Un cadre d'aide à l'exploitation des résultats de prédictions, à destination d'experts de domaine

Gabriel Ferrettini, Julien Aligon, Chantal Soulé-Dupuy

Université de Toulouse, UT1, IRIT (CNRS/UMR 5505), Toulouse, France
prenom.nom@irit.fr

RÉSUMÉ. *L'apprentissage automatique (ML) s'est révélé de plus en plus essentiel dans de nombreux domaines. Pourtant, de nombreux obstacles limitent encore son utilisation par des non-experts. Au premier rang de ceux ci se situe le manque de confiance dans les résultats obtenus et a inspiré plusieurs approches explicatives dans la littérature. Nous proposons ici un cadre pour exploiter cette capacité à expliquer les prédictions de ML de manière simple. Ceci a pour but de permettre aux outils ML existants de fournir une information plus interprétable aux utilisateurs ne maîtrisant pas encore l'apprentissage automatique. Ceci est effectué en fournissant à l'utilisateur une explication détaillée de l'influence des attributs pour chaque instance prédite, en relation avec le modèle d'apprentissage automatique. Nous montrerons également en quoi cette explication aide les utilisateurs non-experts à effectuer certaines tâches d'analyse complexes, telles que la sélection de modèles et l'ingénierie de fonctionnalités, et fournit une assistance pour exploiter efficacement les résultats d'un modèle prédictif.*

ABSTRACT. *Machine learning (ML) has proven increasingly essential in many fields. Yet, a lot obstacles still hinder its use by non-experts. The lack of trust in the results obtained is foremost among them, and has inspired several explanatory approaches in the litterature. We propose here a framework to build upon this ability to explain ML predictions in a simple way. This aims to allow already existing ML tools to provide a more interpretable information to users not already knowledgeable in machine learning. This is performed by providing the user a detailed explanation of the attributes influence for each single predicted instance, related to the machine learning model . We will also show how this explanation helps non-expert users to perform some complex analysis tasks, such as model selection and feature engineering, and provides assistance in exploiting efficiently the results of a predictive model.*

MOTS-CLÉS : *apprentissage automatique, explication de prédictions*

KEYWORDS: *machine learning, prediction explanation*

1. Introduction

Dans la plupart des cas, l'analyse de données suppose des compétences très spécifiques dans la réalisation et l'utilisation de modèles. Par exemple, concevoir un modèle prédictif (modèle finalement très commun et populaire) requiert une certaine expertise dans le domaine de la fouille de données. Ainsi, ces tâches d'analyse sont particulièrement difficiles à appréhender pour des utilisateurs experts de domaine (i.e. ayant une connaissance profonde des données à analyser, sans pour autant avoir des compétences en fouille de données). Un autre problème, pour les utilisateurs non experts, porte sur l'exploitation des résultats fournis par les analyses. En effet, ces résultats n'offrent généralement aucune indication sur la manière dont ils ont été produits, ce qui peut fortement impacter la confiance de ces utilisateurs sur la pertinence des analyses. Ainsi, dans le but de faire confiance aux résultats produits (par des méthodes qu'il ne connaît pas nécessairement) et les appliquer, un utilisateur expert du domaine aura tendance à les rattacher, d'une manière ou d'une autre, à sa propre connaissance de ce domaine.

L'ambition de nos recherches est d'aider les utilisateurs experts de domaine à (re)trouver une motivation pour s'impliquer dans des opérations d'analyse de données en donnant un sens aux processus utilisés. Notamment, notre objectif est de s'appuyer autant que possible sur leurs domaines d'expertise, tout en limitant les connaissance en analyse de données. Plus précisément, nous concentrons nos travaux sur des tâches de classification. Le but est de permettre à un utilisateur final d'effectuer des opérations de sélection de modèle prédictif et de spécifier les caractéristiques du modèle à l'aide des connaissances de domaine (nommée $feature\ engineering$) qui sont des tâches d'analyse connues pour être complexes. L'idée principale est que l'utilisateur n'ait qu'à exercer son expertise critique sur les classifications proposées. Afin de comprendre "comment" ou "pourquoi" une prédiction particulière est faite, nous proposons à l'utilisateur plusieurs types d'explications du comportement du modèle sur ces instances. Notre objectif final est également d'aider l'utilisateur à exploiter le modèle prédictif qu'il a produit. A cette fin, nous proposons donc une méthode de type $sandbox$ pour l'exploitation de modèle, permettant d'étudier le comportement de celui-ci sur de nouvelles instances simulées.

Les contributions, présentées dans ce papier, sont composées de:

– Un cadre d'aide à l'exploitation des résultats de prédictions à destination d'experts de domaine, en s'appuyant sur les travaux de (Strumbelj, Kononenko, 2010) et (Ribeiro *et al*., 2016), .

– Une proposition pour une nouvelle méthode d'explication de classification, basée sur (Strumbelj, Kononenko, 2010).

Ce papier est organisé de la manière suivante. La section 2 présente notre positionnement par rapport à l'état de l'art, portant sur l'aide à l'analyse de données et les explications de prédictions. Par la suite, les sections 3 et 4 développent notre cadre général pour l'aide à l'exploitation de prédictions, en introduisant les concepts de *Dataset*, *Workflow* et *Explication de prédictions*, ainsi qu'un exemple illustrant notre

démarche. Enfin, la section 5 discute des futures expériences à réaliser, permettant de vérifier nos hypothèses énoncées précédemment, et la section 6 conclut ce papier par plusieurs perspectives de travail.

2. Positionnement

Les systèmes existants et les outils pour l'apprentissage automatique et l'analyse de données se concentrent essentiellement sur la mise à disposition et l'explication des *méthodes* et des $algorithmes$. Cette approche est particulièrement utile pour des utilisateurs experts en fouille, mais requiert encore une connaissance avancée en analyse de données. En effet, la plupart des plateformes bien connues en analyse de données, comme Weka (Hall *et al*., 2009) ou Knime (Berthold *et al*., 2007) fournissent des descriptions détaillées des méthodes et algorithmes qu'ils comportent, illustrés souvent d'exemples. Malheureusement, ces descriptions détaillées sont un substitut très limité à la compréhension des informations en analyses de données.

Quelques plateformes, comme par exemple RapidMiner (Hofmann, Klinkenberg, 2013), Orange (Demšar *et al*., 2013), ont tout de même consacré une attention particulière à la présentation et à l'explication de *résultats* d'analyse à l'utilisateur. En fournissant des interfaces de visualisation bien conçues, ces platesformes aident l'utilisateur à comprendre les résultats obtenus, ce qui constitue un premier pas vers une utilisation compréhensible des modèles. Toutefois, ces outils ne peuvent toujours pas expliquer comment les résultats de fouille ont été obtenus, ce qui reste un facteur dissuasif et important pour des utilisateurs de domaine, où de mauvaises décisions peuvent avoir des conséquences graves. Aider ces utilisateurs à comprendre $pourquoi$ une prédiction particulière est faite, de sorte à pouvoir vérifier son raisonnement, pourrait renforcer leur confiance dans un modèle, ou au contraire leur donner une raison valable de le rejeter. Cette intution montre bien la nécessité d'expliquer les prédictions.

Plus récemment, l'outil de Google *What if*[1], principalement basé sur le travail de (Wachter *et al*., 2017), propose de nombreux outils d'exploration pour l'apprentissage automatique. Ceux-ci aident un utilisateur à comprendre et à exploiter les modèles de manière intuitive. Cela se fait principalement en permettant à l'utilisateur d'explorer l'espace de prédictions, à partir d'un modèle entrainé, et en affichant différentes métriques d'une manière facilement interprétable.

Notre travail vise justement à prendre en charge de telles fonctionnalités avec de nouvelles informations plus proches des connaissances du domaine de l'utilisateur : nous fournissons à l'utilisateur une mesure simple et interprétable de l'influence de chaque attribut (d'un ensemble de données) sur une prédiction. Grâce à cette méthode, nous souhaitons l'aider à comprendre le modèle d'apprentissage automatique, non pas de manière globale (existant dans beaucoup d'outils), mais par l'explication des

1. https://pair-code.github.io/what-if-tool/

éléments composant les prédictions (les instances). Cette méthode permettra de mieux appréhender un modèle donné.

L'une des contributions les plus significatives s'orientant dans cette voie peut être trouvé dans (Strumbelj, Kononenko, 2010). Dans ce papier, les auteurs décident de prendre en compte les interactions entre chaque attribut d'un ensemble de données afin d'approcher d'avantage leur influence réelle. Cet article mène par la suite au travail de (Lundberg, Lee, 2017), lequel a théorisé les méthodes d'explications dites "additives", dont la méthode proposée par (Strumbelj, Kononenko, 2010) fait partie. Cet article a notamment mis en lumière plusieures propriétés intéressantes de ces méthodes, qui en font un objet théorique très utile.

Les propriétés d'explication de prédictions ont été principalement étudiées par (Ribeiro *et al*., 2016). D'après ce papier, l'intérêt d'expliquer des modèles à l'utilisateur est triple :

– Premièrement, cela peut être perçu comme un moyen de comprendre le fonctionnement d'un modèle en observant comment il se comporte à divers endroits de l'espace des instances.

– Deuxièmement, cela peut aider un utilisateur non expert à juger de la qualité d'une prédiction et même à identifier la cause des problèmes dans sa classification. Les corriger conduirait alors l'utilisateur à effectuer certaines opérations de *feature engineering*.

– Troisièmement, cela permet à l'utilisateur de décider du modèle lui semblant préférable à un autre, même s'il n'a pas toujours connaissance des principes sous-jacents de chacun d'eux. Il suffit d'expliquer à l'utilisateur le comportement de mêmes instances, classées selon différents modèles.

La capacité d'expliquer une prédiction de n'importe quel modèle semble donc être un élément capital pour permettre à un public plus large, non expert, d'accéder aux modèles d'apprentissage automatique et de les utiliser. Ce besoin nous a amené à considérer les différents systèmes d'explication développés dans la littérature comme ayant un intérêt majeur pour la recherche d'une certaine autonomie des experts de domaine effectuant des activités d'analyse de données. Ainsi, le point clé de ce travail est que chaque choix significatif, effectué par l'utilisateur au cours d'une tâche de fouille de données, doit s'appuyer le plus possible sur la connaissance approfondie qu'il possède de son propre domaine. La section suivante donne un exemple concret de la manière dont cet objectif peut être atteint.

3. Définitions préliminaires et Méthodes d'explication

Dans cette section, nous définissons les éléments important de notre cadre que sont le Dataset, Workflow et l'Explication de Prédictions.

3.1. Dataset et Workflow

Un Dataset est défini comme un ensemble d'instances décrites selon des attributs. Soit $A = \{a_1, ..., a_n\}$, les attributs d'un dataset, une instance x est un vecteur de n valeurs d'attributs de A.

Un Workflow, dans la forme la plus générale, consiste en un graphe orienté d'opérations d'analyse de données (Serban *et al.*, 2013). Il peut inclure la plupart des étapes d'analyse de données, tels que le prétraitrement (nettoyage des données, normalisatin, etc.), la construction de modèles, la recherche de motifs, et même l'optimisation de paramètres pour d'autres opérations de fouille. Concernant les méthodes d'explication, nous considérons, dans ce papier, seulement celles applicables aux classifications supervisées. Ainsi nous ne considèrerons que des workflows résultant de ces types de modèles.

3.2. Explication de Prédictions

Soit un dataset D et un ensemble de n attributs $A = \{a_1, ..., a_n\}, \forall i \in [1..n], a_i \subset \mathbb{R}$. Chaque instance $x \in D$ est définie par les valeurs de chacun de ses attributs : $x = \{x_1, ..., x_n\}, \forall i \in [1..n], x_i \in a_i$. Nous voulons expliquer un modèle prédictif, basé sur la fonction $f : A \rightarrow [0, 1]$, dont le résultat est un score de confiance d'une classification d'une instance x pour la classe C, prédite par le modèle.

Une des premières définitions de l'explication de classification a été proposée dans (Štrumbelj, Kononenko, 2008). Selon leur méthode, l'influence d'un attribut a_i sur la classification d'une instance donnée peut être définie comme la différence entre la prédiction du classifieur (avec a_i) et sa prédiction sans la connaissance de l'attribut a_i. Ainsi, étant donné un ensemble d'instances décrit selon des attributs de A, l'influence de l'attribut a_i sur la classification d'une instance x par la fonction de confiance du classifieur f sur la classe C peut être représentée comme:

$$inf^C_{f,a_i}(x) = f(x) - f(x \backslash a_i) \tag{1}$$

Où $f(x \backslash x_i)$ représente la probabilité, selon le modèle étudié que l'instance x fasse partie de la classe c sans connaître l'attribut a_i.

Afin de simuler cette absence d'attribut a_i, les auteurs de (Štrumbelj, Kononenko, 2008) théorisent trois approches possibles parmi lesquelles nous avons sélectionné les deux plus génériques pour nos tests:

– Réentrainement du classifieur sur le dataset après suppression de l'attribut a_i.

– Utilisation d'une solution approchante à l'aide d'une espérance mathématique selon l'équation 2, comme proposé dans (Robnik-Šikonja, Kononenko, 2008).

$$f(x \backslash a_i) = \sum_{y \in val(a_i)} p(a_i = y) f(x \leftarrow a_i = y) \tag{2}$$

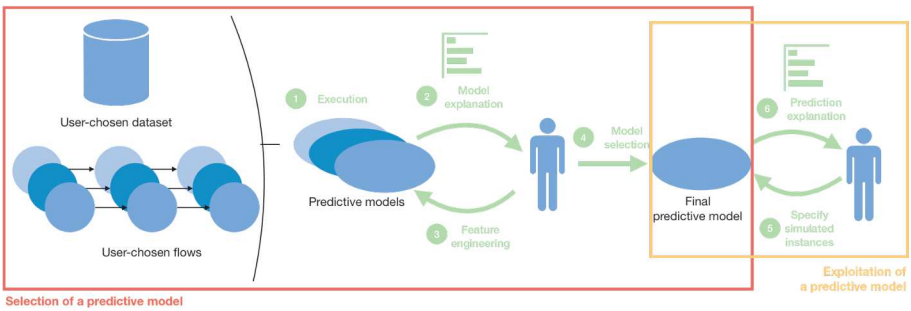

FIGURE *1. Construction et utilisation d'un modèle prédictif*

L'équation 2 consiste en une somme des distributions de probabilités du classifieur f pour une instance x, avec l'attribut a_i prenant toutes ses valeurs possibles, pondérées par la probabilité que l'attribut prenne cette valeur dans le jeu de données d'origine. Ces explications ont de nombreuses applications possibles pour aider les utilisateurs non experts à comprendre et à travailler dans un cadre d'apprentissage automatique. Nous allons démontrer un certain nombre de ces applications dans l'exemple suivant.

4. Exemple illustrant notre démarche

Notre exemple est séparé en deux cas d'utilisation. Dans la section 4.1, nous montrons comment un utilisateur expert de domaine peut être guidé tout au long du processus complexe de construction d'un modèle prédictif, tandis que la section 4.2 illustre comment des explications peuvent apporter de nouvelles connaissances lors de l'exploitation d'un modèle prédictif. Ces deux processus sont représentés dans la Figure 1 et sont basés sur les fonctionnalités les plus courantes de la litterature décrites Section 2.

4.1. Construction d'un nouveau modèle

Le Docteur Smith, un scientifique travaillant dans le domaine de la botanique, a rassemblé un dataset conséquent sur différentes fleurs qu'il a classées par espèce. Son objectif est de comprendre les principales caractéristiques de ces espèces afin de les reconnaître plus facilement par la suite et donc de faciliter son travail dans le futur. Le cadre que nous proposons devrait pouvoir l'aider dans sa tâche de classification.

1. Exécution - Le système propose un ensemble de workflows disponibles et capables de produire des modèles prédictifs. Smith ne connaissant pas grand-chose en apprentissage automatique, il fournit son jeu de données et sélectionne cinq workflows différents à l'aide de leurs descriptions respectives. Ces dernières le renseignent sur les traitements employés pour construire le modèle (e.g. prétraitement des données),

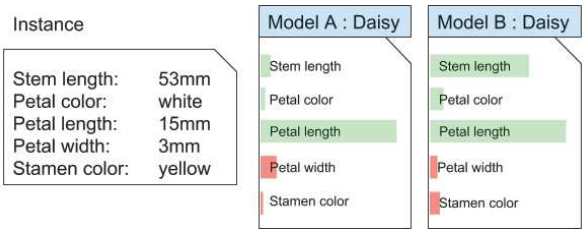

FIGURE 2. *Explications de prédictions*

ainsi que le type même de modèle (e.g. arbre de décision, Bayes naïf) sous forme textuelle. Le système exécute ensuite les cinq workflows sur le jeu de données de Smith, générant cinq modèles prédictifs différents.

2. Explication de modèle - À l'aide de ces modèles, le système peut générer la classification d'une instance donnée du dataset et fournir l'explication afférente à chaque modèle. Ces explications prennent la forme d'influences d'attributs. Par exemple, dans la figure 2, on informe le Docteur Smith qu'une fleur particulière est classée comme espèce S par les modèles A et B, mais A a pris cette décision compte tenu principalement de la longueur des pétales de la fleur, tandis que B a également pris en compte la longueur de la tige. Smith, considérant que la longueur de la tige de cette fleur particulière n'est pas réellement caractéristique de son espèce, il sera donc plus enclin à faire confiance au modèle A. Une manière d'expliquer simplement un modèle est de fournir un ensemble d'instances (les plus diverses possibles) afin de donner un bon aperçu du comportement de chaque modèle. Un algorithme est décrit dans ce sens par (Ribeiro *et al.*, 2016).

3. Feature engineering - A l'aide de ces explications, Smith se rend compte que l'utilisation d'un attribut, $ID\ Flower$, dans son dataset était une erreur : chaque modèle le considère comme très important pour sa prédiction (chaque modèle apprend donc que la fleur avec un identifiant idX appartient à une espèce y). Il est fort probable que l'identifiant de la plupart des fleurs est corrélé à son espèce dans le dataset. A l'aide de l'outil, il peut sélectionner cet attribut "ID Flower" et demander au système de le supprimer. Le système passe ensuite à l'étape (2) une seconde fois. Smith peut ensuite examiner les nouvelles influences générées et décider s'il souhaite exclure un autre attribut.

4. Sélection de modèle - Satisfait de ses modifications, Smith fait ensuite attention aux différences de raisonnement entre les modèles et se rend compte que, si l'un était certes plus précis dans ses classifications, un autre était plus "logique" dans l'utilisation de chaque attribut. En effet, ce dernier modèle donne plus d'importance à la longueur de la tige de fleur ($stem\ lenght$), quand cela est pertinent, et met de côté plus facilement les attributs dont Smith sait qu'ils ne sont pas si importants. Cette vérification de la pertinence de chaque modèle, à l'aide des connaissances du domaine, montre comment Smith peut décider du modèle à utiliser dans la pratique tout en se

fiant uniquement à ses propres connaissances. Le système génère ensuite le workflow correspondant au modèle choisi sur le dataset et le stocke pour une utilisation ultérieure. Ce workflow est choisi parmi ceux déjà réalisés par le passé. Un système de recommandation par filtrage collaboratif est alors nécessaire, comme par exemple, celui proposé dans (Raynaut, 2018).

4.2. Explorer et exploiter un modèle entrainé

Ce deuxième cas d'utilisation reprend l'exemple du docteur Smith après quelques mois. Après avoir étudié et répertorié les fleurs dans une nouvelle région du monde, il souhaite comparer sa population de fleurs à celle qu'il a étudiée dans le premier cas d'utilisation. Le modèle prédictif qu'il a produit à ce moment-là peut maintenant être très utile, lui confiant ainsi la tâche coûteuse d'identifier l'espèce de chaque fleur. Bien qu'il s'agisse de l'utilisation la plus courante d'un modèle prédictif, notre objectif ici est de montrer comment notre cadre permet d'aller encore plus loin. En effet, le but de ce cas d'utilisation est de permettre d'explorer et de présenter les caractéristiques apprises par le modèle pendant son entrainement, afin de trouver de nouveaux liens et corrélations possibles, non identifiés auparavant.

1. Après avoir collecté sur le terrain des fleurs dont les longueur et largeur de pétales sont inhabituels, le docteur Smith souhaite étudier l'importance que le modèle accordera à ces combinaisons d'attributs. L'idée est de pouvoir créer une instance "simulée", donc non présente initialement dans le dataset, afin de tester le comportement du modèle. Smith peut alors affecter aux attributs de longueur et de largeur de pétale une de ces combinaisons inhabituelles qu'il a collectées et demander au système de calculer la prédiction et l'explication.

2. Le système classe l'instance simulée avec le modèle choisi lors du premier cas d'utilisation et génère l'explication de cette classification en indiquant l'influence de chaque attribut que le Dr. Smith souhaite étudier. Avec cette nouvelle information, il peut itérativement concevoir une nouvelle instance simulée, en repartant de l'étape (5), et étudier l'influence des attributs vis à vis de ces nouvelles valeurs. Cette phase peut ainsi aider à l'identification de nouveaux points saillants pour ses recherches.

Le but principal de notre cadre est là : aider à extraire de manière interactive de nouvelles informations, à partir de modèles prédictifs standards. Notre cadre peut ainsi être vu comme un catalyseur pour l'analyse de différents types de données.

5. Comparaison des différentes méthodes d'explication

5.1. Premières expérimentations

5.1.1. Raisonnement préliminaire

Notre but est de comparer et évaluer les deux méthodes d'explication définies Section 3.2. En particulier, l'approche statistique a été préférée dans la littérature (eg.

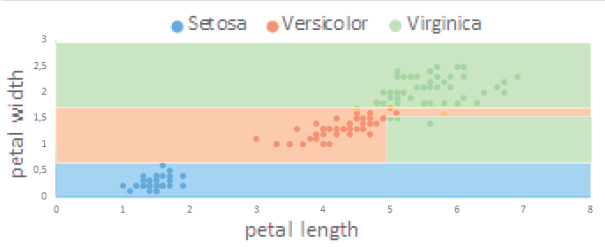

FIGURE *3. Répartition des trois différentes classes par longueurs et largeurs de pétales, avec leur généralisation correspondante à l'arbre de décision*

(Štrumbelj, Kononenko, 2008) and (Strumbelj, Kononenko, 2010)), en raison du coût de calcul potentiellement très élevé de la méthode par réentrainement $(\bigcirc(l * n))$, avec n le nombre d'attributs et l la complexité du modèle d'apprentissage).

5.1.2. Première expérience : exécution

Pour une expérience préliminaire, et afin de faciliter les interprétations, nous appliquons les deux méthodes à un simple arbre de décision ID3, (Quinlan, 1986), formé à partir du dataset *Iris* de Fisher [2]. Comme un arbre de décision est un modèle naturellement interprétable et que le jeu de données *Iris* est bien étudié dans la littérature, il est facile de comparer et d'interpréter les explications générées à partir de modèles prédictifs et de détecter les problèmes éventuels causés par les deux méthodes.

Nous utilisons Weka (Hall *et al.*, 2009) et OpenML (Vanschoren *et al.*, 2014) pour la gestion des données et la création des modèles tout en garantissant la reproductibilité de toutes les expériences. Afin d'estimer la fiabilité des deux modèles, nous appliquons une validation croisée (*5-fold cross validation*) sur l'ensemble des données. Les explications sont générées sur l'ensemble de validation à chaque itération de la validation croisée, pour la méthode statistique et par réentrainement. Nous générons ainsi les explications des prédictions du classifieur d'arbres *J48* de Weka pour l'ensemble des données *Iris*. Nous pouvons ensuite comparer ces explications à l'arbre de décision et obtenir une idée générale de l'exactitude des explications.

5.1.3. Première expérience : résultat et interprétation

Chaque instance du dataset *Iris* est composé de quatre attributs: longueur de pétale (*petal length*), largeur de pétale (*petal width*), longueur de sépale (*sepal length*) et largeur de sépale (*sepal width*). Chacune des instances peut faire partie de l'une de ces trois classes : *Iris Setosa*, *Versicolor* ou *Virginica*.

2. Iris, Fisher : https://en.wikipedia.org/wiki/Iris_flower_data_set

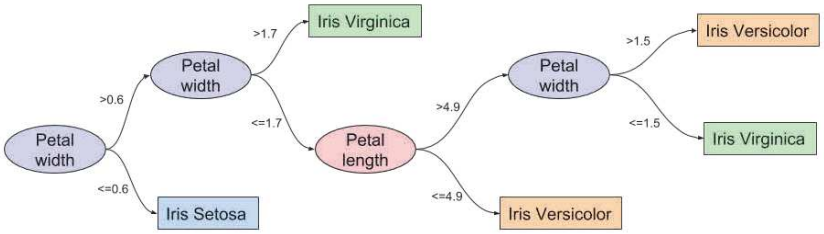

FIGURE *4. Un arbre de décision entrainé sur Iris*

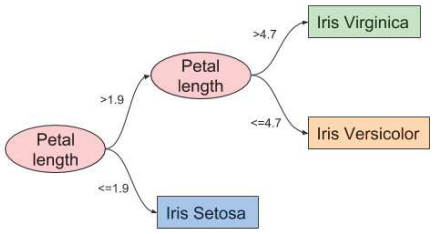

FIGURE *5. Identique à Figure 4 sans l'attribut largeur de pétale (petal width)*

Un simple coup d'œil à l'arbre de décision présenté dans la Figure 4 indique que l'influence doit être nulle pour les attributs longueur et largeur des sépales, car ils ne sont pas du tout utilisés par l'arbre. De plus, les instances de la classe *Setosa* ne devraient être influencées que par la largeur des pétales, car c'est le seul attribut utilisé pour les classer. Pour *Iris Virginica* et *Versicolor*, on peut s'attendre à une influence importante de la largeur des pétales et à une influence plus faible, mais néanmoins significative, de la longueur des pétales, car ces deux attributs sont utilisés (la largeur des pétales demeurant prédominante dans la classification des instances). Nous pouvons maintenant comparer ces suppositions aux résultats du tableau 6.

	Training method				Statistical method			
	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
Setosa	0	0	0	0	0	0	0.005	0.995
Versicolor	0	0	0.785	0.135	0	0	0.269	0.731
Virginica	0	0	-0.0283	0.615	0	0	0.171	0.824
Average	0	0	0.253	0.256	0	0	0.148	0.850

FIGURE *6. Influence moyenne des différents attributs, selon chaque méthode d'explication, pour chaque classe d'instance*

Nous pouvons facilement voir que la méthode par réentrainement ne se comporte pas comme prévu pour les instances de *Setosa*, car tous les attributs reçoivent une influence égale à 0. Cela peut être compréhensible en regardant l'arbre réentrainé sans l'attribut de largeur de pétale dans la figure 5, et la représentation du concept appris par l'arbre de la figure 3. Nous pouvons voir qu'en supprimant l'un des attributs parmi longueur des pétales et largeur des pétales, il reste possible de séparer linéairement la classe *Setosa* des autres en n'utilisant que la longueur de pétale, tout en maintenant une confiance de 100% dans la classification. Ainsi, chaque attribut est considéré comme sans importance par la méthode d'apprentissage lors de la classification des instances de *Setosa*. Cela implique que pour chaque dataset dans lequel deux attributs portent des informations très similaires, la méthode par réentrainement ne pourra pas générer d'explication satisfaisante. Afin de corriger cette aberration et de générer une explication plus précise, nous devons prendre en compte plus que la simple utilité des attributs pris individuellement.

Compte tenu des résultats pour les autres classes, nous ne pouvons pas encore conclure sur les différences entre les influences générées par les deux méthodes, mais nous pouvons voir que le comportement de la méthode statistique semble être plus proche de celui attendu. La simplicité du dataset $Iris$ avantage sans doute la méthode statistique car l'approximation de la répartition d'un petit ensemble d'attributs sera bien entendu plus proche de la situation réelle.

Tout du moins, et au vu de ces résultats préliminaires, il semble donc qu'il y ait une nécessité de considérer non pas les attributs comme des entités indépendantes, mais comme au moins partiellement interdépendantes.

5.2. Intuitions pour de futures expérimentations

5.2.1. Construction de nouvelles méthodes d'explication

Dans le but de répondre aux problèmes d'interaction entre attributs, discuté dans la section précédente, nous proposons de nous inspirer du travail de (Strumbelj, Kononenko, 2010). Nous sommes ici dans un cadre qui se rapproche de la situation d'un jeu dit "à coalitions", où chaque groupes d'attributs peux avoir une influence sur la prédiction du modèle. Nous ne pouvons donc considérer uniquement les attributs seuls mais bien toutes les coalitions possibles d'attributs. L'influence d'un attribut devra se mesurer en fonction de son importance dans chaque coalition. Nous pouvons alors nous rapporter aux jeux de coalition tels que définis par Shapley dans (SHAPLEY, 1953) : Un jeu de coalition de N joueurs est défini comme une fonction de mapping de sous-ensembles de joueurs de gains $g : 2^N \mapsto \mathbb{R}$. Le parallèle peut facilement être établi avec notre situation, où nous souhaitons évaluer l'influence d'un attribut donné *dans toutes les coalitions possibles d'attributs*. Nous examinerions alors non seulement l'influence de l'attribut, mais également son utilisation dans tous les sous ensembles d'attributs. Nous définissons donc l'*influence complète* d'un attribut $a_i \in A$ sur la classification d'une instance x (les notations restent les mêmes que dans 3.2):

$$\mathcal{I}_{a_i}^{C}(x) = \sum_{A' \subseteq A \setminus a_i} shap(A') * (inf_{f,(A' \cup a_i)}^{C}(x) - inf_{f,A'}^{C}(x))) \tag{3}$$

Avec $p(A')$ une fonction de pénalité dépendant de la taille du sous-ensemble A'. En effet, si un attribut change beaucoup le résultat d'un classifieur qui dépend déjà de beaucoup d'attributs, il peut être considéré comme très influent par rapport aux autres. À l'inverse, un attribut qui modifie le résultat d'un classifieur alors que ce classifieur se base sur un petit nombre d'attributs, ne peut pas être considéré comme ayant une influence déterminante.

A cette fin, la valeur de Shapley (SHAPLEY, 1953) est un candidat prometteur, et définis la pénalité comme :

$$shap(A') = \frac{|A'|! * (|A| - |A'| - 1)!}{|A|!} \tag{4}$$

Cette *influence complète* d'un attribut prend désormais en compte son importance parmi toutes les configurations d'attributs possibles, ce qui est plus proche de l'intuition d'origine derrière l'influence des attributs. Cependant, calculer l'*influence complète* d'une seule instance est extrêmement coûteux, avec une complexité de $\bigcirc(2^n * l)$, avec n le nombre d'attributs et l la complexité du modèle à expliquer.

Il n'est donc pas pratique d'utiliser l'*influence complete*. Par conséquent, il devient nécessaire de rechercher un moyen plus efficace d'expliquer les prédictions. Bien que l'*influence complete* soit trop lourde en calculs, son intérêt en fait une excellente baseline de départ, comme le propose d'ailleurs (Strumbelj, Kononenko, 2010). Nous pouvons donc évaluer d'autres méthodes d'explication en étudiant leurs différences par rapport à l'*influence complète*.

5.2.2. Travaux en cours: Trouver de bons estimateurs de l'influence complète

Une approximation de l'*influence complète* pourrait fournir une méthode d'explication à la fois précise et pratique. Certaines ont été proposées dans (Strumbelj, Kononenko, 2010) et (Lundberg, Lee, 2017), bien que ces heuristiques reposent sur plusieurs aprioris : modèle de prédiction linéaire, indépendance des attributs. Cela ne peut être pris en compte dans notre cadre, car nous souhaitons travailler avec n'importe quel ensemble de données et processus. Cela nous conduit donc à rechercher de nouvelles heuristiques.

En particulier, l'évaluation d'un sous-ensemble de tous les sous-groupes d'attributs autorise des limites beaucoup plus pratiques en termes de complexité, tout en produisant des estimateurs à priori plus précis que la simple considération des attributs seuls (*influence linéaire*). Nous pouvons alors considérer une *influence k-complète* (*influence complète* de profondeur k) définie comme :

$$\mathcal{I}_{a_i}^{C_k}(x) = \sum_{A' \subseteq A \setminus a_i | A'| \geq |A| - k} p(A') * (inf(x_{A' \cup \{a_i\}}) - inf(x_{A'})) \tag{5}$$

D'ailleurs, nous pouvons noter que l'*influence linéaire* est alors identique à l'influence de l'*influence* 1-*complète*.

Une autre approche possible consiste à identifier les attributs ayant une relation entre eux, en utilisant un algorithme de regroupement d'attributs (comme dans (Henelius *et al*., 2014)).

Nous pouvons obtenir un groupe tel que $G = \{\{a_1, a_3\}, \{a_2, a_5, a_8\}, \{a_4\}...\}$. Avec de tels regroupements d'attributs, il devient possible de ne considérer que les relations entre les attributs d'un sous-groupe, sans avoir à prendre en compte toutes les combinaisons d'attributs possibles.

Nous obtenons alors une *influence coalitionnelle* d'un attribut $a_i \in g, g \in G$:

$$simple\mathcal{I}^C_{a_i}(x) = \sum_{g' \subseteq g \setminus a_i} shap(g') * (inf^C_{f,(g' \cup a_i)}(x) - inf^C_{f,g'}(x))) \qquad (6)$$

Étant donné que nous pouvons définir un cardinal maximum c pour nos sous-groupes, la complexité serait désormais, dans le pire des cas, $O(2^c * \frac{n}{c} * l) \approx O(n * l)\ avec\ l = m(n)$

Afin de déterminer s'il est possible de générer une approximation satisfaisante de l'influence d'un attribut avec la nouvelle *influence k-complète* et l'*influence coalitionnelle*, il faut évaluer le nombre des combinaisons d'attributs à prendre en compte avant d'être suffisamment proches de l'*influence complète* définie dans 3. De plus, nous devons évaluer si le résultat de l'*influence k-complète* produit bien de meilleures explications que l'*influence linéaire* et l'*influence coalitionnelle*, au vu de son coût de calcul plus élévé.

Pour nos expériences, nous devrons rassembler un ensemble de datasets comportant relativement peu d'attributs, afin de maintenir des limites réalisables en termes de complexité. Pour chaque dataset, l'*influence complète* sera calculée, ainsi que chaque approximation par l'*influence k-complète* et *influence coalitionnelle*. Enfin, l'influence calculée par l'*influence linéaire* sera également générée. Chaque méthode sera ensuite comparée à la méthode complète.

Ces résultats nous diront tout d'abord s'il y a un réel avantage à utiliser l'une des méthodes alternatives et dans quels cas une méthode est préférable aux autres.

6. Conclusion

Nous avons proposé dans cet article un cadre d'explication des modèles prédictifs visant à présenter l'utilisateur expert de domaine comme un élément actif du processus de construction de modèle.

Les différentes expériences que nous proposons sont une étape supplémentaire vers l'aide à l'analyse de données, en particulier l'explication de modèles en appren-

tissage automatique. Notre première tentative de comparaison des méthodes de réentrainement et des méthodes statistiques nous a conduit à identifier des défauts dans leurs conceptions intrinsèques. En prenant en compte les combinaisons d'attributs, nous pensons proposer une base de référence adéquate pour l'explication de prédictions et espérons obtenir de meilleures explications en termes de précision et de réduction des coûts. L'expérience en cours servira à comparer les différentes méthodes et ses résultats nous aideront à concentrer nos efforts sur les candidats les plus prometteurs. La prochaine étape serait ensuite de développer un outil complet sur cette baseline, dans le but de guider un utilisateur non expert tout au long de la construction et de l'exploitation d'un modèle d'apprentissage automatique.

Une perspective à plus long terme porte aussi sur la problématique d'évaluation des différentes méthodes d'explication de la littérarture. En effet, à notre connaissance, il n'existe pas de benchmark permettant d'évaluer objectivement différentes méthodes d'explication. Ce benchmark pourrait inclure, en plus des datasets, un ensemble d'indicateurs liés à la perception qu'en a l'expert de domaine (Kappa de Cohen ou Kappa de Fleiss par exemple, (Cohen, 1960), (Fleiss *et al*., 1971))

Bibliographie

Berthold M. R., Cebron N., Dill F., Gabriel T. R., Kötter T., Meinl T. *et al*. (2007). KNIME: The Konstanz Information Miner. In *Studies in classification, data analysis, and knowledge organization (gfkl 2007).* Springer.

Cohen J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, vol. 20, n° 1, p. 37-46. Consulté sur https://doi.org/10.1177/001316446002000104

Demšar J., Curk T., Erjavec A., Gorup Črt, Hočevar T., Milutinovič M. *et al*. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, vol. 14, p. 2349-2353. Consulté sur http://jmlr.org/papers/v14/demsar13a.html

Fleiss J. *et al*. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, vol. 76, n° 5, p. 378–382.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, vol. 11, n° 1, p. 10–18.

Henelius A., Puolamaki K., Boström H., Asker L., Papapetrou P. (2014). A peek into the black box : exploring classifiers by randomization. *Data mining and knowledge discovery*, vol. 28, n° 5-6, p. 1503–1529. (QC 20180119)

Hofmann M., Klinkenberg R. (2013). *Rapidminer: Data mining use cases and business analytics applications*. Chapman & Hall/CRC.

Lundberg S., Lee S.-I. (2017). A unified approach to interpreting model predictions. In *Nips.*

Quinlan J. (1986, 01 Mar). Induction of decision trees. *Machine Learning*, vol. 1, n° 1, p. 81–106. Consulté sur https://doi.org/10.1023/A:1022643204877

Raynaut W. (2018). *Perspectives de Méta-Analyse pour un Environnement d'aide à la Simulation et Prédiction*. Thèse de doctorat, Université de Toulouse, Toulouse, France. Consulté sur ftp://ftp.irit.fr/IRIT/SIG/2018_These_Raynaut.pdf

Ribeiro M. T., Singh S., Guestrin C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 1135–1144. New York, NY, USA, ACM. Consulté sur http://doi.acm.org/10.1145/2939672.2939778

Robnik-Šikonja M., Kononenko I. (2008, mai). Explaining classifications for individual instances. *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, n° 5, p. 589–600. Consulté sur http://dx.doi.org/10.1109/TKDE.2007.190734

Serban F., Vanschoren J., Kietz J.-U., Bernstein A. (2013, juillet). A survey of intelligent assistants for data analysis. *ACM Comput. Surv.*, vol. 45, n° 3, p. 31:1–31:35. Consulté sur http://doi.acm.org/10.1145/2480741.2480748

SHAPLEY L. S. (1953). A value for n-person games. *Contributions to the Theory of Games*, n° 28, p. 307-317. Consulté sur https://ci.nii.ac.jp/naid/10013542751/en/

Štrumbelj E., Kononenko I. (2008). Towards a model independent method for explaining classification for individual instances. In I.-Y. Song, J. Eder, T. M. Nguyen (Eds.), *Data warehousing and knowledge discovery*, p. 273–282. Berlin, Heidelberg, Springer Berlin Heidelberg.

Strumbelj E., Kononenko I. (2010, mars). An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, vol. 11, p. 1–18. Consulté sur http://dl.acm.org/citation.cfm?id=1756006.1756007

Vanschoren J., Rijn J. N. van, Bischl B., Torgo L. (2014, juin). Openml: Networked science in machine learning. *SIGKDD Explor. Newsl.*, vol. 15, n° 2, p. 49–60. Consulté sur http://doi.acm.org/10.1145/2641190.2641198

Wachter S., Mittelstadt B. D., Russell C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, vol. abs/1711.00399. Consulté sur http://arxiv.org/abs/1711.00399