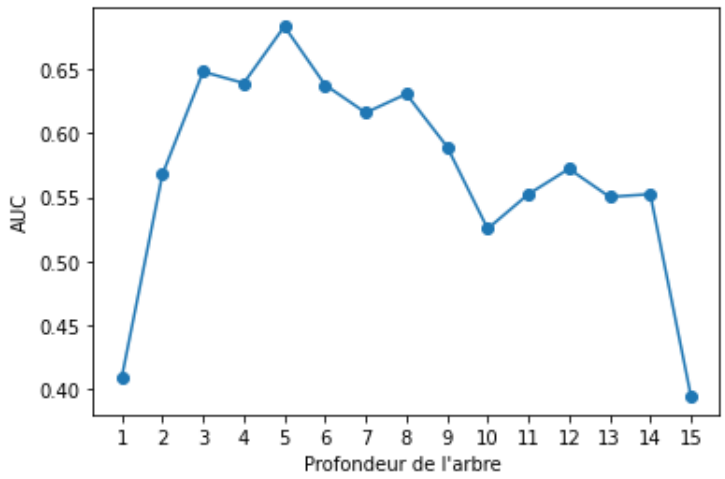


FIGURE 3.15 – Effet de la profondeur des arbres sur l'AUC

Règles de décision

L'avantage des arbres de classification est qu'ils segmentent l'espace d'arrivé et permet d'obtenir visuellement l'influence des variables et de leurs modalités. Les arbres que nous obtenons sont difficilement présentables. Cependant nous pouvons rapporter quelques règles intéressantes. Bien que nous ayons choisi des feuilles terminales de taille minimale 5, aucune n'est constituée uniquement de blessés. La meilleure est constituée de 4 non-blessés pour 6 blessés.

La première règle que nous présentons classe les individus blessés avec une probabilité de 0.58. Elle peut être décrite comme suit :

1. indice de rechute > -0.24,
2. vitesse moyenne > 0.53,
3. nombre d'accélérations entre 0.47 et 0.88,
4. temps de jeu entre -1.23 et -0.70,
5. charge de travail entre -1.23 et -0.65.

De manière plus factuelle, le risque de blessure augmente pour les joueurs ayant un indice de rechute au alentour de la moyenne, une vitesse moyenne et un nombre d'accélération élevé mais un temps de jeu et une charge peu élevé sur les 21 derniers jours.

La seconde règle que nous présentons classe "non-blessés" : la probabilité de "non-blessure" obtenue est 100%. En effet sur les 89 joueurs suivant cette règle aucun ne s'est blessé. Pourtant elle est quasiment identique à la première, les seuls changements sont sur les ligne 4 et 5 :

1. indice de rechute > -0.24,
2. vitesse moyenne > 0.53,
3. nombre d'accélérations entre 0.47 et 0.88,
4. temps de jeu > 0.70,
5. charge de travail > -1.04.

L'augmentation du temps de jeu et de la charge de travail sur les 21 derniers jours, conduit à une protection.

L'une des questions que l'on peut se poser est de déterminer si ces zones de protection ou de risque sont identiques pour l'ensemble des joueurs. Pour mettre en évidence ce phénomène, nous pouvons présenter deux règles impliquant les joueurs. Le joueur 1 a une zone à risque (probabilité de associé de 41%, 5 blessures pour 7 non-blessures) définie par :

— vitesse moyenne <0.27,

— temps de jeu >0.12,

— charge de travail <0.85,

alors que pour le joueur 2, la zone à risque (avec une probabilité à 30%, 3 blessures pour 7 non-blessures) est définie par :

— nombre de décélérations <0.395,

— temps de jeu entre -0.93 et 0.95,

— charge de travail <-0.64.

Il existe une différence notable entre les joueurs puisque le premier a une zone à risque lorsque le temps de jeu augmente mais que la charge est plus faible. Alors que pour le second est à risque s'il joue un temps de jeu moyen mais que la charge de travail est basse. De plus si l'on change simplement la dernière ligne du joueur 2 en augmentant la charge de travail :

— nombre de décélérations <0.395,

— temps de jeu entre -0.93 et 0.95,

— charge de travail >-0.64.

Il passe dans une zone de sécurité avec (37 non-blessures pour aucune blessure).

A travers ces différentes analyses, nous avons pu mettre en évidence la complexité du jeu de données. La non séparabilité des deux classes, que ce soit avec une, deux variables. Les modélisations classique ont également des difficultés à obtenir de bon résultats. Que ce soit avec la régression logistique ou avec les arbres de classification. De plus nous avons vu qu'il existe des liens non linéaire entre les blessures et les variables explicatives comme la charge de travail. Il va donc falloir utiliser des méthodes plus complexes pour permettre d'améliorer ces résultats. Nous utiliserons par exemple les réseaux de neurones, qui permettent d'obtenir une modélisation non linéaire, ou encore les SVM et l'astuce des noyaux.

3.2.3 Modélisation par les réseaux de neurones

Les réseaux de neurones sont aujourd'hui utilisés pour trouver une solution à de nombreux problèmes, reconnaissance d'images, prédiction de risques. Ils peuvent résoudre les problèmes linéaire comme non linéaire (en utilisant des fonctions d'activation non linéaire), ce qui leur confère une grande flexibilité. Leur second avantage est leur capacité d'approximer une fonction inconnue. Un réseau de neurones suffisamment grand avec une couche cachée peut approximer n'importe quelle fonction différentiable. Cela en fait une des méthodes les plus utilisées aujourd'hui. De plus avec les logiciels existants, il est relativement facile de construire son propre réseaux de neurones. Cependant, il est compliqué de trouver celui qui donnera les meilleurs résultats. En effet, les possibilités sont multiples, d'une part pour la construction du réseau, que ce soit dans le nombre de couches, de neurones par couche où la fonction d'activation et d'autre part dans l'apprentissage par le nombre de fois où la base va être utilisée ou le nombre d'observations utilisés à chaque lots. Rappelons que la mises à jour des poids se fait de la manière suivante :

— la base d'apprentissage est divisée en lots,

— les lots sont utilisés un par un pour calculer l'erreur final et effectuer la mise à jour des poids par descente du gradient,

— une fois tous les lots utilisés, on dit qu'une *epoch* est complétée,

— ce processus est répété jusqu'à atteindre le nombre d'epoch souhaité.

D'après cet algorithme, il est donc possible d'utiliser plusieurs fois les mêmes observations. Cela est même conseillé et fait parti des paramètres à optimiser. L'architecture du réseau va dépendre directement du nombre de fois où la base de données a été complètement utilisée.

Pour visualiser l'impact de l'epoch, nous avons utilisé un réseau de neurones à une seule couche cachée de 64 neurones avec une fonction d'activation sigmoïde. Les résultats des différentes courbes ROC sont présentés dans la figure 3.16. Pour construire ces courbes, les poids initiaux étaient initialisés à la même valeur, donc seul le nombre d'epoch varie.

Au cours de nos recherches c'est un paramètre auquel nous ferons particulièrement attention puisque l'apprentissage est mis à jour à chaque prédiction d'un nouveau match. En effet, avec les réseaux de neurones, une fois le modèle construit, si l'on veut ajouter des données d'apprentissage, il est possible d'apprendre uniquement sur ces données, contrairement à d'autre méthodes comme la régression logistique qui nécessite un nouvel apprentissage avec la nouvelle base (ancienne données plus les nouvelles). Cette caractéristique est pratique dans notre cas puisque les données sont renseignées à chaque séance d'entrainement. Le modèle peut donc être rapidement mis à jour. Cependant cette fonctionnalité a tendance à donner plus d'importance aux dernière données d'apprentissage. En effet, les poids sont mis à jour selon l'erreur commise sur ces données. Il faudra donc trouver le bon epoch qui permet de ne pas être en sur-apprentissage sur les données les plus récentes.

En tenant compte de toutes ces problématiques, nous présentons les deux meilleures réseau que nous avons construis. Le premier est un réseau à 2 couches cachées :

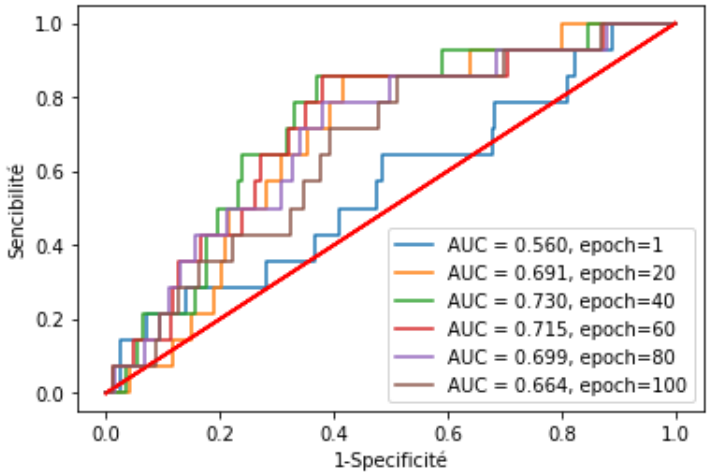
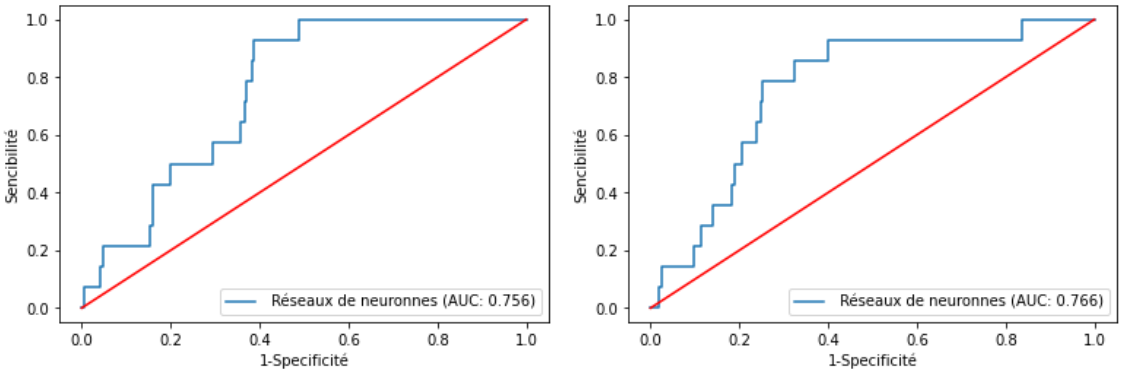


FIGURE 3.16 – Effet du nombre d'epoch sur l'AUC des réseaux de neurones



(a) Réseau à une couche cachée

(b) Réseau à 2 couches cachées

FIGURE 3.17 – Courbes ROC des deux réseaux de neurones

— Première couche de 8 neurones, fonction d'activation : RELU
— Seconde couche de 8 neurones, fonction d'activation : tangente hyperbolique
— Couche de sortie 1 neurone, fonction d'activation : sigmoïde.

Le second à seulement une couche caché :

— Première couche de 8 neurones, fonction d'activation : tangente hyperbolique
— Couche de sortie 1 neurone, fonction d'activation : sigmoïde.

Ce sont deux réseaux de neurones peu profond et qui ont peu de neurones. Les courbes ROC obtenus sont présentés 3.17. Le premier réseau a 3 couche (3.17a) a un AUC à 0.756, le second, légèrement meilleure a un AUC à 0.766. En prenant le meilleur point, il permet de classer correctement 78% des blessures et 77% des blessures. C'est donc le meilleur modèle trouvé jusqu'ici.

3.2.4 Modélisation par les support vector machines

Les SVM ont montré une bonne efficacité dans de nombreux domaines, notamment grâce l'utilisation de l'astuce des noyaux. Cette dernière permet de reconsidérer les problèmes dans des espaces de dimensions supérieures et ainsi trouver une séparation linéaire. Nous l'avons vu, à travers les arbres de classification ou la régression linéaire, sans modifier l'espace de départ, il n'est pas possible de séparer les classes. L'astuce des noyaux peut donc nous aider dans notre modélisation du risque de blessure.

L'inconvénient principal des SVM est la recherche du noyau et des paramètres optimaux associés. En effet, il existe de nombreux noyaux possible. Il est même possible de les combiner entre eux. Le nombre de paramètres varie d'un noyau à un autre : un seul paramètre pour le noyaux polynomial, 3 pour le noyaux sigmoïde. Habituellement la recherche des paramètres optimaux se fait par validation croisée en testant le croisement des différents paramètres. Cette technique relativement rapidement, ne peut pas être utilisée ici. En effet comme nous l'avons expliqué plus haut, la méthode de validation croisée n'est pas appropriée aux jeux de données longitudinales. La recherche des paramètres optimaux se fera donc en utilisant la validation longitudinale. Ce processus entraine une difficulté supplémentaire : le temps de calcul. En effet, l'obtention des résultats d'une paramétrisation peut prendre plusieurs minutes. Au vu de ces problématiques nous présenterons les meilleurs résultats que nous avons eu avec les SVM.

Le premier noyau utilisé est le noyaux linéaire. La courbe ROC est présentée dans la figure 3.18. Le résultat est faible. Le modèle est quasiment équivalent à la classification aléatoire. Ce résultat confirme, que dans l'état actuel des données, les blessures et les non-blessures ne sont pas séparable. Nous allons donc utiliser des noyaux plus complexe.

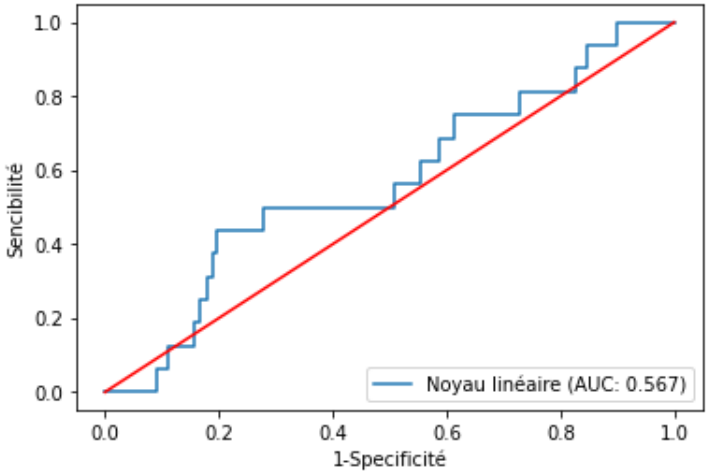


FIGURE 3.18 – Courbe ROC de la modélisation par SVM : noyaux linéaire

Le meilleur résultat obtenu après optimisation des paramètres est le noyau sigmoïde avec les coefficients $c = 2.25$ et $\alpha = 0.05$. L'AUC est alors de à 0.706 (figure 3.19). Bien que les résultats soit nettement meilleurs qu'avec le noyaux linéaire, ils restent cependant dans l'ordre de grandeur des résultats trouvés précédemment.

Les SVM permettent aussi de modifier la contrainte en jouant sur le coût C des mal classés. Plus C est important, plus le coût des mauvais classements est important. Choisir C élevé donnera des marges plus fines qui améliorera le classe de tous les points. Cependant, c'est également donner plus d'importance aux bruits (ou aux blessures accidentelles). Un C faible produira des marges plus grandes. Le coût initiale est 1.

Après recherche, les paramètres donnant le meilleur AUC sont : noyau sigmoïde, $c = 0.2$, $\alpha = 0.09$ et un coût de mauvaise classification C = 5 qui donne un AUC de 0.723 (figure 3.20). C'est le meilleur résultat obtenue avec les SVM jusqu'à présent.

Cette amélioration, grâce à la modification des coûts de mauvais classements, nous montre qu'il est encore possible d'améliorer les prédictions. Pour cela l'une des possibilités est de donner des coûts différents pour les mauvais classements de la classe majoritaire et de la classe minoritaire. Une autre possibilité est d'utiliser des méthodes d'échantillonnage. Par exemple l'utilisation de la méthode random oversampling revient à donner des points plus important à la classe minoritaire : les blessures.

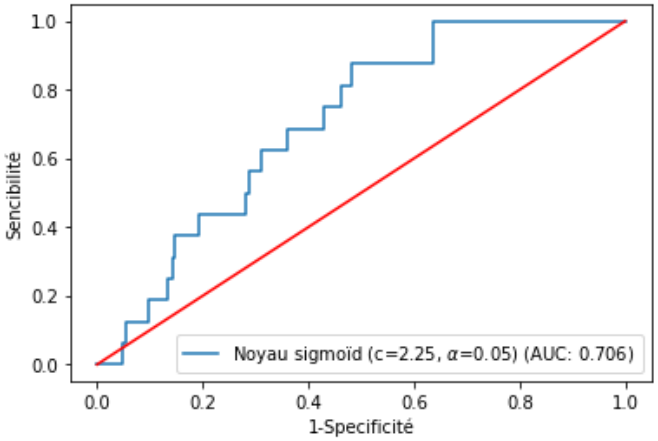


FIGURE 3.19 – Courbe ROC de la modélisation par SVM : noyau sigmoïde ($c = 2.25$, $\alpha = 0.05$)

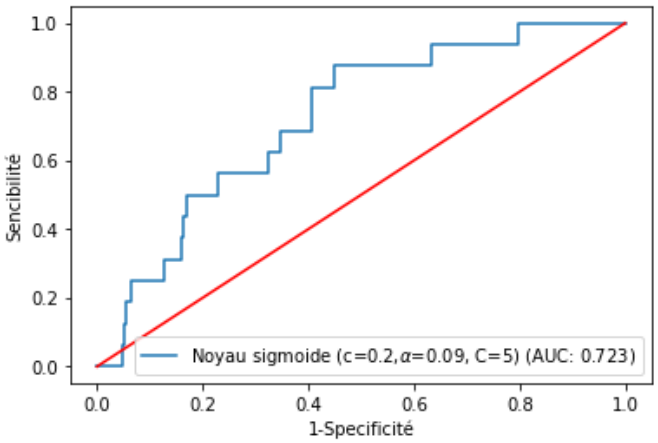


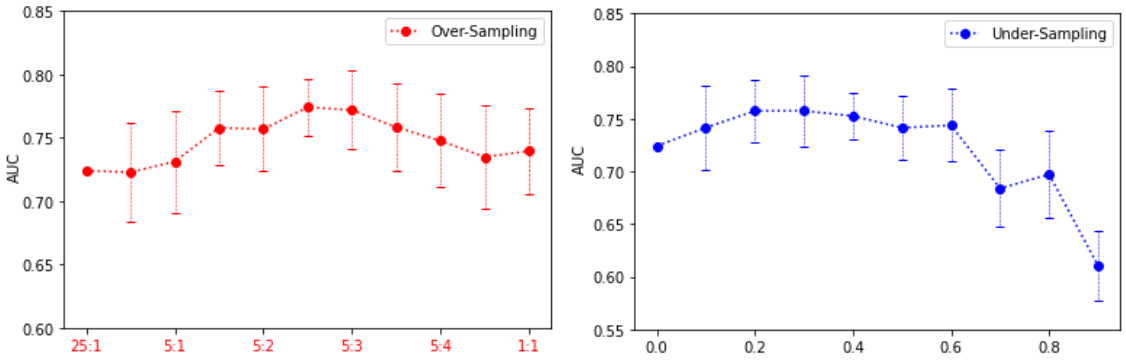
FIGURE 3.20 – Courbe ROC de la modélisation par SVM (noyaux sigmoïde, $c = 0.2$, $\alpha = 0.09$) lorsque l'on ajoute un coût C = 5 de mauvais classement

Conclusion

De manière général, sans méthodes de ré-échantillonnage, la régression logistique et les SVM montrent des résultats équivalents avec un AUC proche de 0.73. Seuls les réseaux de neurones font mieux avec un AUC à 0.76. Cependant la régression logistique est plus simple d'utilisation. En effet, elle ne nécessite pas de calibration contrairement aux SVM et aux réseaux de neurones. En revanche, les arbres de décision montrent des résultats plus faible que les autres.

3.3 Effet des méthodes de ré-échantillonnage

Les méthodes de ré-échantillonnages ont pour objectif d'équilibrer les bases de données en les modifiant. Dans la littérature il est souvent conseillé d'équilibrer 1 : 1, de tels sorte qu'une observation de la classe majoritaire correspond à une observation de la classe minoritaire. Mais cette réalité dépend grandement des données utilisées et de la proportion de chaque classe. Si nous prenons les données que nous avons utilisées, un équilibrage par oversampling 1 : 1 correspond à multiplier par 24 le nombre d'évènements (blessures). Cela peut entrainer un sur-apprentissage et une perte de capacité prédictive de la modélisation. D'un autre côté, si l'oversampling souhaité avec SMOTE est trop important, les points synthétiques générés peuvent détériorer la modélisation. Pour l'undersampling, un équilibrage 1 : 1 correspond à une perte de presque 90% des observations de la base de données. Il va sans dire que cette perte d'information sur la classe majoritaire entraine une mauvaise modélisation. Le choix de l'équilibrage n'est donc pas toujours évident.



(a) Oversampling

(b) Undersampling

FIGURE 3.21 – Effet du taux de ré-échantillonnage sur l'AUC pour la régression logistique

Dans cette partie, nous présentons l'effet des méthodes de sampling sur les modélisations que nous avons utilisées précédemment.

3.3.1 Effet sur la régression logistique

Pour commencer, nous nous intéresserons aux effets des méthodes d'échantillonnage sur la régression logistique. Plus particulièrement aux rééquilibrages par les méthodes de random over/undersampling. Pour cela nous avons tracé les moyennes et les écart-types des indices AUC, sensibilité et spécificité en fonction du taux de ré-échantillonnage. Pour obtenir les courbes présentées, chaque méthode a été effectuée 20 fois comme décrit dans la partie 2.4.

Random over/under sampling

La figure 3.21 présent l'AUC en fonction du taux de sampling lorsque l'on équilibre avec random oversampling (3.21a) et avec random undersampling (3.21a). Chaque point de la courbe représente la moyenne de 20 AUC en utilisant la régression logistique. Pour l'oversampling l'abscisse $x:y$: signifie qu'il a x observations sans évènement pour y observations présentant l'évènement dans la base. Pour l'undersampling, le taux indiqué est le pourcentage d'observations de la classe majoritaire supprimées.

Sur la courbe relative à l'oversampling, on modifie le ratio de 25:1 (le ratio original de la base) à un équilibre 1:1 entre les deux classes. On observe que, pour quasiment l'ensemble des ratios avec oversampling l'AUC est plus élevée que l'équilibrage initial (sans oversampling : AUC=0.724)). Le seul point inférieur étant le second (AUC=0.723). Ensuite, arrive une augmentation d'AUC jusqu'au ratio d'équilibrage 5:3 (AUC=0.772). Après ce point culminant l'équilibrage trop important diminue la qualité du modèle. Le ratio 1:1 a un AUC de 0.74. Il est intéressant de noter que cette augmentation de l'AUC est due à l'information apportée par les observations dupliquées. Mais lorsque l'équilibrage devient trop fort, le poids de la classe minoritaire devient trop important et le modèle est en sur-apprentissage ce qui conduit à une dégradation de la performance.
Pour l'undersampling (figure 3.21b), l'effet est beaucoup plus contrasté puisque la différence entre aucun undersampling (AUC=0.724) et l'AUC le plus haut (AUC=0.757) est de seulement 0.033 obtenue en supprimant 30% des observations de la classe majoritaire. Cependant, cette légère amélioration existe. Ensuite une fois que le taux d'undersampling augmente, la modélisation devient de moins en mois bonne, puisque l'information de la classe majoritaire est totalement perdue.

En résumé sur nos données au sens de l'AUC, l'équilibrage le plus performant est un équilibrage par oversampling entre 5:2 et 5:4. Le graphique 3.21 représente la sensibilité (en bleu) et la spécificité (en rouge) moyenne relative aux meilleurs indices de Pierce de chaque simulation. Une chose importante à observer est le mécanisme antagoniste de la sensibilité et la spécificité.

Lorsque l'un augmente l'autre diminue. C'est l'enjeu des méthodes de sampling : augmenter l'un sans diminuer le second. On observe également très bien que plus le taux d'oversampling est important plus la sensibilité augmente, jusqu'à diminuer à cause du sur-apprentissage.

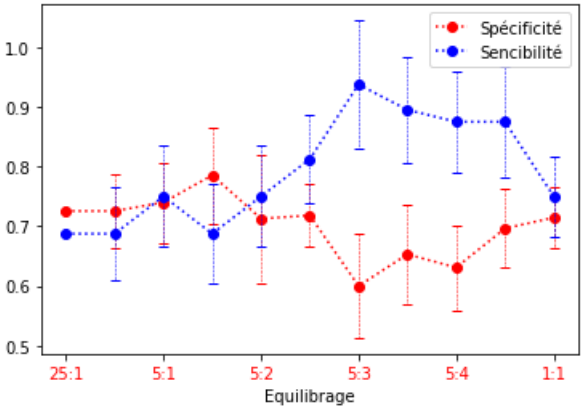
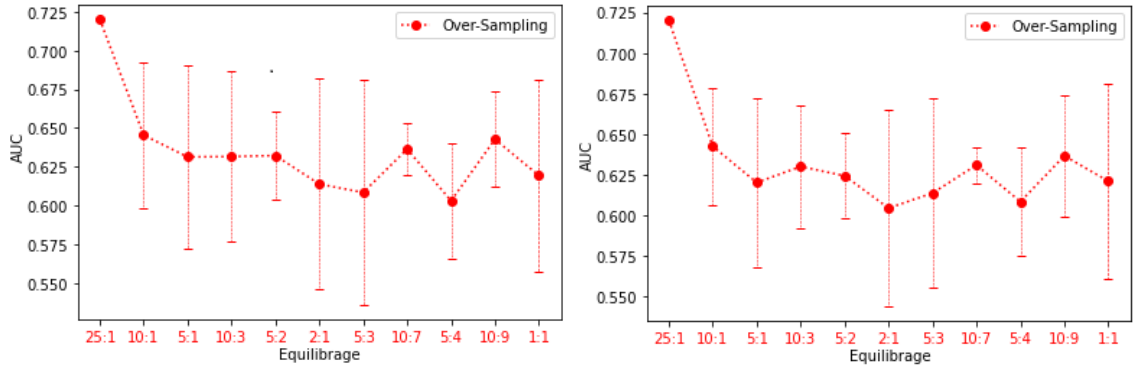


FIGURE 3.22 – Balance de la sensibilité et la spécificité

Les méthodes complexe d'échantillonnage

Nous avons également testé les méthodes d'oversampling présentées dans la section 2.1.2 en combinaison avec la méthode de la régression logistique. En commençant par étudier l'effet des deux plus utilisées : SMOTE et ADASYN. Les courbes de l'AUC obtenues avec ces dernières sont présentées dans la figure 3.23.

Comme nous pouvons le voir, ces méthodes détériorent la modélisation, entrainent une chute de l'AUC. De plus la variation en chaque point est plus importante pour ces méthodes que pour celles de random oversampling. Par exemple, si l'on regarde l'équilibrage 2 : 1, l'écart-types de SMOTE est 0.058, alors qu'il était deux fois moins important pour le random oversampling (0.026). C'est le même ordre de grandeur pour les autres équilibrages. Cette différence de variation peut être expliquée par la génération de blessures synthétiques dans la zone de non-blessures. Ces méthodes, pourtant, ont montré leur efficacité sur d'autres jeux de données. Mais comme nous l'avons montré précédemment, elles peuvent dégrader les modélisation lorsque les données comportent des observations accidentelles.



(a) SMOTE

(b) Adasyn

FIGURE 3.23 – Effet des méthodes SMOTE et ADASYN sur la régression logistique

Les autres méthodes d'oversampling ont été également testées de la même manière mais sans montrer de résultat concluants. Les meilleures AUC obtenues en utilisant les autres méthodes sont :

— Borderline SMOTE : 0.63 pour un équilibrage 10:1

— SMOTE + ENN : 0.64 pour un équilibrage 5:2

— SMOTE + Tomek : 0.64 pour un équilibrage 10:9

Résultats retenus

Il est possible de combiner les méthodes d'échantillonnage entre elles, ce qui multiplie les possibilités. Nous ne présenterons que certaines combinaisons. La table 3.24 présente les résultats de l'indice Peirce et de l'AUC moyen, obtenus sur 20 itérations. La sensibilité et spécificité correspondent à l'une des 20 itérations ayant l'indice de Peirce le plus proche de l'indice de Peirce moyen.

De base, la régression logistique a un AUC de 0.72 et un indice de Peirce de 0.51. La méthode de random oversampling améliore nettement les résultats. Pour l'équilibrage 5:3 (5 non-blessures pour 3 blessures) l'AUC augmente à 0.78 et l'indice de Peirce à 0.56. A l'inverse, l'undersampling doit être utilisée avec plus de précaution, et avec un faible taux d'équilibrage. Les méthodes de SMOTE et ses variantes ne donnent pas de bons résultats sur nos données.

Enfin, la méthode à retenir, est la méthode utilisant de l'undersampling 0.3 et de l'oversampling 5:3 qui a un AUC à 0.78 et un indice de Pierce à 0.547.

Undersampling(1)	Oversampling(2)	SMOTE(3)	AUC (std)	Peirce(4) (std)	Sensitivity	Specificity
None	None	None	0.72	0.51	0.75	0.76
None	5:3	None	0.78 (0.007)	0.560 (0.056)	0.81	0.75
None	1:1	None	0.75 (0.005)	0.551 (0.026)	0.81	0.73
0.3	None	None	0.75 (0.063)	0,518 (0.051)	0.75	0.76
0.5	None	None	0.74 (0.046)	0.536 (0.036)	0.81	0.72
0.3	5:3	None	0.78 (0.009)	0.547 (0.028)	0.81	0.74
0.5	1:1	None	0.77 (0.013)	0.536 (0.037)	0.81	0,73
None	None	10:3	0.62 (0.038)	0.32 (0.082)	0.81	0.70
0.4	None	10:3	0.64 (0.047)	0.34 (0.064)	0.81	0.53

(1) : Oversampling rate
(2) : Undersampling rate
(3) : SMOTE oversampling rate
(4) : Peirce index.

FIGURE 3.24 – Comparaison de l'AUC et de l'indice de Peirce pour plusieurs méthodes d'échantillonnage

3.3.2 Effet sur les arbres de classification

Nous l'avons vu dans le section 3.2.2 les arbres de classification ne donnait pas de bons résultats. En effet, le meilleur AUC trouvé était de 0.684, bien en dessous du modèle de référence. L'un des inconvénients des arbres de classification est leur instabilité lorsque les données sont perturbées. Pour deux bases de données proches, les arbres peuvent engendrer deux modélisations complètement différentes.

Pour illustrer ce phénomène, nous avons perturbé notre jeu de données en retirant aléatoirement 1% des observations de la classe majoritaire cela correspond à la suppression de 30 observations. Ensuite nous utilisons le modèle obtenu pour prédire la saison 2018-2019 en utilisant la validation longitudinale. Les résultats obtenus sont présentés dans la figure 3.25. Au cours des 60 simulations effectuées l'AUC à varié entre 0.522 et 0.71.

Ce problème rend les méthodes d'échantillonnage, dont le but est de perturber les données, compliquées à utiliser. Pour illustrer ce phénomène, nous avons utilisé les méthodes de random oversampling et undersampling en utilisant les arbres de classification (figure 3.26). 20 arbres ont été construits à partir des jeux de données rééquilibrés, puis la moyenne et l'écart-type de

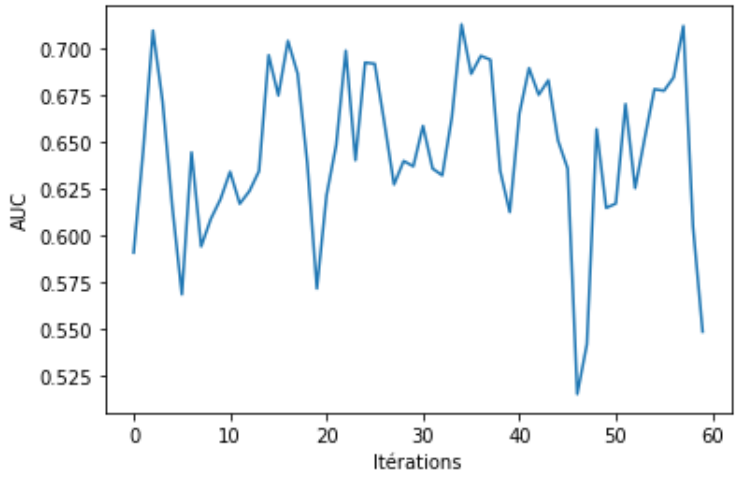
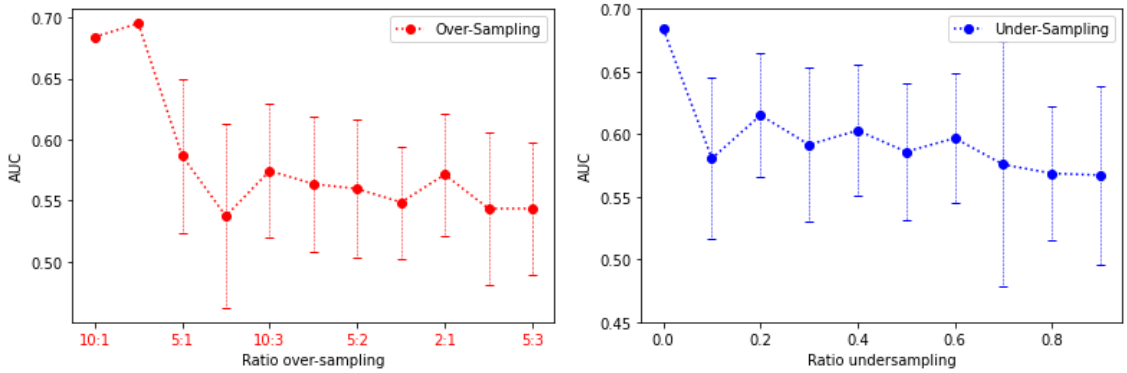


FIGURE 3.25 – Variation de l'AUC lorsque les données sont légèrement perturbées



(a) Oversampling

(b) Undersampling

FIGURE 3.26 – Effet des méthodes d'échantillonnage sur les arbres de décision

l'AUC ont été calculés. On observe que globalement l'oversampling n'a pas un effet positif sur la construction du modèle. Seul un ratio d'oversampling 10 :1 provoque une légère amélioration de l'AUC, avec un écart-type très faible. De plus, les écarts-types montrent la variation importante des arbres lorsque la base de données est modifiée. Concernant l'undersampling (figure 3.26b), plus la base de données est perturbée moins les résultats concernant l'AUC sont bons.

3.3.3 Effet sur le SVM

Il est intéressant d'observer l'effet de l'oversampling et de l'undersampling sur les différentes méthodes. Notamment avec les méthodes qui nécessitent une paramétrisation. Nous l'avons vu plus haut, la meilleure paramétrisation était un noyau sigmoïde de paramètre C = 2.25 et $\alpha = 0.05$. On peut se demander, si ce noyau reste le meilleur avec des données rééquilibrées. La réponse est non, quelque soit le taux d'équilibrage. Par exemple pour 2 : 1 , cette paramétrisation ne permet d'obtenir qu'un AUC moyen de 0.618 sur les 20 itération construites. A l'opposé du noyau linéaire, qui était très mauvais (rappelons que son AUC était de 0.567), devient le meilleur, permettant d'obtenir l'un des meilleurs résultat obtenus jusqu'ici.

Pourtant les SVM sont souvent utilisés pour leur astuce des noyaux et non leur noyau linéaire. En revanche ce dernier est souvent utilisé comme weak learner ou classifieur faible puisqu'il donne rarement de bonnes classifications.

La figure 3.27 présente l'effet de l'oversampling lorsque le noyau linéaire est choisi.

Cette fois, l'effet de l'oversampling est très marqué, puisque l'on observe deux phases :

— jusqu'à l'équilibrage 10 :3, le modèle n'est pas performant. Pour certains équilibrages, le modèle est moins bon que le modèle qui classe aléatoirement les observations. En effet pour les équilibrages 10 : 1 et 5 : 1 l'AUC moyen est de 0.497 et 0.498. Sur cette première phase l'oversampling dégrade le modèle.

— Après l'équilibrage 10:3 on note une amélioration impressionnante du modèle. L'AUC passe de 0.529 pour l'équilibrage 10:3 à 0.761 pour l'équilibrage 5:2. C'est meilleur que la régression logistique sans ré-échantillonnage. Ensuite, le résultat de L'AUC ne fait qu'augmenter jusqu'à son plus haut score pour l'équilibrage 5 : 4 (0.801). C'est le plus haut score observé jusqu'ici. Après ce pic on observe une légère diminution. Cependant il faut noter qu'entre les équilibrages 5:20 et 1:1 les différences sont faibles.

Un autre point à noter est le différence de variation entre les deux phases. Si l'on regarde la première phase de l'équilibrage 25:1 à 10:3 la variation est de l'ordre de 0.06. Alors que sur la seconde phase la variation diminue pour être au alentour de 0.02. A partir de l'équilibrage 5 : 2 le modèle devient plus performant et sa variation diminue.

Cette diminution de variation est un point essentiel pour le fonctionnement de la modélisation dans la vie courante. Les méthodes d'échantillonnage ajoutent de l'aléa et donc de la variabilité dans les prédictions. Or, dans le fonctionnement courant de la méthode, une seule prédiction de risque est donnée. Si la variation du modèle est grande, il est possible, ponctuellement que cette prédiction soit moins précise. En présence d'une faible variation, les résultats restent proches de ceux attendus.

Il existe des méthodes pour diminuer cette variation : ce sont les méthodes d'agrégation.

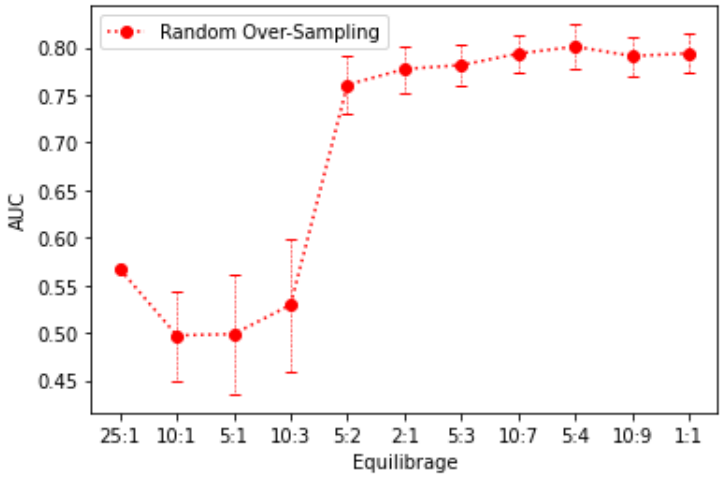


FIGURE 3.27 – Effet du taux de random oversampling sur la modélisation par SVM

Pour rendre compte des résultats des modèles obtenus avec le noyau linéaire et l'équilibrage 5:4 par oversampling, les courbes ROC de trois modèles sélectionnés sont présentées dans la figure 3.28. Parmi l'ensemble des modèles construits avec cet équilibrage nous avons sélectionné :

— le modèle ayant le plus faible AUC (en bleu),

— le modèle ayant l'AUC le plus proche de l'AUC moyen 0.801 (en orange),

— le modèle ayant le plus haut AUC (en vert).

Les courbes sont relativement proches. Le véritable changement se fait à partir d'une sensibilité de 0.29. La différence se fait sur le bon ou mauvais classement de seulement quelques blessures. Sur cette figure, nous avons ajouté la sensibilité et la spécificité correspondant au meilleur indice de Peirce qui est à 0.58. Le taux de blessures bien classées est alors de 88% et le taux de

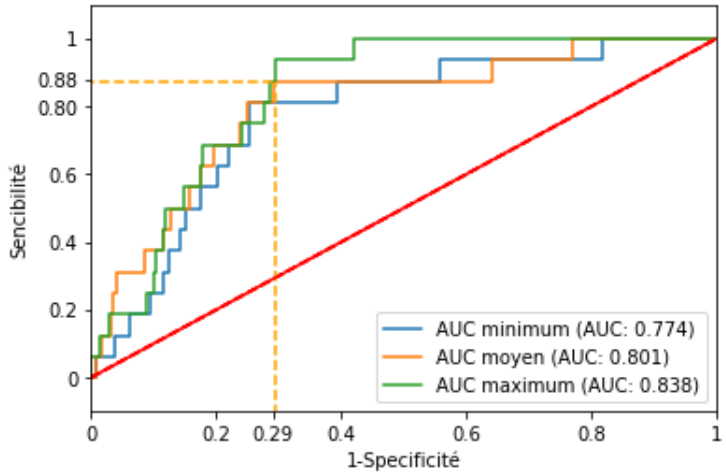


FIGURE 3.28 – Courbes ROC de la modélisation par SVM après équilibrage 5:4

non-blessures bien classées de 71%. Si l'on compare ce résultat à celui obtenu par la régression logistique, c'est une amélioration de 10 points pour le classement des blessures, sans baisse du taux des non-blessures bien classées. C'est donc une amélioration nette.

Concernant les méthode d'undersampling qui supprime les observations éloignées de la frontière des deux classes tel que "One-side sélection", elles n'ont pas de réel intérêt combinées aux méthode de SVM

Lorsque nous présentions les SVM nous avons dit que l'hyperplan était définis par les vecteurs supports, qui, par définition, sont à la frontière des deux classes. Supprimer des observations éloigné de la frontière n'a donc pas d'impact sur l'hyperplan et donc la classification par SVM.

Les méthodes complexes d'échantillonnage

Concernant les méthodes plus complexe d'échantillonnage tel que SMOTE, aucune ne montre de résultats concluants, pour plusieurs raisons :

— trouver le bon paramétrage devient extrêmement compliqué. En effet les méthodes d'oversampling ajoutent des paramètres (nombre de voisin, taux d'équilibrage par exemple). Ce qui multiplie les combinaisons à tester pour trouver les bons paramètres des SVM.

— Ces méthodes modifient l'espace de départ de manière aléatoire. Par exemple SMOTE génère des observations aléatoirement sur le segment entre deux observations de la classe minoritaire. C'est sur ce même espace qu'interviennent les noyaux des SVM, ce qui rend la recherche du bon noyau complexe.

— Nos données longitudinales, font évoluer la base de données par l'ajout d'informations, et donc modifie l'espace de départ. Après plusieurs matchs ajoutés à la base d'apprentissage, un noyau qui semblait bon peut donc devenir "obsolète".

3.3.4 Effet sur les réseaux de neurones

Pour les réseaux de neurones, aucune amélioration n'a été trouvé en utilisant les différentes méthodes de ré-échantillonnage.

3.4 Les méthodes d'agrégation : agrégation par moyenne

Les méthodes d'agrégation consistent à regrouper plusieurs modélisations en une seule. Différentes méthodes existent. Les plus simples regroupent par moyenne ou par vote les classifieurs obtenus. On peut citer par exemple le regroupement de plusieurs classifieur construits sur des

données obtenues en utilisant les méthodes d'échantillonnage. D'autres ajoutent de l'aléa dans les données utilisées pour construire les classifieurs. Comme pour la méthode du Bagging, chaque modèle agrégé est construit sur un échantillon bootstrap. Enfin, d'autres méthodes construisent des classifieurs de manière itérative, chaque classifieur prenant en compte les erreurs du classifieur précédant : c'est le boosting. Dans cette partie nous présenterons et utiliserons les méthodes d'agrégation par moyenne simple et associé au bagging stratifié.

Description des méthodes d'agrégations utilisées

Dans la partie 3.3.1, nous avons montré qu'il était possible d'améliorer les résultats obtenus en utilisant des méthodes d'échantillonnages, notamment en utilisant la méthode de la régression logistique et des svm. Cependant l'échantillonnage entraine une augmentation de la variation des prédictions. Les méthodes d'agrégation peuvent réduire cette variation et améliorer la modélisation. Nous allons donc tester ces méthodes en complément des méthodes d'échantillonnage. La construction de la prédiction finale (utilisant K modèles) se construit donc de la manière suivante :

— construction de K bases de données par over et/ou under sampling ou combinaison des deux,

— construction de K modèles utilisant les K bases de données,

— agrégation par moyenne des prédictions des K modèles.

Lorsqu'une une des deux classes n'est pas modifiée par les méthodes d'échantillonnage, nous avons testé deux stratégies :

- Ne pas modifier la classe, et agréger les modèles par moyenne.
- Créer un échantillon bootstrap de cette classe non modifiée par échantillonnage. Cela nous permet d'ajouter de l'aléa dans les données. Par agrégation nous espérons améliorer les prédictions à l'instar du bagging (stratifié sur la classe non échantillonnée).

Prenons l'exemple du random overampling, les observations de la classe minoritaire sont dupliquées aléatoirement. En revanche la classe majoritaire, n'est pas affectée par cette méthode. Pour cette dernière, nous avons alors deux possibilités :

- dans le premier cas, la classe majoritaire reste inchangée. Seule la classe minoritaire est modifiée par la méthode de ré-échantillonnage,
- dans le second cas, le bagging stratifié : un échantillon bootstrap de la classe majoritaire est créé. La base d'apprentissage est donc constituée de cet échantillon bootstrap de la classe majoritaire et de l'échantillon créé par oversampling de la classe minoritaire.

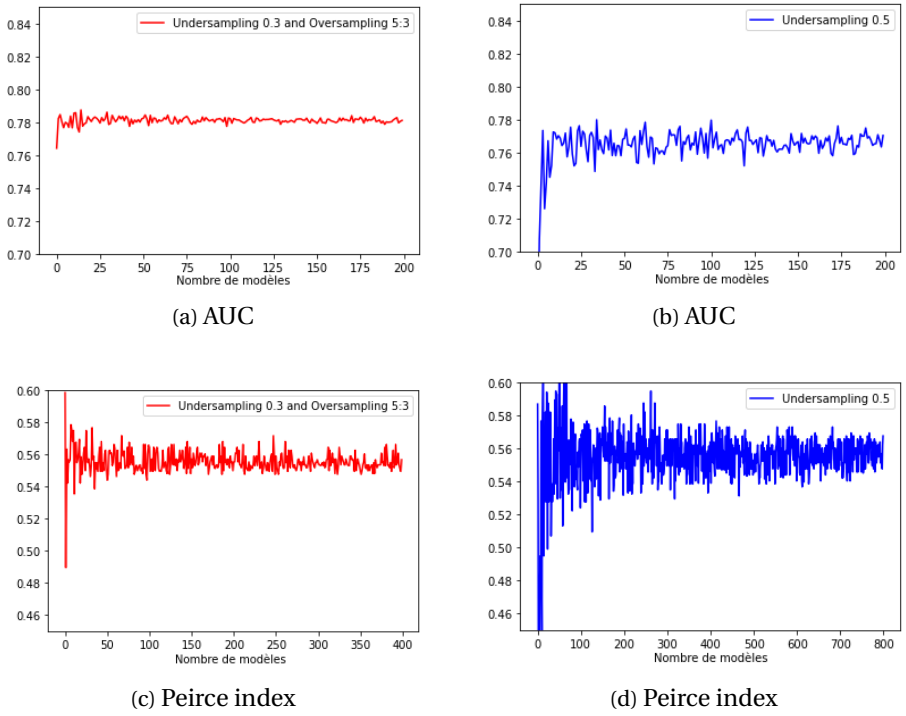
Le seul paramètre à prendre en compte dans les méthodes d'agrégations par moyenne est le nombre de modèle à agrégés. En effet, si le nombre de modèles agréger est trop faible, l'agrégation n'aura pas d'impact. En revanche, s'il est trop élevé, le temps de calcul deviendra trop important, rendant les méthodes d'agrégations inutilisables. Dans la littérature il n'existe pas de consensus. On peut trouver des recommandations, qui sont comprissent entre 20 à 100 itérations [Breiman, 1996], [Bühlmann et Yu, 2002].

Pour choisir le bon nombre d'agrégations nécessaires à la stabilisation du modèle nous avons regardé comment la variation de nos modélisation évoluait en fonction du nombre d'itérations. Puis nous avons choisi le nombre de modèles offrant le meilleur rapport efficacité/rapidité.

3.4.1 Effet sur la régression logistique

La figure 3.29 présente la variation de l'AUC et de l'indice de Peirce pour deux des meilleurs modèles trouvés en utilisant la régression en fonction du nombre de modèles agrégés. Le premier modèle (figure 3.29a et 3.29c) utilise le random undersampling de paramètre $r = 0.3$ et le random oversampling pour un équilibrage final à 5:3. Le second modèle utilise la méthode d'undersampling de paramètre $r = 0.5$ et un échantillon bootstrap pour les blessures.

Globalement, on observe que la variation de l'indice de Peirce est plus importante que pour l'AUC. Pour les deux modèles, l'AUC se stabilise après 20 itérations alors que l'index de Peirce a besoin de plus d'itérations. Dans la suite nous utilisons 20 itérations. C'est le nombre d'itérations qui permet de stabiliser l'AUC et de garder une rapidité de calcul informatique acceptable.



(a) AUC (b) AUC

(c) Peirce index (d) Peirce index

FIGURE 3.29 – Variation de l'AUC et de l'indice de Peirce en fonction du nombre d'agrégation

Pour montrer l'effet des méthodes d'agrégation nous avons comparé les méthodes sans agrégation, avec agrégation et avec agrégation et bagging stratifié. Nous avons sélectionné 4 cas différents pour montrer l'effet de ces méthodes. La figure 3.30 présente les résultats obtenus. Dans ce tableau, pour chaque combinaison utilisée nous indiquons l'AUC moyen et l'indice de Peirce moyen obtenu sur 20 modélisations et leur écart-type. Pour les deux paramètres sensibilité et la spécificité, ils sont obtenus de la manière suivante : parmi les 20 modèles construits, nous sélectionnons celui qui a l'indice de Peirce le plus proche de l'indice moyen. Ensuite nous sélectionnons la sensibilité et la spécificité associé à ce modèle.

Pour le premier modèle, aucune méthode d'échantillonnage n'a été utilisée. On compare donc seulement l'impact du bagging stratifié. Le modèle utilisant l'agrégation est bien meilleur en terme d'AUC puisque celui-ci augmente de 0.4. L'indice de Peirce est également meilleur. La variabilité apporté par le bootstrap reste relativement basse. L'agrégation a donc un véritable intérêt.

Pour le second modèle, les données sont équilibrées par random oversampling de ratio 5 : 3. Dans ce cas il n'y a quasiment aucun effet sur les performances des méthodes d'agrégation. En effet les 3 méthodes ont le même AUC, et quasiment le même indice de Pierce à 0.005 prêt. C'est sur la variabilité et l'AUC et de l'indice de Peirce que les deux méthodes d'agrégation ont un impact. En effet l'agrégation permet de diminuer l'écart type de l'AUC de 0.007 à 0.002. C'est également le cas pour l'écart-type de l'indice de Peirce qui diminue de 0.056 à 0.05.

Pour le troisième modèle, un ramdom undersampling de paramètre 0.3 (30% des observations de la classe majoritaire sont supprimées) est utilisé. Ici, l'effet est moins marqué. L'agrégation simple augmente les performances puisque l'AUC augmente de 0.02 points. En revanche, le bagging stratifié diminue les performances de 0.01 points. Cependant les deux augmentent l'indice de Peirce de 0.03 à 0.04, ce qui représente un gain important. Encore une fois, l'apport le plus important de l'agrégation est sur la variabilité puisque, pour l'AUC, elle diminue de 0.063 à 0.005

Agrégation	Undersampling(1)	Oversampling(2)	AUC (std)	Peirce (3) (std)	Sensibilité	Specificité
Non	None	None	0.72	0.51	0,75	0,76
OUi	Boots	Boots	0.76 (0.004)	0.526 (0.025)	0.81	0.66
Non	None	5:3	0.78 (0.007)	0.560 (0.056)	0.81	0.75
OUi	None	5:3	0.78 (0.002)	0.562 (0.005)	0.75	0.81
OUi	Boots	5:3	0.78 (0.002)	0.555 (0.008)	0.81	0,74
Non	0.3	None	0.75 (0.063)	0,518 (0.051)	0.75	0.76
OUi	0.3	None	0,77 (0.005)	0.554 (0.035)	0.81	0.75
OUi	0.3	Boots	0.74 (0.007)	0.549 (0.030)	0.81	0.75
Non	0.3	5:3	0.77 (0.013)	0.547 (0.028)	0.81	0.74
OUi	0.3	5:3	0.78 (0.009)	0.566 (0.010)	0.81	0.74

(1) : Taux d'oversampling ou échantillon boostrap
(2) : Taux d'undersampling ou échantillon boostrap
(3) : Index de Peirce.

FIGURE 3.30 – Résultats des méthodes d'agrégations sur la régression logistique

pour l'agrégation simple, et 0.004 avec bootstrap sur la classe non échantillonnée. On observe également une baisse de la variation de l'indice de Peirce.

Enfin pour le dernier modèle, une combinaison d'undersampling 0.3 et d'oversampling 5 : 3, est utilisé. Encore une fois l'agrégation améliore les résultats de l'AUC (de 0.77 à 0.78) et de l'indice de Peirce de (0.547 à 0.566). La variation des deux indices diminue également (pour l'AUC de 0.013 à 0.009 et Pour l'indice de Peirce de 0.028 à 0.010).

Sur ces quatre exemples nous avons montré l'apport des méthodes d'agrégation. Pour la régression logistique, les deux méthodes d'agrégation font quasiment toujours aussi bien que sans agrégation et permettent de réduire significativement la variation des modélisations. Nous allons observer des résultats similaires en utilisant les SVM.

3.4.2 Effet sur les arbres de classification

Nous l'avons vu dans la section 3.2.2 les arbres de classification ne donnait pas de bons résultats. Cela peut être un avantage pour les méthodes d'agrégation. En effet les arbres de classification peu profond sont souvent utilisés dans les méthodes d'agrégation.

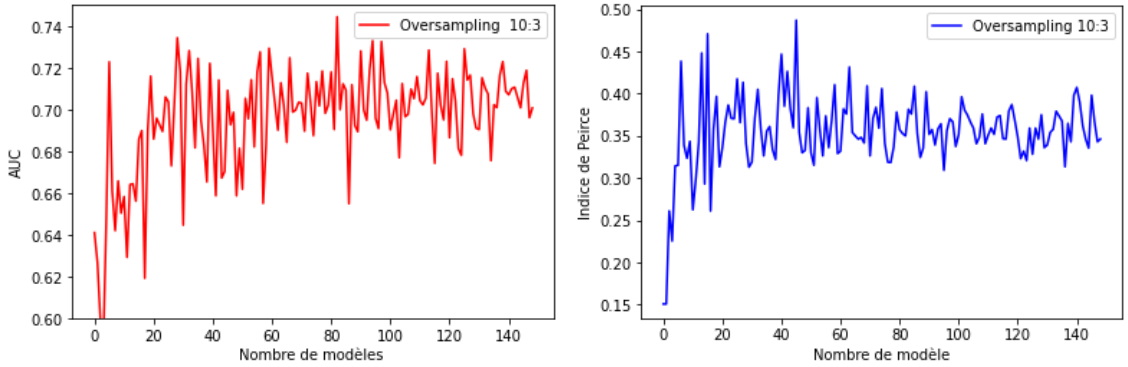
Dans la partie modélisation sans agrégation, les meilleurs arbres trouvés avaient une profondeur de 4. Ici nous utiliserons des arbres moins profonds, qui ont une profondeur de 2. Bien que ces arbres soient moins performants (AUC=0.58), Lorsque l'on utilise des méthodes d'agrégations leur performance est largement améliorée.

Comme pour la régression logistique, nous avons cherché le nombre de modèles à agréger nécessaire pour la stabiliser les prédictions. Les figures 3.31a et 3.31b présente la variation de l'AUC et de l'indice de Peirce en fonction du nombre de modèles agréger.

Contrairement à la régression logistique, les arbres de classifications nécessitent plus de modèles pour être stabilisés. En effet, il faut environ 50 modèles pour stabiliser l'AUC, qui reste néanmoins très fluctuant. Pour l'indice de Peirce, plus le nombre de modèles augmente, plus la variation diminue, mais à partir de 50 modèles la variabilité a déjà bien diminuée. Nous avons donc choisi d'utiliser les agrégations de 50 modèles.

Pour montrer l'effet des méthodes d'agrégation, nous avons comparé les méthodes sans agrégation, avec agrégation simple par moyenne et avec agrégation et bagging stratifié. Nous avons dans un premier temps pour évaluer l'effet de l'agrégation, nous avons utilisé la méthode du random l'oversampling. La figure 3.32a présente l'effet de l'agrégation avec bagging stratifié pour différent équilibrage.

Sur cette figure, l'effet de l'agrégation est remarquable. En effet, la courbe du modèle avec agrégation est toujours au dessus de la courbe du modèle sans agrégation. Sans agrégation le mo-



(a) AUC

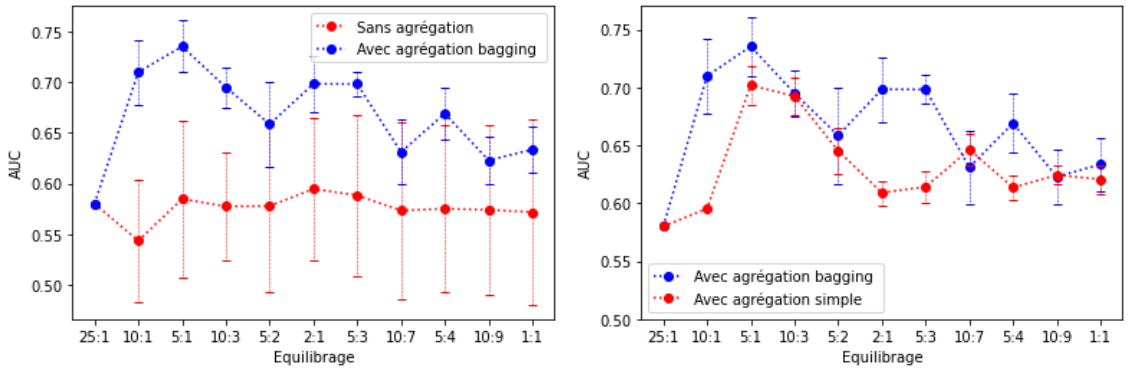
(b) Indice de Peirce

FIGURE 3.31 – Variation de l'AUC et de l'indice de Peirce en fonction du nombre d'agrégation pour les arbres de classification

dèle est très moyen avec une AUC moyen de 0.58. Alors qu'avec agrégation, l'AUC moyen est de 0.67. Nous pouvons également noter que sans agrégation l'effet de l'équilibrage par oversampling est quasi inexistant. Au contraire pour le modèle avec agrégation bagging, l'équilibrage a un réel impact sur l'AUC. On observe une augmentation de celui-ci jusqu'à son maximum (AUC=0.74) pour l'équilibrage 5:1. Ensuite l'AUC diminue globalement même si on observe une légèrement augmentation pour les ratio 2:1 5:3, jusqu'à atteindre un AUC de 0.63 pour l'équilibrage 1:1.

Concernant la variation des modèles, l'agrégation permet de la diminuer nettement. En effet pour l'ensemble des équilibrages la variabilité du modèle sans agrégation est systématiquement 2 à 4 fois moins importantes. Par exemple pour l'équilibrage 10:3, l'écart-type du modèle sans agrégation est de 0.053, alors qu'avec agrégation, il est de 0.020. Pour l'équilibrage 5:3, sans agrégation 0.080, avec agrégation 0.012.

La figure 3.32b, permet de comparer les modèles avec agrégation par moyenne simple et les modèles avec agrégation associé au bagging stratifié. Le modèle avec agrégation bagging stratifié est légèrement meilleur. Son AUC est quasiment toujours au dessus du modèle avec agrégation simple. Le meilleur AUC des deux méthodes est pour l'équilibrage 5:1, 0.74 pour le bagging starifié et 0.71 pour l'agrégation simple. L'avantage de l'agrégation simple est sa faible variabilité. En effet, elle toujours inférieure à celle de l'agrégation avec bagging. Par exemple pour le ratio 2 : 1, elle est de 0.011, alors que pour le bagging stratifié, elle est de 0.028. Cette différence entre les deux méthodes peut s'expliquer simplement. La méthode du bagging stratifié ajoute de l'aléa dans les données. En effet, il construit des échantillons bootstrap ce qui induit une variation plus élevée.



(a) Comparaison des arbres de classification avec agrégation bagging

(b) Comparaison de l'agrégation simple et l'agrégation bagging

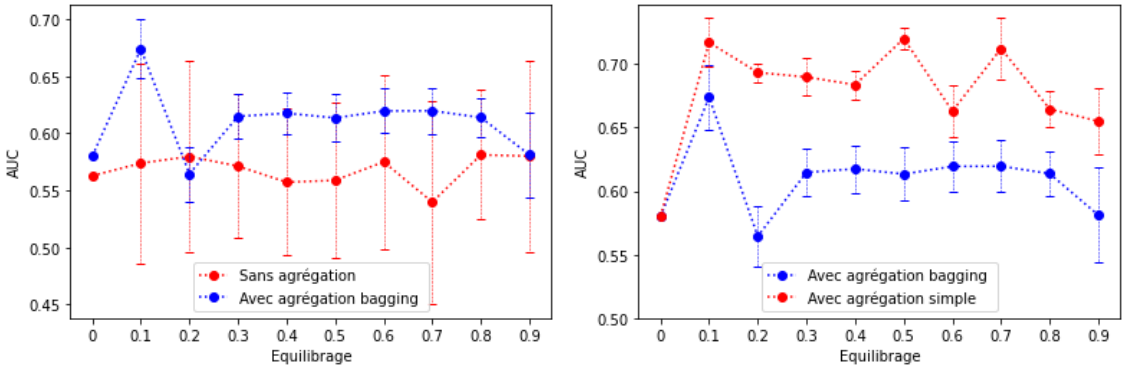
FIGURE 3.32 – Effet des méthodes d'agrégation sur les arbres de classification

Nous avons effectué la même comparaison sur l'équilibrage par undersampling. La figure

3.33a présente la courbe du modèle sans agrégation et du modèle avec bagging stratifié lorsque les données sont équilibrées par random oversampling.

La première chose à noter est que l'undersampling donne de moins bon résultats que l'oversampling. En effet le meilleur AUC est de 0.68 pour l'équilibrage 10 : 1. Comme pour l'oversampling, le modèle avec agrégation est meilleur et moins variant que celui sans agrégation.

Le point intéressant se trouve sur la figure 3.33a. Nous voyons qu'avec l'undersampling, c'est le modèle avec agrégation simple qui est le meilleur. Tout en étant toujours moins variant que le modèle avec bagging stratifié. Cependant les résultats reste plus faibles qu'avec oversampling.



(a) Comparaison des arbres de classification avec agrégation bagging

(b) Comparaison de l'agrégation simple et l'agrégation bagging

FIGURE 3.33 – Effet des méthodes d'agrégation sur les arbres de classification

Pour rendre compte des résultats des modèles obtenus avec les arbres de classification agrégé par bagging stratifié et équilibré 10 : 3 par oversampling, nous présentons les courbes ROC de 3 modèle (minimum, moyenne et maximum) obtenus dans la figure 3.34.

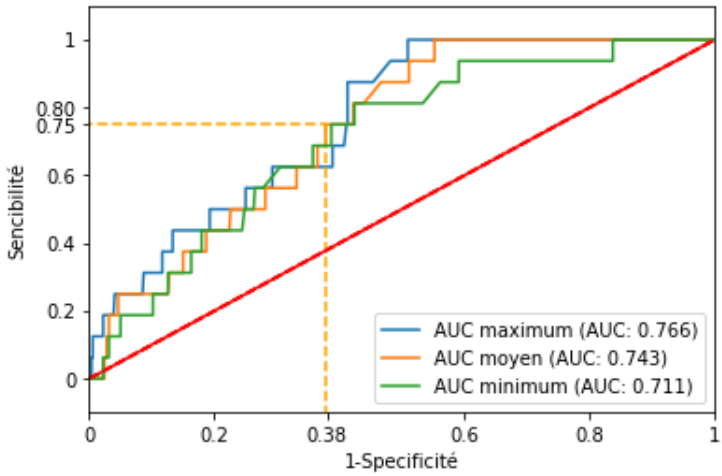


FIGURE 3.34 – Courbes ROC de la modélisation par SVM après équilibrage 5 : 4

Sur ces courbes, bien que l'AUC soit plus haut (0.74) que celui de la régression logistique simple (0.73) L'indice de Peirce est plus faible (0.44) alors qu'il était de 0.51. Ce qui entraine une diminution de la sensibilité qui passe de 0.71 à 0.67. Cette modélisation est donc moins bonne que la modélisation de référence.

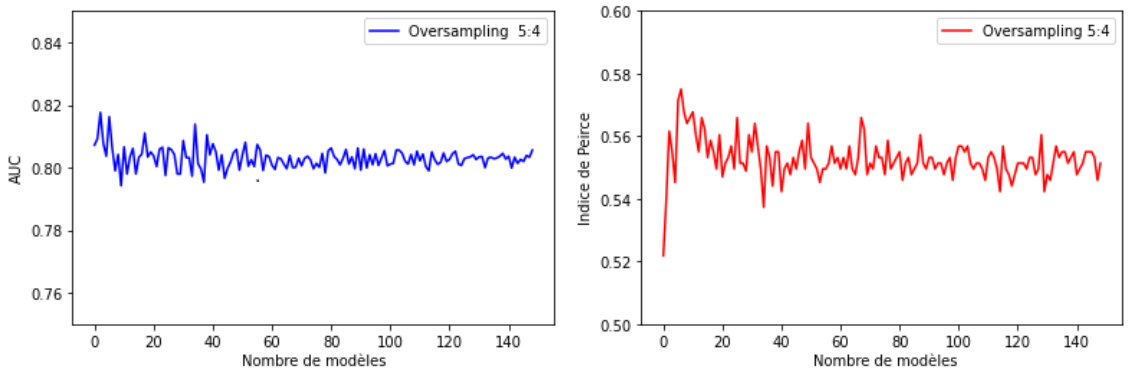
A travers l'utilisation des arbres de classification, nous avons vu trois points importants.

- les méthodes d'agrégation peuvent améliorer les résultats obtenus. En effet sans agrégation les arbres que nous avons choisis avaient un AUC de 0.58, alors qu'avec agrégation l'AUC peut atteindre 0.74.

- La variabilité engendrée par les modèles d'échantillonnage peut être grandement diminuée par ces méthodes. La méthode d'agrégation simple étant la plus stable.
- Selon le type d'échantillonnage, (under ou over sampling) les méthodes d'agrégation ne vont pas avoir le même effet.

3.4.3 Effet sur les SVM

L'impact des méthodes d'échantillonnage sur les méthodes des SVM est très important, notamment la méthode d'oversampling. En effet, en utilisant un équilibrage 5 : 4, l'AUC augmente de 0.22 points. Nous avons souhaité regarder l'effet des méthodes d'agrégation sur les SVM. Pour choisir le nombre de modèles à agréger nécessaires, nous avons regardé les variations de l'AUC (figure 3.35a) et de l'indice de Peirce (figure 3.35b) avec une équilibrage 5 :3 par random oversampling.



(a) AUC

(b) Indice de Peirce

FIGURE 3.35 – Variation de l'AUC et de l'indice de Peirce en fonction du nombre d'agrégation pour les SVM

Contrairement à la régression logistique, l'AUC et l'indice de Peirce se stabilisent au bout de 40 modèles agrégés. L'indice de Peirce est toujours plus variant que l'AUC, comme nous l'avons vu avec la régression logistique. Nous avons choisi d'utiliser 40 modèles suite à ces graphiques.

Pour observer l'effet de l'agrégation, nous avons comparé les résultats obtenus avec et sans agrégation en utilisant la méthode de random oversampling. La figure 3.36 présente l'AUC moyen (et l'écart type) obtenu par la construction de 20 modélisations selon l'équilibrage par oversampling.

— La courbe en rouge représente les modèles sans agrégation. Cette courbe a déjà été présentée (figure 3.27).

— La courbe en bleu représente les modèles construit avec agrégation. Pour mieux comparer et éviter que les courbes ne se croisent, nous avons légèrement déphasé ou décalé cette courbe. Les deux courbes ont donc bien été testées sur les mêmes équilibrages.

Nous voyons que touts les points des modèles avec agrégation se situent au dessus de ceux sans agrégation. C'est très marqué pour les équilibrages 5:1 et 10:3. La différence est moins importante pour équilibrage de 5:2 à 1:1. Cela montre quand même un léger gain d'AUC global. Le meilleur modèle est toujours obtenu pour l'équilibrage 10 : 9 avec un AUC de 0.805 (sans agrégation le meilleur AUC était 0.801 pour l'équilibrage 10 :9).

Le véritable apport, comme nous l'avons dit précédemment, est le gain de stabilité. En effet si l'on regarde en détail chaque écart-type, il est systématiquement diminué. Par exemple pour l'équilibrage 5:1 (resp. 10:3), sans agrégation, l'écart-type est de 0.064 (resp. 0.071), alors qu'avec agrégation il est de 0.045 (resp. 0.027). Pour les valeurs d'équilibrage plus élevées, la méthode est

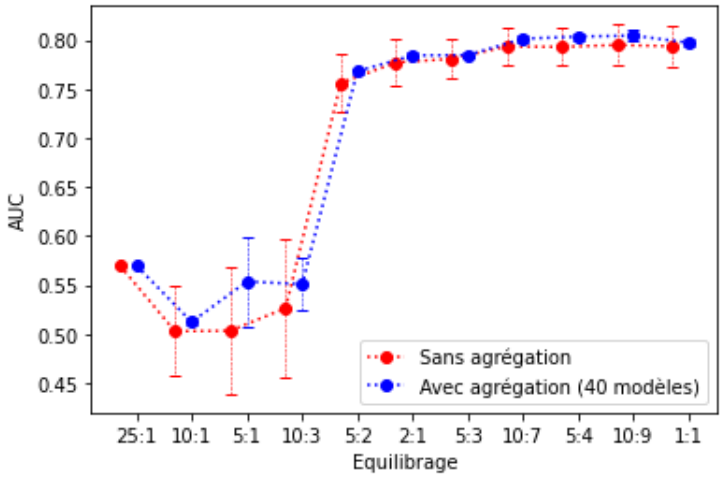


FIGURE 3.36 – Effet de l'agrégation sur la modélisation par SVM

complètement stabilisée. Les écart-types sont tous inférieurs à 0.007, alors qu'ils étaient autour de 0.02 sans agrégation.

Nous voyons donc ici l'importance d'utiliser les méthodes d'agrégation : ils nous assurent une plus grande stabilité dans le modèle et donc une meilleure confiance dans celui-ci.

3.5 Autres méthodes d'agrégation

Dans la partie précédente, nous avons présenté les méthodes d'agrégation par moyenne, mais il en existe d'autre. Dans cette partie, nous présenterons les résultats obtenus en utilisant certaine d'entre elles, en commençant par le plus connue : la méthode des forêts d'arbres aléatoires.

3.5.1 Forêts d'arbres aléatoires

Les forêts d'arbres aléatoires présentées dans la section 3.2.2, sont une méthode d'agrégation utilisant les arbres de classification en tant que weak learner. En effet les arbres de classification sont de mauvais prédicateurs de la blessure. Il est donc intéressant de voir si les résultats peuvent être améliorés en les regroupant.

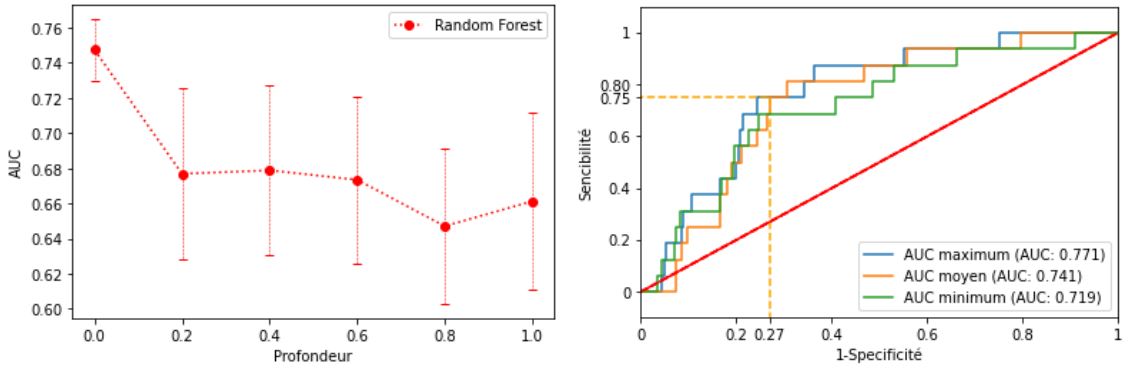
L'agrégation des random Forest se fait par vote des différents arbres construits. Ainsi la probabilité qu'une joueur j se blesse est donnée par le rapport entre le nombre d'arbres classant ce joueur blessé et le nombre total d'arbres générés. Les particularités des random forest se trouve dans les données utilisées pour construire chaque arbre. En effet, pour chaque arbre, les données d'apprentissage sont construites de la manière suivante

— Tirage aléatoire des observations avec remise.

— Tirage aléatoire des variables utilisées sans remise

Cela évite de sélectionner plusieurs fois une variable fortement liée à la réponse pour séparer les données. Ainsi, aux paramètres des arbres de classification à ajuster, il faut ajouter les paramètres : nombre de variables à tirer aléatoirement et nombre d'arbres à construire. Comme pour la méthode des arbres de classification, nous avons testé différentes paramétrisations afin d'obtenir le meilleur résultat. Nous concernant, nous avons construit nos forêts d'arbres en utilisant des arbres de profondeur maximum 2. Nous avons ensuite généré 300 arbres en utilisant pour chaque arbre un nombre de variables égal à la racine carré des variables totales.

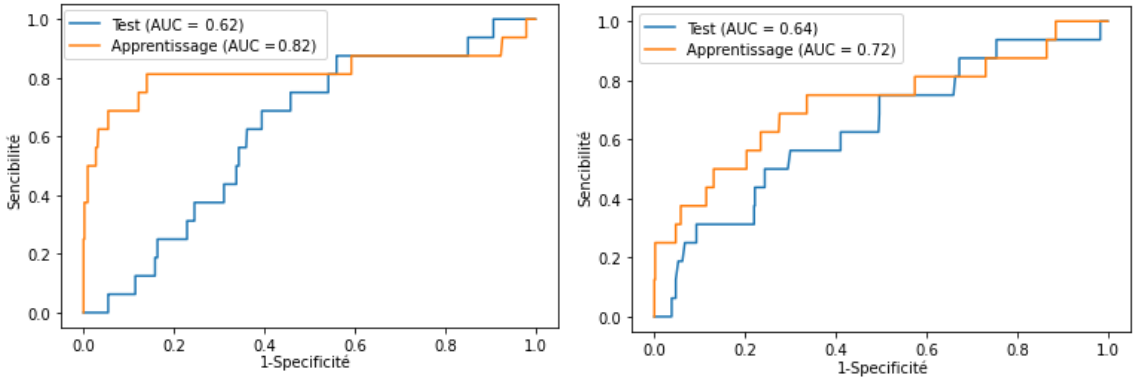
Le premier commentaire que nous pouvons faire est la taille des arbres que nous avons choisis. En effet, si l'on regarde la figure 3.37a où l'AUC moyen sur 20 tirages est présenté en fonction de la profondeur des arbres, la meilleure valeur est pour les arbres les moins profonds. L'AUC moyen est à 0.747. C'est d'autant plus intéressant que dans les méthodes d'agrégation de type boosting



(a) Effet de la profondeur des arbres

(b) Courbes ROC

FIGURE 3.37 – Méthode des Random Forest



(a) Sur-apprentissage

(b) Modélisation par XGboost

FIGURE 3.38 – Courbes ROC de la modélisation par Xgboost

utilisant les arbres, ce sont des arbres de cette profondeur qui sont conseillés. C'est également le cas dans les méthodes de Gradient tree boosting.

Le modèle obtenu avec ces paramètres est meilleur que le modèle de référence utilisant la régressions logistiques sans méthode de ré-échantillonnage et d'agrégation. Les courbes ROC du meilleur, du moins bon et de celui proche de l'AUC moyen sont présentées dans la figure 3.37b. Pour la courbe proche de l'AUC moyen, l'indice de Pierce étant à 0.48 ce qui correspond à une sensibilité de 0.73 et une spécificité de 0.75.

3.5.2 Xgboost

Pour tester la méthode d'arbres boosté, nous avons choisi d'utiliser Xgboost qui montre des résultats intéressants dans la littérature, et notamment dans le domaine des compétitions de data science. Cependant, ces méthodes souffrent de sur-apprentissage qu'il est important de gérer. Connaissant la problématique de nos données nous avons fait particulièrement attention à ce que l'algorithme ne soit pas en sur-apprentissage. Nous avons donc choisi de prendre des arbres de petite taille (profondeur maximum de 2 nœuds), 200 itérations de boosting et 70% des variables à chaque itération. Mais les résultats n'ont pas été convaincants. Soit le modèle sur-apprenait, soit les prédictions étaient mauvaises. La figure 3.38 présente ces deux cas.

3.5.3 Boosting

Une méthode plus complexe d'agrégation est le boosting. En effet il va chercher à améliorer la prédiction des points mal classés, en leur donnant un poids plus important aux modèles suivants. Habituellement les modèles arbres sont utilisés, les SVM ont également montrés des résul-

tats intéressant dans la littérature. En revanche la régression logistique est rarement utilisée. Nous avons testé le boosting sur ces 3 méthodes. Mais malheureusement aucun résultats concluant n'a été trouvé. Toutes les méthodes utilisées montrent des résultat plus faible que sans boosting. Par exemple, la figure 3.39 présente l'AUC obtenus à l'aide du boosting sur la régression logistique en fonction du nombre d'itération de boosting. Les SVM ont montré un fort potentiel lorsqu'ils sont utilisés avec des méthodes d'échantillonnage et notamment les méthodes d'oversampling. Cependant comme pour le sur la régression logistique les résultats sont moins bons que lorsque le boosting n'est pas utilisé.

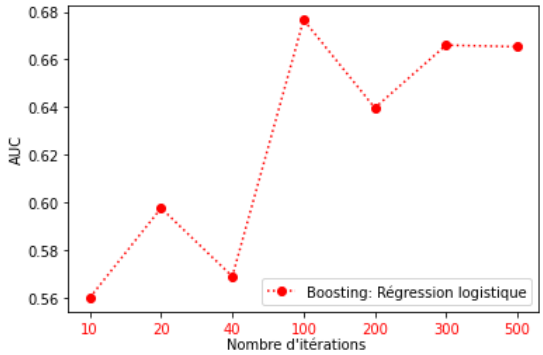


FIGURE 3.39 – Boosting de la régression logistique

Deux choses sont intéressantes à noter : la méthode du boosting n'améliore pas les prédictions faites. C'est en partie dû au faite que, le boosting donne trop d'importance aux blessures accidentelles, ce qui conduit à une perte d'information sur les autres blessures. La deuxième chose à noter est qu'à partir de 100 itérations de boosting les résultats sont meilleurs, même s'ils sont toujours en dessous des résultats de la régression logistique simple. Après ce seuil à 100 itérations le modèle ne s'améliore plus.

3.6 Les meilleures modélisations

Finalement les meilleurs modèles que nous retenons sont dans la figure 3.40. Les deux méthodes utilisées sont les méthodes de régression logistique et de SVM. Les méthodes des réseaux de neurones montraient des résultats légèrement plus faibles. Mais c'est surtout leur complexité à généraliser qui nous a poussé à ne pas les retenir. En effet nous voulons un modèle capable de garder ses capacités prédictives dans le temps. C'est le même constat pour les SVM à noyau. En revanche, SVM sans noyau est un bon candidat puisqu'avec un équilibrage des données, elles présentent le meilleur AUC trouvé de (0.81) en plus d'être très stable. La régression logistique est également un choix intéressant puisqu'elle n'a aucun paramètre à calibrer. Elle montre des résultats très compétitifs que ce soit avec de l'over ou de l'undersampling.

La figure 3.41 présente de les courbes ROC des trois modèles :

— SVM noyaux linaire : oversampling 10:3 et agrégation avec bagging.

— Régression logistique : undersampling 0.5 et oversampling 1:1 et agrégation.

— Réseau de neurones à 2 couches cachées

Clairement la différence entre les deux meilleurs modèle est très faible. En effet, l'AUC et l'indice de Peirce sont quasiment égaux.

Méthodes	Agrégation	Undersampling(1)	Oversampling(2)	AUC (std)	Peirce (3) (std)	Sensibilité	Specificité
Reg. log.	Non	None	None	0.72	0.51	0,75	0,76
Reg. log.	Non	None	5:3	0.78 (0.007)	0.560 (0.056)	0.81	0.75
Reg. log.	OUi	None	5:3	0.78 (0.002)	0.562 (0.005)	0.75	0.81
Reg. log.	Oui	0.5	1:1	0.79 (0.003)	0.564 (0.021)	0.81	0.75
SVM (linéaire)	Non	None	10:3	0.80 (0.020)	0.536 (0.004)	0.81	0.73
SVM (linéaire)	Oui	None	10:3	0.81 (0.006)	0.554 (0.004)	0.81	0.74

(1) : Taux d'oversampling ou échantillon boostrap
(2) : Taux d'undersampling ou échantillon boostrap
(3) : Index de Peirce.

FIGURE 3.40 – Meilleurs méthodes

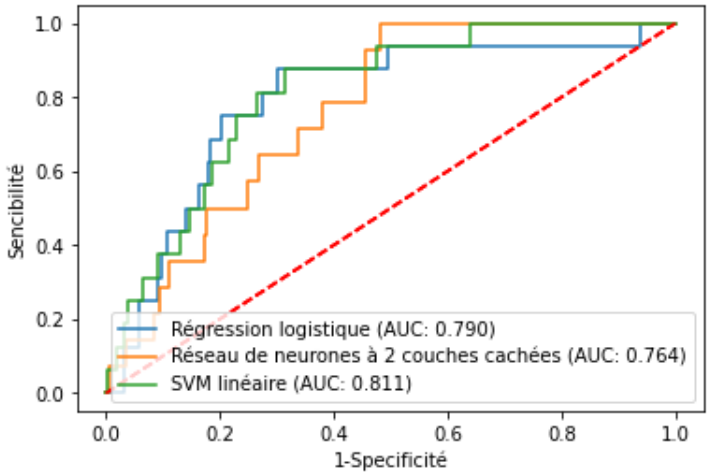


FIGURE 3.41 – Courbe ROC des meilleures modélisation

3.6.1 L'évolution du risque

Enfin pour terminer, nous présentons un exemple de l'évolution de l'indicateur au cours d'une saison. La figure 3.42 présente l'évolution du risque de blessures chez un joueur . Les blessures survenant sans contact en match sont indiquées par les pointillés rouges.

Deux choses sont à noter :

— Il y a 3 pics où le joueur est à risque. Sur le pic le plus important (0.952), le joueur se blesse. Sur les deux autres pics de niveaux comparables (0.286 et 0.332) le joueur ne se blesse que sur le second.

— Bien que le joueur 2 soit à risque, sa probabilité "moyenne" est faible. Elle est de 0.045 en dehors des 3 pics qui représente la probabilité de survenance d'une blessure dans l'effectif. En comptant ces 3 valeurs elle passe à 0.091.

Enfin une 3 troisième blessures est passée plus inaperçu. Le joueur était moins à risque. Néanmoins elle arrive sur l'un des 6 pics où la probabilité de se blesser est d'environ 0.1.

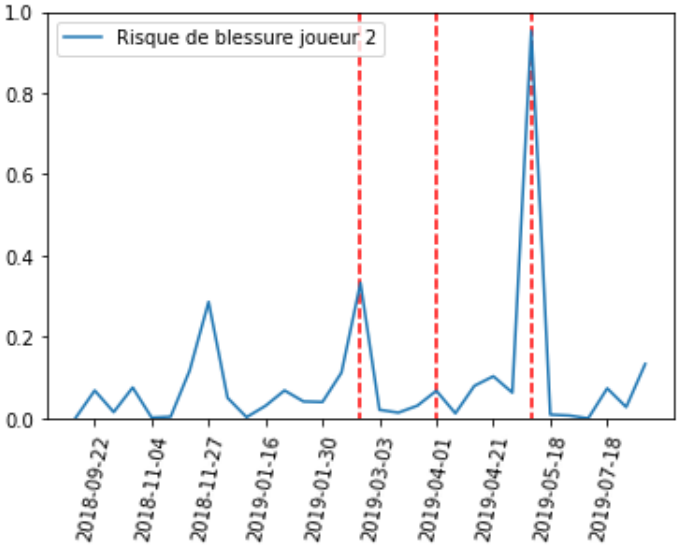


FIGURE 3.42 – Évolution de l'indice chez le joueur 1 durant la saison 2018-2019

3.7 Références

ARNI, A., S. STEFAN et E. LARS. 2004, «Risk factors for injuries in football», *The American Journal of Sports Medicinel.* 58

BAHR, R. et T. KROSSHAUG. 2005, «Understanding injury mechanisms : a key component of preventing injuries in sport», *Br J Sports Med.* 58

BÜHLMANN, P. et B. YU. 2002, «Analyzing bagging», *Annals of Statistics*, vol. 30. 83

BILLY, T. H., G. TIM J, W. L. DANIEL, C. PETER et S. JOHN A. 2015, «The acute :chronic workload ratio predicts injury : high chronic workload may decrease injury risk in elite rugby league players», *Br J Sports Med.* 59, 70

BREIMAN, L. 1996, «Bagging predictors», *Machine Learning*, vol. 24, p. 123–140. 83

CRISTIANO, E., T. J L, F. ABDULAZIZ, S. FATEN et C. HAKIM. 2012, «Low injury rate strongly correlates with team success in qatari professional football», *Br J Sports Med.* 58

DANIEL, C. et R.-G. JAVIER. 2017, «The prevalence of injuries in professional soccer players», *Journal of Orthopedic Research and Therapy.* 58, 60

EKSTRAND, J., T. TIMPKA et M. HÄGGLUND. 2006, «Risk of injury in elite football played on artificial turf versus natural grass : a prospective two-cohort study», *Br J Sports Med.* 58

FOSTER, C. G. 1998, «Monitoring training in athletes with reference to overtraining syndrome.», *Medicine and science in sports and exercise*, vol. 30 7, p. 1164–8. 59

JAN, E., H. MARTIN et W. MARKUS. 2011, «Epidemiology of muscle injuries in professional football (soccer)», *The American Journal of Sports Medicine.* 58, 60

JOHANN, W. et G. TIM J. 2016, «How do training and competition workloads relate to injury? the workload—injury aetiology model», *Br J Sports Med.* 59

KONSTANTINOS, F., T. ELIAS et A. SPYROS. 2010, «Intrinsic risk factors of non-contact quadriceps and hamstring strains in soccer : A prospective study of 100 professional players», *Br J Sports Med.* 59

MARTIN, H., W. MARKUS, M. HENRIK, K. KAROLINA, B. HÅKAN et E. JAN. 2013, «Injuries affect team performance negatively in professional football : an 11-year follow-up of the uefa champions league injury study», *Br J Sports Med.* 58

MARTIN, H., W. MARKUS et E. JAN. 2016, «Previous injury as a risk factor for injury in elite football : A prospective study over two consecutive seasons», *Br J Sports Med.* 59

POLISSAR, L. et P. DIEHR. 1982, «Regression analysis in health services research : The use of dummy variables», *Medical Care*, vol. 20, nº 9, p. 959–966. 66

TIM J, G. 2016, «The training-injury prevention paradox : should athletes be training smarter and harder?», *Br J Sports Med.* 59, 70

Chapitre 4

Analyse de la sensibilité : Méthode de Sobol

Sommaire

L'analyse de la sensibilité est l'étude de l'impact des données d'entrées $X \in \mathbb{R}^p$ sur la sortie $Y \in \mathbb{R}$ donnée par un modèle. Elle permet de :

— connaitre les variables d'entrées les plus influentes et donc supprimer les variables non influentes,

— montrer et quantifier les interactions entre les variables CHAN et collab. [1997].

Elle est donc souvent utilisée sur des modèles de type "boite noire" (réseaux de neurones, bagging par exemple). Effectuer une analyse de la sensibilité permet d'améliorer la compréhension et la performance des modèles utilisés.

Il existe plusieurs méthodes d'analyse de la sensibilité, elles se distinguent en deux groupes :

— l'approche locale, connue sous le nom de "one-at-a-time" (OAT). Elle mesure l'effet de la variation d'une variable d'entrée sur la sortie lorsque les autres variables d'entrées sont fixées,

— l'approche globale qui permet d'estimer l'effet sur la sortie Y lorsque les variables d'entrées varient indépendamment.

L'approche locale repose principalement sur des calculs de gradients ou de dérivées partielles. Elle permet de décrire la variabilité localement au voisinage de quelques points. Plusieurs techniques existent et sont détaillées dans l'article de Zhou [ZHOU et LIN, 2008]. L'une des plus utilisées est la méthode de Moriss MORRIS [1991]. Elle permet, en utilisant des dérivées partielles, d'obtenir graphiquement l'influence des variables sur le modèle et de savoir si l'influence des variables est non linéaire ou avec de l'interaction. Cependant, il n'est pas possible de quantifier ni d'expliciter les interactions entre les variables.

L'approche globale, plus complexe, permet d'obtenir plus d'information sur les interactions entre les variables et leur influence globale. L'une de ces techniques, au centre des recherches ces dernières années, est la méthode de Sobol I. M. [2001]. Nous la présenterons et l'utiliserons sur nos modèles. Chaque méthode possède ses avantages et ses inconvénients, [CARIBONI et collab., 2007] proposent une manière de choisir la bonne méthode à utiliser selon la problématique.

4.1 Les indices de Sobol

Les indices de Sobol I. M. [2001] sont utilisés lorsque le modèle considéré est défini de la manière suivante :

$$y = f(X), \tag{4.1}$$

avec $X = (X_1, ..., X_p)$ les variables d'entrées et y le scalaire de sortie (par exemple une probabilité). Les variables d'entrées $X \subset [0,1]^p$ sont définies sur l'hypercube d'unité 1 et considérées indépendantes. Les indices de Sobol permettent de mesurer la proportion de la variance expliquée par les variables d'entrées du modèle. Plus précisément, ils permettent d'obtenir les informations suivantes :

— Les variables qui engendrent le plus de variabilité sur la variable de sortie.

— Les interactions multiples entre les variables.

Les indices de Sobol sont basés sur la décomposition de la variance totale. Pour faciliter les calculs et leurs compréhensions, nous ferons quelques rappels.

— L'espérance mathématique conditionnelle de Y sachant $X = x$ est donnée par :

$$\mathbb{E}(Y \mid X = x) = \int y f_{Y|X=x}(y)\,dy = \int y \frac{f_{Y,X}(y)}{f_X(y)}\,dy, \tag{4.2}$$

avec $f_{Y|X=x}$, $f_{Y,X}$ et f_X les fonctions de densité.

Dans la suite nous poserons que $\mathbb{E}_X$ et $\mathbb{E}_Y$ sont les espérances selon X et Y respectivement de sorte que $\mathbb{E}_X(X) = \int x f_X(x)\,dx$. Lorsqu'il ne peut pas avoir de confusion nous utiliserons simplement $\mathbb{E}()$.

Il est maintenant aisé de vérifier la formule de la variance totale

Theorem 2 *Pour toutes variables aléatoire* X *et* Y *la formule de la décomposition de la variance totale :*

$$Var_Y(Y) = \mathbb{E}_X(Var_Y(Y \mid X)) + Var_X(\mathbb{E}_Y(Y \mid X)). \tag{4.3}$$

Nous pouvons maintenant présenter les indices de Sobol I. M. [1993] et leurs constructions. Ils proviennent de la décomposition dites de Hoeffding-Sobol qui repose sur les travaux initiaux de la décomposition de Hoeffding, HOEFFDING [1948]. Par souci de compréhension les notations suivantes seront utilisées pour présenter cette décomposition.

Notation.

$u, v \subseteq \{1,2,...,p\}$, signifie que u et v sont deux ensembles formés des éléments de l'ensemble $\{1,2,...,p\}$. On attirera l'attention sur le fait que les ensemble u et v peuvent être égaux à l'ensemble $\{1,2,...,p\}$ du fait du signe "$\subseteq$", ce qui n'est pas été le cas avec le signe "$\subset$". Enfin l'ensemble u privé de v sera défini par $u \setminus v$ et $\mid u \mid \in \mathbb{N}$ est le cardinal de u .

Lemme 1 *Décomposition de Hoeffding-Sobol : Soit* $(X1,...,X_p)$ *p variables supposées indépendantes, g la fonction densité jointe,* $g^p \to \mathbb{R}^+$ *et* $f : I \to \mathbb{R}$ *une fonction de carré intégrable par rapport à la fonction de densité jointe g des p variables. i.e.*

$$\int_I f^2(x)g(x)dx < \infty. \tag{4.4}$$

Alors il existe une unique décomposition de la fonction $y = f(x_1,...,x_p)$ *en somme de fonction de dimension croissante de la forme :*

$$f(x_1,...x_p) = h_0 + \sum_i h_i(x_i) + \sum_{1\le i\le j\le p} h_{i,j}(x_i,x_j) + ... + h_{i,...,p}(x_i,...,x_p). \tag{4.5}$$

avec h_0 *constant et pour tout* $i_j \in (1,..,p)$*, et* $k = 1,...s$*, chaque fonction h vérifie :*

$$\int h_{i_1,..,i_s}(x_{i_1},..,x_{i_s}) f_{X_{i_k}}(x_{i_k})dx_{i_k} = 0, \tag{4.6}$$

où $f_{X_{i_k}}$ *est la fonction de densité associée à* X_{i_k}.

De plus pour tout sous ensemble $u \subseteq \{1,2,...,p\}$*, les fonctions* h_u *sont définies par :*

$$h_u(X_u) = \mathbb{E}(y \mid X_u) - \sum_{v\subset u} h_v(X_v) = \mathbb{E}(y \mid X_u) + \sum_{v\subset u} (-1)^{|u|-|v|}\mathbb{E}(y \mid X_v). \tag{4.7}$$

Demonstration.

Prenons une fonction $f : I = I_1 \times ... \times I_p \subset \mathbb{R}$ de carré intégrable par rapport à g, ou g est la fonction de densité jointe des p variables indépendantes $(X_1,...X_p)$. Supposons que la fonction f admette une décomposition de la forme présentée dans l'équation 4.5, avec les h_u, $u \subseteq \{1,2,...,p\}$ vérifiant la condition 4.6. Alors, on a pour tout $u \subseteq \{1,2,...,p\}$:

$$\begin{aligned}
\mathbb{E}(y \mid X_u) &= \mathbb{E}\left(h_0 + \sum_i h_i(X_i) + \sum_{1\le i\le j\le p} h_{i,j}(X_i,X_j) + ... + h_{i,...,p}(X_i,...,X_p) \mid X_u\right) \\
&= \sum_{v\not\subseteq u} \mathbb{E}(h_v(X_v)) + \sum_{v\subseteq u} h_v(X_v) \\
&= \sum_{v\not\subseteq u} \int h_v(X_v) f_{X_w}(x_w)dx_w + \sum_{v\subseteq u} h_v(X_v),
\end{aligned}$$

avec f_{X_w} la fonction de densité jointe des X_i, pour $i \in \{1,2,...,p\} \setminus u$. Les X_i étant indépendantes, on peut écrire :

$$\mathbb{E}(y \mid X_u) = \sum_{v\not\subseteq u} \int h_v(X_v) \prod_{i\in x} f_{X_i}(x_i)dx_i + \sum_{v\subseteq u} h_v(X_v).$$

En utilisant les propriété des $h_v(\mathrm{X}_v)$ 4.6 :

$$\mathbb{E}(y \mid \mathrm{X}_u) = \sum_{v \subseteq u} h_v(\mathrm{X}_v).$$

Nous avons donc montré que pour tout $u \subseteq \{1,2,...,p\}$, l'unique décomposition de $h_u(\mathrm{X}_u)$ est :

$$h_u(\mathrm{X}_u) = \mathbb{E}(y \mid \mathrm{X}_u) - \sum_{v \subset u} h_v(\mathrm{X}_v).$$

Dans un second temps nous allons montrer l'égalité :

$$h_u(\mathrm{X}_u) = \mathbb{E}(y \mid \mathrm{X}_u) + \sum_{v \subset u} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_v).$$

Pour cela prenons la fonction $k_u(\mathrm{X}_u) = \mathbb{E}(y \mid \mathrm{X}_u) + \sum_{v \subset u} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_v)$ et montrons qu'elle est égale à $h_u(\mathrm{X}_u)$. Pour commencer vérifions que $k_u(\mathrm{X}_u)$ respecte les conditions 4.6. D'après les rappels 4.2 et l'indépendance des variables X_i

$$\int \mathbb{E}(y \mid \mathrm{X}_u) f_{\mathrm{X}_i}(x_i) d_{x_i} = \mathbb{E}(y \mid \mathrm{X}_{u \setminus \{i\}}). \tag{4.8}$$

Soient $s \subseteq \{1,2,...,p\}$, $u \subseteq s$ et $i \in u$ un élément de u,

$$\begin{aligned}
\int k_u(x_u) f_{\mathrm{X}_i}(x_i) d_{x_i} &= \int \left(\mathbb{E}(y \mid \mathrm{X}_u) + \sum_{v \subset u} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_v) \right) f_{\mathrm{X}_i}(x_i) d_{x_i} \\
&= \int \mathbb{E}(y \mid \mathrm{X}_u) f_{\mathrm{X}_i}(x_i) d_{x_i} + \sum_{v \subset u} (-1)^{|u|-|v|} \int \mathbb{E}(y \mid \mathrm{X}_v) f_{\mathrm{X}_i}(x_i) d_{x_i} \\
&= \mathbb{E}(y \mid \mathrm{X}_{u \setminus i}) + \sum_{v \subset u} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_{v \setminus i}) \qquad \text{(d'après 4.8)} \\
&= \sum_{v \subseteq u} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_{v \setminus i}) \\
&= \sum_{\substack{v \subset u \\ i \notin v}} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_v) + \sum_{\substack{v \subseteq u \\ i \in v}} (-1)^{|u|-|v \setminus i|-1} \mathbb{E}(y \mid \mathrm{X}_{v \setminus i}) \\
&= 0.
\end{aligned}$$

Nous venons donc de montrer que $k_u(x_u)$ vérifiait bien la condition 4.6. Nous allons maintenant montrer que $\sum_{u \subseteq \{1,2,...,p\}} k_u(x_u) = y$. Pour ce faire nous utiliserons le fait qu'il y a $2^{|u|}$ sous ensembles ayant un nombre d'éléments pair (D'après la formule du binôme de Newton), il y a donc $2^{|u|-1}$ sous ensembles u avec un nombre d'éléments impair. Également, pour tout $v \subseteq \{1,2,...,p\}$, il existe $2^{p-|v|}$ sous ensemble $v \subseteq \{1,2,...,p\}$ tel que $v \subseteq u$ et donc $2^{p-|v|-1}$ d'entre-eux tel que $| u \setminus v |$ est pair.

$$\begin{aligned}
\sum_{u \subseteq \{1,2,...,p\}} k_u(x_u) &= \sum_{u \subseteq \{1,2,...,p\}} \left(\mathbb{E}(y \mid \mathrm{X}_u) + \sum_{v \subset u} (-1)^{|u|-|v|} \mathbb{E}(y \mid \mathrm{X}_v) \right) \\
&= \sum_{u \subseteq \{1,2,...,p\}} \mathbb{E}(y \mid \mathrm{X}_u) + \sum_{u \subseteq \{1,2,...,p\}} \sum_{v \subset u} (-1)^{|u \setminus v|} \mathbb{E}(y \mid \mathrm{X}_v) \\
&= \sum_{u \subseteq \{1,2,...,p\}} \mathbb{E}(y \mid \mathrm{X}_u) + \sum_{v \subseteq \{1,2,...,p\}} ((2^{p-|v|-1} - 1)\mathbb{E}(y \mid \mathrm{X}_v) - 2^{p-|v|-1} \mathbb{E}(y \mid \mathrm{X}_v) \\
&= \mathbb{E}(y \mid \mathrm{X}_{\{1,2,...,p\}}) \\
&= y.
\end{aligned}$$

Les fonctions de la décomposition de Hoeffding-Sobol étant unique, nous avons montré que la fonction $y = f(x_1, ...x_p)$ peut être décomposée en somme de fonctions de dimensions croissantes respectant les bonnes conditions.

Plusieurs affirmations peuvent être déduites du lemme :

— la première, grâce à la condition 4.6 et à l'indépendance des $(X_1,...X_p)$ induit que les h_u ($u \subseteq \{1,2,...,p\}$ sont orthogonaux. En effet avec $u, v \subseteq \{1,2,...,p\}$ et $u \neq v$:

$$\int h_u(x_u) h_v(x_v) f(x) dx = 0, \tag{4.9}$$

— Par identification des fonction h_u, $f(x_1,...x_p) = h_0 + \sum_i h_i(x_i) + \sum_{1\leq i\leq j\leq p} h_{i,j}(x_{i,j}) + ... + h_{i,...,p}(x_{i,...,p})$:

$$\begin{aligned}
h_0 &= \mathbb{E}(Y),\\
h_i &= \mathbb{E}(Y \mid X_i) - \mathbb{E}(Y),\\
h_{i,j} &= \mathbb{E}(Y \mid X_i, X_j) - \mathbb{E}(Y \mid X_j) - \mathbb{E}(Y \mid X_i) - \mathbb{E}(Y),\\
h_{i,j,k} &= \mathbb{E}(Y \mid X_i, X_j, X_k) - \mathbb{E}(Y \mid X_i, X_j) - \mathbb{E}(Y \mid X_i, X_k) - \mathbb{E}(Y \mid X_j, X_k) - \mathbb{E}(Y \mid X_k) - \mathbb{E}(Y \mid X_j) - \mathbb{E}(Y \mid X_i) - \mathbb{E}(Y).
\end{aligned}$$

— Finalement, par indépendance des variables, on peut définir la décomposition de la variance :

$$Var(Y) = \sum_{u\subseteq\{1,2,...,p\}} Var(h_u(X_u)), \tag{4.10}$$

ou autrement :

$$Var(Y) = \sum_i V_i + \sum_{1\leq i\leq j\leq p} V_{i,j} + ... + V_{i,...,p}(x_{i,...,p}). \tag{4.11}$$

Avec

$$\begin{aligned}
V_i &= Var(\mathbb{E}(Y \mid X_i)),\\
V_{i,j} &= Var(\mathbb{E}(Y \mid X_i, X_j)) - V_i - V_j,\\
V_{i,j,k} &= Var(\mathbb{E}(Y \mid X_i, X_j, X_k)) - V_{i,j} - V_{i,k} - V_{j,k} - V_i - V_j - V_k,\\
&\vdots\\
V_{1,..,p} &= V - \sum_i V_i - \sum_{1\leq i\leq j\leq p} V_{i,j} - ... - \sum_{1\leq i_1\leq ...\leq i_{p-1}\leq p} V_{i_1,...,i_{p-1}}.
\end{aligned}$$

Nous pouvons maintenant définir l'intégralité des indices de Sobol. L'indice de premier ordre de Sobol I. M. [2001], aussi appelé "corrélation ratio" par Mckay McKay [1995], mesure l'effet d'une variable X_i, $i = 1...p$ sur les sorties Y d'un modèle. Il est définie par :

$$\begin{aligned}
S_i &= \frac{Var_{X_i}(h_i(X_i)}{Var(Y)}),\\
&= \frac{Var_{X_i}(\mathbb{E}_Y(Y \mid X_i))}{Var(Y)}, \qquad \text{pour } i = 1,...,p\\
&= \frac{V_i}{Var(Y)}.
\end{aligned} \tag{4.12}$$

Cet indice, compris entre 0 et 1, mesure l'influence engendrée par une variation des variables d'entrées sur la sortie Y ou encore la part de la variance de Y expliquée par X_i. Plus X_i engendre une variation de l'espérance de $Y \mid X_i$ plus cet indice de premier ordre sera proche de 1. L'indice de premier ordre donne l'influence direct d'une variable sur un modèle, mais ne renseigne pas sur l'impact de ses interactions avec les autres variables sur le modèle.

Pour mesurer l'effet des interactions il faut définir les indices d'ordres supérieurs.

— Les indices de second ordre :

$$\begin{aligned} S_{i,j} &= \frac{Var_X(h_{i,j}(X_i,X_j)}{Var(Y)}, \\ &= \frac{Var_{X_i,X_j}(\mathbb{E}_Y(Y \mid X_i,X_j))}{Var(Y)} - S_i - S_j, \qquad \text{pour } i = 1,...,p \\ &= \frac{V_{i,j}}{Var(Y)}, \end{aligned} \tag{4.13}$$

expriment la sensibilité de la variance de Y à l'interaction entre les variables X_i et X_j, qui n'est pas prise en compte dans l'indice de premier ordre.

— Plus généralement nous définissions l'ensemble des indices (du premier ordre à l'ordre p) par :

$$\begin{aligned} S_u &= \frac{VarX_u(h_u(X_u)}{Var(Y)}, \\ &= \frac{Var_{X_u}(\mathbb{E}_Y(Y \mid X_u))}{Var(Y)} - \sum_{v \subset u} S_v, \qquad \text{pour } u, v \subseteq \{1,2,...,p\} \\ &= \frac{V_u}{Var(Y)}. \end{aligned} \tag{4.14}$$

Ces différents indicateurs nous renseignent sur l'effet principal et les interactions d'une variable. Ils sont simples d'utilisation puisqu'ils sont tous positifs et que leur somme est égale à 1. Plus l'indicateur sera grand, proche de 1, plus la variable ou ces interactions seront importantes. Cependant lorsque le nombre de variables devient grand, l'interprétation des indices devient compliquée puisque un modèle à p variables engendre $2^p - 1$ indices. L'indice de sensibilité total d'une variables X_i a été introduit par HOMMA et SALTELLI [1996] pour palier ce problème. Il mesure la sensibilité de la sortie du modèle Y à la variable d'entrée, c'est à dire à son effet principal et à ses interactions avec les autres variables. Il est défini par la sommes des indices de sensibilité relatifs à la variable X_i. On définit donc l'indice total S_{T_i} d'une variable X_i :

$$S_{T_i} = \sum_{\substack{v \subseteq u \\ i \in v}} S_v. \tag{4.15}$$

Pour donner un exemple, dans un modèle à trois variables X_1, X_2, X_3, l'indice de sensibilité total de X_1 est $S_{T_1} = S_1 + S_{1,2} + S_{1,3} + S_{1,2,3}$.

Exemple 1 *Soit* X_i, $i = 1,2,3$ *des variables indépendantes et de loi* $\mathcal{N}(\mu_i, \sigma_i^2)$. *Calculons les indices de Sobol pour le modèle* $Y = f(X_1,X_2,X_3) = X_1X_2 + X_3$ *:*

$$\begin{aligned} Var(Y) =& Var(f(X_1,X_2,X_3) \\ =& Var(h_1(X_1)) + Var(h_2(X_2)) + Var(h_3(X_3)) \\ &+ Var(h_{1,2}(X_1,X_2)) + Var(h_{2,3}(X_2,X_3)) \\ &+ Var(h_{1,3}(X_1,X_3)) + Var(h_{1,2,3}(X_1,X_2,X_3)), \end{aligned}$$

avec :

$$h_0 = \mathrm{E}(\mathrm{Y}) = \mathrm{E}(\mathrm{X}_1\mathrm{X}_2 + \mathrm{X}_3) = \mu_1\mu_2 + \mu_3,$$

$$h_1(\mathrm{X}_1) = \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_1) - h_0 = \mathrm{X}_1\mu_2 + \mu_3 - (\mu_1\mu_2 + \mu_3) = \mathrm{X}_1\mu_2 - \mu_1\mu_2,$$

$$h_2(\mathrm{X}_2) = \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_2) - h_0 = \mathrm{X}_2\mu_1 + \mu_3 - (\mu_1\mu_2 + \mu_3) = \mathrm{X}_2\mu_1 - \mu_1\mu_2,$$

$$h_3(\mathrm{X}_3) = \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_3) - h_0 = \mu_1\mu_2 + \mathrm{X}_3 - (\mu_1\mu_2 + \mu_3) = \mathrm{X}_3 - \mu_3,$$

$$\begin{aligned} h_{1,2}(\mathrm{X}_1,\mathrm{X}_2) &= \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_1,\mathrm{X}_2) - h_1(\mathrm{X}_1) - h_2(\mathrm{X}_2) - h_0 \\ &= \mathrm{X}_1\mathrm{X}_2 + \mu_3 - (\mathrm{X}_2\mu_1 - \mu_1\mu_2) - (\mathrm{X}_1\mu_2 - \mu_1\mu_2) - (\mu_1\mu_2 + \mu_3) \\ &= \mathrm{X}_1\mathrm{X}_2 - \mathrm{X}_1\mu_2 - \mathrm{X}_2\mu_1 + \mu_1\mu_2, \end{aligned}$$

$$\begin{aligned} h_{1,3}(\mathrm{X}_1,\mathrm{X}_3) &= \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_1,\mathrm{X}_3) - h_1(\mathrm{X}_1) - h_3(\mathrm{X}_2) - h_0 \\ &= \mathrm{X}_1\mu_2 + \mathrm{X}_3 - (\mathrm{X}_1\mu_2 - \mu_1\mu_2) - (\mu_1\mu_2 + \mathrm{X}_3 - (\mu_1\mu_2 + \mu_3)) - \mu_1\mu_2 - \mu_3 = 0, \\ h_{2,3}(\mathrm{X}_2,\mathrm{X}_3) &= \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_2,\mathrm{X}_3) - h_2(\mathrm{X}_2) - h_3(\mathrm{X}_2) - h_0 \\ &= \mathrm{X}_2\mu_1 + \mathrm{X}_3 - (\mathrm{X}_2\mu_1 - \mu_1\mu_2) - (\mu_1\mu_2 + \mathrm{X}_3 - (\mu_1\mu_2 + \mu_3)) - \mu_1\mu_2 + \mu_3 = 0, \end{aligned}$$

$$\begin{aligned} h_{1,2,3}(\mathrm{X}_1,\mathrm{X}_2,\mathrm{X}_3) &= \mathrm{E}(\mathrm{Y} \mid \mathrm{X}_1,\mathrm{X}_2,\mathrm{X}_3) - h_{1,2}(\mathrm{X}_1,\mathrm{X}_2) - h_{1,3}(\mathrm{X}_1,\mathrm{X}_3) - h_{2,3}(\mathrm{X}_2,\mathrm{X}_3) \\ &\quad - h_1(\mathrm{X}_1) - h_2(\mathrm{X}_2) - h_3(\mathrm{X}_3) - h_0 \\ &= \mathrm{X}_1\mathrm{X}_2 + \mathrm{X}_3 - (\mathrm{X}_1\mathrm{X}_2 - \mathrm{X}_1\mu_2 - \mathrm{X}_2\mu_1 + \mu_1\mu_2) - 0 - 0 \\ &\quad - (\mathrm{X}_1\mu_2 - \mu_1\mu_2) - (\mathrm{X}_2\mu_1 - \mu_1\mu_2) - (\mathrm{X}_3 - \mu_3) - (\mu_1\mu_2 + \mu_3) - \mu_3 = 0. \end{aligned}$$

Pour faciliter les calculs des indices nous allons expliciter les variances suivantes :

$$\begin{aligned} \mathrm{V}ar(\mathrm{Y}) &= \mathbb{E}((\mathrm{X}_1\mathrm{X}_2 + \mathrm{X}_3)^2) - \mathbb{E}(\mathrm{X}_1\mathrm{X}_2 + \mathrm{X}_3)^2 \\ &= \mathbb{E}(\mathrm{X}_1^2)\mathbb{E}(\mathrm{X}_2^2) + \mathbb{E}(\mathrm{X}_3^2) - \mathbb{E}(\mathrm{X}_1)^2\mathbb{E}(\mathrm{X}_2)^2 - \mathbb{E}(\mathrm{X}_3)^2 \\ &= (\sigma_1^2 + \mu_1^2)(\sigma_2^2 + \mu_2^2) + (\sigma_3^2 + \mu_3^2) - \mu_1^2\mu_2^2 - \mu_3^2 \\ &= (\sigma_1^2 + \mu_1^2)(\sigma_2^2 + \mu_2^2) - \mu_1^2\mu_2^2 + \sigma_3^2, \end{aligned}$$

$$\mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_1,\mathrm{X}_2,\mathrm{X}_3)) = \mathrm{V}ar(\mathrm{X}_1\mathrm{X}_2 + \mathrm{X}_3) = \mathrm{V}ar(\mathrm{Y}),$$

$$\begin{aligned} \mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_1,\mathrm{X}_2)) &= \mathrm{V}ar(\mathrm{X}_1\mathrm{X}_2 + \mathbb{E}(\mathrm{X}_3) \\ &= \mathbb{E}(\mathrm{X}_1^2)\mathbb{E}(\mathrm{X}_2^2) - \mathbb{E}(\mathrm{X}_1)^2\mathbb{E}(\mathrm{X}_2)^2 \\ &= (\sigma_1^2 + \mu_1^2)(\sigma_2^2 + \mu_2^2) - \mu_1^2\mu_2^2, \end{aligned}$$

$$\mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_1,\mathrm{X}_3)) = \mathrm{V}ar(\mathbb{E}(\mathrm{X}_2)\mathrm{X}_1 + \mathrm{X}_3) = \mu_2^2\mathrm{V}ar(\mathrm{X}_1) + \mathrm{V}ar(\mathrm{X}_3) = \mu_2^2\sigma_1^2 + \sigma_3^2,$$

$$\mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_2,\mathrm{X}_3)) = \mathrm{V}ar(\mathbb{E}(\mathrm{X}_1)\mathrm{X}_2 + \mathrm{X}_3 = \mu_1^2\mathrm{V}ar(\mathrm{X}_2) + \mathrm{V}ar(\mathrm{X}_3) = \mu_1^2\sigma_2^2 + \sigma_3^2,$$

$$\begin{aligned} \mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_1)) &= \mathrm{V}ar(\mathrm{X}_1\mathbb{E}(\mathrm{X}_2) + \mathbb{E}(\mathrm{X}_3)) = \mu_2^2\mathrm{V}ar(\mathrm{X}_1) = \mu_2^2\sigma_1^2, \\ \mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_2)) &= \mathrm{V}ar(\mathbb{E}(\mathrm{X}_1)\mathrm{X}_2 + \mathbb{E}(\mathrm{X}_3)) = \mu_1^2\mathrm{V}ar(\mathrm{X}_2) = \mu_1^2\sigma_2^2, \end{aligned}$$

$$\mathrm{V}ar(\mathbb{E}(\mathrm{Y} \mid \mathrm{X}_3)) = \mathrm{V}ar(\mathbb{E}(\mathrm{X}_1)\mathbb{E}(\mathrm{X}_2) + \mathrm{X}_3) = var(\mathrm{X}_3) = \sigma_3^2.$$

Nous pouvons donc maintenant calculer :

— *les indices de Sobol du premier ordre* S_i, $i = 1,2,3$ *qui mesurent la sensibilité de* Y *par rapport à chaque variable d'entrées, sans prendre compte les interactions entre elles.*

$$\begin{aligned} S_1 &= \frac{Var(h_1(X_1))}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_1))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} \\ &= \frac{\mu_2^2\sigma_1^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2}, \end{aligned}$$

$$\begin{aligned} S_2 &= \frac{Var(h_2(X_2))}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_2))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} \\ &= \frac{\mu_1^2\sigma_2^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2}, \end{aligned}$$

$$\begin{aligned} S_3 &= \frac{Var(h_3(X_3))}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_3))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} \\ &= \frac{\sigma_3^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2}. \end{aligned}$$

— *Les indices de Sobol de second ordre* $S_{i,j}$, $i,j = 1,2,3$ *et* $i \neq j$ *mesurant la variance engendrée par l'interaction entre deux variables, les effets du premier ordre étant retiré :*

$$\begin{aligned} S_{1,2} &= \frac{Var(h_{1,2}(X_1,X_2))}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_1,X_2))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} - S_1 - S_2 \\ &= \frac{\sigma_1^2\sigma_2^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2}, \end{aligned}$$

$$\begin{aligned} S_{1,3} &= \frac{Var(h_{1,3}(X_1,X_3))}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_1,X_3))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} - S_1 - S_3 \\ &= \frac{0}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} = 0, \end{aligned}$$

$$\begin{aligned} S_{2,3} &= \frac{Var(h_{2,3}(X_2,X_3))}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_2,X_3))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} - S_2 - S_3 \\ &= \frac{0}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} = 0. \end{aligned}$$

— *L'indice de Sobol de troisième ordre* $S_{1,2,3}$ *mesure l'interaction entre les 3 variables sans prendre en compte les effets de premier et second ordre :*

$$\begin{aligned} S_{1,2,3} &= \frac{Var(h_{1,2,3}(X_1,X_2,X_3)}{Var(Y)} = \frac{Var(\mathbb{E}(Y \mid X_1,X_2,X_3))}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} - S_1 - S_2 - S_{1,2} - S_{1,3} - S_{2,3} \\ &= 1 - \frac{\mu_2^2\sigma_1^2+\mu_1^2\sigma_2^2+\sigma_3^2+\sigma_1^2\sigma_2^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2)-\mu_1^2\mu_2^2+\sigma_3^2} = 0. \end{aligned}$$

— *Finalement nous pouvons calculer les indices totaux de Sobol* S_{T_i}, $i = 1,2,3$ *mesurant l'effet total engendré par l'une des variables sur la variance de Y*

$$S_{T_1} = S_1 + S_{1,2} + S_{1,3} + S_{1,2,3} = \frac{\sigma_1^2\sigma_2^2 + \sigma_1^2\mu_2^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2) - \mu_1^2\mu_2^2 + \sigma_3^2},$$

$$S_{T_2} = S_2 + S_{1,2} + S_{2,3} + S_{1,2,3} = \frac{\sigma_1^2\sigma_2^2 + \sigma_2^2\mu_1^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2) - \mu_1^2\mu_2^2 + \sigma_3^2},$$

$$S_{T_3} = S_3 + S_{1,3} + S_{2,3} + S_{1,2,3} = \frac{\sigma_3^2}{(\sigma_1^2+\mu_1^2)(\sigma_2^2+\mu_2^2) - \mu_1^2\mu_2^2 + \sigma_3^2}.$$

Cet exemple soulève plusieurs points intéressants. Pour commencer le modèle $Y = X_1X_2 + X_3$ est construit avec une variable sans interaction : X_3 et 2 variables en interaction double : X_1 et X_2, il n'y a pas d'interaction triple. On remarque que l'indice de troisième ordre $S_{1,2,3}$ est nul et que les indices de second ordre relatifs à X_3, $S_{1,3}$ et $S_{2,3}$, sont nuls puisqu'il n'y a pas d'interaction entre ces variables. Sans interaction, l'indice totale S_{T_3} est égal à l'indice de premier ordre. Dans le cas des variables identiquement distribuées, les variables avec interaction ont un indice total plus élevé que les variables sans interaction. Enfin il est à noter l'importance donnée aux lois des variables. Par exemple si l'on prend le cas de X_1, X_2 qui jouent le même rôle dans le modèle et que $\mu_1 = 0, \mu_2 = 1$ et $\sigma_1^2 = \sigma_2^2 = 1$ alors $S_{T_1} > S_{T_2}$, ce qui ne reflète pas correctement le symétrie de ces 2 variables.

4.2 Analyse de la sensibilité appliquée

Grâce aux indices de Sobol appliqués aux modélisations de la blessure sans contact chez les sportifs professionnel, nous pouvons obtenir plus d'informations sur les paramètres participant aux mécanismes de la blessure.

Dans cette section, nous allons montrer les variables intervenant dans la modélisation de la blessure par réseau de neurones. Pour cela nous avons choisi le réseau de neurones qui avait le meilleur résultat (AUC à 0.766), celui à trois couches cachées définie par :

— une couche de 30 neurones, fonction d'activation : RELU,

— une couche de 30 neurones, fonction d'activation : tangente hyperbolique,

— une couche de 15 neurones, fonction d'activation : tangente hyperbolique,

— couche de sortie 1 neurone, fonction d'activation : sigmoïde.

L'un des avantages des réseaux de neurones est qu'ils sont hautement non linéaire et peuvent donc faire intervenir des interactions entre les variables. Cependant leur fonctionnement interne est inconnu et non accessible. Il est donc compliqué d'extraire les paramètres importants. L'utilisation de l'analyse de la sensibilité permet donc de comprendre l'influence des variables sur la modélisation. La courbe ROC obtenue par le modèle est présentée dans la figure 4.1. Le premier indice à observer est l'indice total S_{T_i} (figure 4.2). Il représente l'impact total d'une variable sur la variable réponse, son effet principal (de premier ordre) plus toutes ses interactions (d'ordre 2 ou supérieur).

La variable ayant le plus d'impact est la variable "identifiant du joueur". Son indice total est de 0.652. Cela nous montre l'importance d'avoir ajouté cette variable à la modélisation et donc le caractère personnel de la blessure. Cela peut également nous amener à penser qu'il y a certain profil de joueurs à risque se blessant régulièrement. Et un autre profil de joueurs se blessant rarement.

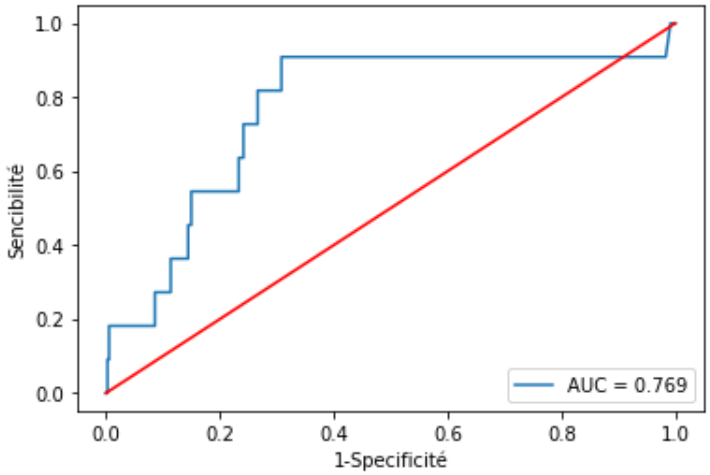


FIGURE 4.1 – Courbe ROC du modèle utilisé pour l'analyse de la sensibilité

Cette hypothèse peut être confirmée par le fait que la variable "indice de récidive" est la troisième variable la plus influente. Son indice total est de 0.234. Ainsi les joueurs avec un indice haut se blessent, ce qui corrobore avec l'hypothèse que certains joueurs se blessent plusieurs fois de suite puisque cette indice indique une période à risque après une blessure.

La seconde variable la plus influente, est la charge de travail dont l'indice total est à 0.614. Nous ne sommes pas surpris que cette variable ait un impact majeur sur la blessure. Elle est au cœur de la préparation physique des joueurs et de nombreux articles ont déjà montré son application dans le mécanisme de la blessure, au même titre que le temps de récupération qui a un indice de 0.187.

Enfin, les variables GPS n'ont quasi aucune influence sur la prédiction de blessures puisque la variable ayant le plus grand indice est le nombre d'accélérations, avec une valeur à 0.005.

Les indices de Sobol d'ordre 1 sont présentés dans la figure 4.3a. La figure 4.3b présente le pourcentage expliqué de l'indice total par l'indice d'ordre 1, i.e. le ratio S_i/S_{T_i} pour $i = 1,...,8$. Il est intéressant de voir que l'indice d'ordre 1 de la variable "identifiant du joueur", qui est la plus influente, explique seulement 26.6% de l'indice total. Le même commentaire s'applique à la variable "charge de travail", avec seulement 14.9% de l'indice total expliqué par l'effet d'ordre 1. Ceci montre qu'il existe de fortes interactions entre les variables, et confirme l'importance des variables "charge de travail" et "identifiant du joueur".

Les effets d'ordre 2 sont présentés dans la figure 4.4. Chaque case de la matrice représente un indice de Sobol d'ordre 2.

La plus forte interaction est entre l'identifiant joueur et la charge de travail (0.205). On peut donc naturellement penser que la charge optimale, i.e. la charge où le joueur est le plus performant et où son risque de blessure est le plus faible, est différente selon les individus. C'est une donnée à prendre en compte par les équipes médicales, et montre que l'encadrement des joueurs doit être individualisé. Ensuite la charge de travail est en interaction avec les autres variables avec des indices proches de 0.06. Finalement, trois variable GPS qui avaient des indices totaux proche de 0, ont une interaction avec la charge mais également l'identifiant du joueur.

L'analyse de la sensibilité et les indices de Sobol permettent d'améliorer la compréhension de notre modélisation. Ils permettent de montrer les variables importantes, mais pas de quelle manière elle influent. Par exemple, la charge de travail est importante mais faut-il avoir une charge de travail élevée ou plutôt faible? Faut-il jouer toutes les semaine? plusieurs fois par semaine? Pour répondre à ces questions nous allons montrer la variation du risque de blessures en fonction des variables explicatives.

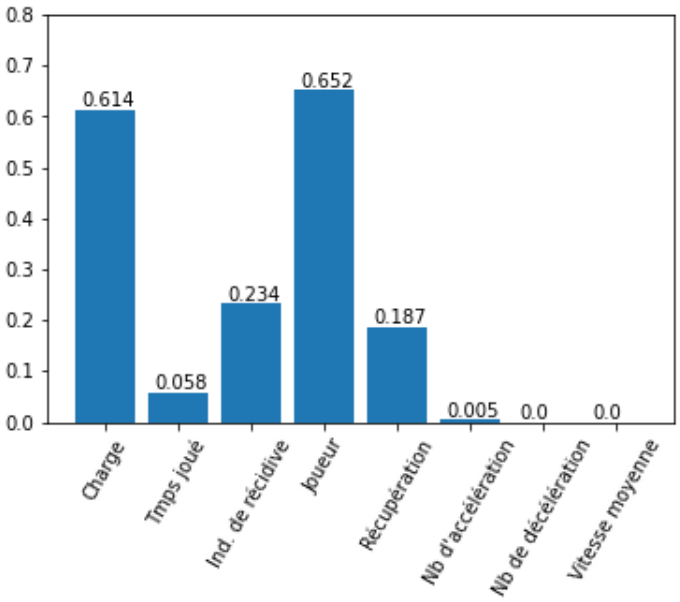
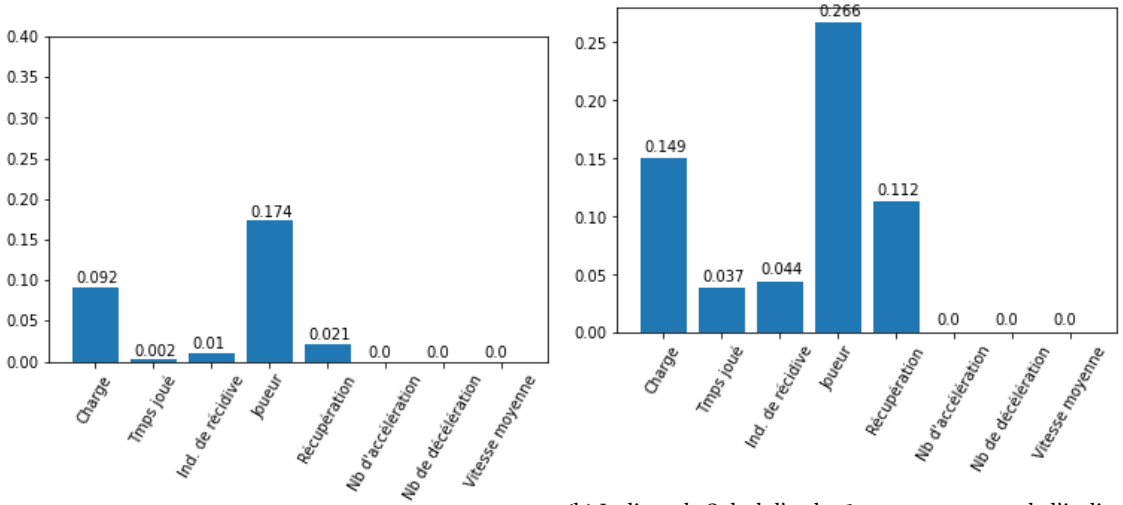


FIGURE 4.2 – Indices de Sobol totaux



(a) Indices de Sobol d'ordre 1

(b) Indices de Sobol d'ordre 1 en pourcentage de l'indice total respectif

FIGURE 4.3 – Indice d'ordre 1

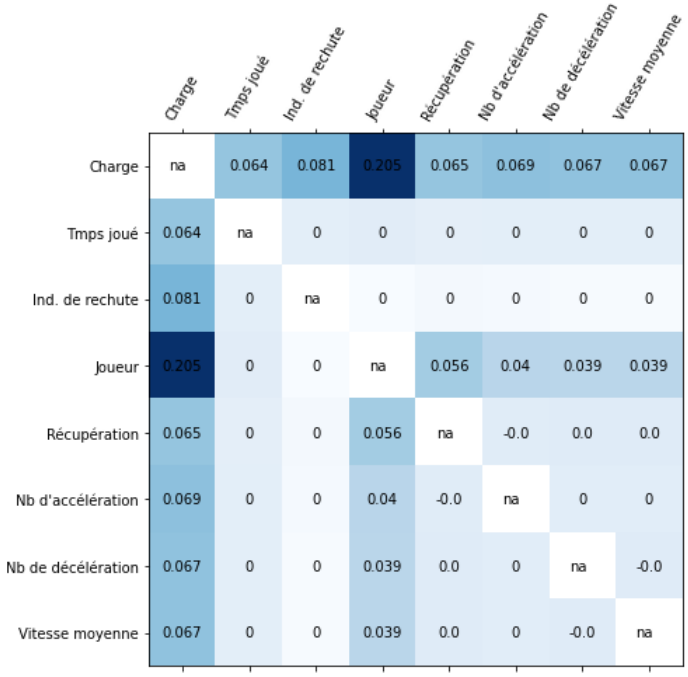


FIGURE 4.4 – Indices de Sobol d'ordre 2

4.3 Risque individuel

Nous avons parlé, tout au long de ce manuscrit, de l'importance d'individualiser le risque. Nous avons montré l'apport de la variable joueur comme effet fixe sur la bonne prédiction des blessures. L'analyse de sensibilité a également montrée que c'était une des variables les plus influentes. Elle a également montré que les variables : charge de travail, temps de jeu et indice de récidive avaient une part importante dans la variation du risque de blessure.

Cependant, nous n'avons toujours aucune idée de la manière dont la variation des variables influe sur la probabilité de se blesser. Pour essayer de visualiser cette influence nous allons suivre quatre joueurs du club ayant des profils différents. Nous analyserons comment varie leur risque de blessure lorsque les variables varient. Nous appellerons ces joueurs 1,2,3 et 4. Il est important de comprendre la différence entre ces joueurs. Pour cela nous allons détailler leur profil. Cependant par souci de confidentialité, les chiffres données sont indicatifs. Les profils des joueurs sont les suivant :

- **Joueur 1 :**
  - Classe d'âge : 29-35 ans.
  - Près de 500 matchs joués dans sa carrière. L'intégralité de ces matchs ont été joué dans la première ligue de plusieurs championnats européens.
  - Sélections en équipe nationale.
  - Participation des compétitions européennes.
  - Titulaire 70% des matchs de son équipe.
  - Plus de 3 blessures durant la saison 2018-2019.
- **Joueur 2 :**
  - Classe d'âge : 29-35 ans.
  - Près de 500 matchs joués dans sa carrière dont plus de 200 en ligue 1.
  - Participation à des compétitions européennes.
  - Titulaire dans Près de 60% des matchs de son équipe.
  - Plus de 3 blessures durant la saison 2018-2019.
- **Joueur 3 :**
  - Classe d'âge : 26-29 ans.
  - Près de 260 matchs joués dans sa carrière. 130 de ces matchs ont été joués dans la première ligue de plusieurs championnats européens.
  - Sélection en équipe nationale.
  - Aucune participation aux compétitions européennes.
  - Titulaire à 60% des matchs de son équipe environ.
  - Moins de 3 blessures durant la saison 2018-2019.
- **Joueur 4 :**
  - Classe d'âge : 17-21 ans.
  - Près de 60 matchs joués dans sa carrière. L'intégralité de ces matchs ont été joués dans la première ligue de plusieurs championnats européen.
  - Sélection en équipe nationale.
  - Participation des compétitions européennes.
  - Titulaire à 50% des matchs de son équipe environ.
  - Moins de 3 blessures durant la saison 2018-2019.

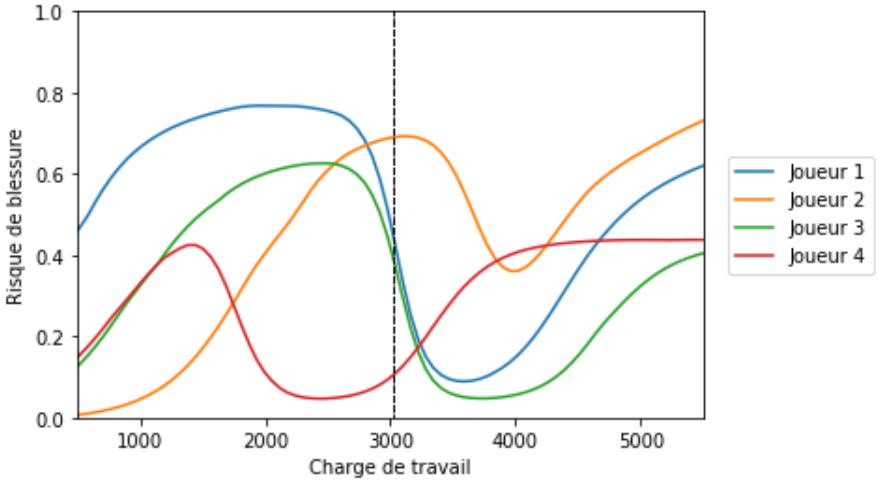


FIGURE 4.5 – Effet de la variable charge de travail sur le risque de blessure

Les 4 joueurs choisis sont des profils différents, en commençant par leur âge : un jeune joueur de 19 ans, un joueur de 27 et deux plus âgés de 33 et 34 ans. En effet nous l'avions vu lorsque nous décrivions le modèle initial (section3.2.1), plus l'âge augmente, plus le risque de blessures augmente. Leur poste est également différent puisque il y a deux défenseurs et deux joueurs plus offensifs (milieu offensif et attaquant). Enfin, le point le plus important est leur nombre de blessures qui est différent. En effet deux des joueurs ont eu plus de trois blessures sur la saison 2018-2019 alors que les deux autres en ont eu moins de deux. Nous avons donc des profils de joueurs à risque et d'autres plus résistants.

Pour montrer l'impact des variables sur le risque de blessure, nous allons graphiquement tracer le risque de blessure en fonction d'une variable. Les autres variables ont été fixées à leur moyenne respectives. Cela permet de visualiser uniquement l'effet de la variable analysée. Pour obtenir le risque de blessure nous utiliserons le même modèle que celui utilisé pour l'analyse de la sensibilité, à savoir, le réseau de neurones à trois couches cachées.

Sur chaque figure, la moyenne de la variable sera indiquée par un trait noir en pointillé. Nous commencerons par analyser la charge de travail.

Charge de travail

L'analyse de notre modèle par la méthode de Sobol nous a montré que la charge de travail jouait un rôle important. En effet, son indice total était de 0.614. L'effet de cette dernière sur le risque de blessure est présentée sur la figure 4.5. Dans un premier temps, nous voyons que lorsque la charge est faible (0-1000) le risque diminue. Cela pourrait nous laisser penser qu'un joueur peu entrainé est protégé de la blessure. Cependant, nous pouvons attribuer ce phénomène à un faible nombre de joueurs se situant dans cette zone (moins de 1.5% des observations). Comme nous avons peu d'observations dans cette zone, le modèle extrapole, il est donc moins précis. Habituellement un joueur peu entrainé commence un match s'il a au minimum une charge de 1000.

Si l'on regarde les quatre courbes, on peut trouver trois zones caractéristiques :

— Sous-entrainement : les joueurs ne sont pas assez entrainés et présentent donc un risque de blessure élevée. Pour les joueurs 1 et 3, le pic se situe entre 1000 et 2500 de charge. La probabilité de blessure au plus haut est environ de 0.7. Le joueur 4 a un pic, plus étroit entre 1000 et 2000, avec une probabilité plus faible 0.4. Le joueur 2 plus singulier, est à risque plus tard que les 3 autres entre 2500 et 4000. Pourtant son plus haut risque est quasiment égal à celui des joueurs 1 et 3.

— Zone optimale : les joueurs sont entrainés selon leur besoin. Le risque de blessure est plus faible. Elle se situe entre 3200 et 4500 pour les joueurs 1 et 3. Concernant les joueurs 1,3 et

4, elle est très protectrice, leur probabilité diminuant jusqu'à 0.05. Pour le joueur 4, elle se situe entre 2000 et 3200. Concernant le joueur 2, la zone est très étroite, entre 3700 et 4200. De plus le risque reste très élevé puisque, même ici, la probabilité de blessure est de 0.4.

— Sur-entrainement : les joueurs sont en sur-charge de travail. Le risque de blessure augmente. Pour les joueurs 1,2 et 3, cette zone commence à partir de 4500. Pour le joueur 4, elle débute à partir de 3300.

Nous pouvons dire que globalement le staff respecte ces trois zones. En effet, la majorité des joueur (75%) débutent un match avec une charge entre 2000 et 4000. Cela peut correspondre à la charge de travail optimal. Cependant il faut faire attention puisque cette zone dépend du profil des joueurs. Par exemple, pour le joueur 4, son risque augmente à partir de 3000 de charge. Il faut également préciser que 5% des joueurs sont en sous-charge (moins de 2000) avant de débuter un match, et plus de 10% ont une charge élevée (plus de 4000). Au total, c'est donc 15% observations qui ont débuté un match dans une zone à fort risque.

Temps joué en match

Le temps en match est un facteur important dans la blessure. Il est inévitablement lié à celle-ci, puisque plus le joueur passe du temps en match plus il risque de se blesser. La figure 4.6, présente la fluctuation du risque de blessure en fonction du temps joué. Un point intéressant est la différence de risque de blessure lorsque les joueurs débutent un match sans avoir joués durant les trois dernières semaines. En effet, deux duos se forment. Le premier avec un risque élevé entre 0.7 et 0.8 et le second avec un risque faible entre 0.05 et 0.1. C'est d'autant plus intéressant que les profils dans chaque duo sont différents. En effet, chaque duo est constitué d'un joueur de plus de 30 ans et d'un plus jeune. Mais également d'un joueur qui se blesse souvent et d'un autre plus résistant.

Comme pour la charge de travail nous pouvons définir les trois zones au moins pour les joueurs 1,2 et 3 :

— Manque de temps de jeu : le joueur est en manque de rythme, son risque de blessure est élevé. Pour les joueurs 1 et 3, cette zone et entre 0 à 170. Pour ces deux joueurs le risque est élevé (0.7 à 0.8). Le joueur 3 est en manque de rythme entre 100 et 200 minutes joués en 3 semaines. Son risque maximum est de 0.45, plus faible que pour les joueurs 1 et 3.

— Temps de jeu optimal : Le joueur est dans son rythme optimal. Le risque de blessure est plus faible. Pour les joueurs 1 et 3 cette zone est entre 180 et 330 ce qui correspond à 2-3 matchs par 3 semaines. Pour le joueur 3 c'est entre 260 et 320, soit 1 match par semaine

— Sur-utilisation du joueur : le joueur est trop sollicité, il accumule de la fatigue, son risque de blessure augmente. Pour les 3 joueurs cette zone est atteinte à partir de 330 minutes jouées en 3 semaines. Le risque final des 3 joueurs est différent, 0.4 pour le joueur 2, 0.65 pour le 3 et 0.85 pour le 1.

Ces zones ne sont pas retrouvés chez le joueur 4. En effet il n'en présente que deux : une où le joueur est à risque de blessure faible de 0 à 200 minutes jouées, le risque étant à 0.05. Puis ensuite le risque augmente à partir de 220 jusqu'à attendre son maximum de 0.4.

Un chose intéressante est le duo de joueurs 1 et 3 qui ont une courbe quasiment identique alors que leur profil est différent. De plus, bien que le joueur 3 a un risque plus important que le joueur 1 au départ, cette ordre s'inverse lorsque le temps de jeu dépasse 330 minutes. Le joueur 3 est donc plus à risque lorsqu'il joue peu mais est plus résistant lorsque les matchs s'enchainent. Il faut encore une fois noter le travail de l'équipe encadrant puisque seulement 4% des joueurs ont joué plus de 400 minutes. En revanche prés de 25% des joueurs débutent un match avec moins de 90 minutes jouées sur les trois dernières semaines. C'est un phénomène inévitable, puisque après chaque rupture de saison (pour blessure ou trêve), le joueur doit retrouver son rythme et faire son premier et second matchs. C'est donc à ce moment que l'équipe médical doit être particulièrement attentive, et traiter les joueurs de manière individuelle.

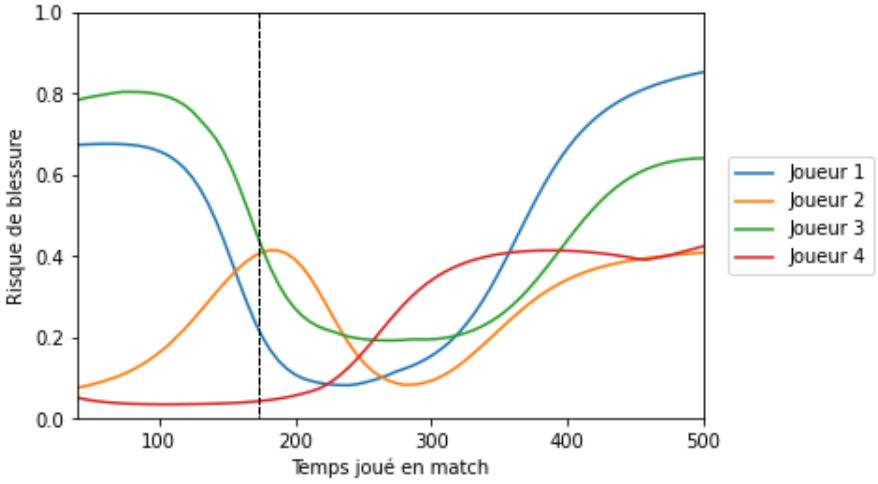


FIGURE 4.6 – Effet de la variable temps joué en match sur le risque de blessure

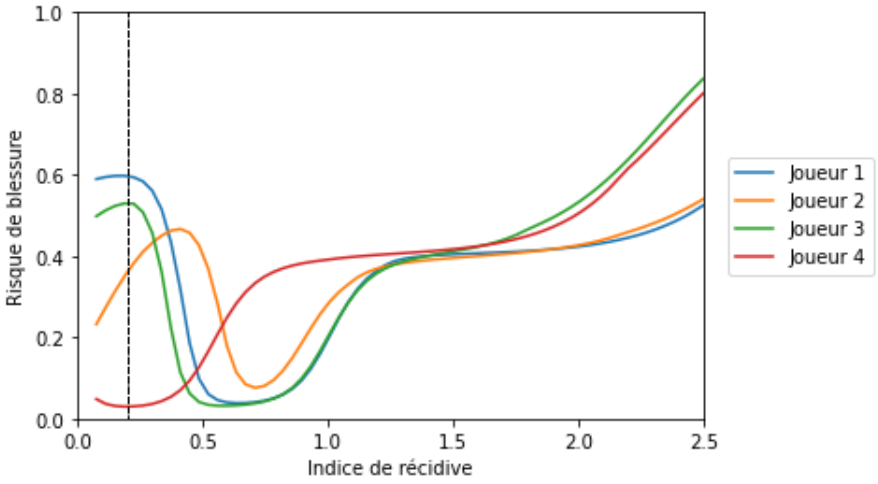


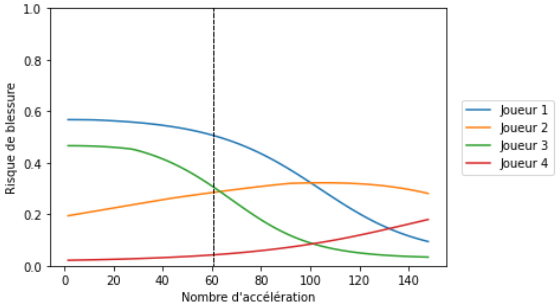
FIGURE 4.7 – Effet de la variable indice de récidive sur le risque de blessure

Indice de récidive

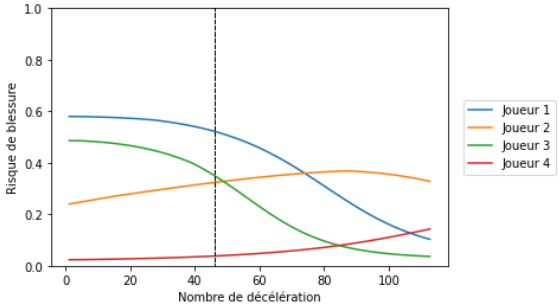
Selon l'analyse de la sensibilité, l'indice de récidive est la variable la plus influente après l'identifiant du joueur. Sur la figure 4.7, on observe que lorsque l'indice est faible (proche de 0), le risque de blessure est élevé. Il est possible que ce soit un biais produit par la construction de l'indice. En effet, ce dernier a été créé pour mesurer le risque qu'un joueur se reblesse. Or souvent, lors de la première blessure du joueur, cette indice est proche de 0, ce qui peut expliquer ce phénomène. Après cette zone à risque, on trouve une zone de protection entre 0.5 et 1. Ensuite, plus l'indicateur augmente, plus le risque augmente.

Variables nombre d'accélérations et décélérations

Les indices de Sobol précédemment calculés montraient que ces deux variables n'influaient pas sur le risque de blessure. Pourtant lorsque l'on regarde la figure 4.8, on observe deux duos différents. Les joueurs 2 et 4 dont l'augmentation du nombre d'accélérations augmente le risque de blessure. Pour les joueurs 1 et 3, cette augmentation en revanche entraine une baisse du risque. Pour le premier duo, l'augmentation reste modérée (environ 0.1 point en plus), alors que pour le second duo la diminution est de 0.5. Cela nous montre que la pratique de certains exercices permet de gérer le risque de blessure chez les joueurs. Enfin, nous pouvons noter que les deux variables accélérations et décélérations ont le même effet.



(a) Effet de la variable accélération sur le risque de blessure



(b) Effet de la décélération sur le risque de blessure

FIGURE 4.8 – Variables GPS

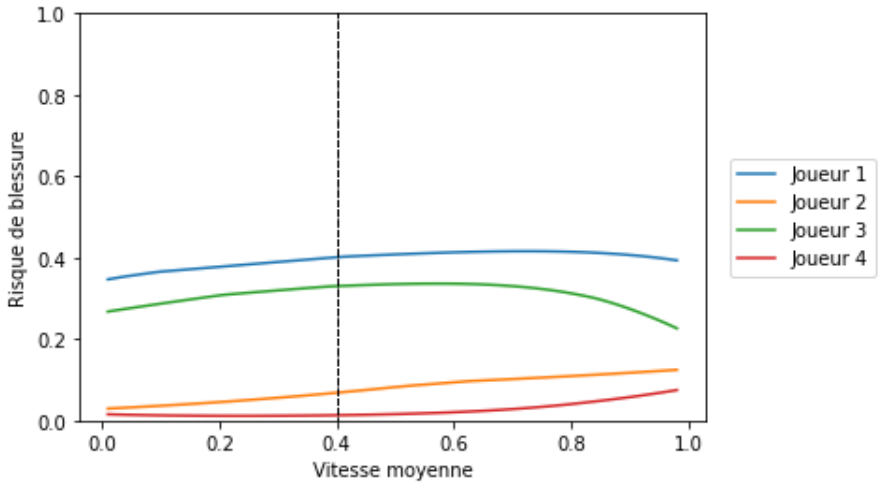


FIGURE 4.9 – Effet de la variable vitesse moyenne sur le risque de blessure

Variable vitesse moyenne

La vitesse moyenne était l'une des variables les moins influentes. La figure 4.9 confirme cette première conclusion. En effet, la variation est identique pour tous les joueurs. L'augmentation de la vitesse moyenne à l'entrainement augmente faiblement le risque de blessure (seulement 0.1)

4.3.1 Profil des joueurs

Nous avons analysé globalement les variables dans la partie précédente. Nous avons indiqué, lorsque c'était intéressant, les différences entre les joueurs. Dans cette partie, nous nous intéresseront plus en détail à chaque joueur, et nous essayerons de construire leur profil.

- **Joueur 1 :**
  - — Joueur à haut risque de blessure. Il a subi plusieurs blessures durant la saison 2018-2019. Comme nous pouvons le voir sur l'ensemble des graphiques, son risque est systématiquement plus élevé lorsqu'il se trouve dans des zone de sous/sur-utilisation.
  - — Il a besoin d'un contrôle de sa charge de travail précise. Sa zone de confort se trouvent entre 3000 et 4200. Il résiste mieux à des charges élevées qu'à des charges faibles. C'est donc un joueur qui a besoin d'une charge de travail importante pour être à son meilleur niveau. On le voit notamment sur les graphiques (figure 4.8), son risque de blessure diminue lorsque le joueur pratique des accélérations et décélérations.

  - Concernant le temps de jeu, c'est un joueur qui a besoin de rythme. Il est dans sa forme optimale lorsqu'il joue entre 180 et 310 minutes pour 3 semaines. Cependant, il résiste mieux à un sous-régime qu'à une sur-régime.
  - Profil du titulaire, il supporte une charge de travail et un enchainement de matchs. Cependant c'est un joueur à haut risque de blessure. Il faudra éviter de le mettre en sur-régime.
- **Joueur 2 :**
  - Joueur à haut risque de blessure. Il a subi plusieurs blessures durant la saison 2018-2019.
  - Il nécessite une attention particulière sur le contrôle de la charge de travail. En effet sa zone de sécurité est très étroite entre 3500 et 4200. Et même dans cette zone le risque reste encore élevé.
  - En revanche c'est un joueur qui est peu influencé par l'enchainement des matchs. Même si ça zone d'état de forme maximale se situe entre 220 et 330, son risque augmente peu lorsque le joueur est en sous ou sur-régime.
  - Profil du titulaire, il peut enchainer les matchs sans se mettre dans une zone de risque élevé. En revanche le contrôle de sa charge doit être précise pour ne pas le mettre en risque de blessure.
- **Joueur 3 :**
  - Joueur à faible risque de blessure.
  - Profil presque identique au joueur 1 (qui est à risque de blessure élevé). Cependant, il supporte mieux le sous-régime que le sur-régime. Profil du titulaire, il supporte une charge de travail et un enchainement des matchs. Cependant, ce joueur supporte mal le sous-régime.
- **Joueur 4 :**
  - Joueur à faible risque de blessure.
  - Joueur qui a besoin de peu travail. En effet ça charge optimal est à un niveau faible entre 1800 et 3000.
  - C'est un joueur qui n'a pas besoin de temps de jeu pour être efficace. En revanche plus il enchaine les matchs, plus son risque augmente.
  - Comme le montre l'indicateur de récidive, c'est un joueur qui aura tendance à enchainer les blessures.
  - Profil de remplaçant parfait, qui n'a pas besoin de temps de jeu pour être protégé.

4.4 Références

CARIBONI, J., D. GATELLI, R. LISKAA et A. SALTELLI. 2007, «The role of sensitivity analysis in ecological modelling», *ecological modelling*. 96

CHAN, K., A. SALTELLI et S. TARANTOLA. 1997, «Sensitivity analysis of model output : Variance-based methods make the difference.», p. 261–268. 96

HOEFFDING, W. 1948, «A class of statistics with asymptotically normal distribution.», *The annals of Mathematical Statistics*. 97

HOMMA, T. et A. SALTELLI. 1996, «Importance measures in global sensitivity analysis of nonlinear models», *Reliability Engineering and System Safety*, vol. 52, n° 1, p. 1 – 17, ISSN 0951-8320. 100

I. M., S. 1993, «Sensitivity estimates for nonlinear mathematical models.», *Mathematical Modelling and Computational Experiments*. 97

I. M., S. 2001, «Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates», *Mathematics and Computers in Simulation*. 96, 99

MCKAY, M. 1995, «Evaluating prediciton uncertainty», *US Nuclear Regulatory Commission and Los Alamos National Laboratory*. 99

MORRIS, M. D. 1991, «Factorial sampling plans for preliminary computational experiments», *Technometrics*. 96

ZHOU, X. et H. LIN. 2008, «Local sensitivity analysis», *EDDS. springer US, Boston*. 96

Chapitre 5

Conclusion

La compréhension de la blessure musculaire sans contact est un sujet de recherche très actif dans le milieu du sport professionnel. En effet, un bonne connaissance de ses facteurs de risque permettrait de mieux protéger l'athlète. En particulier, nous souhaiterions obtenir une bonne évaluation de la probabilité du risque de blessure, permettant ainsi de faire jouer ou non les joueurs à risque, au moment opportun. Pour tenter de répondre à cette problématique, nous avons construit un modèle statistique capable de prédire ce type de blessure.

Pour commencer, nous avons dû améliorer notre connaissance des facteurs influant nos modélisations.

Ces connaissances en notre possession, nous avons construit nos premiers modèles en utilisant des méthodes classiques de modélisation telles que la régression logistique ou les supports vector machines. La première difficulté a été de valider les modèles obtenus. Les données que nous utilisions étant longitudinales, il était impossible d'utiliser les méthodes classiques de validation. Ces méthodes ne prennent pas en compte la temporalité des données ce qui peut amener à confondre les causes et les conséquences de la blessure. Souhaitant évaluer le modèle dans les conditions de son utilisation en vie réelle, nous avons mis en place la méthode de validation longitudinale. Cette méthode, en plus de tenir compte de l'ordre des données, nous permettait de tenir compte du passé le plus proche du présent.

Cette validation longitudinale nous a permis de comparer nos premiers modèles entre eux. Rapidement nous nous sommes confrontés aux problématiques des données déséquilibrées et plus particulièrement des évènements rares. En effet, les résultats étaient peu fiables et les prédictions de blessures sous-évaluées. Nous nous sommes dans un premier temps tournés vers les méthodes d'échantillonnage.

Ces dernières permettent de rééquilibrer les données. Il existe une grande quantité de méthodes d'échantillonnage. Les plus simples dupliquent ou suppriment des observations. C'est le cas de random over/under sampling. D'autres, plus évoluées, modifient l'espace en construisant des observations synthétiques. Nous avons montré à travers plusieurs exemples que les méthodes les plus complexes ne donnaient pas les meilleurs résultats. Sur nos données l'effet était même contraire, elles diminuaient les valeurs de l'AUC. Les méthodes plus simples, en l'occurrence le random sampling, les augmentent. Cependant, nous avons également montré qu'il faut utiliser ces méthodes avec attention et bien maîtriser le taux d'équilibrage. En effet selon les taux, l'échantillonnage peut être néfaste pour les modélisations. Nous l'avons montré par exemple pour la régression logistique et l'équilibrage total par undersampling qui entraine une perte d'information trop grande de la classe majoritaire et donc une perte de performance prédictive, ou encore avec les méthodes de SVM combinées à l'oversampling, un petit rééquilibrage rendait la modélisation presque aléatoire, alors qu'un équilibrage fort donnait le meilleur résultat obtenu. Bien sûr ces méthodes, ajoutant de l'aléa dans les données, entrainaient une hausse de la variabilité. L'étape suivante était donc de s'intéresser aux méthodes d'agrégation.

Elles permettent d'agréger différents modèles afin d'en obtenir un plus performant. De nombreuses recherches portent sur ces méthodes qui montrent des résultats très intéressants. Cepen-

dant sur nos données, les méthodes d'agrégation n'ont pas permis d'améliorer significativement les résultats. Les méthodes de bagging apportent un gain léger sur les résultats. Les autres méthodes comme le boosting semblent trop sensibles aux évènements rares et entrainent une diminution des performances prédictives.

En revanche, les méthodes agrégatives ont aidé à diminuer la variation de prédiction et donc ont contribué à rendre les modèles utilisables. En effet sans ces méthodes, il est compliqué d'utiliser les modèles construits avec des méthodes d'échantillonnage. La variabilité des résultats est trop grande, les probabilités trop fluctuantes, ce qui rend l'interprétation compliquée. Nous conseillerions donc d'utiliser systématiquement une méthode d'agrégation en complément des méthodes d'échantillonnage.

L'exploration de toutes ces méthodes et de leurs combinaisons nous a permis de mettre en évidence les modèles qui produisaient les meilleurs résultats. La méthodes avec le plus haut AUC (0.81), est la combinaison des SVM utilisant un noyau linéaire, un équilibrage par random oversampling de 10 : 3 et une agrégation par moyenne simple. Ce modèle est capable de prédire correctement 81% des blessures et 74% des non-blessures. Le second modèle à retenir utilise une agrégation de régressions logistiques combinée à un random undersampling 0.5 et un random oversampling 1 : 1. Son AUC est de 0.79. Il permet de détecter 81% des blessures et 75% des non-blessures.

Nous étions donc en mesure de prédire ainsi la plupart des blessures survenant. L'étape suivante était de comprendre les facteurs entrainant un risque de blessures, pour pouvoir le contrôler. L'analyse de la sensibilité et les indices de Sobol nous ont permis de comprendre qu'elles étaient les variables influentes. Ceci nous a permis de mettre en évidence le caractère personnel de la blessure. En effet la variable identifiant du joueur avait l'indice le plus haut, suivi par les variables charge de travail, indice de récidive et temps de récupération. Nous avons également montré qu'il existait une interaction entre les variables, rendant la modélisation plus complexe. Il nous restait à comprendre encore comment ces variables influaient sur le risque de blessure. En suivant 4 joueurs et en présentant la variation de leurs indices en fonction des différentes variables, nous avons réussi à montrer plusieurs éléments :

— la variation du risque n'est pas linéaire,

— chaque joueur est unique et n'est pas impacté de la même manière par la charge d'entrainement ou l'enchainement des matchs.

Par la suite, des améliorations pourront encore être apportées. Il reste de nombreux facteurs de risque à étudier et à ajouter aux modèles :

— les facteurs environnementaux, tel que le type de terrain ou les conditions météorologiques,

— les facteurs médicaux, comme l'asymétrie de travail des groupes musculaires par exemple,

— le type de blessure, non pris en compte pour l'instant, qui pourrait l'être dans l'indice de récidive. Chaque blessure n'ayant pas le même impact, ni le même temps de guérison, l'indice pourrait être plus ou moins pondéré.

Sur le plus long terme, nous pourrions imaginer de produire le risque pour chaque type de blessures.

Enfin, il faudrait être capable de généraliser et automatiser l'analyse du risque en fonction des variables. Cela permettrait d'expliquer à chaque instant les paramètres entrainant une hausse, et de pouvoir les réguler pour protéger le joueur et ainsi éviter la blessure.

Bibliographie

5.1 Références

ABD RANI, K., H. A. ABD RAHMAN, S. FONG, K. ZURAIDA et N. ABDULLAH. 2013, «An application of oversampling, undersampling, bagging and boosting in handling imbalanced dataset», .

ABRIL, L. G., H. NÚÑEZ, C. ANGULO et F. V. MORENTE. 2014, «Gsvm : An svm for handling imbalanced accuracy between classes inbi-classification problems», *Appl. Soft Comput.*, vol. 17, p. 23–31.

AHA, W., D. KIBLER et M. ALBERT. 1991, «Instance-based learning algorithms», *Machine Learning*, vol. 6, p. 37–66.

ARNI, A., S. STEFAN et E. LARS. 2004, «Risk factors for injuries in football», *The American Journal of Sports Medicinel*.

AYALA, F., A. LÓPEZ-VALENCIANO, J. GÁMEZ MARTÍN, M. DE STE CROIX, F. VERA-GARCIA, M. GARCÍA-VAQUERO, I. RUIZ-PÉREZ et G. MYER. 2019, «A preventive model for hamstring injuries in professional soccer : Learning algorithms.», *Int J Sports Med*.

BATISTA, G., A. BAZZAN et M.-C. MONARD. 2003, «Balancing training data for automated annotation of keywords : a case study.», *the Proc. Of Workshop on Bioinformatics*, p. 10–18.

BATISTA, G., R. PRATI et M.-C. MONARD. 2004a, «A study of the behavior of several methods for balancing machine learning training data», *SIGKDD Explorations*, vol. 6, p. 20–29.

BATISTA, G., R. PRATI et M.-C. MONARD. 2004b, «A study of the behavior of several methods for balancing machine learning training data», *SIGKDD Explorations*, vol. 6, doi :10.1145/1007730.1007735, p. 20–29.

BATUWITA, R. et V. PALADE. 2010, «Efficient resampling methods for training support vector machines with imbalanced datasets», *The 2010 International Joint Conference on Neural Networks (IJCNN)*, p. pp. 1–8.

BATUWITA, R. et V. PALADE. 2010, «Efficient resampling methods for training support vector machines with imbalanced datasets», , p. 1–8.

BÜHLMANN, P. et B. YU. 2002, «Analyzing bagging», *Annals of Statistics*, vol. 30.

BILLY, T. H., G. TIM J, W. L. DANIEL, C. PETER et S. JOHN A. 2015, «The acute :chronic workload ratio predicts injury : high chronic workload may decrease injury risk in elite rugby league players», *Br J Sports Med*.

BISHOP, C. 2006, *Pattern Recognition and Machine Learning*.

BREIMAN, L. 1996, «Bagging predictors», *Springer*.

BREIMAN, L. 1996, «Bagging predictors», *Machine Learning*, vol. 24, p. 123–140.

BREIMAN, L. 1998, «Arcing classifiers», *The Annals of Statistics*, vol. 26, n° 3, p. 801–824.

BREIMAN, L. 1999, «Prediction games and arcing algorithms», *Neural Computation*, vol. 11, p. 1493–1517.

BREIMAN, L. 2000, «Bias, variance , and arcing classifiers», *Technical Report 460, Statistics Department, University of California*.

BREIMAN, L., J. FRIEDMAN, C. STONE et R. OLSHEN. 1984, «Classification and regression trees», .

BREIMAN, M. 2001, «Random forests», *Machine Learning*, vol. 45(1), p. 5–32.

BRESLOW, N. E. 1996, «Statistics in epidemiology : The case-control study», *Journal of the American Statistical Association*, vol. 91, n° 433, p. 14–28.

BROOMHEAD, D. et D. LOWE. 1988, «Radial basis functions, multi-variable functional interpolation and adaptive networks», *ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM)*, vol. RSRE-MEMO-4148.

CALABRESE, R. et S. OSMETTI. 2015, «Improving forecast of binary rare events data : A gam-based approach», *Journal of Forecasting*, vol. 34, doi :10.1002/for.2335.

CARIBONI, J., D. GATELLI, R. LISKAA et A. SALTELLI. 2007, «The role of sensitivity analysis in ecological modelling», *ecological modelling*.

CHAN, K., A. SALTELLI et S. TARANTOLA. 1997, «Sensitivity analysis of model output : Variance-based methods make the difference.», p. 261–268.

CHAWLA, N., K. BOWYER, L. HALL et W. KEGELMEYER. 2002, «Smote : Synthetic minority oversampling technique», *J. Artif. Intell. Res. (JAIR)*, vol. 16, p. 321–357.

CHEN, T. et C. GUESTRIN. 2016a, «Xgboost : A scalable tree boosting system», p. 785–794, doi : 10.1145/2939672.2939785.

CHEN, T. et C. GUESTRIN. 2016b, «Xgboost : A scalable tree boosting system», p. 785–794, doi : 10.1145/2939672.2939785.

CHOMBOON, K., P. CHUJAI, P. TEERARASSAMMEE, K. KERDPRASOP et N. KERDPRASOP. 2015, «An empirical study of distance metrics for k-nearest neighbor algorithm», , p. 280–285.

CORTES, C. et V. VAPNIK. 2009, «Support-vector networks», *Chem. Biol. Drug Des.*, vol. 297, p. 273–297.

CRISTIANO, E., T. J L, F. ABDULAZIZ, S. FATEN et C. HAKIM. 2012, «Low injury rate strongly correlates with team success in qatari professional football», *Br J Sports Med*.

CUTLER, A., D. CUTLER et J. STEVENS. 2011, «Random forests», *Machine Learning - ML*, vol. 45, doi :10.1007/978-1-4419-9326-7_5, p. 157–176.

DANIEL, C. et R.-G. JAVIER. 2017a, «The prevalence of injuries in professional soccer players», *Journal of Orthopedic Research and Therapy*.

DANIEL, C. et R.-G. JAVIER. 2017b, «The prevalence of injuries in professional soccer players», *Journal of Orthopedic Research and Therapy*.

DRUMMOND, C. et R. HOLTE. 2003, «C4.5, class imbalance, and cost sensitivity : Why under-sampling beats oversampling», *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets*.

DUDA, R., P. HART et D. G.STORK. 2001, *Pattern Classification*.

EFRON, B. et R. . TIBSHIRANI. 1993, «An introduction to the bootstrap», *Chapman and Hall*.

EKSTRAND, J., T. TIMPKA et M. HÄGGLUND. 2006, «Risk of injury in elite football played on artificial turf versus natural grass : a prospective two-cohort study», *Br J Sports Med*.

FAN, W., S. STOLFO, J. ZHANG et P. CHAN. 1999, «Adacost : Misclassification cost-sensitive boosting», *Proceedings of the Sixteenth International Conference on Machine Learning (ICML'99)*.

FAUZAN, M. et H. MURFI. 2018, «The accuracy of xgboost for insurance claim prediction», *International Journal of Advances in Soft Computing and its Applications*, vol. 10, p. 159–171.

FERNÁNDEZ, A., S. GARCÍA, M. GALAR, R. PRATI, B. KRAWCZYK et F. HERRERA. 2018, «Learning from imbalanced data sets», .

FIX, E. et J. L. HODGES. 1989, «Discriminatory analysis. nonparametric discrimination : Consistency properties», *International Statistical Review / Revue Internationale de Statistique*, vol. 57, n° 3, p. 238–247.

FOSTER, C. G. 1998, «Monitoring training in athletes with reference to overtraining syndrome.», *Medicine and science in sports and exercise*, vol. 30 7, p. 1164–8.

FREUND, Y. et R. SCHAPIRE. 1997, «A decision-theoretic generalization of on-line learning and an application to boosting», *J. Comput. Syst. Sci.*, vol. 55, p. 119–139.

FREUND, Y. et R. E. SCHAPIRE. 1996, «Experiments with a new boosting algorithm», .

FRIEDMAN, J., T. HASTIE et R. TIBSHIRANI. 2000, «Additive logistic regression : a statistical view of boosting (with discussion and a rejoinder by the authors)», *Ann. Statist.*, vol. 28, n° 2, doi : 10.1214/aos/1016218223, p. 337–407. URL https://doi.org/10.1214/aos/1016218223.

FRIEDMAN, J. et B. POPESCU. 2003, «Importance sampled learning ensembles», .

FRIEDMAN, J. H. 2001, «Greedy function approximation : A gradient boosting machine», *The Annals of Statistics*, vol. 29, n° 5, p. 1189–1232.

GALAR, M., A. FERNÁNDEZ, E. BARRENECHEA, H. SOLA et F. HERRERA. 2012, «A review on ensembles for the class imbalance problem : Bagging-, boosting-, and hybrid-based approaches», *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, vol. 42, p. 463 – 484.

GARAVAGLIA, S., A. DUN et B. M. HILL. 1998, «A smart guide to dummy variables : Four applications and a macro», .

GEORGIOS, K., M. NIKOLAOS, P. RICARD et M. NICOLA. 2019, «Artificial intelligence. a tool for sports trauma prediction», *Br J Sports Med*.

GIVENS, G. H. et J. A. HOETING. 2012, «Computational statistics», *John Wiley and Sons*.

GREENE, W. 2008, «Econometric analysis», *Macmillan Publishing Company*, vol. 7.

HAN, H., W.-Y. WANG et B.-H. MAO. 2005, «Borderline-smote : A new over-sampling method in imbalanced data sets learning», p. 878–887.

HART, P. 1968, «The condensed nearest neighbour rule», *IEEE Transactions on Information Theory*, vol. 14, p. 515–516.

HASTIE, T., R. TIBSHIRANI, J. FRIEDMAN et J. FRANKLIN. 2004, «The elements of statistical learning : Data mining, inference, and prediction», *Math. Intell.*, vol. 27, p. 83–85.

HAYKIN, S. 1999, *Neural Networks : A Comprehensive Foundation*.

HE, H., Y. BAI, E. GARCIA et S. LI. 2008, «Adasyn : Adaptive synthetic sampling approach for imbalanced learning», *Proceedings of the International Joint Conference on Neural Networks*, p. 1322 – 1328.

HE, H. et E. GARCIA. 2009, «Learning from imbalanced data», *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, p. 1263 – 1284.

HE, H. et Y. MA. 2013, «Imbalanced learning : Foundations, algorithms, and applications», .

HILBE, J. M. 2009, *Logistic Regression Models*.

HILBORN, C. G. et D. LAINIOTIS. 1967, «The condensed nearest neighbor rule», .

HOEFFDING, W. 1948, «A class of statistics with asymptotically normal distribution.», *The annals of Mathematical Statistics*.

JAN, E., H. MARTIN et W. MARKUS. 2011a, «Epidemiology of muscle injuries in professional football (soccer)», *The American Journal of Sports Medicine*.

JAN, E., H. MARTIN et W. MARKUS. 2011b, «Epidemiology of muscle injuries in professional football (soccer)», *The American Journal of Sports Medicine*.

JOHANN, W. et G. TIM J. 2016, «How do training and competition workloads relate to injury? the workload—injury aetiology model», *Br J Sports Med*.

JOÃO GUSTAVO, C., D. O. C. DANIEL, V. D. S. THIAGO, S. JULIO CERCA, M. P. ADRIANO C. et N. GEORGE P. 2019, «Current approaches to the use of artificial intelligence for injury risk assessment and performance prediction in team sports : a systematic review», *Sports Medicine - Open*.

KARAKOULAS, G. I. et J. SHAWE-TAYLOR. 1998, «Optimizing classifers for imbalanced training sets», dans *NIPS*.

KASS, G. V. 1980, «An exploratory technique for investigating large quantities of categorical data», *Journal of Applied Statistics*, vol. 2, p. pp. 119–127.

KING, G. et L. ZENG. 2002, «Logistic regression in rare events data», *Political Analysis*, vol. 9, doi : 10.1093/oxfordjournals.pan.a004868.

KÖKNAR-TEZEL, S. et L. J. LATECKI. 2009, «Improving svm classification on imbalanced data sets in distance spaces», , p. 259–267.

KOHONEN, T. 2001, *Self-Organizing Maps*.

KONSTANTINOS, F., T. ELIAS et A. SPYROS. 2010, «Intrinsic risk factors of non-contact quadriceps and hamstring strains in soccer : A prospective study of 100 professional players», *Br J Sports Med*.

KUBAT, M. 2000, «Addressing the curse of imbalanced training sets : One-sided selection», *Fourteenth International Conference on Machine Learning*.

KUHN, H. W. et A. W. TUCKER. 1951, «Nonlinear programming», , p. 481–492.

LANCASTER, T. et G. IMBENS. 1996, «Case-control studies with contaminated controls», *Journal of Econometrics*, vol. 71, n° 1, p. 145 – 160.

LAURIKKALA, J. 2001, «Improving identification of difficult small classes by balancing class distribution», p. 63–66, doi :10.1007/3-540-48229-6_9.

LEVERSHA, G. 2003, «Statistical inference (2nd edn)», *The Mathematical Gazette*, vol. 87, n° 509, p. 401–403.

LIU, T.-Y. 2009, «Easyensemble and feature selection for imbalance data sets», p. 517–520.

LIU, X.-Y., J. WU et Z.-H. ZHOU. 2009a, «Exploratory undersampling for class-imbalance learning», *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 39, p. 539 – 550.

LIU, X.-Y., J. WU et Z.-H. ZHOU. 2009b, «Exploratory undersampling for class-imbalance learning», *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 39, p. 539 – 550.

LÓPEZ, V., A. FERNÁNDEZ et F. HERRERA. 2014, «On the importance of the validation technique for classification with imbalanced datasets : Addressing covariate shift when data is skewed», *Information Sciences*, vol. 257, p. 1 – 13.

M., H., W. M. et E. J. 2005, «Injury incidence and distribution in elite football–a prospective study of the danish and the swedish top divisions», *Scan J Med Sci Sports*.

MAALOUF, M. 2011, «Logistic regression in data analysis : An overview», *International Journal of Data Analysis Techniques and Strategies*, vol. 3, doi :10.1504/IJDATS.2011.041335, p. 281–299.

MAALOUF, M., D. HOMOUZ et T. TRAFALIS. 2017, «Logistic regression in large rare events and imbalanced data : A performance comparison of prior correction and weighting methods : Maalouf et al .», *Computational Intelligence*, vol. 34.

MAALOUF, M. et M. SIDDIQI. 2014, «Weighted logistic regression for large-scale imbalanced and rare events data», *Knowledge-Based Systems*, vol. 59.

MARTIN, H., W. MARKUS, M. HENRIK, K. KAROLINA, B. HÅKAN et E. JAN. 2013, «Injuries affect team performance negatively in professional football : an 11-year follow-up of the uefa champions league injury study», *Br J Sports Med*.

MARTIN, H., W. MARKUS et E. JAN. 2016, «Previous injury as a risk factor for injury in elite football : A prospective study over two consecutive seasons», *Br J Sports Med*.

MCCULLAGH, P. et J. NELDER. 1989, *Generalized Linear Model*.

MCKAY, M. 1995, «Evaluating prediciton uncertainty», *US Nuclear Regulatory Commission and Los Alamos National Laboratory*.

MIHELICH, M., C. DOGNIN, Y. SHU et M. BLOT. 2019, «A characterization of mean squared error for estimator with bagging», .

MORRIS, M. D. 1991, «Factorial sampling plans for preliminary computational experiments», *Technometrics*.

NUÑEZ, H., L. GONZALEZ-ABRIL et C. ANGULO. 2017, «Improving svm classification on imbalanced datasets by introducing a new bias», *Journal of Classification*, doi :10.1007/s00357-017-9242-x.

PEIRCE, C. S. «The numerical measure of the success of predictions.», *Science*, vol. 4 93, p. 453–4.

PLATT, J. C. 1999, «Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods», , p. 61–74.

POISSON, S. D. 1887, *Recherches sur la probabilité des jugements en matière criminelle et en matière civile*, Elibron Classics.

POLISSAR, L. et P. DIEHR. 1982, «Regression analysis in health services research : The use of dummy variables», *Medical Care*, vol. 20, nº 9, p. 959–966.

R., B. et K. T. 2005, «Understanding injury mechanisms : a key component of preventing injuries in sport», *Br J Sports Med*.

RAMIREZ, F. et H. ALLENDE. 2012, «Dual support vector domain description for imbalanced classification», , p. 710–717.

REFAEILZADEH, P., L. TANG et H. LIU. 2009, *Cross-Validation*, Springer US, Boston, MA, ISBN 978-0-387-39940-9, p. 532–538.

SAITO, T. et M. REHMSMEIER. 2015, «The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets», *PloS one*, vol. 10, p. e0118 432.

SCHAPIRE, R. et Y. SINGER. 1998, «Boostexter : A system for multiclass multi-label text categorization», .

SEIFFERT, C., T. KHOSHGOFTAAR, J. VAN HULSE et A. NAPOLITANO. 2008, «Building useful models from imbalanced data with sampling and boosting.», p. 306–311.

SHAWE-TAYLOR, J. et N. CRISTIANINI. 2004, *Kernel Methods for Pattern Analysis*, Cambridge University Press.

SOBOL, I. M. 1993, «Sensitivity estimates for nonlinear mathematical models.», *Mathematical Modelling and Computational Experiments*.

SOBOL, I. M. 2001, «Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates», *Mathematics and Computers in Simulation*.

STONE, M. 1974, «Cross-validatory choice and assessment of statistical predictions», *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 36, nº 2, p. 111–133.

TIM J, G. 2016, «The training-injury prevention paradox : should athletes be training smarter and harder?», *Br J Sports Med*.

TOMEK, I. 1976, «Two modifications of cnn», *EEE Transactions on Systems Man and Communications*, p. 769–772.

VALVERDE-ALBACETE, F. J. et C. PELÁEZ-MORENO. 2014, «100normalized information transfer factor explains the accuracy paradox», *PLoS ONE*, vol. 9.

VAN HULSE, J. et T. KHOSHGOFTAAR. 2009, «Knowledge discovery from imbalanced and noisy data», *Data and Knowledge Engineering*, vol. 68, nº 12, p. 1513 – 1542.

VAPNIK, V. et A. LERNER. 1963, «Pattern recognition using generalized portrait method», *Automation and Remote Control*, vol. 24, p. 774–780.

VEROPOULOS, K., C. CAMPBELL et N. CRISTIANINI. 1999, «Controlling the sensitivity of support vector machines», *Proceedings of International Joint Conference Artificial Intelligence*.

VISA, S. et A. RALESCU. 2005, «Issues in mining imbalanced data sets - a review paper», *Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conference*.

WILLIAM P., Z. 1989, «Weakly differentiable functions», .

WILSON, D. 1972, «Asymptotic properties of nearest neighbor rules using edited data», *IEEE Trans. Syst. Man Cybern.*, vol. 2, p. 408–421.

WILSON, D. et T. MARTINEZ. 2000, «Reduction techniques for instance-based learning algorithms», *Machine Learning*, vol. 38, p. 257–286.

WU, J., J. REHG et M. MULLIN. 2003, «Learning a rare event detection cascade by direct feature selection», .

YANG, T., L. CAO et C. ZHANG. 2010, «A novel prototype reduction method for the k-nearest neighbor algorithm with k>1», p. 89–100.

YPMA, T. J. 1995, «Historical development of the newton-raphson method», *SIAM review*, vol. 21, p. 531–551.

ZHOU, X. et H. LIN. 2008, «Local sensitivity analysis», *EDDS. springer US, Boston*.

ZHU, B., B. BAESENS, A. BACKIEL et S. VANDEN BROUCKE. 2017, «Benchmarking sampling techniques for imbalance learning in churn prediction», *Journal of the Operational Research Society*, vol. 69, p. 1–17.

ZIANG, J. 2003, «Knn approach to unbalanced data distributions : a case study involving information extraction», *Proc. Int'l. Conf. Machine Learning1 (ICML'03), Workshop Learning from Imbalanced Data Sets*.