Inspection d'une boîte noire via une analyse de robustesse

Arthur Maillart

Inspection d'une boîte noire via une analyse de robustesse

Arthur MAILLART*[1]

[1]Université Lyon 1, Laboratoire SAF, Lyon, France

Abstract

The rise of Machine Learning models has led insurers to create DataLabs in order to build more efficient models than the existing ones. However, for the most critical tasks, some models considered too complex and secretive are struggling to go into production. Indeed, for actuaries responsible of risk assessment, it is difficult to give the same level of confidence to these nebulous models as to more familiar models such as generalized linear models. There is therefore a real need for actuaries to reduce the gap between what is understood from the model and what the model has learnt. However, providing a general explanation of a black box predictor, i.e. an explanation of the general decision-making mechanism of the predictor, is one of the most difficult tasks. Therefore, a significant part of the efforts to increase the intelligibility of models focuses on more affordable tasks, such as providing a local explanation, or visual or textual information on the model's reactions. This article proposes to adapt the robustness approach of [Koh and Liang, 2017], to reconcile the global and local aspect of intelligibility.

Résumé

L'essor des modèles de Machine Learning a poussé les assureurs à se doter de DataLabs dans le but de créer des modèles plus performants que ceux déjà existants. Cependant, pour les tâches les plus critiques, certains modèles considérés trop complexes et opaques peinent à passer en production. En effet, pour les actuaires, chargés d'évaluer le risque, il est difficile d'accorder le même niveau de confiance à ces modèles considérés comme des boîtes noires qu'à des modèles plus familiers comme les modèles linéaires généralisés. Il y a donc un réel besoin pour l'actuaire de réduire l'écart entre ce qu'il comprend du modèle et ce que le modèle a appris. Cependant, fournir une explication globale d'un prédicteur boîte noire, c'est à dire une explication de son mécanisme décisionnel général, est une des tâches les plus difficiles. C'est pourquoi une part importante des efforts pour accroître l'intelligibilité des modèles se concentre sur des tâches plus abordables, à savoir fournir une explication locale ou des informations visuelles ou textuelles sur les réactions du modèle. Cet article propose d'adapter l'approche par robustesse de [Koh and Liang, 2017], pour collecter des informations globales pertinentes et fournir une explication fidèle au voisinage d'un point dont on veut expliquer la prédiction.

Keywords: Machine Learning, Black-Box Inspection, Interpretability, Explainability, Car insurance

Mots-clefs: Apprentissage statistique, Inspection d'une boîte noire, Intelligibilité, Explicabilité, Assurance automobile

*arthur.maillart@etu.univ-lyon1.fr

Introduction

Parmi les méthodes de Machine Learning, les réseaux de neurones font partie des plus complexes. Pour une prédiction en un point donné il n'est souvent pas possible de comprendre pourquoi le modèle prend une décision plutôt qu'une autre. C'est ce constat qui motive le développement de méthodes permettant de rendre plus intelligibles les modèles sophistiqués de Machine Learning. Mais qu'entend-on par intelligible ? Selon [Guidotti et al., 2018] et [Doshi-Velez and Kim, 2017], pour les modèles de Machine Learning, rendre intelligible signifie expliquer en des termes compréhensibles par l'humain. Cependant, expliquer a un sens différent en fonction de l'utilisation que l'on a de ces méthodes. Si l'on souhaite améliorer ou déboguer son modèle, des informations sur l'importance que la boîte noire accorde aux variables peuvent être satisfaisantes. En revanche, si l'on a besoin d'une compréhension plus fine du modèle pour être conforme à la réglementation RGPD, de simples informations sur les variables ne peuvent suffire puisqu'il faut être capable d'expliquer en détail à chaque assuré son tarif par exemple. Évidemment, plus l'on désire que l'explication soit précise, plus le problème est compliqué. Pour une vue d'ensemble sur les méthodes liées à l'intelligibilité des modèles de Machine Learning se référer à [Guidotti et al., 2018].

Dans cet article, nous nous intéressons à une méthode qui permet de mettre en évidence des points influents sur une méthode de Machine Learning paramétrique. Nous appelons méthode paramétrique toute méthode synthétisant des données avec un ensemble de paramètres fixes. Il peut s'agir des réseaux de neurones, des modèles linéaires généralisés ou encore des machines à vecteur de support (SVM). Par opposition, les méthodes non paramétriques permettent de ne pas faire d'hypothèse sur la structure du modèle. Parmi ces dernières, on trouve les arbres de décisions et dérivés tels que les Random Forest et Gradient Boosting. Pour estimer l'influence des points de l'échantillon d'apprentissage, une approche naturelle consiste à entraîner un modèle de référence sur l'ensemble de l'échantillon d'apprentissage puis à mesurer l'écart sur des quantités d'intérêt. Par exemple, l'écart sur les paramètres estimés avant et après retrait d'un point, ou la différence des valeurs de fonction de perte avant et après retrait d'un point. Grâce à ces indicateurs, il est possible de savoir si un point contribue à améliorer ou détériorer les résultats d'un modèle boîte noire. Cependant, cette méthode peut s'avérer très coûteuse en calculs pour des jeux de données volumineux. Nous présentons ici trois indicateurs basés sur les fonctions d'influence qui permettent d'approximer : l'effet du retrait d'un point sur les paramètres estimés, l'effet du retrait ou de la perturbation d'un point de l'échantillon d'apprentissage sur la prédiction d'un point quelconque. Si l'on se réfère à la classification de [Guidotti et al., 2018], la méthode que nous présentons ici se place dans le cadre du "Black-Box inspection problem". En d'autres termes, nous cherchons des éléments visuels ou textuels permettant de nous renseigner sur ce que le modèle a appris. Des exemples connus de méthodes qui appartiennent à cette classe sont les Partial Dependance Plots introduits par [Friedman, 2001] et les Variable Importance Plots introduits par [Breiman, 2001]. Cette approche n'est pas nouvelle puisque très rapidement pour les modèles linéaires généralisés les statisticiens s'y sont intéressés comme Weisberg et Cook [Weisberg, 1982]. Depuis, elle a été approfondie entre autres dans [Wojnowicz et al., 2016] et [Koh and Liang, 2017] qui généralisent les résultats et donnent un cadre mathématique harmonisé pour les modèles paramétriques.

S'il est vrai que déterminer l'influence des points du modèle ne fournit pas une vision globale de la chaîne de décision d'une boîte noire, il est en revanche possible de collecter des informations pertinentes pour déboguer un modèle ou obtenir des informations locales. En effet, en identifiant l'influence des observations de l'échantillon d'apprentissage, il est possible d'identifier les points considérés comme anormaux par le modèle. Ainsi, il est envisageable d'assainir ses données en

traitant les individus aberrants mis en évidence par les indicateurs que nous allons présenter. Par ailleurs, en combinant les indicateurs appropriés avec des modèles naturellement interprétables tels que les modèles linéaires généralisés ou encore les arbres de décision, nous pouvons en extraire des explications sous la forme d'hyperplans (modèles linéaires). L'idée est proche des méthodes LIME [Ribeiro et al., 2016] et SHAP [Lundberg and Lee, 2017] qui consistent à trouver un modèle de substitution[1] au voisinage du point à expliquer. Dans le cas particulièrement complexe des réseaux de neurones, ces informations ne sont pas faciles à obtenir et sont souvent très coûteuses du point de vue informatique. Les méthodes présentées ici permettent de considérablement réduire les temps de calcul en approximant l'influence des points sur le modèle. L'intérêt principal est de localiser les zones de l'espace où il est pertinent d'avoir un modèle de substitution. La deuxième force de cette méthodologie est de restreindre le nombre de points à expliquer.

Nous illustrerons systématiquement les indicateurs sur un jeu de données jouet en deux dimensions avant de les appliquer à un jeu réel de données d'assurance. Par ailleurs, nous utiliserons deux types de modèles paramétriques pour mettre en évidence nos conclusions, une régression logistique et un réseau de neurones. La première permettra de comprendre le fonctionnement des indicateurs avec un modèle qui fait partie de la zone de confort de l'actuaire tandis que le second démontrera l'intérêt concret de la méthode. Dans une première partie, nous présenterons l'indicateur $\mathcal{I}_{up,params}$ qui permet de déterminer l'effet de la suppression d'un point de l'échantillon d'apprentissage sur les paramètres du modèle. Il peut s'avérer utile dans le cas des modèles linéaires généralisés pour détecter des valeurs aberrantes. Ensuite, nous déduirons de ce premier l'indicateur $\mathcal{I}_{up,loss}$ qui caractérise l'impact du retrait d'un point de l'échantillon d'apprentissage sur la prédiction en un point quelconque. Grâce à cet indicateur, nous développerons des explications fidèles localement au modèle boîte noire. Ces dernières prendront la forme d'hyperplans. Par ailleurs, nous montrerons que cette technique permet de réduire drastiquement le nombre de points à analyser. Enfin, nous définirons $\mathcal{I}_{pert,loss}$, qui donne la direction dans laquelle un point doit être perturbé pour maximiser l'impact sur le modèle boîte noire. Nous établirons que cet indicateur peut en lui-même servir d'explication pour une prédiction donnée.

Remarque. *Les preuves figurent en annexe pour fluidifier la lecture.*

Préliminaire sur les fonctions d'influence

Dans son livre, Hampel [Hampel et al., 2005] définit de manière intuitive les fonctions d'influence comme l'effet approché et standardisé, sur une statistique T, de l'ajout d'une observation supplémentaire z étant donné un large échantillon de fonction de répartition F. La fonction d'influence d'une statistique T, en une fonction de répartition sous-jacente F dans la direction z est notée $IF(z, T, F)$. Elle correspond à une dérivée directionnelle en la distribution F dans la direction z.

La notion de fonction d'influence repose sur le concept de distribution contaminée, que nous définissons de la manière suivante.

Définition 0.1. *Soit F une fonction de répartition. On définit la distribution ϵ-contaminée au point z par :*

$$F_{\epsilon,z} = (1 - \epsilon)F + \epsilon\Delta_z$$

[1] Surrogate

où Δ_z est la mesure de Dirac au point z.

Remarque. *Lorsque $\epsilon > 0$ cela peut s'interpréter comme une loi mélange.*

Définition 0.2. *Soit T une fonctionnelle, $T : \begin{cases} \mathcal{F} & \longrightarrow \mathbb{R} \\ F & \longmapsto T(F) \end{cases}$, où $\mathcal{F}$ est l'ensemble des fonctions de répartition. On définit la fonction d'influence par :*

$$IF(z; T; F) = \lim_{\varepsilon \to 0} \frac{T(F_{\epsilon,z}) - T(F)}{\epsilon}.$$

Dans toute la suite de l'article, nous nous placerons dans le cadre d'un problème de classification. Notons $\mathcal{X} = \mathbb{R}^p$ l'espace des variables explicatives et $\mathcal{Y}$ l'espace des labels. On se donne de plus un jeu de données $D = \{z_i = (x_i, y_i) \text{ tels que } x_i \in \mathcal{X} \text{ et } y_i \in \mathcal{Y}\}$ contenant n observations $z_1, \ldots, z_n$. La fonction de répartition empirique du vecteur aléatoire $(X_1, \ldots, X_p)$ est définie par

$$F_n : \begin{cases} \mathbb{R}^p & \longrightarrow \mathbb{R} \\ (x_1, \ldots, x_p) & \longmapsto \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{X_{1,i} \leq x_1, \ldots, X_{p,i} \leq x_p\}}. \end{cases}$$

Intuitivement, prendre la fonction de répartition empirique à la place de F ($F = F_{n-1}$) et $\epsilon = 1/n$ permet de voir que la fonction d'influence mesure n fois la variation subie par T lors de l'ajout d'un point z pour un échantillon assez grand. Dans notre cas, F sera une fonction de répartition multi-dimensionnelle définie comme $F : \mathbb{R}^p \to \mathbb{R}$ et l'on considérera plutôt le retrait d'un point de l'échantillon d'apprentissage. Ce qui revient à prendre $F = F_n$ et $\epsilon = -1/n$

Exemple simple sur l'espérance

Soit Z la variable observée. On considère une perturbation au point $z \in \mathbb{R}$. Alors en notant $\mathbb{E}_{F_{\epsilon,z}}(Z)$ l'espérance de la distribution contaminée on obtient :

$$\mathbb{E}_{F_{\epsilon,z}}(Z) = \int s \; d\left[(1-\epsilon)F + \epsilon \Delta_z\right](s),$$

donc

$$\mathbb{E}_{F_{\epsilon,z}}(Z) = (1-\epsilon) \int s dF(s) + \epsilon z.$$

A partir de cette expression, on peut facilement déterminer l'expression de la fonction d'influence.

$$\boxed{IF(z; E; F) = z - \mathbb{E}(Z)}$$

Remarque. *La fonction d'influence de la moyenne dépend de la distance du point z à la moyenne. Ainsi, plus un point est éloigné en valeur absolue, plus son influence sur la moyenne sera grande. Ou encore, si la contamination intervient au point z et si ce dernier est grand alors la contamination aura un impact significatif sur la moyenne. En revanche, si z est proche de la moyenne alors la contamination aura un impact assez faible. C'est pour cela que la moyenne est considérée comme peu robuste par rapport à la médiane par exemple.*

1 Positionnement du problème théorique et définitions

Les méthodes de Machine Learning telles que les RandomForest [Breiman, 2001], XGBoost [Chen and Guestrin, 2016], et de Deep Learning telles que les réseaux de neurones se sont montrées très performantes dans certaines tâches de prédiction. Suite à cela, le paradigme de l'apprentissage statistique a évolué. Désormais, on cherche à satisfaire deux concepts a priori contradictoires : prédire et comprendre. De ce fait, de nombreuses recherches ont été entreprises ces dernières années pour pallier le manque d'intelligibilité des modèles actuels. D'un point de vue opérationnel, l'intelligibilité peut être déclinée selon quatre approches différentes d'après [Guidotti et al., 2018]. Ces dernières sont synthétisées Figure 1. En fonction de l'usage des méthodes de Machine Learning et du niveau de détail souhaité, certaines méthodes d'interprétation sont plus pertinentes que d'autres. En tant qu'actuaire, il est toujours préférable d'avoir le niveau d'information maximal, c'est-à-dire d'avoir un modèle intelligible par nature ou un modèle surrogate qui reproduit globalement et fidèlement les prédictions du modèle boîte noire. Cependant, ce n'est pas toujours aisé. Dans le cas des réseaux de neurones, il est très difficile pour un humain de comprendre la chaîne de décision globale. Cette dernière faisant intervenir un grand nombre de poids. Il peut donc être intéressant d'affaiblir le problème et de chercher soit de l'information valable uniquement localement, c'est-à-dire répondre au "Black-Box Outcome Explanation Problem", soit obtenir des informations pertinentes pour la compréhension du modèle, mais pas suffisantes pour expliquer globalement les décisions du modèle c'est-à-dire répondre au "Black-Box Inspection Problem".

1.1 Éléments de contexte

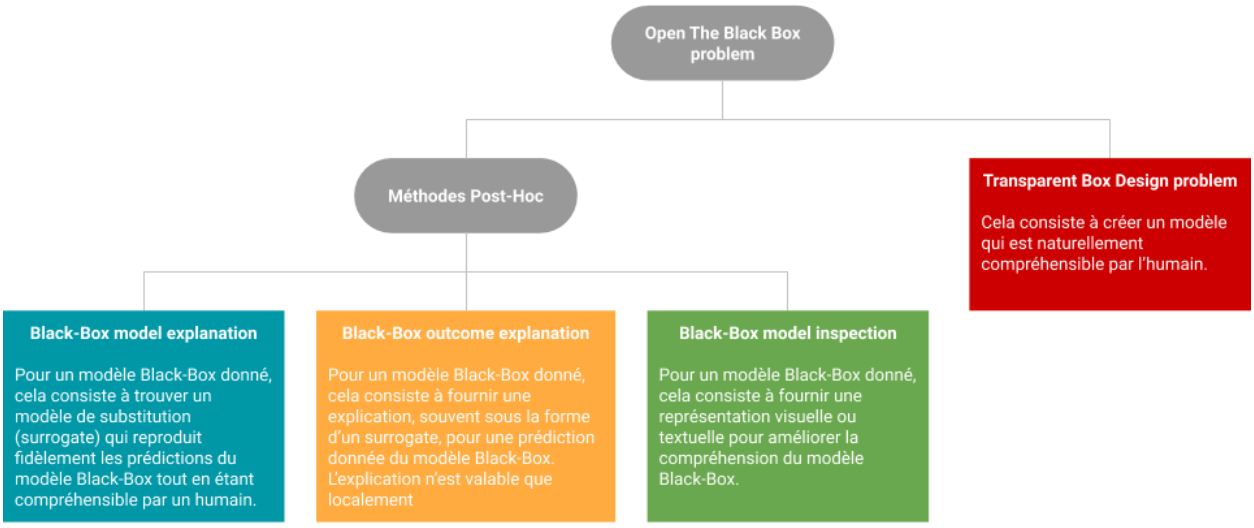


Figure 1 – Les différentes approches du problème de l'intelligibilité

Nous définissons ici formellement le problème de classification. Soit $b : \mathcal{X} \longrightarrow \mathcal{Y}$ un prédicteur boîte noire, que nous pourrons aussi nommer classifieur. Ce dernier s'obtient par une fonction d'apprentissage

$$\mathbb{L}_b : \begin{cases} \mathcal{Z} \longrightarrow (\mathcal{X} \longrightarrow \mathcal{Y}) \\ D \longrightarrow b \end{cases},$$

où $\mathcal{Z}$ représente un ensemble de jeux de données. Ainsi, pour chaque $x \in \mathcal{X}$ le prédicteur b peut fournir une probabilité (ou un vecteur de probabilités) d'appartenance à une classe notée $b(x)$. Dans le contexte de l'apprentissage supervisé dans lequel nous nous plaçons, le jeu de données D est subdivisé en deux échantillons : D_{train} et D_{test}. Le premier sert à construire le classifieur et le second à en évaluer les performances.

Dans cet article, nous présentons des indicateurs qui répondent en premier lieu au "Black-Box Inspection Problem". Par définition, pour une boîte noire b et un ensemble d'instances X, cela consiste à fournir une représentation visuelle ou textuelle notée r. Cette représentation s'obtient à partir de b et X grâce à une procédure $f : (b, X) \longrightarrow r$. Dans notre cas à partir de notre boîte noire et des instances, nous pourrons extraire des représentations r sous la forme de boxplots caractérisants l'influence des points.
Il est aussi possible de se servir des indicateurs et des propriétés des modèles paramétriques étudiés pour répondre au "Black-Box Outcome Explanation Problem". Par définition, ce problème consiste à trouver une explication e qui appartient à $\mathcal{E}$, l'ensemble des explications compréhensibles par l'humain. Cette explication repose sur un modèle intelligible localement c_l qui se déduit de b et de x en utilisant une procédure $f : (b, x) \longrightarrow c_l$. L'explication e s'obtient grâce à c_l et x. Dans notre cas, c_l est un modèle linéaire, et l'explication e est la chaîne de décision (combinaison linéaire des poids) pour l'instance x. L'avantage par rapport au "Black-Box Inspection Problem" est que nous pouvons mesurer la qualité des explications fournies. Cette mesure appelée fidelité dans le domaine du Machine Learning intelligible quantifie à l'aide de métriques usuelles[2] à quel point le modèle de substitution réplique les prédictions du modèle boîte noire au voisinage du point à expliquer.

1.2 Définitions

Nous nous placerons dans le cas de données tabulaires car elles constituent la majorité des données exploitées par les assureurs aujourd'hui. Cependant, il est aussi possible d'utiliser cette méthode avec des images ou du texte. Le lecteur intéressé pourra se référer à [Koh and Liang, 2017].

Les modèles de Machine Learning auxquels s'appliquent les méthodes présentées dans cet article sont paramétriques. Nous modélisons la loi conditionnelle de Y sachant X. Pour un modèle paramétrique donné, nous noterons l'ensemble des paramètres de ce dernier Θ. Soit $z \in D_{train}$ et $\theta \in \Theta$, la fonction de perte au point z pour les paramètres du modèle θ est notée $\mathcal{L}(z, \theta)$. Enfin, notons $\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(z_i, \theta)$ le risque empirique et définissons le minimiseur du risque empirique par :

$$\hat{\theta}_n = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(z_i, \theta). \tag{1}$$

[2] *accurracy, AUC, precision, etc.*

Dans cette étude nous nous limiterons aux cas où $\mathcal{L}$ et $\nabla_\theta \mathcal{L}$ (où ∇_θ est l'opérateur gradient par rapport à θ) sont respectivement trois fois continûment différentiables en θ et deux fois continûment différentiables en x. Nous notons θ_0 le vrai vecteur de paramètres. Celui vers lequel $\hat{\theta}_n$ converge. On suppose de plus que $\mathbb{E}(\nabla_\theta(Z, \theta_0)) = 0$, $\mathbb{E}(||\nabla^2_\theta(Z, \theta_0)||) < \infty$ et que la matrice $\mathbb{E}(\nabla^2_\theta(Z, \theta_0))$ existe et n'est pas singulière. Par ailleurs, on suppose que $\mathbb{E}(\nabla^3_\theta(Z, \theta_0))$ est borné en probabilités lorsque n tend vers l'infini. Lorsque les fonctions de perte ne vérifient pas ces conditions, il est possible d'adapter la méthode pour qu'elle reste valable comme le montrent [Koh and Liang, 2017].

2 Méthodologie d'analyse

Nous souhaitons dans cet article mettre en œuvre les indicateurs pour en vérifier empiriquement la qualité puis étudier leur comportement sur différents jeux de données. Dans un premier temps, nous nous familiarisons avec les indicateurs et leurs limites sur un jeu de données jouet en deux dimensions. Nous employons les indicateurs sur une régression logistique (zone de confort de l'actuaire) puis sur un réseau de neurones. Dans un second temps, nous appliquons les mêmes indicateurs aux mêmes modèles sur un jeu de données d'assurance.

2.1 Les données simulées

Pour tester les indicateurs nous utiliserons un jeu de données simulées en deux dimensions. La Figure 2 le représente. Nous détaillons le procédé pour l'obtenir ci-dessous.

Après avoir tiré un échantillon de 2000 points uniformément sur le carré $[0, 1] \times [0, 1]$.

$$(x_1, x_2) \sim \mathcal{U}([0, 1])$$

Nous définissons un seuil théorique r en fonction de x_1. Ce dernier représente notre vraie fonction de décision.

$$r(x_1) = 0.25 + \frac{0.5}{1 + \exp\left(-20(x_1 - 0.5)\right)} + 0.05 \cos\left(2\pi x_1\right)$$

Nous rajoutons du bruit au voisinage de la frontière.

$$\begin{cases} y = 1 \text{ si } x_2 > r(x_1) + \epsilon \\ y = 0 \text{ si } x_2 < r(x_1) - \epsilon \\ \mathbb{P}(y = 1) = \frac{1}{2} \text{ si } |r(x_1) - x_2| < \epsilon \text{ où } \epsilon = 0.1 \end{cases}$$

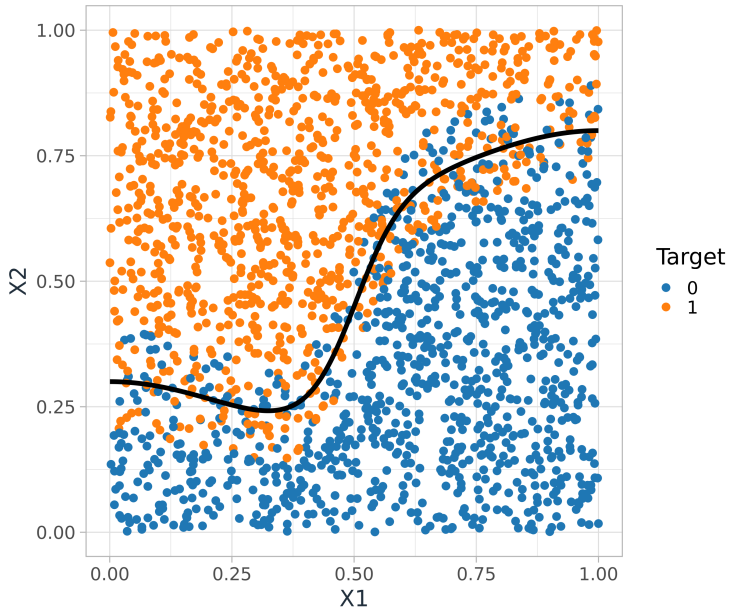


Figure 2 – Jeu de données jouet : le trait noir plein représente la frontière de décision théorique.

2.2 Le problème d'assurance

Cet exemple jouet permet de s'approprier les indicateurs et de développer une intuition dessus. Pour illustrer l'intérêt des indicateurs dans un cadre plus réaliste, nous utiliserons des données d'assureurs ayant déjà servies pour un challenge de DataScience : Le Pricing Game[3] (troisième édition). Les données[4] [5] sont initialement composées de 27 variables exploitables pour ajuster un modèle. Nous n'en utiliserons que 8, toutes numériques pour pouvoir illustrer tous les indicateurs. En effet, $\mathcal{I}_{pert,loss}$ requiert des variables numériques et continues. L'échantillon contient 100 000 observations. Nous centrons et réduisons les données avant de les découper en un échantillon d'apprentissage de validation dans les proportions suivantes (75%/25%). Nous dénombrons 9 490 sinistres dans la base d'apprentissage et 3 164 présents dans la base de validation. Les sinistres sont équitablement répartis entre les deux échantillons. Nous transformons le problème initial de pricing (modèle coût fréquence) en un problème de classification pour être cohérent avec la théorie développée dans l'article. Pour cela, nous choisissons de créer une nouvelle variable cible à partir du nombre d'accidents. Si la police d'assurance a eu au moins un sinistre alors la nouvelle variable cible contiendra un, et zéro sinon. De cette manière, nous créons un problème de prévention. L'idée étant de créer un modèle qui recommande une action de prévention envers les automobilistes en portefeuille.

2.3 Les modèles

Pour commencer nous implémentons les indicateurs pour une régression logistique binaire. Puisque nous avons dans ce cas des formules fermées simples pour le gradient et la hessienne, nous pouvons

[3] http://freakonometrics.free.fr/PG3/3rdPricingGame.pdf
[4] http://freakonometrics.free.fr/PG3/PG_2017_YEAR0.csv contenant les variables explicatives
[5] http://freakonometrics.free.fr/PG3/PG_2017 CLAIMS_YEAR0.csv contenant les sinistres

vérifier empiriquement la précision de l'approximation. Puis, nous étendons à un réseau de neurones pénalisé.

2.3.1 Régression logistique

Formellement, pour la régression logistique binaire, nous cherchons à modéliser la probabilité d'appartenance à la classe 1 par

$$p(x_i;\beta) = \mathbb{P}(Y=1|X=x_i) = \frac{1}{1+e^{-(\beta_0+\beta^\intercal x_i)}},$$

où $\beta = (\beta_0, \ldots, \beta_p)$. Les coefficients β_i du modèle sont ajustés par maximum de vraisemblance, c'est à dire en optimisant la fonction de perte

$$\mathcal{L}(z_i,\beta) = -\left[y_i \log\left(p(x_i;\beta)\right) + (1-y_i)\log\left(1-p(x_i;\beta)\right)\right],$$

sur toutes les observations z_i. Généralement, c'est un algorithme de type Newton-Raphson qui est utilisé pour optimiser les paramètres.

Remarque. *Pour les exemples développés dans cet article, nous n'avons pas pénalisé la régression logistique. Cependant, il est possible d'introduire un terme de pénalisation de type L_2. La fonction de perte au point z_i deviendrait alors :*

$$\mathcal{L}(z_i,\beta) = -\left[y_i \log\left(p(x_i;\beta)\right) + (1-y_i)\log\left(1-p(x_i;\beta)\right)\right] + \frac{\alpha}{2}||\beta||_2^2.$$

Nous avons pénalisé le réseau de neurones décrit ci-dessous avec $\alpha = 0.001$.

2.3.2 Réseau de neurones

Nous utilisons un réseau de neurones simple mais assez souple pour approcher correctement nos problèmes jouet et d'assurance. L'objectif de l'article n'étant pas la performance du modèle mais l'explicabilité, nous n'avons pas cherché à améliorer les performances de nos boîtes noires. Pour des hyperparamètres $q_{m-1} \in \mathbb{N}$ et $q_m \in \mathbb{N}$ nous définissons la couche m de notre réseau de neurones comme suit :

$$\boldsymbol{Z}^{(m)} = \begin{cases} \mathbb{R}^{q_{m-1}} & \longrightarrow \mathbb{R}^{q_{m-1}} \\ \boldsymbol{Z} & \longrightarrow \left(Z_1^{(m)}(\boldsymbol{Z}), \ldots, Z_{q_m}^{(m)}(\boldsymbol{Z})\right)' \end{cases}$$

où le neurone j de la couche m s'écrit comme le produit scalaire suivant :

$$Z_j^{(m)} = Z_j^{(m)}(\boldsymbol{Z}) = \phi\left(\beta_{j,0}^{(m)} + \sum_{l=1}^{q_{m-1}} \beta_{j,l}^{(m)} Z_l\right) = \phi\left(<\boldsymbol{\beta}_j^{(m)}, \boldsymbol{Z}>\right)$$

où les $\boldsymbol{\beta}_j^{(m)} = (\beta_{j,l}^{(m)})_{0<l<q_{m-1}} \in \mathbb{R}^{q_{m-1}+1}$ sont les poids de la couche m du réseau de neurones. Notons M le nombre de couches totales du réseau de neurones (y compris couches d'entrées et sorties). Alors chacune des m couches avec $1 \leq m < M$, de notre réseau s'écrit :

$$\boldsymbol{Z}^{(m:1)}(x_i) = \left(\boldsymbol{Z}^{(m)} \circ \cdots \circ \boldsymbol{Z}^{(1)}\right)(x_i)$$

Comme il est d'usage, nous utilisons la fonction softmax notée σ :

$$\sigma = \begin{cases} \mathbb{R}^K & \longrightarrow \mathbb{R}^K \\ \boldsymbol{Z} & \longrightarrow \left(\frac{e^{Z_1}}{\sum_{k=1}^K e^{Z_k}}, \ldots, \frac{e^{Z_K}}{\sum_{k=1}^K e^{Z_k}} \right) \end{cases}$$

pour que la couche de sortie M retourne un vecteur de probabilités. Finalement

$$p(x_i; \beta) = \boldsymbol{Z}^{(M:1)}(x_i) = \sigma\left(\boldsymbol{Z}^{(M-1:1)}(x_i)\right)$$

Nous avons choisi arbitrairement 2 couches cachées pour nos deux réseaux de neurones[6]. Dans le cas de l'exemple jouet nous choisissons les hyperparamètres suivants : $q_1 = 2$, $q_2 = 4$, $q_3 = 2$, $q_4 = 2$ et $\phi = \tanh$. La Figure 3 représente l'architecture de ce réseau.

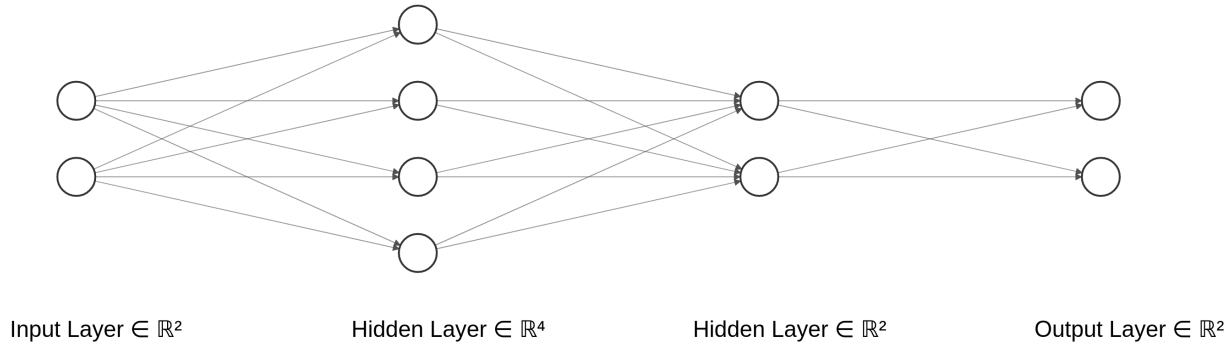


Figure 3 – Architecture du réseau de neurones pour l'exemple jouet

Dans le cas de l'exemple d'assurance nous choisissons les hyperparamètres suivants : $q_1 = 8$, $q_2 = 4$, $q_3 = 2$, $q_4 = 2$ et $\phi = \tanh$. Nous utilisons dans ce cas précis la régularisation L_2 définie plus haut. La Figure 4 représente l'architecture de ce réseau.

[6]Celui appliqué à l'exemple jouet et celui appliqué à l'exemple réel

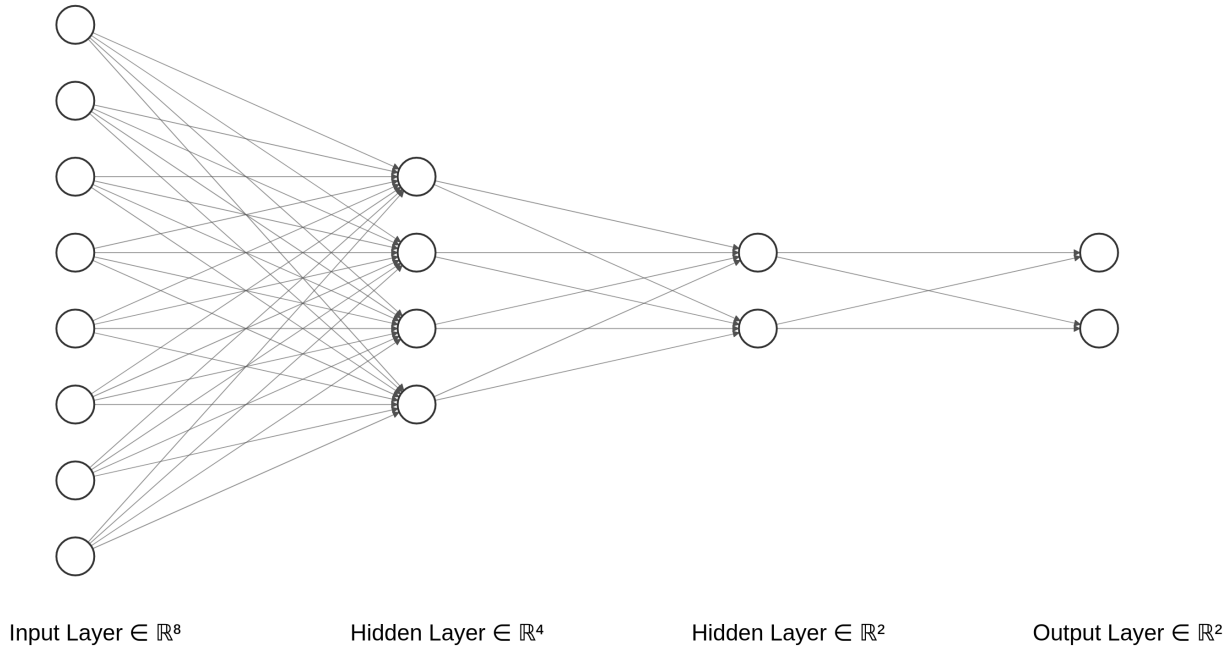


Figure 4 – Architecture du réseau de neurone pour l'exemple d'assurance

3 Influence des observations sur les paramètres

La question à laquelle nous nous intéressons dans cette partie est : Comment réagiraient nos paramètres si nous perturbions légèrement l'échantillon d'apprentissage ? La première forme de perturbation que nous allons étudier ici sera le retrait d'un point de D_{train}. Informatiquement, il n'est pas envisageable de réentraîner un modèle sur une version de l'échantillon d'apprentissage que l'on priverait successivement de chacun de ses points. Le coût d'une telle procédure dépasserait de loin les apports en termes d'intelligibilité. Cependant, grâce aux fonctions d'influence, il est possible d'obtenir efficacement une approximation de cet effet sur les paramètres d'une boîte noire.

Nous souhaitons approcher $\hat{\theta}_n - \hat{\theta}_{n,-z}$, où $\hat{\theta}_{n,-z} = \arg\min_{\theta\in\Theta} \frac{1}{n-1} \sum_{z_i \neq z} \mathcal{L}(z_i, \theta)$. De plus, nous noterons $\hat{\theta}_{n,\epsilon,-z}$ le vecteur des paramètres optimaux pour la distribution contaminée ($F_{\epsilon,z} = (1-\epsilon)F + \epsilon\delta_z$). Nous restreignons notre étude au cas des M-estimateurs [Huber, 1981]. Ces derniers forment une classe importante de statistiques. En pratique, elles s'obtiennent par la minimisation d'une fonction dépendant des données et des paramètres du modèle. Les M-estimateurs peuvent être vus comme une généralisation de l'estimation par maximum de vraisemblance.

Dans le cas d'un M-estimateur on cherche les paramètres qui vérifient $\arg\min_{\theta\in\Theta} (\sum_{i=1}^{n} \rho(z_i, \theta))$, où ρ est la fonction à minimiser sur l'ensemble des données. Il peut s'agir de la log-loss (ou log-vraisemblance) par exemple dans le cas de la régression logistique binaire. En posant $\rho = 1/n \times \mathcal{L}$ on est ramené à notre problème initial de minimisation de risque empirique décrit équation (1).

Définition 3.1. *On définit $\mathcal{I}_{up,params}(z)$ la variation du vecteur de paramètres θ liée à une perturbation infinitésimale ϵ au point z par*

$$\mathcal{I}_{up,params}(z) = \left.\frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon}\right|_{\epsilon=0}.$$

Sous cette forme, $\mathcal{I}_{up,params}$ est difficilement exploitable. Il convient de trouver une formule fermée plus facilement implémentable. C'est l'objet de la Proposition 3.1.

Proposition 3.1.

$$\mathcal{I}_{up,params}(z) = -H^{-1}_{\hat{\theta}_n} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n),$$

où $\nabla_\theta \mathcal{L}(z, \hat{\theta}_n) \in \mathbb{R}^d$ est le gradient de la fonction de perte $\mathcal{L}$ par rapport au paramètre θ et évalué au point z pour le paramètre optimal $\hat{\theta}_n$ et $H_{\hat{\theta}_n} = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_\theta \mathcal{L}(z_i, \hat{\theta}_n)$ avec $\nabla^2_\theta \mathcal{L} \in \mathcal{M}_d(\mathbb{R})$ la matrice contenant toutes les dérivées partielles d'ordre deux par rapport à θ.

La Proposition 3.1 permet de quantifier l'impact sur les paramètres d'une diminution de masse ($\epsilon = -1/n$). Il reste maintenant à montrer que cette fluctuation de masse est bien équivalente au retrait du point z dans l'échantillon d'apprentissage. C'est l'objet de la Proposition 3.2. Nous définissons d'abord la notion de o_p.

Définition 3.2. *Un suite de vecteurs aléatoires $\{Z_n,\ n \in \mathbb{N}\}$ définie sur l'espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ est infiniment petite en probabilité si Z_n converge en probabilité vers 0. Ceci est noté $Z_n = o_p(1)$. De manière immédiate, on a l'équivalence*

$$Z_n = o_p(1) \iff \forall \epsilon > 0 \lim_{n \longrightarrow +\infty} \mathbb{P}(||Z_n|| > \epsilon) = 0.$$

Il est possible de trouver plus de détails dans [Gourieroux and Monfort, 1995].

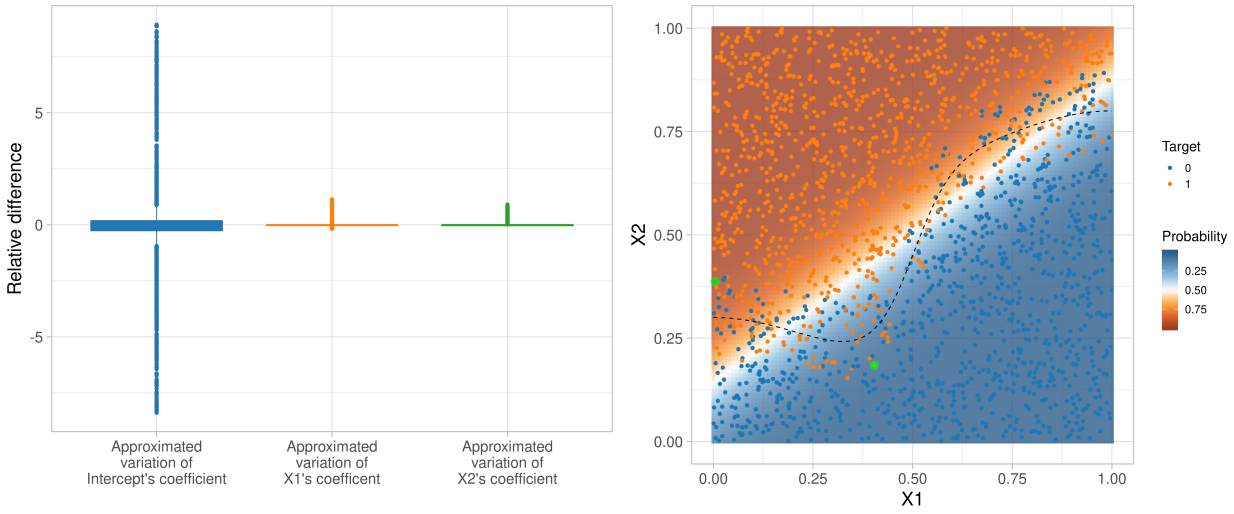
Proposition 3.2. *Soient z un point de l'échantillon d'apprentissage, $\hat{\theta}_n$ et $\hat{\theta}_{n,-z}$ les coefficients estimés respectivement avec et sans la présence du point z dans l'échantillon d'apprentissage.*

$$\hat{\theta}_{n,-z} - \hat{\theta}_n = \frac{1}{n} H^{-1}_{\hat{\theta}_n} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) + o_p(\frac{1}{n}).$$

Grâce à ces deux propositions, il est maintenant possible d'estimer l'influence des points sans avoir à supprimer un à un les points de l'échantillon d'apprentissage. Il suffit en effet de calculer l'inverse de la hessienne associée à la perte ainsi qu'un gradient.

Interprétation géométrique

Il est possible de voir $\mathcal{I}_{up,params}$ comme un unique pas de l'algorithme de *Newton-Raphson* appliqué à la fonction de perte pour minimiser cette dernière. En d'autres termes, c'est la direction qui minimise la perte pour notre modèle et pour un jeu de données. Ceci correspond graphiquement à la variation de pente de notre frontière de décision.



(a) Boxplot des variations relatives des coefficients

(b) Frontière de décision

Figure 5 – Variation des coefficients d'une régression logistique consécutive au retrait d'un point de D_{train}

3.1 Mise en application $\mathcal{I}_{up,params}$

Les graphiques et calculs sont réalisés avec **R** [R Core Team, 2018] et les packages **tensorflow**[7],**keras** et **h2o**.

Nous nous intéressons désormais aux informations que l'indicateur $\mathcal{I}_{up,params}$ peut nous apporter. Rappelons que $\mathcal{I}_{up,params}$ est un vecteur contenant la variation, de chaque paramètre, équivalente au retrait d'un point z de D_{train}. Dans le cas de la régression logistique binaire que nous développons ici, cela permet de connaître de manière immédiate les points qui sont les plus influents sur une variable donnée. En revanche, pour un réseau de neurones, ce n'est pas si simple puisque les poids du réseau n'ont pas de signification immédiate. C'est pourquoi exceptionnellement nous n'appliquerons pas cet indicateur aux réseaux de neurones.

Contrairement à d'autres méthodes d'intelligibilité, l'output n'est pas directement une explication de la chaîne de décision de la boîte noire. Pour obtenir des informations utiles, il faut examiner les points les plus influents. Cela n'est pas toujours facile a fortiori en grande dimension. Toutefois, le boxplot Figure 5a permet d'avoir un aperçu général de la sensibilité des coefficients aux observations de D_{train}.

Sur ce boxplot, il est possible de voir que le coefficient d'intercept du modèle est le plus sensible au retrait d'un point de l'échantillon d'apprentissage. En effet, si l'on retire un point correspondant aux extremas du boxplot bleu, le coefficient d'intercept du modèle augmente ou diminue d'environ 8.5%. Ces deux points sont représentés en vert sur la Figure 5b. Ils correspondent à des points mal classés par le modèle, dans une zone où le modèle est très confiant dans sa prédiction e.g. un orange dans une zone bleu foncé.

Ce premier indicateur renseigne donc sur la sensibilité des paramètres de la boîte noire à certaines observations. De cette manière il est possible de faire émerger des points aberrants ou très influents

[7]https://tensorflow.rstudio.com/

et de les analyser au cas par cas.

3.2 Identification de points atypiques

Appliquons maintenant l'indicateur $\mathcal{I}_{up,params}$ sur notre régression logistique ajustée sur notre jeu de données d'assurance. L'objectif est d'identifier les variables les plus sensibles au retrait d'un point de D_{train}. Comme pour l'exemple jouet, l'information peut-être synthétisée sous la forme d'un boxplot Figure 6. Il semble que les coefficients des variables "Ancienneté permis 2", "Vitesse du véhicule", "Valeur du véhicule" et "Poids du véhicule" soient sensibles à certaines observations de D_{train}. En effet, le retrait d'un seul point parmi les 74983 observations de D_{train} peut entraîner des variations pouvant aller jusqu'à -15%. En revanche, le coefficient d'intercept semble très peu sensible aux points de D_{train}. Notons que les variables les plus sensibles sont cohérentes avec ce que l'on peut attendre du point de vue actuariel. En effet, les variables identifiées comme pertinentes pour l'identification de la sinistralité dans la littérature actuarielle sont liées à l'expérience du conducteur et à la puissance du véhicule. Citons notamment [Esbjörn Ohlsson, 2010] et [Charpentier, 2014] ainsi qu'une étude faisant le lien entre des données classiques et des données *telematics* [Verbelen et al., 2018].

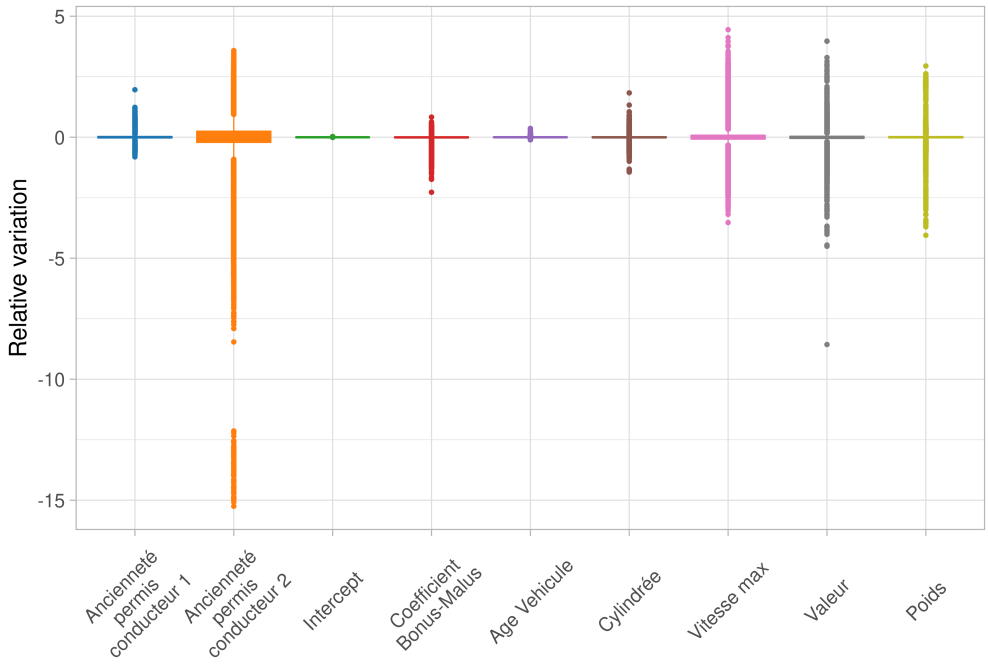


Figure 6 – Sensibilité de chaque prédicteur au retrait de $z \in D_{train}$

Nous pouvons aussi analyser dans le détail les points dont le retrait perturbe le plus les coefficients. Par exemple, en regardant les caractéristiques du point qui entraîne la plus forte variation du coefficient affecté au "Poids du véhicule", nous pouvons constater que c'est un des véhicules les plus légers parmi les plus puissants et plus chers. Ce point atypique est donc plus difficile à classer pour le modèle et apparaît donc comme très influent. Il fait varier les coefficients de "Poids du véhicule" et de "Valeur du véhicule" respectivement de $+2.95\%$ et -8.57%. Un autre exemple est le point

Table 1 – Points influents

	Bonus-Malus	Ancienneté 1	Ancienneté 2	Age Véhicule	Cylindrée	Vitesse max	Valeur	Poids	Target
7743	0.50	41	0	19	3299	270	112538	1260	1
49950	0.57	16	0	51	425	88	1060	880	1
65499	0.50	111	0	12	1598	195	16770	1230	1

dont le retrait perturbe le plus l'estimation du coefficient "Ancienneté de permis 2". Ce point a une ancienneté de 111 ans. Ici, il est probable que ce soit une valeur mal saisie. Ainsi, en analysant les observations dont le retrait modifie le plus l'estimation des coefficients affectés aux variables, il est possible d'identifier des anomalies dans la base D_{train} ou bien simplement des individus atypiques qu'il convient de traiter par la suite.

Limites

Pour que cet indicateur soit pertinent, il paraît nécessaire que les paramètres du modèle soient directement liés aux variables comme c'est le cas pour la régression logistique. Pour les réseaux de neurones, par exemple, cet indicateur n'est pas sans intérêt mais ce qui a été développé ci-dessus ne tient plus.

Une autre limite importante est la difficulté à synthétiser l'information contenue dans la sensibilité du modèles par rapport aux points de D_{train}. Ici, nous adoptons une approche graphique pour pallier ce problème.

4 Identification des points influents pour une prédiction

Nous nous intéressons maintenant à l'effet du retrait d'un point de D_{train} sur la fonction de perte évaluée en un point quelconque que nous noterons z_{test}. Nous souhaitons donc approximer la différence de perte $\mathcal{L}(z_{test}, \hat{\theta}_{n,-z}) - \mathcal{L}(z_{test}, \hat{\theta}_n)$. Comme pour $\mathcal{I}_{up,params}$ nous développerons la preuve dans le cas d'un M-estimateur. Identifier des points influents selon $\mathcal{I}_{up,loss}$, l'indicateur que nous développons ci-après, s'avère utile pour détecter des valeurs aberrantes mais permet aussi de localiser des points proches de la frontière de décision. Grâce à ces points, nous sommes donc capables de localiser la frontière de décision et de fournir des modèles de substitution fidèles à cette dernière localement. Par ailleurs, $\mathcal{I}_{up,loss}$ est très utile pour prioriser les points à analyser, c'est-à-dire ceux pour lesquels il est utile de construire un hyperplan pour se substituer à la frontière complexe.

Définition 4.1. *Soient z et z_{test} deux points appartenant respectivement à l'échantillon d'apprentissage et de test. On définit la variation de perte au point z_{test} liée à une variation infinitésimale de ϵ par*

$$\mathcal{I}_{up,loss}(z, z_{test}) = \left. \frac{d\mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z})}{d\epsilon} \right|_{\epsilon=0}.$$

L'idée sous-jacente est grossièrement la même que pour $\mathcal{I}_{up,params}$: approximer l'effet du retrait de z sur $\mathcal{L}$ au point z_{test} en perturbant la fonction de répartition des observations de $\epsilon = -1/n$ en z. La Proposition 1.3 établit une formule fermée pour cet indicateur.

Proposition 4.1.

$$\mathcal{I}_{up,loss}(z, z_{test}) = \left.\frac{d\mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z})}{d\epsilon}\right|_{\epsilon=0} = -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n)^\intercal H_{\hat{\theta}_n}^{-1} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n),$$

En estimant l'inverse de la hessienne et les gradients aux points d'intérêt, on a une bonne approximation de l'effet du retrait de z sur la prédiction en z_{test}.

Interprétation géométrique

Pour certains modèles, il est possible de montrer que

$$\phi : (x, y) \to x^\intercal H_{\hat{\theta}_n}^{-1} y$$

est un produit scalaire. Ceci nous permettra de fournir une interprétation géométrique de cet indicateur. Nous traitons ici le cas de la régression logistique binaire. Pour cela, nous devons montrer que la forme linéaire associée à $H_{\hat{\theta}_n}$ est définie et positive. La Proposition 1.4 formalise cela.

Proposition 4.2. *Dans le cas d'une régression logistique binaire régularisée, $H_{\hat{\theta}_n}$ est définie positive.*

Soit un modèle de régression logistique binaire et ayant une fonction de perte L_2-régularisée de paramètre $\alpha > 0$, et $H_{\hat{\theta}_n}$ la matrice Hessienne associée à cette dernière. Dans ces conditions,

$$H_{\hat{\theta}_n} \textit{ est définie positive.}$$

En montrant que $H_{\hat{\theta}_n}$ est définie positive, on montre par la même occasion que $H_{\hat{\theta}_n}$ est inversible. Ceci montre que ϕ existe. Il est alors assez facile de voir que ϕ est un produit scalaire.

Cette propriété est intéressante, puisqu'en calculant $\phi(z, z)$, on obtient la norme du gradient au point z qui se trouve être aussi $\mathcal{I}_{up,loss}(z, z)$. En faisant cela, on a placé les deux gradients dans la même direction, ce qui maximise le projeté. Cela permet d'avoir une mesure de l'importance de chaque point de D_{train} pour le modèle.

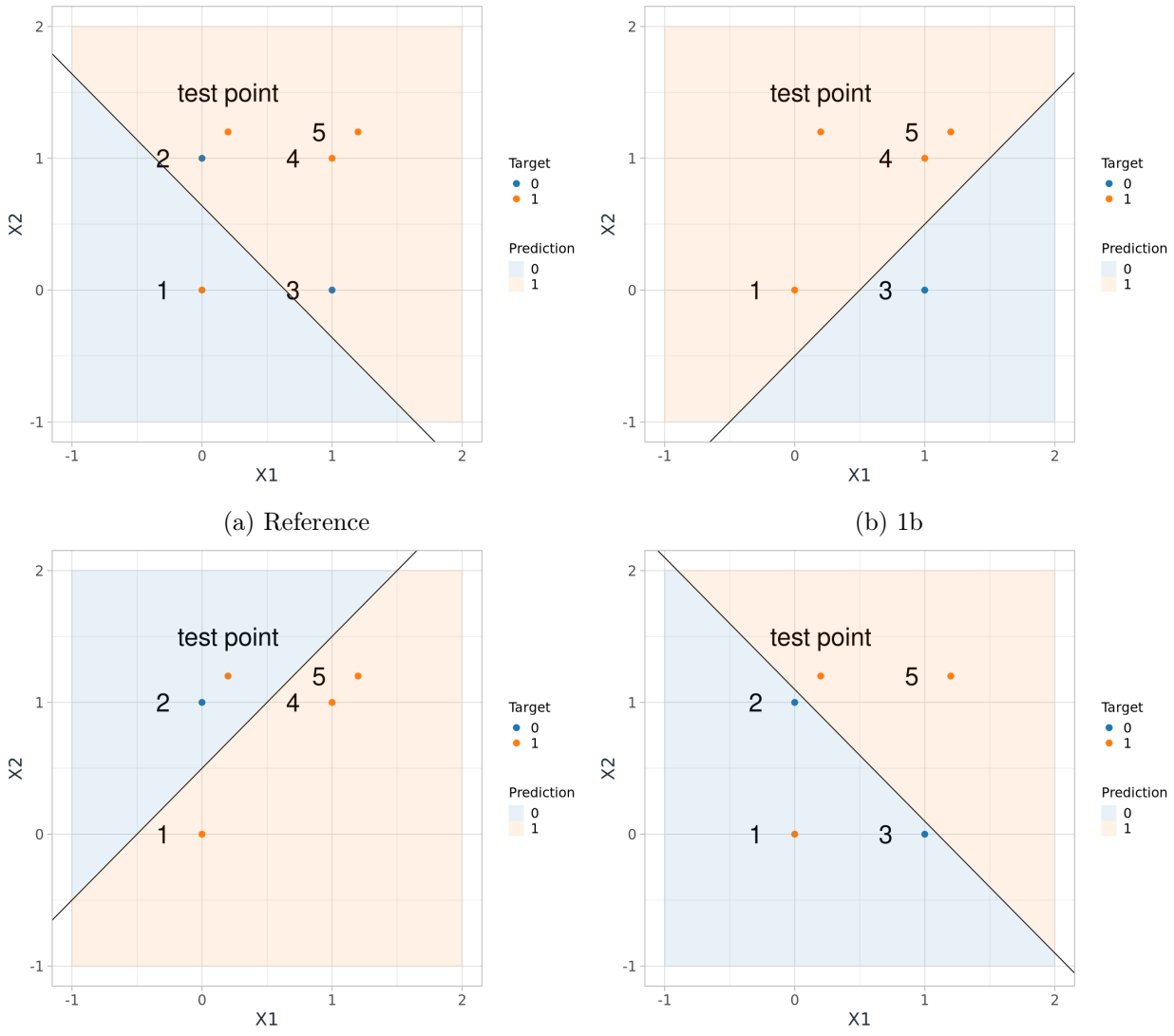
4.1 Mise en oeuvre de $\mathcal{I}_{up,loss}$

Pour bien comprendre comment l'influence varie dans le cas de la log loss, nous utilisons un jeu de données constitué de 5 points d'apprentissage et un point de test. Dans cette première phase, une régression logistique est ajustée sur toutes les données de l'échantillon d'apprentissage. Cette dernière sera notre modèle de référence. Ensuite, les points de l'échantillon d'apprentissage sont retirés un à un et une régression logistique est ajustée sur les échantillons générés par le retrait d'un point. Enfin, pour chaque modèle [8] les classes/probabilités du point z_{test} sont prédites. Pour cette étude, si la probabilité est supérieure à 0.5, la classe prédite sera 1 sinon 0. Rappelons que :

$$\mathcal{I}_{up,\ loss}(z, z_{test}) = \mathcal{L}(z_{test}, \hat{\theta}_{n,-z}) - \mathcal{L}(z_{test}, \hat{\theta}_n).$$

[8] modèle de référence et les modèles sur les données privées d'un point

Notons que cette définition permet de rendre la notion d'influence, définie ci-dessus, cohérente avec l'influence telle qu'on la conçoit au quotidien. En effet, si le retrait du point améliore le modèle (diminue la perte par rapport au modèle de référence) alors l'influence de ce dernier est négative. Garder ce point dégrade donc la qualité du modèle et a une influence négative sur lui. De manière équivalente, un point a une influence positive si son retrait augmente la perte.



(a) Reference

(b) 1b

(c) 1c

(d) 1d

Figure 7 – Influences dans un cas simple

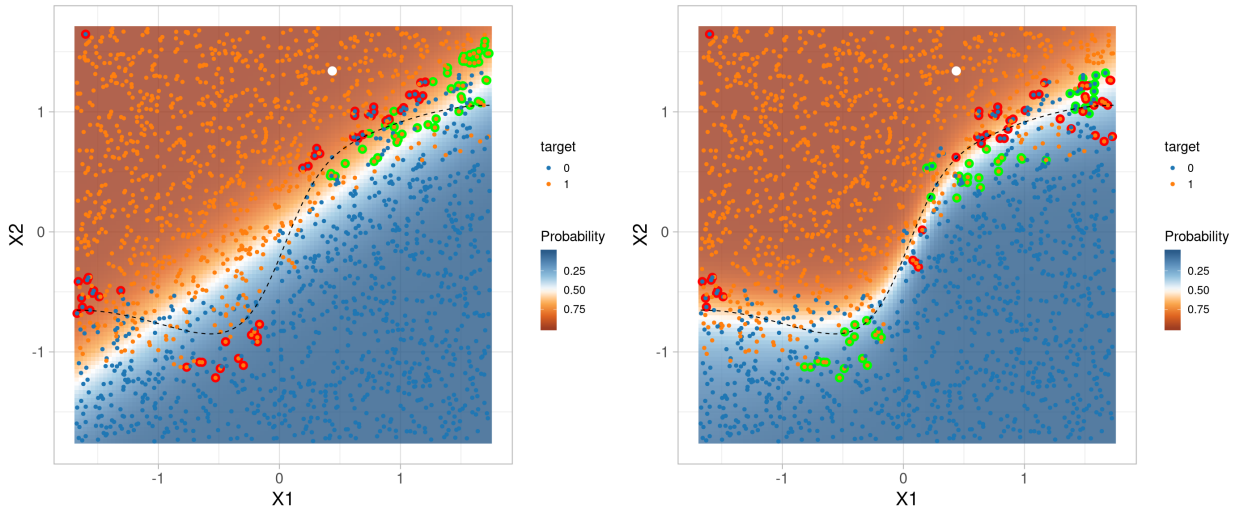
Dans ce cas particulièrement simple Figure 7, il est facile de voir que le classifieur prend ses décisions en fonction de la position des points à classer par rapport à la frontière. Le retrait d'un point de l'échantillon d'apprentissage modifie la frontière et de ce fait les choix du classifieur (valeur ou probabilité). En effet, si l'on regarde le graphique de référence Figure 7a et Figure 7b, on s'aperçoit que le retrait du point 2 modifie fortement la frontière de décision. Cette nouvelle frontière

sépare parfaitement les points de l'échantillon d'apprentissage et par chance l'échantillon du point test. Ainsi, le modèle prédira le point test comme étant de classe 1 avec une probabilité élevée puisque le point test est "à l'intérieur de sa zone" et éloigné de la frontière. La perte dans ce cas est plus faible que celle du modèle de référence pour une prédiction du point test. En conséquence, l'influence est négative comme le montre la Table 5 en annexe. Un raisonnement similaire peut s'appliquer pour le reste des graphiques.

Remarque. *Il est intéressant de remarquer que pour le point test à prédire (classe 1) les points ayant la plus forte influence positive et la plus forte influence négative sont d'une classe différente. C'est un résultat qui n'est pas intuitif mais qui permet de comprendre que la méthode identifie les points liés à la prise de décision d'un algorithme donné. Dans la suite de l'article, nous verrons que les points identifiés sont intrinsèquement liés au modèle boîte noire que l'on étudie.*

4.2 Identification de points aberrants

L'information apportée par $\mathcal{I}_{up,loss}$ est intéressante pour mettre en évidence des points de l'échantillon d'apprentissage importants pour une prédiction donnée. Cela peut s'avérer très utile pour déboguer un modèle. Par exemple, en vérifiant que les données n'ont pas été mal labellisées comme le propose Koh and Liang [2017] ou en analysant les caractéristiques des points les plus négativement influents pour une prédiction au point z_{test}. Ainsi, il est possible de corriger ces observations ou de les supprimer au besoin. En effet, pour une prédiction au point z_{test}, tous les points de D_{train} se voient attribuer une valeur d'influence positive ou négative. Rappelons que l'intensité de cette influence est fonction de l'importance de la perturbation qu'engendrerait le retrait d'un point de l'échantillon d'apprentissage sur la prédiction en un point z_{test}. Par conséquent, les points considérés comme aberrants par un modèle donné doivent avoir une influence négative parmi les plus hautes.



(a) Modèle linéaire

(b) Réseau de neurones

Figure 8 – Modèles boîte noire

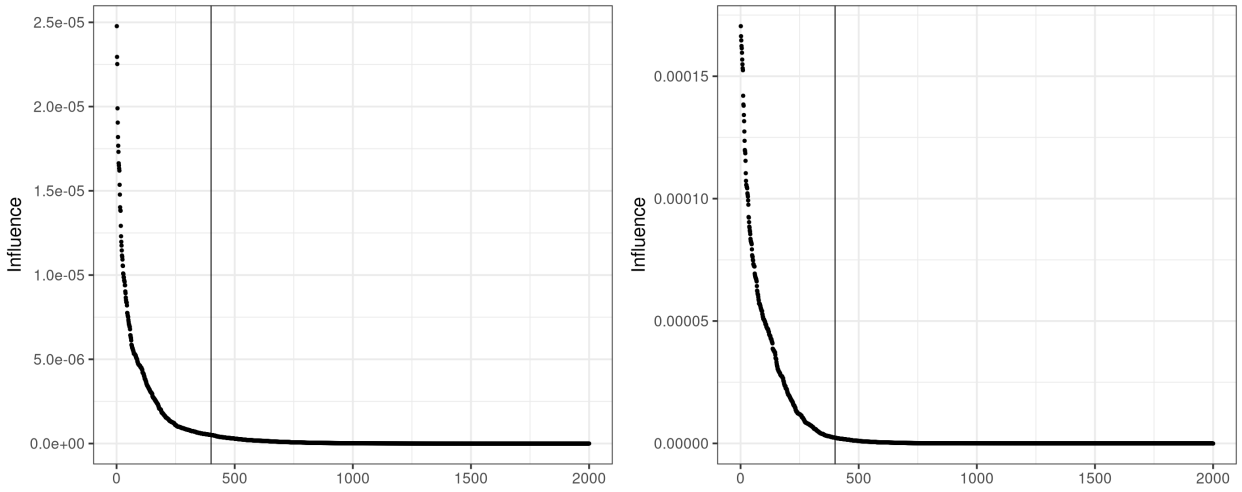
Nous voulons maintenant vérifier s'il est possible d'identifier un point aberrant grâce à $\mathcal{I}_{up,loss}$ et par la même occasion, illustrer l'information fournie par cet indicateur. Pour cela, nous avons créé

deux modèles qui sont représentés Figure 8a et 8b. Le gradient de couleur représente ce que chacun des modèles prédit. Plus le modèle a confiance dans sa prédiction, plus la couleur est foncée. Le blanc marque la zone où le modèle a le plus de mal à trancher, c'est la frontière de décision. Nous pouvons d'ores et déjà noter que le modèle linéaire n'est pas assez souple pour approcher correctement la vraie frontière de décision en pointillés. En revanche, le réseau de neurones régularisé s'en approche bien mieux. Nous avons volontairement modifié le label d'un point en haut à gauche du graphique. De cette manière, nous avons créé un point aberrant. Nous fixons arbitrairement un point à prédire z_{test} qui appartient à D_{test}. Ce point est matérialisé par un halo blanc sur le graphique. Les points cerclés d'un halo vert sont les 50 points les plus influents positivement sur cette prédiction, tandis que les rouges sont les 50 points les plus négativement influents. Nous pouvons noter que les points identifiés pour les deux modèles ne sont pas nécessairement les mêmes. En revanche, ils sont localisés à proximité des frontières de chacun des modèles. Par ailleurs, notre point aberrant en haut à gauche a bien été identifié par la méthode. Pour les deux modèles, ce dernier est le point le plus négativement influent. Dans le cas du modèle linéaire, l'influence de ce point, en valeur absolue, est plus de 27 fois supérieure au quantile à 5% de la distribution des influences, tandis que pour le réseau de neurones l'influence de ce point est 9 fois supérieure par rapport au quantile 5%. Notons que cette fois, l'indicateur peut s'utiliser aussi bien pour la régression logistique que pour les réseaux de neurones. Cela peut donner une méthodologie pour diagnostiquer efficacement les points aberrants pour un modèle donné. Cependant, pour être exhaustif, il faudrait analyser autant de distributions qu'il existe de points à tester ce qui dans la pratique peut s'avérer difficile.

4.3 Extraction d'explications locales

Nous pouvons constater qu'il est très difficile d'extraire d'autres informations pertinentes de cet indicateur étant donné que la localisation des points varie énormément dans l'espace. Si nous décidions d'utiliser une méthode interprétable par nature comme un arbre de régression par exemple prenant comme nouvelle cible l'influence signée de chaque point, nous aurions du mal à avoir un arbre peu profond et fidèle. Ce qui signifie que notre explication locale du modèle boîte noire ne serait pas satisfaisante. Par conséquent, cette technique proposée par Molnar [2019] ne donnerait pas de résultats fiables et intéressants. Cette remarque est valable pour notre exemple en petite dimension et a fortiori en grande dimension.

Exemple en 2 dimensions



(a) Modèle linéaire

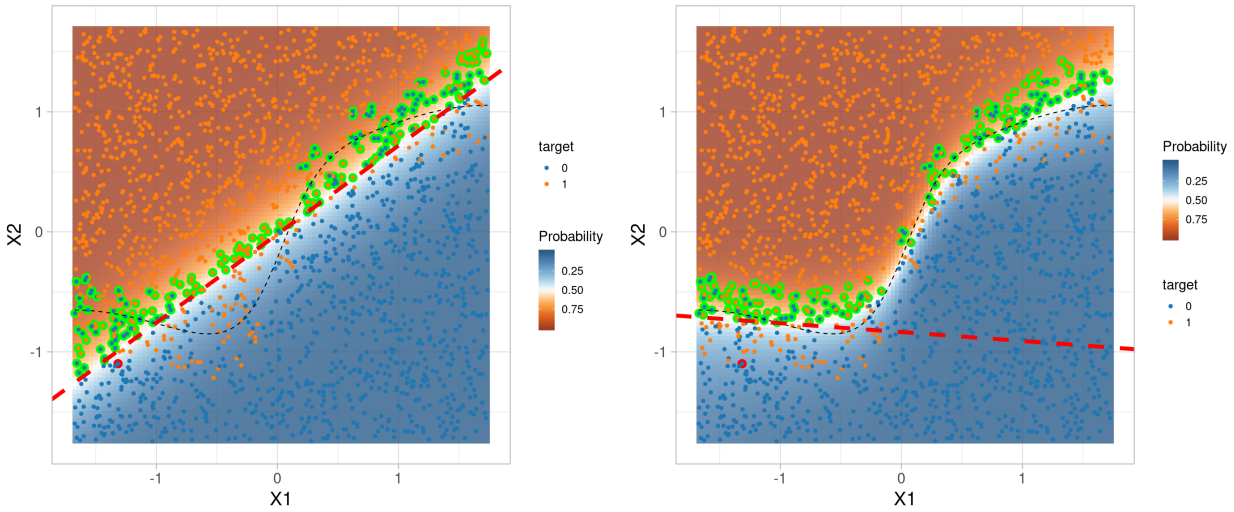
(b) Réseau de neurones

Figure 9 – Observations ordonnées par influence

Nous proposons donc une approche différente pour extraire de l'information pertinente en vue d'expliquer notre régression linéaire et notre réseau de neurones qui jouent le rôle de boîtes noires. Puisque la formule $\mathcal{I}_{up,loss}$ est valable pour un point quelconque z_{test}, elle l'est à fortiori pour les points de D_{train}. Or, les deux matrices hessiennes sont définies positives. Dans ce cas précis, $\mathcal{I}_{up,loss}(z, z)$ représente donc une norme pour le gradient de la fonction de perte évaluée au point z. Cette propriété nous permet d'avoir une distribution d'influences positives ou nulles. De plus, nous avons considérablement réduit la quantité d'informations puisque nous n'avons qu'une valeur d'influence par point de D_{train}. Ce qui est plus simple à interpréter. En effet, le retrait d'un point d'influence élevée perturbera fortement le modèle tandis que le retrait d'un point d'influence nulle ou quasi nulle n'entraînera que de faibles modifications sur les valeurs prédites par la boîte noire. Dans les cas où les matrices hessiennes ne sont pas définies positives, nous prendrons simplement la valeur absolue des valeurs d'influences pour pouvoir appliquer la méthodologie décrite plus bas. Nous représentons pour chaque modèle boîte noire les points influents par ordre décroissant sur les Figures 9a et 9b. Le choix d'un seuil pour le nombre de points influents incombe toujours à l'utilisateur. Cependant, sous cette forme, il devient beaucoup plus aisé de privilégier les points considérés comme pertinents pour le modèle. Nous retenons arbitrairement 400 observations pour les deux modèles. Ces dernières sont ensuite représentées dans le plan. Elles sont identifiées par un halo vert[9]. Comme nous pouvons le voir Figure 10a et 10b les points identifiés comme influents sont différents pour chaque modèle. En revanche, ils ont la propriété intéressante d'être localisés à proximité de la frontière de décision. Ce qui est cohérent avec ce que l'on a développé plus haut. Cependant, il reste difficile d'extraire une explication depuis cet ensemble de points. En effet, étant donné la localisation des points, prendre la moyenne sur chacune des coordonnées n'est d'aucune

[9]Pour que les graphiques restent lisibles et cohérents avec les explications qui suivront, nous n'affichons que les points influents qui se trouvent du côté opposé (de la frontière) au point rouge.

utilité. Néanmoins, nous pouvons nous servir de l'influence comme d'une nouvelle information. Contrairement, à ce qui est proposé par Molnar [2019] nous n'allons pas ajuster un modèle directement sur l'influence de chaque observation mais plutôt nous servir des propriétés liées à ces points et aux modèles boîte noire.



(a) Modèle linéaire

(b) Réseau de neurones

Figure 10 – Points les plus influents pour les modèles boîtes noires et explications.

Nous proposons la méthodologie suivante pour fournir une explication locale, c'est à dire valable uniquement au voisinage du point expliqué.

1. Sélectionner une instance à expliquer que nous noterons z_{exp}.

2. Parmi les points les plus influents, conserver uniquement ceux qui sont de l'autre côté de la frontière par rapport au point à expliquer. Nous pouvons facilement obtenir cette information puisque nous disposons de la boîte noire que nous pouvons interroger dans tout l'espace.

3. Déterminer les points d'intersection entre la frontière de décision et les droites passant par le point à expliquer et les points influents de l'autre côté de la frontière identifiés à l'étape précédente. Nous utilisons ici une méthode de dichotomie ce qui nous permet de fixer arbitrairement la précision.

4. Déterminer pour chaque point d'intersection l'hyperplan $\mathcal{H}_i$ tangent à la surface de décision en ce point. Les hyperplans sont l'équivalent géométrique en dimension d d'un modèle linéaire. Ils constituent ici un ensemble d'explications[10] locales admissibles.

5. Calculer pour chaque point de l'étape 3, la distance au point que l'on cherche à interpréter ainsi que la distance à l'hyperplan $\mathcal{H}_i$. Calculer systématiquement la somme de ces deux distances. Retenir uniquement la distance minimale. Nous noterons cette distance d_{min}. Nous venons de trouver un hyperplan qui peut servir d'explication pour l'instance que l'on étudie. Il nous reste maintenant à en évaluer la fidélité locale.

[10] Aussi appelée surrogate.

6. Il y a plusieurs façons raisonnables de définir un voisinage pour le point à interpréter. Nous pouvons par exemple prendre une boule centrée sur le point que l'on cherche à interpréter et de rayon suffisamment large pour contenir des observations des 2 classes. Il est aussi possible de choisir une boule de même rayon mais centrée sur le point tangent à la surface de décision. C'est cette dernière que nous retenons. Tirer uniformément k points[11] dans cette boule d-dimensionnelle. Ces points sont de nouvelles observations destinées à évaluer la fidélité de l'explication par rapport au modèle boîte noire.

7. Évaluer la fidélité des explications en faisant prédire le modèle linéaire de substitution (notre explication) et le modèle boîte noire sur l'échantillon créé à l'étape précédente. Nous pouvons utiliser l'*accuracy*, l'AUC, la *precision* et le *recall* pour mesurer la fidélité. Cette étape sert à valider l'explication. Une fidélité trop faible nous inciterait à ne pas retenir l'explication.

Les Figures 10a et 10b montrent un exemple d'explication[12] pour le point entouré d'un halo rouge et ce pour nos deux modèles boîtes noires. Selon la métrique *accuracy*, l'explication est fidèle à 100% dans le cas du modèle linéaire et à 99.8% dans le cas du réseau de neurones. Bien entendu ce score de fidélité varie selon l'explication et la complexité du modèle boîte noire. Comme l'on peut s'y attendre, dans le cas du modèle linéaire uniquement, l'explication est très proche de la vraie frontière de décision. Ainsi dans le cas du modèle linéaire uniquement, cette dernière est cohérente avec l'explication globale. Dans le cas plus général du réseau de neurones, elle ne paraît que localement bonne. En plus grande dimension, nous ne pourrons mesurer la qualité d'une explication qu'avec la mesure de fidélité.

Problème d'assurance

Désormais, nous revenons à notre problème de prévention sur des données d'assurance. Nous avons ajusté un modèle linéaire et un réseau de neurones contenant deux couches cachées. La première est composée de 4 neurones tandis que la seconde en contient 2. En général, pour une classification binaire, un neurone de sortie suffit. Cependant, nous avons utilisé le framework **h2o** qui par défaut impose un nombre de neurones de sortie égal au nombre de classes. Ainsi, notre réseau est composé de 44 poids et de 8 biais. Ce qui fait donc 52 coefficients à ajuster au total. Nous avons choisi d'introduire une régularisation de type *L2*. La valeur de cet hyperparamètre a été fixée à 0.001. Les performances des deux modèles sont détaillées dans la Table 2.

Table 2 – Performances des modèles boîtes noires

	threshold	AUC	PRAUC	precision	recall
Modèle linéaire	0.114	0.616	0.171	0.158	0.761
Réseau de neurones	0.121	0.619	0.174	0.157	0.781

Nous souhaitons obtenir une explication au voisinage d'individus influents et ce pour les deux modèles. Cela nous permettra de comparer localement la manière dont les modèles prennent leur décision. Nous prenons les deux premiers individus de la Table 3 pour leur fournir une explication.

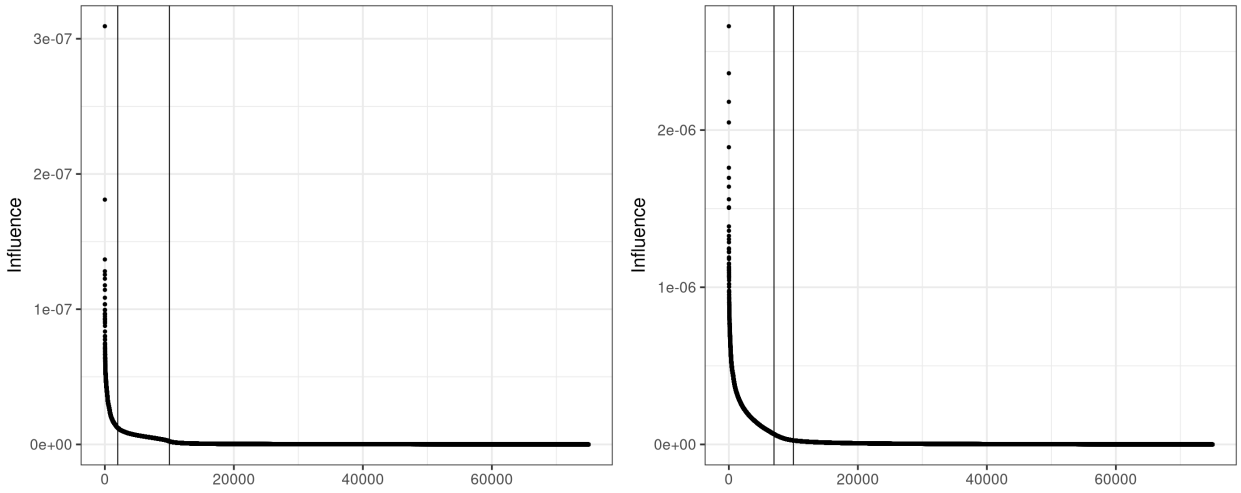
[11] Le choix est laissé à l'utilisateur.
[12] En trait discontinu rouge.

Table 3 – Points influents

	Bonus-Malus	Ancienneté 1	Ancienneté 2	Age Vehicule	Cylindrée	Vitesse max	Valeur	Poids	Target
984	0.5	20	0	23	1913	118	19381	1480	1
1619	0.5	60	0	11	1998	230	30250	1490	1
32294	0.5	41	0	28	3980	126	17982	1970	1
7934	0.5	58	0	13	1396	185	13568	1000	0

En appliquant la méthodologie proposée précédemment, nous obtenons une distribution d'influences représentées sur les Figures 11a et 11b. Les deux points les plus influents pour le modèle linéaire sont les mêmes que ceux qui avaient été identifiés avec l'indicateur précédent. Pour le réseau de neurones, la matrice hessienne est inversible mais pas définie positive. C'est pourquoi nous obtenons des valeurs d'influence positives et négatives. Pour choisir notre seuil de points à conserver, nous prenons donc la valeur absolue des influences. Notez qu'ici encore, des points semblent se distinguer par leur très forte influence. Il s'agit des points 32294 et 7934 dont les caractéristiques sont données dans la Table 3. En observant la cylindrée et la vitesse du point 32294, nous constatons qu'il y a un problème de cohérence entre la cylindrée renseignée et la vitesse maximale affichée. Ceci peut expliquer cette forte valeur d'influence pour le modèle linéaire. Comme nous venons de le montrer, dans la pratique, $\mathcal{I}_{up,loss}$ s'avère très utile pour identifier des points anormaux. Les modèles linéaires généralisés étant encore très répandus chez les assureurs, l'utilisation de cet indicateur peut s'inscrire dans une démarche d'amélioration de la qualité des données.

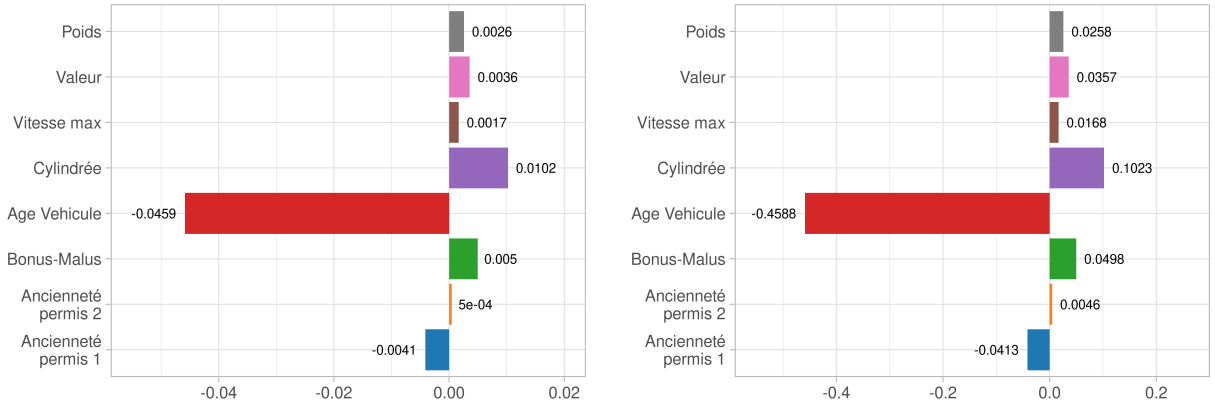
Nous voulons maintenant extraire des explications locales pour les points que nous avons sélectionnés. En nous basant sur les Figures 11a et 11b nous choisissons 2250 points pour le modèle linéaire et 7000 pour le réseau de neurones. En fournissant les explications pour le modèle linéaire, nous souhaitons nous assurer qu'en plus grande dimension, les explications fournies sont toujours cohérentes avec le modèle linéaire lui même. Les Figures 12b et 12a abondent dans ce sens. Dans le cas particulier du modèle linéaire, cette observation tient bien évidemment pour tous les points et pas uniquement pour ces deux exemples. Nous l'avons vérifié mais ne pouvons pas tout afficher. Pour le réseau de neurones, nous attendons des hyperplans différents pour les explications des individus 984 et 1619 puisque le réseau de neurones est non linéaire. C'est ce que nous constatons sur les Figures 13a et 13b. Selon la métrique *accuracy*, l'explication du point 984 pour le modèle linéaire est fidèle à 100% tandis que pour le réseau de neurones, elle n'est fidèle qu'à $97,4\%$. Ce qui reste assez élevé pour considérer que l'explication est fiable au voisinage de ce point.



(a) Modèle linéaire (b) Réseau de neurones

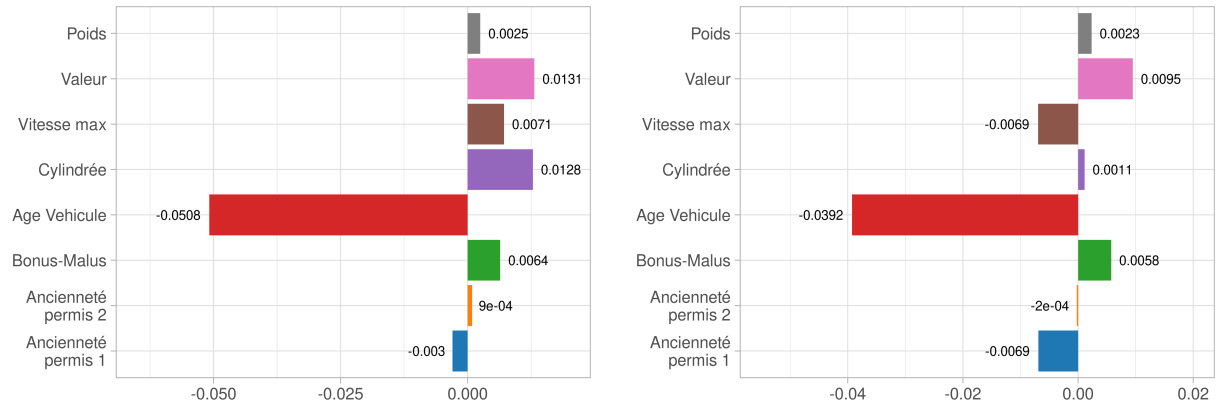
Figure 11 – Points les plus influents pour les modèles boîtes noires et explications

Remarque. *Souvent, lorsqu'un modèle de substitution est ajusté pour expliquer les résultats d'un modèle plus complexe, la nature de l'explication peut paraître incohérente avec notre intuition. En effet, en tant qu'humain, nous serions tentés d'accorder une importance plus grande à l'ancienneté du conducteur principal qu'à l'âge du véhicule pour prédire la survenance d'un sinistre. Cependant, il ne faut pas perdre de vue que l'on cherche à expliquer ce que le modèle a "compris". Or, un modèle peut être efficace (avoir une bonne performance) mais capter des corrélations non désirées ou solution statistique efficace mais non satisfaisante pour servir d'explication à un humain. Bien que les explications fournies ici ne nous semblent pas cohérentes du point de vue actuariel, elles expliquent bien ce que les différents modèles ont appris.*



(a) Explication point 984 (b) Modèle linéaire

Figure 12 – Explication d'un point et modèle linéaire de référence



(a) Explication point 984

(b) Explication point 1619

Figure 13 – Explications des points les plus influents pour le réseau de neurones

5 Identification d'une direction de perturbation maximale

Contrairement aux indicateurs précédents, dans cette partie, nous ne nous intéressons pas aux variations liées au retrait d'un point de D_{train} mais à la perturbation de ce dernier. En effet, nous souhaitons définir la direction dans laquelle déplacer un point de D_{train} pour maximiser la variation de perte pour une prédiction quelconque. L'intérêt d'un tel indicateur est dans notre cas d'avoir une direction dans laquelle trouver un exemple adverse "adversarial exemple" c'est à dire une observation fictive, proche de l'observation initiale et qui serait prédite avec une classe différente par le modèle boîte noire. Dans notre cadre d'étude, cela nous permettra de localiser la frontière de décision.

Cette fois, l'idée est de transférer la masse ϵ d'un point existant z vers un point $z_\delta = (x + \delta, y)$ perturbé. Cela peut être vu en deux temps, comme le retrait du point z de D_{train} puis l'ajout d'un point z_δ dans D_{train}. Nous notons en conséquence, $\hat{\theta}_{n,z_\delta,-z}$ le minimiseur du risque empirique où z_δ remplace z dans D_{train} et $\hat{\theta}_{n,\epsilon,z_\delta,-z}$ le vecteur des paramètres optimaux associé à la distribution contaminée ($F_{\epsilon,z,z_\delta} = F - \epsilon\delta_{z_\delta} + \epsilon\delta_z$). Dans un premier temps, nous ne faisons pas d'hypothèses particulières sur la nature des variables explicatives ni sur δ.

Définition 5.1. *Soient z et z_{test} deux points appartenant respectivement à l'échantillon d'apprentissage et test. On définit la variation de perte au point z_{test} liée à un transfert infinitésimal de masse ϵ de z vers z_δ par*

$$\mathcal{I}_{pert,loss}(z, z_{test})^\intercal = \nabla_\delta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal \Big|_{\delta=0}.$$

La Proposition 5.1 donne une formule fermée qui est valable même pour des données discrètes. Nous verrons qu'il est possible de pousser plus loin l'approximation dans le cas de données numériques continues.

Proposition 5.1.

$$\mathcal{I}_{pert,params}(z, z_\delta) = \frac{d\hat{\theta}_{n,\epsilon,z_\delta,-z}}{d\epsilon}\Big|_{\epsilon=0} = \mathcal{I}_{up,params}(z_\delta) - \mathcal{I}_{up,params}(z).$$

Dans le cas où certaines variables sont discrètes, cela permet de fixer un vecteur de perturbation δ et de connaître instantanément l'effet de cette perturbation sur la fonction de perte du modèle.

Dans le cas de variables continues, il est possible de pousser l'approximation. C'est l'objet de la proposition suivante.

Proposition 5.2. *Soient z un point de l'échantillon d'apprentissage et z_δ sa version perturbée, $\hat{\theta}_n$ et $\hat{\theta}_{n,z_\delta,-z}$ les coefficients estimés respectivement avec la version originale ou perturbée du point z dans l'échantillon d'apprentissage. Dans le cas de données mixtes (continues et discrètes),*

$$\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n = \frac{1}{n} H^{-1}_{\hat{\theta}_n} (\nabla_\theta \mathcal{L}(z_\delta, \hat{\theta}_n) - \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)) + o_p(\frac{1}{n}) + o(||\delta||).$$

dans le cas de données uniquement continues,

$$\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n = \frac{1}{n} H^{-1}_{\hat{\theta}_n} \left[\nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) \right] \delta + o_p(\frac{1}{n}) + o(||\delta||).$$

où $\nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) \in \mathbb{R}^{p \times d}$

Dans le cas où les données sont numériques continues, il est possible d'obtenir un indicateur plus fin. Ce dernier permet de connaître pour une prédiction au point test les directions/variables dans lesquelles les perturbations δ auront le plus d'impact.

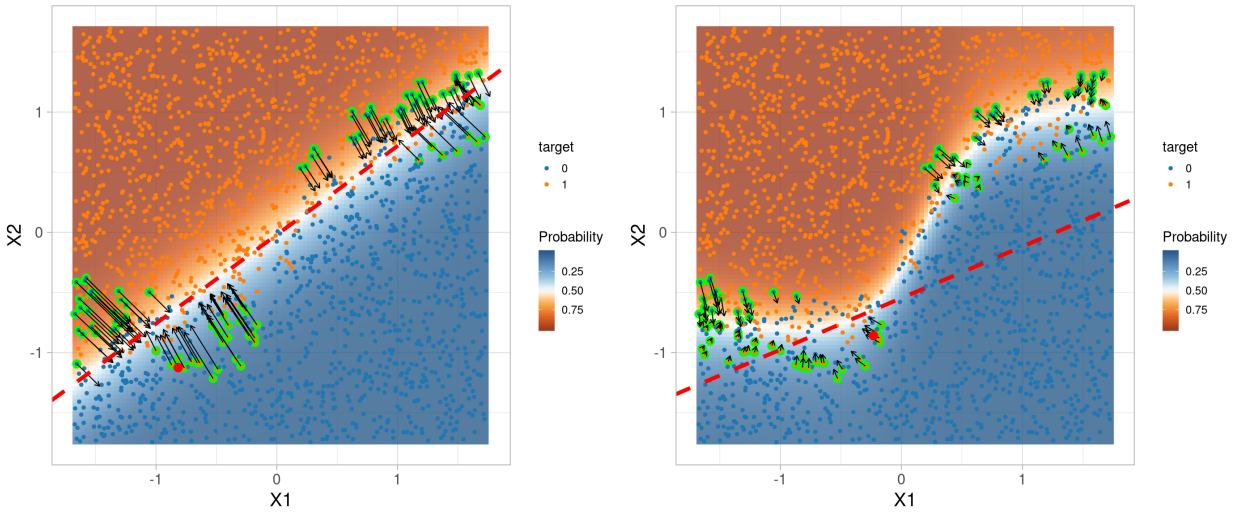
Proposition 5.3. *Supposons que les variables explicatives sont continues ie $x \in \mathcal{X} \in \mathbb{R}^p$ et $||\delta|| \to 0$. Si $\mathcal{L}$ est différentiable par rapport à x et θ alors*

$$\mathcal{I}_{pert,loss}(z, z_{test})^\intercal = -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n)^\intercal H^{-1}_{\hat{\theta}_n} \nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n).$$

Grâce à cet indicateur, nous pourrons obtenir des informations complémentaires sur les variables influentes lors d'une prédiction.

5.1 Mise en oeuvre de $\mathcal{I}_{pert,loss}(z, z_{test})^\intercal$

Nous commençons par illustrer dans le plan l'information portée par $\mathcal{I}^\intercal_{pert,loss}$. Les formules que nous venons d'établir sont valables pour un z_{test} quelconque et a fortiori pour un point de D_{train}. Pour les mêmes raisons que précédemment, nous ne calculerons cet indicateur que pour les points de D_{train} les plus influents. En effet, ce que nous avons établi précédemment nous a permis de définir une notion d'importance des points de D_{train}. Nous pouvons donc privilégier les points d'intérêt plutôt que d'analyser l'intégralité des points. Nous représentons sur les Figures 14a et 14b les vecteurs portant la direction dans laquelle déplacer les cent points les plus influents pour perturber le plus la prédiction de ces mêmes points. Par ailleurs, nous avons représenté en rouge un point choisi au hasard parmi les points les plus influents. Puisque nous disposons d'une direction qui pointe vers la frontière de décision, nous avons un vecteur orthogonal à un ensemble d'hyperplans dont l'un doit être tangent à la surface de décision. C'est pour cela que l'on peut se servir de cet indicateur comme d'une explication. L'idée avait été évoquée par Koh and Liang [2017].



(a) Modèle linéaire (b) Réseau de neurones

Figure 14 – Direction de perturbation privilégiée pour les points les plus influents des modèles boîte noire

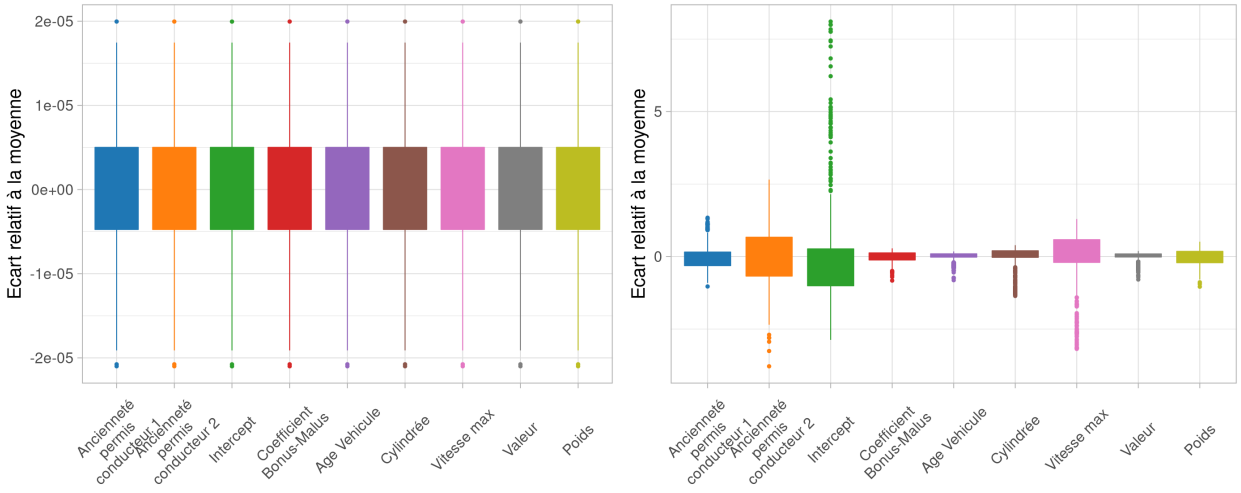
Nous affichons de plus sur ces figures l'hyperplan qui constitue l'explication pour l'observation repérée par un point rouge. Encore une fois pour le modèle linéaire, l'hyperplan est cohérent avec la frontière de décision. Pour le réseau de neurones, il semble que l'approximation soit bonne. Toutefois pour vérifier la qualité de cette approximation, nous devons comme précédemment définir un voisinage pour ce point et comparer pour différentes métriques la fidélité de l'hyperplan, notre explication, par rapport au modèle boite noire. La table 4 synthétise les mesures de fidélité pour ces explications.

Table 4 – Points influents

	Modèle	Accuracy	AUC	F1 Score	Precision	Recall	PRAUC
361	Modèle linéaire	1	1	1	1	1	0.998
74973	Modèle linéaire	1	1	1	1	1	0.998
15540	Réseau de neurones	0.915	0.994	0.919	0.998	0.852	0.990
3973	Réseau de neurones	1	1	1	1	1	0.998

Appliquons désormais cette méthodologie à notre exemple sur des données réelles d'assurance. Grâce à la notion d'importance des points établie dans la partie précédente, nous restreignons le nombre d'observations pour lesquelles nous souhaitons une explication à 2250 pour le modèle linéaire et 7000 pour le réseau de neurones. Nous calculons ensuite $\mathcal{I}_{pert,loss}(z,z)^\intercal$ pour chacune de ces observations. Grâce à cela nous pouvons définir un hyperplan pour chaque point influent qui constitue notre explication. Ensuite, nous évaluons, toujours selon le même procédé la qualité de cette dernière. Puisque nous disposons d'un ensemble d'explications, nous pouvons au choix les analyser une par une ou représenter globalement la variabilité associée à celles-ci. Pour cela, nous considérerons, Figures 15a et 15b, les boxplots des distributions des valeurs de chacune des variables

constituant les hyperplans. Nous avons au préalable centré chacune des variables. Comme nous pouvons le voir, dans le cas du modèle linéaire, les explications varient assez peu, ce qui est plutôt rassurant. En revanche, dans le cas du réseau de neurones, les hyperplans sont plus variés. Cela se justifie par la non linéarité du réseau de neurones. Cette notion de variabilité des explications n'est pas directement exploitable. Cependant, elle pourrait permettre de définir des groupes homogènes d'explications et d'améliorer notre compréhension globale du modèle. Cependant, c'est hors de notre cadre d'étude. Les explications fournies sont fidèles dans l'ensemble. Néanmoins, pour le réseau de neurones, elles le sont moins que pour le modèle linéaire comme en témoigne la Table 4.



(a) Modèle linéaire

(b) Réseau de neurones

Figure 15 – Distributions des hyperplans pour les points les plus influents

Limites

Cet indicateur permet donc de définir efficacement un hyperplan fidèle localement pour les points les plus influents. Cependant, il requiert une hypothèse plus forte qu'$\mathcal{I}_{up,loss}$, la continuité des variables explicatives. Or, en assurance, nombreux sont les prédicteurs binaires. Ainsi, cet indicateur n'est pas toujours utilisable dans la pratique. En revanche, s'il n'est pas possible de l'utiliser, nous pouvons tout de même employer la méthode développée en partie précédente avec $\mathcal{I}_{up,loss}$. Cela requiert plus de calculs mais nous permet d'avoir une explication fidèle sous des hypothèses moins fortes. Par ailleurs, nous avons constaté lors de nos tests que pour des points ayant une faible influence, le calcul de $\mathcal{I}_{pert,loss}$ est moins précis et peu donner des hyperplans de moins bonne qualité.

Conclusion et travaux futurs

Dans cet article, nous avons présenté trois indicateurs basés sur les fonctions d'influence. Grâce à ces derniers, il est possible d'identifier efficacement des valeurs considérées comme anormales par un modèle donné ou de prioriser des points à analyser. Par ailleurs, en combinant les informations portées par ces indicateurs à des propriétés des modèles paramétriques, nous avons été capables d'extraire des explications fidèles localement pour des points que nous souhaitons analyser. Dans

la pratique, nous pouvons donc utiliser ces indicateurs dans une démarche d'amélioration de la qualité des données. En effet, les modèles que nous avons étudiés ici incluent les modèles linéaires généralisés, encore très répandus chez les assureurs. Enfin, il est aussi possible d'utiliser les méthodes présentées dans cet article pour déboguer un modèle localement tel que cela pourrait être fait avec LIME ou SHAP. L'avantage, par rapport à ces dernières est que nous pouvons restreindre le nombre de points à analyser.

Il pourrait être pertinent de prolonger ces travaux en adaptant la technique pour la régression. De plus, comme nous avons pu le voir dans le cas du réseau de neurones, nous pouvons envisager de regrouper les explications similaires entre elles afin d'en diminuer le nombre et ainsi d'améliorer encore notre compréhension des modèles boîtes noires paramétriques.

Remerciements

Nous tenons à remercier Arthur Charpentier pour les données qu'il nous a fournies.

References

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

A. Charpentier. *Computational Actuarial Science with R*. Chapman & Hall/CRC The R Series. Chapman and Hall/CRC, 1 edition, 2014. ISBN 1466592591,9781466592599.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

B. J. a. Esbjörn Ohlsson. *Non-life insurance pricing with generalized linear models*. EAA lecture notes. Springer-Verlag Berlin Heidelberg, 1 edition, 2010. ISBN 3642107907,9783642107900.

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29 (5):1189–1232, 10 2001. doi: 10.1214/aos/1013203451. URL `https://doi.org/10.1214/aos/1013203451`.

C. Gourieroux and A. Monfort. *Statistics and Econometric Models*, volume 2 of *Themes in Modern Econometrics*. Cambridge University Press, 1995. doi: 10.1017/CBO9780511751950.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti. A Survey Of Methods For Explaining Black Box Models. *arXiv:1802.01933 [cs]*, Feb. 2018. URL `http://arxiv.org/abs/1802.01933`. arXiv: 1802.01933.

F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust statistics: the approach based on influence functions*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 2005. ISBN 9780471735779.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York, 2009. ISBN 9780387848587.

P. J. Huber. *Robust Statistics*. Wiley series in probability and mathematical statistics. Wiley, 1981. ISBN 9780471418054,0471418056.

P. W. Koh and P. Liang. Understanding Black-box Predictions via Influence Functions. *arXiv:1703.04730 [cs, stat]*, Mar. 2017. URL `http://arxiv.org/abs/1703.04730`. arXiv: 1703.04730.

S. Lundberg and S. Lee. A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874, 2017. URL `http://arxiv.org/abs/1705.07874`.

C. Molnar. *Interpretable machine learning*. Lulu. com, 2019.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org/`.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv:1602.04938 [cs, stat]*, Feb. 2016. URL `http://arxiv.org/abs/1602.04938`. arXiv: 1602.04938.

R. Verbelen, K. Antonio, and G. Claeskens. Unravelling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67 (5):1275–1304, 2018.

R. D. C. S. Weisberg. *Residuals and influence in regression*. Monographs on statistics and applied probability (Series). Chapman and Hall/CRC, 1st edition, 1982. ISBN 041224280X,9780412242809.

M. Wojnowicz, B. Cruz, X. Zhao, B. Wallace, M. Wolff, J. Luan, and C. Crable. "Influence Sketching": Finding Influential Samples In Large-Scale Regressions. *arXiv e-prints*, art. arXiv:1611.05923, Nov. 2016.

A Annexe : Data

Nom	Description
Coefficient Bonus/Malus	est le coefficient de Bonus/Malus de la police. Il est compris entre 0.5 et 3.5. Le meilleur est le plus bas. Il commence à 1 pour les jeunes conducteurs.
Niveau couverture police	est le niveau de couverture de la police : Mini, Median1, Median2, Maxi.
Durée de couverture	est l'ancienneté en années de la police.
Durée depuis dernier avenant	représente l'ancienneté de la police actuelle (ie) depuis le dernier changement.
Fréquence de paiement	est la fréquence de paiement : annuelle, biannuelle, trimestrielle, mensuelle
Pay As You Drive	indique si le client a souscrit à une offre Pay As You Drive.
Utilisation	décrit l'utilisation du véhicule par l'assuré.
Code INSEE	est le code INSEE identifiant la commune ou le département de l'assuré.
Sexe conducteur 1	est le sexe du premier conducteur.
Sexe conducteur 2	est le sexe du deuxième conducteur.
Ancienneté permis 1	est l'ancienneté du permis du premier conducteur en années.
Ancienneté permis 2	est l'ancienneté du permis du deuxième conducteur en années.
Age du véhicule	est l'âge en années du véhicule depuis sa sortie.
Cylindrée véhicule	représente la cylindrée du véhicule.
Type alimentation	est le type d'alimentation du moteur du véhicule. La variable a été encodée par "Target Encoding".
Fabriquant du véhicule	est le nom du constructeur du véhicule : Renault, Peugeot et Citroën.
Modèle du véhicule	est le modèle du véhicule.
Vitesse max véhicule	est la vitesse maximale du véhicule.
Type de véhicule	type de véhicule : tourisme, commercial.
Poids véhicule	est la masse du véhicule (en kg)

Figure 16 – Dictionnaire des données

B Annexe : Preuves

Preuve proposition 3.1. Montrons que $\mathcal{I}_{up,params}(z) = -H_{\hat{\theta}_n}^{-1}\nabla_\theta \mathcal{L}(z, \hat{\theta}_n)$.

Soient $Z \sim F$ où F est la fonction de répartition liée aux données et $Z_\epsilon \sim F_{\epsilon,z} = (1-\epsilon)F + \epsilon\delta_z$ sa version contaminée. On souhaite minimiser $\mathbb{E}[\mathcal{L}(Z,\theta)]$ c'est à dire $\frac{1}{n}\sum_{i=1}^n \mathcal{L}(z_i,\theta)$ qui réalise son minimum en $\hat{\theta}_n$. Sous l'hypothèse que le minimum est atteint où ∇_θ s'annule,

$$\nabla_\theta[\mathbb{E}(\mathcal{L}(Z,\hat{\theta}_n))] = \nabla_\theta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{L}(z_i,\hat{\theta}_n)\right) = 0,$$

ce qui se réécrit

$$\mathbb{E}(\nabla_\theta \mathcal{L}(Z,\hat{\theta}_n)) = \frac{1}{n}\sum_{i=1}^n \nabla_\theta\left(\mathcal{L}(z_i,\hat{\theta}_n)\right) = 0.$$

Notons $\hat{\theta}_{n,\epsilon,-z}$ le vecteur des paramètres optimaux pour la distribution contaminée. On cherche à calculer $\frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon}$. Pour la distribution contaminée l'équation précédente se réécrit

$$\mathbb{E}(\nabla_\theta \mathcal{L}(Z_\epsilon, \hat{\theta}_{n,\epsilon,-z})) = \frac{(1-\epsilon)}{n}\sum_{i=1}^{n} \nabla_\theta \mathcal{L}(z_i, \hat{\theta}_{n,\epsilon,-z}) + \epsilon \nabla_\theta \mathcal{L}(z, \hat{\theta}_{n,\epsilon,-z}) = 0.$$

En dérivant par rapport à ϵ on a

$$-\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \mathcal{L}(z_i, \hat{\theta}_{n,\epsilon,-z}) + \frac{(1-\epsilon)}{n} \times \frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon} \times \sum_{i=1}^{n} \nabla^2_\theta \mathcal{L}(z_i, \hat{\theta}_{n,\epsilon,-z}) + \nabla_\theta \mathcal{L}(z, \hat{\theta}_{n,\epsilon,-z}) + \epsilon \times \frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon} \times \nabla^2_\theta \mathcal{L}(z, \hat{\theta}_{n,\epsilon,-z}) = 0.$$

En faisant tendre ϵ vers 0, puis en se servant de la condition de nullité du gradient on obtient

$$\boxed{\mathcal{I}_{up,params}(z) = -H^{-1}_{\hat{\theta}_n} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)}$$

□

Preuve proposition 3.2. Il reste à montrer que $\hat{\theta}_{n,-z} - \hat{\theta}_n = -\frac{1}{n}\mathcal{I}_{up,params}(z) + o_p(\frac{1}{n})$

Soient, $Z_1, \ldots, Z_n$ des variables indépendantes identiquement distribuées. Montrons dans le cadre des échantillons finis que l'influence (asymptotique) du *n-ième* élément peut s'exprimer comme suit :

$$\hat{\theta}_{n,-z} - \hat{\theta}_n = -\frac{1}{n} H^{-1}_{\theta_0} \nabla_\theta \mathcal{L}(z, \theta_0) + o_p(\frac{1}{n})$$

On écrit le développement de Taylor pour la condition de premier ordre en θ_0.

$$0 = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \mathcal{L}(Z_i, \theta_0) + \frac{1}{n}\sum_{i=1}^{n} \nabla^2_\theta \mathcal{L}(Z_i, \theta_0)(\hat{\theta}_n - \theta_0) + \frac{1}{2}(\hat{\theta}_n - \theta_0)^\intercal \frac{1}{n}\sum_{i=1}^{n} \nabla^3_\theta \mathcal{L}(Z_i, \theta_0)(\hat{\theta}_n - \theta_0) + o(||(\hat{\theta}_n - \theta_0)||^2). \quad (2)$$

On peut écrire le même développement pour $\hat{\theta}_{n-1}$.

$$0 = \frac{1}{n-1}\sum_{i=1}^{n-1} \nabla_\theta \mathcal{L}(Z_i, \theta_0) + \frac{1}{n-1}\sum_{i=1}^{n-1} \nabla^2_\theta \mathcal{L}(Z_i, \theta_0)(\hat{\theta}_{n-1} - \theta_0) + \frac{1}{2}(\hat{\theta}_{n-1} - \theta_0)^\intercal \frac{1}{n-1}\sum_{i=1}^{n-1} \nabla^3_\theta \mathcal{L}(Z_i, \theta_0)(\hat{\theta}_{n-1} - \theta_0) + o(||(\hat{\theta}_{n-1} - \theta_0)||^2). \quad (3)$$

A partir des équations (2) et (3) on obtient l'égalité suivante :

$$
\begin{aligned}
-\frac{1}{n}\nabla_\theta \mathcal{L}(Z_n,\theta_0) = &-\frac{1}{n(n-1)}\sum_{i=1}^{n-1}\nabla_\theta \mathcal{L}(Z_i,\theta_0) + \frac{(\hat\theta_n-\hat\theta_{n-1})}{n-1}\sum_{i=1}^{n-1}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0)\\
&+\frac{(\hat\theta_n-\theta_0)}{n-1}\left[\frac{1}{n}\sum_{i=1}^{n}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0) - \frac{1}{n-1}\nabla^2_\theta \mathcal{L}(Z_n,\theta_0)\right]\\
&+\frac{1}{2}\left[(\hat\theta_n-\theta_0)^\intercal \frac{1}{n}\sum_{i=1}^{n}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0)(\hat\theta_n-\theta_0) - (\hat\theta_{n-1}-\theta_0)^\intercal \frac{1}{n-1}\sum_{i=1}^{n-1}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0)(\hat\theta_{n-1}-\theta_0)\right]\\
&+o(||(\hat\theta_n-\theta_0)||^2). \quad (4)
\end{aligned}
$$

Dans le développement (4), on pose $Z_n = z$. Le premier terme du membre de droite peut se réécrire $-\frac{1}{n}\left[\frac{1}{n-1}\sum_{i=1}^{n-1}\nabla_\theta \mathcal{L}(Z_i,\theta_0)\right]$. De cette manière il est facile de voir que le terme entre crochets multiplié par $\sqrt{n}$ tend en loi vers une loi normale multivariée centrée d'après le théorème *central limit*. Ce terme est donc borné en probabilité et est un $O_p(\frac{1}{n^{3/2}})$. Le deuxième terme du membre de droite peut s'écrire $(\hat\theta_n-\hat\theta_{n-1})\left[\frac{1}{n-1}\sum_{i=1}^{n-1}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0)\right]$ où le terme entre crochet tend vers la matrice H_{θ_0} en probabilité d'après la loi des grands nombres. Ceci grâce à l'hypothèse $\mathbb{E}[||\nabla^2_\theta \mathcal{L}(Z,\theta_0)||] < \infty$. Le troisième terme est un $o_p(\frac{1}{n})$ après multiplication par $\sqrt{n}$. Le quatrième et dernier terme est une somme de termes bornés en probabilité. En effet, nous avons fait l'hypothèse que $\frac{1}{n}\sum_{i=1}^{n}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0) = O_p(1)$

Il reste à rassembler les morceaux. On multiplie à gauche et à droite par $\sqrt{n}$. On obtient :

$$
\begin{aligned}
-\frac{1}{\sqrt{n}}\nabla_\theta \mathcal{L}(Z_n,\theta_0) = O_p(\frac{1}{n}) + (\hat\theta_n-\hat\theta_{n-1})\sqrt{n}(H_{\theta_0}+o_p(1)) + o_p(\frac{1}{n})\\
+\underbrace{\frac{\sqrt{n}}{2}\left[O_p(1)o_p(\frac{1}{n}) - O_p(1)o_p(\frac{1}{n})\right]}_{o_p(\frac{1}{\sqrt{n}})}. \quad (5)
\end{aligned}
$$

Soit encore,

$$
\left[-\frac{1}{n}\nabla_\theta \mathcal{L}(Z_n,\theta_0) + o_p(\frac{1}{n})\right]\left[H^{-1}_{\theta_0} + o_p(1)\right] = (\hat\theta_n-\hat\theta_{n-1})
$$

Donc,

$$
(\hat\theta_n-\hat\theta_{n-1}) = -\frac{1}{n}H^{-1}_{\theta_0}\nabla_\theta \mathcal{L}(Z_n,\theta_0) + \underbrace{H^{-1}_{\theta_0}o_p(\frac{1}{n})}_{o_p(\frac{1}{n})} - \underbrace{\frac{1}{n}\nabla_\theta \mathcal{L}(Z_n,\theta_0)o_p(1)}_{o_p(\frac{1}{n})} + \underbrace{o_p(\frac{1}{n})o_p(1)}_{o_p(\frac{1}{n})}
$$

$$
\begin{aligned}
\hat\theta_{n,-z} - \hat\theta_n &= -\frac{1}{n}H^{-1}_{\theta_0}\nabla_\theta \mathcal{L}(z,\theta_0) + o_p(\frac{1}{n})\\
&= -\frac{1}{n}H^{-1}_{\hat\theta_n}\nabla_\theta \mathcal{L}(z,\hat\theta_n) + o_p(\frac{1}{n})
\end{aligned}
$$

Finalement,

$$\boxed{\hat{\theta}_{n,-z} - \hat{\theta}_n = \frac{1}{n} H_{\hat{\theta}_n}^{-1} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) + o_p(\frac{1}{n})}$$

□

Preuve proposition 4.1. Montrons que $\mathcal{I}_{up,loss}(z, z_{test}) = -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n) H_{\hat{\theta}_n}^{-1} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)$

On applique la règle de la dérivation en chaîne à $\dfrac{d\mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z})}{d\epsilon}$. Il vient

$$\frac{d\mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z})}{d\epsilon} = \nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z}) \frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon}.$$

En évaluant en $\epsilon = 0$, on obtient

$$\begin{aligned}
\left.\frac{d\mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z})}{d\epsilon}\right|_{\epsilon=0} &= \left.\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_{n,\epsilon,-z})\right|_{\epsilon=0} \underbrace{\left.\frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon}\right|_{\epsilon=0}}_{=-H_{\hat{\theta}_n}^{-1} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)} \\
&= -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n) H_{\hat{\theta}_n}^{-1} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n).
\end{aligned}$$

Finalement,

$$\boxed{\mathcal{I}_{up,loss}(z, z_{test}) = -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n) H_{\hat{\theta}_n}^{-1} \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)}$$

□

Preuve proposition 4.2. Soient $z_1 \dots z_n$ un ensemble d'observations telles que $z_i = (x_i, y_i)$ avec $x_i \in \mathbb{R}^p$ et $y_i \in \{0, 1\}$. On note

$$\mathbb{P}(Y = 1 | X = x_i) = p(x_i, \theta) = \frac{1}{1 + e^{-\theta^\intercal x_i}}.$$

Rappelons que la fonction de perte dans le cas de la régression logistique binaire est la log-vraisemblance (log-loss) régularisée avec un paramètre ($\alpha > 0$). C'est à dire,

$$\mathcal{L}(z_i, \theta) = -\sum_{i=1}^{n} \left[y_i \log(p(x_i, \theta)) + (1 - y_i) \log(1 - p(x_i, \theta)) + \alpha ||\theta||_2^2 \right].$$

Il est facile de montrer que

$$H_{\hat{\theta}_n} = \sum_{i=1}^{n} p(x_i, \theta)(1 - p(x_i, \theta)) \underbrace{\begin{pmatrix} x_{1,1}^2 & \cdots & x_{1,1}x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{1,1}x_{1,n} & \cdots & x_{n,n}^2 \end{pmatrix}}_{=x_i x_i^\intercal} + 2\alpha I d_p.$$

voir [Hastie et al., 2009] page 120. Montrons que $H_{\hat{\theta}_n}$ est positive.

Soit $a \in \mathbb{R}^p$, montrons que $a^\intercal H_{\hat{\theta}_n} a \geq 0$

$$a^\intercal H_{\hat{\theta}_n} a = \sum_{i=1}^{n} p(x_i,\theta)(1-p(x_i,\theta))a^\intercal x_i x_i^\intercal a + 2\alpha||a||_2^2.$$

Soit $i \in \{1,\dots,n\}$ on pose $b_i = x_i^\intercal a$ alors

$$a^\intercal H_{\hat{\theta}_n} a = \sum_{i=1}^{n} \underbrace{p(x_i,\theta)}_{>0}\underbrace{(1-p(x_i,\theta))}_{>0}\underbrace{b_i^\intercal b_i}_{||b||_2^2} + +2\underbrace{\alpha}_{>0}||a||_2^2.$$

donc d'une part

$$\boxed{a^\intercal H_{\hat{\theta}_n} a \geq 0}$$

et d'autre part, $a^\intercal H_{\hat{\theta}_n} a = 0 \implies a = 0$. En effet, on a une somme de deux termes positifs. Pour qu'elle soit nulle il est nécessaire que les deux s'annulent simultanément. Ceci s'obtient uniquement lorsque $a = 0$ (grâce à la norme 2).

Remarque. *On voit grâce à la régularisation que l'on peut montrer le caractère défini. Sans ce terme, nous ne pourrions démontrer que le fait que $a^\intercal H_{\hat{\theta}_n} a \geq 0$. En pratique pourtant, même sans ce paramètre de régularisation, on est la plupart du temps dans le cas défini.*

□

Preuve proposition 5.1. Montrons que $\mathcal{I}_{pert,params}(z, z_\delta) = \mathcal{I}_{up,params}(z) - \mathcal{I}_{up,params}(z_\delta)$

Soient $Z \sim F$ où F est la fonction de répartition liée aux données et $Z_{\epsilon,\delta} \sim F_{\epsilon,z_\delta,z} = F + \epsilon\delta_z - \epsilon\delta_{z_\delta}$ sa version contaminée. On reprend la même méthodologie que pour la proposition 3.1. Nécessairement,

$$\nabla_\theta[\mathbb{E}(\mathcal{L}(Z,\hat{\theta}_n)] = \nabla_\theta\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(z_i,\hat{\theta}_n)\right) = 0.$$

Ce qui se réécrit

$$\mathbb{E}(\nabla_\theta(\mathcal{L}(Z,\hat{\theta}_n))) = \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta\left(\mathcal{L}(z_i,\hat{\theta}_n)\right) = 0.$$

Notons $\hat{\theta}_{n,\epsilon,z_\delta,-z}$ le vecteur des paramètres optimaux pour la distribution contaminée. On cherche à calculer $\frac{d\hat{\theta}_{n,\epsilon,z_\delta,-z}}{d\epsilon}$. Pour la distribution contaminée l'équation précédente se réécrit

$$\mathbb{E}(\nabla_\theta(\mathcal{L}(Z_{\epsilon,\delta},\hat{\theta}_{n,\epsilon,z_\delta,-z}))) = \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta\mathcal{L}(z_i,\hat{\theta}_{n,\epsilon,z_\delta,-z}) + \epsilon(\nabla_\theta\mathcal{L}(z_\delta,\hat{\theta}_{n,\epsilon,z_\delta,-z}) - \nabla_\theta\mathcal{L}(z,\hat{\theta}_{n,\epsilon,z_\delta,-z})) = 0.$$

En dérivant par rapport à ϵ

$$\frac{1}{n}\times\frac{d\hat{\theta}_{n,\epsilon,z_\delta,-z}}{d\epsilon}\times\sum_{i=1}^{n}\nabla_\theta^2\mathcal{L}(z_i,\hat{\theta}_{n,\epsilon,z_\delta,-z}) + A = 0,$$

avec

$$A = (\nabla_\theta \mathcal{L}(z_\delta, \hat{\theta}_{n,\epsilon,z_\delta,-z}) - \nabla_\theta \mathcal{L}(z, \hat{\theta}_{n,\epsilon,z_\delta,-z})) + \epsilon \times \frac{d\hat{\theta}_{n,\epsilon,z_\delta,-z}}{d\epsilon} \times (\nabla^2_\theta \mathcal{L}(z_\delta, \hat{\theta}_{n,\epsilon,z_\delta,-z}) - \nabla^2_\theta \mathcal{L}(z, \hat{\theta}_{n,\epsilon,z_\delta,-z})).$$

En évaluant en $\epsilon = 0$, puis en se servant de la condition de nullité du gradient et en simplifiant on obtient

$$\mathcal{I}_{pert,params}(z, z_\delta) = -H^{-1}_{\hat{\theta}_n}(\nabla_\theta \mathcal{L}(z, \hat{\theta}_n) - \nabla_\theta \mathcal{L}(z_\delta, \hat{\theta}_n)).$$

Ce qui correspond bien à

$$\boxed{\mathcal{I}_{pert,params}(z, z_\delta) = \mathcal{I}_{up,params}(z) - \mathcal{I}_{up,params}(z_\delta)}$$

□

Preuve proposition 5.2. Formule fermées pour $\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n$

En réutilisant le résultat de la Proposition 1.2, il vient naturellement que $\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n = -\frac{1}{n}(\mathcal{I}_{up,params}(z_\delta) - \mathcal{I}_{up,params}(z)) + o_p(\frac{1}{n})$.

Pour montrer la seconde partie du résultat, nous allons utiliser un développement de Taylor à l'ordre 1. Soit $f : \mathbb{R}^p \to \mathbb{R}$ une fonction deux fois différentiables en un point $a \in \mathbb{R}^p$, rappelons que la formule de Taylor nous permet d'écrire

$$f(a + h) = f(a) + \nabla_x f(a)h + o(||h||).$$

En posant $f = \nabla_\theta \mathcal{L}$, $a = z$ et $h = \delta$, on obtient

$$\nabla_\theta \mathcal{L}(z_\delta, \theta_0) = \nabla_\theta \mathcal{L}(z, \theta_0) + \nabla_x \nabla_\theta \mathcal{L}(z, \theta_0)\delta + o(||\delta||).$$

Soit encore

$$\nabla_\theta \mathcal{L}(z_\delta, \theta_0) - \nabla_\theta \mathcal{L}(z, \theta_0) = \nabla_x \nabla_\theta \mathcal{L}(z, \theta_0)\delta + o(||\delta||).$$

En injectant cette nouvelle approximation dans la formule précédente, on obtient

$$\begin{aligned}\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n &= \frac{1}{n} H^{-1}_{\theta_0} \left[\nabla_x \nabla_\theta \mathcal{L}(z, \theta_0)\right] \delta + o_p(\frac{1}{n}) + o(||\delta||) \\ &= \frac{1}{n} H^{-1}_{\hat{\theta}_n} \left[\nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)\right] \delta + o_p(\frac{1}{n}) + o(||\delta||)\end{aligned}$$

Finalement,

$$\boxed{\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n = \frac{1}{n} H^{-1}_{\hat{\theta}_n} \left[\nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n)\right] \delta + o_p(\frac{1}{n}) + o(||\delta||)}$$

□

Preuve proposition 5.3. Formule fermée pour $\mathcal{I}_{pert,loss}(z, z_{test})^\intercal$

Supposons que les variables explicatives sont continues ie $x \in \mathcal{X} \in \mathbb{R}^p$ et $||\delta|| \to 0$. Si $\mathcal{L}$ est différentiable par rapport à x et θ alors d'après l'approximation développée dans la preuve précédente

$$\nabla_\theta \mathcal{L}(z_\delta, \theta_0) - \nabla_\theta \mathcal{L}(z, \theta_0) = \nabla_x \nabla_\theta \mathcal{L}(z, \theta_0)\delta + o(||\delta||).$$

En injectant ceci dans la Proposition 1.4, on obtient

$$\left.\frac{d\hat{\theta}_{n,\epsilon,z_\delta,-z}}{d\epsilon}\right|_{\epsilon=0} = -H_{\theta_0}^{-1} \nabla_x \nabla_\theta \mathcal{L}(z, \theta_0)\delta + o_p(\frac{1}{n}) + o(||\delta||).$$

Par définition,

$$\mathcal{I}_{pert,loss}(z, z_{test})^\intercal = \left.\nabla_\delta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal\right|_{\delta=0}.$$

On applique la règle de la dérivation en chaîne à $\left.\nabla_\delta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal\right|_{\delta=0}$. Il vient

$$\nabla_\delta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal = \nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal \frac{d\hat{\theta}_{n,z_\delta,-z}}{d\delta}.$$

Or, d'après les approximations établies plus haut on a

$$\frac{d\hat{\theta}_{n,z_\delta,-z}}{d\delta} = -H_{\hat{\theta}_n}^{-1} \nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) + o_p(\frac{1}{n}) + o(1).$$

En évaluant en $\delta = 0$ il vient

$$\begin{aligned}
\left.\nabla_\delta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal\right|_{\delta=0} &= \left.\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})\right|_{\delta=0} \underbrace{\left.\frac{d\hat{\theta}_{n,z_\delta,-z}}{d\delta}\right|_{\delta=0}}_{=-H_{\hat{\theta}_n}^{-1} \nabla_x \nabla_\theta \mathcal{L}(z,\hat{\theta}_n) + o_p(\frac{1}{n}) + o(1)} \\
&= -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n) H_{\hat{\theta}_n}^{-1} \nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) + o_p(\frac{1}{n}) + o(1).
\end{aligned}$$

Ce qui permet de conclure que

$$\boxed{\mathcal{I}_{pert,loss}(z, z_{test})^\intercal = -\nabla_\theta \mathcal{L}(z_{test}, \hat{\theta}_n)^\intercal H_{\hat{\theta}_n}^{-1} \nabla_x \nabla_\theta \mathcal{L}(z, \hat{\theta}_n) + o_p(\frac{1}{n}) + o(1)}$$

□

C Annexes : Figures

Table 5 – Table des coefficients estimés

Point retiré	Influence au point test
1	4.77
2	-0.47
3	13.35
4	0.19
5	0.23