Some explainability methods for statistical learning models in actuarial science

Arthur Maillart

N° d'ordre NNT : 2021LYSE1105

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de :

l'Université Claude Bernard Lyon 1

École Doctorale ED 486

Sciences Économiques et de Gestion

Spécialité de doctorat : Sciences de Gestion

Soutenue publiquement le 29 juin 2021, par :

Arthur Maillart

Quelques Méthodes d'Explicabilité pour les Modèles d'Apprentissage Statistique en Actuariat

---

Devant le jury composé de :

Katrien Antonio Professeure, KU Leuven, Louvain	**Rapporteure**
Arthur Charpentier Professeur, Université du Québec, Montréal	**Examinateur**
Caroline Hillairet Professeure, ENSAE, Paris	**Examinatrice**
Olivier Lopez Professeur, Sorbonne Université, Paris	**Rapporteur**
Frédéric Planchet Professeur, Université Lyon 1, Lyon	**Président**
Christian Y. Robert Professeur, ENSAE, Paris	**Directeur de thèse**
Arnaud Cohen Président, Forsides France, Paris	**Invité**

Université Claude Bernard Lyon 1

Arthur Maillart

Quelques Méthodes d'Explicabilité pour les Modèles d'Apprentissage Statistique en Actuariat

Sciences de Gestion, 29 juin 2021

Directeur de thèse : Christian Y. Robert

Université Claude Bernard Lyon 1

Laboratoire de Sciences Financières et d'Assurance

École Doctorale ED 486

50 Avenue Tony Garnier

69007 Lyon

Forsides France

52 Rue de la Victoire

75009 Paris

Résumé

Les méthodes d'apprentissage statistique se sont implantées dans de nombreux secteurs de l'industrie. Ces algorithmes génèrent des modèles complexes qui peuvent avoir des conséquences concrètes sur notre vie quotidienne. Par exemple, les modèles produits par ces algorithmes peuvent influer sur l'acceptation d'un crédit immobilier ou sur un tarif d'assurance. Il est donc essentiel que nous puissions comprendre la chaîne de décisions logiques de ces modèles ou du moins avoir des éléments pour les interpréter. En particulier, pour l'actuaire, expliquer ces modèles complexes permet de découvrir de nouvelles connaissances scientifiques. Cela permet aussi d'expliquer à un public non-expert les décisions d'un modèle sophistiqué. Ainsi, dans cette thèse, nous examinons plusieurs méthodes qui fournissent des explications de modèles pour des publics cibles expérimentés ou non. Nous montrons à travers trois problématiques indépendantes l'intérêt des méthodes d'explicabilité pour l'apprentissage statistique en science actuarielle.

Nous commençons ce manuscrit par un chapitre introductif. Bien que le thème de l'interprétabilité des modèles statistiques ait fortement stimulé la recherche ces dernières années, ce sujet reste pour le moment assez peu développé par la communauté actuarielle. Ainsi, nous proposons une brève revue historique permettant de retracer les étapes importantes qui structurent aujourd'hui la recherche sur ce point. Nous mettons de cette manière en évidence que les différentes communautés qui ont contribué à l'apprentissage statistique ont des besoins propres en matière d'interprétabilité. Par exemple, pour un modèle de vision assistée par ordinateur, il n'est souvent pas possible d'expliquer simplement la chaîne de décisions qui entraîne une prédiction pour une image donnée. Par conséquent, les experts du domaine proposent des explications sous la forme d'images où les zones responsables d'une prédiction sont colorées, ou encore en affichant les filtres ajustés pour le modèle. Avec ces mécanismes, pour chaque image une explication est générée. L'explication n'est donc valable que pour cette image, ce qui ne permet pas d'avoir une vision globale du modèle. Ce type d'explication ne peut dès lors pas convenir à toutes les communautés. En particulier, les actuaires emploient rarement ce genre de méthodes. Ces statisticiens, experts du risque, préféreront souvent un modèle interprétable par nature. Un modèle interprétable par nature est, par exemple, un modèle linéaire généralisé ou un arbre de régression. Dans le cas où le modèle utilisé n'est pas interprétable par nature, comme pour un ensemble d'arbres, il est courant d'utiliser des graphiques présentant l'importance des variables ou d'autres quantités d'intérêt. Nous montrons ainsi qu'il n'y a pas de consensus et que la notion d'interprétabilité rassemble de nombreuses approches et méthodes. Néanmoins, il est possible d'harmoniser les travaux des différentes communautés en établissant une taxonomie des méthodes pour expliquer des modèles complexes, nommés boîtes

noires. Parmi les classifications proposées, nous en retenons une qui est composée de quatre groupes de méthodes et qui tient compte du type d'explication proposé par une méthode d'explicabilité donnée. Conséquemment, elle tient compte de l'audience cible de l'explication. Pour chaque classe de méthode d'explicabilité, nous décrivons le public visé par l'explication fournie, et les limites de ces méthodes. Nous terminons ce chapitre en montrant qu'il est très difficile de quantifier une notion que l'on peine à définir comme l'interprétabilité. Bien qu'il n'y ait actuellement pas de consensus sur la manière de mesurer l'interprétabilité d'un modèle par rapport à un autre, il est tout de même possible d'évaluer la fidélité et la stabilité de l'explication par rapport au modèle à expliquer.

Les deux chapitres suivants se concentrent sur deux méthodes qui extraient une explication depuis un ensemble d'arbres appelé boîte noire. Dans les deux cas, cette explication prend la forme d'un arbre de régression qui reproduit fidèlement les prédictions de l'ensemble d'arbres tout en étant plus simple. Ainsi, nous conservons la performance prédictive du modèle boîte noire, mais nous pouvons expliquer ses prédictions grâce à notre arbre de régression. Nous illustrons ces techniques sur deux problématiques différentes.

Dans le chapitre 2, nous construisons un modèle de fréquence sinistre avec un Poisson random forest. C'est une approche data-driven que nous implémentons sur des données télématiques. C'est ce modèle de fréquence sinistre qui est notre boîte noire et dont nous voulons extraire une explication. Or, un ensemble d'arbres crée une partition de l'espace des covariables en rectangles de dimension d, où d est la dimension de l'espace des variables explicatives. Pour chacun de ces rectangles, une unique valeur est prédite. Par conséquent, une approche assez intuitive consiste à agréger ces trop nombreux rectangles pour en fournir un ensemble parcimonieux. Cet ensemble de rectangles devient alors un modèle de substitution permettant d'expliquer la boîte noire. Pour effectuer cette agrégation, il existe une méthode nommée DefragTrees. Cette méthode consiste à exprimer la distribution de couples (rectangle, prédiction) en fonction de paramètres à estimer. Grâce à cela, il est possible de formuler le problème d'agrégation comme un problème de maximisation de la vraisemblance dont nous pouvons obtenir une solution numérique avec l'algorithme espérance-maximisation. Une fois les paramètres optimaux obtenus, nous avons à disposition un ensemble de rectangles et donc de règles qui expliquent la boîte noire. Toutefois, si aucune contrainte supplémentaire n'est imposée, un individu peut être caractérisé par deux règles différentes, car deux règles peuvent se chevaucher. L'explication est dans ce cas plus compliquée à comprendre. C'est pourquoi nous modifions cette méthode pour qu'elle produise un arbre de régression plutôt que des règles. Cela facilite l'interprétation. Grâce au modèle de substitution que nous extrayons, nous pouvons ensuite expliquer les prédictions du modèle de fréquence sinistre en fonction des variables télématiques. De cette manière, nous pouvons découvrir de nouvelles relations intéressantes du point de vue actuariel. Dans notre étude de cas, nous mettons en évidence que les jeunes assurés qui ne conduisent pas régulièrement dans l'année ont une fréquence sinistre significativement plus élevée que les autres.

Dans le chapitre 3, nous voulons estimer l'indice de queue qui mesure l'importance d'un évènement extrême. Pour cela, nous supposons que ce paramètre prend un nombre fini de valeurs sur une partition de l'espace des variables explicatives. Nous proposons une stratégie en deux étapes. La

première étape consiste à ajuster un modèle gamma gradient boosting, notre boîte noire, pour obtenir une estimation de l'indice de queue en fonction des variables explicatives. Puis, nous agrégeons les nombreux rectangles générés en utilisant une méthode de classification ascendante hiérarchique avec une contrainte spatiale. La contrainte spatiale permet de tenir compte de la proximité entre les rectangles dans l'espace des covariables. Ainsi, nous obtenons notre partition de l'espace en un nombre fini d'indices de queues en fonction des variables explicatives. Toutefois, ce modèle n'est pas interprétable puisque la forme des régions créées par l'agrégation est trop complexe pour être compréhensible par un humain. Notre seconde étape consiste donc à extraire un modèle de substitution pour interpréter les prédictions de la boîte noire. Nous adoptons une stratégie similaire à celle du chapitre précédent c'est-à-dire que nous voulons agréger des rectangles contigus pour en former de plus larges. Pour ce faire, nous créons une méthode originale qui permet de produire une suite d'arbres de régression emboîtés qu'il suffit d'élaguer pour obtenir un modèle de substitution et donc une explication de la profondeur souhaitée. Nous commençons par déterminer la partition générée par le croisement de tous les arbres de l'ensemble. Puisque nous pouvons connaître la prédiction de la boîte noire pour chacun de ces rectangles, nous avons une connaissance complète des prédictions de la boîte noire sur tout l'espace des covariables. Pour agréger ces rectangles, nous modifions un algorithme d'apprentissage d'arbre de régression en restreignant ses seuils de coupe possibles à ceux déjà effectués par l'ensemble d'arbres. En entrées de ce nouvel algorithme, nous fournissons un point par rectangle de la partition et la prédiction de la boîte noire associée. De cette manière, l'algorithme génère une suite d'arbres de plus en plus profonds jusqu'à atteindre l'arbre maximal. Cet arbre réplique parfaitement la partition initiale, mais est trop profond. Il suffit de l'élaguer pour trouver un meilleur compromis entre la fidélité par rapport à la boîte noire et la profondeur de l'arbre qui est notre modèle de substitution. Grâce à cette méthode, nous expliquons les prédictions du gamma gradient boosting ajusté sur un jeu de données qui contient les coûts estimés des ouragans aux États-Unis depuis les années 50. Cela nous permet de comprendre les caractéristiques qui génèrent un indice de queue faible ou élevé et donc de comprendre les caractéristiques d'ouragans qui génèrent des sinistres plus ou moins graves. Dans ce cas particulièrement simple, l'explication est en ligne avec l'intuition. Cela confirme l'intérêt de l'approche pour extraire efficacement des connaissances depuis ces données.

Notre dernier chapitre se distingue des deux précédents puisque nous nous intéressons cette fois-ci à un type de modèle différent et donc à une représentation interne différente. En effet, nous étudions ici des modèles paramétriques d'apprentissage statistique, dont les paramètres sont fixés a priori puis sont estimés par la suite. Parmi ces modèles, nous trouvons les modèles linéaires généralisés, et les réseaux de neurones. Ici, nous considérons que notre boîte noire est un réseau de neurones. Contrairement aux deux premiers chapitres, nous n'avons plus accès à une représentation interne qui maille l'espace des variables explicatives et que nous pouvons exploiter pour donner une explication globale des prédictions de la boîte noire. Dès lors, expliquer devient plus compliqué. C'est pourquoi nous changeons d'objectif par rapport aux deux chapitres précédents et ne cherchons que des explications valables au voisinage d'une prédiction à expliquer. Nous nous plaçons dans un cadre de classification binaire et proposons une méthode pour expliquer une prédiction pour une observation donnée de l'espace. La stratégie que nous suggérons ici est d'identifier des points influents pour une

prédiction. Ces points, peu nombreux, permettent de localiser la frontière de décision au voisinage de la prédiction à expliquer. Ensuite, nous développons un algorithme qui permet de construire un hyperplan tangent à la frontière de décision au voisinage de la prédiction à expliquer. Cet hyperplan est un modèle de substitution local qui permet d'expliquer une prédiction à la fois. Nous présentons donc des explications sous la forme de modèles linéaires. Nous perdons ainsi la vision globale du modèle, mais nous pouvons toutefois employer cette approche pour comprendre des phénomènes sur une sous-partie de l'espace. Nous illustrons cette stratégie avec des données d'assurance automobile. Nous créons un problème de prévention dans lequel notre objectif est de trouver les profils les plus susceptibles de faire une déclaration de sinistre à horizon 1 an. En appliquant notre méthodologie, nous sommes capables d'extraire des explications fidèles aux voisinages des prédictions que nous voulons expliquer. Toutefois, dans ce cadre, nous ne pouvons pas extraire de nouvelles connaissances scientifiques puisque nous n'avons plus de vision globale.

Mots-clés : Apprentissage statistique; Apprentissage statistique intelligible; Explicabilité; Modélisation de la fréquence sinistre; Indice de queue

Abstract

Machine learning methods have become established in many industry sectors. The related algorithms generate complex models that can have concrete consequences on our daily lives. For example, models produced by these algorithms can influence the acceptance of a loan or an insurance premium. Therefore, it is essential that we can understand the logical decision chain of such a model or at least have ways of interpreting it. In particular, for actuaries, explaining the relevant complex models can lead to discoveries of new scientific knowledge. Interpretability also makes it possible to explain the decisions of a sophisticated model to a non-expert audience. Thus, in this thesis we examine several methods that provide model explanations for both experienced and non-expert target audiences. We also illustrate, considering three independent aspects, the appeal of explainability methods for machine learning in actuarial science.

We begin this manuscript with an introductory chapter. Although there has been a strong stimulus to research the topic of machine learning model interpretability in recent years, this subject remains relatively undeveloped by the actuarial community at the moment. Thus, we perform a brief historical review to trace the important steps that structure research on this topic today. In doing so, we highlight that the different communities that have contributed to machine learning have their own needs in terms of interpretability. For example, for a computer vision model, it is often impossible to simply explain the decision chain that leads to a prediction for a given image. Therefore, experts in the field offer explanations that are images where the areas responsible for a prediction are colored, or display the filters fitted for the model. With these mechanisms, an explanation is generated for each image. The explanation is therefore only valid for that image, which prevents having a global vision of the model. Hence, this type of explanation cannot be appropriate for all

communities. In particular, actuaries rarely use such methods. As statisticians and experts of risk assessment, they often prefer inherently interpretable models. An inherently interpretable model is, for example, a generalized linear model or a regression tree. In cases where the model used is not inherently interpretable, such as a set of trees, it is common to use graphs showing the variables' importance or other quantities of interest. We thus show that there is no consensus and that the notion of interpretability brings together many approaches and methods. Nevertheless, it is possible to harmonize the approaches of different communities by establishing a taxonomy of methods used to explain complex models referred to as black-boxes. Among the proposed classifications, we retain that composed of four groups of methods. This taxonomy takes into account the type of explanation proposed by a given explainability method. Consequently, it considers the target audience of the explanation. For each class of explanation methods, we describe the audience for the explanation provided, and the limitations of the respective methods. We conclude this chapter by showing that it is very difficult to quantify the notion of interpretability because it is difficult to define. Although there is currently no consensus on how to measure the interpretability of one model compared to another, it is still possible to assess the fidelity and stability of the explanation for a model to be explained.

The next two chapters focus on two methods that extract an explanation from a tree ensemble. Such a set of trees is referred to as a black-box. In both cases, this explanation takes the form of a regression tree that faithfully reproduces the predictions of the tree ensemble while being simpler. Thus, we preserve the predictive performance of the black-box model while being able to explain its predictions using our regression tree. We illustrate these techniques on two different use cases.

In chapter 2, we construct a claim frequency model with a Poisson random forest. It is a data-driven approach that we implement on telematic data. The claim frequency model is our black-box, and we want to extract an explanation from it. A tree ensemble creates a partition of the covariate space into rectangles of dimension d, which is also the dimension of the explanatory variables' space. For each of such rectangles, a unique value is predicted. Therefore, an intuitive approach is to aggregate such numerous rectangles to provide a parsimonious set. This set of rectangles then becomes a surrogate model used to explain the black-box. To perform this aggregation, a method named DefragTrees can be used. It consists of expressing the distribution of pairs (rectangle, prediction) as a function of parameters to be estimated. As a result, it is possible to formulate the aggregation problem as a maximum likelihood problem from which we can obtain a numerical solution with the expectation-maximization algorithm. Once the optimal parameters have been obtained, we have at our disposal a set of rectangles and therefore rules that explain the black-box. However, if no additional constraint is imposed, an individual can be characterized by two different rules because two rules may overlap. The explanation in this case is more complicated to understand. This is why we modify this method so that it produces a regression tree rather than rules, which eases the interpretation. Using the substitution model we extract, we can then explain the predictions of the claim frequency model as a function of telematic variables. In this way, we can discover new actuarially interesting relationships. In our case study, we show that young policyholders who do not drive regularly during the year have a significantly higher claim frequency than do others.

In chapter 3, we want to estimate the tail index that measures the importance of an extreme event. To do so, we assume that this parameter takes a finite number of values on a partition of the explanatory variables' space. To perform estimation under this assumption, we propose a two-step strategy. The first step consists of fitting a gamma gradient-boosting model, our black-box, to obtain an estimate of the tail index as a function of explanatory variables. Then, we aggregate the numerous rectangles using an ascending hierarchical classification method with a spatial constraint. The latter allows us to take into account the proximity between the rectangles in the covariates' space. Thus, we obtain our partition of the space into a finite number of tail indexes as a function of explanatory variables. However, this model is not interpretable since the shape of regions created by the aggregation is too complex to be understood by a human. Therefore, our second step is to extract a surrogate model to interpret the variations of the black-box. We adopt a strategy similar to that of the previous chapter; i.e., we want to aggregate contiguous rectangles to form larger ones. To this end, we create an original method that produces a sequence of nested regression trees. To obtain a surrogate model and thus an explanation of the desired depth, the sequence of trees simply needs to be pruned. We start by determining the partition generated by crossing all the trees in the set. Since we can determine the black-box prediction for each of these rectangles, we have a complete knowledge of black-box predictions over the entire covariates' space. To aggregate these rectangles, we modify a regression tree learning algorithm by restricting its possible split points to those already made by the tree ensemble. We provide as inputs to this new algorithm one point per rectangle of the partition and the prediction of the associated black-box. As a result, the algorithm generates a sequence of trees that becomes increasingly deeper until the maximum tree depth is reached. This tree perfectly replicates the initial partition but is too deep. We merely have to prune it to attain a better trade-off between the fidelity with respect to the black-box and the depth of our surrogate model. Using this method, we explain the predictions of the gamma gradient boosting model fit on a dataset that contains the estimated costs of hurricanes in the United States over the last 50 years. This allows us to comprehend the characteristics that generate a high or low tail index. Thus, we can understand the characteristics of hurricanes that generate more or less severe claims. In this particularly simple case, the explanation is consistent with intuition. This confirms the appeal of the approach to effectively extracting knowledge from such data.

Our last chapter differs from the two previous chapters because in it we are interested in a different type of model and therefore a different internal representation. Indeed, we study parametric machine learning models with parameters that are set a priori and then estimated afterwards. Among these models, we find generalized linear models and neural networks. Here, we presume that our black-box is a neural network. Contrary to the first two chapters, we no longer have access to an internal representation that meshes the space of explanatory variables. Hence, we cannot exploit the internal representation to provide a global explanation of the black-box predictions. Consequently, explaining becomes more complicated. This is why we change our objective compared to the two previous chapters and only look for explanations that are valid close to a prediction to be explained. We study a binary classification framework and propose a method to explain a prediction for a given observation of the space. The strategy we suggest here is to identify influential points for a prediction. These points, which are few in number, make it possible to locate the decision boundary near the

prediction to be explained. Then, we develop an algorithm that allows us to construct a hyperplane tangent to the decision boundary in the vicinity of the prediction to be explained. This hyperplane is a local surrogate model that allows explaining one prediction at a time. Therefore, we present explanations in the form of linear models. Thus, we lose the global vision of the model but can, however, use this approach to understand phenomena on a subset of the space. We illustrate this strategy with car insurance data. We create a prevention problem in which our objective is to find the profiles most likely to file a claim within 1 year. By applying our methodology, we are able to extract locally faithful explanations for the predictions we want to explain. However, within this framework, we cannot extract new scientific knowledge since we no longer have a global vision.

Keywords : Machine Learning; Interpretable Machine Learning; Explainability; Claim Frequency Modeling; Tail-Index

Acknowledgement

Tout d'abord, je tiens à remercier très sincèrement Christian Robert, mon directeur de thèse. Christian, tu m'as accompagné durant ces années de doctorat et ton soutien sans faille m'a permis de conclure aujourd'hui ce travail de longue haleine. Merci pour tes précieux conseils et nos nombreux échanges. Ils ont contribué à nourrir ma réflexion et à faire évoluer mon travail de thèse.

Un grand merci à Katrien Antonio et Olivier Lopez d'avoir accepté de rapporter ma thèse et d'y avoir consacré du temps. J'adresse mes sincères remerciements à Arthur Charpentier, Caroline Hillairet et Frédéric Planchet qui ont accepté de faire partie du jury.

Je remercie l'ensemble des membres du laboratoire SAF pour leur accueil. J'ai beaucoup apprécié mes passages à Lyon grâce à vous. Merci en particulier à Pierre, Sarah, Steve, Romain, Morgane, Maximilien (qui me doit une bière soit dit en passant), Pierrick, Claire et Pierre Olivier (alias P-O) pour ces précieux moments.

Je remercie également Arnaud Cohen, directeur de Forsides. Merci, Arnaud, de m'avoir ouvert des portes essentielles au lancement de mon projet de thèse. Aussi, merci d'avoir veillé à ce que j'accomplisse mon travail de recherche dans de bonnes conditions. Comme tu le sais, je garde d'excellents souvenirs de mes années passées à Forsides, et en particulier des personnes que j'y ai rencontrées.

Enfin, je souhaite remercier ma famille sans qui rien n'aurait été possible. Un grand merci à mes parents, Patrick et Enza, à mon frère Théo et à Mélodie qui partage ma vie et qui m'a donc supporté pendant ces années.

Contents

1 Introduction

> *There is no true interpretation of anything; interpretation is a vehicle in the service of human comprehension. The value of interpretation is in enabling others to fruitfully think about an idea.*
>
> **— Andreas Buja**

1.1 Historical context

What we call machine learning today includes, among other things, computer science and statistics. Thus, several communities of researchers have helped develop models that learn from data. According to Breiman (2001b), each of these communities is trying to model a complex phenomenon with a different strategy. Statisticians often use data models, i.e., parametric equations to fit a model to the data. In contrast, computer scientists do not make assumptions about the underlying process generating the data. To be concise, these two cultures have divergent goals; statisticians want to extract information from their models, while computer scientists are driven by performance of prediction. Therefore, the simplest models, such as linear models or trees, have been preferred in the statistical community. This is still so in, for example, the actuarial science.

However, recent successes of more sophisticated methods are of interest to statisticians and actuaries. We offer a brief history of methods developed by various communities. We will observe that interpretability is intrinsically linked to the development of machine learning models. The early methods of explainability appeared a few years after the first opaque models, which outperformed the interpretable models of the time. Moreover, this historical review shows that the adoption of machine learning in actuarial sciences has been slower than in other fields. This was partly due to the lack of interpretability. Several approaches attempting to integrate machine learning models into actuarial topics have been developed in recent years. However, as we shall observe, interpretability issues remain relatively little addressed.

In this introduction and in the manuscript, we limit our study to generalized linear models, tree ensemble models and neural networks produced by supervised learning algorithms. These models are the most commonly used by the machine learning community and by actuaries. In this part

dedicated to the historical context, we systematically present a brief history of each method. Then, we show how the methods are used in an actuarial framework.

Through this short historical overview, we aim to provide the reader an insight into interpretable machine learning and various approaches that have been developed through years. Considering the very large number of articles and the multiplicity of strategies, this overview cannot be exhaustive.

Parametric methods

Let us begin with parametric models such as linear models, neural networks and support vector machines. The first statistical learning method was described by Gauss and Legendre in the early 19th century for astronomy applications. The scholars' work on the least squares method has been seminal in the theory of linear models. Thereafter, the development of linear models has spanned nearly 200 years. Among the proposed adaptations of the linear model are the logit model introduced by Berkson (1944) in biostatistics, the probit model presented by Bliss (1934) and the ridge regression, a penalized regression proposed by Hoerl and Kennard (1970). Later, J. A. Nelder and Wedderburn (1972) unified these approaches, introducing generalized linear models (GLM) to have a common framework and a similar learning method for all types of linear models. The above study drew much interest from the statistical community. Then, McCullagh and J. Nelder (1989) synthesized research related to generalized linear models in their influential book. Afterwards, research on generalized linear models continued with attempts to find ways to improve performance without losing intelligibility. To keep these models simple and transparent, penalized regression methods such as LASSO proposed by R. Tibshirani (1996) were developed. As to performance, the structure of linear models is often too simple to capture nonlinear relationships with the target. Hence, Hastie and R. J. Tibshirani (1990) proposed generalized additive models to further extend generalized linear models. In this framework, a linear dependence in predictors is replaced by a linear dependence in unknown smooth functions of predictors. In some cases, this greatly improves predictive power without sacrificing intelligibility.

Generalized linear models have gradually matured and now have solid theoretical foundations. Their applied uses have expanded from the mid-1990s, particularly in regard to insurance, with the influential studies of Brockman and Wright (1992), Daykin et al. (1994) and Haberman and Renshaw (1996) and the deregulation of car insurance pricing. A comprehensive overview can be obtained from Ohlsson and Johansson (2010), Wuthrich and Buser (2017) and Denuit et al. (2019). Today, generalized linear models and generalized additive models still have a definitive presence in insurance companies due to their ease of implementation and their interpretability.

In contrast to linear models that have quickly become established in the scientific community, neural networks have experienced a more chaotic development. A timeline is proposed in **Fig. 1.1**. The current formalization of such networks is a result of numerous trials and failures. The first machine learning methods were developed in the early 1940s. McCulloch and Pitts (1944) created an original

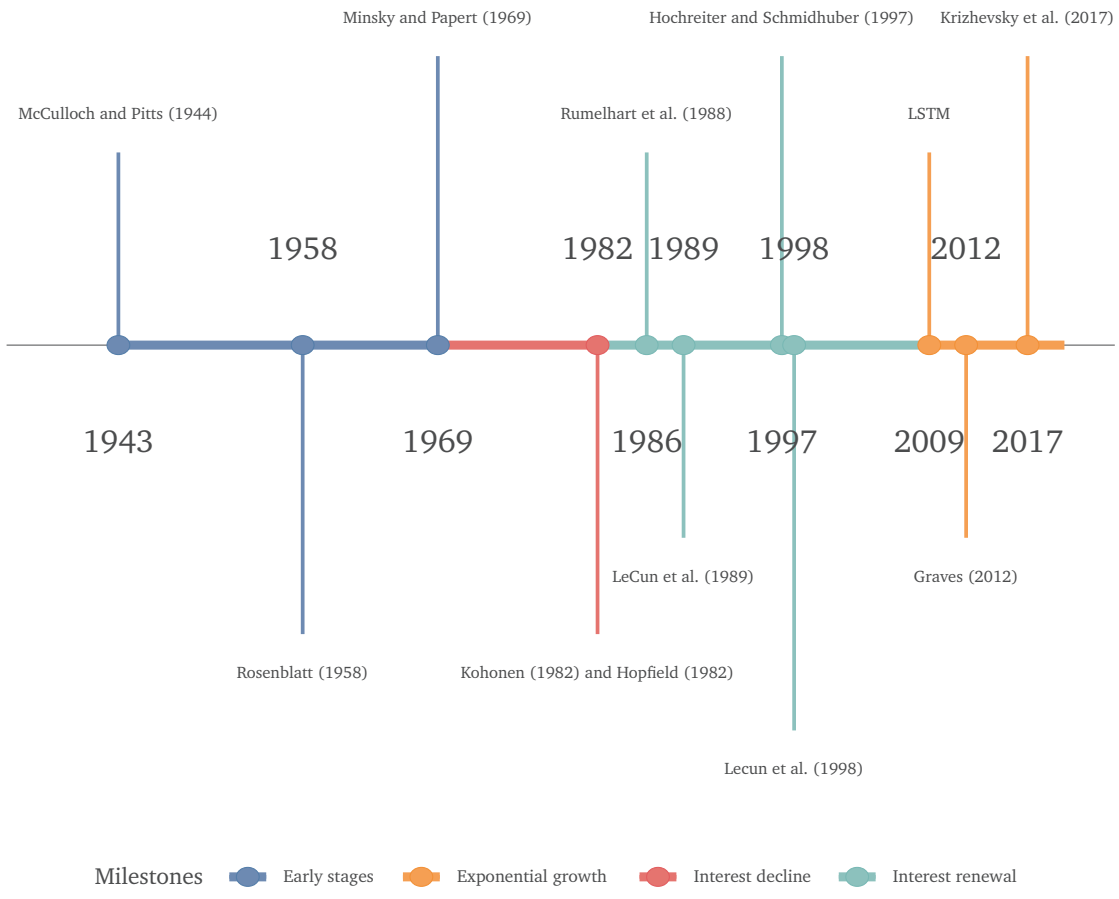

Fig. 1.1: Important milestones of neural networks' development.

mathematical model inspired by biological neurons and conceived an artificial neuron. Following this study, many researchers suggested topologies for artificial neural networks and methods to train these models. Nearly a decade later, Rosenblatt (1958) set a milestone with a single-layer perceptron. Due to the perceptron's ability to recognize simple patterns, it caught the interest of the computer science community. The cited study stimulated research around the early forms of neural networks for a while. Afterwards, the book by Minsky and Papert (1969) pointed out the impossibility for a single-layer neural network such as the perceptron to separate nonlinearly separable classes. Indeed, the researchers showed the necessity to add a hidden layer to solve, for example, the XOR (exclusive or) problem. Nevertheless, at that time no algorithm could train such a model. This led to a period of a declining interest in neural networks. That gloomy period ended in 1982 with the self-organizing map introduced by Kohonen (1982) and Hopfield (1982) with a formulation of a recurrent network based on energy functions. Nevertheless, the interest in neural networks only returned fully with the study by Rumelhart et al. (1988). This influential article proposed the backpropagation algorithm for training multilayer neural networks. Hence, that algorithm provided a generic method of training many architectures of neural networks and solved the problem highlighted by Minsky and Papert (1969). Unfortunately for interpretability, adding a hidden layer makes a neural network difficult or impossible to interpret in most cases. In fact, the weights of the hidden layer have no meaning by themselves in contrast to generalized linear models where weights correspond directly to a variable.

This was a beginning of success for neural networks since shortly afterwards, LeCun et al. (1989) and Lecun et al. (1998) proved that neural networks could perform well on images and real-world data. Neural networks have also been shown to be efficient on textual data after the introduction of long-short-term memory (LSTM) cells by Hochreiter and Schmidhuber (1997). In 2012, neural networks powered by a growing computing capacity have made it possible to achieve considerable performance gains on image classification tasks (Krizhevsky et al. (2017)) and optical character recognition (Graves (2012)). Since then, performance of these neural networks has increased to approach the human performance on simple classification tasks. The synergy between research and industry has recently helped generalize the use of neural networks for tasks such as character recognition for postal sorting, image classification, and automatic text translation.

In the actuarial literature, neural networks have been used for pricing in the tutorial by Noll et al. (2018) focusing on French automobile third-party liability data, to extract covariates from telematic data in M. V. Wüthrich (2017), Gao and M. V. Wüthrich (2018) and Gao, Meng, et al. (2018) and to classify drivers based on telematic data with a convolutional neural network in Gao and M. Wüthrich (2019). These papers illustrate a diversity of neural networks' architectures, considering feed-forward networks, auto-encoders and convolutional neural networks. The promising results of these techniques show that there is room for these methods in actuarial science.

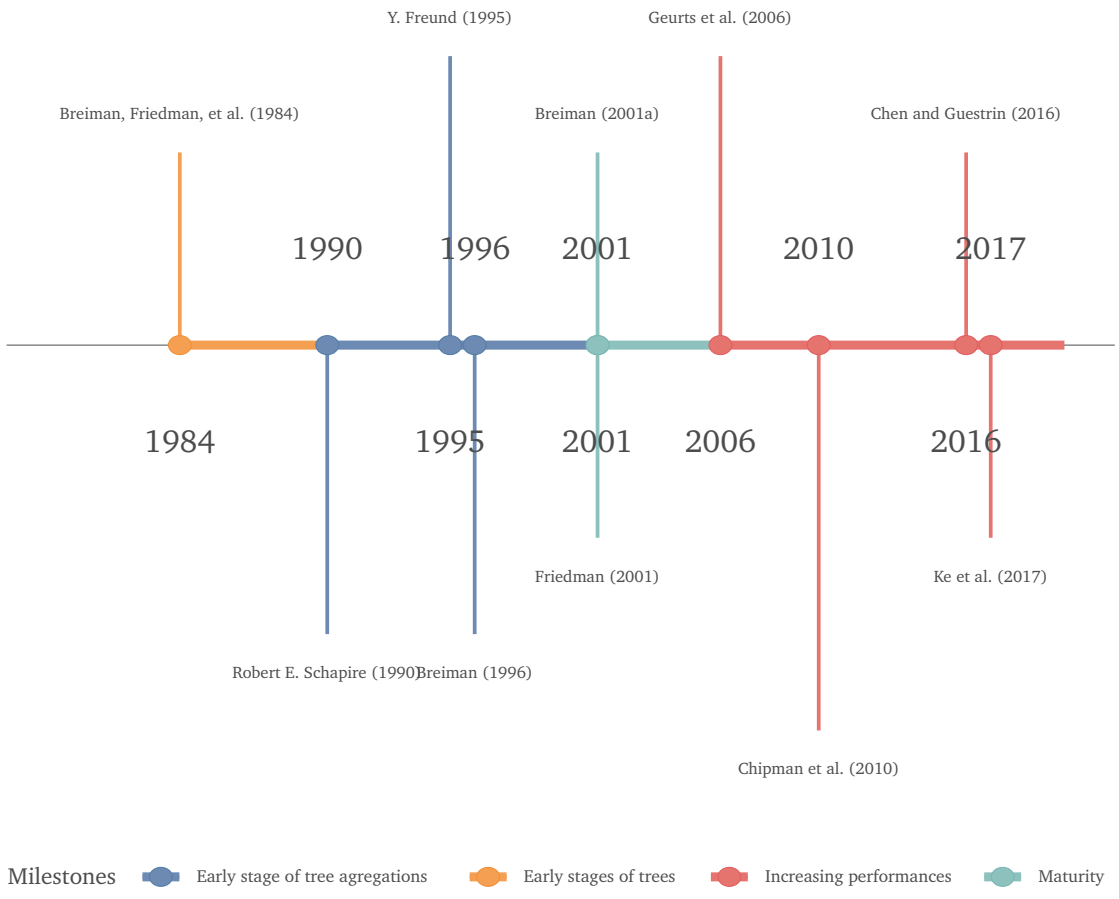

Fig. 1.2: Important milestones for tree-based methods' development.

Nonparametric methods

We now present a short history of nonparametric methods such as tree-based methods developed from the mid-1980s onwards. A timeline is proposed in **Fig. 1.2**. In his well-known book Breiman, Friedman, et al. (1984), Leo Breiman proposed an original method called classification and regression trees (CART). This is currently the most widely used form of tree in machine learning. Furthermore, it is the weak learner of many tree-ensemble methods. Other kinds of trees have been suggested, such as C4.5 trees by Quinlan (1992), but did not achieve the same popularity. CART stimulated the research community, and such trees were also implemented in a concrete use case by Breiman himself as a consultant. However, despite the popularity of this method, researchers identified numerous drawbacks, including its low precision and lack of robustness. Indeed, if the learning sample is slightly modified, the built tree can be significantly different according to Breiman (1996). These problems have since been studied, and a significant number of solutions have been proposed. This is the starting point of interpretability problems for tree-based models.

Quickly thereafter, the idea of combining several trees appeared. Nevertheless, the formulation of tree aggregation as we know it today came few years later with the articles of Robert E. Schapire (1990), Y. Freund (1995) and Yoav Freund and Robert E Schapire (1997) laying the foundations of boosting, and Breiman (1996) suggesting the basics of bagging (bootstrap aggregating). These tree-based ensemble methods produced promising results but the gain in performance and robustness came at the price of models' understandability. Hence, solving one problem raised another: the intelligibility of these models. While a (shallow) tree is intelligible and its output understandable, a tree aggregation is extremely complex to grasp for a human. Later, Breiman (2001a) and Friedman (2001) concurrently presented methods known as random forest and gradient boosting. Even though the learning algorithms were different, in Quinlan et al. (1996) and Breiman (2001a) a connection between these two methods is highlighted. Few years later, extremely randomized trees was presented in Geurts et al. (2006) and Bayesian additive regression trees were described in Chipman et al. (2010). The first method tried to improve the predictive accuracy of the "sum of trees" models generating trees with little correlation between them, while the second constrained trees of the ensemble with Bayesian methods. Recently, Chen and Guestrin (2016) and Ke et al. (2017) proposed fast implementations of gradient boosting. Hence, research on tree ensembles seemed to focus more on the accuracy and scalability of models and less on their intelligibility. Several authors indeed mentioned the intelligibility problem, but few suggested methods to increase intelligibility of models.

In actuarial science, several studies have tried an approach relying on tree-based methods to model claim frequency or severity. Among them, Henckaerts, Coté, et al. (2020) model claim frequency and severity with tree-based methods and try to obtain an insight from the related models, and Noll et al. (2018) propose a tutorial with a comparative approach to different machine learning methods. As established in Wuthrich and Buser (2017) and Henckaerts, Coté, et al. (2020), the classical L^2 loss is not well suited for insurance data since such data features an excess of zeros for claim frequency and a long tail for claim severity. Thus, the authors propose changing the splitting criterion of base

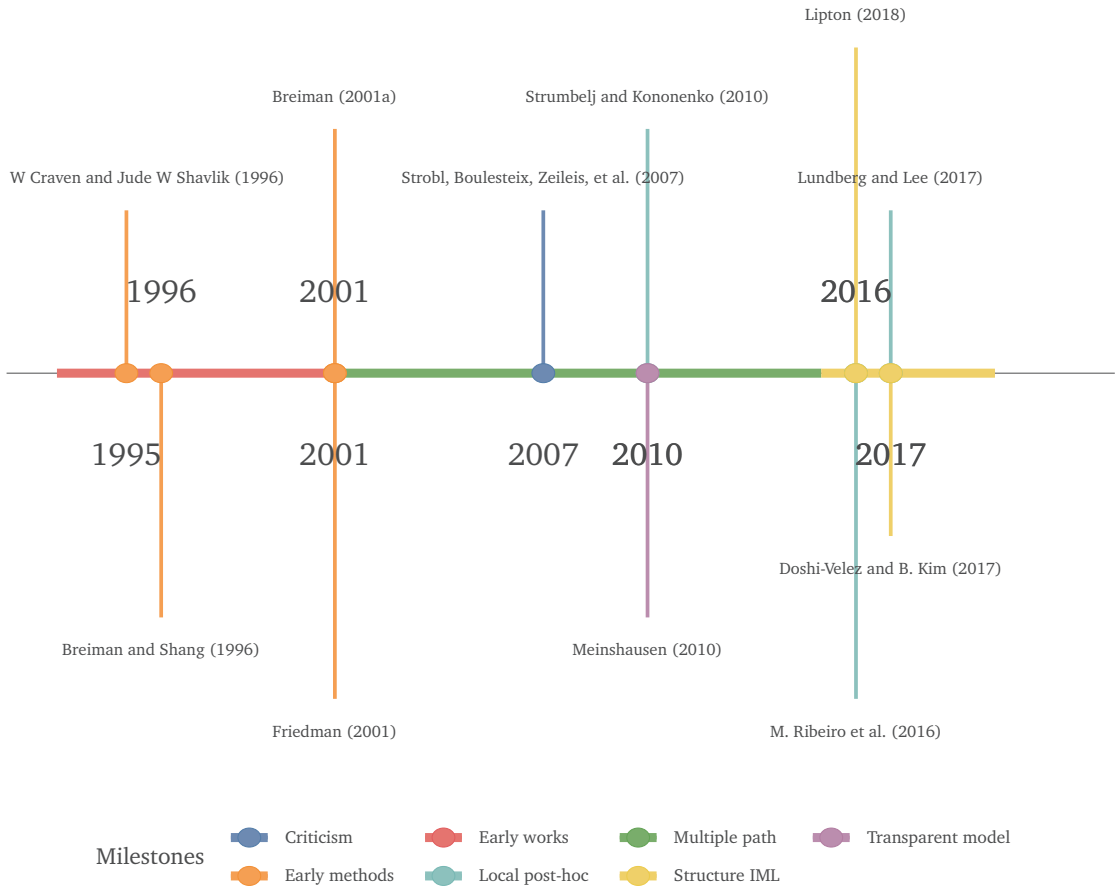

Fig. 1.3: Important milestones for tree-based methods' development.

learners to a deviance-based splitting criterion. An overview of these approaches is available in Wuthrich and Buser (2017) and Denuit et al. (2020).

Explainability methods

If we ignore models that are inherently interpretable, early studies of interpretability of machine learning models appeared shortly after the emergence of neural networks. Indeed, after Rumelhart et al. (1985) and LeCun et al. (1989), the first concrete use cases of neural networks were developed. This empirical confirmation of neural networks' performance raised concerns about their use in industry, and the lack of interpretability was pointed out. Indeed, in contrast to symbolic AI, neural networks do not have a simple structure that would allow translating their internal logic into an explanation. Consequently, the early 1990s witnessed the progression of the first methods for interpreting machine learning models such as neural networks.

In a survey, Andrews et al. (1995) reported a significant diversity of methods and strategies. However, the majority of methods tried to translate a neural network into a sparse set of simple rules. This was

probably the legacy of symbolic AI. In addition, the researchers suggested the first taxonomy of these approaches. The paper distinguished between decompositional and pedagogical approaches. The former rely on characteristics such as the architecture of the neural network model to be explained. In contrast, pedagogical methods rely not on specific characteristics but only on inputs and outputs generated by the black-box model to be explained. This terminology can still be used today. However, it seems that the terms model-agnostic (rather than pedagogical) and model-specific (instead of decompositional) are currently preferred. Among the studies of the time, a basic and important idea emerged for model-agnostic methods: the validity domain of a black-box model is the entire covariates' space.

Thus, it is possible to query the model anywhere in space and as many times as desired. This is a considerable asset since any amount of data can be generated as needed. That is the reason a black-box model is sometimes called an oracle. Mark W. Craven and Jude W. Shavlik (1994) and Mark W Craven and Jude W Shavlik (1996) developed this notion, extracting, respectively, the rules and a tree representative of the underlying black-box model. In interpretable machine learning, the model mimicking the predictions of the black-box is called a surrogate. Since this approximation is made on the whole space, the surrogate is global. Soon after tree-based ensemble methods were introduced, the same interpretability problem arose with respect to them as in the case of neural networks. Shortly thereafter, Breiman and Shang (1996) used the idea of Craven and Shavlik to extract a tree surrogate from tree ensemble models.

A few years later, tree-based ensemble methods were mature. To extract information from such complex models, Breiman (2001a) and Friedman (2001) proposed, respectively, the variable importance and the partial dependence plots. These two methods, which are now widely used, have been studied and criticized; relevant studies include those by Molnar (2020), Strobl, Boulesteix, Zeileis, et al. (2007) and Hooker and Mentch (2019). Other methods have also been developed: for example, by Apley and Zhu (2020). Noting that tree-based ensemble models were often very efficient on tabular data, several authors recommended methods that built an intrinsically interpretable model based on a tree-based ensemble model. This was the case for a rule ensemble proposed by Friedman and Popescu (2008) and the node harvest by Meinshausen (2010). The latter used d-dimensional rectangles generated by a set of trees to create a parsimonious set of rules. The objectives of the two methods were similar; however, the approaches differed. Meinshausen (2010) formulated a quadratic optimization problem, while Friedman and Popescu (2008) used penalized regressions. Other methods have also been proposed, such as evtree developed by Grubinger et al. (2014) that learns a tree model with an evolutionary algorithm and SLIM by Ustun and Rudin (2015) who formulate a mixed-integer programming problem, learning a tree representation. Nevertheless, these methods significantly increase the computational time.

Although researchers have been interested in interpretability since the early 1990s, few studies have suggested a rigorous framework for it. Thus, each community proposed its own approach. Since the beginning of the study of interpretability, relatively little attention has been paid to the target audience or the objective of an interpretability method: debugging, trust, knowledge, etc.

Consequently, the majority of the tested procedures consisted of models that were interpretable by design or methods for extracting models that were inherently interpretable. This was in fact the most natural way to proceed. After 2015, with the growing interest in interpretability methods, more emphasis was placed on techniques that had been underexploited until then. This was so for local methods that explain only one prediction at a time, and thus an explanation is only valid in the vicinity of the prediction to be explained. For this, a local surrogate is fitted near the instance of interest. One of the influential articles on the subject introduced the locally interpretable model explanation (LIME) method by M. Ribeiro et al. (2016). This, however, was not the first study of local explanation since Strumbelj and Kononenko (2010) suggested a technique based on game theory. Later, the SHapley Additive exPlanations (SHAP) method introduced by Lundberg and Lee (2017) unified the cited studies. These methods can perform a fit to any black-box model, and therefore are model-agnostic.

Currently, studies are trying to establish the structure of interpretable machine learning. Thus, several articles describe the desirable properties of a transparent model. The article by Lipton (2018) has been one of the most influential for this task. Over the preceding decade, the strong growth in the number of studies has led to a step toward the development of a more rigorous framework for interpretability Doshi-Velez and B. Kim (2017). The field is maturing but it is not yet fully developed. Although there is no consensus on the definition of interpretability and how to measure it, we can now characterize and classify techniques with the help of the numerous literature reviews, including those by Guidotti et al. (2019), Burkart and Huber (2021), Carvalho et al. (2019) and Molnar et al. (2020). By now, many methods have been produced. Example surveys include those by Guidotti et al. (2019), Adadi and Berrada (2018) and Arrieta et al. (2020). However, no method seems to have become a dominant one for everyone, and presently studies must be done to ensure the fidelity of the explanations provided, and their quality and usefulness.

In actuarial science, research on interpretability has been rare and very recent. Until now, the approaches of inherently interpretable models such as GLM and GAM were privileged. However, we note that Henckaerts, Coté, et al. (2020) and Henckaerts, Antonio, et al. (2020) tried to open the black-box and integrate their insights into generalized linear models. A tutorial that applies various techniques described by Molnar (2020) was developed by Lorentzen and Mayer (2020). So far, only already existing techniques such as variable importance, partial dependence plots, LIME and SHAP have been illustrated in these articles, and the limits of these methods in terms of intelligibility and the risk of misleading the end user remain underexplored.

Thus, history shows that the earliest statistical learning models are inherently interpretable. Indeed, statisticians have been using them for a long time and have contributed to building a solid theoretical foundation. However, these methods are unsuitable for modeling some complex and highly nonlinear phenomena. This is why various research communities introduced other methods such as tree-based methods or neural networks. These methods often showed significant gains compared to linear models; nonetheless, this improvement in performance came at the expense of interpretability. Hence, part of the scientific community has been trying to create models that are interpretable by design and

that perform as their black-box equivalents when possible. When the task is too difficult, interpretable elements are provided, but the model is no longer transparent. Hence, we observe a diversity of approaches to making models interpretable. This diversity makes the notion of interpretability difficult to define as we will in Sect.1.4.

Complex models empirically provide an upper bound on the performance that any model can achieve. Thus, the objective of research on interpretability is to reach performance comparable to that of black-box models while remaining understandable. For a long time, inherently interpretable models remained the standard. Consequently, most approaches consisted of developing a model that was interpretable by nature or extracting one from a black-box. Although these techniques are reasonable, it is not always possible to extract a simple global surrogate from a sophisticated model. Hence, the increasing complexity of neural network architectures has popularized other methods for interpretability, including local methods. Since 2015, studies of interpretability have been growing rapidly and have implemented previously underexploited methods. This profusion of ideas and techniques needs a structure to move forward. This is the current challenge of this recent discipline.

Terminology

As we have just observed, researchers in interpretable machine learning have developed a specific vocabulary since the creation of the first methods. The terms of this vocabulary are useful for characterizing and classifying explainability methods. In the literature, there is a consensus on the main terms to be used to describe these methods. We define these terms below.

post hoc: Interpretability is developed after the training of a black-box model. This means that the latter has been learned, and we want to extract information from it. For example, if we fit a random forest model and apply a method to extract the importance of variables to the model, such a method is post hoc.

ante-hoc: Interpretability is built-in from the beginning of model creation. In contrast to post hoc methods, for ante-hoc methods interpretability has been created beforehand. This means that the learning algorithm has been designed to produce a model that is interpretable by nature. For example, generalized linear models and decision or regression trees are derived from ante-hoc methods.

model-specific: The explainability method is tied to a specific class of models. For example, a method that applies only to tree-ensemble models but not to neural networks is a method that is model-specific.

model-agnostic: In contrast to model-specific methods, model-agnostic methods do not make assumptions about the model to be explained. Therefore, they can be applied to any machine learning model.

local: The explanation only holds for an instance and a close vicinity.

global: The explanation holds over the whole space.

data-dependent: A data-dependent method generates new data to construct an explanation. Since a black-box model can predict any point in space, many methods randomly generate new observations that are labeled by the black-box. Using this new learning sample, it is possible to fit an explanatory model for the black-box.

data-independent: The explainability method works without additional data. Unlike data-dependent methods, these methods do not generate new data to create an explanation.

surrogate model: This is a model that is a local or global substitute for the black-box. For example, to globally explain a random forest, we can fit a regression tree that faithfully reproduces the predictions of the black-box. This regression tree is a surrogate model. It serves as a global explanation for the complex model.

Some authors suggest other terms such as expressive power, portability, translucency and algorithmic complexity to describe properties of explainability methods. Carvalho et al. (2019) and Molnar et al. (2020) provide the relevant overviews. In this thesis, for simplicity, we prefer not to introduce these terms.

1.2 Motivations for interpretable machine learning

A recurring question when discussing interpretability is as follows: why do we need interpretable models? Before answering this question and to be fair, let us remark that interpretability is not systematically required. As pointed out by Doshi-Velez and B. Kim (2017), in at least two situations interpretability is not needed: when there are no critical consequences of wrong results and when the problem has been well studied and validated in real-world applications. To illustrate the first case, we consider an image classifier developed by a programmer for own use. This classifier is supposed to help sort through vacation photos. If the model is bad and incorrectly classifies images, the repercussions are not severe. In contrast, an image classifier implemented by Google that confuses a black person with a gorilla[1] can have a significant impact on the company's reputation and therefore on its business. Another real-life example of what could go wrong with a complex model involves Amazon. Indeed, the company's model trained to help human resources in recruitment discriminated against women.[2]. Let us consider, as the last example, an optical character recognition[3] model that extracts ZIP codes and addresses from envelopes. After years of improvement, the technique has been mastered and validated in real-life conditions. Even if the model is imperfect, it is more efficient

[1]https://www.theguardian.com/technology/2018/jan/12/google-racism-ban-gorilla-black-people
[2]https://fortune.com/2018/10/10/amazon-ai-recruitment-bias-women-sexist/
[3]This example has been suggested by Molnar (2018).

than a human and makes fewer mistakes. Furthermore, the knowledge we could obtain from this model is not particularly useful. Therefore, in this case, interpretability is not required.

Questions about interpretability arise from an incompleteness problem, as mentioned by Doshi-Velez and B. Kim (2017). The reason stems from our inability to encode the entirety of a problem in a model. For example, ethics and fairness are two notions that are difficult to formalize. Therefore, introducing them into a model is a complex task. For the moment, the best-performing models only optimize a metric. However, a single scalar cannot capture all the subtleties of a real-world problem. Thus, by making the model interpretable, we benefit from a high-performance model while maintaining control over it, and can therefore make more informed decisions. We detail below the different motivations for more interpretable models.

Human curiosity

For decades, we have been training models to automate repetitive tasks to achieve near-human performance. Today's computer vision and NLP or scoring models are powerful enough to be used in companies. This explains why they are widespread in our society. Since these models are currently used in our daily life, they can have an impact on the acceptance of a credit application or on getting a new job. According to Miller (2019), as human beings, we have the natural need to understand models or at least to understand model-driven decisions. Authors such as Miller (2019) and Molnar (2020) agree that we all have a mental model that we update to conform to unexpected events that occur. For example, when a prospect subscribes to a car insurance policy, the insurer is, for now, capable of detailing its tariff. However, if the insurer changes to a black-box pricing model, the explanation becomes much more complicated to provide and probably to understand. For instance, the prospect might not be able to understand why that prospect's rate is higher than that of another person with approximately the same characteristics. Beyond the frustration related to the lack of a satisfactory answer, the prospect cannot even know how to adapt to lower the cost. This is also true for binary models, for example, for real estate loan acceptance. An interpretable model allows one to understand why a loan application is refused and what needs to be changed for it to be accepted. These motivations concern a non-expert audience, i.e., one with no knowledge of, for example, machine learning or actuarial science. Other motivations may exist for a more expert audience. We detail them below.

Scientific findings and knowledge discovery

Facing increasingly complex problems containing thousands of variables, scientists have opened up to machine learning techniques. Indeed, methods such as random forests, gradient boosting and neural networks do not require a priori knowledge of the data. Therefore, they are often implemented in data-driven approaches. Additionally, they are capable of handling large and high-dimensional datasets. Hence, these are interesting candidates for scientists who deal with challenging data and

want to provide solutions to difficult problems. Nevertheless, scientific findings remain hidden if the model is a black-box that only provides predictions without explanations, according to Molnar (2020). To discover new scientific knowledge from these models, it is then necessary to extract understandable explanations. Finding a plausible explanation is not a guarantee of a useful discovery. Nevertheless, as mentioned by Lipton (2018), if we can clearly formulate a hypothesis based on what we learn from a black-box, we can test that hypothesis to confirm or reject it.

Consider a model that is accurate enough to predict if one will be diagnosed with cancer in the next decade. Once fitted, the model becomes a source of knowledge. Extracting knowledge from this black-box and testing hypotheses for causality will help mitigate the risks for a single person and can be of great value in conducting effective healthcare campaigns. This justifies the need for a simple and clear model to provide an explanation of a black-box model. In actuarial sciences, interpretable machine learning applied to original datasets could lead to the identification of new risk factors. For example, several articles demonstrate the utility of *telematic* data in car insurance risk evaluation. Such particularly relevant studies are those by Boucher et al. (2017) and Verbelen et al. (2018) in the pay-as-you-drive framework and those by M. V. Wüthrich (2017), Gao, Meng, et al. (2018) and Gao and M. Wüthrich (2019) in the pay-how-you-drive framework. The first two articles implement GLM and GAM to show that predictive power increases when *telematic* data is added. These inherently interpretable models can be used to explain phenomena observed in insurance. A notable example is the difference in claims frequency between men and women, who have different behavioral patterns according to Verbelen et al. (2018). However, models such as GLM or GAM require the model to be stated a priori. As a result, it is difficult to discover new relationships per se. Alternatively, we can fit the best available model in the first step and can then learn an explainable representation of that model in the second step. Hence, we could bring together the best of both approaches described in Breiman (2001b). At the time of writing, few articles in the actuarial sciences have tried this strategy.

Regulation

The General Data Protection Regulation (GDPR) replaced in April 2018 the Data Protection Directive of 1995. This update of the regulation was mandatory to allow the laws to reflect the current issues. Indeed, our personal data are being collected on a very large scale through our connected devices, and decision-making algorithms are being implemented in an increasing number of business areas. Hence, we need rules to establish a framework. This regulation poses two problems for the machine learning community, as Goodman and Flaxman (2017) pointed out. First, the subjects of the data, i.e., the people to whom the data is related, can ask for an explanation when an algorithm has made a decision concerning them (pertaining to, for example, credit acceptance or insurance pricing). Second, data subjects have the right to non-discrimination. Among other things, this means for the machine learning community that one should be very careful in the choice of predictors and one should not introduce sensitive predictors such as race, gender or wealth. Additionally, since predictors may be correlated with each other, removing sensitive variables may not be sufficient.

While interpretability is not the miracle answer to these two problems, it can clearly contribute to the solution. Indeed, interpretable machine learning methods can provide explanations for decisions made by an algorithm. These explanations are often partial, but represent a good starting point for opening the black-box. Moreover, obtaining explanations about what drives a model to make a decision makes it easier to identify the important predictors and their relationships in decision-making. As a result, it becomes easier to detect possible biases and discrimination.

Social acceptance, trust and ethics

In explaining, we naturally favor interactions with humans. Indeed, for decisions that affect us we prefer explanations and have more confidence in something we understand than in something we do not. Additionally, we place more trust in explanations that are consistent with our knowledge. These ideas are reflected in studies by Miller (2019), Martens et al. (2011) and Elomaa (1994). Therefore, methods that increase our understanding of a model allow us to raise our confidence in such models, according to M. Ribeiro et al. (2016). Moreover, as we established earlier, the ethical issues raised by machine learning models are prominent today. Without being the only option to deal with this problem, interpretability can help highlight discriminatory biases (with respect to gender, race, sexual orientation, etc.). A clear explanation of what makes the algorithm predict one value instead of another puts the stress on the variables the black-box relies on. Hence, we can more easily identify a variable or a combination of variables discriminating against a protected group. For instance, in car insurance pricing, we can no longer use gender as a variable. Nevertheless, removing the gender variable from the dataset might not be sufficient since a combination of other variables can act as a proxy of this variable. Furthermore, identifying such biases is important to avoid training a model that repeats mistakes made by humans. Let us suppose that our tariff silently discriminates against a population by assigning that population a rate higher than the market rate. Then, it would seem reasonable that the portfolio gradually becomes distorted and that such a population will disappear from the portfolio. In contrast, if a tariff segment is accidentally favored, the portfolio could acquire bad risks and progressively deteriorate. This is called adverse selection. Notably, discrimination and ethics are current topics, and the general population is increasingly concerned by these issues. Failure to take into account such considerations of ethics, social acceptance and trust could lead to a lack of confidence among policyholders. For an insurance company, this can result in a reputation problem or significant portfolio distortions.

Safety

Machine learning algorithms are already used in situations where the environment is nonstationary. Models are also deployed in settings where their use might alter the environment. Due to interpretability, it is also possible to improve the safety of machine learning systems. Indeed, interpretable machine learning models can be tested, debugged and audited more easily. Caruana et al. (2015) provide us with an instructive case study, where the authors compare intelligible and non-intelligible

machine learning models trained to predict the mortality rate of patients suffering from pneumonia. The objective is to determine whether it is possible to send the least risky patients home and treat them outside the hospital. This is not a new task, as the authors retroactively review studies performed years earlier based on similar data. These studies highlight, considering an intelligible model, a counterintuitive pattern in the data: patients with asthma have a lower risk than do those without. Indeed, for safety reasons, doctors systematically send patients with asthma to intensive care. Thus, such patients' mortality rate is low. Nevertheless, if this pattern had not been demonstrated with an intelligible model and a black-box model had recommended sending asthmatic patients home, the results would have been catastrophic for the patients. The study by Caruana et al. (2015) highlights potential new patterns of the same type with GAMs. The appeal of having an intelligible model in this case is to be able to explain the decision chains of the models and thus to identify dangerous patterns in the data. Once these patterns have been recognized, it is possible to modify and correct them, i.e., to perform debugging. This can be accomplished by replacing the variables in the data or by acting directly on the model. In insurance, risks are related to financial risks incurred by the company. Thus, a poorly segmented automobile insurance rate can have considerable financial consequences for the business. As mentioned by Lipton (2018), when a model is deployed in a non-stationary environment, e.g., to price a car insurance product, it is preferable to understand precisely what the model does. In fact, if an event occurs and the distribution of claims or policyholders is perturbed, it is important to know precisely how the model will react to prevent undesirable behavior. Furthermore, in a concurrent market, the model itself can alter the environment where it is deployed (adverse selection). This is not the only critical area. Indeed, reserving is another field with high stakes. It seems natural in these domains to have a model with understandable decisions. A sample relevant study is that by Andrews et al. (1995). Rudin (2019) goes further by stating that it is preferable to use only intrinsically interpretable models in high-stake domains.

1.3 Stakeholders in the insurance industry

In the motivations we have just provided, several target audiences for explanations appeared. Now, in a simplified insurance framework, we explicitly define the stakeholders of a machine learning system. As we have observed, motivations for interpretable models can vary according to the target audience. In the remaining part of this introduction, we develop an example of claim frequency modeling for car insurance. This example is interesting because it details the complete chain of stakeholders from the early stages of modeling to deployment. We can identify at least three main stakeholders: the actuary, the machine learning expert and the policyholder.

Actuaries are domain experts. They have a thorough knowledge of risks and statistical models. However, they are not necessarily familiar with sophisticated machine learning models. Their objective is to fit the best model consistent with their risk assessment. To do so, they may use data and their experience. Actuaries are responsible for the pricing model. Therefore, they need to be

confident in it. Additionally, they must be able to explain their model in simple terms to members of the board of directors and the management. This requires a deep understanding.

Machine learning experts and data scientists are experts in machine learning models. Generally, they do not have an in-depth knowledge of the business domain they work with. Instead, they master techniques for data preprocessing, imputation, exploration and modeling. Their objective is to propose a model that is more accurate than that already in place. During the modeling process, they may implement explainability methods. However, since they are not responsible for the model, they apply these methods to debug or improve the performance of the model.

Policyholders are lay users. We assume that they do not have specialized knowledge of statistical learning models. They want to understand the decision imposed by the system.

Remark 1 *Members of the executive board and the management can be either domain experts or lay users. However, they do not directly interact with the model. Thus, it is the actuary who must explain the model to them. In addition, the actuary may also have skills in machine learning.*

1.4 What is interpretability?

> *If no one asks me, I know it; if I try to explain it to the one who questions me, I no longer know it*. Saint-Augustin.

According to Lipton (2018), although the first methods of interpretability originate from the early 1990s, the notion of interpretability has long remained without a precise definition. With the appearance of the first complex models, the first explainability methods were developed. A sample survey of rule-based explanations for neural networks is provided by Andrews et al. (1995) . Most often, articles proposed methods for creating interpretable models without providing a definition of interpretability probably because the notion of interpretability seems so familiar to us. Over time, methods have accumulated, revealing significant diversity in the descriptions of interpretable models. As noted by Lipton (2018), the definitions of interpretability are sometimes discordant. Hence, this suggests that interpretability is not universal and refers to more than one concept. Since then, multiple definitions of interpretability have been suggested in the literature. Biran and Cotton (2017) define machine learning systems as interpretable if their operations can be understood by a human either through introspection or through produced explanations. Doran et al. (2017) further define an interpretable system as a system where a user can not only observe but also study and understand how inputs are mathematically mapped to outputs. This definition is close to that of a transparent model by Lipton (2018). Another definition is proposed by Miller (2019), who defines interpretability of a model as the degree to which an observer can understand the cause of a decision. This definition is slightly different, as it introduces the notion of causality into an explanation and a fictitious degree of understandability. From this, we observe that the formal definition of explanation remains elusive.

Krishnan (2019) goes further and states that the difficulty of defining interpretability involves the question of what the notion is supposed to capture.

Hence, the concept of interpretability is ill-defined. Furthermore, as mentioned by Adadi and Berrada (2018) and Arrieta et al. (2020), sometimes the terms interpretability, explainability, understandability and comprehensibility are used interchangeably in the literature. We agree that in some contexts it may be useful to distinguish between these terms. However, the aim of this thesis is not to feed the semantic debates but to study the problem of interpretability and to propose concrete solutions to it. Consequently, in this manuscript we use the terms interchangeably. We now define interpretability pragmatically, as suggested by Doshi-Velez and B. Kim (2017) and Guidotti et al. (2019).

Interpretability is the ability to explain or present a concept in terms that are understandable to a human.

Interpretability is therefore based on the concept of explanation, which is defined as follows by Guidotti et al. (2019).

Explanation is an interface between humans and a machine that, at the same time, both is an accurate proxy of the decision-maker and is comprehensible to humans.

An explanation in machine learning is a very subjective concept. According to Ruping et al. (2006), some prefer explanations in the form of graphs, while others prefer concise textual explanations. Still others prefer a formal model based on data, while others are more comfortable with a limited set of representative examples. Additionally, as humans, we do not all have the same ability to analyze and synthesize many graphs or rules, and neither do we have the same knowledge of a specific domain such as medicine, finance or commerce. Hence, each stakeholder of a machine learning system requires a different type of explanation. Considering the concept discussed by Woodward (1979), T. W. Kim (2018) suggests a distinction between pragmatic and non-pragmatic theories of explanation. For non-pragmatists, there exists only one correct explanation for a given event. In this approach, there is no consideration of the audience, and regardless of the complexity of the explanation and the level of knowledge required, there is only one correct explanation. In practice, only domain experts can understand complex explanations since they both have the knowledge and are used to reasoning regarding their topics. Moreover, depending on our level of education, we may be able to grasp concepts such as vector spaces, statistical graphs such as boxplots, or formal logic. Freitas (2014) and later Bibal and Frénay (2016) go further in their survey: an overly simplistic model can be rejected by a domain expert, e.g., a doctor or an actuary. In contrast, lay users (non-experts) such as patients or policyholders prefer simple explanations according to Miller (2019). Therefore, the pragmatic theory of explanations states that an explanation must be tailored to the audience. In **Fig. 1.4**, we provide a conceptual representation of what Guidotti names the "open the black-box problem". This is why we choose the definition suggested by Guidotti et al. (2019). In an extensive review of the

literature, the cited study proposes one of the first classification of explainability methods based on the form of explanations provided, e.g., graphs, text, a set of weights, a set of rules, trees, etc.

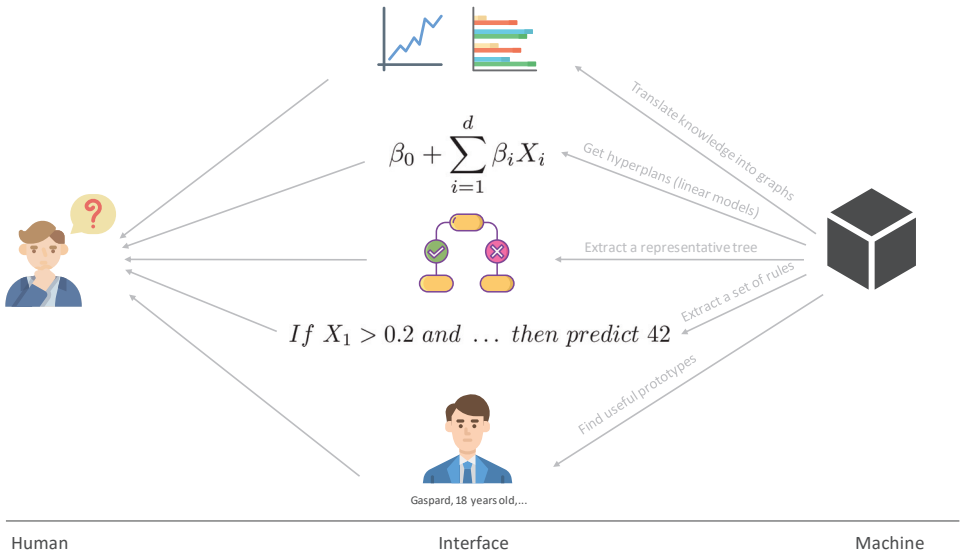

Fig. 1.4: Black-box explanation problem. How can a human learn from a learned black-box?

To summarize, we think that the definition proposed by Guidotti et al. (2019) is relevant. Indeed, it is close to the common definition.[4] Moreover, it is oriented toward the target audience due to some flexibility in the proposed explanation. It suffices to change the interface by using a tree rather than a graph to change the explanation and thus adapt to another audience. Moreover, for the same interface, for example a tree, it is possible to increase the complexity depending on the person who receives the explanation, also called the explainee.

1.4.1 Focusing on what makes a good explanation

Before we present these approaches, we think that it is a good idea to consider how humans deliver explanations to each other. An interesting study on this topic has been proposed by Miller (2019). This article synthesizes the work of many fields of research such as philosophy, psychology and social sciences to derive insight into how people define, generate, select, evaluate and present explanations. An agent is called an explainer; this is the person who explains something to the explainee. The review highlights four important facts about explanations:

1. Explanations are contrastive. An explanation is often sought in response to particular counterfactual cases. This means that most of the time if we ask a "why" question, we would rather ask why a particular event occurred instead of another. For example, if we ask "why does this one have cancer?", we mean to say "why does this person have cancer instead of not having

[4] Merriam-Webster dictionary, accessed 2020-07-27.

cancer?" Hence, this implies trying to compare characteristics that cause one person to have cancer while another does not. This point is also supported in machine learning by Wachter et al. (2017).

2. Explanations are selected (with a bias). For a real-world event, there may be dozens or hundreds of causes. Nevertheless, the human brain cannot handle so many causes. Hence, we naturally prefer to select a small subset of them. However, the selection is influenced by certain cognitive biases such as abnormality or intentionality. Indeed, explanations highlighting abnormal events seem to have more impact on the explainee. Thus, if a complete explanation is based on approximately 20 causes, we tend to select the events that seem abnormal in relation to a reference scenario rather than other causes. To illustrate this, Miller considers the example of the Challenger shuttle that exploded in 1986. Part of the full explanation is that a chemical compound reacted with oxygen in the air, resulting in an explosion of the shuttle. However, as humans, we prefer to say that the cause was related to a faulty seal. Indeed, the shuttles did not explode in most missions. However, oxygen is always present in the atmosphere. On the other hand, all shuttles did not have a faulty seal. In addition, the intentionality bias causes people to favor explanations that reflect an intention, e.g., driving fast, buying powerful cars, etc.

3. For an explanation, probability can be understood in two ways. The probability that an explanation is true, or the use of statistics and probability in the explanation itself. Miller (2019) observed that referring to probabilities or statistical relationships was not as effective as we might expect. This means, on the one hand, that using a level of likelihood for an explanation is not as convincing as an explanation exhibiting causes. Saying, for example, that the explanation we propose is 95% true is not enough to convince the explainee. On the other hand, using probabilities as elements of an explanation for an explainee is unsatisfactory. Miller proposed the following example: "a student coming to their teacher to ask why they only received 50% on an exam. An explanation that most students scored around 50% is not going to satisfy the student. Adding a cause for why most students only scored 50% would be an improvement. Explaining to the student why they specifically received 50% is even better, as it explains the cause of the instance itself.".

4. Explanations are social. An explanation is essentially a transfer of knowledge between two persons. According to Miller (2019), explanations are presented relative to the explainer's belief about the explainee's knowledge or beliefs on the topic. It is also the idea underlying the distinction between pragmatic and non-pragmatic explanations, supported by T. W. Kim (2018), De Graaf and Malle (2017) and Páez (2019) in machine learning. Thus, it is very important to assess a lay user's knowledge and beliefs to provide a credible explanation.

From these four propositions, we can infer that it is preferable to look for short explanations while not degrading their truthfulness or at least likelihood. Such explanations become even more accepted

when they are consistent with the knowledge of the explainee and allow the latter to build one's own mental model of reasoning using counterfactual cases. Hence, domain experts are not in the best position to assess the relevance of an explanation to a lay user. The idea behind this is that different levels of expertise require different explanations.

As Miller (2019) points out, there may be divergences between the expectations of a lay user and those of a domain expert. For the moment, many researchers build explanatory models (global or local surrogates) for themselves, rather than for the intended users. Miller (2019) refers to this phenomenon as "the inmates running the asylum". According to Mittelstadt et al. (2019), explanations in the machine learning sense are closer to scientific models than is an explanation in the Miller's sense. Indeed, for domain experts and machine learning experts, a surrogate model offers more information than does a single explanation. A surrogate model allows experts to rationalize, ask questions, and obtain counterfactual explanations. Furthermore, a scientist using machine learning methods to find new knowledge is unlikely to have the same selection bias. As Freitas (2014) mentions, an expert domain may in some cases select more complex models. This is also supported by Páez (2019) who states that scientific experts will be able to master the best possible explanations of the phenomena within their field of study. Thus, depending on the target audience, a good explanation may have a different meaning. However, we think that a carefully chosen surrogate can provide useful information for experts and be queried to offer good explanations for a lay user.

1.4.2 Desiderata of transparent models

Before presenting the taxonomy we have chosen, it appears important to us to present the desiderata of transparent models formulated by Lipton (2018). By showing the mechanisms of transparent models, we can easily highlight by contrast what is missing from black-boxes that prevents them from being interpretable.

- The notion of simulatability proposed by Lipton (2018) refers to a model simple enough for a human to take the input data together with the parameters of the model and, in a reasonable time, step through every calculation required to produce a prediction.

- The second notion of transparency introduced by Lipton (2018) is decomposability. A model is said to be decomposable if it (its inputs, parameters, and calculation) admits an intuitive explanation.

- The final notion of transparency is algorithmic transparency. It relates to the extent a learning algorithm is transparent.

We illustrate these on models recognized as transparent, such as GLM or CART. This application to models familiar to actuaries allows us to highlight explanation mechanisms of these models and what we would like to identify in more sophisticated models. To be more concrete, we use existing models

already fit to real-world data in Noll et al. (2018). Since in this thesis our goal is to illustrate the mechanisms of interpretation of these models, we do not redevelop the parts related to their fitting. For each model, we show that it is transparent in the sense of Lipton (2018). Then, we highlight the type of explanations each model can provide. A list of common explanation types in machine learning can be found in Mohseni, Zarei, et al. (2018).

For theoretical results and developments on GLM, the interested reader can refer to McCullagh and J. Nelder (1989). Practical applications to actuarial problems are developed in Ohlsson and Johansson (2010), Wuthrich and Buser (2017) and Denuit et al. (2019). For CART, the reference for theoretical elements is Breiman, Friedman, et al. (1984), while applications are implemented in Wuthrich and Buser (2017) and Denuit et al. (2020). Finally, for an overview of the interpretable mechanisms related to these models, the reader can refer to Molnar (2020).

Generalized linear models

As mentioned above, we illustrate these properties on a fitted Poisson GLM model with a log-link function. We are uninterested in the learning process, and presume that an algorithm provides us with this model. Once the weights $\boldsymbol{\beta} = \{\beta_0, \ldots, \beta_d\}$ have been fit, the equation that makes it possible to predict is of the form

$$\log(\mu_i) = \beta_0 + \beta_1 x_{i,1} + \ldots + \beta_d x_{i,d} \tag{1.1}$$

where μ_i is the estimated mean for policyholder $i = 1, ..n$, and $x_{i,j}$ is the value of the j-th variable for the i-th policyholder. Hence, the dependence structure between the response and covariates is completely specified a priori. This plays a major role in the ease of interpretation.

If there is a reasonable number of variables, then the model is observable all at once. Even though the term "reasonable" is not rigorous, we cannot do better since it depends on the cognitive capacity and experience of the user. In our example in **Tab. 1.3**, we have approximately 50 weights. We think this number is manageable. Apart from this consideration, if we want to calculate the output for a specific instance, we only need to multiply the exponential value of the product of input values and weights to compute it. This may require some time if the model has many variables but is rather easy. Hence, this model is simulatable. If the number of weights is too high, for example, $10,000$, the model loses simulatability and hence cannot be considered transparent.

If our predictors are not built from complex feature engineering, we can presume that the model is decomposable since each input is understandable by itself and is related to a unique weight. This means that the parameters could be described as representing the strengths of association between

each feature and the label. If a model contains complex feature-engineered variables, it is no longer transparent, as we do not know what the weights relate to.

Finally, the learning algorithm can be proven to converge and is considered transparent. In this thesis, we do not explore algorithmic transparency further. Instead, we focus on what we can learn from a fitted model.

Now, let us examine the types of explanations and information this transparent model can provide. Due to simulatability and decomposability, we can easily interact with the model.

Tab. 1.1: Policyholder to be explained.

VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Area	Density	Region
11	0	54	50	B12	Diesel	C	291	Midi-Pyrenees

Tab. 1.2: Standardized policyholder to be explained.

VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Area	Density	Region
11	-1.24	0.6	-0.62	B12	Diesel	C	-0.38	Midi-Pyrenees

"Why" explanation

For a random policyholder in the training set, we can calculate the claim frequency and decompose the effect of each variable. This provides an explanation for a given policyholder. Since the model is fit on standardized data, we can provide policyholders' characteristics in **Tab. 1.1** because they are quantities understandable to humans, and show in **Tab. 1.2** the standardized policyholders' characteristics used to compute the output. We only need 10 (9 variables + intercept) coefficients to obtain the annual predicted claim frequency for this observation. These coefficients are presented in **Tab. 1.3**. The following formula gives the predicted annual claim frequency for this policyholder:

$$e^{-2.63}e^{0.17\times 1}e^{-1.24\times(-0.22)}\dots e^{0.00\times(-0.38)}e^{-0.15\times 1}\approx 10.96\%$$

The predicted claim frequency for this policyholder is approximately the same as the annual empirical claim frequency on the whole dataset, 10.06%. The above equation clearly shows that the brand new car of this policyholder increases the predicted claim frequency of the latter by 32% ($e^{-1.24\times(-0.22)}=1.32$), all other things being equal. In contrast, the bonus equal to 50 lowers the prediction by 19.63% ($e^{-0.62\times 0.35}=0.8036$), all other things being equal. Applying this reasoning to every variable, we

can understand for this individual which characteristics increase or decrease the predicted claim frequency.

Tab. 1.3: Summary of the Poisson GLM.

Feature	Estimate	Std. Error	z value	$Pr(>\|z\|)$
(Intercept)	-2.63	0.09	-28.30	3.89e-176
VehPower5	0.15	0.02	7.78	7.43e-15
VehPower6	0.18	0.02	9.46	3.14e-21
VehPower7	0.11	0.02	5.88	4.12e-09
VehPower8	-0.05	0.03	-1.72	8.49e-02
VehPower9	0.29	0.03	9.95	2.47e-23
VehPower10	0.24	0.03	8.35	6.93e-17
VehPower11	0.17	0.04	4.45	8.55e-06
VehPower12	-0.01	0.06	-0.11	9.15e-01
VehPower13	-0.01	0.09	-0.06	9.50e-01
VehPower14	0.17	0.10	1.72	8.62e-02
VehPower15	-0.09	0.10	-0.91	3.61e-01
VehAge	-0.22	0.01	-31.77	1.53e-221
DrivAge	0.09	0.01	15.74	7.60e-56
BonusMalus	0.35	0.01	69.21	0.00e+00
VehBrandB10	0.05	0.04	1.16	2.45e-01
VehBrandB11	0.13	0.04	3.11	1.89e-03
VehBrandB12	0.19	0.02	10.21	1.74e-24
VehBrandB13	0.01	0.04	0.28	7.76e-01
VehBrandB14	-0.10	0.08	-1.24	2.14e-01
VehBrandB2	0.00	0.02	-0.29	7.71e-01
VehBrandB3	0.00	0.02	0.09	9.29e-01
VehBrandB4	-0.02	0.03	-0.51	6.09e-01
VehBrandB5	0.07	0.03	2.74	6.16e-03
VehBrandB6	-0.03	0.03	-1.09	2.76e-01
VehGasRegular	0.06	0.01	5.14	2.68e-07
AreaB	0.06	0.02	2.73	6.28e-03
AreaC	0.10	0.02	5.33	9.84e-08
AreaD	0.20	0.02	9.56	1.17e-21
AreaE	0.23	0.03	8.40	4.60e-17
AreaF	0.18	0.09	1.90	5.75e-02
Density	0.00	0.02	0.15	8.82e-01
RegionAquitaine	-0.11	0.09	-1.22	2.22e-01
RegionAuvergne	-0.31	0.12	-2.60	9.35e-03
RegionBasse-Normandie	-0.01	0.10	-0.12	9.02e-01
RegionBourgogne	-0.02	0.10	-0.17	8.68e-01
RegionBretagne	0.06	0.09	0.67	5.00e-01
RegionCentre	0.04	0.09	0.49	6.25e-01
RegionChampagne-Ardenne	0.11	0.12	0.93	3.50e-01
RegionCorse	0.06	0.11	0.50	6.18e-01
RegionFranche-Comte	-0.14	0.17	-0.82	4.13e-01
RegionHaute-Normandie	-0.07	0.11	-0.67	5.03e-01
RegionIle-de-France	-0.03	0.09	-0.34	7.37e-01
RegionLanguedoc-Roussillon	-0.05	0.09	-0.57	5.69e-01
RegionLimousin	0.14	0.11	1.28	2.00e-01
RegionMidi-Pyrenees	-0.15	0.10	-1.58	1.14e-01
RegionNord-Pas-de-Calais	-0.19	0.09	-2.01	4.42e-02
RegionPays-de-la-Loire	-0.04	0.09	-0.42	6.78e-01
RegionPicardie	0.03	0.10	0.33	7.38e-01
RegionPoitou-Charentes	-0.07	0.10	-0.71	4.80e-01
RegionProvence-Alpes-Cotes-D'Azur	-0.03	0.09	-0.32	7.46e-01
RegionRhone-Alpes	0.06	0.09	0.61	5.41e-01

Why not? A contrastive explanation

In this model, the prediction for any individual is considered with respect to a reference policyholder. As explained by Molnar (2020), if all predictors have been standardized, the reference policyholder is represented by the mean of each numerical predictor and the reference category of each categorical predictor. For this individual, the predicted annual claim frequency is e^{β_0}. Therefore, we can propose a form of contrastive explanation by comparing the characteristics that differ with respect to this reference individual. We answer the following question: why is our policyholder riskier than the reference?

Generally, in insurance the goal is to create tariff cells Ohlsson and Johansson (2010). Thus, continuous predictors are often discretized to create categorical ones, and the final model is only composed of categorical predictors. This makes it easier to create a meaningful reference policyholder.

"What if" explanation

We can also ask ourselves, for a given policyholder, what would happen if one or more characteristics were different. For example, if the policyholder lived in Limousin instead of Midi-Pyrenees, the response of the model would have been 14.7%. In this case, changing this attribute increases significantly the predicted annual claim frequency, all other things being equal. We can also perform this analysis, altering numerical predictors, and more than one value at a time.

"How to" explanation

Additionally, we can ask a slightly different question: how do we alter a characteristic of an individual to obtain a given output? To this end, we only need to solve a simple equation. Thus, it is easy to explain to a non-expert the conditions under which that individual could obtain a better tariff. Let us consider an actionable variable such as vehicle age. A policyholder could ask the following question : what vehicle age could lower that person's annual claim frequency to 8%? Using the equation and weights, we can tell the policyholder that an 8-year-old vehicle could lower the predicted annual claim frequency to 8%.

"What else" explanation

The last type of explanation consists of finding other instances that generate the same output as does the policyholder we try to explain. For our policyholder, buying a car with $\texttt{VehPower} = 14$ does not change the prediction. In the case of numerical variables, we could find an infinite number

of points that predict the same output. Nevertheless, as humans, we cannot process too many points. In contrast, in the case of a categorical variable there is only a finite set of predictions. Hence, if the number of variables and categories is reasonable, as is the case for pricing models, we can exhaustively describe all predictions. Therefore, we can find all instances with the same predictions.

We have just reviewed how we can extract an explanation at an individual level. We have also shown that provided explanations are short and can be clearly formulated.

The model can also be used to extract scientific knowledge. Let us go back to equation (1.1). We can write the expected claim frequency as a function of predictors by applying an exponential function on both sides of the equation:

$$\mu_i = e^{\beta_0} \prod_{j=1}^{d} e^{\beta_j x_{i,j}}.$$

Assuming that the j-th variable is continuous, increasing $x_{i,j}$ by one unit leads to

$$\mu_i = e^{\beta_j} \left(e^{\beta_0} \prod_{j=1}^{d} e^{\beta_j x_{i,j}} \right).$$

Hence, increasing $x_{i,j}$ by one unit multiplies the response by a factor of e^{β_j}. In our concrete example, increasing the driver's age by one unit increases the annual claim frequency by 9.68%, all other things being equal.

Assuming that the j-th variable is categorical, the interpretation is slightly different. The response is multiplied by e^{β_j} if the individual has the modality represented by variable j instead of the reference category, all other things being equal. In our example, if a policyholder has a vehicle with `VehPower` $= 11$, then the annual predicted claim frequency increases by 18.71%, all other things being equal. If this vehicle has the reference level 4, the predicted claim frequency is unchanged. Due to this, we can conclude that in this model it is riskier to have a car with `VehPower` $= 11$ than a car with `VehPower` $= 4$. Unfortunately, we cannot conclude that the more powerful the car is, the riskier the policyholder is because this variable is considered categorical, and the relationship with the target is not monotonic.

Finally, an important advantage of GLM models is that they assess the importance of a predictor in the model. It is indeed possible to test whether a coefficient is significantly different from 0 according to the z-statistic. There also exist procedures to select models with a parsimonious set of covariates.

Remark 2 *We recall that correlation is not causation. The relationships identified by the model are not guaranteed to be causal. However, they can be useful to a scientist (for example, an actuary) to understand various phenomena. We can then test their causality.*

Classification and regression trees

We have just described one of the most widely used transparent models in actuarial science. GLM models belong to the family of parametric models. This means that the structure must be defined a priori before the parameters are estimated. From the interpretability point of view, this is a considerable asset since we can introduce our beliefs into the structure of the model, and hence are more willing to accept it. However, it is unlikely that this model correctly reflects reality. It may therefore be interesting to adopt a data-driven approach to create a model from the data. For this, it is common to use nonparametric models and in particular those of the CART type. Here, we fit a Poisson CART model on the same data as that used for the Poisson GLM. The documentation [5] for the `rpart` package and a tutorial [6] are both available online. The theoretical elements and a use case are developed in Wuthrich and Buser (2017) and Denuit et al. (2020). The only difference is that we do not need to standardize data since the learning algorithm is not sensitive to the scale effect. The resulting Poisson regression tree is displayed in **Fig. 1.5**.

In regard to GLM, we show that a tree with a reasonable number of leaves has the three desirable properties of a transparent model. First, a tree of a reasonable depth is observable all at once. An example is shown in **Fig. 1.5**. It is very easy to compute a prediction for a given individual since it is sufficient to check at each node whether the individual satisfies the condition. Depending on the result, the individual is placed in the right or left node. Following the chain of conditions from the root to the leaf (a terminal node), we obtain the rule and the prediction for that individual. Thus, the model is simulatable.

Tab. 1.4: Another policyholder to be explained.

VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Area	Density	Region
7	5	39	58	B1	Diesel	A	24	Centre

[5] https://cran.r-project.org/web/packages/rpart/rpart.pdf
[6] https://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf

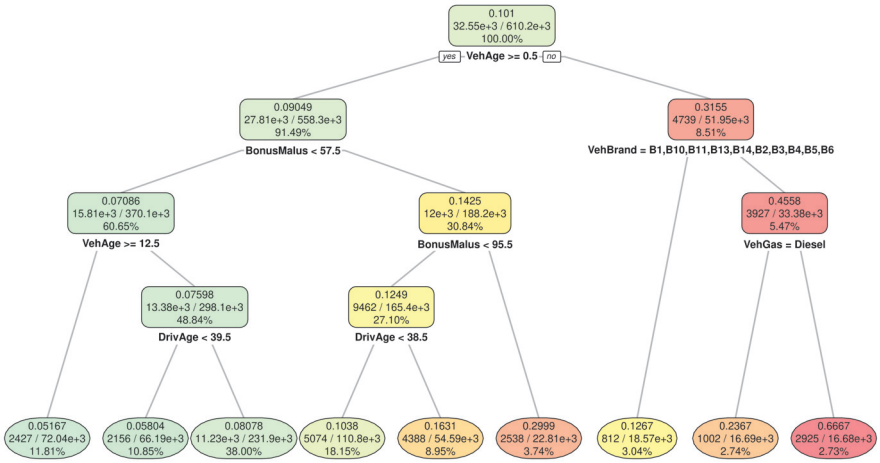

Fig. 1.5: Regression tree predicting the annual claim frequency.

Nodes are the components of a CART model. These are nothing more than d-dimensional rectangles in the covariates' space. The rectangles are expressed as a combination of conditions in the form of (variable, threshold) pairs. In this sense, we can state that the model is decomposable since each node (an elementary component of the model) has a simple explanation: that a characteristic is above or under a specific threshold. For example, all individuals of the first child node on the left satisfy `VehAge` $>= 0.5$. Once again, the learning algorithm can be proven to converge. Therefore, a tree with a reasonable depth is considered transparent.

Let us now examine the type of explanations and information that can be extracted from such a model.

"Why" explanation

Above, we have already shown the mechanism for individual explanations. Let us follow the path for the policyholder in **Tab. 1.4**. As `VehAge` above 0.5, the policyholder is placed on the left node. Because `BonusMalus` is between 57.5 and 95.5, the policyholder is successively moved to the right and left node. Finally, since `DrivAge` greater than 38.5, the policyholder ends up at the right leaf. This path can be converted into a rule:

If `VehAge` $\in [0.5, \infty)$ and `BonusMalus` $\in [57.5, 95.5)$ and `DrivAge` $\in (38.5, \infty)$ then 16.31%.

This rule serves as a "*why*" *explanation*.

Why not? A contrastive explanation

Here, we again can offer contrastive explanations. For example, if we consider the individual in Tab. 1.4, the predicted claim frequency is approximately 16.31%. Going back to the first ancestor of this leaf, we observe that if the individual were younger than 38.5 years of age, then that person's claim frequency would decrease to 10.38%. This answers the question of why the claim frequency is 16% rather than 10%. We can also ask the following: why is our policyholder riskier than the three leaves at the bottom left? The answer is as follows: because $\texttt{BonusMalus} >= 57.5$. Another question could be as follows: why is our policyholder riskier than the less risky profiles? The reason is that $\texttt{BonusMalus} >= 57.5$ and $\texttt{VehAge} < 12.5$. A binary tree naturally provides a form of contrastive explanations.

"What if" explanation

For a given policyholder, we can alter the conditions one-by-one and monitor the cell in which the policyholder is placed. In our example, if $\texttt{VehAge} = 0$, then the predicted claim frequency decreases to 12.67%. Changing the $\texttt{VehBrand}$ has no effect on the predicted claim frequency for this policyholder. We can also change multiple conditions, $\texttt{VehAge} = 20$ and $\texttt{BonusMalus} = 50$, and the response will be 5.17%.

"How to" explanation

Since all outputs are directly visible on the tree, we can very easily answer the following question: how can a given variable be modified to obtain a prediction? For instance, how can the lowest predicted annual claim frequency, i.e., 5.17%, be obtained? By driving one more year without filing a claim and being at fault, our policyholder would reach a predicted claim frequency of between 5% and 8%. However, after one year, the driver's age will be 40. To ensure that the lowest predicted claim frequency, the policyholder will need a vehicle that is over 12.5 years old.

"What else" explanation

In this analysis, we want to find instances with similar predicted values. Let us recall that a leaf is a d-dimensional rectangle containing points with the same output. This rectangle can be translated to a rule: for example,

If $\texttt{VehAge} \in [0.5, \infty)$ and $\texttt{BonusMalus} \in [57.5, 95.5)$ and $\texttt{DrivAge} \in (38.5, \infty)$ then 16.31%.

Finding instances with the same output is easy. From this rule, we can either create a fictitious policyholder that belongs to the cell, or finding instances in the training or testing set.

Furthermore, due to the tree's simple and hierarchical structure, we can quickly identify variables that impact predictions and those that do not. For example, variables `Area`, `Region`, `VehPower` and `Density` do not appear in our tree, and the prediction does not depend on them. Therefore, altering these variables has no impact on predictions.

In addition, we can instantly identify the leaves with the highest or lowest predicted values. In our example in **Fig. 1.5**, the color codes correspond to higher or lower claim frequencies. We can therefore follow the reverse path, i.e., starting from a leaf, and infer the characteristics necessary to reach it. This is very useful for understanding phenomena on a global scale. Indeed, we can determine a set of conditions that leads to a high or low predicted annual claim frequency, which is valuable knowledge for the actuary's risk assessment. Therefore, scientific knowledge can be extracted from such a model.

If the hierarchical structure does not suit us, we can transform the partition into a set of rules. To evaluate the importance of a rule, we estimate the proportion of individuals covered by it. The advantage of a rule is that we can switch conditions and present them in a suitable order. However, presenting information in this way can make the analysis complex, as the tree size and hence the number of rules increases.

We have presented the mechanisms and properties that help interpret transparent models, i.e., to explain a model globally or to provide an explanation for an individual. Next, we will explore the extraction of interpretable elements from models that are no longer transparent, such as the models produced by a random forest, gradient boosting or deep learning. To this end, many methods have been proposed since the advent of the first explainability methods in the early 1990s. These different approaches to the "open the black-box" problem attempt to mimic some of the properties of transparent models. For example, the notion of variable importance that is present in GLM and a CART tree has been proposed by Breiman (2001a) to help in the interpretation of random forest models. Local methods such as LIME by M. Ribeiro et al. (2016) or SHAP by Lundberg and Lee (2017) attempt to reproduce the ability to obtain an explanation for a given individual. Other methods such as TREPAN by Mark W Craven and Jude W Shavlik (1996) or "born again trees" by Breiman and Shang (1996) provide a global explanation of the model by approximating a black-box model with a simpler model. At this time, none of these approaches can transform an opaque model into a transparent one. However, by mimicking some desirable properties of transparent models, such approaches can make black-box models more interpretable or at least provide insights. In the following section, we propose the taxonomy we have chosen to classify the approaches.

1.4.3 Taxonomy of methods for interpretability

To take into account the latest explainability methods, several classification schemes have recently been proposed in the literature. Let us mention in particular those of Carvalho et al. (2019), Molnar (2020), Adadi and Berrada (2018), Arrieta et al. (2020), Guidotti et al. (2019) and Burkart and Huber (2021). We prefer the classification of Guidotti et al. (2019). The latter is based on four different problems presented in **Fig. 1.6**.

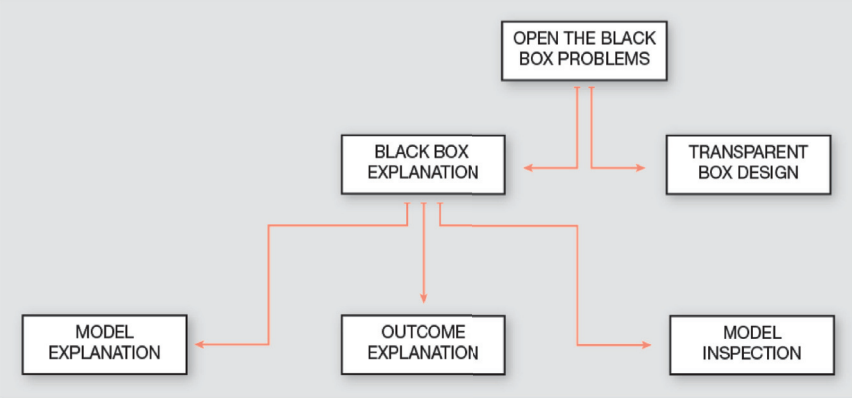

Fig. 1.6: Different machine learning model intelligibility approaches. Source: Guidotti et al. (2019).

At the highest level, "open the black-box" problems can be split into two categories. On the one hand, there are problems related to the explanation of a trained black-box (a black-box explanation); on the other hand, there are problems related to the design of a model that is transparent by nature (the transparent box design). Black-box model explanations can be further divided into three subproblems: the problems that consist of exhibiting the internal logic of a black-box model (model explanation), problems with solutions that consist of finding an explanation for a particular instance (outcome explanation), and finally the model inspection problems with the objective of studying how the outputs of a black-box vary when the input is perturbed.

To clarify this taxonomy, we provide the formulation of each problem as proposed by Guidotti et al. (2019). Then, we illustrate this taxonomy with popular explainability methods and discuss the limitations and the targeted audience of these methods.

Inspection methods

The model inspection problem consists of providing a representation (visual or textual) for understanding some specific property or predictions of a black-box model.

For example, variable importance of Breiman (2001a) and partial dependence plots (PDP) of Friedman (2001) are among the most well-known methods among black-box inspection methods.

More recently, Goldstein et al. (2015) suggested using individual conditional expectation (ICE) plots to refine PDP, and Apley and Zhu (2020) described the use of accumulated local effects (ALE) plots to obtain a better representation of PDP in the presence of highly correlated predictors. Another prototype of this class is a generalization of variable importance proposed in Fisher et al. (2019). The interested reader can refer to Molnar (2020) for an overview of these methods.

For the purpose of illustration, we apply inspection methods to the `Poisson GBM` model created in Noll et al. (2018). The data and hyperparameters are described in the cited article. The code to reproduce the model is available[7] online. We presume here that we only have access to the black-box `Poisson GBM` and the data. We want to explain the variability of this black-box. A set of explanations provided by these methods is displayed in **Fig. 1.7**.

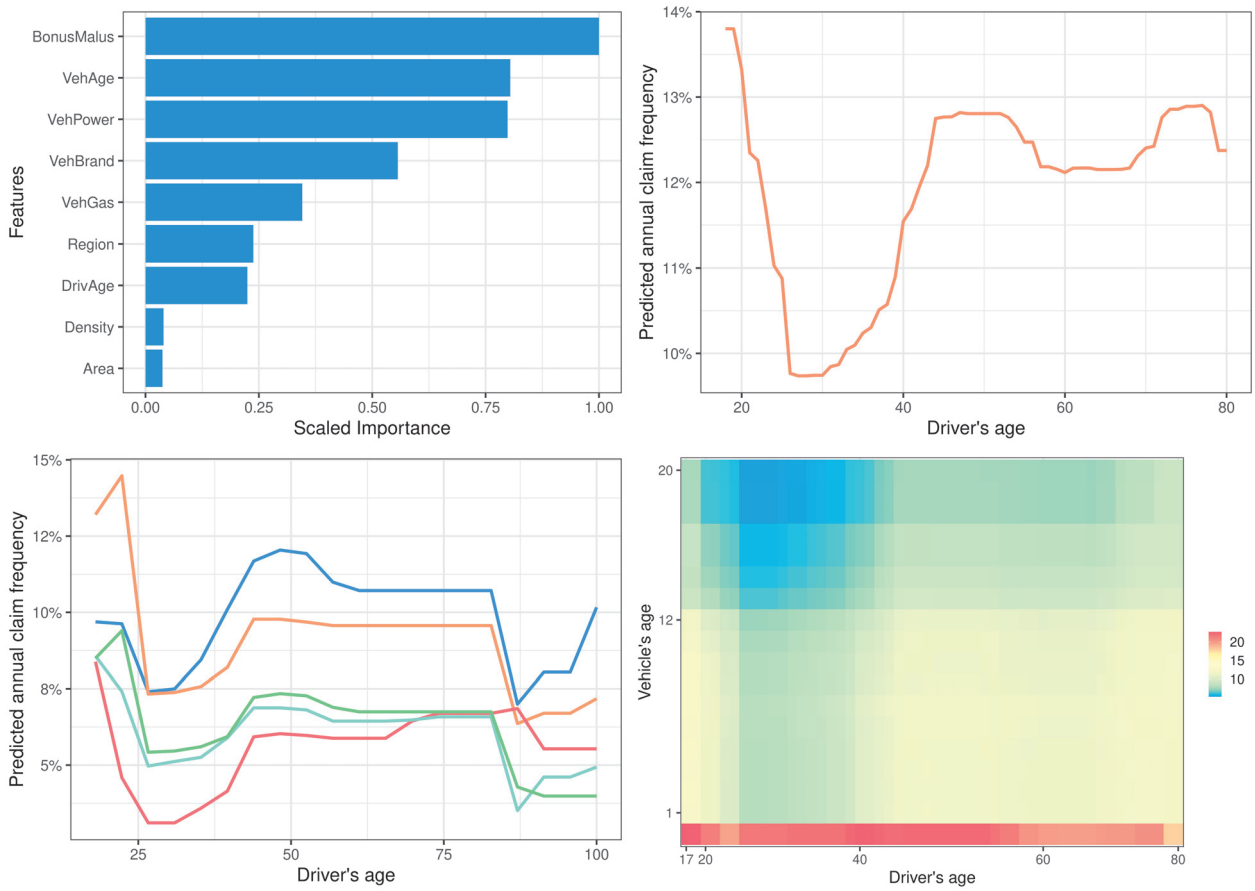

Fig. 1.7: Most common inspection methods applied to `Poisson GBM`.

Top left: a variable importance plot for the `Poisson GBM`. Variables are sorted from the most influential to the least influential.

Top right: a partial dependence plot for driver's age. We can observe the mean response of the model as a function of driver's age.

[7] https://github.com/JSchelldorfer/ActuarialDataScience

Bottom left: an ICE plot for 5 individuals. This plot shows for each individual the response of the model as a function of driver's age. Note that we cannot represent 600,000 trajectories on the same graph because it would not be readable.

Bottom right: a 2D partial dependence plot representing the interaction between driver's age and vehicle's age. We can observe the mean response of the model as a function of driver's age and vehicle's age.

Remark 3 *As noted in the limitations below, these methods provide less reliable results when variables are correlated. This is the case in our dataset. For example, the linear correlation coefficient between Bonus-Malus and driver's age is* -0.48.

Limitations and target audience

One of the most widespread forms of interpretability among machine learning community is black-box inspection. A look at Kaggle's forums can convince the reader of this. Methods such as variable importance and partial dependence plots are almost systematically used to analyze trained black-boxes. The success of inspection methods is mostly due to their ease of implementation and the simplicity of reading the graphs provided.

Nevertheless, we believe that these methods are not very well suited to a lay user audience. According to Miller (2019), a lay user is not very comfortable with explanations in the form of probabilities or statistics in general. Furthermore, Wachter et al. (2017) state that these forms of explanation are arguably more difficult to use and understand for a non-expert.

Moreover, many criticisms of reliability of these methods in the presence of correlated predictors have appeared in the literature. Strobl, Boulesteix, Zeileis, et al. (2007) compare three different methods of obtaining variable importance from a random forest, namely, selection frequency, Gini importance, and permutation importance. The study shows that all of these measures are unreliable and underscores that permutation importance is biased toward correlated features. This bias is also highlighted by Strobl, Boulesteix, Kneib, et al. (2008) in bioinformatics. A more general critical analysis is provided by Hooker and Mentch (2019). Indeed, the latter point out that a partial dependence plot and individual conditional expectation plots increase in variability in some cases if features are correlated. The paper concludes that these methods can be highly misleading in the presence of statistical dependence between predictors. Some recent methods such as the accumulated local effects plot introduced in Apley and Zhu (2020) propose an alternative to a partial dependence plot to resolve these issues. However, it often comes at a price of a more difficult plot to interpret. The pros and cons of these methods are discussed by Molnar (2020). In light of these criticisms, we believe that to correctly interpret these graphs, it is necessary to have some experience with machine learning.

Finally, the provided level of information is low. For example, a variable importance plot only offers an estimate of important variables for the model, while a partial dependence plot provides a mean response effect as a function of one or two variables at a time. Nevertheless, for high-dimensional models with high-order interactions, analyzing a black-box with univariate or bivariate graphs does not suffice for obtaining a concise idea of what is happening in the black-box. Inspection methods are therefore fundamentally limited in the amount of information they can graphically convey. It is impossible for a human to understand the inner logic of a black-box or to comprehend how the latter computes predictions. Consequently, it is unfeasible to measure the faithfulness of an explanation. It is also implausible to observe the big picture with a method of this kind or even to obtain a holistic view described in Lipton (2018). However, despite these numerous criticisms, these methods can be very useful for a machine-learning technician. Indeed, they allow debugging a model and confirming that variables are as important in the model as we think they are in reality. This can be done only from a univariate or bivariate point of view.

Outcome explanation methods

Given a black-box and an input instance, the outcome explanation problem consists of providing an explanation for the outcome of the black-box on that instance. It is not mandatory to explain the whole logic underlying the black- box, and only the reason for the prediction on a specific input instance is required.

An intuitive method is to fit a decision tree in the vicinity of x, the instance for which we want an outcome explanation. The latter could be the path in the decision tree followed by attribute values in x as in our approach in Sect. 1.4.2. Nevertheless, in practice the most relevant methods belonging to this class are probably the locally interpretable model explanation (LIME) introduced by M. Ribeiro et al. (2016) and SHAP presented by Lundberg and Lee (2017). Some refinements of these methods have been proposed recently. For example, the Anchors technique suggested by M. T. Ribeiro et al. (2018) tries to improve LIME, and TreeSHAP and KernelSHAP attempt to increase the computational efficiency of SHAP.

Tab. 1.5: Policyholder to be explained.

VehPower	VehAge	DrivAge	BonusMalus	VehBrand	VehGas	Area	Density	Region
11	0	54	50	B12	Diesel	C	291	Midi-Pyrenees

Similarly to inspection methods, we illustrate outcome explanation methods with two of the best-known methods of this class. Hence, we apply LIME and SHAP to our Poisson GBM and require an explanation of the outcome for the policyholder in **Tab 1.5**. The results are shown in **Fig. 1.8**.

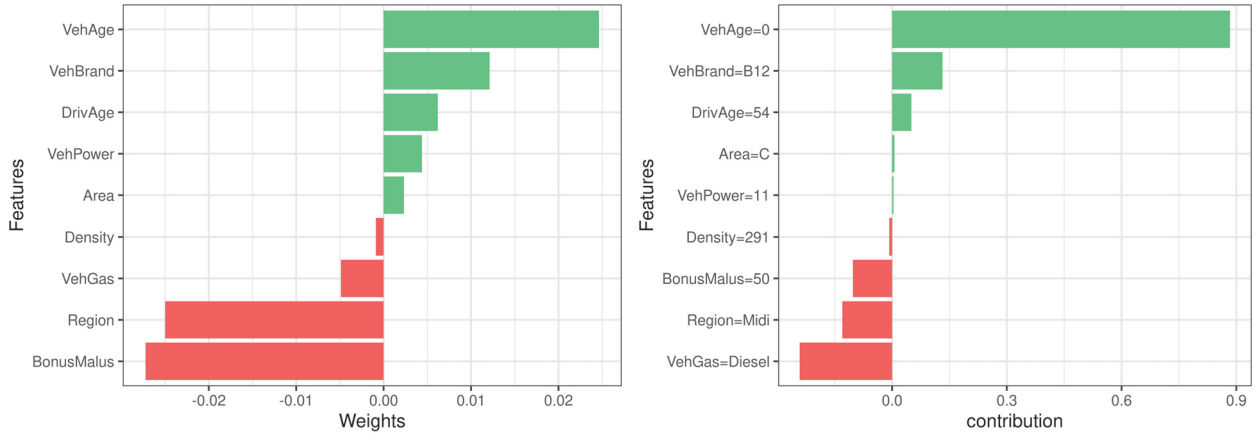

Fig. 1.8: Left: LIME explanation for the observation in **Tab. 1.5**. The value predicted by the local model is 10%, and the true value is 16.7%. For the local fit, $R^2 = 0.285$. Right: SHAP explanation for the same observation.

The two explanations for the same point are different. We have little confidence in the LIME explanation since the quality of the local fit is very low. For SHAP, the explanation is valid only for that point. Therefore, it is very difficult to have hindsight.

Limitations and target audience

Outcome explanation methods recently popularized by LIME and SHAP have also been highly appreciated by the machine learning community, and interest in them has been growing since their introduction. The basic idea is to note that in the absence of a global explanation of the model, we can at least provide an explanation for a given instance. This explanation will only be valid in the vicinity of the point to be explained. Consequently, the explanation may be inconsistent with the true global explanation. Several studies such as those by Páez (2019) and Mittelstadt et al. (2019) compare this to scientific models. Nevertheless, compared to scientific models, most local methods provide little or no information about the validity domain of the explanation. For example, the Newton's law is applicable to explaining the collapse of a bridge but is not valid in quantum physics. Similarly, locally interpretable models should provide a clear validity domain and highlight their limitations. The authors of LIME propose an exponential kernel to weight observations, which represents their definition of the neighborhood. We do not think that a mathematical tool of this kind is easily comprehensible to a lay user. Therefore, using such methods could be misleading for the user if their limitations are not fully understood. For example, the user could draw erroneous conclusions from the explanations provided by the methods of this class.

Another, more general, argument commonly used against outcome explanation methods is that understanding a model with a collection of pointwise explanations is hardly conceivable. A method of this type is therefore not suitable for a user seeking a global view of the model. This is also not

appropriate for discovering scientific knowledge and detecting bias or discrimination, as mentioned by Wachter et al. (2017).

A synthesis of responses to the French "Autorité de Contrôle Prudentiel et de Régulation" (ACPR) consultation[8] shows concerns about the stability and reliability of explanations provided by these methods. A more detailed version is also available. [9] Alvarez-Melis and Jaakkola (2018) show that some of these approaches (at least LIME and SHAP) are locally not sufficiently stable. The idea behind stability is that for an explanation to be usable correctly, it should be coherent locally, i.e., its neighbors should have similar explanations if they have the same predictions by the black-box. Laugel, Renard, et al. (2018) show that creating a too stable local surrogate could lead to a local explanation that is not faithful to the initial model. Hence, there is a tradeoff between stability and fidelity. Such stability and fidelity issues are critical. Indeed, as Lipton (2018) points out, we are inclined to believe plausible explanations. Unfortunately, plausible does not mean true. If we want to avoid misleading the end user, we must provide stable and highly faithful explanations.

Even though using a local explanation has many drawbacks, these methods can convey more information than can black-box inspection methods. Most outcome explanation methods use a local surrogate model to provide an explanation. Hence, an explanation is a set of weights, a path in a tree or a set of rules. We think this gives clearly formulated and concise explanations. Therefore, we can measure the fidelity of an explanation. This is an important improvement compared to inspection methods because we can assess whether we can trust an explanation. Furthermore, for a domain expert or a machine learning technician with a deep knowledge of the limitations of a local model, an approach of this kind allows debugging a model or understanding the local behavior of a black-box model.

Due to the local surrogate, in some cases it is also possible to obtain counterfactual explanations. An explanation of this kind is required for the lay user according to Miller (2019), Wachter et al. (2017) and Laugel, Lesot, et al. (2019). More research has to be done to ensure that these methods provide stable and locally faithful explanations. If such a method were available, it could therefore be suitable for explaining the prediction of an instance to both a lay user and a more knowledgeable audience.

Model explanation methods

The black-box explanation problem consists of providing a global explanation of the black-box model through a transparent model. The latter should both be able to mimic the behavior of the black-box and be understandable by humans. In other words, an interpretable model approximating the black-box must be globally interpretable.

[8] https://acpr.banque-france.fr/sites/default/files/medias/documents/20201215_analyse_gouvernance_evaluation_ia.pdf

[9] https://acpr.banque-france.fr/sites/default/files/medias/documents/20201215_synthese_gouvernance_evaluation_ia.pdf

Remark 4 *The model explanation problem differs from the inspection problem since it requires the extraction of an interpretable global predictor. Recall that the inspection problem focuses on the analysis of specific properties of the black- box without requiring a global understanding of it.*

Unfortunately, few methods of this class are as well-known as inspection and outcome explanation methods. However, we can mention TREPAN by Mark W Craven and Jude W Shavlik (1996) and born again trees by Breiman and Shang (1996). Both propose extracting the structure of a tree from a learned black-box. Generally, training a CART model on the predictions of a black-box model produces a global explanation. We illustrate methods of this type with one that is among the simplest to implement. We first create a new database from our learning database. To do so, we replace the number of claims by the claim frequency predicted by the `Poisson GBM` and remove the offset. At this point, the target is the predicted annual claim frequency. We fit a regression tree on these data. The resulting tree – our global surrogate – is presented in **Fig. 1.9**.

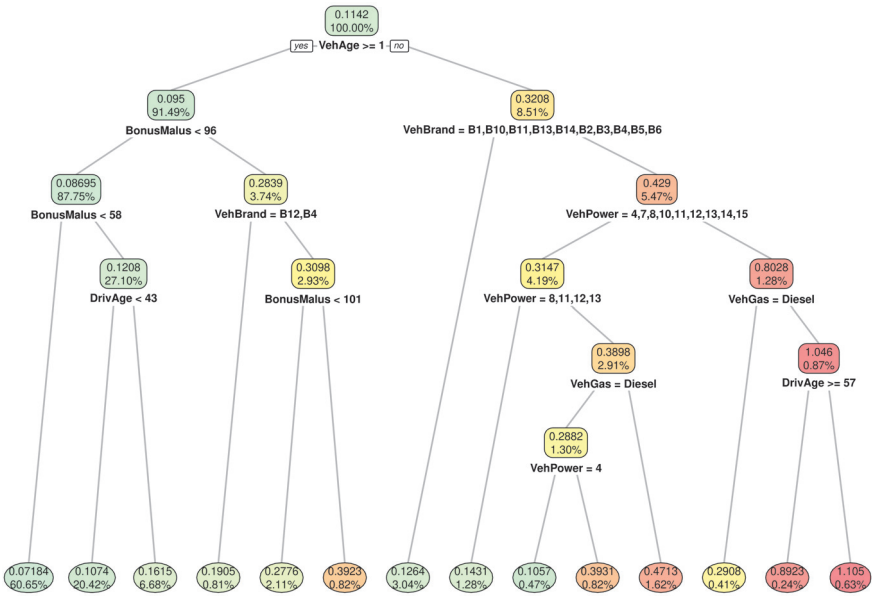

Fig. 1.9: Global surrogate for a `Poisson GBM`.

We evaluate the fidelity with the R^2 score between the predictions of the `Poisson GBM` and those of our surrogate tree. On the training set, $R^2 = 85.7\%$; on a 7-fold cross-validation, $R^2 = 85.7\%$, and on the test set, $R^2 = 87.5\%$. This explanation is faithful and allows us to observe which policyholders are considered risky by the `Poisson GBM` model.

Remark 5 *We do not evaluate the stability of this explanation. However, as CART trees are not robust, if we change our training set, the resulting explanation could be different.*

Limitations and the target audience

Providing a simple explanation for an entire and complex model is a challenging task. The most powerful methods such as tree-ensemble methods and neural networks create models with a very large number of trees, rules or weights. This complexity generally improves performance. This is where the black-box model explanation and transparent box design approaches diverge. In the case of model explanation, the objective is to find a surrogate model. The latter has to mimic the predictions of the trained black-box over the entire space. Thus, there are two distinct models: one used to predict, and the other used to explain. This keeps performance intact since the complex model produces predictions without any parsimony constraints. The second model, simpler, integrates a parsimony constraint that makes it less accurate but more interpretable. Once again, the criticism formulated by Lipton (2018) holds. Indeed, if the surrogate model is not sufficiently faithful and does not capture the correct interactions between variables, it can mislead the user about the predictions of the black-box. Nevertheless, compared to outcome explanation methods, we do not need to worry about the validity domain of the surrogate since it is global.

Compared to outcome explanation methods, model explanation methods can result in unstable explanations. Indeed, if a method generates data to learn the interpretable model, there is a risk of instability in the explanation. Let us consider the simplest possible example: we want to mimic our black-box with a CART tree. To do so, we build a learning base from the available data and create a target variable with our oracle (the black-box model). CART trees are known for their low robustness. Therefore, the choice of the initial learning sample changes the explanation. Bastani et al. (2017) mention this stability problem not only for their proposed method but also for born again trees by Breiman and Shang (1996) and consequently for TREPAN by Mark W Craven and Jude W Shavlik (1996). However, the definition of stability is different from that proposed for a local approach. Indeed, here stability is in a sense close to robustness. In the methods we study in chapters 2 and 3, we are confronted with this problem. We can also safely state that this problem is present for most data-dependent methods. For the moment, few studies assess stability. In particular, Bastani et al. (2017) propose a metric without going into detail.

Maintaining a simple model while approaching a complex one is not easy. As a result, the explanations provided may be less accurate than local explanations. However, this should not be generalized. Suppose now that we can extract a stable and faithful tree. This surrogate can be used not only to explain global variations of the black-box but also to approximate predictions of the latter. An explanation of this type can therefore be used by all types of users: lay users, domain experts, and machine learning experts.

Transparent design methods

The transparent box design problem consists of directly providing a model that is locally or globally interpretable.

Among the methods that belong to this class, there are generalized linear models of J. A. Nelder and Wedderburn (1972), classifications and regression trees of Breiman, Friedman, et al. (1984), and generally the penalization methods such as LASSO by R. Tibshirani (1996). We have already illustrated models of these kinds in Sect. 1.4.2.

Limitations and the target audience

Creating a model that is transparent by design is the second approach to obtaining a global explanation of the model. There are several ways to do this; examples include making a priori assumptions about the target distribution and the structure of the model (linear), generating a tree with complexity constraints, generating a set of rules with length and number constraints, introducing a penalty term, etc. Since intelligibility is introduced beforehand in the model, the target audience can naturally be taken into account. As we have demonstrated in Sect.1.4.2, a policyholder (a lay user) can calculate own premium from the GLM coefficients and an actuary (an expert user) can also extract relevant information such as risk factors. Since the model that provides predictions is also the model that provides explanations, there is no risk of making a different interpretation of a prediction. There is also no stability problem. When the stakes are high, Rudin (2019) suggests using only models that are transparent by nature. This seems obvious at first sight, and this is why this approach has long prevailed in interpretable machine learning. However, performance is often degraded by the imposition of constraints during learning, and it is quite rare to see simple models such as GLM or CART winning in machine learning competitions. This, however, does not mean that these models always perform worse than their black-box alternatives.

Synthesis

In this taxonomy, explainability methods are thus classified according to the problem to which they provide a solution. The latter takes different forms depending on the problem. Hence, the solution may

- explain the model,
- explain the outcome,

- inspect the black-box internally, or
- provide a transparent solution.

Here, the answer to the problem provides an interface between humans and machines. Consequently, it is the nature and the extent (global or local) of the interface that allow these methods to be classified. The target audience is implicitly taken into account in this taxonomy. Even if the limitations we have pointed out are not exhaustive and do not systematically apply to all methods of the same class, this should guide the reader in the selection of an explainability method. Moreover, we are convinced that even if some methods are more suitable for certain tasks than others, it is preferable to conceive interpretability at multiple levels.

1.5 Evaluation

Evaluating the interpretability of a machine learning model is necessary to be able to compare models with each other. It is a step toward a more rigorous approach to interpretability in machine learning, as mentioned by Doshi-Velez and B. Kim (2017). However, assessing interpretability of models is not an easy task.

First, there are several aspects of interpretability evaluation, as discussed by, for example, Mohseni, Block, et al. (2018). In particular, when contemplating post hoc interpretability, it seems important to assess the quality (fidelity and stability) and the usability for a target audience. To some extent, we can quantify fidelity and stability with several metrics, but usability is much more difficult to measure. An explanation that did not satisfy the first two criteria would be useless. Even worse, it could mislead the end user. When post hoc methods are used, Lipton (2018) recommends the utmost caution. Indeed, insufficient local or global fidelity could result in the end user of the explanation being misled. Recent studies also point in this direction: for example, Schneider et al. (2020) show that it is possible to create false explanations that deceive humans.

An interpretable model should allow a human to answer different types of questions correctly within a limited time. Unfortunately, evaluating interpretability of a model on a task that can be performed by a human is expensive and complex. Doing so requires one to carefully select a panel, prepare a set of questions to study the understanding of each participant and analyze the results. Consequently, few authors provide such studies to evaluate their explainability methods. As Doshi-Velez and B. Kim (2017) state, the majority of articles assess interpretability via a proxy such as the tree depth. In a recent paper, Páez (2019) confirms that too few articles propose genuine evaluations of interpretability of their models.

Some interesting research has already been done to evaluate interpretability of models such as trees for different target audiences. In several studies, participants were asked to directly assess the

interpretability of the model they used with a score. However, interpretability is a subjective notion that depends on the individual, and one's experience and level of knowledge. Therefore, results may vary depending on the target audience, namely, domain experts, lay users, machine learning experts, etc. In Piltaver et al. (2016), tree-structured models are presented to experts and non-experts to the purpose of performing tasks such as classifying an instance, explaining a prediction, validating an assertion or discovering new relationships. For each task, the accuracy of results and the response time are measured. The study shows that data-mining experts perform better and faster on the proposed tasks. The study also shows that the structure of the proposed tree affects the response time for all audiences, and in particular the number of leaves increases it. Nevertheless, as mentioned by Allahyari and Lavesson (2011), trees that are too shallow can be regarded as not very interpretable because they do not provide enough information to the user. This is also the opinion of Freitas (2014) for domain experts. Huysmans et al. (2011) compare different ways of presenting information: decision trees, decision tables and rules. It is indeed possible under certain conditions to switch from one representation to the other. Subject to these hypotheses, the article shows that non-expert users of its panel preferred decision tables to trees or rules. Other authors such as Martens et al. (2011) propose a justifiability metric to evaluate and compare interpretability of models. Overall, the appeal of these studies could entail having proxies that make it possible to quantify for each target audience and for each model the level of interpretability of the latter.

Synthesis of the introductory chapter

This introduction allows us to present the notions necessary to understand the contemporary concept of interpretable machine learning. We show that there is no consensus on the definition of interpretability. Consequently, many approaches have been developed since the beginning of interpretable machine learning in the early 1990s. In addition to discussing the vocabulary, we present a taxonomy to classify methods according to the type of explanations they provide and the target audience. Moreover, for the insurance industry, we suggest a simplified description of stakeholders for a machine learning model. This simplified vision should enable the reader to better understand which methods are to be applied to provide explanations for different stakeholders and problems. Finally, we point out the difficulties related to the evaluation of interpretability methods. It is quite challenging to quantify a notion that is hard to define. However, we show that interpretable machine learning is trying to structure itself. If it is impossible, at this time, to measure interpretability, we can at least evaluate the quality of the provided explanations by using the notions of fidelity and stability. This manuscript is mainly interested in explainability methods that provide global explanations for tree ensemble models. These models often perform well on tabular data. Moreover, we believe that obtaining a global explanation allows both extracting scientific knowledge and providing explanations to a non-expert audience. Chapters 2 and 3 are dedicated to the presentation of two such methods. Nevertheless, we also want to study interpretability of neural networks. The latter are gaining in popularity and can be interesting in some cases, as we showed in the beginning of this introduction. However, by losing the strong structure imposed by trees in the space, we also lose explanatory

capability. Therefore, we weaken the problem of interpretability and limit ourselves to local methods in the last chapter of this thesis.

1.6 Our contributions

To conclude this general introduction, we now summarize the contributions resulting from this thesis. They will be detailed later in chapters 2-4.

Chapter 2: Toward an explainable machine learning model for claim frequency

This chapter corresponds to the article by Maillart (2021b) published in the European Actuarial Journal in 2021. We propose a claim frequency model based on a tree-ensemble model that we make interpretable. In the context of car insurance pricing, tree-based methods are an interesting alternative to generalized linear models. Indeed, these methods require little data preprocessing and perform as well as or better than generalized linear models without too much hyperparameter tuning. Nevertheless, these methods generate black-box models, which constrains their deployment. Some methods that allow extracting interpretable elements from tree ensembles such as variable importance and partial dependence plots already exist. However, they do not provide a clear understanding of what is happening inside the black-box. There are also methods explaining locally any observation of any black-box model. These methods can therefore be used on tree ensembles. Nevertheless, these methods make it difficult to grasp a black-box model all at once. Few methods specifically aim at providing a global explanation of a tree-based model. Such methods include born again trees developed by Breiman and Shang (1996), the node harvest suggested by Meinshausen (2010), and DefragTrees proposed by Hara and Hayashi (2018). All of these techniques attempt to create a model that is interpretable by nature using only predictions of the tree such as done by Breiman and Shang (1996) or predictions of the tree and the tree structure as done by the node harvest and DefragTrees. Our approach is different since we propose extracting a surrogate model from the tree set to explain this black-box model globally. Hence, we retain the performance of the black-box model while improving its interpretability.

To obtain our global surrogate, we modify DefragTrees, the method developed by Hara and Hayashi. This method allows extracting a set of rules from a tree ensemble model, i.e., a random forest, gradient boosting, etc. To do so, the authors propose using the structure of trees contained in the tree ensemble. Indeed, the d-dimensional rectangles generated by the trees can be written as a probabilistic model depending on parameters. Thus, this is possible to express with parameters the probability distribution of pairs (prediction, rectangle). Therefore, we can express the likelihood and attempt to maximize it. In practice, we use the EM algorithm to obtain a numerical solution of this problem. The approach of Hara and Hayashi produces a parsimonious set of rules used both to predict and to explain. We differ from this approach since our strategy is to keep two models, one to

predict and the other to understand. As a result, we do not reduce the performance of our best model. Additionally, the DefragTrees method creates rectangles that may overlap or leave gaps in space. In other words, it does not induce a partition of space. Therefore, a point can have many explanations if it belongs to several rectangles simultaneously. We propose a modification of DefragTrees that allows generating a partition of the space and obtaining a unique explanation for each individual. We think this is preferable from the interpretability perspective.

From the actuarial science point of view, we develop a method we can use to extract scientific knowledge from a tree-based model. Tree-ensemble methods learn the structure of the model from data without a human prior and are data-driven methods. Thus, when we do not have a priori knowledge about the data, such a method provides a strategy for extracting scientific knowledge from the data. We illustrate this on telematic data already studied by Verbelen et al. (2018). The latter use an approach based on generalized additive models. The structure of the model is thus additive. In our case, we make no hypothesis about the structure of the model and let the algorithm learn from data. In the second step, we extract knowledge from our black-box, which is a Poisson random forest. As a result of this strategy, we highlight a relationship that was not pointed out by Verbelen et al. (2018): young drivers who do not drive regularly during the year have a significantly higher claim frequency than do others.

In addition, we propose an alternative strategy. This consists of extracting knowledge from a tree-ensemble model and injecting it into a model that is easier to deploy. To do so, we integrate the interactions created by the leaves of the surrogate tree into a generalized linear model. This significantly improves the performance of the latter while remaining within a known framework. This approach also has the advantage that it can be deployed quickly and at no additional cost to the insurance company.

Chapter 3: Tail-index partition-based rules extraction

This chapter corresponds to Maillart and Robert (2021). Here, we propose modeling the tail index function with a gamma gradient-boosting machine. The tail index is an important parameter that measures how often extreme events occur. In many applied fields, this index depends on explanatory variables. In this chapter, we assume that it takes a finite number of values over a partition of the explanatory variables' space. To produce this simpler model, we aggregate the small rectangles created by the gamma gradient-boosting model with an ascending hierarchical clustering method with a spatial constraint. Hierarchical clustering is performed on the predictions of the gamma gradient boosting within the rectangles, and the spatial constraint takes into account proximity between rectangles. Thus, close rectangles with approximately the same predicted values are grouped early in the process. We obtain our final model by selecting an appropriate number of classes. Nevertheless, this model is not easy for a human to interpret because the shapes of regions are no longer rectangular and become too complex to understand in a high-dimensional space. Therefore, in this chapter

we introduce an original approach to extracting a surrogate from the partition created by the tree ensemble. This idea is similar to that in the previous chapter. Indeed, we note that too many small rectangles fragment the covariate space. Thus, we want a method that aggregates these rectangles to form larger ones while maintaining a good level of fidelity. To this end, we rely on being able to easily access the partition generated by each tree in the set. Furthermore, we can determine the cross-partition for the set of partitions, and can query the unique prediction of the tree ensemble within each rectangle of the cross partition. Thus, we have the knowledge of the tree ensemble at any point in space. However, there is too much information (d-dimensional rectangles) for a human to interpret this model. Consequently, we modify the CART learning algorithm so that it splits only where the tree ensemble has already split. Finally, we provide as input to this new algorithm a database composed of a unique observation within each rectangle of the cross-partition and the corresponding black-box predictions. Hence, the algorithm learns to clone the cross-partition. The created maximum tree perfectly reproduces the cross-partition but is not interpretable. To make it interpretable, we only need to prune the tree to a depth that suits us. The pruned tree is our surrogate model.

From the interpretability point of view, this method stands out in several aspects. Proceeding with a regression tree to learn the cross-partition, we obtain a set of nested trees. Hence, each split creates a deeper tree that is included in the previous one. This is a form of consistency between each pair of nested trees and thus between explanations. This is what allows us to transform the interpretability problem into a tree pruning problem. Choosing a tree consists of measuring the fidelity for each nested tree and selecting the trade-off between fidelity and complexity that suits us. Another consideration is that most explainability methods provide either a global explanation or a local explanation but not both. Although this was not the initial objective, our method can also extract local explanations that are faithful to a set of trees. Moreover, our method differs from other global interpretability methods for tree ensembles because we measure fidelity not on a few observations, but over the whole space. Finally, our method is deterministic. It is therefore much less sensitive to stability problems than is, for example, DefragTrees.

From an actuarial point of view, our nonparametric approach allows us to obtain an estimate of the tail index as a function of covariates. The value of this index plays a fundamental role in the pricing of XS reinsurance contracts. Similarly to Goegebeur et al. (2021), we can obtain an estimator of the conditional reinsurance premium. Goegebeur et al. (2021) propose a local estimation method. This means that only the observations with covariate values that are close to the value considered for pricing are retained. Our method allows obtaining an estimator over the whole covariate space and handling many covariates because it does not suffer from the dimensionality curse problem.

Chapter 4: Black-box inspection via robustness analysis

This chapter corresponds to Maillart (2021a) and was published in the Bulletin Français d'Actuariat in 2020. In this chapter, we explore interpretability of parametric models such as generalized linear models and neural networks. Our starting point is the method proposed by Koh and Liang (2017). It identifies points in the learning sample that are influential for a given prediction. To this end, the authors develop three indicators that approximate the removal or a perturbation of a point from the training sample. These indicators are derived from influence functions used in robust statistics. The first measures the variation of the black-box's parameters following the deletion of a point from the learning sample. The second approximates the loss variation in prediction resulting from the removal of a point. The last indicator quantifies the loss variation in prediction after the perturbation of a point in the learning sample. These yardsticks only depend on gradients and the Hessian that can be computed using deep learning frameworks such as TensorFlow or PyTorch. This avoids refitting the black-box model each time a point in the learning sample is modified. As a result, this technique is feasible for real-world datasets.

In contrast to Koh and Liang (2017), who use this method for image classification, we apply it to tabular data. We demonstrate that the first two indicators can be used to efficiently spot points considered abnormal by the black-box model. Thereby, it is possible to implement this approach in a data quality procedure. This is not the only asset of these indicators. In fact, we show in the chapter that the points identified as influential for the model are mainly located around the decision boundary. Thus, we can locate the decision frontier and consequently the points for which it is interesting to have an explanation. However, for tabular data it is not easy to extract clear information from a set of influential points. Indeed, such points for a model are distributed along the decision boundary. As a result, we obtain a set of points with heterogeneous characteristics belonging to both classes. Therefore, we suggest using the points not as prototypes for a specific explanation, but rather as elements that help construct hyperplanes tangent to the decision frontier. Accordingly, we provide local explanations – our hyperplanes – that are faithful to the black-box's decision boundary. Hence, our main contribution consists of providing an algorithm taking influential points and the black-box model and returning a locally faithful explanation in the form of a hyperplane. Finally, we demonstrate exclusively for numerical data that it is possible to use the last indicator to determine a direction in which to perturb a point to maximize the loss variation in prediction. It turns out that this direction is in fact a normal vector to the decision surface and can therefore be used directly as a local explanation for a given point.

After illustrating on a two-dimensional dataset the appeal of each indicator for the extraction of faithful explanations, we apply them to insurance data. To this end, we consider data used in car insurance pricing. However, in this chapter, we study a binary classification problem. Hence, we transform the problem into a prevention problem to fit in a binary classification framework. We examine predicting risky profiles to make preventive recommendations. On this high-dimensional

dataset, we show that we are able to extract local explanations that are faithful to the original black-box model. We verify that for a generalized linear model, the method always returns the same explanation. This confirms empirically that the explanations are consistent with the global decision frontier within the known framework of generalized linear models. Then, we apply this method to extract explanations from a neural network. Our fidelity measures show that the explanations are faithful to the initial decision frontier. This does not allow for a global understanding of the neural network's variations. Nevertheless, it can be useful for providing an explanation for a subgroup of policyholders. At the time of writing this manuscript, there is very little research on this subject in the actuarial literature. This study makes a contribution by introducing a novel technique of interpretable machine learning in actuarial sciences.

Bibliography

Adadi, Amina and Mohammed Berrada (2018). "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160. DOI: 10.1109/access.2018.2870052. URL: https://doi.org/10.1109%2Faccess.2018.2870052.

Allahyari, Hiva and Niklas Lavesson (2011). "User-oriented assessment of classification model understandability". In: *11th scandinavian conference on Artificial intelligence*. IOS Press.

Alvarez-Melis, David and Tommi S Jaakkola (2018). "On the robustness of interpretability methods". In: *arXiv preprint arXiv:1806.08049*.

Andrews, Robert, Joachim Diederich, and Alan B. Tickle (Dec. 1995). "Survey and critique of techniques for extracting rules from trained artificial neural networks". In: *Knowledge-Based Systems* 8(6), pp. 373–389. DOI: 10.1016/0950-7051(96)81920-4. URL: https://doi.org/10.1016%2F0950-7051%2896%2981920-4.

Apley, Daniel W. and Jingyu Zhu (June 2020). "Visualizing the effects of predictor variables in black box supervised learning models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82(4), pp. 1059–1086. DOI: 10.1111/rssb.12377. URL: https://doi.org/10.1111%2Frssb.12377.

Arrieta, Alejandro Barredo et al. (June 2020). "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58, pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012. URL: https://doi.org/10.1016%2Fj.inffus.2019.12.012.

Bastani, Osbert, Carolyn Kim, and Hamsa Bastani (2017). "Interpreting blackbox models via model extraction". In: *arXiv preprint arXiv:1705.08504*.

Berkson, Joseph (Sept. 1944). "Application of the Logistic Function to Bio-Assay". In: *Journal of the American Statistical Association* 39(227), pp. 357–365. DOI: 10.1080/01621459.1944.10500699. URL: https://doi.org/10.1080%2F01621459.1944.10500699.

Bibal, Adrien and Benoit Frénay (2016). "Interpretability of machine learning models and representations: an introduction." In: *ESANN*.

Biran, Or and Courtenay Cotton (2017). “Explanation and justification in machine learning: A survey”. In: *IJCAI-17 workshop on explainable AI (XAI)*. Vol. 8. 1, pp. 8–13.

Bliss, Chester I (1934). “The method of probits.” In: *Science*.

Boucher, Jean-Philippe, Steven Coté, and Montserrat Guillen (Sept. 2017). “Exposure as Duration and Distance in Telematics Motor Insurance Using Generalized Additive Models”. In: *Risks* 5(4), p. 54. DOI: 10.3390/risks5040054. URL: https://doi.org/10.3390%2Frisks5040054.

Breiman, Leo (Aug. 1996). “Bagging predictors”. In: *Machine Learning* 24(2), pp. 123–140. DOI: 10.1007/bf00058655. URL: https://doi.org/10.1007%2Fbf00058655.

Breiman, Leo (2001a). “Random forests”. In: *Machine learning* 45(1), pp. 5–32.

Breiman, Leo (Aug. 2001b). “Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)”. In: *Statistical Science* 16(3), pp. 199–231. DOI: 10.1214/ss/1009213726. URL: https://doi.org/10.1214%2Fss%2F1009213726.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone (Oct. 1984). *Classification And Regression Trees*. Routledge. DOI: 10.1201/9781315139470. URL: https://doi.org/10.1201%2F9781315139470.

Breiman, Leo and Nong Shang (1996). “Born again trees”. In: *University of California, Berkeley, Berkeley, CA, Technical Report* 1(2), p. 4.

Brockman, M. J. and T. S. Wright (1992). “Statistical motor rating: making effective use of your data”. In: *Journal of the Institute of Actuaries* 119(3), pp. 457–543. DOI: 10.1017/s0020268100019995. URL: https://doi.org/10.1017%2Fs0020268100019995.

Burkart, Nadia and Marco F. Huber (Jan. 2021). “A Survey on the Explainability of Supervised Machine Learning”. In: *Journal of Artificial Intelligence Research* 70, pp. 245–317. DOI: 10.1613/jair.1.12228. URL: https://doi.org/10.1613%2Fjair.1.12228.

Caruana, Rich, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad (Aug. 2015). “Intelligible Models for HealthCare”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2783258.2788613. URL: https://doi.org/10.1145%2F2783258.2788613.

Carvalho, Diogo V., Eduardo M. Pereira, and Jaime S. Cardoso (July 2019). “Machine Learning Interpretability: A Survey on Methods and Metrics”. In: *Electronics* 8(8), p. 832. DOI: 10.3390/electronics8080832. URL: https://doi.org/10.3390%2Felectronics8080832.

Chen, Tianqi and Carlos Guestrin (Aug. 2016). “XGBoost”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145%2F2939672.2939785.

Chipman, Hugh A., Edward I. George, and Robert E. McCulloch (Mar. 2010). “BART: Bayesian additive regression trees”. In: *The Annals of Applied Statistics* 4(1), pp. 266–298. DOI: 10.1214/09-aoas285. URL: https://doi.org/10.1214%2F09-aoas285.

Craven, Mark W and Jude W Shavlik (1996). “Extracting tree-structured representations of trained networks”. In: *Advances in neural information processing systems*, pp. 24–30.

Craven, Mark W. and Jude W. Shavlik (1994). “Using Sampling and Queries to Extract Rules from Trained Neural Networks”. In: *Machine Learning Proceedings 1994*. Elsevier, pp. 37–45. DOI: 10.1016/b978-1-55860-335-6.50013-1. URL: https://doi.org/10.1016%2Fb978-1-55860-335-6.50013-1.

Daykin, C. D., T. Pentikainen, and M. Pesonen (Sept. 1994). “Practical Risk Theory for Actuaries.” In: *Biometrics* 50(3), p. 897. DOI: 10.2307/2532828. URL: https://doi.org/10.2307%2F2532828.

De Graaf, Maartje MA and Bertram F Malle (2017). “How people explain action (and autonomous intelligent systems should too)”. In: *2017 AAAI Fall Symposium Series*.

Denuit, Michel, Donatien Hainaut, and Julien Trufin (2019). *Effective Statistical Learning Methods for Actuaries I*. Springer International Publishing. DOI: 10.1007/978-3-030-25820-7. URL: https://doi.org/10.1007%2F978-3-030-25820-7.

Denuit, Michel, Donatien Hainaut, and Julien Trufin (2020). *Effective Statistical Learning Methods for Actuaries II*. Springer International Publishing. DOI: 10.1007/978-3-030-57556-4. URL: https://doi.org/10.1007%2F978-3-030-57556-4.

Doran, Derek, Sarah Schulz, and Tarek R Besold (2017). “What does explainable AI really mean? A new conceptualization of perspectives”. In: *arXiv preprint arXiv:1710.00794*.

Doshi-Velez, Finale and Been Kim (2017). “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608*.

Elomaa, Tapio (1994). “In defense of C4. 5: Notes on learning one-level decision trees”. In: *Machine Learning Proceedings 1994*. Elsevier, pp. 62–69.

Fisher, Aaron, Cynthia Rudin, and Francesca Dominici (2019). “All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously.” In: *Journal of Machine Learning Research* 20(177), pp. 1–81.

Freitas, Alex A. (Mar. 2014). “Comprehensible classification models”. In: *ACM SIGKDD Explorations Newsletter* 15(1), pp. 1–10. DOI: 10.1145/2594473.2594475. URL: https://doi.org/10.1145%2F2594473.2594475.

Freund, Y. (Sept. 1995). “Boosting a Weak Learning Algorithm by Majority”. In: *Information and Computation* 121(2), pp. 256–285. DOI: 10.1006/inco.1995.1136. URL: https://doi.org/10.1006%2Finco.1995.1136.

Freund, Yoav and Robert E Schapire (Aug. 1997). “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. In: *Journal of Computer and System Sciences* 55(1), pp. 119–139. DOI: 10.1006/jcss.1997.1504. URL: https://doi.org/10.1006%2Fjcss.1997.1504.

Friedman, Jerome H. (2001). “Greedy function approximation: A gradient boosting machine.” In: *The Annals of Statistics* 29(5), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

Friedman, Jerome H. and Bogdan E. Popescu (Sept. 2008). “Predictive learning via rule ensembles”. In: *The Annals of Applied Statistics* 2(3), pp. 916–954. DOI: 10.1214/07-aoas148. URL: https://doi.org/10.1214%2F07-aoas148.

Gao, Guangyuan, Shengwang Meng, and Mario V. Wüthrich (Sept. 2018). “Claims frequency modeling using telematics car driving data”. In: *Scandinavian Actuarial Journal* 2019(2), pp. 143–162. DOI: 10.1080/03461238.2018.1523068. URL: https://doi.org/10.1080%2F03461238.2018.1523068.

Gao, Guangyuan and Mario Wüthrich (Jan. 2019). “Convolutional Neural Network Classification of Telematics Car Driving Data”. In: *Risks* 7(1), p. 6. DOI: 10.3390/risks7010006. URL: https://doi.org/10.3390%2Frisks7010006.

Gao, Guangyuan and Mario V. Wüthrich (Oct. 2018). "Feature extraction from telematics car driving heatmaps". In: *European Actuarial Journal* 8(2), pp. 383–406. DOI: 10.1007/s13385-018-0181-7. URL: https://doi.org/10.1007%2Fs13385-018-0181-7.

Geurts, Pierre, Damien Ernst, and Louis Wehenkel (Mar. 2006). "Extremely randomized trees". In: *Machine Learning* 63(1), pp. 3–42. DOI: 10.1007/s10994-006-6226-1. URL: https://doi.org/10.1007%2Fs10994-006-6226-1.

Goegebeur, Yuri, Armelle Guillou, and Jing Qin (2021). "Extreme value estimation of the conditional risk premium in reinsurance". In: *Insurance: Mathematics and Economics* 96, pp. 68–80. ISSN: 0167-6687. DOI: https://doi.org/10.1016/j.insmatheco.2020.10.010. URL: https://www.sciencedirect.com/science/article/pii/S0167668720301426.

Goldstein, Alex, Adam Kapelner, Justin Bleich, and Emil Pitkin (Jan. 2015). "Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation". In: *Journal of Computational and Graphical Statistics* 24(1), pp. 44–65. DOI: 10.1080/10618600.2014.907095. URL: https://doi.org/10.1080%2F10618600.2014.907095.

Goodman, Bryce and Seth Flaxman (Oct. 2017). "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". In: *AI Magazine* 38(3), pp. 50–57. DOI: 10.1609/aimag.v38i3.2741. URL: https://doi.org/10.1609%2Faimag.v38i3.2741.

Graves, Alex (2012). "Offline Arabic Handwriting Recognition with Multidimensional Recurrent Neural Networks". In: *Guide to OCR for Arabic Scripts*. Springer London, pp. 297–313. DOI: 10.1007/978-1-4471-4072-6_12. URL: https://doi.org/10.1007%2F978-1-4471-4072-6_12.

Grubinger, Thomas, Achim Zeileis, and Karl-Peter Pfeiffer (2014). "evtree: Evolutionary Learning of Globally Optimal Classification and Regression Trees inR". In: *Journal of Statistical Software* 61(1). DOI: 10.18637/jss.v061.i01. URL: https://doi.org/10.18637%2Fjss.v061.i01.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi (Jan. 2019). "A Survey of Methods for Explaining Black Box Models". In: *ACM Computing Surveys* 51(5), pp. 1–42. DOI: 10.1145/3236009. URL: https://doi.org/10.1145%2F3236009.

Haberman, Steven and Arthur E. Renshaw (1996). "Generalized Linear Models and Actuarial Science". In: *The Statistician* 45(4), p. 407. DOI: 10.2307/2988543. URL: https://doi.org/10.2307%2F2988543.

Hara, Satoshi and Kohei Hayashi (2018). "Making tree ensembles interpretable: A bayesian model selection approach". In: *International conference on artificial intelligence and statistics*. PMLR, pp. 77–85.

Hastie, Trevor J and Robert J Tibshirani (1990). *Generalized additive models*. Vol. 43. CRC press.

Henckaerts, Roel, Katrien Antonio, and Marie-Pier Coté (2020). "Model-Agnostic Interpretable and Data-driven suRRogates suited for highly regulated industries". In: *arXiv preprint arXiv:2007.06894*.

Henckaerts, Roel, Marie-Pier Coté, Katrien Antonio, and Roel Verbelen (July 2020). "Boosting Insights in Insurance Tariff Plans with Tree-Based Machine Learning Methods". In: *North American Actuarial Journal*, pp. 1–31. DOI: 10.1080/10920277.2020.1745656. URL: https://doi.org/10.1080%2F10920277.2020.1745656.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Computation* 9(8), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162%2Fneco.1997.9.8.1735.

Hoerl, Arthur E. and Robert W. Kennard (Feb. 1970). “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. In: *Technometrics* 12(1), pp. 55–67. DOI: 10.1080/00401706.1970.10488634. URL: https://doi.org/10.1080%2F00401706.1970.10488634.

Hooker, Giles and Lucas Mentch (2019). “Please stop permuting features: An explanation and alternatives”. In: *arXiv preprint arXiv:1905.03151*.

Hopfield, J. J. (Apr. 1982). “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the National Academy of Sciences* 79(8), pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554. URL: https://doi.org/10.1073%2Fpnas.79.8.2554.

Huysmans, Johan, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens (Apr. 2011). “An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models”. In: *Decision Support Systems* 51(1), pp. 141–154. DOI: 10.1016/j.dss.2010.12.003. URL: https://doi.org/10.1016%2Fj.dss.2010.12.003.

Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu (2017). “Lightgbm: A highly efficient gradient boosting decision tree”. In: *Advances in neural information processing systems* 30, pp. 3146–3154.

Kim, Tae Wan (2018). “Explainable artificial intelligence (XAI), the goodness criteria and the grasp-ability test”. In: *arXiv preprint arXiv:1810.09598*.

Koh, Pang Wei and Percy Liang (2017). “Understanding black-box predictions via influence functions”. In: *International Conference on Machine Learning*. PMLR, pp. 1885–1894.

Kohonen, Teuvo (1982). “Self-organized formation of topologically correct feature maps”. In: *Biological Cybernetics* 43(1), pp. 59–69. DOI: 10.1007/bf00337288. URL: https://doi.org/10.1007%2Fbf00337288.

Krishnan, Maya (Aug. 2019). “Against Interpretability: a Critical Examination of the Interpretability Problem in Machine Learning”. In: *Philosophy & Technology* 33(3), pp. 487–502. DOI: 10.1007/s13347-019-00372-9. URL: https://doi.org/10.1007%2Fs13347-019-00372-9.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (May 2017). “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60(6), pp. 84–90. DOI: 10.1145/3065386. URL: https://doi.org/10.1145%2F3065386.

Laugel, Thibault, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki (2019). “Issues with post-hoc counterfactual explanations: a discussion”. In: *arXiv preprint arXiv:1906.04774*.

Laugel, Thibault, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki (2018). “Defining locality for surrogates in post-hoc interpretablity”. In: *arXiv preprint arXiv:1806.07498*.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel (Dec. 1989). “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1(4), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541. URL: https://doi.org/10.1162%2Fneco.1989.1.4.541.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86(11), pp. 2278–2324. DOI: 10.1109/5.726791. URL: https://doi.org/10.1109%2F5.726791.

Lipton, Zachary C. (Sept. 2018). “The mythos of model interpretability”. In: *Communications of the ACM* 61(10), pp. 36–43. DOI: 10.1145/3233231. URL: https://doi.org/10.1145%2F3233231.

Lorentzen, Christian and Michael Mayer (2020). “Peeking into the Black Box: An Actuarial Case Study for Interpretable Machine Learning”. In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3595944. URL: https://doi.org/10.2139%2Fssrn.3595944.

Lundberg, Scott and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *arXiv preprint arXiv:1705.07874*.

Maillart, Arthur (2021a). “Black box inspection via robustness analysis.” In: *Bulletin Français d’Actuariat*.

Maillart, Arthur (2021b). “Toward an explainable machine learning model for claim frequency : a une case in car insurance pricing with telematics data.” In: *European Actuarial Journal*.

Maillart, Arthur and Christian Robert (2021). “Tail-index partition-based rules extraction.” In: *Working paper*.

Martens, David, Jan Vanthienen, Wouter Verbeke, and Bart Baesens (Nov. 2011). “Performance of classification models from a user perspective”. In: *Decision Support Systems* 51(4), pp. 782–793. DOI: 10.1016/j.dss.2011.01.013. URL: https://doi.org/10.1016%2Fj.dss.2011.01.013.

McCullagh, P. and J.A. Nelder (Jan. 1989). *Generalized Linear Models*. Routledge. DOI: 10.1201/9780203753736. URL: https://doi.org/10.1201%2F9780203753736.

McCulloch, Warren and Walter Pitts (June 1944). “Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133.” In: *Journal of Symbolic Logic* 9(2), pp. 49–50. DOI: 10.2307/2268029. URL: https://doi.org/10.2307%2F2268029.

Meinshausen, Nicolai (Dec. 2010). “Node harvest”. In: *The Annals of Applied Statistics* 4(4), pp. 2049–2072. DOI: 10.1214/10-aoas367. URL: https://doi.org/10.1214%2F10-aoas367.

Miller, Tim (Feb. 2019). “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267, pp. 1–38. DOI: 10.1016/j.artint.2018.07.007. URL: https://doi.org/10.1016%2Fj.artint.2018.07.007.

Minsky, Marvin and Seymour Papert (Aug. 1969). “Perceptrons. An Introduction to Computational Geometry. Marvin Minsky and Seymour Papert. M.I.T. Press, Cambridge”. In: *Science* 165(3895), pp. 780–782. DOI: 10.1126/science.165.3895.780. URL: https://doi.org/10.1126%2Fscience.165.3895.780.

Mittelstadt, Brent, Chris Russell, and Sandra Wachter (Jan. 2019). “Explaining Explanations in AI”. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM. DOI: 10.1145/3287560.3287574. URL: https://doi.org/10.1145%2F3287560.3287574.

Mohseni, Sina, Jeremy E Block, and Eric D Ragan (2018). “A human-grounded evaluation benchmark for local explanations of machine learning”. In: *arXiv preprint arXiv:1801.05075*.

Mohseni, Sina, Niloofar Zarei, and Eric D Ragan (2018). “A multidisciplinary survey and framework for design and evaluation of explainable AI systems”. In: *arXiv preprint arXiv:1811.11839*.

Molnar, Christoph (2018). “A guide for making black box models explainable”. In: *URL: https://christophm. github. io/interpretable-ml-book*.

Molnar, Christoph (2020). *Interpretable machine learning*. Lulu. com.

Molnar, Christoph, Giuseppe Casalicchio, and Bernd Bischl (2020). “Interpretable Machine Learning–A Brief History, State-of-the-Art and Challenges”. In: *arXiv preprint arXiv:2010.09337*.

Nelder, J. A. and R. W. M. Wedderburn (1972). “Generalized Linear Models”. In: *Journal of the Royal Statistical Society. Series A (General)* 135(3), p. 370. DOI: 10.2307/2344614. URL: https://doi.org/10.2307%2F2344614.

Noll, Alexander, Robert Salzmann, and Mario V. Wuthrich (2018). “Case Study: French Motor Third-Party Liability Claims”. In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3164764. URL: https://doi.org/10.2139%2Fssrn.3164764.

Ohlsson, Esbjörn and Björn Johansson (2010). *Non-Life Insurance Pricing with Generalized Linear Models*. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-10791-7. URL: https://doi.org/10.1007%2F978-3-642-10791-7.

Páez, Andrés (May 2019). “The Pragmatic Turn in Explainable Artificial Intelligence (XAI)”. In: *Minds and Machines* 29(3), pp. 441–459. DOI: 10.1007/s11023-019-09502-w. URL: https://doi.org/10.1007%2Fs11023-019-09502-w.

Piltaver, Rok, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić (Nov. 2016). “What makes classification trees comprehensible?” In: *Expert Systems with Applications* 62, pp. 333–346. DOI: 10.1016/j.eswa.2016.06.009. URL: https://doi.org/10.1016%2Fj.eswa.2016.06.009.

Quinlan, J Ross (1992). *C4. 5: programs for machine learning*. Elsevier.

Quinlan, J Ross et al. (1996). “Bagging, boosting, and C4. 5”. In: *Aaai/iaai, Vol. 1*, pp. 725–730.

Ribeiro, Marco, Sameer Singh, and Carlos Guestrin (2016). ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics. DOI: 10.18653/v1/n16-3020. URL: https://doi.org/10.18653%2Fv1%2Fn16-3020.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2018). “Anchors: High-precision model-agnostic explanations”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Rosenblatt, F. (1958). “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological Review* 65(6), pp. 386–408. DOI: 10.1037/h0042519. URL: https://doi.org/10.1037%2Fh0042519.

Rudin, Cynthia (May 2019). “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. In: *Nature Machine Intelligence* 1(5), pp. 206–215. DOI: 10.1038/s42256-019-0048-x. URL: https://doi.org/10.1038%2Fs42256-019-0048-x.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (Sept. 1985). *Learning Internal Representations by Error Propagation*. Tech. rep. DOI: 10.21236/ada164453. URL: https://doi.org/10.21236%2Fada164453.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1988). “Learning Internal Representations by Error Propagation”. In: *Readings in Cognitive Science*. Elsevier, pp. 399–421. DOI: 10.1016/b978-1-4832-1446-7.50035-2. URL: https://doi.org/10.1016%2Fb978-1-4832-1446-7.50035-2.

Ruping, Stefan et al. (2006). “Learning interpretable models”. In.

Schapire, Robert E. (June 1990). “The strength of weak learnability”. In: *Machine Learning* 5(2), pp. 197–227. DOI: 10.1007/bf00116037. URL: https://doi.org/10.1007%2Fbf00116037.

Schneider, Johannes, Joshua Handali, Michalis Vlachos, and Christian Meske (2020). “Deceptive ai explanations: Creation and detection”. In: *arXiv preprint arXiv:2001.07641*.

Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis (July 2008). “Conditional variable importance for random forests”. In: *BMC Bioinformatics* 9(1). DOI: 10.1186/1471-2105-9-307. URL: https://doi.org/10.1186%2F1471-2105-9-307.

Strobl, Carolin, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn (Jan. 2007). “Bias in random forest variable importance measures: Illustrations, sources and a solution”. In: *BMC Bioinformatics* 8(1). DOI: 10.1186/1471-2105-8-25. URL: https://doi.org/10.1186%2F1471-2105-8-25.

Strumbelj, Erik and Igor Kononenko (2010). “An efficient explanation of individual classifications using game theory”. In: *The Journal of Machine Learning Research* 11, pp. 1–18.

Tibshirani, Robert (Jan. 1996). “Regression Shrinkage and Selection Via the Lasso”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1), pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x. URL: https://doi.org/10.1111%2Fj.2517-6161.1996.tb02080.x.

Ustun, Berk and Cynthia Rudin (Nov. 2015). “Supersparse linear integer models for optimized medical scoring systems”. In: *Machine Learning* 102(3), pp. 349–391. DOI: 10.1007/s10994-015-5528-6. URL: https://doi.org/10.1007%2Fs10994-015-5528-6.

Verbelen, Roel, Katrien Antonio, and Gerda Claeskens (Apr. 2018). “Unravelling the predictive power of telematics data in car insurance pricing”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67(5), pp. 1275–1304. DOI: 10.1111/rssc.12283. URL: https://doi.org/10.1111%2Frssc.12283.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). “Counterfactual explanations without opening the black box: Automated decisions and the GDPR”. In: *Harv. JL & Tech.* 31, p. 841.

Woodward, James (1979). “Scientific Explanation”. In: *The British Journal for the Philosophy of Science* 30(1), pp. 41–67. ISSN: 00070882, 14643537. URL: http://www.jstor.org/stable/687416.

Wuthrich, Mario V. and Christoph Buser (2017). “Data Analytics for Non-Life Insurance Pricing”. In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.2870308. URL: https://doi.org/10.2139%2Fssrn.2870308.

Wüthrich, Mario V. (Apr. 2017). “Covariate selection from telematics car driving data”. In: *European Actuarial Journal* 7(1), pp. 89–108. DOI: 10.1007/s13385-017-0149-z. URL: https://doi.org/10.1007%2Fs13385-017-0149-z.

2

Toward an explainable machine learning model for claim frequency : a use case in car insurance pricing with telematics data

> *We're drowning in information and starving for knowledge.*
>
> — **Rutherford D. Rogers**

Abstract

In this paper, we suggest an explainable machine learning approach to model the claim frequency of a telematics car dataset. In fact, we use a data-driven method based on tree ensembles, namely, the random forest, to create a claim frequency model. Then, we present a method to build a tree that faithfully synthesizes the predictions of a tree ensemble model such as those derived from the random forest or gradient boosting. This tree serves as a global explanation of the predictions of the black-box. Thanks to this surrogate model, we can extract knowledge from a black-box tree ensemble model. Then, we provide an application to improve the performance of a generalized linear model. Indeed, we integrate this new knowledge into a generalized linear model to increase the predictive power.

Keywords: Pay-as-you-drive insurance, Usage-based insurance, Risk classification, Explainability, Interpretability, Random forest, Gradient boosting, Black-box model explanation

2.1 Introduction

For many years, the actuary's understanding of risk was restricted to a set of simple and fairly static variables such as gender, license seniority, age, areas or vehicle information. Although this approach already provides differentiated pricing, risk assessment can be improved with telematics. Typically, trips are recorded by sequences of GPS coordinates for such data. These coordinates are often enriched with other data collected at the same frequency such as the engine velocity, for example.

Alternatively, trips can be stored as metadata. For example, these metadata can be the number of kilometers driven, the time spent in a speed zone, and the kilometers driven on a road. GPS coordinates are more suited to pay-how-you-drive (PHYD) insurance, while metadata is preferred in pay-as-you-drive (PAYD) insurance. While PHYD insurance focuses on driving style, PAYD insurance focuses on driving habits. The first one requires more details and is outside the scope of our study. We will focus on PAYD insurance and the behavior of policyholders.

Telematics have improved the understanding of risk exposure. Indeed, before its use, actuaries could only measure it with duration of observation. The time exposure characterizes the intra-annual coverage period of a policyholder. However, a policyholder may be covered without driving regularly. Therefore, the insurer carries a different risk. This idea, initially developed by Vickrey (1968), has since been supported by numerous studies. These include Tselentis et al. (2016), Boucher et al. (2017), and Verbelen et al. (2018) more recently. All of these works highlight the potential of telematics data for PAYD insurance. The generalized linear model (GLM) is most often applied to gain insight of telematics data. It is still widely used in the research community and in insurance companies because it is easy to interpret and has been extensively studied in the actuarial literature. However, it is not well suited to these data. Indeed, when the relationship between the variables and the target (claim frequency), for example, is not linear, it becomes necessary to discretize the continuous variables; see, e.g., Henckaerts, Antonio, et al. (2018), Ohlsson and Johansson (2010), and Frees et al. (2016). There are some procedures to automate variable splitting. Nevertheless, expert judgment is most often used in companies. However, we have little experience with telematics data. Therefore, it seems difficult to discretize the variables without prior knowledge. Furthermore, nothing guarantees the optimality of the breaks. In addition, the embedded devices produce many variables. It can be hard to manage in day-to-day company business.

In contrast, machine learning algorithms manage continuous variables, interaction effects and colinear predictors very well. This makes them good candidates to model a claim frequency problem with these data. Unfortunately, the latter are often difficult to interpret. Authors interested in these methods generally focus only on performance; see, e.g., Yang et al. (2017), Wuthrich and Buser (2017), and Noll et al. (2018). The first works to increase intelligibility in actuary concentrate on methods that provide a local perspective (LIME in Ribeiro et al. (2016) and SHAP in Lundberg and Lee (2017)) or too little information to obtain an overview of the model (variable importance Breiman (2001) or partial dependence plots J. H. Friedman (2001) or SHAP). In Henckaerts, Coté, et al. (2020), a methodology is proposed to gain an insight into machine learning models.

In this paper, we introduce a method to provide an explanation for a tree-based black-box such as the random forest (RF) or gradient boosting (GBM). The explanation is global, which means here that it explains the entire behavior of the model in the form of a regression tree. Thanks to this tree, we can extract knowledge from a tree-based model and then inject this new knowledge in a generalized linear model to boost its performance. To do so, we propose to adapt DefragTrees, a method initially developed by Hara and Hayashi (2016) and then extended in Hara and Hayashi (2018). The latter

suggests defragging the highly fragmented covariate space generated by the tree ensemble model to produce a parsimonious set of rules.

The article is organized as follows. In Sect. 2.2, we present Poisson regression trees, which are the basic elements of our frequency models. Then, we fit and compare different claim frequency models based on tree ensemble methods to our data. We also fit a reference GLM to benchmark performances and hence measure the increase in performance due to telematics variables. In Sect. 2.3, we develop theoretical elements of DefragTrees. This method allows us to obtain a faithful tree from the tree ensemble model. In addition, we suggest an alternative method to generate outputs as a tree and not as rules. We apply the method on the black- box predictions of our fitted claim frequency models and analyze extracted knowledge. Finally, we integrate this new knowledge into our benchmark GLM to improve its performance.

2.2 Machine learning approach for frequency modeling

Nonlife insurance is mainly based on the law of large numbers. Indeed, it is extremely difficult to predict whether a policyholder will have a claim over a year. However, it is easier to provide a good approximation of the portfolio's total loss within a year. This idea makes it possible to define the pure premium as the expected amount that the insurer will have to pay for the risk transferred to it. In practice, since crash conditions depend on the policyholder, insurers seek to refine their ratings by segmenting their population according to the frequency and intensity of claims. Hence, they can provide a rating that is more representative of the policyholder's risk. Of course, the challenge of this segmentation is to create subgroups that are broad enough to apply the law of large numbers and sufficiently homogeneous so that policyholders belong to an identifiable class of risk. This is what motivates the construction of statistical models for pricing.

2.2.1 The basics of pricing in car insurance

A classical way to estimate pure premium is to consider a frequency/severity model. In this context, we must fit two models, one for the claim severity and the other for the claim frequency. Here, we only focus on the second one. We denote by N_i and N, respectively, the number of claims recorded for the individual i over the observation period ω_i and the total number of claims over the total observation period ω. The observation period is also called time exposure. It allows individuals to be weighted according to their observation time. Hence, uncertainty decreases as the exposure increases.

We want to model the ratio $Y = N/\omega$ according to the variables available in the database, i.e., we want to fit a model to estimate $E[Y|X = x]$. We will assume that each N_i has a Poisson distribution. The choice of a Poisson distribution is motivated by the fact that the claims process is a Poisson process. See Ohlsson and Johansson (2010) and Beard et al. (1984) for more details. Therefore, we

assume that there exists an expectation of the claim frequency denoted by $\lambda_i > 0$ and an exposure $\omega_i > 0$ such that:

$$\forall k \in \mathbb{N}, P\left[N_i = k\right] = e^{-\lambda_i \omega_i} \frac{(\lambda_i \omega_i)^k}{k!}.$$

Remark 6 *We work under the assumption that policies are independent. Hence, at an aggregated level, in a tariff cell, for example, the number of claims still has a Poisson distribution.*

In this section, we try two different approaches to fit a claim frequency model. The first one is classical. We fit a parametric model, namely, a Poisson GLM to the data. The second one is a data-driven approach. In this case, we fit a nonparametric tree ensemble model to the data. We need a metric to compare these models with each other. Therefore, we recall the definition of the mean Poisson deviance loss as follows :

$$D(N, \lambda) = \sum_{k=1}^{n} 2N_k \left[\frac{\lambda\omega_k}{N_k} - 1 - \log\left(\frac{\lambda\omega_k}{N_k}\right)\right]$$

where n is the number of observations and the k-th term is set to $2\lambda_k\omega_k$ if $N_k = 0$. The interested reader can refer to chapter 1 of Wuthrich and Buser (2017).

2.2.2 Use case in telematics car insurance pricing

Since we want to show the contribution of telematics data in assessing claim frequency, we divide our dataset into two parts. The first one contains only the policy variables, while the second one contains the policy variables and the telematics variables. **Tab. 2.7** in Appendix 2.5.2 describes the available variables. We train the random forest and gradient boosting methods on the first dataset and a GLM to create a reference score. This provides a benchmark of methods before the contribution of telematics data. Then, we train the random forest and gradient boosting methods on the second dataset. This allows us to measure the contribution of telematics data. Subsequently, by extracting the knowledge acquired by the random forest, we will be able to strengthen our GLM model.

Remark 7 *Since we have the number of kilometers driven by the policyholder, we have the choice of the exposure variable.* ***Fig. 2.25*** *in Appendix 2.5.2 shows the claim number average by buckets of width of* 0.05 *year. As we can see, the relationship between the average number of claims and the duration seems to be linear* ($R^2 = 0.92$)*. In contrast, if we choose the kilometers by buckets of* $5,000$ *kilometers in* ***Fig. 2.26*** *in Appendix 2.5.2, the relationship is not linear. Those results are consistent with Boucher et al. (2017). We decide to set the time as exposure because we do not want to introduce another*

difficulty at this step. We leave the impact analysis of a change in exposure variable for a future study. The interested reader can refer to Boucher et al. (2017) and Verbelen et al. (2018) for more detailed articles on time and distance as exposure variables.

2.2.3 Exploratory data analysis

The dataset comes from a Belgian portfolio. It contains motor third-party liability (MTPL) insurance data. This insurance is mandatory and covers any damage caused to a third party. The MTPL insurance we are studying here is specific since the data were collected on a portfolio of very young policyholders. See **Fig. 2.1** and **Fig. 2.2**.

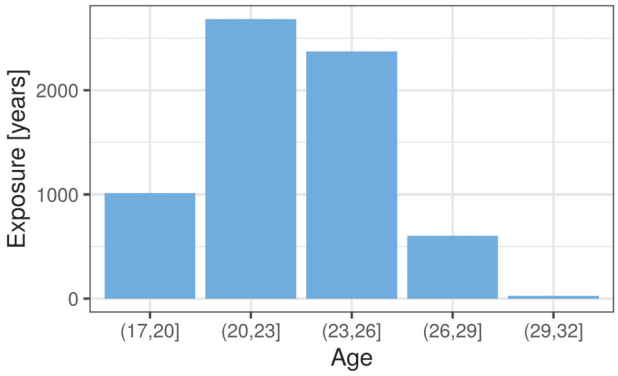

Fig. 2.1: Exposure by ages

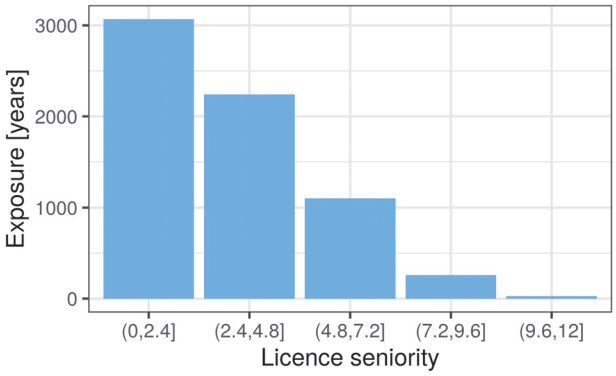

Fig. 2.2: Exposure by license seniority

Indeed, this population is considered risky and is often assigned very high premium levels. To encourage these individuals to set an embedded device in their vehicle, the insurer offered discounted premiums. Nevertheless, it is worth noting that policyholders knew the data would not be used to change their future premium. The embedded device in the vehicle permanently collects statistics on the driver's habits, particularly on the roads used, the kilometers driven and the time slots of the trips. However, it does not collect information on driving style such as acceleration, speed, braking frequency or intensity.

Tab. 2.1: Reported claim rates

Claims	Number of policies	Years at risk	Driven distance (x 10 000 km)
0	11967	6337.53	9412.14
1	565	345.68	572.67
2	23	16.29	29.77

The data collected by the embedded device are preprocessed by the data provider. We did not have access to the raw data but only to an aggregated version. The latter contains the kilometers driven by the policyholder and the details concerning the type of roads and the durations of all the trips. These data were linked by the insurer to policy variables such as `Age`, `Gender`, `License seniority` and the reported claims number. Policyholders are observed for a maximum of one year. If the observed

period is less than a year, it means that something evolved in their policy, for example, a change of residence, vehicle or guarantee. There are 7 levels of coverage available in the data. We decide to focus on the minimal mandatory guarantee because this is the most represented. By doing so, we end up with a dataset containing $4,718$ policyholders observed between January 1, 2010 and December 31, 2015. It represents a total of $6,700$ policy years observed or 100 million kilometers driven. For these $12,555$ observations, only 611 claims were reported at fault (see **Tab. 2.1** for details). It means that considering the time as exposure, the whole portfolio has a claim rate of 9.12%.

In **Fig. 2.3**, we illustrate the observed marginal log claim frequencies of the continuous policy features. We can see that these variables have a nonlinear relationship with the claim frequency. Based on these graphs, we can observe that the most informative feature is `License seniority`. In the policy variable subset, we have two categorical variables `Gender` and `Area`. Globally, men tend to have a higher claim frequency than women. In fact, the observed claim frequency for men is 10.46% and 7.55% for the women. The variable `Area` does not exist in the dataset; however, we build it to give sense to GPS coordinates. Hence, we transform the GPS coordinates of the insured home cities with a variable containing eleven modalities. These modalities represent the 10 provinces of Belgium and the capital Brussels, which is treated separately. We represent the observed claim frequency inside this administrative division of Belgium in **Fig. 2.4**.

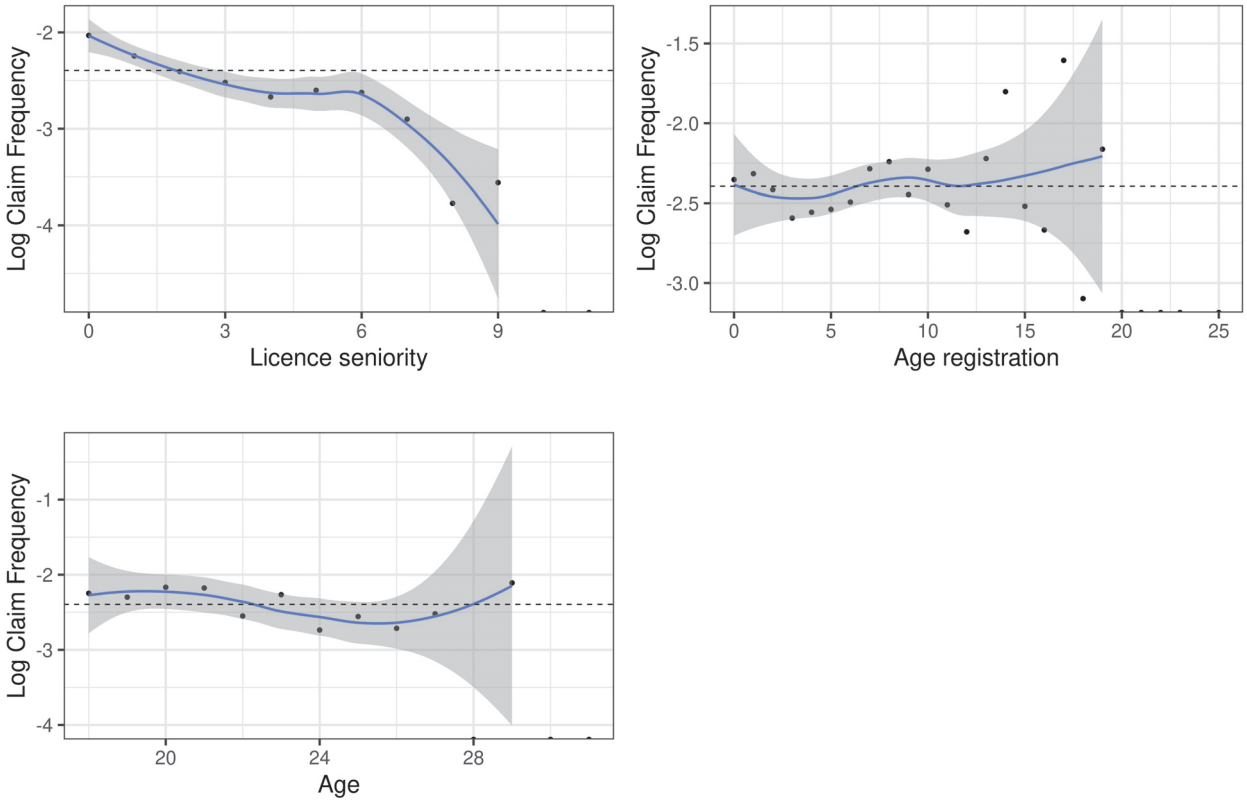

Fig. 2.3: Observed log claim frequencies.

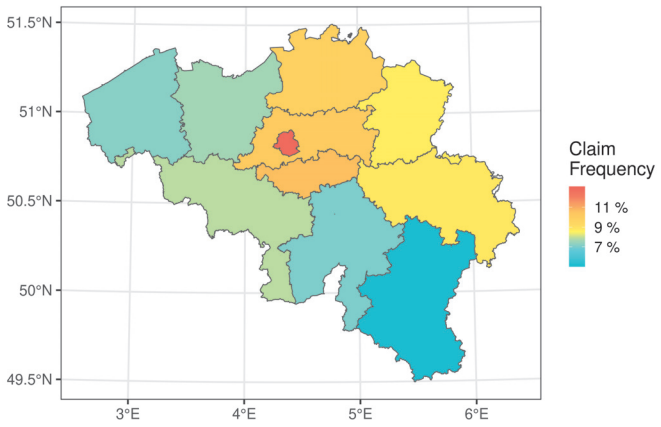

Fig. 2.4: Claim frequency map

2.2.4 Encoding variables and fitting models

GLM and tree-based methods require different data encoding for numeric variables. Indeed, GLM cannot handle nonlinearities efficiently. This is why, in practice, continuous variables are discretized. In contrast, tree-based methods handle numerical variables very well, whether linear or not. It may be necessary to scale the numerical variables to help the gradient descent of the GBM, for example. As far as categorical variables are concerned, both the GLM and the tree-based methods have a mechanism to manage them. Thus, we can integrate them without preprocessing them.

Parametric approach: GLM

To discretize our variables, we fit a Poisson regression tree for each variable to discretize. The target variable is the claim number, the offset is the log of the exposure variable and the explanatory variable is the variable to discretize. We impose that the tree does not exceed two levels of depth and set the complexity parameter to $cp = 0.0001$. Hence, the Poisson regression tree is forced to grow until a maximum of four leaves. We do this because we have very few data points, so we do not want too many modalities by variable in our GLM to avoid overfitting. Therefore, we do not want too many leaves in our tree. The `Age registration` variable is not retained in the rest of the analysis because it leads to a discretization that is too thin (i.e., with too many levels). This is due to a low predictive power of this variable. We end up with the variables presented in **Tab. 2.2**.

To select the best GLM model, we exhaustively search among the possible models with our variables subset. The resulting model is called `GLM policy` in **Tab. 2.3**.

Tab. 2.2: Description of the policy variables for the GLM and their modalities after discretization

Variable	Description	Modalities
`Age`	Driver's age	(18, 24], (24, 32]
`License seniority`	The duration since the license was obtained	(0, 2.6], (2.6, 7.5], (7.5, 12]
`Area`	The provinces of Belgium and the capital Brussels	Antwerp, East Flanders, Flemish Brabant, Limburg, West Flanders, Hainaut, Liège, Luxembourg, Namur, Walloon Brabant, Brussels-Capital Region
`Fuel`	Fuel type	diesel, petrol
`Gender`	Gender of the insured	male, female
`Kwatt`	Power of the vehicle	(21, 42] (42, 160]

Tab. 2.3: Best GLM models with policy variables

	`GLM policy`
(Intercept)	$-2.29\ (0.15)^{***}$
Gender male	$0.32\ (0.08)^{***}$
Flemish Brabant	$0.24\ (0.22)$
Walloon Brabant	$0.16\ (0.21)$
West Flanders	$-0.24\ (0.19)$
East Flanders	$-0.14\ (0.19)$
Hainaut	$-0.24\ (0.15)$
Liège	$0.04\ (0.16)$
Limburg	$-0.12\ (0.21)$
Luxembourg	$-0.61\ (0.33)$
Namur	$-0.34\ (0.22)$
Brussels-Capital	$0.26\ (0.20)$
License seniority [2.6, 7.5)	$-0.38\ (0.09)^{***}$
License seniority [7.5, 12)	$-1.40\ (0.42)^{**}$
AIC	4895.89
BIC	5014.89
Log-likelihood	-2431.94
Deviance	3673.77
5-fold strat. CV - Poisson deviance Loss	0.2952532

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$

2.2.5 Nonparametric approach: tree-based models

The random forest proposed by Breiman (2001) and boosted trees, such as gradient boosting J. H. Friedman (2001) and XGBoost Chen and Guestrin (2016), rely on the same weak learner: the regression tree. Nevertheless, in car insurance pricing, the target is a counting variable; therefore, it may be useful to adapt these methods to a Poisson distribution. The Poisson regression tree is implemented in the rpart package Therneau, Atkinson, et al. (2015). A comprehensive study of Poisson tree ensemble methods is proposed by Wuthrich and Buser (2017). The methods we are interested in use Poisson regression trees as base learners, which is not particularly common. This is why we develop hereafter some theoretical elements.

2.2.6 Poisson regression trees

Let $\boldsymbol{x_i} = (x_i^{(1)}, \dots, x_i^{(p)})$ be a vector of characteristics for an individual i with $\boldsymbol{x_i} \in \mathcal{X} = \mathbb{R}^p$. We assume all variables are numerical and $y_i \in \mathcal{Y} = \mathbb{R}^+$ is our numerical target. If this is not the case, we can encode our variables to bring us back to this case. As mentioned above, we assume that the number of claims for each observation $(N_i, \boldsymbol{x_i}, \omega_i)$, $i = 1, .., n$ follows a Poisson law with expectation:

$$E[N_i] = f(\boldsymbol{x_i})\omega_i$$

where $f : \mathcal{X} \longrightarrow \mathcal{Y}$ is a regression function to be determined. Regression trees do not make assumptions about the structure of the function f. Indeed, the regression tree recursively splits the variable space $\mathcal{X}$ into I regions $(R_i)_{i\in\{1,..,n\}}$. A region can be seen as a p-dimensional rectangle of the space. These rectangular regions form a partition of $\mathcal{X}$, meaning that $\forall i \neq i', R_i \bigcap R_{i'} = \emptyset$ and $\bigcup_{i=1}^{I} R_i = \mathcal{X}$. For each region, a frequency λ_i is estimated and will be the predicted value by the tree in that region. By denoting $\mathcal{R} = \{R_i\}_i^I$ and $\Lambda = \{\lambda_i\}_i^I$, it becomes possible to define a regression function over the entire space:

$$f(\boldsymbol{x_i}; \mathcal{R}, \Lambda, I) = \left\{ \begin{array}{l} \mathcal{X} \longrightarrow \mathcal{Y} \\ \boldsymbol{x_i} \longrightarrow \sum_{i=1}^{I} \lambda_i \mathbb{1}_{\boldsymbol{x_i} \in R_i}. \end{array} \right.$$

The Poisson regression tree uses a different objective function from the classical regression tree described by Breiman et al. (1984) to perform its splits. Each node of a tree is associated with a p-dimensional rectangular cell. At each node R_l, the algorithm looks for the variable and the threshold $z \in \mathbb{R}$ that minimize the Poisson deviance resulting from the split:

$$\min_{(j,z)\in\mathcal{C}_{R_l}}\left[\min_{\lambda_0\geq 0}\sum_{\{k=1,..,n\ s.t\ \boldsymbol{x_k}\in R_l,\ x_k^{(j)}<z\}} D^*(N_k,\lambda_0)+\min_{\lambda_1\geq 0}\sum_{\{k=1,..,n\ s.t\ \boldsymbol{x_k}\in R_l,\ x_k^{(j)}\geq z\}} D^*(N_k,\lambda_1)\right]$$

where j represents the index of the split variable at node R_l, $\mathcal{C}_{R_l}$ the set of all possible cuts in node R_l and $D^*(N_k,\lambda)$ is the Poisson deviance loss for a single observation. It can be expressed as:

$$D^*(N_k,\lambda)=2(l_N(N_k)-l_N(\lambda))$$

with the log-likelihoods $l_N(\lambda)=\sum_k\left[-\lambda\omega_k+N_k\log\left(\lambda\omega_k\right)-\log(N_k!)\right]$ and $l_N(N_k)=-\lambda\omega_k+N_k\log\left(\lambda\omega_k\right)-\log(N_k!)$. The region R_l is split into two subregions $R_{l,0}=\{\boldsymbol{x}\in R_l\ s.t\ x^{(j)}<z\}$ and $R_{l,1}=\{\boldsymbol{x}\in R_l\ s.t\ x^{(j)}\geq z\}$ whose claim frequencies $\lambda_{l,0}$ and $\lambda_{l,1}$, respectively, can be calculated explicitly by maximum likelihood methods. Let $\tau=\{0,1\}$; then:

$$\lambda_{l,\tau}=\underset{\lambda_\tau\geq 0}{\arg\min}\sum_{\{k=1,..,n\ s.t\ \boldsymbol{x_k}\in R_{l,\tau}\}} D^*(N_k,\lambda_\tau)=\frac{\sum_{\{k=1,..,n\ s.t\ \boldsymbol{x_k}\in R_{l,\tau}\}}N_k}{\sum_{\{k=1,..,n\ s.t\ \boldsymbol{x_k}\in R_{l,\tau}\}}\omega_k}.$$

It is possible to view this problem as the search for the split that minimizes the sum of deviances $D^*_{R_{l,0}}(N,\lambda_{l,0})+D^*_{R_{l,1}}(N,\lambda_{l,1})$, where the Poisson deviance loss inside the regions $R_{l,\tau},\ \tau=\{0,1\}$ generated by the split is:

$$D^*_{R_{l,\tau}}(N,\lambda_{l,\tau})=\sum_{\{k=1,..,n\ s.t\ \boldsymbol{x_k}\in R_{l,\tau}\}} 2N_k\left[\frac{\lambda_{l,\tau}\omega_k}{N_k}-1-\log\left(\frac{\lambda_{l,\tau}\omega_k}{N_k}\right)\right]$$

where the right-hand side is set equal to $2\lambda_{l,\tau}\omega_i$ if $N_i=0$.

2.2.7 Fitting tree ensemble models

Multiple packages for random forest and gradient boosting exist. For example, gradient boosting is implemented in XGBoost, CatBoost, LightGBM, h2o GBM or gbm (R packages). Each of these packages has its own subset of specific features but relies on an aggregation of tree learners. In this article, we focus on claim frequency modeling, and we want to implement it using a Poisson distribution with tree-based methods. We also require an implementation that handles an offset variable. Hence, we choose the R packages RfCountData[1] for random forest and h2o.gbm (package h2o) for gradient boosting.

[1] https://github.com/fpechon/rfCountData

For the random forest and the gradient boosting, we replace the values of numerical variables by their rank divided by the number of rows in the dataset. Hence, each predictor is between 0 and 1. The preprocessing helps tree ensemble methods to learn and is useful for the extraction method. Furthermore, it partially anonymizes data and results. We do not apply other preprocessing since these methods have a mechanism to deal with nonlinearity. Moreover, the implementations we have chosen automatically manage categorical variables. As with GLM, we transform our GPS coordinates into a variable indicating in which administrative region of Belgium the policyholder lives.

Hyperparameter search for machine learning models

Telematics data are fairly recent and often have a high number of predictors. It is therefore interesting to use tree-based methods to explore them and extract new knowledge. Indeed, tree-based method learning mechanisms handle interactions between predictors and make it possible to process a large volume of predictors without being sensitive to the correlation between them. Moreover, in most cases, these methods require few hyperparameters to be tuned for performance. In our case, however, we will perform a grid search for hyperparameters. This consists of defining a set of possible values for each hyperparameter we want to tune and then testing all possible combinations. This means that the number of models to test is equal to the product of the number of elements of each set. This will help us to obtain good models with respect to our metric. **Tab. 2.8** and **Tab. 2.9** in Appendix 2.5.3 indicate that we have to test, respectively, 576 random forest models and 2520 gradient boosting models.

2.2.8 Evaluation setup

We evaluate the models with the mean Poisson deviance loss. To select the best model among those proposed on the grids, we perform a 5-fold stratified cross validation. The Poisson deviance is evaluated on the holdout samples. Chapter 1 of Wuthrich and Buser (2017) explains the procedure in detail. Each model of the grids, including the reference GLM, is evaluated with this process. First, we apply this methodology on the dataset containing only policy variables to find policy models. Subsequently, we evaluate with the same principle only the random forest and gradient boosting on the dataset containing policy variables and telematics variables. The results are shown in **Tab. 2.4**. `RF policy` has a lower performance than `GLM policy` but `GBM policy` is slightly better than `GLM policy`. With few numerical variables, tree-based methods do not perform well. This can partially explain why the tree-based methods do not systematically outperform `GLM policy`. In the following section, after presenting the surrogate extraction method, we will display the rules learned by the model. We observe that with more explanatory variables, tree-based models are significantly better than the reference GLM. There is therefore a signal in the telematics variables, and it is easily captured by tree-based models. Once again, we will use the surrogate extraction method to display the new knowledge acquired by tree-based models.

Remark 8 *In Tab. 2.4, we only display the parameters that override the default parameters.*

Tab. 2.4: Table of allowed hyperparameters for grid search.

Model name	Description	Hyperparameters	5-fold strat. CV ($\times 10^{-2}$)
`GLM policy`	Reference GLM fitted exclusively on policy variables	None	29.52532
`RF policy`	A random forest fitted exclusively on policy variables	`nodesize` $= 1200$, `ntree` $= 50$, `maxnodes` $= 32$, `mtry` $= 1$	29.57799
`GBM policy`	A gradient boosting machine fitted exclusively on policy variables	`learn_rate_opt` $= 0.01$, `max_depth_opt` $= 5$, `sample_rate_opt` $= 0.8$, `min_rows` $= 2000$, `col_sample_rate_opt` $= 0.3$, `ntree` $= 400$	29.51324
`RF policy + telematics`	A random forest fitted on policy and telematics variables	`nodesize` $= 800$, `ntree` $= 30$, `maxnodes` $= 12$	28.98369
`GBM policy + telematics`	A gradient boosting machine fitted on policy and telematics variables	`learn_rate_opt` $= 0.03$, `max_depth_opt` $= 1$, `sample_rate_opt` $= 0.9$, `min_rows` $= 2500$, `col_sample_rate_opt` $= 0.8$, `ntree` $= 400$	28.84483

2.2.9 Limitations of existing explainability methods

Before describing the surrogate extraction method, we present the limitations of the currently used methods to interpret the models. We only introduce the most commonly used methods in insurance companies. Now, let us take a look at the most widely used method among practitioners, the so-called variable importance. Different definitions of the variable importance plot exist. We use the one that consists of fitting a tree-based model and then for each variable, randomly swap values for each observation and measure the difference between the mean Poisson deviance loss before and after the perturbation. The latter gives us a first idea of the variables that seem relevant to the model. Unfortunately, it does not provide information about the influence of predictors on the outcome of the black-box. For example, it is impossible to know the impact of a change in a predictor's value. We

present the variable importance plots for the random forest fitted on policy variables in Fig. 2.5 and on policy variables and telematics variables in Fig. 2.6.

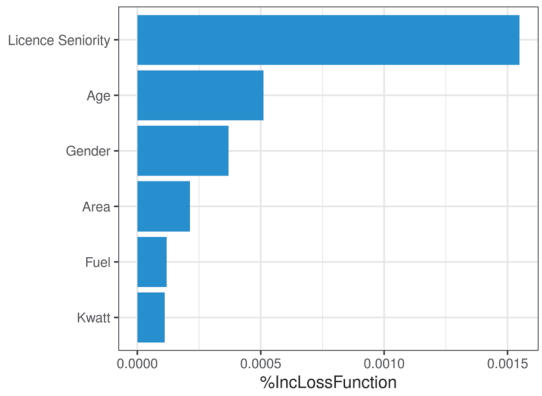

Fig. 2.5: Variable importance for `RF policy`

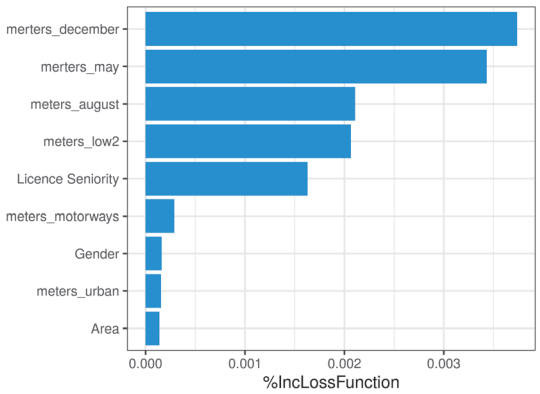

Fig. 2.6: Variable importance for `RF policy + telematics`

The variable importance plot of `RF policy` is consistent with our exploratory data analysis. The most influential predictors are `License seniority`, `Age`, `Gender` and `Area`. Considering now the `RF policy + telematics` variable importance plot, we see that among the 9 most influential predictors, `RF policy + telematics` mostly relies on telematics variables but we cannot tell more from these plots.

Let us continue the analysis with a second common analysis tool: the partial dependence plot (PDP). It was first described in J. H. Friedman (2001). In its univariate version, all variables are fixed except one. A first dataset is created by replacing all the values of this variable of interest by the first value of a range, often the min and max of this variable. Then, the black-box model predicts on this dataset, and the mean of the responses is taken. The process is repeated with the next values of the range. Thus, it is possible to visualize the mean response of the black-box model as a function of this variable. We represent these plots for `License seniority` as a variable of interest for our two fitted random forests in Fig. 2.7 and Fig. 2.8. We are seeing an already observed effect on the data. The claim frequency decreases with `License seniority` on both plots. It is consistent with our univariate analysis and our general knowledge in motor insurance, but this cannot help us to display a clear decision chain from our two random forest models. We could have represented the response of our black-box models based on the crossover of two variables, but we could not have gone much further in understanding these models. Indeed, if we want to be exhaustive, we would need p graphs from the univariate point of view and $p(p-1)/2$ from the bivariate point of view. It means 28 graphs for the simple model on policy variables and 1891 graphs on policy and telematics variables. It quickly becomes prohibitive because it is not parsimonious. Furthermore, we cannot represent more than two variables all at once.

Another criticism that can be made of PDPs is that they mask the variability associated with the remaining variables. Indeed, our random forest models predict not only with the `License seniority` but also with the rest of the "frozen" predictors. The ICEplot method from Goldstein et al. (2015)

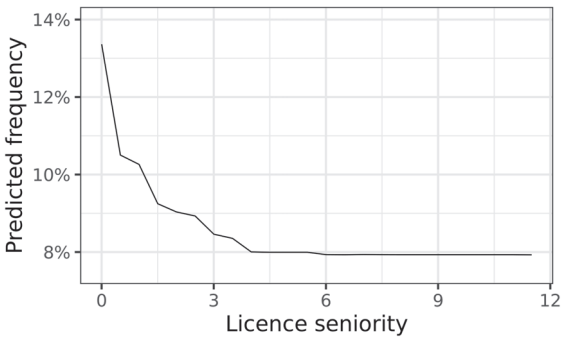

Fig. 2.7: Partial dependence plot of `RF policy`

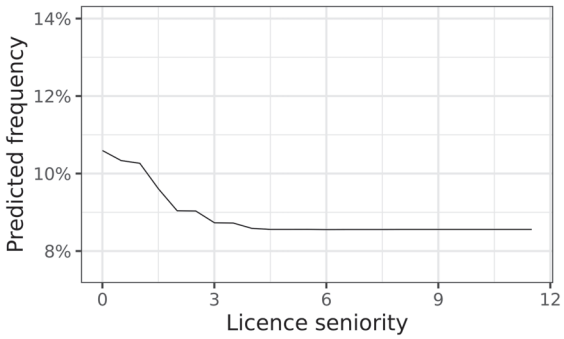

Fig. 2.8: Partial dependence plot of `RF policy + telematics`

tries to add some supplemental information. However, it suffers from the same parsimony problem as the PDPs. Therefore, PDPs complement the variable importance. Nevertheless, the information provided by this method is still very incomplete.

From this, we conclude that currently used methods in insurance companies bring us insights on our models but are not sufficient. Indeed, they fail to extract a transparent and parsimonious set of rules describing the tree-based model's variations. From now on, we will focus on clarifying the main rules identified by the fitted Poisson random forest.

2.3 Extracting a surrogate

A practical application of Poisson random forests has been developed in Sect. 2.2.2. At this point, how can we explain the fitted tree-based model? The problem we are trying to solve is a specific approach to the larger "open the black-box problem". We begin this section by presenting the general problem of machine learning model interpretability and explain our specific approach. Then, we present the necessary theoretical elements of the surrogate extraction method.

2.3.1 Basic notions of intelligibility

According to Guidotti et al. (2019) and Doshi-Velez and Kim (2017), the word "interpret" in machine learning means to explain or to provide the meaning of some concept in understandable terms to a human. An explanation is then an interface between humans and the machine that is an accurate and comprehensible proxy of the black-box. In practice, each community has its own definition of explanation. This means that an explanation can take a wide variety of forms depending on how the models are used. For example, in computer vision, the explanation often takes the form of an image on which red or green pixels are overlaid according to their contribution to the prediction. In contrast, for medical decision support applications, physicians want to understand the full reasoning

behind the model. The explanation is therefore often sought in the form of weights, rules or trees. For pricing actuaries, it is important to have a thorough understanding of the pricing model as the impact of a poorly priced product could be significant for the company. According to each definition of explanation, four different approaches have emerged to tackle the "open the black-box problem". They are summarized in **Fig. 2.9**.

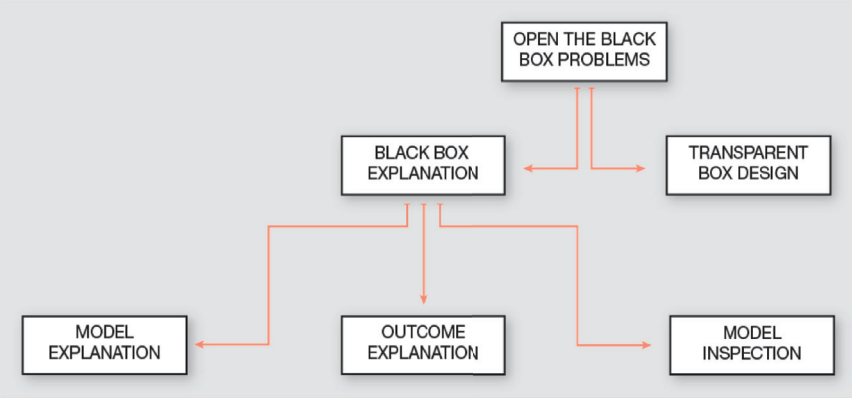

Fig. 2.9: Different machine learning model intelligibility approaches, Source: Guidotti et al. (2019)

The method presented below is used to answer either the "black-box model explanation problem" or the "transparent box design problem". We only deal with the first one here. Thus, we look for a surrogate model (our explainer) that reproduces as much as possible the predictions of our black-box Poisson tree-based model. From the definitions above, the surrogate model has to fulfill two objectives:

- be understandable by humans. In our case, it is equivalent to setting a limited number of regions denoted by K. This will be represented by a decision tree with only K leaves.

- be faithful to the black-box Poisson tree-based model. In machine learning model intelligibility, this is called fidelity. We want to measure this fidelity to be confident about the quality of the explanation. If we denote by f_{bb} and f_{sur}, respectively, the black-box regressor and its surrogate and $\bar{y}_{bb}$ the mean of the black-box response, we evaluate the fidelity as:

$$fidelity = 1 - \frac{\sum_{i=1}^{n} \left(f_{bb}(\boldsymbol{x_i}) - f_{sur}(\boldsymbol{x_i})\right)^2}{\sum_{i=1}^{n} \left(f_{bb}(\boldsymbol{x_i}) - \bar{y}_{bb}\right)^2}.$$

Remark 9 *This is the R^2 metric between predictions of the black-box Poisson tree-based model and the predictions of the surrogate with K leaves.*

2.3.2 Additive tree models

As mentioned above, the regression tree is the basic component of many efficient methods but also more sophisticated. This is why we have detailed it in Sect. 2.2.6. Now, we describe additive tree models following the definition of Cui et al. (2015). This abstraction layer gives a common formalism to most tree-based methods. Since the surrogate extraction method applies to ATMs, we will be able to explain a large number of models.

Definition 1 *An additive tree model is a set of regression or classification trees that are combined in an affine manner. Mathematically, an ATM takes the following form:*

$$F : \boldsymbol{x_i} \longrightarrow w_0 + \sum_{t=1}^{T} w_t f_t(\boldsymbol{x_i}; \mathcal{R}_t, \Lambda_t, I_t)$$

where $f_t(\boldsymbol{x_i}; \mathcal{R}_t, \Lambda_t, I_t)$ is the output of the tree $t = 1, .., T$ for the observation $\boldsymbol{x_i}$ and $w_t \in \mathbb{R}$ is the weight of the tree t among the tree ensemble. We recall that $\mathcal{R}_t$ is the t-th partition, Λ_t is the set of predicted values associated with $\mathcal{R}_t$ and I_t is the number of regions in $\mathcal{R}_t$.

Remark 10 *There are several references with similar names to additive tree models in the literature. These include chapter 10 of Hastie et al. (2009) and J. Friedman et al. (2000) and J. H. Friedman (2001). These are closely related concepts. However, these references are only associated with tree-based boosting methods such as multiple additive regression trees (MART) and gradient boosting machines, for example. We need a more general definition that can also include models derived from bagging, such as random forest or extremely randomized trees Geurts et al. (2006). In Chipman et al. (2010), the authors refer to the concept we need in terms of "sum of trees" models and establish a link with additive models. In this article, we are interested in a method of post hoc interpretability, i.e., it considers that the black-box model is already fitted. Therefore, we are not interested in how the methods learn and combine trees but just in the result of this aggregation. This is why we choose to follow the proposed **Definition 1**.*

This definition formalizes and generalizes the classical point of view about tree-based methods. Indeed, we can see that each prediction of an ATM is a weighted sum of tree predictions.

Now, we slightly change our point of view to build the framework for the surrogate extraction method. First, it is worth noting that a binary tree is a partition of the covariate space $\mathbb{R}^p$. From this, we can see an ATM as the crossing of T partitions of the same space $\mathbb{R}^p$ and a linear combination of T

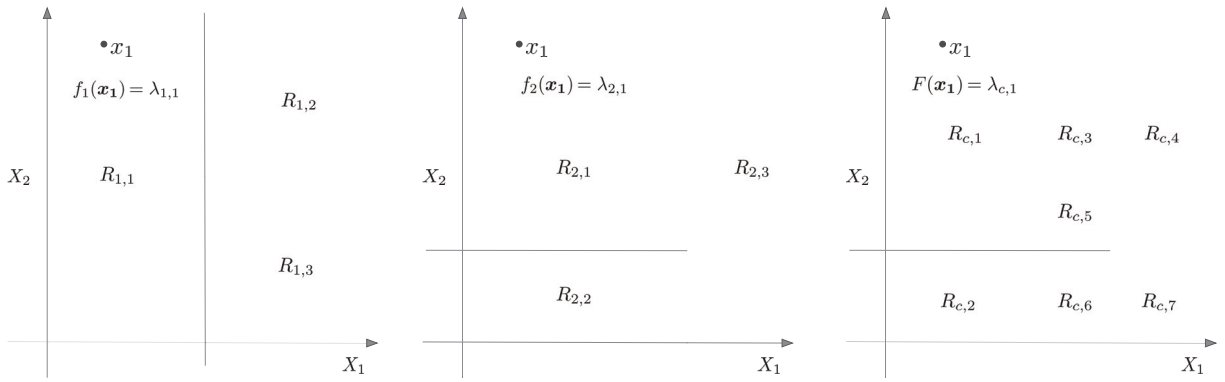

Fig. 2.10: The first two partitions of $\mathbb{R}^2$ are crossed to form $\mathcal{R}_c = \{R_{c,i}\}_{i=1}^{I_c}$.

predictions (one for each tree). A two-dimensional example is displayed **Fig. 2.10**. The first two partitions are crossed to form the last. Thus, we need to define what crossing two partitions is.

Definition 2 *(Cross-partition)*

Let us define $\mathcal{R}_1 = \{R_{1,i}\}_{i=1}^{I_1}$, $\mathcal{R}_2 = \{R_{2,i}\}_{i=1}^{I_2}$ *two partitions of the same space* $\mathbb{R}^p$. *We define the cross-partition by the collection:*

$$\mathcal{R}_c = \mathcal{R}_1 \cap \mathcal{R}_2 = \{R_{1,i} \cap R_{2,k} \mid i = 1, .., I_1,\ k = 1, .., I_2\} \backslash \{\emptyset\}.$$

Remark 11 *The cross-partition is a partition. See Appendix 2.5.1*

Let $\boldsymbol{x_i} \in \mathbb{R}^p$, and $\mathcal{R}_c = \{R_c\}_{c=1}^{C}$ and $\Lambda_c = \{\lambda_c\}_{c=1}^{C}$, respectively, be the cross-partition containing C regions and the predicted values inside each region. The previous lemma means that after the fitting step, an ATM is a partition of $\mathbb{R}^p$. Hence, we can write the prediction for $\boldsymbol{x_i}$ similar to a tree prediction

$$F(\boldsymbol{x_i}) = \sum_{c=1}^{C} \lambda_c \mathbb{1}_{\boldsymbol{x_i} \in R_c}.$$

Let us denote by $R_{c'}$ the hyperrectangle to which $\boldsymbol{x_i}$ belongs. By construction, $R_{c'} = \bigcap_{t=1}^{T} R_{t,k_t}$. Then, for each $t = 1, .., T$, there exists k_t such that $\boldsymbol{x_i} \in R_{t,k_t}$ and $f_t(\boldsymbol{x_i}; \mathcal{R}_t, \Lambda_t, I_t) = \lambda_{t,k_t}$. Hence, from the classical point of view, we can alternatively express the prediction for $\boldsymbol{x_i}$ as follows

$$F(\boldsymbol{x_i}) = \sum_{t=1}^{T} \omega_t f_t(\boldsymbol{x_i}; \mathcal{R}_t, \Lambda_t, I_t) = \sum_{t=1}^{T} \omega_t \lambda_{t,k_t}.$$

Therefore, if we set $\lambda_{c'} = \sum_{t=1}^{T} \omega_t \lambda_{t,k_t}$, it is easy to see that obtaining the ATM's prediction for $\boldsymbol{x_i}$ is equivalent to finding the cell containing $\boldsymbol{x_i}$ in the cross-partition and predicting a linear combination of tree predictions.

Remark 12 *If we combine multiple ATMs linearly, which means that the predicted values within the cross-partition's regions can be expressed as a linear combination, the result is still an ATM. This is interesting because we can combine a random forest and gradient boosting linearly and keep ATM properties. Hence, we could use the technique that we will describe in Sect. 2.3.3 to explain the resulting model.*

2.3.3 Defragmenting the space

One of the main difficulties when trying to explain the results of an ATM is that the number of trees leads to a significant increase in the number of regions and therefore in the number of rules. By approximating the set of trees by a small number of larger regions we should be able to obtain a simple surrogate model that is faithful to the black-box model.

Problem

Knowing the G regions generated by the crossing of the partitions of the fitted Poisson ATM $\mathcal{R} = \{R_g\}_{g=1}^{G}$ and its predicted values $\Lambda = \{\lambda_g\}_{g=1}^{G}$ in these areas, we want to find $K << G$ regions $\mathcal{R}' = \{R'_k\}_{k=1}^{K}$ and their predicted values $\Lambda' = \{\lambda'_k\}_{k=1}^{K}$ so that

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(f_{bb}(\boldsymbol{x_i}) - f_{sur}(\boldsymbol{x_i})\right)^2}{\sum_{i=1}^{n} \left(f_{bb}(\boldsymbol{x_i}) - \bar{y}_{bb}\right)^2}$$

the value characterizing the percentage of variations explained by the surrogate is as high as possible. The parameter K is arbitrarily chosen by the user. To address this problem, we adopt an approach developed by Hara and Hayashi (2016) and Hara and Hayashi (2018).

Remark 13 *If the surrogate has only one leaf that predicts the mean response of the black-box, i.e., $f_{sur} = \bar{y}_{bb}$, then $R^2 = 0$. Furthermore, if the surrogate predicts exactly the predictions of the black-box model, then $R^2 = 1$. Nevertheless, this metric is not lower bounded by 0. As a consequence, a bad surrogate can generate a negative R^2.*

Informal description of the objective

The idea of the method we present is based on generative models. These models try to provide the joint probability distribution of the pair $(Y_{i,bb}, \boldsymbol{X_i})$, where $Y_{i,bb}$ is a random variable representing the ATM's black-box prediction and $\boldsymbol{X_i}$ is a random vector of characteristics. Realizations of $Y_{i,bb}$ and $\boldsymbol{X_i}$ are denoted by $y_{i,bb}$ and $\boldsymbol{x_i}$ respectively. In our case, we want to express a slightly different probability distribution. We need to introduce $\boldsymbol{s(x_i)}$, the binary representation of $\boldsymbol{x_i}$. We denote by $\boldsymbol{S_i}$ the random vector of this binary representation. This representation will be detailed in Sect. 2.3.3. Hence, we aim to express the probability distribution of $(Y_{i,bb}, \boldsymbol{S_i})$ as a function of the ATM's parameters. For this, we will use a mixture of experts model introduced by Robert A. Jacobs, Michael I. Jordan, and Barto (1991) and Robert A. Jacobs, Michael I. Jordan, Nowlan, et al. (1991). A mixture of experts consists of a set of specialized models each of which is responsible for a particular region of the input space. The concept was extended to hierarchical mixture of experts by Michael I Jordan and Robert A Jacobs (1992) and M. Jordan and R. Jacobs (n.d.) For an overview see McLachlan and Peel (2000). In this setting, we suppose that the probability distribution of $(Y_{i,bb}, \boldsymbol{S_i})$, denoted by g, can be written in the form :

$$g(y_{i,bb}, \boldsymbol{s(x_i)};\ \Pi) = \sum_{k=1}^{K} \alpha_k g_k(y_{i,bb}, \boldsymbol{s(x_i)};\ \boldsymbol{\theta_k}) \tag{2.1}$$

where $g_k(y_{i,bb}, \boldsymbol{x_i};\ \boldsymbol{\theta_k})$ is a probability distribution parametrized by the vector of parameters $\boldsymbol{\theta_k}$, $\Pi = \{\alpha_1, \ldots, \alpha_K, \boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_K}\}$ is the set of unknown parameters, and α_k are nonnegative quantities that sum to one, i.e.,

$$\sum_{k=1}^{K} \alpha_k = 1.$$

Hence, in a mixture of experts model, there are two types of components. The first ones are the K modules also called experts. These last approximate the probability distribution of $(Y_{i,bb}, \boldsymbol{S_i})$ within each region of the covariate space. It is assumed that different experts are appropriate in different regions of the covariate space. Thus, by modifying these modules, it is possible to adapt this method to a wide variety of problems. According to our developments in Sect. 2.2.6, it seems appropriate to search for regions in the form of rectangles of dimension p. Hence, the experts' parameters $\boldsymbol{\theta_k}$ are taken to characterize the region indexed by k and the prediction within. The second type of component is a module to identify for any $\boldsymbol{s(x_i)}$ the expert module whose output is most likely to approximate the probability distribution of $(Y_{i,bb}, \boldsymbol{S_i})$. This is the role of the mixing proportions α_k, the probabilities that an instance $\boldsymbol{s(x_i)}$ belongs to the region k. Once the components of the

model are expressed as a function of the parameters, we can formalize the problem as a maximum likelihood problem to obtain the parameters of our mixture of experts, i.e.,

$$\max_{\Pi} \log\left(g(y_{i,bb}, \boldsymbol{s}(\boldsymbol{x_i}); \Pi)\right).$$

In this kind of situation, it is quite common to use the EM algorithm. Before developing the elements related to the EM algorithm framework, we have to propose an expression for the expert probability distributions. This is the purpose of the next sections.

Binary expressions of the regions and the observations

Before going into the detail of the mixture of experts, we need to introduce the binary representation of regions and observations. Suppose that the tree ensemble generated a set of L_j splits for each variable $j \in \{1, .., p\}$. We denote by L the total number of splits for the tree ensemble:

$$\sum_{j=1}^{p} L_j = L.$$

To simplify notation, we re-index the thresholds by z_l with $l \in \{1, .., L\}$ and introduce d_l as the index of the variable split at z_l. According to Sect. 2.3.2, these trees form a partition of the space. It means that a region $R_g \in \mathcal{R}$ can be uniquely characterized by a binary vector $\tilde{\boldsymbol{\eta}}_g \in \{0, 1\}^L$. The l-th component of this vector $\tilde{\eta}_{g_l}$ is equal to 1 if all $\boldsymbol{x_i}$ that fall into R_g satisfy the statement $x_{i,d_l} > z_l$, and $\tilde{\eta}_{g_l} = 0$ otherwise. This is illustrated in **Fig. 2.11**. The thresholds of the splits are denoted by $z_l,\ l \in \{1, 4\}$. The first element of the binary vector $\tilde{\boldsymbol{\eta}}_2$ is equal to 1 because there are only $\boldsymbol{x_2}$ and $\boldsymbol{x_{10}}$ in R_2 and they both satisfy $x_{.,1} > z_1$. The second element is 0 because $x_{.,1} \leq z_2$ and so on.

We now need a similar expression to characterize whether an observation is inside a region or not. Let us define by $\boldsymbol{s}(\boldsymbol{x_i}) \in \{0, 1\}^L$ the binary expression of an observation $\boldsymbol{x_i}$ where the l-th element s_l is equal to $\mathbb{1}_{x_{i,d_l} > z_l}$. This is also represented in **Fig. 2.11**. Now, if we want to determine whether an observation is inside a specific region R_g, it is equivalent to comparing the two vectors $\boldsymbol{s}(\boldsymbol{x_i})$ and $\tilde{\boldsymbol{\eta}}_g$. This is useful to represent the region shapes of the (deterministic) ATM, but we are not able to derive a generative model if we do not make this representation more flexible.

In fact, we want to express the probability of a binary representation for an observation as a function of the shape parameters. Therefore, for any macro region R_k, we want to express the probability to observe a binary representation of $\boldsymbol{x_i}$. This is because now, the region R_k contains a mix of binary representations for $\boldsymbol{x_i}$. See **Fig. 2.12**. The region R_k can "generate" different binary representations for the observations it contains. The idea to extend the definition of shape parameters to fit our needs

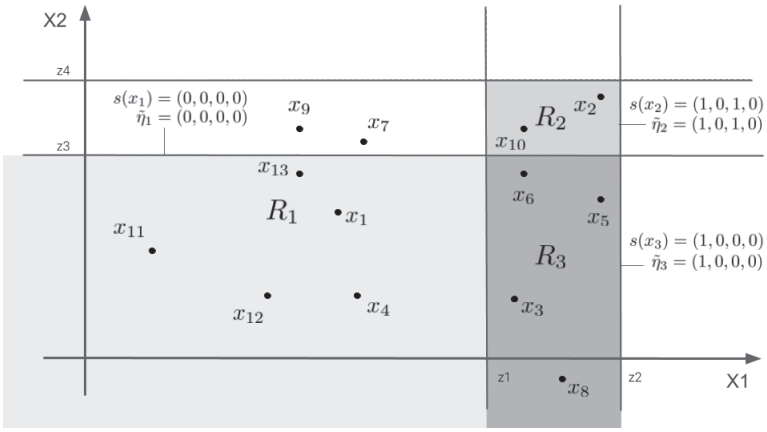

Fig. 2.11: Binary expression of an ATM's regions

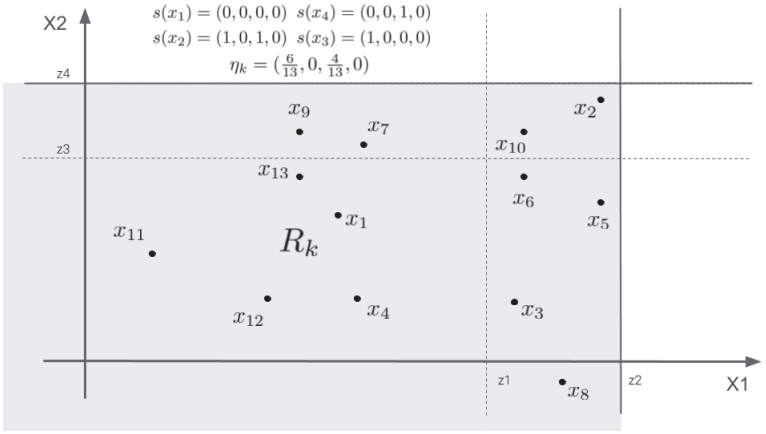

Fig. 2.12: Binary expression of an arbitrary region R_k. The component $\eta_{k,1}$ is equal to $\frac{6}{13}$ because six points are above the threshold z_1. Note that the vector component does not sum to one.

is based on the fact that in R_k, some $\boldsymbol{x_i}$ satisfy the statement $x_{i,d_l} > z_l$, while some others do not. We denote by $\boldsymbol{\eta_k}$ this extended version of the vector of parameters. We can naturally think of η_{kl} as a probability for an $\boldsymbol{x_i}$ to meet the condition $x_{i,d_l} > z_l$. Therefore,

- if all $\boldsymbol{x_i} \in R_k$ satisfy the statement $x_{i,d_l} > z_l$, then $\eta_{kl} = 1$;
- if no $\boldsymbol{x_i} \in R_k$ satisfy the statement $x_{i,d_l} > z_l$, then $\eta_{kl} = 0$, which is the case for thresholds z_2 and z_4 in **Fig. 2.12**;
- if some $\boldsymbol{x_i} \in R_k$ satisfy the statement $x_{i,d_l} > z_l$, then $\eta_{kl} = P(x_{i,d_l} > z_l)$. It is the case for thresholds z_2 and z_3 in **Fig. 2.12**.

As proposed by Hara and Hayashi (2018), we choose a Bernoulli distribution for $\boldsymbol{S_i}$ the random vector of binary representations $\boldsymbol{s(x_i)}$. We denote by b this probability distribution:

$$b(\boldsymbol{s(x_i)}; \boldsymbol{\eta_k}) = \prod_{l=1}^{L} \eta_{k_l}^{s_l} (1 - \eta_{k_l})^{(1-s_l)}$$

where $\eta_{k_l} = P(s_l = 1) = P(x_{i,d_l} > z_l)$. Now, we have a generative model for the binary representation of $\boldsymbol{x_i}$ as a function of $\boldsymbol{\eta_k}$.

Probabilistic expression of the experts

We have just defined the probability distribution of the binary expression of an observation as a function of the parameters $\boldsymbol{\eta_k}$. To complete our mixture of experts expression, we have to introduce $\boldsymbol{\phi_k}$ as the parameters on which the surrogate's prediction depends within a region indexed by k. Hereafter, $\boldsymbol{\theta_k}$ is written as $(\boldsymbol{\eta_k}, \boldsymbol{\phi_k})$. Considering this, we can rewrite equation (2.1):

$$g(y_{i,bb}, \boldsymbol{s(x_i)};\ \Pi) = \sum_{k=1}^{K} \alpha_k g_k(y_{i,bb}, \boldsymbol{s(x_i)};\ \boldsymbol{\eta_k},\ \boldsymbol{\phi_k}).$$

Let us introduce $\boldsymbol{U_i}$ as a binary random vector such that the k-th element of this vector is 1 if the i-th observation belongs to the region k. We denote by $U_{i,k}$ the k-th element of the random vector $\boldsymbol{U_i}$:

$$U_{i,k} = \begin{cases} 1 \text{ if } \boldsymbol{x_i} \in R_k \\ 0 \text{ elsewhere.} \end{cases}$$

Suppose that the conditional probability distribution of $(Y_{i,bb}, \boldsymbol{S_i})$ given $\boldsymbol{U_i} = \boldsymbol{e_k}$ is the component distribution $g_k(y_{i,bb}, \boldsymbol{s(x_i)};\ \boldsymbol{\phi_k},\ \boldsymbol{\eta_k})$, where $\boldsymbol{e_k}$ denotes a binary vector taking the value 1 in it is k-th component and 0 elsewhere. With this natural idea, we can generate $(y_{i,bb}, \boldsymbol{s(x_i)})$ from our mixture of experts. Thanks to this binary representation, we can write the probability for an observation to be inside a region as a function of the mixing proportions:

$$P(\boldsymbol{U_i} = \boldsymbol{u_i}) \overset{\text{def}}{=} p(\boldsymbol{u_i}; \boldsymbol{\alpha}) = \prod_{k=1}^{K} \alpha_k^{u_{i,k}}.$$

The introduction of $\boldsymbol{U_i}$ allows us to fit our likelihood maximization problem into the framework of the EM algorithm. We now want to express the complete data log-likelihood, i.e., the observed

variables and the unobserved random vectors. These latent vectors allow us to express the joint probability distribution $(Y_{i,bb}, \boldsymbol{s_i}, \boldsymbol{U_i})$ as a function of the parameters $\boldsymbol{\phi_k}, \boldsymbol{\eta_k}, \boldsymbol{\alpha}$. This expression can be decomposed in the following way:

$$h(y_{i,bb}, \boldsymbol{s(x_i)}, \boldsymbol{u_i}; \boldsymbol{\phi_k}, \boldsymbol{\eta_k}, \boldsymbol{\alpha}) = \underbrace{g(y_{i,bb}, \boldsymbol{s(x_i)}|\boldsymbol{u_i}; \boldsymbol{\phi_k}, \boldsymbol{\eta_k})}_{(a)} \times \underbrace{p(\boldsymbol{u_i}; \boldsymbol{\alpha})}_{(b)}.$$

Next, we need to select a distribution for $Y_{i,bb}$. We choose a Gaussian distribution, meaning that $\boldsymbol{\phi_k} = \{\lambda_k, \sigma_k^2\}$. We denote by q the probability distribution:

$$q(y_{i,bb}; \lambda_k, \sigma_k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{y_{i,bb}-\lambda_k}{\sigma_k})^2}.$$

Remark 14 *Thanks to this flexibility on the distribution of $Y_{i,bb}$, we can theoretically adapt the method to find a surrogate of an ATM that performs Tweedie regression for example.*

We make the assumption that $Y_{i,bb}$ and $\boldsymbol{S_i}$ are conditionally independent given $\boldsymbol{U_i} = \boldsymbol{u_i}$, where $\boldsymbol{u_i} = \boldsymbol{e_k}$, so we can rewrite (a):

$$g(y_{i,bb}, \boldsymbol{s(x_i)}|\boldsymbol{u_i}; \boldsymbol{\phi_k}, \boldsymbol{\eta_k}, \boldsymbol{\alpha}) = q(y_{i,bb}|\boldsymbol{u_i}; \boldsymbol{\phi_k}) \times b(\boldsymbol{s(x_i)}|\boldsymbol{u_i}; \boldsymbol{\eta_k}).$$

By definition of $U_{i,k}$ we can further rewrite:

$$g(y_{i,bb}, \boldsymbol{s(x_i)}|\boldsymbol{u_i}; \boldsymbol{\phi_k}, \boldsymbol{\eta_k}, \boldsymbol{\alpha}) = \prod_{k=1}^{K} [q(y_{i,bb}; \boldsymbol{\phi_k}) \times b(\boldsymbol{s(x_i)}; \boldsymbol{\eta_k})]^{u_{i,k}}.$$

Finally, we obtain our probabilistic expression for the ATM. We denote by h the probability distribution of $(Y_{i,bb}, \boldsymbol{S_i}, \boldsymbol{U_i})$:

$$h(y_{i,bb}, \boldsymbol{s(x_i)}, \boldsymbol{u_i}; \lambda_k, \sigma_k, \boldsymbol{\eta_k}, \boldsymbol{\alpha}) = \prod_{k=1}^{K} [q(y_{i,bb}; \lambda_k, \sigma_k) \times b(\boldsymbol{s(x_i)}; \boldsymbol{\eta_k}) \times \alpha_k]^{u_{i,k}}$$

where each probability distribution is expressed as a function of parameters $\Pi = \{\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_K}, \boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta_K}, \alpha_1, \ldots, \alpha$
From this expression, we can write the complete data likelihood for our problem and solve it with the EM algorithm. We detail the calculation in Sect. 2.3.3.

Solve the maximum likelihood problem with the EM algorithm

The random vector $\boldsymbol{U_i}$ allows us to formulate the problem as an incomplete data problem. Thus, we can fit into the EM framework. Our observations $\Delta = \{(y_{1,bb}, \boldsymbol{s(x_1)}), \ldots, (y_{n,bb}, \boldsymbol{s(x_n)})\}$ are seen as incomplete since the associated $\boldsymbol{U_i}$ vectors are not observable. Recall that in this framework, we assume that each observation comes from an expert module once fitted. It is the vector $\boldsymbol{U_i}$ that encodes the belonging of an observation to a module. The complete observations are therefore $\{(y_{1,bb}, \boldsymbol{s(x_1)}), \ldots, (y_{n,bb}, \boldsymbol{s(x_n)}), \boldsymbol{u_1}, \ldots, \boldsymbol{u_n}\}$. We can now write the complete data log-likelihood:

$$L_c(\Pi) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{i,k} \left[\log\left(\alpha_k\right) + \log\left(q(y_{i,bb}; \lambda_k, \sigma_k)\right) + \log\left(b(\boldsymbol{s(x_i)}; \boldsymbol{\eta_k})\right)\right].$$

Now, we apply the EM algorithm considering $u_{i,k}$ are missing data. This method first introduced by Dempster et al. (1977) alternates between two steps, the expectation step denoted by E and the maximization step denoted by M to solve the log-likelihood maximization problem. In the E step, the algorithm tries to find a lower bound given the current parameter estimation $\Pi^{(t)} = \{\boldsymbol{\phi_1}^{(t)}, \ldots, \boldsymbol{\phi_K}^{(t)}, \boldsymbol{\eta_1}^{(t)}, \ldots, \boldsymbol{\eta_K}^{(t)}, \alpha_1^{(t)}, \ldots, \alpha_K^{(t)}\}$. The superscript t indicates the states at the t-th iteration. The M step searches for the parameters $\Pi^{(t+1)}$ that maximize the lower bound found in the E step.

E step

In the E step, we take the conditional expectation of the complete data log-likelihood given the observed data Δ using the current fit $\Pi^{(t)}$. Let us denote by Q the lower bound we try to find in this step. This can be written :

$$Q(\Pi; \Pi^{(t)}) = E_{\Pi^{(t)}}\left[\log\left(L_c(\Pi)\right) \mid \Delta\right].$$

To obtain the lower bound at step $(t+1)$, we only need to calculate the conditional expectation of $U_{i,k}$ given Δ. By noting that the complete data log-likelihood is linear in $u_{i,k}$, this calculation expectation simplifies as:

$$
\begin{aligned}
E_{\Pi^{(t)}}[U_{i,k} \mid \Delta] &= P_{\Pi^{(t)}}(U_{i,k} = 1 \mid \Delta) \\
&= \alpha_k^{(t)} \frac{g_k(y_{i,bb}, \boldsymbol{s(x_i)}; \lambda_k^{(t)}, \boldsymbol{\eta_k}^{(t)})}{\sum_{m=1}^{K} \alpha_m^{(t)} g_m(y_{i,bb}, \boldsymbol{s(x_i)}; \lambda_m^{(t)}, \boldsymbol{\eta_m}^{(t)})}. \\
&\stackrel{\text{def}}{=} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}).
\end{aligned}
$$

The last quantity we defined, $\tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)})$, is the posterior probability that the observation $(y_{i,bb}, \boldsymbol{s(x_i)})$ belongs to the i-th module of the mixture of experts. We can now rewrite the lower bound:

$$
Q(\Pi; \Pi^{(t)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}) \left[\log(\alpha_k) + \log(q(y_{i,bb}; \lambda_k, \sigma_k)) + \log(b(\boldsymbol{s(x_i)}; \boldsymbol{\eta_k}))\right].
$$

M step

To obtain the updated parameters $\Pi^{(t+1)}$ from this step, we need to maximize $Q(\Pi; \Pi^{(t)})$ with respect to Π. We can observe that this maximization problem can be broken into three simpler maximization problems. On the one hand:

$$
\begin{cases}
\max\limits_{\boldsymbol{\alpha}} \sum_{k=1}^{K} \sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}) \log(\alpha_k) \\
s.t \ \sum_{k=1}^{K} \alpha_k = 1.
\end{cases}
$$

On the other hand:

$$
\begin{cases}
\max\limits_{\boldsymbol{\phi}} \sum_{k=1}^{K} \sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}) \log(q(y_{i,bb}; \lambda_k, \sigma_k)) \\
\max\limits_{\boldsymbol{\eta}} \sum_{k=1}^{K} \sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}) \log(b(\boldsymbol{s(x_i)}; \boldsymbol{\eta_k})).
\end{cases}
$$

We can find admissible solutions for the first maximization problem with the Lagrange multiplier method. The last two problems can be classically solved by maximum likelihood. Finally, we obtain the updated parameters $\forall l \in \{1, .., L\}, \ \forall k \in \{1, .., K\}$:

$$\alpha_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}), \quad \lambda_k^{(t+1)} = \frac{\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}) \times y_{i,bb}}{\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)})}$$

$$\eta_{k,l}^{(t+1)} = \frac{\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)}) \times s_{i,l}}{\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)})}, \quad (\sigma_k^2)^{(t+1)} = \frac{\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)})(y_{i,bb} - \lambda_k^{(t+1)})^2}{\sum_{i=1}^{n} \tau_k(y_{i,bb}, \boldsymbol{s(x_i)}; \Pi^{(t)})}.$$

Remark 15 *The objective function is not concave. As a result, the algorithm may get stuck in a local maximum. This is why, in practice, we restart the method several times with a randomly defined starting point.*

We can observe that the regions no longer form a partition. This can cause several problems. On the one hand, if two rectangles overlap, observations can belong to both at the same time. This can be challenging in explaining the results. In fact, it is easier to understand a set of simple rules forming a hard partition rather than a soft partition. By a soft partition, we mean that a point belongs to a region with a given probability as opposed to a hard partition where a point belongs to only one region with certainty. Furthermore, if there are empty spaces in this soft partition, an observation that has never been seen can have any assigned area. In DefragTrees, this problem is handled by imputing the average value of the ATM's predictions to the new instance. However, this is unsatisfactory since it does not provide information about the class to which this observation belongs. Therefore, it does not provide an explanation. The problems we point out are related to the fact that DefragTrees creates a soft partition of the space. However, to be easily explained, as humans, we prefer a hard partition. This is why we propose an alternative version of DefragTrees in which we try to rebuild a hard partition of space from the soft partition. To accomplish this, we use a modified version of the classical regression tree. This version differs from the standard classification tree because the tree is forced to choose its splits only among those created by the ATM, and the tree must have only K leaves.

2.3.4 Extracting a tree from the fitted black-box claim frequency model

In Sect. 2.2.2, we fitted four claim frequency models with Poisson ATM. We checked these models with classic tools at our disposal. Now, we want to go further in the explanations and extract faithful trees from these last ones. In this article, we only focus on the two Poisson random forest models, `RF policy` and `RF policy + telematics`, but the method presented above also works for `GBM policy` and `GBM policy + telematics`. By extracting a tree surrogate, we want to understand the predictions of `RF policy` to eventually detect problems. We also want to highlight what was learned by `RF policy + telematics` to discover new interesting relationships between explanatory variables and claim frequency.

In DefragTrees Hara and Hayashi (2018), the method we use and adapt to fit our needs, the only parameter left to the user is K, the region number of our surrogates. This is why we fit this

method with several arbitrary values of $K \in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 16]$. Hence, we are able to evaluate the trade-off between the fidelity of the K region surrogates and the parsimony represented by K. Since the objective function is not concave, the method can find local maxima for some random initial states. Hence, we restart the algorithm 20 times for each K with a different seed. Then, we select the most faithful model for each K.

Surrogate for RF policy

In **Fig. 2.13**, we present the fidelity/parsimony trade-off for `RF policy`. The R^2 fidelity metric is described in Sect. 2.3.3 and is evaluated on soft partitions. We compare these results to the ones evaluated on hard partitions in **Fig. 2.14**. In this second case, it gives us the percentage of black-box `RF policy` variations explained with only K leaves. As we can see, the results are not as good as expected but not so bad considering that the 5-region surrogate captures approximately 75% of the black-box prediction variability.

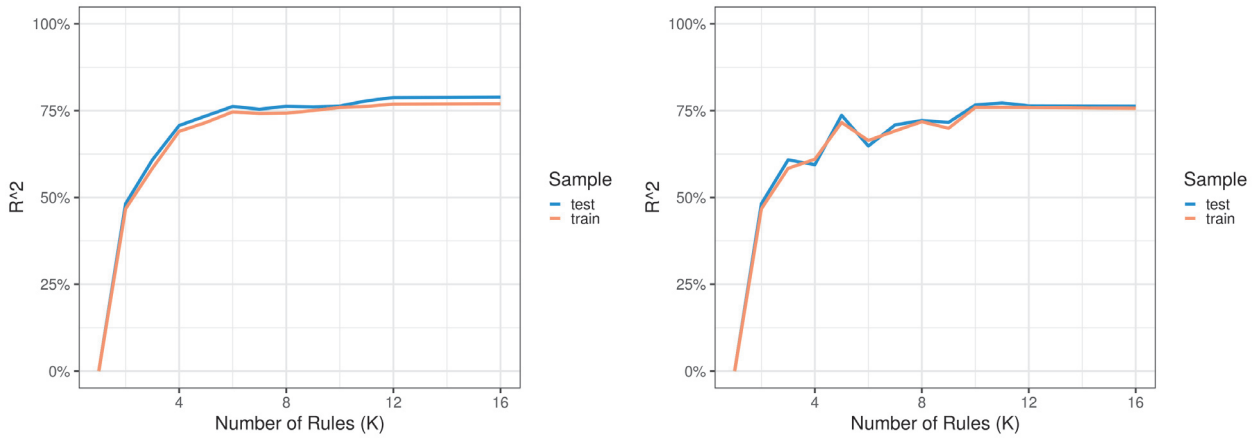

Fig. 2.13: Trade-off of Fidelity/Parsimony with `RF policy` (soft partition)

Fig. 2.14: Trade-off of Fidelity/Parsimony with `RF policy` (hard partition)

Remark 16 *Intuitively, the more we add regions in the surrogate, the more we can accurately describe the black-box variations. We see such a trend in the graphs, but we could not perform the calculations for the maximum of regions because there are at least* $12,100$ *different regions in* `RF policy`*. Moreover, it would be of little interest since we want to be parsimonious to ensure the explainability.*

There is no recommended method to choose the number of regions. It is possible to use a kind of elbow method thanks to the globally concave profile of the curves. Indeed, we can pick each method that preserves as much as possible the fidelity with a parsimony constraint on K. Here, we choose the 5-region surrogate that gives the best level of fidelity (approximately 75%) with the minimum number of leaves (rules). Now, we may further be interested in representing the errors of `RF policy`'s surrogate predictions within each region. This is presented in **Fig. 2.16**. The relative errors made by the 5-region surrogate are inferior at 10% in absolute value for most of the observations. Even

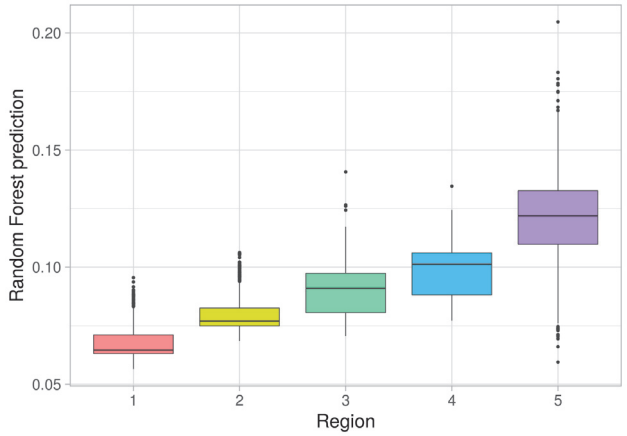

Fig. 2.15: `RF policy`'s predictions by region.

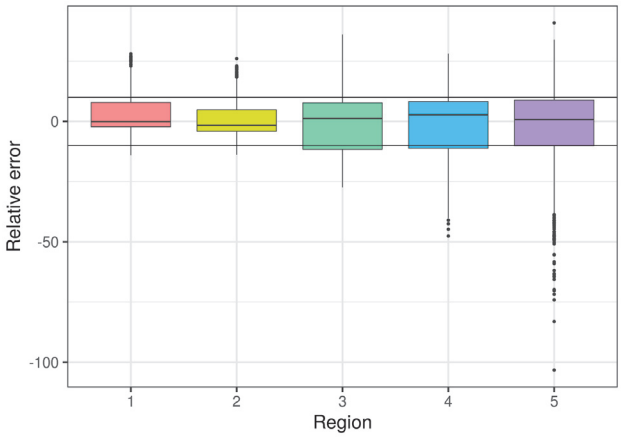

Fig. 2.16: `RF policy`'s relative errors by region.

though the surrogate makes mistakes when predicting the `RF policy`'s predictions because of its small number of regions, it is faithful to the black-box predictions.

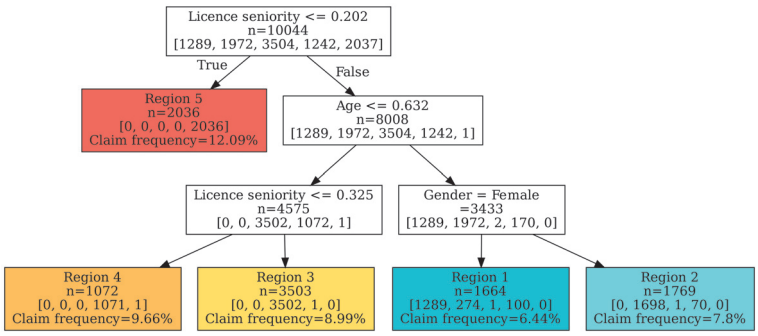

Fig. 2.17: The 5-region surrogate for the `RF policy` model. Each leaf represents a region numbered from 1 to 5. Please note that Fig. 2.15 and Fig. 2.16 refer to these regions.

The examination of these plots makes us confident of the faithfulness of our surrogate. We show in Fig. 2.17 the surrogate as a tree. On this tree, we can clearly see that `RF policy` relies on `Age` and `License seniority`, which are two correlated predictors. Unlike the GLM, we do not have to take care of correlated predictors since trees, and therefore the random forest, are known to be insensitive to correlated variables. Nevertheless, if we include those predictors in the training set, the random forest creates many splits because of the sampling mechanisms on `Age` and `License seniority`. Because of this, the space is unnecessarily fragmented and therefore harder to understand. We can also observe that `Area` is not present in the surrogate. This is because this variable is not as important as displayed predictors to explain black-box variability. Nevertheless, if we had selected a deeper tree, we would have observed this predictor. From this surrogate tree, we can see that the `RF policy` relies on variable/threshold couples that are relevant for actuaries. Unfortunately, in this case, we cannot diagnose precisely the performance problem. To go further, it would be necessary to inspect the `RF policy`'s predictions and compare them with those of the GLM.

It is also interesting to note that the regions determined by the surrogate separate the `RF policy`'s predictions relatively well as a function of the predicted claim frequency. We can see in **Fig. 2.15** that individuals who fall into region 5 have a high predicted claim frequency compared to other groups, for example. Hence, a surrogate is helpful to identify combinations of variables that can explain the predicted claim frequency. In other words, it is useful to isolate the characteristics of good and bad risks.

Surrogate for RF policy + telematics

In regard to the `RF policy`, we test different values for K to identify a good surrogate candidate. To choose the parameter K of the surrogate, we use **Fig. 2.18** and **Fig. 2.19**.

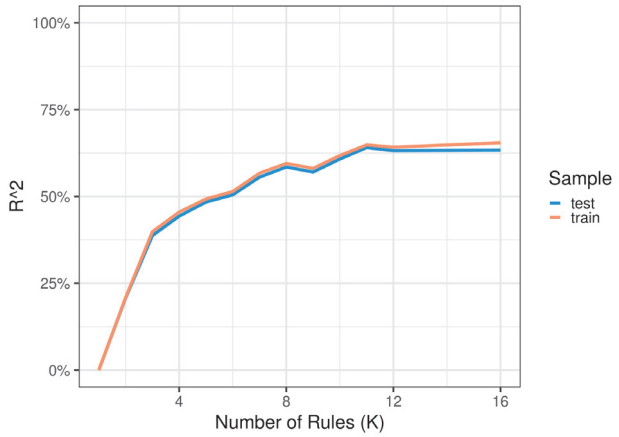

Fig. 2.18: Trade-off for the `RF policy` and `telematics` (soft)

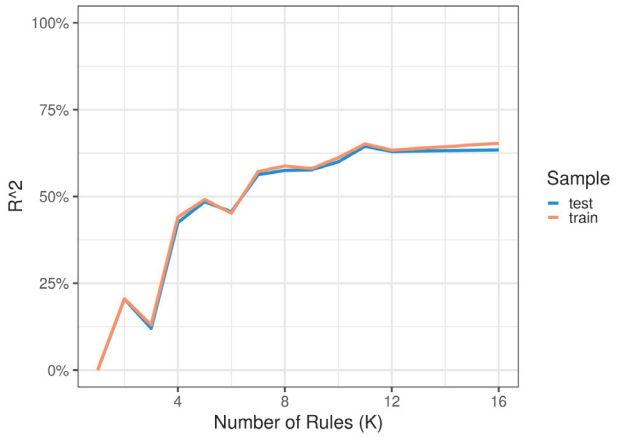

Fig. 2.19: Trade-off for the `RF policy` and `telematics` (hard)

As we can see, finding an appropriate surrogate is now a more complex task because of the dimension of the problem and the complexity of the random forest model. In fact, we need a deeper surrogate tree to capture the variability of the black-box. For this one, we set $K = 12$ because it yields a better fidelity score (approximately 63%) with the lowest number of regions. We display the tree representing the surrogate in **Fig. 2.22**. We also represent a boxplot of errors by region in **Fig. 2.21** and a boxplot of `RF policy + telematics` predictions in **Fig. 2.20**. We also show in **Tab. 2.5** the predicted claim frequencies and the observed claim frequency in each region identified by the surrogate model.

Remark 17 *To implement the explainability models developed above, we modified the DefragTrees*[2] *code.*

According to the surrogate, the model seems to rely mostly on telematics variables. This surrogate is completely explicit but hard to understand because we have relatively little experience with

[2] https://github.com/sato9hara/defragTrees

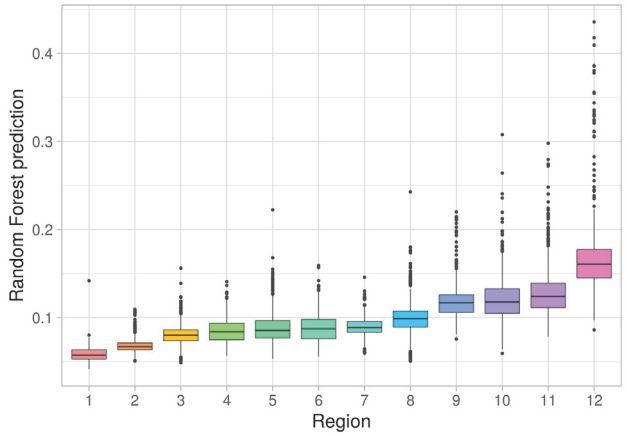

Fig. 2.20: `RF policy + telematics`'s relative errors by region.

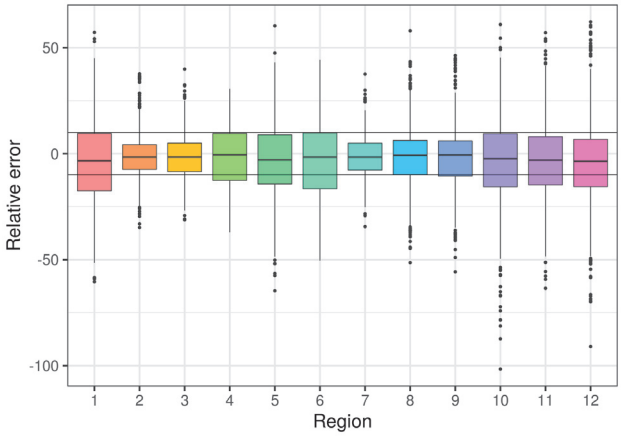

Fig. 2.21: `RF policy + telematics`'s predictions by region.

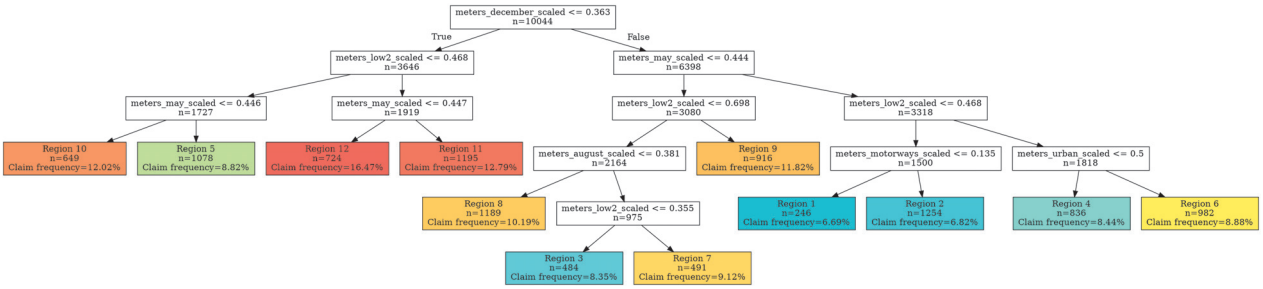

Fig. 2.22: The 12-region surrogate for `RF policy + telematics` model. Each leaf represents a region numbered from 1 to 12. Please note that Fig. 2.20 and Fig. 2.21 refer to these regions. Furthermore, regions 1 to 5 in the surrogate for `RF policy` are not the same as regions 1 to 5 in the surrogate for `RF policy + telematics`.

Tab. 2.5: Predicted vs observed claim frequency by region.

Region	Surrogate	Random forest	Observed claim frequency
1	6.69%	5.87%	3.75%
2	6.82%	6.83%	5.54%
3	8.35%	8.06%	8.82%
4	8.44%	8.51%	7.80%
5	8.82%	8.82%	7.47%
6	8.88%	8.83%	8.15%
7	9.12%	9%	9.50%
8	10.19%	9.86%	9.98%
9	11.82%	11.78%	12.98%
10	12.02%	12.07%	12.27%
11	12.79%	12.8%	14.73%
12	16.47%	16.69%	25.28%

telematics data and few general conclusions have been given in the literature. Hence, from an actuarial point of view, it is hard to unequivocally accept this model.Indeed, humans tend to more easily accept explanations that are in line with their knowledge. This phenomenon is explained in Miller (2019). However, we can see that intuitive predictors such as the proportions of meters

traveled between 19 h and 22 h meters_low2 or in a specific month intervene on part of the tree. In the light of this knowledge extracted from RF policy + telematics, we were able to check that there are drivers in the database who do not drive uniformly over the year. These drivers have very high proportions of kilometers driven in certain months, in particular January, May, June, August, September and December. In the remaining months of the year, these same drivers drive very little. This, together with the driving slot, reflects occasional driving behavior. Given that the portfolio is very young, it is likely that a significant portion of the portfolio is composed of students. These students may occasionally use their car to drive home on weekends or vacations. Therefore, they do not drive continuously. This may explain the lack of experience and consequently the claim frequency. Thanks to the surrogate model, we were able to quickly extract knowledge and build a hypothesis. Furthermore, we can observe that meters_low2 systematically discriminates riskier profiles. The surrogate tree helps to discriminate between riskier and less risky profiles. These relationships are free from variables such as License seniority. Hence, it would be interesting to integrate these in an a posteriori tariff. Therefore, it can help reduce premiums for some individuals in this risky population and, consequently, be more competitive.

Inspecting regions

Thanks to the surrogate, we have a partition of the feature space and we know approximately the black-box predicted value inside. However, the surrogate does not necessarily use all the available variables because of its shallow depth. As a consequence, some actuarially important variables such as Gender and License seniority in the surrogate do not appear. This does not mean that they are not used by the black-box model but that they are less useful for characterizing black-box variations than those appearing in the surrogate. In fact, telematics variables occur higher up in the surrogate tree. Therefore, they are more discriminating for RF policy + telematics than License seniority. However, License seniority has not disappeared from RF policy + telematics and is not useless. We represent in **Fig. 2.23** the mean response of RF policy + telematics conditional on the surrogate's leaves. We thus seek to determine whether there remains explanatory power carried by License seniority in each of the leaves. There is still a trend indicating that the more experienced the driver is, the lower the frequency predicted by RF policy + telematics. However, we can see that in leaves 1 to 5, this trend is very clearly attenuated. Thus, without using the License seniority variable, the model isolates the riskiest profiles from the less risky ones. In the higher-risk leaves, some explanatory power remains in the License seniority variable. However, it no longer seems very useful in leaves 1 to 5.

In **Fig. 2.24**, we represent the proportion of male and female inside each leaf. We can observe that in the surrogate regions associated with the highest claim frequencies, the proportion of males is more important than the proportion of females. In contrast, the lowest predicted claim frequencies are associated with a higher proportion of females. Each leaf represents a pattern in driving habits. Thus, we see that there is a correlation between the less risky driving habits and a higher proportion

of women. This would tend to confirm that the `Gender` variable is a proxy for more or less risky behavior patterns. This had already been highlighted by Verbelen et al. (2018).

Remark 18 *This analysis can also help to detect whether a black-box model discriminates according to a particular variable. Hence, it can be useful for ethical challenges.*

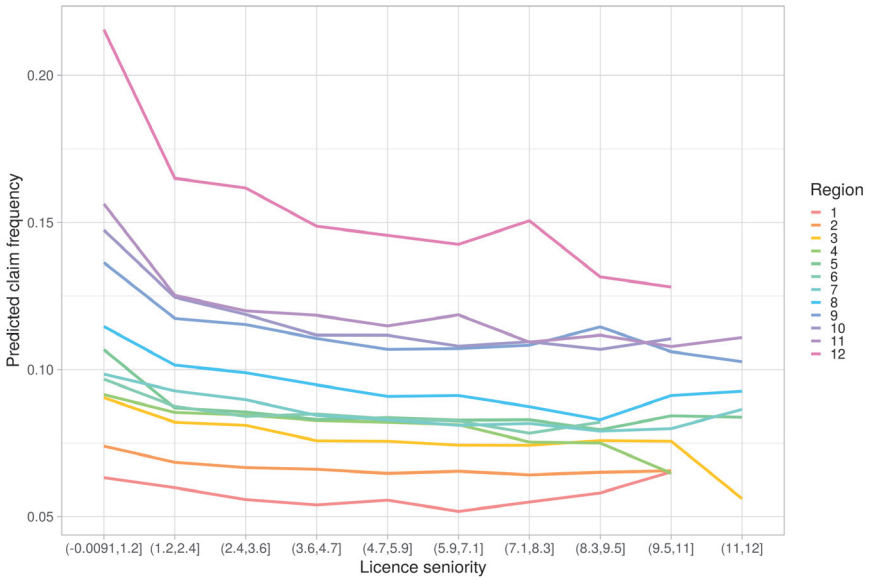

Fig. 2.23: Mean `License seniority` by areas for the RF policy + telematics model

Alternatively, we can decide to extract the patterns found by the surrogate to integrate them in a classic GLM. In fact, many insurers are not ready to move to a fully data-driven strategy for car insurance pricing. Here, we propose a simple alternative that consists of integrating the 2- or 3-level interactions displayed in the surrogate and include them in a GLM.

Export new knowledge to GLM

Insurance companies are still very supportive of the GLM. Indeed, the lack of confidence in new machine learning methods associated with IT systems that are often complex and costly to modify does not encourage insurers to develop new technologies. This is why we propose an intermediate technique that does not require the modification of an insurance company's IT infrastructure. Since we have a decision tree that faithfully replicates `RF policy + telematics`'s predictions, we have high-order interactions at our disposal. We create a new variable that indicates the leaf an observation falls into. Then, we add this variable to our `GLM policy`. To evaluate the relevance of this new variable, we use a stratified cross-validation as before. In addition, we have a limited number of variables at our disposal. Consequently, we can make an exhaustive search for the best model

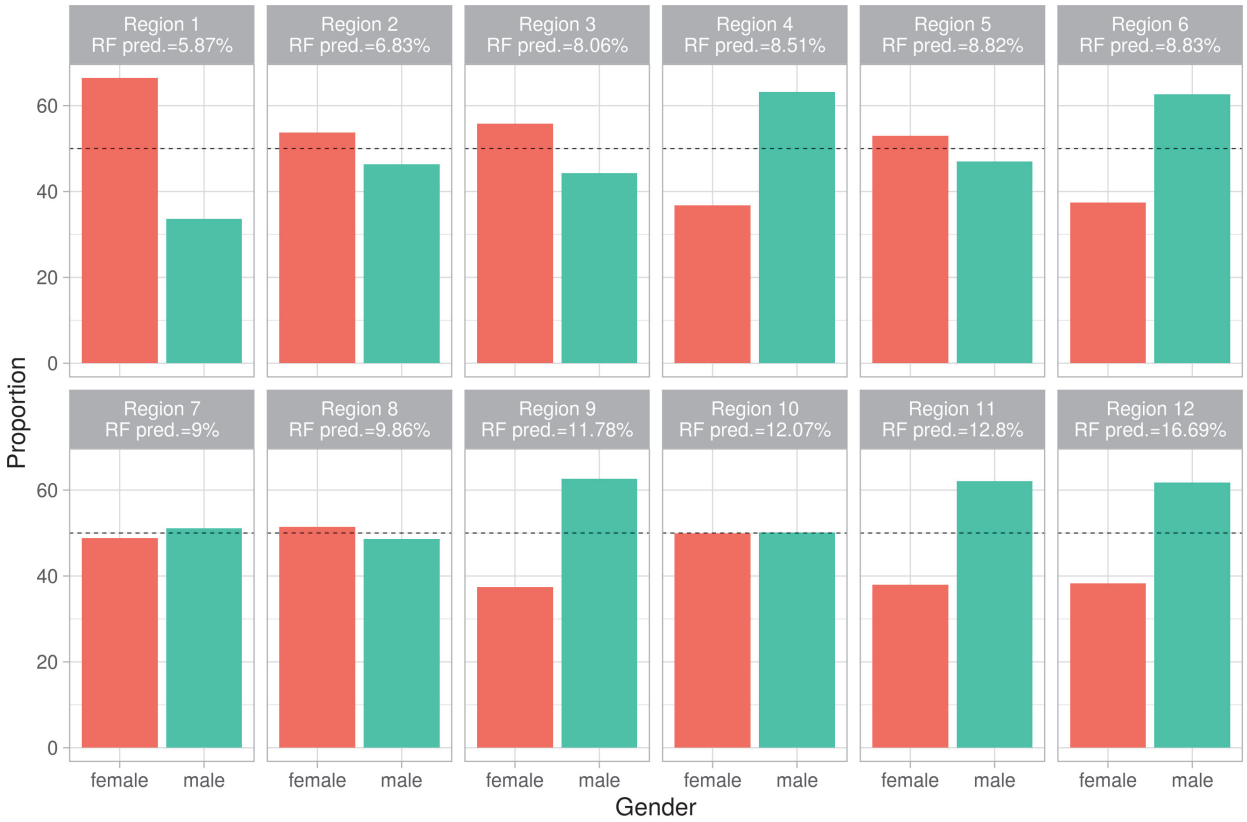

Fig. 2.24: `Gender` by areas for the RF policy + telematics model

among all possible combinations. We present the best models for `GLM policy` and `GLM policy + telematics` in **Tab. 2.3**.

The integration of these new interactions in the GLM has made it possible to significantly reduce the mean Poisson deviance loss. Moreover, the `GLM policy + telematics` model confirms the interest of the identified interactions. Indeed, the variable containing the leaves was selected in the best model. Moreover, we can see that leaves 7 to 11, which code for high claim frequencies, are considered significant by the `GLM policy + telematics` model.

Thus, by integrating these new interactions, we have improved the `GLM policy` frequency model by approximately 2% according to the Poisson deviance loss. This may not seem to be much, but tariff variables generally have a fairly low predictive power. Thanks to DefragTrees, we were able to find significant interactions for the GLM and thus enrich it. This makes it an efficient extraction process since it requires little human intervention.

Tab. 2.6: Statistical models

	GLM Policy	GLM Policy + telematics
(Intercept)	-2.29^{***} (0.14)	-3.17^{***} (0.36)
Gender male	0.32^{***} (0.08)	0.26^{**} (0.08)
Flemish Brabant	0.24 (0.22)	0.27 (0.22)
Walloon Brabant	0.16 (0.21)	0.19 (0.21)
West Flanders	-0.24 (0.19)	-0.17 (0.19)
East Flanders	-0.14 (0.19)	-0.12 (0.19)
Hainaut	-0.24 (0.15)	-0.21 (0.15)
Liège	0.04 (0.16)	0.04 (0.16)
Limburg	-0.12 (0.21)	-0.05 (0.21)
Luxembourg	-0.61 (0.33)	-0.53 (0.33)
Namur	-0.34 (0.22)	-0.29 (0.22)
Brussels-Capital	0.26 (0.20)	0.25 (0.20)
License seniority [2.6, 7.5)	-0.38^{***} (0.08)	-0.40^{***} (0.08)
License seniority [7.5, 12)	-1.40^{***} (0.41)	-1.38^{***} (0.41)
Leaf1		0.50 (0.35)
Leaf2		0.87^{*} (0.39)
Leaf3		0.68 (0.36)
Leaf4		0.82^{*} (0.36)
Leaf5		0.68 (0.35)
Leaf6		0.90^{*} (0.38)
Leaf7		1.07^{**} (0.37)
Leaf8		1.12^{**} (0.36)
Leaf9		1.13^{**} (0.41)
Leaf10		1.39^{***} (0.35)
Leaf11		1.90^{***} (0.36)
AIC	4897.47	4824.36
BIC	5001.60	5010.31
Log-likelihood	-2434.73	-2387.18
Deviance	3679.35	3584.25
Num. obs.	12555	12555
5-fold strat. CV - Poisson deviance Loss	0.2952532	0.2898846

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

2.4 Conclusion

In this article, we introduced DefragTrees, a method for extracting a surrogate model from a large class of black-box models, namely, additive tree models. We modified the initial method to produce a tree-like output that is easier to understand for a human. Furthermore, we believe that this method is more consistent than the current standards used in insurance companies such as the variable importance and partial dependence plots. Indeed, DefragTrees tries to produce a tree that is faithful to the ATM black-box while respecting the splits already made by the set of trees.

Through our case study, we managed to show that it is a useful tool to understand an ATM black-box. Indeed, by displaying a clear decision chain for a human, it becomes easier to understand tree-ensemble models. Moreover, this method is very useful in the case of new data such as telematics data. These kinds of data often contain many numerical predictors that are often nonlinearly related to the target. If we wanted to introduce them into a GLM model, for example, it would require considerable analysis and discretization to achieve the best out of them. By extracting a clear surrogate from our Poisson random forest fitted on the totality of the data, we were able to both understand the relationships found by the ATM black-box and to reinject these relationships into a GLM model to improve its performance. Finally, we audited the various regions that form the gross partition of our surrogate model. Focusing on predictors known to actuaries, we found that gender appears as a proxy for behavioral variables. This could potentially be used to create more ethical models.

Acknowledgments

We thank the two anonymous reviewers for their careful reading of our manuscript and their insightful and relevant suggestions.

2.5 Appendix

2.5.1 Results on partitions

Proof 1 *The proof is organized in two parts. First, we prove that the reunion of $\tilde{\mathcal{R}}_g$ elements covers $\mathbb{R}^p$, i.e.,*

- *For each $x \in \mathbb{R}^p$, there exists $i \in \{1, .., I_1\}$ and $k \in \{1, .., I_2\}$ such that $x \in \tilde{R}_{1,i} \cap \tilde{R}_{2,k}$.*

Subsequently, we prove that the elements of $\tilde{\mathcal{R}}_g$ are pairwise disjoint, i.e.,

- *if $\tilde{R}_{g,1} \in \tilde{R}_g$ and $\tilde{R}_{g,2} \in \tilde{R}_g$ then $\tilde{R}_{g,i_1} \cap \tilde{R}_{g,i_2} = \emptyset$.*

Let $x \in \mathbb{R}^p$, $\tilde{\mathcal{R}}_1$ be a partition of $\mathbb{R}^p$, so there exists $i \in \{1, I_1\}$ such that $x \in \tilde{R}_{1,i}$. Symmetrically, there exists $k \in \{1, I_2\}$ such that $x \in \tilde{R}_{2,k}$. Hence, $x \in \tilde{R}_{1,i} \cap \tilde{R}_{2,k}$. It means that $\tilde{R}_{1,i} \cap \tilde{R}_{2,k} \neq \emptyset$ and prove the first point.

Now, let us suppose that $\tilde{R}_{g,1} \in \tilde{\mathcal{R}}_g$ and $\tilde{R}_{g,2} \in \tilde{\mathcal{R}}_g$. There exists $i_1, i_2 \in \{1, I_1\}$ and $k_1, k_2 \in \{1, I_2\}$ such that $\tilde{R}_{g,1} = \tilde{R}_{1,i_1} \cap \tilde{R}_{2,k_1}$ and $\tilde{R}_{g,2} = \tilde{R}_{1,i_2} \cap \tilde{R}_{2,k_2}$ where $\tilde{R}_{1,i_1}, \tilde{R}_{1,i_2} \in \tilde{\mathcal{R}}_1$ and $\tilde{R}_{2,k_1}, \tilde{R}_{2,k_2} \in \tilde{\mathcal{R}}_2$. Suppose that $\tilde{R}_{g,1} \cap \tilde{R}_{g,2} \neq \emptyset$, then there exists $x \in \mathbb{R}^p$ such that

$$x \in \tilde{R}_{g,1} \cap \tilde{R}_{g,2} = (\tilde{R}_{1,i_1} \cap \tilde{R}_{1,i_2}) \cap (\tilde{R}_{2,k_1} \cap \tilde{R}_{2,k_2})$$

It means that $x \in \tilde{R}_{1,i_1} \cap \tilde{R}_{1,i_2}$ and hence that $\tilde{R}_{1,i_1} \cap \tilde{R}_{1,i_2} \neq \emptyset$. However, $\tilde{\mathcal{R}}_1$ is a partition. Therefore, the only way that $\tilde{R}_{1,i_1} \cap \tilde{R}_{1,i_2} = \emptyset$ is that $\tilde{R}_{1,i_1} = \tilde{R}_{1,i_2}$. The same argument holds for $\tilde{R}_{2,k_1} \cap \tilde{R}_{2,k_2}$. It follows directly that $\tilde{R}_{g,1} = \tilde{R}_{g,2}$. Hence, we prove that if two elements of $\tilde{\mathcal{R}}_g$ are not disjoint, they are equal. This concludes the proof.

Tab. 2.7: Table of predictors

Claims information	
`Claims`	Number of MTPL claims reported at fault during the policy period
Policy variables	
`Exposure`	The fraction of the year during which the insured was observed (at most 1)
`Age`	Driver's age
`License seniority`	The duration since the license was obtained
`Area`	The provinces of Belgium and the capital Brussels
`Fuel`	Fuel type (*diesel* or *petrol*)
`Gender`	Gender of the insured (*male* or *female*)
`Kwatt`	Power of the vehicle
Telematics variables	
`Distance`	Distance driven by the insured during the policy period (in meters)
`Yearly distance`	Rescaled `Distance` to a full year (in meters)
`Trips`	Number of trips of the insured during the policy period
`Average distance`	Distance driven on average by trip (`Distance/Trips`)
`Road type`	Division of `Distance` into 4 road types (*motorways*, *urban areas*, *abroad*, and *other*). We also have the proportion of meters within each type of road.
`Time slot`	Division of the `Distance` into 5 time slots (*22h-6h*, *6h-9h30*, *9h30-16h*, *16h-19h* and *19h-22h*). We also have the proportion of meters within each time slot.
`Month`	Division of the `Distance` into 12 months (*january* to *december*).
`Season`	Division of the `Distance` into 4 seasons (*spring*, *summer*, *fall*, *winter*).
`Week/weekend`	Division of `Distance` into week (*monday* to *friday*) and weekend (*saturday* to *sunday*)

2.5.2 Exploratory data analysis

Exposure variable selection

To visualize the relationships between the claim frequency and explanatory variables, we represent a scatter plot of the log claim frequency as a function of each continuous policy predictor: `License seniority`, `Age`, and `Age registration`. Those graphs allow us to see immediately that there is no log-linear relationship between these variables and the target. Hence, we need to discretize continuous predictors to utilize the complex relationship between predictors and the target. As we can see, there is a clear trend for `License seniority`: the more experienced the driver is, the lower the claim frequency. This is less obvious for `Age` and `Age registration`. This is why we choose to discretize our variables with a tree- based method described below.

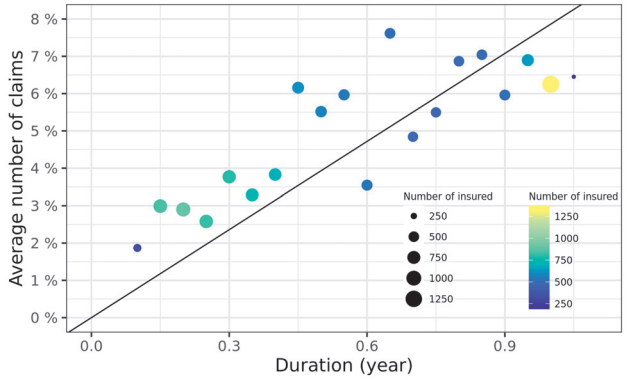

Fig. 2.25: Time as exposure

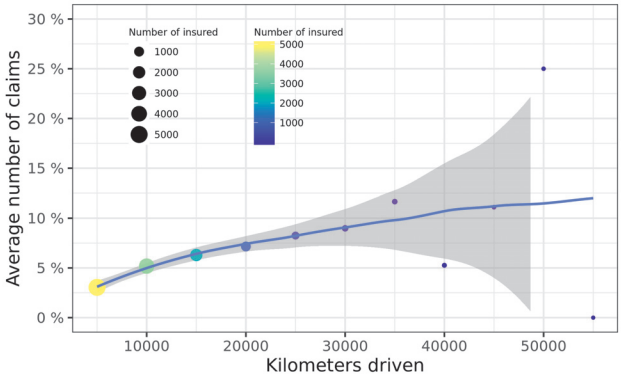

Fig. 2.26: Distance as exposure

Remark 19 *This dataset contains very few points. Sometimes, for the highest values of continuous predictors, the claim frequency is null because no claim was reported. To represent this in* ***Fig. 2.3****, we remove those points because the log is not defined for these observations. Moreover, the* `Kwatt` *variable has too many zeros to be useful in a scatter plot. Therefore, we decided to represent only the discretized variable.*

2.5.3 Grid search

Tab. 2.8: Table of allowed random forest hyperparameters for grid search.

Hyperparameter	Set of values
`nodesize`	$\{50, 500, 800, 1000, 1200, 2000\}$
`ntree`	$\{30, 50, 100, 200\}$
`maxnodes`	$\{2, 4, 8, 12, 16, 32\}$
`mtry`	$\{\lfloor p * 0.1 \rfloor, \lfloor p * 0.3 \rfloor, \lfloor p * 0.5 \rfloor, \lfloor p * 0.7 \rfloor\}$

Tab. 2.9: Table of allowed gradient boosting hyperparameters for grid search.

Hyperparameter	Set of values
`learn_rate_opt`	$\{0.01, 0.03, 0.1\}$
`max_depth_opt`	$\{1, 2, 3, 4, 5\}$
`sample_rate_opt`	$\{0.7, 0.8, 0.9, 1.0\}$
`min_rows`	$\{50, 100, 500, 1000, 2000, 2500\}$
`col_sample_rate_opt`	$\{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$

Bibliography

Beard, Robert Eric, Teivo Pentikäinen, and Erkki Pesonen (1984). *Risk Theory*. Springer Netherlands. DOI: 10.1007/978-94-011-7680-4. URL: https://doi.org/10.1007%2F978-94-011-7680-4.

Boucher, Jean-Philippe, Steven Coté, and Montserrat Guillen (Sept. 2017). "Exposure as Duration and Distance in Telematics Motor Insurance Using Generalized Additive Models". In: *Risks* 5(4), p. 54. DOI: 10.3390/risks5040054. URL: https://doi.org/10.3390%2Frisks5040054.

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45(1), pp. 5–32.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone (Oct. 1984). *Classification And Regression Trees*. Routledge. DOI: 10.1201/9781315139470. URL: https://doi.org/10.1201%2F9781315139470.

Chen, Tianqi and Carlos Guestrin (Aug. 2016). "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145%2F2939672.2939785.

Chipman, Hugh A., Edward I. George, and Robert E. McCulloch (Mar. 2010). "BART: Bayesian additive regression trees". In: *The Annals of Applied Statistics* 4(1), pp. 266–298. DOI: 10.1214/09-aoas285. URL: https://doi.org/10.1214%2F09-aoas285.

Cui, Zhicheng, Wenlin Chen, Yujie He, and Yixin Chen (Aug. 2015). "Optimal Action Extraction for Random Forests and Boosted Trees". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2783258.2783281. URL: https://doi.org/10.1145%2F2783258.2783281.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39(1), pp. 1–22.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608*.

Frees, Edward W., Glenn Meyers, and Richard A. Derrig, eds. (2016). *Predictive Modeling Applications in Actuarial Science*. Cambridge University Press. DOI: 10.1017/cbo9781139342681. URL: https://doi.org/10.1017%2Fcbo9781139342681.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (Apr. 2000). "Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors)". In: *The Annals of Statistics* 28(2), pp. 337–407. DOI: 10.1214/aos/1016218223. URL: https://doi.org/10.1214%2Faos%2F1016218223.

Friedman, Jerome H. (2001). "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29(5), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

Geurts, Pierre, Damien Ernst, and Louis Wehenkel (Mar. 2006). "Extremely randomized trees". In: *Machine Learning* 63(1), pp. 3–42. DOI: 10.1007/s10994-006-6226-1. URL: https://doi.org/10.1007%2Fs10994-006-6226-1.

Goldstein, Alex, Adam Kapelner, Justin Bleich, and Emil Pitkin (Jan. 2015). "Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation". In: *Journal of Computational and Graphical Statistics* 24(1), pp. 44–65. DOI: 10.1080/10618600.2014.907095. URL: https://doi.org/10.1080%2F10618600.2014.907095.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi (Jan. 2019). "A Survey of Methods for Explaining Black Box Models". In: *ACM Computing Surveys* 51(5), pp. 1–42. DOI: 10.1145/3236009. URL: https://doi.org/10.1145%2F3236009.

Hara, Satoshi and Kohei Hayashi (2016). “Making tree ensembles interpretable”. In: *arXiv preprint arXiv:1606.05390*.

Hara, Satoshi and Kohei Hayashi (2018). “Making tree ensembles interpretable: A bayesian model selection approach”. In: *International conference on artificial intelligence and statistics*. PMLR, pp. 77–85.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Henckaerts, Roel, Katrien Antonio, Maxime Clijsters, and Roel Verbelen (2018). “A data driven binning strategy for the construction of insurance tariff classes”. In: *Scandinavian Actuarial Journal* 2018(8), pp. 681–705. DOI: 10.1080/03461238.2018.1429300. eprint: https://doi.org/10.1080/03461238.2018.1429300. URL: https://doi.org/10.1080/03461238.2018.1429300.

Henckaerts, Roel, Marie-Pier Coté, Katrien Antonio, and Roel Verbelen (July 2020). “Boosting Insights in Insurance Tariff Plans with Tree-Based Machine Learning Methods”. In: *North American Actuarial Journal*, pp. 1–31. DOI: 10.1080/10920277.2020.1745656. URL: https://doi.org/10.1080%2F10920277.2020.1745656.

Jacobs, Robert A., Michael I. Jordan, and Andrew G. Barto (Apr. 1991). “Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks”. In: *Cognitive Science* 15(2), pp. 219–250. DOI: 10.1207/s15516709cog1502_2. URL: https://doi.org/10.1207%2Fs15516709cog1502_2.

Jacobs, Robert A., Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton (Feb. 1991). “Adaptive Mixtures of Local Experts”. In: *Neural Computation* 3(1), pp. 79–87. DOI: 10.1162/neco.1991.3.1.79. URL: https://doi.org/10.1162%2Fneco.1991.3.1.79.

Jordan, M.I. and R.A. Jacobs (n.d.). “Hierarchical mixtures of experts and the EM algorithm”. In: *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*. IEEE. DOI: 10.1109/ijcnn.1993.716791. URL: https://doi.org/10.1109%2Fijcnn.1993.716791.

Jordan, Michael I and Robert A Jacobs (1992). “Hierarchies of adaptive experts”. In: *Advances in neural information processing systems*. Citeseer, pp. 985–992.

Lundberg, Scott and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *arXiv preprint arXiv:1705.07874*.

McLachlan, Geoffrey and David Peel (Sept. 2000). *Finite Mixture Models*. John Wiley & Sons, Inc. DOI: 10.1002/0471721182. URL: https://doi.org/10.1002%2F0471721182.

Miller, Tim (Feb. 2019). “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267, pp. 1–38. DOI: 10.1016/j.artint.2018.07.007. URL: https://doi.org/10.1016%2Fj.artint.2018.07.007.

Noll, Alexander, Robert Salzmann, and Mario V. Wuthrich (2018). “Case Study: French Motor Third-Party Liability Claims”. In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3164764. URL: https://doi.org/10.2139%2Fssrn.3164764.

Ohlsson, Esbjörn and Björn Johansson (2010). *Non-Life Insurance Pricing with Generalized Linear Models*. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-10791-7. URL: https://doi.org/10.1007%2F978-3-642-10791-7.

Ribeiro, Marco, Sameer Singh, and Carlos Guestrin (2016). ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 2016 Conference of the North American*

Chapter of the Association for Computational Linguistics: Demonstrations. Association for Computational Linguistics. DOI: 10.18653/v1/n16-3020. URL: https://doi.org/10.18653%2Fv1%2Fn16-3020.

Therneau, Terry M, Elizabeth J Atkinson, et al. (2015). *An introduction to recursive partitioning using the Rpart routines*. Tech. rep. Technical report Mayo Foundation.

Tselentis, Dimitrios I., George Yannis, and Eleni I. Vlahogianni (2016). "Innovative Insurance Schemes: Pay as/how You Drive". In: *Transportation Research Procedia* 14, pp. 362–371. DOI: 10.1016/j.trpro.2016.05.088. URL: https://doi.org/10.1016%2Fj.trpro.2016.05.088.

Verbelen, Roel, Katrien Antonio, and Gerda Claeskens (Apr. 2018). "Unravelling the predictive power of telematics data in car insurance pricing". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67(5), pp. 1275–1304. DOI: 10.1111/rssc.12283. URL: https://doi.org/10.1111%2Frssc.12283.

Vickrey, William (1968). "Automobile Accidents, Tort Law, Externalities, and Insurance: An Economist's Critique". In: *Law and Contemporary Problems* 33(3), p. 464. DOI: 10.2307/1190938. URL: https://doi.org/10.2307%2F1190938.

Wuthrich, Mario V. and Christoph Buser (2017). "Data Analytics for Non-Life Insurance Pricing". In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.2870308. URL: https://doi.org/10.2139%2Fssrn.2870308.

Yang, Yi, Wei Qian, and Hui Zou (May 2017). "Insurance Premium Prediction via Gradient Tree-Boosted Tweedie Compound Poisson Models". In: *Journal of Business & Economic Statistics* 36(3), pp. 456–470. DOI: 10.1080/07350015.2016.1200981. URL: https://doi.org/10.1080%2F07350015.2016.1200981.

Tail-index partition-based rules extraction

3

> *When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one.*
>
> — **Vladmir Vapnik**

Abstract

The tail index is an important parameter that measures how extreme events occur. In many applied fields, this tail index depends on covariates. In this paper we assume that it takes a finite number of values over a partition of the covariate space. This article proposes a tail-index partition-based rules extraction method that is able to construct estimates of the partition subsets and estimates of the tail index values. The method combines two steps: first an additive tree ensemble based on the Gamma deviance is fitted (which includes random forest and gradient tree boosting), second a hierarchical clustering with spatial constraints is used to estimate the subsets of the partition. We also propose a global tree surrogate model to approximate the partition-based rules while providing an explainable model from the initial covariates. Our procedure is illustrated on simulated data. A real case study that is valuable for insurance of wind damage caused by tornadoes is also presented.

Keywords: Tail index, Additive tree ensembles, Partitioning methods, XAI.

3.1 Introduction

A central topic in extreme value statistics is the estimation of the tail index that is directly related to the tail behavior of random events. It is often assumed that this tail index is a constant independent on explanatory variables while it is observed in many applied fields that covariates play an important role in leading to extreme events. In this paper we are interested in the tail index estimation for heavy tailed (i.e. Pareto-type) models when a multivariate random covariate $\mathbf{X}$ is observed simultaneously with the variable of interest Y. We assume that the tail index function takes a finite number of values over a partition of the covariate space.

Non-parametric local estimators of the tail index function using Kernel regression methods and classical estimators of the tail index without covariables have already been proposed for a decade. Local Hill type estimators have been introduced and studied in Goegebeur, Guillou, and Schorgen (2013), Goegebeur, Guillou, and Stupfler (2015), Gardes and Stupfler (2013). Stupfler (2013) considered the moment estimator introduced in Dekkers et al. (1989) and provided estimators for the three domains of attraction. Daouia et al. (2010) studied the estimation of extreme quantiles under a conditional Pareto-type model with random covariates and plugged a fixed number of such quantile estimates in Pickands and Hill estimators (Pickands (1975), Hill (1975)). Gardes and Girard (2012) generalized their method to the case when the covariate space is infinite-dimensional. More recently Farkas et al. (2021) have proposed a regression tree method where the quadratic loss used in the "growing" phase of the tree has been replaced by a log-likelihood loss based on the likelihood of Generalized Pareto distributions.

However such previous non-parametric estimators of the tail index function are not able take into account the assumption that this function only takes a finite number of values over the covariate space. The aim of this paper is to provide a method to estimate both the partition subsets of the covariate space as well as the values of the tail index function.

Rule-based methods are a popular class of techniques in machine learning and data mining that share the goal of finding regularities in data that can be expressed in the form of simple rules. The most common example is the IF-THEN rule which, from a condition based on the covariate $\mathbf{X}$, provides an associate estimates value for Y. Regression trees typically generate such rules where the condition is build from intersections of sub-conditions like "the i-th component of $\mathbf{X}$ is larger or smaller than a specific threshold". Although these conditional statements can easily be understood by humans, they generate a partition of the covariate space composed of rectangles that are not necessarily suitable to depict regularities in data.

In this paper we are interested in a partition-based method which, from a small-size partition of the covariate space, provides an accurate prediction for Y for any subset of the partition. We propose a partition-based rules extraction method that combines two steps: first an additive tree ensemble based on a specific deviance is fitted (which includes random forest and gradient tree boosting), second a hierarchical clustering with spatial constraints is used to estimate the subsets of the partition.

Although our procedure provides a finite number of subsets of the covariate space and can make accurate predictions by modeling the complex structure of the partition, it can be viewed as a black-box model producing predictions without explaining how the partition subsets have been made from the covariate vector $\mathbf{X}$. Interpretability techniques can then come into play by providing a lens through which the complex structure of the partition can be viewed. Therefore we also propose a global tree surrogate model to approximate the partition-based rules while providing an explainable model from the initial covariates. This surrogate model combines a binary encoder representation of the additive tree ensemble with a regression tree.

The rest of this paper is structured as follows. Section 2 presents the model and its assumptions, and then details the methodology. Section 3 first illustrates our approach on simulated data and then provides an application to insurance that is valuable for wind damage caused by tornadoes. Section 4 concludes this paper.

3.2 Methodology

The goal of supervised learning is to predict a scalar random variable Y by a covariate vector $\mathbf{X} \in \mathcal{X}$ where $\mathcal{X}$ is called the covariate space and is assumed to be included in $\mathbb{R}^p$. We denote by $\mathcal{P} = \{\mathcal{A}_i : i = 1, \ldots, I\}$ a small-size partition of $\mathcal{X}$.

3.2.1 Model

Let Z be a positive, real-valued and heavy-tailed random variable. We assume that its conditional distributions given $\mathbf{X}$ are characterized in the following way

$$P\left(Z > z | \mathbf{X} = \mathbf{x}\right) = z^{-\alpha(\mathbf{x})} L\left(z; \mathbf{x}\right), \quad z > 0, \mathbf{x} \in \mathcal{X},$$

where

$$\alpha\left(\mathbf{x}\right) = \sum_{i=1}^{I} \alpha_i \mathbb{I}_{\{\mathbf{x} \in \mathcal{A}_i\}} > 0, \quad \mathbf{x} \in \mathcal{X},$$

is the (unknown) tail index function that characterizes the dependence of the tail behavior of Z on $\mathbf{X}$, and $L\left(z; \mathbf{x}\right)$ are slowly varying functions in the sense that $\lim_{z \to \infty} L\left(tz; \mathbf{x}\right) / L\left(z; \mathbf{x}\right) = 1$ for any $t > 0$ and $\mathbf{x} \in \mathcal{X}$.

We seek to estimate the Hill index function $\xi\left(\mathbf{x}\right) = \alpha^{-1}\left(\mathbf{x}\right)$ from a set of independent random variables $\mathcal{D}_n = \{(Z_i, \mathbf{X}_i)_{i=1,\ldots,n}\}$ distributed as the independent pair $(Z, \mathbf{X})$. To do this, we introduce a family of positive threshold functions $t_u\left(\cdot\right) = ut\left(\cdot\right)$ with $u > 0$ and where $t : \mathcal{X} \to \mathbb{R}_+$ satisfies $\inf_{\mathbf{x} \in \mathcal{X}} t\left(\mathbf{x}\right) > 0$. We only keep the observations $(Z_i, \mathbf{X}_i)$ for which $Z_i > t_u\left(\mathbf{X}_i\right)$ for a large u. Let us define $Y^{(u)} = \ln\left(Z / t_u\left(\mathbf{X}\right)\right)$ given $Z > t_u\left(\mathbf{X}\right)$ and note that the distribution of $Y^{(u)}$ may be approximated by an Exponential distribution with mean $\xi\left(\mathbf{x}\right)$ since

$$\lim_{u \to \infty} P(Y^{(u)} > y | \mathbf{X} = \mathbf{x}) = e^{-\alpha(\mathbf{x}) y}, \quad y > 0.$$

We will therefore work with the set of observations $\mathcal{D}_n^{(u)} = \{(Y_i^{(u)}, \mathbf{X}_i) \in \mathcal{D}_n : Z_i > t_u\left(\mathbf{X}_i\right)\}$ in order to build an estimate $\xi_n^{(u)} : [0,1]^p \to \mathbb{R}_+$ of the Hill index function ξ, and we will use an appropriate loss function adapted to Exponential distributions.

For this purpose, we consider the Gamma deviance. A deviance D is a bivariate function that satisfies the following conditions: $D\left(y, y\right) = 0$ and $D\left(y, \xi\right) > 0$ for $y \neq \xi$. In statistics, the deviance is used

to build goodness-of-fit statistics for a statistical model. It plays an important role in exponential dispersion models and generalized linear models. Let $f(y;\xi)$ be the density function of an Exponential random variable with mean ξ. The Gamma deviance function is defined by

$$D(y,\xi) = 2\left(\ln f(y;y) - \ln f(y;\xi)\right) = 2\left[\frac{y-\xi}{\xi} - \ln\left(\frac{y}{\xi}\right)\right], \quad y,\xi > 0.$$

Note that $D(y,\xi) \sim (y-\xi)^2$ as $y \to \xi$, and therefore the Gamma deviance and the L^2 distance are equivalent when y is close to ξ. The Gamma deviance is not only more appropriate than the L^2 distance because the observations $Y_i^{(u)}$ are asymptotically distributed as exponential random variables, but also because it prevents the estimates $\xi_n^{(u)}$ from taking negative values.

Thereafter we will consider families of estimators statisfying

$$\xi_n^{(u)}(\cdot) = \arg\min_{f \in \mathcal{F}_n^{(u)}} \sum_{i \in \mathcal{I}_n^{(u)}} D(Y_i^{(u)}, f(\mathbf{X}_i))$$

where $\mathcal{I}_n^{(u)} = \{i : (Y_i^{(u)}, \mathbf{X}_i) \in \mathcal{D}_n^{(u)}\}$ and $\mathcal{F}_n^{(u)} = \mathcal{F}_n(\mathcal{D}_n^{(u)})$ is a class of functions $f : \mathcal{X} \to \mathbb{R}_+$ that may depend on the data $\mathcal{D}_n^{(u)}$.

3.2.2 Tree-based tail index estimators

Tree-based algorithms, such as Classification and Regression Trees (CART) Breiman et al. (1984), random forests Breiman (2001) and boosted regression trees Friedman (2001), are popularly used in all kinds of data science problems because they are considered to be one of the best supervised learning methods. They constitute a class of predictive models with high accuracy, stability and capability of interpretation.

The CART algorithm is the cornerstone of this class of algorithms. It makes a tree by splitting the sample into two or more homogeneous sets based on most significant splitter/differentiator in explanatory variables. Both random forests and boosted regression trees create tree ensembles by using randomization during the tree creations. However, a random forest builds the trees in parallel and average them on the prediction, whereas a boosted regression tree creates a series of trees, and the prediction receives incremental improvement by each tree in the series.

Tail regression trees

A decision tree is derived from a rule-based method that generates a binary tree through a recursive partitioning algorithm that splits subsets (called nodes) of the data set into two subsets (called sub-nodes) according to the minimization of a split/heterogeneity criterion computed on the resulting

sub-nodes. The root of the tree is $\mathcal{X}$ itself. Each split involves a single variable. While some variables may be used several times, others may not be used at all.

For tail index regression, the split criterion will be based on the Gamma deviance. To properly define it, we let A be a generic node and $n^{(u)}(A)$ be the number of data points of $\mathcal{I}_n^{(u)}$ falling in A. A cut in A is a pair (j,x), where j is an element of $\{1,...,p\}$, and x is the position of the cut along coordinate j, within the limits of A. Let $\mathcal{C}_A$ be the set of all such possible cuts in A. Then, for any $(j,x)\in\mathcal{C}_A$, the Gamma deviance split criterion takes the form

$$\begin{aligned} L_n(j,x) &= \frac{1}{n^{(u)}(A)}\sum_{i\in\mathcal{I}_n^{(u)}} D\left(Y_i^{(u)},\bar{Y}(A)\right)\mathbb{I}_{\{\mathbf{X}_i\in A\}} - \frac{1}{n^{(u)}(A)}\sum_{i\in\mathcal{I}_n^{(u)}} D\left(Y_i^{(u)},\bar{Y}(A_L)\right)\mathbb{I}_{\{\mathbf{X}_i\in A, X_i^{(j)}\leq x\}} \\ &\quad -\frac{1}{n^{(u)}(A)}\sum_{i\in\mathcal{I}_n^{(u)}} D\left(Y_i^{(u)},\bar{Y}(A_U)\right)\mathbb{I}_{\{\mathbf{X}_i\in A, X_i^{(j)}>x\}}, \end{aligned}$$

where $A_L=\{\mathbf{x}\in A : x_i^{(j)}\leq x\}$, $A_L=\{\mathbf{x}\in A : x_i^{(j)}>x\}$, and $\bar{Y}(A)$ (resp., $\bar{Y}(A_L)$, $\bar{Y}(A_U)$) is the average of the $Y_i^{(u)}$'s belonging to A (resp., A_L, A_U). Note that $L_n(j,x)$ simplifies in the following way

$$L_n(j,x)=\ln\left(\bar{Y}(A)\right)-\frac{n^{(u)}(A_L)}{n^{(u)}(A)}\ln\left(\bar{Y}(A_L)\right)-\frac{n^{(u)}(A_U)}{n^{(u)}(A)}\ln\left(\bar{Y}(A_U)\right).$$

At each node A, the best cut (j_n^*,x_n^*) is finally selected by maximizing $L_n(j,x)$ over $\mathcal{C}_A$. The best cut is always performed along the best cut direction j_n^*, at the middle of two consecutive data points.

Let us denote by $\mathcal{T}_n^{(u)}=\{A_{i,n}^{(u)} : i=1,\ldots,I_n^{(u)}\}$ the partition of $\mathcal{X}$ obtained where $I_n^{(u)}$ is the total number of leaf nodes in the tree. Then the estimate of ξ takes the form of a piecewise step function

$$\xi_n^{(u)}(\mathbf{x};\mathcal{T}_n^{(u)})=\sum_{i=1}^{I_n^{(u)}}\bar{Y}(A_{i,n}^{(u)})\mathbb{I}_{\{\mathbf{x}\in A_{i,n}^{(u)}\}},$$

where $\mathbb{I}_{\{\mathbf{x}\in A_{i,n}^{(u)}\}}$is the indicator function that $\mathbf{x}$ is in leaf node $A_{i,n}^{(u)}$ of the tree partition.

Decision tree output is very easy to understand and does not require any statistical knowledge to read and interpret it. Decision tree is one of the fastest way to identify most significant variables and relationships between two or more variables. One of the most practical issues is overfitting. A first way to solve it is to set constraints on model parameters: the users may e.g. fix the minimum number of observations which are required in a node to be considered for splitting, or the maximum depth of the tree (the number of edges from the root node to the leaf nodes of the tree), or else the maximum number of terminal nodes or leaves in the tree. A second way is to use pruning that consists in reducing the size of the decision tree by removing sections of the tree that are non-critical. By reducing the complexity, pruning improves predictive accuracy and mitigates overfitting.

Tail random forest Maillart and Robert (2021))

A random forest is a predictor consisting in a collection of several randomized regression trees. Let Θ be a random variable independent of $\mathcal{D}_n^{(u)}$ that will characterize the set of covariates among the components of $\mathbf{X} = (X^{(1)}, ..., X^{(p)})$ and the set of observations among $\mathcal{D}_n^{(u)}$ that will be used to build a tail regression tree. Let $\Theta_1, \ldots, \Theta_M$ be independent random variables, distributed as Θ and independent of $\mathcal{D}_n^{(u)}$. For the j-th tree in the collection $\mathcal{T}_n^{(u)}(\Theta_j)$, we denote by $\xi_n^{(u)}(\cdot;\Theta_j, \mathcal{D}_n^{(u)})$ the estimate of ξ.

For our tail index regression problem, the trees will be combined through a harmonic mean to form the forest estimate

$$\frac{1}{\xi_{M,n}^{(u)}(\mathbf{x};\Theta_1, \ldots, \Theta_M)} = \frac{1}{M}\sum_{j=1}^{M} \frac{1}{\xi_n^{(u)}(\mathbf{x};\Theta_j, \mathcal{D}_n^{(u)})}, \tag{3.1}$$

or equivalently

$$\alpha_{M,n}^{(u)}(\mathbf{x};\Theta_1, \ldots, \Theta_M, \mathcal{D}_n^{(u)}) = \frac{1}{M}\sum_{j=1}^{M} \alpha_n^{(u)}(\mathbf{x};\Theta_j, \mathcal{D}_n^{(u)})$$

where $\alpha_{M,n}^{(u)}$ and $\alpha_n^{(u)}$ denote the respective tail index estimates. Note that such an aggregation is different from the one done for the usual random forest and ensures that the Gamma deviance loss of $\xi_{M,n}^{(u)}$ will be smaller than the average of the individual Gamma deviance losses of the $\xi_n^{(u)}$'s (see Maillart and Robert (2021)). Let us denote by $\mathcal{R}_n^{(u)} = \{B_{j,n}^{(u)} : j = 1, \ldots, J_n^{(u)}\}$ the partition of rectangles of $\mathcal{X}$ obtained from crossing the partitions of the regression trees $\mathcal{T}_n^{(u)}(\Theta_1), \ldots, \mathcal{T}_n^{(u)}(\Theta_M)$. The tail random forest estimate of ξ also takes the form of a piecewise step function

$$\xi_{M,n}^{(u)}(\mathbf{x};(\Theta_m)) = \sum_{j=1}^{J_n^{(u)}} b_{j,n}\mathbb{I}_{\{\mathbf{x}\in B_{j,n}^{(u)}\}},$$

where

$$\frac{1}{b_{j,n}} = \frac{1}{M}\sum_{j=1}^{M} \frac{1}{\xi_n^{(u)}(\mathbf{x};\Theta_j, \mathcal{D}_n^{(u)})}\mathbb{I}_{\{\mathbf{x}\in B_{j,n}^{(u)}\}}.$$

As for the usual random forests, the algorithm needs two additional parameters: the number of pre-selected covariates for splitting, the number of sampled data points in a tree. Tail random forests can be used to rank the importance of variables in a natural way as described in Breiman's original paper Breiman (2001). Tail random forests achieve higher accuracy than a single tail regression tree and suffer less from the overfitting issue, but they sacrifice the intrinsic interpretability present in decision trees and may can appear as a black-box approach for statistical modelers.

Tail gradient tree boosting

Tail gradient tree boosting combines weak tree "learners" (small depth tail regression trees) into a single strong learner in the following iterative way:

- Initialize the model with a small depth tail regression tree $\xi_{0,n}^{(u)}$. Let M be an integer.

- For $m = 1, \ldots, M$, compute the pseudo-residuals

$$r_{i,m}^{(u)} = - \left. \frac{\partial D\left(y_i, \xi\right)}{\partial \xi} \right|_{\xi = \xi_{m-1,n}^{(u)}(\mathbf{x}_i)}, \quad i \in \mathcal{I}_n^{(u)}.$$

- At the m-th step, the algorithm fits a regression tree $h_{m,n}^{(u)}\left(\mathbf{x}\right)$ to pseudo-residuals $\left(r_{i,m}^{(u)}\right)_{i \in \mathcal{I}_n^{(u)}}$ such that

$$h_{m,n}^{(u)}\left(\mathbf{x}\right) = \sum_{k=1}^{K_{m,n}^{(u)}} c_{k,m} \mathbb{I}_{\{\mathbf{x} \in C_{k,m,n}^{(u)}\}},$$

where $\mathcal{G}_{m,n}^{(u)} = \{C_{k,m,n}^{(u)} : k = 1, \ldots, K_{m,n}^{(u)}\}$ is the associated partition of the tree and $\left(c_{k,m}\right)_{k=1,\ldots,K_{m,n}^{(u)}}$ are the predicted values in each region.

- The model update rule is then defined as

$$\xi_{m,n}^{(u)}(\mathbf{x}) = \xi_{m-1,n}^{(u)}(\mathbf{x}) + \sum_{k=1}^{K_{m,n}^{(u)}} \gamma_{k,m} \mathbb{I}_{\{\mathbf{x} \in C_{k,m}\}}$$

where

$$\gamma_{k,m} = \arg\min_{\gamma} \sum_{\mathbf{x}_i \in C_{k,m}} D\left(y_i, \xi_{m-1,n}^{(u)}(\mathbf{x}_i) + \gamma\right).$$

One of the outputs of this algorithm is a partition of rectangles of the covariate space: $\mathcal{G}_n^{(u)} = \{C_{j,n}^{(u)} : j = 1, \ldots, K_n^{(u)}\}$ such that the gradient tree boosting estimate of ξ takes the form of a piecewise step function

$$\xi_{M,n}^{(u)}(\mathbf{x};) = \sum_{j=1}^{K_n^{(u)}} c_j \mathbb{I}_{\{\mathbf{x} \in C_{j,n}^{(u)}\}}.$$

The number of gradient boosting iterations M appears as a regularization parameter. Increasing M reduces the error on training set, but setting it too high may lead to overfitting. An optimal value of M is often selected by monitoring prediction error on a separate validation dataset. At each iteration of the algorithm, it is also possible to only consider a subsample of the training set (drawn at random without replacement) to fit the weak tree learner. Friedman (2001) observed a substantial

improvement accuracy with this modification in the case of the usual gradient boosting. Subsampling may introduce randomness into the algorithm and help prevent overfitting, acting also as a kind of regularization. Another useful regularization techniques for gradient boosted trees is to penalize model complexity of the learned model. The model complexity is in general defined as the number of leaves in the learned trees.

3.2.3 Hierarchical clustering with spatial constraints

The tail random forest as well as the tail gradient tree boosting provide as outputs large-size partitions of the covariate space that divide it into a very fine structure which does not however reflect the partition $\mathcal{P} = \{A_i : i = 1, \ldots, I\}$ on which is defined the Hill index function ξ. We must now gather subsets of these too fine partitions to reveal the partition $\mathcal{P}$.

Many methods have been proposed to find a partition according to a dissimilarity-based homogeneity criterion, but in our case it is necessary to impose contiguity constraints (in the covariate space). Such restrictions are needed because the objects in a cluster are required not only to be similar to one other, but also to comprise a contiguous set of objects. How to create contiguous set of objects?

A first approach consists in defining a contiguity matrix $\mathrm{C} = (c_{ij})$ where $c_{ij} = 1$ if the i-th and the j-th objects are regarded as contiguous, and 0 if they are not. A cluster C is then considered to be contiguous if there is a path between every pair of objects in C (the subgraph is connected). Several classical clustering algorithms have been modified to take into account this type of constraint. A survey of some of these methods can be found in Murtagh (1985) and in Gordon (1996). When considering the ordinary hierarchical clustering procedure as a particular case, the relational constraints introduced by the contiguity matrix can however lead to reversals (i.e. inversions, upward branchings in the tree). It was proven that only the complete link algorithm is guaranteed to produce no reversals. Recent implementation of strict constrained clustering procedures are available in the R package `constr.hclust` (Legendre (2011)).

A second approach consists in introducing soft contiguity or spatial constraints. It has been proposed to run clustering algorithms on a modified dissimilarity matrix which would be a combination of the matrix of spatial distances and the dissimilarity matrix computed from non-spatial variables. According to the weight given to the spatial dissimilarities in this combination, the solution will have more or less spatially contiguous clusters. A typical example is the Ward-like hierarchical clustering algorithm including spatial/geographical constraints proposed in Chavent et al. (2018). The authors introduce two dissimilarity matrices D_0 and D_1 and consider a convex combination as the criterion to minimize. The first matrix gives the dissimilarities in the feature space and the second matrix gives the dissimilarities in the spatial space. The value of the weight parameter $\gamma \in [0, 1]$ is determined to increase the spatial contiguity without deteriorating too much the quality of the solution based on the variables of the feature space. The procedure is available in the R package `ClustGeo` (Chavent et al. (2018)).

It is noteworthy that packages which implement tree additive ensemble methods do not provide the large-size partition, but only the set of generated trees. When the number of trees increases, the number of rectangles generated by the intersections of the tree partitions increases tremendously. In practice, a naive approach which consists in testing the crossing of each rectangle with all the other rectangles is computationally too heavy. Another approach is required: we used a classical method from computational geometry known as segment trees. By constructing a tree composed of segments that represent the edges of rectangles, it is possible to infer very efficiently what the rectangles with a non-empty intersection are. This method is described in Zomorodian and Edelsbrunner (2000). An implementation in C++ is available in the CGAL library (The CGAL Project (2021)).

3.2.4 Equivalent tree: a tail tree surrogate

Our procedure described in the previous subsections provides a small-size partition of the covariate space and ultimately makes accurate predictions by modeling the complex structure of the partition. However it is not able to explain easily how the partition subsets have been made from the covariate vector $\mathbf{X}$, and then it can be viewed as a black-box model producing predictions. Therefore we attach to our procedure a global tree surrogate model that approximates the partition-based rules while providing an explainable model using the initial covariates. This surrogate model combines a binary encoder representation of the additive tree ensemble with a regression tree. It is called the equivalent tree model.

Let us denote by R_g a rectangle of the partition of the covariate space obtained from the additive tree ensemble. We are interested in a binary representation of R_g. We assume that the additive tree ensemble has L internal nodes/splits of the form $X_{i_l} > z_l$ for $l = 1, \ldots, L$. The rectangle R_g may then be represented by the binary vector $\boldsymbol{\eta}_g \in \{0,1\}^L$ such that $\mathbf{x} \in R_g$ if $x_{i_l} > z_l$ for $l = 1, \ldots, L$. Fig. 3.1 illustrates this representation on a simple example where $p = 2$ and $L = 4$.

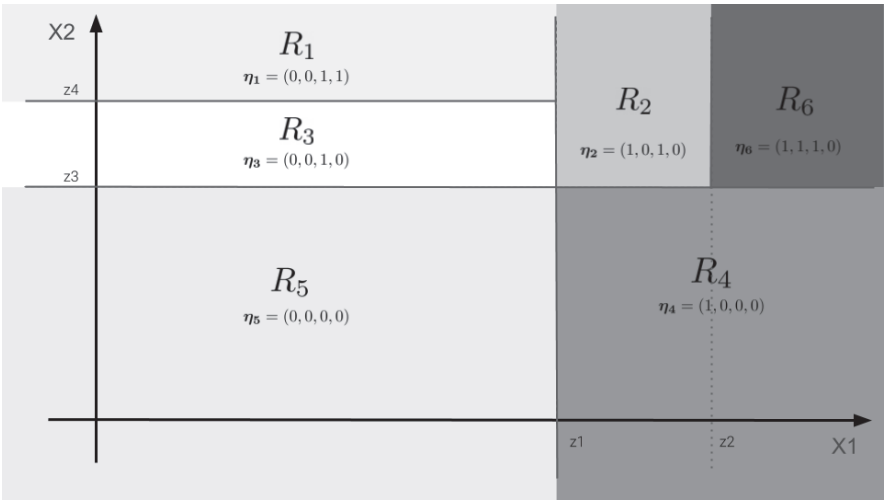

Fig. 3.1: A simple example of the binary representation of the rectangles of the covariate space obtained from an additive tree ensemble.

We now propose an alternative partition based on a regression tree and proceed as follows:

1. we associate to each rectangle R_g its predicted value (as a new label), its binary representation (as a new feature vector) and a weight corresponding to the proportion of the observations $(Y_i^{(u)}, \mathbf{X}_i)$ such that $\mathbf{X}_i$ belongs to it;

2. we build a maximal regression tree from these new data.

By choosing an appropriate depth we obtain an approximation of the large-size partition of the additive tree ensemble with a reasonable explanation. Note that the fidelity measured by the R^2 measure (i.e. the percentage of variance that our surrogate model is able to capture from the additive tree ensemble) increases at each split. Since the regression tree method is applied to binary predictors that are linked to the splits made by the tree-based model, it divides the covariate space from the same rectangles as the tree-based model. The regression tree method thus provides a nested set of trees that approximates faithfully the large-size partition.

It should be noted that additive tree ensemble models natively provide local model interpretations for analyzing predictions. Actually the vector of covariates associated to a prediction belongs to a unique rectangle of the large-size partition whose edges are intervals belonging to the support of the covariate components. Such local explanations can be used to answer questions like: why did the model make this specific prediction? or what effect did this specific feature value have on the prediction?. But additive tree ensemble models need additional surrogate models for good global explanations.

3.3 Numerical experiments

3.3.1 Simulation studies

This section investigates how our approach performs on simulated data. We consider two toy datasets $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$ where $p = 2$, $\mathcal{X} = [0,1] \times [0,1]$, X_1 and X_2 are two independent random variables uniformly distributed over $[0,1]$, and

$$P\left(Z > z | \mathbf{X} = \mathbf{x}\right) = z^{-\alpha(\mathbf{x})}, \quad z > 1, \mathbf{x} \in \mathcal{X}.$$

The partitions $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ of $\mathcal{X}$ are of size 4 and are depicted in the following figure

Since the slowly varying functions of the family of conditional survival distribution functions of Z are equal to 1, it is not necessary to choose a family of thresholds for these datasets and to select observations whose values are higher than the thresholds. The sizes of the datasets are equal to

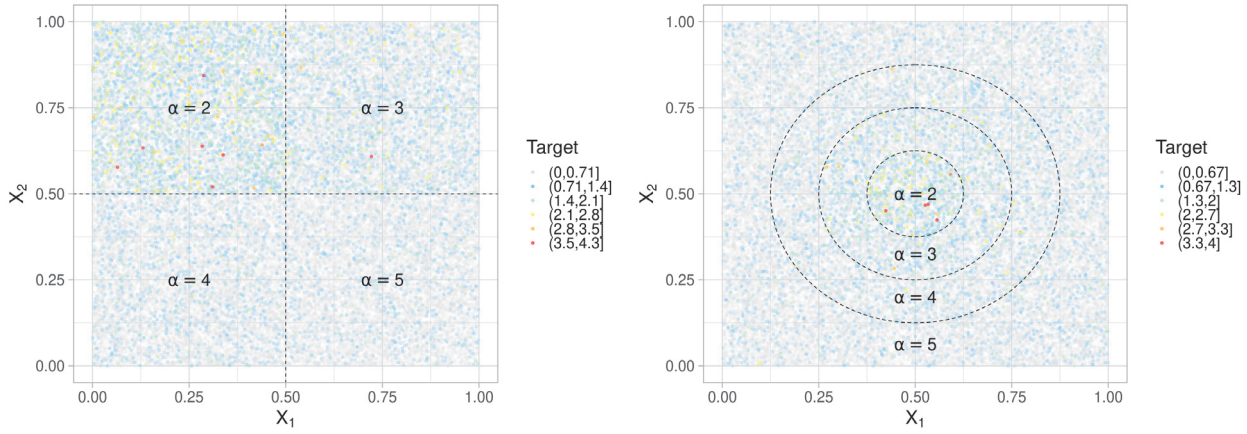

Fig. 3.2: Partitions $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ of X with their associated tail index values.

$n = 50,000$. On **Fig. 3.2**, each point represents an observation (the color of a point is given by the value of its associated observation).

We choose the tail gradient tree boosting method to build the large-size partitions. The gradient tree boosting combines 100 weak tree learners. We have performed a grid-search with a 5-fold cross-validation to find a set of good parameters. We evaluate the performance of the models with the negative Gamma log-likelihood. We have simulated test datasets with the same size to understand how the models generalize. The performances are presented in **Tab. 3.1**.

Tab. 3.1: Performance comparison between models.

Model name	Description	Hyperparameters	5-folds CV ($\times 10^{-2}$)
`GBM Gamma` $\mathcal{D}_1$	Gamma Gradient Boosting fitted on $\mathcal{D}_1$.	`col_sample_rate` = 0.9 `learn_rate` = 0.1 `max_depth` = 4 `sample_rate` = 0.7 `distribution` = "gamma"	-0.35
`GBM Gamma` $\mathcal{D}_2$	Gamma Gradient Boosting fitted on $\mathcal{D}_2$.	`col_sample_rate` = 0.9 `learn_rate` = 0.1 `max_depth` = 4 `sample_rate` = 0.7 `distribution` = "gamma"	-14.81

Fig. 3.3 provides the large-size partitions obtained for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$. The color inside a rectangle is the value of the prediction of the tail index for this rectangle. The tail gradient tree boosting model performs well on $\mathcal{D}^{(1)}$ which is not surprising because the subsets of the partition $\mathcal{P}^{(1)}$ are characterized by intersections of conditions that only depend on X_1 or X_2, but not both. The learning task is more difficult and challenging for $\mathcal{D}^{(2)}$ because the subsets of the partition $\mathcal{P}^{(2)}$ depend on an intricate way of X_1 and X_2.

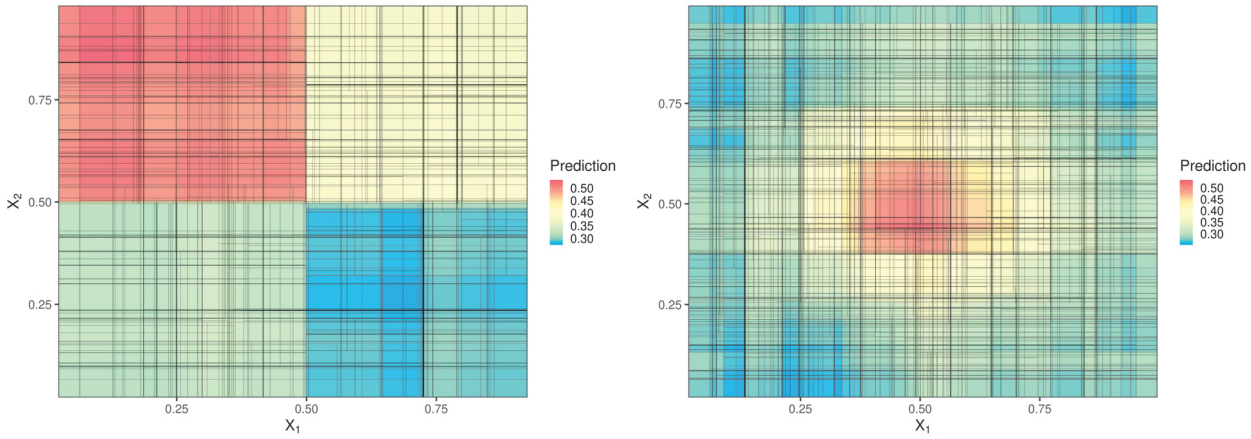

Fig. 3.3: Tail gradient tree boosting partitions for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$.

We then run `ClustGeo` to gather the rectangles of the tail gradient tree boosting partitions in order to reveal the partitions $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$. The matrix D_1 which gives the dissimilarities in the spatial space is defined in the following way: if two rectangles are adjacent, the distance value is set to 0.1 and if this is not the case, the distance value is set to a large value $d = 9.10^6$. **Fig. 3.4** shows the fidelity curves (R^2 measure) between the aggregated partition and the initial partition created by the tail gradient tree boosting for different values of the weight parameter $\gamma \in [0, 1]$. Based on these curves, we choose $\gamma = 0.3$ for $\mathcal{D}^{(1)}$ and $\gamma = 0.1$ for $\mathcal{D}^{(2)}$, while fixing the size K of the small-size partitions to 4. The fidelity is then equal to 94.91% for $\mathcal{D}^{(1)}$ and 91.18% for $\mathcal{D}^{(2)}$.

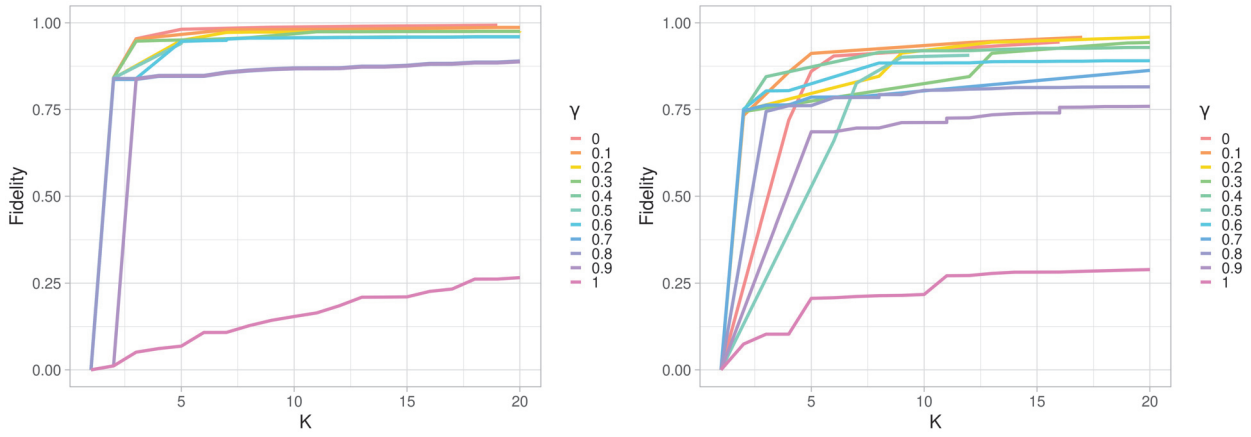

Fig. 3.4: Fidelity curves for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$.

Fig. 3.5 provides the estimated small-size partitions for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$. For $\mathcal{D}^{(1)}$, we observe that the hierarchical clustering algorithm has some difficulties in separating the region where the tail index is equal to 0.2 from the one where it is equal to 0.25. This reason is that these values are actually too close. Moreover it creates a fourth small subset for the partition in the region where the tail index is equal to 0.25 without providing a very different predicted value. A partition with three subsets would therefore be sufficient here. Nevertheless the shapes of the estimated subsets of the partition are close to the shapes of the real subsets. For $\mathcal{D}^{(2)}$, we observe that the circular subsets for the tail indices equal to 0.25, 0.33 and 0.5 are relatively well estimated, but the subset for the value 0.25 is

the union of two disjoint subsets. We conclude that `ClustGeo` does a good job for identifying the small-size partitions.

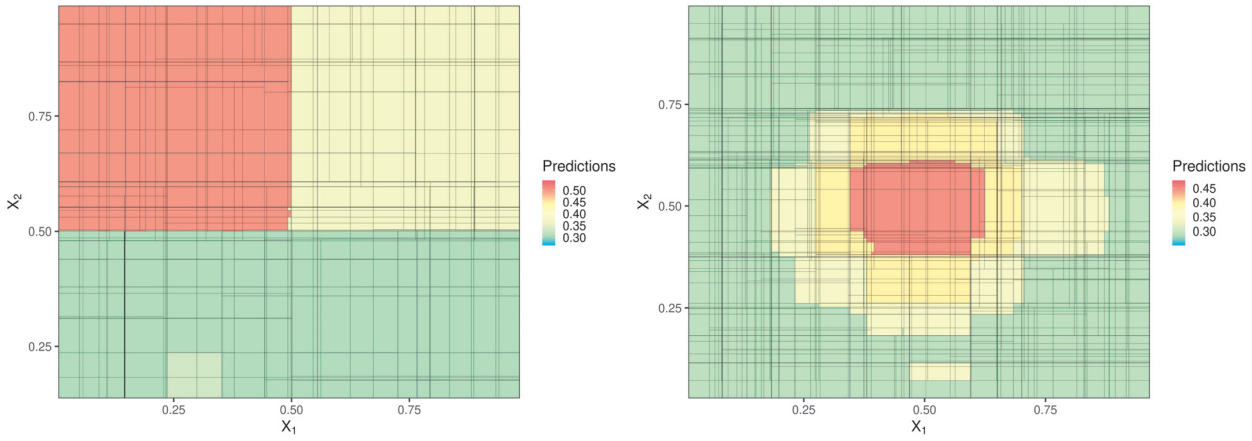

Fig. 3.5: Estimated small-size partitions for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$.

In **Fig. 3.6**, QQ-plots compare the distributions of the observations inside each subsets of the estimated partitions with the Exponential distributions (whose means are equal to the empirical means). The alignment of the points shows good fits and validates the assumption of Exponential distributions for $\log(Z)$. The tail indices for the orange and yellow subsets are very well estimated.

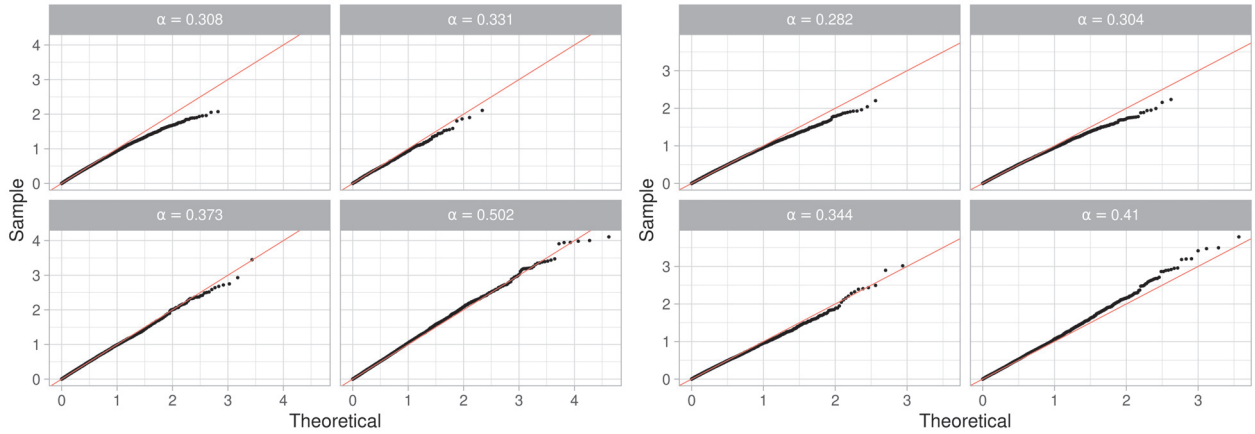

Fig. 3.6: QQ-plots comparing the distributions of the observations inside each subsets of the estimated partitions with the Exponential distributions.

Finally, we compare the estimated partitions with the outputs of the equivalent tree model. **Fig. 3.7** provides the fidelity curves (R^2 measure) between the approximated partition and the initial partition created by the tail gradient tree boosting. We decide to choose $K = 4$ subsets for $\mathcal{D}^{(1)}$ with a high fidelity (98.10%) and $K = 7$ subsets for $\mathcal{D}^{(2)}$ with a good fidelity (79.24%). **Fig. 3.8** shows the approximated small-size partitions for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$. The equivalent tree model provides the exact partition for $\mathcal{D}^{(1)}$ doing better than the hierarchical clustering algorithm. Although the approximated small-size partition differs from the true one for $\mathcal{D}^{(2)}$, the explanations that could be made would be very convincing. In **Fig. 3.9**, QQ-plots compare the distributions of the observations inside each

subsets of the approximated partitions with the Exponential distributions (whose means are equal to the empirical means). The alignment of the points also shows good fits.

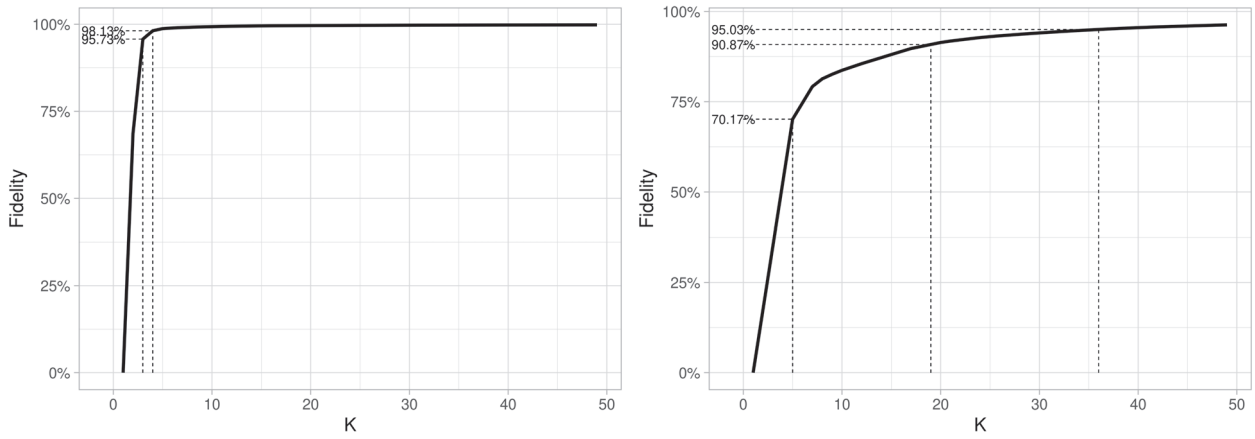

Fig. 3.7: Fidelity curves of the equivalent tree models for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$.

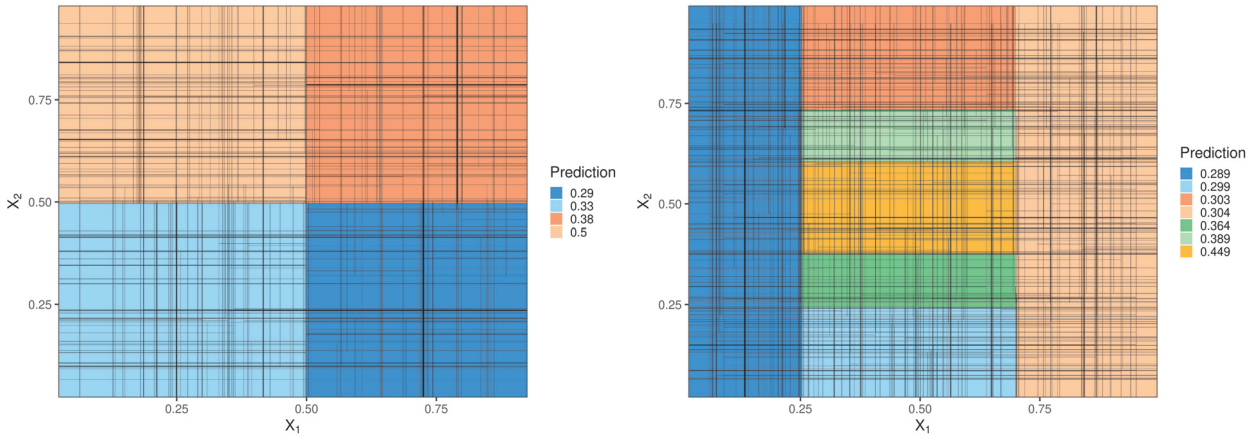

Fig. 3.8: Approximated small-size partitions of the equivalent tree models for $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$.

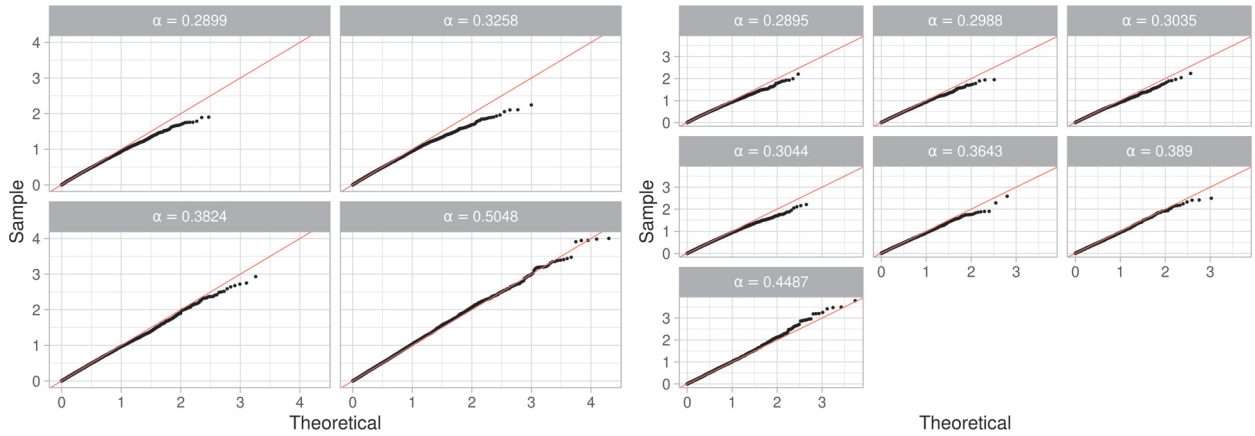

Fig. 3.9: QQ-plots comparing the distributions of the observations inside each subsets of the estimated partitions with the Exponential distributions.

3.3.2 A case study for the insurance industry

Each year, about 1,200 tornadoes with wind speeds as high as 300 mph touch down in the United States. They are more frequent than hurricanes and can also cause severe damage over small areas, as well as many deaths. In the decade 1965-1974, they were responsible for an average of 141 deaths each year, compared with 57 in the 10 years 1995-2004. The peak of the tornado season is April through June or July. Spring tornadoes tend to be more severe and strike the Southeast, which is more densely populated than the Great Plains, thus causing more deaths than those in the summer months.

Standard homeowners and business insurance policies cover wind damage caused by tornadoes and severe weather. Homeowner's insurance policies also provide coverage for additional living expenses policyholders will need to finance temporary housing costs and other daily necessities. Damage to vehicles is covered under the comprehensive section of standard auto insurance policies, which is optional.

We consider a NOAA (National Oceanic and Atmospheric Administration) tornado dataset containing $60,652$ events from 1950 to 2011. The variables that we kept from this dataset are given in **Tab. 3.2**. We combine the variables `PROPDMGEXP` and `PROPDMG` to a single variable `PROPERTY DAMAGE` containing the estimated costs (in dollars) of the damages caused by the tornadoes. We only retain strictly positive amounts of `PROPERTY DAMAGE` and end up with $39,036$ observations. We finally extract `YEAR` and `MONTH` from the variable `DATE`.

Tab. 3.2: Descriptive table of variables.

Variable	Description
`LENGTH`	Length of the tornado's path in miles.
`WIDTH`	Maximum width of tornado's path in yards.
`MAG`	Hail in inches.
`LONGITUDE`	Longitude where the tornado occurred.
`LATITUDE`	Latitude where the tornado occurred.
`STATE`	State where the tornado occurred.
`DATE`	Start date of the tornado.
`PROPDMG`	Property damage in dollars.
`PROPDMGEXP`	Magnitude of the damages. (H=hundreds, K=thousands, M=millions, B=billions)

We want to characterize the extremal behavior of property damages caused by tornadoes. The documentation provided with the dataset does not mention whether inflation has been taken into account to evaluate property damages, but after analyzing the amounts of damages, we concluded that it was not the case and decided to adjust these amounts for inflation. We also decided to

add contextual information about the population densities since the damages caused by tornadoes are correlated to the human constructions in the area they strike. We used a shapefile containing geographical frontiers of U.S. counties that we linked with the census data to extract population densities by year and county (DENSITY). We thereafter consider as our variable of interest the following variable: DAMAGE BY DENSITY=PROPERTY DAMAGE/DENSITY.

Fig. 3.10 illustrates the linear relationships between log (DAMAGE BY DENSITY) with respectively log (LENGTH) and log (WIDTH), while **Fig. 3.11** provides a scatter plot showing that the longer and the wider a tornado track, the greater the damage will be.

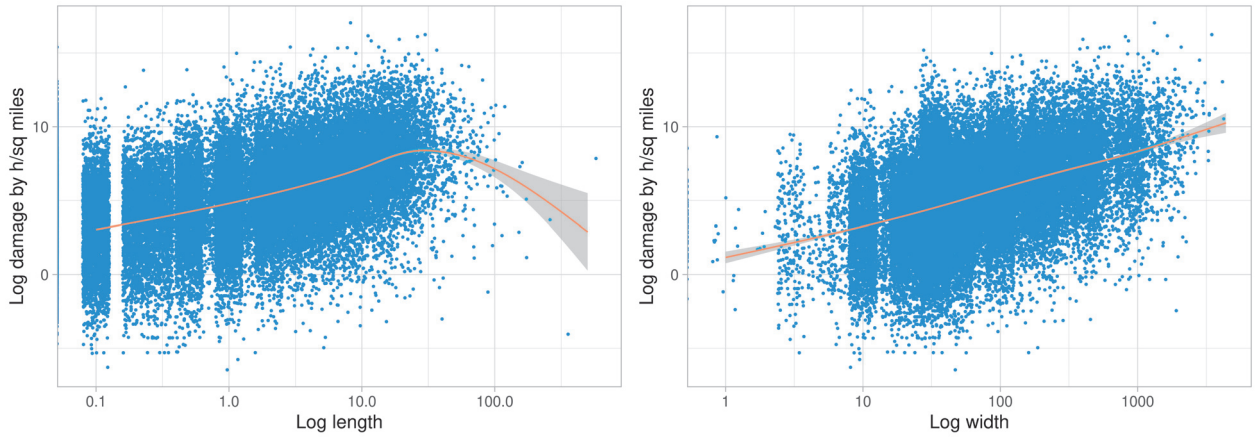

Fig. 3.10: Linear relationships between log (DAMAGE BY DENSITY) with respectively log (LENGTH) and log (WIDTH).

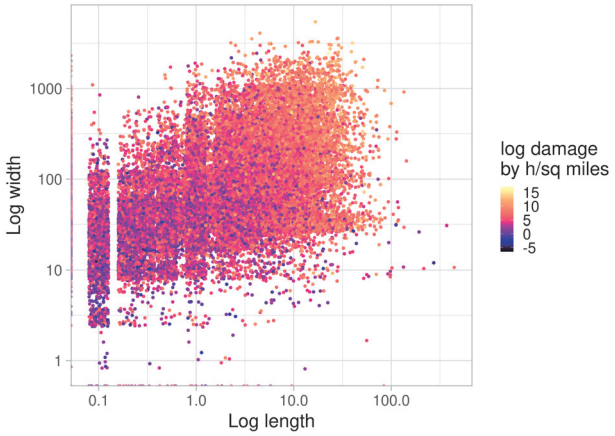

Fig. 3.11: Scatter plot of log (LENGTH) and log (WIDTH).

Fig. 3.12 displays a map of the United States with the population density per county as well as the locations of a random subsample of the tornadoes depicted with different colors depending on the amount of damages.

After a detailed study of the tails of the distributions of the variable DAMAGE BY DENSITY, we concluded that it was necessary to make a deterministic transformation of this variable so that the observations are compatible with the hypothesis of a Pareto-type distribution. The tails of the distri-

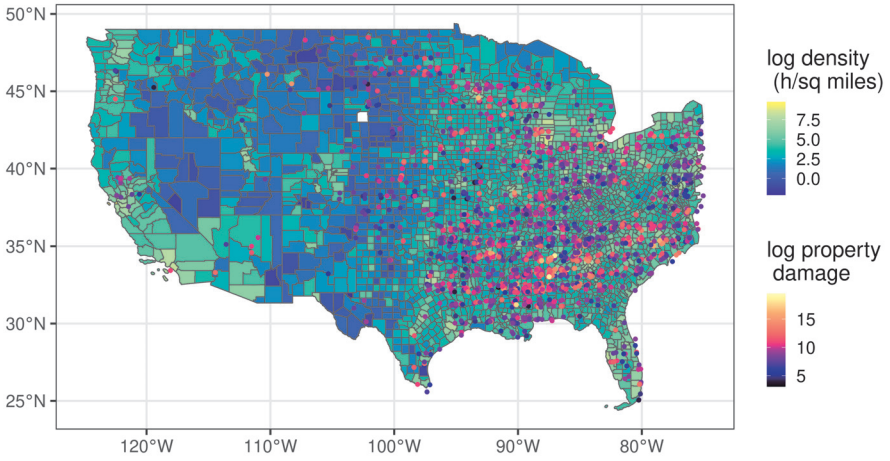

Fig. 3.12: Population density in the US and tornado locations.

butions are in fact of Weibull-type with a coefficient θ that can be estimated following the approach developed in Girard (2004). The values of the estimators of the Weibull tail-coefficient $\hat{\theta}_n$ based on the k_n upper order statistics are given in the left panel of Figure 13. In practice, the choice of the parameter k_n is the key problem to obtain a correct estimation of θ. If k_n is too small, the variance of $\hat{\theta}_n$ may be high and conversely, if k_n is too large, the bias may be important. We have chosen an approach like in Girard (2004) and have selected the values $k_n = 5000$ and $\hat{\theta}_n = 4$. Therefore we transform the variable `DAMAGE BY DENSITY` in the following way

$$Z = \exp\left((\texttt{DAMAGE BY DENSITY})^{1/4}\right).$$

Fig. 3.13 displays a QQ plot comparing the distribution of $Y = \log(Z)$ with an exponential distribution. The good alignment of the points let us expect that an Exponential distribution with a finite number of values for its parameter is an appropriate choice. We therefore choose to keep all the observations of Y in the dataset.

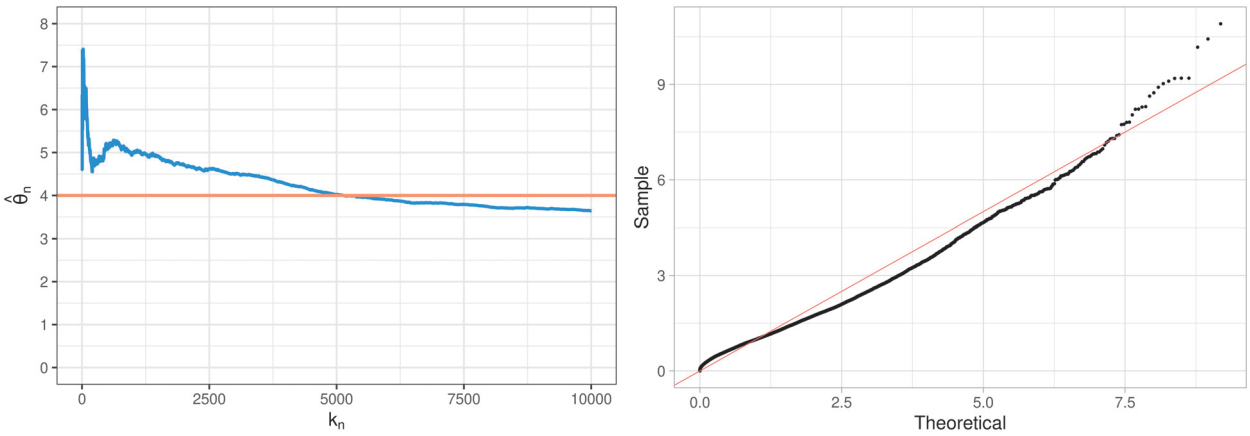

Fig. 3.13: Left panel: Weibull tail-coefficient estimator $\hat{\theta}_n$; Right panel: QQ-plot comparing the distributions of the observations with the Exponential distribution.

The covariates that have been retained for building an additive tree model are: `MAG`, `LENGTH`, `WIDTH`, `LATITUDE`, `LONGITUDE`, `YEAR`. We divided the observations into two subsets : a training set (80% of observations) and a validation set (20% of observations). We fitted a tail gradient tree boosting. The choice of the hyper-parameters are given in the following table.

Tab. 3.3: Selected model.

Model name	Description	Hyperparameters	5-folds CV
`GBM Gamma`	A Gradient Boosting Machine that uses the Gamma log-likelihood criterion.	`col_sample_rate` = 0.7 `learn_rate` = 0.1 `max_depth` = 4 `sample_rate` = 0.7 `distribution` = "gamma"	0.89

The grid search for the hyper-parameters was performed with a 5-fold cross-validation. Results are displayed in **Tab. 3.4** in appendix. We selected the model with the smallest Gamma deviance and we denote by `GBM Gamma` this model (the Gamma deviance on the train set is equal to 0.8827, on the validation set 0.8917 and by cross-validation 0.8899).

We then run `ClustGeo` to gather subsets of the tail gradient tree boosting partition. To help us choose the mixing parameter γ, we plot the proportions (resp. normalized proportions) of explained inertia of the partitions in K clusters obtained with the `ClustGeo` procedure for a range of γ. When the proportion (resp. normalized proportion) of explained inertia based on D_0 decreases, the proportion (resp. normalized proportion) of explained inertia based on D_1 increases. The plots are given in **Fig. 3.14**.

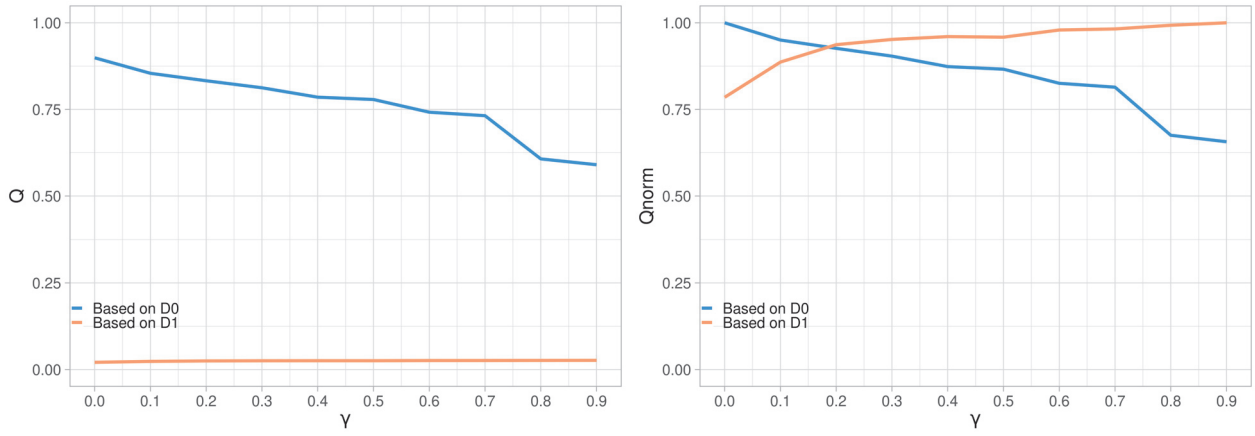

Fig. 3.14: Proportions (resp. normalized proportions) of explained pseudo-inertias versus γ in the left panel (in the right panel).

We also compute the fidelity curves with respect to γ (see **Fig 3.15**).

On the basis of these plots, we chose $\gamma = 0.2$ and $K = 12$ (R^2 measure is equal to 84.95%.). **Fig. 3.16** displays box plots of the predicted values by `GBM Gamma` for each subset of the estimated partition, as

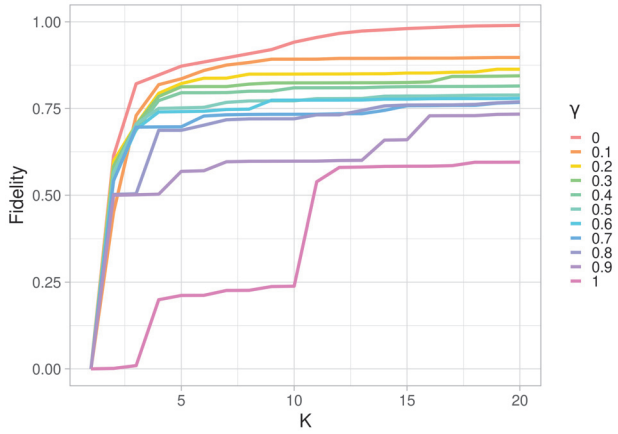

Fig. 3.15: Fidelity curves versus γ.

well as box plots of their relative differences with their means. For almost every subsets, more than 50% of the predictions have relative differences less than 15% in absolute value.

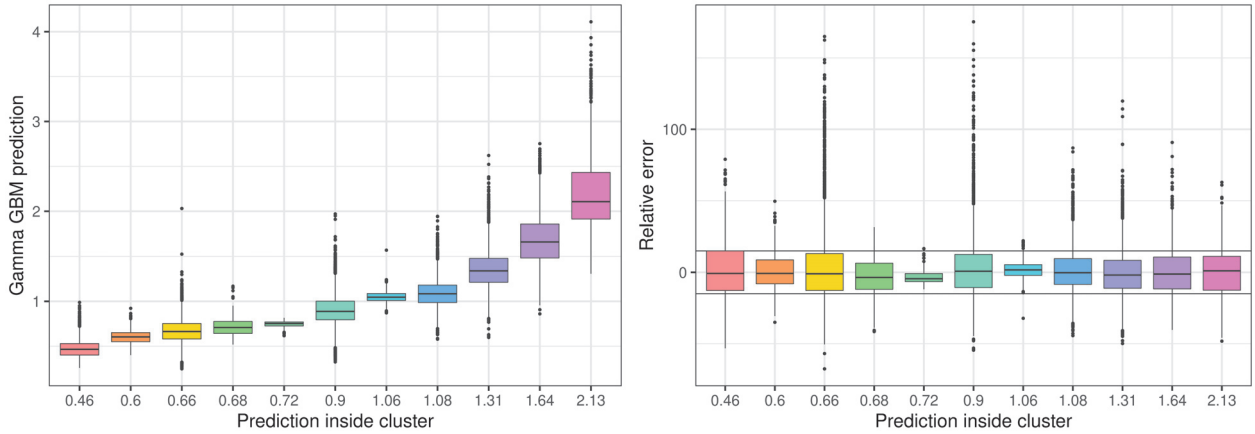

Fig. 3.16: Left panel: GBM Gamma's predictions by subset of the partition; Right panel: Relative errors.

We finally provide in the left panel of **Fig. 3.17** the QQ-plots comparing the distributions of the observations inside each subsets of the estimated partitions with the Exponential distributions and in the right panel the box plots of the predicted values for each subset of the estimated partitions and for the covariates: `LATITUDE`, `LENGTH`, `LONGITUDE`, `WIDTH`, `YEAR`. We note that the empirical distributions inside the subsets of the estimated partition are not well approximated by Exponential distributions. We observe that the covariates `LENGTH` and `WIDTH` are the covariates are the covariates that most influence the means of observations through the subsets of the estimated partition.

Instead of taking for D_1 the distance value which is set to 0.1 if two rectangles are adjacent and to a large value if this is not the case, we now take the Euclidean distance between the gravity centers of the rectangles. We obtain the following results.

On the basis of these plots, we chose $\gamma = 0.2$ and $K = 12$ (R^2 measure is equal to 87.50%.). **Fig. 3.20** displays box plots of the predicted values by `GBM Gamma` for each subset of the estimated partition, as

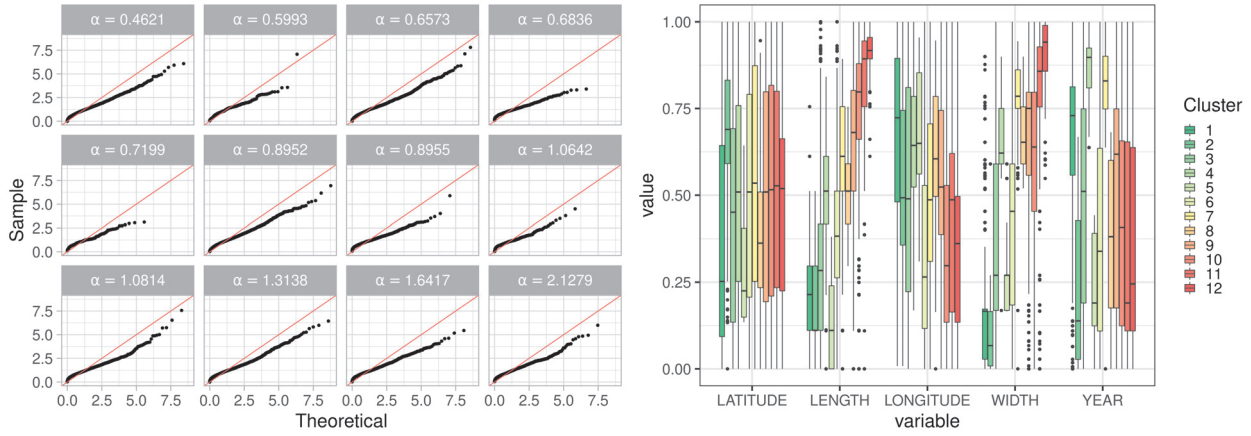

Fig. 3.17: Left panel: QQ-plots comparing the distributions of the observations inside each subset of the estimated partitions with the Exponential distributions; Right panel: Box plots of the predicted values for each subset of the estimated partitions and for the covariates: `LATITUDE`, `LENGTH`, `LONGITUDE`, `WIDTH`, `YEAR`.

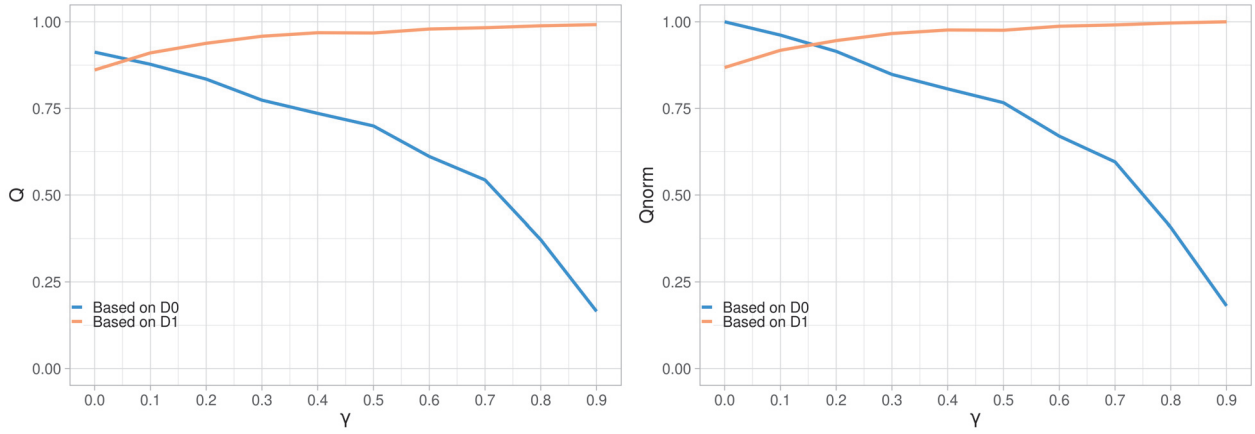

Fig. 3.18: Proportions (resp. normalized proportions) of explained pseudo-inertias versus γ in the left panel (in the right panel).

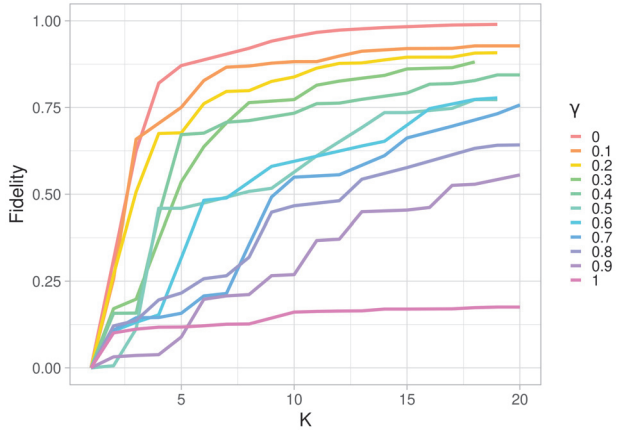

Fig. 3.19: Fidelity curves versus γ.

well as box plots of their relative differences with their means. For almost every subsets, more than 50% of the predictions have also relative differences less than 15% in absolute value.

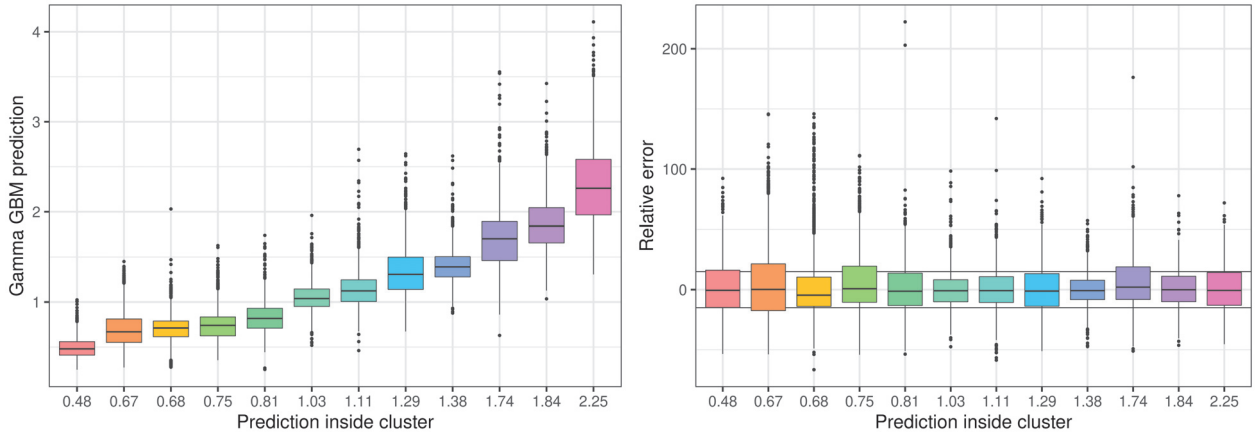

Fig. 3.20: Left panel: GBM Gamma's predictions by subset of the partition; Right panel: Relative errors.

We finally provide in the left panel of Figure 21 the QQ-plots comparing the distributions of the observations inside each subsets of the estimated partition with the Exponential distributions and in the right panel the box plots of the predicted values for each subset of the estimated partitions and for the covariates: `LATITUDE`, `LENGTH`, `LONGITUDE`, `WIDTH`, `YEAR`. The assumption of Exponential distributions for each subset of the partition is now more convincing. Moreover the box plots provide evidence of clearer links between the covariates considered and the empirical distributions of the observations through the subsets.

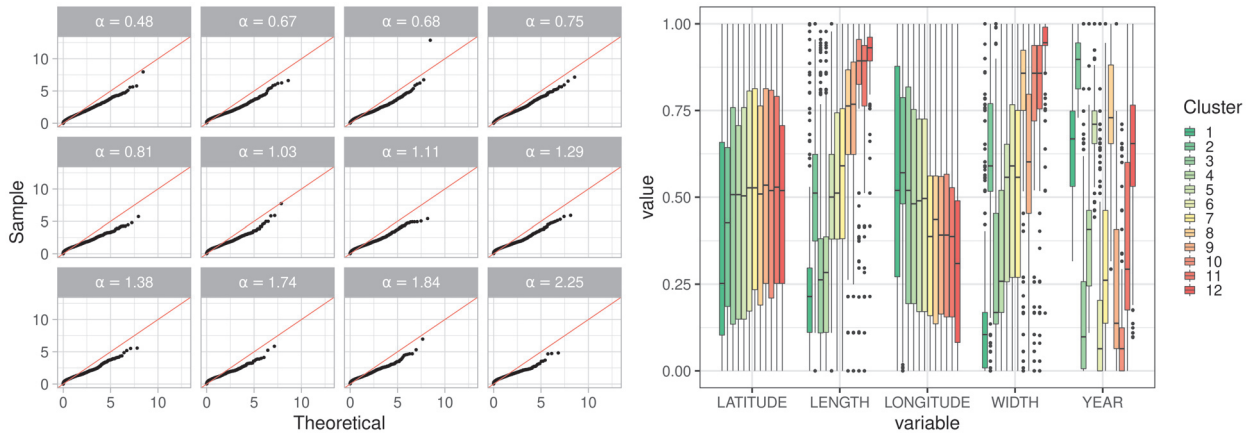

Fig. 3.21: Left panel: QQ-plots comparing the distributions of the observations inside each subset of the estimated partitions with the Exponential distributions; Right panel: Box plots of the predicted values for each subset of the estimated partitions and for the covariates: `LATITUDE`, `LENGTH`, `LONGITUDE`, `WIDTH`, `YEAR`.

We finally consider the equivalent tree method. **Fig. 3.22** displays the fidelity curve which led us to also choose $K = 12$ subsets with R^2 measure equal to 75.75%.

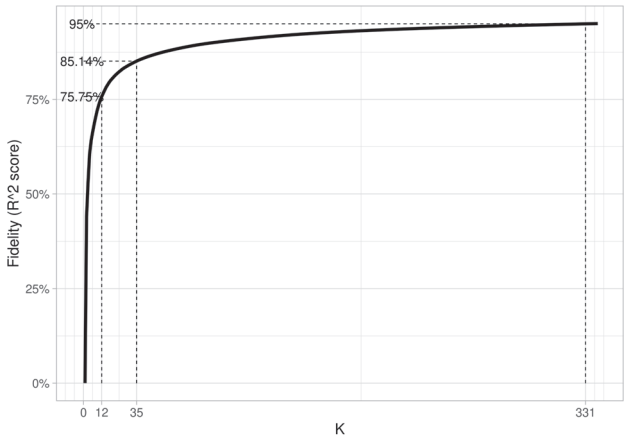

Fig. 3.22: Fidelity curve.

The tree that led to the approximated partition is depicted in **Fig. 3.23**.

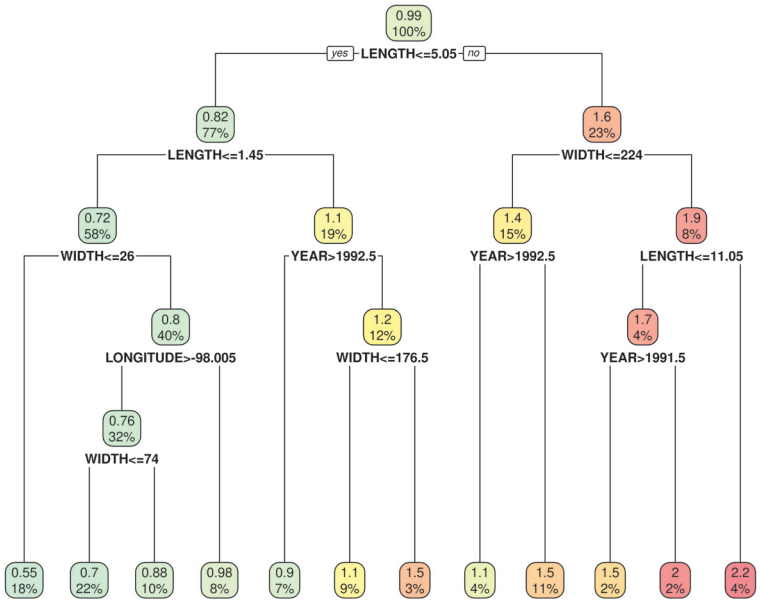

Fig. 3.23: Surrogate tree ($R^2 = 75\%$).

Fig. 3.24 provides box plots of the predicted values by `GBM Gamma` for each subset of the estimated partition, as well as box plots of their relative differences with their means.

Fig. 3.25 shows that the approximated partition of the equivalent tree gives distributions of observations inside each subset that are relatively close to Exponential distributions.

From these different analyses, we can draw the following conclusions:

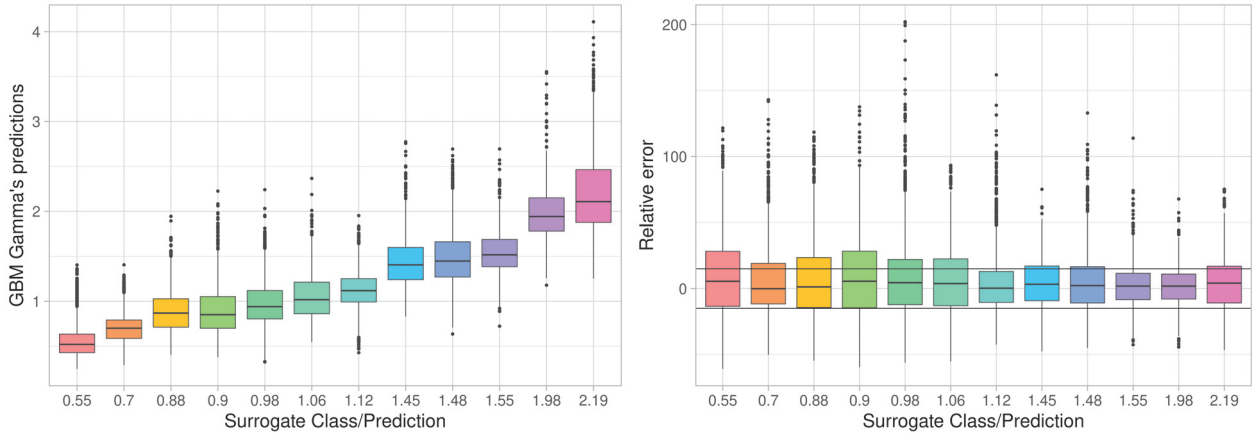

Fig. 3.24: Left panel: GBM Gamma's predictions by subset of the partition; Right panel: Relative errors.

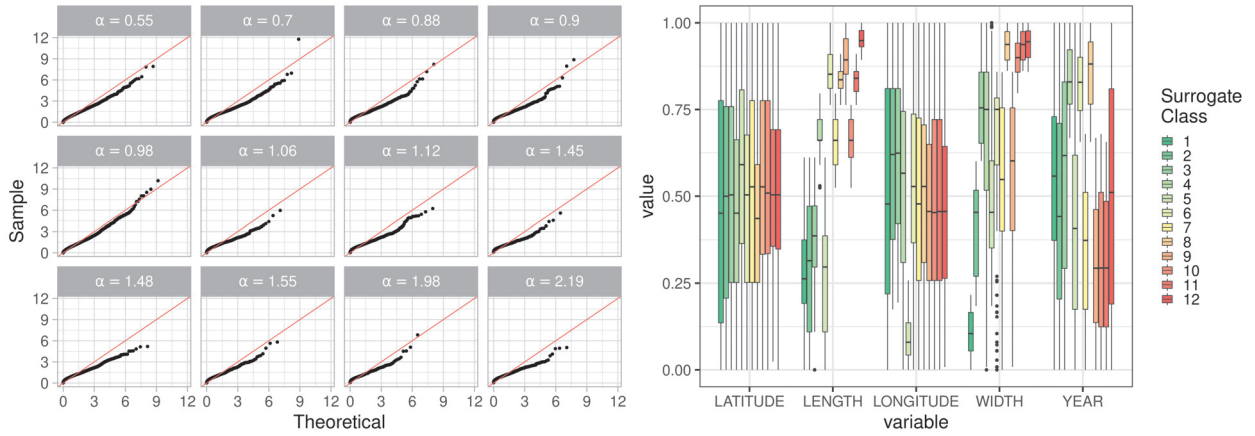

Fig. 3.25: Left panel: QQ-plots comparing the distributions of the observations inside each subset of the estimated partitions with the Exponential distributions; Right panel: Box plots of the predicted values for each subset of the estimated partitions and for the covariates: `LATITUDE`, `LENGTH`, `LONGITUDE`, `WIDTH`, `YEAR`.

- The small-size partitions obtained from the additive tree ensemble and the hierarchical clustering with spatial constraints have a better fidelity than the one obtained with the tree surrogate model for the same number of classes because they are built in a more flexible way.

- The empirical distributions of the observations within each subset of the partitions are closer to Exponential distributions for the tree surrogate model while providing natural explanations for the classes in terms of covariates and a simple division of the covariate space.

3.4 Conclusions

Estimating the tail index can become highly complex in the presence of covariates in order to gain a competitive advantage in risk assessment. However providing simple but accurate models is a key requirement for any high-stakes decision. In this paper we assume that the tail index function only takes a small number of values over a partition of the covariate space. We propose a tail-index partition-based rules extraction method that is able to construct estimates of the partition subsets and estimates of the tail index values.

The method combines two steps: first an additive tree ensemble based on the Gamma deviance is fitted (which includes random forest and gradient tree boosting), second a hierarchical clustering with spatial constraints (`ClustGeo`) is used to estimate the subsets of the partition. The number of subsets of the partition is selected first by determining a weight coefficient between the dissimilarity matrices that provides a high degree of spatial contiguity without deteriorating too much the quality of the solution based only on the predictions of the additive tree ensemble, second by ensuring a sufficiently high level of fidelity (R^2 measure). The quality of the choice of the partition is finally checked by comparing the fit of the distributions of the observations to Exponential distributions with QQ plots for each subset of the partition.

Our procedure provides a small number of subsets of the covariate space whose shape may be however highly complex because they were constructed with constraints to form homogeneous subsets in terms of predictions but also homogeneous in the covariate space. It may be difficult to find simple covariate-based explanations for these subsets. We have therefore proposed a global tree surrogate model to approximate the partition-based rules while providing an explainable model from the initial covariates. If explanations are to be provided, fidelity should be sacrificed in order to generate a more “rigid” model with cuts aligned with the covariate axes. Our numerical experiments as well as the case study show that the drop in quality is actually not that great.

Appendix

Tab. 3.4: Grid search for the gamma gradient boosting model.

col_sample_rate	distribution	learn_rate	max_depth	sample_rate	Gamma deviance
0.7	gamma	0.1	4	0.7	0.8898692
0.8	gamma	0.1	4	0.9	0.8899294
0.8	gamma	0.1	4	0.8	0.8899373
0.6	gamma	0.1	4	0.9	0.8905952
0.4	gamma	0.1	4	0.9	0.8918163
0.8	gamma	0.1	3	0.7	0.8939461
0.7	gamma	0.1	3	0.8	0.8939524
0.8	gamma	0.1	3	0.9	0.8941586
0.5	gamma	0.1	3	0.8	0.8943843
0.6	gamma	0.1	3	0.7	0.8947193
0.4	gamma	0.1	3	0.9	0.8951752
0.2	gamma	0.1	3	0.7	0.8980810
0.7	gamma	0.03	4	1.0	0.8986279
0.6	gamma	0.03	4	0.8	0.8991031
0.4	gamma	0.03	4	0.7	0.8999548
0.7	gamma	0.03	3	1.0	0.9032651
0.6	gamma	0.03	3	0.8	0.9034957
0.5	gamma	0.03	3	1.0	0.9037465
0.2	gamma	0.03	4	1.0	0.9065550
0.3	gamma	0.03	4	1.0	0.9065550
0.7	gamma	0.01	4	0.9	0.9228733
0.7	gamma	0.01	4	1.0	0.9229140
0.7	gamma	0.01	3	0.8	0.9294435
0.8	gamma	0.01	3	0.7	0.9294608
0.7	gamma	0.01	3	0.9	0.9294780
0.5	gamma	0.01	3	0.9	0.9302424
0.5	gamma	0.01	3	0.7	0.9302508
0.2	gamma	0.01	4	1.0	0.9378801
0.3	gamma	0.01	3	1.0	0.9440475
0.3	gamma	0.01	3	0.9	0.9441168

Bibliography

Breiman, Leo (2001). “Random forests”. In: *Machine learning* 45(1), pp. 5–32.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone (Oct. 1984). *Classification And Regression Trees*. Routledge. DOI: 10.1201/9781315139470. URL: https://doi.org/10.1201%2F9781315139470.

Chavent, Marie, Vanessa Kuentz-Simonet, Amaury Labenne, and Jérome Saracco (Jan. 2018). “ClustGeo: an R package for hierarchical clustering with spatial constraints”. In: *Computational Statistics*

33(4), pp. 1799–1822. DOI: 10.1007/s00180-018-0791-1. URL: https://doi.org/10.1007%2Fs00180-018-0791-1.

Daouia, Abdelaati, Laurent Gardes, Stéphane Girard, and Alexandre Lekina (June 2010). "Kernel estimators of extreme level curves". In: *TEST* 20(2), pp. 311–333. DOI: 10.1007/s11749-010-0196-0. URL: https://doi.org/10.1007%2Fs11749-010-0196-0.

Dekkers, A. L. M., J. H. J. Einmahl, and L. De Haan (Dec. 1989). "A Moment Estimator for the Index of an Extreme-Value Distribution". In: *The Annals of Statistics* 17(4). DOI: 10.1214/aos/1176347397. URL: https://doi.org/10.1214%2Faos%2F1176347397.

Farkas, Sébastien, Olivier Lopez, and Maud Thomas (May 2021). "Cyber claim analysis using Generalized Pareto regression trees with applications to insurance". In: *Insurance: Mathematics and Economics* 98, pp. 92–105. DOI: 10.1016/j.insmatheco.2021.02.009. URL: https://doi.org/10.1016%2Fj.insmatheco.2021.02.009.

Friedman, Jerome H. (2001). "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29(5), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

Gardes, Laurent and Stéphane Girard (Jan. 2012). "Functional kernel estimators of large conditional quantiles". In: *Electronic Journal of Statistics* 6(none). DOI: 10.1214/12-ejs727. URL: https://doi.org/10.1214%2F12-ejs727.

Gardes, Laurent and Gilles Stupfler (May 2013). "Estimation of the conditional tail index using a smoothed local Hill estimator". In: *Extremes* 17(1), pp. 45–75. DOI: 10.1007/s10687-013-0174-5. URL: https://doi.org/10.1007%2Fs10687-013-0174-5.

Girard, Stéphane (Jan. 2004). "A Hill Type Estimator of the Weibull Tail-Coefficient". In: *Communications in Statistics - Theory and Methods* 33(2), pp. 205–234. DOI: 10.1081/sta-120028371. URL: https://doi.org/10.1081%2Fsta-120028371.

Goegebeur, Yuri, Armelle Guillou, and Antoine Schorgen (May 2013). "Nonparametric regression estimation of conditional tails: the random covariate case". In: *Statistics* 48(4), pp. 732–755. DOI: 10.1080/02331888.2013.800064. URL: https://doi.org/10.1080%2F02331888.2013.800064.

Goegebeur, Yuri, Armelle Guillou, and Gilles Stupfler (Aug. 2015). "Uniform asymptotic properties of a nonparametric regression estimator of conditional tails". In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques* 51(3). DOI: 10.1214/14-aihp624. URL: https://doi.org/10.1214%2F14-aihp624.

Gordon, A.D. (Jan. 1996). "A survey of constrained classification". In: *Computational Statistics & Data Analysis* 21(1), pp. 17–29. DOI: 10.1016/0167-9473(95)00005-4. URL: https://doi.org/10.1016%2F0167-9473%2895%2900005-4.

Hill, Bruce M. (Sept. 1975). "A Simple General Approach to Inference About the Tail of a Distribution". In: *The Annals of Statistics* 3(5). DOI: 10.1214/aos/1176343247. URL: https://doi.org/10.1214%2Faos%2F1176343247.

Legendre, P (2011). "const. clust: space-and time-constrained clustering package. R package version 1.2". In: *URL: http://adn. biol. umontreal. ca/~ numericalecology/Rcode*.

Maillart, Arthur and Christian Robert (2021). "Hill random forests". In: *Working paper*.

Murtagh, F. (Jan. 1985). “A Survey of Algorithms for Contiguity-constrained Clustering and Related Problems”. In: *The Computer Journal* 28(1), pp. 82–88. DOI: 10.1093/comjnl/28.1.82. URL: https://doi.org/10.1093%2Fcomjnl%2F28.1.82.

Pickands, James (Jan. 1975). “Statistical Inference Using Extreme Order Statistics”. In: *The Annals of Statistics* 3(1). DOI: 10.1214/aos/1176343003. URL: https://doi.org/10.1214%2Faos%2F1176343003.

Stupfler, Gilles (Jan. 2013). “A moment estimator for the conditional extreme-value index”. In: *Electronic Journal of Statistics* 7(none). DOI: 10.1214/13-ejs846. URL: https://doi.org/10.1214%2F13-ejs846.

The CGAL Project (2021). *CGAL User and Reference Manual*. 5.2.1. CGAL Editorial Board. URL: https://doc.cgal.org/5.2.1/Manual/packages.html.

Zomorodian, Afra and Herbert Edelsbrunner (2000). “Fast software for box intersections”. In: *Proceedings of the sixteenth annual symposium on Computational geometry - SCG ’00*. ACM Press. DOI: 10.1145/336154.336192. URL: https://doi.org/10.1145%2F336154.336192.

4 Black-Box Inspection via Robustness Analysis

> *To deal with a 14-dimensional space, visualize a 3-D space and say 'fourteen' to yourself very loudly. Everyone does it.*
>
> — **Geoffrey Hinton**

Abstract

The rise of machine learning models has led insurers to create DataLabs to build models that are more efficient than existing models. However, for the most critical tasks, some models that are considered too complex and secretive struggle to go into production. Indeed, for actuaries responsible for 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 learned. 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 aspects of intelligibility.

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

4.1 Introduction

Among machine learning methods, neural networks are the most complex. For a prediction at a given point, it is often impossible to understand why the model makes one decision rather than another. This legitimates the development of methods to make sophisticated machine learning models more intelligible. However, what is meant by intelligible? According to Guidotti et al. (2019) and Doshi-Velez and Kim (2017), for machine learning models to make intelligible signifies to explain in

human-understandable terms. However, explaining has a different sense depending on how we use these methods. If we want to improve or debug our model, information on the importance that the black box gives to variables can be satisfactory. Nevertheless, if we need a deeper comprehension of the model to comply with the General Data Protection Regulation (GDPR), simple information about the variables may not be sufficient. Indeed, we want to be able to, for example, explain the price in detail to each policyholder. Obviously, the more precise we want the explanation to be, the more complicated the problem becomes. For an overview of explainability methods in machine learning, please refer to Guidotti et al. (2019).

In this article, we are interested in a method that lets us highlight influential points on a parametric machine learning model. We call a parametric method any method that learns a model written as a set of parameters. These can be neural networks, generalized linear models or support vector machines. In contrast, nonparametric methods do not make assumptions about the structure of the model. These include decision trees and tree ensembles such as random forest and gradient boosting. To estimate the influence of points in the learning sample, a natural approach is to train a reference model on the entire learning sample and then measure the deviation on quantities of interest. For example, the difference in estimated parameters before and after removing a point, or the variation in loss function values before and after removing a point. Using these indicators, it is possible to know whether a point contributes to improving or deteriorating the performance of a black-box model. However, this method can be very expensive in calculations for large datasets. In this paper, we present three indicators based on influence functions. The first allows us to approximate the effect of removing a point on the estimated parameters. The two others approximate the impact on any prediction of removing or altering a point from the training sample. If we refer to the classification of Guidotti et al. (2019), the method we show here is part of the "black-box inspection problem". In other words, we are looking for visual or textual elements that enable us to learn about what the model has learned. Known examples of methods belonging to this class are the partial dependence plots presented in Friedman (2001) and the variable importance plots introduced by Breiman (2001). This approach is not new since statisticians such as Cook and Weisberg (1982) were very quickly interested in generalized linear models. Since then, it has been deepened, among others, in Wojnowicz et al. (2016) and Koh and Liang (2017), which generalize the results and give a harmonized mathematical framework for parametric models.

Determining the influence of observations on a model does not provide a global explanation of a black-box model. Nevertheless, it is possible to collect relevant information to debug the model or to obtain local information. By identifying the influential observations of the learning sample, we can find the points considered abnormal by the model. Thus, it is possible to clean the data to address these anomalous individuals. To achieve this, we present methods to compute indicators that highlight these observations. Moreover, by combining the appropriate indicators with naturally interpretable models such as generalized linear models, we can extract explanations in the form of hyperplanes (linear models). The idea is similar to the Local Interpretable Model-Agnostic Explanations (LIME) Ribeiro et al. (2016) and SHapley Additive exPlanations (SHAP) Lundberg and Lee (2017) methods that consist of finding a local surrogate model near the point to be explained. In

the complex case of neural networks, this information is not easy to obtain and is often expensive from a computational point of view. The methods presented here make it possible to reduce calculation times by approximating the influence of the points on the model. The main interest is to locate the areas where it is relevant to have a surrogate model. The second strength of this strategy is to restrict the number of points to be explained.

We systematically illustrate the indicators on a toy dataset in two dimensions before applying them to a real insurance dataset. Additionally, we use two types of parametric models to highlight our findings, a logistic regression model and a neural network. The logistic regression model allows us to understand how indicators work with a model within the actuary's comfort zone, while the second demonstrates the specific interest of the method. In the first part, we present the indicator $\mathcal{I}_{up,params}$ which determines the variation in the model's parameters when we delete a point from the learning sample. This can be useful for generalized linear models, especially to detect outliers. Then, we derive $\mathcal{I}_{up,loss}$, which characterizes the loss shift on a given prediction when we remove a point from the training sample. From the latter, we develop explanations that are locally faithful to the black-box model. These explanations take the form of hyperplanes. Moreover, we show that this technique allows us to reduce the number of points to be analyzed. Additionally, we define $\mathcal{I}_{pert,loss}$, which gives the direction in which a point must be perturbed to maximize the impact on the black-box model. Finally, we establish that this indicator can serve as an explanation for a given prediction.

Remark 20 *The proofs are in the appendix to make the reading more fluid.*

Foreword on influence functions

In Law et al. (1986), influence functions are intuitively the approximated and standardized effect on a statistic T when adding an observation z given a large sample with a distribution function F. The influence function of T of an underlying distribution function F in direction z is denoted $IF(z, T, F)$, which corresponds to a directional derivative in the F distribution in the direction z.

The notion of influence function is based on the concept of contaminated distribution. It is defined as follows.

Definition 3 *Let F be a distribution function. We define the ϵ-contaminated distribution at point z by:*

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

where Δ_z is the Dirac measure at point z.

Remark 21 *When $\epsilon > 0$, it can be interpreted as a mixture.*

Definition 4 *Let T be a functional, $T : \begin{cases} \mathcal{F} & \longrightarrow \mathbb{R} \\ F & \longmapsto T(F) \end{cases}$, where $\mathcal{F}$ is the set of distribution functions. We define the influence function by:*

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

Throughout the rest of the article, we study a binary classification problem. Let $\mathcal{X} = \mathbb{R}^p$ denotes the space of the explanatory variables and $\mathcal{Y}$ denotes the space of the labels. In addition, we have a dataset $D = \{z_i = (x_i, y_i)$ such that $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}\}$ containing n observations $z_1, \ldots, z_n$. The empirical distribution function of the random vector $(X_1, \ldots, X_p)$ is defined by

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

Intuitively, if we take the empirical distribution function rather than F ($F = F_{n-1}$) and $\epsilon = 1/n$, we show, for a sufficiently large sample, that the influence function measures n times the variation in T when adding a point z. In our case, F is a multidimensional distribution function defined as $F : \mathbb{R}^p \to \mathbb{R}$ and we consider instead the removal of a point from the training sample. This is equivalent to taking $F = F_n$ and $\epsilon = -1/n$.

Simple example on expectancy

Let Z be the observed variable. We consider a perturbation at point $z \in \mathbb{R}$. Thus, denoting $\mathbb{E}_{F_{\epsilon,z}}(Z)$ as the expected value of Z under the contaminated distribution function, we obtain:

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

hence,

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

From this expression, we can easily determine the expression of the influence function.

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

Remark 22 *The influence function of the average depends on the distance from the point z to the average. Thus, the more distant a point is in absolute value, the greater its influence on the average. Or if the contamination occurs at point z, and if this point has a high value, then the contamination has a significant impact on the mean. In contrast, if z is close to the average, then contamination has a relatively small impact. This is why, for example, the mean is considered not robust compared to the median.*

4.2 Theoretical problem setting and definitions

Machine learning methods such as random forest Breiman (2001) and XGBoost Chen and Guestrin (2016), and deep learning methods such as neural networks have proven to perform well in some prediction tasks. As a result, the paradigm of statistical learning has evolved. Now we seek to satisfy two contradictory concepts: predicting and understanding. Consequently, numerous works have been undertaken in recent years to overcome the lack of intelligibility. From an operational point of view, intelligibility can be divided into four approaches according to Guidotti et al. (2019). These are summarized in **Fig. 4.1**. Depending on the use of machine learning methods and the target audience, some interpretation methods are more relevant than others. As an actuary, it is often preferable to have the maximum level of information. This means to have a model intelligible by nature or a surrogate model that globally and faithfully reproduces the predictions of the black-box model. However, this is not always easy. For neural networks, it is difficult for a human to understand the global decision chain, since it involves too many weights. It may, therefore, be interesting to weaken the problem and either look for information only valid locally or to extract relevant information to obtain insight from the model. In the first case, we deal with the "black-box outcome explanation problem", while in the second we answer the "black-box inspection problem".

4.2.1 Contextual elements

Here, we define the classification problem. Let $b : \mathcal{X} \longrightarrow \mathcal{Y}$ be a black-box predictor, also called a classifier. The latter is provided by a learning function

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

where $\mathcal{Z}$ represents a set of datasets. Thus, for each $x \in \mathcal{X}$ the predictor b can provide a probability (or vector of probabilities) of belonging to a class denoted $b(x)$. In the supervised learning framework,

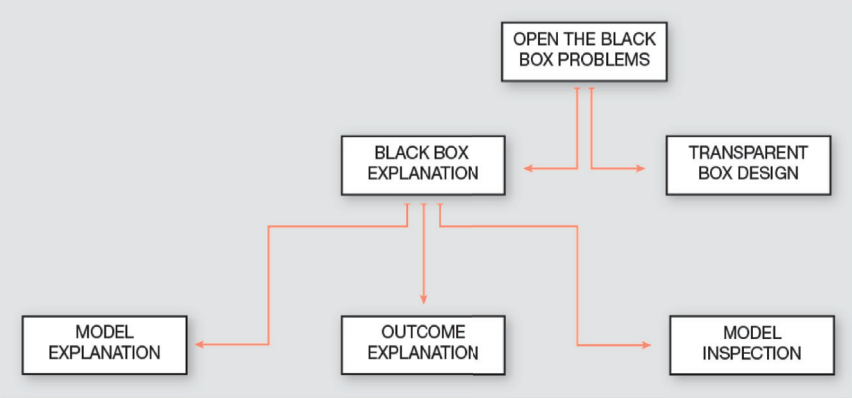

Fig. 4.1: Different machine learning model intelligibility approaches, Source: Guidotti et al. (2019)

D is divided into two samples: D_{train} and D_{test}. The first is used to build the classifier, and the second is used to evaluate its performance.

In this paper, we present indicators that first answer the "black-box inspection problem". By definition, for a black box b and a set of instances X, this consists of providing a visual or textual representation denoted by r. This representation is obtained from b and X using a procedure $f : (b, X) \longrightarrow r$. In our case, from our black box and the instances, we can extract representations r in the form of boxplots. These graphs characterize the influence of the points.

It is also possible to use the indicators and properties of the studied parametric models to answer the "black-box outcome explanation problem". This problem consists of finding an explanation e that belongs to $\mathcal{E}$, the set of explanations understandable by humans. This explanation is extracted from a locally intelligible model c_l, which is deduced from b and x using a procedure $f : (b, x) \longrightarrow c_l$. The explanation e is extracted from c_l and x. In our case, c_l is a linear model, and e is the decision chain (linear combination of weights) for x. The advantage over an inspection method is that we can measure the quality of the explanations provided. This measure is called fidelity in interpretable machine learning. Fidelity quantifies how well the surrogate model replicates the predictions of the black-box model in the vicinity of the point to be explained using usual metrics.

4.2.2 Definitions

We limit our study to tabular data because this is the most common form of data for insurers today. However, it is also possible to use this method with images or text. The interested reader may refer to Koh and Liang (2017).

The methods we study apply only to parametric models. We model the conditional law of Y given X. For a given parametric model, we denote the set of parameters by Θ. Let $z \in D_{train}$ and $\theta \in \theta$; the loss function at point z for the parameters of model θ is denoted by $\mathcal{L}(z, \theta)$. Finally, let us denote $\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(z_i, \theta)$ as the empirical risk and define the empirical risk minimizer as:

$$\hat{\theta}_n = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(z_i, \theta). \tag{4.1}$$

We assume in this study that $\mathcal{L}$ and $\nabla_\theta \mathcal{L}$ (where ∇_θ is the gradient operator with respect to θ) are three times continuously differentiable in θ and two times continuously differentiable in x. We denote by θ_0 the vector of true parameters, the one to which $\hat{\theta}_n$ converges. We further assume that $\mathbb{E}(\nabla_\theta(Z, \theta_0)) = 0$, $\mathbb{E}(||\nabla^2_\theta(Z, \theta_0)||) < \infty$ and that the matrix $\mathbb{E}(\nabla^2_\theta(Z, \theta_0))$ exists and is not singular. Furthermore, we assume that $\mathbb{E}(\nabla^3_\theta(Z, \theta_0))$ is bounded in probabilities when n tends toward infinity. When the loss functions do not verify these conditions, it is possible to adapt the method so that it remains valid as shown by Koh and Liang (2017).

4.3 Methodology of analysis

In this article, we implement the indicators to empirically verify their quality, then we study their behavior on different datasets. First, we manipulate these indicators and show their limitations on a two-dimensional toy dataset. We use the indicators on logistic regression (actuary comfort zone) and on neural network. In the second step, we apply the same indicators to the same models, but this time on an insurance dataset.

4.3.1 Simulated data

To test the indicators, we use a two-dimensional simulated dataset, shown in **Fig. 4.2**. We detail the process below.

After drawing a sample of $2,000$ points uniformly over the $[0, 1] \times [0, 1]$ square,

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

we define a theoretical threshold r as a function of x_1. The latter represents our true decision function.

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

We add noise near the decision boundary.

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

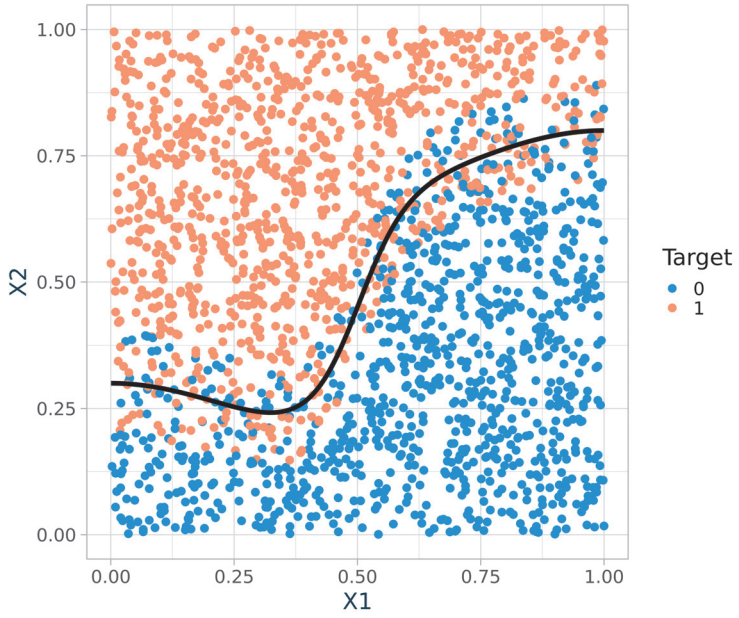

Fig. 4.2: Toy dataset: the solid black line represents the theoretical decision boundary.

4.3.2 The insurance problem

This toy example lets us manipulate the indicators and develop an understanding of them. To illustrate the interest of the indicators in a more realistic framework, we use data from insurers. These data have already been exploited for a DataScience challenge: The Pricing Game[1] (third edition). The data[2] [3] are initially composed of 27 variables that can be used to fit a model. We use only 8 numerical variables to illustrate the indicators. This is because $\mathcal{I}_{pert,loss}$ requires numerical and continuous variables. The sample contains 100,000 observations. We center and scale the data before splitting them into a validation sample and a learning sample. These samples are split into the following proportions (75%/25%). A total of 9,490 claims are reported in the learning sample

[1] http://freakonometrics.free.fr/PG3/3rdPricingGame.pdf
[2] http://freakonometrics.free.fr/PG3/PG_2017_YEAR0.csv containing the explanatory variables
[3] http://freakonometrics.free.fr/PG3/PG_2017 CLAIMS_YEAR0.csv containing the claims

and 3,164 in the validation sample. The claims are evenly distributed between the two samples. We transform the initial pricing problem (severity/frequency model) into a classification problem to be consistent with the theory developed in the article. To do so, we choose to create a new target variable from the number of claims. If the insurance policy has at least one claim then the new target variable contains one, and zero otherwise. In this way, we create a prevention problem. The idea is to create a model that recommends preventive action toward a driver.

4.3.3 The models

First, we implement the indicators for binary logistic regression. Since, in this case, we have simple closed formulas for the gradient and the Hessian, we can empirically check the accuracy of the approximation. Then, we extend to a penalized neural network.

Logistic Regression

For binary logistic regression, we attempt to model the probability for an observation of belonging to class 1 by

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

where $\beta = (\beta_0, \ldots, \beta_p)$. The coefficients β_i of the model are fitted by maximum likelihood, i.e., by optimizing the loss function

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

over all observations z_i. Usually, a Newton-Raphson algorithm is used to optimize the parameters.

Remark 23 *For the examples developed in this article, we have not penalized logistic regression. However, it is possible to introduce an L_2 penalty term. The loss function at the point z_i would then become:*

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

We have penalized the neural network described below with $\alpha = 0.001$.

Neural Network

We use a sufficiently simple and flexible neural network to correctly fit our toy and insurance datasets. The objective of the article is not the performance of the models. We want explanations for their outcomes. Hence, we do not seek to improve the performance of our black boxes. For hyperparameters $q_{m-1} \in \mathbb{N}$ and $q_m \in \mathbb{N}$, we define the layer m of our neural network as follows:

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

where neuron j of layer m is written as the following dot product:

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

where $\boldsymbol{\beta}_j^{(m)} = (\beta_{j,l}^{(m)})_{0<l<q_{m-1}} \in \mathbb{R}^{q_{m-1}+1}$ are the weights of the layer m of the neural network. Let us denote by M the total number of layers (including input and output layers). Then, each of the $m \in [[1, M]]$ layers is written as:

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

We use the softmax function denoted below by σ:

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

so that the output layer M returns a probability vector. Finally, the probability for an individual belonging to class 1 is expressed as follows:

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

We arbitrarily chose 2 hidden layers for our two neural networks[4]. In the case of the toy example, we chose the following hyperparameters: $q_1 = 2$, $q_2 = 4$, $q_3 = 2$, $q_4 = 2$ and $\phi = \tanh$. **Fig. 4.3** represents the architecture for this neural network.

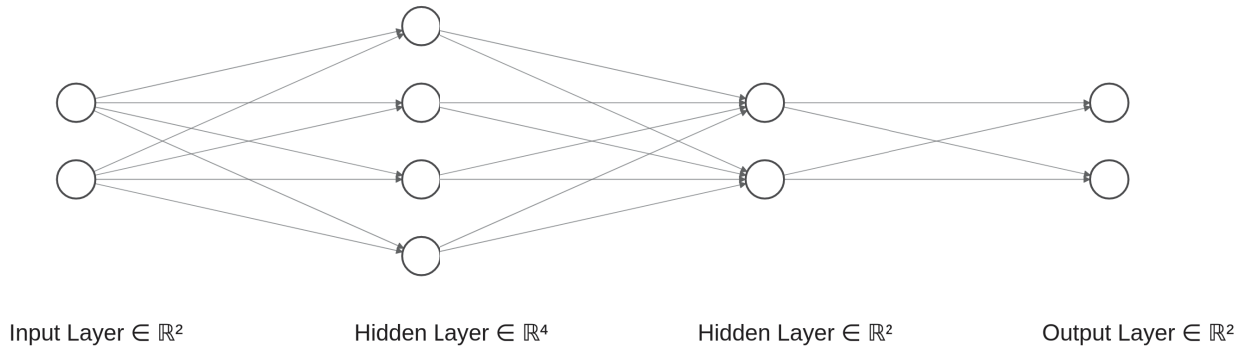

Fig. 4.3: Neural network architecture for the toy example

For the insurance problem, we chose the following hyperparameters: $q_1 = 8$, $q_2 = 4$, $q_3 = 2$, $q_4 = 2$ et $\phi = \tanh$. In this case, we use regularization L_2 previously defined. **Fig. 4.4** represents the architecture for this neural network.

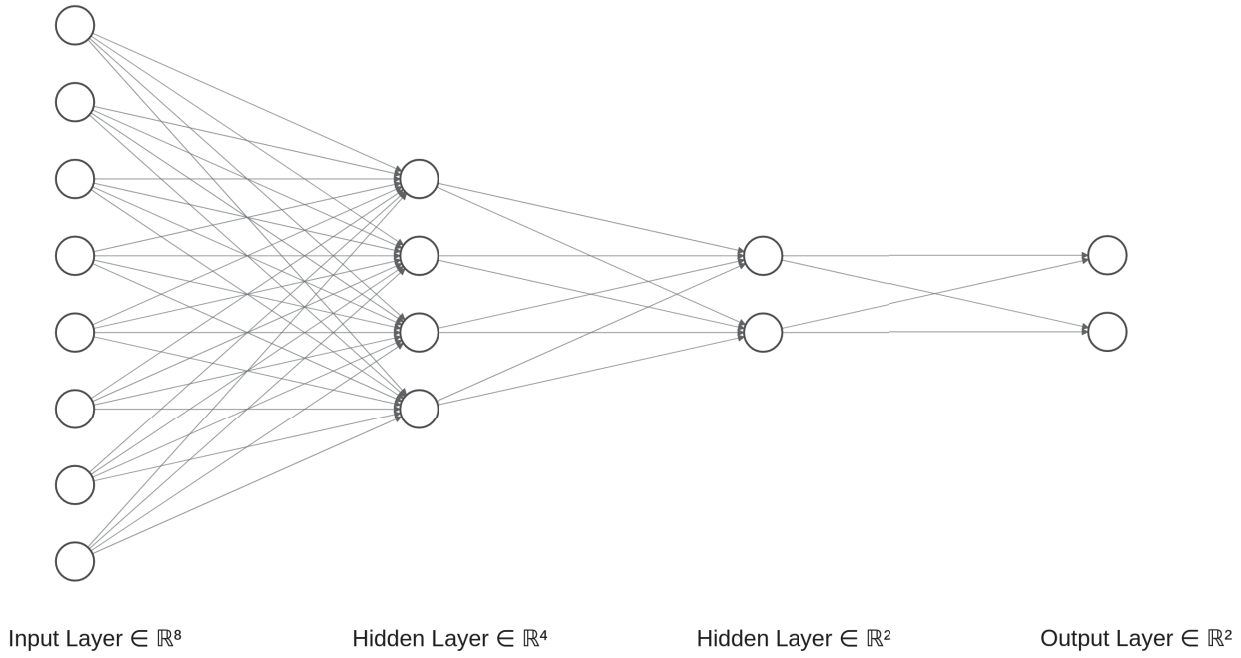

Fig. 4.4: Neural network architecture for the insurance problem

[4]The one applied to the toy example and the one applied to the real example

4.4 Influence of observations on parameters

The question we focus on in this section is how would our parameters change if we slightly modify the learning sample? The first form of perturbation we study here is the removal of a point in D_{train}. Informally, it is not conceivable to retrain a model for each dataset we can create by removing only one observation at a time. The cost of such a procedure would far exceed the benefits in terms of intelligibility. However, due to influence functions, it is possible to efficiently obtain an approximation of this variation on the black-box parameters.

We want to approximate $\hat{\theta}_n - \hat{\theta}_{n,-z}$, where $\hat{\theta}_{n,-z} = \arg\min_{\theta\in\Theta} \frac{1}{n-1} \sum_{z_i \neq z} \mathcal{L}(z_i, \theta)$. Furthermore, we denote the vector of optimal parameters by $\hat{\theta}_{n,\epsilon,-z}$ for the contaminated distribution ($F_{\epsilon,z} = (1-\epsilon)F + \epsilon\delta_z$). We restrict our study to the case of the M-estimators Huber (1981). In practice, they are obtained by minimizing a function depending on the data and the model parameters. The M-estimators can be seen as a generalization of maximum likelihood estimation.

In the case of an M-estimator, we look for the parameters that verify $\arg\min_{\theta\in\Theta} \left(\sum_{i=1}^{n} \rho(z_i, \theta)\right)$, where ρ is the function to be minimized over the observations. This can be the log-loss (or log-likelihood) in the case of binary logistic regression, for example. By setting $\rho = 1/n \times \mathcal{L}$ we are led back to our initial problem of empirical risk minimization described in equation (4.1).

Definition 5 *We define $\mathcal{I}_{up,params}(z)$ as the variation of the parameters vector θ related to an infinitesimal perturbation ϵ at point z by*

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

$\mathcal{I}_{up,params}$ is hardly exploitable. It is better to find a closed formula that is easier to implement. This is the purpose of the Proposition 1.

Proposition 1

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

where $\nabla_\theta \mathcal{L}(z, \hat{\theta}_n) \in \mathbb{R}^d$ is the gradient of the loss function $\mathcal{L}$ with respect to the parameters θ and evaluated at point z for the optimal parameters $\hat{\theta}_n$ and $H_{\hat{\theta}_n} = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 \mathcal{L}(z_i, \hat{\theta}_n)$ with $\nabla_\theta^2 \mathcal{L} \in \mathcal{M}_d(\mathbb{R})$ the matrix containing all the partial derivatives of order two with respect to θ.

The proposition 1 lets us quantify the impact on the parameters of a diminution of the mass ($\epsilon = -1/n$). It now remains to show that this mass fluctuation is equivalent to the removal of the point z in the learning sample. This is the purpose of the Proposition 2. We first define the notion of o_p.

Definition 6 *A sequence of random vectors $\{Z_n,\ n \in \mathbb{N}\}$ defined on the probabilized space $(\Omega, \mathcal{A}, \mathbb{P})$ is infinitely small in probability if Z_n converges in probability to 0. This is denoted by $Z_n = o_p(1)$. Immediately, we have the equivalence*

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

More details can be found in Gourieroux and Monfort (1995).

Proposition 2 *Let z be a point of the learning sample, $\hat{\theta}_n$ and $\hat{\theta}_{n,-z}$ the coefficients estimated with and without z in the learning sample.*

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

Due to these two propositions, it is now possible to estimate the influence of the points without having to remove the points one by one from the learning sample. It is sufficient to calculate the inverse of the Hessian associated with the loss and a gradient.

Geometrical interpretation

It is possible to see $\mathcal{I}_{up,params}$ as a single step of the *Newton-Raphson* algorithm applied to the loss function to minimize it. In other words, it is the direction that minimizes the loss for our model and for a given dataset. This corresponds graphically to the slope variation in our decision boundary.

4.4.1 Implementation of $\mathcal{I}_{up,params}$

Graphs and calculations are made with **R** Team et al. (2013) and the packages **tensorflow**[5], **keras** and **h2o**.

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

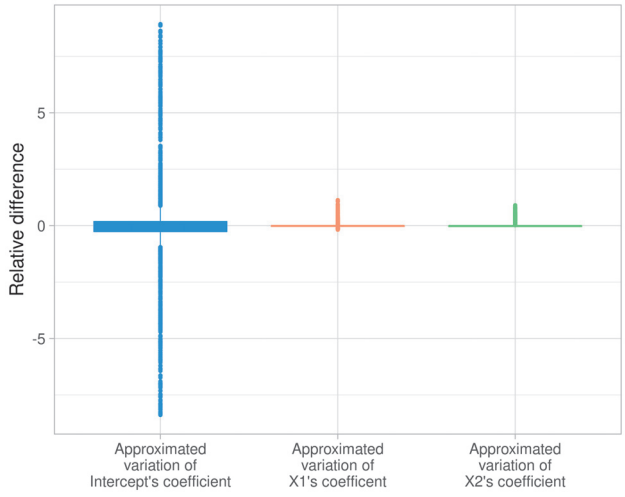

Fig. 4.5: Relative coefficient variations

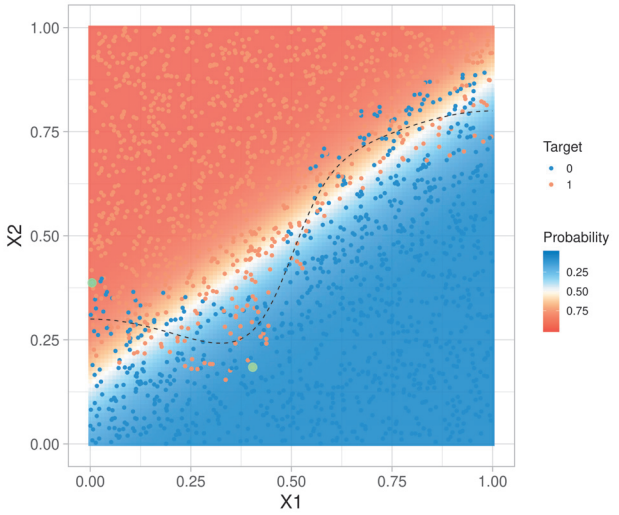

Fig. 4.6: Decision boundary

We are now interested in the information that the $\mathcal{I}_{up,params}$ indicator can bring us. Remember that $\mathcal{I}_{up,params}$ is a vector containing the variation in each parameter, equivalent to the deletion of z from D_{train}. For the binary logistic regression that we develop here, this allows us to know which points are the most influential on a given variable. Nevertheless, for a neural network, it is not so simple since the weights of the network have no immediate significance. This is why, exceptionally, we do not apply this indicator to neural networks.

Unlike other intelligibility methods, the output is not directly an explanation of the black-box decision chain. To obtain useful information, we have to examine the most influential points. This is not always easy, especially for large-scale datasets. However, the boxplot **Fig. 4.5** gives a general overview of the coefficient's sensitivity to observations of D_{train}.

On this boxplot, it is possible to see that the model's intercept coefficient is most sensitive to the removal of a point from the training sample. If we remove a point corresponding to the extremes of the blue boxplot, the model's intercept coefficient increases or decreases by approximately 8.5%. These two points are represented in green in **Fig. 4.6**. They correspond to points misclassified by the model. The model is also very confident in its prediction. This is represented by an orange or dark blue color.

This first indicator, therefore, provides information on the sensitivity of the black-box parameters to any observation. In this way, it is possible to point out aberrant or very influential points and analyze them.

4.4.2 Identification of atypical points

Let us now apply the $\mathcal{I}_{up,params}$ indicator to our logistic regression fitted on our insurance dataset. We want to identify the most sensitive variables to the removal of a point in D_{train}. The information can be synthesized as a boxplot **Fig. 4.7**. It seems that coefficients related to variables `License seniority 2`, `Max speed`, `Vehicle value` and `Vehicle weight` are sensitive to some observations of D_{train}. Removing a single point among the $74,983$ observations of D_{train} can result in variations of up to -15%. In contrast, the intercept coefficient seems to be insensitive to the D_{train} points. Note that the most sensitive variables are consistent with what we can expect from an actuarial point of view. In fact, the variables identified as relevant for the identification of claims in the actuarial literature are related to the driver's experience and the power of the vehicle. These include Ohlsson and Johansson (2010) and Charpentier (2014) and a study linking conventional data with *telematics* Verbelen et al. (2018) data.

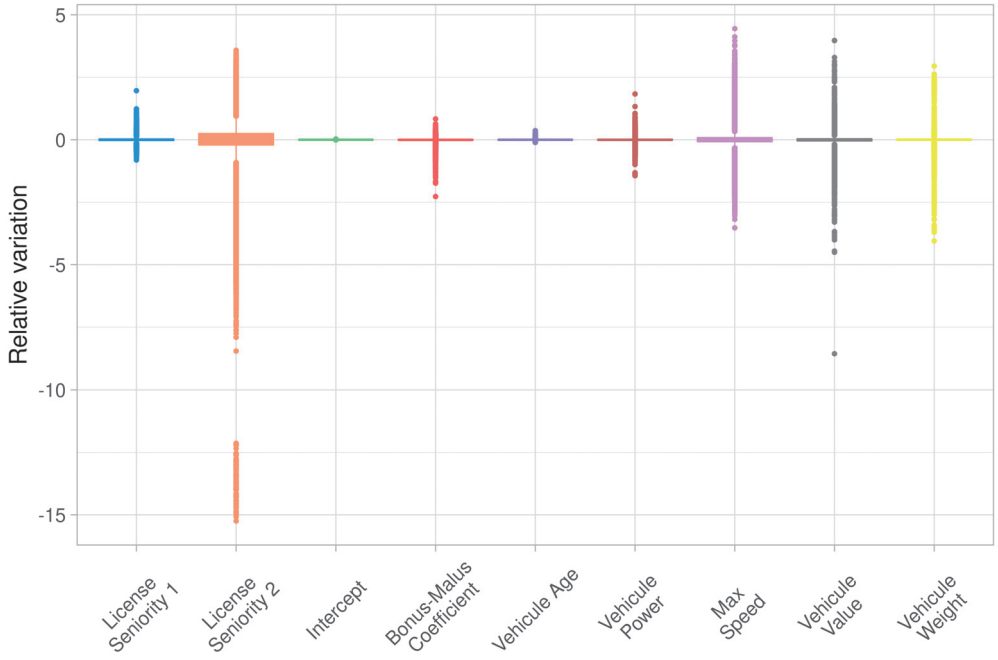

Fig. 4.7: Sensitivity of each predictor to the deletion of $z \in D_{train}$

Tab. 4.1: Influential points

	Bonus/Malus	License seniority 1	License seniority 2	Vehicle age	Vehicle power	Max speed	Vehicle value	Vehicle weight	Target
7,743	0.50	41	0	19	3,299	270	112,538	1,260	1
49,950	0.57	16	0	51	425	88	1,060	880	1
65,499	0.50	111	0	12	1,598	195	16,770	1,230	1

We can also analyze in detail the points that perturbed the coefficients the most. For example, looking at **Tab. 4.1** we can see that the vehicle responsible for the greatest variation in "Vehicle weight" is the lightest among the most powerful and most expensive vehicles. This atypical point is, therefore, more difficult to classify for the model and consequently appears to be very influential. It causes

the coefficients of `Vehicle weight` and `Vehicle value` to vary by $+2.95\%$ and -8.57%, respectively. Another example is the point whose removal most disturbs the estimate of the `License seniority` 2 coefficient. This policyholder has license seniority of 111 years, which means that the driver is at least 119 years old. Here, it is likely that it is a mistyped value. Thus, we can easily identify anomalies in D_{train} or atypical individuals who should be treated afterward.

Limits

For this indicator to be relevant, the parameters need to be directly related to the variables as for logistic regression. For neural networks, for example, this indicator is not without interest, but what was developed above no longer holds.

Another important limitation is the difficulty in summarizing the information. We generate a vector for each point of D_{train}. It is unreasonable to expect a human to manage so much data. Here, we adopt a graphical approach to overcome this problem.

4.5 Identification of influential points for a prediction

We now want to measure the variation in the loss function evaluated at any z_{test} if we remove a point from D_{train}. We, therefore, approximate the difference in loss $\mathcal{L}(z_{test}, \hat{\theta}_{n,-z}) - \mathcal{L}(z_{test}, \hat{\theta}_n)$. We develop the proof for M-estimators. Identifying influential points according to $\mathcal{I}_{up,loss}$, the indicator we develop below, is useful for detecting outliers but also allows locating points close to the decision boundary. These points allow us to locate the decision frontier and to provide surrogate models that are locally faithful to the decision boundary. Moreover, $\mathcal{I}_{up,loss}$ is very useful for prioritizing the points to be analyzed, i.e., those for which it is useful to build a hyperplane to substitute the complex boundary.

Definition 7 *Let z and z_{test} be points of D_{train} and D_{test}, respectively. We define the loss variation at point z_{test} related to an infinitesimal variation in ϵ by*

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

The underlying idea is approximately the same as for $\mathcal{I}_{up,params}$: approximate the variation in $\mathcal{L}$ at point z_{test} when we remove z. We do this by perturbing the distribution function of the observations upweighting the point z by $\epsilon = -1/n$. The Proposition 1.3 is a closed formula for this indicator.

Proposition 3

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

By estimating the inverse of the Hessian and the gradients at the points of interest, we approximate the effect of removing z on the prediction in z_{test}.

Geometrical interpretation

For some models, it is possible to show that

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

is a dot product. This allows us to provide a geometrical interpretation of this indicator. We consider here the case of binary logistic regression. To do so, we must show that the linear form associated with $H_{\hat{\theta}_n}$ is definite and positive. Proposition 1.4 formalizes this.

Proposition 4 *In the case of a regularized binary logistic regression, $H_{\hat{\theta}_n}$ is positive definite.*

Let us consider a binary logistic regression model with a loss function L_2-regularized with parameter $\alpha > 0$, and $H_{\hat{\theta}_n}$ the Hessian matrix associated with it. Under these conditions,

$$H_{\hat{\theta}_n} \textit{ is positive definite.}$$

By showing that $H_{\hat{\theta}_n}$ is definite positive, we show at the same time that $H_{\hat{\theta}_n}$ is invertible. This proves that ϕ exists. It is then quite easy to see that ϕ is a dot product.

This property is interesting since by calculating $\phi(z, z)$, we obtain the gradient's norm at point z, which also happens to be $\mathcal{I}_{up,loss}(z, z)$. By doing this, we placed both gradients in the same direction, which maximizes the projection. This provides a measure of importance for points in D_{train}.

4.5.1 Implementation of $\mathcal{I}_{up,loss}$

To understand how the influence varies in the case of log-loss, we use a data set of 5 learning points and 1 test point. In the first step, logistic regression is fitted on the learning sample. This is our reference model. Then, the training observations are removed one by one, and logistic regression is fitted on the resulting samples. Finally, for each model [6] the classes/probabilities are predicted for z_{test}. For these models, if the probability is greater than 0.5, the predicted class is 1; otherwise, 0. Recall that:

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

Note that this definition makes the notion of influence, as defined above, consistent with influence, as we conceive it in everyday life. Indeed, if the removal of a point improves the model (reduces the loss compared to the reference model), then the influence of this observation is negative. Keeping this point, therefore, degrades the quality of the model and has a negative influence on it. In contrast, a point has a positive influence if its removal increases the loss.

[6] reference model and models without one point

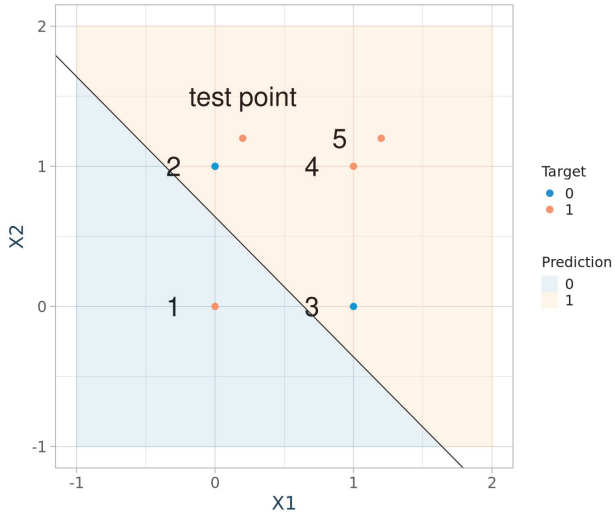

Fig. 4.8: Reference

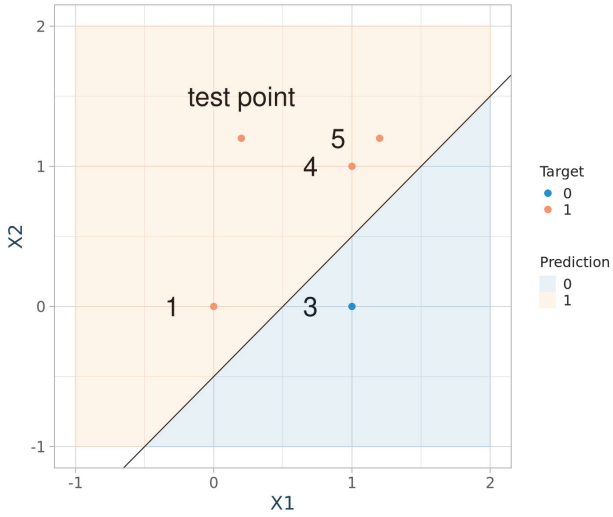

Fig. 4.9: Without point 2

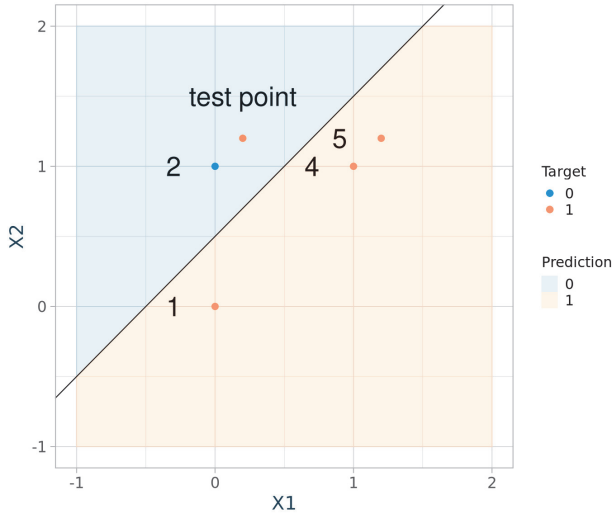

Fig. 4.10: Without point 3

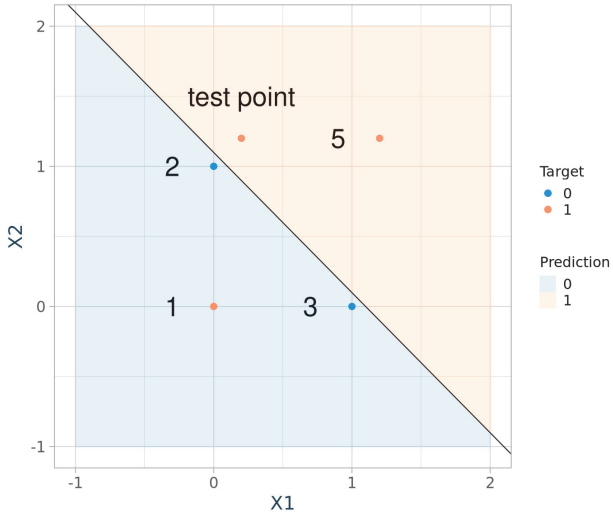

Fig. 4.11: Without point 4

In this simple case in **Fig. 4.8** to **Fig. 4.11**, it is easy to see that the classifier makes its decisions according to the position of points with respect to the decision boundary. Removing a point from the training sample changes the decision boundary and thus the classifier's prediction (value or probability). If we look at the reference graph in **Fig. 4.8** and **Fig. 4.9**, we notice that the removal of point 2 significantly modifies the decision boundary. This new decision boundary separates perfectly the points of the training sample and by chance the sample from the test point. Hence, the model predicts the test point as 1 with a high probability since the test point is "inside its zone" and far from the decision boundary. The loss in this case is lower than that of the reference model for the test point prediction. Consequently, the influence is negative, as shown in **Tab. 4.6** in the appendix. Similar reasoning can be applied to the rest of the graphs.

Remark 24 *It is interesting to note that for the test point to be predicted (class 1), the points with the highest positive influence and the highest negative influence are of a different class. This is not an intuitive result, but it shows us that the method identifies key points for a given model. In the rest of the article, we will see that the identified points are intrinsically linked to a black-box model.*

4.5.2 Identification of outliers

The information provided by $\mathcal{I}_{up,loss}$ is useful for highlighting important points in the learning sample for a given prediction. This can be very helpful for debugging a model, for example, by determining that the data have not been mislabeled as suggested by Koh and Liang (2017) or analyzing the characteristics of the most negatively influential points for a prediction at point z_{test}. Thus, it is possible to rectify these observations or delete them if necessary. For a prediction at z_{test}, every point of D_{train} is assigned a value of positive or negative influence. Remember that the intensity of this influence depends on the loss variation evaluated at z_{test}. Therefore, outliers for a given model should have one of the highest negative influences.

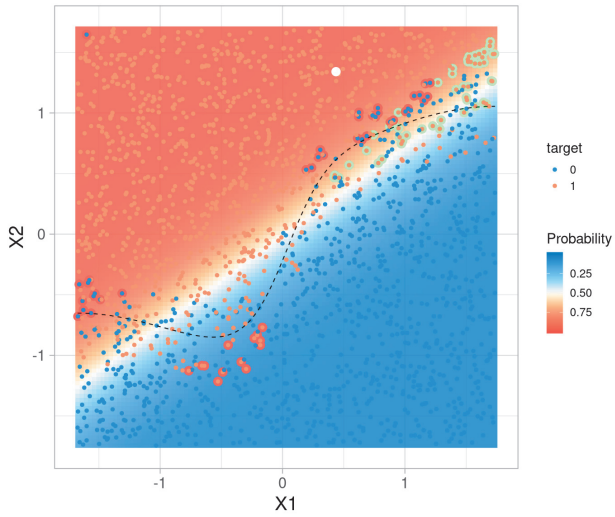

Fig. 4.12: Linear model

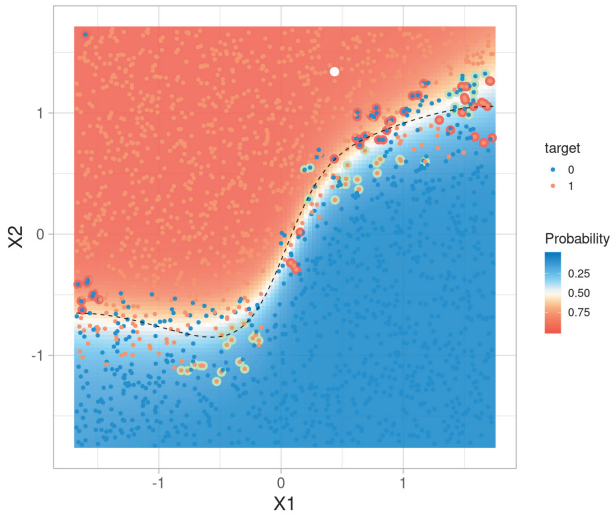

Fig. 4.13: Neural network

We now want to determine whether it is possible to identify an outlier using $\mathcal{I}_{up,loss}$. Additionally, we seek to illustrate the information provided by this indicator. For this purpose, we create two models which are shown in **Fig 4.12** and **4.13**. The color gradient represents the prediction of each model. The more confident the model is in its prediction, the darker the color. White marks the area where the model is uncertain, which is the decision boundary. We can already see that the linear model is not flexible enough to fit correctly to the true decision boundary in the dotted lines. In contrast, the regularized neural network fits better. We intentionally modify the label of a point at the top left of the graph. This way, we create an aberrant point. We randomly select a point to predict z_{test} that belongs to D_{test}. This point is represented by a white halo on the graph. The points circled with a green circle are the 50 points that have the highest positive influence on this prediction. The red

points are the 50 points that have the highest negative influence on the prediction. We notice that the points identified for the two models are not necessarily the same. Nonetheless, they are located close to the decision boundary of each model. Moreover, our aberrant point on the top left has been identified by the method. For both models, this is the most negatively influential point. For the linear model, the influence of this point is 27 times higher in absolute value than the 5% percentile of the influence distribution. For the neural network, the influence of this point is 9 times higher than the 5% percentile. The indicator can be used for both logistic regression and neural networks. This can provide a method for efficiently diagnosing outliers for a given model. However, to be exhaustive, as many distributions as there are test points should be analyzed, which in practice can be difficult.

4.5.3 Extraction of local explanations

We can see that it is hard to extract other relevant information from this indicator since the location of influential points varies greatly in space. Let us assume that we want to obtain extra information from the influences. It is reasonable to locally fit a transparent model, such as a regression tree, for instance, taking as a new target the signed influence of each point. Hence, we can model the influence of points as a function of predictors and see where influential points are for any prediction. Nevertheless, negative and positive influences are too mixed up to produce a shallow tree. This means that our local explanation of the black-box model would be too complex. Therefore, this technique suggested by Molnar (2020) does not give reliable and interesting results. This remark is valid for our example in small dimensions and a fortiori in large dimensions.

Example in 2 dimensions

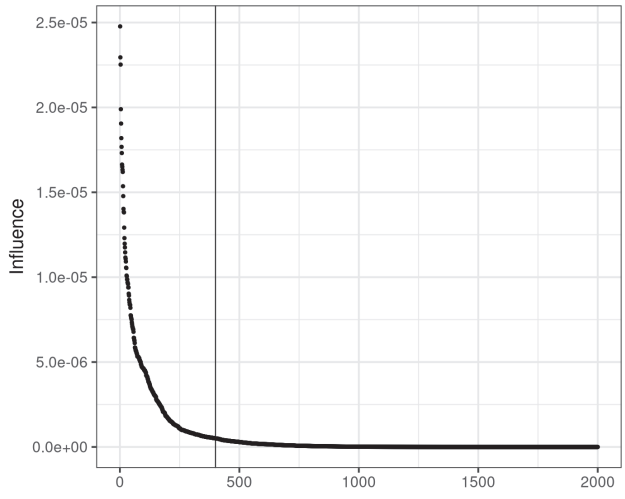

Fig. 4.14: Linear model

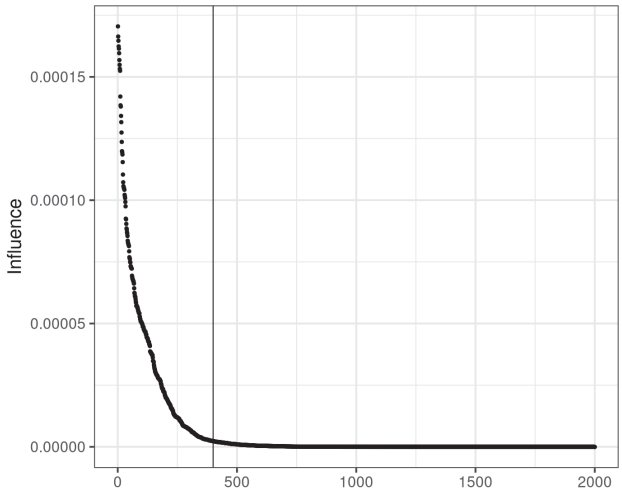

Fig. 4.15: Neural network

We, therefore, propose a different approach to extract relevant information to explain our models. Since the formula $\mathcal{I}_{up,loss}$ is valid for any point z_{test}, it is also valid for the points in D_{train}. Both Hessian matrices are definite positive. In this case, $\mathcal{I}_{up,loss}(z, z)$ is a norm for the gradient of the loss function evaluated at point z. Thus we obtain a distribution of positive influences. Moreover, we considerably reduced the information since we have only one influence value for each point in D_{train}. This is easier to interpret. The removal of a highly influential point greatly disturbs the model while the removal of a null or almost null influence point causes only small modifications on the predicted values. When Hessian matrices are not definite positive, we can take the absolute value of the influence values to be able to apply the method described below. We represent for each model the influential points in descending order in the **Fig. 4.14** and **4.15**. The choice of a threshold for the number of influential points is left to the user. We arbitrarily select a threshold of 400 observations for both models. Points are then represented in the 2 dimensional space. Now, influential points are identified by a green circle[7] As can be seen in **Fig. 4.16** and **4.17** the points identified as influential are different for each model. In addition, they have the interesting property of being located close to the decision boundary. This is consistent with what we developed above. However, it remains difficult to extract an explanation from this set of points. Given the location of the points, taking the average over each of the coordinates is of no use. Nevertheless, we can use the influence as new information. Contrary to what is proposed by Molnar (2020) we do not fit a model directly on the influence of each observation but rather use the properties related to these points and the models.

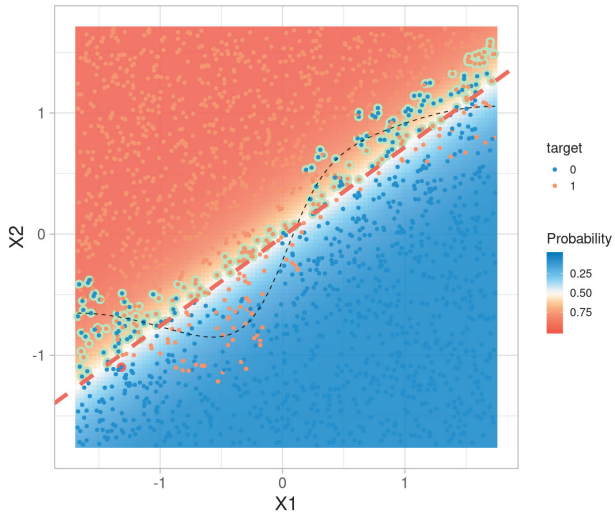

Fig. 4.16: Linear model

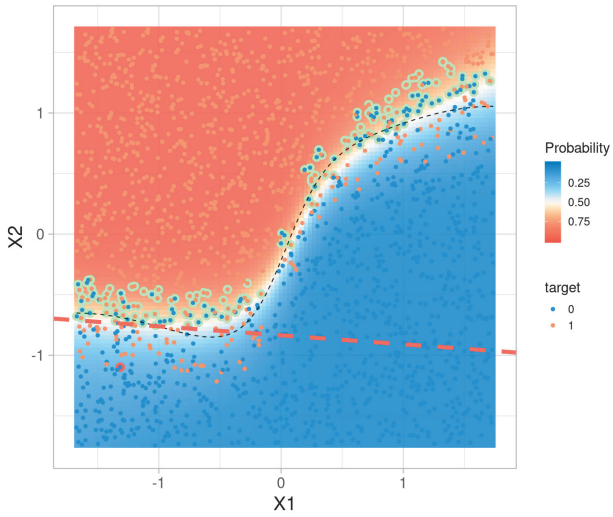

Fig. 4.17: Neural network

We suggest the following strategy to provide a local explanation only valid in the vicinity of the point being explained.

1. Select an instance to explain that we denote by z_{exp}.

[7] To keep the graphs simple and consistent with the explanations that follow, we display only the influential points that are on the opposite side (of the boundary) to the red point.

2. Among the most influential points, keep only those that are on the other side of the border with respect to the point to be explained. We can easily obtain this information since we can query the black-box model over the whole space.

3. Find the intersection point between the decision boundary and the straight lines passing through the point to be explained and an influential point on the other side of the boundary. These points are those identified in the previous step. We use a dichotomy method here, which lets us arbitrarily set the precision.

4. For each point of intersection, locate the hyperplane $\mathcal{H}_i$ tangent to the decision surface at that point. Hyperplanes are the geometric equivalent of linear models in dimension d. Here they are a set of local explanations that are admissible.

5. Calculate the distance to the point we are trying to explain and the distance to the hyperplane $\mathcal{H}_i$ for each point of step 3. Calculate the sum of these two distances. Use only the minimum distance. We note this distance d_{min}. We have found a hyperplane that can be used as an explanation for the instance we are studying. We now have to evaluate its local fidelity.

6. There are several reasonable methods for defining a neighborhood for the point to be explained. For example, we can take a ball centered on the point we are attempting to interpret and with a radius large enough to contain observations of both classes. It is also possible to choose a ball of the same radius but centered on the point tangent to the decision surface. It is the latter that we retain. k points[8] are pulled uniformly in this d-dimensional ball. These points are new observations intended to evaluate the fidelity of the explanation with respect to the black-box model.

7. Assess the fidelity of the explanations predicting the linear surrogate model (our explanation) and the black-box model on the previously created sample. We can use the *accuracy*, *AUC*, *precision* and *recall* to measure the fidelity. This step validates the explanation. If the fidelity is too low, we reject the explanation.

Fig. 4.16 and **4.17** show an example of an explanation[9] for the point surrounded by a red circle for our two models. According to the *accuracy*, the explanation is 100% faithful to the linear model and 99.8% faithful to the neural network in the vicinity of the point to be explained. Of course, this fidelity score varies according to the explanation and the complexity of the black-box model. As expected, in the case of the linear model only, the explanation is very close to the true decision frontier. For the linear model, the local explanation is consistent with the global explanation. For the neural network, it only appears to be locally good. In higher dimensions, we can only measure the quality of an explanation with the fidelity measure to assert its quality.

[8] The choice is left to the user.
[9] Red dashed line.

Insurance problem

Now, we return to our prevention problem on insurance data. We fit a linear model and a neural network containing two hidden layers. The first hidden layer is composed of 4 neurons, while the second contains 2 neurons. Generally, for binary classification, one output neuron is sufficient. However, we use the **h2o** framework, which imposes the number of output neurons to match the number of classes. Thus, our network has 44 weights and 8 bias. This requires 52 coefficients to be estimated. We introduce a regularization of type *L2*. The value of this hyperparameter is 0.001. The performances of the two models are detailed in **Tab 4.2**.

Tab. 4.2: Performance of black-box models

	threshold	AUC	PRAUC	precision	recall
Linear model	0.114	0.616	0.171	0.158	0.761
Neural network	0.121	0.619	0.174	0.157	0.781

We want an explanation in the vicinity of influential observations for both models. This allows us to locally compare the way the models make their decision. We illustrate this by giving an explanation for the first two individuals of the **Tab 4.3**.

Tab. 4.3: Influential points

	Bonus-Malus	Ancienneté 1	Ancienneté 2	Age Vehicule	Cylindrée	Vitesse max	Valeur	Poids	Target
984	0.5	20	0	23	1913	118	19,381	1,480	1
1,619	0.5	60	0	11	1998	230	30,250	1,490	1
32,294	0.5	41	0	28	3,980	126	17,982	1970	1
7,934	0.5	58	0	13	1,396	185	13,568	1,000	0

Applying the strategy proposed above, we obtain a distribution of influences represented in **Fig. 4.18** and **4.19**. The two most influential points for the linear model are the same as those identified with the previous indicator. For the neural network, the Hessian matrix is invertible but not definite positive. This is why we obtain positive and negative influence values. To choose our threshold of points to keep, we, therefore, take the absolute value of the influences. Note that here again, some points can be differentiated by their very high influence. These are points 32,294 and 7,934 whose characteristics are given in the **Tab. 4.3**. Observing `Vehicle power` and `Max speed` for point 32,294, we see that there is a consistency problem between `Vehicle power` and `Max speed`. This may explain this strong influence value for the linear model. As we have just shown, in practice, $\mathcal{I}_{up,loss}$ is very useful for identifying abnormal points. As generalized linear models are still very widespread among insurers, the use of this indicator can be part of a data quality improvement approach.

We now want to extract local explanations for the selected points. Based on the **Fig. 4.18** and **4.19** we choose $2,250$ points for the linear model and $7,000$ points for the neural network. Giving

the explanations for the linear model, we want to ensure that in higher dimensions, the provided explanations are still consistent with the linear model. Fig. 4.20 and 4.21 empirically confirm this. For the linear model, this observation holds for every point and not only for these two examples. We checked it but cannot display all the results. For the neural network, we expect different hyperplanes for the explanations of individuals 984 and $1,619$ since the neural network is nonlinear. This is what we see in Fig. 4.22 and 4.23. According to the *accuracy*, the explanation of point 984 for the linear model is 100% faithful. For the neural network, the fidelity is 97.4%. This remains high enough to consider that the explanation is reliable near this point.

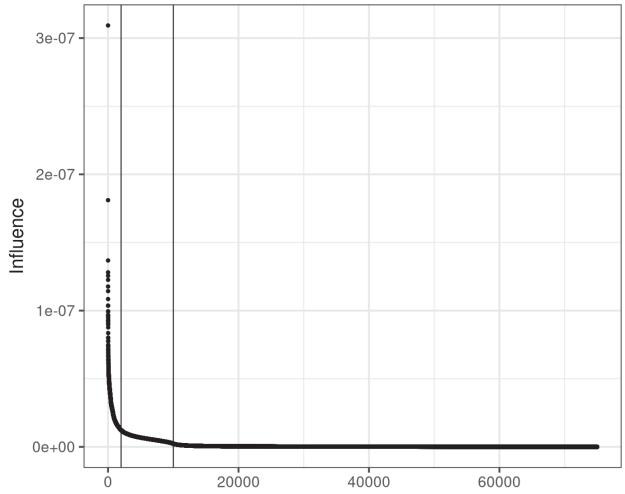

Fig. 4.18: Linear model

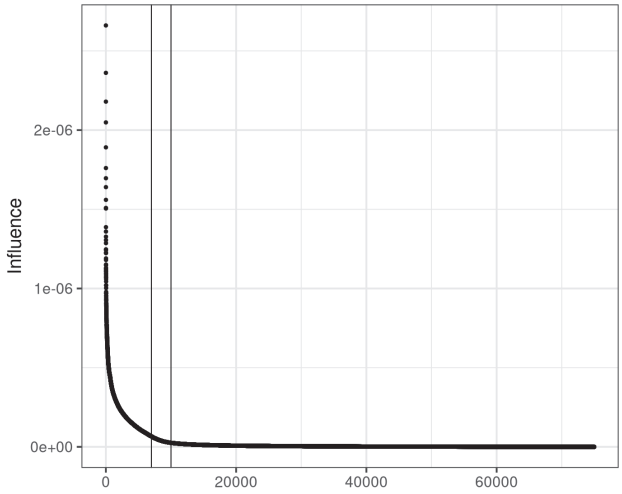

Fig. 4.19: Neural network

Remark 25 *Often, when a surrogate model is fitted to explain the results of a more complex model, the nature of the explanation may seem inconsistent with our expectations. As humans, we prefer explanations that correspond with our expectations. Hence, it can be disturbing to see that* `Vehicle age` *has greater importance than* `License seniority 1` *in predicting the occurrence of a claim. However, we should consider that we are trying to explain what the model has "understood". A model can perform well and capture spurious correlations. Thus, a black-box model can be an efficient statistical solution and produce poor explanations for a human. Although the explanations provided here do not seem to us to be actuarially consistent, they do explain what the different models have learned locally.*

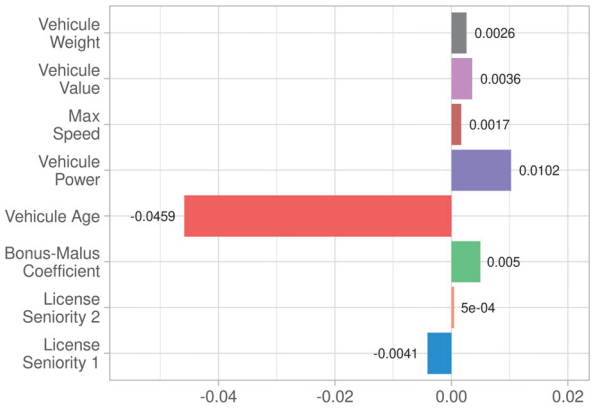

Fig. 4.20: Explanation for point 984

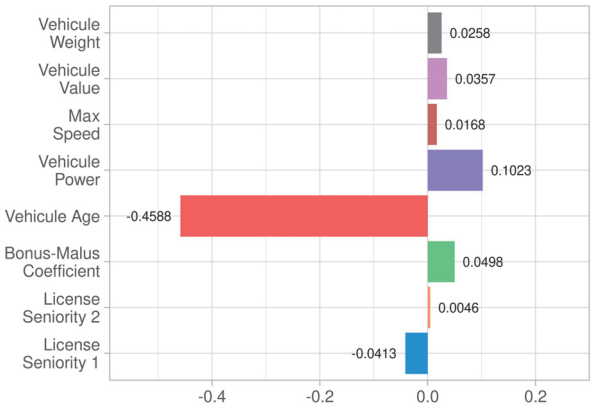

Fig. 4.21: Linear model

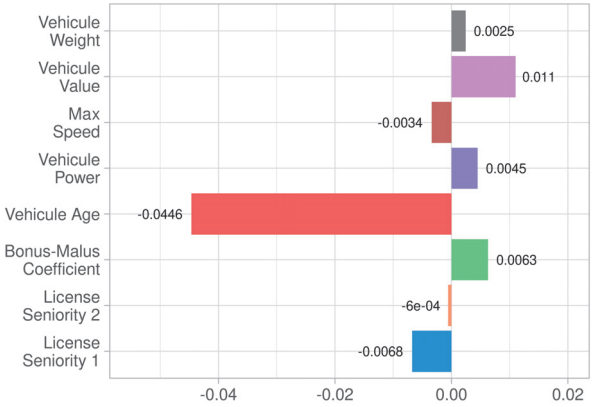

Fig. 4.22: Explanation for point 984

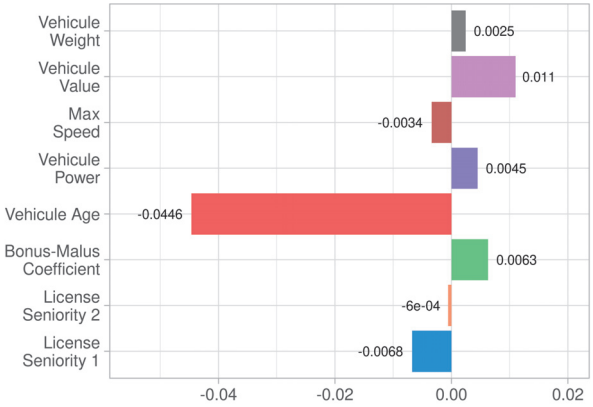

Fig. 4.23: Explanation for point 1,619

4.6 Identification of a maximal perturbation direction

Contrary to the previous indicators, in this section, we are not interested in the variations caused by the removal of a point. Instead, we are interested in the perturbation of a point. We want to define the direction in which moving a point of D_{train} maximizes the loss variation for any prediction. Such an indicator provides a direction in which to find an "adversarial example". This means a fictitious observation close to the initial observation, which is predicted with a different class by the black box model. In our framework, this allows us to locate the decision frontier.

This time, the idea is to transfer the mass ϵ of a point z to a perturbed point $z_\delta = (x + \delta, y)$. We can decompose this transfer into two steps. The first step is to remove z from D_{train}. Then, in a second step, a perturbed point z_δ is added to D_{train}. We denote by $\hat{\theta}_{n,z_\delta,-z}$, the empirical risk minimizer where z_δ replaces z in D_{train} and $\hat{\theta}_{n,\epsilon,z_\delta,-z}$ is the vector of optimal parameters associated

with the contaminated distribution ($F_{\epsilon,z,z_\delta} = F - \epsilon\delta_{z_\delta} + \epsilon\delta_z$). Initially, we do not make any particular assumptions about the nature of the explanatory variables or about δ.

Definition 8 *Let z and z_{test} be two points belonging to the training sample and the test sample. We define the loss variation at point z_{test} related to an infinitesimal mass transfer ϵ from z to z_δ by*

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

Proposition 5 gives a closed formula that is valid even for discrete data. In the rest of the article, we demonstrate that it is possible to refine the approximation further in the case of continuous numerical data.

Proposition 5

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

If some variables are discrete, this makes it possible to set a vector of perturbation δ to evaluate the loss variation in this direction.

In the case of continuous variables, it is possible to refine the approximation.

Proposition 6 *Let z be a point of the training sample and z_δ its perturbed version, $\hat{\theta}_n$ and $\hat{\theta}_{n,z_\delta,-z}$ are the coefficients estimated with the original and perturbed versions, respectively, of point z. In the case of mixed data (continuous and discrete),*

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

in the case of continuous data only,

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

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

In the case of continuous numerical data, we can obtain a finer indicator. For a prediction at a test point, it gives the directions/variables in which the δ perturbations have the greatest impact.

Proposition 7 *Let us assume that the explanatory variables are continuous, i.e.,.* $x \in \mathcal{X} \in \mathbb{R}^p$ *and* $||\delta|| \to 0$. *If* $\mathcal{L}$ *is differentiable with respect to* x *and* θ, *then*

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

This indicator allows us to obtain additional information on the influential variables when making a prediction.

4.6.1 Implementation of $\mathcal{I}_{pert,loss}(z, z_{test})^\intercal$

We begin by illustrating the information carried by $\mathcal{I}_{pert,loss}^\intercal$ in two dimensions. The formulas are valid for any z_{test} and a fortiori for a point of D_{train}. For the same reasons as before, we only calculate this indicator for the most influential D_{train} points. We can sort points according to their importance to avoid analyzing the totality of the observations. In Fig. 4.24 and 4.25, we represent the vectors bearing the direction in which to move the hundred most influential points to perturb the prediction the most. In addition, we represent in red a point chosen at random among the most influential points. Since we have a direction pointing to the decision boundary, we have a vector orthogonal to a set of hyperplanes, one of which is tangent to the decision surface. This is why we can use this indicator as an explanation. The idea was suggested by Koh and Liang (2017).

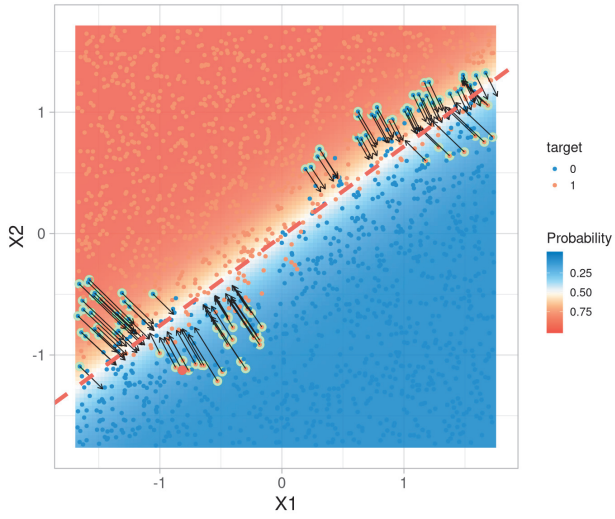

Fig. 4.24: Linear model

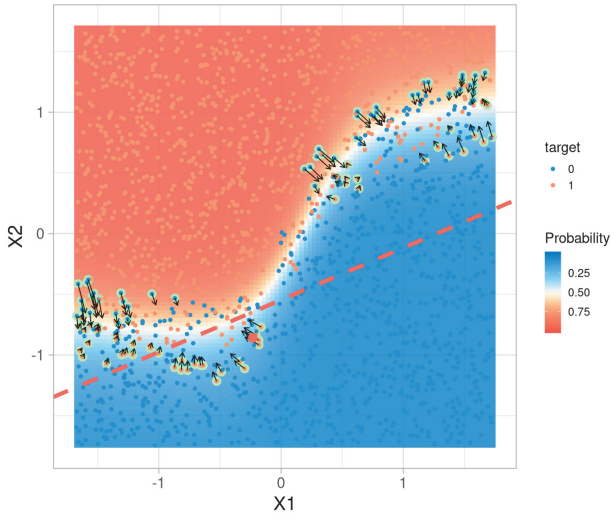

Fig. 4.25: Neural network

We also display in these figures the hyperplane with a red dotted line for the observation to be explained. Once again for the linear model, the hyperplane is consistent with the decision boundary. For the neural network, the approximation is good. However, to check the quality of this approximation, we must, as before, define a neighborhood for this point and evaluate the fidelity. For this, we predict each point of the neighborhood with the hyperplane (our explanation of the black-box model). Then, we measure the fidelity between these predictions and the black-box prediction for the same predictions. The **Tab. 4.4** synthesizes the fidelity measures for these explanations.

Tab. 4.4: Influential points

	Model	Accuracy	AUC	F1 Score	Precision	Recall	PRAUC
361	Linear model	1	1	1	1	1	0.998
74,973	Linear model	1	1	1	1	1	0.998
15,540	Neural network	0.915	0.994	0.919	0.998	0.852	0.990
3,973	Neural network	1	1	1	1	1	0.998

Let us now apply this method to our example on real insurance data. With the same technique as before, we restrict the number of observations for which we want an explanation to $2,250$ for the linear model and $7,000$ for the neural network. We thus calculate $\mathcal{I}_{pert,loss}(z,z)^\intercal$ for each of these observations. Therefore, we can define a hyperplane for each influential point. Each hyperplane is an explanation. Then, we evaluate the quality of the explanations. Since we have a set of explanations, we can either analyze them one by one or represent the variability of the collection. For this, we consider, **Fig. 4.26** and **4.27**. These are the boxplots of the hyperplane's coefficient distributions. We previously centered each of the variables. As can be seen, for the linear model, the explanations do not differ much, which is rather reassuring. In contrast, for the neural network, the hyperplanes fluctuate more. This is justified by the nonlinearity of the neural network. This variability of explanations is not directly exploitable. However, it can allow us to define homogeneous groups of explanations and improve our global understanding of the model, though, this is outside the scope of this study. The explanations provided are generally faithful. Nevertheless, for the neural network, they are less accurate than for the linear model as shown in **Tab. 4.4**.

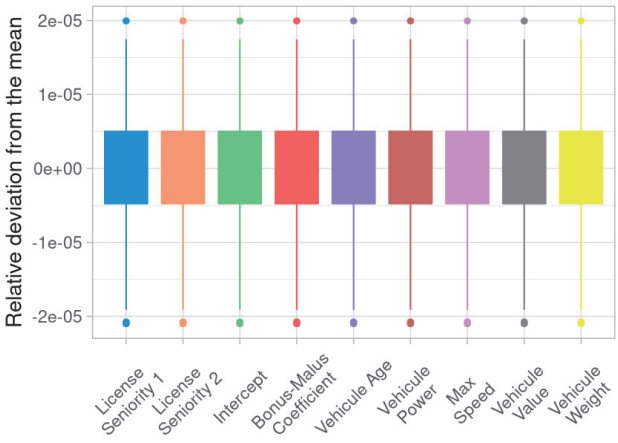

Fig. 4.26: Linear model

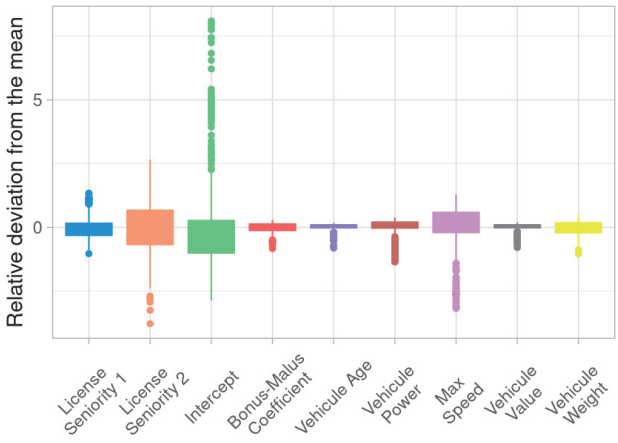

Fig. 4.27: Neural network

Limits

This indicator, therefore, allows us to efficiently define a locally faithful hyperplane for the most influential points. However, it requires a stronger assumption than for $\mathcal{I}_{up,loss}$, the continuity of the explanatory variables. In insurance, there are many categorical predictors. Thus, this indicator is not always exploitable in practice. However, if it is not possible to use it, we can still use the strategy developed in the previous section with $\mathcal{I}_{up,loss}$. This requires more calculations but allows us to have a faithful explanation under weaker assumptions. Moreover, we noticed during our tests that for points with a weak influence, the calculation of $\mathcal{I}_{up,loss}$ is less accurate and can give hyperplanes of lower quality.

4.7 Conclusion and future work

In this article, we presented three indicators based on influence functions. Therefore, it is possible to efficiently identify observations considered abnormal by a given model or to prioritize points to be analyzed. Moreover, by combining the information carried by these indicators with properties of parametric models, we can extract locally faithful explanations for points that we want to analyze. In practice, we can, therefore, use these indicators in a data quality improvement process. The models we studied here include generalized linear models, which are still very widespread among insurers. Finally, it is also possible to use the methods presented in this article to debug a model locally as with LIME or SHAP. The advantage over the latter is that we can restrict the number of points to be analyzed.

It might be relevant to extend this work by adapting the technique for regression. Moreover, as we have seen in the case of neural networks, we may consider grouping similar explanations together to reduce their number and thus further improve our understanding of parametric black-box models.

Acknowledgments

We would like to thank Arthur Charpentier for the data he provided.

4.8 Appendix

4.8.1 Data

Tab. 4.5: Data dictionnary

Variable name	Description
`Bonus/Malus`	is the Bonus/Malus coefficient of the policyholder. It is between 0.5 and 3.5. The lowest is the best. It starts at 1 for young drivers.
`Policy coverage level`	is the level of coverage of the policy: Mini, Median1, Median2, Maxi.
`Coverage period`	is the number of years of the policy.
`Duration since the last modification`	represents the seniority of the current policy since the last change.
`Payment frequency`	is the frequency of payment: annual, biannual, quarterly, monthly.
`Pay-as-you-drive`	indicates whether the customer has subscribed to a Pay-as-you-drive offer.
`Usage`	describes the policyholder's use of its vehicle.
`INSEE code`	is the INSEE code identifying the commune or department of the policyholder.
`Driver's gender 1`	is the gender of the first driver.
`Driver's gender 2`	is the gender of the second driver.
`License seniority 1`	is the license seniority of the first driver.
`License seniority 2`	is the license seniority of the second driver.
`Vehicle age`	is the age (in years) of the vehicle since its release.
`Vehicle power`	represents vehicle power.
`Fuel type`	is the fuel type of the vehicle.
`Vehicle brand`	is the name of the vehicle manufacturer: Renault, Peugeot or Citroen.
`Vehicle model`	is the model of the vehicle.
`Max speed`	is the maximum speed of the vehicle.
`Vehicle type`	tourism, commercial.
`Vehicle weight`	is the mass of the vehicle (en kg).

4.8.2 Proofs

Proof 2 (Proof for proposition 1) *Let us show that* $\mathcal{I}_{up,params}(z) = -H_{\hat{\theta}_n}^{-1}\nabla_\theta \mathcal{L}(z, \hat{\theta}_n)$.

Let $Z \sim F$*, where* F *is the distribution function for the data and* $Z_\epsilon \sim F_{\epsilon,z} = (1-\epsilon)F + \epsilon\delta_z$ *its contaminated version. We want to minimize* $\mathbb{E}[\mathcal{L}(Z,\theta)]$ *i.e.,* $\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(z_i,\theta)$*, which is minimal in* $\hat{\theta}_n$*, under the assumption that the minimum is reached where* ∇_θ *is null,*

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

which is rewritten as

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

Let us denote by $\hat{\theta}_{n,\epsilon,-z}$ the vector of optimal parameters for the contaminated distribution. We seek to calculate $\frac{d\hat{\theta}_{n,\epsilon,-z}}{d\epsilon}$. The contaminated distribution of the previous equation is rewritten as

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

Deriving with respect to ϵ we have

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

When ϵ goes toward 0, then using the nullity condition of the gradient we obtain

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

Proof 3 (Proof for proposition 2) *It remains to show that $\hat{\theta}_{n,-z} - \hat{\theta}_n = -\frac{1}{n}\mathcal{I}_{up,params}(z) + o_p(\frac{1}{n})$*

Let $Z_1,\ldots,Z_n$ be independent and identically distributed variables. Let us show in the framework of finite samples that the (asymptotic) influence of the n-i$\grave{e}me$ element can be expressed as follows:

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

We write the first-order of Taylor's development in θ_0.

$$0 = \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta \mathcal{L}(Z_i,\theta_0) + \frac{1}{n}\sum_{i=1}^{n}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0)(\hat{\theta}_n - \theta_0) + \frac{1}{2}(\hat{\theta}_n - \theta_0)^\intercal \frac{1}{n}\sum_{i=1}^{n}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0)(\hat{\theta}_n - \theta_0) + o(||(\hat{\theta}_n - \theta_0)||^2). \quad (4.2)$$

The same development can be written for $\hat{\theta}_{n-1}$.

$$0 = \frac{1}{n-1}\sum_{i=1}^{n-1}\nabla_\theta \mathcal{L}(Z_i,\theta_0) + \frac{1}{n-1}\sum_{i=1}^{n-1}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0)(\hat{\theta}_{n-1} - \theta_0) + \frac{1}{2}(\hat{\theta}_{n-1} - \theta_0)^\intercal \frac{1}{n-1}\sum_{i=1}^{n-1}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0)(\hat{\theta}_{n-1} - \theta_0) + o(||(\hat{\theta}_{n-1} - \theta_0)||^2). \quad (4.3)$$

From equations (4.2) and (4.3) we obtain the following equality:

$$\begin{aligned} -\frac{1}{n}\nabla_\theta \mathcal{L}(Z_n,\theta_0) = &-\frac{1}{n(n-1)}\sum_{i=1}^{n-1}\nabla_\theta \mathcal{L}(Z_i,\theta_0) + \frac{(\hat{\theta}_n - \hat{\theta}_{n-1})}{n-1}\sum_{i=1}^{n-1}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0) \\ &+ \frac{(\hat{\theta}_n - \theta_0)}{n-1}\left[\frac{1}{n}\sum_{i=1}^{n}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0) - \frac{1}{n-1}\nabla^2_\theta \mathcal{L}(Z_n,\theta_0)\right] \\ &+ \frac{1}{2}\left[(\hat{\theta}_n - \theta_0)^\intercal \frac{1}{n}\sum_{i=1}^{n}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0)(\hat{\theta}_n - \theta_0) - (\hat{\theta}_{n-1} - \theta_0)^\intercal \frac{1}{n-1}\sum_{i=1}^{n-1}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0)(\hat{\theta}_{n-1} - \theta_0)\right] \\ &+ o(||(\hat{\theta}_n - \theta_0)||^2). \quad (4.4) \end{aligned}$$

In the development (4.4), we set $Z_n = z$. *The first term of the right member can be rewritten* $-\frac{1}{n}\left[\frac{1}{n-1}\sum_{i=1}^{n-1}\nabla_\theta \mathcal{L}(Z_i,\theta_0)\right]$. *In this way it is easy to see that the bracketed term multiplied by* $\sqrt{n}$ *converges in law toward a centered multivariate normal distribution according to the central limit theorem. This term is, therefore, bounded in probability and is* $O_p(\frac{1}{n^{3/2}})$. *The second term of the right member can be written as follows* $(\hat{\theta}_n - \hat{\theta}_{n-1})\left[\frac{1}{n-1}\sum_{i=1}^{n-1}\nabla^2_\theta \mathcal{L}(Z_i,\theta_0)\right]$, *where the term between brackets converges in probability toward the matrix* H_{θ_0} *due to the law of large numbers. It is due to the hypothesis* $\mathbb{E}[||\nabla^2_\theta \mathcal{L}(Z,\theta_0)||] < \infty$. *The third term is* $o_p(\frac{1}{n})$ *after multiplying by* $\sqrt{n}$. *The fourth term is a sum of probability bounded terms. We make the hypothesis that* $\frac{1}{n}\sum_{i=1}^{n}\nabla^3_\theta \mathcal{L}(Z_i,\theta_0) = O_p(1)$

It remains to put the pieces together. We multiply left and right by $\sqrt{n}$ *to obtain:*

$$-\frac{1}{\sqrt{n}}\nabla_\theta \mathcal{L}(Z_n,\theta_0) = O_p(\frac{1}{n}) + (\hat{\theta}_n - \hat{\theta}_{n-1})\sqrt{n}(H_{\theta_0} + o_p(1)) + o_p(\frac{1}{n}) + \underbrace{\frac{\sqrt{n}}{2}\left[O_p(1)o_p(\frac{1}{n}) - O_p(1)o_p(\frac{1}{n})\right]}_{o_p(\frac{1}{\sqrt{n}})}. \quad (4.5)$$

Hence,

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

thus,

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

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

Finally,

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

Proof 4 (Proof for proposition 3) *Let us show that* $\mathcal{I}_{up,loss}(z,z_{test}) = -\nabla_\theta \mathcal{L}(z_{test},\hat{\theta}_n)H_{\hat{\theta}_n}^{-1}\nabla_\theta \mathcal{L}(z,\hat{\theta}_n)$

The chain rule is applied to $\frac{d\mathcal{L}(z_{test},\hat{\theta}_{n,\epsilon,-z})}{d\epsilon}$ *to obtain*

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

By evaluating in $\epsilon = 0$*, we obtain*

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

Finally,

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

Proof 5 (Proof for proposition 4) *Let $z_1 \dots z_n$ be a set of observations such that $z_i = (x_i, y_i)$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$. We denote by*

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

Recall that the loss function in the case of binary logistic regression is the log-likelihood (log-loss) regularized with a parameter ($\alpha > 0$). This means,

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

It is easy to show that

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

see page 120 of Hastie et al. (2009). Let us show that $H_{\hat{\theta}_n}$ is positive.

Let $a \in \mathbb{R}^p$ and show that $a^\intercal H_{\hat{\theta}_n} a \geq 0$

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

Let $i \in \{1, \ldots, n\}$; we set $b_i = x_i^\intercal a$, then

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

on the one hand,

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

on the other hand, $a^\intercal H_{\hat{\theta}_n} a = 0 \implies a = 0$. A sum of positive terms is null if and only if each term is null.

Remark 26 *It is possible to show that $H_{\hat{\theta}_n}$ is definite due to the regularization term.*

Proof 6 (Proof for proposition 5) *Let us show that $\mathcal{I}_{pert,params}(z, z_\delta) = \mathcal{I}_{up,params}(z) - \mathcal{I}_{up,params}(z_\delta)$*

Let $Z \sim F$, where F is the distribution function related to the data and $Z_{\epsilon,\delta} \sim F_{\epsilon,z_\delta,z} = F + \epsilon\delta_z - \epsilon\delta_{z_\delta}$ its contaminated version. We use the same methodology as for the proposition 1. Necessarily,

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

This can be rewritten as

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

Let us denote by $\hat{\theta}_{n,\epsilon,z_\delta,-z}$ the vector of optimal parameters for the contaminated distribution. We seek to calculate $\frac{d\hat{\theta}_{n,\epsilon,z_\delta,-z}}{d\epsilon}$. For the contaminated distribution the previous equation is rewritten

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

Deriving with respect to ϵ

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

with

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

By evaluating in $\epsilon = 0$*, then using the nullity condition of the gradient and simplifying we obtain*

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

This corresponds to

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

Proof 7 (Proof for proposition 6) *Closed formula for* $\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n$

Reusing the result of Proposition 1.2, it comes naturally that $\hat{\theta}_{n,z_\delta,-z} - \hat{\theta}_n = -\frac{1}{n}(\mathcal{I}_{up,params}(z_\delta) - \mathcal{I}_{up,params}(z)) + o_p(\frac{1}{n})$.

To show the second part of the result, we use a Taylor development at order 1. Let $f : \mathbb{R}^p \to \mathbb{R}$ *be a function that can be differentiated twice in a point* $a \in \mathbb{R}^p$*. Let us recall that Taylor's formula allows us to write*

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

By setting $f = \nabla_\theta \mathcal{L}$, $a = z$ and $h = \delta$, we obtain

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

Then,

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

By injecting this new approximation into the previous formula, we obtain

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

Finally,

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

Proof 8 (Proof for proposition 7) *Closed formula for $\mathcal{I}_{pert,loss}(z, z_{test})^\intercal$*

Assume that the explanatory variables are continuous, i.e., $x \in \mathcal{X} \in \mathbb{R}^p$ and $||\delta|| \to 0$. If $\mathcal{L}$ is differentiable with respect to x and θ then according to the approximation developed in the previous proof

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

By injecting this into Proposition 1.4, we obtain

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

By definition,

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

We apply the chain rule to $\nabla_\delta \mathcal{L}(z_{test}, \hat{\theta}_{n,z_\delta,-z})^\intercal\Big|_{\delta=0}$. *We obtain*

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

According to the approximations above we have

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

By evaluating in $\delta = 0$ *we obtain*

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

This leads to the conclusion that

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

4.8.3 Figures

Tab. 4.6: Table of estimated coefficients

Removed point	Influence at test point
1	4.77
2	-0.47
3	13.35
4	0.19
5	0.23

Bibliography

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45(1), pp. 5–32.

Charpentier, Arthur, ed. (Aug. 2014). *Computational Actuarial Science with R*. Chapman and Hall/CRC. DOI: 10.1201/b17230. URL: https://doi.org/10.1201%2Fb17230.

Chen, Tianqi and Carlos Guestrin (Aug. 2016). "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145%2F2939672.2939785.

Cook, R Dennis and Sanford Weisberg (1982). *Residuals and influence in regression*. New York: Chapman and Hall.

Doshi-Velez, Finale and Been Kim (2017). "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608*.

Friedman, Jerome H. (2001). "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29(5), pp. 1189–1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

Gourieroux, Christian and Alain Monfort (Oct. 1995). *Statistics and Econometric Models*. Cambridge University Press. DOI: 10.1017/cbo9780511751950. URL: https://doi.org/10.1017%2Fcbo9780511751950.

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi (Jan. 2019). "A Survey of Methods for Explaining Black Box Models". In: *ACM Computing Surveys* 51(5), pp. 1–42. DOI: 10.1145/3236009. URL: https://doi.org/10.1145%2F3236009.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Huber, Peter J. (Feb. 1981). *Robust Statistics*. John Wiley & Sons, Inc. DOI: 10.1002/0471725250. URL: https://doi.org/10.1002%2F0471725250.

Koh, Pang Wei and Percy Liang (2017). "Understanding black-box predictions via influence functions". In: *International Conference on Machine Learning*. PMLR, pp. 1885–1894.

Law, John, F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel (1986). "Robust Statistics-The Approach Based on Influence Functions." In: *The Statistician* 35(5), p. 565. DOI: 10.2307/2987975. URL: https://doi.org/10.2307%2F2987975.

Lundberg, Scott and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *arXiv preprint arXiv:1705.07874*.

Molnar, Christoph (2020). *Interpretable machine learning*. Lulu. com.

Ohlsson, Esbjörn and Björn Johansson (2010). *Non-Life Insurance Pricing with Generalized Linear Models*. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-10791-7. URL: https://doi.org/10.1007%2F978-3-642-10791-7.

Ribeiro, Marco, Sameer Singh, and Carlos Guestrin (2016). ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics. DOI: 10.18653/v1/n16-3020. URL: https://doi.org/10.18653%2Fv1%2Fn16-3020.

Team, R Core et al. (2013). “R: A language and environment for statistical computing”. In.

Verbelen, Roel, Katrien Antonio, and Gerda Claeskens (Apr. 2018). “Unravelling the predictive power of telematics data in car insurance pricing”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67(5), pp. 1275–1304. DOI: 10.1111/rssc.12283. URL: https://doi.org/10.1111%2Frssc.12283.

Wojnowicz, Mike, Ben Cruz, Xuan Zhao, Brian Wallace, Matt Wolff, Jay Luan, and Caleb Crable (Dec. 2016). ““Influence sketching”: Finding influential samples in large-scale regressions”. In: *2016 IEEE International Conference on Big Data (Big Data)*. IEEE. DOI: 10.1109/bigdata.2016.7841024. URL: https://doi.org/10.1109%2Fbigdata.2016.7841024.

5 Conclusions

The purpose of this concluding section is not to detail once again the contributions of this thesis. Indeed, they have already been described in the overall summary, the general introduction and in the conclusions of each chapter. In this general conclusion, we propose some research perspectives that are in line with the work we presented.

In chapter 2, we study a method named DefragTrees. This method is based on a statistical procedure that randomly initializes rectangles in the covariates' space before optimizing their shape to obtain a global surrogate. Unfortunately, this introduces randomness into the method. Furthermore, the objective function is not concave. Therefore, if we run the algorithm twice with different seeds, we may become stuck in local maxima. Hence, we may obtain two different trees and therefore different explanations. In practice, this is undesirable since we only want one explanation for a model. However, even if we can say that the explanations are different, i.e., that the trees generated by the method are different, we are unable to quantify the extent of the differences. We have not proposed a metric to quantify the difference between two trees, i.e., that between two of our explanations. However, such a metric may be of considerable interest since it would allow us to measure the stability of explanations provided by this method. This would not improve the method, but would be an important step in assessing the quality of explanations for methods of these kinds.

Our chapter 3 proposes an original method for solving a problem similar to that of chapter 2. Although this method has many advantages, it also has some drawbacks. For instance, the crossing of several partitions in a high-dimensional space can lead to a very large number of rectangles. This means that such a number of rectangles will have to be kept track of by the computer, which in practice will require considerable resources. For the moment, we have found a temporary solution. We only keep in the cross-partition the rectangles that contain points from the database. This solution allows our method to always work. However, we no longer have the knowledge of the tree ensemble over the whole space, and instead only know it on a subset of rectangles. Therefore, it seems interesting to consider other solutions to better manage complex cases. For example, we could divide the space into several subparts. Then, for each part of the space, we could generate a regression or classification tree that would faithfully represent the black-box. Finally, we could aggregate such faithful trees into a single one.

Moreover, as we show in chapter 3, our method can extract a local explanation that faithfully represents the black-box. Thanks to this we could compare our explanations with those produced by other local methods such as LIME or SHAP. The latter has drawn much attention from insurers

interested in implementing machine learning approaches. We could propose an approach that performs comparisons between these two methods to validate or reject the SHAP method on a black-box class for which we can now produce faithful explanations.

Finally, in our last chapter we show how to identify the influential points of a parametric black-box model. Additionally, we demonstrate how to extract local explanations for these points. However, contrary to chapters 2 and 3, this approach does not provide a global vision of the model. Consequently, it is rather difficult to draw general conclusions about the model from all of these explanations. One direction for improvement would be to generate the set of explanations for the influential points and then classify these explanations into K groups. Then, if we could determine the validity domain of explanations in each of these K groups, we could create a global model that would be valid across the space. This global model would be composed of valid explanations for each group and the validity domain of each group in the space.

List of Figures

List of Tables