AutoXAI: Un cadre pour sélectionner automatiquement la solution d'XAI la plus adaptée

Robin Cugny*,**, Julien Aligon**, Max Chevalier**, Geoffrey Roman Jimenez*
Olivier Teste**

*SolutionData Group
{rcugny, groman-jimenez}@solutiondatagroup.fr,
**Université Toulouse 1, Université Toulouse 2, Université Toulouse 3, IRIT
Toulouse, France
{robin.cugny, julien.aligon, max.chevalier, olivier.teste}@irit.fr

Résumé. Ce papier est un résumé des travaux publiés à la conférence CIKM 2022, Cugny et al. (2022). Un grand nombre de solutions d'XAI (eXplainable Artificial Intelligence) ont été proposées ces dernières années. Récemment, grâce à de nouvelles mesures d'évaluation des explications, il est devenu possible de les comparer. Cependant, la sélection de la solution d'XAI la plus pertinente reste une tâche fastidieuse, surtout si l'utilisateur a des besoins et des contraintes spécifiques. Dans cet article, nous proposons d'introduire AutoXAI, un cadre qui recommande la meilleure solution d'XAI et ses hyperparamètres au regard du contexte de l'utilisateur (ensemble de données, modèle d'apprentissage, besoins et contraintes liées à l'XAI). Notre approche s'inspire des travaux liés au domaine des systèmes de recommandation adaptés au contexte ainsi que de l'AutoML (Automated Machine Learning) pour nos stratégies d'optimisation et d'évaluation. Dans ce papier résumé, nous illustrons notre approche au travers d'un cas d'usage montrant qu'AutoXAI recommande bien une solution adaptée (avec les meilleurs hyperparamètres) aux besoins et contraintes de l'utilisateur.

1 Introduction

Le présent article est un résumé de l'article publié dans la conférence CIKM 2022, Cugny et al. (2022). Les modèles d'apprentissage machine (ML) sont désormais largement utilisés dans l'industrie. Il y a, cependant, un vrai besoin de mieux comprendre le problème de boite noire lié à l'utilisation de modèles prédictifs. Au cours de la dernière décennie, le domaine de l'intelligence artificielle explicable (XAI) a proposé une grande variété de solutions pour faciliter la compréhension des modèles de ML (Carvalho et al., 2019).

Cependant, les data scientists souhaitant appliquer une solution d'XAI adaptée se trouvent confrontés aux problèmes suivants :

— Ils doivent vérifier que les solutions d'XAI sont compatibles avec le type de données et le modèle prédictif ;
— Les solutions d'XAI doivent expliquer, dans un format approprié, ce qu'ils veulent comprendre ;

— Ils doivent manuellement évaluer l'efficacité des explications produites ;
— Le contexte de l'utilisateur (données, modèle, besoins et contraintes liés à l'XAI) exige que les explications répondent à des critères de qualité spécifiques (appelés propriétés des explications), ce qui implique l'utilisation de mesures d'évaluation appropriées ;
— Ils doivent trouver les meilleurs hyperparamètres pour chacune des solutions d'XAI sélectionnées afin de conserver la meilleure d'entre elles.

Ainsi, nous proposons dans cet article un cadre, appelé AutoXAI, recommandant des solutions d'XAI avec des hyperparamètres optimisés en adéquation avec le contexte du data scientist. Nous nous inspirons principalement des travaux issus des systèmes de recommandation adaptés au contexte et des approches liées à l'AutoML et adaptables au domaine de l'XAI.

Le reste de cet article est organisé comme suit. La section 2 propose un état de l'art, présentant en particulier les approches existantes sur la sélection de méthodes d'XAI. Les définitions formalisant le contexte du data scientist sont proposées dans la section 3.1. Le cadre d'AutoXAI est introduit dans la section 3.2. Le cas d'usage illustrant l'intérêt de notre approche est décrit dans la section 4. Une conclusion et les possibles perspectives de travail sont finalement présentées dans la section 5.

2 État de l'art

De nombreuses solutions d'XAI existent désormais. Par exemple, Carvalho et al. (2019) proposent de regrouper les solutions d'XAI en fonction du type d'explication produite : résumé d'attributs, mécanismes internes du modèle, exemple de données, modèle de substitution intrinsèquement interprétable, ensembles de règles, explications en langage naturel et réponses à des questions. Plus tard, Liao et al. (2020) suggèrent que les explications répondent à des questions spécifiques sur les données, leur traitement et les résultats de ML. Ils assimilent ainsi les solutions d'XAI à des questions et créent une banque de questions d'XAI ouvrant la voie à la conception d'applications d'XAI centrées sur l'utilisateur. Overton (2011) définit une explication comme un *explanan* : la réponse à la question et un *explanandum* : ce qui doit être expliqué. Ces deux éléments fournissent une caractérisation des explications et permettent ainsi à l'utilisateur de préciser quelle explication est la plus adaptée.

Il existe également de nombreux critères de qualité permettant d'évaluer la pertinence des explications. Ces mesures sont associées à des propriétés d'explication que Nauta et al. (2022) proposent d'unifier. Voici les propriétés qui seront étudiées dans cet article : la *Continuité* décrit à quel point la fonction d'explication est continue et généralisable, l'*Exactitude* décrit à quel point l'explication est fidèle à la boîte noire et la *Compacité* décrit la taille de l'explication.

En considérant la variété des méthodes d'explication existantes et les critères de qualité associés, l'établissement d'un système de recommandation apparaît judicieuse. Pour recommander des solutions d'XAI adaptées, il faut, cependant, prendre en compte l'ensemble du contexte d'un data scientist. Selon Adomavicius et al. (2011), les systèmes de recommandation adaptés au contexte offrent des recommandations plus pertinentes que les systèmes plus classiques (comme ceux basés sur un filtrage collaboratif simple). Le contexte utilisateur est alors intégré au cours de trois phases : le préfiltrage contextuel qui sélectionne un sous-ensemble de candidats possibles avant la recommandation, la modélisation contextuelle qui utilise le contexte dans le processus de recommandation et le postfiltrage contextuel qui ajuste la recommandation.

Un système de recommandation n'est cependant pas suffisant pour adapter des méthodes d'XAI liées le plus souvent à des modèles de ML. La conception d'algorithmes de ML est une tâche itérative consistant à tester et à modifier à la fois l'architecture et les hyperparamètres de l'algorithme. C'est une tâche répétitive et chronophage. C'est pour cette raison qu'une partie des travaux s'est concentrée sur l'automatisation de la conception d'algorithmes de ML, à savoir l'AutoML (He et al., 2021). La principale stratégie qui nous intéresse ici est l'optimisation des hyperparamètres (HPO) qui consiste à trouver les meilleurs hyperparamètres d'un algorithme de ML au regard d'une fonction de coût.

Concernant les approches pour le choix d'une solution d'XAI, un data scientist a actuellement la possibilité de choisir une solution d'XAI parmi des bibliothèques d'XAI, des comparatifs/benchmarks et des frameworks d'AutoML. Les bibliothèques d'XAI disponibles rassemblent des solutions récentes mais n'intègrent pas l'évaluation automatique de l'explication et ne recommandent pas de solutions d'XAI en fonction des besoins et des contraintes des data scientists. Les comparatifs et benchmarks Yeh et al. (2019); Alvarez-Melis et Jaakkola (2018) comparent l'efficacité des solutions d'XAI à l'aide de mesures d'évaluation d'XAI. Cependant, les résultats obtenus dépendent d'ensembles de données et modèles de ML spécifiques, qui ne sont pas forcément ceux que le data scientist compte utiliser. Enfin, Vermeire et al. (2021) soulignent que les utilisateurs devraient être guidés dans le choix des solutions d'XAI et proposent une première méthodologie pour cette problématique, tandis que Palacio et al. (2021) proposent un cadre théorique pour faciliter la comparaison entre les solutions d'XAI.

3 Notre approche

3.1 Définitions

Soit X, Y un **ensemble de données** avec les observations $X = \{x_i\}_{i=1}^n | x_i \in \mathbb{R}^d$ et les labels correspondants $Y = \{y_i\}_{i=1}^n | y_i \in \mathbb{R}$, n est le nombre d'observations et d de dimensions (aussi appelés attributs).

Un **modèle de ML** est entraîné avec un ensemble de données X, Y en inférant des relations statistiques entre X et Y. Ce modèle peut alors être utilisé comme une fonction prédictive qu'on note $f : X \to \hat{Y}$ avec $\hat{Y} = \{\hat{y}_i\}_{i=1}^n | \hat{y}_i \in \mathbb{R}$, les prédictions produites.

On note $\mathcal{E} = \{\mathcal{E}_i\}_{i=1}^k$ l'ensemble de tous les **explanandum**, où l'explanandum $\mathcal{E}_i$ est un descripteur pour les fonctions d'explication spécifiant *ce qui est expliqué*. On note aussi $\mathcal{E}' = \{\mathcal{E}'_j\}_{j=1}^{k'}$ l'ensemble de tous les **explanan**, où l'explanan $\mathcal{E}'_j$ est un descripteur pour les fonctions d'explication spécifiant *comment c'est expliqué*.

Les **propriétés** des explications sont des critères descriptifs de qualité pour les explications. On note P_r, l'ensemble des propriétés que les explications vérifient ou non. Le data scientist peut ainsi spécifier ses besoins avec $(\mathcal{E}, \mathcal{E}')$ et ses contraintes avec P_r.

Une **solution d'XAI** agit comme une fonction qui produit une ou plusieurs explications. On note $E = \{e_t\}_{t=1}^l$, l'ensemble des explications avec $l \in \mathbb{N}$ le nombre d'explications. On note $f_e^{(h)} : P(X, Y, F, \hat{Y}) \to E$ la fonction d'explication avec $P(X, Y, F, \hat{Y})$ une partition de $\{X, Y, F, \hat{Y}\}$ et h les hyperparamètres de la solution d'XAI. $f_e^{(h)} \in F_e$ avec F_e l'ensemble des fonctions d'explication. Les hyperparamètres sont des paramètres statiques qui déterminent le comportement de la solution d'XAI. Pour les modèles transparents $f = f_e^{(h)}$.

Une **mesure d'évaluation d'XAI** évalue une propriété et est souvent adaptée à un type spécifique d'explication. On note l'ensemble des mesures d'évaluation d'XAI $M = \{m_q\}_{q=1}^{c}$, où $m_q : P(X, F, F_e, Y) \to \mathbb{R}$, avec $P(X, F, F_e, Y)$ une partition de $\{X, F, F_e, Y\}$, telle que m_q évalue $p_q \in P_r$.

Par exemple, on apprend f un modèle de ML (un Perceptron multicouche), sur X, Y l'ensemble de données Diabetes [1], et on utilise $f_e^{(h)}$, une solution d'XAI (LIME [2]) pour obtenir des explications. On peut alors mesurer p_q la propriété de *Continuité* avec m_q, la **Robutesse** (Alvarez-Melis et Jaakkola, 2018), une mesure d'évaluation d'XAI.

3.2 AutoXAI

La Figure 1a décrit l'architecture globale d'AutoXAI :

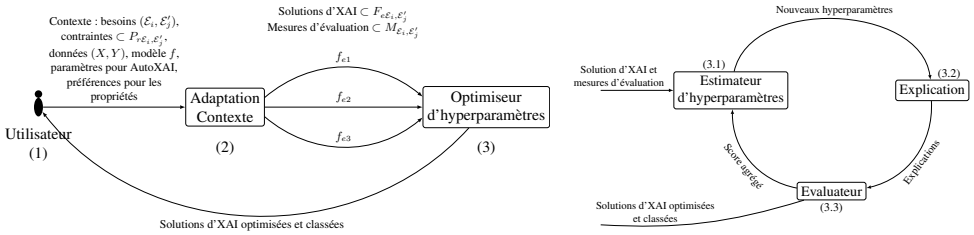
1. L'**utilisateur** donne les éléments de son contexte, les paramètres pour AutoXAI et ses préférences concernant les propriétés des explications ;
2. Le **composant d'adaptation** au contexte sélectionne un sous-ensemble de solutions d'XAI correspondant aux besoins et un sous-ensemble de mesures d'évaluation pour s'assurer que les contraintes de l'utilisateur soient respectées ;
3. Pour chaque solution d'XAI, l'**optimiseur d'hyperparamètres** recherche les hyperparamètres qui réduiront la fonction de perte basée sur les scores agrégés des mesures d'évaluation. Pour ce faire, il effectue les opérations suivantes en boucle (voir Figure 1b).
   (a) L'**estimateur d'hyperparamètres** propose de nouveaux hyperparamètres en fonction de l'algorithme d'optimisation choisi ;
   (b) Le **composant d'explications** utilise la solution d'XAI et les hyperparamètres nouvellement proposés pour produire des explications ;
   (c) L'**évaluateur** applique les mesures d'évaluation d'XAI aux explications et agrège les scores ainsi obtenus.

Certaines solutions d'XAI et certaines mesures d'évaluation d'XAI n'ont pas été conçues pour être utilisées plusieurs fois de suite et présentent une complexité algorithmique importante. Pour réduire le coût en temps de ces algorithmes, sans modifier leur architecture, nous avons adapté les stratégies heuristiques qui existent dans le domaine de l'AutoML : l'early stopping, qui consiste à arrêter l'optimisation des hyperparamètres et/ou l'évaluation d'XAI lorsqu'un seuil de stabilisation est atteint, et le partage d'informations qui consiste à réutiliser les résultats intermédiaires et à partager les informations entre les évaluations.

4 Cas d'usage

Pour le cas d'usage, nous avons Alice, une data scientist dans un laboratoire médical, et Bob, un médecin. Bob utilise un modèle de ML boîte noire comme outil d'aide à la décision et demande une explication pour les prédictions du modèle afin de vérifier certains cas rares. Ici

1. https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html
2. https://github.com/marcotcr/lime



(a) Architecture globale d'AutoXAI

(b) Détails sur l'optimiseur d'hyperparamètres

FIG. 1 – Architecture d'AutoXAI.
Les figures se lisent en suivant le numéro des étapes. Dans la Figure 1a, pour chaque solution d'XAI, l'étape (3) optimise les hyperparamètres par rapport aux scores agrégés des mesures d'évaluation en entrant dans une boucle décrite en Figure 1b.

les besoins de Bob sont les suivants : il veut savoir "Pourquoi on obtient telle prédiction ?" et il souhaite connaître les contributions des attributs pour la prédiction. Concernant les contraintes, Bob veut des explications stables à cause du bruit dans les mesures, précises à causes des enjeux et concises pour ne pas perdre de temps. Ces contraintes correspondent respectivement aux propriétés de *Continuité*, d'*Exactitude* et de *Compacité*.

Les ensembles de données utilisés sont *Diabetes dataset* et *Pima Indians dataset* [3]. *Diabetes dataset* a 10 attributs et est conçu pour une tâche de régression visant à prédire la progression de la maladie. *Pima Indians dataset* a 8 attributs et est fait pour une classification binaire afin de prédire si les patients sont diabétiques. Le modèle de boîte noire utilisé est l'implémentation scikit-learn d'un perceptron multicouche [4]. Pour la régression nous utilisons le MLPRegressor et pour la classification nous utilisons le MLPClassifier.

Les solutions d'XAI implémentées sont LIME et Kernel SHAP [5].

Les mesures d'évaluation d'XAI et leurs propriétés correspondantes sont les suivantes :

— **Robustesse**, Alvarez-Melis et Jaakkola (2018), *Continuité*
— **Infidélité**, Yeh et al. (2019), *Exactitude*
— **Nombre d'attributs**, Rosenfeld (2021), *Compacité*

Pour l'agrégation dans ce scénario, Alice et Bob ont fixé les poids à 1, 2 et 0,5 pour respectivement la robustesse, l'infidélité et le nombre d'attributs. Alice, fixe le nombre d'itérations à 25. La stratégie d'HPO est une optimisation bayésienne. Enfin, les stratégies d'évaluation utilisées sont l'early stopping pour le calcul des mesures d'évaluation d'XAI et le HPO, et le partage d'informations pour la robustesse et l'infidélité. Le code permettant de reproduire les résultats de ce cas d'usage est disponible à l'adresse suivante : `https://github.com/RobinCugny/AutoXAI`.

Un extrait du classement produit par AutoXAI pour *Diabetes dataset* est dans le Tableau 1 et celui pour le *Pima Indians dataset* est dans le Tableau 2. Les solutions d'XAI sont triées par ordre décroissant en fonction du score agrégé. Pour montrer diverses solutions d'XAI,

3. `https://www.kaggle.com/uciml/pima-indians-diabetes-database`
4. `https://scikit-learn.org/stable/modules/neural_networks_supervised.html`
5. `https://github.com/slundberg/shap`

TAB. 1 – Extrait du classement produit par AutoXAI sur *Diabetes dataset*.

Score agrégé	Robustesse standardisée	Fidélité standardisée	Nb attributs standardisé	Solution d'XAI	Hyperparamètres
1.023	0.727	0.833	1.351	LIME	1 ;3656
1.019	0.703	0.991	0.745	LIME	3 ;8782
0.963	0.682	1.068	0.139	LIME	5 ;5392
-0.287	0.310	-0.924	1.351	SHAP	1 ;1304 ;auto
-0.633	-0.319	-0.975	0.745	SHAP	3 ;1571 ;aic
-0.639	0.014	-1.000	0.139	SHAP	5 ;1148 ;aic

TAB. 2 – Extrait du classement produit par AutoXAI sur *Pima Indians dataset*.

Score agrégé	Robustesse standardisée	Fidélité standardisée	Nb attributs standardisé	Solution d'XAI	Hyperparamètres
1.412	0.744	1.435	1.243	LIME	1 ;5347
1.282	0.575	1.325	1.243	SHAP	1 ;509 ;bic
0.361	0.633	0.117	0.430	LIME	3 ;8329
0.176	0.339	-0.014	0.430	SHAP	3 ;713 ;auto
0.070	0.262	0.070	-0.383	SHAP	5 ;537 ;bic
-0.185	0.599	-0.481	-0.383	LIME	5 ;7023

nous présentons trois combinaisons d'hyperparamètres avec LIME et trois avec SHAP. Pour visualiser les explications, nous avons opté pour un nombre d'attributs de 1, 3 et 5. Ainsi, l'utilisateur peut vérifier si des explications courtes sont suffisantes pour comprendre la prédiction ou si plus de caractéristiques seraient utiles. Dans les colonnes Hyperparamètres, les deux premiers hyperparamètres pour LIME comme pour SHAP sont : premièrement, le nombre de caractéristiques dans l'explication, et deuxièmement, le nombre de perturbations utilisées pour construire le modèle linéaire. Le dernier hyperparamètre pour SHAP est la régularisation l1 à utiliser pour la sélection des attributs.

LIME est systématiquement supérieur à SHAP dans les classements avec ces mesures d'évaluation d'XAI bien qu'il semble que SHAP soit légèrement plus performant sur *Pima Indians dataset* que sur *Diabetes dataset*. Dans *Pima Indians dataset*, SHAP est plus fidèle et réussit à capturer les changements de la fonction de prédiction. Ainsi, il semble trouver les mêmes relations entre attributs que celles utilisées par le modèle.

La compacité a un impact sur les autres propriétés d'après Nauta et al. (2022). En outre, Bob, le médecin, doit observer les explications pour confirmer le nombre d'attributs nécessaires à la compréhension de la prédiction. Pour cela, prenons une observation de *Diabetes dataset*. Le modèle fait une prédiction et Bob demande *Pourquoi cette prédiction ?* et veut connaître les attributs qui contribuent à celle-ci. Les explications produites par les solutions d'XAI recommandées par AutoXAI sont en Figure 2. En bas à droite se trouve LIME avec les hyperparamètres par défaut. Certaines caractéristiques ont peu d'influence sur la prédiction et sont inutiles pour répondre à la question de Bob. Avec ces explications de tailles différentes, Bob peut voir ce qui est important et ce qui est négligeable pour lui. Il peut donc choisir la taille de l'explication qu'il souhaite, en gardant à l'esprit les scores des propriétés et le score agrégé.

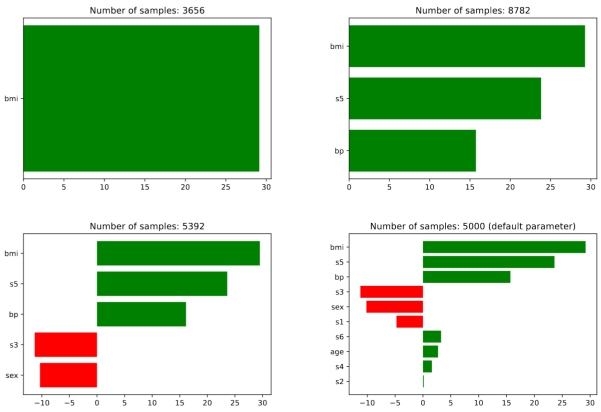


FIG. 2 – Explications produites par LIME avec différents hyperparamètres pour une même observation de *Diabetes dataset*. Les explications sont de différentes tailles et présentent ainsi différents attributs.

5 Conclusion et perspectives

Dans cet article, nous proposons AutoXAI, un cadre qui recommande les meilleures solutions d'XAI en fonction du contexte de son utilisateur. AutoXAI automatise la tâche fastidieuse de sélection d'une solution d'XAI et de ses hyperparamètres. Il produit un classement des solutions en tenant compte des préférences de l'utilisateur. Bien qu'AutoXAI soit centré sur le système comme l'AutoML, ici l'utilisateur spécifie les besoins, les contraintes et choisit la solution d'XAI dans le classement. Nous montrons également qu'il peut y avoir un compromis entre les propriétés. La compacité, en particulier, devrait être surveillée et l'utilisateur devrait décider en vérifiant les explications. Par ailleurs, le choix d'une explication plutôt qu'une autre peut soulever des questions éthiques.

Un travail à court terme serait de compléter AutoXAI avec de nouvelles adaptations des méthodes d'AutoML. Des perspectives à plus long terme consistent à appliquer AutoXAI dans un cadre réel et à analyser les retours des utilisateurs pour évaluer son efficacité et l'améliorer. Enfin, l'étude de l'influence des propriétés d'XAI les unes aux autres sera un sujet d'étude important dans le domaine de l'évaluation des solutions d'XAI.

Références

Adomavicius, G., B. Mobasher, F. Ricci, et A. Tuzhilin (2011). Context-aware recommender systems. *AI Magazine 32*(3), 67–80.

Alvarez-Melis, D. et T. S. Jaakkola (2018). On the robustness of interpretability methods. *arXiv preprint arXiv :1806.08049*.

Carvalho, D. V., E. M. Pereira, et J. S. Cardoso (2019). Machine learning interpretability : A survey on methods and metrics. *Electronics 8*(8).

Cugny, R., J. Aligon, M. Chevalier, G. R. Jimenez, et O. Teste (Eds.) (2022). *AutoXAI : A Framework to Automatically Select the Most Adapted XAI Solution*. ACM.

He, X., K. Zhao, et X. Chu (2021). Automl : A survey of the state-of-the-art. *Knowledge-Based Systems 212*, 106622.

Liao, Q. V., D. Gruen, et S. Miller (2020). *Questioning the AI : Informing Design Practices for Explainable AI User Experiences*, pp. 1–15. New York, NY, USA : Association for Computing Machinery.

Nauta, M., J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, et C. Seifert (2022). From anecdotal evidence to quantitative evaluation methods : A systematic review on evaluating explainable ai. *arXiv preprint arXiv :2201.08164*.

Overton, J. (2011). Scientific explanation and computation. In T. Roth-Berghofer, N. Tintarev, et D. B. Leake (Eds.), *Explanation-aware Computing, Papers from the 2011 IJCAI Workshop, Barcelona, Spain, July 16-17, 2011*, pp. 41–50.

Palacio, S., A. Lucieri, M. Munir, S. Ahmed, J. Hees, et A. Dengel (2021). Xai handbook : Towards a unified framework for explainable ai. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3766–3775.

Rosenfeld, A. (2021). Better metrics for evaluating explainable artificial intelligence. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, Richland, SC, pp. 45–50. International Foundation for Autonomous Agents and Multiagent Systems.

Vermeire, T., T. Laugel, X. Renard, D. Martens, et M. Detyniecki (2021). How to choose an explainability method ? towards a methodical implementation of xai in practice. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Cham, pp. 521–533. Springer International Publishing.

Yeh, C.-K., C.-Y. Hsieh, A. Suggala, D. I. Inouye, et P. K. Ravikumar (2019). On the (in)fidelity and sensitivity of explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, et R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Summary

This paper is a summary of the work published at the CIKM 2022 conference, Cugny et al. (2022). A large number of XAI (eXplainable Artificial Intelligence) solutions have been proposed in recent years. Recently, thanks to new XAI evaluation methods, it has become possible to compare them. However, selecting the most relevant XAI solution remains a tedious task, especially if the user has specific needs and constraints. In this paper, we propose to introduce AutoXAI, a framework that recommends the best XAI solution and its hyperparameters while taking into account the user's context (dataset, learning model, XAI needs and constraints). Our approach draws on work related to the field of context-based recommender systems as well as AutoML (Automated Machine Learning) for our optimization and evaluation strategies. In this summary paper, we illustrate our approach through a use case showing that AutoXAI recommends the most suitable solution (with the best hyperparameters) to the user's needs and constraints.