THÈSE de DOCTORAT

de l'UNIVERSITÉ TOULOUSE CAPITOLE

Présentée et soutenue par

Madame Elodie ESCRIVA

Le 1 mars 2024

Amélioration des explications attributives locales pour appuyer l'analyse prédictive par apprentissage automatique : application au secteur de la santé et aux outils d'aide à la décision médicale.

École doctorale : **Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche : **IRIT : Institut de Recherche en Informatique de Toulouse**

Thèse dirigée par *Madame Chantal SOULE-DUPUY et Monsieur Julien ALIGON*

Composition du jury

Rapporteure : Mme Marie-Jeanne LESOT
Rapporteur : M. Mourad BOUNEFFA
Examinateur : M. Olivier TESTE
Examinateur : M. Nicolas LABROCHE
Directrice de thèse : Mme Chantal SOULE-DUPUY
Co-directeur de thèse : M. Julien ALIGON

“ L’université n’entend ni approuver ni désapprouver les opinions particulières de l’auteur. ”

Remerciements

La réalisation de ma thèse a été un beau parcours de trois ans, ponctué de nombreuses rencontres percutantes et décisives, ébauchant ce que mon travail de recherche est devenu.

Je remercie premièrement ma directrice et mon co-directeur de thèse, Chantal Soulé-Dupuy Chantal et Julien Aligon, ainsi que mon encadrant industriel, Jean-Baptiste Excoffier. Vous m'avez accompagnée dans tout ce travail de thèse. Vous avez enrichi mes réflexions et encouragé à donner le meilleur de moi-même. Merci pour votre temps et vos conseils. Merci aux personnes qui ont accompagné mes travaux de thèses et y ont contribué, Manon Martin, Tom Lefrere, Gabriel Ferrettini, Paul Montsarrat et Julien May.

Je souhaite également remercier les institutions et personnes qui ont permis la réalisation de cette thèse. D'abord l'Institut de Recherche en Informatique de Toulouse et l'Université Toulouse Capitole, pour le cadre académique fourni, et Kaduceo pour le contexte en santé, notamment Matthieu Ortala, Cédric Giorgi et Noémie Salaün-Penquer. Ce fut un réel plaisir d'évoluer à l'intersection de ces deux contextes et d'avoir l'opportunité de poursuivre mes travaux dans le domaine de la santé.

Il me semble aussi important de remercier les membres de mon jury qui ont accepté d'évaluer mon travail. Notamment mes rapporteurs de thèse, Marie-Jeanne Lesot et Mourad Bouneffa, pour les rapport précis, positif et de qualité qu'ils ont écrit. Leurs remarques étaient particulièrement constructives dans l'analyse de mes travaux et leurs perspectives. Merci également à Nicolas Labroche et Olivier Teste, examinateurs de mon jury, pour les discutions, les retours, et l'expérience que vous m'avez apporté, en tant qu'examinateur et même avant cela, lors de nos rencontres, de nos discussions. Merci Olivier d'avoir cru en moi en tant que chercheuse en premier, dès 2019 pendant mon stage auprès de toi. Merci Nicolas de croire en moi et de vouloir continuer à collaborer avec moi, même si tu veux me faire déménager dans le froid de Blois.

Ma thèse n'aurait également pas été ce qu'elle est sans mes rencontres avec mes collègues de Kaduceo et les autres doctorants de l'Université. Votre soutien, votre intérêt pour mes travaux, votre présence, nos discussions, nos glaces, nos chocoblast, les soirées du jeudi et du vendredi soir m'ont permi de traverser cette thèse plus sereinement et avec pleins de moment de joie. Côté Kaduceo, merci aux survivants Benjamin, Corentin, Hugo, Jean-Baptiste, Samuel, et aux anciens, Arthur, Arthur, Cécile, Cédric, Christelle, Elise, Jordan, Mélissa. Mention spéciale pour Corentin, tu me suis depuis longtemps maintenant dans mon parcours universitaire et professionel, et mon aventure Kaduceo a aussi commencé grâce à toi. Côté doctorants, merci à la team des Julianitos, Emmanuel, Haomiao et Robin, pour nos discussions passionnantes sur le XAI, nos casse-têtes à comprendre les mathématiques, notre entraide, nos travaux communs, et pour vos retours toujours précieux sur mon travail. J'espère que je vous ai montré la voie pour que vous réussissiez en beauté vos thèses. Merci aux autres doctorants, les anciens et les petits nouveaux, qui ne comprenaient souvent pas grand choses à nos délires : Bastien, Clé-

ment, Flavien, Landy, Ronan, Théo, Vincent-Nam, Vlada. Vous avez rythmé mes fins de semaines de rires, de discussions politiques enflammés, de choco-bananes, et de belles discussions. Mention spéciale à Bastien, merci pour ton soutien, nos discussions et à tout ce que tu m'apportes encore aujourd'hui dans ma vie de tous les jours. Dans mon coeur, tu es un Docteur et tu mérites ce titre à plein d'égard. Je sais que ma vie serait différente aujourd'hui si je ne t'avais pas rencontré.

Enfin, dans un aspect plus personnel, merci aux personnes qui partagent ma vie de tout les jours, notamment sportive. Merci au club de volley-ball de Fonsorbes et à l'équipe R2 pour tous les matchs, les entrainements, l'esprit de compèt. Vous m'avez aidé à traverser cette thèse sûrement sans vous en rendre compte. Merci également à l'équipe de Baseball5 de Montpellier, pour leur accueil, leur amitié, leur soutien à distance. Merci à Romy pour sa présence, pour son soutien, pour nos discussions interminables et pour la joie qu'être ton amie m'apporte.

Merci Flavien pour ton soutien et ton amour sans faille, dans les moments difficiles et dans les moments de joie de cette thèse et de notre vie. Tu crois en moi depuis deux ans, sans jamais flancher et ce soutien a été un des plus important quand j'avais besoin d'aide pour rester concentrer sur l'accomplissement de ma thèse. J'ai grandi avec cette thèse et à tes côtés, et je suis fière de la personne que je suis aujourd'hui. Merci à la famille de Flavien pour votre accueil et votre soutien dès que vous m'avez rencontré, je me sens soutenu et comme à la maison chaque fois que je viens chez vous. Merci d'avoir ajouté une nouvelle corde à mon arc, en me permettant de devenir une valeur ajoutée à votre belle famille. Merci à mes parents et à mes frères, pour tout ce que vous avez fait pendant mes nombreuses années d'études, lors de mon retour à la maison l'année dernière, pour votre soutien et votre confiance.

Résumé

Dans le domaine de la santé, l'analyse des données est un enjeu fort dans l'amélioration des prises en charge, la prévention des maladies ou l'adaptation des thérapies à chaque patient. Initialement, cette analyse de données est basée sur des méthodes statistiques telles que les statistiques descriptives et inférentielles, afin de découvrir de nouvelles connaissances dans les données. L'apparition de l'apprentissage automatique a permis de nouveaux cas d'usages grâce à son pouvoir prédictif. Son essor a également été favorisé par des applications aux retombées positives. Cependant, les modèles prédictifs sont qualifiés de boite fermées à cause de leur complexité, leur architecture ou de leur statut propriétaire. Comprendre leur fonctionnement et leur prédiction est alors critique, notamment dans les domaines sensibles.

Le domaine de l'Explicabilité du Machine Learning (XML) a émergé afin d'expliquer le comportement des modèles prédictifs et leurs prédictions. Deux approches s'opposent entre les modèles intrinsèquement interprétables et les méthodes post-hoc d'explication. Les modèles interprétables ont des structures pouvant être directement analysées, tels que les arbres ou les règles de décisions. Les méthodes post-hoc agnostiques, à l'opposé, s'appliquent sur des modèles déjà entraînés pour expliquer leur fonctionnement et leurs prédictions. Les méthodes post-hoc agnostic d'explication locale permettent notamment d'expliquer individuellement chaque prédiction, quelque soit le modèle. Ces méthodes populaires subissent cependant de nombreuses critiques quant à leur efficacité, leurs hypothèses restrictives et la nécessité de privilégier les modèles interprétables dans les domaines sensibles. Cependant, ces derniers peuvent devenir boite-fermée à cause de leur complexité ou pour des raisons propriétaires. Les méthodes post-hoc sont alors la seule alternative restante actuellement pour expliquer les prédictions.

Pour intégrer les explications post-hoc dans des applications médicales, plusieurs problématiques se posent autour des forces et des faiblesses des explications locales, de leur analyse et de leur utilisation par des personnels de santé non-experts en modèle prédictif. Le comportement et les limites des méthodes explicatives sont un point critique lors de leur utilisation, notamment dans les domaines sensibles. De plus, afin de comprendre les liens entre les prédictions, le modèle et les données, les explications peuvent être vues comme de nouvelles données à analyser et à explorer. Enfin, dans le cadre d'applications métiers, les utilisateurs finaux sont souvent peu impliqués dans la conception des applications intégrant des modèles prédictifs et des explications.

Dans cette thèse, nous contribuons à l'amélioration et l'implémentation des explications dans le domaine de la santé sous quatre axes. Premièrement, nous dressons un état de l'art du domaine, des méthodes post-hoc locales et leurs évaluations, des différentes utilisations des explications et des tests réalisés avec des utilisateurs. Puis, nous améliorons la méthode d'explication locale *Coalitionnelle* et la comparons à la littérature afin de mettre en lumière leurs forces, faiblesses et limites. Nous montrons les meilleurs contextes d'utilisation de chaque méthode et qu'un gap existe encore pour les larges jeux

de données. Ensuite, nous proposons une approche d'analyse des explications basée sur le clustering afin d'extraire des informations sur le modèle, les données et les prédictions, offrant des perspectives d'analyse de ces explications. Enfin, nous étudions, en définissant les besoins des utilisateurs, comment implémenter efficacement les explications dans des applications médicales, en combinant plusieurs approches et des analyses statistiques. Nous proposons ainsi notre protocole de tests utilisateurs évaluant l'apport des explications pour des professionnels de santé, réalisé en collaboration avec des experts métiers.

Summary

In healthcare, data analysis is a powerful tool for improving treatment, preventing disease and adapting therapies to individual patients. Initially, this data analysis was based on statistical methods, such as descriptive and inferential statistics, to discover new knowledge in the data. The emergence of machine learning has opened up new possibilities thanks to its predictive power, its growth being boosted by applications with positive results and outcomes. However, predictive models are often described as closed-boxes because of their complexity, architecture or proprietary design. Understanding how they work and how they predict is therefore critical, particularly in sensitive fields like healthcare.

The field of Explainable Machine Learning (XML) has arisen to explain the behaviour of predictive models and their predictions. There are two main approaches: intrinsically interpretable models and post-hoc explanation methods. Intrinsically interpretable models have structures that can be directly analysed, such as trees or decision rules. Post-hoc methods, on the other hand, are applied to already trained models to explain their behaviour and prediction. In particular, local explanation agnostic post-hoc methods explain each prediction individually, whatever the model. However, these popular methods are often criticised for their effectiveness, restrictive assumptions and the need to favour interpretable models in sensitive areas. However, these interpretable models can become closed-boxes due to their complexity or for proprietary issues. Post-hoc methods are then the only alternative currently available to explain the predictions.

When integrating post-hoc explanations into medical applications, several issues arise concerning the strengths and weaknesses of local explanations, their analysis and their use by healthcare personnel who are not experts in predictive models. The behaviour and limitations of explanatory methods are critical points when used, particularly in sensitive areas. In addition, to understand the links between modelling, data and predictions, explanations can be seen as new data to be analysed and explored. Finally, in medical applications, end users are often rarely involved in the application development and integration of explanations, making the explanations' contribution challenging to assess.

In this thesis, we contribute to improve and implement explanations in the healthcare domain in four areas. First, we review the current state of the art, including domain definitions, intrinsically interpretable models, post-hoc local explanation methods and their evaluations. We also describe the different uses of explanations and the tests carried out with domain expert users.

We then improve the local *Coalitional* explanation method and compare seven local attributive explanation methods to highlight their strengths, weaknesses and limitations. We show the best contexts for using each explanation method depending on the characteristics of the datasets and the predictive model based on six metrics and more than 300 open datasets and that a gap still exists for large datasets.

We then propose an approach for analysing explanations to discover insights into the

model, data and predictions. We show that clustering explanations offer better groups of instances than clustering raw data for several families of clustering, offering positive outlooks for analysing explanations.

Finally, we detail our implementation of explanations in medical applications. By defining user needs and the purpose of explanations, we study how to present and analyse them efficiently by combining several explanations and statistical analysis approaches. Finally, we present our user-testing protocol for evaluating the contribution of explanations for healthcare professionals, created in collaboration with medical experts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Context

In France, since 2019, the Health Data Hub has been building a unique database of French healthcare data thanks to the collaboration of 56 stakeholders. The project currently brings together 10 different national databases, including health insurance data, death data, hospital data and medico-economic data on breast cancer, rare diseases and COVID-19. In 4 years, more than 7,000 personal health data analysis projects have been launched via the Health Data Hub, with the attraction growing year on year [1].

In healthcare, data analysis presents a twofold challenge. (1) In medical research, data analysis is usually used for exploratory purposes. It is used to create hypotheses about the links and causality between medical data and a diagnosis to improve pathology detection, management or survival rate (Sidey-Gibbons and Sidey-Gibbons, 2019). (2) In everyday practice, data analysis supports the administrative process and the organisation of care pathways, helps to prevent undesirable events and implements preventive measures, to improve patient health and ensure that the right therapeutic protocols are applied. Batko and Ślęzak (2022) defines 8 major fields of application for medical data analysis: diagnostic support, therapeutic support, precision medicine (or personalised medicine), preventive medicine, telemedicine, health population support, medical research and cost reduction.

Initially, data analysis was carried out using classical statistical tools, such as descriptive statistics, inferential statistics and exploratory data analysis techniques. However, these approaches, although still useful today, focus on producing knowledge about known data (such as data descriptions and distributions, correlations, causality, trends, etc). With the emergence of Machine Learning, a paradigm shift has occurred to predict the behaviour of unseen data. Machine Learning (ML) is a field of Artificial Intelligence (AI) defined as "a computer program, learning from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" (Mitchell, 1997). It uses mathematical and statistical approaches to build algorithms capable of 'learning', of generalising behaviour in data to improve their performance in solving a task, without being explicitly programmed. Compared to classical statistical analysis where variables are compared in pairs, ML allows non-linear relationships between attributes to be detected and all attributes to be considered together. Machine learning is usually divided into three categories, based on

[1] `https://www.health-data-hub.fr/` (last visit date: December 5, 2023)

the feedback given to the algorithm during learning: supervised learning, unsupervised learning and reinforcement learning. Typically in healthcare, supervised learning is used as data can be labelled (Jiang et al., 2017; Rajpurkar et al., 2022). This learning type refers to training algorithms with datasets containing both inputs and labels, the expected outputs. Through iteration, algorithms will learn to predict outputs based on the inputs and correct their error by comparing the predictions with the labels, optimising a loss function on unseen data. An optimal, well-trained algorithm will then be able to predict the output for new inputs that it has never seen before. Flexible and well-tuned models often outperform simpler statistical models in performances on unseen data, and ensembles of different models often further outperform individual models (Rokach, 2010). During this thesis, only supervised learning on tabular data was used and taken into consideration. It derives from the current primacy of this scope in real-world healthcare applications (Sidey-Gibbons and Sidey-Gibbons, 2019).

The paradigm shift, from the production of knowledge to the search for predictive performance on unseen data, has added value to the understanding of data. Indeed, being able to predict unseen data correctly implies an understanding of the invisible and underlying patterns in the data. Linking all attributes together and discovering non-linear relationships between them can unlock a new approach to data analysis. The notion of prediction, particularly in precision medicine, as an aid to diagnostic and therapeutic decision-making, is also a significant and powerful contribution of ML. It can provide healthcare professionals with additional assistance in their day-to-day practice. Moreover, following visible success in predictive tasks from all medical specialities, ML techniques have attracted the interest of healthcare professionals, clinicians and health researchers (Jiang et al., 2017; Sidey-Gibbons and Sidey-Gibbons, 2019; Rajpurkar et al., 2022; Batko and Ślęzak, 2022).

Unfortunately, the quest for performance in predicting unseen data has led to the creation and use of increasingly complex ML models, pushing their transparency and interpretability into the background. Models with few structural restrictions, such as neural networks and gradient-boosted trees, have replaced models that are more interpretable because they are structurally restricted, such as decision rules or linear models. In intrinsically interpretable models - also known as glass-box or open-box models -, the individual components can be analysed individually and easily linked to understandable concepts. For example, the coefficients of a linear regression can be extracted, linked to specific inputs and interpreted more or less in isolation. Also, a terminal leaf of a decision tree can be described by a sequence of binary operations leading to a specific prediction. In contrast, complex, closed-box models are difficult to analyse into understandable individual components, making them more complex to interpret and explain (Molnar, 2022a). Since they make medical decisions for patients' health, medical professionals need to understand and evaluate the predictive model, so that they can assess and appropriate the predictions and explain to patients the decisions made using a tool incorporating Machine Learning. The challenge is even greater given that most healthcare staff are not Machine Learning specialists, which means that interpretations and explanations need to be simplified.

In recent years, research into Explainable Machine Learning (XML) has re-emerged in response to the increasing demand for prediction explanations and transparency in ML models. The terms Explainable IA (XAI), Interpretable ML (IML), Interpretability and Explainability often overlap and multiple definitions and uses of each terminology co-exist

in the literature (Broniatowski, 2021). One consensus is that these fields seek to extract insights from the models and understand the reasons behind the predictions made. Alongside intrinsically interpretable models, post-hoc methods have emerged to explain/interpret closed-box models. Post-hoc methods refer to methods applied to already-trained ML models, with no impact on the training. Especially, agnostic post-hoc explanation method brings together all the methods that can be applied whatever the ML model, as opposed to model-specific explanation methods. In addition, post-hoc methods can have a global or local scope, depending on whether they explain the general behaviour of the model or an individual prediction. This thesis focuses especially on the local post-hoc explanation methods, to explain local prediction and provide healthcare professionals support when using ML systems.

However, the use of explainability in high-risk domains, such as healthcare, is criticised in favour of interpretability and intrinsically interpretable models (Rudin, 2019). The author states that high-risk domains require a way to challenge the model outputs, which in turn requires understanding how the decision was made. Intrinsically interpretable models would be the only way to produce 100% faithful explanations that align with how the model truthfully works, and current post-hoc interpretations would be insufficient because they generally simplify relationships. Well-known limits of current post-hoc explanations methods, like *SHAP* (Lundberg and Lee, 2017) and *LIME* (Ribeiro et al., 2016), also mentioned how these methods are time-consuming and consider attributes as independent, creating unwanted behaviours when attributes are correlated - which is very common with real-world data (Garreau and von Luxburg, 2020; Kumar et al., 2020). Explanation methods are also not widely evaluated and compared in the literature due to their subjective nature, creating a gap in understanding their strengths and optimal environments for use (Miller, 2019). The criticism from Rudin (2019) therefore focuses on the choice of models, omitting the potential contribution of closed-box model explanations compared to intrinsically interpretable models and their interpretation (Molnar, 2022b). Interpretations of intrinsically interpretable models do not generally carry the same information as post-hoc explanation methods. The former will focus on the structure of the model itself. In contrast, post-hoc explanation methods can provide information on the importance of variables and their impact on predictions, at the global level of the model or locally for each prediction or minority sub-groups of data.

Although intrinsically interpretable models can perform similarly to complex models and should be preferred wherever possible, ML models can become closed-boxes for reasons other than performances alone. In medicine, many models are considered closed-box, because they are proprietary and therefore inaccessible (Petersen et al., 2022). In the case of proprietary models, even a decision tree would be a closed-box model, which needs to be explained. Intrinsically interpretable models can also become closed-box as their complexity increases, decreasing their interpretability. Post-hoc agnostic explanation methods are then the only solution currently available to provide explanations to users and attempt to explain the reasons for a prediction. The research and optimisation of post-hoc and agnostic explanation methods is therefore an important area of research for managing the current limitations of local post-hoc methods and for guaranteeing the reliability of ML models. The explainability of ML models is a crucial point in the pursuit of "safe, robust, reliable and fair ML systems" (Petersen et al., 2022), whatever the current or future level of legal requirement for explainability.

1.2 Problem Statement

Given the context, criticisms and limitations mentioned above, we can raise the following problem: **How ML explanations can be provided to non-ML-expert medical professionals for predictive analysis?**

This can be broken down into three key issues, which are the main focus of this thesis:

1. How can we manage local post-hoc explanations behaviours, strengths and limitations?

2. How explanations can highlight the behaviour of the analysed data through the predictive model by identifying localities (prototypical or atypical)?

3. How can we efficiently provide explanations to medical end-users, who are not ML-experts?

1.3 Manuscript Outline

This manuscript is divided into four chapters, three with proposals to address the above issues.

Chapter 2 presents a review of the literature on explainability. We detail the definitional issues surrounding explainability and interpretability and the classification of the various existing methods. We describe the intrinsically interpretable models and post-hoc explanation methods in the literature, especially those relevant to this thesis, as well as their limitations and the existing evaluation and comparison metrics. We then discuss the various uses of explanations as a final tool or as new data to be explored. Finally, we provide an introduction to the user tests carried out to evaluate the explanations for end-users.

Chapter 3 covers our contribution to the coalitional explanation method improvement and the evaluation and comparison of local post-hoc attributive explanation methods. We focus on the first issue by taking into account the current limitations of local attributive methods to optimise one and highlight each local attributive XML method behaviour through our evaluation. We complete our contribution with two medical examples to illustrate the explanations of local attributive methods.

Chapter 4 presents our work on local explanations analysis as a new data space. To gain insight into the second issue, we define our analysis framework centred around explanation clustering and evaluate it for several clustering methods against clustering and raw data analysis. We illustrate how explanation clustering can be applied to medical data and used by healthcare professionals.

Chapter 5 details our contribution to the last issue, with the implementation of explanations in medical applications. This contribution focuses on how to efficiently display and analyse explanations to provide them to end medical users and on an experimental protocol for carrying out user tests to evaluate the contribution of explanations for healthcare professionals.

Finally, Chapter 6 summarises our contribution to the XML research domain and discusses the main research perspectives.

Chapter 2

Explanations of Machine Learning predictions : Literature review

Contents

In the last decades, Machine Learning (ML) has become a standard for making decisions, automating tasks, and exploring and analysing data. More and more domains use these techniques for scientific applications, recommendation systems, fraud detection, speech recognition or virtual assistants. Successful applications can be seen in ecology with predictions of water reservoir levels (Obringer and Nateghi, 2018), in streaming platforms with film recommendations (Zhou et al., 2021), in biology for monitoring marine wildlife (Dujon et al., 2021), in banking to prevent card fraud (Ali et al., 2022), in health care for preventing diseases like heart failure (Nagavelli et al., 2022), improving imaging diagnosis (Litjens et al., 2017), or better-managing patient flows (El-Bouri et al., 2021). However, the inability to understand how ML works and makes predictions -called the closed-box effect- becomes alarming in sensitive fields. The risks resulting from an error in the predictive model do not have the same impact and are not considered in the same way in medical, judicial or financial domains (Lipton, 2018). A mistake in a medical robot performing surgeries will have far more consequences than an error in a film recommendation. A new field of study, the interpretability/explainability of models, has therefore emerged, focusing on the problems of understanding ML predictions and closed-box models, with multiple overlapping terminology: *eXplainable ML* (XML), *Interpretable ML* (IML), *eXplainable AI* (XAI), Interpretability, Explainability.

This chapter covers the main concepts concerning the explainability of modelling predictions, its uses and user evaluation of explanations. Section 2.1 discusses the definitions of explainability and interpretability. It presents different explainability methods and ways of comparing them. Section 2.2 shows how local explanations are used for their intended purpose or as new data. Finally, Section 2.3 discusses the evaluation of local explanations by users, both for the construction of user experience and the results of the literature.

2.1 How to explain predictions ?

2.1.1 On defining human comprehension of Machine Learning

The principal assumption behind explainability and interpretability is that more transparent, interpretable and explainable models lead to users understanding and trusting the intelligent system (Miller, 2019). It can also help to uncover causal structure, get helpful information about and from the model, or offer legal right to prediction explanations (Lipton, 2018).

But since the rise of explainability and interpretability, one main criticism is the lack of one formal definition, with multiple definitions arising at the same time (Lipton, 2018; Murdoch et al., 2019; Molnar, 2022a; Flora et al., 2022). In linguistics, *"interpretation"* aims to enable communication between people who speak different languages, whilst *"explanation"* attempts to describe the causes, context and consequences of a thing the way it is, or of a process the way it takes place. Both can be complementary, as explanations may need interpretation to be understandable. In this direction, Roscher et al. (2020) describe interpretability as *"the capability of making sense of an obtained ML model"* and explainability as *"revealing the underlying causes to the decision of an ML method"*. However, both terms are often used interchangeably although they do not have the same semantic meaning. Miller (2019) then defined explainable AI as *"explanatory agent revealing underlying causes to its or another agent's decision making"*, interpretability as

"the degree to which an observer can understand the cause of a decision" and equate interpretability and explainability. This definition of interpretability mixes the linguistic definitions of explanation and interpretation and the explainable AI one introduces the notion of another agent involved in the process of explaining. However, defining and measuring human/observer understanding raises a significant problem. Kim et al. (2016) shortcut this drawback by clarifying interpretability as *"the degree to which a human can consistently predict the model results"*. On the other hand, some papers defined interpretability as understandability or intelligibility, with open- or glass-box models as opposed to incomprehensible closed-box models (Lou et al., 2013). Interpretability then refers to understanding how the model works based on its structure. A third approach was defined by Montavon et al. (2018) and Ribeiro et al. (2016) by using interpretability for the general strategy of understanding a model when explainability is used for a single prediction. Montavon et al. (2018) defined an interpretation as *"the mapping of an abstract concept (e.g. a predicted class) into a domain that the human can make sense of"*, where a domain can be images or texts, that human can readily understand, and an explanation is *"the collection of attributes of the interpretable domain, that have contributed for a given example to produce a decision (e.g. classification or regression)"*. Ribeiro et al. (2016) also defined *"explaining a prediction"* as *"providing a qualitative understanding of the relationship between the instance attributes and the model predictions"*. Finally, a more global definition of interpretability was set by Murdoch et al. (2019) as the *"extraction of relevant knowledge from a machine-learning model concerning relationships either contained in data or learned by the model"*.

By taking into account semantics and established definitions, especially Roscher et al. (2020); Lou et al. (2013), in this thesis, I will use terms as follows:

- *explainability* for extracting knowledge about an interpretable or uninterpretable model and its decisions and explaining its behaviour
- *interpretability* for open-box models -or glass-box models- that can be readily understood without additional method
- *explanations* for explanations of individual predictions

To classify and categorise the numerous methods that have appeared over the years, a classification has been established based on three main characteristics (Lipton, 2018; Burkart and Huber, 2021; Vilone and Longo, 2021; Linardatos et al., 2021):

Intrinsic or post-hoc. Each method can be classified according to whether explainability is obtained by the model itself or by an external method applied to the model. Intrinsic means that models are transparent, glass-box, due to their inner structure, such as linear models, decision trees or decision rules. This refers to what has been defined as interpretability. On the contrary, post-hoc explainability refers to methods applied to the models after training to extract knowledge about their behaviour. These methods can also be applied to intrinsic interpretable models. Methods based on Shapley Values (Štrumbelj and Kononenko, 2008) or Permutation attributes (Fisher et al., 2019) are part of post-hoc methods.

Model-specific or agnostic. This distinction makes it possible to separate methods according to whether they can be applied to all models or just one. A model-specific method can be applied to only one model type and used only in that specific case. The intrinsic interpretability of models is model-specific since they use properties of the models themselves, such as weights in the case of linear regression, and cannot be applied to all

other types of models. In contrast, a "model-agnostic" method is usually applicable to all predictive models, as it analyses the input/output pairs of the model to produce explanations. These methods do not have access to the model properties and are applied after the model has been trained (i.e. post-hoc methods).

Local or global methods. Methods can be classified based on how they consider the model to explain. The global approach focuses on understanding the entire model at once. This approach requires knowledge of both the trained model and the training data. It is based on a holistic view of the model and aims to understand the importance of each attribute and how they interact to produce a prediction. In application, this approach is challenging because human beings cannot mentally picture space in more than three dimensions. To overcome this problem, the local approach focuses on a point of interest, a single instance, and describes the model locally around this point to reduce the overall complexity of the model. This approach makes it possible to understand the factors influencing a prediction and can be more accurate than global explanations (Ribeiro et al., 2016; Guidotti et al., 2018).

These categories efficiently describe all the actual explainability methods, even if they overlap in some characteristics. Two main methods classes can be identified in the literature: intrinsically interpretable models (which are global, model-specific and intrinsic), and post-hoc agnostic methods.

2.1.2 Intrinsically Interpretable Models

In this section, we briefly discuss the "intrinsically interpretable models" as they are used to build local post-hoc model-agnostic explainability methods and can be a first step into interpretability and explainability.

The field of intrinsically interpretable models existed for many years before closed-box problems appeared in ML. Linear regression and rule-based ML models are standard pre-21st century models and fall into this category due to their inner structure. Intrinsically interpretable models are often defined as the easiest way to achieve explainability (Molnar, 2022a).

In ML, *Linear regressions* model the relationship between attributes and a target by a linear predictive function. For each instance of the dataset, the relationship is described as a sum of weighted attributes:

Definition 2.1. *Linear Regression*

Let x_i be the value of the $i-th$ attribute, β_i the learned coefficient for the $i-th$ attribute, β_0 the intercept and ϵ the remaining error. The target y is defined as:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n + \epsilon$$

Coefficients can be used to interpret the model and how each attribute influences the model predictions. For classification tasks and more complex data, adaptations of linear regressions exist as *Logistic Regression*, *Generalized Linear Models* and *Generalized Additive Models*. Thanks to their linearity, all are easy to understand and intrinsically interpretable. Logistic regression also offers a probability for categories. These models are widely used in academic fields such as medicine, biology, epidemiology, environmental sciences and behavioural science, where the need to interpret the link between attributes and targets is strong (Hastie et al., 2009).

Tree-based algorithms, such as Decision Trees, represent the relationship between attributes and labels by partitioning the data space. One attribute recursively partitions the data space into two by maximising the dissimilarity between the two new partitions. Based on these thresholds, called nodes, the instances are divided into groups called leaves, with each instance belonging to only one leaf. These partitions can be graphically represented by one tree, starting with the root node and following each node like a boolean condition until a final leaf that defines the prediction. This characteristic makes Decision Trees interpretable, easy to understand and intuitive for human comprehension.

Figure 2.1 represents a decision tree trained on the well-known Iris Dataset, with default parameters from the sklearn library[1]. Iris Setosa, in orange, can be separated from other classes based on the petal length. Iris Virginica (purple) and Versicolor (green) are then defined based on the petal width and petal and sepal length. Only the sepal width attributes were not used to build this decision tree.

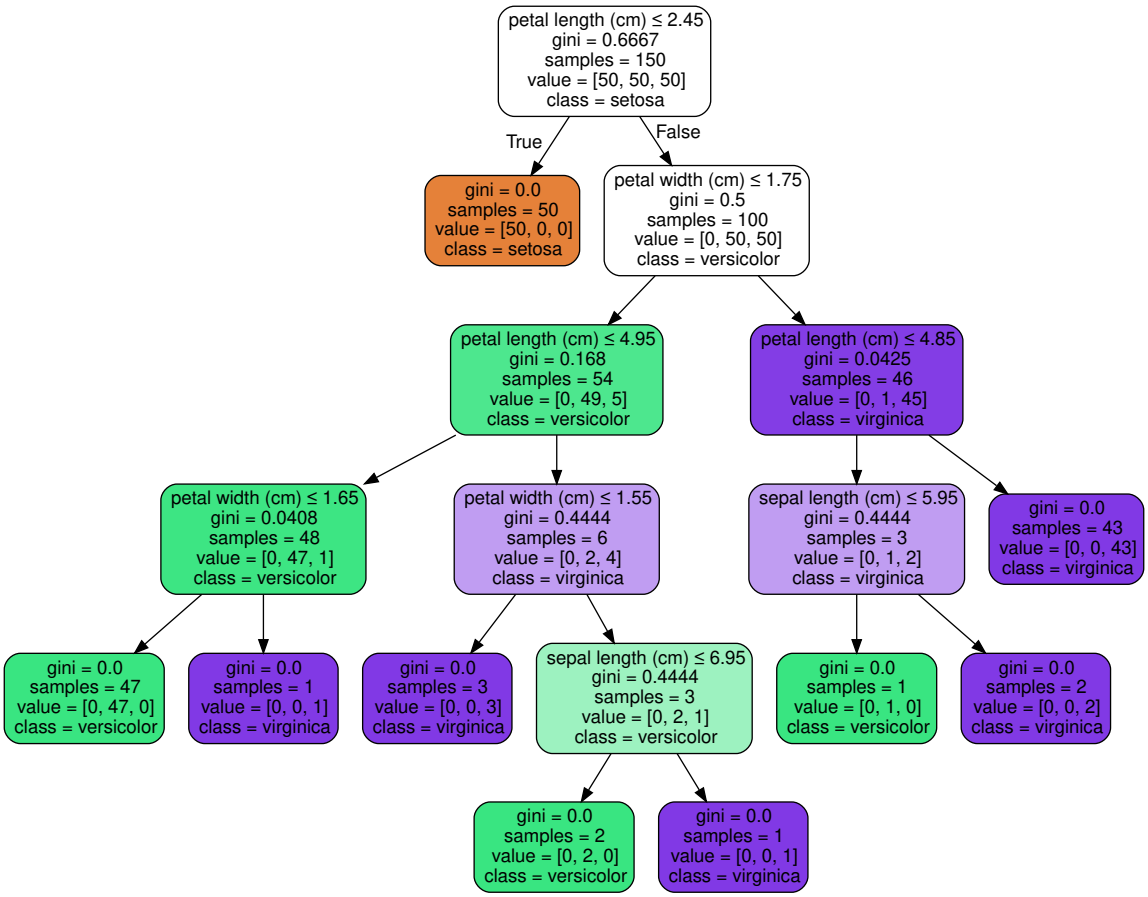


Figure 2.1: Example of a Decision Tree trained on the Iris Dataset.

Decision Rules are based on combinations of IF/THEN conditions to make classification predictions. Like with conditionals in algorithms, the structure can be defined as:

"IF the condition is met THEN a certain prediction is made."

Multiple conditions can be combined with an AND. This structure is intuitive for humans as it is close to the natural way of reasoning, with only relevant attributes used for the

[1] `https://scikit-learn.org/stable/index.html`

modelling. To enhance comprehensibility, conditions can be hierarchised in a decision list or organised in a non-overlapping decision set.

Table 2.1 describes decision rules computed by the *SkopeRules* algorithm [2] on the Iris Dataset (Gardin et al., 2019). Skope Rules is an algorithm that computes decision rules based on random forest. To represent the three classes, 3 out of 4 attributes are used: petal length and width and sepal length. Again, sepal length is not used, as for the decision tree. Precision and Recall are metrics about the rules.

Label	**Rules**	**Precision**	**Recall**
Setosa	Petal length <= 2.45	1.0	1.0
Versicolor	Sepal length > 4.95 & Petal length <= 5.35 & Petal width <= 1.75 & Petal width > 0.80	1.0	0.97
Virginica	Petal width > 1.65	0.95	0.92

Table 2.1: Example of Decision Rules for the Iris Dataset.

Unfortunately, there are limitations to the intrinsic interpretation of these models and their use in modelling complex data (which is usually the case with real-world data). Their internal structure, which allows them to be interpretable, also becomes a weakness. For linear models, their linearity allows to model only linear relationships and dependence or correlation between attributes are not taken into account. Models also have limited performances caused by this restrictive hypothesis, and complex adaptations of Generalised Additive Models (GAM) and Generalised Linear Models (GLM) are made at the cost of reduced interpretability. The analysis of weights can also be counter-intuitive since each weight depends on the importance of all the other attributes. This analysis is further complicated with logistic regression models since the interpretation is multiplicative rather than additive. Unlike linear models, Decision trees and rules are appropriate when the relationship between attributes and labels is non-linear or when attributes interact. Only to fail to model linear relationships as numeric attributes are, implicitly or not, discretised. Decision Trees can also be unstable due to their high reliance on training data, as small changes in the data might completely change the final Trees (Breiman et al., 1984). Another limitation is that complex data often leads to complex models, even for intrinsic interpretable ones. This results in trees with significant depth, a large number of weights to analyse for linear models or numerous rules with a wide range of conditions for decision rules models. This complexity can hinder the interpretation of these models, which then become *closed-box* models. Moreover, intrinsically interpretable models can also be *closed-box* models when they are proprietary -i.e. the one creating the model reserves the rights to use, modify or share. Then, only predictions are reachable and the structure of the model is unknown, thus non-interpretable. Finally, intrinsically interpretable models are not optimal for high performance. They are outperformed by more complex and less interpretable versions like Boosted Trees Ensemble, Random Forest, Neuronal and Convolutional Networks.

While intrinsically interpretable models can provide a first approach and an excellent overview when modelling data, the known limitations and the need for high performances lead to focus on another type of explanation applicable to all models: post-hoc agnostic methods.

[2] `https://github.com/scikit-learn-contrib/skope-rules`

2.1.3 Post-hoc Agnostic Explainability Methods

Post-hoc agnostic explainability methods have become more important in the literature over the last ten years (Linardatos et al., 2021). They aim to explain the relationships between attributes and prediction, an intuitive approach to human understanding. Each prediction is seen as a result of the impact of all the attributes in the data as models are trained on data to make predictions. One of the strengths of these methods is that only the prediction function and the data are requested. Models remain closed-box, facilitating applications in fields where the model cannot be shared for security reasons.

With post-hoc methods, models can be studied globally or locally, depending on the goal and the information wanted. Global methods aim to describe the average model behaviour over all the instances to understand the data behaviour with the model. They are mainly based on analysing the effect of each attribute over all the predictions. For example, Partial Dependence Plots show the marginal effect of one or two attributes on the prediction (Friedman, 1991) and Permutation attribute Importance shows how predictions change when attribute values are swapped (Breiman, 2001; Fisher et al., 2019).

On the other hand, local methods have been popular in recent years, as explaining each prediction is useful in applications, intuitive and easy to understand. Their popularity also comes from their instance-level accuracy, which allows finer differences to be detected between all instances. Local explanations can take multiple forms, such as attributive influences or explanations by examples and counterfactual instances. Example-based explanations relate to other instances of the dataset to explain the prediction, by selecting similar instances or by computing the minimum changes to switch prediction for counterfactual explanations. In this thesis, we focus on explanations by local attributive influences and methods, as it allows more specific analysis than global ones and as the main goal in the medical field is to understand how ML modelling makes each patient prediction. These methods produce a vector of weights to represent the contribution of each attribute to the prediction for a single instance, called influences. The magnitude and sign of the influence provide information about the strength of this contribution. A high magnitude indicates an attribute with a significant impact on the prediction. The influence sign then defines whether the effect goes in the direction (+) or against (-) the model's basal value - the default prediction for a new instance without any other knowledge about it. These methods can also be additive, meaning that, for each instance, the sum of its influences approximates the difference between the instance prediction and the average prediction of the model over the dataset. In this way, local attributive methods make it possible to explain the prediction and the knowledge learned by the closed-box model, regardless of their correctness. It is then possible to measure the variations in predictions, the attributes responsible for the variations and the magnitude of their influence to compare the results of several models and illustrate how these models have learned.

Figure 2.2 shows an example of local explanations on the Indian Pima Diabetes dataset[3], made with *SHAP* on a trained Random Forest. The prediction that this person, a single instance, has diabetes is 0.86. The attributes contributing the most to the prediction are *Glucose, Pregnancies, BMI* and *Age*. Only the *BMI* attribute contributes "against" the prediction, i.e. for the model, BMI value decreases the risk of diabetes. For the model, high glucose levels are the main reason for the high prediction of diabetes.

[3] `https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database`

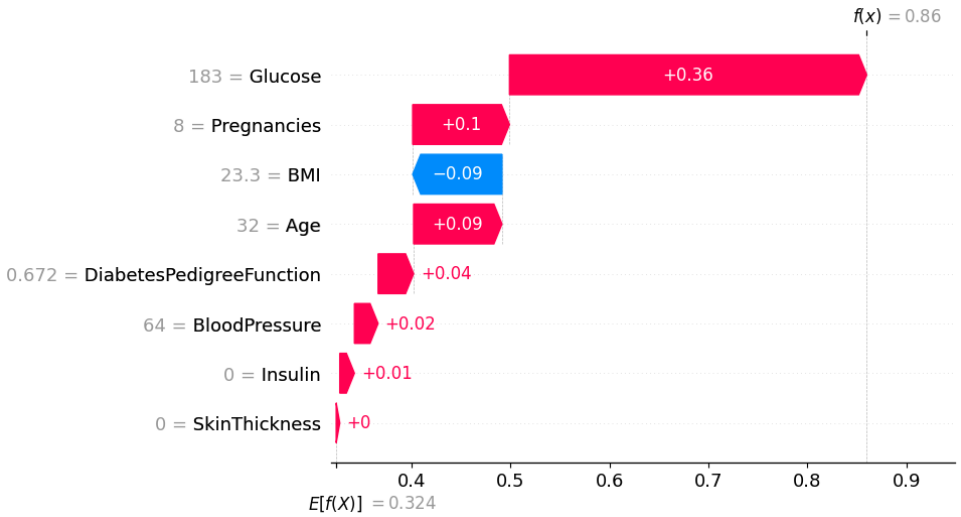


Figure 2.2: Local explanation example for a single instance of the Indian Pima Diabetes dataset.

Shapley Values explanations The first local attributive additive method was based on Shapley values and described in Štrumbelj and Kononenko (2008, 2010, 2014). In cooperative game theory, Shapley values distribute the contribution to the payoff equitably between the players in a coalition (Shapley, 1952). In ML, the gain can be linked to the prediction made by the model and the players to the attributes. The influence of each attribute is computed based on its impact on the prediction for each coalition of attributes. Shapley values are then "the average marginal contribution of an attribute in all possible coalitions [of attributes]" (Štrumbelj and Kononenko, 2010).

Definition 2.2. *Shapley values explanations*

Let A be the set of attributes in the dataset, a_j the j-th attribute, and Δ the difference between the expected prediction for a combination of attributes S and the expected prediction in the absence of data, with $\hat{f}$ the prediction function of the ML model. The influence $\phi_j \in \Re$ of the attribute j for the instance x is defined as:

$$\phi_j(x) = \sum_{S \subseteq A \setminus \{a_j\}} \frac{|S|!(|A| - |S| - 1)!}{|A|!} (\Delta_{S \cup \{a_j\}}(x) - \Delta_S(x))$$

$$\Delta_S(x) = \hat{f}_S(x) - \hat{f}(\varnothing)$$

Thus, for each instance, for each attribute, the Shapley value corresponds to the difference between the prediction with and without this attribute, penalised according to the size of the coalitions of attributes for all possible attributes coalitions. For example, with a 3-attribute dataset $[A, B, C]$, the influence of the attribute A will be :

$$\phi_A(x) = \frac{1}{3}\Delta_A + \frac{1}{6}(\Delta_{AB} - \Delta_B) + \frac{1}{6}(\Delta_{AC} - \Delta_C) + \frac{1}{3}(\Delta_{ABC} - \Delta_{BC})$$

. To compute the prediction with and without data, a data perturbation mechanism is used to simulate the absence of data. For each attribute or group of attributes, values are replaced by another value extracted randomly from the attribute values set. Perturbations

can also be done by sampling from the marginal distribution of the attribute values or by retraining the modelling without the attribute or group of attributes of interest.

The explanation method based on Shapley values is called the *Complete* method. However, this method is expensive to compute, with an exponential complexity concerning the number of attributes in the dataset.

LIME *LIME* is a well-known local attributive explanation method for tabular data, images and text, described in Ribeiro et al. (2016). *LIME* uses local surrogate models to locally approximate a complex closed-box model and, for each instance, explain the influence of each attribute on the prediction. For each instance to be explained, *LIME* generates new data in a close neighbourhood and computes the predictions of these new instances with the closed-box model. An interpretable model, usually a linear model, is trained with the new dataset. The surrogate model is then used to explain the prediction of the instance of interest in the form of a weight vector associating each attribute with its influence on the prediction. The influences are obtained mathematically according to the following formula:

Definition 2.3. *LIME*

Let x be one instance of the dataset, g the surrogate interpretable model, G the set of potential interpretable models, $\Omega(g)$ the complexity of the model g, $\hat{f}$ the predictive function of the closed-box model, π_x the proximity measure between x and the sampled instances z, and $\mathcal{L}$ the cost function between g and $\hat{f}$. The explanation for the instance x is obtained with:

$$\phi(x) = \underset{g \in G}{\operatorname{argmin}}\, \mathcal{L}(\hat{f}, g, \pi_x) + \Omega(g)$$

The aim is to minimise the cost function between the two models to reach a substitution model representative of the original model and simple to interpret. One advantage of the cost function is the ability to assess how reliable the surrogate model is locally in explaining the closed-box model. In addition, the open choice between different surrogate models means that the explanations can be adapted to the users based on the interpretable model they understand best.

The full implementation of *LIME* is available on GitHub: `https://github.com/marcotcr/lime`.

SHAP The SHapley Additive exPlanations method (*SHAP*) (Lundberg and Lee, 2017) is an alternative to *LIME* and the *Complete* method, working on improving computation time and explanation precision, especially for tree-based models (Lundberg et al., 2020). It combines *LIME* (Ribeiro et al., 2016) and Shapley values (Štrumbelj and Kononenko, 2014), along with other methods from the literature (Lipovetsky and Conklin, 2001; Bach et al., 2015; Datta et al., 2016; Shrikumar et al., 2017), in a unique framework to produce local attributive additive explanations. As in *LIME*, the main idea is to create perturbations in the data and to use a linear model to approximate the change in the prediction. Linking local surrogate and Shapley value, data perturbations are employed to simulate the absence of an attribute and avoid retraining the complex model without the attribute of interest to compute the Shapley values.

Definition 2.4. *SHAP Explanation*

Let $\phi_j \in \Re$ be the influence of the attribute j based on Shapley value, M the maximum coalition size, $z' \in 0,1^M$ the vector designating the attributes in the coalition and $\phi_0 = E_X(\hat{f}(x))$ with $\hat{f}$ the prediction function of the closed-box model. The surrogate model g producing explanations is defined as :

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

For each coalition of attributes z', attribute values are replaced with another value from the marginal distribution of the data to simulate the absence of these attributes based on the function $h_x(z') = z$ where $h_x : \{0,1\}^M \rightarrow \mathbb{R}^p$. The function h_x maps 1's to the corresponding value from the instance to explain x. For tabular data, it maps 0's to the values of another instance sampled from the data, based on the data marginal distribution. This means that *SHAP* equates "attribute value is absent" with "attribute value is replaced by random attribute value from data", creating a new dataset Z (Molnar, 2022a).

This dataset Z is used to fit the surrogate linear model g that is optimised based on the following loss function :

Definition 2.5. *SHAP Loss function*

Let Z be the perturbed dataset, g the surrogate model, $\hat{f}$ the prediction function of the closed-box model, h_x the mapping SHAP function, $\pi_x(z')$ the function weighting z'.

$$L(\hat{f}, g, \pi_x) = \sum_{z' \in Z} [\hat{f}(h_x(z')) - g(z')]^2 \pi_x(z')$$

The main difference between *LIME* and *SHAP* lies in how both weigh the samples in the function $\pi_x(z')$. When *LIME* based the weight on the distance between the original instance and samples, *SHAP* is based on the weight the coalition would get in the Shapley value estimation.

Definition 2.6. *SHAP weight kernel*

Let M be the maximum coalition size and $|z'|$ the number of "present" attributes in vector z'. The weight of the coalition z' is defined as :

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)}$$

Finally, based on Definition 2.4 with a linear surrogate model optimised with Definitions 2.5 and 2.6, the influences are the coefficient ϕ_j of the linear surrogate model.

SHAP also provides information about the global behaviour of the model by aggregating local influences. Global and local explanations aim to be consistent together as they have the same foundation. *SHAP* includes an agnostic explainer, *KernelSHAP*, as well as model-specific explainers, such as *TreeSHAP*, *LinearSHAP* or *DeepSHAP* for tree-based models, linear models and deep models respectively.

The full implementation of *SHAP* is available on GitHub: `https://github.com/slundberg/shap`.

K-depth method The *K-depth method* approximates the *Complete* method, based on Shapley Values (Ferrettini et al., 2020a). It aims to reduce the complexity and the computation time while keeping explanations accurate and the agnostic characteristic of the *Complete* method. This method decreases the number of coalitions used to compute Shapley values by considering only coalitions smaller than k. By using only a subset of all the coalitions, the method keeps the interdependence of attributes but eliminates the broadest coalitions that are expensive to compute and have less impact. Influences are defined as :

Definition 2.7. *K-depth method*

Let A be the set of attributes in the dataset, a_j the j-th attribute, k the maximum length of a coalition, Δ the difference between the expected prediction for x for a combination of attributes S and the expected prediction in the absence of data, with $\hat{f}$ the prediction function of the ML model. The influence $\phi_j \in \Re$ of the attribute j for an instance x is defined as:

$$\phi_j(x) = \sum_{S \subseteq A \setminus \{a_j\},\ |S \cup \{a_j\}| \leq k} \frac{|S|!(|A| - |S| - 1)!}{k \times (|A| - 1)!} (\Delta_{S \cup \{a_j\}}(x) - \Delta_S(x))$$

$$\Delta_S(x) = \hat{f}_S(x) - \hat{f}(\varnothing)$$

This method is thus naive since interesting coalitions of attributes can be arbitrarily eliminated.

Coalitional method Another agnostic attributive explanation method based on Shapley values was then introduced to smartly take the interdependence of attributes into account and solve some restrictions of *SHAP*. The *Coalitional* method uses grouping methods such as *Principal Component Analysis* (PCA), *Spearman correlation factor* (Spearman) and *Variance Inflation Factor* (VIF) to select from amongst all the possible combinations of attributes those that would be the most interesting for explanations (Ferrettini et al., 2020a,b). These groups are then used as coalitions to compute Shapley values as in the *Complete* method. The influence of each attribute is defined as its impact on the prediction only on these groups of attributes, approximating the *Complete* method and reducing the computational time. With these grouping methods, groups do not have to be exclusive, meaning that one attribute can be on multiple coalitions. Grouping methods are also defined with a threshold parameter α that changes the number and size of attribute coalition to prioritise a lower computational time or a higher accuracy. Influences are computed similarly to the Shapley values in the *Complete* method, only by modifying the attributes coalitions in parameters.

Definition 2.8. *Coalitional method*

Let G be a pre-computed coalition of attributes, A the set of attributes in the dataset, a_j the j-th attribute, $G_{\{a_j\}}$ the subset of G containing the coalitions of attributes $g \in G$ such as $a_j \in g$, Δ the difference between the expected prediction for x for a combination of attributes S and the expected prediction in the absence of data, with $\hat{f}$ the prediction

function of the ML model. The coalitional influence $\phi_j \in \Re$ *of the attribute* j *for an instance* x *is defined as:*

$$\phi_i(x) = \sum_{g' \subseteq g \setminus \{a_j\}, g \in G_{\{a_j\}}} \frac{|g'|!(|g| - |g'| - 1)!}{\sum_{g \in G_{\{a_j\}}} |g|!} (\Delta_{g' \cup \{a_i\}}(x) - \Delta_{g'}(x))$$

$$\Delta_S(x) = \hat{f}_S(x) - \hat{f}(\varnothing)$$

The full implementation of *Coalitional-based method* is available on GitHub: `https://github.com/kaduceo/coalitional_explanation_methods`.

Limitations of the local attributive explanations methods Although all these local attribution methods are popular or promising, they present limitations regarding complexity, calculation time, accuracy, applicability or their restrictive hypotheses. First, the *Complete* method has an exponential complexity regarding the number of attributes, which stops these methods from being used with large datasets. A well-known limitation of *LIME*, which also applies to *SHAP*, is the restrictive hypothesis on which *LIME* is based, such as local linearity and attribute independence caused by sampling that impacts the precision of both methods (Slack et al., 2020; Garreau and von Luxburg, 2020; Kumar et al., 2020). Biased classifiers can fool explainability methods, whose problem is even more accentuated on *LIME* (Slack et al., 2020). For *LIME* and *SHAP*, defining the locality around an instance of interest can also be a challenge, as the fit of the surrogate model has a significant impact on the accuracy of the explanations (Laugel et al., 2018) as well as their stability (El Shawi et al., 2019; Alvarez-Melis and Jaakkola, 2018). Moreover, computation time is still higher for other models than tree-based models (Van den Broeck et al., 2022). For each method, the choice of the main hyperparameters is then a challenge in itself, especially for non-expert (Garouani et al., 2022): the number of perturbed samples and the kernel width for *LIME*, the data used to create perturbed samples and the number of perturbed samples for *SHAP*, the k for *K-depth* method, the α-threshold for *Coalitional* methods. To the best of our knowledge, no papers explore the hyperparameters of all these methods and their impact on explanations. Finally, explanations from all methods can be misinterpreted, even more with users who are not experts in data science, as explanations are highly linked to the current set of attributes, the coalitions taken into account, the sampling step or the quality of the surrogate model.

Table 2.2 sums up information about local attributive explanation methods. All methods are post-hoc, allow a global approach by combining local explanations and are mostly agnostic. Differences in when to use them lie mainly in the type of data accepted by the methods and the ML tasks they can be applied. On tabular data and classification tasks, *K-depth* and *Coalitional* suggest approaches keeping the attributes interdependence when computing explanations, unlike *SHAP* and *LIME*. *Coalitional* method is therefore long to compute and suffers from hyperparameters complex to optimise. In Chapter 3, especially Section 3.2, we will focus on optimising the *Coalitional* method to offer a strong candidate to rival *SHAP* and *LIME*.

2.1.4 On comparing explainability methods

Because of the subjective nature of explanations, there is no consensus on objective mathematical ways to evaluate the explanations. Comparative studies between local explainability methods are available, such as El Shawi et al. (2019); Ferrettini et al. (2020b);

XML	Post-hoc	Scope	Model	Data	ML Task	Hyper parameters	Attributes dependence	Computation time
Shapley Values	✓	Local Global	Agnostic	Tabular	Classification	None	✓	* * *
LIME	✓	Local *Global* ≈	Agnostic	Tabular Images Texts	Classification Regression	Numerous Complex	✗	*
Kernel SHAP	✓	Local Global	Agnostic	Tabular Images Texts	Classification Regression	Few Complex	✗	**
Tree SHAP	✓	Local Global	Specific	Tabular Images Texts	Classification Regression	One Easy	✗	*
K-depth	✓	Local Global	Agnostic	Tabular	Classification	One Easy	✓	*
Coalitional	✓	Local Global	Agnostic	Tabular	Classification	Few Complex	✓	**

Table 2.2: Characteristics and limitations of the different methods of attributive local explanations

Duell et al. (2021). Most works focused only on *SHAP* and *LIME* when comparing local methods. Duell et al. (2021) compared *LIME*, *SHAP* and *Scoped Rules* (Anchors) using attribute importance ranking. However, they used a single metric with a single prediction model on a single dataset, which limits the generalisability of their results. In Ferrettini et al. (2020b), the *Coalitional* method was compared with *LIME* and *SHAP* considering computation time and accuracy score. Their results showed that their proposal is competitive with the literature. El Shawi et al. (2019) compared six local model-agnostic explanation techniques using custom quantitative measures on two tabular and two text datasets. From these experiments, no single method stood out for all metrics and all datasets. According to the metrics considered, each method has its strengths and weaknesses, the choice being dependent on both the user's goal and dataset content.

One challenge is then to define what is a good explanation and how to show mathematically their relevance. Miller (2019) indicated that evaluating explainability methods is very subjective and that there was no consensus to propose relevant metrics. Nevertheless, the author summarised criteria for good human-friendly explanations such as contrastiveness, social adaptation, focus on the abnormal, truthfulness or consistency with prior beliefs. They presented their work as general guidelines for objectively defining the relevant explanations. Robnik-Šikonja and Bohanec (2018) also defined properties for individual explanations to help characterise good explanations, like accuracy, fidelity, representativity, understandability or consistency. Explanations that comply with these properties can be seen as correct to the model or the data, trustworthy and easy to understand for the end-users. However, all these characteristics are mainly subjective and not defined mathematically, and how to measure them is not unmistakable. The definition of an acceptable explanation can also differ based on the end-user, the application domain, and the objectives when using explanations, making it difficult to objectively assess the quality of the explanations (Miller, 2019).

Other papers defined more precise metrics to evaluate explanations and applied them to use cases. El Shawi et al. (2019) described similarity, bias detection, execution time, and trust as quantitative measures to evaluate explanations. These metrics are generic as

the authors aimed to compare different explainability techniques on several tabular and text datasets. However, only an intuitive description of the metrics is provided, with no mathematical implementation, making it challenging to reuse them.

As we focus on local additive explainability methods producing one vector of attribute influences for each instance, quantitative metrics can be used to evaluate and compare explanations based on their influence vectors, sometimes referred to as attributive-based metrics. In particular, in Nguyen and Martínez (2020), authors defined monotonicity and effective complexity to evaluate explanations quality. Monotonicity is particularly interesting as it assesses the relationship between the values of an explanation and its expectations. Effective complexity relates to conciseness by estimating the minimum number of attributes necessary for one explanation.

Robustness is another frequently mentioned metric in the literature, defined as the capacity of the explanation to be similar when inputs are similar. Several mathematical formulations of this metric exist, based on how authors determine what similarity means and how to compute it (Alvarez-Melis and Jaakkola, 2018). These multiple implementations produce metrics efficient to evaluate each explanation method specifically. However, it can be confusing to compare measures that have not been calculated in the same way or to find inconsistent definitions of the same metrics.

Another approach is to compare the evaluated methods to a baseline and measure the error between the two. This technique compares multiple explainability methods at once, based on the same metric as long as the methods produce similar outputs (Ferrettini et al., 2020a,b; Carmichael and Scheirer, 2021). The error is computed as the distance between the baseline and each method, allowing the use of any existing distance metric that can be applied to all the methods to be evaluated. Although this approach solves the problem of finding a metric relevant to all explanations, it raises the problem of defining a trustable and consensual baseline.

Comparing explainability methods seems to be a problem of its own, as no consensual methodology exists. Multiple metrics are subjectively defined in the literature, sometimes without mathematical translation (Robnik-Šikonja and Bohanec, 2018; Nauta et al., 2023). By contrast, some metrics, such as robustness, have several mathematical descriptions, making it difficult to use them optimally. Another challenge also lies in finding a metric that applies to all the methods to be compared, especially as metrics mainly measure the explanations themselves and not their usability or practicability. The lack of unified metrics and the differences between explainability methods mean that only similar ones can be easily benchmarked, thus increasing the complexity of selecting metrics and evaluating and comparing explanations (Alvarez-Melis and Jaakkola, 2018; Nauta et al., 2023; Yeh et al., 2019). Nevertheless, comparisons of the different XML methods are necessary to assess their performances and the limits of their use. Explainability methods deserve to be studied to make a reliable assessment of the strengths and weaknesses of each method and of each approach - local or global, attributive or by example, in particular. In Chapter 3, Section 3.3, we will focus on evaluating the local attributive methods in the most exhaustive possible way, based on a large collection of datasets, multiple ML models and metrics evaluating both explanations quality and practicability.

2.2 How explanations are used ?

The rise of the local attributive explainability methods previously seen has led to numerous real-life applications, particularly in hospitals and care centres. Using ML capabilities in healthcare provided an alternative for cost-effective and sustainable healthcare systems as ML can outperform human analytical thinking (Rao et al., 2022). A specific characteristic of the medical field is that it is guided by moral principles such as beneficence, respect for human autonomy, prevention of harm, justice, privacy and transparency, which medical ML must respect. Existing regulations already require safety, robustness, reliability, privacy, security, transparency, explainability and nondiscrimination properties (Petersen et al., 2022). This section explores two areas of explanation use: explanations as a final tool and explanations as new exploration data.

2.2.1 Explanations as a final tool

Decision support tools have then been improved through explainability so that healthcare staff can understand the decision provided. Antoniadi et al. (2021) retrieved almost one hundred scientific publications in major conferences and journals about Clinical Decision Support Systems (CDSS) with XML methods. The authors focused on the context of XML use in CDSS. They defined that CDSS must be trustworthy, easy to understand and positively augment the human decision-making process to be effective for practical use. Explainability is perceived as essential for achieving these objectives, even if scepticism about these new techniques persists. The challenge remains how to present explanations in an informative, efficient and clinically meaningful way. In Rao et al. (2022), *LIME* and *SHAP* appeared to be the best way to implement explainability in healthcare as they are model-agnostic and local, making both model and predictions understandable. In the medical context, with applications for medical professionals, local explanations match the need for information about each patient individually. In the literature of *SHAP* and *LIME* applications in healthcare decision support tools, local explainability is also used with multiple aims more than understanding the prediction for one patient: uncovering risk factors for one disease and comparing them against the known literature (Barda et al., 2020; Jiang et al., 2023; Monsarrat et al., 2022), explaining outcomes coherently by selecting informative explanations (Oh et al., 2021) or defining patient profile sharing similar characteristics (Excoffier et al., 2022b; Cooper et al., 2021; Lee et al., 2022). These uses also appeared in medical research, introducing local explainability as a tool for understanding the model, comparing the explanations with the medical literature and retrieving potential new information about a disease.

However, providing individual explanations without a more general context seems insufficient to improve the effectiveness of the user's decision-making. Indeed Weerts et al. (2019) showed that displaying local explanations along with a prediction for a single instance did not significantly enhance the utility for the user as opposed to prediction alone. Moreover, Zhang et al. (2020) indicated that local explanations alone slightly decreased the user understanding compared to showing model prediction alone (as users can misinterpret the explanations). Moreover, knowing all the local explanations of a dataset does not guarantee a complete data understanding since there are as many explanations as instances in the original raw dataset, with the difficulty of finding explainability patterns in this new dataset. It may be helpful to provide a broader view to help understand a particular situation in the dataset, to know whether it is a usual case or a more atypical

one.

2.2.2 Explanations as new exploration data

With the idea of going one step further than just the explanations provided by explainability methods, explorations of explanations have appeared, centred either on attributes or instances. The former shows the global effect of a single attribute or the interactions between at least two attributes on each dataset instance, as done in Barda et al. (2020). It focuses on attributes of the dataset to understand how the model links the raw data and the prediction, based on the explanations. It gives a global view of the model understanding of the dataset and can help assess explanatory trends for a given population.

With LIME, Ribeiro et al. (2016) also introduced a method for recommending instances from explanations, the Submodular Pick method, called *SP-LIME*. Based on a global measure of attribute importance that uses explanations, the instances that best cover the attributes in the dataset are selected to highlight a set of representative instances, provide global information and help users trust the model and its predictions. Sangroya et al. (2020) proposed an amelioration of *SP-LIME* by guiding instance selection through Formal Concept Analysis. The idea of selecting instances to understand global behaviour is applied for glaucoma analysis in Kamal et al. (2022). *SP-LIME* select four representative instances, one for each different glaucoma severity level. Authors found that *SP-LIME* provides trustworthy and understandable results for medical experts so that clinicians and patients can understand the decision-making process and the risk-factor characteristics for different glaucoma severity levels.

One limit of these two approaches is the inability to detect groups of instances sharing common or conversely very different characteristics. That is why the instance-centred approach can help the user to contextualise a new particular instance explanation with a larger group. To offer users a global view of explanations, papers have focused on aggregating instances and describing the explanation groups created (Alkhatib et al., 2023; Excoffier et al., 2022b; Cooper et al., 2021; Lee et al., 2022). Another advantage of this approach is the ability to suggest a restricted number of instances to analyse as a priority for end-users. Depending on the selection method used, it would then be possible to highlight the representative instances with significant explanatory power on one hand and the principal instances whose behaviour differs from the general behaviour on the other. These instances could give context to the user to better understand the explanations.

One well-known method for aggregating similar data is clustering. These techniques create clusters/subgroups of data related to the distance between them. Clusters can help to find relationships between instances based on their similarities. Based on a COVID-19 dataset, Cooper et al. (2021) tried to identify better clusters based on *KernelSHAP* values. Rather than clustering the original dataset, called raw data, they trained a classification model, computed the *KernelSHAP* values for each instance and performed DB-SCAN clustering on these influences. They showed better identification of clusters with influences than raw data and graphically pictured the cluster differences using UMAP, a well-known reduction dimension technique. With clinical and biological data from COVID-19 hospital patients, Excoffier et al. (2022b) uncovered the COVID-19 typology of patients to identify those most at risk of aggravation during their hospital stay. ML combined with Explainability methods was used to highlight the most significant attributes and build an aggravation risk score. Then, clustering techniques on explanations aggregated patients

and defined three clusters of patients that appear to be consistent with three distinct risk-score levels. Instance recommendations based on the medoid of each cluster also allowed an in-depth study of each subgroup's characteristics.

These two papers explored the hypothesis of using influences to acquire more knowledge about the data on specific medical examples. However, no article formally evaluated the contribution of explanation clustering in general. Despite their positive conclusions, these papers only used a single dataset with a single XML method, without generalising the approach or comparing findings with other local attributive XML methods.

Another approach for aggregating instances was proposed in Alkhatib et al. (2023) and used pattern mining methods to create characterisation rules. Association rules mining was applied to local explanations to define some aggregation of instances through their shared characteristics. Results showed better results than other methods creating discrimination rules from explanations available in the literature. However, the experiments did not compare the rules derived from the explanations with the rules from raw data to assess whether using explanations was beneficial.

At the margin of this thesis, research also exists on the use of explanations in the context of attribute selection. Multiple papers (Man and Chan, 2021; Verhaeghe et al., 2023; Xiaomao et al., 2019; Liu et al., 2022) introduced how to use local explanations for attribute selection, by proposing new framework or methods. Methods were compared to well-known attribute selection methods like the Mean Decrease Accuracy approach. Results showed that the explainability methods produced better outcomes for attribute selection than the methods in the literature.

Finally, with the rise of explainability applications, ML research looked beyond simply explaining the ML model. Several papers in the last year have covered use cases using ML explainability for multiple purposes: attribute selection, clustering, and instance recommendations. Influences are considered like new inputs for finer analysis, either directly in the ML pipeline with attribute selection or afterwards to gain a more in-depth and concise understanding of the ML model and the underlying data. In Chapter 4, with the idea of using explanations as a new data space to explore, we propose a first global data exploration approach based on clustering and evaluate it with a large collection of datasets and multiple clustering algorithms.

However, the proliferation of concrete uses for explanations raises the question of the lack of formal user testing and evaluation of explanations from the point of view of their use by users. Little to no user testing is also available in the medical field for the use of explanation, and even less when adding explanation exploration like clustering or explanation selection.

2.3 Are explanations useful for end-users ?

Among all the articles on explainability via local methods, only a minority belongs to the user testing category. The main focus is still on creating new approaches, evaluating them and applying them. Most XML methods are evaluated only through proxy tasks (i.e. without human implication) with only 22% of XML papers from major conferences including human user study and 23% of them with application-domain experts -around

5% of the analysed papers- (Nauta et al., 2023). Even if proxy-task is time and cost-saving, easier to set up and easily scalable (Doshi-Velez and Kim, 2018), that kind of evaluation cannot evaluate the pertinence of explanation for users in real applications. Then, Antoniadi et al. (2021) stated that user tests and studies to understand user needs were insufficient in the literature, especially in the medical field. To bridge the gap between XML research and real-world application, Srinivasan and Chander (2020) also recommended including humans in the loop and adopting user-centred approaches in 4 out of 6 recommendations. The development of adequate real applications requires the implementation of user experience evaluation to improve the user experience.

XML tests involving humans can be separated into two types (Doshi-Velez and Kim, 2018): application-grounded evaluation and human-grounded evaluation. Application-grounded evaluation involves domain experts in real-world applications to evaluate if the XML method works with the end-users for a particular task and prove that the system delivers on its intended tasks (Antunes et al., 2008). In contrast, human-grounded evaluations use a simplified task with users who are not necessarily domain experts. This evaluation is suitable for testing the general quality of explanations with a large pool of users.

Building a user experience requires several framework elements to be defined according to the research question so that the study is relevant and appropriate (Nunes and Jannach, 2017; Chromik and Schuessler, 2020):

- *What data are collected?*: Quantitative, qualitative or both. Quantitative can be objective metrics like the accuracy of participant prediction, score before or after exposition to explanations, the answer time with or without explanations, or subjective metrics using Likert-scales to assess participants' opinions about the usefulness of the explanations, their understanding, their interest, and their acceptance. Qualitative evaluation can be performed with open-response questions, direct or participant observations, interviews, written documents, and videos.

- *How to define groups of users?*: a single group, one group with and one group without the subject of study, groups with multiple alternatives. A well-used distinction is then made with studies between-subjects that evaluate the differences between groups of participants and within-subjects studies assessing the differences within participants using multiple alternatives.

- *How to evaluate users' expertise level?*: expertise in ML, explainability, of the dataset; measured by Likert-scale, by the users or by the study supervisor. Expertise level can be significant to evaluate, as experts and novices may have different preferences regarding the provided explanations, as shown in Ramberg (1996).

Several user experiments were conducted to assess the impact of local post-hoc explanation methods on domain experts. We focus on four human-grounded experiments, including two in the medical field, summarised in Table 2.3. Weerts et al. (2019) aims to show if local post-hoc explanations, here *SHAP* explanations, are helpful for domain experts to assess the correctness of positive prediction. Experiments were conducted with 159 students with basic knowledge of explainable ML, split into three groups. User tasks were evaluated through the qualitative analysis of participant written reasoning with and without explanation and the quantitative analysis of participant performances. Results showed no significant differences in performances when explanations were available, even though the explanations influenced the participants' reasoning. However, some limits

arise as only *SHAP* was evaluated among all local post-hoc explainability methods and the task was a simplified task performed by undergraduate or graduate computer science students. The experiment setup does not seem to suit the initial purpose of the article, which was to assess the impact of the explanations for domain experts. In Jesus et al. (2021), a payment fraud detection task performed by three domain experts was used to evaluate and compare three local post-hoc explainability methods, *TreeSHAP*, *TreeInterpreter* and *LIME*. Tests were performed in three steps, with different access to information: data only; data and modelling prediction; data, modelling prediction and explanations. Explanations were displayed after a transformation into natural language, with the name of the attribute and its original value and a square of colour based on how they contribute to the fraud risk: green for negative contribution as they lower the risk score, red for positive contribution increasing the risk. Explanations were not in their original *attribute-contribution/influences* form and the value of the contribution was not mentioned in the user interface. Participants were evaluated through quantitative objective and subjective metrics on all steps: decision time, participants' accuracy, recall, false positive rate and participants' perception of explanations usefulness, relevance and pertinence to faster decision using 5-point Likert scales. Results showed that adding modelling predictions and explanations significantly reduces the decision time, especially with *TreeInterpreter*. Explanations also increased the participant's agreement over the same data. However, the participants' accuracy dropped significantly when the modelling prediction was added to the data, and neither predictions nor explanations significantly enhanced the user efficacy. Each explainer was also perceived differently by participants in terms of relevance, usefulness and diversity. *TreeInterpreter* was the explainer with the most positive answer while *LIME* had significantly worse results than *SHAP* and *TreeInterpreter* when asking if the explainer helped the user review faster. However, the results were mixed, perhaps due to the limited number of participants, their very high level of expertise, the distinct data presented to each participant and the presentation of local explanations without the influence values for each attribute.

In the medical field, Diprose et al. (2020) and Daudt et al. (2021) studied the impact of explainability with ML risk calculator on domain experts, respectively for pulmonary embolism and cervical cancer.

In Diprose et al. (2020), 170 physicians completed a survey about their understanding, explainability and trust in ML outputs, with or without explainability. The authors aimed to investigate the association between physician understanding of modelling predictions, their ability to explain the decision to patients and their willingness to trust the modelling using multiple explainability methods. Physicians were divided into four groups, each group with the control modelling output without explainability, one of the two global explainability methods and one of the two local explainability methods. Global methods were Variable importance and Individual conditional expectation, and local methods were *LIME* and *Shapley Values*. After each output (control with no explainability, global explainability and local explainability), physicians were asked whether the modelling prediction made sense to them on a 4-point Likert scale, whether they would be able to explain the decision to their patients (yes/no response) and whether they would follow the modelling prediction (yes/no response). Results showed a statistically significant relation between physicians' understanding of modelling output with explainability, their confidence in explaining to patients and their trust to follow modelling decisions for all explainability methods. A significantly higher proportion of physicians favoured global or

local explanations over only the modelling output and no explanations. When comparing explainability methods, physicians significantly preferred local explanations to global methods thanks to the simplicity of the visualisation, the specificity of the explanation and the confirmation of clinical knowledge even if it did not influence physician behaviour. The results may be slightly tempered, however, since 76% of the doctors already found the prediction of the model to be trustworthy even without explanation. In addition, the order of outputs was always the same (no explanations, global explanations, local explanations), possibly favouring local explanations. The high performance of the model may also positively bias physicians' confidence in the risk scores. Based on these results and the differences in experts' and non-experts' behaviour, we can hypothesise that the explanations may be of greater benefit to users who already have some knowledge of data analysis and/or knowledge in the application domain.

Daudt et al. (2021) aimed to assess the explanations impact on different users, based on their level or domain of expertise, and their confidence and understanding of modelling results. They evaluated explanations on two different domains, one critical and one non-critical, to investigate differences in users' behaviours and confidence. Users are divided into three groups: *Layman* for users that used AI products without being ML experts, *Domain Expert* for users that are professionals in the domain from whom data are extracted (here medical experts), *IA Expert* for users who design ML algorithms and/or explainability methods. *SHAP*, *LIME* and Permutation Importance were qualitatively evaluated through questionnaires. Users had to review explanations and also gave their feedback about the important attributes without seeing explanations. In a critical domain, the medical domain, users answered that *SHAP* and *LIME* helped them the most to understand the results, especially AI Experts. *LIME* helped more than *SHAP* only for Layman users. When asked about their preferred methods, Layman and AI Experts responded *LIME* while *SHAP* was selected by Domain Experts. However, all users had the worst results with *LIME* when asked about the attributes that most influence the diagnosis based on explanations and the best results with Permutation Importance. This last method seems to be the best one for domain experts, while Layman and AI Experts performed better with *SHAP*. In comparison, in sports -the non-critical domain studied in Daudt et al. (2021)-, *SHAP* showed the best result at helping understand the results, Permutation Importance at determining the most influential attributes and *LIME* was chosen as the most helpful method. Only Layman users had different results, by favouring the Permutation Importance. Users then reacted differently based on their expertise and the application domain, as already seen in Diprose et al. (2020). The limits of this study lie in the absence of the full questionnaires to understand how users' perception is retrieved, along with information about how explanations are displayed, to counter potential bias when explainability methods are always shown in the same order, as previously seen.

In addition to the previously mentioned limits, validating explanations with users can unintentionally combine the evaluation of explanation correctness with evaluating the correctness of the predictive model. Leavitt and Morcos (2020), therefore, plead for "clear, specific, testable, and falsifiable hypotheses" that dissociate the evaluation of the explainability method from the predictive model. The evaluation of explanations via user tests, as crucial as they are, are not in themselves self-evident, with clear and general protocols, and several pitfalls can disrupt the creation and implementation of valuable and optimal tests. The domain, the user level of expertise, and the type of explanations

seem the first steps to efficient user testing.

Paper	Goal	Task	XML method	Users	Expertise	Experimental conditions
Weerts et al. (2019)	Evaluate the XML utility for the prediction's assessment for domain-experts	Simplified evaluation of positive prediction	SHAP	159	Students, Basic XML knowledge	2 groups : predictions with or without XML
Jesus et al. (2021)	Isolate the impact of gradually providing different levels of information	Payment fraud detection	TreeSHAP, LIME, TreeInterpreter	3	Fraud analysts (domain-experts)	3 steps : (1) Data Only (2) Data+ML score (3) Data+score+XML
Diprose et al. (2020)	Investigate the relation between prediction understanding, ability to explain decision and ML trust	Health Risk Calculator	2 globals: Permutation variable importance, Individual conditional expectation 2 locals: Shapley Values, LIME	170	General Practitioners (domain experts)	4 groups: 1 global & 1 local method (1) Data Only (2) Data + global (3) Data + local
Daudt et al. (2021)	Evaluate the explanations for different types of users (critic and non-critic domains)	2 tasks : Cervical cancer, FIFA Man of the Match	SHAP, LIME, Permutation Importance	49 & 46	3 types: Layman, Domain experts, IA experts	All participants review the three methods

Table 2.3: Overview of the user tests reported

2.4 Conclusion

In the field of explainability, the state of the art shows a proliferation of research, in designing new methods, evaluating them, using them and testing them in real-life situations. This research field is especially relevant as existing regulations and recommendations, and those currently being drafted, tend to impose transparency on the decisions made by automatic algorithms, particularly in Europe and in the medical field (European General Data Protection Regulation[4], European AI Act[5], Report *"The impact of artificial intelligence on the doctor-patient relationship"*[6], Recommendation *"Artificial intelligence in health care: medical, legal and ethical challenges ahead"*[7], UNESCO *"Recommendation on the Ethics of Artificial Intelligence"*[8]).

However, several limitations have been raised previously, and this thesis proposes to address them as follows:

1. **On local attributive explainability methods**. To the best of our knowledge, the *Coalitional* method is the only one other than Shapley values to keep attribute correlations and use smart coalitions to compute explanations. However, this method depends on a hard-to-setup parameter to select useful coalitions and lacks a detailed evaluation against popular methods such as *LIME* and *SHAP*. In Chapter 3, we propose an upgrade to the *Coalitional* method to tackle the initialisation parameter problem and present results from a complete benchmark of this method against *SHAP*, *TreeSHAP* and *LIME*. We also provide recommendations on which method is best to use, depending on the data and the desired precision of the explanations.

[4] https://eur-lex.europa.eu/eli/reg/2016/679/oj

[5] https://www.europarl.europa.eu/doceo/document/TA-9-2023-0236_FR.html

[6] https://www.coe.int/en/web/bioethics/report-impact-of-ai-on-the-doctor-patient-relationship

[7] https://pace.coe.int/en/files/28813

[8] https://unesdoc.unesco.org/ark:/48223/pf0000380455

2. **When using and evaluating explainability, local explanations do not appear to support user decision-making as much as expected.** Moreover, some research on specific datasets used local explanations for further analysis of the underlying data with promising results. In Chapter 4, we propose to explore explanations as a new data space to reveal complex data patterns and better understand both the modelling and the original dataset. We add clustering to a complete explainability framework and investigate the idea of studying separately instances well and unwell predicted by the modelling.

3. **XML User experimentation shows mixed results mostly due to limitations in the experimental protocol, low number of participants or unclear hypotheses, especially in the medical domain.** User needs and expectations are also unwell-defined, leading to unsuitable and incomplete user interfaces. In Chapter 5, we propose an approach to integrate explainability in medical user interfaces to understand the data, their modelling and the predictions. We work on enhancing the understandability, usability, actionability and tractability of explanations for users. We build a user experiment protocol for healthcare professionals and students to assess the contribution of explanations and explanations analysis - clustering and similar patients.

Chapter 3

Explanations of predictions: Coalitional method improvement and evaluation of local attributive methods

Contents

3.1 Introduction

Limitations of current approximation methods of the *Complete* method highlighted in Chapter 2 indicate that potential interactions between attributes must be better taken into account. Combination of correlated attributes should be avoided as possible to minimise the complexity, thus computation time, while staying at high accuracy compared to the *Complete* method. To this end, Ferrettini et al. (2020b) propose several grouping methods based on *Principal Component Analysis* (PCA), *Spearman correlation factor* (Spearman) and *Variance Inflation Factor* (VIF). *Reverse* methods -based on either *Spearman* or *VIF* grouping methods- that only gather uncorrelated attributes were also developed since groups only formed of highly correlated attributes contain mostly redundant information. Explanations through influence for each attribute of the dataset are then computed using *coalitional* influence, which takes as parameters the list of groups generated by a grouping method.

This chapter will provide an in-depth look at coalition methods, followed by our contribution to their improvement in Section 3.2. We will evaluate the coalition methods against the literature in two steps in Section 3.3: first, against *SHAP* and Shapley-based methods and then against more local attributive methods like *LIME* and variation of *SHAP*. Finally, we provide two medical examples of how to interpret and use local explanations and explore local attributive explanation methods hyperparameters in 3.4 and propose recommendations on the best approach to use, depending on the data and the desired precision of the explanations in Section 3.5.

The work mentioned in this Chapter has been published in the following articles: Ferrettini, Escriva, Aligon, Excoffier, and Soulé-Dupuy (2021); Doumard, Aligon, Escriva, Excoffier, Monsarrat, and Soulé-Dupuy (2022, 2023).

3.2 On optimising the *Coalitional* methods

3.2.1 The original coalition method

The *Coalitional* method proposed by Ferrettini et al. (2020a,b) is based on the identification of attributes having interactions between them to obtain a grouping, for example $G = \{\{a_1, a_3\}, \{a_2, a_5, a_8\}, \{a_4\}...\}$. With such groupings of attributes, it becomes possible to consider only the attributes of a subgroup, without having to consider every possible attribute combination like in the *Complete* method. It is important to note that the groups do not necessarily have to be exclusive, which means an attribute a_i can be found in multiple groups of G.

Definition 3.1. *Coalitional method*

Let G be a pre-computed coalition of attributes, A the set of attributes in the dataset, a_i the $i-th$ attribute, $G_{\{}a_i\}$ the subset of G containing the coalitions of attributes $g \in G$ such as $a_i \in g$, Δ the difference between the expected prediction for x for a combination of attributes S and the expected prediction in the absence of data, with $\hat{f}$ the prediction function of the ML model. The coalitional influence $\phi_i \in \Re$ of the attribute i for an instance x is defined as:

$$\phi_i(x) = \sum_{g' \subseteq g \setminus \{a_i\}, g \in G_{\{a_i\}}} \frac{|g'|!(|g| - |g'| - 1)!}{\sum_{g \in G_{\{a_i\}}} |g|!} (\Delta_{g' \cup \{a_i\}}(x) - \Delta_{g'}(x))$$

$$\Delta_S(x) = \hat{f}_S(x) - \hat{f}(\varnothing)$$

Given the fact that we can set a maximum cardinal c for our subgroups, the complexity is now, in the worst case, $O(2^c * \frac{m}{c} * l(m,x)) \approx O(m * l(m,x))$ with m the number of attributes in the dataset. This method calculates fewer groups than the *k-depth* method described in Section 2.1 but tries to make up for it by only grouping the attributes related to each other. To determine which attributes seem to be related and build the coalitions of attributes G, several types of coalition strategies are proposed below, based on well-known correlation calculation and dimension reduction techniques: *PCA*, *VIF* and *Spearman*.

It should also be noted that the following group generation strategies depend only on the correlation between attributes, expressed as an α-threshold, and not on the ML model used. It is unnecessary, therefore, to re-generate the groups when switching to another ML model as opposed to any SHAP-based method that must be completely re-trained for any new model, even if it only differs from the previous one by a slight change in a hyperparameter.

3.2.1.1 PCA-based coalition

The main principle of a Principal Component Analysis (PCA) is to reduce a dataset to its simplest expression in terms of attributes. In other words, if the dataset is considered a multidimensional matrix, the *PCA* aims to reduce its dimensionality as much as possible. To do that, the different attributes of the dataset are combined linearly, the result being a new set of attributes, each new attribute being a linear combination of the previous ones.

Our reasoning, for this approach, is to consider the set of combined attributes (summarised by the new attribute of the *PCA*) as a group of influence.

Given a dataset $D = (A, X)$ composed of a set of m attributes $A = \{a_1, ..., a_n\}$, and a set of instances X where $x \in X, x = \{x_1, ..., x_m\}$ with $\forall i \in [1..m], x_i \in a_i$. We can apply a PCA which produces a new dataset $D' = (A', X')$ such as $A' = \{a'_1, ..., a'_p\}$ with each new attribute being a linear composition of the previous attributes: $\forall i \in [1, ..., p], a'_i \in A', \exists\{\beta_1, ..., \beta_m\} \in R^m, a'_i = \beta_1 * a_1 + ... + \beta_m * a_m$.

Given this set of factors $\beta_1, ..., \beta_m$, for each attribute, we consider each factor as an evaluation of the importance of the attributes in the group. We can then constitute a coalition of attributes by exploiting the groups formed by the most important factors. This gives us the algorithm 1. For the sake of simplicity, we consider each $a' \in A'$ as a vector of its β_i factors.

3.2.1.2 VIF-based coalition

The Variance Inflation Factor (*VIF*) is an estimation of the multicollinearity of the attributes of the dataset regarding a given target attribute.

Given a dataset $D = (A, X)$, the *VIF* value of $a \in A$ is calculated by running a standard linear regression with a as the target for the prediction. Then, given R the coefficient of determination of the linear regression, we have:

$$VIF(a) = \frac{1}{1 - R^2} \tag{3.1}$$

It is commonly accepted that a VIF superior to 10 indicates strong multicollinearity of the attribute with other attributes of the dataset. This threshold of 10 is arbitrary

Algorithm 1 PCA-based coalition extraction.

Input: a threshold α and the set of attributes A' of the PCA
Output: σ a coalition of attributes

$\sigma \leftarrow \{\}$
for all $a' \in A'$ **do** ▷ for each attribute generated by the PCA.
  $g \leftarrow \{\}$ ▷ g, a new possible group
  $\beta_{max} \leftarrow max(a' = \beta_1, ..., \beta_m)$ ▷ find the most important attribute
  for all $\beta_i \in a'$ **do**
    if $\beta_i \geq \beta_{max} * (1 - \alpha)$ **then**
      add a_i to g ▷ the attribute is included in the group if close to the max
    end if
  end for
  add g to σ
end for
return σ

but considered a standard in numerous publications (Makki, 2019). Moreover, when an attribute is removed from the dataset, the *VIF* of the attributes multicollinear with it decreases. Then, we can automatically detect groups of attributes by calculating the *VIF* of each attribute (considered as a target) of the dataset and then comparing them with a new *VIF* calculation with an attribute removed. For this purpose, we consider two possible approaches:

- Considering as a priority the calculation of strongly multicollinear groups of attributes: Those are groups of attributes with a dependency on one another. In the context of this approach, attributes whose *VIF* varies strongly when an attribute is removed from the dataset are considered part of the group.
- Considering as a priority the calculation of weakly or non-multicollinear groups of attributes: Given the fact that correlated attributes tend to bring the same information to the model, it may be preferable to prioritise groups for which the addition or removal of an attribute changes greatly the information brought by the group.

These two approaches are named *VIF* coalition and *Reverse-VIF* coalition, respectively. This gives us the algorithm 2, for the *VIF* coalition. The *Reverse-VIF* coalition can be obtained simply by replacing the condition for adding an attribute to a group by $if\ newvifs(a') > oldvifs(a') \times (1 - t \times 0.05)$. This supplementary ratio of 0.05 has been obtained by preliminary experiments, which showed that just keeping the $1 - t$ factor led to a generation of all the possible subgroups, which defeated the principle of an approximation.

3.2.1.3 Spearman-based coalition

A limit of the VIF is the sole consideration of multicollinearity, while a correlation between attributes might not be linear. This problem is addressed through the *Spearman* correlation coefficient, which takes into account non-linear correlations. *Spearman* being not multicollinear, the calculation of the correlation between attributes has to be done by pairs. Thus, the method consists in generating the matrix of all the correlations

Algorithm 2 VIF-based coalition extraction.

Input: a threshold α, the set of attributes of the dataset A and a function $VIF(A)$ calculating the array of all the *VIF* of all the subsets of a set of attributes

Output: σ a coalition of attributes

$\sigma \leftarrow \{\}$
$vifs \leftarrow VIF(A)$ ▷ calculating the initial VIFs of the attributes
for all $a \in A$ **do**
    $g \leftarrow \{\}$
    add a to g
    $new\_vifs \leftarrow VIF(A \backslash a)$
    for all $a' \in (A \backslash a)$ **do**
        if $new\_vifs(a') < vifs(a') \times (0.4 + \alpha)$ **then**
            add a' to g
        end if
    end for
    add g to σ
end for
return σ

of each pair and then deciding which attributes are part of a group. For this method, we have the same two possibilities as for the *VIF* method: we can either prioritise the calculation of strongly correlated attributes or on the contrary, prioritise groups of non-correlated attributes. These two approaches are named respectively *Spearman* coalition and *Reverse-Spearman* coalition.

Given a dataset $D = (A, X)$, with $A = \{a_1, ..., a_m\}$ the correlation matrix C is obtained by computing the *Spearman* correlation coefficient of each attribute couple: $C(1, 2) = corr(a_1, a_2)$. Thus C is symmetrical and has 1 as the value of its whole diagonal. For each line i of the matrix C, we consider as grouped with a_i the attributes strongly (or weakly) correlated with a_i, for the *Spearman* coalition (or the *Reverse-Spearman* coalition).

The algorithm 3 details the *Spearman Coalitional* explanation method. If the most correlated attribute has a coefficient less than 0.1, a is considered as a singleton thanks to the condition $max(corrmat(a)) > 0.1$. The *Reverse-Spearman Coalitional* explanation method can be obtained by replacing the condition for adding an attribute to a group by $corrmat(a, a') < min(corrmat(a)) + max(corrmat(a)) \times \alpha and min(corrmat(a)) < 0.5$. This adds the least correlated attributes up to a threshold: if the attribute least correlated to a has its *Spearman* correlation to a superior to 0.5, we consider the attribute a as a singleton.

3.2.2 Improvements on the coalition method

In its first version, all *Coalitional* methods depended on an α-threshold to determine the size of the attributes coalitions. It is almost the only hyperparameter of the method, other than choosing the *Coalitional* strategy. To evaluate the impact of *alpha* on the results and performances of the *Coalitional* methods, we study the size and number of the coalition groups created, depending on the chosen α and the dataset number of attributes.

Figure 3.1 and 3.2 compare the average number and average size of the groups of attributes generated by *Coalitional* methods for several α-thresholds. The average number

Algorithm 3 Spearman-based coalition extraction.

Input: a threshold α, the set of attributes of the dataset A, and a function $spearman(A)$ calculating the matrix of all the absolute *Spearman* correlation coefficient of all the subsets of a set of attributes. a max and min functions which returns the maximum and minimum of a matrix line.

Output: σ a coalition of attributes

$\sigma \leftarrow \{\}$
$corrmat \leftarrow spearman(A)$ ▷ calculating the correlation matrix
for all $a \in A$ **do**
  $g \leftarrow \{\}$
  for all $a \in A$ **do**
    if $corrmat(a, a') > max(corrmat(a)) \times (1 - \alpha)$ and $max(corrmat(a)) > 0.1$ **then**
      add a' to g
    end if
  end for
  add g to σ
end for
return σ

of groups varies considerably according to the grouping strategy. *PCA* strategy produces more groups than other methods, for all α values, and differences are more visible when the number of attributes in the dataset increases. The impact of α on the number of clusters seems also dependent on the grouping method: differences between $\alpha = 0.01$ and $\alpha = 0.4$ are greater with *PCA* than with *Reverse-Spearman* for all number of attributes. Similar behaviours are seen with the size of the groups, with gaps based on the grouping strategy and the α value. The size of the groups increases proportionally with the number of attributes with *Reverse-Spearman* and *Reverse-VIF* strategies when the size remains similar with *PCA* and *Spearman* strategies. Differences based on the α are also greater for both Reverse strategies. Globally, *Reverse-VIF* generates few large groups, whereas *PCA* generates many small groups. These differences impact the number of distinct combinations of attributes taken into account for the explanations computation.

To evaluate the impact of these differences in the creation of coalition groups, we compute and study the average complexity of the groups of attributes in Figure 3.3. We define the *complexity* as the ratio between the number of distinct attribute combinations for a coalitional strategy and an *alpha*-threshold and the total number of distinct attribute combinations, equal to $2^M - 1$ where M is the number of attributes. We obtain a ratio between 0 and 1, representing the fraction of all the possible attribute combinations taken into account to compute explanations. A value close to 1 indicates that the generated group list is similar to the *Complete* method. A low value indicates that the group list is closer to the *linear*, thus less complex and faster to compute. As expected from previous results, the *PCA* method generates on average the fewest distinct coalitions of attributes for all α-threshold while the *Reverse-VIF* methods generate the most complex ones. Changes in α-threshold have a more significant impact on complexity for the Reverse strategies. When the number of attributes increases, *Reverse* strategies have a complexity near half the complete, especially with α-thresholds above 0.2, while *PCA* and *Spearman* have a decreasing complexity due to the low mean size of each attributes group shown in Figure 3.2.

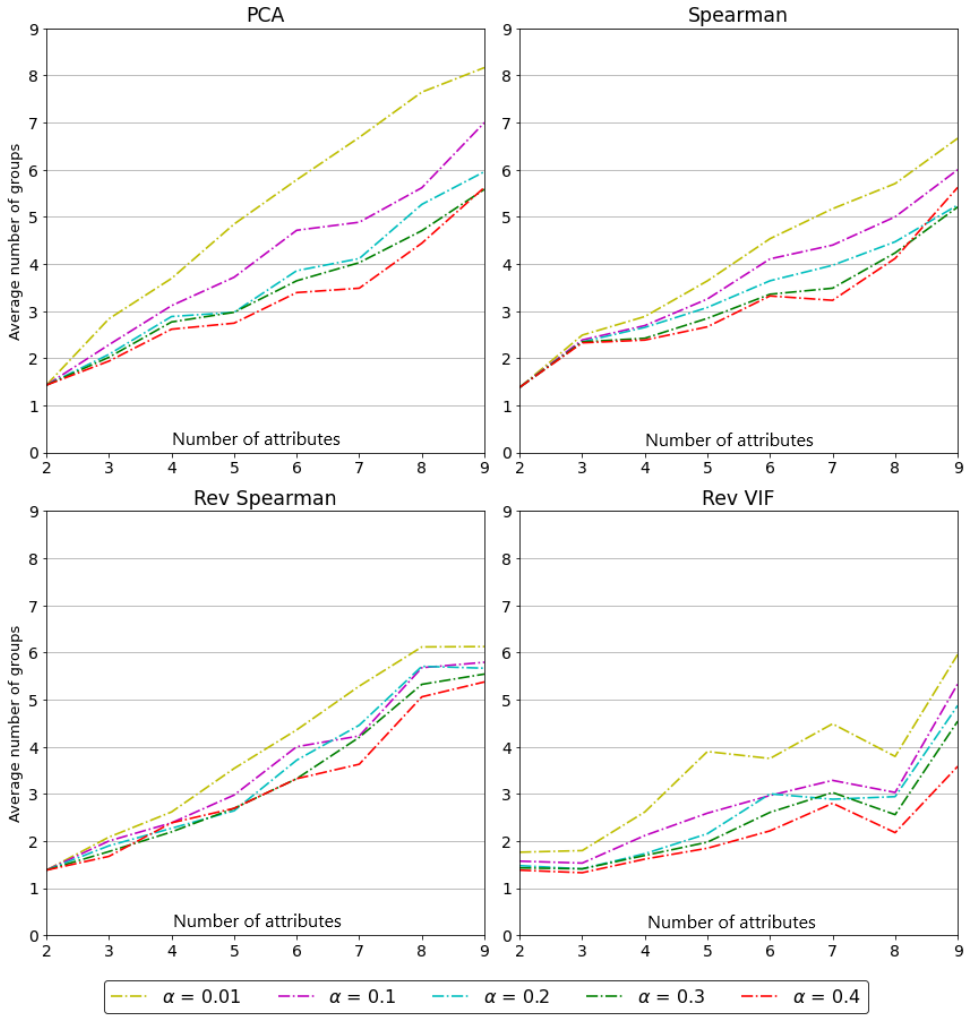


Figure 3.1: Mean number of groups for *Coalitional* methods depending on α-threshold and number of attributes.

To illustrate this phenomenon, Figure 3.4 displays the evolution of the complexity for the groups generated by four *Coalitional* methods for a particular dataset with seven attributes. The *linear* complexity is thus equal to 7 while the *Complete* method one is $2^7 - 1 = 127$. We study the evolution of the number of distinct subgroups, so the complexity, of the coalitions created by each grouping method for $\alpha \in [0, 1]$. There is a clear difference between the evolution of each grouping method confirming that the α-threshold can not be set at the same value for all methods. To reach the complexity of the complete, the α-threshold needs to be set to 0.22 for *Reverse-VIF*, 0.64 for *PCA* and 0.85 for both *Spearman* strategies. If we focus on the complexity, if one wants a 25%-complexity (i.e. a number of distinct subgroups equal to $0.25 \times 127 \approx 63$), the α-threshold for the *Reverse VIF* method would be equal to about 0.08, whereas for *Spearman* it would be about 0.42.

To sum up, the group characterisation for *Coalitional* methods shows that the groups differ greatly depending on the α-threshold and the grouping method. This behaviour can raise a limit when using the *Coalitional* methods to choose hyper-parameters. It can also complicate the experimental setup for evaluating and benchmarking the different *Coalitional* strategies in appropriate and fair conditions. We improve the *Coalitional* methods described in Section 3.2.1 by relying on the proportion of the *Complete* method

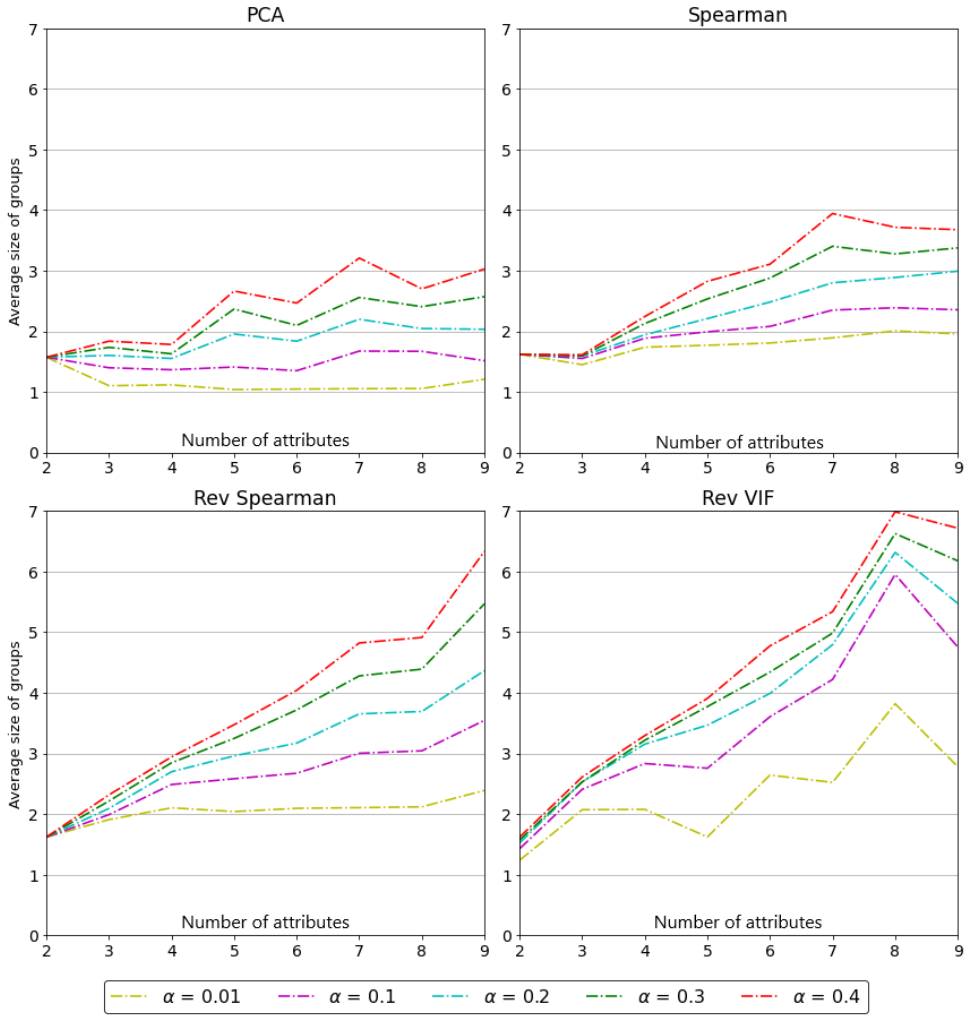


Figure 3.2: Mean size of groups for *Coalitional* methods depending on α-threshold and number of attributes.

complexity rather than the α-threshold. Then, if a short calculation time is required, the proportion can be set at 10% whereas it can be set at 50% if calculation time is not an issue and more precise results are needed.

Our approach is based on a *dichotomic search* to find the most appropriate α value for the chosen complexity, as described in Algorithm 4.

Originally, the dichotomic search works to find a value in an ordered interval by successively dividing the interval into two parts and selecting the interval containing the searched value. In our case, the two limits of the interval are unknown: depending on the grouping strategy, α can be any positive number or a number $\in [0, 1[$. Then, we decided to adapt the dichotomic search by initialising an α value compatible with all grouping methods and limiting the research to a maximum number of iterations rather than finding the perfect solution. Thus, at each iteration, we compare the wanted complexity to the complexity with the current threshold and update the α value. As we cannot split the interval in two as in the original dichotomic search to change the α value, we added or subtracted half the α value to virtually create intervals for searching for the optimal α value. We also keep the nearest result in case the algorithm does not find a perfect solution and the maximum iteration is reached. Indeed, unlike the original dichotomic search, finding a solution is not guaranteed as the numbers of distinct subgroups, so the

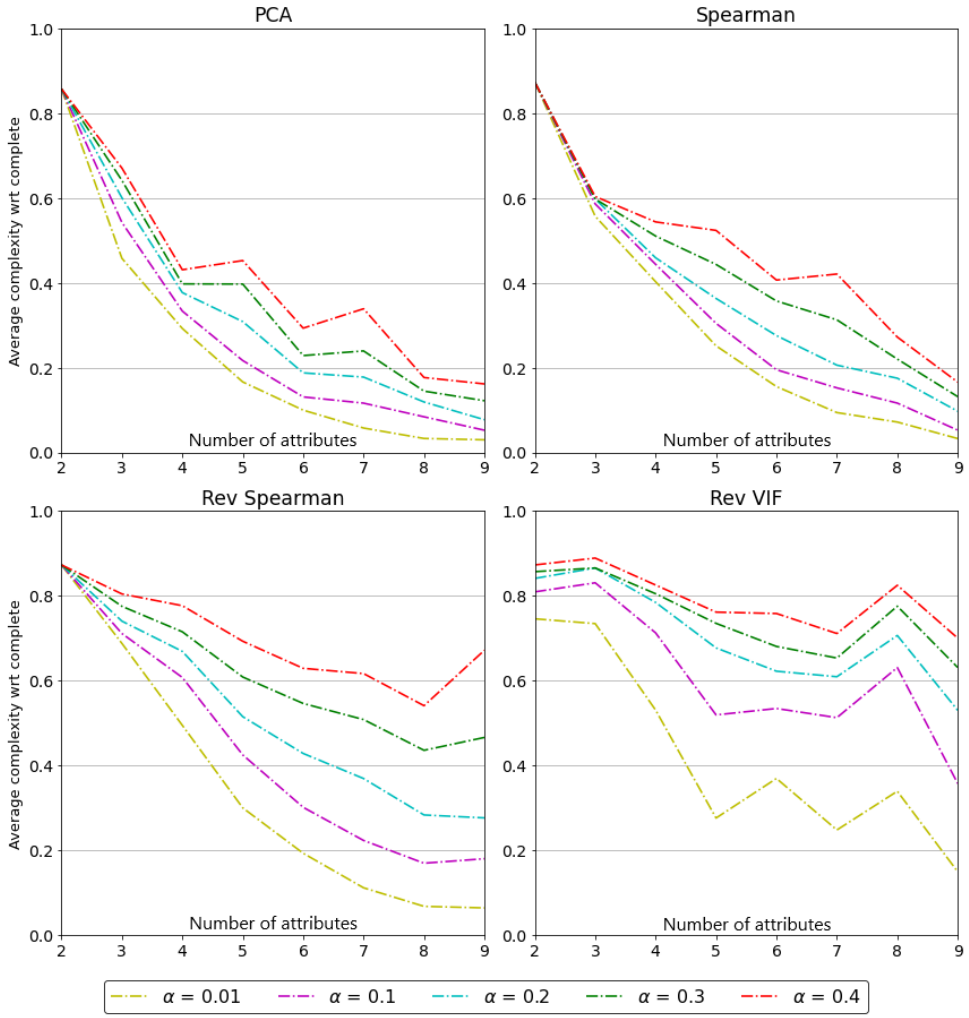


Figure 3.3: Mean complexity proportion compared to *Complete* method complexity for *Coalitional* methods depending on α-threshold and number of attributes.

possible complexity values, are not a continuous range as we can see in the example in Figure 3.4. The logic behind it is relatively trivial: Each time an attribute is added to a group, the number of distinct sub-groups increases according to the number of attributes already present in the group and if the attribute is already in other coalitions. For example, for a dataset containing four attributes $\{A, B, C, D\}$ and with no possible attribute redundancy:

- If attribute A is added to group $\{B\}$, the number of possible subgroups increases from 1 to 3: each singleton plus the combination of A and B.
- If the attribute A is added to the group $\{B, C\}$, then the number of possible subgroups increases from 3 to 7: the subgroups $\{A\}$, $\{A, B\}$ and $\{A, B, C\}$ are added to the subgroups already possible.

In our case, as the groups are not exclusive, an attribute can be redundant between several groups. So identical subgroups may appear several times when all the possible subgroups are calculated. Continuing with the previous example, with the following coalition: $\{\{A, B\}, \{A, C, D\}\}$. The group $\{A, B\}$ gives the subgroups $\{A\}$, $\{B\}$, $\{A, B\}$ and the group $\{A, C, D\}$ the subgroups $\{A\}$, $\{C\}$, $\{D\}$, $\{A, C\}$, $\{A, D\}$, $\{C, D\}$, $\{A, C, D\}$.

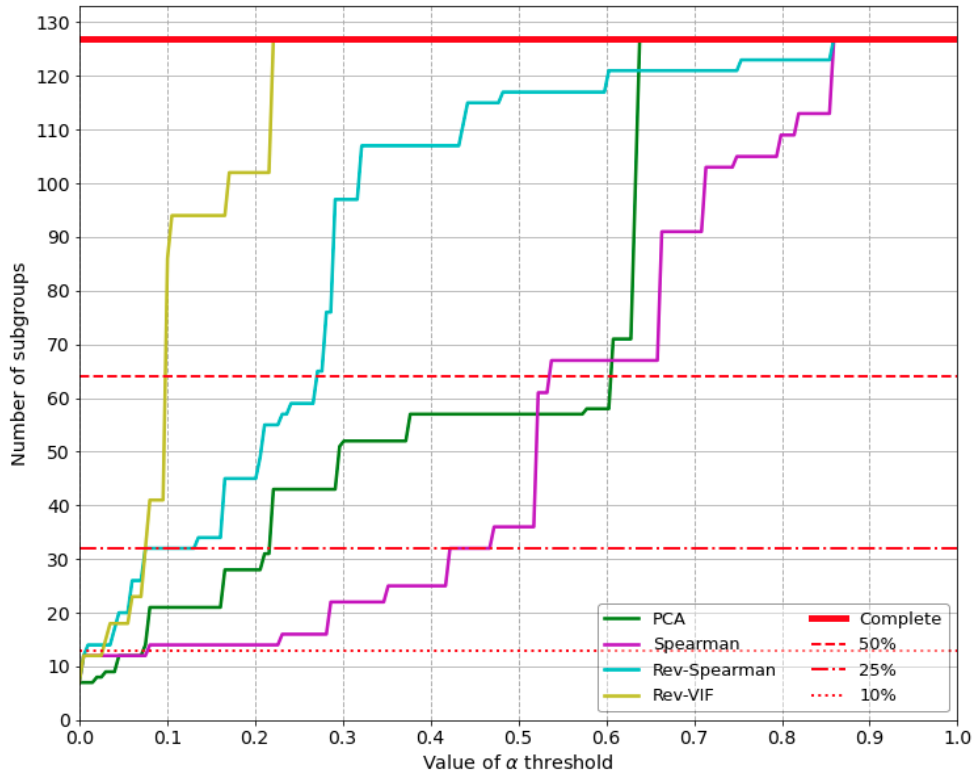


Figure 3.4: Evolution of complexity of *Coalitional* methods depending on α-threshold.

Group $\{A\}$ is then redundant in the list of possible subgroups with these coalitions. The number of distinct subgroups is then equal to 9, not 10, and the complexity as we calculate it is equal to $9/(2^4 - 1) = 0.6$. It may be impossible to achieve a complexity of $10/(2^4 - 1) = 0.67$ using our grouping strategies to compute coalitions. A complexity equal to 0.6 may then be the nearest, so best, complexity we can have.

Our algorithm for the optimal α-threshold search ends in three configurations, ensuring that the algorithm ends when an optimal solution is found and that an infinite while loop does not occur:

- The initialised α is already the optimal solution.
- An optimal α threshold is found during the iterations.
- The maximum iteration is reached, and the nearest solution found is returned.

Finally, our version of the *Coalitional* methods retains the advantage of building groups independently of the ML model used.

3.3 On evaluating the *Coalitional* methods against the literature

In this section, we evaluate the *Coalitional* methods in their updated version in two different experimental setups. First, we compare the optimised *Coalitional* method against literature methods based on Shapley values. We benchmark multiple grouping strategies for three different percentages of complexity and analyse their explanations against the

Algorithm 4 Searching of the optimal α-threshold based on the complexity percentage.

Input: a complexity percentage p, the set of attributes of the dataset A, a grouping function *grouping*, a function *count* that returns the number of distinct possible subgroups based on a list of groups, and a number of maximum iteration μ.

Output: the best α threshold found

$\alpha \leftarrow 0.5$
$\sigma \leftarrow grouping(A, \alpha)$
$complexity \leftarrow 2^{lenght(A)} - 1$ ▷ Compute the total complexity
$n \leftarrow ceiling(p \times complexity)$ ▷ Compute the wanted complexity
$n\_subgroups \leftarrow count(\sigma)$
$n\_subgroups_{best} \leftarrow n\_subgroups$
if $n\_subgroups = n$ **then** ▷ Initial verification
    return α
end if
$i \leftarrow 0$
while $i < \mu$ **do**
    if $n\_subgroups < n$ **then**
        $\alpha \leftarrow \alpha + \frac{\alpha}{2}$
    else if $n\_subgroups > n$ **then**
        $\alpha \leftarrow \alpha - \frac{\alpha}{2}$
    end if
    $\sigma \leftarrow grouping(A, \alpha)$
    $n\_subgroups \leftarrow count(\sigma)$
    if $n\_subgroups = n$ **then**
        return α
    end if
    if $|n\_subgroups - n| < |n\_subgroups_{best} - n|$ **then**
        $n\_subgroups_{best} \leftarrow n\_subgroups$
        $\alpha_{best} \leftarrow \alpha$ ▷ Keep the most optimal solution in memory
    end if
    $i \leftarrow i + 1$
end while
return α_{best}

Complete method (Štrumbelj and Kononenko, 2010), the *k-depth* method (Ferrettini et al., 2020a), *KernelSHAP* (Lundberg and Lee, 2017) and *TreeSHAP* (Lundberg et al., 2020). In a second step, we evaluate the most promising *Coalitional* method with more datasets, ML models and metrics and added two more literature XML methods to the comparison: *LIME* (Ribeiro et al., 2016) and one variation of *TreeSHAP*.

3.3.1 Against Shapley-based XML methods

In this experiment, published in Ferrettini et al. (2021); Escriva et al. (2022), we aim to evaluate our improved *Coalitional* methods against similar XML methods from the literature, those based on the Shapley-values, especially as *SHAP* is one of the most-used explanation methods. We aim to compare them on a large collection of datasets in a similar setup, explicitly detailed below.

3.3.1.1 Experimental protocol

Our tests are realised with the data available on the Openml platform (Vanschoren et al., 2014). We select the biggest collection of datasets [1] on which classification tasks have been run. We also consider two classification ML models: Random Forest and Support Vector Machine (SVM) with the non-linear Radial-Basis-Function (RBF) kernel. Experiments are conducted using Python 3.7.9 with the Scikit-Learn version 1.0.1 implementation on both models [2]. We use default values for model hyperparameters. Due to the heavy computational cost of the *Complete* method -considered the reference of our experiments- we select the datasets with at most nine attributes.

Thus, a collection of 243 datasets is obtained. Table 3.1 details the number of datasets and statistics about the number of instances for each number of attributes. We can see that the number of instances varies greatly depending on the number of attributes in our collection of datasets. This could impact the comparison of XML methods behaviours when the number of attributes in the datasets is considered. We would then adapt the evaluation metrics when required.

# of attributes	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**All**
# of datasets	3	21	44	25	38	26	34	28	24	243
Mean insts #	724	736	1688	560	843	600	456	750	479	760
Median insts #	130	138	475	264	250	229	294	310	281	277
Min insts #	40	27	44	23	38	15	40	34	52	15
Max insts #	2001	5456	10386	5456	7129	3107	4052	4177	1473	10386

Table 3.1: Statistics of the OpenML dataset collection for a given number of attributes.

For each dataset and model, we generate the explanations for each *Coalitional* method described in Section 3.2, for each instance of the 243 datasets: *PCA Coalitional*, *Spearman Coalitional*, *Reverse-Spearman Coalitional*, *VIF Coalitional* and *Reverse-VIF Coalitional*. We also compute the explanations for the following literature methods: the *Complete* method for the baseline, *K-depth* method, *KernelSHAP* and *TreeSHAP*. The *Coalitional* explanations are generated using the different group generation methods based on a percentage of the total complexity of 10%, 25% or 50% (small complexity resulting in less distinct subgroups, and high values in more subgroups). We generate the possible subgroups with these three different values of complexity to study the impact on the computation speed and accuracy of the method.

To compare the different explanation methods, we consider the explanation results as a vector of attribute influences noted $\phi(x) = [i_1, ..., i_m]$ with m the number of attributes in the dataset. Thus, each of the attributes a_k is given an influence $i_k \in [0, 1]$ by the method $\phi : \forall k \in [1..m], i_k = \phi_{a_k}(x)$, with x an instance of the dataset. We then define a difference between two vectors of influences i, j as the Euclidean distance:

$$D(i, j) = \sqrt{\sum_{k=1}^{m} (i_k - j_k)^2}$$

Considering this formula, we define an error score based on the difference between an explanation method and the *Complete* method.

[1] Available in `https://www.openml.org/s/107/tasks`

[2] `https://scikit-learn.org/stable/`

Definition 3.2. *Error with regards to the Complete method*

With X the instances of a given dataset, n the instances number and m the number of attributes. Let $D(i, j)$ be the distance metric as defined in Equation 3.3.1.1, $\phi(X_i)$ be the influence produced by an explanation method ϕ for a given instance X_i and a given ML model, and $\phi^C(X_i)$ the influence given by the Complete method for the same model and same instance. We define the mean error of the explanation method as:

$$err(\phi, X) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}D(\phi(X_i), \phi^C(X_i))$$

We generate the error score of every explanation method, allowing us to compare their performances across the different collected datasets. Each error score is the distance of one of the *Coalitional* methods from the *Complete* method. Thus, a lesser error indicates a more precise estimation of the *Complete* method.

To compare methods, we also consider the time needed to explain a set of data, called computation time. This includes all the steps of each method to explain all the instances of the dataset, without optimisation: the time to compute the influences for all the instances of the dataset, plus the time to determine the coalitions of attributes of interest in the *coalitional* and *k-depth* methods. Since the number of instances impacts the total computation time for a dataset, each computational time is normalised by dividing by the number of instances in the dataset to compare times per instance.

All experiments were run on an AMD Ryzen 3700 processor with 8 x 3.6 GHz cores and 32 GB of RAM.

3.3.1.2 Results

To have an overview of the methods performances, we average the error compared to the *Complete* method and the computation time globally, thus independently of the number of attributes in the datasets. Therefore, it gives us a single representation, called *Performance Map*, with the computation time normalised by one of the *Complete* method on the horizontal axis, and the error for the *Complete* method on the vertical axis. The *Complete* method is thus placed on the point with coordinates $(1, 0)$. All the methods are then placed above the *Complete* method and methods placed at the left of the *Complete* method have lower computation time than the complete, while those placed at the *Complete* method right are slower to compute.

First, we compare the *Coalitional* methods against each other to evaluate their performances on all datasets. Figure 3.5 shows the mean results across the two ML models - Random Forest and RBF-SVM - of the five grouping strategies for the three percentages of complexity. *PCA* 50% have the smallest error compared to the *Complete* method since it is the method furthest down. It is also the slowest one, with a calculation time almost equal to that of the *Complete* method. *Reverse-VIF* and *VIF* 10% are the fastest ones with a time of half the complete, with also the largest error. *VIF* and *Reverse-VIF* perform least well overall, although *Reverse-VIF* 50% performs as well as *Spearman* 50%, *Reverse-Spearman* 50% and *PCA* 25%. Based on all percentages, on both ML models, the best *Coalitional* methods seem to be *PCA* and *Spearman*. *PCA* is better than *Spearman* for the error metric, but slower. *PCA* 25% and *Spearman* 50% have closed results, meaning that *PCA* achieve the accuracy of *Spearman* for lower complexity percentages, but at the cost of computation time.

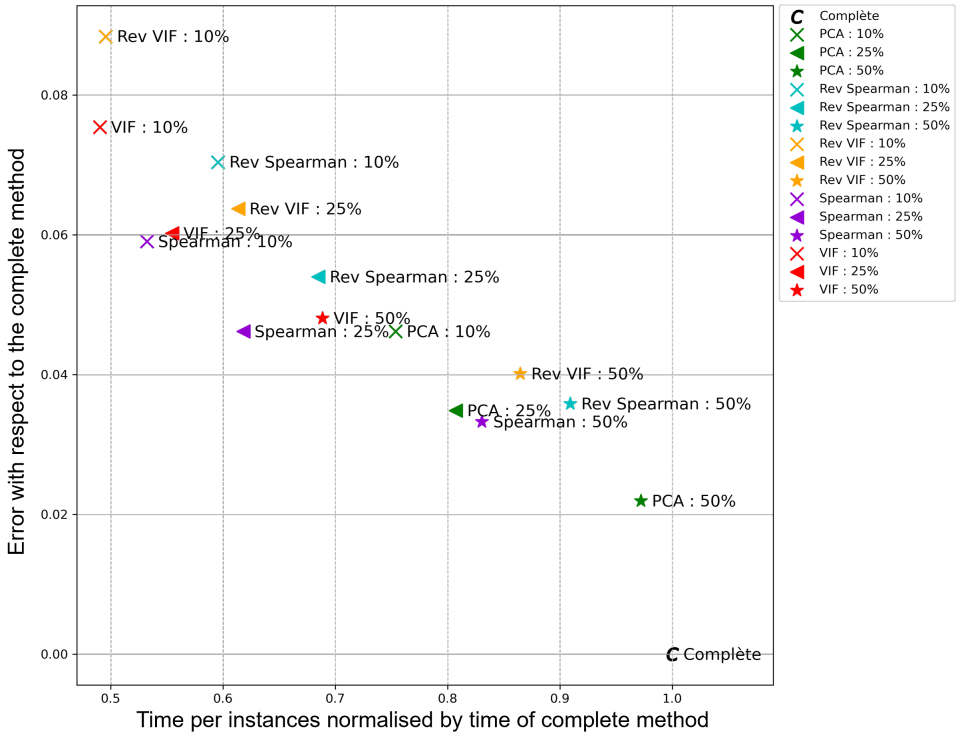


Figure 3.5: Performance maps for the *Coalitional* methods, mean results on both ML models.

To study more in-depth the impact of the grouping strategies, we also display the Performance Map for each ML model and add the *k-depth* XML method to compare results. As the number of attributes and instances in a dataset has a strong impact on the performance, we binary split the collection of datasets. The first part includes datasets that have relatively few instances (strictly less than 500) or attributes (strictly less than 6), whereas the second one only includes datasets with a higher number of instances (at least 500) and attributes (at least 6). There are 213 datasets in the first set and 30 in the second one.

Figure 3.6 shows the results for the Random Forest model for four *Coalitional* methods -*PCA*, *Spearman*, *Reverse Spearman* and *Reverse VIF* as *VIF* method was excluded based on the previous results- and the *k-depth* methods -from *linear* to *Complete* method. The left sub-graph shows the results for the first set containing datasets with few attributes or instances and the sub-graph on the right indicates results for more complex datasets as described previously.

As before, *Coalitional* methods show strong results. *PCA 50%* have a computation time equivalent to a *3-depth* but have an error close to the *4 or 5-depth* for both sets. Similarly, all *Coalitional* methods with a complexity of 25% are in terms of computation time closer to the *2-depth* while being as accurate as the *3-depth*. This highlights that *Coalitional* methods generate smarter groups with less useless or redundant information than *k-depth*, thus being more efficient.

Figure 3.7 shows the *Performance Maps* for the SVM model for both sets of datasets. Unlike Random Forest, there is a clear difference between the results for the two sets. For smaller datasets -either in terms of an attribute or instance number- some *Coalitional* methods are longer to compute than the *Complete* one. This is because the time taken to find the appropriate α-threshold with the bisection method is too large relative to

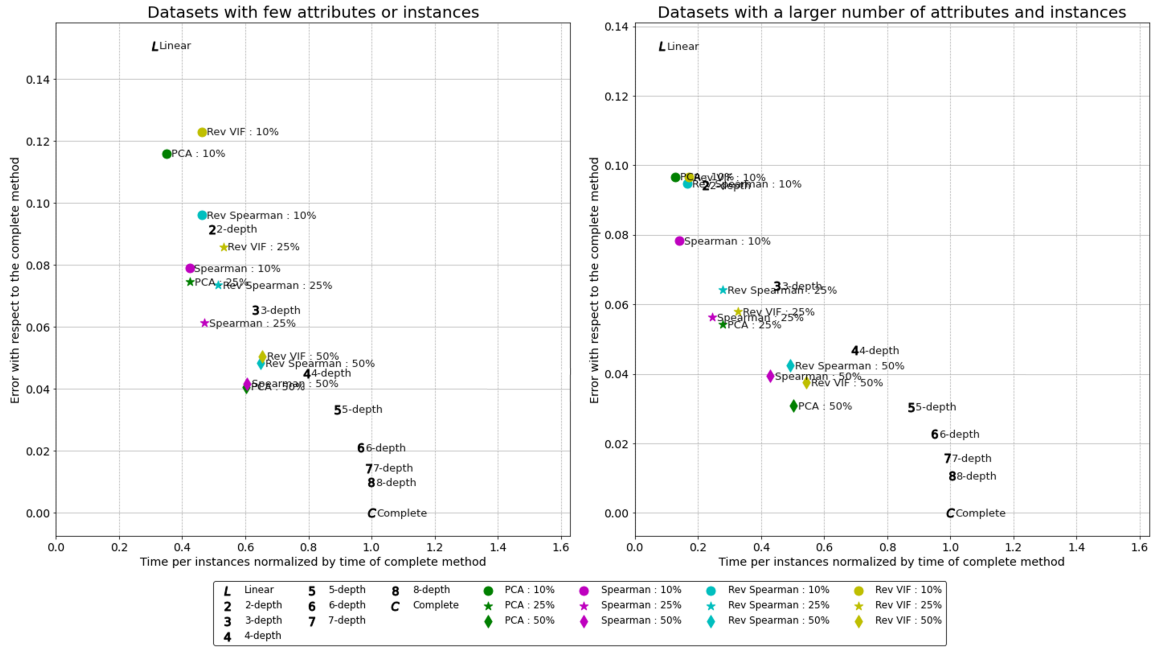


Figure 3.6: Performance maps for two sets of datasets for *Coalitional* methods for Random Forest.

the global computation time, which also includes training models for explanations and influences computation for each attribute. In the case of small datasets, the *Complete* method seems the best one, as the computation time is not as important as with large datasets - since its complexity is exponential.

Nevertheless, for larger datasets, the performances of *Coalitional* methods are satisfying. Indeed, in a similar way to Random Forest results, *Coalitional* methods with a 50%-threshold are mostly faster to compute than *4-depth* method for the same error compared to the *Complete* method -especially *PCA* which is efficient. *Spearman* and *PCA* with a 25% threshold are very efficient as well, with a computation time between those of *2-depth* and *3-depth* while being more accurate than *3-depth*.

Based on these three analyses, *PCA* and *Spearman* methods seem the most promising ones to compete against the literature. Figure 3.8 displays the Performance Map of two *coalitional* methods, with the *k-depth* method -from *linear* to *Complete* method-, *KernelSHAP* and *TreeSHAP*. Results are the mean ones across both ML models, except for *TreeSHAP* which can only be computed on the Random Forest model. The sub-graph on the left displays the *KernelSHAP* method, which gives poor results, being slower than the *Complete* method and with an accuracy between the *3-depth* and the *4-depth*. This method flattens the rest of the graph, thus the sub-graph on the right without *KernelSHAP*. This is a major inconvenience since it suggests that interpretability with models that are not tree-based -then *TreeSHAP* is unusable- would often be intractable in practice. Nevertheless, *TreeSHAP* method is still on average slower to compute than the complete, despite being specially designed for tree-based models, and the accuracy is near the *KernelSHAP*, *3-depth*, *4-depth* ones. *Coalitional* methods then out-performed the *SHAP* ones in this experiment. These bad results when computing the influences for all instances seem to be the reason behind the implementation of a Kmeans in the *SHAP* library, to summarise the instances when computing influences with *SHAP* methods [3].

[3]The documentation of this functionality was set available mid-2020 after the experiments for this

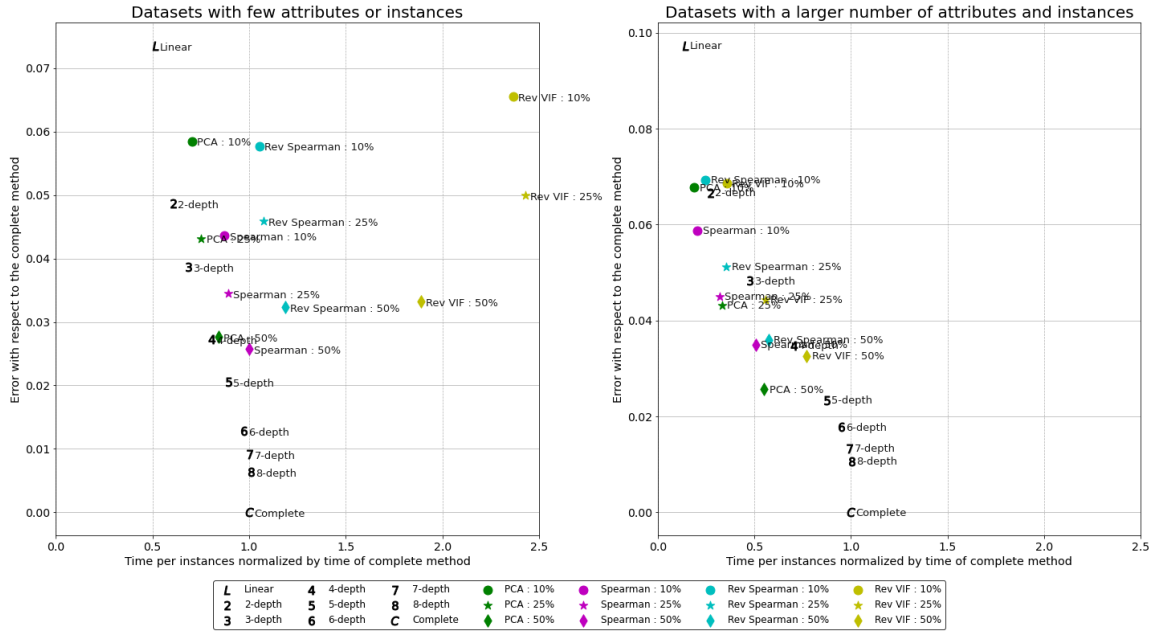


Figure 3.7: Performance maps for two sets of datasets for *Coalitional* methods for SVM.

3.3.2 Against local attributive methods

In this second experiment, published in Doumard et al. (2022, 2023), we aim to benchmark local XML methods from the literature -*LIME*, *KernelSHAP*, *TreeSHAP* and *Coalitional* methods- within the parameters recommended by the documentation on a large collection of datasets. We want to measure the theoretically well-known limitations of these methods, discussed in Section 2.1.3. We aim to identify how each method behaves in different ML modelling and data dimensionality setups - both at the instance and attribute levels - by providing insight into computational time, attribute importance, robustness, readability and clusterability of explanations. In this thesis, we will focus on comparing the *Coalitional* methods with the other local XML methods.

3.3.2.1 Experimental protocol

To compare explanation methods, we apply them to a wide range of 304 datasets available on OpenML[4]. Due to computational constraints of explanation methods, we only considered datasets with at most 13 attributes and a maximum of 10 000 instances. We also only considered classification tasks to use comparable predictive models and metrics. We describe the amount and size of datasets per number of attributes in Table 3.2. We can see that the number of instances varies greatly. This could impact XML methods' behaviours, so the comparison, when the number of instances and attributes in the datasets is considered.

For modelling these data, we choose four widely used types of ML models for classification: Logistic Regression (LR), Support Vector Machines (SVM), Random Forests (RF) and Gradient Boosted Machines (GBM). For the first three models, we use the Python library *scikit-learn* version 1.0.1. For GBM, we use the Python library *XGBoost* version

article was done. Moreover, no paper was found on the reliability of using Kmeans to optimise the computational time of SHAP.

[4] `www.openml.org`

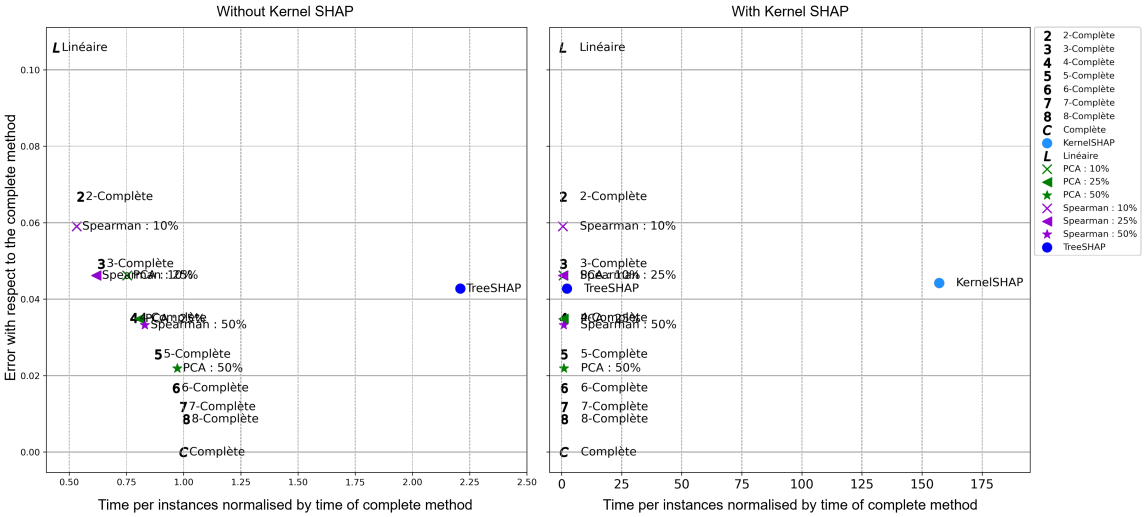


Figure 3.8: Performance maps of two *coalitional* methods, *k-depth* method, *KernelSHAP* and *TreeSHAP*.

1.5. We use default values for model hyperparameters. For explanation methods, we use Python libraries *shap* 0.40 and *lime* 0.2.0.1.

To explain the prediction of these models, we select four different explainability methods, described in Section 2.1.3, not including the *Complete* method used as a reference. In particular, we use one coalitional-based method: the *Spearman* method with a complexity threshold of 25%. Regarding *SHAP*, we use the model-agnostic *KernelSHAP* on all datasets. As this method is slow to execute if we use the whole dataset as background samples for permutations, we choose to follow *SHAP* recommendation[5] by doing a *K-Means* clustering on the input dataset and then taking the centroids as background samples. We choose $K = 10$ clusters for each dataset. In addition, for the two tree-based predictive models XGBoost and Random Forests, we use the model-specific method *TreeSHAP* by two implementations. The first one determines *SHAP* values with background samples, similar to *KernelSHAP* but optimised for tree-based methods. We use the whole dataset as background samples for this method. The second one approximates *SHAP* values by considering the trees structures and does not need background samples in input. We name it *TreeSHAPapprox*. Last, we consider *LIME*, which requires a number of perturbed samples to be created to explain each instance. We choose to set this number to 100 samples for all datasets.

All experiments were run on an Intel Xeon Gold 6230 processor with 125 GB of RAM using Python 3.9.7. All runs are performed on a single CPU core for optimisation and replicability.

To evaluate the performances of explanation methods and compare them over a wide range of datasets, we use six different metrics that only consider the influence values computed by each explainability method. In all the following definitions, let X be a given dataset with n instances and m attributes, and ϕ an explanation method that can be applied to each instance of the dataset given an ML model (that we omit for conciseness).

[5] *KernelSHAP* documentation includes a recommendation to use K-Means algorithm to speed up computation time `https://shap-lrjball.readthedocs.io/en/latest/generated/shap.KernelExplainer.html`

Number of Attributes	Number of Datasets	Number of instances		
		Min	Max	Mean
1	5	130	9100	3079
2	21	52	5456	901
3	43	60	9989	1729
4	23	96	8641	1016
5	35	62	7129	941
6	27	51	9517	949
7	33	54	4052	499
8	32	52	8192	1473
9	23	52	1473	484
10	37	57	5473	712
11	8	66	4898	942
12	12	123	8192	1175
13	5	178	506	293
All datasets	304	51	9989	1035

Table 3.2: Description of the collection of OpenML datasets used for experiments.

Computation time The first metric is the time needed for a given method to compute the local influences for the whole dataset, divided by the dataset number of instances. This includes all the steps in each method: the time to initialise the method, compute the influences for all the instances of the dataset and determine the coalitions of attributes of interest in the *Coalitional* method. Since the number of instances impacts the total time for a dataset, each computational time is normalised by dividing the time by the number of instances in the dataset to compare times per instance.

Error The second one is a quantification of the average deviation of the influence given by a method from the *Complete* method. It is the same error metric as in 3.3.1, defined in Definition 3.2.

Area Under Curve The third metric is inspired by the principle of effective complexity defined in Nguyen and Martínez (2020). However, it benefits from the absence of any parameter. It evaluates the conciseness of an explanation given the distribution of attribute importance. Attribute importance - the mean absolute value of influence assigned to instances for a given attribute- is ranked in decreasing order and then the cumulative sum is calculated. For example, in a dataset with 2 attributes, if a method gives 80% of the importance to the most important attribute (and so 20% to the second), it would have a cumulative importance proportion vector of $0, 0.8, 1$. We can then define the normalised Area Under Curve (AUC) as:

Definition 3.3. *Area under the cumulative attribute importance curve*

Let $C \in [0;1]^{m+1}$ be the cumulative importance proportion vector given by an explanation method over a dataset, with C_i the total importance proportion taken by the i-th most important attributes and m the number of attributes. We define the area under the cumulative attribute importance curve as:

$$AUC(X) = \frac{1}{m} \sum_{i=0}^{m-1} \frac{C_i + C_{i+1}}{2}$$

This metric shows whether an explanation method favours the attribution of great importance to a few attributes or, on the contrary, a more homogeneous distribution among a larger number of attributes. As this cumulative sum is sorted by decreasing value, this value is bound between 0.5 and 1. A value of 0.5 means that the explanation method gives the same importance to all attributes while a value of 1 means that the explanation method gives non-zero influences only to a single attribute, explaining the model predictions with a single attribute.

Robustness The fourth metric is a measure of the robustness of the method. A method is robust if similar instances lead to similar explanations. Formalised in Alvarez-Melis and Jaakkola (2018), we use the discrete version of the local Lipschitz estimation.

Definition 3.4. *Robustness (local Lipschitz estimation)*
Let $\mathcal{N}_\epsilon(X_i) = \{X_j \in X | \|X_i - X_j\| \leq \epsilon\}$ be the ϵ-neighbourhood of the instance X_i, with $\phi(X_i)$ the explanation vector associated to the instance X_i.

$$\tilde{L}_X(X_i) = \max_{X_j \in \mathcal{N}_\epsilon(X_i) \leq \epsilon} \frac{\|\phi(X_i) - \phi(X_j)\|_2}{\|X_i - X_j\|}$$

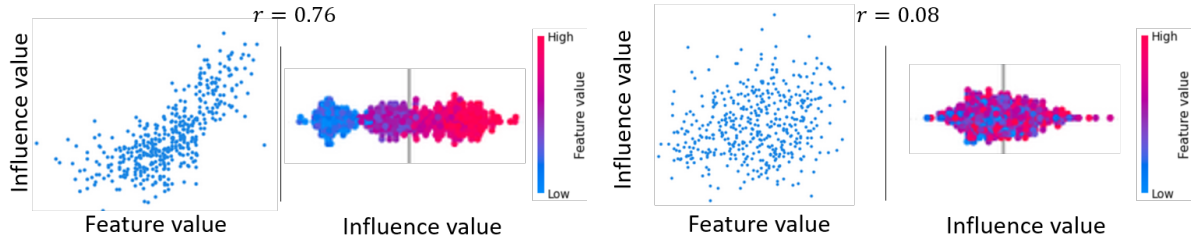
A high value of $\tilde{L}_X(X_i)$ means that the explanation method is not robust for the instance X_i over the dataset X, and a low value means that the explanation is robust for the instance X_i over the dataset X. We average this value over all instances of a dataset to get the value of the metric for a method for a dataset.

Readability The fifth metric is the global explanation readability. It is inspired by the monotonicity metric defined in Nguyen and Martínez (2020), but rather than looking at the correlation between the absolute values of the attributions and the expectations -that we cannot compute-, we look at the correlation between the data values and the influences for an attribute. Even if the explanations are calculated for each instance, we want these explanations to make sense when comparing one to another. To evaluate that, we look at the relationship between the value of an attribute, and the value of the explanation for this attribute, for all instances using the Spearman correlation coefficient r.

Definition 3.5. *Readability*
Let A be the set of attributes in the dataset, m the number of attributes, $a_i \in \mathbb{R}^n$ be the i-th attribute, $\phi(a_i) \in \mathbb{R}^n$ the explanation vector for the attribute a_i on all the instances and $r(i,j)$ the Spearman correlation coefficient of two vectors of equal size. We define the readability of an explanation method over a dataset as:

$$\mathcal{R}(A) = \frac{1}{m} \sum_{i=1}^{m} |r(a_i, \phi(a_i))|$$

In Figure 3.9, we show a visual example of what we consider readable or unreadable according to our definition.



(a) Readable explanation (b) Unreadable explanation

Figure 3.9: Examples of readable and unreadable explanations. Each dot corresponds to an instance. On the compact representation (right of each sub-figure), the colour represents the attribute value in the dataset.

Clusterability The sixth and last metric measures the pairwise attribute interaction as captured by the explanations. To do that, for each pair of attributes within a global explanation, we use a clustering method to create a partition of the explanation of all instances for the pair of attributes, and then evaluate the quality of the clustering created this way. We average this value over all pairs of attributes and name this metric (2-dimensional)-clusterability:

Definition 3.6. *Clusterability*

Let $\phi(a_i) \in \mathbb{R}^n$ the explanation of each instance for the i-th attribute, m the number of attributes, K a clustering function, and S an evaluation function for a clustering. We define clusterability as:

$$Cl(X) = \frac{2}{m * (m-1)} \sum_{\substack{i,j \in [1,\dots,m] \\ i \neq j}} S(K(\phi(a_i), \phi(a_j)))$$

A high clusterability score means that the explanation method draws relationships between pairs of attributes for their joined contribution to the predictions. For our experiments, we use K-Means as the clustering method and the Silhouette score as the clustering quality measure.

3.3.2.2 Results

In this section, following the methodology previously described, we present the results in two ways. To begin with, we aim to compare the four additive methods with one another, focusing on the *Coalitional* method. Then, with a similar methodology, we want to identify the impact of the predictive model on specific explanation methods and show if the behaviour of the *Coalitional* method differs from the other methods.

Supplementary data referenced through the rest of the section are available on Github[6]. Note that the approximate version of *TreeSHAP* is not shown for the Error and Robustness metrics because its implementation forces its *SHAP* values to be in log odds instead of probabilities, making it impossible to compare to other methods as we cannot compute distances.

[6] `https://github.com/EmmanuelDoumard/local_explanation_comparative_study`

Additive method comparison

Computation time We show in Figure 3.10 the evolution of the computation time of each method for each predictive model, averaged over datasets that share the same number of attributes. Error bars are also displayed for each mean time value. *LIME*, having a linear complexity with the number of attributes, is computationally expensive compared to other methods in low dimension (few attributes) but is less expensive than *Coalitional*-based methods and *KernelSHAP* in higher dimensions. *LIME* also seems to have very low inter-dataset time variability, resulting in smaller error bars on the graph. *Spearman* and *Complete* methods show an exponential complexity with the number of attributes, having high execution time in high dimensions, but they have a similar execution time with other methods in low dimensions. *Spearman* method execution time seems naturally correlated to the *Complete* method execution time, taking a fraction of the time (roughly 25%) of the *Complete* method. *KernelSHAP*, despite a limitation on the number of background samples, have a high execution time in high dimensions, comparable to *Spearman* and *Complete* methods for non-tree-based methods. For tree-based methods, *KernelSHAP* is slower in low dimensions, but faster in high dimensions than *Spearman* and *Complete* methods. Last, both *TreeSHAP* methods seem to have constant execution time per instance no matter the number of attributes, despite *TreeSHAP* having the greater variability of all XML methods. The approximate tree path-dependent version of *TreeSHAP* has the lowest execution time per instance.

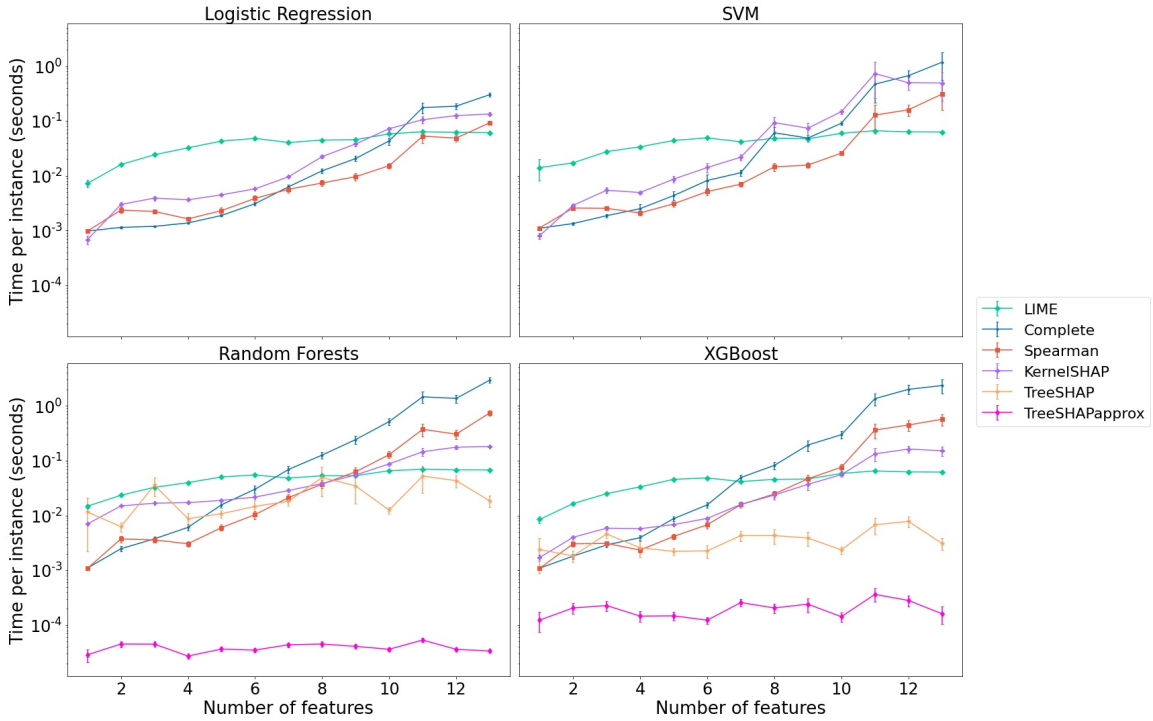


Figure 3.10: Execution time of each method per instance, averaged by number of attributes, for each model

Error Regarding the error, Figure 3.11 shows the average absolute difference in influence between each method and the *Complete* method (reference). First, we can see that overall, the more attributes there are in a dataset, the closest (measured by the second metric) the influences are to the *Complete* method. This is probably because usually, the

more attributes there are, the less influence amplitude each attribute has in the prediction. We also note that methods are ranked the same way over all the models. In low dimension (less than six attributes), *KernelSHAP* is the closest to the *Complete* method, followed by *Spearman*, while *LIME* is the farthest. In higher dimensions, *Spearman* becomes more precise than *KernelSHAP*. *TreeSHAP* (both the approximate and the data-dependent version) is more precise than *KernelSHAP*, but still less precise than *Spearman* in high dimensions.

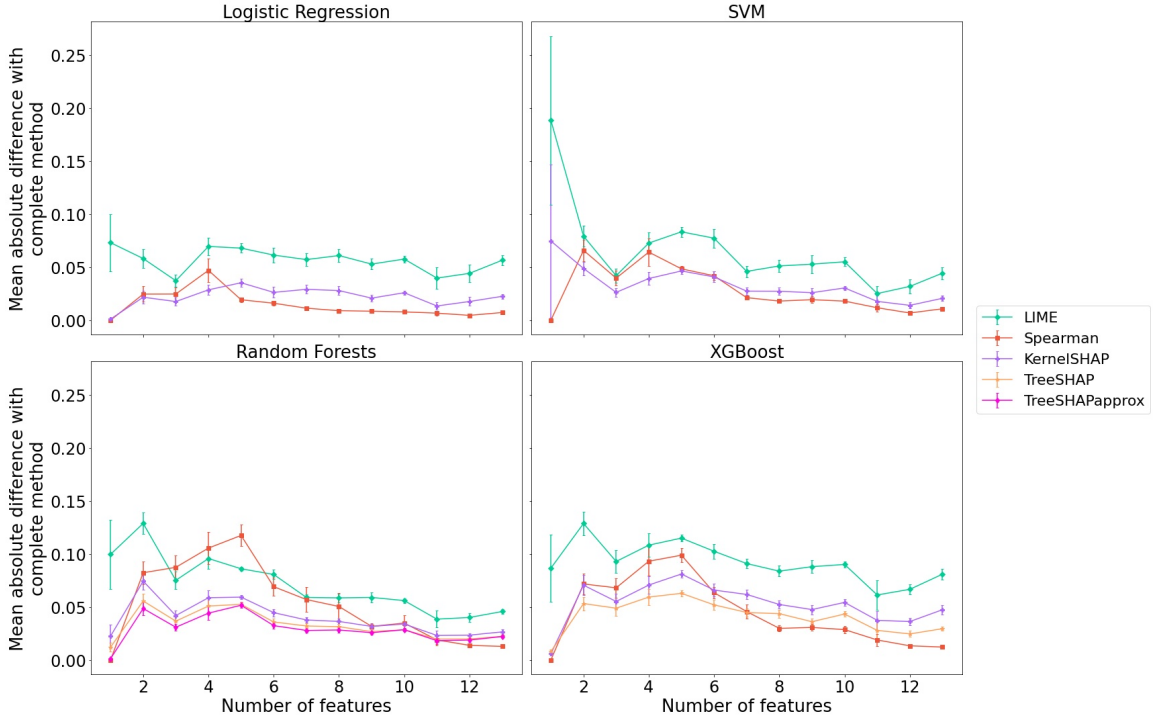
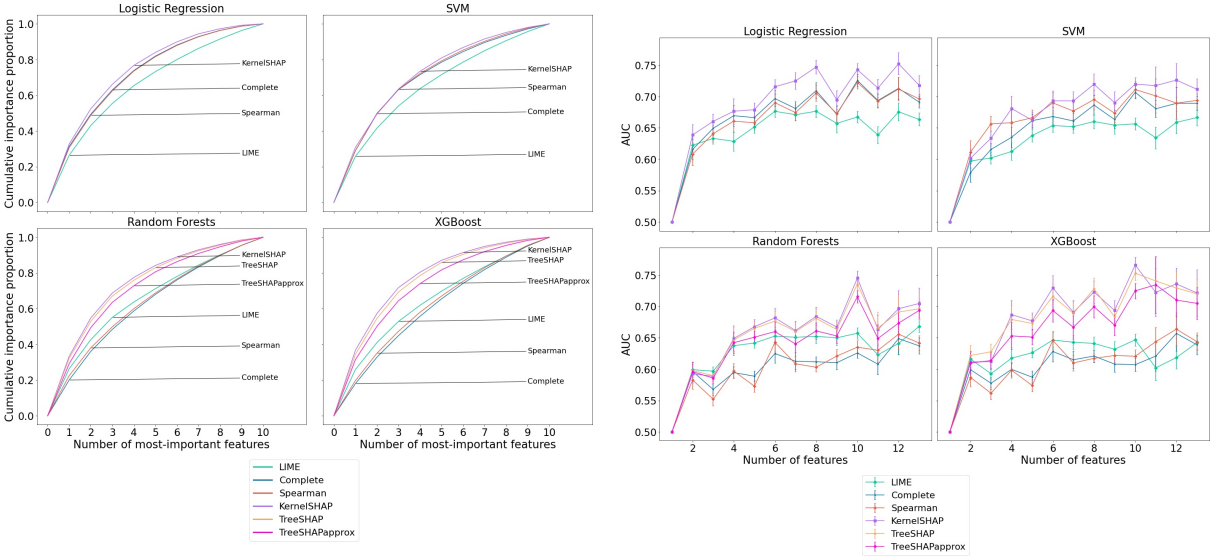


Figure 3.11: Mean absolute difference of each method with the *Complete*, averaged by number of attributes, for each model

AUC We show in Figure 3.12a the graphical representation of an example of the cumulative attribute importance proportion. The figure shows the averaging of the cumulative importance proportion of the most important attributes for the 37 datasets having ten attributes. This way, for each predictive model and each method, we obtain a curve from which we compute the third metric: the AUC of the curve. We see in the figure that some methods present steeper curves than others. For example, with Logistic Regression and SVM, *LIME* gives less proportion of the total importance to the few first most-important attributes, compared to coalitional-based and *SHAP* methods. For tree-based models, we see that *SHAP*, no matter the method, gives much more importance to the first few most important attributes than *LIME*, the *Complete* or *Coalitional* methods. Finally, *Spearman Coalitional* and *Complete* methods have close results, almost indistinguishable. According to the method for computing AUC illustrated in Figure 3.12a, we represent the average values of AUC for datasets from 2 to 13 attributes for each ML model and explanation method in Figure 3.12b. For all models, we can see that *SHAP* methods tend to produce influences with a higher AUC compared to other methods. This means that *SHAP* methods tend to assign most of the attribute importance to fewer most important attributes, while other methods tend to distribute the attribute importance more uni-

formly over all attributes. *Spearman Coalitional* and *Complete* methods seem to generate similar AUCs for attribute importance. Finally, *LIME* tends to produce influences with lower AUCs for non-tree-based methods, while it produces AUCs closer to the *Spearman Coalitional* method for tree-based methods.



(a) Example of AUC for datasets with ten attributes

(b) Average AUC for all datasets

Figure 3.12: (a) Most-important attributes cumulative importance proportion by method, for each model, for datasets with ten attributes. (b) AUC of each method, averaged by the number of attributes, for each model

Robustness Regarding robustness, we show in Figure 3.13 the local Lipschitz estimates for each model, grouped by method. We used the formula 3.4 with $\epsilon = 0.3$. We show in supplementary data that different values of ϵ did not change the relative order of results. Overall, the explanation method does not impact the robustness so much, except for *LIME* with the Logistic Regression and SVM models, for which the method is far less robust. We can also see that the *Spearman Coalitional* method is slightly less robust than the *Complete* method and *SHAP* methods, the *TreeSHAP* method being the most robust.

Readability Figure 3.14, similarly, represents the readability for each model, grouped by method. The explanation method does not impact so much readability. The *Complete* and *Spearman Coalitional* methods have a slightly lower readability than the other ones. It means that the link between an attribute and its explanations tends to be less obvious with these methods than with the others. This is possibly due to the coalitional nature of these methods: by focusing on coalitions, these methods are often able to capture complex interactions between multiple attributes, meaning that the marginal contribution of an attribute is too complex to be explained only by the attribute value. Conversely, since the attributes are considered independent by the *SHAP* and *LIME* methods, some attributes can concentrate the contribution to a prediction and virtually erase the contribution of the attributes to which they were initially correlated.

Clusterability Finally, we show in Figure 3.15 the two-dimensional clusterability of the methods applied to each model. We can see that *LIME* has significantly lower clus-

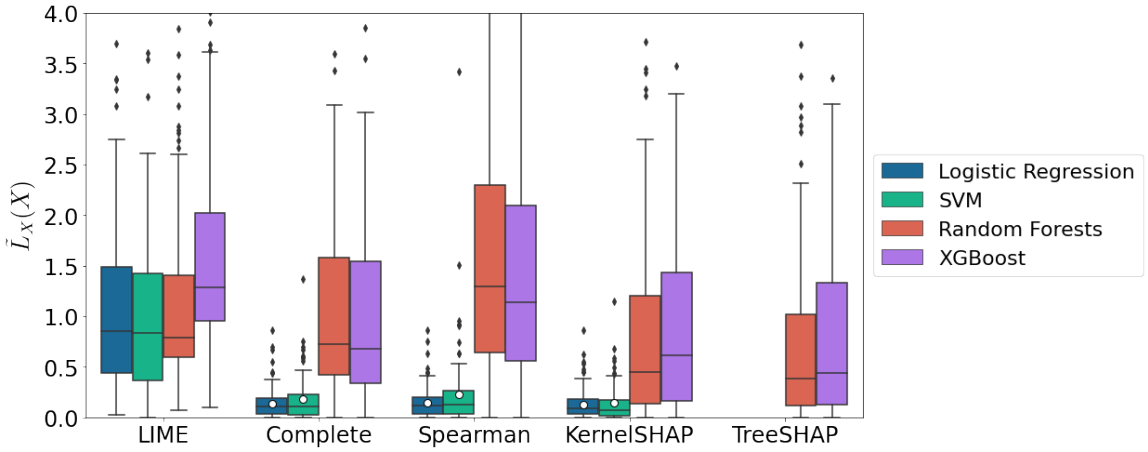


Figure 3.13: Local Lipschitz estimate for each model, grouped by method. Each box represents the results aggregated for all datasets. The white dot represents the mean value. Due to far outliers, we cropped the plot at $\tilde{L}_X(X) = 4$

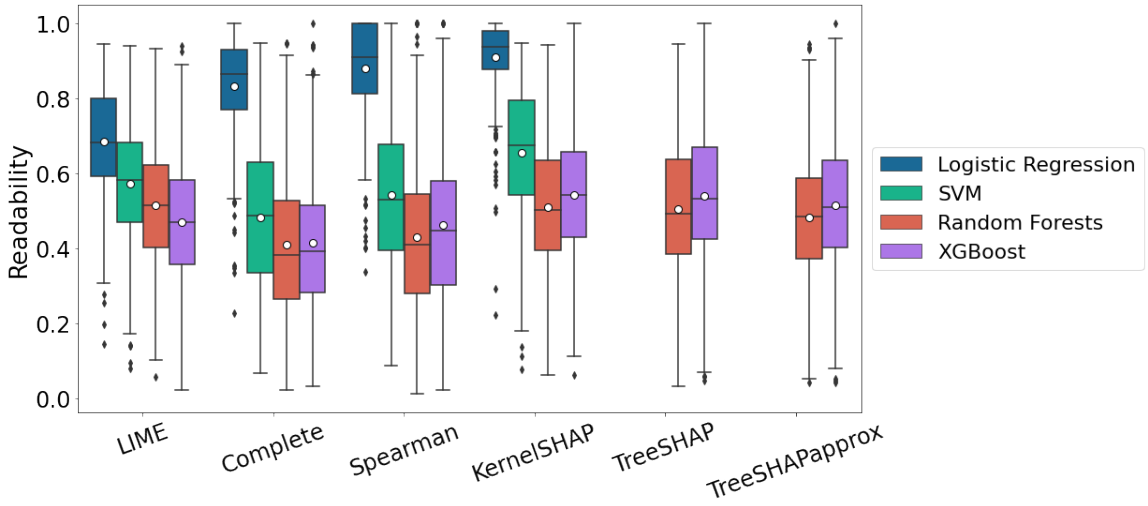


Figure 3.14: Readability for each model, grouped by method. Each box represents the results aggregated for all datasets. The white dot represents the mean value.

terability than the other methods, which have similar clusterability. It means that *LIME* tends to capture fewer interactions between pairs of attributes by groups of instances. This may be due to the discretisation imposed by *LIME* on each attribute independently of the others.

Machine Learning model explanations comparison

Computation time We show in Figure 3.16 the computational time per instance needed to compute the explanations of each predictive model for each explanation method. We can see that *LIME* execution time has almost no inter-model variability: the computation time per instance is the same no matter the model. For the other methods, the ranking of the method computational performances according to the model is roughly the same, from slowest to fastest: Random Forests, XGBoost, SVM and Logistic Regression. SVM has overall higher variability, presenting steeper curves and higher error bars. SVM

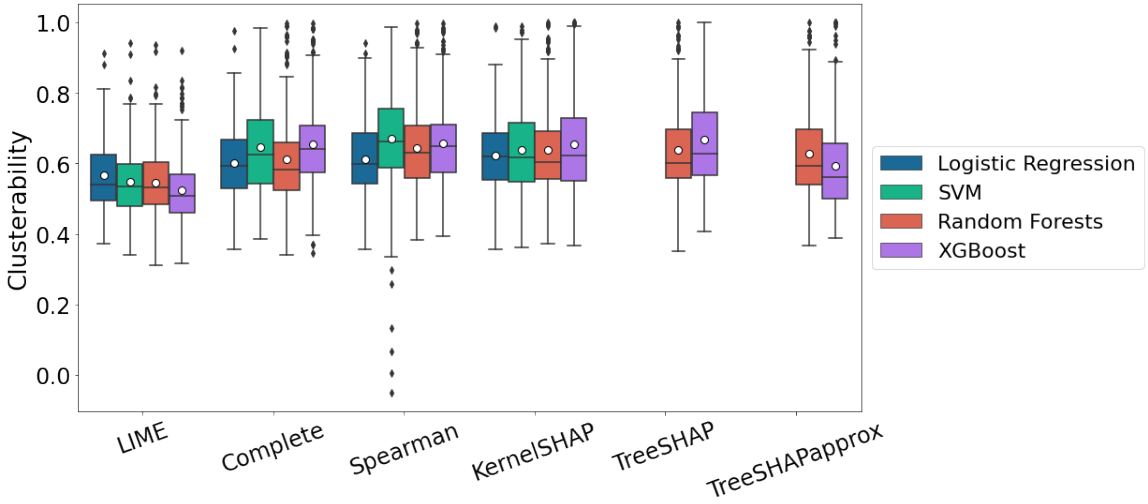


Figure 3.15: Clusterability for each model, grouped by method. Each box represents the results aggregated for all datasets. The white dot represents the mean value.

even presents outlying results when applied to *KernelSHAP* in higher dimensions. Overall, we do not observe any specific behaviour in terms of calculation time depending on the model used, except for *TreeSHAPapprox* where Random Forests are faster to compute. This may be related to the fact that *TreeSHAPapprox* only considers tree structures, as Random Forests tree structures are simpler than XGBoost ones. *Spearman Coalitional* method has a similar behaviour as the *Complete* method. In general, the faster a model is to train and predict values and the simpler it is, the faster the explanations are to compute, no matter the method.

Error We present in Figure 3.17 the error for each method for each model. The figure does not present the results for *TreeSHAPapprox* because the only relevant model for this method is Random Forests, there is no other model to compare the results with. For the three model-agnostic methods (*LIME*, *KernelSHAP* and *Spearman Coalitional*), the Logistic Regression and SVM models generate the most precise explanations compared to the *Complete* method on the same models. We can see that the explanations based on Logistic Regression are usually more precise than the SVM ones, especially in low dimensions. XGBoost explanations are less precise than Random Forest ones, except for the *Spearman Coalitional* method (similar results are observed). Overall, it seems that the simpler the model, the more precise it is in regards to the *Complete* method.

AUC Regarding the AUC, we present all the results in Figure 3.18. We observe that for *LIME* and *KernelSHAP*, there is no significant difference between the AUC of the model explanations. However, for the *Spearman Coalitional* and *Complete* methods, we can see a clear separation between tree-based methods and non-tree-based methods: the latter have higher AUC than the others. When using *Spearman Coalitional* and *Complete* methods, this means that one should be aware that different models may yield different importance distributions over the attributes. For the tree-specific methods, we can see that XGBoost generates explanations with slightly higher AUCs than Random Forests on average.

Regarding robustness, readability, and clusterability, we use the same graphs presented

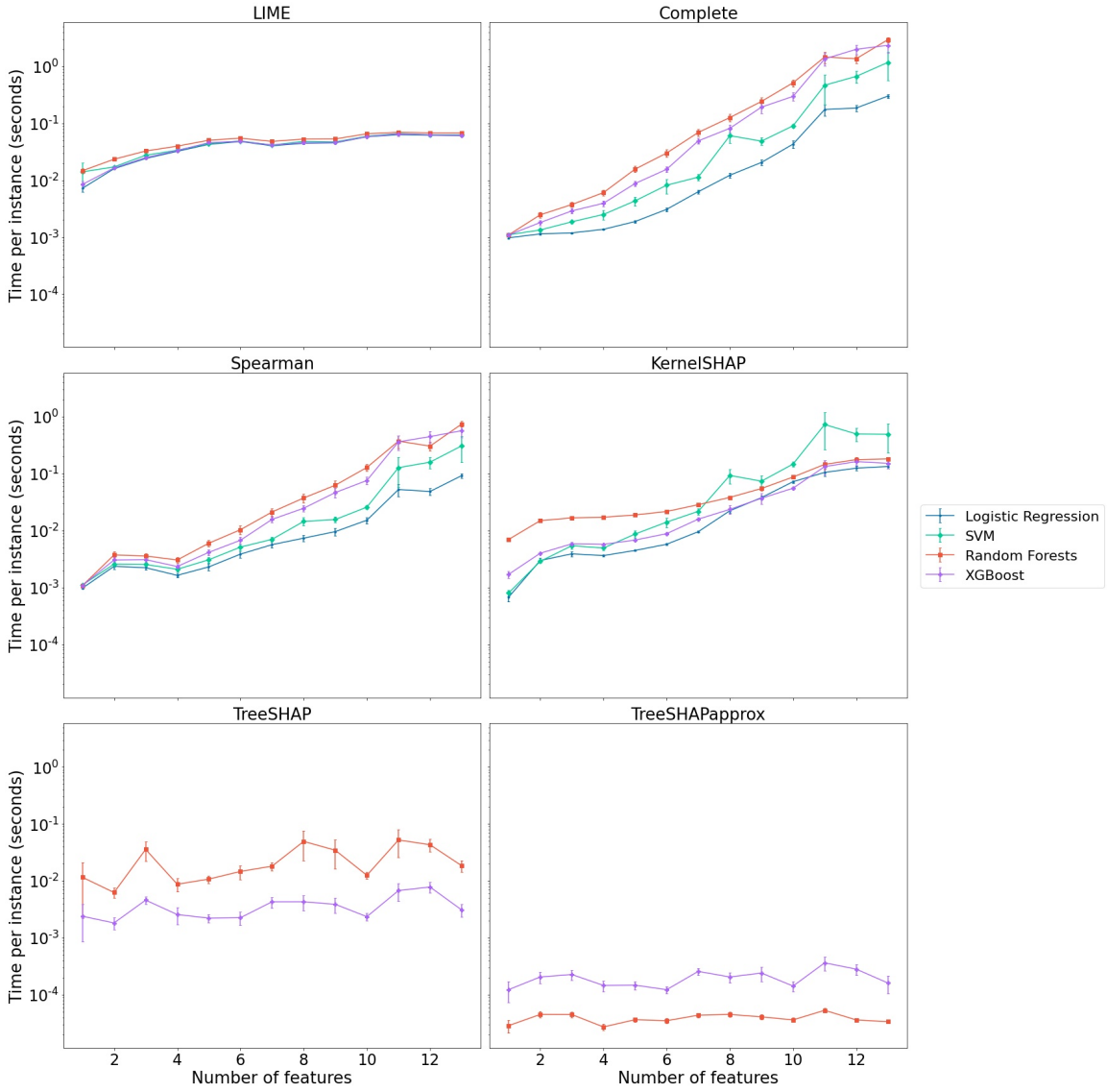


Figure 3.16: Execution time of each model per instance, averaged by the number of attributes, for each method

in the previous part to analyse the impact of the model on the explanations regarding these three metrics.

Robustness We look at Figure 3.13 to compare the robustness of the methods applied to each model. We confirm that, except for LIME, the Logistic Regression and SVM models produce much more robust explanations than the Random Forests and XGBoost models for all XML methods. This is probably tied to the complexity of the models. On one hand, a more complex model is usually harder to explain even for model-agnostic explanation methods, and on the other hand, a more complex model leads to highly non-linear functions, meaning that instances that are close to each other may have different predictions and therefore, different explanations.

Readability For readability, Figure 3.14 shows that the explanations made on the Logistic Regression model are much more readable than the ones on the other models. Explanations made on the SVM model fall between the Logistic Regression and the tree-based models in terms of readability for most explanation methods. This is probably

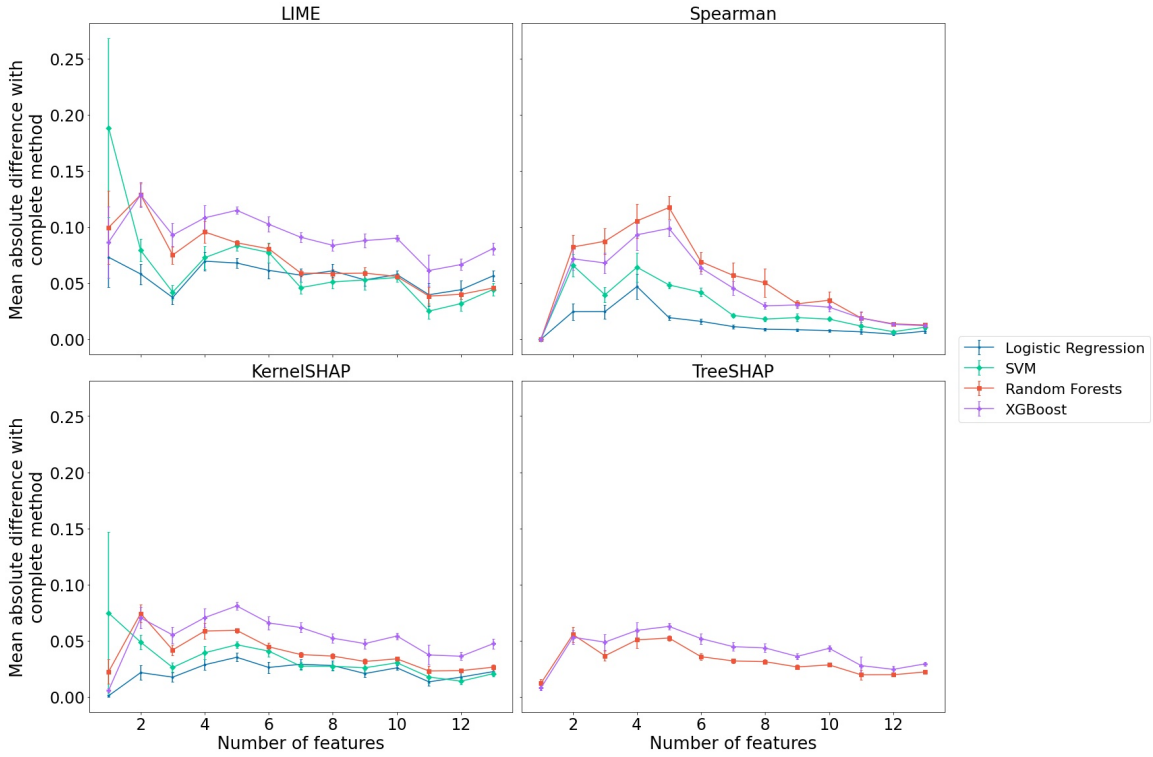


Figure 3.17: Mean absolute difference of each method with the *Complete*, averaged by number of attributes, for each model

because simpler models tend to draw relationships between individual attributes and the output without necessarily considering the interaction between attributes, producing explanations that can be read attribute by attribute.

Clusterability Finally, we look at the clusterability of the explanation applied to the ML models by looking at Figure 3.15. We can see that all models have similar clusterability between them. This may indicate that the model is not important in determining particular sub-populations of explanations by pairs of attributes, or that it depends more on the considered dataset than on the model.

3.4 Medical examples and exploration of explanation methods' hyperparameters

In this section, we provide two medical examples with two different objectives. First, an example of how to use and interpret explanations with a dataset on the COVID-19 disease published in Ferrettini et al. (2021). Then, an exploration of local attributive explanation methods hyperparameters and behaviours on the SA-Heart dataset, published in Doumard et al. (2023). Our global objective is to focus on specific examples that a user could face while analysing their data and building or using explainability tools.

3.4.1 Medical Example: Covid-19 dataset

In this example, we show one use of our *Coalitional* methods on a real use case dataset and for specific instances from this dataset. We explore what exploring explanations

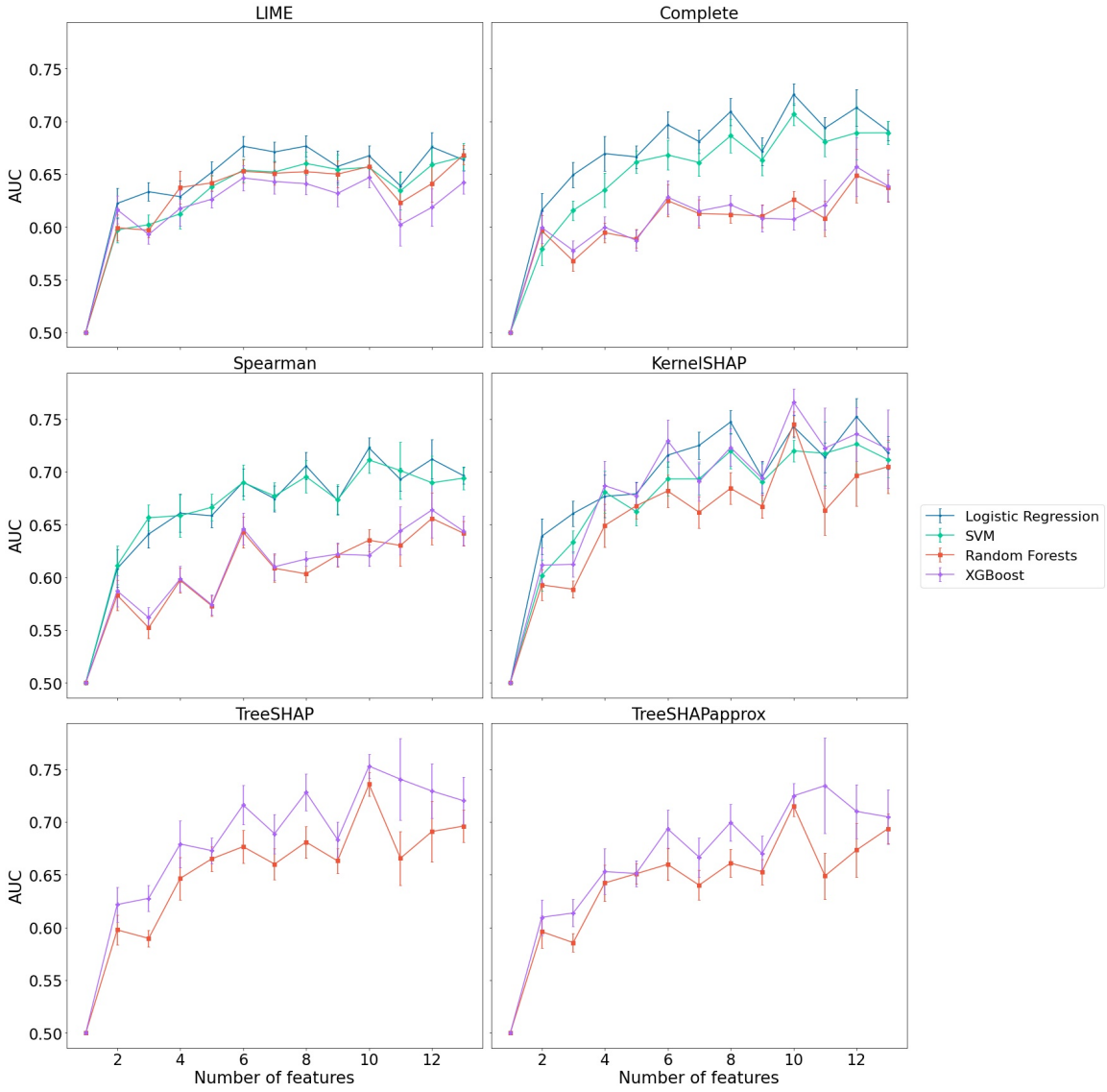


Figure 3.18: AUC of each model, averaged by number of attributes, for each method

could mean for medical professionals. We also include the most used explanation method, *KernelSHAP*, for comparison.

As seen in Section 2.1, the quality of an explainability method is a subjective concept and it would be difficult to theorise measures to assess what constitutes good explainability. Nevertheless, some criteria exist in the literature to evaluate individual explanations (Robnik-Šikonja and Bohanec, 2018). Properties such as fidelity and comprehensibility can help non-experts to evaluate and compare individual explanations, thus explanation methods. Fidelity represents the ability of an explanation to approximate the prediction of the "closed-box" model and comprehensibility evaluates the ability of users to understand the explanations.

The use case dataset concerns the SARS-COV2 - also called COVID-19 - epidemic outbreak in France during the 2020 Spring. Data collection complied with the European GDPR rules and consists of anonymised medical information of 409 patients with Covid-19 virus hospitalised at the Centre Hospitalier Intercommunal de Créteil [7] between March

[7]The use-case dataset was acquired in collaboration with the Centre Hospitalier Intercommunal de

and May 2020. The primary binary outcome consists of the deterioration of the patient's state of health during their stay, also called aggravation. Deterioration was defined as the occurrence of septic shock or acute respiratory distress syndrome, the need for mechanical ventilation or resuscitation during hospitalisation, or in-hospital mortality. Out of the 409 patients, 176 of them had a deterioration in their health state, i.e. 43% of the data set. Each patient profile is established upon the patient's arrival at the hospital. Available information consists of 10 attributes such as basic characteristics (age and gender), exam results of Chest Computed-Tomography (CT) scan severity, and comorbidities like cancer, type-2 diabetes, obesity, intellectual disability, and cardiovascular disease. For this use case, a *Random Forest* model and the *Spearman Coalitional* method with a complexity threshold of 25% are used. The model has an accuracy of 74% with an 80% precision and a 69% recall.

Figure 3.19 and 3.20 give the average absolute influence of each attribute, with or without taking into account the class predicted by the model, for the *Spearman Coalitional 25%* and *KernelSHAP* method respectively. Age and Chest scan severity are the two most important attributes for both methods, with Chest scan severity having a greater impact on aggravation class. This shows a coherence between the medical reality and both explanation methods. Indeed, a high Chest scan severity is strongly associated with an aggravation of the health state as shown in Francone et al. (2020). Both methods also have different results for other attributes, such as cardiovascular disease, cancer and mental disability that have on average almost no impact with *KernelSHAP* and all attributes have on average a higher influence with the *Spearman Coalitional* method. Taking into account classes, the average influences for both classes are relatively similar using *KernelSHAP*, except for the age and severity of the chest CT scan. With the *Spearman Coalitional* method, the average influences of ageusia anosmia, diabetes and insulin treatment are dissimilar. For older patients with high chest scanner severity, type-2 diabetes, insulin treatment, or ageusia anosmia, the model is likely to predict a higher risk of deterioration with *Spearman Coalitional* since the average absolute influence of these attributes is higher for the aggravation class.

All these behaviours from our model are coherent with the clinical literature about COVID-19 (Zheng et al., 2020). In contrast, with the *KernelSHAP* method, the near-zero average influences for some attributes are inconsistent with known risk factors.

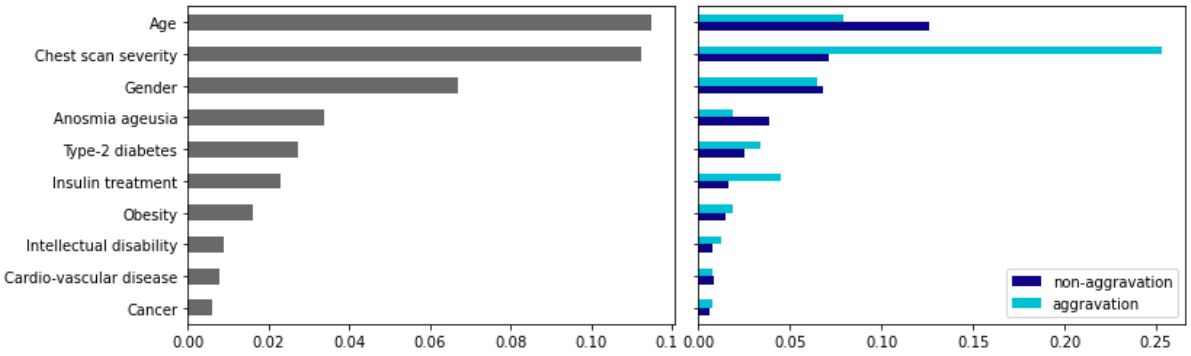


Figure 3.19: Mean absolute influence for each attribute with *Spearman Coalitional* 25% method. (left) for both classes, (right) for each class separately.

Créteil. Therefore, we greatly acknowledge the managers and physicians involved in this project. `https://www.chicreteil.fr/`

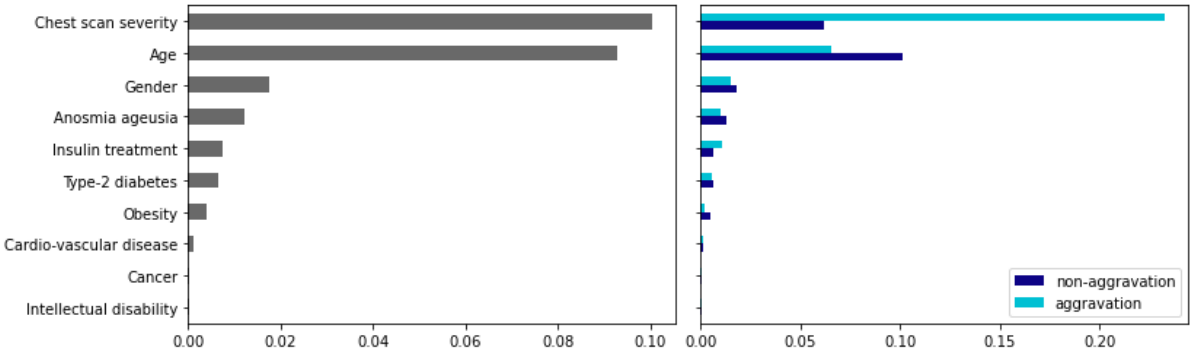


Figure 3.20: Mean absolute influence for each attribute with *KernelSHAP* method. (left) for both classes, (right) for each class separately.

Another important point of the explanations is the fidelity and ease of understanding and interpreting them. Although very subjective, these parameters are essential to take into account in the medical field, since a lack of fidelity to the model and understanding of the explanations can lead to wrong decision-making and consequences for the health of patients. To evaluate this, one instance of each class from the Covid-19 dataset was randomly drawn to describe and evaluate the explanation of the *KernelSHAP* and the *Spearman Coalitional* method. Figures 3.21 and 3.22 show the influence of each attribute for these patients, whose descriptions are given below. Patient A is a 54-year-old obese person with no clinical signs of infection in their chest CT scan. This patient also has insulin treatment and signs of ageusia or anosmia. The two methods find that the value of Chest CT scan severity and age for this patient contributes the most to the prediction of non-aggravation while their gender, their symptoms of anosmia and ageusia, their obesity and their insulin treatment go against the prediction. The explanations allow us to understand that this patient has many risk factors and that the non-aggravation prediction comes mainly from the absence of severity of the chest CT scan and the patient's age. However, for the *KernelSHAP* method, the absence of cardiovascular disease goes against non-aggravation prediction while it contributes to the prediction for the *Spearman Coalitional* method. This seems contrary to medical knowledge about COVID-19 (Zheng et al., 2020) since cardiovascular disease is a risk factor. The absence of disease should therefore be in favour of a non-aggravation of the patient's state of health.

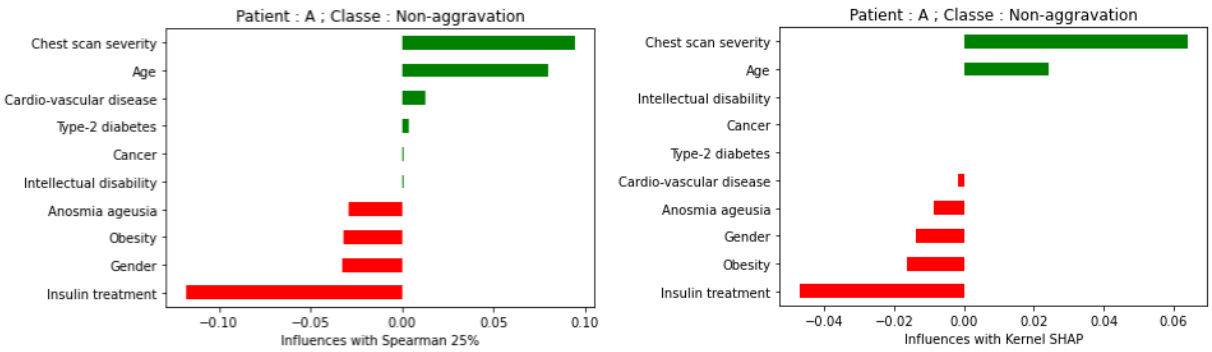


Figure 3.21: Influences of patient A with *KernelSHAP* and *Spearman Coalitional* 25%.

Patient B is a 76-year-old person with type-2 diabetes, insulin treatment, and ageusia anosmia. The severity of their chest CT scan is 4 out of 4 which is a critical value. For

KernelSHAP method, the chest scan severity is way more important than other attributes in the prediction. For *Spearman Coalitional* method, even if the severity of the chest CT scan is significant, the presence of insulin treatment, the patient's gender and age are important. The absence of cancer, cardiovascular disease, intellectual disability, and obesity goes against the prediction, while there is no impact with *KernelSHAP* method. *Spearman Coalitional* explanations are slightly more contrasted than *KernelSHAP* ones.

For this use case, the two methods are easy to understand as they are based on the same additive strategy. For both methods and both examples, influences approximate closely model predictions and therefore have a high fidelity. However, this fidelity is only local, as methods only explain individual instances. Moreover, based on the clinical literature about COVID-19 (Zheng et al., 2020), the explanations from *Spearman Coalitional* method seem more consistent for comorbidities. Finally, the *Complete* method dataset was computed in 51 seconds with the *Spearman Coalitional* method when it took more than 18 minutes for the *KernelSHAP* method, for similar results.

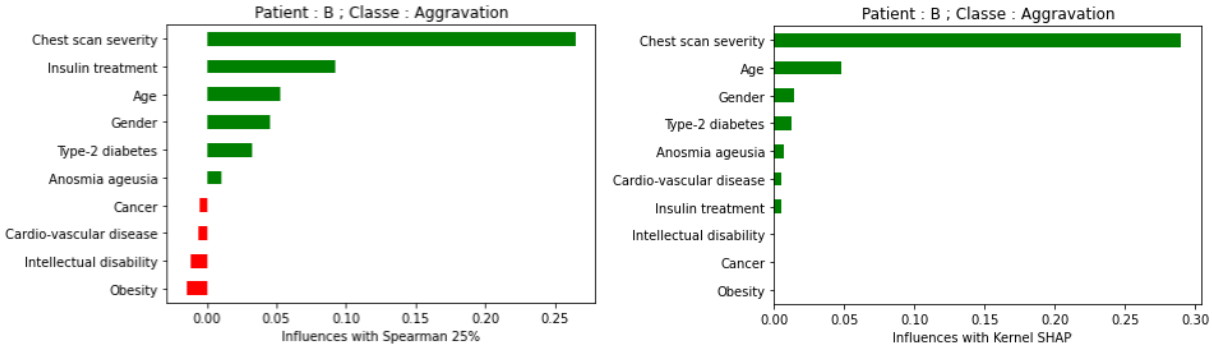


Figure 3.22: Influences of patient B with *KernelSHAP* and *Spearman Coalitional* 25%.

3.4.2 Medical Example: SA-Heart dataset

SA-Heart is a dataset extracted from a larger database of South Africans detailed in a 1983 study (Rossouw et al., 1983). The extracted dataset is a retrospective sample of males in a heart-disease high-risk region of the Western Cape, South Africa. The dataset is composed of 462 individuals for 10 attributes. The main objective is to predict the binary target attribute '**CHD**', a coronary heart disease, according to 9 explanatory factors: **tobacco** (cumulative consumption tobacco), **age** (at the onset), **LDL** (low-density lipoprotein cholesterol), **adiposity** (estimation of the body fat percentage), **obesity** (through the body mass index), **family** (family history of heart disease, present or absent), **alcohol** (current alcohol consumption), **SBP** (systolic blood pressure) and **type-A** (Type-A behaviour scale). After model training, the different explanatory profiles obtained between the different methods of explanation are compared. By considering a reflection on the end-user side, the health care practitioners, explanatory profiles should be used 1) at the population level (global explanations), for example, to highlight high-risk patient profiles, develop new prevention programs, develop new physio-pathological hypotheses but also 2) at the instance level (local explanations), for personalised medicine.

For conciseness, we limit the analysis to a single Machine-Learning model. We choose Random Forests, as every explanation method that we consider applies to it. We present the results with SVM, Logistic Regression and XGBoost models in supplementary data.

In Table 3.3, we show the values of each metric on the SA-Heart dataset. To enforce the robustness of the results, we calculated the explanations 10 times for each method and averaged the metrics. *TreeSHAP* looks promising, giving the best score in AUC, Robustness and Clusterability while maintaining correct performances in Computation Time, Error and Robustness. Confirming trends seen in the previous section, *Spearman Coalitional* is the most precise method compared to the *Complete* method, the approximate version of *TreeSHAP* is the fastest, and *LIME* produce the most readable explanations.

	LIME	Complete	Spearman	KernelSHAP	TreeSHAP	TreeSHAPapprox
Time per instance	0.062	0.141	0.036	0.061	0.011	**<0.001**
Error	0.046	0.000	**0.026**	0.034	0.029	0.033
AUC	0.604	0.560	0.550	0.623	**0.625**	0.614
Readability	**0.686**	0.499	0.427	0.679	0.652	0.621
Robustness	0.116	0.099	0.146	0.086	**0.080**	0.095
Clusterability	0.460	0.485	0.506	0.521	**0.522**	0.520

Table 3.3: Metrics applied to explanations of Random Forests on SA-Heart

To compare the explanations of the different additive methods, we look at global explanations given by each method. We use *SHAP*-like representations to visualise global explanations by aggregating local explanations on the same representation. This way, we build different figures. The first one, in Figure 3.23, represents a global explanation of the predictive model, given by each explanation method, by plotting the explanation profile of each attribute on a separate line. For each method, the attributes are sorted in decreasing attribute importance, the top one being the most contributing attribute on average, while the bottom one being the least contributing attribute on average. For each attribute, each dot represents an individual from the dataset, its colour representing the value of the associated attribute. Its position on the x-axis represents the contribution of the attribute to the prediction of this individual, and overlapping dots are spread on the y-axis.

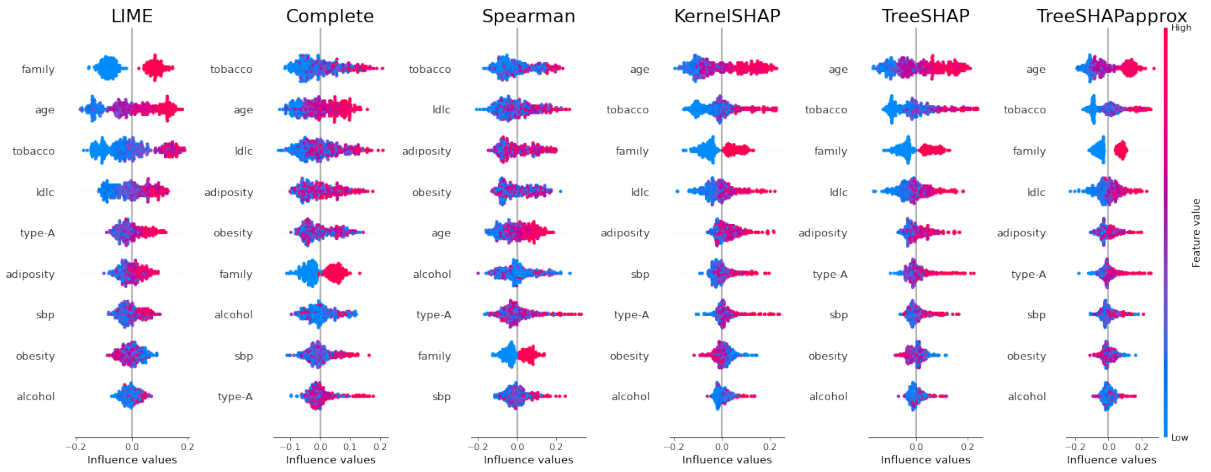


Figure 3.23: Summary plots of each method on the SA-Heart dataset

We can see that most of the attributes have similar ranking among the different methods: tobacco and age are the two most important attributes except for the *Spearman Coalitional* method which ranks age 5$^{\text{th}}$. On the opposite side, alcohol, SBP, and type-A

are always in the 4 least important attributes. These attributes have also similar explanation profiles. Conversely, some other attributes exhibit more marked differences depending on the methods. The most important difference is observed in the binary attribute family history of heart disease. This attribute is assigned fairly low importance by the coalitional-based method, relatively high importance (3rd most important attribute) by *SHAP* methods, and very high importance by *LIME* (most important attribute). Obesity and adiposity have also different influences depending on the method: obesity is ranked second least contributing by *LIME* and *SHAP*, but more important by the coalitional-based methods. It is important to note that obesity and adiposity are highly correlated (Spearman correlation r=0.72). We hypothesise that it may be the reason for such differences. Overall, the three *SHAP* methods give similar explanations and have almost identical rankings of the attributes. From a global perspective, we can also see that *SHAP* and *LIME* present a more homogeneous "gradient" of colours for the explanations, whereas coalitional-based methods present mixed-up colours in the explanations. This means that *LIME* and *SHAP* explanations are more locally monotonic, in the sense that the influence value of an attribute for an individual is more locally correlated to the value of the attribute for *LIME* and *SHAP* than it is for coalitional-based methods. This also illustrates well the values of readability seen in Table 3.3.

The second visualisation that we present is Partial Dependence Plots (PDP). PDPs focus on the relationship between an attribute and the influence of this attribute on the model prediction by plotting each pair of attributes and influence values on a 2-dimensional axis. We compare the PDPs of several important attributes in Figure 3.24.

Looking at the PDPs for the **age** attribute, we show that *LIME* seems to form clusters of points around specific cut-off age values. To a lesser extent, this phenomenon can also be seen in the other *SHAP* methods. Conversely, coalitional-based methods have similar PDPs and do not seem to find such cut-offs. However, it seems to be a special behaviour of the explanation at specific ages. For example, subjects around 50 years have a marked lower contribution of this attribute to the prediction of the presence of coronary heart disease than people even slightly younger or older. This may hint at an over-fitting of the ML model that would not have been captured by the other explanation methods. The explanation of the tobacco attribute also largely differs among explanation methods. Where all the methods agree on attributing a low value to non-smoking individuals, the evolution of the contribution varies with the quantity of tobacco. Once again, *LIME* and *SHAP* explanations seem to find a cut-off value for tobacco consumption, of around 7 and 9 respectively, while coalitional-based methods capture a non-monotonic, more complex relationship.

We also look at adiposity PDPs. Once again, the three *SHAP* explanations are close to each other. Interestingly, they capture a non-monotonic relationship between the attribute and the outcome, giving people around 30% of adiposity a higher influence for this attribute (in absolute value) than people close to this value. This relationship seems to be captured to a lesser extent by coalitional-based methods, but not captured at all by *LIME*. We also note that the *Complete* and *Spearman Coalitional* influences are more scattered, which means that more variance exists amongst subjects of the same adiposity for these methods than for the others.

Lastly, looking at obesity PDPs, *LIME* and *SHAP* methods find a negative relationship between obesity and CHD prediction. This seems counter-intuitive, as obesity is a strongly known comorbidity factor of cardiac disease. As previously mentioned, obesity and adiposity are strongly correlated (r=0.72), and this may be the reason for such ob-

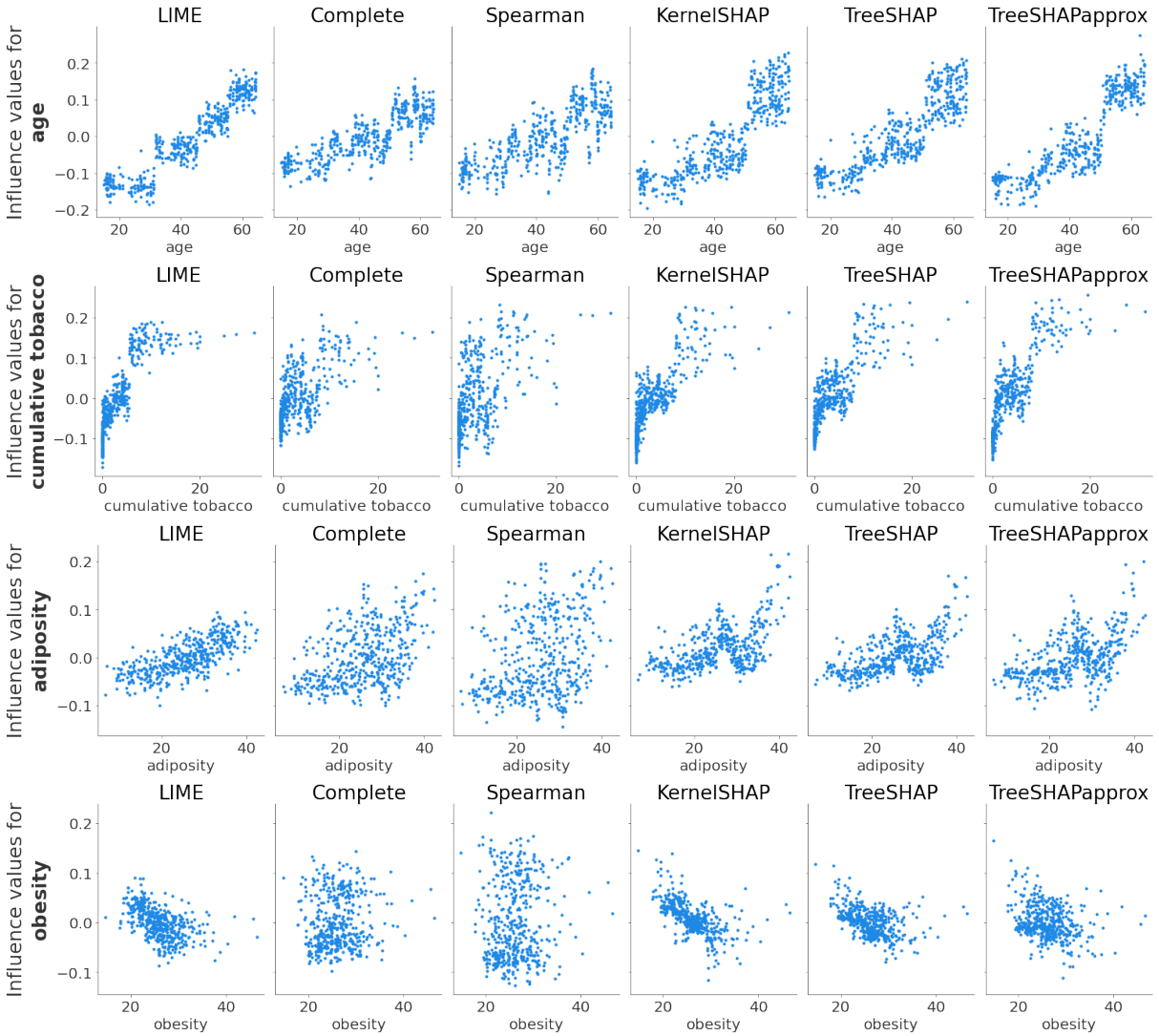


Figure 3.24: Partial dependence plots of age, tobacco, adiposity and obesity for each method

servation. Furthermore, we have mentioned in section 2.1.3 that *SHAP* works under the hypothesis that attributes are independent, but with such correlation, it is very unlikely that obesity and adiposity are independent. To better understand the relationship between these two attributes, as found by the methods, we plot in Figure 3.25 the influence values of adiposity and obesity given by each method.

The *Complete* and *Spearman Coalitional* methods seem to find a positive correlation between the influences of the two attributes: when an individual is assigned a high influence value for obesity, a high influence value for adiposity is usually assigned, and conversely. We can even distinguish two clusters of individuals: one for individuals that have a high influence value for both attributes and one for individuals that have a low influence value for both attributes. Such patterns are not found by *LIME* or *SHAP*, thus confirming the lack of ability of these methods to consider dependent attributes. This shows the limits of clusterability as a global metric to evaluate explanations. As seen in Table 3.3, on this dataset, the three *SHAP* methods have an overall higher clusterability than coalitional-based methods. However, when we consider pairs of attributes individually, we see that coalitional-based methods can capture clusters that *SHAP* fails

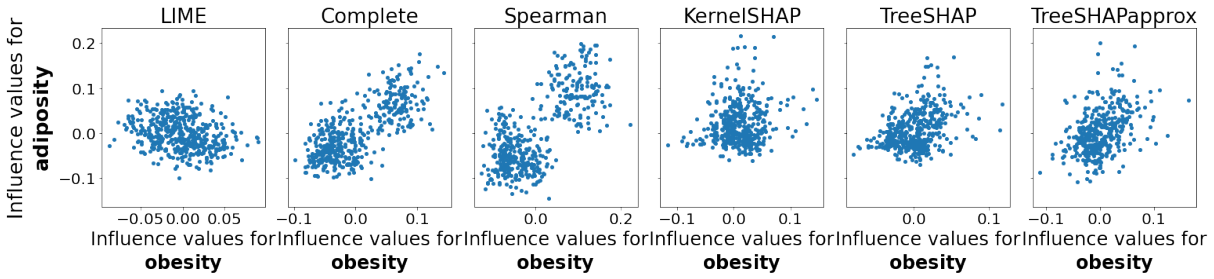


Figure 3.25: Influence value of adiposity against the influence value of obesity

to capture.

On a more global scale, we see that *LIME* and *SHAP* produce explanations that are easier to read at first glance compared to *Complete* and *Spearman Coalitional* explanations. However, *LIME* and *SHAP* seem to capture different cut-offs and relationships, and it is hard to confirm such values without further biological knowledge. Coalitional-based methods seem to produce explanations that are harder to read on a global scale, but more precise at an individual level and able to take into account the dependencies between attributes. PDPs for all attributes are available in supplementary data.

3.4.2.1 Hyper-parameter exploration

Most explanation methods have several parameters that can change the way the explanations are generated, and so their values. Previously, we showed results for a single set of parameters for each method. In this section, we present new results on the SA-Heart dataset by taking different values of several parameters. For conciseness, we present the results for only a single model, Random Forests, although we observe similar results on the other models as well.

LIME The first parameter we investigate is one of *LIME* most important parameters: the number of samples drawn from the distribution to generate the local linear model to explain an instance. Its default value is 5000, but as the computation time scales linearly with this number, we limited this number of samples to 100 in our previous experiments. In Figure 3.26, we visually show the effect of different values of this parameter on the explanations.

We can immediately see that the number of samples impacts the global explanations for the dataset. However, looking at the relative importance of the attributes, knowing they are sorted in descending importance from top to bottom, we can see that this parameter does not change the attribute importance so much. As the number of samples increases, we can see for each attribute that *LIME* explanations are grouped by attribute values around specific influence values. This creates vertical stripes that get thinner when the number of samples increases for each explanation. To have a better visualisation of this phenomenon, we look at the partial dependence plot of the **age** attribute in Figure 3.27. *LIME* tends to discretise the age values, with the influence values becoming increasingly grouped and homogeneous as the number of samples increases. This goes to an extreme case when taking 10 000 samples, with the same influence values for an entire age group. As we see the age category of an individual defines almost entirely the influence value given by *LIME* for this attribute. This can be an incorrect explanation, as this would mean that the model does not consider any interaction between the age and other attributes to make

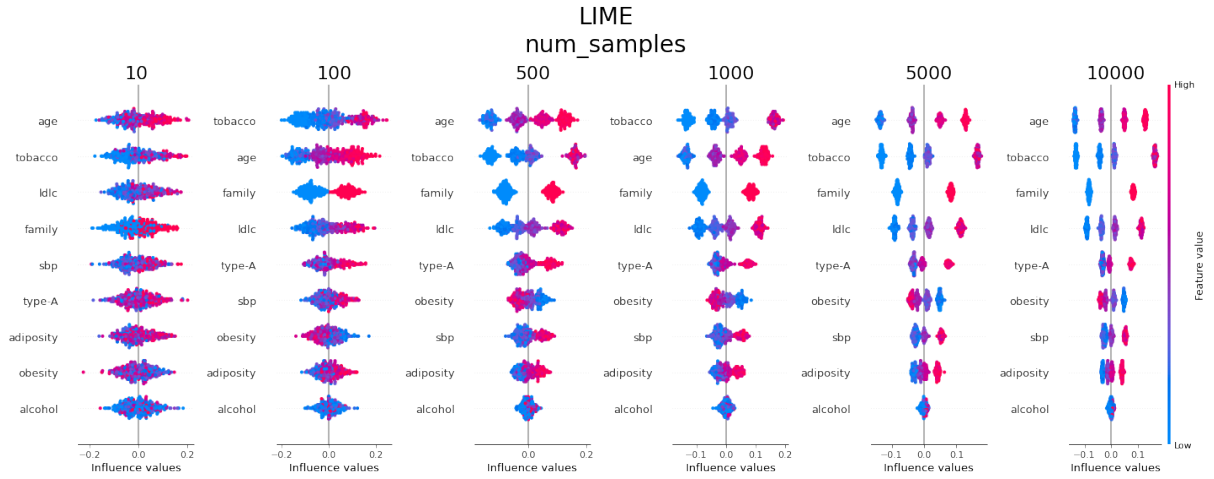


Figure 3.26: Summary plot of the explanations given by *LIME* on the SA-Heart dataset with different values for the number of samples drawn to create the local model for each explanation.

a prediction, while we know that Random Forests use tree depth and node successions to take into account the relationship between attributes. This would also mean that the model does not have enough granularity to consider the attributes as *continuums* and instead considers only categories, which again is certainly incorrect regarding Random Forests. However, when looking at the relationship between the number of samples and the local Lipschitz estimate in Figure 3.28, the robustness increases with the number of samples per explanation. This underlines the limits of robustness and, to a broader extent, the limits of objective metrics to evaluate the explanations. Despite being systematically measurable on all the explanations, they must be taken as a whole to qualify and compare explanations. Human and expert reading is always necessary to validate the quality of the explanations.

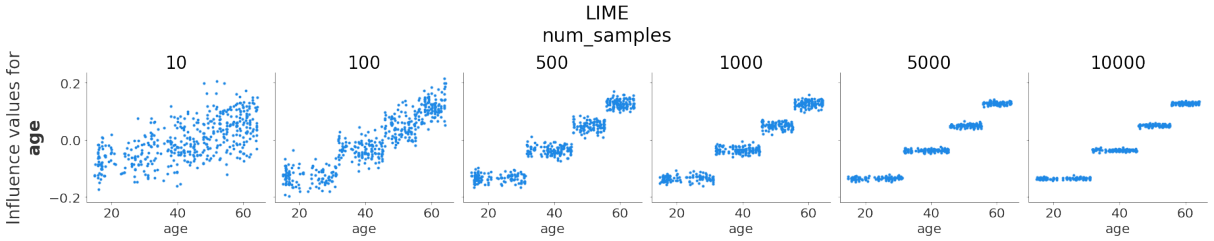


Figure 3.27: Partial dependence plot of the explanations given by *LIME* for the attribute **age** on the SA-Heart dataset with different values for the number of samples drawn to create the local model for each explanation.

Another important parameter for *LIME* is the kernel width. With *LIME*, the kernel weights the sampled instances by their distance from the instance to be explained to create the local surrogate linear model. The farthest the drawn sample is from the instance, the less weight it has in the local linear model. This enforces the notion of locality for the linear model and the higher the kernel width, the less local the linear model is. Visani et al. (2022) insist on the trade-off between stability (the equality of the local model coefficient through repeated trials) and adherence (the R^2 performance of the local model). The article shows that the value of the kernel width mainly determines these trade-offs. The

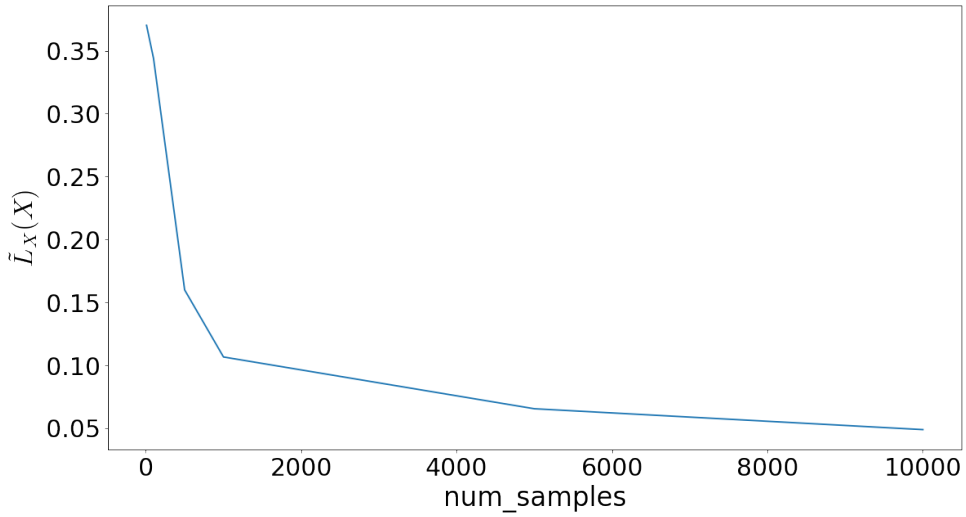


Figure 3.28: Local Lipschitz estimate of *LIME* explanations on the SA-Heart dataset according to the number of samples drawn for each explanation.

base value is $0.75 \times \sqrt{d}$ with d the number of attributes (2.25 for the SA-Heart dataset)[8]. Figure 3.29 shows a partial dependency plot for the age attribute for different values of kernel width. We observe very similar results for each attribute (which can also be seen in the supplementary figures).

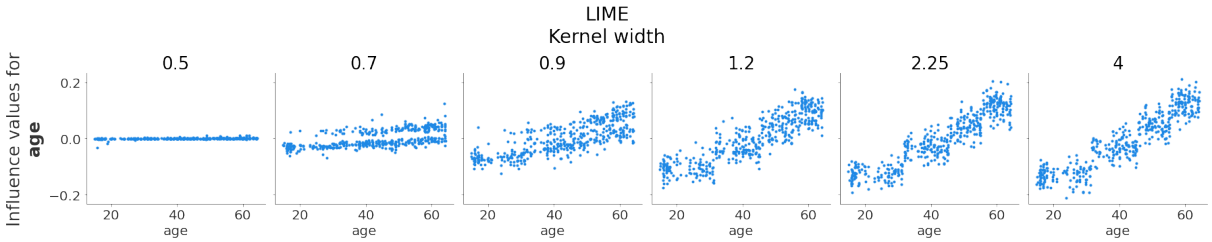


Figure 3.29: Partial dependence plot of the explanations given by *LIME* for the attribute **age** on the SA-Heart dataset with different values for the kernel width.

We can see that the main impact of the kernel width is the amplitude of the explanations: lower values of kernel width result in flattened values of influence that mix in an unreadable fashion, while higher values of kernel width lead to the usual "boxes" that *LIME* creates for the explanations. The default value (2.25 for SA-Heart) seems to be on the higher end. This could mean that the default value of kernel width makes the linear model not local enough, giving high weight to samples far from the instance we want to explain. For this dataset and model, a more appropriate kernel width value may be closer to 1.

Spearman Coalitional For the *Spearman Coalitional* method, we look at its single parameter: the proportion of subsets of attributes (or coalitions) taken into account to compute the influences. This parameter is called *complexity rate* and goes from 0 excluded (we need at least a coalition) to 1 included. A complexity of 1 gives the same algorithm as

[8] *LIME* documentation: `https://lime-ml.readthedocs.io/`

the *Complete* method. Figure 3.30 show the partial dependence plot of the age attribute for different complexity values. Once again, we see very similar behaviour to the other dependence plots.

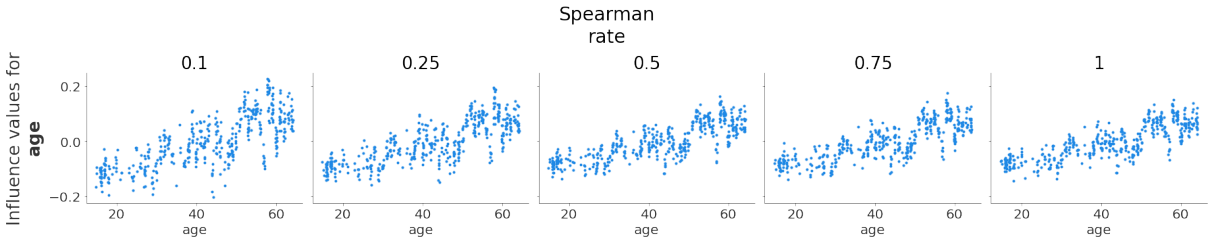


Figure 3.30: Partial dependence plot of the explanations given by the *Spearman Coalitional* method for the attribute **age** on the SA-Heart dataset with different values for the complexity rate.

We can see that a higher complexity produces less scattered explanations. However, after 0.5, the change is barely visible. We can conclude that the complexity effectively controls the degree of approximation, and on this dataset and model, the *Spearman Coalitional* method with a complexity of 0.25 is a good approximation of the *Complete* method, and it is a good approximation for rates of 0.5 and more.

SHAP Next, we look at the *KernelSHAP* method. With this method, a so-called "background" dataset must be used to provide relevant samples to train the XML method. However, this can slow down the process considerably as it creates samples around each background instance. As advised by the documentation, if the method takes too much time to compute, we can use a clustering method (namely KMeans) to extract the few most relevant samples in the training dataset to represent the data distribution. We then refer to this number of "most relevant samples" as "Number of background samples". Figure 3.31 show the age attribute dependence plot for different values of the number of background samples.

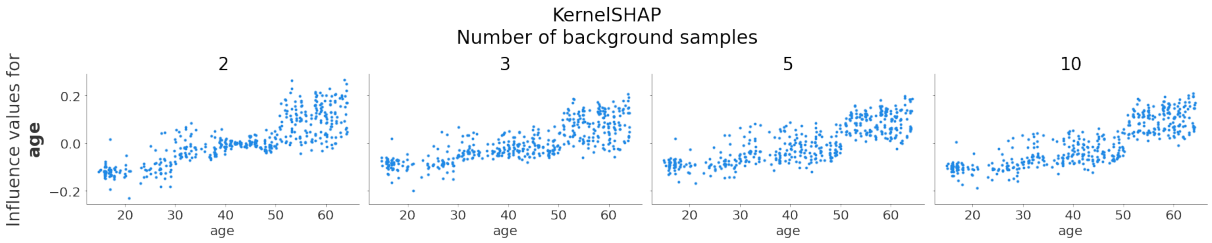


Figure 3.31: Partial dependence plot of the explanations given by the *KernelSHAP* method for the attribute **age** on the SA-Heart dataset with different values for the number of background samples.

We can see that even with two background samples, the explanations are already close to the one with ten background samples and that there is almost no difference between 3, 5 and 10 background samples. Although this depends on the number of samples and the distribution of the dataset, we can still hypothesise that we can significantly reduce the number of background samples with the KMeans algorithm to reduce the time required for the method to compute explanations.

Finally, we examine the number of samples obtained from the distribution (based on the background samples mentioned above) by creating perturbations. This parameter is called nsamples and its default value is $2d + 2048$, with d the number of attributes in the dataset (2066 for the SA-Heart dataset)[9]. We show in Figure 3.32 the age attribute dependence plot for different values of nsamples.

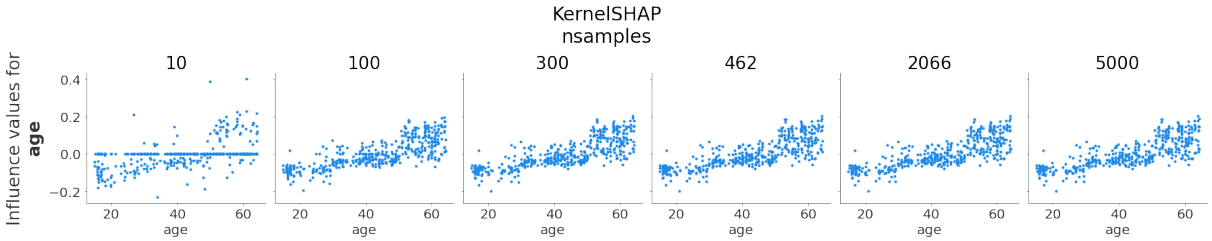


Figure 3.32: Partial dependence plot of the explanations given by the *KernelSHAP* method for the attribute **age** on the SA-Heart dataset with different values for the number of drawn samples.

As we increase the number of samples, we can see that we quickly reach a plateau at around 300 nsamples. Under this value, we can still see the shape of the explanation, but we can also see many samples that are given an influence of 0, which is incorrect. Nevertheless, we can firmly say that the default value (2066) is too many samples for this dataset and model. We can reduce this number a lot to compute the explanations.

Overall, we find that the impact of the parameters depends on the method, and each parameter has a different effect. *Spearman Coalitional* has a main parameter that can control the trade-off between the degree of approximation and the computation time of the method. *KernelSHAP* and *LIME* have several parameters, allowing the user to control the robustness, computation time and locality of the explanations to some extent, but they require good knowledge of the explanation method.

3.5 Recommendations for the use of local attributive explanation methods

In this section, we propose some recommendations to use each local attributive explanation method, based on the results from all our experiments. For simplification's sake, we refer to the explanations produced by the local explanation methods when summarised for all the data as 'global explanations'.

Table 3.4 summarises the advantages and drawbacks of each method studied. Overall, in all our experiments, we highlight that coalitional-based methods should better produce precise local explanations while *SHAP* should be better at creating coherent and easily interpretable global explanations. It is also confirmed by the fact that *SHAP* tends to assign more importance to fewer attributes than other methods, producing global explanations that are more concise but potentially hiding other attribute contributions and inter-dependencies. *Spearman Coalitional* explanations are overall slightly less robust than the other methods and are negligibly less readable. *LIME* has several drawbacks, one of the most distinguishable being its tendency to miss the interactions between attributes and complex influences. Regarding method parameters, each method offers a different

[9] *KernelSHAP* documentation

Method name		Advantages		Drawbacks	
Coalitional based	Complete	Consider feature interdependence	Exact shapley values	Slow in high dimension	
	Spearman		Parameter α to control the level of approximation	Global explanations can be hard to read	Less robust on tree-based models
LIME		Fast in high dimension Various parameters to control robustness and locality trade-offs		Slow in low dimension Low quality explanations Tends to miss non linear and non monotonic influences Not robust with simple models Can miss relationship between pairs of features	
SHAP	KernelSHAP	Easy to interpret global explanations	Various parameters	Approximations may be inprecise	Slow in high dimension
	TreeSHAP		Very fast in low and high dimensions		Tree-based models specific
	TreeSHAPapprox				

Table 3.4: Summary table of advantages and drawbacks of each method

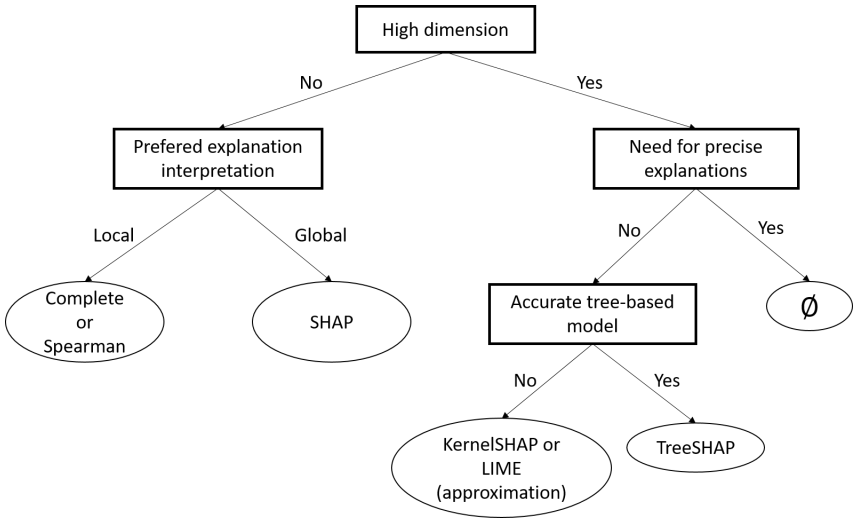


Figure 3.33: Road map for the most appropriate use of methods

number and types of parameters. *Spearman Coalitional* α allows users to easily control the trade-off between computation time and degree of approximation. *LIME* numerous and complex parameters allow a fine-tuning of the method but require extensive knowledge of *LIME* behaviour regarding these parameters and the model and dataset considered. The *KernelSHAP* parameters are similar to those of *LIME* but appear to induce less significant changes in the explanations obtained, which makes it possible to use them to reduce the computation time without degrading the quality of the explanations.

We use all the results presented to show a simplified road map as a decision tree in Figure 3.33 to help readers find the most suitable explanation method according to their datasets and objectives.

In this figure, the high dimensions represent the number of attributes in the studied dataset. Indeed, there is no "hard" cut-off to define when it goes from low to high dimensions. In our experiments, we can consider this cut-off somewhere between 11 and 15 attributes, depending on the dataset complexity, the computational time and the material available. "Accurate tree-based model" represents the ability to train a satisfactory (defined by the users' objectives) tree-based model on the dataset. The model can then be explained thanks to the optimisation in *TreeSHAP*. If the desired model is not tree-based, we advise the user to look at *KernelSHAP* and *LIME* parameters to reduce the number of background samples and perturbation samples until the explanations are computed in

a reasonable time. However, we warn about the potential loss of precision and robustness induced by such method approximations.

Finally, we show that *SHAP* and *LIME* can make significant approximations in some cases and that *Coalitional* and *Complete* methods cannot be executed in a reasonable time in high dimensions. This leaves a space for high-dimensional precise explanations that are not yet addressed to our knowledge.

3.6 Conclusion

This Chapter presents an improved version of the already-existing *Coalitional* method. We work on the hyper-parameters of the method to propose a simpler-to-use version. We use a dichotomic search to find the optimal value of the existing α-threshold. We base our *Coalitional* method on the wanted complexity in the attributes coalitions to optimise the explanations.

Then, we provide a large set of experiments to benchmark local attributive explanation methods and recommendations on when to use each method. Our findings indicate that there is not a single method that is the most appropriate for every usage. Therefore, this thorough analysis allowed us to identify the strengths and limitations of each method. Our improved version of the *Coalitional* methods well-perform against well-known methods from the literature by allowing an acceptable computation time while maintaining a high precision of explanations. Regarding other explanation methods, the *Complete* is the most accurate but suffers from a very long computational time. On the contrary, *LIME* and *SHAP* methods offer a more intelligible global view of feature effects. We have also seen that the choice of the predictive ML model does not impact the general behaviour of the explanation methods much. However, except with *LIME*, simpler predictive models tend to produce more readable and robust explanations, but tree-based models allow for *TreeSHAP* use which is more efficient.

The greatest problem arises when a high dimension (*i.e.,* a high number of attributes) is involved, as is often the case in statistics and ML. In this case, the exponential complexity of *Coalitional-based* methods makes them too long to compute. Indeed, the worst-case scenario is the need for high-precision local explanations in high dimensions since there is a clear lack of methods addressing this problem in the current literature. Another problem is that computing the *Complete* influences as the baseline becomes near impossible with larger attribute numbers. Thus, it isn't easy to monitor the performance of our different methods with this baseline. A possible way to address this problem could be first to run a global attribute importance study for large datasets using methods such as *Permutation Importance* that is model agnostic, or *Gini Importance* for tree-based models. Then use this information to compute influences only for the most important attributes during the individual explanation generation.

Finally, we provide two medical examples to illustrate how to use, interpret and explore explanations. These examples show how explanations can behave in real-world applications and how different XML methods can produce explanations easier or harder to understand. In the next chapter, we will explore the explanations exploitation as new data to discover new information about the data and its modelling.

Chapter 4

Explanations as a new data space: exploring explanations through clustering

Contents

4.1 Introduction

Local explanations are increasingly used in AI-assisted tools to offer more information than a single prediction (Antoniadi et al., 2021). Their popularity is due to the instance-level accuracy of these explanations, which links the impact of each attribute to the prediction made for each instance and allows differences to be detected between all instances. Yet, providing only local influences seems insufficient to improve decision-making efficiency. Indeed Weerts et al. (2019); Zhang et al. (2020) show that displaying influences along with an individual prediction did not significantly enhance the utility and understanding for the user as opposed to prediction alone. Moreover, knowing all the local explanations of a dataset does not guarantee a complete data understanding since there are as many explanations as instances in the original raw dataset, with the difficulty of finding explainability patterns in this new dataset.

In this context, we hypothesise that XML influences can be seen as a new data space that can be explored and used as a basis for further analysis. Indeed, these influences represent ideally the importance of each attribute for the task at hand in an ML model and may convey less noise or spurious indicators than the original space as only the most significant information is preserved this way. Influence analysis is thus a good candidate to identify the main trends in the dataset, i.e. the characteristic relationships between the attributes.

In this Chapter, we propose a new framework for data exploration that, instead of analysing raw data space, focuses on the analysis of relations between XML influences in Section 4.3. As such, this work can be perceived as a contribution to the novel domain of Actionable XAI (Holzinger et al., 2020), which considers actionable concepts, measures, and metrics for explainable learning and reasoning to improve data analysis or ML models based on explanations. As a first contribution in this direction, we introduce in this chapter a thorough analysis of the benefit of XML influence space for data exploration based on clustering algorithms. The main benefit of our approach is that, by reducing perturbations in the description of instances via explanations, we expect to achieve better cluster quality than in the original raw data space with more homogeneous subgroups of influences. In turn, these clusters will help identify and understand the relationship between the data and its use by the ML model. Clustering approaches are also the most straightforward approach for understanding the behaviour of the modelling and the underlying dataset. To the best of our knowledge, this is the first work that studies in a general framework the benefits of using local influences as a new input for clustering to identify more informative and homogeneous groups.

In Section 4.4, we extensively evaluate our approach on 104 datasets paired with multiple local attributive XML methods and clustering techniques for a large variety of cluster numbers, to compare the use of raw data and influences. We detail our research questions and hypothesis, the metrics used to evaluate the clusters' quality and Specifically, we propose an in-depth study for the K-medoid clusters quality to show the efficiency of considering influences space even for misclassified instances and ML models with low-performances. We finally evaluate the clustering of explanations when the optimal number of clusters is used for each clustering technique, on multiple local attributive XML methods to evaluate our proposal in a best scenario use case. We also discuss the advantages of our approach in a broader context, linking results from clustering with knowledge from modelling and explanation methods.

The work mentioned in this Chapter has been published in the following articles: Es-

criva, Aligon, Excoffier, Monsarrat, and Soulé-Dupuy (2023a); Cugny, Doumard, Escriva, and Wang (2023).

4.2 Prerequisites: Analysis of clustering algorithms

According to Jain et al. (1999), clustering consists of the unsupervised classification of patterns (being data items, attributes vectors, time series, graphs) into groups called clusters. This problem is complex since there are no unique criteria to assess the quality of a grouping. For example, internal criteria such as Davies-Bouldin index (Davies and Bouldin, 1979) ensure that groups are compact and well-separated but impose to shape the clusters as hyper-spheres, similar to the well-known Silhouette index (Rousseeuw, 1987). External criteria such as (Adjusted) Rand Index (Hubert and Arabie, 1985) assess the quality of the grouping with a ground-truth knowledge that is to be known beforehand. Even if an evaluation criterion is known, clustering is an NP-hard problem since one would have to build all partitions for all possible numbers of clusters to determine the best clustering (Jain and Dubes, 1988). As such, there exist a large variety of clustering algorithms (Jain, 2010) depending if they produce a disjoint partition of the dataset such as k-means (MacQueen, 1965) or k-medoid (Kaufman and Rousseeuw, 1990), a fuzzy or soft partition (Bezdek, 1981) or a dendrogram that is a nested set of partitions such as in the hierarchical clustering (Kaufman and Rousseeuw, 1990). Jain (2010) identifies new trends for clustering algorithms such as the introduction of semi-supervision to take into account expert knowledge when available (Bilenko et al., 2004; Vu et al., 2012). Other challenges involve dealing with large-scale datasets or streams (Labroche, 2014) or proposing efficient co-clustering approaches that build a clustering of instances and attributes at the same time (Parsons et al., 2004). In our work, we focus first on simple use cases of data exploration, thus avoiding the impact of streams or external constraints on our experiments. Finally, another recent tendency in clustering is related to the use of deep architecture to build end-to-end clustering systems that go from data representation to clustering in a single algorithm. The most well-known approaches in this context are DEC (Deep Embedded Clustering) and its variants (Xie et al., 2016). We are not going to focus on these types of approaches either as they build their own embedding that would defect, to some extent, the interest of our study to compare the raw data space with the attribute influence space. However, one important aspect of clustering is the metrics that define the topology of the space and that are generally attached to the geometry of the clusters. To preserve a variety of cluster shapes, we will consider in our work, clustering approaches relying on minimisation of variance in Euclidean space (k-means, k-medoids, hierarchical clustering with Ward criterion (MacQueen, 1965; Kaufman and Rousseeuw, 1987; Ward Jr and Hook, 1963)), Gaussian Mixture Models that leverage the constraint of uniform variance of k-means (Dempster et al., 1977), a Mahalanobis distance that leverage variance and correlation between attributes to discover anisotropic clusters (Mahalanobis, 1936) and finally, a density-based algorithm (HDB-SCAN, based on DB-SCAN algorithm) that can find any type of cluster shape (Ester et al., 1996; McInnes and Healy, 2017).

4.3 Influence-based clustering framework

With the idea of exploring explanations as completely new data, clustering techniques are the most straightforward to gain insight and challenge our hypothesis. Clustering

is a common tool of exploratory data analysis and statistical data analysis to discover interesting patterns in data. In this section, we detail our influence-based clustering framework.

Figure 4.1 shows the step-by-step process to cluster instances based on their influences:

1. A machine learning model is trained with raw data and predicts classes of all the instances from the raw dataset.

2. A local attributive XML method explains the trained model. Users can choose the data used as input for the method. Influences are computed to explain why the ML model made such predictions.

3. A clustering algorithm is used on influences to create homogeneous groups of instances to detect their important attributes based on the modelling. Users can define the number of clusters they want to compute.

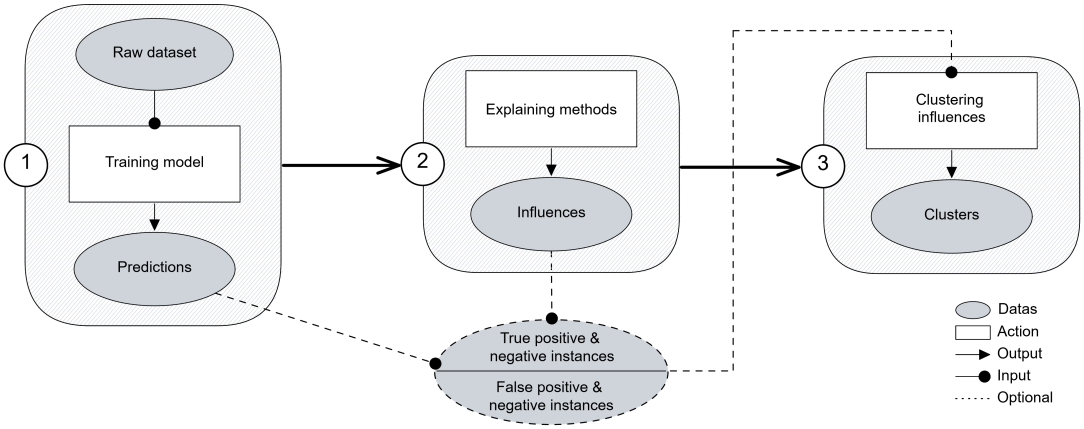


Figure 4.1: Our proposed Framework for explanation exploration.

In this framework, various elements can be modified according to user preferences. Any classification model can be used in Stage 1, as they are all designed to compute predictions. Stage 3 allows any clustering method that produces a disjoint partition of the dataset.

In Stage 2, the framework is designed to accept local attributive XML methods. These influences are represented as tabular data, where each instance has a value associated with each attribute. We directly use these influences data as input for the clustering step. Influences are valuable because they provide additional information that the raw data does not: the link between the modelling predictions and the dataset attributes. Compared to raw data, explanations produced by local attributive XML methods have the same unit across all attributes, thus avoiding any problem of value ranges. Another advantage is that influence values are less noisy since the ML model mainly focuses on attributes relevant to the underlying predictive task and excludes information not explained by the complex attributes interaction, hence the relevance of carrying out clustering. For supervised tasks, local attributive XML methods usually generate a dataset for each class with identical dimensions as the raw data. For example, if the raw data consists of n instances and m attributes and the supervised task is a multi-class problem with c classes, the generated dataset (also called the influence dataset) is shaped as a tensor with $n \times m \times c$ dimensions. To have an influence dataset with the same dimension as raw

data ($n \times m$) one can only select a single class and its associated influences. For example, regarding binary classification, the positive class is often chosen as the class of interest for influences.

An additional and optional step is to select a particular subset of the data for clustering. Indeed, it is possible to study the instances correctly and incorrectly classified by the model separately via instance clustering. Considering the model predictions against the data labels, the influences are separated into two distinct groups before being clustered. Two different sets of clusters are then proposed to the users. This option can have several advantages. Since the influences represent the model decisions, separating the instances can provide new knowledge. Studying the well-classified instances can help to identify their characteristic patterns by removing noise and outliers from the misclassified instances. This can give a more accurate idea of general patterns, for example, to check that there is no bias in the dataset. Regarding misclassified instances, they may cover different realities. They can be outliers in the data and not correspond to the general behaviours without bias or error. However, misclassified instances may also constitute a particular sub-group of the data that is worth studying. For example, this would be the case of children with some types of cancers usually associated with older people. Due to age, the model may misunderstand this subgroup, as there are few children with non-pediatric cancers, or the input variables may be insufficient to identify this subgroup. However, it is necessary to study this subgroup to understand whether there is any specific behaviour in this subgroup and ultimately understand the overall dataset. Separating the instances can therefore allow the exploration of new patterns that can be invisible if all the data were kept. This may be even more important for influences because of their direct link to the model. Indeed, when the model prediction is incorrect, the influences reflect this error and are directly impacted by the wrong prediction of the model.

The full implementation of our proposal is available here: https://github.com /kaduceo/XAI-based-instance-selection. The source code will evolve with future works. Additional materials are also available.

4.4 Evaluation of our framework

In this section, we describe the experiments carried out to show the value of our framework, and more broadly of explanation clustering. We have focused on four different research questions (RQ) to show the usefulness of clustering explanations.

- RQ1. Does clustering explanations produce better-quality clusters than raw data clustering?
- RQ2. Does explanation clustering give effective results (even ?) for low-performance models?
- RQ3. How do misclassified instances behave when clustered based on their explanations?
- RQ4. Are there differences in cluster quality between XML methods for clustering explanations?

Based on these questions, we expect that clustering explanations will be beneficial for all clustering techniques and local XML methods included in the experiment. We hypothesise that this approach is also relevant for low-performance models, and that clustering

misclassified instances can provide important clusters to study alongside well-classified instance clusters. Finally, we speculate that Shapley-based XML methods -*SHAP*, *Tree-SHAP* and *Spearman Coalitional*- will be better for clustering explanations than *LIME*.

Thus, to support our hypothesis, we will first compare several clustering techniques for multiple local attributive explanation methods on various numbers of clusters for answering RQ1. Secondly, based on the best clustering method, we will compare the clustering of explanations - from multiple local explanation methods - separately for models with high and low performances to relate to RQ2, and then for instances well- or mis-classified by the models for RQ3. Finally, to bring awareness about RQ4, we will compare the locale XML methods clustering based on the optimal number of clusters for multiple clustering techniques and the corresponding clusters' qualities.

Part of this work was made in collaboration with the University of Tours, with the participation of Tom Lefrere and Manon Martin as Master 1 internships and Nicolas Labroche as one of their co-supervisors.

4.4.1 Experimental protocol

Clustering algorithms For our experiments, we select six clustering techniques to compare them and to achieve diversity in terms of clustering techniques families: three partitioning clustering, one hierarchical clustering, one density-based clustering and one modelling model-based clustering. As both raw data and influences data are tabular data of the same dimensions, clustering can be easily applied to both datasets without adapting the clustering method to a specific input.

First, we use the *K-medoids* algorithm (Kaufman and Rousseeuw, 1987). K-medoids assign data to k clusters iteratively based on their distance to a centroid point. This central point is always an instance from the dataset. Each iteration tries to maximise the distance between points from different clusters and minimise the distance inter-cluster. The number of clusters k is pre-defined. In this experiment, we chose two distinct distance metrics: the Euclidean distance and the Mahalanobis distance. The second one considers correlations between variables and is suitable for data following a multivariate distribution. We also use the *k-means* algorithm (MacQueen, 1965). We can expect some differences as prototypes representative of clusters may not necessarily be part of the original instances with k-means. However, due to its continuous representation of prototypes, k-means can reach better compactness and separability between clusters when compared to k-medoids at the expense of the interpretability of cluster prototypes.

For these three clustering methods, to ensure the stability of the clustering, we use *k-means++* for the initialisation -and the equivalent *kmedoid++* for the Kmedoid clustering. Based on these initialisation methods, our results have shown to be very consistent from one run to the next. For this reason, the paper only shows one result for each of these approaches even though there might exist a small variability due to the non-deterministic nature of the choice of the initial clusters.

Agglomerative Nesting (Agnes) is a hierarchical clustering method mentioned in Kaufman and Rousseeuw (1990). Hierarchical clustering creates a hierarchy of clusters and therefore a pre-specified number of clusters is not required if one wants the complete hierarchy. A number can be specified to extract the clusters from one level of the hierarchy. Agglomerative clustering works in a bottom-up manner: at first, each instance is considered as a single-element cluster and at each iteration, the two most similar clusters are combined. The similarity between elements is based on their distance, the Euclidean

distance being the default distance. Agglomerative clustering is good at identifying small clusters thanks to its bottom-up approach and clusters are persistent over runs. We use the Ward linkage to choose the pair of clusters to merge.

HDBSCAN (McInnes and Healy, 2017; McInnes et al., 2017) is a popular hierarchical density-based clustering technique. Based on the density in some space, HDBSCAN groups together the points where the density is high (i.e. the points closely packed, that have many neighbours). Density is defined based on the distance between points and HDBSCAN performs multiple iterations of clustering for all possible density scales. This allows the detection of meaningful clusters in data of varying densities and the robustness to parameter selection, as opposed to DBSCAN. HDBSCAN is stable over runs and resistant to noise and outliers. This clustering technique does not require a pre-specified number of clusters.

Expectation-Maximum clustering algorithm (EM) was proposed by Dempster et al. (1977) to cluster points based on statistical modelling and data distribution. EM clustering assigns data points to clusters iteratively to maximise the overall probability or likelihood of the data. Unlike other clustering methods, EM is a soft clustering technique: each point has a probability of belonging to each cluster, rather than a single assigned cluster. In our case, we assign instances to the cluster with the highest probability.

Datasets and classification task We use 104 datasets from an Open ML collection[1] (Vanschoren et al., 2014) that meet the following criteria: binary classification, more than 100 instances, more than four attributes and at most nine attributes due to the computational cost of producing influences. Table 4.1 details statistics about the datasets used.

Table 4.1: Statistics of the experimental datasets based on the number of attributes.

Number of attributes	**4**	**5**	**6**	**7**	**8**	**9**	**All**
Number of datasets	14	25	17	16	15	17	104
Mean number of instances	465	1197	654	554	650	503	670
Min number of instances	125	100	100	108	130	100	100
Max number of instances	1372	7129	3107	4052	4177	1473	7129

Binary classification is chosen to facilitate the interpretation of influences. We consider that all influences are based on class 1. In this case, influences represent the impact of each attribute on the probability of the instance being in class 1. We train a Random Forest model (RF) with a Grid Search Cross-Validation to optimise hyperparameters. This model was chosen to test tree-specific explanation methods while keeping a limited number of hyperparameters to avoid overfitting (compared to boosted trees). Only to evaluate the performances of the modelling, each dataset is divided into train and test sets according to the 75%/25% ratio. Table 4.2 shows the performances of all the models trained in our experiments. Models are trained adequately to capture most information of the dataset. The mean and median balanced accuracy are respectively 0.79 and 0.85, meaning most models can accurately classify test instances. Some models also have very low accuracy, the minimum being 0.42. When we separate models based on an accuracy threshold set to 0.8, high-accuracy models have a median balanced accuracy of 0.92,

[1] Available in `https://www.openml.org/s/107/tasks`

whereas low-accuracy models have a median of 0.6. This distinction between the models will be used for the experiments related to RQ2, to show whether the approach works whatever the performance of the models.

Table 4.2: **Statistics of models trained**. Balanced accuracy and percentages of true and false instances are presented for the 104 datasets and separately based on the 0.8 accuracy threshold. For true and false instances, the median number of instances is presented along with the percentage.

	Balanced Accuracy			**% of True instances**			**% of False instances**		
Models (#)	Median	Min	Max	Median	Min	Max	Median	Min	Max
All (104)	0.85	0.42	1.0	94% (307)	61%	100%	6% (21)	0%	39%
Acc $\geq$ 0.8 (60)	0.92	0.81	1.0	97% (404)	85%	100%	3% (11)	0%	15%
Acc $<$ 0.8 (44)	0.60	0.42	0.79	82% (252)	61%	98%	18% (62)	2%	39%

We also study the number of instances well classified and misclassified by the ML modelling in Table 4.2. In all experiments, we call *true instances* well-classified instances, referring to True positive and True negative terms. *False instances* is then related to False positive and False negative instances, so misclassified instances. We use three different separations of data: all instances together, only true instances and only false instances. For the experiments about RQ3, as we separate true and false instances, we choose not to evaluate high-accuracy models on false instances as there are not enough instances in most datasets to create clusters and properly evaluate them and compare the results. Then, when studying false instances, we only work with models with low accuracy as the number of false instances is higher and sufficient. Also, the number of true instances is adequate to perform clustering for all models.

Explainability methods For exhaustive purposes, we choose four different local attributive XML methods to compute influences: *KernelSHAP* -called *SHAP*- (Lundberg and Lee, 2017), *TreeSHAP* (Lundberg et al., 2020), *LIME*(Ribeiro et al., 2016) and *Spearman Coalitional* method (Ferrettini et al., 2021). As explained in Chapter 3.5, each XML method provides influences with different strengths and disadvantages. Thus, we want to study the relevance of using local influence clustering compared to raw clustering in a global way.

Setting the number of clusters To define the number of clusters, we used two different setups. First, to strictly compare the clustering methods to answer RQ1, we use multiple percentages of the total number of instances in the dataset as the number of clusters. We will also use these percentages for RQ2 and RQ3 when evaluating the impact of the model performances and the classification. We use the following percentages: 1%, 2%, 3%, 4%, 5%, 10%, 20%, 30%, 40% and 50%. The number of clusters is then $n_{cluster} = p * n_{instances}$ with p the selection percentage between 0 and 1 and a minimum number of two clusters. As the size of the datasets varies greatly as shown in Table 4.1, we prefer to select a percentage rather than fixed numbers of instances to take into account the diversity of the datasets. As we first aim to show how clustering on influences exhaustively performs against the raw data, multiple percentages per dataset can show how cluster quality evolves without looking for the optimal number of clusters (which may be different for each method). For RQ1 -so RQ2 and RQ3-, we only use the parametric

clustering approach -Partitioning clustering methods (K-medoids and K-means), Agnes and EM- which allows us to change the number of clusters and exclude HDBSCAN. Then, to study the differences in raw and each XML data spaces clustering for RQ4, we will compute the optimal number of clusters and use the Silhouette score (Rousseeuw, 1987) as the metric that automatically selects the optimal number of clusters when needed.

Let i be an instance of the dataset D assigned to the cluster C_I, C the set of c clusters, n the number of instances in the dataset D, and $d()$ a distance function.

$$
\begin{aligned}
a(i) &= \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j) \\
b(i) &= \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j) \\
s(i) &= \frac{b(i) - a(i)}{max(b(i), a(i))} \\
Silhouette &= \frac{1}{n} \sum_{i \in D} s(i)
\end{aligned} \tag{4.1}
$$

For a range of cluster numbers, each score is computed and the highest value is chosen as the optimal cluster number. In case of a tie (which is unlikely given the workings of the metrics), we take the first value. We defined a range between 2 and 30 clusters to compute the optimal number. The Silhouette score was used for the parametric clustering approaches only, as HDBSCAN automatically compute its optimal number of clusters.

Comparison to ground-truth labels Finally, we evaluate if clusters are well-defined and manage to group similar instances and separate dissimilar instances based on their *a-priori* labels. We select two external clustering metrics, *Entropy* and *Purity*. With external metrics, class labels are needed as metrics assess the distribution of labels within clusters to evaluate how clusters and labels are related and how clusters manage to group similar instances. Entropy measures the distribution of labels in a cluster, i.e. the ability of the algorithm to differentiate between data that do not have the same "real" class. A perfect entropy means all instances from the same class are in the same clusters. In addition, Purity measures the relative size of the majority class in a cluster to evaluate its dominance over other classes. Perfect purity describes that each cluster has only one class. These two metrics give values between 0 and 1. A perfect clustering will usually have an entropy equal to 0 and a purity equal to 1. These metrics are defined as follows (Conrad et al., 2005):

$$
E(C_k) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_k^i}{n_k} \log \frac{n_k^i}{n_k} \qquad\qquad Entropy = \sum_{k=1}^{K} \frac{n_k}{n} E(C_k)
$$

$$
P(C_k) = \frac{1}{n_k} \max_i (n_k^i) \qquad\qquad Purity = \sum_{k=1}^{K} \frac{n_k}{n} P(C_k)
$$

where C_k is a particular cluster of size n_k, q is the number of class in the dataset, K the number of clusters and n_k^i is the number of instances of the ith class assigned to the kth cluster.

4.4.2 Results

In this section, we describe the results of the experiments by first comparing multiple clustering methods either based on the clusters of influences from XML methods or obtained from raw instances, in relation to RQ1. We then study the impact of model performances and data classification on the cluster quality to answer RQ2 and RQ3, for *SHAP* and *Spearman Coalitional* for the K-medoid clustering. The fourth part shows how XML methods perform against each other for K-medoid, Agnes and HDBSCAN clustering when searching for the optimal number of clusters to gain insight into RQ4.

4.4.2.1 On comparing the explanations clustering from multiple family of techniques

Figures 4.2 display the purity and entropy values for all percentages of selection, on raw and influences data, for K-medoids with Euclidean and Mahalanobis distance and K-means. The three clustering methods have almost identical behaviours. Clusters have better purity and entropy for *Spearman Coalitional*, *SHAP* and *TreeSHAP* methods. *LIME* and Raw have worse results, especially in entropy where differences with other explanation methods are greater.

Figure 4.3 shows the mean purity and entropy for the Agnes and EM clustering method. Results are identical to the ones with partitioning clustering techniques: explanations methods produce clusters with better quality than Raw data. *LIME* is below other explanation methods in terms of clusters' quality, nearest to Raw results than other XML results.

Figure 4.4 shows the clusters' quality for all clustering techniques for *SHAP* and *LIME* Explanations. We exclude *TreeSHAP* and *Spearman Coalitional* as their results are identical to the *SHAP* ones. For all explanations, as described before, all clustering methods based on distance have similar results. In entropy, K-medoids with the Mahalanobis distance have slightly worse results than the other clustering approaches using the Euclidean distance. These differences are therefore not statistically significant. Agnes and EM also have almost identical results to K-medoids and K-means.

Our results show that multiple families of clustering techniques - partitioning, hierarchical and modelling model-based - perform well in clustering explanations. Mean Purity and Entropy are better than with Raw data, indicating that clusters are more meaningful relative to the labels of instances, answering RQ1 and confirming our hypothesis. Only *LIME* explanations produce clusters of lower quality. However, since all the other explanation and clustering methods produce better clusters, this difference is probably due more to *LIME* than to any problem with the proposed approach. Noticeably, *LIME* exhibits a slightly higher variability in its explanation (Visani et al., 2022) that may not fully ensure that neighbours in the original space share close attribute influence representations. Moreover, *LIME* tends to generate too general explanations (Laugel et al., 2018; Alvarez-Melis and Jaakkola, 2018) that might be less accurate (depending on the shape of the original decision boundary) than those proposed by other methods. All these aspects can lead to the observation that *LIME* attribute influence space is under-performing for clustering, related to our hypothesis of RQ4.

The performance of all these clustering techniques is valuable as they can have other advantages outside the boundaries of our approach and experiments. Hierarchical clustering shows the instances' relationships in the clustering process, which can be used to work around clusters. Density-based clustering can be used to detect prototypes and outliers

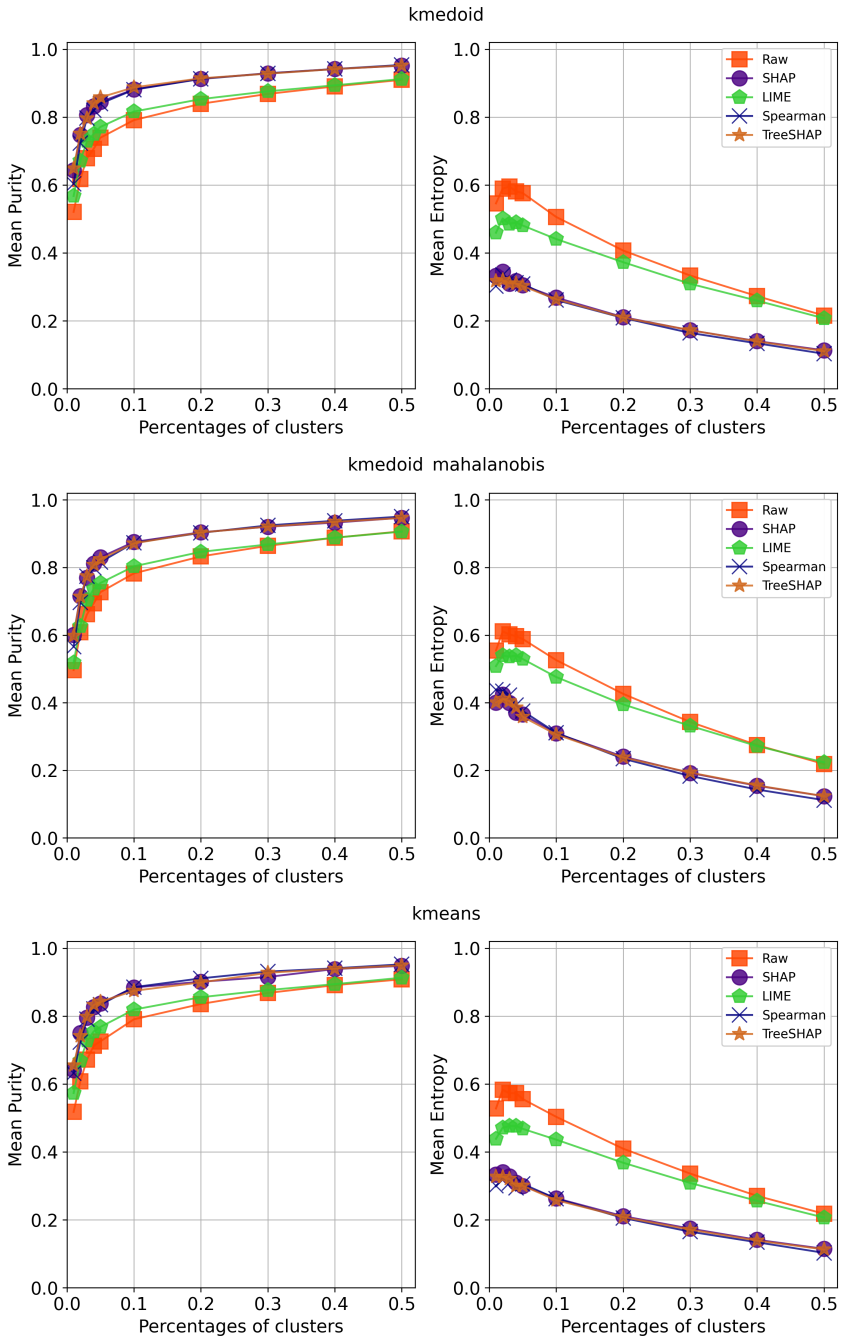


Figure 4.2: Comparison of clustering quality for partitioning clustering techniques: K-medoids with Euclidean and Mahalanobis distance and K-means.

differently from other methods. EM clustering can allow the study of the attributes' importance for each cluster and their interpretation based on the distribution and variance of attributes in each cluster.

4.4.2.2 Impact of models performances on clusters quality

Based on the previous results, we will focus in this experimental step on the K-medoids with Euclidean distance to evaluate the impact of model performances on the clusters' quality.

When comparing raw data clusters to the influence ones, for all instances, Figure 4.5

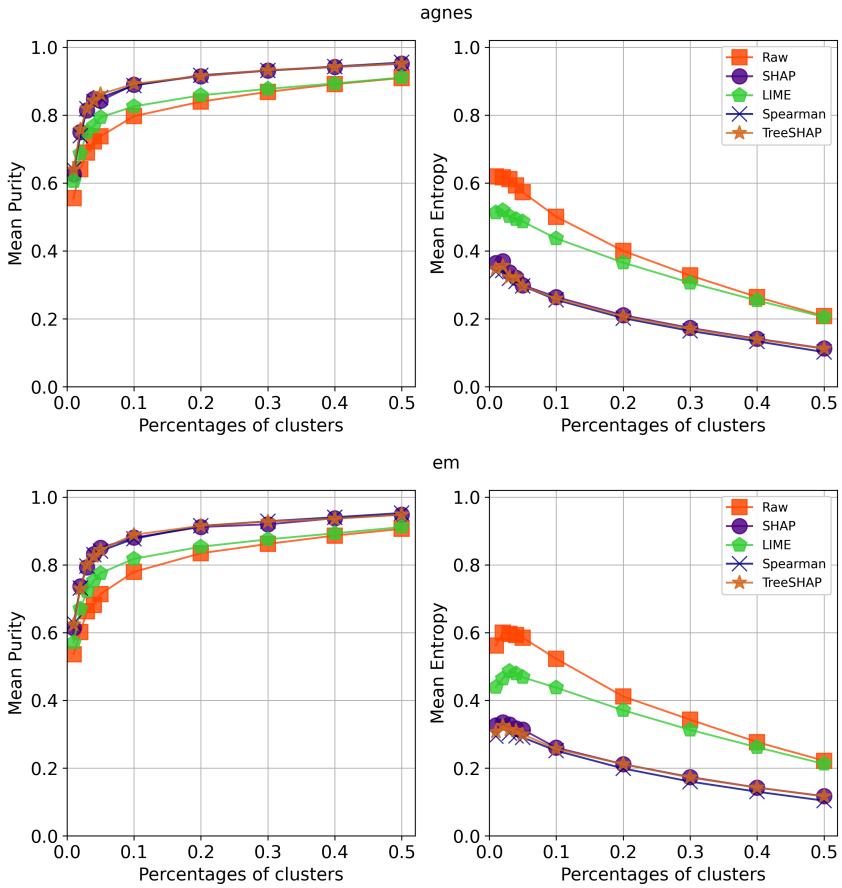


Figure 4.3: Comparison of clustering quality for Agnes and EM clustering.

shows raw data clusters have lower purity and greater entropy than other clusters, regardless of the model performance, the percentages or the XML methods, as we have already seen previously. When we compare results based on model performances, purity is higher for high-performance models for raw data and influences clustering and the differences between methods seem similar. For entropy, results are also better for high-performance models. However, differences between methods are even greater when the model has an accuracy greater than 80%. For low-performance models, *Spearman Coalitional* influences clustering has the lowest entropy, followed by *SHAP* and *TreeSHAP* -which have almost the same entropy-, and then by *LIME* and Raw. With high-performance models, *SHAP* and *TreeSHAP* have similar entropy to *Spearman Coalitional*, even slightly better ones, and the differences with *LIME* and Raw are more visible.

To relate to our RQ2, this indicates that clustering explanations from models with low performance provide clusters with one majority class -from the purity metric- but that instances from the same class are dispatched between clusters -then the high entropy. Also, we expect that clusters have a lower purity and entropy when models have a lower accuracy, whatever the data or cluster percentages. Indeed, when the model performance is poor while the model is adequately trained, this may indicate that the data is less generalisable or of lower quality. The consequences are a higher proportion of misclassified instances, impacting the explanations. This hypothesis seems to be reflected in the quality of the clusters created.

To gain insight into the pertinence of clustering explanations even for low-performance models, we take into account only the true instances (the instances well predicted by the

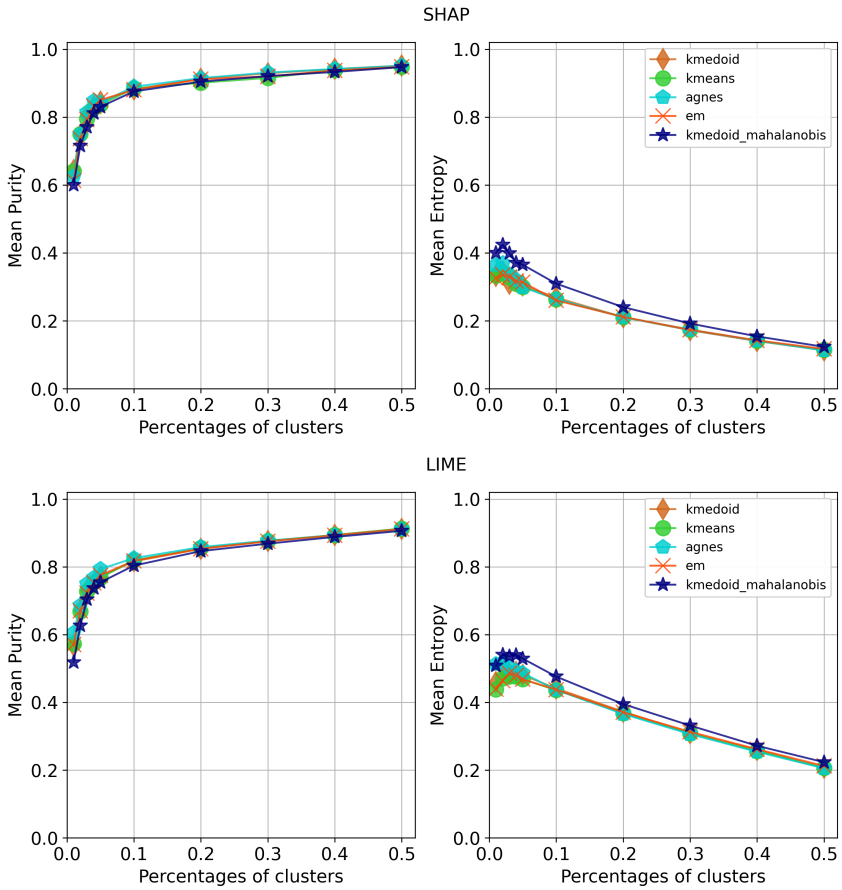


Figure 4.4: Comparison of clustering quality for each clustering technique, for *SHAP* and *LIME* explanations.

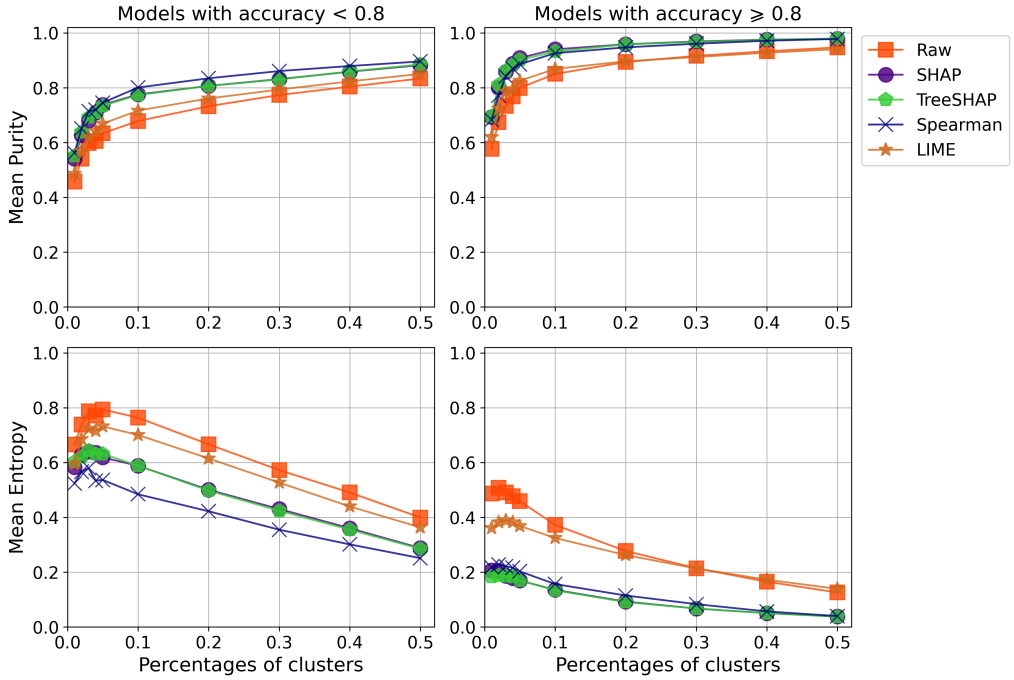


Figure 4.5: Comparison of K-medoid clustering for XML methods trained on all instances.

model) to compare the cluster quality of the two subsets of models. Figure 4.6 displays the true instances clusters' quality for models with low and high performances. In Purity, both subsets of models have high, almost perfect, purity, especially for high percentages of clusters. *LIME* and Raw have lower Purity than *SHAP* and *TreeSHAP* for both model performances, and all four have similar purity across all models. *Spearman Coalitional* is the only XML method with a different behaviour based on model performances. For low-performance models, *Spearman Coalitional* has a purity similar to *SHAP* methods, while is near Raw and *LIME* and lower for high-performance models than for low-performance models. In entropy, almost the same behaviour appears for all raw and XML methods. For all models, *SHAP*, *TreeSHAP* and *Spearman Coalitional* have lower entropy than *LIME* and Raw. *Spearman Coalitional* has also a slightly worse entropy for high-performance models, although it is still better than *LIME* and Raw.

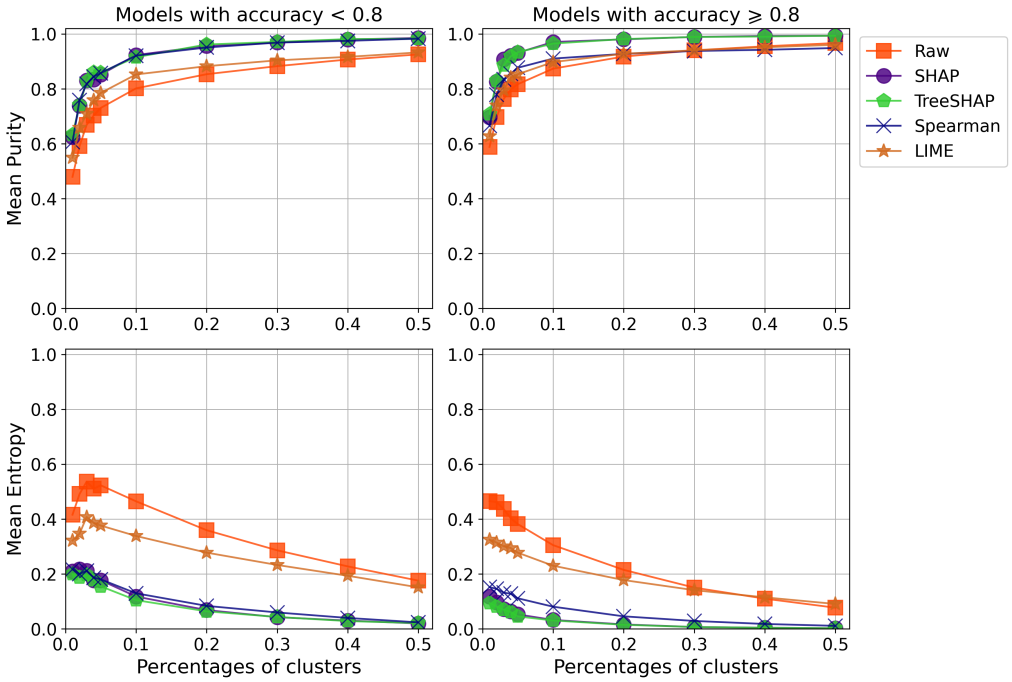


Figure 4.6: Comparison of K-medoid clustering for low- and high-performance models on "true" instances.

Based on these results and our RQ2, clustering explanations from low-performance models give worse results than for high-performance models. However, the quality of the clusters may be sufficient, particularly in terms of purity, to analyse and extract relevant information from them. This point is supported by the analysis of true instances, where cluster quality is similar and high whatever the performance of the models.

Moreover, we can compare XML methods based on cluster quality, to support the RQ4 hypothesis. Although all are attributive methods with similar global behaviour, the calculated influences appear to be sufficiently different to produce dissimilar cluster results, especially in entropy. Clusters based on *LIME* have purity and entropy close to the clusters based on raw data, making this XML method the one with the worst results. More than evaluating clustering on only true instances, comparing all subgroups must be useful to show more differences between XML methods.

4.4.2.3 Impact of using different data subgroups for clustering

In this subsection, we aim to show in which circumstances well-classified or misclassified instances can be used to produce clusters of good quality (or not), notably in the worst case (degraded accuracy on a set of misclassified instances). We want to see if well and unwell instances behave differently when they are clustered together or separately and the impact of using only specific subgroups of data, to support our RQ3 hypothesis. As in the previous step, we focus only on the Kmedoids clustering technique. We also show the results for only *Spearman Coalitional* and *SHAP* XML methods as *TreeSHAP* is almost identical to *SHAP*, and *LIME* have the worst results.

Figure 4.7 and 4.8 show the cluster quality for the three data modalities, with influences respectively from *SHAP* and *Spearman Coalitional*. For both XML methods, clusters have better quality with only the true instances than with all the instances of the dataset, especially for low-performance models.

Figure 4.7 shows little difference in cluster quality for *SHAP* between *all instances* and *true instances* subgroups for models with high accuracy. Both metrics give good -high for purity, low for entropy- and almost equal results for both modalities. Influences from true instances produce almost perfect clusters even with low cluster percentages and are little affected by the model accuracy. As models with high accuracy have fewer false instances, their influences may only produce noises for the clustering. Removing them gives slightly better global results, as clusters have better entropy. For models with low accuracy, there are more differences between the subgroups, presumably because the proportion of false instances is greater. The *all instances* and *true instances* subgroups have a 0.4 difference in entropy and a 0.1 difference in purity for almost all percentages. The *false instances* subgroups also have similar purity and better entropy as the *all instances* subgroups. Separating true and false instances to study them separately produces more homogeneous and coherent clusters for *SHAP* than keeping all instances together, especially on low-accuracy models. With these models, the number of false instances is higher, and they often represent behaviours not caught by the model.

For the *Spearman Coalitional* method, Figure 4.8 reveals a similar overall behaviour to *SHAP* regarding the cluster quality depending on the subgroups, especially on high-accuracy models and on the *true instances* subgroups. However, for low accuracy models and unlike *SHAP*, there are some differences when using only false instances. The *false instances* subgroups have slightly higher purity and lower entropy, especially on low percentages. The different use of input data by both methods can explain this behaviour. *SHAP* uses the input to produce perturbations for the model, creating new instances and studying a larger area of the data space than just the input data (here, the false instances). In contrast, *Spearman Coalitional* does not produce any perturbations and uses the input data as is to explain the model. The data space is then smaller and, therefore, less exhaustive. Using only false instances may lead to influences more precise for this subgroup, compared to using all instances or instances with perturbations, hence the difference between the two subgroups for *Spearman Coalitional* and the difference with *SHAP*. Moreover, for low-accuracy models, clusters from *true instances* and *false instances* subgroups are better than the clusters from all instances.

Globally, the cluster quality is degraded when considering all instances. Since purity checks the proportion of the majority class in each cluster and entropy how labels are dispatched across the clusters, grouping instances misclassified with well-classified ones logically lowers the cluster purity. With false instances, we analyse cases where the model fails to generalise or describe the data correctly. As influences represent the model

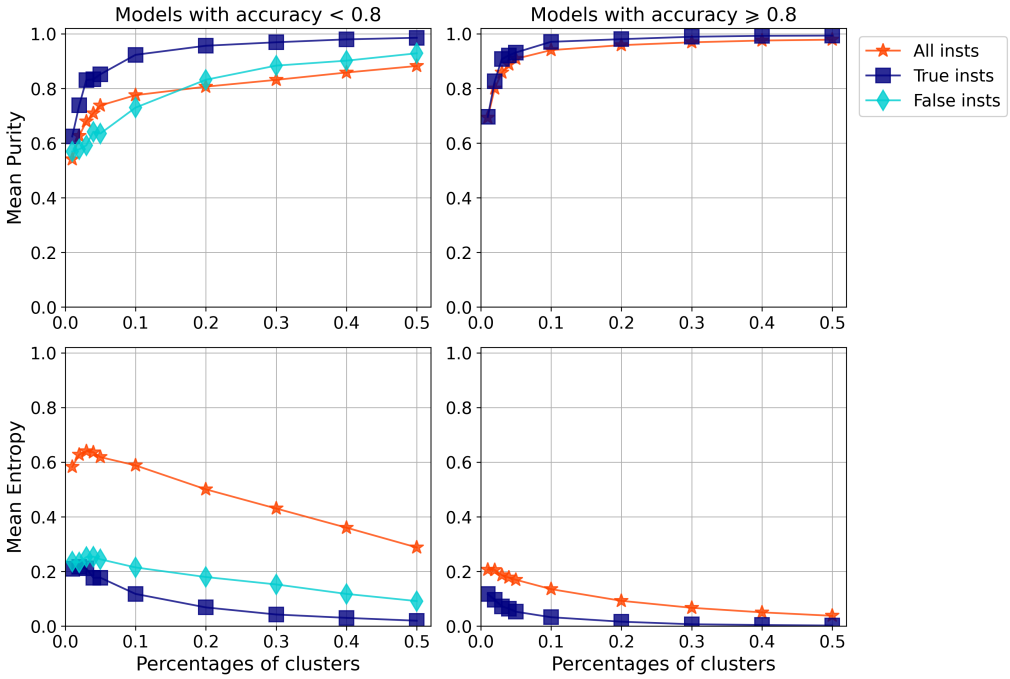


Figure 4.7: Comparison of K-Medoid clustering of *SHAP* influences.

decision, influences of misclassified instances may have lower quality than true instances influences and can bring noise to them. They may, however, be representative of why the model does not generalise and understand these data. Thus, these clusters can indicate where the problems lie in the data or the model. Concerning our RQ3, results are mixed as the clusters' quality with misclassified instances is not similar to the ones with well-classified instances. However, information may be extracted from these instances and their explanations to understand the model, and separating the instances according to their prediction gives valuable results.

These two figures also show that different XML methods can lead to clusters with distinct qualities or behaviours based on the data subgroups selected. *Spearman Coalitional* seems slightly better at clustering false instances than SHAP. And *SHAP* at clustering true instances, especially for high-performance models. These methods can then produce diverse and meaningful clusters to understand the modelling and dataset. Based on the subgroup of data studied, one method may also be preferable to another depending on the context. This seems consistent with the findings of Chapter 3.5, where depending on the dataset, the interdependence of attributes, the dimensionality or the model, one XML method can be more efficient than others. The same reasoning seems to apply here, supporting our RQ4 hypothesis, where according to the subgroup studied, one XML method can be better than the others.

4.4.2.4 On comparing the optimal number of clusters and the related clusters performances

When clusters are used in real-world applications, one important constraint is the number of clusters to compute. Too few clusters can result in inconsistent groups of instances with strong disparities, making it impossible to study them effectively. On the contrary, too many clusters can make it too complex to study all the clusters, resulting in a scattering of information and too much specificity in the information uncovered, without highlighting

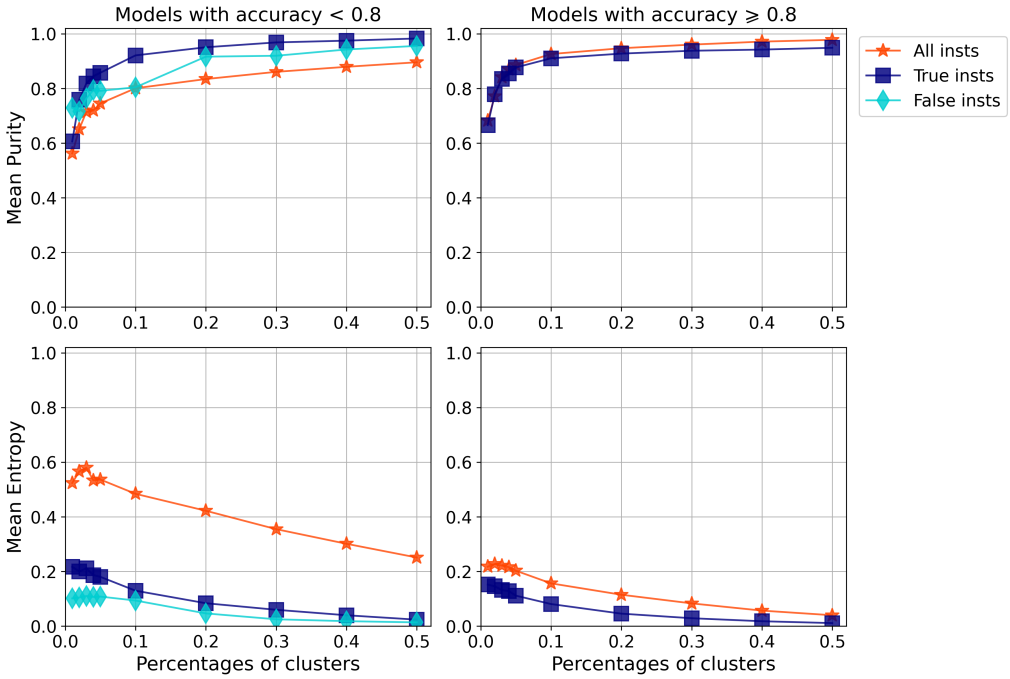


Figure 4.8: Comparison of K-medoid clustering of *Spearman Coalitional* influences.

the most important information. In both cases, time can be wasted due to the lack of clarity and usefulness of the clusters. Evaluating clusters in a setup with their optimal number of clusters is another important step to confirm the pertinence of our approach and evaluate the performance differences between each XML method.

For this experiment step, we keep one method previously evaluated per family to compare how the clusters on all raw data and influences behave in terms of quality: K-medoid for partitioning, Agnes for hierarchical and EM for modelling. With HDBSCAN for density-based techniques, this selection allows a diversity of clustering techniques to study the impact of the optimal number of clusters on cluster quality and how explanations clustering performs in this setup, to explore more deeply our RQ4. As previously said, the optimal number of clusters for K-medoids, Agnes and EM was computed with the Silhouette Score, defined in Definition 4.1. HDBSCAN is non-parametric and automatically defines the optimal number of clusters for each dataset.

Table 4.3 shows the mean optimal number of clusters for each clustering method and each type of data. The standard deviation is shown in parenthesis and Student statistical tests were performed between Raw data clustering and each XML data clustering.

Globally, no clustering method stands out from the others in terms of the number of clusters. Agnes and EM are the only two with greater numbers of clusters for Raw than for all explanation methods, with Agnes producing significantly fewer clusters with explanations from all methods than with Raw. EM produces similar numbers of clusters for Shapley-based XML methods - *Spearman Coalitional*, *SHAP* and *TreeSHAP* -, slightly less than with Raw data, without this difference being significant. K-medoid and HDBSCAN produce more clusters with Shapley-based XML methods than with Raw data, with significant differences for *SHAP* and *TreeSHAP* with K-medoid, and for the three Shapley-based XML methods for HDBSCAN.

LIME is the only XML method with significant differences from all clustering methods. The optimal number of clusters for *LIME* is significantly lower than with Raw data. This

	Kmedoid	**AgNes**	**EM**	**HDBScan**
Raw	6.83 (8.73)	11.09 (12.15)	8.45 (10.43)	6.30 (5.06)
SHAP	9.61 (11.34) *	5.68 (7.59) ***	6.44 (8.28)	9.99 (6.83) ***
LIME	3.32 (3.91) ***	3.55 (4.61) ***	3.44 (4.38) ***	4.41 (2.70) ***
Spearman	7.35 (10.17)	7.02 (9.91) **	6.24 (8.70)	7.89 (6.53) *
TreeSHAP	9.37 (11.29) *	6.24 (8.39) ***	6.36 (8.39)	9.04 (6.45) ***

Table 4.3: Mean and std of the optimal clusters number for each XML method and clustering techniques. For Kmedoid, AgNes and EM, the number is based on the Silhouette Score. P-values are represented as follows: (*) $p < 0.05$, (**) $p < 0.01$, (***) $p < 0.001$.

can be due to *LIME* discretisation behaviour studied in Chapter 3.4.2. As influence values are close for a range of raw values, influences may be grouped more easily and with fewer clusters. However, as mentioned above, a small number of clusters can lead to the aggregation of very disparate instances in terms of labels. Then, the significant results in the number of clusters are not an indicator of better explanations or better clustering with *LIME* and this study of the optimal number of clusters alone did not assess the quality of these clusters.

In this direction, Table 4.4 shows how each Raw and XML clustering performs against each other based on the quality of the clusters. For each type of data, for each clustering method, with clusters computed based on the optimal number, we retrieve the number of times each one has the best entropy and purity - the highest purity and lowest entropy.

	Purity				**Entropy**			
	Kmedoid	AgNes	EM	HDBScan	Kmedoid	AgNes	EM	HDBScan
Raw	14	22	17	7	14	10	15	29
SHAP	29	21	34	21	28	29	24	28
LIME	9	7	9	11	17	21	21	20
Spearman	**32**	**40**	**36**	25	**52**	**49**	**52**	**47**
TreeSHAP	31	32	32	**35**	31	28	30	34

Table 4.4: Number of times each XML method performs best for four clustering methods (highest purity and lowest entropy) over the hundred datasets. In the event of a tie, each XML method gets one point and the best score is shown in bold for each clustering method.

Regarding Purity, the best results vary depending on the clustering techniques and the XML methods. *Spearman Coalitional* has the best results for K-medoids, Agnes and EM. *TreeSHAP* is better than other XML methods for HDBSCAN to produce clusters with high purity. Except with HDBSCAN clustering, *LIME* has the worst scores and is below Raw clustering. *LIME* having low numbers of clusters seems to drastically impact the purity metric, as instances of each label must be split into different clusters and no single class must take the lead in each cluster, validating our RQ4 hypothesis. Conversely, for the best purity, there seems to be no link between a high number of clusters and high

entropy, since *Spearman Coalitional* and *TreeSHAP* are not always the methods with the most clusters on average for the clustering techniques in which they perform best.

In entropy, *Spearman Coalitional* has the best results for all clustering methods. These results also do not correlate with the lower or greater number of clusters for each clustering technique. From these two analyses when the optimal number of clusters is considered, *Spearman Coalitional* and *TreeSHAP* influences produce clusters with a higher quality than Raw data. Even *SHAP* has good results, with high numbers of better purity and entropy. This correlates with the results of our previous study with non-optimal numbers of clusters based on percentages, supporting our RQ4. Comparing the results from K-medoid, Agnes and EM with HDBSCAN shows that our approach is also valuable with density-based and non-parametric clustering techniques.

Finally, when choosing a clustering technique and an XML method, one can choose the best combination based on their clusters' quality preferences. When minimising entropy -i.e. creating clusters where instances with the same labels are in the same clusters-, one can focus on *Spearman Coalitional* influences, whatever clustering method. Based on purity and focusing on clusters with one majority label, a best practice can be to try *Spearman Coalitional* and *TreeSHAP* with multiple clustering techniques to make the most of explanation clustering.

4.4.3 Discussion

Clustering on XML influences showed better results than clustering on raw data, regardless of the percentage/number of clusters or the performance of the modelling, especially for Shapley-based XML methods. This behaviour was seen in multiple clustering techniques working differently around data to compute clusters. The influences seem to contain information allowing a better clustering, probably by highlighting the most significant attributes for each instance or removing noises from raw data. This finding seems consistent with the results of Cooper et al. (2021) while showing a more global approach, working with other XML methods than *SHAP* and a hundred of datasets.

Separating the instances correctly and incorrectly classified by the model also gives better results than keeping all the instances together. Since the information in the two subgroups is different, they each seem to create noise in the information of the other subgroup. Indeed, the misclassified instances are often outliers or critical instances in the dataset. Their behaviour is different from the general behaviour of the data, whereas correctly classified instances follow the behaviour that the model detects. However, as some misclassification may result from bias in a subgroup of the data or from the atypical behaviour of that subgroup compared to the whole dataset, it is of great interest to study them as a priority. When separating correctly and incorrectly classified instances, the differences in cluster quality seem to be more pronounced with the *Spearman Coalitional* method than with *SHAP*. The contribution seems to depend on the XML method used, probably because of the calculation of influences since *SHAP* creates perturbations on the instances and *Spearman Coalitional* keeps the input data as it is. A limit to these subgroups' separation is also the decrease of its relevance when the accuracy of the model increases. Indeed, the number of false instances logically decreases with increasing accuracy. Creating an XML model and clusters with a low instance count does not make sense and can only lead to data misunderstanding. However, as the accuracy increases, the false instances become mostly outliers of the dataset or biased instances rather than subgroups with their behaviours to analyse. Their small number can be analysed manually without

any particular clustering method.

Explanations clustering being better than Raw clustering also emerges when focusing on the optimal number of clusters for each dataset, clustering technique and explanation type. *Spearman Coalitional* and *TreeSHAP* appear to be the best local explanation methods to perform explanation exploration. The mean number of clusters also did not correlate with a better purity or entropy across all the datasets. In this setup, *LIME* again produce worse results than the other explanation methods, suggesting that *LIME* influences are not suitable for clustering, and data exploration through clustering. Shapley-based methods then seem more reliable for exploration, as *SHAP* also perform well with all clustering techniques.

Finally, the proposed approach also adds another use of influences. Clusters based on influences can be used to focus on sub-groups of data to be studied. Clustering can be combined with other approaches to understand the clusters created, like rule-based algorithms or instance selection. As mentioned before, the inner properties of each clustering technique can be used to explore clusters. Hierarchical clustering can be valuable for exploring similar influences and instances or how influences behave with different numbers of clusters/on different hierarchical levels. The data distribution and variance in each cluster from EM clustering can explain what attributes are important for each cluster and may explain how clusters are built. Medoids from K-medoids can be used to select representative instances and summarise each cluster. Density-based approach may allow the discovery of prototypes from high-density clusters and outliers/critic instances from low-density clusters, enhancing the understanding of the dataset and summarising it as with medoids. Our results reinforce the idea that influences can be considered as new inputs for finer analysis on the ML modelling pipeline, to gain a more in-depth and concise understanding of the ML model and the underlying data.

4.5 Conclusion

This Chapter details the work carried out on the use of explanations as a new set of data to be explored. We propose a complete framework to analyse local explanations for data exploration. We aim to discover categories of explanations, so instances, to study them from a perspective other than the raw data and with the knowledge retrieved from modelling.

We combine local attributive XML methods with clustering to explore the space of influence data and uncover new insights about the explanations, the prediction, the modelling and the dataset. We provide clusters of similar instances based on their explanations to assist and improve data analysis based on explanations. We then add another use of influences to the ones from the literature.

We provide a set of experiments to validate the valuable contribution of influence-based clustering for multiple XML methods, clustering techniques families and varying numbers of clusters. The clusters from the influence-based framework are more homogeneous and of better quality whatever the XML methods and the clustering techniques used.

We prove that the explanations clusters are of good quality and pertinent, even for low-performance models and misclassified instances. We show the advantages of splitting the well- and misclassified instances by the model when studying a dataset as a whole, as it highlights the most important subgroups of data and the behaviour of outliers simultaneously. We finally highlight the different behaviours of explanation clustering when considering the optimal number of clusters and provide a medical example of how clusters

of explanations can be used in real-world applications and support data analysis.

Based on our results, our approach could be extended for other supervised tasks. Clusters can also help select informative instances and provide a small number of instances to users. These instances can help to understand datasets and modelling using examples rather than statistical information. Based on the different advantages of each clustering technique family, we can explore how to make the most of each cluster, to better understand the explanations, the prediction, the modelling and the dataset.

Chapter 5

Explanations in user's hands: explanations for medical applications

Contents

5.1 Introduction

Among the potential uses of explanations, a significant part involves their contribution to the end-users of machine learning models. Providing appropriate information to help users understand ML functions and decisions is one of the critical aspects of explanations. However, user tests are often absent from evaluations of explanation methods and the number of implementations of explanations in real-world applications documented in the literature is limited (Liao et al., 2020). Bhatt et al. (2020) find that the majority of explanation deployments are meant for machine learning engineers and to debug the model itself, not for end-users and the persons affected by the model. Limitations to providing explanations to end-users include the lack of framework on how to provide explanations, the need for domain experts to evaluate explanations, the risk of spurious correlations explanations, the lack of causal intuition, and the latency in computing and showing explanations in real-time (Bhatt et al., 2020). Baniecki et al. (2023) also claim that a closed-box machine learning model cannot be explained by only a single explanation method, which highlights only one perspective of the model functions and decisions. They define isolated explanations as "prone to misunderstanding, leading to wrong or simplistic reasoning".

Based on the works in Liao et al. (2020); Baniecki et al. (2023), we focus on how combining multiple explanations methods can lead to better application of explainability for end-users in real-world applications. Based on our previous work, we hypothesise that analysing explanations can also enhance their usability, actionability and tractability for end-users, especially in user interfaces built for them.

In this Chapter, we want to emphasise the importance of users in the use of explanations. In Section 5.2, we show how users' understanding of explanations can be enhanced with appropriate user interface and applications of explanations. We build question-driven user interfaces displaying explanations and implement them in medical applications. We show our creation process from the first mockups to the actual user interface and we justify the combined use of several explainability methods, types of graphical and textual visualisations and explanation-based analysis tools. In Section 5.3, we highlight how the same dataset, modelling and explanations can be used in different setups, to also enhance the understanding of explanations by users, and their usability and actionability. In our case, we present one "real-world" medical application and one medical exploratory data analysis on the selected dataset. Then, in Section 5.4, we describe our exhaustive experimental protocol for carrying out user tests, to assess the relevance and impact of the explanations and their analysis. We expose our primary and secondary hypotheses, the materials used - dataset, modelling and explanations -, and detail our methods: the selection of users, the experimental design, the statistical techniques to evaluate our results and the expected bias and limits we consider in our experiments.

Part of the work mentioned in this Chapter has been published in the following articles: Excoffier, Escriva, Aligon, and Ortala (2022a); Escriva, Doumard, Excoffier, Aligon, Monsarrat, and Soulé-Dupuy (2023b)

5.2 Displaying explanations in medical applications

As part of my thesis, I work at Kaduceo company in the implementation of explainability in medical applications aimed at hospitals and healthcare professionals. In this context, we explore how multiple approaches of explanations, especially global and local, can be

combined. We aim to give an overview of the modelling and the prediction, as medical practitioners can easily and efficiently understand the model, the data and the prediction for each patient. We also want users to be able to carry out their own analyses and compare their expertise with the information extracted from the explanations.

5.2.1 Mock-ups design

To build our user interface, we combine multiple explanation approaches: (1) a global approach to give the users an overview of the modelling and the data analysed, and (2) a local approach to explain prediction for each patient precisely and enhance the understanding of each patient.

For the global approach, based on the "XAI Question Bank" in Liao et al. (2020), we select the question we want to answer thanks to explainability, based on our users' needs. We build the first mock-up[1] for global explanations in Figure 5.1 by focusing on the following: "How does the system make predictions?". Especially, we focus on the questions "What attributes does the system consider?", "How does attribute X impact the predictions ?" and "What are the top attributes that determine its predictions?".

Visualisation 1 represents the beeswarm plot of the explanations, where each point is one instance of the dataset. The colour range represents the attribute value in the dataset, from low-value in blue and high-value in pink, and the X-axis represents the influence value of the attribute. This information is completed by visualisation 2 which displays the average influence for each attribute over the entire dataset. For qualitative attributes, average influence can be displayed based on each category by clicking on the attribute name. With these visualisations, we aim to summarise how attributes values and influences interact based on the modelling and how each attribute impacts the predictions.

In addition to global explanations based on post-hoc explainability, we choose to focus on classical statistical analysis with univariate and bivariate analysis. Visualisations 3 and 4 are Partial Dependence Plot (PDP) between the values of each attribute and respectively the influence of the attribute or the prediction. These visualisations highlight for each attribute the interaction between attribute values and the influences or the prediction, in more detail than the previous visualisations 1 and 2. PDPs are known to be intuitive and understandable, and the broad outlines of the model's behaviour can be extracted from these visualisations. Next, bivariate analysis is used to study the relationship between two variables in the dataset, at both the raw data and influence levels. Firstly, the x and y axes are used to position each instance according to the influence of the attributes of interest. Secondly, the colours and sizes of the circles represent the raw values of the attributes. This highlights the relationship between attributes in the raw data and the impact of these relations in the explanations. Although less intuitive than simpler visualisation, it is useful for discovering potential correlations between variables and analysing how the explanations between these two variables behave.

Finally, we display patient profiles to summarise the dataset. These patients are selected based on clusters computed on influences. These patients can be seen as prototypes of the dataset, representative of the behaviour of groups of patients. It can help understand the factors that are important for each group of patients, and therefore gain an overview by looking at specific cases.

[1]Figures and diagrams in the Mock-ups only serve as examples, data between each visualisation are unrelated.

Figure 5.2 is the mock-up for local explanations, when focusing on a particular patient. As for global explanations, we focus on providing diverse information to understand prediction comprehensively and efficiently. Based on the "XAI Question Bank" (Liao et al., 2020), we focus on the "Why?" and grasp some intuitions on "How to be that? (a different prediction)" and "How to still be that? (the current prediction)". With local prediction, we try to answer to "Why/how is this instance given this prediction?" especially "What feature(s) of this instance determine the system's prediction of it?"

First, we provide a view of the dataset and their prediction, to select the patient of interest. When chosen, the user accesses the local explanation in visualisation 7 in the mockup 5.2, where the influence of each attribute is shown and helps understand which attributes contribute to the prediction for this particular patient. Visualisation 8 allows a deeper view by providing the average influence of each attribute for the predicted class. The aim of combining these two visualisations is to help the user understand the prediction for this patient and to position the explanations to other explanations for this same prediction.

Then, we provide profiles of similar patients to give more context to the selected patient and how some changes in attributes can change or not the prediction. We want to use the idea of counterfactual explanations and minimal/maximal changes to change/conserve a prediction while using existing examples in the dataset. This idea links to the questions "How should this instance change to get a different prediction ?" and "What is the scope of change permitted for this instance to still get the same prediction?" of the "XAI Question Bank" (Liao et al., 2020), especially "What kind of instance gets the same prediction?" and "What would the system predict for a different similar instance?". This section can also be easily updated to display both similar and dissimilar patients, in terms of prediction and/or raw data, to give an even more global overview of the modelling behaviour for similar patients.

The highlighted patient is also linked to the possible actions displayed. Based on the dataset and the prediction, we highlight attribute values that increase or decrease the prediction probability. These actions can help define a therapeutic protocol for patients, supplementing and summarising the information already gathered from explanations and similar patients.

These mock-ups were implemented in Kaduceo company's applications, and adapted according to the modelling scenarios and the needs and preferences of the healthcare users.

5.2.2 Implementation in medical applications

Figure 5.3 shows the main user interface for model performances and global explanations, on a task that computes the patients' readmission risk based on their information and their previous hospital stays. This page is displayed after the modelling training. The top half of the interface summarises the model performances based on multiple well-known metrics and visualisations for Machine Learning: F1 score, accuracy, AUC score, Sensitivity, Specificity, Confusion Metric and ROC Curve. It aims to give the user insight into how well the model performs on this task.

The model explanation part is then divided into the ranking of attributes by importance in the prediction and the main groups of patients identified. Attributes importance ranking is displayed through text and visual representation. Each attribute can also be explored more specifically, as seen in 5.4. This figure shows the interface accessible with

the "See more" button, which displays the Partial Dependence Plot between the dataset values and the influences for the attribute of interest.

The "main groups identified" section displays the information for the clusters computed on influences. For each cluster, the average risk or readmission, the number of patients in the clusters, the proportion of these patients in the total dataset and the most important attributes of the clusters are directly visible in the interface. Then, thanks to the "Variables" button, for each cluster, the 5 most important attributes with their values and influences on the prediction and the influences of the central patient of the group appear, as visible in Figures 5.5. This information can help understand each cluster and important behaviours in the dataset.

Finally, Figure 5.6 shows how we implemented local explanation in the context of predicting patients' hospital stay duration. For each patient, we display the prediction in days, the confidence in this prediction and, on the right, a chronology of the main events that occurred during the hospitalisation. The local explanation is displayed along with the patient information, to help understand the prediction and what attributes most increase or decrease the prediction.

In summary, the implementation of explanations in current medical applications is a challenge in itself, requiring mainly adaptations for each prediction task. An interesting approach seems to be to combine several explanation approaches to provide the broadest view to healthcare professionals and let them analyse the information with their medical knowledge.

5.3 Analysis of medical ML explanations

More than only explaining a single prediction for one patient, we already show in Chapter 4 that explanations can also be used in exploratory data analysis. In this section, we use medical exploratory data analysis to understand the modelling and retrieve new knowledge from the data. On the same dataset, we aim to show how explanations can be used for multiple purposes: risk stratification for precision medicine and exploratory analysis to better understand disease and/or the behaviour of typical and atypical patients.

5.3.1 Materials & Methods

For this study, we use an open dataset, the Acute Inflammation dataset[2]. The Acute Inflammation dataset was created to develop an expert system for urinary disease. It consists of 120 patients, described by six attributes: Temperature *(35°C-42°C)*, Occurrence of nausea *(yes-no)*, Lumbar pain *(yes-no)*, Urine pushing (continuous need for urination, *yes-no*), Micturition pain *(yes-no)* and Burning of urethra, itch, swelling of urethra outlet (abbreviated as Urethra burning, *yes-no*). Each patient can have two different diseases of the urinary system: acute inflammation of the urinary bladder (AIUB) and acute nephritis of renal pelvis origin. Patients may suffer from both diseases simultaneously, so this dataset is a multi-output problem. We only focus on the AIUB disease to have a binary classification problem. Medical staff defined AIUB as *"a sudden occurrence of pains in the abdomen region and the urination in form of constant urine pushing, micturition pains and sometimes lack of urine keeping. The temperature of the body is rising, most often not*

[2]Dataset: `https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations`

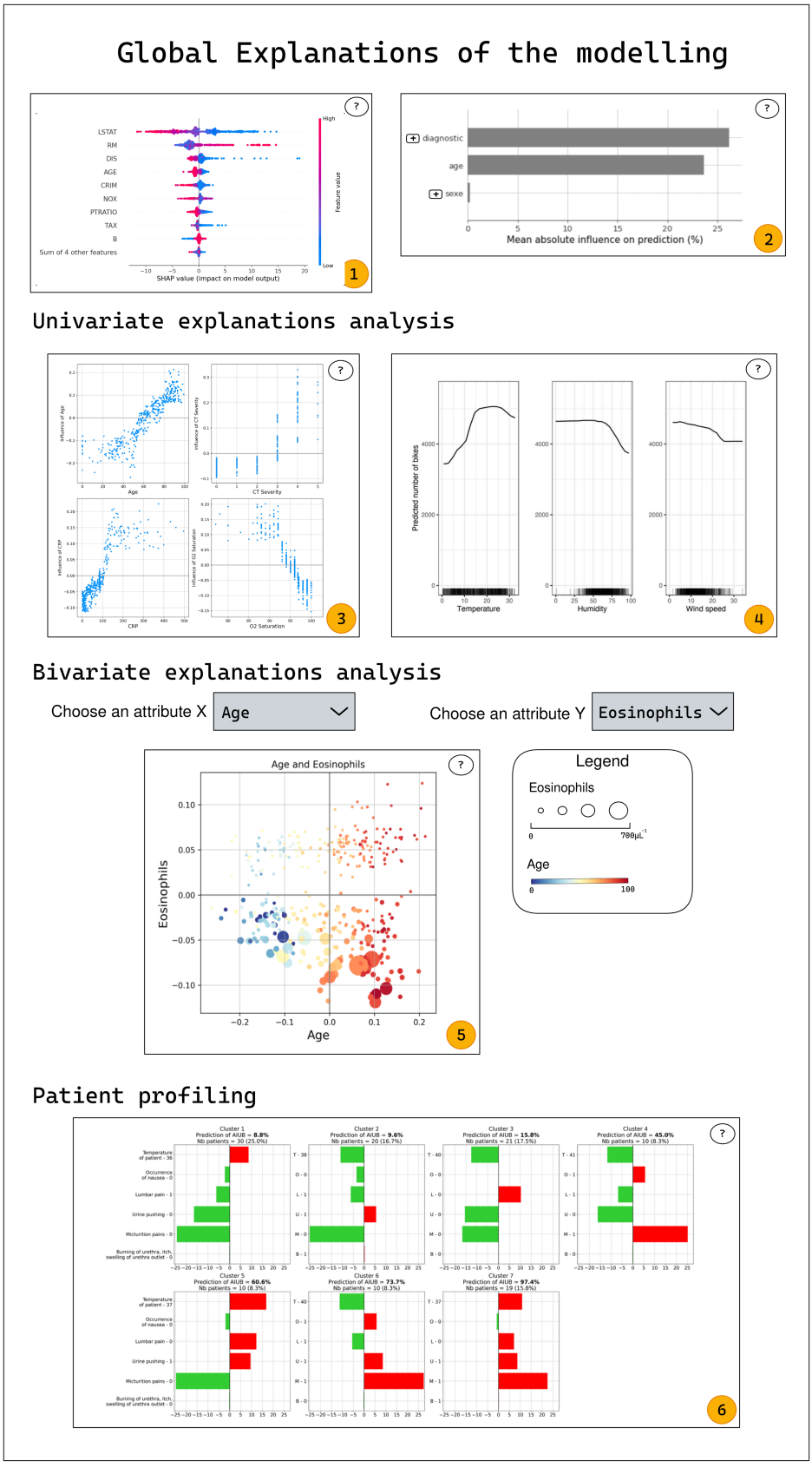
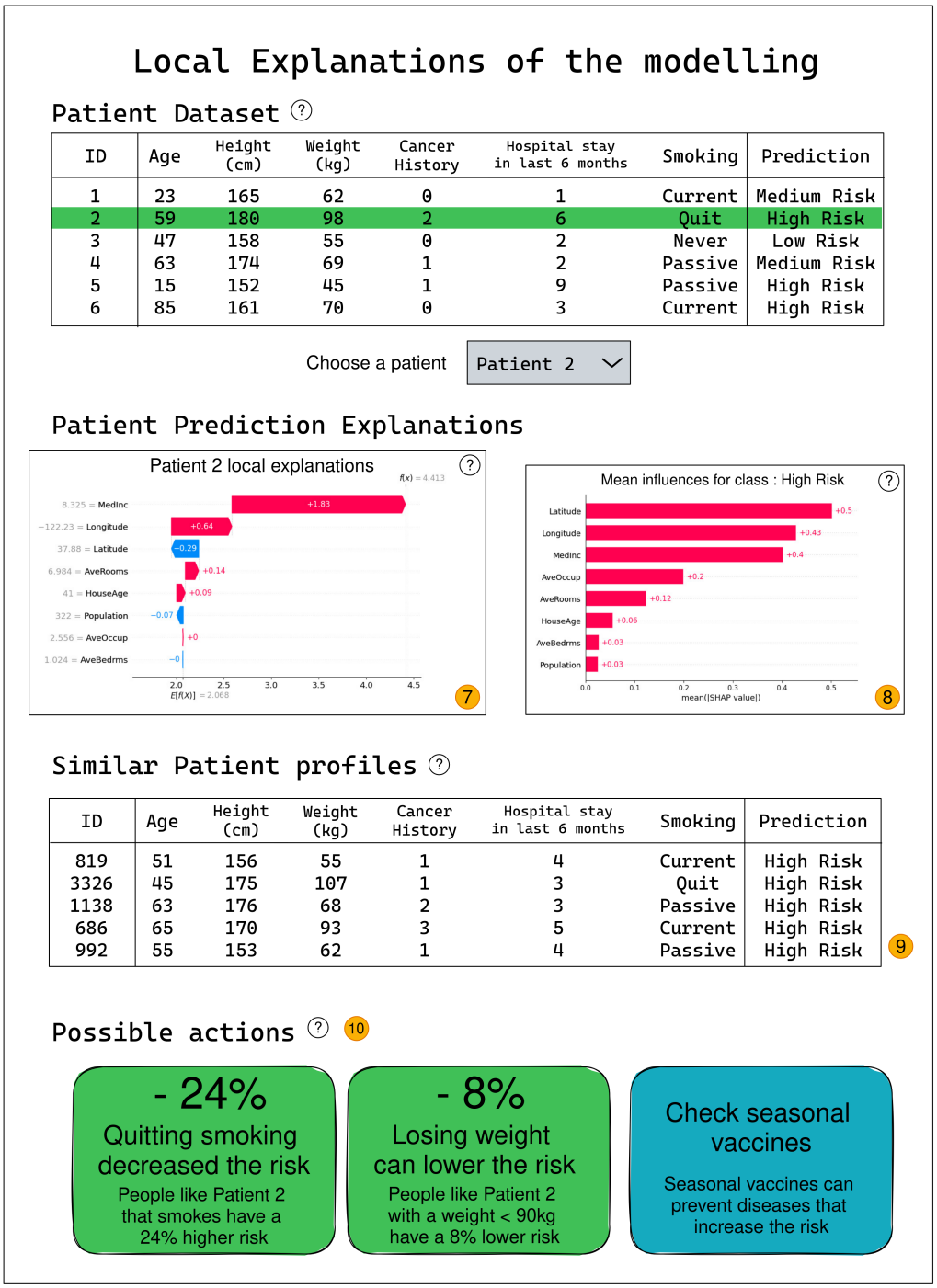


Figure 5.1: Mock-up for global explanations for a medical application.



ID	Age	Height (cm)	Weight (kg)	Cancer History	Hospital stay in last 6 months	Smoking	Prediction
1	23	165	62	0	1	Current	Medium Risk
2	59	180	98	2	6	Quit	High Risk
3	47	158	55	0	2	Never	Low Risk
4	63	174	69	1	2	Passive	Medium Risk
5	15	152	45	1	9	Passive	High Risk
6	85	161	70	0	3	Current	High Risk

ID	Age	Height (cm)	Weight (kg)	Cancer History	Hospital stay in last 6 months	Smoking	Prediction
819	51	156	55	1	4	Current	High Risk
3326	45	175	107	1	3	Quit	High Risk
1138	63	176	68	2	3	Passive	High Risk
686	65	170	93	3	5	Current	High Risk
992	55	153	62	1	4	Passive	High Risk

Figure 5.2: Mock-up for local explanations for a medical application.

above 38C. The excreted urine is turbid and sometimes bloody" (Czerniak and Zarzycki, 2003).

In this dataset, 59 out of 120 patients have AIUB disease. Only the *Temperature of patient* is a quantitative attribute while the five others were all binary variables. Table 5.1 indicates the main characteristics of the global, non-AIUB and AIUB populations. Statistical tests were performed between Non-AIUB and AIUB. Results are presented with mean and standard deviation for quantitative attributes, and numbers and proportions for qualitative attributes which were all binary indicators. P-values were adjusted using Bonferroni correction to control the family-wise error rate.

		Total	**Non-AIUB**	**AIUB**	**p-value**
	Nb patients	120	61 (50.8%)	59 (49.2%)	
Quanti.	Temperature	38.72 (±1.8)	39.15 (±1.9)	38.29 (±1.7)	0.0552
Quali.	Nausea	29 (24.2%)	10 (16.4%)	19 (32.2%)	0.4224
	Lumbar pain	70 (58.3%)	51 (83.6%)	19 (32.2%)	<0.01 **
	Urine pushing	80 (66.7%)	21 (34.4%)	59 (100.0%)	<0.01 **
	Micturition pain	59 (49.2%)	10 (16.4%)	49 (83.1%)	<0.01 **
	Urethra Burning	50 (41.7%)	21 (34.4%)	29 (49.2%)	0.8814

Table 5.1: Population characteristics.

On this dataset, we apply a three-step methodology, common to our two analysis goals. This approach aims to analyse datasets through ML modelling and local explanations. Based on the dataset of interest consisting of patients' medical records and their disease diagnosis, this method allows an understanding of interactions between patients' characteristics and the disease.

(1) The first step consists of *ML predictive modelling*, to evaluate the risk of AIUB disease for each patient based on the understanding of the complex statistical relationship of the dataset. An XGBoost model, a boosted tree ensemble technique (Chen and Guestrin, 2016), is used for its efficiency. We use a nested cross-validation (CV) procedure to provide unbiased modelling (hyperparameters optimisation with an inner 5-fold CV) and to evaluate performances and compute local explanations (through an outer 5-fold CV). As for the model performances, there was an accuracy of 98.33%, a sensitivity of 96.72%, a specificity of 100% and an AUC ROC Score of 99.06%.

(2) Second step is the *explanation of the modelling* to provide individual explanations of the prediction for each patient, corresponding to individual risk and protective factors. *TreeSHAP* (Lundberg et al., 2020), a local attributive XML method for tree-based predictive models, is used to compute influence explanations.

(3) Last step consists of *identifying subgroups of similar patients* to discover local patterns in the data and explain the subgroups' characteristics. K-Medoids algorithm (Park and Jun, 2009) is used for the clustering task to ensure robustness against outliers, while the optimal number of groups was chosen with the Silhouette score. K-medoids algorithm is used on the influence explanations from step (2), with the advantages of taking into account the non-linear interactions discovered by the model while having all features at the same unit.

5.3.2 Risk Stratification

This analysis is an exploration of healthcare risk stratification based on influences from *TreeSHAP*. In healthcare, physicians need to link a patient to a more global context to deliver the most relevant care. Care management through risk stratification (abbreviated as RS) is thus commonly used in healthcare (Dera, 2019). It consists of the identification of several groups, where patients in the same group have similar conditions and risk levels. It also has been shown that physicians' usefulness and confidence in such RS workflows is the highest when it lets a large place for human presence and action (Ross et al., 2017). Predictive tools, associated with explanations explorations, should be able to provide explanations about the individual prediction and enable the user to contextualise the observation and its related prediction: *Is it a well-known or conversely an atypical case?*

Can it be linked to a more general group?. This medical use case would provide physicians and medical staff a clearer view of the different profiles of patients, also called typologies, so that they can adjust or create medical protocols that best fit the specific needs, either in terms of condition and risk level, for each identified patient typology, to deliver the most appropriate care for every patient efficiently.

The influence of each attribute is displayed in Figure 5.7. For each sub-graph, a dot represents a single patient, with the attribute value on the x-axis and the associated influence on the y-axis. *Lumbar pain* and an higher *Temperature of patient* were associated with an lower risk of AIUB, while *Occurrence of nausea*, *Micturition pains*, *Urine pushing* and *Burning of urethra, itch, swelling of urethra outlet* all increased the AIUB risk.

The optimal number of clusters using the Silhouette score is 7. The K-Medoid method also allowed us to find out the most representative patient profile of each group to get a clear view of their characteristics and specificity. Figure 5.8 indicates for each cluster its most representative patient with the associated influences. For each cluster, the associated title indicates the predicted probability of having AIUB and the number of patients in the cluster. Attribute names are represented by their initials for the sub-graph not located on the far left. Initial values of attributes are indicated after the hyphen. A positive influence (represented in red for visual help) increases the AIUB risk. The population is rather uniformly distributed among the clusters, indicating no atypical cases in the dataset. Clusters are different in terms of risk level and risk factors. Indeed, even when two clusters have close risk levels, they significantly differ when considering the risk factors. Clusters 1, 2 and 3 have low levels of AIUB risk, with only one slight risk factor. Cluster 4 has two risk factors, including *Micturition pains*, a heavy risk factor. Cluster 5 has no *Micturition pains* but three other risk factors. Clusters 6 and 7 had the highest risk level with several risk factors.

To sum up, this exploration identified several medical cases in this context (*i.e.* the different typologies of patients). Identified effects of attributes are coherent with medical indications about AIUB given in the original paper (Czerniak and Zarzycki, 2003). Mixing ML modelling and computation of attributes' influences allow us to quantify and identify both the risk level and the associated risk factors which were then used to construct the final RS. It extends the use of influences (*i.e.* local explanations) to get a more efficient view of the situation by identifying the existing typical subgroups that differ either in their situation (risk level) or condition (protective and risk factors). The focus we also made on the most representative case of each group (*i.e.* the medoid) gives an even clearer understanding of the global situation. Moreover, this approach can help doctors to adapt or create medical guidelines and protocols to best meet each group's specific needs. For example, using group decision rules, practitioners could associate a new patient with a larger group and then provide that new patient with the appropriate care defined for that specific group.

5.3.3 Exploratory data analysis

Exploratory data analysis techniques are meant to investigate data and discover patterns, make and test hypotheses with the help of statistics, graphical representation, clustering or predictive tools. In particular, Bottom-Up approaches aim to find patterns and gain insight by analysing data without making *a-priori* hypotheses (Morgenthaler, 2009; Wirsch, 2014). Among the tools for exploratory data analysis, predictive approaches, primarily through machine learning, have made it possible to capture more complex sta-

tistical phenomena in the data that classical statistical techniques cannot understand. Local explanations also allow investigation of the reasons behind the model prediction for each instance.

Then, our objective is to apply a bottom-up exploratory data analysis approach on a medical dataset, on both explanations and raw data, to highlight and compare the knowledge retrieved in both data spaces. We show that explanations can allow a deeper dataset investigation. This study can also show the usefulness of seeing explanations not only as an outcome but also as a tool.

First, we focus on analysing raw data, statistically and based on the clusters created. Then, we explore the *TreeSHAP* explanations for all the instances and the explanations clusters. To understand clusters, decision rules are created with the Skope-Rules algorithm (Gardin et al., 2019). Rules are computed to ensure perfect precision and recall of all rules: all instances of the cluster respect the rule, and all instances respecting the rule belong to the cluster.

Raw data Analysis *Populations and statistical tests.* Table 5.1 shows the main characteristics of the dataset using raw data only, with results from statistical tests performed on AIUB and non-AIUB patients. Three attributes are defined as statistically significant to detect AIUB: Lumbar pain, Urine pushing and Micturition pain. Patients with lumbar pain seem to have less AIUB while having urine pushing and micturition pain correlate with an AIUB diagnosis.

Clustering and rule-based analysis. To create homogeneous groups of patients, one method consists of performing clustering. The optimal number of clusters was 11, based on the silhouette scores in Table 5.2. Table 5.3 shows the rules defined by Skope Rules to describe each cluster. Rules have a median of 2.5 attributes per rule. All rules have perfect precision and recall with a maximum of three attributes, which is a small enough number of attributes to facilitate the interpretation of each rule. The most used attributes are urethra burning and temperature with six distinct occurrences, both previously defined as not significantly discriminating for AIUB diagnosis in Table 5.1. Only one cluster, Cluster 2, uses only significantly discriminating attributes. Also, having eleven clusters makes it challenging to easily understand the rules and clusters.

Table 5.2: Silhouette Score for multiple numbers of clusters for Raw data.

K	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Raw	0.56	0.44	0.37	0.42	0.46	0.51	0.54	0.54	0.56	**0.57**	0.56	0.56	0.56	0.56

XML analysis *Local post-hoc explanations.* Figure 5.9 shows the *SHAP* mean absolute influences and the distributions of influences based on the attribute value. In the distribution plot, the attributes are sorted in decreasing attribute importance from top to bottom and each dot represents an instance from the dataset, its colour representing the raw value of the attribute. The position on the x-axis represents the contribution of the attribute to the prediction of this individual, and overlapping dots are spread on the y-axis.

The three most important attributes were Micturition pain, Urine pushing and Temperature. Micturition pain and Urine pushing increases the risk of having AIUB. On the contrary, a higher temperature decreases the probability of having AIUB. In particular,

Table 5.3: Decision Rules for clusters based on raw data, with the number of patients per cluster and the mean percentage of AIUB-risk.

	Rules	**Nb**	**Mean %**
1	Nausea = 1 & Urine pushing = 0	10	45.6
2	Lumbar pain = 0 & Urine pushing = 0	10	10.7
3	Nausea = 1 & Urethra burning = 1	9	72.2
4	Temperature < 39.85 & Micturition pain = 0 & Urethra burning = 1	10	13.0
5	Lumbar pain = 0 & Urethra burning = 1	20	97.1
6	Temperature < 38.95 & Temperature > 36.65 & Urine pushing = 0	13	11.0
7	Temperature < 38.95 & Lumbar pain = 0 & Micturition pain = 0	10	59.9
8	Nausea = 1 & Urine pushing = 1 & Urethra burning = 0	10	73.6
9	Temperature > 39.85 & Nausea = 0 & Urethra burning = 1	11	11.2
10	Lumbar pain = 0 & Micturition pain = 1 & Urethra burning = 0	10	97.1
11	Temperature < 36.65 & Urethra burning = 0	7	11.2

having urine pushing also seems to have less impact on the prediction than not having urine pushing. In contrast, Nausea and Urethra burning have little to no impact on the predictions. For nausea, *SHAP* describes that having them increases the risk of AIUB for some patients and a subgroup of patients is identified.

Figure 5.10 shows the distribution of influences only for patients having Nausea. Looking in detail at these patients, they all suffer from lumbar pain, micturition pain and temperature above 40°C (which is higher than the dataset mean). There seems to be a subgroup of patients with a strong relationship between these four attributes. Moreover, for this subgroup of patients, there is a strong correlation between the attribute Urine Pushing and the presence of AIUB: when patients have urine pushing, they have an AIUB; when they do not have urine pushing, there is no AIUB. This subgroup is probably best to study, as the nausea attribute may create a real-world bias due to its strong association with other attributes in the dataset.

Clustering and rule-based analysis. As one subgroup is already discovered, clustering can help to find other subgroups of interest. For clustering on *SHAP* influences, the optimal number of clustering is set as 7, based on the silhouette score in Table 5.4. Table 5.5 shows rules defined by SkopeRules for clusters based on influences. These rules have a median of two attributes per rule and focus mainly on statistically relevant attributes. Only one decision rule consists of three attributes, and the most used attribute is Urine Pushing, with five occurrences. As shown before for the "Nausea subgroups", this attribute is the most important for patients with Nausea (clusters 4 and 6) and also for patients with lumbar pain (clusters 3 and 5). Urine Pushing does not appear in rules only for clusters 2 and 7, the two biggest clusters, where AIUB-risk is respectively very low and very high. These clusters may be interesting to study from a medical point of view to understand patients' characteristics and why the Urine-pushing variable is not the most relevant variable to distinguish them from other clusters. Also, although Micturition pain is the most influential attribute for SHAP, it is not very present in the rules, mainly

because this attribute seems replaced by the attribute Nausea in the clusters since there is a strong link between having Nausea and Micturition pain.

Table 5.4: Silhouette Score for multiple numbers of clusters for XML data.

K	2	3	4	5	6	7	8	9	10	11	12	13	14	15
XML	0.59	0.59	0.52	0.62	0.69	**0.76**	0.74	0.69	0.63	0.61	0.61	0.61	0.62	0.67

Table 5.5: Decision Rules for clusters based on influences, with the number of patients per cluster and the mean percentage of AIUB-risk.

	Rules	**Nb**	**Mean %**
1	Temperature <= 38.89 & Urine pushing = 0	20	11.0
2	Micturition pain = 0 & Urethra burning = 1	21	12.0
3	Lumbar pain = 0 & Urine pushing = 0	10	10.7
4	Nausea = 1 & Urine pushing = 0	10	45.6
5	Lumbar pain = 0 & Urine pushing = 1 & Micturition pain = 0	10	59.9
6	Nausea = 1 & Urine pushing = 1	19	73.0
7	Lumbar pain = 0 & Micturition pain = 1	30	97.2

To sum up, both raw data and explainability methods detect patterns in the data, subgroups of patients and information about the relationship between the AIUB disease and patients' symptoms. In addition to the information learned in the literature (Czerniak and Zarzycki, 2003) and found in the raw data analysis, the explanation-based data analysis allowed risk and protective factors to be identified more concisely. Rules are mainly based on statistically significant attributes, adding interactions between attributes, and with the target class, compared to raw data analysis. The smaller number of clusters and attributes in each rule also simplifies the understanding of patient subgroups and the relationship of each attribute to the AIUB risk. With raw data, multiple clusters have similar mean percentages of AIUB risk and almost identical patients. The differences between these clusters are often based on attributes not important for detecting AIUB. This behaviour can be beneficial to study the dataset in-depth, less for discovering the attributes that truly impact the diagnosis of the disease and for capturing concise knowledge. The conciseness provided by influences also makes it easier to assign a new patient to a subgroup of patients to study their disease and risk factors. This advantage comes from the ability of ML modelling to capture more complex relationships than traditional statistical methods. Finally, the explanation data allowed the discovery of relevant subgroups of patients, including those with nausea. This subgroup has strong relationships between several attributes, and the presence of AIUB is based solely on the attribute Urine Pushing, making its study engaging for understanding the mechanisms of the disease in some patients. Finding this type of subgroup can help to investigate biases in the dataset, especially around the attribute Nausea.

5.3.4 Discussions

Based on these two explorations of the Acute Inflammation dataset, similar conclusions can be drawn: the results were obtained thanks to the explanations, and would not have

been possible with the initial data alone. Using explanations appears a strong option to consider when the final objective of the data analysis is either the construction of an RS or patient profile discovery. In the second example, we show that combining raw data and explanations adds significant value to exploratory data analysis. We want to bear in mind that exploring the raw data, the model and the explanations can facilitate users' understanding and increase the contribution of the explanations by giving them meaning. Explanations do not appear as data that is independent of the initial data but as a complement and support to its understanding.

Finally, the proposed exploration should be applied and tested in more complex medical contexts, with datasets having different characteristics, such as more observations, more attributes, and more variability leading to lower model performances. Confronting our analysis with medical experts is also necessary to ensure the relevance, tractability and actionability of our experiments, interface design introduced in 5.2 and explanation-based analysis.

5.4 Medical User Tests Protocol

This experimental protocol was created in collaboration with Julien May, a final-year student at Toulouse Health University and Paul Monsarrat, Professor at Toulouse Health University.

5.4.1 Purpose

In the existing literature described in 2.3, few domain experts are present in studies evaluating the performance of explanations. While there are now many publications on explainability, experiments on its contribution to an expert's performance are rarer. With this in mind, we have designed an experiment focused on medical experts, to evaluate the interest and potential benefits that XML could bring to healthcare professionals, compared with the contribution of AI alone. We also want to assess the value of the explanations about the user's level of expertise in the medical task used in the experiments.

Primary Hypothesis If we formulate the following two hypotheses:

- H_0 (null hypothesis): The use of XML does not improve the accuracy and/or speed of diagnosis for the medical expert compared to the use of data alone or combined with machine learning.
- H_1 (alternative hypothesis): The use of XML improves the accuracy and/or speed of diagnosis for the medical expert compared with the use of data alone or combined with machine learning.

The main objective of this study will therefore be to attempt to demonstrate that hypothesis 1 is true and that the null hypothesis is false, or that the null hypothesis is not shown to be false. For that, we define α, "the probability of the study rejecting the null hypothesis given that the null hypothesis is true" (Dalgaard, 2008), to 0.05.

Secondary assumptions In addition, several secondary targets can be defined in the form of the following queries, which we will attempt to answer:

- Does the improvement in diagnosis accuracy and/or speed thanks to the explanations differ according to the practitioner's expertise?
- Is there an improvement in the accuracy and/or speed of diagnosis with the use of data associated with the ML compared with the use of data alone?
- Is there a more significant improvement in the accuracy and/or speed of diagnosis for patients misdiagnosed with the use of data alone?
- Do the medical expert and the *SHAP* method give the same order of importance to the variables when making their diagnosis?

These secondary objectives are of interest to understand how medical experts react to the predictions and explanations and to be able, if necessary, to adapt the explanations to the medical experts after analysing the results.

5.4.2 Materials

To design this experiment, we chose an open dataset to ensure data reproducibility and anonymity. We use the open dataset SA-Heart (Rossouw et al., 1983). It retrieves retrospective information about 462 South Africans from the heart-disease high-risk region of the Western Cape. The main objective of this dataset is to predict the presence of coronary heart diseases (CHD) according to nine attributes: **tobacco** (cumulative consumption tobacco), **age** (at the onset), **LDL** (low-density lipoprotein cholesterol), **adiposity** (estimation of the body fat percentage), **obesity** (through the body mass index), **family** (family history of heart disease, present or absent), **alcohol** (current alcohol consumption), **SBP** (systolic blood pressure) and **type-A** (Type-A behaviour scale).

Table 5.6 indicates the main characteristics of the global, non-CHD and CHD populations. Statistical tests were performed between Non-CHD and CHD. Results are presented with mean and standard deviation for quantitative attributes, and numbers and proportions for qualitative attributes which were all binary indicators. P-values were adjusted using Bonferroni correction to control the family-wise error rate. Most of the attributes are statistically significant to detect the two classes. Only the *Alcohol*, *Obesity* and *Type-A* attributes are not significant to distinguish between the patients with or without CHD. This feature - most attributes are significant between the classes - in the choice of the dataset is motivated by the presence of patients with strong characteristics for detecting coronary heart disease. This will make it easier to detect sick patients, particularly for cardiology experts, and therefore polarise the effects of the explanations for doctors. In addition, the presence of patients with strong characteristics of the disease and other patients with less strong characteristics may make it possible to detect a different impact of the explanations.

For modelling the data, we train an XGBoost classifier with default parameters (Chen and Guestrin, 2016) and achieve a 0.91 accuracy and 0.90 ROC-AUC score. We exclude the *Type-A* attribute as modelling performances were better without it. Table 5.7 shows additional classification metrics on each class for the trained models and the correspondence between the predictions and the true labels on all the dataset instances (called confusion matrix). It shows that the model has better performances in predicting the "Non-CHD" class than the "CHD" class. However, the number of *False positive* and *False negative* predictions are low based on the confusion matrix.

		Total	**Non-CHD**	**CHD**	**p-value**
	Nb patients	462	302 (65.4%)	160 (34.6%)	
Quanti.	SBP	138.33 (±20.5)	135.46 (±18.0)	143.74 (±23.7)	<0.01 **
	Tobacco	3.64 (±4.6)	2.63 (±3.6)	5.52 (±5.6)	<0.01 **
	LDL	4.74 (±2.1)	4.34 (±1.9)	5.49 (±2.2)	<0.01 **
	Adiposity	25.41 (±7.8)	23.97 (±7.8)	28.12 (±7.1)	<0.01 **
	Type-A	53.1 (±9.8)	52.37 (±9.5)	54.49 (±10.2)	0.2394
	Obesity	26.04 (±4.2)	25.74 (±4.1)	26.62 (±4.4)	0.2835
	Alcohol	17.04 (±24.5)	15.93 (±23.5)	19.15 (±26.2)	1.0
	Age	42.82 (±14.6)	38.85 (±14.9)	50.29 (±10.6)	<0.01 **
Quali.	Family	192 (41.6%)	96 (31.8%)	96 (60.0%)	<0.01 **

Table 5.6: SA-Heart patients characteristics.

		Prediction				
		Non-CHD	CHD	Precision	Recall	F1-Score
True Labels	**Non-CHD**	286	16	0.93	0.95	0.94
	CHD	23	137	0.90	0.86	0.88

Table 5.7: Classification performance metrics and confusion matrix for the model trained on the SA-Heart dataset.

For explanation, we use the *TreeSHAP* approach (Lundberg et al., 2020) and compute explanations for all patients based on the probability of coronary heart disease. Figure 5.11 displays global explanations of the modelling, based on the local explanations.

5.4.3 Methods

Users The target population for the experiment is doctors, particularly in cardiology, to assess the potential benefit of using explanations in association with ML. However, having a population representative of the target population in this study is not easy. The main reason for this is the work overload of French healthcare staff due to understaffing. We will therefore be trying to recruit as many doctors as possible, as well as lecturers and researchers from medical schools, ideally experts in cardiology, and final-year students with two to three years of clinical experience, to increase our numbers. The eligibility criteria for recruitment are as follows:

- be a doctor who has been trained or is in training;
- have cardiology knowledge;
- a minimum of two years of clinical experience.

The recruitment criteria lead to a population with a heterogeneous level of expertise. To correct a potential bias linked to this heterogeneity, the experts will therefore be clustered according to two criteria:

- a subjective criterion regarding their level of expertise in cardiology;
- an objective criterion based on their status (cardiologist, doctor from other specialities, 6th-year student, professor-researcher in medicine faculty).

These clusters can be compared with the whole group of experts to measure an increase or decrease in the criteria observed. This process will make it possible to measure the impact of the explanations compared to the users' expertise levels.

The target is to recruit at least 100 users, to have sufficient results to establish statistics with a reasonably solid basis.

Experimental Design Each user will be randomly allocated 9 anonymised patients from the dataset, whom they will see during three different trial phases explained below. Each patient will also be monitored by 10 practitioners.

Each patient presentation will be subdivided into three modes, inspired by Jesus et al. (2021):

- Mode 1 (Control mode): presentation of the patient with the data alone;
- Mode 2: presentation of the patient with the data and the ML result;
- Mode 3: patient presentation with data, ML result and associated explanations.

The patients will be randomly divided into three groups A, B and C, and the experiments into 3 phases. During the first phase, patients in Group A will be presented in Mode 1, those in Group B in Mode 2 and those in Group C in Mode 3. During the second phase, patients in Group A will be presented in Mode 2, patients in Group B in Mode 3 and patients in Group C in Mode 1. During the third phase, each group will be presented in the remaining mode. Thus, during each phase, each practitioner will see 3 patients in mode 1, 3 in mode 2 and 3 in mode 3. He will see the same 9 patients in each phase, each presented in a different mode to the others. At the end of the three phases, each practitioner will have seen each patient in each mode with sufficient time -a minimum of one week- between each phase to ensure that the data from one mode does not influence the others. We will use Mode 1, where users only access the patient's data, as the control mode to access the impact of providing explanations and Mode 2 to isolate the explanations' impact from that of the ML modelling.

The phases will be spaced a few weeks apart, to avoid practitioners recognising their patients from their data and being influenced in their diagnosis by a diagnosis previously made in another phase.

During each experimental phase, practitioners will be asked the following queries via an online questionnaire made with LimeSurvey[3]:

1. Self-assessment of the healthcare professional's level of expertise in cardiology on a scale of 1 to 10;
2. Query on the health professional's situation: cardiologist, doctor from other specialities, 6th-year student, professor-researcher in medicine faculty;
3. Training: One mode 1 training patient, one mode 2 training patient and one mode 3 training patient. Queries about the user prediction, their trust levels from 0 to 10, their attributes ranking from most useful to least useful;

[3] `https://www.limesurvey.org/`

4. Presentation of the 9 patients (3 according to mode 1, 3 according to mode 2 and 3 according to mode 3) ordered randomly, one after the other. Queries about the user prediction, their trust levels from 0 to 10, their attributes ranking from most useful to least useful;

5. Query about how the patients felt about the XML contribution.

As previously mentioned, we expect to have users with varying levels of expertise. So, the first two queries enable us to characterise users to construct differentiated statistical indicators. We also ask these queries at each phase to monitor how users' self-assessments can differ with time, as it is a subjective evaluation. Queries about expertise levels are displayed to users as in Figure 5.12.

Then, for training and test queries, for each mode, we display the patient characteristics along with the specific information for modes 2 and 3: ML prediction and ML performances for mode 2, with global and local explanations, clusters rules and similar patients from the same clusters added for mode 3. For each mode, users are asked if the patient has a coronary disease with a yes/no query and their level of confidence in their answer on a 10-point scale. Then, users are asked to rank the attributes from most to least useful in determining whether the patient has coronary heart disease. The different user interfaces for Mode 1, 2, and 3 are presented in Figures 5.13, 5.14 and 5.15.

At the end of each phase, users are asked to evaluate assertions based on 10-point scales, about their opinion on the XML contribution. These statements are the following, also displayed in Figure 5.16:

- I better understand patient data with explanations.
- I make a faster decision with explanations.
- Explanations are more confusing than helpful.
- I easily understand the explanations.
- Viewing similar patients is useful in making a decision.
- The model's predictions are sufficient and explanations are unnecessary.
- Explanations are easy to use.
- Explanations are useful in reinforcing my decision.
- Explanations provide too much information.
- Explanations are coherent with my medical knowledge.

Evaluation Methods Based on our primary hypothesis, secondary assumptions and experiment design, we want to mainly evaluate the users' accuracy, the users' response speed, users' confidence and the users' feelings about explanations.

For users' accuracy and response speed, we plan to compare, for each mode, the user accuracy/time in predicting coronary heart disease. We want to assess how the average accuracy/time of users changes as a function of mode, and whether the differences are significant between each mode. We also want to split users based on their expertise

level and their status to study if the average accuracy/time behaves differently for each subgroup of users. More globally, comparisons of performances from each mode can be performed with multiple focuses:

- comparison patient-focus: for each patient independently, how do users' performances vary?
- comparison user-focus: for each user, how does their performance vary between modes?
- comparison expertise-focus: for each cluster of expertise, how does the expertise impact the performances?
- comparison mode-focus: for each mode, how do performances vary between users?
- comparison classification-focus: how do users' performances vary between each mode depending on the correctness of the users' prediction with data alone?

For each focus, statistics about the average accuracy/time will be calculated, as well as p-values when relevant. These p-values will be used to validate or reject H_0 and H_1 based on the previously defined α threshold of 0.05. We will also compute how many times explanations improve users' performances and their proportion among all the results.

To gain more insight into our secondary assumptions, we will use metrics on users' confidence in their predictions, the ranking of variables and users' perceptions of the contribution of explanations. Based on the users' confidence in their prediction, we will compute statistics in the same fashion as for accuracy and response speed metrics and relate these results to previous ones. With this confidence metric, we want to evaluate the link between the confidence of the user and the mode, their performance for the task, and their level of expertise.

For the ranking of attributes based on their usefulness for the user, we would first compare how user responses differ for each mode for the same patient. We want to assess the extent to which users change their ranking for the same patient according to the different modes, particularly between modes 1 and 2, since no additional information on the importance of the attributes is provided. Thus, based on the rankings for modes 1 and 2, we will be able to assess better the differences with mode 3 ranking and the impact of explanations - i.e. the local importance of attributes. For this evaluation, we will use the Kendall-Tau distance to compare the ranked lists (Kendall, 1948). For each patient and each user, we will first compare the mode 1 and 2 rankings to evaluate the inter-variability, then compare the mode 3 ranking to the two first modes and influences to evaluate the impact of explanations. Finally, we will compare results based on the expertise lever of the users.

Finally, to study the users' feelings, we will evaluate the answers to the 10-point scale queries. As mentioned in Norman (2010), parametric statistics are robust and can be used on Likert scales even if the answers are ordinal, not normally distributed and if the number of users is small. We will then study each query individually based on the average response and the distribution of answers, using parametric tests to check statistical significance between users with different expertise levels. We will also search for correlations between the answers to multiple queries using the Pearson correlation factor to see if users from the same expertise cluster have the same answers between all queries.

Expected bias and limits When designing the study, several challenges and potential biases emerged:

- How to recruit healthcare professionals with an already busy schedule?
- How could we obtain as many responses (and therefore as much data) as possible from each practitioner interviewed, without the survey being abandoned along the way or carried out without the necessary care because it was too long?
- How can we avoid getting incorrect answers because of poor understanding or use of the computer interface or the presentation or wording of the query?
- An increase in the practitioner's performance could be observed when answering due to learning and habituation, which would not be linked to any ML or XML contribution. Habituation may also be observed for a particular patient if the different stages of the tests are carried out in succession, always in the same order - in the case, for example, of displaying the data alone, then the ML results and then the XML results, always in the same order. This habituation can impact the contribution of each method - ML and XML - for users.

Some of the biases mentioned above are resolved in the following four ways. To maximise the number and quality of responses simultaneously, we limit the time of the experiment by limiting the number of patients studied by each user. If we consider that the time needed to evaluate a patient is a maximum of 1 minute per mode and that we propose 10 different patients to each user, then the experiment is limited to a 10-minute duration per phase. In this way, we give priority to the quality of the results. This solution also makes it easier to recruit people with busy schedules, since the duration of the test will remain relatively short for each phase. However, there needs to be a sufficient ratio between the number of users and the number of patients to ensure that the results can be used for statistical analysis. Finally, as the test duration is short, the risk of users adapting to the test is reduced.

Secondly, we divide the test mode into different phases so that users do not study all the data in the same order for all patients and practitioners do not become accustomed to each patient. A random allocation of each stage is set up to limit this bias. By separating the stages, we limit the risk of practitioners distinguishing their patients from their data and being influenced by a diagnosis previously made in another stage. This strategy also contributes to reducing the length of the tests and therefore possibly improving the quality of the practitioners' responses.

To recruit healthcare professionals, we can also publish our questionnaire online and share it on national mailing lists. Remove usability tests are as effective as in-lab tests (Selvaraj, 2004). They remove geographical barriers and reduce the cost of meeting room reservations and travel. The non-moderated nature of the tests may also allow real-life user behaviour to be more spontaneous than in a moderated setting. However, the upstream organisation and logistics of this type of test need to be taken into account, as well as the lack of control over the profiles recruited and real-time support, and the possible loss of additional insights.

Finally, to limit incorrect answers in the event of misunderstandings, training examples are included before the test so users can familiarise themselves with the task. However, these examples should not be too long to avoid a negative impact on the test.

5.4.4 Discussion

This experimental protocol is designed to evaluate a local attributive explanation method as extensively as possible, by proposing several analyses of the explanations to users and by focusing on several areas when analysing the results. We also wanted a protocol that could be adapted to several medical specialities and several types of users, so that the tests could be extended in the future. There are also no constraints on the model used or the local attributive explanation method evaluated. We focused on the limitations of the literature described in Chapter 2.3 on the hypotheses, the number of participants and their differences in expertise, and the confounding effects of machine learning and explanations. We have clearly defined a primary hypothesis and secondary hypotheses, also the statistical constraints for validating and rejecting the hypotheses. We hope to increase the chances of recruiting users by splitting the test into several short phases and offering this test to professionals with different levels of expertise in the task. We compensate for confusion between the effects of model predictions and explanations by temporally separating the presentation of each mode for each patient and by presenting each mode in a random order.

This experiment design could also be extended to include more local attributive explanation methods and make hypotheses comparing the effect of different explanation methods. Based on the desired exhaustiveness of the experiment, constraints are to assign patients and explanations to each user so that:

- One patient is seen by multiple users for each explainable method;
- One patient is seen by the same user for each explainable method;
- Each user views multiple patients for each explainable method;

Increasing the number of explanation methods also means guaranteeing a sufficient number of users. It is therefore possible to increase the number of phases, still following the logic of separating the modes for each patient in the different phases, to increase the number of patients seen by each user and guarantee a sufficient volume of results for each method of explanation and patient to be able to validate or invalidate the hypotheses. With these constraints, one could study how users are impacted by the explanations globally and how differences in explanations impact the user for the same patient.

With a high volume of users in the experiment, the impact of the model accuracy could also be estimated by providing two different model accuracies to the users in Mode 2. We could then see if the model accuracy and the user trust in the model impact the confidence in the explanations and how users behave with explanations. This step could also help to isolate the effect of the predictions and the effect of the explanations by comparing the results between the two models.

Due to a lack of time, this experimental protocol has not yet been implemented. We hope to be able to recruit users and carry out these user tests in the coming months.

5.5 Conclusion

This chapter details the work on explanations for medical users, both in terms of the user interface and how the explanations can be used and analysed from different perspectives - direct medical application or exploratory data analysis - and around user experiences to

validate the reliability of our approach combining local and global explanations and their analysis.

We have successfully implemented explanations in a medical application for healthcare professionals based on previously produced mock-ups. We used a question-based approach, based on the questions that users want to answer via the explanations. We designed an interface offering a range of graphical and textual information to ensure that the information is conveyed regardless of the user's viewing preferences. Our intuitive approach also allows users to analyse the explanations provided themselves and combine them with their medical knowledge to gain a critical view. This autonomy encourages users to adhere to and use the explanations.

To reinforce the discussions around explanations for users, we have also proposed an analysis of explanations under two different axes for the same dataset, to show that the use of explanations can pursue several aims, which should not be overlooked when considering the presentation of explanations to users. Using a dataset on urinary tract infections, we have constructed a risk stratification study based on the explanations, which can be used to personalise care for each profile of patients. Secondly, we used exploratory data analysis techniques on the explanations to find new information on the patients in the dataset and analyse the links between the data, the modelling and the disease.

Finally, based on the limitations of the literature on user testing in the field of explanations, we propose a detailed experimental protocol to evaluate the impact of local explanations for medical professionals. We define specific research hypotheses, evaluated according to several focuses - patient, user, users' level of expertise, users' prediction accuracy without explanations - and with objective and subjective metrics - decision speed, decision accuracy and confidence, feelings and usefulness of each variable. We have set up a method for analysing the results for each metric, based on statistical tests, and we have provided information on the various biases and limitations expected and taken into account during the creation of the experimental protocol. We have also considered possible future extensions to our protocol and hope to be able to start conducting tests with the protocol presented in the coming months.

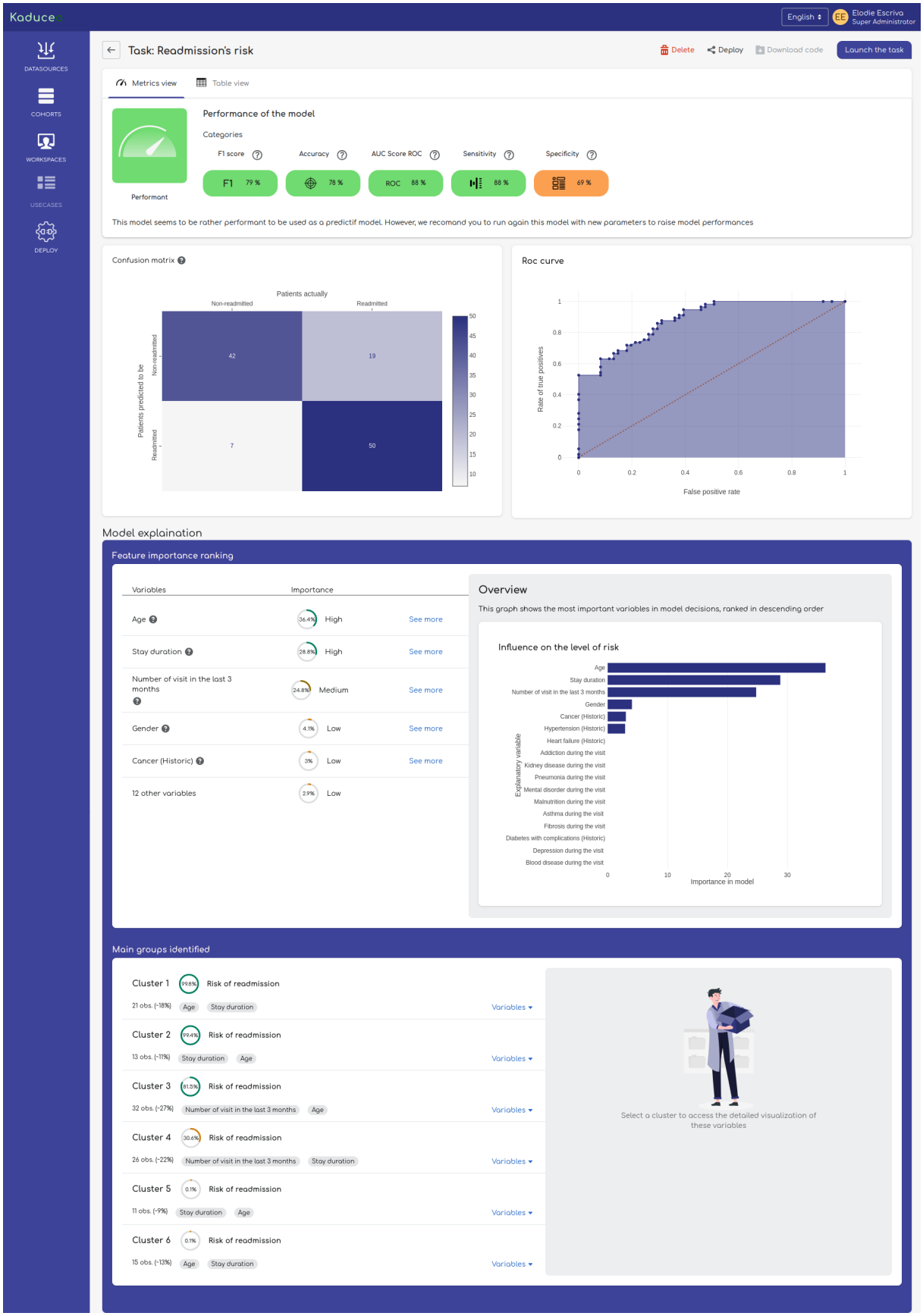
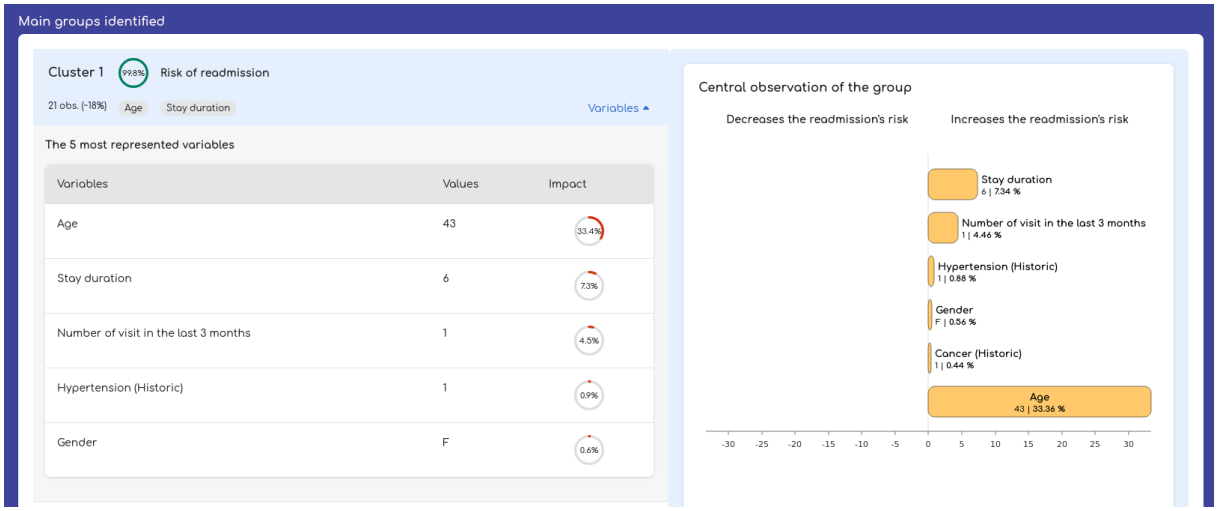
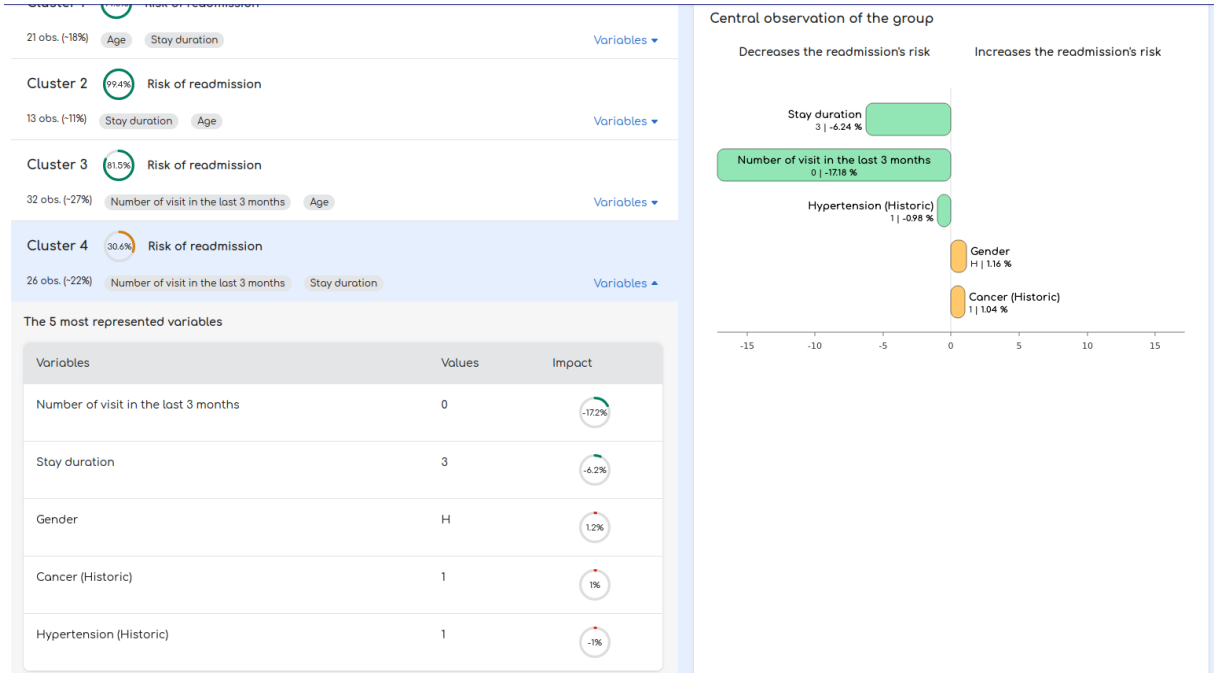


Figure 5.3: User Interface for model performances and global explanations. This scenario predicts the readmission risks for patients.

Figure 5.4: User Interface for global explanations with focus on the "Age" Attribute.



(a) Cluster 1 focus.



(b) Cluster 4 focus.

Figure 5.5: User Interface for the patients' clusters in global explanations, with focus on clusters 1 and 4.

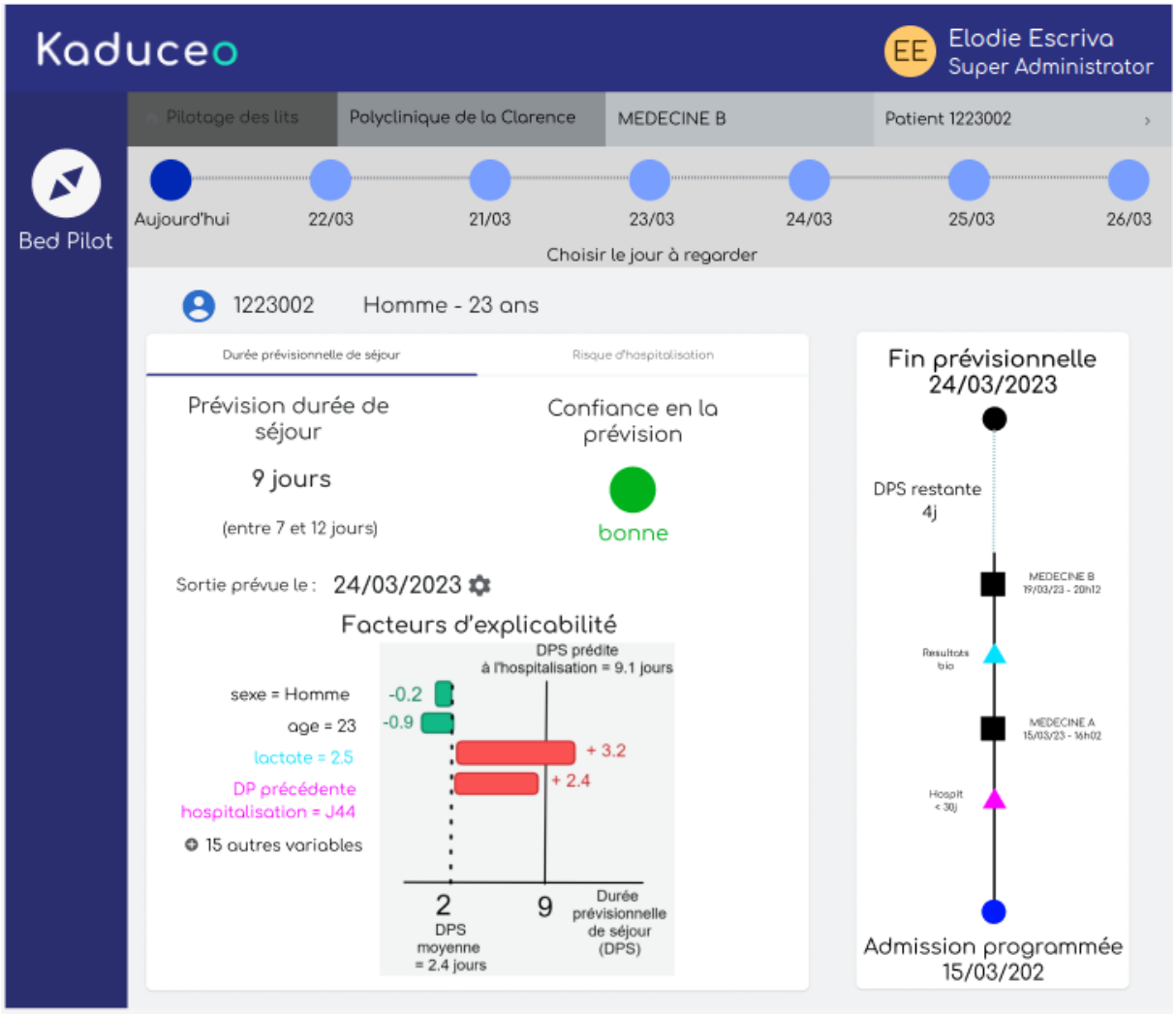


Figure 5.6: User Interface for local explanations. This scenario predicts the hospitalisation stay duration for patients.

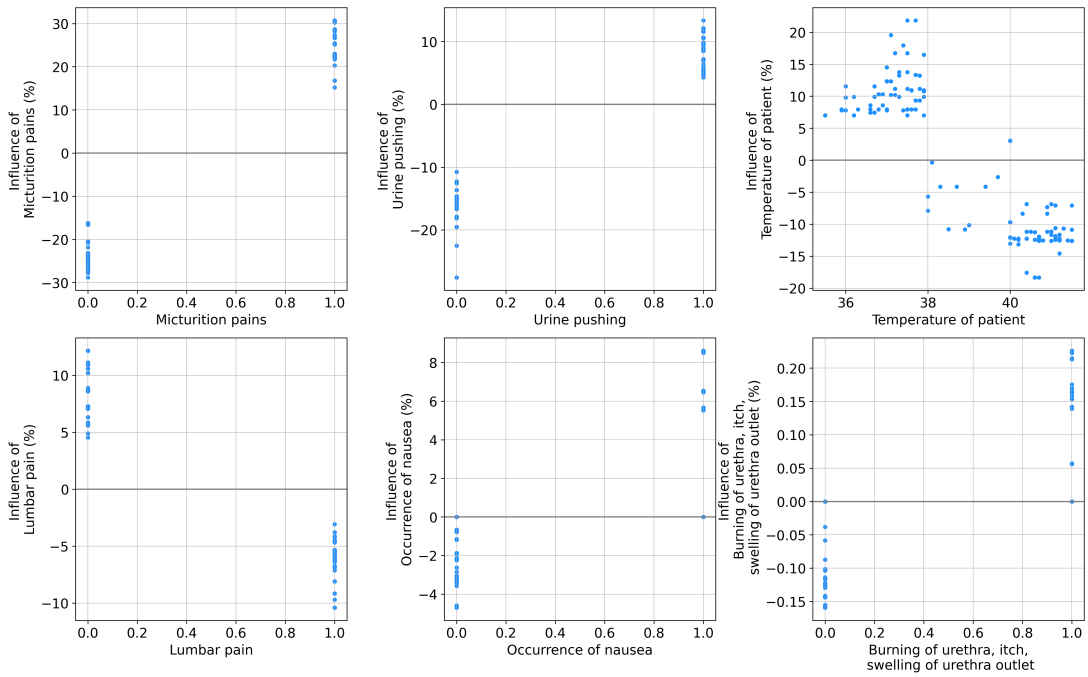


Figure 5.7: Univariate view of each attribute's effect.

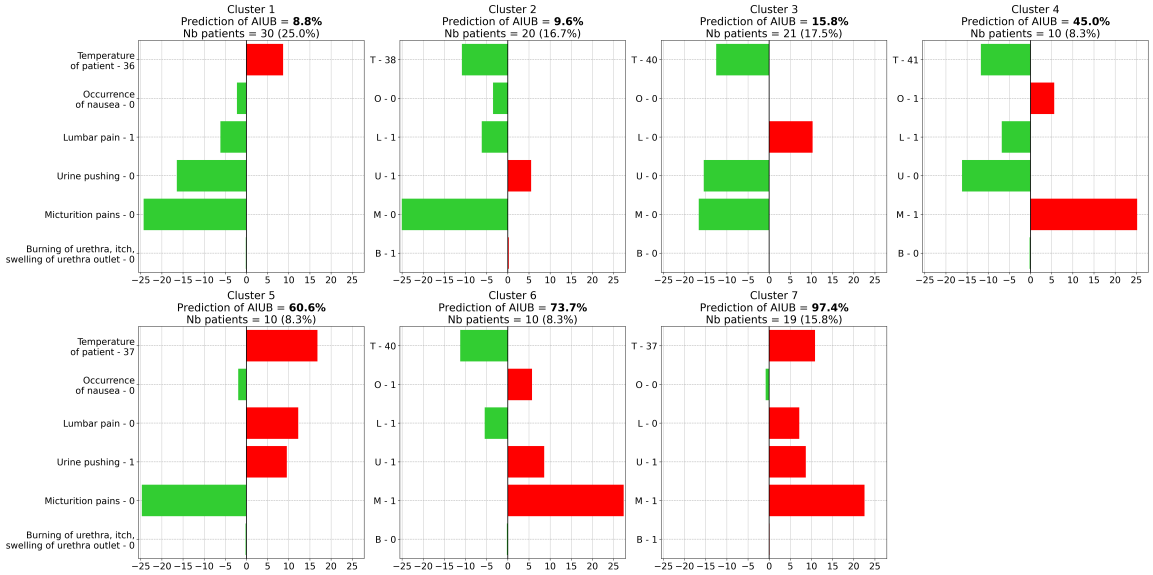


Figure 5.8: Influences of patients corresponding to the medoids of the three identified clusters.

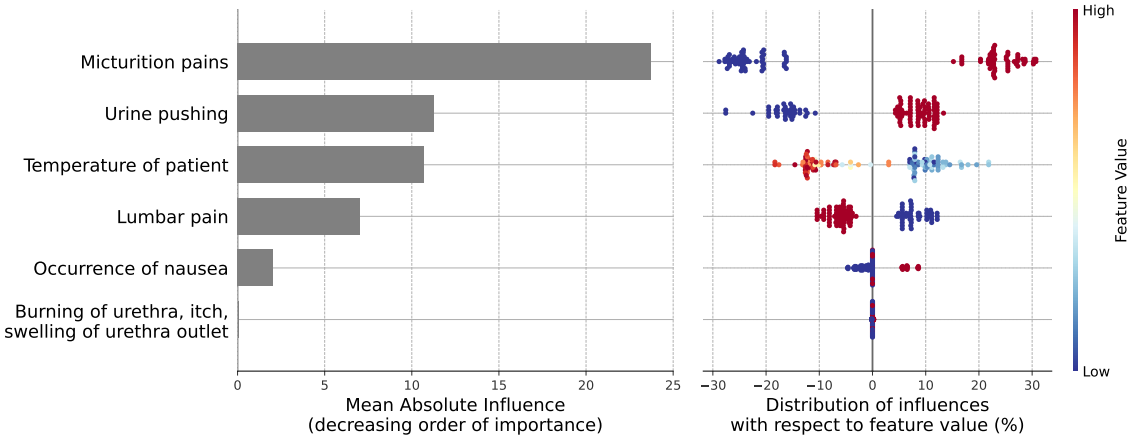


Figure 5.9: *SHAP* mean absolute influences and Distribution of influences for the trained modelling.

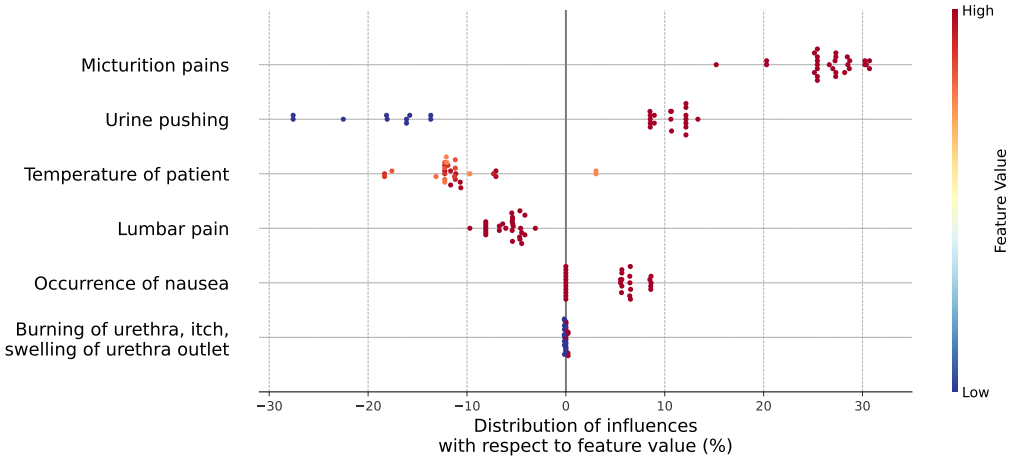


Figure 5.10: Distribution of *SHAP* influences for patients with nausea.

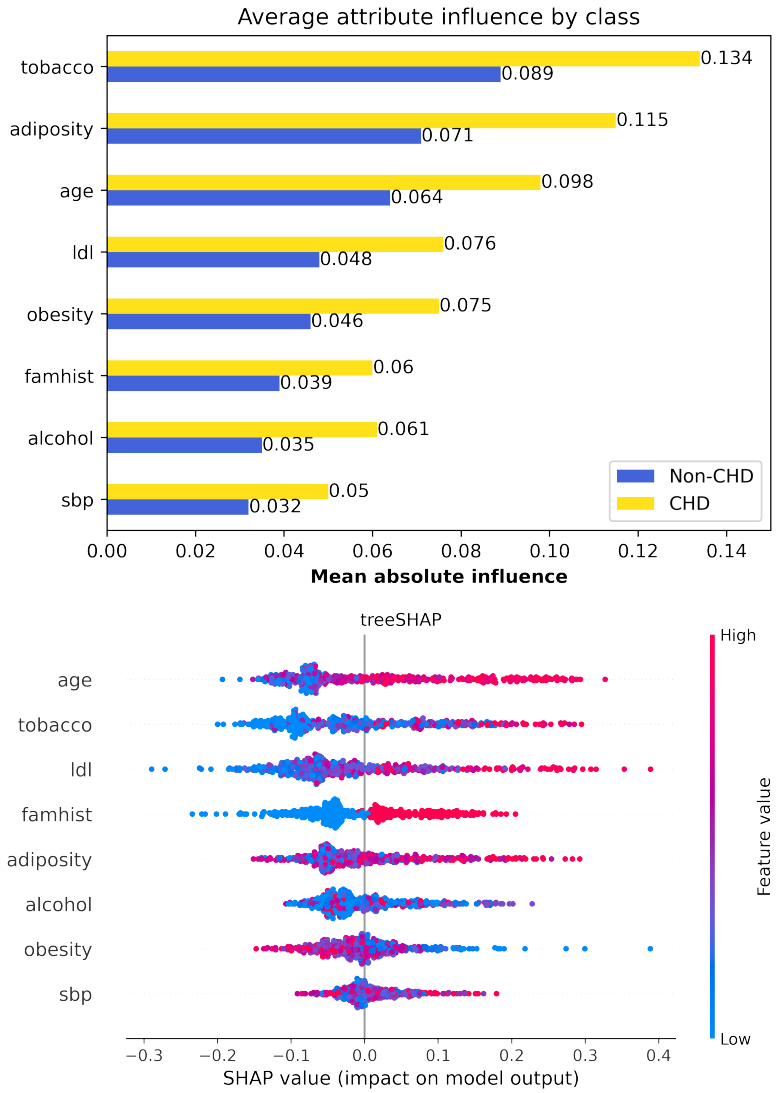


Figure 5.11: Global explanation for the SA-Heart dataset, based on the *TreeSHAP* local explanations. *(top)* Average influence per class. *(bottom)* Beeswarm plot, coloured by feature values, with each point being a patient.

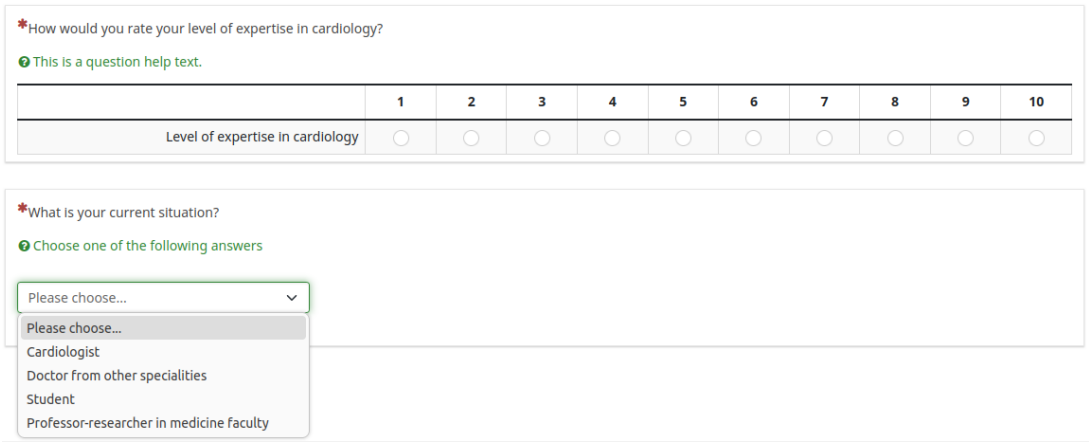


Figure 5.12: Queries for assessing the expertise levels of users.

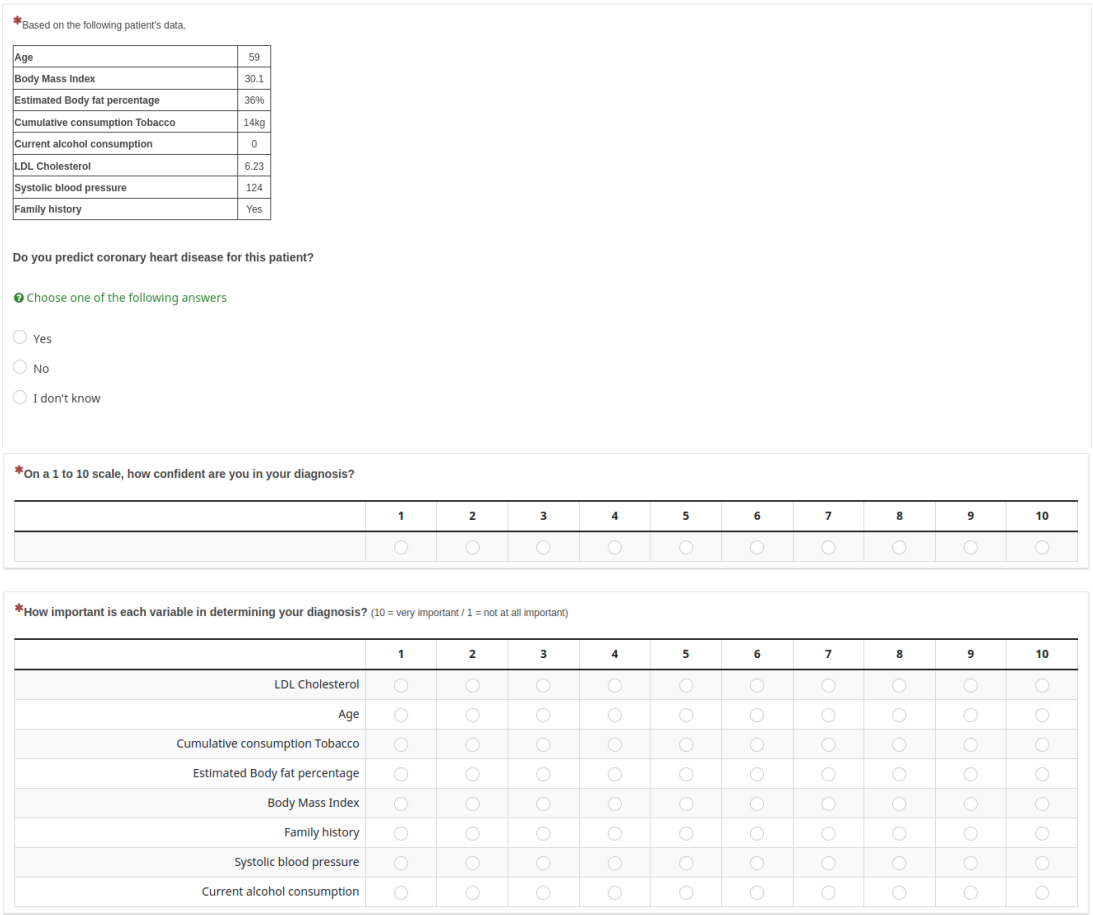
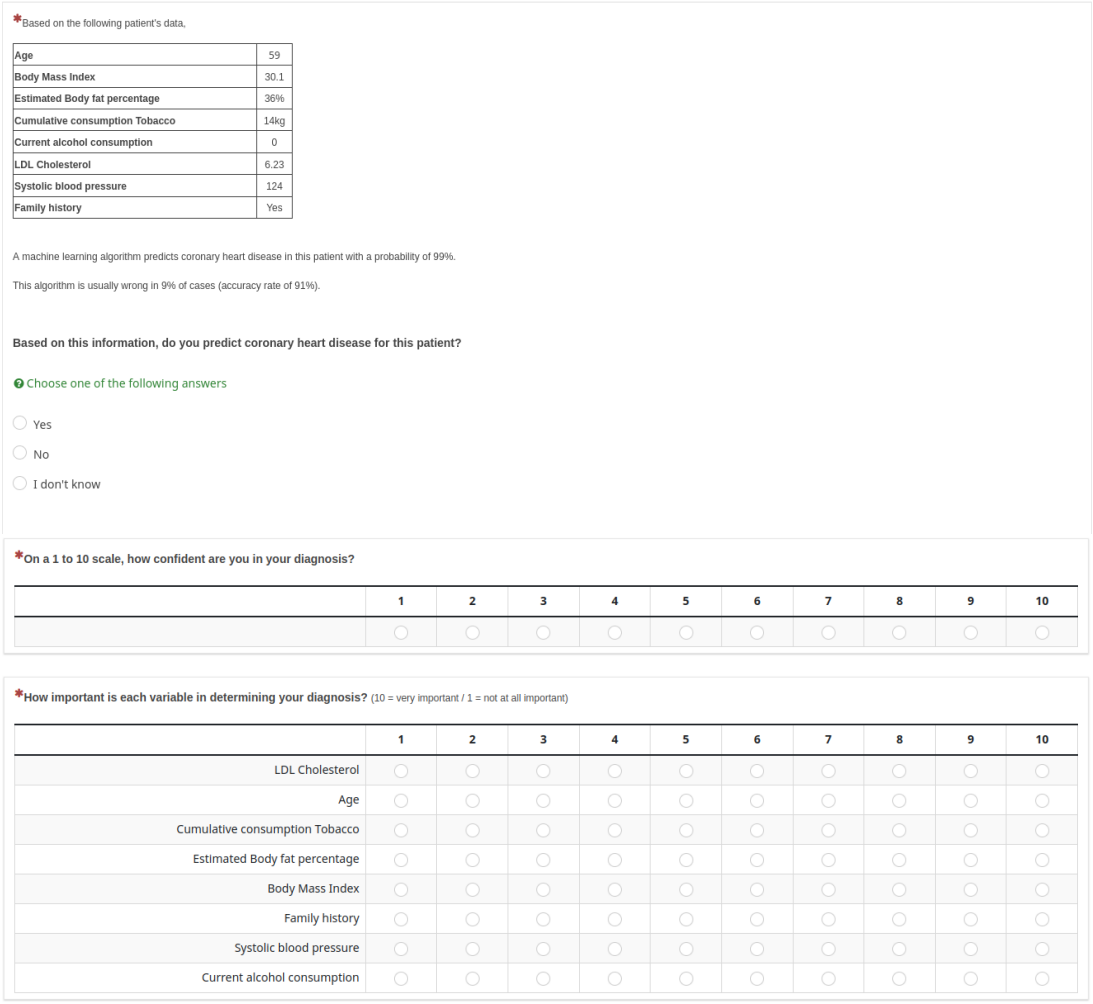


Figure 5.13: User interface for the Mode 1.



*Based on the following patient's data,

Age	59
Body Mass Index	30.1
Estimated Body fat percentage	36%
Cumulative consumption Tobacco	14kg
Current alcohol consumption	0
LDL Cholesterol	6.23
Systolic blood pressure	124
Family history	Yes

A machine learning algorithm predicts coronary heart disease in this patient with a probability of 99%.

This algorithm is usually wrong in 9% of cases (accuracy rate of 91%).

Based on this information, do you predict coronary heart disease for this patient?

Choose one of the following answers

- Yes
- No
- I don't know

***On a 1 to 10 scale, how confident are you in your diagnosis?**

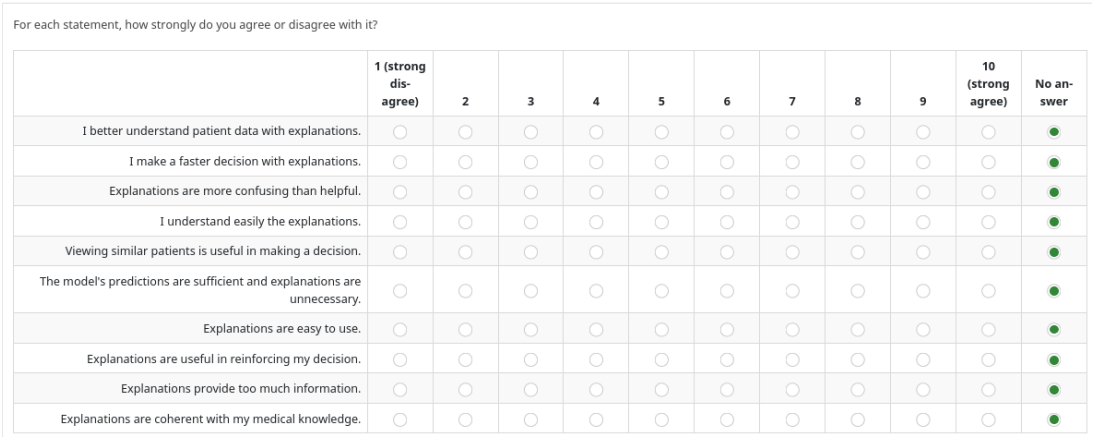
	1	2	3	4	5	6	7	8	9	10

***How important is each variable in determining your diagnosis?** (10 = very important / 1 = not at all important)

	1	2	3	4	5	6	7	8	9	10
LDL Cholesterol										
Age										
Cumulative consumption Tobacco										
Estimated Body fat percentage										
Body Mass Index										
Family history										
Systolic blood pressure										
Current alcohol consumption										

Figure 5.14: User interface for the Mode 2.

Figure 5.15: User interface for the Mode 3.



For each statement, how strongly do you agree or disagree with it?

	1 (strong dis-agree)	2	3	4	5	6	7	8	9	10 (strong agree)	No an-swer
I better understand patient data with explanations.	○	○	○	○	○	○	○	○	○	○	●
I make a faster decision with explanations.	○	○	○	○	○	○	○	○	○	○	●
Explanations are more confusing than helpful.	○	○	○	○	○	○	○	○	○	○	●
I understand easily the explanations.	○	○	○	○	○	○	○	○	○	○	●
Viewing similar patients is useful in making a decision.	○	○	○	○	○	○	○	○	○	○	●
The model's predictions are sufficient and explanations are unnecessary.	○	○	○	○	○	○	○	○	○	○	●
Explanations are easy to use.	○	○	○	○	○	○	○	○	○	○	●
Explanations are useful in reinforcing my decision.	○	○	○	○	○	○	○	○	○	○	●
Explanations provide too much information.	○	○	○	○	○	○	○	○	○	○	●
Explanations are coherent with my medical knowledge.	○	○	○	○	○	○	○	○	○	○	●

Figure 5.16: User interface for the evaluation of the ten assertions.

Chapter 6

Conclusion & Perspectives

6.1 Conclusion

The work in this thesis focuses on the explainability of Machine Learning model predictions, and their possible uses, particularly for end-users in the medical field. Explanation methods have grown quickly over the last few years and since the start of my thesis, with increasing popularity, visible in the number of methods, articles, software currently available and start-ups offering ML model explainability services. This growing interest, and the importance of explainability, are easily understandable given the scale of current applications of Machine Learning and its closed-box characteristic for many models. The closed-box characteristic of many Machine Learning models is a crucial point in the ML research field, and its application in sensitive areas. Today, from my point of view and supported by French and European regulations[1], it seems to me ethically questionable not to provide explanations for the predictions of ML models in sensitive areas and/or areas involving human or animal life. In these fields, local explanation methods, and more generally post-hoc methods, are backup alternatives for explaining complex, closed-box ML models. Back-up as the methods in the literature are still based on restrictive hypotheses -local linearity and attribute independence caused by data perturbations- which are problematic in many complex and sensitive domains such as medicine. The proposed optimised version of the *Coalitional* method is therefore a good candidate for resolving these limitations. In the course of this thesis, we have shown the usefulness of this method and other existing attributive local methods for explaining predictions on numerous datasets from different contexts and with various metrics. Unfortunately, limitations such as the application to large datasets or the large-scale validation of explanations by users persist in the current literature.

A good practice would therefore be, whatever the domain and when the data allow it, to first relate to intrinsically interpretable models to certify that they are not sufficient for the problem being addressed. In the opposite case, a closed-box model approach and an explanation of this model by a combination of global and local explanatory approaches is a potential solution. The existence of methods for explaining closed-box models should not be a justification for using exclusively and automatically increasingly complex closed-box ML models, where there is a risk of oversimplification and error in the explanations.

[1]European General Data Protection Regulation, European AI Act, Report *"The impact of artificial intelligence on the doctor-patient relationship"*, Recommendation *"Artificial intelligence in health care: medical, legal and ethical challenges ahead"*, UNESCO *"Recommendation on the Ethics of Artificial Intelligence"*

Especially if an intrinsically interpretable model such as a tree or decision rules would have been sufficient, effective and interpretable models for modelling the data.

In this thesis, instead of developing new methods, we deliberately focused on optimising and using existing methods rather than designing new ones. We work on the *Coalitional* methods, on understanding the behaviour of these existing methods and on how we can use explanations and make the most of them to efficiently deliver them to end-users. Focusing on end-users is a major aspect of the future development of explanations, to bring these new data face to face with the real world. To this end, we have shown that exploratory analysis of explanations via clustering is a relevant approach, whatever the performance of the model, the instance classification or the local attributive method used. This new data space offers possibilities to efficiently provide end-users with explanations and explanation analyses to have the most positive impact possible. The question-based approach and the analysis of explanations have been our basis for integrating explanations in medical applications and proposing user interfaces adapted to the medical users' constraints of time, non-expertise in Machine Learning models and the human lives potentially at stake. We have also proposed a full experimental protocol to evaluate the relevance of the approach via explanation analysis, with end-users. Our idea is that end-users can provide extensive feedback on the use, usability, reliability, comprehensibility and actionability of explanations and their analyses. It is therefore interesting to have this twofold approach to explainability and end-users: generic and rigorous tests in laboratories, to consolidate and enrich existing methods and large-scale applications of explanations on various use cases with users in the field to observe the real-world behaviour of the approaches. Links between these two approaches would then be essential to enrich them, for example via a user feedback loop to understand the problems in the field on the one hand and the improvement of explanations and how they are provided to users on the other hand.

Finally, this thesis offers a complete framework for explaining predictions, aimed at healthcare workers, i.e. the end users, from the raw data to the user interface. Figure 6.1 displays the entire framework, with the proposals of this thesis. We built on the well-known ML pipeline (step 1 in Figure 6.1) and the existing work on explainability to better explain individual predictions (step 2). We add new steps to explore these explanations to better understand them (step 3), and provide them to end-users (step 4). Multiple steps of this new framework have already been validated by experimentation both on a large collection of general open datasets and on specific medical datasets close to the real-world environment.

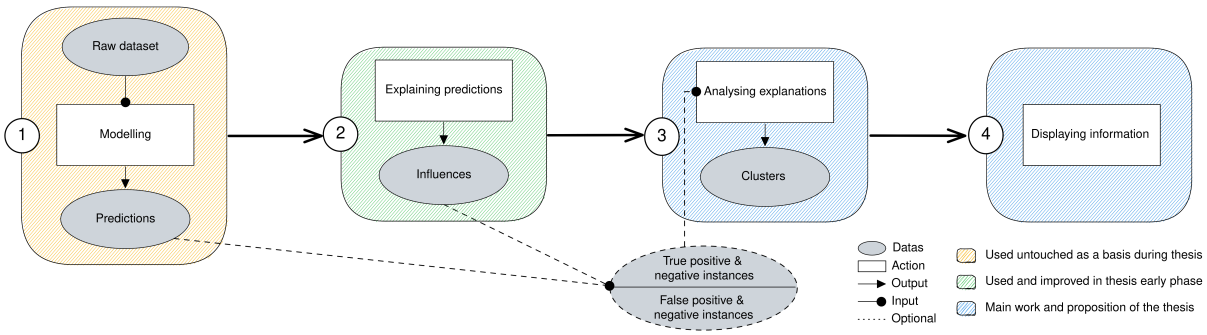


Figure 6.1: Complete XML Framework for local explanations aimed at end-users.

6.2 Perspectives

This thesis consists of several contributions to the research field of explaining the predictions of Machine Learning models. However, research directions are still open to enrich our work and research.

From a short-term perspective, carrying out user tests according to our experimental protocol is a central point of our research. The results will form the basis of our future theoretical work on the attributive local explanation methods, the analysis of the explanation space and the improvement of our user interfaces. In addition, to further validate the explanations, we want to explore the link between the performance of ML models and the quality of the explanations. This work would involve exploring a performance threshold below which local explanations would no longer be sufficiently reliable and robust, in addition to the performance constraints on the model that may be applied by the application domains. We want to apply metrics currently used to compare explanations between multiple XML methods on explanations built when the performances of one ML model are degraded.

In explanation exploration, an important focus of our mid-term research will be the search for links and new information in the data via explanations. We want to pursue our work on explanation clustering, in particular by analysing and characterising the clusters formed and looking for biases in the model and the data. This research would also aim to gain a better understanding of the explanations, predictions, modelling and initial data. Clustering based on influences may then help to understand *why* the model is wrong and not just *where* the model is wrong, and allow for detecting bias in the data.

Through these axes of research, our long-term objective is dual.
Firstly, as already introduced in Cugny et al. (2023), we want to go further than just explaining predictions and integrate explanations into a complete data analysis framework using ML and XML methods. In this way, we could imagine an efficient involvement of explanations at all stages of the ML analysis. Explanations could be used upstream of modelling for pre-processing as in Man and Chan (2021) and during modelling to select a model and hyper-parameters as in Garouani et al. (2022). As we already proposed, explanations can serve the analysis and understanding of predictions, the search for information and biases in the model and the data. Difficulties then lie in how to combine all these researches in one ML analysis pipeline, explainable from start to end.
Secondly, we want to focus on integrating user feedback on explanations. This will enable us to identify inconsistencies between the model and users' domain knowledge and to focus the need for explanations on the end-users. This area would involve collecting feedback, processing it and carrying out research to improve the data and modelling based on this feedback. These two axes could also be linked to make the most of explanations and end-users knowledge. Finally, our approach could be applied in fields such as aeronautics (e.g. analysis and prediction of incidents on aircraft for maintenance purposes) or agronomy (e.g. risks of disease propagation on olive trees or seeds) with our research team's industrial partners.

Bibliography

A. Ali, S. Abd Razak, S. H. Othman, T. A. E. Eisa, A. Al-Dhaqm, M. Nasser, T. Elhassan, H. Elshafie, and A. Saif. Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Applied Sciences*, 12(19):9637, Sept. 2022. ISSN 2076-3417. doi: 10.3390/app12199637. URL `https://www.mdpi.com/2076-3417/12/19/9637`.

A. Alkhatib, H. Boström, and M. Vazirgiannis. Explaining Predictions by Characteristic Rules. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 389–403, Cham, 2023. Springer International Publishing. ISBN 978-3-031-26387-3. doi: 10.1007/978-3-031-26387-3_24.

D. Alvarez-Melis and T. S. Jaakkola. On the Robustness of Interpretability Methods. In *Proceedings of the 2018 ICML Workshop on Human Interpretability in Machine Learning*, Stockholm, Sweden, 2018. arXiv. doi: 10.48550/arXiv.1806.08049. URL `http://arxiv.org/abs/1806.08049`.

A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, and C. Mooney. Current Challenges and Future Opportunities for XAI in Machine Learning-Based Clinical Decision Support Systems: A Systematic Review. *Applied Sciences*, 11(11):5088, 2021. ISSN 2076-3417. doi: 10.3390/app11115088. URL `https://www.mdpi.com/2076-3417/11/11/5088`.

P. Antunes, V. Herskovic, S. F. Ochoa, and J. A. Pino. Structuring dimensions for collaborative systems evaluation. *ACM Comput. Surv.*, 44(2), 2008. ISSN 0360-0300. doi: 10.1145/2089125.2089128. URL `https://doi.org/10.1145/2089125.2089128`.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7), 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0130140.

H. Baniecki, D. Parzych, and P. Biecek. The grammar of interactive explanatory model analysis. *Data Mining and Knowledge Discovery*, 2023. ISSN 1573-756X. doi: 10.1007/s10618-023-00924-w. URL `https://doi.org/10.1007/s10618-023-00924-w`.

N. Barda, D. Riesel, A. Akriv, J. Levy, U. Finkel, G. Yona, D. Greenfeld, S. Sheiba, J. Somer, E. Bachmat, G. N. Rothblum, U. Shalit, D. Netzer, R. Balicer, and N. Dagan. Developing a COVID-19 mortality risk prediction model when individual-level data are not available. *Nature Communications*, 11(1):4439, 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18297-9. URL `https://www.nature.com/articles/s41467-020-18297-9`.

K. Batko and A. Ślęzak. The use of Big Data Analytics in healthcare. *Journal of Big Data*, 9(1):3, 2022. ISSN 2196-1115. doi: 10.1186/s40537-021-00553-4. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8733917/`.

J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, USA, 1981. ISBN 0306406713.

U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. F. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, page 648–657, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3375624. URL https://doi.org/10.1145/3351095.3375624.

M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*, page 11. ACM, 2004. doi: 10.1145/1015330.1015360.

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

L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, New York, 1984. ISBN 978-1-315-13947-0. doi: 10.1201/9781315139470.

D. A. Broniatowski. Psychological Foundations of Explainability and Interpretability in Artificial Intelligence. *NIST*, 2021. URL https://www.nist.gov/publications/psychological-foundations-explainability-and-interpretability-artificial-intelligence.

N. Burkart and M. F. Huber. A Survey on the Explainability of Supervised Machine Learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021. ISSN 1076-9757. doi: 10.1613/jair.1.12228. URL https://www.jair.org/index.php/jair/article/view/12228.

Z. Carmichael and W. J. Scheirer. A framework for evaluating post hoc feature-additive explainers. *arXiv preprint arXiv:2106.08376*, 2021. doi: 10.48550/ARXIV.2106.08376. URL https://arxiv.org/abs/2106.08376.

T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, San Francisco California USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL https://dl.acm.org/doi/10.1145/2939672.2939785.

M. Chromik and M. Schuessler. A Taxonomy for Human Subject Evaluation of Black-Box Explanations in XAI. In *Proceedings of the IUI 2020 Workshop*, volume 2582 of *CEUR Workshop Proceedings*, Cagliari, Italy, 2020. CEUR-WS.org. URL https://ceur-ws.org/Vol-2582/paper9.pdf.

J. Conrad, K. Al-Kofahi, Y. Zhao, and G. Karypis. Effective Document Clustering for Large Heterogeneous Law Firm Collections. In *Proceedings of the 10th international conference on Artificial intelligence and law*, ICAIL '05, pages 177–187, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930817. doi: 10.1145/1165485.1165513.

A. Cooper, O. Doyle, and A. Bourke. Supervised Clustering for Subgroup Discovery: An Application to COVID-19 Symptomatology. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Communications in Computer and Information Science, pages 408–422, Cham, 2021. Springer International Publishing. ISBN 978-3-030-93733-1. doi: 10.1007/978-3-030-93733-1_29.

R. Cugny, E. Doumard, E. Escriva, and H. Wang. L'explicabilité au service de l'extraction de connaissances : application à des données médicales. 2023. doi: 10.48550/ARXIV.2302.02653. URL `https://hal.science/hal-03978252`.

J. Czerniak and H. Zarzycki. Application of rough sets in the presumptive diagnosis of urinary system diseases. In *Artificial Intelligence and Security in Computing Systems*, The Springer International Series in Engineering and Computer Science, pages 41–51, Boston, MA, 2003. Springer US. ISBN 978-1-4419-9226-0. doi: 10.1007/978-1-4419-9226-0_5.

P. Dalgaard. *Power and the computation of sample size*, pages 155–162. Springer New York, New York, NY, 2008. ISBN 978-0-387-79054-1. doi: 10.1007/978-0-387-79054-1_9. URL `https://doi.org/10.1007/978-0-387-79054-1_9`.

A. Datta, S. Sen, and Y. Zick. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 598–617, 2016. doi: 10.1109/SP.2016.42.

F. Daudt, D. Cinalli, and A. C. B. Garcia. Research on Explainable Artificial Intelligence Techniques: An User Perspective. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2021. doi: 10.1109/CSCWD49262.2021.9437820.

D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x`.

J. D. Dera. Risk stratification: A two-step process for identifying your sickest patients. *Family practice management*, 26(3):21–26, 2019.

W. K. Diprose, N. Buist, N. Hua, Q. Thurier, G. Shand, and R. Robinson. Physician understanding, explainability, and trust in a hypothetical machine learning risk calculator. *Journal of the American Medical Informatics Association: JAMIA*, 27(4):592–600, 2020. ISSN 1527-974X. doi: 10.1093/jamia/ocz229.

F. Doshi-Velez and B. Kim. *Considerations for Evaluation and Generalization in Interpretable Machine Learning*, pages 3–17. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98131-4. doi: 10.1007/978-3-319-98131-4_1. URL `https://doi.org/10.1007/978-3-319-98131-4_1`.

E. Doumard, J. Aligon, E. Escriva, J.-B. Excoffier, P. Monsarrat, and C. Soulé-Dupuy. A comparative study of additive local explanation methods based on feature influences. In *Proceedings of the EDBT Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP)*, volume 3130, page 31. CEUR-WS.org, 2022. URL `https://hal.science/hal-03687554`.

E. Doumard, J. Aligon, E. Escriva, J.-B. Excoffier, P. Monsarrat, and C. Soulé-Dupuy. A quantitative approach for the comparison of additive local explanation methods. *Information Systems*, 114:102162, 2023. ISSN 0306-4379. doi: 10.1016/j.is.2022.102162.

J. Duell, X. Fan, B. Burnett, G. Aarts, and S.-M. Zhou. A Comparison of Explanations Given by Explainable Artificial Intelligence Methods on Analysing Electronic Health Records. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, 2021. doi: 10.1109/BHI50953.2021.9508618.

A. Dujon, J. Geeson, J. Arnould, B. Allan, K. Katselidis, and G. Schofield. Machine learning to detect marine animals in UAV imagery: effect of morphology, spacing, behaviour and habitat. *Remote Sensing in Ecology and Conservation*, 7, May 2021. doi: 10.1002/rse2.205.

R. El-Bouri, T. Taylor, A. Youssef, T. Zhu, and D. A. Clifton. Machine learning in patient flow: a review. *Progress in Biomedical Engineering (Bristol, England)*, 3(2): 022002, Apr. 2021. ISSN 2516-1091. doi: 10.1088/2516-1091/abddc5. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8559147/`.

R. El Shawi, Y. Sherif, M. Al-Mallah, and S. Sakr. Interpretability in HealthCare A Comparative Study of Local Machine Learning Interpretability Techniques. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 275–280, 2019. doi: 10.1109/CBMS.2019.00065.

E. Escriva, G. Ferrettini, J. Aligon, J.-B. Excoffier, and C. Soulé-Dupuy. Stratégies coalitionnelles pour une explication efficace des prédictions individuelles. In *Conférence francophone sur l'Extraction et la Gestion des Connaissances (EGC 2022)*, page 395. RNTI : Revue des Nouvelles Technologies de l'Information, 2022. URL `https://hal.science/hal-03845757`.

E. Escriva, J. Aligon, J.-B. Excoffier, P. Monsarrat, and C. Soulé-Dupuy. How to Make the Most of Local Explanations: Effective Clustering Based on Influences. In *Advances in Databases and Information Systems*, volume 13985, pages 146–160. Springer Nature Switzerland, Cham, 2023a. ISBN 978-3-031-42913-2 978-3-031-42914-9. doi: 10.1007/978-3-031-42914-9_11.

E. Escriva, E. Doumard, J.-B. Excoffier, J. Aligon, P. Monsarrat, and C. Soulé-Dupuy. Data Exploration Based on Local Attribution Explanation: A Medical Use Case. In *New Trends in Database and Information Systems*, volume 1850, pages 315–323. Springer Nature Switzerland, Cham, 2023b. ISBN 978-3-031-42940-8 978-3-031-42941-5. doi: 10.1007/978-3-031-42941-5_27.

M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery,and Data Mining (KDD-96), Portland, Oregon,*

USA, pages 226–231. AAAI Press, 1996. URL http://www.aaai.org/Library/KDD/1996/kdd96-037.php.

J.-B. Excoffier, E. Escriva, J. Aligon, and M. Ortala. Local Explanation-Based Method for Healthcare Risk Stratification. In *Medical Informatics Europe 2022. Challenges of Trustable AI and Added-Value on Health*, pages 555–556. IOS Press, 2022a. doi: 10.3233/SHTI220520.

J.-B. Excoffier, N. Salaün-Penquer, M. Ortala, M. Raphaël-Rousseau, C. Chouaid, and C. Jung. Analysis of COVID-19 inpatients in France during first lockdown of 2020 using explainability methods. *Medical & Biological Engineering & Computing*, 60(6): 1647–1658, 2022b. ISSN 1741-0444. doi: 10.1007/s11517-022-02540-0. URL https://doi.org/10.1007/s11517-022-02540-0.

G. Ferrettini, J. Aligon, and C. Soulé-Dupuy. Explaining Single Predictions: A Faster Method. In *SOFSEM 2020: Theory and Practice of Computer Science*, Lecture Notes in Computer Science, pages 313–324, Cham, 2020a. Springer International Publishing. ISBN 978-3-030-38919-2. doi: 10.1007/978-3-030-38919-2_26.

G. Ferrettini, J. Aligon, and C. Soulé-Dupuy. Improving on Coalitional Prediction Explanation. In *Advances in Databases and Information Systems*, Lecture Notes in Computer Science, pages 122–135, Cham, 2020b. Springer International Publishing. ISBN 978-3-030-54832-2. doi: 10.1007/978-3-030-54832-2_11.

G. Ferrettini, E. Escriva, J. Aligon, J.-B. Excoffier, and C. Soulé-Dupuy. Coalitional Strategies for Efficient Individual Prediction Explanation. *Information Systems Frontiers*, 2021. ISSN 1572-9419. doi: 10.1007/s10796-021-10141-9.

A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019. URL http://jmlr.org/papers/v20/18-760.html.

M. Flora, C. Potvin, A. McGovern, and S. Handler. Comparing Explanation Methods for Traditional Machine Learning Models Part 1: An Overview of Current Methods and Quantifying Their Disagreement, 2022. URL http://arxiv.org/abs/2211.08943.

M. Francone, F. Iafrate, G. M. Masci, S. Coco, F. Cilia, L. Manganaro, V. Panebianco, C. Andreoli, M. C. Colaiacomo, M. A. Zingaropoli, et al. Chest ct score in covid-19 patients: correlation with disease severity and short-term prognosis. *European radiology*, 30(12):6808–6817, 2020.

J. H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19 (1):1 – 67, 1991. doi: 10.1214/aos/1176347963. URL https://doi.org/10.1214/aos/1176347963.

F. Gardin, R. Gautiern, N. Goix, B. Ndiaye, and J.-M. Schertzer. Skope-rules, 2019. URL https://github.com/scikit-learn-contrib/skope-rules.

M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, and A. Lewandowski. Towards big industrial data mining through explainable automated machine learning. *The International Journal of Advanced Manufacturing Technology*, 120(1):1169–1188, 2022. ISSN 1433-3015. doi: 10.1007/s00170-022-08761-9.

D. Garreau and U. von Luxburg. Explaining the explainer: A first theoretical analysis of lime. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence an Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1287–1296. PMLR, 2020. URL `https://proceedings.mlr.press/v108/garreau20a.html`.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5): 93:1–93:42, 2018. ISSN 0360-0300. doi: 10.1145/3236009. URL `https://dl.acm.org/doi/10.1145/3236009`.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2nd edition, 2009. ISBN 978-0-387-84857-0 978-0-387-84858-7. doi: https://doi.org/10.1007/978-0-387-84858-7.

A. Holzinger, R. Goebel, R. Fong, T. Moon, K. Müller, and W. Samek. xxai - beyond explainable artificial intelligence. In *xxAI - Beyond Explainable AI - International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, pages 3–10, 2020. doi: 10.1007/978-3-031-04083-2_1.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.*, 31(8): 651–666, 2010. doi: 10.1016/J.PATREC.2009.09.011.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.

A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999. doi: 10.1145/331499.331504.

S. Jesus, C. Belém, V. Balayan, J. Bento, P. Saleiro, P. Bizarro, and J. Gama. How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 805–815, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 978-1-4503-8309-7. doi: 10.1145/3442188.3445941.

F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and Vascular Neurology*, 2(4):230–243, 2017. ISSN 2059-8696. doi: 10.1136/svn-2017-000101.

P. Jiang, H. Suzuki, and T. Obi. Interpretable machine learning analysis to identify risk factors for diabetes using the anonymous living census data of japan. *Health and Technology*, 13(1):119–131, 2023. doi: 10.1007/s12553-023-00730-w.

M. S. Kamal, N. Dey, L. Chowdhury, S. I. Hasan, and K. Santosh. Explainable AI for Glaucoma Prediction Analysis to Understand Risk Factors in Treatment Planning. *IEEE Transactions on Instrumentation and Measurement*, 71, 2022. ISSN 1557-9662. doi: 10.1109/TIM.2022.3171613.

L. Kaufman and P. Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis*. John Wiley & Sons, 01 1990. ISBN 0-471-87876-6. doi: 10.2307/2532178.

L. Kaufman and P. J. Rousseeuw. Clustering by Means of medoids. In *Proceedings of the Statistical Data Analysis Based on the L1 Norm Conference*, volume 31, pages 405–416, Neuchatel, Switzerland, 1987. URL https://cir.nii.ac.jp/crid/1571698601146627840.

M. Kendall. *Rank correlation methods*. Rank correlation methods. Griffin, Oxford, England, 1948.

B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5680522b8e2bb01943234bce7bf84534-Abstract.html.

I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler. Problems with shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5491–5500. PMLR, 2020. URL https://proceedings.mlr.press/v119/kumar20e.html.

N. Labroche. Online fuzzy medoid based clustering algorithms. *Neurocomputing*, 126: 141–150, 2014. doi: 10.1016/J.NEUCOM.2012.07.057.

T. Laugel, X. Renard, M.-J. Lesot, C. Marsala, and M. Detyniecki. Defining locality for surrogates in post-hoc interpretablity. *Proceedings of the ICML Workshop on Human Interpretability for Machine Learning (WHI)*, 2018. URL https://hal.sorbonne-universite.fr/hal-01905924.

M. L. Leavitt and A. Morcos. Towards falsifiable interpretability research. *arXiv preprint arXiv:2010.12016*, 2020. URL https://arxiv.org/abs/2010.12016.

K. Lee, M. V. Ayyasamy, Y. Ji, and P. V. Balachandran. A comparison of explainable artificial intelligence methods in the phase classification of multi-principal element alloys. *Scientific Reports*, 12(1):11591, 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-15618-4. URL https://www.nature.com/articles/s41598-022-15618-4.

Q. V. Liao, D. Gruen, and S. Miller. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pages 1–15, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376590. URL https://doi.org/10.1145/3313831.3376590.

P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021. ISSN 1099-4300. doi: 10.3390/e23010018.

S. Lipovetsky and M. Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001. ISSN 1526-4025. doi: 10.1002/asmb.446.

Z. C. Lipton. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018. ISSN 1542-7730. doi: 10.1145/3236386.3241340.

G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, Dec. 2017. ISSN 1361-8415. doi: 10.1016/j.media.2017.07.005. URL `https://www.sciencedirect.com/science/article/pii/S1361841517301135`.

Y. Liu, Z. Liu, X. Luo, and H. Zhao. Diagnosis of Parkinson's disease based on SHAP value feature selection. *Biocybernetics and Biomedical Engineering*, 42(3):856–869, 2022. ISSN 0208-5216. doi: 10.1016/j.bbe.2022.06.007. URL `https://www.sciencedirect.com/science/article/pii/S0208521622000638`.

Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 623–631, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747. doi: 10.1145/2487575.2487579. URL `https://doi.org/10.1145/2487575.2487579`.

S. M. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30 of *NIPS'17*, page 4768–4777. Curran Associates, Inc., 2017. ISBN 9781510860964. URL `https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html`.

S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020. ISSN 2522-5839. doi: 10.1038/s42256-019-0138-9.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics & Probability*, volume 5.1, pages 281–297. University of California Press, 1965.

P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936. doi: 10.1007/s13171-019-00164-5.

S. Makki. *An Efficient Classification Model for Analyzing Skewed Data to Detect Frauds in the Financial Sector*. Theses, Université de Lyon ; Université Libanaise, 2019. URL `https://theses.hal.science/tel-02457134`.

X. Man and E. P. Chan. The Best Way to Select Features? Comparing MDA, LIME, and SHAP. *The Journal of Financial Data Science*, 3(1):127–139, 2021. ISSN 2640-3943. doi: 10.3905/jfds.2020.1.047.

L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference*, pages 33–42. IEEE, 2017. doi: 10.1109/ICDMW.2017.12.

L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017. doi: 10.21105/joss.00205.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. ISSN 0004-3702. doi: 10.1016/j.artint.2018.07.007.

T. M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill, international ed. edition, 1997. ISBN 978-0-07-115467-3 978-0-07-042807-2.

C. Molnar. *Interpretable Machine Learning*. Independently published, 2 edition, 2022a. URL https://christophm.github.io/interpretable-ml-book/.

C. Molnar. *Model-agnostic interpretable machine learning*. Text.PhDThesis, Ludwig-Maximilians-Universität München, 2022b. URL https://edoc.ub.uni-muenchen.de/30374/. ISSN: 1930-3742.

P. Monsarrat, D. Bernard, M. Marty, C. Cecchin-Albertoni, E. Doumard, L. Gez, J. Aligon, J.-N. Vergnes, L. Casteilla, and P. Kemoun. Systemic Periodontal Risk Score Using an Innovative Machine Learning Strategy: An Observational Study. *Journal of Personalized Medicine*, 12(2):217, 2022. ISSN 2075-4426. doi: 10.3390/jpm12020217. URL https://www.mdpi.com/2075-4426/12/2/217.

G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2017.10.011.

S. Morgenthaler. Exploratory data analysis. *WIREs Comp Stats 1*, 2009.

W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. doi: 10.1073/pnas.1900654116.

U. Nagavelli, D. Samanta, and P. Chakraborty. Machine Learning Technology-Based Heart Disease Detection Models. *Journal of Healthcare Engineering*, 2022:7351061, Feb. 2022. ISSN 2040-2295. doi: 10.1155/2022/7351061. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8898839/.

M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *ACM Computing Surveys*, 2023. ISSN 0360-0300. doi: 10.1145/3583558.

A.-p. Nguyen and M. R. Martínez. On quantitative aspects of model interpretability. *arXiv:2007.07584 [cs, stat]*, 2020. URL http://arxiv.org/abs/2007.07584.

G. Norman. Likert scales, levels of measurement and the "laws" of statistics. *Advances in Health Sciences Education*, 15(5):625–632, 2010. ISSN 1573-1677. doi: 10.1007/s10459-010-9222-y. URL https://doi.org/10.1007/s10459-010-9222-y.

I. Nunes and D. Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27 (3), 2017. ISSN 1573-1391. doi: 10.1007/s11257-017-9195-0.

R. Obringer and R. Nateghi. Predicting Urban Reservoir Levels Using Statistical Learning Techniques. *Scientific Reports*, 8(1):5164, Mar. 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-23509-w. URL https://www.nature.com/articles/s41598-018-23509-w. Number: 1 Publisher: Nature Publishing Group.

S. Oh, Y. Park, K. J. Cho, and S. J. Kim. Explainable machine learning model for glaucoma diagnosis and its interpretation. *Diagnostics*, 11(3), 2021. ISSN 2075-4418. doi: 10.3390/diagnostics11030510. URL https://www.mdpi.com/2075-4418/11/3/510.

H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009. doi: https://doi.org/10.1016/j.eswa.2008.01.039. URL https://www.sciencedirect.com/science/article/pii/S095741740800081X.

L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor.*, 6(1):90–105, 2004. doi: 10.1145/1007730.1007731.

E. Petersen, Y. Potdevin, E. Mohammadi, S. Zidowitz, S. Breyer, D. Nowotka, S. Henn, L. Pechmann, M. Leucker, P. Rostalski, and C. Herzog. Responsible and Regulatory Conform Machine Learning for Medicine: A Survey of Challenges and Solutions. *IEEE Access*, 10, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3178382.

P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol. AI in health and medicine. *Nature Medicine*, 28(1):31–38, 2022. ISSN 1546-170X. doi: 10.1038/s41591-021-01614-0. URL https://www.nature.com/articles/s41591-021-01614-0.

R. Ramberg. Construing and testing explanations in a complex domain. *Computers in Human Behavior*, 12(1):29–48, 1996. ISSN 0747-5632. doi: https://doi.org/10.1016/0747-5632(95)00017-8. URL https://www.sciencedirect.com/science/article/pii/0747563295000178.

S. Rao, S. Mehta, S. Kulkarni, H. Dalvi, N. Katre, and M. Narvekar. A Study of LIME and SHAP Model Explainers for Autonomous Disease Predictions. In *2022 IEEE Bombay Section Signature Conference (IBSSC)*, 2022. doi: 10.1109/IBSSC56953.2022.10037324.

M. T. Ribeiro, S. Singh, and C. Guestrin. "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*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL https://arxiv.org/abs/1602.04938.

M. Robnik-Šikonja and M. Bohanec. *Perturbation-Based Explanations of Prediction Models*, pages 159–175. Springer International Publishing, Cham, 2018. ISBN 978-3-319-90403-0. doi: 10.1007/978-3-319-90403-0_9.

L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010. ISSN 1573-7462. doi: 10.1007/s10462-009-9124-7. URL https://doi.org/10.1007/s10462-009-9124-7.

R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020. doi: 10.1109/ACCESS.2020.2976199.

R. L. Ross, B. Sachdeva, J. Wagner, K. Ramsey, and D. A. Dorr. Perceptions of risk stratification workflows in primary care. *Healthcare*, 5(4):78, 2017. ISSN 2227-9032. doi: 10.3390/healthcare5040078.

Rossouw, du Plessis, Benade, Jordaan, Kotze, Jooste, and Ferreira. Coronary risk factor screening in three rural communities-the coris baseline study. *South African medical journal*, 64(12):430–436, 1983.

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL `https://www.sciencedirect.com/science/article/pii/0377042787901257`.

C. Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature machine intelligence*, 1(5):206–215, 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9122117/`.

A. Sangroya, M. Rastogi, C. Anantaram, and L. Vig. Guided-LIME: Structured Sampling based Hybrid Approach towards Explaining Blackbox Machine Learning Models. In *Proceedings of the CIKM 2020 Workshops*, volume 2699 of *CEUR Workshop Proceedings*, Galway, Ireland, 2020. CEUR-WS.org. URL `https://ceur-ws.org/Vol-2699/paper01.pdf`.

P. V. Selvaraj. *Comparative study of synchronous remote and traditional in-lab usability evaluation methods*. PhD thesis, Virginia Polytechnic Institute and State University, 2004.

L. S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi: 10.7249/P0295.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017. URL `https://proceedings.mlr.press/v70/shrikumar17a.html`.

J. A. M. Sidey-Gibbons and C. J. Sidey-Gibbons. Machine learning in medicine: a practical introduction. *BMC Medical Research Methodology*, 19(1):64, 2019. ISSN 1471-2288. doi: 10.1186/s12874-019-0681-4. URL `https://doi.org/10.1186/s12874-019-0681-4`.

D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 180–186, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371100. doi: 10.1145/3375627.3375830. URL `https://doi.org/10.1145/3375627.3375830`.

R. Srinivasan and A. Chander. Explanation perspectives from the cognitive sciences - a survey. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4812–4818. International Joint Conferences on Artificial Intelligence Organization, 2020. doi: 10.24963/ijcai.2020/670. URL `https://doi.org/10.24963/ijcai.2020/670`. Survey track.

G. Van den Broeck, A. Lykov, M. Schleich, and D. Suciu. On the Tractability of SHAP Explanations. *Journal of Artificial Intelligence Research*, 74:851–886, 2022. ISSN 1076-9757. doi: 10.1613/jair.1.13283.

J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014. ISSN 1931-0145. doi: 10.1145/2641190.2641198. URL https://doi.org/10.1145/2641190.2641198.

J. Verhaeghe, J. Van Der Donckt, F. Ongenae, and S. Van Hoecke. Powershap: A Power-Full Shapley Feature Selection Method. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 71–87, Cham, 2023. Springer International Publishing. ISBN 978-3-031-26387-3. doi: 10.1007/978-3-031-26387-3_5.

G. Vilone and L. Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.009.

G. Visani, E. Bagli, and F. Chesani. OptiLIME: Optimized LIME Explanations for Diagnostic Computer Algorithms. *arXiv:2006.05714 [cs, stat]*, 2022. URL http://arxiv.org/abs/2006.05714.

V. Vu, N. Labroche, and B. Bouchon-Meunier. Improving constrained clustering with active query selection. *Pattern Recognit.*, 45(4):1749–1758, 2012. doi: 10.1016/J.PATCOG.2011.10.016.

J. H. Ward Jr and M. E. Hook. Application of an hierarchical grouping procedure to a problem of grouping profiles. *Educational and Psychological Measurement*, 23(1):69–81, 1963.

H. J. P. Weerts, W. van Ipenburg, and M. Pechenizkiy. A Human-Grounded Evaluation of SHAP for Alert Processing. In *Proceedings of 2019 SIG-KDD Workshop on Explainable AI (KDD-XAI)*. arXiv, 2019. doi: 10.48550/arXiv.1907.03324. URL http://arxiv.org/abs/1907.03324.

A. Wirsch. *Analysis of a top-down bottom-up data analysis framework and software architecture design.* Thesis, Massachusetts Institute of Technology, 2014. URL https://dspace.mit.edu/handle/1721.1/107346.

X. Xiaomao, Z. Xudong, and W. Yuanfang. A Comparison of Feature Selection Methodology for Solving Classification Problems in Finance. *Journal of Physics: Conference Series*, 1284(1):012026, 2019. ISSN 1742-6596. doi: 10.1088/1742-6596/1284/1/012026. URL https://dx.doi.org/10.1088/1742-6596/1284/1/012026.

J. Xie, R. B. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 478–487. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/xieb16.html.

C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar. On the (In)fidelity and Sensitivity of Explanations. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a7471fdc77b3435276507cc8f2dc2569-Abstract.html.

Y. Zhang, Q. V. Liao, and R. K. E. Bellamy. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 295–305, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372852. URL https://dl.acm.org/doi/10.1145/3351095.3372852.

Z. Zheng, F. Peng, B. Xu, J. Zhao, H. Liu, J. Peng, Q. Li, C. Jiang, Y. Zhou, S. Liu, et al. Risk factors of critical & mortal covid-19 cases: A systematic literature review and meta-analysis. *Journal of Infection*, 2020.

J. Zhou, Z. Wei, B. Peng, and F. Chi. Research and Application of Film and Television Literature Recommendation Based on Secure Internet of Things and Machine Learning. *Mobile Information Systems*, 2021:e4066267, Oct. 2021. ISSN 1574-017X. doi: 10.1155/2021/4066267. URL https://www.hindawi.com/journals/misy/2021/4066267/. Publisher: Hindawi.

E. Štrumbelj and I. Kononenko. Towards a model independent method for explaining classification for individual instances. In *Data Warehousing and Knowledge Discovery*, pages 273–282, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-85836-2.

E. Štrumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(1):1–18, 2010. URL http://jmlr.org/papers/v11/strumbelj10a.html.

E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, 2014. ISSN 0219-3116. doi: 10.1007/s10115-013-0679-x. URL https://doi.org/10.1007/s10115-013-0679-x.