Une interface XAI pour des modèles d'apprentissage automatique à base d'arbres

Gilles Audemard*, Sylvie Coste-Marquis*
Pierre Marquis*,** Mehdi Sabiri* Nicolas Szczepanski*

*Univ. Artois, CNRS, CRIL, Lens, France
**Institut Universitaire de France, France
nom@cril.fr

Résumé. Dans le cadre d'une tâche de prédiction en IA explicable, nous proposons un protocole pour régir les interactions entre un modèle d'apprentissage automatique à base d'arbres (le système d'IA) et son utilisateur U. Nous nous plaçons dans le cas où les connaissances de U sur la tâche de prédiction peuvent être représentées par un ensemble cohérent mais incomplet de règles de classement qui sont supposées fiables. Le protocole proposé a pour objectif d'aider U à décider quoi faire de chaque prédiction effectuée par l'IA : l'accepter ou la rejeter. Il vise également à améliorer la qualité des futures prédictions faites par l'IA en l'améliorant grâce à l'expertise de U, et, réciproquement, à compléter les connaissances de U en s'appuyant sur les prédictions faites par l'IA.

1 Introduction

Dans cet article, nous décrivons un protocole XAI pour régir les interactions entre un utilisateur U et un système AI d'intelligence artificielle à base d'apprentissage automatique (ML). Le but est d'améliorer la confiance que U possède en AI, de corriger les prédictions erronées de AI et d'enrichir les connaissances de U grâce aux interactions. Nous supposons que U détient des connaissances fiables mais incomplètes sur le domaine d'application ciblé par AI : il n'est pas attendu que la couverture de U soit complète, i.e. que U soit capable d'associer une prédiction à chaque instance possible (sinon, U n'aurait probablement pas besoin de l'aide d'AI). Nous supposons aussi que U est confiant dans les connaissances qu'il détient et estime qu'il est plus fiable que AI. De plus, U est supposé classer les instances de manière cohérente. Cela signifie que les raisons utilisées par U pour classer les instances (qui peuvent être modélisées de manière abstraite sous forme de règles de classement) ne sont pas contradictoires. Par ailleurs, pour être mis en œuvre, notre protocole demande qu'AI soit capable d'engendrer des explications locales et fidèles et de corriger des prédictions erronées. C'est le cas lorsqu'AI est un modèle ML basé sur des arbres. Nous pouvons utiliser dans ce cadre la bibliothèque open source PyXAI (`https://github.com/crillab/pyxai`) pour réaliser les tâches d'explication et de correction. Des expérimentations ont été réalisées et les résultats obtenus (présentés plus en détail dans (Audemard et al., 2024)) montrent que des bénéfices en terme de performance prédictive de AI et de couverture de U peuvent être obtenus en utilisant le protocole proposé.

2 Préliminaires formels

Classement et explications Nous supposons que les instances $\boldsymbol{x}$ prises en compte sont décrites à l'aide de paires attribut / valeur. $A = \{A_1, \ldots, A_k\}$ est l'ensemble des *attributs* utilisés. Chaque attribut de A est booléen, catégorique ou numérique et prend ses valeurs dans un *domaine* D_i. Les attributs de A ne sont pas nécessairement indépendants et une *théorie du domaine* Σ (représentée par un circuit propositionnel ou une formule propositionnelle) qui précise comment les attributs (et leurs valeurs) sont logiquement connectés peut être exploitée (Gorji et Rubin, 2022). Une *instance* $\boldsymbol{x}$ sur A est un k-uplet de $D_1 \times \ldots \times D_k$. $\boldsymbol{x} = (v_1, \ldots, v_k)$ est également vu logiquement comme un ensemble $t_{\boldsymbol{x}}$ de *caractéristiques* $\{(A_i = v_i) : i \in [k]\}$, interprété de manière conjonctive.

Dans le cas d'un classement à une seule étiquette, on considère un ensemble unique C d'étiquettes représentant des classes. Un *classeur* f sur A est alors une fonction qui associe chaque instance $\boldsymbol{x}$ de l'ensemble $\boldsymbol{X}$ de toutes les instances à une classe dans C. Il s'agit d'un *classeur binaire* lorsque $C = \{0, 1\}$. Pour un classeur binaire f, une instance $\boldsymbol{x} \in \boldsymbol{X}$ est dite *positive* lorsque $f(\boldsymbol{x}) = 1$ et elle est dite *négative* lorsque $f(\boldsymbol{x}) = 0$.

Lorsque le classeur f est un modèle ML à base d'arbres (un arbre de décision (Breiman et al., 1984; Quinlan, 1986), une forêt aléatoire (Breiman, 2001) ou un arbre boosté (Freund et Schapire, 1997)), f peut également être vu comme une fonction booléenne sur l'ensemble des conditions booléennes utilisées dans f. Autrement dit, nous pouvons supposer que f est un classeur sur un ensemble d'attributs booléens correspondant aux conditions booléennes utilisées dans f. Dans ce cas, les attributs booléens ne sont généralement pas indépendants. En effet, ils peuvent provenir des mêmes attributs numériques ou catégoriques utilisés au départ pour l'apprentissage du classeur. Par exemple, nous pouvons considérer un attribut booléen $x_1 = (\hat{a}ge > 21)$ lié à un attribut numérique $\hat{a}ge$, mais aussi un attribut booléen $x_2 = (\hat{a}ge > 18)$ qui est connecté à $\hat{a}ge$ et logiquement lié à x_1, car x_1 ne peut être vrai pendant que x_2 serait faux. Les caractéristiques correspondantes sont des *littéraux* (ici, x_1, x_2 et les littéraux complémentaires $\overline{x_1}$, $\overline{x_2}$). La théorie du domaine indiquant comment les caractéristiques sont liées pourrait être la formule $\Sigma = \overline{x_1} \vee x_2$ dans ce cas.

Étant donné un classeur f sur A et une instance $\boldsymbol{x} \in \boldsymbol{X}$, une *explication abductive* t pour $\boldsymbol{x}$ donné f (Ignatiev et al., 2019) est un ensemble de caractéristiques de $\boldsymbol{x}$, $t \subseteq t_{\boldsymbol{x}}$, interprété de manière conjonctive, tel que toute instance $\boldsymbol{x'} \in \boldsymbol{X}$ couverte par t (c'est-à-dire satisfaisant $t \subseteq t_{\boldsymbol{x'}}$) est telle que $f(\boldsymbol{x'}) = f(\boldsymbol{x})$. Chaque instance $\boldsymbol{x}$ possède une explication abductive donnée f, puisque $t = t_{\boldsymbol{x}}$ satisfait les conditions requises. Bien sûr, une telle explication triviale est généralement inutile, et lorsque cela est possible, les explications abductives qui ne coïncident pas avec $t_{\boldsymbol{x}}$ sont préférées. Souvent, les explications abductives minimales pour l'inclusion ensembliste (Ignatiev et al., 2020) (aussi appelées raisons suffisantes (Darwiche et Hirth, 2020) ou PI-explications (Shih et al., 2018)) et les explications abductives de taille minimale (Barceló et al., 2020; Audemard et al., 2022) sont ciblées.

Règles de classement Une *règle de classement* r est un couple $r = t \rightarrow c$ où t est une conjonction de caractéristiques sur A et c est un élément de C. t est la *condition* de r, et c est la *conclusion* de r. Une règle de classement $r = t \rightarrow c$ *classe* une instance $\boldsymbol{x}$ sur A comme c si et seulement si $\boldsymbol{x}$ satisfait t. Dans ce cas, nous notons $r(\boldsymbol{x}) = c$.

Étant donné une théorie du domaine Σ et deux règles de classement $r_1 = t_1 \rightarrow c_1$ et $r_2 = t_2 \rightarrow c_2$, nous disons que r_1 *spécialise* r_2 (ou, de manière équivalente, que r_2 *généralise* r_1) si et seulement si $c_1 = c_2$ et t_2 est une conséquence logique de $t_1 \wedge \Sigma$. Une règle $r_1 = t_1 \rightarrow c_1$ est *en conflit* avec un ensemble R de règles de classement étant donné une théorie du domaine Σ si et seulement s'il existe une règle $r_2 = t_2 \rightarrow c_2$ dans R telle que $c_1 \neq c_2$ et que $t_1 \wedge t_2 \wedge \Sigma$ est cohérent. Un ensemble R de règles de classement est dit :

— *cohérent* si et seulement si, pour chaque paire r_1, r_2 de règles de R telles que $r_1 = t_1 \rightarrow c_1$ et $r_2 = t_2 \rightarrow c_2$, si $c_1 \neq c_2$, alors $t_1 \wedge t_2 \wedge \Sigma$ est incohérent ;
— *complet* si et seulement si, pour chaque instance $\boldsymbol{x}$ sur A, il existe au moins une règle $r = t \rightarrow c$ telle que $\boldsymbol{x}$ satisfait t. Autrement dit, toute instance $\boldsymbol{x}$ sur A est classée par une règle de R ;
— *simplifié* si et seulement si, pour chaque paire r_1, r_2 de règles distinctes de R, r_1 ne spécialise pas r_2 étant donné Σ.

Pour une instance $\boldsymbol{x} \in \boldsymbol{X}$, lorsque toutes les règles de R qui classent $\boldsymbol{x}$ ont la même conclusion, R classe $\boldsymbol{x}$ sans ambiguïté, à condition qu'au moins une règle r de R classe $\boldsymbol{x}$. Dans ce cas, nous disons que $R(\boldsymbol{x})$ est défini et nous notons $R(\boldsymbol{x}) = r(\boldsymbol{x})$. Dans le cas contraire, nous disons que $R(\boldsymbol{x})$ est indéfini, noté $R(\boldsymbol{x}) = \perp$. Lorsque R est cohérent, complet et simplifié, chaque instance $\boldsymbol{x}$ sur A est classée par une règle unique r de R.

Correction de classeurs à base d'arbres par rectification La rectification est une approche pour la mise à jour des classeurs AI (Coste-Marquis et Marquis, 2021), qui peut être utilisée pour mettre en œuvre une opération de correction d'AI par les règles de R_U.

Par construction, dans le cas d'un classement à étiquette unique, la rectification d'un classeur AI par une règle de classement r_U conduit à un classeur qui classe chaque instance comme le fait AI, sauf pour les instances classées par r_U, qui sont classées par le classeur rectifié comme le demande r_U. De plus, lorsque AI est un modèle à base d'arbres, sa rectification par une règle de classement r_U peut être réalisée efficacement, i.e., en temps polynomial en la taille de l'entrée, AI et r_U (Coste-Marquis et Marquis, 2023).

3 Un protocole XAI pour les modèles à base d'arbres

Dans ce qui suit, nous supposons que U est représenté par un ensemble de règles R_U qui est cohérent et simplifié, mais non complet (sinon, l'utilisateur n'aurait pas besoin d'AI !). Les éléments de R_U peuvent être considérés comme des connaissances dans lesquelles l'utilisateur U a confiance. Le classeur AI (quel qu'il soit) peut, de son côté, toujours être associé à un ensemble équivalent R_{AI} de règles de classement qui est cohérent et complet. En particulier, $\{t \rightarrow AI(\boldsymbol{x}) \mid t$ est une explication abductive pour $\boldsymbol{x}$ étant donné $AI \mid \boldsymbol{x} \in \boldsymbol{X}\}$ est un tel ensemble de règles de classement. Cet ensemble exponentiel en taille par rapport au nombre d'attributs utilisés pour décrire les instances, n'a pas besoin d'être entièrement calculé.

Dans notre approche, des règles de classement r de R_{AI} sont calculées à la demande, i.e., chaque fois qu'une prédiction concernant une instance $\boldsymbol{x}$ est calculée. De telles règles r sont extraites d'explications fidèles, reflétant la façon dont AI procède pour classer les instances $\boldsymbol{x}$. Ainsi, si t est une explication abductive pour $\boldsymbol{x}$ étant donné AI, alors $r = t \rightarrow AI(\boldsymbol{x})$ est une règle de classement qui peut être déduite de AI : pour chaque instance $\boldsymbol{x}'$ satisfaisant t, il est garanti que $AI(\boldsymbol{x}') = AI(\boldsymbol{x})$. Les règles de R_U sont supposées plus fiables que celles de R_{AI}.

Supposons qu'une règle $r = t \rightarrow c$ de R_{AI} ait été engendrée à partir d'une explication abductive pour l'instance $\boldsymbol{x}$ considérée. Cette règle r classe $\boldsymbol{x}$ comme étant de classe c. Quatre

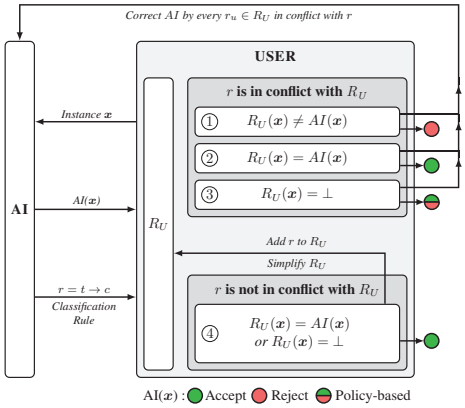

FIG. 1 – *Un protocole XAI permettant diverses interactions entre U et AI.*

cas distincts (1) à (4) méritent d'être examinés. La figure 1 illustre les quatre cas sur lesquels notre protocole XAI est fondé. Une interaction commence dès que l'utilisateur fournit une instance $\boldsymbol{x}$ au système AI et demande une prédiction $AI(\boldsymbol{x})$.

Cas (1). Supposons d'abord que $\boldsymbol{x}$ soit également classée par R_U (donc $R_U(\boldsymbol{x})$ est défini), de telle manière que $R_U(\boldsymbol{x}) \neq AI(\boldsymbol{x})$. Dans ce cas, il existe au moins une règle $r_U = t_1 \rightarrow c_1$ dans R_U telle que t_1 couvre $\boldsymbol{x}$ et $c_1 \neq c$. Par définition, puisque t couvre $\boldsymbol{x}$, r est en conflit avec r_U, et donc avec R_U. Dans ce cas, *la prédiction $AI(\boldsymbol{x})$ doit être rejetée, et AI doit être corrigé par r_U*. Bien sûr, il peut y avoir plusieurs règles r_U dans R_U qui classent $\boldsymbol{x}$. Dans ce cas, *AI doit être corrigé par chaque règle r_U*. En effet, r est en conflit avec chacune d'elles.

Par exemple, supposons que $R_U = \{(a \wedge b) \rightarrow 1, (b \wedge c) \rightarrow 1, \overline{b} \rightarrow 0\}$. On peut facilement vérifier que R_U est cohérent et simplifié, mais pas complet (par exemple, $(0, 1, 0)$ n'est pas classé par R_U). Supposons que AI soit équivalent à l'ensemble de règles de classement suivant : $\{\overline{a} \rightarrow 1, \overline{b} \rightarrow 1, (a \wedge b) \rightarrow 0\}$, qui est cohérent, complet et simplifié. Considérons l'instance $(1, 1, 1)$. Elle est classée par R_U comme une instance positive en utilisant la règle $r_U = (a \wedge b) \rightarrow 1$ (et aussi par la règle $(b \wedge c) \rightarrow 1$). En revanche, AI la classe comme une instance négative en utilisant la règle $r = (a \wedge b) \rightarrow 0$. Ainsi, la prédiction d'AI doit être rejetée, et AI doit être corrigé par les deux règles $r_U = (a \wedge b) \rightarrow 1$ et $(b \wedge c) \rightarrow 1$. On peut vérifier que $r = (a \wedge b) \rightarrow 0$ est en conflit avec $r_U = (a \wedge b) \rightarrow 1$, mais aussi avec $(b \wedge c) \rightarrow 1$.

Cas (2). Supposons maintenant que l'instance $\boldsymbol{x}$ soit classée par R_U (donc $R_U(\boldsymbol{x})$ est défini), de telle manière que $R_U(\boldsymbol{x}) = AI(\boldsymbol{x})$. Dans ce cas, r n'est pas en conflit avec les règles de R_U qui classent $\boldsymbol{x}$, puisque toutes ces règles ont nécessairement la même conclusion c. *La prédiction $AI(\boldsymbol{x})$ peut être acceptée puisqu'elle est conforme à la prédiction faite par U en utilisant des connaissances plus fiables sur la tâche de prédiction*. Cependant, il reste possible que r soit en conflit avec d'autres règles r_U de R_U. Si tel est le cas, AI doit être corrigé par chaque règle r_U en conflit. Cette correction permet d'anticiper et résoudre en amont

des incompatibilités entre les prédictions de R_U et de AI qui pourront survenir plus tard sur d'autres instances.

Par exemple, supposons que $R_U = \{(a \wedge b) \rightarrow 1, (b \wedge c) \rightarrow 1, \overline{b} \rightarrow 0\}$, comme dans l'exemple précédent. Supposons cette fois que AI soit équivalent à l'ensemble de règles de classement suivant : $\{c \rightarrow 1, \overline{c} \rightarrow 0\}$, qui est cohérent, complet et simplifié. Considérons à nouveau l'instance $(1, 1, 1)$. Elle est classée comme une instance positive par R_U et par AI. La prédiction faite par le classeur concorde avec les connaissances de l'utilisateur et peut donc être acceptée. Cependant, il s'avère que la règle de classement $r = c \rightarrow 1$ utilisée par AI pour classer $(1, 1, 1)$ est en conflit avec la règle de classement $r_U = \overline{b} \rightarrow 0$ de R_U. Il faut donc corriger AI par $r_U = \overline{b} \rightarrow 0$. Une fois cette correction effectuée, l'instance $(1, 0, 1)$ sera classée de la même manière (comme une instance négative) par R_U et AI. Corriger AI par r_U dès que l'instance $(1, 1, 1)$ est traitée empêche AI de produire une mauvaise prédiction qui serait réalisée si l'instance $(1, 0, 1)$ était considérée ensuite.

Cas (3). Supposons maintenant que l'instance $\boldsymbol{x}$ ne soit pas classée par R_U (i.e., $R_U(\boldsymbol{x})$ n'est pas défini). Dans cette situation, comme dans le cas (2), il peut cependant arriver que r soit en conflit avec certaines règles r_U de R_U. Supposons que ce soit le cas. Alors AI doit être corrigé par chaque règle r_U en conflit. Encore une fois, cette correction permet d'éviter dans le futur qu'AI et R_U soient en désaccord.

Par exemple, supposons que $R_U = \{(a \wedge b) \rightarrow 1, (b \wedge c) \rightarrow 1, \overline{b} \rightarrow 0\}$, qui est cohérent, simplifié, mais pas complet. Supposons que AI soit équivalent à l'ensemble de règles de classement suivant : $\{c \rightarrow 1, \overline{c} \rightarrow 0\}$, qui est cohérent, complet et simplifié. Considérons maintenant l'instance $(0, 1, 0)$. Cette instance n'est pas classée par R_U et elle est classée par AI comme une instance négative. Cependant, la règle de classement $r = \overline{c} \rightarrow 0$ utilisée par AI pour classer $(0, 1, 0)$ est en conflit avec la règle de classement $r_U = (a \wedge b) \rightarrow 1$ de R_U. Ainsi, AI devrait être corrigé par $r_U = (a \wedge b) \rightarrow 1$. Si cette correction est effectuée, l'instance $(1, 1, 0)$ sera classée de la même manière (comme une instance positive) par R_U et AI.

À propos de la décision à prendre concernant la prédiction $AI(\boldsymbol{x})$ dans ce cas, plusieurs politiques peuvent être envisagées. Selon une *politique aventureuse*, *la prédiction* $AI(\boldsymbol{x})$ *peut être acceptée* puisque aucune règle de R_U n'indique que la prédiction doit être rejetée. En revanche, selon une *politique prudente*, *la prédiction* $AI(\boldsymbol{x})$ *doit être rejetée* car la règle r de AI utilisée pour prendre la décision n'est pas entièrement correcte. En effet, elle classe certaines instances d'une manière différente de R_U (donc r doit être spécialisée). D'autres politiques pourraient être facilement définies en prenant en compte le nombre et la spécificité des règles de R_U avec lesquelles r est en conflit.

Cas (4). Le cas restant couvre les situations où $R_U(\boldsymbol{x}) = AI(\boldsymbol{x})$ ou $R_U(\boldsymbol{x}) = \perp$, et où il n'y a aucun conflit entre r et R_U. Dans ce cas, il est logique *d'accepter la prédiction faite* par AI. Aucune étape de correction n'est nécessaire, r peut simplement être ajouté à R_U et l'ensemble résultant peut ensuite être simplifié (en effet, il est possible que r spécialise ou généralise certaines règles de R_U).

Par exemple, considérons $R_U = \{(a \wedge b) \rightarrow 1, (\overline{a} \wedge \overline{b}) \rightarrow 0\}$, qui est cohérent, simplifié, mais pas complet ($(0, 1, 0)$ n'est pas classé par R_U). Supposons que AI soit équivalent à l'ensemble suivant de règles de classement : $\{a \rightarrow 1, \overline{a} \rightarrow 0\}$, qui est cohérent, complet, et simplifié. L'instance $(1, 1, 1)$ est classée par R_U comme une instance positive et par AI comme une instance positive. La règle de classement $r = a \rightarrow 1$ utilisée par AI pour classer $(1, 1, 1)$ n'est en conflit avec aucune règle de classement de R_U. La règle $r_U = (a \wedge b) \rightarrow 1$ de R_U

peut être remplacée par la règle plus générale $r = a \rightarrow 1$ de R_{AI}, conduisant ainsi à un nouvel ensemble de règles pour U, donné par $\{a \rightarrow 1, (\overline{a} \wedge \overline{b}) \rightarrow 0\}$. Cet ensemble est cohérent et simplifié. Considérons maintenant l'instance $(0, 1, 0)$. Cette instance n'est pas classée par R_U et elle est classée par AI comme une instance négative. La règle de classement $r = \overline{a} \rightarrow 0$ utilisée par AI pour classer $(0, 1, 0)$ n'est en conflit avec aucune règle de classement de R_U. La règle $r_U = (\overline{a} \wedge \overline{b}) \rightarrow 0$ de R_U peut être remplacée par la règle plus générale $r = \overline{a} \rightarrow 0$ de R_{AI}, ce qui conduit à l'ensemble de règles $\{(a \wedge b) \rightarrow 1, \overline{a} \rightarrow 0\}$, qui est cohérent et simplifié.

On peut facilement observer sur cet exemple que la capacité à dériver des explications abductives aussi générales que possible (en particulier, minimales pour l'inclusion ensembliste) a un impact sur l'ensemble résultant de règles pour U. En effet, supposons que AI soit maintenant donné par l'ensemble de règles de classement $\{(a \wedge b) \rightarrow 1, (a \wedge \overline{b}) \rightarrow 1, \overline{a} \rightarrow 0\}$, qui est cohérent, complet, et simplifié, et que l'instance à classer soit $(1, 1, 1)$. Il est évident que cet ensemble de règles est équivalent à $\{a \rightarrow 1, \overline{a} \rightarrow 0\}$, puisque chacune des deux règles $(a \wedge b) \rightarrow 1$ et $(a \wedge \overline{b}) \rightarrow 1$ pourrait être remplacée par la règle plus générale $a \rightarrow 1$, reflétant le fait que l'explication abductive correspondante $(a \wedge b)$ pour $(1, 1, 1)$ donnée par AI n'est pas minimale pour l'inclusion ensembliste. Ajouter à R_U la règle de classement $(a \wedge b) \rightarrow 1$ utilisée par AI pour classer $(1, 1, 1)$ laisserait R_U inchangé.

4 Expérimentations

Dans nos expérimentations, nous nous sommes concentrés sur un problème de classement binaire : $C = \{1, 0\}$. U est simulé à partir d'une forêt aléatoire qui considère qu'une instance est classée seulement si une proportion assez élevée (70% dans les expérimentations) des arbres de la forêt en conviennent. AI est un arbre de décision unique choisi uniformément au hasard dans F, parmi les arbres dont la précision dépasse 50%.

Dans le tableau 1, la première colonne "Dataset" donne le nom du jeu de données, la deuxième colonne $\#F$ donne le nombre de caractéristiques une fois les attributs catégoriels *one-hot* encodés, la troisième colonne $\#I$ indique le nombre d'instances dans le jeu de données, la quatrième colonne $(\#B)$ indique le nombre de conditions booléennes distinctes utilisées dans F, et la dernière colonne "Repository" précise la source des données.

Dataset	#F	#I	#B	Repository
arrowhead	249	146	86	UCR
australian	38	690	51	openML
balance	4	576	10	UCI
biodegradation	41	1055	69	openML
breastTumor	37	286	38	openML
cleveland	22	303	25	openML
compas	11	6172	13	openML
contraceptive	21	1140	30	UCI
divorce	54	170	36	UCI

TAB. 1 – *Description des ensembles de données utilisés dans les expérimentations.*

Le tableau 2 présente quelques statistiques concernant la trace complète des interactions. La colonne $\#R$ indique le nombre de rectifications effectuées sur AI. La colonne $\#G$ indique le nombre de généralisations (strictes) des règles de R_U qui ont été détectées (rappelons que de telles généralisations peuvent survenir dans le Cas (4)). La colonne "Cas" indique pour chaque

cas, de (1) à (4), le nombre d'instances déclenchantes de ce cas. Enfin, la colonne "$I\#R_U$" donne le nombre Initial de règles dans R_U ("+" indique le nombre de règles concluant à 1, tandis que "-" indique le nombre de règles concluant à 0), et de même pour la colonne "$F\#R_U$" concernant le nombre Final de règles dans R_U.

Dataset	#R	#G	Cas				I#R_U		F#R_U	
			(1)	(2)	(3)	(4)	+	-	+	-
arrowhead	112	7	0	4	10	11	12	4	9	4
australian	453	0	1	67	27	5	8	13	8	13
balance	128	0	0	47	25	28	5	4	5	4
biodegradation	144	10	0	22	14	64	3	9	3	5
breastTumor	48	38	0	0	12	38	0	6	0	5
cleveland	122	0	0	32	15	6	4	5	4	5
compas	0	100	0	0	0	100	2	1	1	1
contraceptive	118	12	0	8	73	19	5	4	5	4
divorce	64	0	0	22	5	2	3	4	3	4

TAB. 2 – *Quelques statistiques sur la nature des interactions réalisées.*

Le tableau 2 montre que les cas les plus fréquents rencontrés lors des expérimentations varient profondément en fonction du jeu de données considéré. Le cas (1) a été le moins fréquent, ce qui peut s'expliquer par le fait que la précision initiale de AI n'était en général pas très inférieure à celle de U.

Dataset	IAccAI	FAccAI	IAccU	FAccU	ICU	FCU
arrowhead	0.596	0.923	1.000	0.953	9.308e-09	0.500
australian	0.702	0.723	0.909	0.909	0.122	0.122
balance	0.711	0.836	0.863	0.863	0.694	0.694
biodegradation	0.746	0.778	0.801	0.799	0.257	0.531
breastTumor	0.580	0.570	0.529	0.581	0.318	0.826
cleveland	0.651	0.783	0.892	0.892	0.328	0.328
compas	0.660	0.660	0.675	0.660	0.640	1.000
contraceptive	0.609	0.627	0.754	0.832	0.159	0.304
divorce	0.917	0.933	0.982	0.982	0.003	0.003

TAB. 3 – *Évolution de la précision de AI, de la précision de R_U et de la couverture de R_U.*

Le tableau 3 indique les précisions initiales (IAccAI et IAccUser) de AI et U, leurs précisions finales (FaccAI et FAccUser), et la couverture initiale (ICU) et finale de R_U. L'étape finale est obtenue une fois qu'au plus 100 instances ont été prises en compte pour déclencher une interaction. Ce tableau montre que selon les ensembles de données utilisés, les interactions entre AI et U peuvent permettre de faire croître la performance prédictive de AI (même si cela n'est pas systématique) et l'ensemble des instances que U est capable de classer (i.e., la couverture de U).

5 Conclusion

Dans ce résumé, nous avons présenté un protocole qui définit des interactions possibles entre un utilisateur U et un classeur (à base d'arbres) AI utilisé par U. Expérimentalement, nous avons montré que ces interactions peuvent être utiles aux deux acteurs en présence.

Remerciements Le travail correspondant a été réalisé dans le cadre de la chaire ANR d'enseignement et de recherche EXPEKCTATION (ANR-19-CHIA-0005-01).

Références

Audemard, G., S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, et P. Marquis (2022). Trading complexity for sparsity in random forest explanations. In *Proc. of AAAI'22*, pp. 5461–5469.

Audemard, G., S. Coste-Marquis, P. Marquis, M. Sabiri, et N. Szczepanski (2024). Designing an XAI interface for tree-based ML models. In *ECAI 2024*, pp. 1075–1082.

Barceló, P., M. Monet, J. Pérez, et B. Subercaseaux (2020). Model interpretability through the lens of computational complexity. In *Proc. of NeurIPS'20*.

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

Breiman, L., J. H. Friedman, R. A. Olshen, et C. J. Stone (1984). *Classification and Regression Trees*. Wadsworth.

Coste-Marquis, S. et P. Marquis (2021). On belief change for multi-label classifier encodings. In *Proc. of IJCAI'21*, pp. 1829–1836.

Coste-Marquis, S. et P. Marquis (2023). Rectifying binary classifiers. In *Proc. of ECAI'23*, pp. 485–492.

Darwiche, A. et A. Hirth (2020). On the reasons behind decisions. In *Proc. of ECAI'20)*, pp. 712–720.

Freund, Y. et R. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci. 55*(1), 119–139.

Gorji, N. et S. Rubin (2022). Sufficient reasons for classifier decisions in the presence of domain constraints. In *Proc. of AAAI'22*, pp. 5660–5667.

Ignatiev, A., N. Narodytska, N. Asher, et J. Marques-Silva (2020). On relating 'why ?' and 'why not ?' explanations. *CoRR abs/2012.11067*.

Ignatiev, A., N. Narodytska, et J. Marques-Silva (2019). Abduction-based explanations for machine learning models. In *Proc. of AAAI'19*, pp. 1511–1519.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*(1), 81–106.

Shih, A., A. Choi, et A. Darwiche (2018). A symbolic approach to explaining bayesian network classifiers. In *Proc. of IJCAI'18*, pp. 5103–5111.

Summary

We present an XAI protocol for ruling interactions between a tree-based ML model (the AI system) and its user U, in the context of a prediction task. The pieces of knowledge held by U concerning the prediction task are supposed to be representable by a consistent yet incomplete set of reliable classification rules. The proposed protocol aims to help U deciding what to do with each prediction made by AI (accept it, reject it). It also aims to improve the quality of further predictions made by AI thanks to the expertise of U, and, reciprocally, to complete the pieces of knowledge held by U by leveraging the predictions made by AI.