



**UNIVERSITÀ
DEGLI STUDI
DI TRIESTE**

UNIVERSITÀ DEGLI STUDI DI TRIESTE

**XXXVIII CICLO DEL DOTTORATO DI RICERCA IN
APPLIED DATA SCIENCE AND ARTIFICIAL
INTELLIGENCE**

SU FONDI DI ASSICURAZIONI GENERALI S.p.A.

**Enhancing Trustworthiness via Explanation
Robustness and Aggregations**

Settore scientifico-disciplinare: INF/01 INFORMATICA

**DOTTORANDA
ILARIA VASCOTTO**

Ilaria Vascotto

**COORDINATORE
PROF. FRANCESCO PAULI**

Francesco Pauli

**SUPERVISORE DI TESI
PROF. LUCA BORTOLUSSI**

Luca Bortolussi

**CO-SUPERVISORE DI TESI
PROF. ALEX RODRIGUEZ**

Alexandro Rodriguez Garcia

**CO-SUPERVISORE DI TESI
DR. ALESSANDRO BONAITA**

Alessandro Bonaïta

ANNO ACCADEMICO 2024/2025

Abstract

This Thesis focuses on the trustworthiness of eXplainable Artificial Intelligence (XAI) approaches, with a focus on the fields of insurance and asset management. As a first research direction, this work investigates the role of explanation robustness to increase the system’s trustworthiness. Robustness is, in fact, an essential property of both Artificial Intelligence (AI) systems and their explanations, but may be overlooked at times. Popular XAI techniques such as LIME and SHAP have been proved to be highly unstable but are still widely used in high-risk real-world applications.

The second research direction explores whether aggregating explanations may increase trustworthiness, by providing a more conservative and multi-perspective explanation. In particular, assuming that one of the individual explanations may fail or be non-robust for a single datapoint, an aggregation of explanations is able to account for this and provide a single explanation that summarizes multiple points of view. Critically, XAI methods suffer from the disagreement problem, that is, may at times return explanations that disagree with one another. An aggregation can be constructed to mitigate the risk of individual method failure and increase the overall trustworthiness of the AI system.

This work investigated both facets starting from the definition of a new robustness estimator via a properly constructed neighbourhood to aid the local robustness analysis. Aggregations have been explored both from a single-model and multi-model point of view, addressing the need of collecting explanations either from different XAI approaches or Machine Learning (ML) models. The initial analysis also allowed for the development of a pipeline to determine whether an explanation can be trusted as well as investigating the problem of unbalanced datasets. Furthermore, two new feature attributions were crafted and used in the multi-model aggregation.

Results have shown that explanation aggregation as well as the robustness evaluation are useful tools towards the development and deployment of trustworthy AI solutions.

Sommario

Questa Tesi si focalizza sulla affidabilità delle tecniche di Intelligenza Artificiale Spiegabile (in inglese, *eXplainable Artificial Intelligence*, XAI di seguito), con un focus sugli ambiti assicurativo e degli investimenti. Come prima direzione di ricerca, questo lavoro analizza il ruolo della robustezza delle spiegazioni al fine di aumentare l'affidabilità dei sistemi. La robustezza è, infatti, una proprietà essenziale sia dei sistemi di Intelligenza Artificiale (IA) che delle loro spiegazioni, ma talvolta risulta un aspetto trascurato. È stato provato che approcci di XAI molto popolari, quali LIME e SHAP, sono molto instabili ma ciò nonostante il loro utilizzo rimane esteso anche in applicazioni reali ad alto rischio.

La seconda direzione di ricerca esplora come l'aggregazione di spiegazioni possa aumentare l'affidabilità, proponendo una spiegazione più conservativa e sfaccettata. In particolare, assumendo che una delle singole spiegazioni possa fallire o risultare non robusta per uno specifico dato, una aggregazione di spiegazioni è in grado di tenere conto di questa vulnerabilità e proporre una singola spiegazione che riassuma molteplici punti di vista. E' importante sottolineare che i metodi di XAI sono soggetti al problema della discordanza (*disagreement problem*), ovvero possono talvolta restituire spiegazioni che sono in disaccordo tra loro. Una aggregazione può quindi essere costruita per mitigare il rischio di fallimento di singole tecniche e aumentare l'affidabilità del sistema di IA.

Questo lavoro affronta entrambi gli aspetti a partire dalla definizione di un nuovo stimatore della robustezza mediante un intorno appositamente costruito che supporta l'analisi locale della robustezza. Le aggregazioni sono state esplorate sia da un punto di vista del singolo modello che di molteplici modelli, tenendo conto della necessità di raccogliere spiegazioni o da diversi approcci XAI o da diversi modelli di apprendimento automatico. Le analisi iniziali hanno inoltre permesso lo sviluppo di una pipeline per determinare se una spiegazione sia affidabile ma anche per analizzare il problema dei dataset sbilanciati. Infine, due nuovi metodi di attribuzione sono stati sviluppati e usati nelle aggregazioni multi-modello.

I risultati ottenuti dimostrano che la aggregazione di spiegazioni, così come la valutazione della robustezza, sono strumenti utili per lo sviluppo e l'utilizzo di soluzioni IA affidabili.

Acknowledgments

The following contributions were made possible thanks to the support of Alessandro Bonaita and the Data Science group of Assicurazioni Generali, whom we thank for their interest and involvement in the project as well as for the sponsorship of my PhD scholarship.

I would like to thank Professor Luca Bortolussi for the opportunity to carry out this research within the AI Lab. My gratitude goes to Professor Alex Rodriguez for his interest, constant support and kind words.

Contents

Abstract	i
Sommario	iii
Acknowledgments	v
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	2
1.3 Thesis Structure	4
1.4 Publications	4
I Background	7
2 Foundations of Machine Learning	9
2.1 Branches of Machine Learning	9
2.2 Supervised Learning	10
2.2.1 k-Nearest Neighbours	10
2.2.2 Random Forest	11
2.2.3 Neural Networks	12
2.3 Unsupervised Learning	15
2.3.1 Clustering	15
2.3.2 Dimensionality Reduction	17
3 Introduction to eXplainable Artificial Intelligence	19
3.1 Why Do We Need Explanations?	19
3.2 Legislative Context	20
3.2.1 The AI Act	20
3.2.2 The GDPR	22
3.3 Taxonomy of XAI Methods	23
3.4 Types of Explanations	24
3.4.1 Local Explanations	25
3.4.2 Global Explanations	25

3.5	Evaluation of XAI Approaches	26
4	An Overview of XAI Techniques	29
4.1	LIME	29
4.2	SHAP	31
4.2.1	Kernel SHAP	32
4.2.2	Deep SHAP	33
4.3	Sensitivity and InputXGradient	33
4.4	Layerwise Relevance Propagation	34
4.5	DeepLIFT	36
4.6	Integrated Gradients	38
II	Contributions and Results	41
5	Robustness Estimation	43
5.1	Related Work	43
5.2	Robustness Estimator	47
5.3	Neighbourhood Generation	48
5.3.1	Random Neighbourhood ($\mathcal{N}_R$)	48
5.3.2	Medoid-based Neighbourhood ($\mathcal{N}_M$)	48
5.4	Results	49
5.4.1	Dataset and Model Description	49
5.4.2	Neighbourhood Generation and Robustness Evaluation	51
6	Aggregating Explanations	57
6.1	Related Work	57
6.1.1	The disagreement problem	57
6.1.2	Aggregations	58
6.2	Single-Model Aggregation	60
6.2.1	Ensemble Approach	60
6.2.2	Mean Aggregation	61
6.3	Results	62
6.3.1	Qualitative Evaluation	62
6.3.2	Robustness Analysis	63
7	An Evaluation Perspective	67
7.1	When Can You Trust Your Explanations?	67
7.2	The Pipeline	68
7.3	Validation	69
7.4	Results	70
7.4.1	Hyperparameter selection	70
7.4.2	Pipeline	71
7.4.3	Validation	71
7.5	Unbalanced Datasets	74
7.5.1	Minority Class Evaluation	74
7.6	Results	75

7.6.1	Dataset Description and Model Training	75
7.6.2	Reliability of Explanations	76
8	Feature Attributions for k-NN and RF	79
8.1	Related Work	79
8.2	Explaining k-Nearest Neighbours	80
8.2.1	Dataset Condensation	80
8.2.2	k-NN Explanation	83
8.3	Explaining Random Forests	84
8.4	Results	86
8.4.1	Qualitative Evaluation	86
8.4.2	Robustness Analysis	89
9	Aggregating Explanations from Multiple Models	91
9.1	Related Work	91
9.2	Multi-Model Aggregation	92
9.3	Results	93
9.3.1	Dataset and Model Description	93
9.3.2	Qualitative Evaluation	94
9.3.3	Robustness Analysis	95
9.3.4	Additional Metrics and Validation	96
9.3.5	Comparison to the Ensemble	98
10	Conclusions	101
10.1	Future Work	102
	Bibliography	105

List of Figures

2.1	A scheme of the branches of Machine Learning.	9
2.2	A visualization of the structure of a neuron and an example of a MLP for classification.	13
2.3	A comparison of k -means and HDBSCAN clustering on the Aggregation dataset.	16
2.4	A comparison of PCA and UMAP projections on the Fashion MNIST dataset.	18
3.1	A simple example to showcase the need for explanations.	19
3.2	The accuracy-interpretability trade-off.	24
4.1	A visual representation of LIME.	30
4.2	An example of SHAP explanations.	31
4.3	A visual representation of Saliency and Input X Gradient.	34
4.4	An example of LRP explanation mechanism on image data.	35
4.5	A visualization of DeepLIFT's backpropagation.	37
4.6	An example of the straight line path used in Integrated Gradients.	39
5.1	An example of neighbourhood generation on the Swiss roll dataset.	50
5.2	The robustness scores on the adult and heloc datasets with medoid-based neighbourhood.	55
5.3	A comparison between random and medoid-based neighbourhoods on the adult dataset.	55
6.1	The robustness scores on the adult dataset for the ensemble and mean aggregations, with medoid-based neighbourhood.	64
6.2	The robustness score on the adult dataset for the ensemble and mean aggregation with both a random and medoid-based neighbourhood.	65
6.3	HDBSCAN clusters, colored by mean robustness scores, over two dimensional UMAP projections.	66
7.1	The complete pipeline to test explanation trustworthiness.	69
7.2	ROC curves on the bank dataset with both aggregations and neighbourhood generation schemes.	73
7.3	A representation of the proposed steps for minority class evaluation.	75
7.4	The robustness of Integrated Gradients with a random and medoid-based neighbourhood, separated by class.	77
7.5	The robustness and consistency scores for the minority class.	78

8.1	The dataset condensation approach on the moons and circles datasets.	83
8.2	Distribution of the explanations for the first toy dataset.	87
8.3	Distribution of the explanations for the second toy dataset.	88
8.4	The robustness scores on the adult, bank and cancer datasets for the two explanations.	89
9.1	A representation of the proposed steps for multi-model explanations.	93
9.2	An example of the feature attributions derived on the adult dataset. .	94
9.3	The robustness scores on the bank and heloc dataset for the three models and their aggregation.	96
9.4	ROC curves on the bank and heloc datasets.	97
9.5	Comparison of the robustness scores and ROC curves on the bank dataset for multiple aggregations.	100

List of Tables

5.1	Dataset details.	51
5.2	Hyperparameter selection for both neighbourhood generation schemes.	52
5.3	Robustness scores with random neighbourhood.	53
5.4	Robustness scores with medoid-based neighbourhood.	53
6.1	Percentage of top k features of the individual methods within the ensemble's top k	63
6.2	Robustness scores for the ensemble and mean aggregation with medoid-based neighbourhood.	64
6.3	Robustness scores for the ensemble and mean aggregation with random neighbourhood.	65
7.1	Pipeline hyperparameter selection for both neighbourhood generating schemes and aggregation types.	70
7.2	Comparison of robustness scores for the ensemble and the mean aggregation with $r_{th} = 0.80$	71
7.3	Model accuracy (%) over the training set.	72
7.4	Comparison of ensemble robustness ($r_{th} = 0.80$) according to Models concordance in the prediction.	72
7.5	Average AUC of the three models for both aggregation types and neighbourhoods.	73
7.6	F1 score on train, validation and test sets divided by class.	76
7.7	Robustness and consistency scores on frost events.	78
8.1	Robustness scores for the k -NN and RF explanations.	89
9.1	Model accuracy (%) over the training set.	94
9.2	Robustness scores for the three explanations and the aggregation.	96
9.3	Comparison between models prediction agreement and average neighbourhood size.	97
9.4	AUC values for the three models and the aggregation.	97
9.5	Robustness scores for the two aggregations and the individual attributions (in absolute value).	99
9.6	AUC values for the three aggregations and the individual methods (in absolute sign).	100

Chapter 1

Introduction

1.1 Motivation

The role of Machine Learning (ML) and Artificial Intelligence (AI) has become pivotal in modern society, spreading in usage and importance in multiple areas of application. Among these, AI tools have gained popularity in high-risk applications such as healthcare [20] and finance [19].

Their rapid growth and increase in performance have brought up ethical questions on their usages, setting the basis for the development of an AI legislative regulation. The European Union has proposed two innovative regulations, the AI Act [38] and parts of the General Data Protection Regulation (GDPR) [37], stressing the need for a fairer and more transparent approach to Artificial Intelligence. Similarly, the United States of America have proposed the new Blueprint for an AI bill of rights [128] that strives for a fair development and deployment of AI systems. Notable inclusions in these regulations regard the need for a transparent AI and for explanations, aimed either at end-users or practitioners.

Modern AI systems have introduced a new degree of complexity due to the elevated number of parameters involved to solve challenging tasks and are often referred to as *black boxes* given the opaque nature of their predictions. Transparency is, instead, a fundamental property that AI systems should guarantee, aiming to provide detailed descriptions of the model reasoning, even in natural language. Imagine that a model is being used in healthcare for patient diagnosis. If doctors can understand how it reached a given prediction, they gain a valuable tool for evaluating the correctness of such diagnosis and grow confidence in the AI system, even if they don't fully comprehend the technical aspects of the black box model.

Explanation robustness is a critical aspect for the development of trustworthy AI systems. The robustness of an explanation method (also known as explainer) can be broadly defined as its ability to provide similar explanations for similar instances. In the context of local XAI methods, the robustness is strictly connected to data point locality and the manifold learned by the model. Robustness naturally ensures a higher degree of trust-ability as it provides by construction a guarantee of equal treatment under similar circumstances. Assume for example that two people with comparable characteristics (same age, same education, same annual income and so forth) apply for a loan at a bank and both loans are rejected. If the decision was

made by an automated decision system, it would be possible to retrieve explanations to elucidate on the decision. If the two explanations were contrasting (for instance, one saying that age was an important variable towards the rejection and the other stating that it was important towards the acceptance), the users could naturally lose a degree of confidence in the predictions and the system as a whole, effectively damaging the positive usages of AI in society.

Another practical problem may hinder the effectiveness of complex AI systems: the disagreement problem [69]. It needs to be accounted for in practical applications, especially when considering high-risk ones, as it represents the scenarios in which multiple explanations, applied to the same model and instances, return contrasting results. Consider for example a feature for which method A returns a positive attribution while method B a negative one. The discrepancies between explainability approaches may negatively affect the perception of model reliability and hinder its effective deployment. The debate over which explanation to choose (or trust) is still open, as explanation disagreement posits practical impediments to trustable AI systems. It has been shown in [69] that even experts tend to choose one method in these scenarios based on personal preferences, not quantitative evaluations. This can be also traced back to the lack of standardized criteria to evaluate explanations and their heterogeneity approach-wise.

Before introducing the contributions of this Thesis, let it be acknowledged that this work was supported by Assicurazioni Generali SpA, sponsors of my PhD scholarship. The research topic central to this Thesis plays a natural role in the insurance and asset management fields. Given the new transparency requirements pushed forwards by the Legislators to protect both the provider and the user, the usage of AI applications in high-risk use-cases can only be considered trustworthy if accompanied by explanations on the proposed outcome. In this scenario, explanation robustness becomes pivotal to achieve legislative compliance and guarantee a fair and equal treatment for the company's clients. Indeed, robustness measures the ability of providing similar explanations for similar inputs, effectively measuring how fair (that is, stable) a given explanation is.

The use-cases presented in Part II are derived from publicly-available datasets (with one exception in which only the response variable is derived from proprietary data and therefore not shareable) to preserve internal proprietary data and information of the Generali group. To mimic the data structure used in practical insurance use-cases, we have focused in the following on tabular datasets, made of both numerical and categorical variables, addressing binary or multiclass classification tasks (either balanced or unbalanced).

1.2 Contributions

This Thesis aims at tackling the previously-introduced research directions by addressing the disagreement problem via explanation aggregation and investigating the trustworthiness aspect through robustness estimation. The two facets go hand in hand towards the development of a trustworthy approach to AI in practical applications, enhancing the reliability of the AI system as well as the usage of explanations in high-risk applications.

The first research direction investigates the role of explanation robustness, introducing a novel robustness estimator for local feature attribution XAI approaches and a neighbourhood generation based on a clustering step, needed for the correct computation of the robustness score. The neighbourhood generation step ensures that the generated datapoints are on manifold and that the robustness score is computed on a consistent area of the data space, in contrast to random perturbations which may be off-manifold. We have tested the robustness to non-adversarial perturbations of multiple XAI methods applicable to Neural Networks, confirming that many popular techniques are - in fact - lacking from a robustness point of view and that may yield untrustworthy results.

Following these findings, we investigated the role of explanation aggregation (or *ensemble* of explanations) to increase trustworthiness in a single-model setting. This approach stems from the above-mentioned disagreement problem, in which multiple explanations are derived from the same model, and acts as a conservative solution towards feature importances disagreements. Moreover, the ensemble introduces a level of trustworthiness thanks to its robustness score, whose average behaviour yields more consistent results, even when the composing approaches disagree or are less robust. Its adoption in practical use-cases supports both reliability and transparency, as the explanation incorporates signals derived from multiple approaches.

Preliminary results on explanation aggregation and robustness have allowed us to devise a pipeline to test the trustworthiness of the explanations. It is intended to be used by practitioners to quantify the level of trust for a given local explanation. The analysis is made of both a local robustness computation and a prediction made by leveraging close datapoints within the manifold, effectively capturing aggregative behaviours in the neighbourhood. A novel validation assessment of explanation trustworthiness was developed, to address the lack of a ground truth (robustness-wise) in real-world use-cases.

We further investigated the role of explanation reliability in the context of unbalanced datasets, where the learning process and the derived explanation are particularly sensible for the majority class. Being more easily predictable, there are little guarantees that the prediction is based on learned data patterns. Investigating more in details the minority class instead produces a more reliable analysis and provides insights on the prediction of the class of interest.

To conclude, we shifted the focus from single-model aggregation to multi-model aggregation. The first contribution in this setting is the introduction of two new feature attribution approaches tailored for k -Nearest Neighbours and Random Forests. These have then been used, along with a Neural Network-specific explanation, to evaluate the effectiveness of multi-model aggregations. Specifically, this approach leverages the predictive power of multiple models and aggregated explanations derived from different decision-making processes, allowing for an effective analysis of the datapoint prediction. The robustness analysis was adapted to take into account possible disagreements in the prediction and has shown that the aggregation is a promising approach to increase reliability.

Code for the experimental evaluations is available on my Github page (<https://github.com/ilariavascotto>), with specific repositories cited in the relevant Chapters of this Thesis.

1.3 Thesis Structure

Following the Introduction, the Thesis will be divided into:

- **Part I. Background** Chapter 2 will present the basis of the Machine Learning models and techniques used in the remainder of the Thesis. Chapter 3 will present an introductory overview to the field of eXplainable Artificial Intelligence (XAI), its connections to the legislative field, the types of explanations and their evaluation. Chapter 4 will focus more in details on a subset of XAI techniques used in the following, delving into their theoretical background.
- **Part II. Contributions and Results** Chapter 5 will discuss explanation robustness and our proposal for the robustness estimator, its properties and evaluation on publicly-available datasets. Chapter 6 will introduce the role of explanation aggregation, with a focus on single-model aggregations. Chapter 7 will focus on the evaluation of the proposal, with a pipeline to test trustworthiness and an analysis on unbalanced datasets. Chapter 8 will propose two new feature attribution approaches tailored for k -Nearest Neighbours and Random Forests. Chapter 9 will conclude the contribution section with multi-model aggregations.

While both Chapters 6 and 9 tackle the role of explanation aggregation, they have been discussed separately in order to better integrate Chapter 7 within the single-model aggregation setting and address new explanations in Chapter 8 before considering the multi-model aggregation in Chapter 9.

Lastly, Chapter 10 will present a summary overview of the obtained results and a section on future steps for the research carried out so far.

1.4 Publications

The following list includes the publications that form the basis of this Thesis (listed in chronological order):

- [134] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi. When can you trust your explanations? A robustness analysis on feature importances. In R. Guidotti, U. Schmid, and L. Longo, editors, *Explainable Artificial Intelligence*, pages 225–249, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-032-08327-2. doi: https://doi.org/10.1007/978-3-032-08327-2_11
- [131] I. Vascotto, V. Blasone, A. Rodriguez, A. Bonaita, and L. Bortolussi. Assessing reliability of explanations in unbalanced datasets: a use-case on the occurrence of frost events. In P. Biecek, S. Nowaczyk, G. Kutyniok, and L. Longo, editors, *xAI-2025 Late-breaking Work, Demos and Doctoral Consortium Joint Proceedings*, volume 4017 of *CEUR Workshop Proceedings*, pages 73–80, 2025. URL https://ceur-ws.org/Vol-4017/paper_10.pdf

- [133] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi. Beyond single-model XAI: aggregating multi-model explanations for enhanced trustworthiness. In A. Følstad, D. Apostolou, S. Taylor, A. Palumbo, E. Tsalapati, G. Stamatellos, and R. Catelli, editors, *Proceedings of TRUST-AI 2025 - the European Workshop on Trustworthy AI*, volume 4132 of *CEUR Workshop Proceedings*, pages 276–286, 2025. URL <https://ceur-ws.org/Vol-4132/short38.pdf>

Additionally, the following work was accepted as a workshop abstract:

- [132] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi. Aggregating explanations to increase trustworthiness: current directions and challenges. Abstract presented at Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy, 2025

Part I

Background

Chapter 2

Foundations of Machine Learning

This Chapter will introduce the theoretical foundations to the Machine Learning models and techniques used in the following Chapters of this Thesis.

2.1 Branches of Machine Learning

Machine Learning (ML) can be defined as the science of getting computers to learn without being explicitly programmed for a given task. It encompasses a set of techniques that can range from classification to regression tasks, as well as clustering or dimensionality reduction.

The learning paradigms of Machine Learning (Figure 2.1) can be classified as:

- Supervised Learning: given an $(input, output)$ pair, it is possible to learn a function that maps the input to the output.
- Unsupervised Learning: given a dataset, it is possible to discover relationships and structures in the data.
- Reinforcement Learning: it focuses on the decision making of autonomous agents, which typically learn how to navigate a dynamic environment.

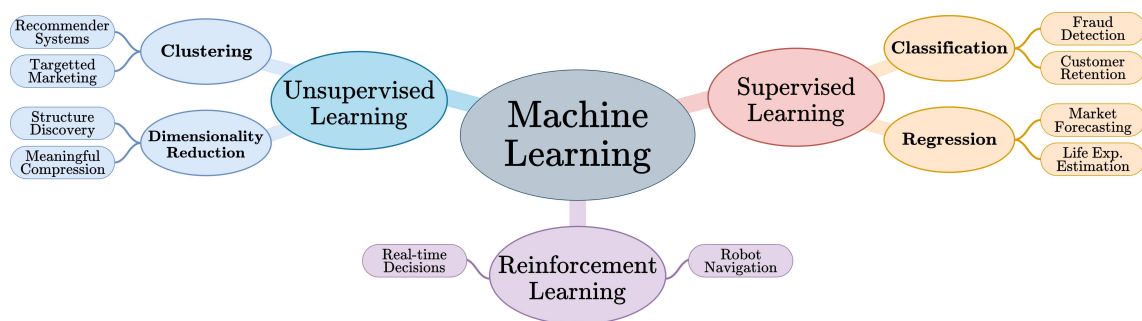


Figure 2.1: A scheme of the branches of Machine Learning.

2.2 Supervised Learning

Let $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ be a dataset of N datapoints, where $\mathbf{X} = (\mathbf{x}^1, \dots, \mathbf{x}^N) \in \mathbb{R}^{N \times m}$ is the input matrix and $\mathbf{y} = (y^1, \dots, y^N) \in \mathbb{R}^N$ is the output. Supervised learning is the task of learning a model $f : \mathbb{R}^m \rightarrow \mathbb{R}$ such that $f(\mathbf{x}) = y$. According to the output type, supervised learning algorithms can be used to solve either classification (binary or multiclass) or regression problems.

The dataset is usually split into a training and test datasets, namely $\mathcal{D}_{train}, \mathcal{D}_{test}$. The training set is used to train the model f such that $f(\mathbf{x}) \approx y$ while the test set represents previously unseen datapoints on which it is possible to evaluate the performance of the learned model. For classification problems, the accuracy is a good measure of quality, $acc(\mathbf{y}, \hat{\mathbf{y}}) = 1/N \cdot \sum_{i=1}^N \mathbb{I}[y^i = \hat{y}^i]$, while for regression problems the Mean Squared Error (MSE) is preferred, $MSE(\mathbf{y}, \hat{\mathbf{y}}) = 1/N \cdot \sum_{i=1}^N (y^i - \hat{y}^i)^2$. In both cases, $\hat{y}^i = f(\mathbf{x}^i)$ represents the predicted value while the vector $\mathbf{y}$ stores the ground truths.

2.2.1 k-Nearest Neighbours

k -Nearest Neighbours (k -NNs) is a non-parametric model that can be applied to both classification and regression tasks. It is based on a majority voting between the closest datapoints in the training set to predict the value of a new unseen datapoint in the test set.

More specifically, let $\mathcal{D}_{train}$ be the training set and let $d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ be a valid distance metric, such as the Euclidean distance. Given a test point $\mathbf{x} \in \mathcal{D}_{test}$, let $\pi_1(\mathbf{x}), \dots, \pi_N(\mathbf{x})$ be a reordering of the indices of the training set $\{1, \dots, N\}$ according to the distance from $\mathbf{x}$, $d(\mathbf{x}, \cdot)$, i.e. for all $i < N$ it holds that $d(\mathbf{x}, \mathbf{x}^{\pi_i(\mathbf{x})}) \leq d(\mathbf{x}, \mathbf{x}^{\pi_{i+1}(\mathbf{x})})$.

- Given a number k of neighbours, the k -NN rule for binary classification assigns point $\mathbf{x}$ a label $\hat{y}$ as the majority label among the set $\{y^{\pi_i(\mathbf{x})} : i \leq k\}$.
- For regression problems, the label $\hat{y}$ is usually defined as the average target of the k -NN, i.e. $\hat{y} = 1/k \cdot \sum_{i=1}^k y^{\pi_i(\mathbf{x})}$. This choice can be extended to consider more complex assignation strategies, such as the weighted average according to the distance from point $\mathbf{x}$.

Distance Metrics for Tabular Datasets Tabular datasets are characterized by the presence of both numerical and categorical variables. To correctly take them into account, the following distance metrics are used:

- Euclidean distance, when only numerical variables are present.

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| \quad (2.1)$$

- Hamming distance, when only categorical variables are present.

$$d(\mathbf{u}, \mathbf{v}) = \frac{1}{m} \cdot \sum_{j=1}^m \mathbb{I}[u^j \neq v^j] \quad (2.2)$$

- Gower distance, in the case of mixed numerical and categorical variables.

$$d(\mathbf{u}, \mathbf{v}) = \frac{1}{m} \cdot \sum_{j=1}^m s(u^j, v^j) \quad (2.3)$$

where the similarity $s(u^j, v^j)$ is computed according to the type of the j -th variable. If it is categorical, $s(u^j, v^j) \in \{0, 1\}$ where 1 denotes equality ($\mathbb{I}[u^j = v^j]$) and 0 inequality. If the variable is numerical, $s(u^j, v^j) = 1 - (|u^j - v^j|/R^j)$ with R^j the range of the j -th variable.

Hyperparameter Tuning Let $Err(\mathbf{y}, \hat{\mathbf{y}})$ be an appropriate error metric. The hyperparameter k representing the number of nearest neighbours can be chosen as the one minimizing the error Err on the training set. However, the division of the dataset $\mathcal{D}$ into training and test sets may influence the result, so a common practice is to perform k -fold Cross Validation (CV) and evaluate the performance of the model over different train-test combinations. If the value of k is too low, the predictions showcase greater variability, while, on the other hand, a large value of k reduces the variability of the predictions when the dataset is noisier but may be overall less accurate. A trade-off between accuracy and generalizability can be found by looking at the elbow plot, where various values of k are plotted against the error of the corresponding k -NN model.

2.2.2 Random Forest

Random Forests (RFs) [15] are an example of *ensemble* methods, a set of models that use multiple instances of a *base learner* to generate a more robust and accurate prediction.

Decision Tree A Decision Tree (DT) is a model that aims at subdividing the input space $\mathbb{R}^m$ into J non-overlapping regions R_j . For each observation $\mathbf{x} \in R_j$, the prediction is the same and is either the mean (regression) or the majority voted class (classification) of the outputs observed for the training observations that fall within the same region, $\mathbf{x}^i \in R_j$.

Decision trees are made of *nodes*: the first node is named *root node* and, along with the *internal* nodes, corresponds to the IF-THEN condition to be tested. Each node has two *children nodes*: the *left* child is reached if the condition is not met, while the *right* one otherwise. The terminal condition is reached at the *leaf nodes* which store the predicted values for the region R_j , whose bounds are defined by the set of conditions of the *decision path* traversed within the tree.

DTs are built with a top-down greedy approach known as recursive binary splitting. At each step of the building process, the best split is selected as the one maximising the reduction in error at that time. Specifically, given a feature x^j and a cutpoint s , the split creates the two regions $R_1(j, s) = \{\mathbf{x} | x^j < s\}$ and $R_2(j, s) = \{\mathbf{x} | x^j \geq s\}$. In a regression setting, for any feature j and split-point s , the goal is to minimize the Residual Sum of Squares (RSS):

$$RSS = \sum_{i: \mathbf{x}^i \in R_1(j, s)} (y^i - \hat{y}^i)^2 + \sum_{i: \mathbf{x}^i \in R_2(j, s)} (y^i - \hat{y}^i)^2 \quad (2.4)$$

The process is repeated until a stopping criterion is reached, for instance, the regions contain no more than n_R datapoints each or a maximum depth of the tree (number of nodes between the root and the leaf nodes over a single path) is reached.

Classification trees are instead built based on two possible measures to evaluate the quality of a split (*node impurity*). Having $\hat{p}_{jk}$ the proportion of training datapoints in the j -th region that belong to the k -th class, the Gini Index is defined as:

$$G = \sum_{k=1}^K \hat{p}_{jk}(1 - \hat{p}_{jk}) \quad (2.5)$$

An alternative is the entropy, defined as:

$$H = - \sum_{k=1}^K \hat{p}_{jk} \log(\hat{p}_{jk}) \quad (2.6)$$

Random Forest Random Forests [15] are a collection of decision trees that jointly contribute to make a prediction. They are more robust and accurate than individual DTs, and are an improvement over bagged trees as they decorrelate the trees.

This is achieved with two main caveats: the first one is dataset bootstrapping and the second one is feature randomization. As in bagging, the trees are built on a different bootstrapped training set of size n_b . During the building procedure of each of the B trees, the split is selected only on a random subset of p variables from the set of m total ones. Each split is associated with a new subset of features, and it usually holds that $p \approx \sqrt{m}$.

Assuming that T_b indicates the b -th tree in the Random Forest T , a prediction for a new point $\mathbf{x} \in \mathcal{D}_{test}$ is made as:

- Regression: $\hat{y} = 1/B \cdot \sum_{b=1}^B T_b(\mathbf{x})$
- Classification: $\hat{y} = \text{majority_vote}\{T_b(\mathbf{x})\}_{b=1}^B$

In the classification scenario, the prediction can be made using a *hard* or a *soft* assignation. In the former scenario, the majority vote is based on the most commonly predicted class by the individual trees in the forest. In the latter one instead, an average of the class probabilities derived from each tree is computed and the class is predicted as the one with maximum average probability. There may be scenarios in which the two assignations do not coincide.

2.2.3 Neural Networks

Neural Networks (NNs) are the fundamental model in the field of Deep Learning (DL). DL is a subfield of Artificial Intelligence that introduces complex representations derived from other simpler concepts.

The simplest example of a DL model is the feedforward deep network, also known as Multi-Layer Perceptron (MLP). The goal is to learn a model f_θ which accurately approximates the output y such that $f_\theta(\mathbf{x}) \approx y$. This is done by learning the set of parameters θ that best approximates the desired relationship between input $\mathbf{x}$ and output y .

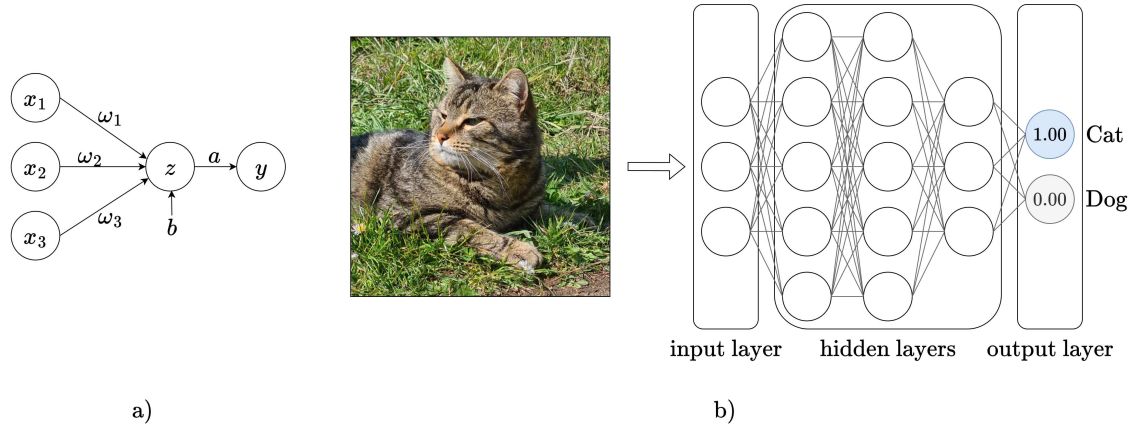


Figure 2.2: a) The structure of a neuron taking as input x_1, x_2, x_3 , with weights $\omega_1, \omega_2, \omega_3$ and bias b . It holds that $\mathbf{z} = b + \sum_i \omega_i x_i$ and $y = a(\mathbf{z})$. b) An example of a neural network taking as input the image of a cat and correctly classifying it.

Neurons The building block of NNs is the neuron, a mathematical concept inspired by biology and the neurons present in the human and animal brains. A neuron is a function that outputs a non-linear transformation of the input according to:

$$y = a(\boldsymbol{\omega} \cdot \mathbf{x} + b) \quad (2.7)$$

where a is known as *activation function*, while the parameters $\boldsymbol{\omega}$ represent the weight terms and b the bias. The term $\mathbf{z} = \boldsymbol{\omega} \cdot \mathbf{x} + b$ is also known as pre-activation.

Activations Common choices of the activation functions are the sigmoid, the hyperbolic tangent ($\tanh$) and the Rectified Linear Unit (ReLU).

$$a_{\text{sigmoid}}(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}} \quad a_{\text{tanh}}(\mathbf{z}) = \frac{e^{\mathbf{z}} - e^{-\mathbf{z}}}{e^{\mathbf{z}} + e^{-\mathbf{z}}} \quad a_{\text{ReLU}}(\mathbf{z}) = \max(0, \mathbf{z}) \quad (2.8)$$

Layers and MLPs A layer is a collection of neurons that operate on the same input. As in biology, neurons can operate in parallel to process a signal. A sequence of interconnected layers takes the name of Multi Layer Perceptron (MLP). Starting from an *input layer* in which each neuron is associated with one of the m features in the input vector, a series of *hidden layers* is concatenated until the *output layer* is reached. The l -th hidden layer of the network takes as input $\mathbf{y}_{l-1}$, the output of the previous layer:

$$\mathbf{y}_l = a(\boldsymbol{\omega}_l \cdot \mathbf{y}_{l-1} + b_l) \quad (2.9)$$

The input layer takes as input $\mathbf{x}$ while the result of the output layer is the model's prediction. In the case of regression tasks, the output layer is made of a single neuron, while for classification ones the final layer is made of K neurons (K being the number of classes in the dataset) and is usual preceded by a softmax activation.

Training The training of neural networks entails the learning of $\theta_l = (\boldsymbol{\omega}_l, b_l) \forall l \in (1, \dots, L)$, that is, of all the weights and biases of the neurons composing the

layers of the net, generally identified by $\boldsymbol{\theta}$. This is achieved via a gradient-based optimization of a chosen loss function, which measures the accuracy of the prediction. Given $\mathcal{L}(\mathbf{x}, \boldsymbol{\theta})$ a loss function, the model weights are learned via the backpropagation algorithm [102], which allows to compute the gradients with respect to the weights $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}, \boldsymbol{\theta})$ and propagate them through the network.

Typical choices for the loss function are the Mean Squared Error (MSE) for regression cases and the Cross Entropy for classification ones. If the output layer is preceded by a softmax layer, p_c represents the predicted probability associated to the ground truth class c .

$$\mathcal{L}_{regr}(\mathbf{x}, \boldsymbol{\theta}) = (f(\mathbf{x}, \boldsymbol{\theta}) - y)^2 \quad \mathcal{L}_{class}(\mathbf{x}, \boldsymbol{\theta}) = -\log(p_c) \quad (2.10)$$

Backpropagation For each layer l in the network, it is possible to compute the gradient of the loss with respect to the layer's weights.

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_l} = \boldsymbol{\delta}_l \cdot \mathbf{y}_{l-1}^T \quad (2.11)$$

where $\boldsymbol{\delta}_l$ represents the error signal propagated backwards from the output layer and $\mathbf{y}_{l-1}$ is the output of the previous layer. The term $\boldsymbol{\delta}_l$ is updated at each layer and starting from the output via:

$$\boldsymbol{\delta}_l = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{z}^l} \right) \cdot a'(\mathbf{z}^l) \quad (2.12)$$

where $\mathbf{z}^l$ is the pre-activation and a' is the derivative of the activation function.

For the preceding layer of the network, $l - 1$, the error can be computed as:

$$\boldsymbol{\delta}_{l-1} = (\boldsymbol{\theta}_l^T \boldsymbol{\delta}_l) \cdot a'(\mathbf{z}^{l-1}) \quad (2.13)$$

having $\boldsymbol{\theta}_l^T$ the transpose of the weight matrix that connects layer $l - 1$ and l .

Optimizer Optimization algorithms are used to update the weights $\boldsymbol{\theta}$ after the backward pass. The simplest one is Gradient Descent (GD) where, at iteration t :

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_{t-1}} \quad (2.14)$$

where η is known as learning rate parameter.

When the amount of data to be processed is elevated, computing the full loss becomes a practical impediment to computational efficiency. This problem is usually dealt with by using small batches of data for the optimization step, other than the whole dataset. The procedure is known as Stochastic Gradient Descent (SGD) but improvements such as Adaptive Moment Estimation (ADAM) [66] have gained popularity.

2.3 Unsupervised Learning

Unsupervised Learning encompasses a set of approaches in which the goal is to learn patterns and relationships within an unlabelled dataset, therefore with no indication on the desired outcome.

Unsupervised Learning approaches can be roughly divided into two categories: clustering and dimensionality reduction techniques.

2.3.1 Clustering

Clustering is the task of finding groups within a dataset $\mathcal{D} = \{\mathbf{X}\}$ such that datapoints with similar characteristics are grouped together and dissimilar datapoints are associated to different groups.

More specifically, a clustering algorithm takes as input a dataset $\mathcal{D}$, a distance metric d and outputs a collection of clusters $C = (C_1, \dots, C_k)$ such that $\bigcup_{i=1}^k C_i = \mathcal{D}$ and $\forall i \neq j, C_i \cap C_j = \emptyset$. The number of clusters k may be defined by the user or be extracted automatically during the clustering procedure. Moreover, clustering assignments may be *hard* (each point is assigned to only one cluster), *soft* (each point is assigned to each cluster with a given probability) or following a *dendrogram* (in the case of *hierarchical* clustering algorithms).

k-Means k -means [79] is one of the simplest clustering algorithms. k -means associates each cluster C_j to a cluster centre (*centroid*) c_j , such that:

$$c_j = \arg \min_{c \in \mathbb{R}^m} \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, c)^2 \quad (2.15)$$

Given k the number of desired clusters, k -means optimizes the following objective function:

$$O(\mathcal{D}, d, C) = \min_{c_j \in \mathbb{R}^m} \sum_{j=1}^k \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, c_j)^2 \quad (2.16)$$

This is achieved in practice via an iterative algorithm of *it* steps. The first k cluster centres are chosen randomly and the iterative procedure first assigns each datapoint $\mathbf{x} \in \mathcal{D}$ to the cluster of its closest centroid. It then re-computes the new cluster centres and repeats the assignments. This is done until a stopping condition is met: either the number of iterations is reached or the difference between two consecutive iterations is below a threshold value ϵ .

An improvement over plain k -means is k -means++ [8]. This variation modifies the initial random assignment such that the first cluster centres are far away from one another, stabilizing the optimization.

Nonetheless, k -means and k -means++ are based on the Euclidean distance and cannot be adapted to other distance measures. Moreover, the cluster centres - computed as the arithmetic mean of the clusters - may not coincide with the observations or the data distributions and are susceptible to outliers.

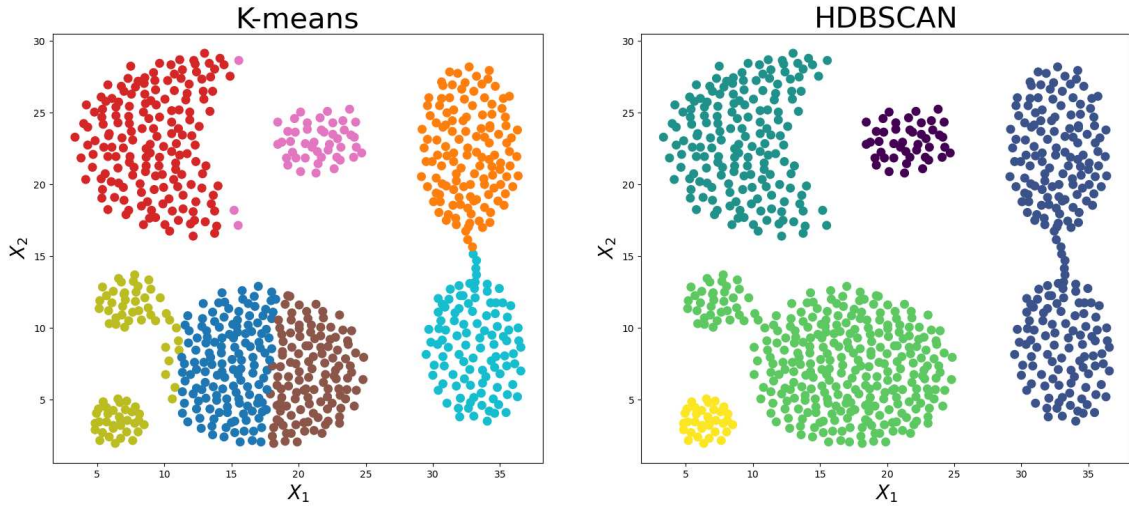


Figure 2.3: A comparison of k -means ($k = 7$) and HDBSCAN clustering on the Aggregation dataset. HDBSCAN identifies 5 clusters within the dataset.

k-Medoids k -medoids [91] improves over k -means by allowing to consider any distance metric, such as those introduced in Subsection 2.2.1. In particular, k -medoids constraints the cluster centres to belong to the dataset $\mathcal{D}$ (in this case, instead of centroid they are referred to as *medoids*).

In particular, the objective function becomes:

$$O(\mathcal{D}, d, C) = \min_{c_j \in \mathcal{D}} \sum_{j=1}^k \sum_{\mathbf{x} \in C_j} d(\mathbf{x}, c_j)^2 \quad (2.17)$$

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [18] is a density-based clustering algorithm. It extends DBSCAN [36] by converting it to a hierarchical algorithm and by selecting a flat clustering based on the computed cluster stability.

Density-based clustering algorithm identify regions of high-density the are separated by regions of low-density. In DBSCAN this is achieved by considering as clusters the density-reachable points from a reference *core point* p . A point is defined as core point if it has at least *min_points* points within its ϵ neighbourhood (a neighbourhood of radius ϵ). The points which are far from any cluster are considered noise or outliers.

The two hyperparameters of DBSCAN are $\theta = (\epsilon, \text{min_points})$, while the number k of clusters is automatically derived during the procedure. HDBSCAN extends DBSCAN and requires no hyperparameter selection on ϵ , as varying values are considered within the algorithm. In practice, it considers a Minimum Spanning Tree (MST) and uses it to build a hierarchy from which the most stable clusters are extracted. An advantage over DBSCAN is that clusters associated to very different densities can be extracted.

2.3.2 Dimensionality Reduction

Dimensionality Reduction is the task of finding a reduced representation of the dataset $\mathcal{D} = \{\mathbf{X} \in \mathbb{R}^{N \times m}\}$ such that the reduced representation lies in a lower-dimensional space, namely $\mathbf{Z} \in \mathbb{R}^{N \times p}$ with $p < m$.

PCA Principal Component Analysis (PCA) [93] is among the simplest dimensionality reduction techniques. It maps the dataset $\mathcal{D}$ into a lower dimensional representation $\mathbf{Z}$ via a linear transformation such that $\mathbf{Z} = \mathbf{X}W$. The reverse mapping can be computed via a matrix U such that $\mathbf{X} = \mathbf{Z}U$.

PCA aims at finding the matrices $W \in \mathbb{R}^{m \times p}$ and $U \in \mathbb{R}^{p \times m}$ such that the error between the original and the recovered vector is minimal.

$$\arg \min_{W \in \mathbb{R}^{m \times p}, U \in \mathbb{R}^{p \times m}} \sum_{i=1}^N \|\mathbf{x}^i - \mathbf{x}^i W U\|^2 \quad (2.18)$$

Having $\mathbf{X}$ centred, let $A = \sum_{i=1}^N \mathbf{x}^i \cdot (\mathbf{x}^i)^T$ and $\mathbf{u}_1, \dots, \mathbf{u}_p$ be the p eigenvectors of A associated with the p largest eigenvalues of A . The solution to PCA is given by setting U to the matrix whose columns are $\mathbf{u}_1, \dots, \mathbf{u}_p$ and, consequently, $W = U^T$.

The p dimensions over which the data is projected are known as Principal Components (PC).

UMAP Uniform Manifold Approximation and Projection (UMAP) [80] is a dimensionality reduction technique rooted in a theoretical framework based on Riemannian geometry and algebraic topology.

UMAP uses local manifold approximations and their local fuzzy simplicial set representations to construct a topological representation of the input. Given a lower dimensional representation of the data, the same steps are used to construct another topological representation. UMAP optimizes the layout in the lower dimensional space to minimize the cross entropy between the two topological representations.

UMAP represents an improvement over other techniques, such as t-SNE [130], as it is more computationally efficient and is better able to preserve global structures within the data.

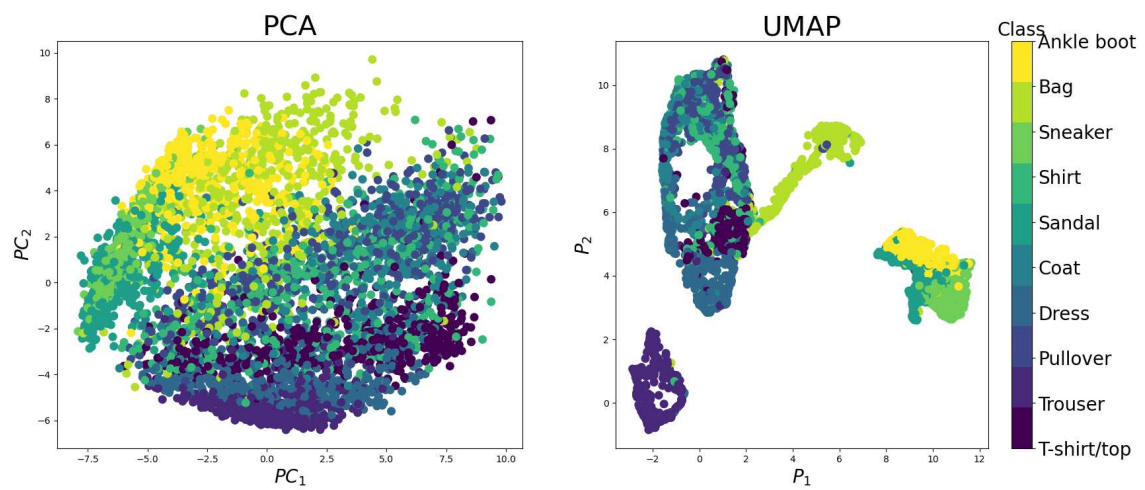


Figure 2.4: A comparison between the PCA and UMAP projections (with $p = 2$ components) on the Fashion MNIST dataset.

Chapter 3

Introduction to eXplainable Artificial Intelligence

The field of eXplainable Artificial Intelligence (XAI) aims at bridging the gap between complex, yet very accurate, models and interpretable ones, effectively attempting to *open the black box*. It is a sub-field of Artificial Intelligence which proposes a set of explainability approaches to shed the light on the decisional process of ML models. As modern AI and ML tools have increased their complexity, due to the elevated number of parameters involved, the need for explanations has become ever so prominent. This need was also strengthened by recent regulations, promulgated both in the European Union and the United States of America, that - for the first time - call for *trustworthy* Artificial Intelligence.

This Chapter will introduce the reader to the building blocks of XAI, its taxonomy, the types of explanations as well as their evaluation. It will moreover delve into the EU legislation on AI and their relationship with respect to the use of XAI in practical applications.

3.1 Why Do We Need Explanations?

Let us begin with a simple example (Figure 3.1). Assume that a person is interested in applying for a loan at the local bank. The request is processed and the decision is to reject the loan request. The user may want to know the reasons behind the rejections, to improve the application if a new request was to be submitted again. Two situations may arise:

- The loan request was processed by a human operator. A solution would be to ask for an appointment and go through the application with the employee.



Figure 3.1: A simple example to showcase the need for explanations.

- The loan request was processed by an automated decision system. If the reasoning behind a model is unknown, how can an explanation be retrieved?

In this latter case, the role of XAI methods becomes crucial: they, in fact, allow us to understand how a given output was produced, and possibly question the results. In this case, for example, the application may be re-examined under human supervision.

Explanations can not only be useful to understand how a model makes its decisions, but also to detect possible biases or faulty patterns learned by the model itself. A notable example is related to the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), a legal tool used in the United States to predict the risk of recidivism of inmates. While not explicitly using race as a variable (therefore being considered *race-blind*), COMPAS was found to be highly discriminatory against people of African descendant [126]. Another example can be found in [20], where the authors discuss how a ML model was trained to predict the risk of death associated with pneumonia. Through the usage of an *intelligible model* - namely a Generalized Additive Model with pairwise interactions (GA2M) [76] - they were able to discover that asthma was associated with a lower pneumonia severity. After better investigating the data and the model, it was found that patients with these characteristics were instead very sick and urgently admitted to the ICU, and the heavy treatment they received contributed to the successful recovery rate. Had the model been used in real applications, it would have caused severe damages, as it was making the predictions based on a correlation, other than causation, basis.

Similar examples of faulty behaviours have been discussed also in [41], where the authors mention a fictional use case in which a tank recognition tool was developed by the military. Despite having exceptional accuracy on the test set, in real-world applications it performed poorly. It was later discovered that the model was learning to discriminate between sunny and wet weather, other than recognizing if the tank was friendly or enemy. The pictures of friendly tanks in the training set were in fact always taken with sunny weather while those of enemy ones in wet conditions. In a similar manner, [96] crafted an example in which huskies and wolves were recognized by a binary classifier due to a mischievous training set. In particular, the wolves were always portrayed with snow in the background and the use of an XAI approach was able to reveal this error in the model.

3.2 Legislative Context

Recent legislative regulations have sparked the discussion on the usage of AI tools in the European Union [37, 38] and in the United States of America [128]. The fast pace at which AI and ML tools are evolving represents an inherent obstacle to legal systems, which often struggle to keep up with the novelties in the field. The proposal of state-of-the-art innovative regulations comes as a powerful statement and attempts at limiting damaging uses of AI in real-world scenarios.

3.2.1 The AI Act

The Artificial Intelligence Act (AI Act) [38] is a European regulation on AI, first proposed in 2021, that entered into force on August 1st 2024 and is expected to

become effective starting August 2026. It is the first comprehensive regulation on AI by a major regulator anywhere¹. The AI Act introduces a classification of AI systems into the minimal-risk, limited-risk, high-risk and unacceptable-risk categories. The first two are either unregulated or with limited, lighter obligations. Most of the AI Act focuses on the regulation of high-risk applications, as defined in Article 6 and Annexes II and III, while explicitly prohibiting unacceptable-risk applications (as per Article 5) such as social scoring systems and manipulative AI.

In the following, sections of Articles 13 and 14 are reported due to their importance for trustworthy AI systems, and their connection to XAI will be discussed.

Article 13, AI Act (Transparency and provision of information to users). [...]

2. *High-risk AI systems shall be accompanied by instructions for use in an appropriate digital format or otherwise that include concise, complete, correct and clear information that is relevant, accessible and comprehensible to users.*
3. *The information referred to in paragraph 2 shall specify: [...]*
 - d. *the human oversight measures referred to in Article 14, including the technical measures put in place to facilitate the interpretation of the outputs of AI systems by the users; [...]*

Recital 47 of the proposal (The principle of transparency) further states that *to address the opacity that may make certain AI systems incomprehensible to or too complex for natural persons, a certain degree of transparency should be required for high-risk AI systems*[...].

Article 14, AI Act (Human oversight). [...]

4. *The measures referred to in paragraph 3 shall enable the individuals to whom human oversight is assigned to do the following, as appropriate to the circumstances:*
 - a. *fully understand the capacities and limitations of the high-risk AI systems and be able to duly monitor its operation, so that signs of anomalies, dysfunctions and unexpected performance can be detected and addressed as soon as possible;*
 - b. *remain aware of [...] ‘automation bias’, [...];*
 - c. *be able to correctly interpret the high-risk AI system’s output, taking into account in particular the characteristics of the system and the interpretation tools and methods available; [...]*

While from a legislative point of view the AI Act is very sound, it may be difficult and complex to translate the legislative notion of transparency into a strict and well-defined mathematical concept which can be used by computer scientists. As stated in [34], the AI Act leaves to the practitioners the design of appropriate measures to guarantee legislative compliance, creating a gap between regulatory intent and

¹<https://artificialintelligenceact.eu/>

technical execution. The dis-homogeneity is derived also from the purposely vague statements in the proposal: in Article 13 for example the transparency is not referred to specifically, nor is its intended target. As far as the scope is concerned, it has been argued in [34] that the transparency shall be guaranteed towards the users of AI systems, yet XAI can contribute to both the user and the developer’s side, offering different-level inspections. Article 14 stresses the need for transparency by specifying that human oversight must be ensured and enabled by correct interpretative tools and methods, without providing more details on possible *desiderata*.

This ambiguity is, in part, intended as, considering the pace at which AI tools developed and evolved in the latest years, a too strict regulation may have become inapplicable before it even entered into force. This is, unfortunately, a consequence of the slow pace of legislative systems and the complexity of the already-available regulations. In fact, [34] argue that the AI Act still needs a deeper level of analysis. For example, compliance to regulations, when concerning privacy and trade secrets, should be taken into account to dually protect a company’s rights and the people affected by the outcome. Opening the black box in privacy-related cases may require the intervention of authorities or auditors, supervising the process and obliged to keep the results secrets.

3.2.2 The GDPR

The General Data Protection Regulation (GDPR) [37] entered into full force on May 25th 2018 as the main legislation in the European Union that refers to data usage and privacy. Its role in regulating AI systems spans from data collection, usage and ownership. We cite in particular Article 13 and Article 22 which underline that, in the legislative context, the data subject (i.e. *the identified or identifiable living individual to whom personal data relate*, Article 4, EU GDPR) has several rights concerning the automated processing of its data.

Article 13, EU GDPR (Information to be provided where personal data are collected from the data subject). [...]

2. *In addition to the information referred to in paragraph 1, the controller shall, at the time when personal data are obtained, provide the data subject with the following further information necessary to ensure fair and transparent processing: [...]*

f. the existence of automated decision-making, including profiling, referred to in Article 22(1) and (4) and, at least in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject. [...]

Article 22, EU GDPR (Automated individual decision-making, including profiling) is often referred to as the one describing the so-called ‘right to explanation’ and has contributed to active research in the field of XAI from companies looking to fulfil its requirements. Nonetheless, it must be noted that the GDPR does mention the right to explanation for individual decisions, but it does so in its non-binding recitals [53, 109].

The GDPR’s focus on transparency and individual rights has influenced the development and deployment of AI systems in Europe, requiring companies to take privacy and potential risks into account from the beginning. The GDPR provides a strong framework on AI governance with a focus on data protection, and now operates along with the AI Act for a comprehensive regulatory landscape on AI.

The role of XAI The Council of Europe’s ad-hoc committee on Artificial Intelligence [17] recently adopted a document pointing to transparency, explainability and accountability as central elements for the protection of individual’s rights. Similarly, the Ethics guidelines for trustworthy AI [47] posit that explainability is - together with fairness, the respect for human autonomy and the prevention of harm - one of the four ethical principles for trustworthy AI. In 2022, the European Commission has started the development of AI harmonised standards [39]: a common ground on the choice, development and usage of XAI approaches may have a considerable impact on the global state of AI transparency. Moreover, [119] investigated the extent by which XAI evaluation metrics align with the AI Act requirements, underlining the need for standardized metrics to ensure the effectiveness of the regulation. XAI can be a powerful tool to *open the black box* and comply with the legislations, provided that its approaches can be trusted and their effectiveness proven. Nonetheless, other points of view, such as [90], posit that XAI is not a mandatory requirement for more transparent and trustworthy AI, as it is not explicitly mentioned by the proposal. This raises the question about whether current research efforts on transparency can adequately support the AI Act’s objectives on trustworthiness.

3.3 Taxonomy of XAI Methods

As XAI is a growing field of research, multiple works have been proposed to classify the numerous XAI approaches that have been developed in the last ten years [1, 11, 21, 50, 103, 108, 122, 136, 137]. Despite the lack of an agreed-upon taxonomy in the field, XAI approaches can generally be classified according to the following axis, as first suggested in [1]:

- Scope of the explanation: a *local* approach aims at explaining how an individual prediction is made while a *global* one focuses on the model as a whole, analysing its overall reasoning.
- Model of interest: *model-specific* techniques are tailored to the structure of the model under investigation, while *model-agnostic* ones can be applied to any model.
- Transparency: *intrinsically transparent* models are interpretable by construction (also referred to as *glass boxes*) while *post-hoc* techniques are applied after the model is fully trained.

For neural network approaches, it is also possible to distinguish between *perturbation* and *gradient*-based approaches [105, 106]. The former ones are often model-agnostic while the latter ones are specific to neural networks and harness their inner structure, usually proposing backpropagation-like procedures.

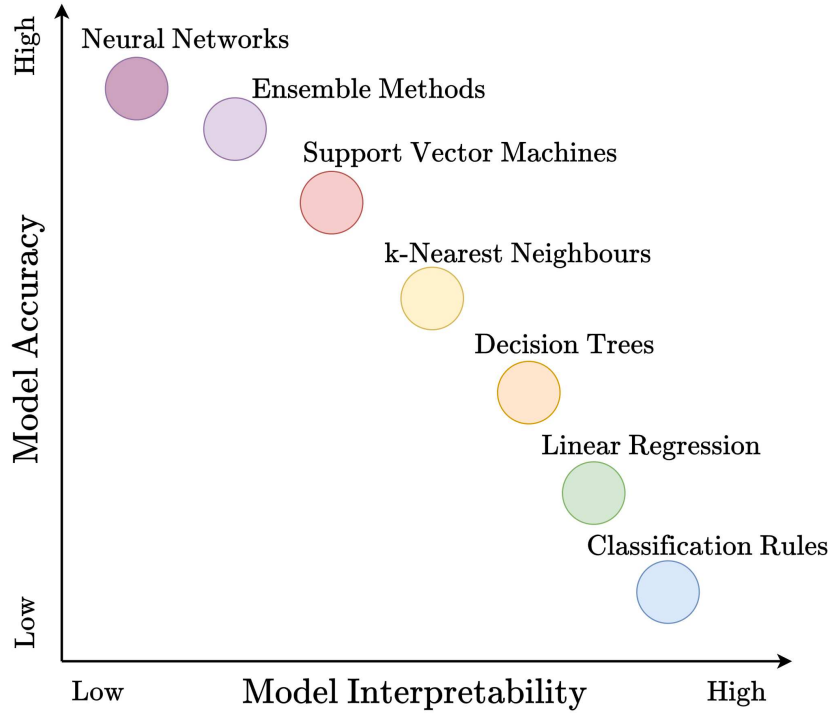


Figure 3.2: The accuracy-interpretability trade-off.

As discussed in multiple surveys, interpretability often comes at the cost of model accuracy (Figure 3.2). Models which are simpler by construction, such as classification rules, linear regression models or decision trees, can be easily interpreted but are usually less accurate. Of course, interpretability has its limitations: assume to be evaluating a classification rule with $n_{rules} = 4$ or with $n_{rules} = 250$. For a human, the first scenario can be assessed easily, while the second one is more complex and the elevated number of rules may hinder the effectiveness of the model's choice. On the other hand, highly-accurate models, such as ensemble methods and neural networks, are complex by construction, considering either an elevated number of base learners (individually interpretable by construction) or of weight parameters. In these cases, the models are able to effectively solve more complex tasks but their inner reasoning is often unknown.

3.4 Types of Explanations

XAI techniques can be classified according to the taxonomy presented in Section 3.3, but the variety of available approaches allows for the identification of explanation types as well. The following Subsections will present a brief overview of the explanation types, divided by explanation scope.

3.4.1 Local Explanations

Feature attributions A feature attribution method applied to a model f and a datapoint $\mathbf{x} = (x^1, \dots, x^m)$ returns a vector of size m of the form $\mathbf{a}(\mathbf{x}) = (a^1, \dots, a^m)$. Each entry of the feature attribution vector (also referred to as *feature importance* vector at times) represents the importance of the corresponding feature towards the model's outcome $f(\mathbf{x})$. A positive signed attribution indicates a positive influence towards the prediction, while a negative sign a negative one. Zero-valued attributions are usually associated with non-influencing features. Examples of feature attribution methods are LIME [96] and SHAP [77], as well as LPR [9], DeepLIFT [112] and Integrated Gradients [125]. They will be discussed in details in Chapter 4.

Heatmaps Feature attributions in the cases of images are also known as *heatmaps*. Let $\mathcal{I} \in \mathbb{R}^{w \times h \times c}$ be an image of width w , height h and with c colour channels. A heatmap is an explanation $\mathcal{H}(\mathcal{I}) \in \mathbb{R}^{w \times h}$ in which each pixel is assigned to an importance value relative to the prediction of model f . The aforementioned feature attributions can be used in these cases, but Convolutional Neural Networks-specific approaches exist, to take into account convolutional and pooling layers. Popular approaches are guided-backpropagation [123], CAM [143] and Grad-CAM [110].

Counterfactuals Inspired by sociological notions, counterfactual explanations try to answer the question "What would have happened if one condition had changed?". In particular, they make use of counterfactuals of the form $\mathbf{c} = (c^1, \dots, c^m)$, having m the number of features in the dataset. A counterfactual is constructed as the minimally-changed vector that changes class label such that $f(\mathbf{x}) \neq f(\mathbf{c})$. The use of counterfactuals should aid a user in understanding which features were the most important for a prediction and, specifically, how it is possible to change the predicted class label. Examples of counterfactuals are CLEAR [139], DiCE [85] and FACE [95].

Rule Extraction Given their inherent interpretability, decision rules can be used as an efficient explainability method themselves. The class of rule extraction methods aims at generating a rule (or set of rules) to explain a given prediction from model f . According to the class of models under investigation, the rules may be derived following a decision path (decision tree, random forest or ensemble methods) or a surrogate model may be constructed first. XAI methods that make use of rule extraction are Anchors [97], LORE [49], Bellatrex [28] and G-REX [68].

3.4.2 Global Explanations

Model Distillation To understand models at a global level, model distillation techniques use simpler interpretable models g to mimic the reasoning of the black box f , which in this context acts as an *oracle*. In particular, the predictions of the *oracle* are used as labels during the training of the interpretable model g . Examples of model distillation techniques are Trepan [26], STA [145] and CMM [32].

Prototypes and Criticisms With a reasoning similar to that of counterfactuals, prototypes represent instances of a given category or concept while criticisms are those that are not well-represented by the set of prototypes. Jointly considered, they can be used to explain a model both globally and locally, using representative datapoints to synthesize the model’s reasoning or the similarities within the data and a datapoint $\mathbf{x}$. Examples are MDD [63] and ABELE [51].

Other Techniques Graphical representations of the features, and their importance with respect to the outcome, can be found via PDP [42], ICE [44], ALE [6] plots and VEC curves [24]. Graph-based approaches for tree-based models such as DPG [7] preserve relations among features, logical decisions and predictions to bring to light global patterns. Other interesting approaches consider concept activation vectors [64] or cluster local explanations to provide a more global overview [72]. Note that some of the previously cited methods can be adapted to extract also global explanations, such as in the case of SP-LIME [96], SP-Anchors [97] and SHAP [77].

3.5 Evaluation of XAI Approaches

The variety of XAI approaches necessarily requires a set of evaluation metrics to compare them and validate their effectiveness, from both a technical standpoint and a user perspective.

The evaluation necessitates a strict definition of what should be tested: [75] proposes three directions for evaluation, namely *simulatability* (transparency at the level of the entire model), *decomposability* (at the level of the individual components, i.e. the parameters) and *algorithmic transparency* (at the level of the training algorithm). Similarly, [33] divides evaluation approaches into three categories: application-grounded (experiments with the end users), human-grounded (experiments with lay humans) and functionality-grounded (proxies based on a formal definition of interpretability).

[117] proposes that the explainability approaches tailored to predictive systems should be accompanied by a fact sheet assessing their functional and operational requirements, as well as an evaluation against user-usability criteria, a discussion on security, privacy and vulnerability and a validation via user studies or synthetic experiments. All the requirements operate as a potential trade-off between model accuracy and explanation faithfulness. The context in which an XAI approach operates can drastically change their usage and user understanding.

[121] introduces three criteria to evaluate the quality of the retrieved explanations, namely comprehensibility (for the users), fidelity (to the system) and assessability. The authors nonetheless test only six approaches, mainly applicable to NNs-only. They find LIME and TCAV being the most comprehensible with respect to, for instance, Integrated Gradients, but at the same time LIME was shown to be the least faithful method. For the assessability criterion instead they find it the most difficult to test, but underline that, as noted in [101], heatmaps may not generate information which can be used meaningfully. For example, the heatmap associated

with the predicted class, say *dog*, may not differ enough from that of another class, say *wolf*.

Surveys and toolkit for evaluation and benchmarking have been proposed in multiple works, such as [14, 58, 61, 87, 99, 144]. For example [99] proposes four objective metrics: performance differences D , number of rules R , number of features used F and stability of the explanations S . Nonetheless, these metrics are not widely applicable to XAI approaches and appear to be over simplistic. [61] is one of the few papers introducing mathematical formulations of the desired metrics and evaluations. Often, the discussion is presented at a high-level, with possible connections to the field of sociology when user studies are discussed, but lacking the mathematical rigour needed for a thorough evaluation and comparison of different approaches.

A more complete survey [86] introduces twelve properties on explanation quality (referred to as the Co-12 properties), subdivided into content, presentation and user dimensions. The work introduces also an interactive website for XAI approaches and functionally grounded evaluation methods. Notably, the proposed evaluation is accompanied by an XAI taxonomy underlining which metrics can be used according to the explanation type at hand.

Quantus [54] is an XAI toolkit to evaluate NNs with respect to a set of evaluation metrics, categorized into faithfulness, robustness, localization, complexity, randomization and axiomatic properties.

OpenXAI [2] is, according to the authors, a more complete toolkit with respect to Quantus, but only six explanation methods can be tested within their framework. However, they implement a set of quantitative metrics [69] which can be used to test faithfulness, stability and fairness. A benchmarking analysis is also discussed.

As a final note, [120] proposes an evaluation of XAI methodologies with respect to their legal compliance, as discussed in Subsection 3.2. They found that - for the time being - no single XAI approach satisfies all of the requirements, having SHAP the most legally faithful method but suffering from runtime and complexity issues.

Specific metrics relevant to the content of this Thesis will be discussed in details in Section 5.1 of Chapter 5.

Chapter 4

An Overview of XAI Techniques

This Chapter will present an overview of the XAI techniques used in the remainder of this Thesis. It will focus on the theoretical foundations of local feature attribution approaches, both of model-agnostic and model-specific nature, applicable to tabular dataset.

4.1 LIME

Local Interpretable Model-Agnostic Explanations (LIME) [96], as the name suggests, is a local and model-agnostic explainability technique. Developed by Ribeiro et al. in 2016, it rapidly gained popularity as one the first XAI approaches being proposed in the field. Nonetheless, its shortcomings - such as its inherent *instability* - were quickly detected and multiple works have tried to overcome them since [23, 49, 97, 111, 127, 138, 142]. Despite its faults, it is still used extensively as an explanation tool by uninformed practitioners.

LIME is built starting from interpretable representations of features in the form of binary vectors. In general, having $\mathbf{x} \in \mathbb{R}^m$ the input feature vector, $\mathbf{z} \in \{0, 1\}^b$ identifies its binary interpretable representation. In the context of natural language data, $\mathbf{z}$ indicates the presence or absence of a word, while for images it refers to contiguous patches of pixels (also known as super-pixels). For tabular data, a categorical variable is identified by the presence/absence of a selected class while numerical ones are categorized in a binary fashion as being above or below a chosen quantile.

Let $\mathcal{I}$ be a class of interpretable models, such as linear models, decision trees or falling rule lists. Let $e \in \mathcal{I}$ be an explanation model, defined on $\{0, 1\}^b$, and whose complexity is measured by $\Omega(e)$. For example, $\Omega(e)$ may be the depth of a decision tree or the number of non-zero coefficients in a linear model. Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be the model under investigation. Finally, consider $\pi(\mathbf{x}, \tilde{\mathbf{x}})$ a measure of proximity between $\mathbf{x}$, the original datapoint, and $\tilde{\mathbf{x}}$, its perturbation.

LIME computes its explanations by minimizing the following loss:

$$\xi(\mathbf{x}) = \arg \min_{e \in \mathcal{I}} \mathcal{L}(f, e, \pi) + \Omega(e) \quad (4.1)$$

where $\xi(\mathbf{x})$ is the explanation for the datapoint $\mathbf{x}$ derived by an explanation model e on the interpretable representation $\mathbf{z}$.

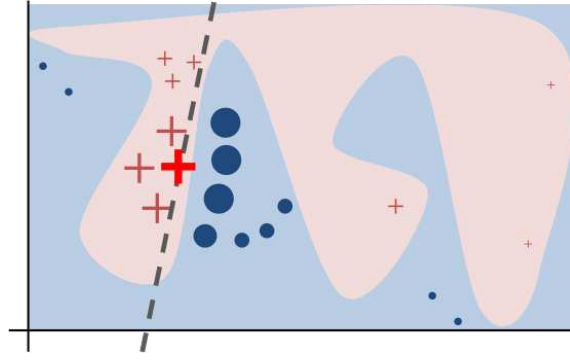


Figure 4.1: A visual representation of LIME, as depicted in [96]. The input of interest, $\mathbf{x}$, is represented by the red bold cross. The perturbed dataset $\mathcal{N}$ is made of the other red crosses and the blue dots, whose size is proportional to the distance weights $\pi(\mathbf{x}, \tilde{\mathbf{x}})$. The linear model e (dotted gray line) is fitted on such dataset and acts as a local linear approximator of the black box model f , whose decision boundaries are coloured in the background with paler tones.

The locality-aware loss $\mathcal{L}(f, e, \pi)$ measures the fidelity of the approximating model e to f by means of the proximity measure π , and a penalization term $\Omega(e)$ is applied to limit the complexity of the explanation model e .

The loss is practically optimized by means of a local sampling (Figure 4.1): starting from the interpretable representation of the input, $\mathbf{z}$, its non-zero elements are drawn uniformly at random. With $\tilde{\mathbf{z}} \in \{0, 1\}^b$ a perturbed sample, its original representation $\tilde{\mathbf{x}} \in \mathbb{R}^m$ is retrieved and $f(\tilde{\mathbf{x}})$ is computed and used as reference class label. Having $\mathcal{N}$ a dataset of perturbed samples and associated class labels, an explanation $\xi(\mathbf{x})$ is computed via Equation 4.1.

Assuming that $\mathcal{I}$ is set to be the class of linear models, it is possible to rewrite the loss term as:

$$\mathcal{L}(f, e, \pi) = \sum_{\tilde{\mathbf{x}}, \tilde{\mathbf{z}} \in \mathcal{N}} \pi(\mathbf{x}, \tilde{\mathbf{x}}) \cdot (f(\tilde{\mathbf{x}}) - e(\tilde{\mathbf{z}}))^2 \quad (4.2)$$

where $\pi(\mathbf{x}, \tilde{\mathbf{x}}) = \exp(-d(\mathbf{x}, \tilde{\mathbf{x}})^2/\sigma^2)$ is an exponential kernel defined on the distance function d with width σ . Equation 4.1 is then approximated by means of Lasso regularization, where the top k features are selected. This effectively includes the penalization term $\Omega(e)$ into the formulation as it governs the number of non-zeros coefficients of the model.

Remark Despite its simplicity, LIME is an inherently *unstable* method due to the sampling step required in the approximation phase. In particular, it has been shown that applying LIME multiple times on the same datapoint may result in different explanations, both in terms of selected features and in the sign of the coefficient associated with a specific feature. This behaviour is clearly undesirable as it hinders the utility of the explanations themselves. Attempts at improving the methods focus on the introduction of stability by construction, often with a modification of the sampling step [49, 142].

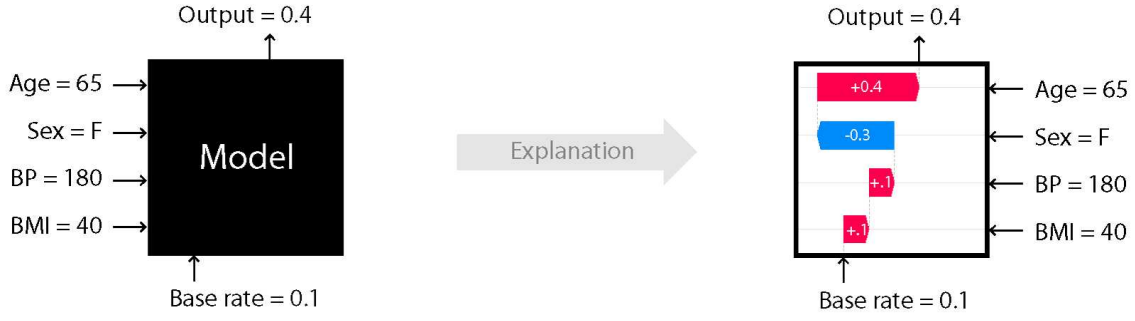


Figure 4.2: An example of SHAP explanations, as proposed in the official documentation². The input features of the datapoint of interest $\mathbf{x}$ are presented on the left side of the image. A black box model predicts the output value $f(\mathbf{x}) = 0.4$ and is associated with a baseline value $\mathbb{E}[f] = 0.1$. In the right-most part of the picture, the red bars represent positive valued coefficients ϕ_i associated with an explanation $e(\mathbf{x})$, while the blue one represents a negative valued one. It holds that $f(\mathbf{x}) = 0.4 = \mathbb{E}[f] + \phi_{Age} + \phi_{Sex} + \phi_{BP} + \phi_{BMI}$.

4.2 SHAP

SHapley Additive exPlanations (SHAP) [77] is a model-agnostic explainability technique first proposed in 2017 by Lundberg and Lee. Nowadays, it is one of the most popular techniques in both academia and industry, thanks to its theoretical background and open source python library². Despite its widespread usage, SHAP may result in unreliable explanations when its theoretical assumptions are not met in practise.

Definition 4.2.1. Additive feature attributions methods have an explanation model e that is a linear function of binary variables:

$$e(\mathbf{z}) = \phi_0 + \sum_{i=1}^b \phi_i \cdot z^i \quad (4.3)$$

where $\mathbf{z} = (z^1, \dots, z^b) \in \{0, 1\}^b$, b is the number of simplified input features and $\phi_i \in \mathbb{R}$.

It has been proved in [77] that the following local explanations belong to the class of additive feature attributions: LIME [96], DeepLIFT [112], Layer-Wise Relevance Propagation [9] and Shapley value estimation methods such as Shapley regression values [74] and sampling values [124]. The latter ones derive their reasoning from cooperative game theory to compute local explanations. Shapley values describe the average contribution of one feature (the *player*) in the model prediction (the *game*), over all possible combinations of the other features. Being computationally expensive to compute, an approximation is often used instead on a subset of feature combinations, adequately sampled.

²<https://shap.readthedocs.io/en/latest/>

Property 4.2.1 (Local accuracy).

$$f(\mathbf{x}) = e(\mathbf{z}) = \phi_0 + \sum_{i=1}^b \phi_i \cdot z^i \quad (4.4)$$

The explanation model $e(\mathbf{z})$ matches the original model $f(\mathbf{x})$ when $\mathbf{x} = h(\mathbf{z})$, having $h : \{0, 1\}^b \rightarrow \mathbb{R}^m$ the reverse mapping function.

Property 4.2.2 (Missingness).

$$z^i = 0 \Rightarrow \phi_i = 0 \quad (4.5)$$

Missingness constraints features where $z^i = 0$ to have no attributed impact.

Property 4.2.3 (Consistency). Let $f_x(\mathbf{z}) = f(h(\mathbf{z}))$ and $\mathbf{z} \setminus i$ denote the setting $z^i = 0$. For any two models f and g , if:

$$g_x(\mathbf{z}) - g_x(\mathbf{z} \setminus i) \geq f_x(\mathbf{z}) - f_x(\mathbf{z} \setminus i) \quad (4.6)$$

for all inputs $\mathbf{z} \in \{0, 1\}^b$, then $\phi_i(g, \mathbf{x}) \geq \phi_i(f, \mathbf{x})$.

Theorem 1. *Only one possible explanation model e follows Definition 4.2.1 and Properties 4.2.1, 4.2.2 and 4.2.3:*

$$\phi_i(f, \mathbf{x}) = \sum_{\tilde{\mathbf{z}} \subseteq \mathbf{z}} \frac{|\tilde{\mathbf{z}}|! \cdot (b - |\tilde{\mathbf{z}}| - 1)!}{b!} [f_x(\tilde{\mathbf{z}}) - f_x(\tilde{\mathbf{z}} \setminus i)] \quad (4.7)$$

where $|\tilde{\mathbf{z}}|$ is the number of non-zero entries in $\tilde{\mathbf{z}}$, and $\tilde{\mathbf{z}} \subseteq \mathbf{z}$ represents all $\tilde{\mathbf{z}}$ vectors where the non-zero entries are a subset of the non-zero entries of $\mathbf{z}$.

Following results from combined cooperative game theory, the coefficients ϕ_i are known as Shapley values. In particular, SHAP values are the Shapley values of a conditional expectation function of the original model f and are the solution to equation 4.7. It holds that $f_x(\tilde{\mathbf{z}}) = f(h(\tilde{\mathbf{z}})) = \mathbb{E}[f(\tilde{\mathbf{x}}) | \tilde{\mathbf{x}}_S]$ where S is the set of non-zero indices in $\tilde{\mathbf{z}}$.

As their exact computation can be particularly challenging, approximations are proposed. We will discuss in this section the model-agnostic approximation proposed in [77] and the version tailored for neural networks. Other model-specific approximations, taking advantage of the model's structure, have been proposed by the same authors, as in the case of TreeSHAP [78].

4.2.1 Kernel SHAP

Kernel SHAP combines LIME [96] and Shapley values for a faster approximation compared to classical estimation. Assuming feature independence and taking into account the linearity assumption on $e(\mathbf{z})$ (as per Definition 4.2.1), the following choice of loss function, complexity measure and kernel allow us to retrieve the Shapley values via linear regression.

$$\begin{aligned}
\Omega(e) &= 0 \\
\pi(\mathbf{z}, \tilde{\mathbf{z}}) &= \frac{(b-1)}{(b \text{ choose } |\tilde{\mathbf{z}}|) \cdot |\tilde{\mathbf{z}}| \cdot (b - |\tilde{\mathbf{z}}|)} \\
\mathcal{L}(f, e, \pi) &= \sum_{\tilde{\mathbf{z}} \in \mathcal{N}} [f(h(\tilde{\mathbf{z}})) - e(\tilde{\mathbf{z}})]^2 \cdot \pi(\mathbf{z}, \tilde{\mathbf{z}})
\end{aligned} \tag{4.8}$$

where $|\tilde{\mathbf{z}}|$ is the number of non-zero entries in $\tilde{\mathbf{z}}$, sampled from non-zero entries of $\mathbf{z}$.

4.2.2 Deep SHAP

Model-specific versions of SHAP allow for faster approximations of the Shapley values as they leverage the model’s structure during computation. Deep SHAP merges an older version of DeepLIFT [113] and Shapley values, as it was shown that DeepLIFT approximates SHAP values under the assumptions of feature independence and model linearity for the neural networks. In practical implementations, it makes use of a selection of *background* samples during the approximation step instead of a single reference value, as in [112].

Remark SHAP suffers from being computationally expensive, especially when the number of evaluated datapoints is high. Its reliance on approximations may lead to unexpected results when its theoretical assumptions (e.g. feature independence) are not met. The same authors proposed also the SAGE [25] method, a global feature importance approach that is able to deal with feature interactions by construction, effectively addressing one of SHAP’s shortcomings as feature independence is hardly met in practical applications. Nonetheless, the reliance of SHAP on the sampling step for the approximation makes it susceptible to instability and unreliable results [56, 57, 70]. Attempts at improving SHAP have been proposed to address both its computational cost and sampling step [16, 45, 59, 67, 88].

4.3 Sensitivity and InputXGradient

Among the first neural network-specific explainability techniques, saliency [115] acts as a local inspection method which is based on backpropagation. Instead of computing the derivatives of the loss function with respect to the neurons’ weights, saliency computes the derivatives of the output score function with respect to the input layer, effectively capturing the gradients’s influences towards the prediction.

$$a_{\text{saliency}}^i(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x^i} \tag{4.9}$$

A first naive improvement over saliency is Input $\times$ Gradient [89, 113], which multiplies the result of saliency by the input feature vector.

$$a_{\text{IxG}}^i(\mathbf{x}) = x^i \cdot \frac{\partial f(\mathbf{x})}{\partial x^i} \tag{4.10}$$

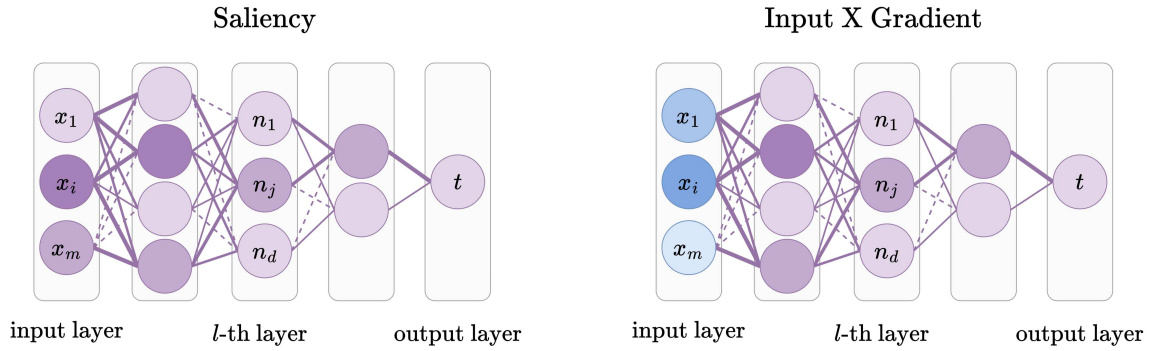


Figure 4.3: A visualization of the gradient's propagation in Saliency (left) and the same gradients multiplied by the input vector in the initial layer (right).

This allows to flag as non influential the features which were not originally present in the feature vector and to introduce a magnitude and sign adjustment to the computed explanations.

Nonetheless, both approaches rely on local gradient information which may introduce a bias due to gradient saturation or discontinuities in the neural network's activations [112]. Moreover, the mere computation of the gradients may carry insufficient information to effectively capture the inner mechanisms of complex black box models.

4.4 Layerwise Relevance Propagation

Layerwise Relevance Propagation (LRP) [9, 84] is one of the first proposed explainability methods tailored to neural networks. Published in 2015 by author S. Bach et al., LRP aims at explaining the output score $f(\mathbf{x})$, also known as *relevance*, by backpropagating the signal through the net.

Having j and k neurons in two consecutive layers of the neural network f , the following rule allows to propagate the relevance scores R_k at a given layer to the lower one:

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \quad (4.11)$$

where the quantity z_{jk} represents the amount by which neuron j has contributed to making neuron k relevant.

Property 4.4.1 (Conservation). Having j and k neurons in two consecutive layers of the neural network f , it holds that $\sum_j R_j = \sum_k R_k$ (*layer-wise conservation*). By applying the conservation property to the whole net, it can be proved that $\sum_i R_i = f(\mathbf{x})$ (*global conservation*).

When applied to deep neural networks with rectifier linearities (ReLU), it is easy to derive specific propagation rules. Neurons in networks of this type are of the form: $a_k = \max(0, \sum_{j \in \text{prev}} a_j \cdot w_{jk})$, having w_{jk} the weight in the network connecting neurons j and k in two subsequent layers.

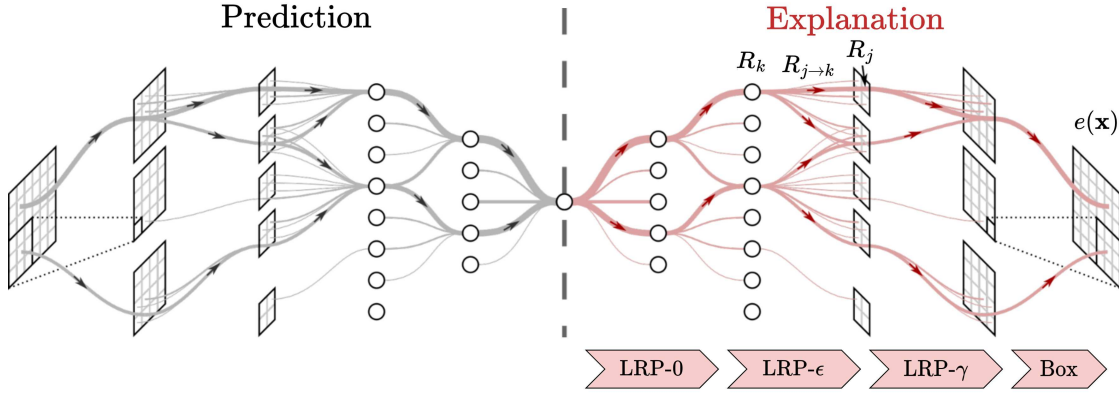


Figure 4.4: An example of how LRP works on images³. First the image $\mathbf{x}$ is processed through a neural network f and a prediction in the final layer is obtained. The relevance $f(\mathbf{x})$ is backpropagated through the net via a series of rules (composite LRP), tailored for each layer type and depth within the network, and the explanation $e(\mathbf{x})$ is obtained.

$$\begin{aligned}
 \text{LRP-0} : R_j &= \sum_k \frac{a_j \cdot w_{jk}}{\sum_{0,j} a_j \cdot w_{jk}} R_k \\
 \text{LRP-}\epsilon : R_j &= \sum_k \frac{a_j \cdot w_{jk}}{\epsilon + \sum_{0,j} a_j \cdot w_{jk}} R_k \\
 \text{LRP-}\gamma : R_j &= \sum_k \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j \cdot (w_{jk} + \gamma w_{jk}^+)} R_k
 \end{aligned} \tag{4.12}$$

It has been shown that a uniform application of the LRP-0 rule to the whole network is equivalent to applying the Input X Gradient method [113].

The LRP- ϵ rule introduces the coefficient $\epsilon > 0$ in the denominator to ensure numerical stability. The LRP- γ rule, instead, makes use of the parameter γ to govern how much positive contribution are favoured. w_{jk}^+ identifies a positive weight and $\sum_{0,j}$ indicates a summation over all lower-layer neurons.

Considering a rule applied to a neuron generally defined by $a_j \cdot \rho(w_{jk})$, LRP can be computed via Algorithm 4.1.

Algorithm 4.1 LRP computation

$\forall k : z_k = \epsilon + \sum_{0,j} a_k \cdot \rho(w_{jk})$	▷ forward pass
$\forall k : s_k = R_k / z_k$	▷ element-wise division
$\forall j : c_j = \sum_k \rho(w_{jk}) \cdot s_k$	▷ backward pass
$\forall j : R_j = a_j c_j$	▷ element-wise product

³<https://www.hhi.fraunhofer.de/en/departments/ai/technologies-and-solutions/layer-wise-relevance-propagation.html>

LRP exhibits links to Deep Taylor Decomposition (DTD) [82], another explainability technique which is based on Taylor expansions. They are applied to each neuron in the net and a baseline $\mathbf{x}^0$ (*root point*) is chosen such that it has near-zero output, $f(\mathbf{x}^0) = 0$. It can be shown that an appropriate choice of the baseline for DTD results in the LRP procedure [84].

LRP can moreover be extended to deal with composite rules [84, 104] (Figure 4.4) or for applications to other architectures of neural networks, such as Graph Neural Networks (GNNs) [107]. Nonetheless, the choice of individual versus composite rules (see Figure 10.4 from [84]) can result in great differences within the retrieved feature importances. The complexity in selecting the appropriate rule (or sets of rules) introduces a level of tuning which may damage the effectiveness of the explanations, as it may be unclear which one works best without introducing confirmation bias from the users.

4.5 DeepLIFT

Deep Learning Important FeaTures (DeepLIFT) [112, 113], originally presented by Shrikumar et al. in 2016, introduced a new paradigm in which explanations are presented with respect to a reference state. This allows to propagate an importance even when the gradient is zero, and alleviates the problem of discontinuities in the gradient. It moreover relies on a backpropagation-based algorithm, which makes its computation more efficient than perturbation-based methods.

DeepLIFT explains the difference in output from a *reference* output in terms of the difference in input from a *reference* input. The *reference* represents a baseline value which is selected as a suitable candidate according to the problem at hand.

Let t be the target neuron of interest, and let $n_1, \dots, n_d$ represent the neurons in an intermediate layer l . Having t^0 the reference activation of the neuron t , the difference-from-reference can be defined as $\Delta t = t - t^0$.

Property 4.5.1 (Summation to Delta). A contribution score $C_{\Delta n_j \Delta t}$ is assigned to Δn_j following:

$$\sum_{j=1}^d C_{\Delta n_j \Delta t} = \Delta t \quad (4.13)$$

It is important to note that $C_{\Delta n_j \Delta t}$ can be non-zero even when the gradient $\frac{\partial t}{\partial n_j}$ is null, addressing a fundamental limitation of gradient-based approaches.

For an input neuron x with difference-from-reference Δx , and a target neuron t with difference-from-reference Δt , a multiplier $m_{\Delta x \Delta t}$ is defined by:

$$m_{\Delta x \Delta t} = \frac{C_{\Delta x \Delta t}}{\Delta x} \quad (4.14)$$

Property 4.5.2 (Chain rule for multipliers). Assume to have an input layer with neurons $x_1, \dots, x_m$, a hidden layer with neurons $n_1, \dots, n_d$ and a target neuron t . The following rule is consistent with the summation-to-delta property:

$$m_{\Delta x_i \Delta t} = \sum_j m_{\Delta x_i \Delta n_j} \cdot m_{\Delta n_j \Delta t} \quad (4.15)$$

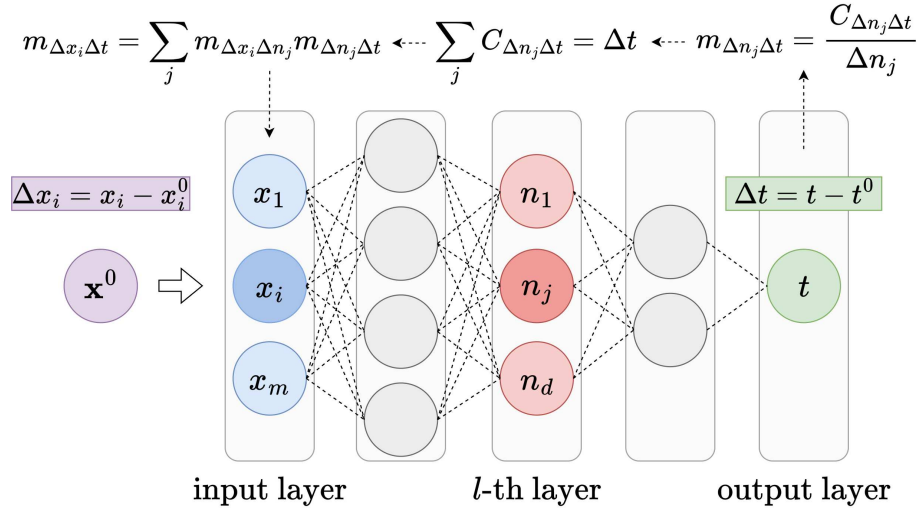


Figure 4.5: A visualization of the contribution propagation through a net, starting from the *difference-from-reference* in the output layer and following DeepLIFT's rules.

Positive and negative contributions for each neuron n_j , namely Δn_j^+ and Δn_j^- , can be defined such that:

$$\begin{aligned}\Delta n_j &= \Delta n_j^+ + \Delta n_j^- \\ C_{\Delta n_j \Delta t} &= C_{\Delta n_j^+ \Delta t} + C_{\Delta n_j^- \Delta t}\end{aligned}\tag{4.16}$$

The following rules allow to assign contribution scores to each neuron of its immediate input, effectively defining the backpropagation of the scores through the net via the chain rule. Let us consider a neuron n_j which receives as input a vector x and whose output is identified, without loss of generality, by y .

The Linear Rule It should be used for Dense and Convolutional layers, excluding non-linearities. Having y a linear function of its inputs, $y = w_0 + \sum_i w_i \cdot x_i$, it follows that $\Delta y = \sum_i w_i \cdot \Delta x_i$. The positive and negative parts of Δy can be defined as:

$$\begin{aligned}\Delta y^+ &= \sum_i 1\{w_i \Delta x_i > 0\} w_i \Delta x_i = \sum_i 1\{w_i \Delta x_i > 0\} w_i (\Delta x_i^+ + \Delta x_i^-) \\ \Delta y^- &= \sum_i 1\{w_i \Delta x_i < 0\} w_i \Delta x_i = \sum_i 1\{w_i \Delta x_i < 0\} w_i (\Delta x_i^+ + \Delta x_i^-)\end{aligned}\tag{4.17}$$

The contributions can be defined as:

$$\begin{aligned}C_{\Delta x_i^+ \Delta y^+} &= 1\{w_i \Delta x_i > 0\} w_i \Delta x_i^+ \\ C_{\Delta x_i^- \Delta y^+} &= 1\{w_i \Delta x_i > 0\} w_i \Delta x_i^- \\ C_{\Delta x_i^+ \Delta y^-} &= 1\{w_i \Delta x_i < 0\} w_i \Delta x_i^+ \\ C_{\Delta x_i^- \Delta y^-} &= 1\{w_i \Delta x_i < 0\} w_i \Delta x_i^-\end{aligned}\tag{4.18}$$

It follows that the multipliers take the form $m_{\Delta x_i^+ \Delta y^+} = m_{\Delta x_i^- \Delta y^+} = 1\{w_i \Delta x_i > 0\} w_i$ (and similarly for those containing Δy^-).

The Rescale Rule It should be used for non-linear transformations taking a single input, such as ReLU, tanh or sigmoid. Let y be a non-linear transformation of its input, $y = a(x)$. For the summation-to-delta property it holds that $C_{\Delta x \Delta y} = \Delta y$ and $m_{\Delta x \Delta y} = \frac{\Delta y}{\Delta x}$. Considering the positive and negative effects separately, and assuming that they are proportional to Δx^+ and Δx^- respectively:

$$\begin{aligned}\Delta y^+ &= \frac{\Delta y}{\Delta x} \Delta x^+ = C_{\Delta x^+ \Delta y^+} \\ \Delta y^- &= \frac{\Delta y}{\Delta x} \Delta x^- = C_{\Delta x^- \Delta y^-}\end{aligned}\tag{4.19}$$

It follows that $m_{\Delta x^+ \Delta y^+} = m_{\Delta x^- \Delta y^-} = m_{\Delta x \Delta y} = \frac{\Delta y}{\Delta x}$.

The Reveal Cancel Rule An improved version of DeepSHAP considers the effects of positive and negative contributions separately, effectively computing the Shapley values of Δx^+ and Δx^- contribution to y .

In particular, the Rescale rule can be rewritten starting from:

$$\begin{aligned}\Delta y^+ &= \frac{1}{2}(f(x^0 + \Delta x^+) - f(x^0)) + \frac{1}{2}(f(x^0 + \Delta x^- + \Delta x^+) - f(x^0 + \Delta x^-)) \\ \Delta y^- &= \frac{1}{2}(f(x^0 + \Delta x^-) - f(x^0)) + \frac{1}{2}(f(x^0 + \Delta x^+ + \Delta x^-) - f(x^0 + \Delta x^+)) \\ m_{\Delta x^+ \Delta y^+} &= \frac{C_{\Delta x^+ \Delta y^+}}{\Delta x^+} = \frac{\Delta y^+}{\Delta x^+} \quad m_{\Delta x^- \Delta y^-} = \frac{\Delta y^-}{\Delta x^-}\end{aligned}\tag{4.20}$$

The choice of the reference input is essential for the retrieval of informative explanations. The authors of DeepLIFT suggest that the choice of the baseline should be domain-driven and, at times, it may be necessary to compare the attributions derived from multiple baselines. For example, a good baseline for the MNIST dataset could be a black image (which corresponds to the background for all the digits) but this may not be an appropriate choice for other image datasets. Other studies [22, 40] question the use of a baseline, due to possible bias injection or artifacts.

4.6 Integrated Gradients

Sundarajan et al. presented Integrated Gradients (IG) [125] in 2017. It is a feature importance method that aims at solving the limitations encountered in previously-introduced approaches and is based on path methods for gradient computation.

As in the case of DeepLIFT, Integrated Gradients makes use of a baseline, against which explanations are computed. In this context, a baseline is defined as a point in the input space for which the model's prediction is neutral. Again, for the MNIST dataset a good choice would be a black image, while for a tabular dataset or a bag-of-words a zero-vector, representing the absence of signal, is considered more appropriate.



Figure 4.6: An example of the straight line path from baseline $\mathbf{x}^0$ (left, $\alpha = 0.00$) to the input $\mathbf{x}$ (right, $\alpha = 1.00$). In the context of images, the straight line path corresponds to increasingly more luminous images.⁴

Axiom 4.6.1 (Sensitivity - a). An attribution method e satisfies the axiom of sensitivity if, for every input and baseline that differ in one feature but have different predictions, then the differing feature should be given a non-zero attribution.

Axiom 4.6.2 (Implementation invariance). Two networks f and g are functionally equivalent if their outputs are equal for all inputs, despite having different implementations.

It has been shown in [125] that DeepLIFT [112] and LRP [9], among other methods, break at least one of the above axioms, which are instead deemed fundamental for attribution methods.

Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be a deep network. Let $\mathbf{x}$ be the input and let $\mathbf{x}^0 = \{x^{(0,i)}\}_{i=1}^m$ be the baseline input. Consider the straight-line path (in $\mathbb{R}^m$) from the baseline $\mathbf{x}^0$ to the input $\mathbf{x}$, and compute the gradient for all points along the path. Integrated Gradients is obtained by aggregating such gradients and, in particular, it is defined along the i -th dimension, a^i , as:

$$a^i(\mathbf{x}) = (x^i - x^{(0,i)}) \cdot \int_{\alpha=0}^1 \frac{\partial f(\mathbf{x}^0 + \alpha \cdot (\mathbf{x} - \mathbf{x}^0))}{\partial x^i} d\alpha \quad (4.21)$$

Integrated gradients satisfies also the following propositions.

Proposition 4.6.1 (Completeness). If $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is differentiable almost everywhere then $\sum_{i=1}^m a^i(\mathbf{x}) = f(\mathbf{x}) - f(\mathbf{x}^0)$.

In deep networks it is often possible to define a baseline such that the model prediction is near zero, i.e. $f(\mathbf{x}^0) \approx 0$. The black image (Figure 4.6) satisfies this property for image models: the interpretation in this case ignores the baseline and distributes the output score to individual features directly.

Integrated Gradients can be easily generalized to Path Integrated Gradients, assuming that the gradients are accumulated along a path (non necessarily the straight-line path). Let $\gamma : [0, 1] \rightarrow \mathbb{R}^m$ be a smooth path function that specifies the path in $\mathbb{R}^m$ from the baseline $\mathbf{x}^0$ to the input $\mathbf{x}$ such that $\gamma(0) = \mathbf{x}^0$ and $\gamma(1) = \mathbf{x}$. Then, for $\alpha \in [0, 1]$, the path integrated gradients along $\gamma(\alpha)$ is computed as:

$$a_{\gamma}^i(\mathbf{x}) = \int_{\alpha=0}^1 \frac{\partial f(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \cdot \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha \quad (4.22)$$

⁴Created following https://www.tensorflow.org/tutorials/interpretability/integrated_gradients

Integrated Gradients corresponds to a path method for $\gamma(\alpha) = \mathbf{x}^0 + \alpha \cdot (\mathbf{x} - \mathbf{x}^0)$. Path methods are the only methods satisfying the following desirable axioms.

Axiom 4.6.3 (Sensitivity - b). If the function implemented by the deep network does not depend (mathematically) on some variable, then the attribution of that variable is always zero.

Axiom 4.6.4 (Linearity). Suppose that we linearly composed two deep networks modelled by the functions f and g to form a third network that models the function $a \times f + b \times g$ to be the weighted sum of the attributions for f and g with weights a and b respectively. Intuitively, we would like the attributions to preserve any linearity within the network.

Proposition 4.6.2. Path methods are the only attribution methods that always satisfy Implementation Invariance, Sensitivity - b, Linearity and Completeness.

Property 4.6.1 (Symmetry-preserving). Two input variables x^i and x^j are symmetric preserving w.r.t f if and only if $f(x^i, x^j) = f(x^j, x^i)$ for all values of x^i and x^j .

This property is expected to be satisfied for attribution methods as, if two variable play the exact same role, they are expected to receive the same importance. In particular, the following Theorem holds.

Theorem 2. *Integrated gradients is the unique path method that is symmetry-preserving.*

In practical terms, Integrated Gradients can be computed via an approximation of s steps:

$$a^i(\mathbf{x})^{\text{approx}} = \frac{x^i - x^{(0,i)}}{s} \cdot \sum_{k=1}^s \frac{\partial f(\mathbf{x}^0 + \frac{k}{s} \cdot (\mathbf{x} - \mathbf{x}^0))}{\partial x^i} \quad (4.23)$$

The authors of [125] show that $s \in (20, 300)$ is enough to provide good approximations of the integrals, but the computation can nonetheless become expensive when the number of step grows.

Works aiming at improving Integrated Gradients have been proposed [35, 60, 62, 140], where the usage of the baseline and of the chosen path is discussed and computationally-efficient methods are presented. Notably, neuron-level attributions have also been investigated with the introduction of Neuron IG [114], a method which proposes a scalable implementation of Total Conductance [29].

Part II

Contributions and Results

Chapter 5

Robustness Estimation

This Chapter will introduce the notion of *explanation robustness*, which has been central for the research carried out during the development of this Thesis.

As we will focus on tabular datasets and classification problems in the remainder of this Thesis, let us remark some useful notation. Let $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ be a dataset with N datapoints and m features such that $(\mathbf{x}^i, y^i) = (x^{(i,1)}, \dots, x^{(i,j)}, \dots, x^{(i,m)}, y^i)$ where y^i is a class label. The dataset is split into a training, validation and test sets, $\mathcal{D}_{train}, \mathcal{D}_{valid}, \mathcal{D}_{test}$ respectively. Let f be a model trained on $\mathcal{D}_{train}$ and let t indicate a target class. Note that, for classification tasks and neural networks models, the selection of the output neuron to be explained is required: in those scenarios we will select the target t as the predicted class neuron such that $t = \arg \max(f(\mathbf{x}))$. In general, the target t will coincide with the predicted class of the model f . Let e be an explanation method (also called *explainer*) and $e(\mathbf{x}^i) := e(\mathbf{x}^i, f)$ the explanation of model f prediction for the point $\mathbf{x}^i$. If e is a feature attribution method, then the feature attribution vector is indicated by $\mathbf{a}^i = (a^{(i,1)}, \dots, a^{(i,j)}, \dots, a^{(i,m)})$.

Robustness of explanations can be broadly defined as their ability to *produce similar explanations for similar inputs*. Considering $\mathbf{x}$ the original datapoint, $\tilde{\mathbf{x}}$ a perturbation, e an explanation method and $e(\mathbf{x})$ the corresponding explanation then:

$$\mathbf{x} \rightarrow \tilde{\mathbf{x}}, e(\mathbf{x}) \rightarrow e(\tilde{\mathbf{x}}) \Rightarrow r(\mathbf{x}, e) = g(\mathbf{x}, \tilde{\mathbf{x}}, e) \quad (5.1)$$

that is, the robustness r of $e(\mathbf{x})$ is a function of the chosen explanation method, the original datapoint and the perturbed one.

More formally, considering a constraint of the form $d(\mathbf{x}, \tilde{\mathbf{x}}) < \epsilon$ with $\epsilon > 0$ and d a distance metric (for example, the Euclidean distance) it is possible to introduce the notion of local robustness.

Definition 5.0.1. An explanation method e is locally robust if perturbing the input results in a similar explanation. If $\mathbf{x} \rightarrow \tilde{\mathbf{x}}$ with $d(\mathbf{x}, \tilde{\mathbf{x}}) < \epsilon$ ($\epsilon > 0$) $\Rightarrow e(\mathbf{x}) \approx e(\tilde{\mathbf{x}})$.

5.1 Related Work

As mentioned in Chapter 4, LIME [96] and SHAP [77] are among the most widely used XAI techniques in real-world applications. Despite their popularity, both meth-

ods lack robustness: LIME is, in fact, an unstable method by design as its neighbourhood generation steps yields different sets of datapoints at each call of the method. Undesirably, this implies that, even when the method is applied to the same datapoint $\mathbf{x}$, a different model f is being fitted and the extracted explanations may be different, differing feature-wise in magnitude or attribution sign. Similarly, SHAP lacks stability due to its susceptibility to feature correlations, sampling variability and data distribution shifts due to the way Shapley values are approximated.

The instability of both methods was first proved in [116], where the authors showed that an adversarial classifier can be easily defined to mask biased decisions via unbiased explanations. In particular, assume to have a biased model f that is applied to the input $\mathbf{x} \in \mathbb{R}^m$. As in LIME and SHAP, let $\tilde{\mathbf{x}} \in \mathbb{R}^m$ be a perturbation which lies off-manifold or is out-of-distribution. It is then possible to train an adversarial model $\tilde{f}$ that acts like the original model on the original input data, but follows an unbiased behaviour ψ on the perturbations $\tilde{\mathbf{x}}$. The explanations derived on $\tilde{\mathbf{x}}$ will therefore behave in an unbiased manner, while the original model f remains biased, effectively fooling any practitioner investigating it, making it look unbiased and therefore safe.

The work of [46] investigates if additive explanations can be trusted when used on non-additive models. More specifically, they underline how the theoretical assumptions from both LIME and SHAP on feature independence are hardly met in real-world scenarios, therefore feature interactions are not taken into account in the explanations.

[81] suggests explanation robustness to be tested along three directions: robustness to input perturbations (either from adversaries or from changes in the data distribution), to model changes (model modifications that act on the same input) and to hyperparameter selection (for the XAI methods). For the first direction, works on feature attributions usually focus on saliency maps. In [3], for example, the authors show the LIME and SHAP are significantly less robust than gradient-based saliency methods while [65] demonstrates that input perturbations of the form $\tilde{\mathbf{x}} = \mathbf{x} + \lambda$ with $\lambda \in \mathbb{R}^m$ a constant can cause gradient-based saliency methods to produce wrong explanations. Other works [30, 43] demonstrate how gradient-based methods can be forced to generate arbitrary explanations due to the geometrical properties of the underlying NNs.

In the case of model changes, [5] show that it is always possible to find an adversarial model $\tilde{f}$ with the same behaviour of the original one f but with manipulated explanations (*fairwashing of explanations*). The authors propose a version of the analysed gradient-based explanations ([9, 125]) robust to model manipulations by projecting the explanations on the tangent space of the data manifold.

An adversarial attack in the context of XAI is a perturbation of the input such that the model prediction is unchanged but the explanation marks different features as important or not [10]. An evaluation is provided in [43], where attacks to NNs explanations maximise the change in the explanations (with respect to different change functions) under the constraint that the perturbation is small and the model's prediction unchanged.

While random perturbations and adversarial attacks to explanations in the context of images can be more easily examined by human experts, as the changes are

often evident to the naked eye, it is not sufficient to rely on non-quantitative evaluations. This is inherently required when working with tabular datasets, for which small changes may be meaningless for a human supervisor but of high importance for the black box model.

A first attempt at formalizing robustness can be found in [3], where robustness of explanations is defined through point-wise neighbourhood-based local Lipschitz continuity.

Definition 5.1.1. A model $f : \mathbf{X} \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$ is locally Lipschitz if for every $\mathbf{x}^0$ there exists $\delta > 0$ and $L \in \mathbb{R}$ such that $\|\mathbf{x} - \mathbf{x}^0\| < \delta$ implies $\|f(\mathbf{x}) - f(\mathbf{x}^0)\| \leq L\|\mathbf{x} - \mathbf{x}^0\|$.

This definition can be easily extended to consider an explainer e instead of a generic predictive model f . The constant L must be estimated as, for every point, the following optimization must be solved :

$$\hat{L}(\mathbf{x}^i, e) = \arg \max_{\mathbf{x}^j \in \mathcal{N}_\epsilon^i} \frac{\|e(\mathbf{x}^i) - e(\mathbf{x}^j)\|_2}{\|\mathbf{x}^i - \mathbf{x}^j\|_2} \quad (5.2)$$

where $\mathcal{N}_\epsilon^i$ is a ball of radius ϵ centred at $\mathbf{x}^i$.

A (weaker) empirical notion of stability allows to trivially compute the value of $\hat{L}$ in practical applications. Having $\mathbf{X} = \{\mathbf{x}^i\}_{i=1}^N$ and for every $\mathbf{x}^i \in \mathbf{X}$, let $\mathcal{N}_\epsilon^i = \{\mathbf{x}^j \in \mathbf{X} \mid \|\mathbf{x}^i - \mathbf{x}^j\| \leq \epsilon\}$ then:

$$\hat{L}_{\mathbf{X}}(\mathbf{x}^i, e) = \arg \max_{\mathbf{x}^j \in \mathcal{N}_\epsilon^i} \frac{\|e(\mathbf{x}^i) - e(\mathbf{x}^j)\|_2}{\|\mathbf{x}^i - \mathbf{x}^j\|_2} \quad (5.3)$$

Nonetheless, it yields robustness scores which are not bounded, making it difficult to effectively compare methodologies and results.

The survey [86] proposes a set of metrics to evaluate explainability approaches in terms of their robustness. In particular, they refer to similarities measures such as rank order correlations, top- k intersections, rule matching and structural similarity indexes. [99] suggested the Jaccard similarity as a similarity metric. Of course, according to the type of explanation at hand, one or more of these metrics may not be applicable.

The toolkit [54] introduces the following metrics for robustness evaluation, the first one being the local Lipschitz continuity as per Equation 5.3 [3]. A closely related metric is the max-sensitivity defined in [141] for an explainer e :

$$\text{max-sensitivity}(\mathbf{x}^i, \mathcal{N}_\epsilon^i, e, f) = \max_{\tilde{\mathbf{x}}^i \in \mathcal{N}_\epsilon^i} \|e(\mathbf{x}^i, f) - e(\tilde{\mathbf{x}}^i, f)\| \quad (5.4)$$

It measures the maximum change in the explanation with respect to a perturbed input $\tilde{\mathbf{x}}^i$.

Similarly, [83] proposes explanation continuity be measured by the strongest variation of an explanation $e(\mathbf{x})$ in the input domain such that:

$$\text{continuity}(\mathbf{x}^i, e) = \max_{\mathbf{x}^j \in \mathbf{X} \text{ s.t. } \mathbf{x}^i \neq \mathbf{x}^j} \frac{\|e(\mathbf{x}^i) - e(\mathbf{x}^j)\|_1}{\|\mathbf{x}^i - \mathbf{x}^j\|_2} \quad (5.5)$$

[27] introduces the notion of consistency to measure the similarity between datapoints which are associated with the same explanation. For example, assume that two images are assigned to the same explanation $e(\mathbf{x}^i) = e(\mathbf{x}^j) = \text{"contains zebra"}$, then they should also be classified to the same class. For an explanation ξ , let $C_\xi = \{\mathbf{x}^j \in \mathbf{X} \mid e(\mathbf{x}^j) = \xi\}$ be the set of instances assigned to the same explanations. Let μ be a distribution over the instances $\mathbf{X}$. Then, the local consistency can be defined as:

$$\text{local consistency}(\mathbf{x}^i, e, f) = \Pr_{\mathbf{x}^j \in C_\xi^\mu} (f(\mathbf{x}^j) = f(\mathbf{x}^i)) \quad (5.6)$$

where $\mathbf{x}^j \in C_\xi^\mu$ means that $\mathbf{x}^j$ is drawn from the distribution μ restricted to the set C_ξ . The global consistency can be retrieved as the expected value of the local consistency over the whole dataset $\mathbf{X}$.

Lastly, the toolkit implements the metrics of [2], namely relative input stability, relative representation stability and relative output stability. In particular, they extend the metric of [3] so that the relative input stability (RIS) becomes:

$$\text{RIS}(\mathbf{x}^i, \mathcal{N}^i, e, f) = \max_{\mathbf{x}^j \in \mathcal{N}^i \text{ s.t. } f(\mathbf{x}^i) = f(\mathbf{x}^j)} \frac{\left\| \frac{e(\mathbf{x}^i) - e(\mathbf{x}^j)}{e(\mathbf{x}^i)} \right\|_p}{\max \left(\left\| \frac{\mathbf{x}^i - \mathbf{x}^j}{\mathbf{x}^i} \right\|_p, \epsilon_{min} \right)} \quad (5.7)$$

with $\epsilon_{min} > 0$ needed for stability. The other two metrics modify the denominator of this formulation to include, in the first case, the relative difference between latent representations and, in the second one, the l_p norm of the output logits for the two datapoints. These metrics are the only one used in the OpenXAI toolkit [2] for explanation robustness.

The work of [69] defines metrics to capture explanation disagreement with respect to the top- k features or some features of interest. They select feature agreement (fraction of common features between the top- k sets of two explanations), rank agreement (the features in the top- k sets that are common and in the same position in the rank orders of the two explanations), sign agreement (common features that also share the same sign in the two ranks) and signed rank agreement (common features sharing the same sign and rank position). For the features of interest, they include rank correlation (computed via Spearman's rho rank correlation coefficient) and pairwise rank agreement (fraction of feature pairs for which the relative ordering is equal within the two explanations).

Robust-by-construction explanations are an interesting area of research [5].

[31] investigates instead how to train NNs to yield more robust explanations: they indicate weight decay, smoother activation function and regularizers are key factors. More complete frameworks are, for example, ROBust Posthoc Explanations (ROPE) [71] and ROBust Algorithmic Recourse (ROAR) [129]. The former one generates explanations which are robust to changes in the input and data distribution (showing an improvement over LIME and SHAP) while the latter one focuses on robust counterfactuals under model changes.

[4] introduces self-explaining neural networks, a class of models for which faithfulness and robustness are enforced by construction through a specific regularization, and thanks to a generalization of the local Lipschitz continuity of Equation 5.3.

5.2 Robustness Estimator

This Section will introduce a set of *desiderata* for the robustness estimator which we have originally proposed in [134].

Let $\hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i) := \hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f)$ be a robustness estimator that measures the robustness of the explanation method e applied to the neighbourhood $\mathcal{N}^i$ of the point $\mathbf{x}^i$ over the model f .

Property 5.2.1. If $r(\mathbf{x}^i, \tilde{\mathbf{x}}^i) := r(\mathbf{x}^i, \tilde{\mathbf{x}}^i, e, f)$ is the robustness of $e(\mathbf{x}^i)$ with respect to the perturbation $\tilde{\mathbf{x}}^i$, then the robustness $\mathcal{R} = \mathbb{E}[r]$ is estimated by:

$$\hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i) = \frac{1}{|\mathcal{N}^i|} \sum_{\tilde{\mathbf{x}}^i \in \mathcal{N}^i} r(\mathbf{x}^i, \tilde{\mathbf{x}}^i) \quad (5.8)$$

where $\mathcal{N}^i = \{\tilde{\mathbf{x}}^i | \tilde{\mathbf{x}}^i = \mathbf{x}^i + \lambda \text{ with } d(\mathbf{x}^i, \tilde{\mathbf{x}}^i) < \epsilon (\epsilon > 0), \lambda \in \mathbb{R}^m\}$

Two points which are close to each other within the dataspace produce explanations that have comparable robustness scores.

Property 5.2.2. If $\hat{\mathcal{R}}$ is a local robustness estimator, then for two distinct points $\mathbf{x}^i, \mathbf{x}^j$ such that $\mathbf{x}^i \neq \mathbf{x}^j$ and $d(\mathbf{x}^i, \mathbf{x}^j) < \epsilon (\epsilon > 0)$ it holds that:

$$\exists \delta > 0 \text{ s.t. } |\hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i) - \hat{\mathcal{R}}(\mathbf{x}^j, \mathcal{N}^j)| < \delta \quad (5.9)$$

Robustness estimation is linked to the uncertainty in the estimates, both due to possible errors in the explanations themselves and the lack of a ground truth for the robustness.

Property 5.2.3. $\hat{\mathcal{R}}$ is such that $\hat{\mathcal{R}} = \mathcal{R} + \theta_\epsilon$ where $\mathcal{R} = \mathbb{E}[r]$ is the true robustness and $\theta_\epsilon \neq 0$ is an error term.

Being a central part of the definition, the neighbourhood generation plays an important role in the robustness estimation process when non-adversarial perturbations are considered. On-manifold perturbations better reflect the true data distribution and the manifold learned by the model, therefore exhibit greater robustness scores than random perturbations which may be off-manifold.

Property 5.2.4. If $\mathcal{N}^i$ is an on-manifold neighbourhood and $\bar{\mathcal{N}}^i$ is an off manifold-one, then $\hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i) > \hat{\mathcal{R}}(\mathbf{x}^i, \bar{\mathcal{N}}^i)$.

The robustness of an aggregation of explainers is bounded by the robustness of the individual components.

Property 5.2.5. If the explainer $\bar{e}$ is an aggregation of explainers, $\bar{e} = \text{agg}(e_1, \dots, e_l)$ then, if $\hat{\mathcal{R}}(e) := \hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f)$, it holds that $\hat{\mathcal{R}}(\bar{e}) \leq \max(\hat{\mathcal{R}}(e_1), \dots, \hat{\mathcal{R}}(e_l))$

Two equivalent models predicting the same class for a datapoint will be related to explanations with comparable robustness scores. The data manifold learned by the models influences the robustness score.

Property 5.2.6. Let $f(\cdot)$ and $g(\cdot)$ be two models with comparable accuracy on the dataset $\mathcal{D}$. If the point $\mathbf{x}^i$ is predicted to belong to the same class by both models, say $\hat{y}_f^i = \hat{y}_g^i$ and let $\hat{\mathcal{R}}(f) := \hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f)$, then:

$$|\hat{\mathcal{R}}(f) - \hat{\mathcal{R}}(g)| < \delta \text{ with } \delta > 0 \quad (5.10)$$

We propose the robustness r be computed as $r(\mathbf{x}^i, \tilde{\mathbf{x}}^i) = \rho(e(\mathbf{x}^i, f), e(\tilde{\mathbf{x}}^i, f))$, where ρ is the Spearman's rho rank correlation coefficient and $e(\mathbf{x}^i, f)$ is the explanation of model f prediction of point $\mathbf{x}^i$. By Property 5.2.1, it then holds that the robustness $\mathcal{R}$ can be estimated via:

$$\hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f) = \frac{1}{|\mathcal{N}^i|} \sum_{\tilde{\mathbf{x}}^i \in \mathcal{N}^i} \rho(e(\mathbf{x}^i, f), e(\tilde{\mathbf{x}}^i, f)) \quad (5.11)$$

where $\mathcal{N}^i = \{\tilde{\mathbf{x}}^i | \tilde{\mathbf{x}}^i = \mathbf{x}^i + \lambda \text{ with } \lambda \in \mathbb{R}^m, d(\mathbf{x}^i, \tilde{\mathbf{x}}^i) < \epsilon (\epsilon > 0) \text{ and } \hat{y}^i = \hat{y}^{\tilde{i}}\}$. By definition, it holds that $0 \leq \hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f) \leq 1$. This robustness estimator satisfies also Properties 5.2.2 to 5.2.6.

5.3 Neighbourhood Generation

In the previous Section, we have limited the constraints over the neighbourhood $\mathcal{N}^i$ to only consider perturbed points $\tilde{\mathbf{x}}^i$ which are close to the original datapoint $\mathbf{x}^i$ and for which the model's prediction is the same, $\hat{y}^i = \hat{y}^{\tilde{i}}$. In the following, we will compare two neighbourhood generating schemes that can influence the robustness computation. Let us first distinguish between the numerical and categorical variable of a point $\mathbf{x}^i$ with $x^{(i,num)}$ and $x^{(i,cat)}$ respectively, as perturbations act differently according to the data type of the variable.

5.3.1 Random Neighbourhood ($\mathcal{N}_R$)

A naive approach to neighbourhood generation is the random generation, in which white noise is added to the numerical variables while categorical ones are flipped at random.

$$\begin{cases} \tilde{x}^{(i,num)} &= x^{(i,num)} + \delta^i \text{ with } \delta^i \leftarrow \mathcal{N}(0, \sigma^2) \\ \tilde{x}^{(i,cat)} &= \text{flip}(x^{(i,cat)}) \text{ with probability } \gamma_{cat} \end{cases} \quad (5.12)$$

The flip of a categorical variable entails a random sampling among the possible modalities associated with the variable, the observed value of $x^{(i,cat)}$ being excluded.

5.3.2 Medoid-based Neighbourhood ($\mathcal{N}_M$)

In [134] we have introduced a medoid-based neighbourhood generating scheme which leverages the manifold hypothesis to generate perturbations which are still on-manifold. This is crucial as our aim for robustness estimation is to test it against non-adversarial perturbations, which therefore have to be consistent with the observed data distribution.

Consider a k -medoid clustering on $\mathcal{D}_{valid}$, with $k_{medoids}$ selected such that, on average, each cluster is of size $n_k = 10$. For each cluster c , the medoid $\mathbf{x}^c$ acts as a representative observation and its k_M nearest neighbours among the other cluster centres are stored in the set $NN^c = (\mathbf{x}^1, \dots, \mathbf{x}^{k_M})$. Let $\mathbf{x}^i$ be a datapoint for which a perturbation $\tilde{\mathbf{x}}^i$ must be computed. First, let c be the cluster associated with $\mathbf{x}^i$ and let NN^c be the list of nearest medoids associated to the cluster. The generation of the perturbation $\tilde{\mathbf{x}}^i$ begins by selecting one of the medoids at random from this set, say $\mathbf{x}^M \in NN^c$. Having then α and α_{cat} the probabilities of perturbing a numerical and a categorical variable respectively, the perturbation is performed following:

$$\begin{cases} \tilde{x}^{(i,num)} &= (1 - \bar{\alpha}) \cdot x^{(i,num)} + \bar{\alpha} \cdot x^{(M,num)} \text{ with } \bar{\alpha} \leftarrow \text{Beta}(\alpha \cdot 100, (1 - \alpha) \cdot 100) \\ \tilde{x}^{(i,cat)} &= \begin{cases} x^{(i,cat)} & \text{with probability } 1 - \alpha_{cat} \\ x^{(M,cat)} & \text{with probability } \alpha_{cat} \end{cases} \end{cases} \quad (5.13)$$

For both generating schemes, the resulting neighbourhood should be of size at least $n = 100$ to ensure statistical significance. A filtering step is performed after $\mathcal{N}^i = (\tilde{\mathbf{x}}^1, \dots, \tilde{\mathbf{x}}^n)$ are generated, removing all the perturbations for which the model prediction is different with respect to that of the original datapoint, i.e. $f(\tilde{\mathbf{x}}^i) \neq f(\mathbf{x}^i)$. Hyperparameter tuning is performed on $\theta_1 = (\sigma, \gamma_{cat})$ and $\theta_2 = (\alpha, \alpha_{cat}, k_M)$ to ensure that, on average, around 95% of the datapoints are kept.

A comparison between the two neighbourhoods is presented in Figure 5.1, where the Swiss Roll dataset is used as an example. Both approaches were applied to the same datapoint $\mathbf{x}^i$: the left most column represents a 3D visualization of the dataset, which has the shape of a rolled piece of paper with a small degree of "thickness" to it, achieved by adding a small amount of random noise. The middle column is a view from above while the right-most one represents a zoomed-in visualization of the neighbourhoods $\mathcal{N}_R$ and $\mathcal{N}_M$. It is quite evident from the comparison between top and bottom rows that the random neighbourhood $\mathcal{N}_R$ produces perturbations $\tilde{\mathbf{x}}^i$ which lie outside of the data manifold, even when the perturbation parameters are small ($\sigma = 0.05$). On the other hand, the medoid-based neighbourhood $\mathcal{N}_M$ is able to remain within the bounds, even if a larger interpolation coefficient is being used ($\alpha = 0.30$). The points in the bottom row depicted in orange correspond to the $k_M = 5$ nearest medoids of the cluster, which in this case coincide with the perturbation directions over which the datapoints are being interpolated.

5.4 Results

5.4.1 Dataset and Model Description

This subsection introduces the datasets used in the remainder of this thesis. Specifically, we selected the following publicly-available datasets from the UC Irvine Machine Learning Repository⁵: **beans**, **cancer**, **mushroom**, **white wine**, **adult** and **bank marketing**. We have additionally worked on the **heloc** and **ocean** datasets,

⁵<https://archive.ics.uci.edu/datasets>

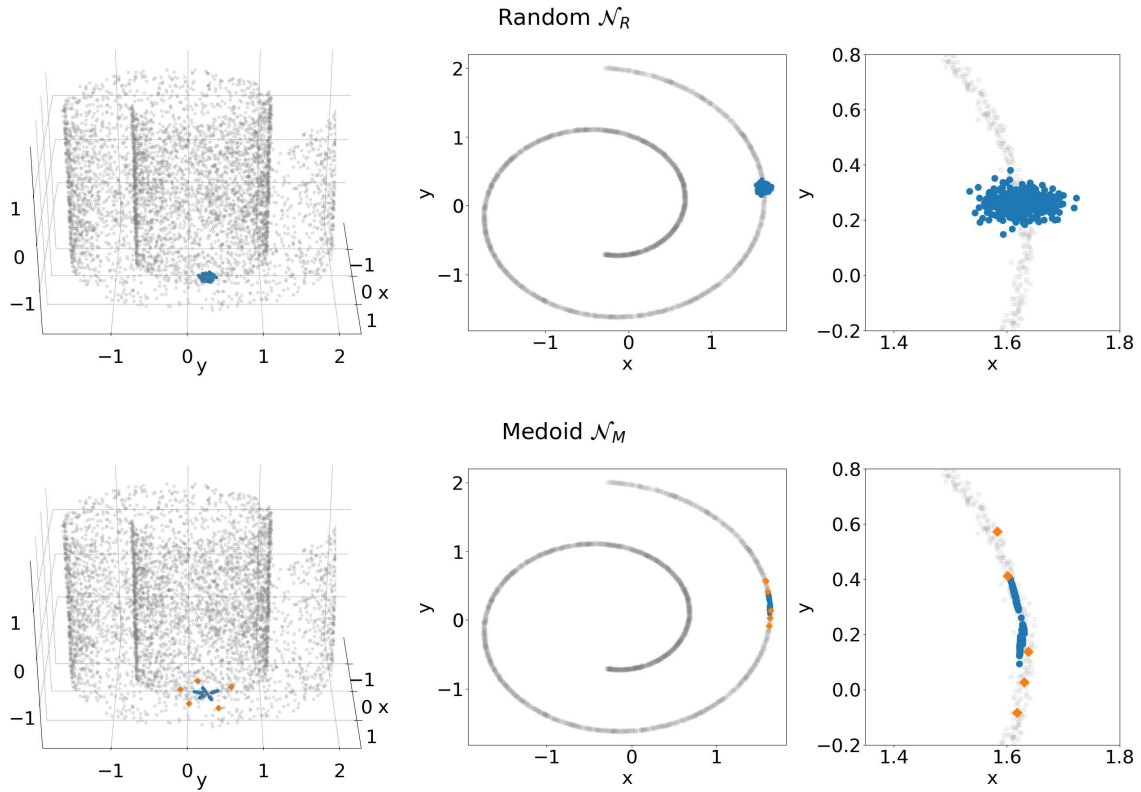


Figure 5.1: An example of neighbourhood generation ($n = 500$) on the Swiss roll dataset: the top row depicts the random neighbourhood generation $\mathcal{N}_R$ with $\sigma = 0.05$ while the bottom one the medoid-based generation $\mathcal{N}_M$ with $\alpha = 0.30$ and $k_M = 5$. Blue datapoints represent the generated perturbations while the orange ones in the bottom row are the k_M neighbouring cluster centres.

following [92] and [118] respectively. The first four are toy examples, showcasing simpler classification tasks that can be solved even with non-neural models and with an overall lower number of both features and datapoints. All the eight datasets propose binary or multiclass classification problems and contain both numerical and categorical variables. A brief overview of the datasets can be found in Table 5.1.

The following preprocessing steps were performed:

- Standardization of numerical variables and one-hot encoding of the categorical ones.
- Removal of correlated features. Spearman’s rho rank correlation coefficient was used for numerical variables while the Normalized Mutual Information criterion (NMI) for the categorical ones. This step is necessary in order to ensure that generated perturbations are more akin to the observed data distribution, and also that the computed attributions are non distorted by unconsidered correlations existing among the features.

For each dataset $\mathcal{D}_j$, a neural network f_j was trained on the training set. The nets are fully-connected and have ReLU activations, with the exception of the one applied to the `ocean` dataset in which the tanh activation was used, as in [118]. In

Table 5.1: Dataset details: number of output classes, number of numerical and categorical variables, train, validation and test sizes as well as the number of k -medoids clusters used.

Dataset	Classes	#Num	#Cat	$\mathcal{D}_{train}$	$\mathcal{D}_{valid}$	$\mathcal{D}_{test}$	$k_{medoids}$
beans	7	7	0	10'888	2'223	500	225
cancer	2	15	0	397	121	50	10
mushroom	2	0	21	6'498	1'225	400	120
white wine	2	9	0	3'918	780	200	80
adult	2	5	7	36'177	8'045	1'000	1'000
bank	2	5	9	36'168	8'043	1'000	1'000
heloc	2	14	2	8'367	1'592	500	130
ocean	6	8	0	109'259	30'328	10'000	3'000

all networks, the softmax is used in the final layer - even in the binary classification examples. This choice is governed by the necessity of applying a set of attributions to the output of the network. The use of the softmax in the output layer reduces the effects of the vanishing gradient problem, that is, the scenarios in which the attributions are returned as zero-vectors. In these cases, the signal from the output neuron which is being explained is lost when the output score is propagated through the net.

5.4.2 Neighbourhood Generation and Robustness Evaluation

We evaluated the robustness of seven XAI approaches applicable to neural networks⁶: LIME (Section 4.1), SHAP (Section 4.2), Saliency, Input X Gradients (Section 4.3), LRP (with the gamma rule, the one minimizing the vanishing gradient problem - Section 4.4), DeepLIFT (Section 4.5) and Integrated Gradients (Section 4.6).

As discussed in Section 5.3, we consider two possible neighbourhood generating schemes: the random and the medoid-based neighbourhoods. Hyperparameters were selected in a dataset-specific manner with respect to the output of the neural network f_j , and are summarized in Table 5.2. For the medoid-based neighbourhood generation, the number of k -medoid clusters identified on the validation set is specified in Table 5.1 and yields clusters with an average size of $n_k = 10$. A dataset-specific distance metric was used in the clustering step to correctly capture variable heterogeneity: the Gower distance (Subsection 2.2.1) was used for all datasets except the mushroom one, where the Hamming one was selected. Each neighbourhood is constructed with $n = 100$ datapoints before the final filtering step is applied. Average neighbourhood sizes after filtering can be found in Table 5.2, confirming that hyperparameter tuning was performed in a satisfactory manner, as most of the average neighbourhood sizes are greater than the 95.00% mark.

The robustness score in Equation 5.11 is then computed for each datapoint in the test set $\mathcal{D}_{(j,test)}$ over both neighbourhoods.

⁶The used code can be found at https://github.com/ilariavascotto/XAI_robustness_analysis

Table 5.2: Hyperparameter selection for the Random and Medoid-based neighbourhood generation schemes, along with average neighbourhood size per dataset, $|\mathcal{N}|$.

Dataset	Random $\mathcal{N}_R$			Medoid-based $\mathcal{N}_M$			
	σ	γ_{cat}	$ \mathcal{N} $	α	α_{cat}	k_M	$ \mathcal{N} $
beans	0.02	-	94.94	0.10	-	10	98.32
cancer	0.10	-	92.56	0.10	-	4	95.50
mushroom	-	0.15	95.89	-	0.15	10	95.96
white wine	0.03	-	96.48	0.15	-	5	97.32
adult	0.05	0.05	93.55	0.05	0.05	5	95.87
bank	0.05	0.10	95.09	0.05	0.10	5	96.28
heloc	0.03	0.10	89.57	0.05	0.05	5	94.97
ocean	0.001	-	96.55	0.05	-	5	98.23

A note on categorical variables. A common practice in ML applications is to preprocess categorical features with a one-hot encoding before a neural network is trained. For example, let $x^{(\cdot, cat)}$ be a categorical feature that can assume four modalities and let $(0, 0, 1, 0)$ be its one-hot encoding vector, where the non-zero entry corresponds to the observed modality (the third one in this example). A feature attribution method applied to the one-hot encoded feature vector returns a set of importances of the form $(0, 0, v, 0)$: the non-zero entry is the only one associated with a non-null value v . This is to be expected as the missingness property (Property 4.2.2) holds for the methods under consideration.

Considering that one-hot encoding can increase the size of the input vector from m to m' (with, potentially, $m' \gg m$), we aim at limiting the effects of zero-entries in the Spearman’s rho computation of Equation 5.11 via a *reverse encoding* of categorical variables. They can be represented as a whole by the observed modality such that the variable $x^{(\cdot, cat)}$ is associated with an attribution $a_l^{(\cdot, cat)} = v$. This allows us to consider feature vectors of size m (instead of m'), aligning with the shape of the original feature vector, and to perform a more effective comparison of the robustness. The *reverse encoding* should be applied when categorical variables are present before the robustness computation.

Results As stated in Section 5.2, it holds by definition that $0 \leq \hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f) \leq 1$. Values closer to 1 are preferable as they indicate greater robustness of the explanation, while lower values are symptomatic of explanation’s instability.

Tables 5.3 and 5.4 present the average robustness scores (expressed in percentage) for the seven XAI approaches over the eight datasets, both with a random (Table 5.3) and medoid-based (Table 5.4) neighbourhoods. In both tables, values in blue indicate the maximum robustness achieved dataset-wise, while the orange ones the minimum. The bottom row depicts the average robustness achieved by the individual XAI methods over all the dataset.

LIME and SHAP are - on average - the least robust methods in both scenarios. This is to be expected as they are model-agnostic approaches while the remaining methods are tailored to neural networks. Saliency and Input X Gradients show an intermediate behaviour, being extremely robust for some of the datasets but

Table 5.3: Average robustness scores (%) on the test sets, for each dataset and XAI method, with random neighbourhood. Values in **blue** represent the maximum dataset-wise while those in **dark orange** the minimum. The last row represents the average robustness scores achieved by each method over all the datasets.

Dataset	LIME	SHAP	Saliency	IxG	LRP	DeepLIFT	IG
beans	55.63	39.32	71.02	66.43	76.88	88.19	81.94
cancer	6.44	14.07	91.72	76.58	56.32	77.45	78.94
mushroom	0.08	7.17	98.98	63.11	63.85	61.80	64.40
white wine	7.21	6.06	8.03	11.02	61.60	21.56	81.61
adult	4.62	11.84	1.08	0.40	76.73	75.87	60.82
bank	4.74	9.94	82.62	65.77	74.84	74.50	77.39
heloc	12.72	8.48	12.82	10.58	59.36	67.44	63.70
ocean	57.73	43.99	86.07	82.76	51.06	86.16	87.48
average	18.65	17.61	56.54	47.08	65.08	69.12	74.54

Table 5.4: Average robustness scores (%) on the test sets, for each dataset and XAI method, with medoid-based neighbourhood. Values in **blue** represent the maximum dataset-wise while those in **dark orange** the minimum. The last row represents the average robustness scores achieved by each method over all the datasets.

Dataset	LIME	SHAP	Saliency	IxG	LRP	DeepLIFT	IG
beans	60.26	41.94	89.44	90.53	86.81	84.79	95.93
cancer	16.14	17.24	98.26	97.98	72.30	98.45	98.52
mushroom	0.10	7.29	99.01	63.74	64.24	62.36	65.61
white wine	7.67	6.41	16.80	1.44	69.79	19.09	90.83
adult	5.23	13.55	2.03	1.64	85.27	84.86	76.26
bank	4.80	10.34	89.22	75.14	79.41	78.76	80.98
heloc	14.11	9.31	24.71	24.34	76.44	84.14	81.40
ocean	58.24	45.86	94.46	94.56	70.23	93.19	94.03
average	20.82	18.99	64.24	58.42	75.56	75.71	85.44

failing with others robustness-wise. In particular, it seems that Saliency excels in simpler datasets (namely cancer and mushroom) but struggles with more complex ones (notably, the adult dataset). A more consistent and desirable behaviour is presented by LRP, DeepLIFT and Integrated Gradients. These methods are, in fact, associated to the three highest average robustness scores over the datasets, for both random and medoid-based neighbourhoods. The greater robustness of these approaches can be appreciated also in Figure 5.2, where the distribution of the robustness scores computed on the test set is presented for both the adult (left panel) and heloc (right panel) datasets. The robustness threshold of 0.50 is depicted by a dashed grey vertical line. It can be seen that, in both scenarios, LIME and SHAP showcase a poor performance while LRP, DeepLIFT and Integrated Gradient are the most robust methods under investigation. Moreover, Saliency and Input X Gradient present a very similar behaviour (the blue dotted line is almost superimposed over the light red one). This is to be expected as Input X Gradient is derived from Saliency and simply modifies its explanation by multiplying it with the input vector.

Notably, as stated by Property 5.2.4, the robustness computed on an on-manifold neighbourhood is greater than that computed on an off manifold one. In fact, the vast majority (96.43%) of the average robustness scores for all methods and datasets is greater in the case of the medoid-based neighbourhood with respect to the random one. This can be further appreciated by the results from Figure 5.3, where the distribution of the robustness scores computed on the test set for the adult datasets are presented. The solid blue lines represent the medoid-based neighbourhood while the dashed red ones the random version. The upper four panels are limited to the $(0.0, 0.5)$ range for readability purposes, as scores above 0.50 are not recorded for these methods. It can be seen that the two robustness scores present a similar behaviour, with the medoid-based one being marginally advantageous. More evidently, the second row presents larger differences between the two approaches, with the medoid-based neighbourhood producing greater robustness scores for the three methods.

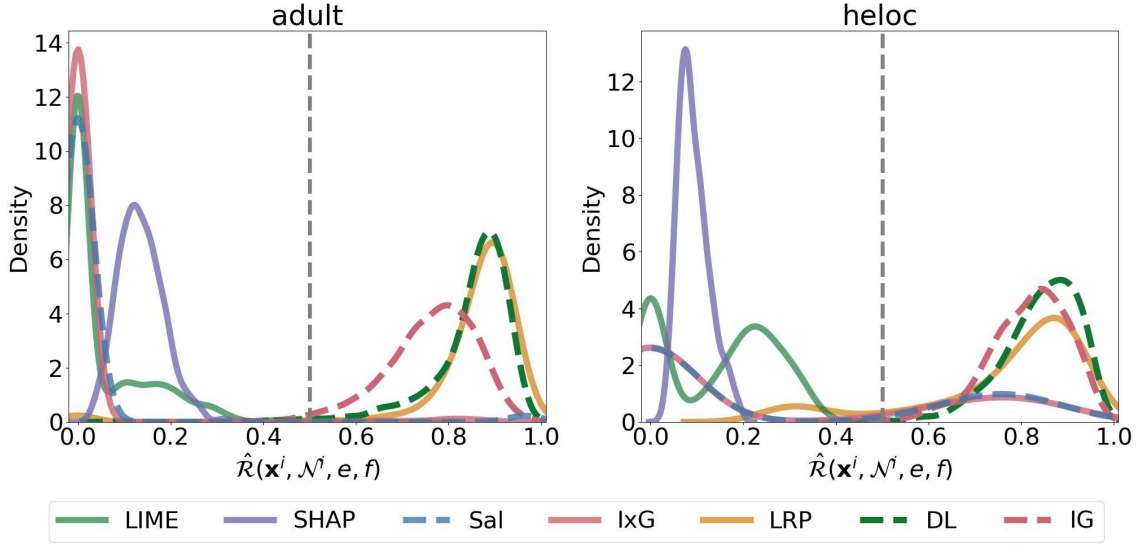


Figure 5.2: The robustness scores for the adult (left) and heloc (right) datasets for the seven XAI methods with the medoid-based neighbourhood. The gray dashed line represents the 0.50 robustness threshold.

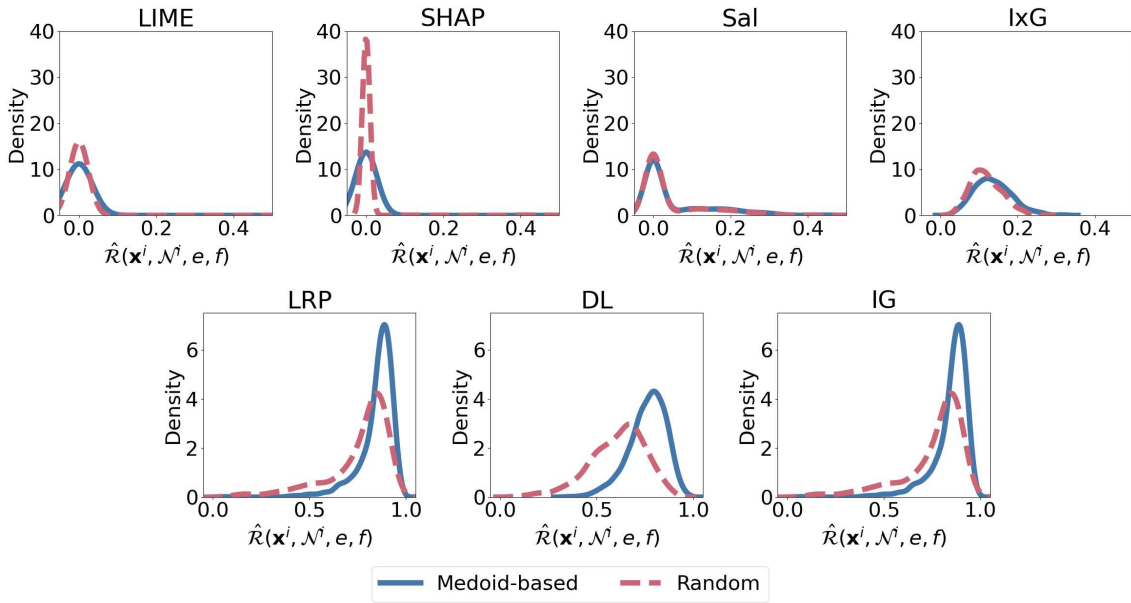


Figure 5.3: A comparison between the robustness scores derived from a medoid-based neighbourhood (solid blue) and random one (dashed red), for the seven XAI methods on the adult dataset. The first row is limited within the (0.0, 0.5) range for readability, while the second row encompasses the full range (0.0, 1.0).

Chapter 6

Aggregating Explanations

This Chapter will introduce the notion of explanation aggregation (or *ensemble* of explanations) and its connection to robustness. In this Chapter we will discuss the reasoning behind aggregation as a trustworthy approach, as well as a solution to the disagreement problem [69]. We will discuss in details single-model aggregations, that is, the aggregation of explanations or XAI techniques applied to the same model, while Chapter 9 will focus on the aggregation of explanations derived from different predictive models.

The main difference can be defined as follows:

- Single-model Aggregation: let f be a predictive model and let $\{\mathbf{a}_l\}_{l=1}^L$ be L attribution methods applied to it. A single-model aggregation is then defined as $\mathbf{a}_{agg} = g(\mathbf{a}_l|f)$, a function of the individual attributions over the model f .
- Multi-model Aggregation: let $f_1, \dots, f_L$ be L predictive models, each associated with a corresponding attribution method $\mathbf{a}_l$. A multi-model aggregation is then defined as $\mathbf{a}_{agg} = g(\mathbf{a}_l|f_l)$, a function of the individual attributions applied to the corresponding model.

The contents of this Chapter will be strictly linked to those discussed in Chapter 7, as they lie within the single-model aggregation setting. On the other hand, the multi-model aggregation of Chapter 9 is strongly connected to the topic of Chapter 8 and will therefore be discussed in the following.

6.1 Related Work

6.1.1 The disagreement problem

The disagreement problem in XAI posits that multiple explanation methods, applied to the same datapoint and model, may produce explanations which are contrasting either in sign, magnitude or feature rankings. This naturally raises trustworthiness questions: firstly, the contrasting results may hinder the usage of XAI approaches in their entirety (how can you trust the explanations if they differ significantly?) and secondly, may require a rigorous methodology to being able to choose one explainer over the others. Practical disagreements that may be encountered are sign disagreement or rank disagreement. Sign disagreement refers to the cases in which a feature

j has positive sign for the explainer e_l and a negative one for $e_{l'}$, while the rank disagreements refers to the cases in which the same feature is considered more or less important relative to the other features under evaluation. Both scenarios give rise to possible contrasting results and affect the reliability of the explanations.

[69] presents an overview of the disagreement problem according to a user study with 25 participants from both academia and industry. While it is underlined that the problem is of non-negligible size, the authors do not propose solutions or good practices to limit its undesirable effects. Critically, the results from the study underline poor practices from the practitioners themselves, such as choosing the results of one method they particularly like with an apparent lack of objective evaluations.

6.1.2 Aggregations

The role of aggregations in Machine Learning and the use of ensemble approaches have been extensively studied. The aggregation of multiple weak learners is, in fact, a popular technique in ML literature. For example, tree-based ensembles such as the Random Forest (Subsection 2.2.2) are based on this idea, which increases predictive power while reducing model variance. This approach can naturally be extended to other domains. The following will provide a brief overview on aggregating explanations.

[100] presents a first step towards explanation aggregation, aimed at aiding the end-users in interpreting explanation disagreements. Specifically, they focus on three agreement metrics: Feature Agreement (FA), Rank Agreement (RA) and Sign Agreement (SA). Results show that it may be sufficient to evaluate the sign disagreement, specifically addressing the aggregated set:

$$A = \{j \in \mathcal{S} : \text{sign}(e_l(\mathbf{x})^{(\cdot,j)}) = \text{sign}(e_{l'}(\mathbf{x})^{(\cdot,j)})\} \quad (6.1)$$

with $\mathcal{S} = \text{top}k(e_l(\mathbf{x})) \cap \text{top}k(e_{l'}(\mathbf{x}))$

The set $\mathcal{S}$ represents the set of features which are among the top k for both explainers $e_l, e_{l'}$. The features which are not included in the set A are those over which the two attribution approaches disagree: the authors suggest that these should not be shown to the user, reducing the perception of explanation disagreement. This approach was only tested via a comparison of LIME [96] and SHAP [77].

AGREE (AGgregation for Robust Explanations) [94] is a feature attribution aggregation framework that is built to tackle the disagreement problem. It focuses on the ensemble of different explainers and works on the ranks of the derived attributions. The key idea of AGREE is weighting the feature attribution ranks, $\mathbf{r}_l$, based on the level of confidence of each explainer, resulting in a more robust explanation. The confidence level of each explainer is derived from pairwise alignment scores of the L composing explainers, specifically addressing the disagreement between each pair of explainers. Formally, having $M \in \mathbb{R}^{L \times L}$ a pairwise alignment matrix, the explainer confidence vector A is defined as:

$$A_l = \begin{cases} \frac{1}{L} \sum_{l'=1}^L M[l, l'] & \text{if } M \text{ is symmetric} \\ \frac{1}{2L} \sum_{l'=1}^L (M[l, l'] + M[l', l]) & \text{otherwise} \end{cases} \quad (6.2)$$

The aggregation is then computed as:

$$AGREE(A, \mathbf{r}) = \bar{\omega}^{(g)} = \frac{1}{L} \sum_{l=1}^L \bar{\omega}_l^{(g)} \quad \text{with} \quad \bar{\omega}_l^{(g)} = \frac{\sum_{l'=1}^L (A_{l'} \cdot \mathbf{r}_l)}{\sum_{l'=1}^L A_{l'}} \quad (6.3)$$

where the notation $\bar{\omega}_l^{(g)}$ refers to the weighted attributions obtained via an alignment function g as the basis for the confidence scores in the vector A for the l -th explainer.

Given a dataset $\mathbf{X}$, it is also possible to derive a global explanation as the average of the aggregated explanations over the dataset.

A practical application of ensemble methods in the medical field [146] showcases the advantage of merging heatmaps applied to chest X-ray images. The images are manually annotated by radiologists and SHAP [77] and Grad-CAM++ [110] are used to generate the heatmaps. These are then combined through an EnsembleXAI via stacking. Specifically, Kernel Ridge regression is applied to the normalized Grad-CAM++ and normalized positive SHAP values to generate a mapping layer which identifies the useful features. A user study performed with radiologists showed that the EnsembleXAI approach was deemed the most trustworthy (mean vote 70.2%).

[13] introduces the InXAI framework: it builds an ensemble of explanations starting from a set of desirable metrics such as consistency, stability and area under the loss curve. In practice though, any metric can be used in the computation of the ensemble. The ensemble is built as a weighted average of the individual explanations, weighted by an ensemble score, defined as:

$$ES(M, \omega) = \sum \omega_i \cdot M_i \quad (6.4)$$

where ω_i are a set of weights and M_i is a metric of choice. The weights capture the importance of the individual metrics and are assigned arbitrarily depending on the user preferences over the possible metric. The ensemble is then computed as:

$$e_{ens} = \frac{ES(M, \omega) \cdot [\gamma_1 e_1, \dots, \gamma_L e_L]}{\sum_{l=1}^L ES_l(M, \omega)} \quad (6.5)$$

where γ_l are scaling factors necessary to compare different XAI explanations. Generally, the scaling could be performed with min-max normalization as in [13]. Unfortunately, the framework is presented on model-agnostic explainers only, namely LIME [96], SHAP [77] and ANCHORS [97].

Aggregations can also be performed leveraging multiple explanations from the same method: [12] derives a more robust Shapley-value explanation by aggregating the explanations computed on a carefully crafted neighbourhood, minimizing explanation sensitivity. The authors prove that convex combinations of explanations yield lower sensitivity with respect to individual methods and that an explanation average reduces errors. Moreover, they introduce an aggregation which combines the Shapley values from a set of nearest neighbours, weighted by the inverse distance. This step allows to maintain high faithfulness within the explanation.

[48] investigates the role of an Ensemble of Counterfactual Explainers (ECE) to cover a set of *desiderata*. These are usually achieved by individual explainers through optimization problems focusing on only one of the desired properties. The ensemble

approach is instead able to boost the individual weak explainers, combining their results via a diversity-driven selection function. A set $\mathcal{C}$ of counterfactuals from the L base counterfactual explainers is obtained. Each base explainer is invoked on a sample without replacement of the instances $\mathbf{X}$ and a random subset $\mathcal{A}'$ of actionable features is selected, as in random forest, ensuring diversity. Actionable features in the context of counterfactual explanations are the set of features which can be changed by the counterfactual. An example of non-actionable features are age, gender and race.

Starting from the set $\mathcal{C}$, the selection of k -counterfactuals is performed. The set of final explanations, $\mathcal{S}$, is found by maximising the following density-based objective function:

$$\arg \max_{\mathcal{S} \subseteq \mathcal{C} \wedge |\mathcal{S}| \leq k} \left| \bigcup_{\mathbf{c} \in \mathcal{S}} knn_{\mathcal{C}}(\mathbf{c}) \right| - \lambda \sum_{\mathbf{c} \in \mathcal{S}} d(\mathbf{c}, \mathbf{x}) \quad (6.6)$$

where $knn_{\mathcal{C}}(\mathbf{c})$ returns the k most similar counterfactuals to $\mathbf{c}$ within the complete set $\mathcal{C}$ and d is an appropriate distance metric that measures the similarity with respect to the datapoint $\mathbf{x}$. The Cost Scaled Greedy algorithm is used for the maximization. Experimental results show that this approach excels at providing diverse counterfactuals that satisfy the selected *desiderata*.

To conclude, in the context of adversarial attacks, [98] showcases the efficacy of ensemble methods as defences on explanations. They test their aggregated explanation (the average of normed explanations) on image data and show that it is more resilient to attacks, when a composing explanations method or the model itself is being fooled.

6.2 Single-Model Aggregation

This Chapter focuses on the aggregation of multiple XAI approaches applied to the same *black box* model. In the following we will focus on neural networks as model of choice and will consider local model-specific XAI approaches.

6.2.1 Ensemble Approach

We presented in [134] a novel aggregation approach for neural network's local explanations which merges feature importances and focuses on the ranking of the features.

Let $\mathbf{a}_l^i$ be the feature importance vector derived from applying the explanation method e_l on point $\mathbf{x}^i$. Let L be the number of methods which are being aggregated and let $\mathbf{r}_l^i$ be the l -th ranking, a vector of size m that stores the indices that would sort the array $|\mathbf{a}_l^i|$ in decreasing order, i.e. $\mathbf{r}_l^i = (r_l^{(i, \tau_1)}, \dots, r_l^{(i, \tau_m)})$ where $\boldsymbol{\tau} = (\tau_1, \dots, \tau_m)$ is a reordering of the indices $j \in \{1, \dots, m\}$ such that $|a_l^{(i, \tau_j)}| \geq |a_l^{(i, \tau_{j+1})}|$.

Then, let an average attribution $\mathbf{a}_{ens}^i$ be defined by:

$$a_{ens}^{(i, j)} = \frac{\sum_{l=1}^L r_l^{(i, j)} \cdot w_l^{(i, j)}}{\sum_{l=1}^L w_l^{(i, j)}} \cdot (1 + \lambda \bar{n}^{(i, j)}) \quad (6.7)$$

where $\mathbf{w}_l$ is the weight vector corresponding to the l -th feature attribution vector, $\lambda = 0.15$ is a penalization term and $\bar{n}^{(i,j)}$ is the number of methods for which there is a disagreement on the sign of the attribution $a_l^{(i,j)}$ for $l = 1, \dots, L$. The ranking $\mathbf{r}_{ens}^i$ storing the indices that would sort the array $\mathbf{a}_{ens}^i$ in increasing order is the feature ranking resulting from the proposed approach. Formally, $\mathbf{r}_{ens}^i = (r_{ens}^{(i,\pi_1)}, \dots, r_{ens}^{(i,\pi_m)})$ where π is a reordering of the indices $j \in \{1, \dots, m\}$ such that $a_{ens}^{(i,\pi_j)} \leq a_{ens}^{(i,\pi_{j+1})}$.

The aggregation defined by Equation 6.7 tackles practical issues that emerge when jointly considering multiple explanations. One of the first notable issues is that attributions may present coefficients which are very close, but not equal, to zero. For the missingness property (Property 4.2.2), it holds that a zero coefficient $a_l^{(i,j)} = 0$ implies that the feature $x^{(i,j)}$ is not important towards the prediction. Nonetheless, the lack of a common scale for feature attributions influences the ability to discriminate between important and unimportant features while taking into account only the absolute value of the corresponding coefficients. Case-specific thresholds may vary significantly according to the model at hand and the class which is being explained.

This issue is taken into account in the ensemble aggregation by using the following weighting scheme where, for the l -th method, i -th point and j -th feature, it holds:

$$w_l^{(i,j)} = \frac{\sigma(\mathbf{a}_l^i)}{\sqrt{|\bar{\mathbf{a}}_l^i| \cdot |a_l^{(i,j)}|}} \quad (6.8)$$

where $\mathbf{a}_l^i$ is the l -th feature importance vector, $\bar{\mathbf{a}}_l^i$ its average and $\sigma(\mathbf{a}_l^i)$ the standard deviation.

With this choice of weights, smaller values of the feature importance coefficients are associated to larger weights, effectively contributing to moving a feature into the set of non-informative ones in the ranking (as the largest indices are associated with non-informative features). Considering that the feature attributions may differ greatly in magnitude between multiple methods, a normalization is introduced with the consideration of the average and standard deviation within the weights.

The ensemble is computed as the weighted average of the L rankings, and a penalization term λ is introduced to penalize the features for which the L methods exhibit sign disagreement, as it can be considered a symptom of explanation instability. The penalization furthermore favours features for which there is sign concordance ($\bar{n}^{(i,j)} = 0$), either positive or negative, and when the attribution magnitude is non negligible. As said, considering that larger weights move features towards the set of non-influential ones, the penalization contributes to lower attribution values for concordant features, capturing both their (absolute) relevance and importance towards the prediction.

6.2.2 Mean Aggregation

A baseline aggregation against which the ensemble approach will be tested (Section 6.3) is the mean aggregation. It consists in the arithmetic average of the attributions, assuming that the vectors have been previously normalized to have norm 1. Having

$\mathbf{a}_l^i$ such that $\|\mathbf{a}_l^i\|_2 = 1$, then:

$$a_{mean}^{(i,j)} = \frac{1}{L} \sum_{l=1}^L a_l^{(i,j)} \quad (6.9)$$

with $\mathbf{a}_{mean}^i$ such that $\|\mathbf{a}_{mean}^i\|_2 = 1$.

One of the disadvantages of the mean aggregation is that it may assign a zero attribution even when the L individual coefficients are non-null. Assume for example that, for the j -th feature, $L = 3$ coefficients derived from as many methods are $(\nu, \nu, -2\nu)$ with $\nu > 0$. The mean will be equal to zero, implying that the feature is non relevant towards the prediction, but the importance for each of the L methods indicates a relevance of magnitude at least ν .

6.3 Results

Starting from the analysis presented in Subsection 5.4, we have selected $L = 3$ methods to be aggregated as LRP, DeepLIFT and Integrated Gradients (of which a theoretical introduction is presented in Subsections 4.4, 4.5 and 4.6). They have been shown to be the most robust XAI approaches under consideration and therefore act as a strong baseline for the aggregation. As the aim is constructing explanations which are meaningful and robust, it only comes as a natural choice the selection of robust methods to start with. Nevertheless, both aggregations can be easily extended to include a higher number of feature attribution methods.

6.3.1 Qualitative Evaluation

Table 6.1 presents the average percentage of top- k features (from the attributions taken in absolute value) which are within the top- k features of the Ensemble. $k = \{3, 5\}$ are chosen as they allow for comparability between datasets with a varying number of features. Overall, the concordance between the ensemble's top- k features and the individual approaches is significantly high, underlining how the ensemble is able to correctly take into account the most important features as signalled by its composing approaches. Top- k concordance is an important metric of validation for this scenario as the ensemble is built on the absolute ranking of the features.

Average top- k stability can be evaluated for the ensemble also on the neighbourhoods, to test the concordance of the approach over small datapoint perturbations. The following includes the average concordances over the whole test set with a medoid-based neighbourhood:

- $k = 3$ - beans 62.2, cancer 65.5, mushroom 74.8, wine 67.8, adult 97.0, bank 85.3, heloc 85.5 and ocean 84.0
- $k = 5$ - beans 89.5, cancer 74.7, mushroom 80.5, wine 86.2, adult 95.8, bank 90.7, heloc 87.5 and ocean 94.1

These results confirm that the ranking is preserved within the neighbourhood, but a robustness analysis could be better suited to understand the quality of the proposed approach.

Table 6.1: Percentage of top- k features (in absolute value) of the individual methods within the top- k features of the ensemble, averaged over the test set.

Dataset	$k = 3$			$k = 5$		
	LRP	DL	IG	LRP	DL	IG
beans	83.3	100.0	66.7	100.0	90.0	90.0
cancer	50.0	66.7	83.3	90.0	80.0	70.0
mushroom	83.3	83.3	83.3	80.0	90.0	90.0
white wine	33.3	50.0	33.3	70.0	90.0	60.0
adult	100.0	100.0	66.7	90.0	90.0	80.0
bank	100.0	100.0	83.3	80.0	100.0	100.0
heloc	66.7	66.7	83.3	70.0	50.0	70.0
ocean	66.7	100.0	83.3	70.0	80.0	90.0

6.3.2 Robustness Analysis

A robustness analysis on the two aggregation is proposed in the following ⁷. For the sake of comparability, as the Ensemble is constructed on the absolute rankings, the results presented for its composing methods will be computed on the feature attributions in absolute value. For the Mean aggregation instead the approaches will be left with the computed signs. Difference between the columns of LRP, DeepLIFT (DL) and Integrated Gradients (IG) in Tables 6.2 and 6.3 are due to this intrinsic aspect of the evaluation.

Figure 6.1 depicts the robustness scores derived from the ensemble (left) and mean (right) aggregation with a medoid-based generation mechanism on the adult dataset. Hyperparameters for the neighbourhood generation can be found in Table 5.2. As stated by Property 5.2.5 for the robustness estimator, in both scenarios the aggregated explanations robustness acts as an average of the robustness of its composing approaches, and is limited by the observed span. For the adult dataset, Integrated Gradients is, on average, the least robust method while DeepLIFT and LRP showcase greater values of the robustness score. This behaviour though is dataset-dependant: Integrated Gradients is not, in general, the least robust method.

Table 6.2 presents the average robustness scores achieved by the three approaches and the aggregation for the ensemble (left) and mean (right) over the eight datasets. Values in bold indicate the largest average robustness achieved dataset-wise in the two settings. As suggested by Figure 6.1, the average robustness score of the ensemble is in between the observed values of the other techniques. Moreover, the largest robustness is achieved by dataset-dependant methods, with DeepLIFT being preferable in more than half of the instances.

The advantage of the aggregated explanation, being either the ensemble or the mean, is that they acts as conservative explanations, taking into account the individual method’s robustness and their feature-wise agreement. While the average robustness may be lower than that of the composing approaches, the aggregation is able to take into account the possible undesirable effects of a less robust and disagreeing approach, flagging possible untrustworthiness for a given datapoint.

⁷The used code can be found at https://github.com/ilariavascotto/XAI_robustness_analysis

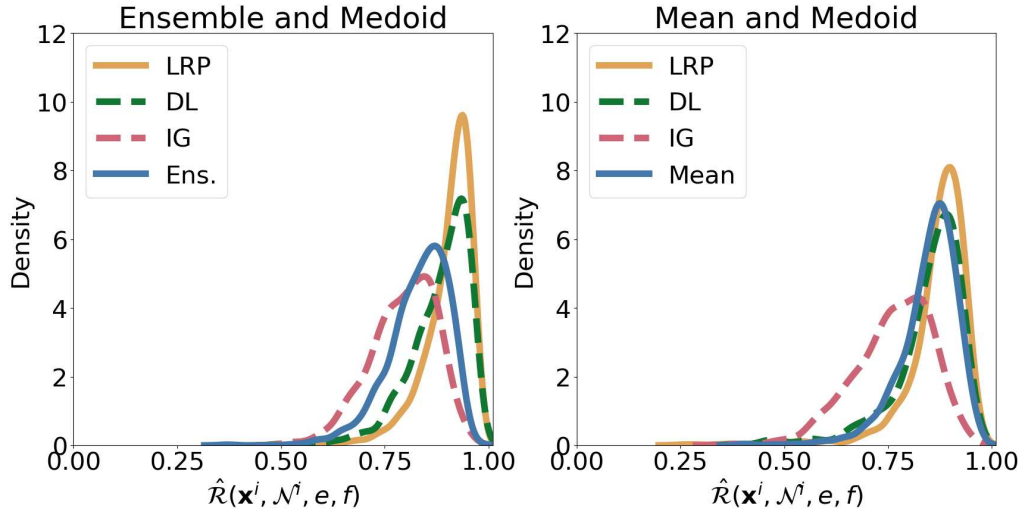


Figure 6.1: The robustness scores distribution for the adult dataset of the ensemble (left) and the mean (right) aggregations with a medoid-based neighbourhood generation.

Table 6.2: Average robustness scores (%) on the test sets, for each dataset and the XAI method with a medoid-based neighbourhood. The left table compares the Ensemble (Ens.) with the feature attributions of the other approaches in absolute value, while the right-most table compares the Mean aggregation with the attributions taken with sign. Values in bold indicate the largest robustness dataset-wise.

Dataset	LRP	DL	IG	Ens.	LRP	DL	IG	Mean
beans	86.69	95.60	93.96	87.85	86.54	98.12	95.74	94.28
cancer	73.35	96.99	96.98	85.71	72.80	98.57	98.80	94.60
mushroom	70.30	72.12	76.72	73.56	64.21	62.20	65.59	64.44
white wine	73.22	90.38	88.27	69.18	69.09	92.78	91.53	84.32
adult	90.25	82.72	87.55	83.07	86.91	84.66	76.19	84.75
bank	81.16	82.72	87.55	82.83	79.33	78.80	81.03	80.73
heloc	83.88	89.45	86.54	83.27	77.55	85.30	82.36	84.33
ocean	86.13	97.39	94.32	88.75	72.10	94.89	94.55	84.11

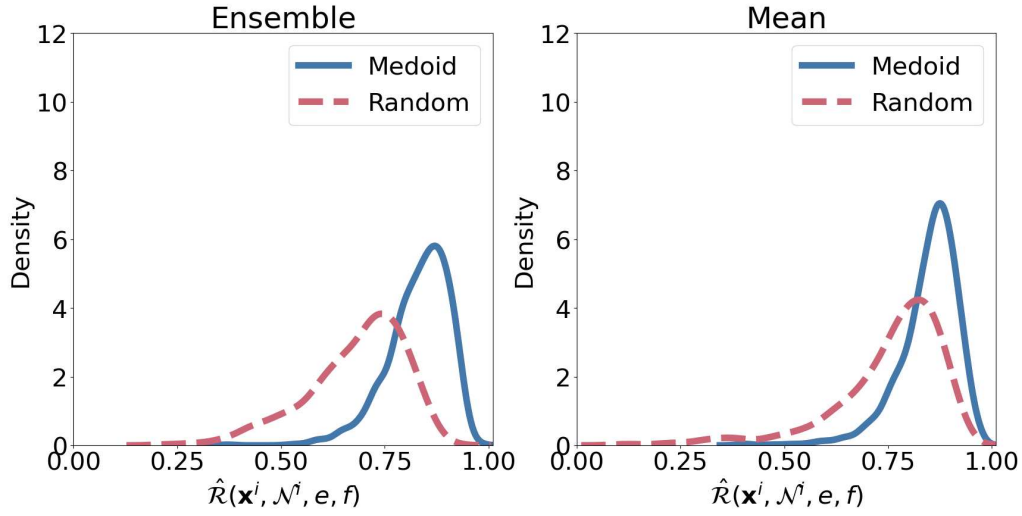


Figure 6.2: The robustness scores distribution for the adult dataset of the ensemble (left) and the mean (right) aggregations with both a medoid-based (solid blue line) and a random (dashed red) neighbourhood.

Table 6.3: Average robustness scores (%) on the test sets, for each dataset and the XAI method with a random neighbourhood. The left table compares the Ensemble (Ens.) with the feature attributions of the other approaches in absolute value, while the right-most table compares the Mean aggregation with the attributions taken with sign. Values in bold indicate the largest robustness dataset-wise.

Dataset	LRP	DL	IG	Ens.	LRP	DL	IG	Mean
beans	73.65	83.61	77.26	75.54	77.0	89.37	81.97	86.21
cancer	53.96	69.87	68.76	64.88	56.91	77.57	78.81	75.2
mushroom	69.18	70.93	75.63	72.32	63.82	61.37	64.46	63.68
white wine	63.04	72.67	76.42	58.48	61.70	82.12	81.79	76.05
adult	79.39	75.63	60.20	68.21	80.18	75.86	60.72	75.51
bank	73.82	75.64	80.60	76.12	74.98	74.58	77.56	76.56
heloc	56.82	61.15	56.80	54.03	59.44	67.84	64.02	66.1
ocean	74.23	93.37	89.05	82.18	51.14	86.23	87.47	67.36

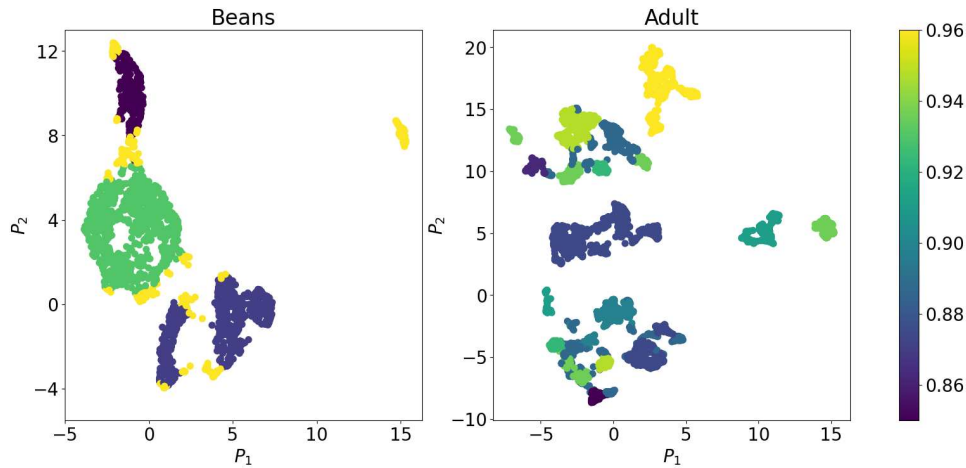


Figure 6.3: HDBSCAN clusters, coloured by mean robustness scores, over the two dimensional UMAP projections (P_1, P_2) for the beans (left) and adult (right) datasets.

We compared the aggregations also in the case of randomly generated perturbations. Figure 6.2 depicts the effects of the two neighbourhood generating schema on the aggregation’s non-adversarial robustness. Akin to Figure 5.3, the solid blue curve represents the average robustness with a medoid-based neighbourhood while the dashed red one with a random neighbourhood. Confirming the statement of Property 5.2.4, the on-manifold neighbourhood is able to generate greater robustness scores than the random counterpart, also in the scenario in which the explanation is an aggregation.

Table 6.3 presents the average robustness with the random neighbourhood. In this case as well, it is possible to appreciate that the robustness of the aggregation is an average of the composing methods and that the largest robustness is achieved by a different method depending on the dataset.

Interestingly, to support the claims of Property 5.2.4, it is possible to note that all the values in Table 6.3 are lower than those in Table 6.2, confirming the advantage of using the on-manifold neighbourhood.

In Figure 6.3, we present a two dimensional visualization of the UMAP [80] projections of the validation set $\mathcal{D}_{valid}$, clustered with the HDBSCAN algorithm [18], in which each cluster is coloured by its mean robustness score (of the Ensemble with a medoid-based neighbourhood). UMAP, as introduced in Subsection 2.3.2, is a dimensionality reduction technique that allows us to visualize a lower dimensional projection of the data. We applied the density-based algorithm HDBSCAN (see Subsection 2.3.1) to such two-dimensional projections, producing clusters without the need of setting hyperparameters for the desired number of cluster, as is the case in k -means instead. As can be seen from the figure, the projected dataspace may be more or less complex according to the dataset under evaluation. Robustness homogeneity within the derived clusters supports the claims of Property 5.2.2, which states that close points are expected to exhibit smaller differences between their robustness scores.

Chapter 7

An Evaluation Perspective

This Chapter will present an evaluation perspective on explanation *trustworthiness*. Starting from the contributions of Chapters 5 and 6, we will present a pipeline to test the trustability of the explanations as well as a new validation approach to tackle the lack of a ground truth for the robustness. Additionally, we will investigate the role of unbalanced datasets (Section 7.5) and the reliability of the explanations for the minority class in a practical insurance use-case.

7.1 When Can You Trust Your Explanations?

The estimate of the explanation robustness defined via Equation 5.11 may not be sufficient when evaluating a previously unseen datapoint. In particular, we argue that the punctual estimate may not reflect the whole robustness concept, as the datapoint may lie in an unstable area of the feature space. Robustness is not only important from a punctual point of view: it is necessary to verify also that the datapoint lies in a robust area of the feature space. Assume that it is known that a given neighbourhood is non robust: this information allows us to *doubt* the robustness score of a new datapoint, even if the score itself is seemingly very high.

To address this issue, in [134] we proposed to leverage local information from the validation set $\mathcal{D}_{valid}$ via a k -Nearest Neighbours regressor fitted on such set's robustness scores. In particular, having $\mathbf{x}^i \in \mathcal{D}_{test}$ a new datapoint, we compute both the robustness score $\hat{\mathcal{R}}(\mathbf{x}^i) := \hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f)$ via Equation 5.11 and the regressor's prediction $\hat{\mathcal{R}}_{knn}(\mathbf{x}^i)$. Let r_{th} be a selected threshold value, then it is possible to distinguish between the following scenarios:

1. if $\hat{\mathcal{R}}(\mathbf{x}^i) \geq r_{th}$ and $\hat{\mathcal{R}}_{knn}(\mathbf{x}^i) \geq r_{th}$ then $\mathbf{x}^i$ is a *robust* point;
2. if $\hat{\mathcal{R}}(\mathbf{x}^i) \geq r_{th}$ and $\hat{\mathcal{R}}_{knn}(\mathbf{x}^i) < r_{th}$ then $\mathbf{x}^i$ is an *uncertain* point, as it lies in an area of the feature space over which robustness may be lacking;
3. if $\hat{\mathcal{R}}(\mathbf{x}^i) < r_{th}$ then $\mathbf{x}^i$ is a *non robust* datapoint.

Among these scenarios, the second one represents a set of conditions that practitioners should evaluate carefully. In fact, despite having a computed robustness score that is greater than the selected threshold r_{th} , the local information derived from the neighbours suggests otherwise. This scenario aims at *ringing a bell* in the

practitioner evaluating a given explanation: knowing that it may be misleading, the analysis of the specific instance will be more careful and may require a more detailed human evaluation of both the model’s prediction and the derived explanation.

The advantage of using a k -NN regressor instead of a k -NN classifier is that it allows to be independent from the selected robustness threshold r_{th} , requiring the model be fitted only once. The number of neighbours k_R is a dataset-dependant hyperparameter and relies on the goodness of the approximation of the robustness score by the regressor, fitted on the validation set $\mathcal{D}_{valid}$.

7.2 The Pipeline

The following steps describe in detail the complete framework (summarized in Figure 7.1) for robustness evaluation and explanation trustworthiness.

1. Split the dataset into $\mathcal{D}_{train}$, $\mathcal{D}_{valid}$, $\mathcal{D}_{test}$ and perform the required preprocessing steps.
2. Train a neural network model f on $\mathcal{D}_{train}$.
3. Perform k -medoid clustering on $\mathcal{D}_{valid}$ and retrieve the k_M nearest neighbours among the medoids.
4. For each point $\mathbf{x}^j \in \mathcal{D}_{valid}$, generate a neighbourhood $\mathcal{N}^j$ of size n , following Subsection 5.3.2.
5. Compute the attributions of LRP, DeepLIFT and Integrated Gradients and merge them following the ensemble aggregation scheme (Equation 6.7).
6. Compute the robustness score $\hat{\mathcal{R}}$ via Equation 5.11 for each point $\mathbf{x}^j \in \mathcal{D}_{valid}$.
7. Use the robustness scores computed in Step 6 to train a k -Nearest Neighbours regressor with k_R neighbours and select an appropriate threshold r_{th} .
8. For each point $\mathbf{x}^i \in \mathcal{D}_{test}$, predict the medoid cluster it belongs to and generate a neighbourhood $\mathcal{N}^i$ of size n , following Subsection 5.3.2.
9. Compute the L feature attribution vectors and the resulting aggregation.
10. Compute the robustness score $\hat{\mathcal{R}}$ via Equation 5.11 and the predicted robustness $\hat{\mathcal{R}}_{knn}$ via the k -NN regressor trained at Step 7.
11. Assess the trustworthiness following Subsection 7.1.

The computational cost of this procedure is dominated by the computation of the individual attributions, as a pass through the network is needed for each datapoint, for each perturbation and for each of the three methods. The subsequent robustness method inherits this complexity.

This framework is particularly useful for practitioners aiming at investigating explanation robustness and overall trustworthiness, as it includes an ensemble-based

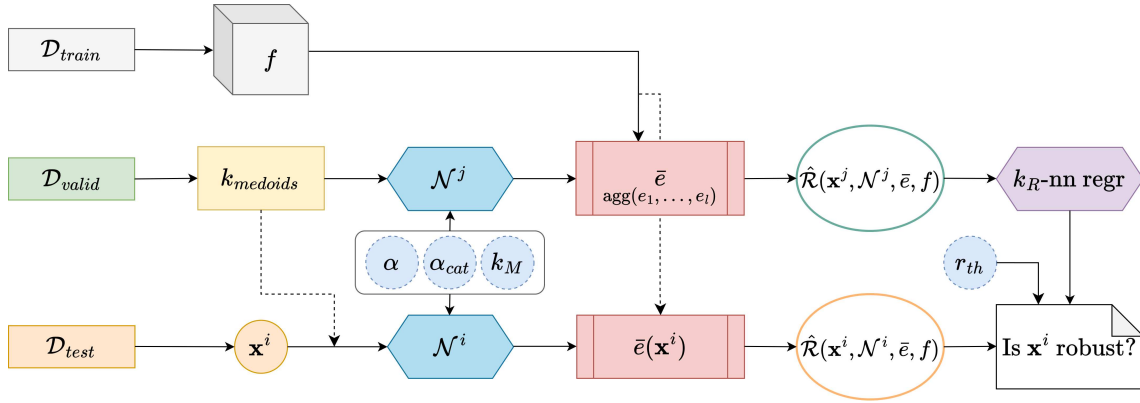


Figure 7.1: The complete pipeline to test explanation trustworthiness.

explanation that captures the signals of L individual explanation methods and computes the robustness on a carefully constructed neighbourhood. The pipeline is designed to flag possibly non robust datapoints, even in the cases in which they may appear as such, ensuring greater trustworthiness in the system as a whole, as it allows for a more aware analysis from a practitioners perspective.

A note on categorical variables. The pipeline, as presented, is applicable to datasets made entirely of numerical variable, while a small adjustment is required to correctly take into account categorical variables as well. As discussed in Chapter 5, the *reverse encoding* should be applied when categorical variables are present in Steps 5 and 9, prior to the ensemble’s computation.

7.3 Validation

One of the aspects that impede robustness estimation in practical applications is the frequent lack of a ground truth, as the data generation process is unknown in real-world applications. Comparisons against expert-provided explanations may be also unfeasible due to the large amount of data under evaluation and the significant costs that may be needed to support such manual labelling.

In [134], we argued that robust explanation-wise datapoints lie in a robust area of the feature space, and should be considered robust even when passed through different models. This assumption implies that non robust points will exhibit differences both in the explanations and in the predictions of multiple models, as the lack of robustness can be seen as a somewhat intrinsic characteristic of the area of the manifold in which they lie.

Let us consider three neural networks, say f_1 , f_2 and f_3 , which have comparable accuracy over $\mathcal{D}_{train}$ and that differ either in the number of hidden layers or neurons per layer. Let $\mathcal{D}_{agree}$ be the subset of datapoints for which model f_1, f_2, f_3 predict the same class and $\mathcal{D}_{disagree}$ the subset for which one the models predicts a different class, i.e. $\mathcal{D}_{agree} = \{\mathbf{x}^i \in \mathcal{D}_{train} \mid f_1(\mathbf{x}^i) = f_2(\mathbf{x}^i) = f_3(\mathbf{x}^i)\}$ and $\mathcal{D}_{disagree} = \{\mathbf{x}^i \in \mathcal{D}_{train} \mid f_I(\mathbf{x}^i) = f_{II}(\mathbf{x}^i) \neq f_{III}(\mathbf{x}^i)\}$ with $\{I, II, III\}$ indicating any ordering of the indices $\{1, 2, 3\}$, with no loss of generality.

Consider a point to be robust if $\hat{\mathcal{R}}(\mathbf{x}^i) \geq r_{th}$ and non-robust otherwise. We propose the validation to follow a ROC/AUC analysis, where the True Positive Rate (TPR) and False Positive Rate (FPR) are defined as:

$$\text{TPR} = \frac{\#\{\text{Robust \& Agree}\}}{\#\{\text{Agree}\}} \quad \text{FPR} = \frac{\#\{\text{Robust \& Disagree}\}}{\#\{\text{Disagree}\}} \quad (7.1)$$

Varying the threshold value r_{th} , it is possible to plot the ROC curve and compute the corresponding AUC value.

Note that in this case, the difference between robust and non-robust datapoints is linked only to the computed robustness score $\hat{\mathcal{R}}$ and the threshold r_{th} , unlike in Subsection 7.1 where the k NN prediction is additionally taken into account.

7.4 Results

7.4.1 Hyperparameter selection

The pipeline introduced in the previous Subsections is mainly governed by two hyperparameters: the number of neighbours used to fit the k NN regressor and the robustness threshold. The number of neighbours k_R is chosen as the one minimising the approximation error over the robustness scores. A good default value is $k_R = 7$. The robustness threshold r_{th} is selected by looking at the distribution of robust, non robust and uncertain datapoints at varying levels of the threshold. In particular, it is chosen as the threshold value that corresponds to the first inflection point of the robust percentage curve. The default value $r_{th} = 0.80$ works well in most scenarios. Case-specific hyperparameters can be found in Table 7.1 where, for both neighbourhood generating schemes and aggregation methods, the chosen hyperparameters are presented.

Table 7.1: Hyperparameter selection for the Random and Medoid-based neighbourhood generation schemes. For both the Ensemble and the Mean aggregation, both the number of neighbours (k_R) used in the k NN regressor fitting and the chosen threshold (r_{th}) are presented.

	Random $\mathcal{N}_R$				Medoid-based $\mathcal{N}_M$			
	Ensemble		Mean		Ensemble		Mean	
Dataset	k_R	r_{th}	k_R	r_{th}	k_R	r_{th}	k_R	r_{th}
beans	5	0.75	7	0.80	9	0.85	9	0.90
cancer	5	0.55	7	0.65	11	0.85	9	0.90
mushroom	9	0.70	11	0.60	7	0.70	11	0.60
white wine	9	0.45	7	0.65	11	0.85	7	0.80
adult	5	0.70	5	0.70	9	0.80	9	0.80
bank	7	0.75	9	0.75	7	0.80	11	0.80
heloc	11	0.45	5	0.60	15	0.80	13	0.80
ocean	5	0.75	5	0.65	5	0.65	5	0.75

Table 7.2: When can you trust your explanations? Comparison of the robustness scores with Ensemble and Mean aggregations with $r_{th} = 0.80$. Datapoints are classified as robust (Rob.), uncertain (Uncert.) or non robust (Non rob.).

Dataset	$\mathcal{D}_{test}$	Ensemble (%)			Mean (%)		
		Rob.	Uncert.	Non Rob.	Rob.	Uncert.	Non Rob.
beans	500	74.6	7.2	18.2	93.0	2.2	4.8
cancer	50	62.0	12.0	26.0	100.0	0.0	0.0
mushroom	400	0.0	1.2	98.8	0.0	0.0	100.0
white wine	200	7.5	26.5	66.0	56.5	11.0	32.5
adult	1'000	63.4	7.1	29.5	72.6	7.3	20.1
bank	1'000	65.2	7.1	27.7	44.6	16.4	39.0
heloc	500	62.8	9.2	28.0	60.6	14.0	25.4
ocean	10'000	79.1	5.6	15.3	53.5	16.0	30.5

7.4.2 Pipeline

Table 7.2 presents the classification of the test set into robust, uncertain and non robust datapoints (as per Subsection 7.1⁸). For comparability between datasets and aggregators, we set $r_{th} = 0.80$, even if the Ensemble often requires lower values of r_{th} compared to the mean. The results are computed with a medoid-based neighbourhood, as it was shown in the previous Chapters that it produces larger robustness scores.

For almost all the datasets, robust datapoints are the majority, with the exception of the white wine dataset with the ensemble and the mushroom dataset in both scenarios. This can be traced back to the choice of equal threshold for all datasets, as dataset-specific hyperparameters for the robustness threshold would have it set at values much lower than 0.80.

The percentage of uncertain and non-robust datapoints are non-negligible. In particular, the set of datapoints which are flagged as uncertain are the most interesting - and valuable - to investigate. In fact, they represent areas of the feature space in which the robustness estimation is more uncertain (as per Property 5.2.3). In practical applications, these points are those over which a practitioner should exhibit a greater level of care during an evaluation.

Results from Chapter 6, such as Figure 6.3, moreover suggest that the use of the k NN regressor in the procedure is justified as the datapoints are naturally grouped in clusters with similar robustness scores. Therefore, the k NN regressor helps extract additional local information in terms of robustness.

7.4.3 Validation

To test our validation assessment proposal (Subsection 7.3⁸), for each dataset we have trained three neural networks, referred to as Model 1, Model 2 and Model 3. The three models showcase a similar accuracy score (Table 7.3) and have varying architectures. In particular, Model 1 acts as a baseline (and it is the model used

⁸The used code can be found at https://github.com/ilariavascotto/XAI_robustness_analysis

Table 7.3: Model accuracy (%) over the training set.

Dataset	Model 1	Model 2	Model 3
beans	93.41	93.44	89.47
cancer	99.5	99.75	99.24
mushroom	100.00	100.00	100.00
white wine	89.23	89.10	86.55
adult	91.39	91.38	91.09
bank	91.99	91.76	91.45
heloc	85.50	85.57	85.01
ocean	92.04	87.88	92.27

Table 7.4: Comparison of ensemble’s robustness classification (with $r_{th} = 0.80$) according to Model 1,2,3 concordance in the prediction. The two right-most columns for the Robust and Non Robust cases indicate the number of datapoints of the test set falling within each scenario, such that $|\mathcal{D}_{test}| = n_{Rob} + n_{\overline{Rob}}$.

Dataset	Robust			Non Robust		
	Agree (%)	Disagree (%)	n_{Rob}	Agree (%)	Disagree (%)	$n_{\overline{Rob}}$
beans	95.35	4.65	409	95.60	4.40	91
cancer	91.89	8.11	37	92.31	7.69	13
mushroom	100.00	0.00	5	100.00	0.00	395
white wine	94.11	5.88	68	92.42	7.58	132
adult	93.76	6.24	705	80.68	19.32	295
bank	95.99	4.01	723	90.25	9.75	277
heloc	82.22	17.78	360	56.43	43.57	140
ocean	85.21	14.8	8’475	80.33	19.67	1’525

for the previously presented results of Chapters 5 and 6), Model 2 has more layers and more neurons per layer while Model 3 is a more compact version, with a limited number of layers. Model 2 and 3 have the same activations as Model 1 (ReLU in all cases but the ocean dataset, in which tanh was used, and with softmax in the output layer).

Table 7.4 shows how the percentage of datapoints over which the three above-introduced models (dis)agree on the prediction varies according to the computed robustness score, having set also for this comparison $r_{th} = 0.80$. Note that the mushroom dataset is a particular case, as all three models reach 100% accuracy and are, therefore, always agreeing prediction-wise.

It can be seen that the percentage of disagreeing points within the non-robust ones is, for most datasets, greater than that of the robust ones. This supports the validation proposal, as we deem the disagreement prediction-wise to be a symptom of non robustness, as stated by Property 5.2.6.

The ROC/AUC analysis presented in Subsection 7.3 allows us to jointly consider the aggregation method and the neighbourhood generation scheme which better fit the dataset at hand. Fig. 7.2 depicts the ROC curves of the three models in varying scenarios: Model 2 is consistently associated with the highest ROC curve,

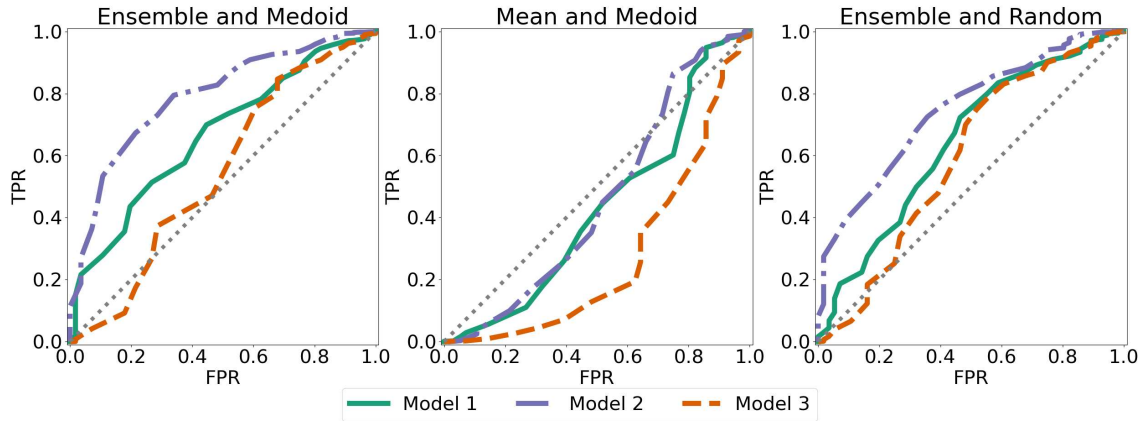


Figure 7.2: ROC curves on the bank dataset with ensemble aggregation and medoid-based neighbourhood (left), mean aggregation and medoid-based neighbourhood (centre) and ensemble aggregation with random neighbourhood (right). The gray dotted line represents the bisecting line.

Table 7.5: Average AUC value of Model 1, 2, 3 for both aggregation types and neighbourhoods. For each dataset, the largest average AUC is presented in bold.

Dataset	Medoid		Random	
	Ensemble	Mean	Ensemble	Mean
beans	0.5711	0.4854	0.4842	0.4067
cancer	0.7346	0.7400	0.6766	0.6612
mushroom	0.0000	0.0000	0.0000	0.0000
white wine	0.4762	0.5931	0.4819	0.6614
adult	0.7018	0.6695	0.8284	0.8077
bank	0.6670	0.3883	0.6612	0.4696
heloc	0.6640	0.6673	0.6262	0.5875
ocean	0.5194	0.5128	0.4350	0.4053

suggesting that, for the bank dataset, a deeper net is better able to propose robust explanations. Moreover, the ROC curves are preferable for all three models with the ensemble aggregation and the medoid-based neighbourhood, while the mean aggregation (as shown in the central plot) presents ROC curves even below the bisecting line.

This is further confirmed by the results presented in Table 7.5, where the average AUC is computed for the four possible combinations of aggregation method and neighbourhood generation. The dataset-wise maximum values of AUC are presented in bold, showing how the ensemble is, on average, preferable with respect to the mean aggregation and how the medoid-based neighbourhood is related to larger AUC values in most examples. The mushroom dataset represents, as in Table 7.4, a peculiar case: having only agreeing datapoints the ROC and AUC cannot be computed and are therefore exempt from this analysis.

7.5 Unbalanced Datasets

Real-world datasets in a variety of domains are intrinsically unbalanced as the critical events, such as rare diseases, natural catastrophes or fraudulent transactions, occur less frequently than the normal instances. The fields of insurance and asset management often face problems with an intrinsic unbalance in the response variable, for example when predicting the probability of churn of a customer or the risk of fraud.

Despite the numerous practical examples that may come to mind, dataset unbalance may pose significant challenges for the AI and ML models, leading at times to biased model predictions. In high-risk scenarios especially, it is crucial that the predictions are accurate and unbiased, due to the serious consequences a wrongful prediction may have. In this context, it becomes particularly valuable to gain insights on a model's decision to enhance the trustworthiness of the system.

Let us introduce a simple example. Assume that a model f reaches an accuracy of 99% on the training set $\mathcal{D}_{train}$ for a given real-world problem. If the dataset is balanced, or presents a limited factor of unbalance, it is possible to consider the behaviour as an appropriate one. If, instead, we assume that the dataset is highly unbalanced, i.e. $Pr(y = 0) = 0.99$ and $Pr(y = 1) = 0.01$, we cannot deem the model to be as well-behaved as in the previous case. In fact, an accuracy equal to 99% could be easily obtained with a model which predicts all the datapoints to belong to class 0, the majority class. This is, of course, an undesirable behaviour and highlights an important facet of explanation reliability with unbalanced datasets. Consider that, if the majority class is easily predicted by the model, there are no guarantees on the quality of the derived explanations and their trustworthiness. Moreover, a common practice in ML when dealing with an unbalanced response variable is to adjust the model to concentrate on the minority class (the most difficult one to predict). In this scenario as well it is not possible to assume that the model has correctly learned the specific facets of the majority class, even when the accuracy is high.

7.5.1 Minority Class Evaluation

In [131] we introduced a framework to test the minority class explanations. The focus on the minority class is derived from the above-discussed considerations and on the relevance in accurately predicting rare events in high-risk applications. In the following, let us assume that $y = 0$ indicates the majority class and $y = 1$ the minority one, such that $Pr(y = 0) \gg Pr(y = 1)$.

Given a neighbourhood $\mathcal{N}^i$ of size n , constructed as in Subsection 5.3.2, let e be an explanation method applied to both the original datapoint $\mathbf{x}^i$ and its perturbations $\tilde{\mathbf{x}}^i$ and let $e(\mathbf{x}^i), e(\tilde{\mathbf{x}}^i)$ be the corresponding explanations. For the datapoints belonging to the minority class $y = 1$, it is possible to compute a locally weighted attribution as follows:

$$\bar{e}(\mathbf{x}^i) = \frac{\sum_{\tilde{\mathbf{x}}^i \in \mathcal{N}^i} e(\tilde{\mathbf{x}}^i) \cdot \pi(\mathbf{x}^i, \tilde{\mathbf{x}}^i)}{\sum_{\tilde{\mathbf{x}}^i \in \mathcal{N}^i} \pi(\mathbf{x}^i, \tilde{\mathbf{x}}^i)} \quad \text{where } \pi(\mathbf{x}^i, \tilde{\mathbf{x}}^i) = \frac{1}{d(\mathbf{x}^i, \tilde{\mathbf{x}}^i)} \quad (7.2)$$

This aggregated explanations allows us to take into account the results of a data augmentation on the minority class, which is ensured to be faithful to the

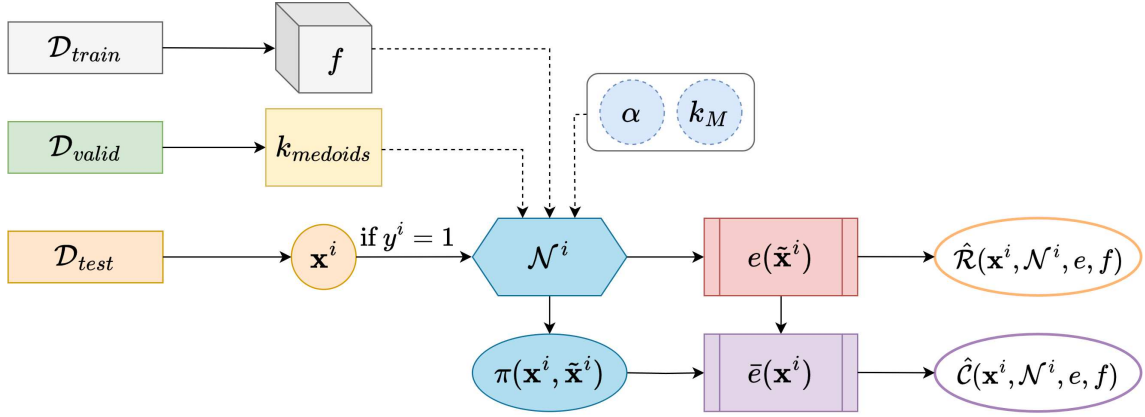


Figure 7.3: A representation of the proposed steps for minority class evaluation.

data manifold thanks to the medoid-based neighbourhood generation. In Equation 7.2, the explanations are computed on perturbed datapoints which themselves are predicted to belong to the minority class and are weighted according to their distance from the original datapoint $\mathbf{x}^i$. This ensures that same-class perturbations which are more distant will be associated to a smaller weight in the aggregated explanation.

To assess the reliability of explanations for sensible datapoints, we proposed to verify the local robustness of the derived explanations with two locally-computed scores on $\mathcal{N}^i$. The first one - referred to in this context as *local robustness* - is the robustness score $\hat{\mathcal{R}}$ introduced in Chapter 5, where the average Spearman's rank correlation is computed between the original datapoint's explanation $e(\mathbf{x}^i)$ and that of its perturbations $e(\tilde{\mathbf{x}}^i)$.

We introduced a second score - known as *consistency* [131]- which computes the rank correlation between the original explanation and the locally weighted aggregation (Equation 7.2).

$$\hat{\mathcal{C}}(\mathbf{x}^i, \mathcal{N}^i, e, f) = \rho(e(\mathbf{x}^i), \bar{e}(\mathbf{x}^i)) \quad (7.3)$$

The consistency score allows us to investigate if the aggregated explanation is locally faithful to the original one, ensuring that it summarizes useful information on the model decision making.

A summary of the complete steps is presented in Figure 7.3.

7.6 Results

To test the effectiveness of the proposed approach, we used a real-world insurance use-case provided by Assicurazioni Generali S.p.A. Despite being a proprietary dataset - therefore not publicly shareable - we can outline the main steps of the work and some of the characteristics of the data.

7.6.1 Dataset Description and Model Training

The dataset addresses the problem of identifying frost events, in particular their occurrence, using atmospheric data. This occurrence estimation is necessary for

Table 7.6: The results in terms of F1-score on the train, validation and test sets for the majority class 0 and the minority class 1 and the support, i.e. the number of points in each set.

	Train		Validation		Test	
	F1-score	Support	F1-score	Support	F1-score	Support
Class 0	1.00	$\sim 225'000$	1.00	$\sim 45'000$	1.00	$\sim 30'000$
Class 1	0.66	$\sim 2'500$	0.50	~ 450	0.51	~ 300

insurance practitioners in order to estimate agricultural business policies, especially in regions in which the data is limited. The explanatory variables were constructed starting from publicly-available data from the ERA5 reanalysis data [55] from the European Centre for Medium-Range Weather Forecasts (ECMWF). The final dataset was made of 8 numerical variables which were standardized during a preprocessing phase. Data are aggregated by days and by municipalities and only spring and summer months are considered, as they represent the time span over which frost events can be more damaging in agricultural fields. The target data were instead derived from proprietary datasets on insurance policies and claims. They take on value of either 0 or 1, indicating whether the frost even occurred during a given day and in the selected municipality. The available time window spans over a period of 15 years, from 2009 to 2024, and the focus was the country of Poland. The dataset presents an extremely unbalance within the possible class labels, having the majority class present with a frequency of 0.99 while the minority one of 0.01.

The dataset was split into a training, validation and test set, $\mathcal{D}_{train}$, $\mathcal{D}_{valid}$, $\mathcal{D}_{test}$ respectively. The training set was used to train a Multi Layer Perceptron (MLP) with ReLU activation functions in the hidden layers and sigmoid activation in the output one. The model was trained with Focal Loss (FL) [73], as it is specifically tailored for unbalanced datasets. Hyperparameters for the loss were set via a random search, from which $\gamma = 2.5$ and $\alpha = 0.75$ were selected as the best performing on the validation set. Table 7.6 presents the F1-score computed separately for the majority and minority class over the three datasets. In particular, it is possible to note the the model excels (as expected) in predicting the majority class ($y = 0$) while performs modestly on the minority one ($y = 1$). Nonetheless, given the complexity of the problem at hand and the limited number of data points for the minority class, we consider these results to be satisfactory.

7.6.2 Reliability of Explanations

We focused on the test set datapoints for an evaluation⁹. Firstly, we investigated the effects of the neighbourhood generation on the robustness of both majority and minority class. As can be seen in Figure 7.4, the two neighbourhood generating processes discussed in Chapter 5 present significant differences in the robustness values of the Integrated Gradients explanations. Interestingly, the minority class (the blue line - Frost) appears to be the most affected by the choice of neighbourhood, as it is associated with a larger distribution shift between the left and right plots.

⁹The used code can be found at https://github.com/ilariavascotto/Reliability_Unbalanced

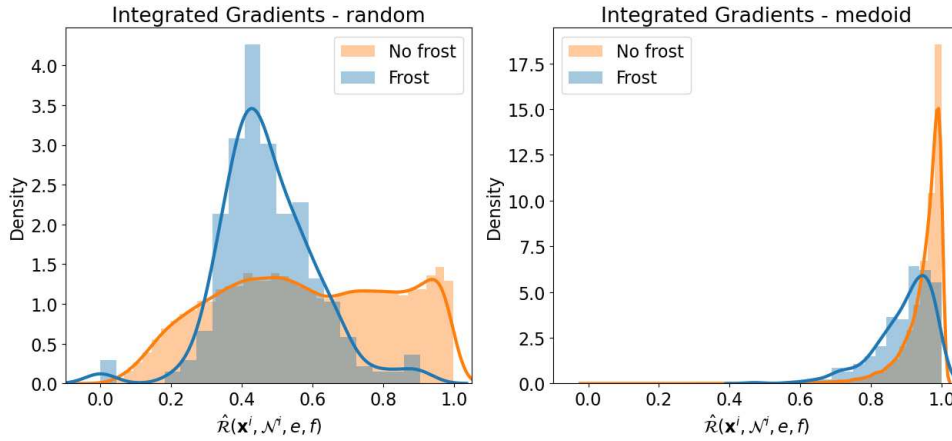


Figure 7.4: The robustness of Integrated Gradients computed with a random neighbourhood (left) and a medoid-based one (right). In blue, the distribution of the robustness scores for the frost events (minority class) and in orange for the non-frost events (majority class).

Similar results have been obtained with DeepLIFT, LRP and the Ensemble (Chapter 6). The results support our hypothesis on the vulnerability of the minority class, suggesting the need for careful evaluations and deeper analysis for these instances.

Focusing specifically on the minority class datapoints, $\mathcal{D}_{frost} = \{\mathbf{x}^i \in \mathcal{D}_{test} \text{ s.t. } y^i = 1\}$, a neighbourhood $\mathcal{N}^i$ of size $n = 100$ was built for each point with a medoid-based scheme ($\alpha = 0.05$ and $k_M = 5$) and the robustness score $\hat{\mathcal{R}}$ was computed.

Figure 7.5 (left) depicts the robustness scores on $\mathcal{D}_{frost}$, showing how LRP is the least stable method in this scenario (also confirmed by the lower average robustness and standard deviation in Table 7.7). To limit the effects of the vanishing gradient problem, the epsilon rule (Section 4.4) was selected for this dataset. The computation of the ensemble’s robustness is affected by the lower performance of LRP. DeepLIFT and Integrated Gradients instead produce greater robustness scores.

The robustness computation alone is not sufficient to describe the minority class explanation quality nor to ensure that they are meaningful. Our proposal [131] is to compute the locally weighted attribution $\bar{e}$ and test its consistency with respect to the original datapoint’s explanation (Equation 7.3). Having datapoints which lie in a robust area of the dataspace (as confirmed by the robustness scores), allows us to leverage on-manifold information to enrich the set of datapoints belonging to the minority class. The individual explanation which can be derived by this data augmentation takes into account also the neighbouring points and helps understanding the decision making process of the model.

Figure 7.5 (centre, right) presents the consistency scores of the four explainers on $\mathcal{D}_{frost}$. It is possible to note that the two most robust methods, DeepLIFT and Integrated Gradients, are associated with the largest consistency scores. This is to be expected as the local explanations are more similar to one another and therefore the averaged explanation remains highly correlated to that of the original datapoint. While LRP and the ensemble suffer from lower robustness scores, their consistency scores are still satisfactory and indicate a higher correlation when considering the average explanation.

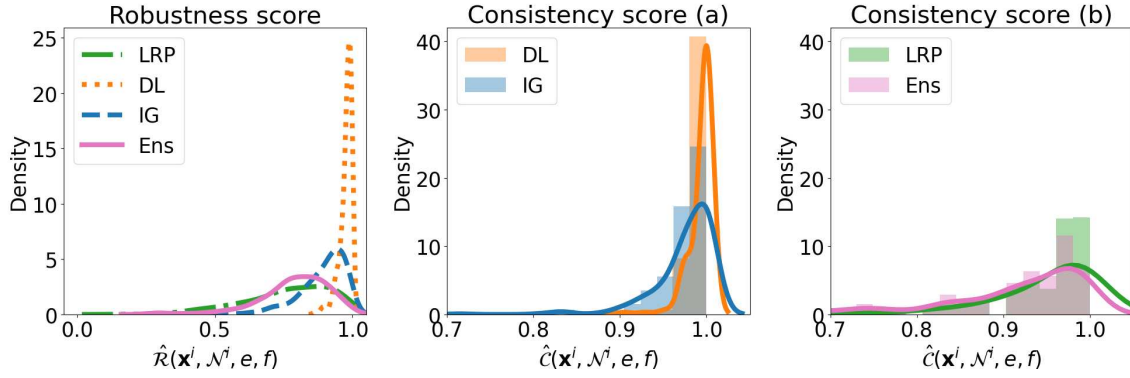


Figure 7.5: Average robustness scores on the test set for the minority class (left). Average consistency scores on the test set for the minority class (centre, right). The division between the two panels was made for readability purposes.

Table 7.7: The average robustness and consistency scores (%) on frost events for the four explainability methods. Values within brackets represent the standard deviation. Values in bold represent the maximum value over the four methods.

Method	$\hat{\mathcal{R}}(\mathbf{x}^i, \mathcal{N}^i, e, f)$	$\hat{\mathcal{C}}(\mathbf{x}^i, \mathcal{N}^i, e, f)$
LRP	76.77 (± 15.70)	89.86 (± 19.95)
DeepLIFT	97.69 (± 2.26)	99.40 (± 1.51)
Integrated Gradients	89.34 (± 8.35)	97.56 (± 3.58)
Ensemble	79.03 (± 12.56)	89.20 (± 13.73)

Table 7.7 presents the mean robustness and consistency scores with the standard deviation reported within brackets. DeepLIFT is the most robust and most consistent method in this use-case, and it is associated both with the highest average and lower standard deviation.

The locally averaged explanation summarizes more local information with respect to a single explanation for a datapoint and it has been proved to encapsulate the explanations of a robust area of the feature space.

Chapter 8

Feature Attributions for k -NN and RF

This Chapter will present the feature attribution approaches tailored to k -Nearest Neighbours and Random Forests we have developed. Specifically, as both models usually prefer explanations of other forms, it has been necessary to craft new local feature attribution approaches to mimic the models reasoning. These explanations will be used in Chapter 9 in combination with one neural network-specific approach to investigate the effects of multi-model aggregations.

8.1 Related Work

As introduced in Chapter 3, model-agnostic approaches that are applicable also to k -NNs and RFs are LIME [96], SHAP [77], Anchors [97] and LORE [49]. The first two are examples of feature attribution methods while the latter ones return explanations in the form of (sets of) decision rules.

Literature on k -nearest neighbours is certainly limited as far as explanations are concerned. As k -NNs are considered a class of mostly-interpretable models, research on model-specific XAI techniques is limited. Most examples in the literature simply apply LIME or SHAP to the outputs of a k -NN classifier or regressor. Nonetheless, as it has been proved also in Chapter 5, the instability of these approaches may hinder the validity of such explanations even when the underlying model is considered interpretable. [52] attempts at increasing the interpretability of k -NNs by introducing the learning of a locally adaptive metric, which is naturally interpretable.

RF-specific approaches, such as Bellatrex [28], produce explanations in the form of sets of rules, derived from the composing trees in the forest. Specifically, given a random forest f and a datapoint $\mathbf{x}^i$ being explained, a rule pre-selection is performed to consider only those trees whose prediction $T_b(\mathbf{x}^i)$ is as close as possible to that of the random forest. The paths traversed in these trees are mapped to m -dimensional vectors such that each feature is weighted by the proportion of training set instances that have passed through the node. Next, PCA is applied to generate a low-dimensional projection of such decision paths, which in the final step are clustered via k -means. Representatives from each clusters are the final rules which make up the explanations.

8.2 Explaining k -Nearest Neighbours

k -Nearest Neighbours (Subsection 2.2.1) are considered a class of interpretable models as it is quite intuitive to understand (and mimic) the reasoning behind a new prediction. Nonetheless, it is not as simple to identify the features which were most influential for the class assignation.

Leveraging k -NN inner mechanisms, we have proposed a new feature importance approach that locally explains the model decisions. Having selected a value of the hyperparameter k , a new datapoint is classified based on the most frequent class which is found among its k nearest neighbours within the training set. The difference between the, say, two possible classes can mainly be attributed to the subset of features for which a larger distance is recorded between the two sets of corresponding points. Yet, in the case in which the k -NNs all belong to the same class, it is hard to perform such comparison. In fact, locally there are no distinctions to be highlighted, but the classification can be imputed to the fact that the datapoint is closer to those of the predicted class rather than the opposing one. This concept naturally aligns with the reasoning of k -NNs on a more general scale: datapoints are classified based on the most similar class, where the similarity is defined with distance-based metrics.

Our proposal leverages these characteristics, producing a feature importance vector as local explanation and is an extension of what was originally presented in [133]. Notably, a dataset condensation step was introduced as a starting point to limit the variability of the derived explanations and increase their local robustness.

8.2.1 Dataset Condensation

We have crafted a novel approach that substitutes the numerous training set datapoints ($\mathcal{D}_{train}$) with a smaller subset of points ($\mathcal{D}_{cond}$) to aid the retrieval of meaningful and robust explanations. We refer to this approach as condensation-based, as it was inspired by dataset condensation approaches in 1-NN literature.

An outline of the dataset condensation procedure can be found in Algorithm 8.2. The key idea of the condensation is to focus on representative datapoints which can be used as references during the explanation process of k -NN (Subsection 8.2.2).

The first step of the procedure in the CONDENSATION function is creating an array, called *reference*, that stores the indices of the trails which are found within the dataset. A trail can be defined as a cluster of datapoints which belong to the same class and that can be reached following the nearest neighbours of the composing datapoints. In particular, trails are built starting from the datapoints $\mathbf{x}^i$ that have all their k -NNs belonging to the same class, y^i . This allows to consider trails of class-specific points which can be later used to extract representative instances.

Datapoints for which at least one of the k -NNs belongs to the opposing class are discarded at the beginning of the procedure and assigned a reference value of 0. The remaining ones are scanned and, if they have not already been assigned to a trail, act as the starting point for the FOLLOW_TRAIL procedure in which the index C is used to identify the membership to the new trail. The procedure incrementally adds datapoints to the trail according to their reachability from the closest neighbours.

Having selected a point p as starting point, its neighbours are retrieved and are added to the trail (if previously unassigned, i.e. with reference = -1). The k -NNs

Algorithm 8.2 Dataset condensation

```

function FOLLOW_TRAIL(p, reference, C)
    reference[p] = C                                ▷ Assign first point to C
    neighbours_list = [p]                            ▷ Starting point
    i ← 0
    while i < length(neighbours_list) do
        x ← neighbours_list[i]                        ▷ Extract one neighbour from trail
        neighs ← K-NN(x)                             ▷ Find its closest neighbours
        for  $\mathbf{x}^j \in \text{neighs}$  do
            if reference[ $\mathbf{x}^j$ ] = -1 then                ▷ If previously unassigned
                reference[ $\mathbf{x}^j$ ] = C                    ▷ Add to C
                new_neighs = K-NN( $\mathbf{x}^j$ )                ▷ Find its closest neighbours
                if NEIGHS_SAME_CLASS( $\mathbf{x}^j$ ) then        ▷ All k-NNs of same class
                    neighbours_list.append(new_neighs)    ▷ Add to trail
                end if
            end if
        end for
        i ← i + 1
    end while
    return
end function

function FIND_MEDOIDS(reference, X, max_size, min_count)
    trail_list ← [C in unique(reference) if count(C) > min_count]
                                                    ▷ Remove small trails
    trail_size ← count(trail_list)                ▷ Array: size of each trail in trail_list
     $k_{\text{trail}}$  ← CEIL(trail_size/max_size)          ▷ Array: k-medoids param. for each trail
    medoids ← []
    for i, trail in ENUMERATE(trail_list) do
        K-MEDOIDS(X,  $k_{\text{trail}}[i]$  | trail)          ▷ Compute  $k_{\text{trail}}[i]$  medoids in trail
        medoids.append(K-MEDOIDS.centres)          ▷ Save medoids
    end for
    return medoids
end function

function CONDENSATION(X, max_size, min_count)
    reference ← array(0, dim=length(X))
    idx ← NEIGHS_SAME_CLASS(X) ▷ Points with all k-NNs of same class as x
    reference[idx] = -1
    C ← 1
    for p in idx do
        if reference[p] ≠ -1 then                    ▷ If already assigned, ignore
            continue
        end if
        FOLLOW_TRAIL(p, reference, C)                ▷ Follow neigh. trail
        C ← C+1
    end for
    medoids ← FIND_MEDOIDS(reference, X, max_size, min_count)
    return medoids                                ▷ Condensed dataset
end function

```

of the added points are computed and, if they all belong to the same class as point p , are added to the list of neighbours over which the procedure iterates. This allows to retrieve sets of datapoints which are connected via a neighbours' path (thus, trail) and that belong to the same class.

These trails are used in the `FIND_MEDOIDS` procedure which computes the condensed dataset. In particular, a first filtering is performed to discard trails with too few datapoints (the choice is governed by the hyperparameter `min_count`). This allows to focus on more populated trails which may be more meaningful in effectively describing the dataset.

For each of the remaining trails, the number k of clusters to be computed via k -medoids is selected as the ratio between the number of points in the trail (i.e. `count(trail)`) and the hyperparameter `max_size`, which governs the size of the individual clusters (expected to have an average size of `max_size`). For each trail then, the k -medoids algorithm is applied with the number of clusters set to $k_{trail}[i]$ and the cluster centres are stored. The collection of cluster centres acts as the condensed dataset ($\mathcal{D}_{cond}$), which encompasses representative datapoints class-wise and that are part of class-specific reachable trails.

A qualitative example of the condensation can be found in Figure 8.1, where the approach is applied to the moons (top row) and circles (bottom row) datasets. A small amount of random noise ($\epsilon \sim \mathcal{N}(\mu = 0, \sigma = 0.05)$) was added to increase the complexity of the task. The right column presents the condensed dataset (black circles) and the lighter datapoints correspond to the trails over which the medoids were computed. In particular, they are coloured with the same tone as the corresponding class (Orange for class 0 and Blue for class 1) with varying degrees of luminescence to distinguish the trails. It can be seen that the moons dataset with the following hyperparameters leads to two trails and 41 medoids: $k = 15$, `max_size`=20 and `min_count`=1 (in this case, the `min_count` parameter does not filter out any trails prior to the medoids retrieval, as the two classes are well separated and the two retrieved trails make up the whole dataset). For the circles dataset instead, 14 trails and 38 medoids are retrieved: $k = 5$, `max_size`=20 and `min_count`=10. In this case, it can be seen that part of the original dataset is not being taken into account for the trails: the portion of datapoints which are on the border between the two classes has been discarded. This allows to consider only representative datapoints of both classes. The hyperparameter `min_count` further filters out trails which are considered too small: the resulting condensed dataset presents a set of medoids which are equidistant on the two circumferences.

Of course, hyperparameter setting can influence the quality of the results: firstly, the parameter k represents the degree of sensibility of the trails with respect to the class constraint of the class. Lower values will likely retain a larger number of datapoints over which trails should be computed, while larger values will lead to more conservative results. Secondly, the `max_size` parameter determines the size of the condensed dataset. Class-specific sizes should be considered as to ensure that a minimum number of condensed datapoints is kept for each class. This becomes particularly relevant when considering unbalanced classes. Lastly, the `min_count` allows to focus only on more populated trails, limiting the effects of borderline datapoints or small trails which may not reflect the true dataset distribution.

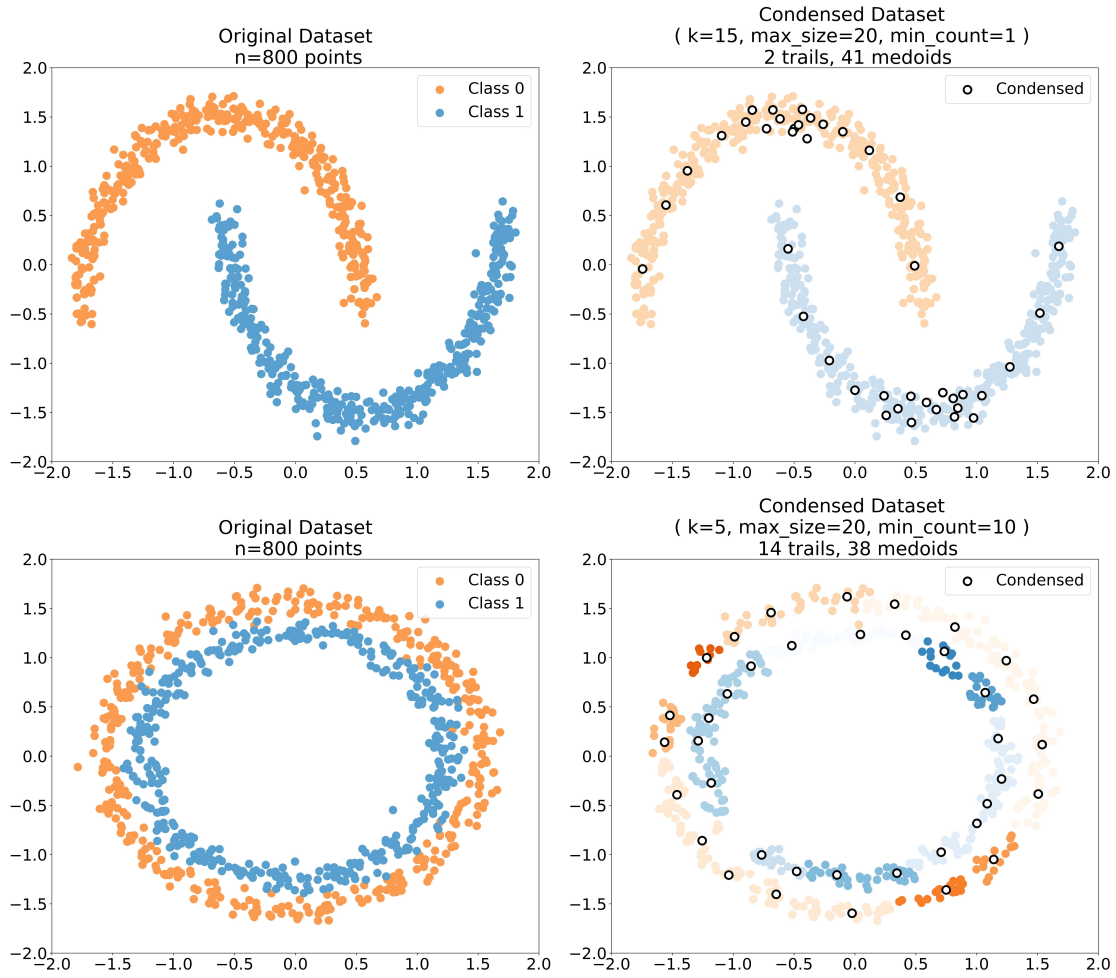


Figure 8.1: The dataset condensation approach on the moons (top row) and circles (bottom row) datasets. The left columns depicts the original training set (to which a small amount of random noise was added): in both cases, there are two distinct classes. The right column presents the condensed datasets (black circles) and depicts the trails over which the medoids were computed (coloured by class, different luminescence corresponds to a different trail).

8.2.2 k-NN Explanation

Consider two sets of datapoints, namely $\mathcal{N}_c$ and $\mathcal{N}_{\neg c}$, which correspond to the k_e nearest neighbours retrieved, respectively, between the condensed dataset $\mathcal{D}_{cond}$ datapoints belonging to class c and those of the opposing class $\neg c$. The average feature-wise distance within each set of points and the datapoint of interest $\mathbf{x}$ is computed and stored as D_c and $D_{\neg c}$. Assuming that $\mathbf{x}$ is predicted to belong to class c , the final feature attribution is the feature-wise difference between the two averages:

$$\mathbf{a} = D_{\neg c} - D_c \quad (8.1)$$

A final normalization can be applied to ensure comparability with other approaches (as will be the case in Chapter 9).

Algorithm 8.3 k -NN explanation

```

x: input point
f:  $k$ -nearest neighbours model

function AVERAGE_DISTANCE(x,  $\mathcal{N}$ )
   $D \leftarrow 1/|\mathcal{N}| \cdot \sum_{\mathbf{x}' \in \mathcal{N}} d(\mathbf{x}, \mathbf{x}')$ 
  return  $D$ 
end function

function EXPLAIN_KNN(x,  $k_e$ ,  $\mathcal{D}$ )
   $c = f(\mathbf{x})$  ▷ Class predicted by the  $k$ -nn model
   $\mathcal{D}_{cond} \leftarrow \text{CONDENSATION}(\mathcal{D}, \text{max\_size}, \text{min\_count})$ 
   $\mathcal{N}_c \leftarrow \text{K-NN}(k_e, \mathcal{D}_{cond}, \mathbf{x}|c)$  ▷  $k_e$  nearest neighbours belonging to class  $c$ 
   $\mathcal{N}_{\neg c} \leftarrow \text{K-NN}(k_e, \mathcal{D}_{cond}, \mathbf{x}|\neg c)$ 
   $D_c \leftarrow \text{AVERAGE\_DISTANCE}(\mathbf{x}, \mathcal{N}_c)$  ▷ Average feature-wise distance
   $D_{\neg c} \leftarrow \text{AVERAGE\_DISTANCE}(\mathbf{x}, \mathcal{N}_{\neg c})$ 
   $\mathbf{a} \leftarrow D_{\neg c} - D_c$ 
   $\mathbf{a} \leftarrow \mathbf{a} / \|\mathbf{a}\|_2$  ▷ Normalize to have norm 1
  return  $\mathbf{a}$ 
end function

```

The resulting normalized explanation assumes values in the $[-1, 1]$ range, with positive values associated with features which were relevant towards class c prediction and negative values otherwise. As per Property 4.2.2, a null value corresponds to a non-informative feature.

The attributions derived following Equation 8.1, in fact, assume positive value when the feature-wise difference between the datapoints belonging to the opposite class envelope greater differences (thus, distances) with respect to those of the same class. This aligns with the class separation inherent to k -NN: assume, for example, that for point $\mathbf{x}^i$ the j -th feature is associated with a positive attribution $a^{(i,j)} > 0$. This reflects that the j -th feature is important in the separation of class c and $\neg c$ as the feature assumes values more similar to the subset of k_e datapoints of the same class with respect to those of the opposite one. The larger the difference, the more the feature is relevant into separating the two possible classes.

This reasoning resonates with the choice of considering also datapoints which are outside of the k considered in the prediction phase, as it would otherwise hinder the ability of retrieving meaningful explanations in the case in which the k datapoints belong to the same class.

8.3 Explaining Random Forests

Explanations for Random Forests (Subsection 2.2.2) are often proposed in the form of single decision trees or sets of rules. Despite the high level of interpretability, these explanations are not directly comparable to feature attributions derived from other classes of models. For this reason, we have proposed a new feature attribution

method [133] tailored to Random Forests (Algorithm 8.4).

Consider a datapoint $\mathbf{x}$ which is passed through a random forest f made of B trees. Assume, with no loss of generality, that the random forest predicts point $\mathbf{x}$ belonging to class c , having predicted probabilities $[p_c, p_{-c}]$. For each of the B trees composing the forest, the datapoint's prediction is made by traversing the tree along a single decision path. According to the individual tree predicted class, the decision path may include features that contribute to the RF majority vote (and therefore the predicted class) or to the opposite one. As introduced in the Background section, each node in the classification decision tree is selected as the one with minimal node impurity, measured either by the Gini index or the entropy. Considering the relevance of such measures during the training phase, we proposed to use the node impurities as feature importances.

Specifically, consider two feature attribution vectors, namely η_c and η_{-c} , of length equal to the number of features in the input vector $\mathbf{x}$, say m . As outlined in Algorithm 8.4, for each tree in the forest, the decision path which is being followed by point $\mathbf{x}$ is retrieved and it is checked whether the prediction of the individual tree is the same as that of the random forest or not. For a tree agreeing with the RF prediction, the feature importance vector η_c is retrieved and, for all the features which are used at the split nodes of the traversed decision path, the corresponding node impurity is summed to the respective element of the vector η_c . The vector η_{-c} is otherwise used to store the impurities of the decision paths leading to the opposite class. A final calibration step is performed to properly take into account the effects of positively and negatively influencing features. In particular, the two above-defined feature importance vectors are weighted by the opposite class probability scores, returning a final vector of the form:

$$\mathbf{a} = (p_{-c} + \epsilon) \cdot \eta_c - p_c \cdot \eta_{-c} \quad (8.2)$$

where the parameter $\epsilon = 0.01$ is added to yield non-null coefficients in the case in which $p_{-c} = 0.00$. As for the k -NN explanation (Algorithm 8.3), it is possible to perform a final normalization step to ensure comparability.

The resulting normalized attributions assume theoretical value in the $[-1, 1]$ range, with positive values associated with features which are relevant towards the majority voted class and negative otherwise. Also in this scenario, a null value corresponds to non-informative features. It may be the case that a feature is never used in any of the traversed decision paths therefore it is, as expected, considered to be non influential.

While from a theoretical standpoint the attributions from Equation 8.2 can assume negative values and reach the lower bound of $a^{(i,j)} = -1$, it is hard to observe this behaviour in practice. In particular, it would be required not only that a feature j is used only in the decision paths leading to the opposite class prediction $-c$ but also that the decision trees composing the forest are not too unbalanced towards the majority class prediction. This ensures both that the feature is meaningful towards $-c$ class prediction and that it appears consistently across the trees in the forest.

In most scenarios though, the resulting feature attributions will present a positive sign and the magnitude will be diminished by considering also the paths which disagree with the final prediction.

Algorithm 8.4 RF explanation

```

x: input point
f: random forest model

function SUM_NODE_IMPURITY(path,  $\eta$ )
  for node  $\in$  path do
     $i \leftarrow \text{node.feature}$ 
     $\eta[i] \leftarrow \text{ADD}(\text{node.impurity})$ 
  end for
  return  $\eta$ 
end function

function EXPLAIN_RF(x, f)
   $p_c, p_{-c} \leftarrow f(\mathbf{x})$   $\triangleright$  Predicted class probabilities
   $c \leftarrow \arg \max(f(\mathbf{x}))$   $\triangleright$  Predicted class
   $\eta_c, \eta_{-c} \leftarrow \text{array}(0, \text{dim} = m)$ 
  for tree  $\in f$  do
    path  $\leftarrow \text{DECISION\_PATH}(\text{tree}, \mathbf{x})$   $\triangleright$  Follow the traversed path in the tree
    if tree.predict(x) = c then
       $\eta_c \leftarrow \text{SUM\_NODE\_IMPURITY}(\text{path}, \eta_c)$ 
    else
       $\eta_{-c} \leftarrow \text{SUM\_NODE\_IMPURITY}(\text{path}, \eta_{-c})$ 
    end if
  end for
   $\mathbf{a} \leftarrow (p_{-c} + \epsilon) \times \eta_c - p_c \times \eta_{-c}$ 
   $\mathbf{a} \leftarrow \mathbf{a} / \|\mathbf{a}\|_2$   $\triangleright$  Normalize to have norm 1
  return  $\mathbf{a}$ 
end function

```

This explanation strategy is more expressive than alternatives such as: feature counts across the decision paths leading to the RF prediction, features counts weighted by node depth along the decision paths or the same strategy as Algorithm 8.4 without considering the trees which disagree with the RF prediction. Moreover, introducing the node impurity within the explanation allows us to take into account the computed relevances of the training process, remaining faithful to the model's decision making process.

8.4 Results

8.4.1 Qualitative Evaluation

To evaluate the effectiveness of the proposed explanation, let us consider a set of toy datasets which illustrate the behaviour under controlled settings. The first example concerns the binary classification problem of two clouds of datapoints, both generated from a bivariate normal with mean vector $\mu^0 = (0, 0)$ and $\mu^1 = (4, 0)$

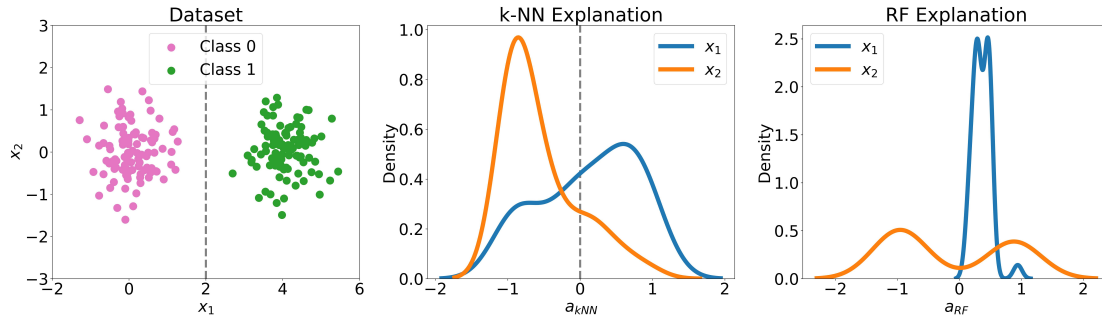


Figure 8.2: The first toy dataset with the two classes (left), the distribution of the explanations for feature x^1 (blue) and x^2 (orange) for the k -NN (centre) and the RF (right).

respectively. In both cases the covariance matrix is set to $\Sigma = 0.3 * \mathbf{I}_2$ (with $\mathbf{I}_2$ the identity matrix of size 2). A total of $n = 100$ datapoints were generated for each class, yielding two distinct clusters that can be linearly separated when considering only the first feature (Figure 8.2 - left). In this scenario then, we expect feature x^1 to have more relevance than feature x^2 towards the prediction.

A k -NN with $k = 5$ and a RF with $B = 10$ were trained on a training set (obtained with an 80 – 20% split) and reached 100% accuracy on both the training and test set. Both approaches were applied to the corresponding models and the obtained explanations for the test set are depicted in Figure 8.2 (centre for k -NN and right for RF). For each of the two features, x^1 and x^2 , the corresponding coefficient distribution is plotted, showcasing how feature x^1 is associated to larger coefficients and is therefore deemed as the most important feature among the two.

A trivial analysis would expect feature x^1 to be associated with a positive value while x^2 to a null one, denoting it as non important towards the prediction. This, of course, does not take into account the model reasoning and the way in which variables are used during a prediction. In particular, for the k -NN the importance of feature x^2 is skewed towards negative values while for the RF it spans across both negative and positive signs. This can be attributed to the way in which the trees are built: having a random selection of features, most of the trees take into account both features or only feature x^2 and, being the problem easy to solve, all the trees are predicting the majority class, effectively confirming the relevance (and usage) of both features for the prediction. Feature x^1 is associated with positive values for the RF and derives larger average coefficients due to the larger node impurities scores recorded within the decision paths of the forest.

A second toy dataset was built with the same modalities, having now three features of which the first two are generated from a bivariate normal with mean vector $\mu^0 = (0, 0)$ and $\mu^1 = (4, 4)$ and covariance matrices $\Sigma = 0.3 * \mathbf{I}_2$. The third variable is constructed as random noise $x^3 \sim \mathcal{N}(0, \sigma = 0.0001)$, denoting an irrelevant variable (Figure 8.3 - left - only shows the two-dimensional view on x^1 and x^2).

The k -NN and RF models were constructed with the same hyperparameters of the previous case: we expect both features x^1, x^2 to be associated with a degree of relevance as the dataspace can now be separated considering either of the two vari-

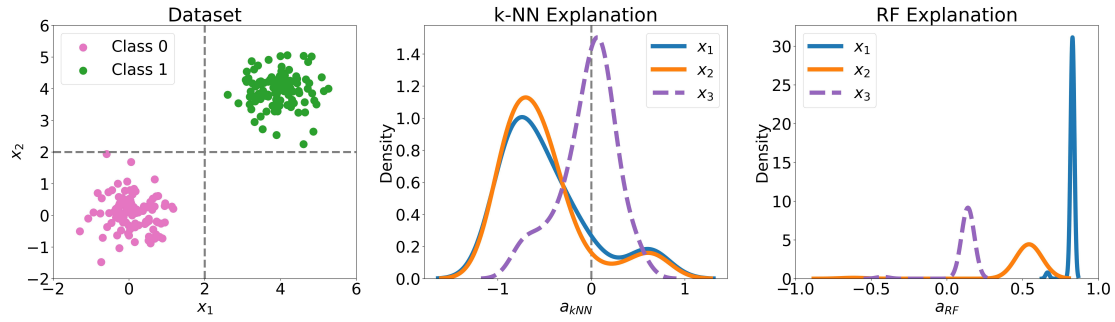


Figure 8.3: The second toy dataset with the two classes (left, bidimensional view), the distribution of the explanations for feature x^1 (blue), x^2 (orange) and x^3 (dashed purple) for the k -NN (centre) and the RF (right).

ables, while feature x^3 should be associated with null coefficients. As can be seen from the distribution of the attributions in the central and right panel of Figure 8.3, feature x^3 (dashed violet line) is centred around zero in both scenarios while features x^1, x^2 show both negative and positive attribution coefficients, denoting relevance towards the given class prediction or the opposing one. As previously said, the observed coefficients align with the models training phase and intrinsic characteristics. These two toy example aim at showcasing the efficacy of the proposed explanations: it must be taken into account that the lack of a ground truth in terms of the desired attribution vector for the individual models hinders the ability of effectively capturing the desired behaviour. Nonetheless, the proposed explanations take into account by construction the decision-making process of the models.

Hyperparameter tuning The k -NN explanation relies on the condensation step and the choice of how many neighbours to extract in the computation of the feature-wise distances (Algorithm 8.3). We found the following choices of the hyperparameters `max_size`, `min_count` and k_e to suit most use-cases, but a grid-search could be used to select more appropriate values for the dataset at hand or with the aim of providing a more robust explanation.

- `max_size`: should be chosen such that the average cluster size of the k -medoids step is, on average, $n = 10$. Larger `max_size` values yield a lower number of datapoints in the condensed dataset: it should be ensured that at least k_e datapoints are found per each class.
- `min_count`: a default value of 1 works well in most scenarios but, in the cases in which there are a lot of trails composed of very few datapoints, a threshold value could be selected to ensure that the representatives are extracted only from more populated trails (for example, setting `min_count` = k , with k the hyperparameter of the k -NN).
- $k_e = k$ ensures that the same number of datapoints are being evaluated both in the model and in the explanation's computation (class-wise).

Table 8.1: Average robustness scores (%) over the whole test set for the k -NN and the RF explanations.

Dataset	k -NN	RF
adult	82.54	88.67
bank	64.99	73.52
heloc	79.46	80.56
cancer	93.59	81.07
white wine	78.82	66.60

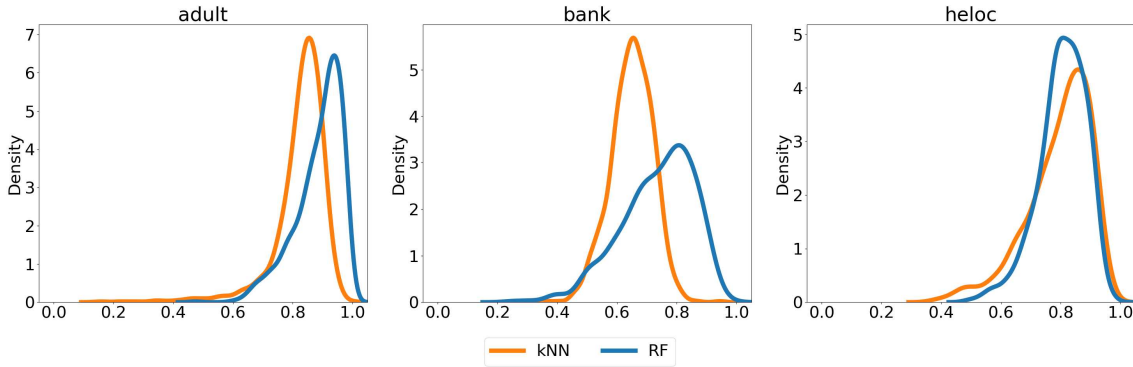


Figure 8.4: Distribution of the robustness scores for the two explanation approaches tailored for k -NN (orange) and RF (blue) models on the adult (left), bank (centre) and cancer (right) datasets.

8.4.2 Robustness Analysis

As they will be used in Chapter 9 for a multi-model aggregation, it is necessary to ensure that the proposed explanations are robust. This section will present the robustness analysis¹⁰ of the two approaches on a subset of the datasets introduced in Subsection 5.4.1, specifically addressing binary classification problems. Results from Table 8.1 and Figure 8.4 present the average robustness scores for the two approaches: it can be seen that the average robustness scores assume acceptable values, with the RF being preferable on the adult, bank and heloc dataset while the k -NN on the two simpler ones, namely cancer and white wine. The bank dataset (Figure 8.4) is associated to a higher degree of variability in terms of robustness for both approaches, while for the other datasets the variance is usually lower.

These results were derived from a medoid-based neighbourhood with $n = 100$ datapoints before the filtering step. Let it be noted that the two models take into account different neighbourhoods as prediction agreement must be ensured. As it will be more evident in the following Chapter (see Table 9.1), the models have comparable accuracy and can therefore be analysed in a similar fashion.

Additional insights The dataset condensation step introduces a significant decrease in the explanation’s standard deviation for the k -NN with respect to that originally proposed in [133]. Depending on the dataset, the variability decreased by

¹⁰The used code can be found at https://github.com/ilariavascotto/Aggregation_MultiModel

a factor of 30 to 50 %. This reflected also in larger robustness scores as the inherent variability necessarily affected also the rank correlations.

Similarly, for the RF explanation, the variability of the explanations affected the robustness. In this case though, the simpler versions of the explanations (mentioned at the end of Subsection 8.3) were not expressive enough to guarantee an accurate explanation. The proposed approach is instead more flexible and, despite yielding slightly lower robustness scores, was deemed preferable from an explanation point of view.

Chapter 9

Aggregating Explanations from Multiple Models

This Chapter focuses on the aggregation of explanations derived from different predictive models. In particular, the task of multi-model aggregation, as briefly introduced in Chapter 6, can be defined starting from $f_1, \dots, f_L$ predictive models of different nature. If each model is associated to a corresponding attribution method $\mathbf{a}_l$, then the aggregation can be defined as $\mathbf{a}_{agg} = g(\mathbf{a}_l | f_l)$.

9.1 Related Work

[135] is one of the few works that addresses this research question. It introduces the Weighted Importance Score and Frequency Count (WISFC) framework, investigating the disagreement problem in the context of feature rankings and proposing a more stable and reliable ranking, inspired by AGREE [94].

The WISFC framework integrates two metrics: the Weighted Importance Score (WIS) and the Frequency Count (FC). The first metric measures the average importance of a feature over the considered models, while the second how often it is within the top- k subset for each model. The combined metric is able to take into account two facets, presenting as important features that both have high magnitude and are consistently important across the methods.

The approach is global and model-agnostic, requiring only the explanations be expressed as feature importance vectors. To increase flexibility, the WIS of the j -th feature is computed with weights proportional to the performance (perf) of the individual models, having more accurate models being associated to a larger weight in the ranking.

$$WIS(j) = \sum_{l=1}^L \omega_l \cdot \mathbf{r}_l(j) \quad \text{with} \quad \omega_l = \frac{\exp(\text{perf}_l)}{\sum_{l=1}^L \exp(\text{perf}_l)} \quad (9.1)$$

Lower values of the WIS indicate that features are often placed in higher ranks across the models. After the selection of a relevance criterion, such as a rank or score threshold, the FC is computed as:

$$FC\%(j) = \frac{FC(j)}{L} \times 100 \quad (9.2)$$

It measures the overlap across the L models.

Following the WISFC framework, multiple aggregation strategies can be employed to obtain a final set of consistently important features. Unfortunately, despite being a model-agnostic method, it is tested only on relatively simple models and global feature attributions which may be unreliable. Specifically, the setting is as follows (chosen attributions are within brackets): logistic regression (coefficients), decision tree, Random Forest, XGBoost (Gini importance) and support vector machine (Kernel SHAP).

9.2 Multi-Model Aggregation

To introduce the task of multi-model explanation aggregation, it is necessary to define the models whose explanations are being merged. In particular, in this scenario we have decided to focus on the following three classes of models: k -Nearest Neighbours (Subsection 2.2.1), Random Forest (Subsection 2.2.2) and Neural Networks (Subsection 2.2.3). The choice of these models was driven by the differences in which the models make a prediction, having k -NN a distance-based method and RF an ensemble of rule-based base learners. Both significantly differ with respect to the NNs in terms of complexity, accuracy and inherent interpretability.

What will be described in the following is a natural generalization of the neural network-specific analysis discussed in [134]. As in the previous case, the class of explanations which we focused on is that of feature attributions. For the k -NN and RF we'll consider the tailored explanations introduced in Chapter 8, while for the NNs we will use DeepLIFT [112] (Section 4.5). Results from Chapter 5 showed that DeepLIFT's behaviour robustness-wise is, on average, more consistent than the considered alternatives for neural networks. This made it more suitable for a multi-model evaluation, even if any of the other approaches could be used instead.

The working setup of this proposal is focused around binary classification problems, instead of the multi-class ones previously discussed. This is a necessary simplification as it allows to focus more carefully on the models' agreement and perform a small adjustment when there are prediction disagreements.

Specifically, let f_l be a predictive model associated to a feature importance method $\mathbf{a}_l$ with $l \in (1, \dots, L)$. To compute an aggregative explanation, $\mathbf{a}_{agg} = g(\mathbf{a}_l | f_l)$, it is necessary that the explanations are referring to the same class. While in the single-model aggregation this was ensured by construction, as all explanations were referring to the same model, datapoint and prediction, it is now necessary to guarantee comparability. This can be done following two possible scenarios, assuming that $\{I, II, III\}$ represent a re-ordering of the models indices $\{1, 2, 3\}$:

- a. $f_I(\mathbf{x}^i) = f_{II}(\mathbf{x}^i) = f_{III}(\mathbf{x}^i)$ then the considered aggregations are of the form $\mathbf{a} = (\mathbf{a}_I, \mathbf{a}_{II}, \mathbf{a}_{III})$ from which $\mathbf{a}_{agg}$ can be computed.
- b. $f_I(\mathbf{x}^i) = f_{II}(\mathbf{x}^i) \neq f_{III}(\mathbf{x}^i)$ then the considered aggregations are of the form $\mathbf{a} = (\mathbf{a}_I, \mathbf{a}_{II}, -\mathbf{a}_{III})$ from which $\mathbf{a}_{agg}$ can be computed. This sign change ensures that the disagreeing model is considered with respect to the most voted class prediction-wise and, consequently, that the explanations are comparable.

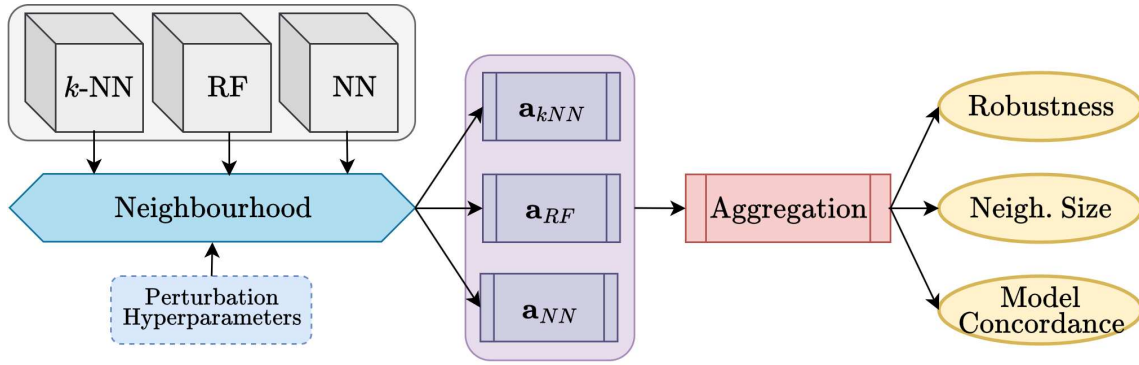


Figure 9.1: A representation of the proposed steps for multi-model explanations.

With this consideration in mind, an aggregated attribution can be crafted as [133]:

$$\mathbf{a}_{agg} = \frac{1}{L} \sum_{l=1}^L \mathbf{a}_l \text{ with } \|\mathbf{a}_l\|_2 = 1 \quad (9.3)$$

The average is computed feature-wise and does not require the final explanation to have norm 1, only that the composing approaches do.

To test the effectiveness of the proposed explanation, both from a qualitative and robustness standpoint, it is possible to follow a some-what similar analysis to that introduced for the NN case. More specifically, the robustness analysis can be performed in the same form but additional information can be derived to test whether model agreement is a symptomatic behaviour of explanations stability. Having now three different models at hand, model agreement is an even more necessary property for explanation trustworthiness. A synthetic representation of the proposed steps can be found in Figure 9.1, where the three models are applied to a properly-constructed neighbourhood prior to the individual attributions and the aggregation computation. Metrics of interest in this context are the robustness, the neighbourhood size and the validation on model concordance. They will be discussed more in details in the following Section.

9.3 Results

9.3.1 Dataset and Model Description

In this Chapter we considered a subset of 5 datasets from those described in Section 5.4.1, specifically limiting the analysis to binary classification problems: **adult**, **bank marketing**, **heloc**, **cancer** and **white wine**¹¹.

The datasets were processed as in the previously described scenario, being split in to a training, validation and test set, $\mathcal{D}_{train}$, $\mathcal{D}_{valid}$, $\mathcal{D}_{test}$ respectively. For all use-cases, the three models (k -NN, RF, NN) were trained on $\mathcal{D}_{train}$ and the resulting accuracy scores can be found in Table 9.1. The k -NNs (with Gower distance) had the number of neighbours set at $k = 15$ for the adult and bank dataset, and $k = 5$

¹¹UC Irvine Machine Learning Repository <https://archive.ics.uci.edu/datasets>

Table 9.1: Models accuracy (%) over the training set. Values in bold indicate the highest accuracy achieved for each dataset.

Dataset	k -NN	RF	NN
adult	81.45	84.85	91.39
bank	88.57	91.58	91.99
heloc	75.51	80.36	85.50
cancer	95.71	99.75	93.41
white wine	86.63	94.69	89.23

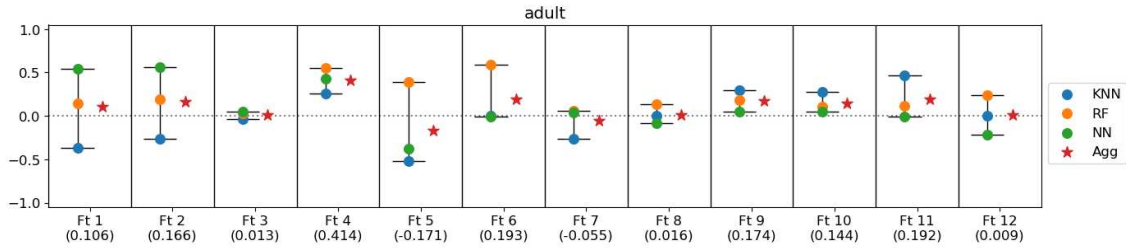


Figure 9.2: An example of the feature attributions derived on the adult dataset. The k -NN (blue), RF (orange) and NN (green) explanations are connected by a vertical black line (observed range). The red star represents the aggregation, whose value is presented within brackets close to the corresponding feature index (1 to 12).

in the other cases. The RFs were built with $B = 25$ base learners for all cases. The NNs correspond to those used in Chapter 5 and to Model 1 in Chapter 7 (the NN column of Table 9.1 coincides with the Model 1 in Table 7.3 for the selected datasets).

All models achieved an adequate performance, as the accuracy scores are above the 80% mark, with the only exception being the k -NN on the heloc dataset. As expected, the k -NNs are often associated to the lowest accuracy (being instances of simpler models) while the NNs excel in the most complex datasets (adult, bank, heloc) and the RF in the two simpler ones (cancer, wine).

9.3.2 Qualitative Evaluation

This Subsection presents a brief discussion of an example of aggregation, with a datapoint $\mathbf{x}^i \in \mathcal{D}_{test}$ of the adult dataset being explained. A graphical representation can be found in Figure 9.2 where, for each of the twelve features, the three individual normalized explanations (coloured circles) and the aggregation (red star) are depicted. This example illustrates two key aspects of the aggregation: first, cases in which the three explanations have coefficients very close to zero (and possibly differing in sign) are shrunk towards the zero (that is, the feature is considered to be non important) - Features 3, 7, 8, 12. Secondly, when features are differing in sign and the magnitudes are large, the aggregation penalizes the disagreement but produces a coefficient which is coherent with the stronger signal - Features 1, 2, 5. Lastly, concordance between the three methods is maintained - Features 4, 5, 9, 10, 11. This is an expected (and desirable) behaviour as consistent information is kept and dis-

agreements are penalized more strongly according to the magnitude.

9.3.3 Robustness Analysis

A robustness evaluation akin to Chapter 5 can be proposed also in the scenario in which multiple models are being evaluated at once¹². In the context of the neighbourhood generation (Section 5.3), a clarification must be made. Specifically, focusing on the filtering step, two scenarios now arise:

- Single-model evaluation: the filtering removes all perturbations that change the predicted class label with respect to the original one, i.e. $f(\tilde{\mathbf{x}}^i) = f(\mathbf{x}^i)$
- Multi-model aggregation: the filtering removes all the perturbations that change at least one class label with respect to the L considered model predictions. While it is not necessary that the L models agree on the prediction, the class agreement pattern must be ensured within the neighbourhood. Assuming that $\{I, II, III\}$ represents a re-ordering of the models indices $\{1, 2, 3\}$, with no loss of generality, two scenarios may occur:

- a. $f_I(\mathbf{x}^i) = f_{II}(\mathbf{x}^i) = f_{III}(\mathbf{x}^i)$ then $f_I(\tilde{\mathbf{x}}^i) = f_{II}(\tilde{\mathbf{x}}^i) = f_{III}(\tilde{\mathbf{x}}^i) \forall \tilde{\mathbf{x}}^i \in \mathcal{N}^i$
- b. $f_I(\mathbf{x}^i) = f_{II}(\mathbf{x}^i) \neq f_{III}(\mathbf{x}^i)$ then $f_I(\tilde{\mathbf{x}}^i) = f_{II}(\tilde{\mathbf{x}}^i) \neq f_{III}(\tilde{\mathbf{x}}^i) \forall \tilde{\mathbf{x}}^i \in \mathcal{N}^i$

Results from both Figure 9.3 and Table 9.2 distinguish between the two types of filtering according to the model being explained. In particular, the initial neighbourhood $\mathcal{N}^i$ is the same for all models, but it is filtered according to the individual model prediction for the k -NN, RF and NN models while the joint prediction pattern is considered within the Aggregation robustness computation. Hyperparameter tuning for the neighbourhood ensures that individual model's neighbourhood retains an average size of about 95% of the generated datapoints. As it was shown in the previous chapters, the neighbourhood generating scheme used in these experimental evaluation is the medoid-based one (Subsection 5.3.2), as it contributes to larger robustness scores and its perturbations are more faithful to the data distribution.

Figure 9.3 presents the distributions of the robustness scores on the test set for the bank (left) and heloc (right) datasets. It can be seen that the heloc dataset is generally associated with larger robustness scores for all methods under consideration, while for the bank one k -NN appears to be less robust than its counterparts. The aggregated explanation acts, expectedly, as an average robustness-wise between the three individual components in both scenarios.

Table 9.2 presents the average robustness scores (in percentage) over the test set. It can be seen that NN is the method with largest robustness on average, while RF outperforms k -NN on the more complex datasets while it underperforms on the two simpler ones. The robustness scores are more or less high depending on both the complexity of the datasets and the model concordances. For example, while the aggregation is associated with the largest average robustness for the heloc dataset, the distribution of the robustness scores for the NN (Figure 9.3) is associated with slightly larger scores with respect to the aggregation. Considering the medians of

¹²The used code can be found at https://github.com/ilariavascotto/Aggregation_MultiModel

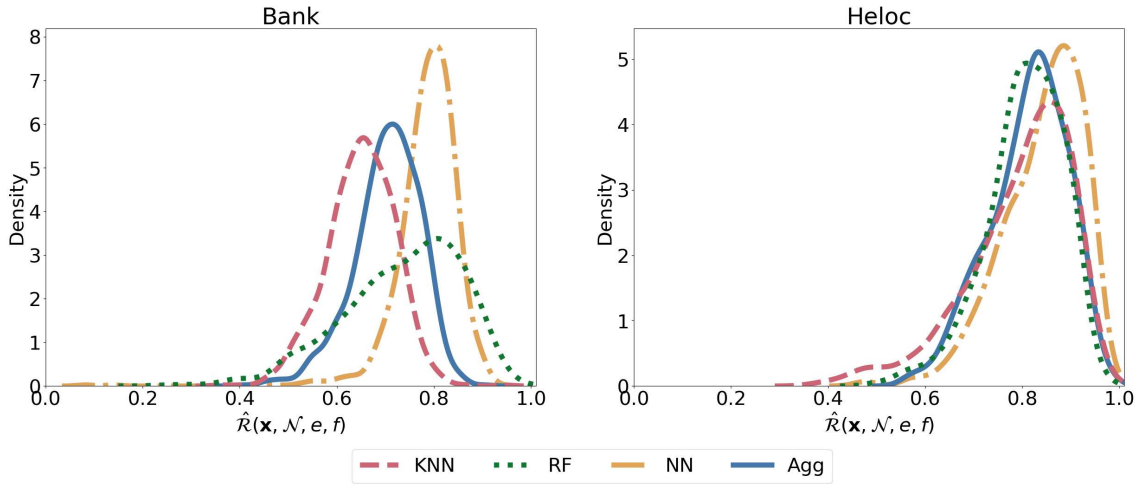


Figure 9.3: Distribution of the robustness scores for the three models and their aggregation on the bank (left) and heloc (right) datasets.

Table 9.2: Average robustness scores (%) over the whole test set. Values in bold indicate the highest robustness achieved within each dataset.

Dataset	k -NN	RF	NN	Agg.
adult	82.54	88.67	85.03	85.31
bank	64.99	73.52	78.74	74.79
heloc	79.46	80.56	84.23	84.50
cancer	93.59	81.07	98.40	87.23
white wine	78.82	66.60	92.96	71.72

the two distributions, in fact, the NN’s median is larger than the aggregation one ($85.81 > 82.59$), despite having similar average values.

The advantage of using the aggregation in place of an individual model-explanation pair is bifold. On one hand, the aggregation’s robustness is still acceptable but, on the other, it encompasses the explanations derived from multiple models. In practice, it acts as a conservative and trustworthy explainer.

9.3.4 Additional Metrics and Validation

Having now three predictive models under consideration, it is possible to define a new robustness-related metric to aid the understanding of the trustworthiness of an explanation. The main assumption which governed the choice of this metric is that robust explanations usually lie in a robust area of the data manifold. Results from Table 9.3 show the average neighbourhood size associated with the aggregation, starting from $n = 100$ datapoints and after the filtering step. The average neighbourhood size varies greatly depending on the dataset being analysed (columns Test %). The remaining columns of the table show the number of observations in the test set (n_{obs}) and corresponding average neighbourhood size for the datapoints over which the three models agree on the prediction (Agree) and those for which one model (which names the column) predicts the opposite class with respect to the

Table 9.3: Comparison between the models prediction agreement (n_{obs}) and average neighbourhood size in percentage (after filtering step). The columns Test presents the results over the whole test set, while the column Agree over the datapoints for which the three models agree on the prediction. The remaining columns are named by the model which disagrees on the prediction. Values in bold in each columns for the percentage indicate the largest neighbourhood size achieved per dataset.

Dataset	Test		Agree		k -NN		RF		NN	
	n_{obs}	%	n_{obs}	%	n_{obs}	%	n_{obs}	%	n_{obs}	%
adult	1000	77.41	802	77.78	100	75.86	35	73.80	63	77.14
bank	1000	89.16	929	91.03	30	70.30	10	49.10	31	64.35
heloc	500	67.23	358	67.72	70	61.27	24	73.25	48	69.25
cancer	50	84.02	46	89.61	0	N/A	3	25.67	1	2.00
white wine	200	78.04	176	81.97	12	56.50	6	39.33	6	44.67

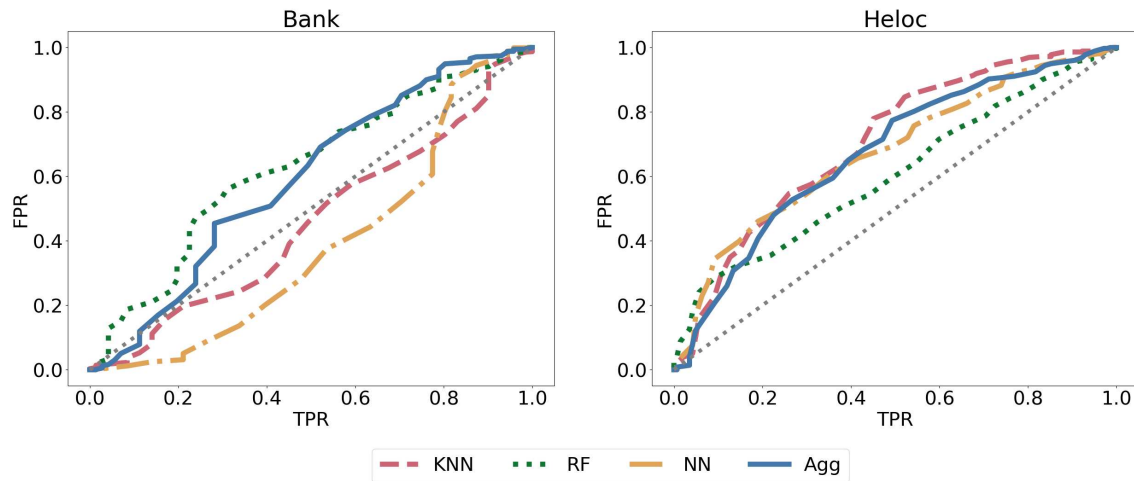


Figure 9.4: ROC curves for the three models and the aggregation for the bank (left) and heloc (right) datasets. The gray dotted line represents the bisecting line.

Table 9.4: AUC values computed for the three models and the aggregation. Values in bold indicate the largest score recorded per dataset.

Dataset	k -NN	RF	NN	Agg
adult	0.5510	0.5417	0.6970	0.6682
bank	0.4558	0.6257	0.3861	0.5913
heloc	0.7059	0.6049	0.6748	0.6737
cancer	0.8288	0.9212	0.7120	0.8967
white wine	0.5379	0.4698	0.0469	0.5696

other two. Values in bold in the columns % are associated with the largest observed neighbourhood sizes after the filtering step. It can be noted that, in most cases, the largest size is achieved under the full concordance of the three models, effectively supporting the robustness to model agreement linkage. Note that, for the cancer datasets, there are 0 datapoints for which the k -NN is the disagreeing model, therefore the corresponding value N/A indicates that no data was available for the specific scenario during our experimental evaluation.

Results from Table 9.3 suggest that, even when considering models of different ML classes, it is possible to investigate more deeply the relationship between explanation robustness and model agreement. To verify this assumption, we have followed the validation proposal of Section 7.3, computing both the ROC curve (Figure 9.4) and AUC (Table 9.4) with the TPR and FPR tailored for the robustness-agreement evaluation. Results from both the Figure and the Table suggest that the aggregation can act as a good explanation, being - in most cases - the method associated to the second largest AUC value. The preferable method varies dataset-wise and is, interestingly, not associated with a unique approach. Additionally, for the bank dataset it is possible to note that the k -NN and NN ROC curves are below the bisecting line and are, in fact, also associated with AUC values below the 0.50 mark. This behaviour was not appreciated for the other datasets but it may be due to the lower average robustness scores achieved by the two methods (as per Table 9.2).

9.3.5 Comparison to the Ensemble

Aiming at investigating explanation robustness with respect to the aggregation method, we now present a comparison of the above-defined aggregation to the ensemble introduced in Chapter 6. Critically, a few considerations must be made. First of all, the results associated with the Ensemble will be compared to the robustness of the explanations taken in absolute value, to ensure comparability. This necessarily implies that the results for the individual attributions may differ from what was presented up to now in this Chapter. Secondly, the considered attributions were normalized to have norm 1 for comparability purposes. The ensemble was built from un-normalized vectors, therefore it is necessary to slightly modify the weights of Equation 6.8 to ensure that the variability of the attribution vectors is correctly taken into account when computing the ranking.

In the following we will compare three aggregation approaches:

- the aggregation of Equation 9.3 (Agg.);
- the ensemble as per Equation 6.7 (Ensemble or Ens.);
- the ensemble with modified weights (Ensemble with $\omega = 1$).

The Ensemble with $\omega = 1$ is a modified version of the Ensemble, where the weights were simplified in order to better capture the dynamics between the feature attribution vectors considered in this scenario. More specifically, the weights are set to $w_l^{(i,j)} = 1$ for all the features while the penalization term is left untouched. This allows the aggregation to be more strictly dependant on the computed rankings and

Table 9.5: Average robustness score (%) over the whole test set. Values in bold indicate the highest robustness achieved within each dataset. The individual attributions are evaluated with absolute value for comparability.

Dataset	Ensemble	Ens. $\omega = 1$	k -NN	RF	NN
adult	82.04	84.76	84.31	90.69	88.46
bank	69.85	77.84	70.14	82.28	82.76
heloc	72.95	77.74	79.08	75.60	83.12
cancer	80.51	87.24	91.17	84.32	96.51
white wine	49.26	56.82	71.95	58.80	88.39

the penalization term. As in the previous case, the ensemble is computed on the feature rankings in absolute values and is, as well as the Ensemble, compared to the individual attributions taken in absolute value.

Table 9.5 presents the average robustness scores achieved over the whole test set. The left most part of the table presents the robustness of the two ensemble methods (Ensemble and Ens. $\omega = 1$), denoting how the second approach is consistently more robust than the first one. The right most part instead presents the average robustness of the absolute valued composing attributions: the NN appears to be the most robust approach in this context. Figure 9.5 (left) presents the distribution of the robustness scores for the bank dataset, showcasing how the ensemble $\omega = 1$ (purple) is preferable with respect to the ensemble (light blue) and presents, as expected, an average behaviour robustness-wise.

An effective comparison of the approaches can be performed via the ROC/AUC analysis previously introduced, which allows us to compare the different aggregations even if one is computed on the absolute valued ranking and the other on the ranks with sign. Figure 9.5 (right) depicts the ROC curves obtained on the bank dataset for the Aggregation (dashed blue), Ensemble (light blue) and Ensemble with $\omega = 1$ (purple). The curves behaviour is generally preferable with respect to what was observable in Figure 9.4.

Table 9.6 instead showcases the AUC values obtained for the three aggregation methods and the three models (with absolute sign). According to the dataset, the Aggregation and the modified Ensemble are associated with the largest AUC scores in all considered datasets. Notably, the k -NN results in the lowest AUC scores, denoting the largest deviation from the AUC of Table 9.4. Overall, the aggregation methods are best suited to describe the robustness-agreement relationship.

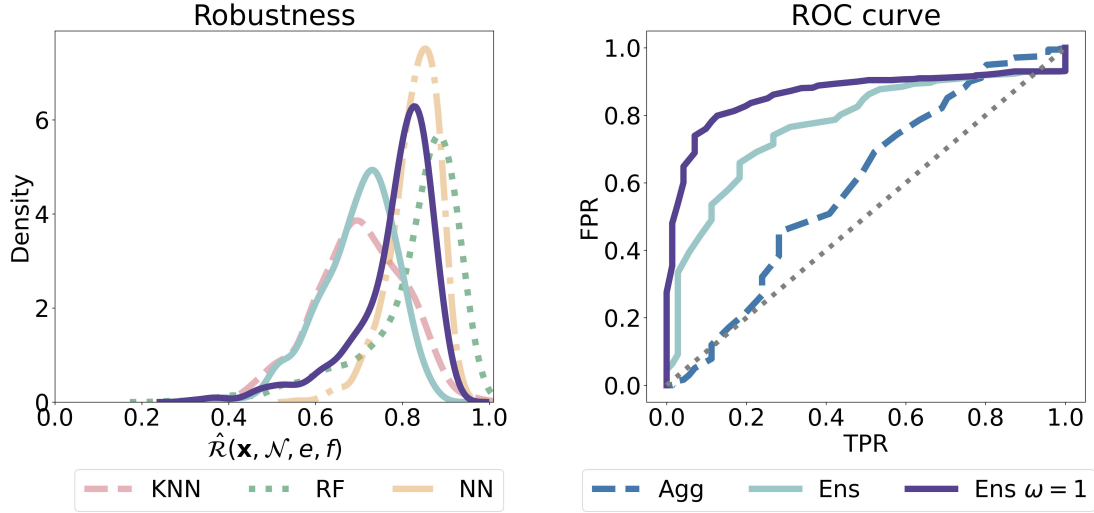


Figure 9.5: Distribution of the robustness scores on the bank dataset for the individual attributions in absolute value and the two ensemble approaches (left); ROC curves for the three aggregations (right). The gray dotted line represents the bisecting line.

Table 9.6: AUC values computed for the three aggregations (left) and the individual methods (right), taken in absolute sign. Values in bold indicate the largest score recorded per dataset.

Dataset	Agg	Ens	Ens. $\omega = 1$	$ k\text{-NN} $	$ RF $	$ NN $
adult	0.6682	0.4171	0.6780	0.2121	0.4818	0.6074
bank	0.5913	0.7721	0.8624	0.3117	0.7582	0.5215
heloc	0.6737	0.5433	0.5309	0.3276	0.4584	0.4347
cancer	0.8967	0.8750	0.8967	0.4348	0.8913	0.6168
white wine	0.5696	0.5530	0.4929	0.3539	0.5407	0.0477

Chapter 10

Conclusions

This Thesis investigated the role of explanations robustness and the effectiveness of aggregation approaches to enhance the trustworthiness of AI systems in practical applications.

As seen in the previous Chapters, we devised a set of desirable properties that robustness estimators should satisfy, introducing a novel neighbourhood generation mechanism. We addressed the disagreement problem by proposing a novel aggregation method to merge explanations in a single-model aggregation scenario, focusing on the ranking of the features and we introduced a framework to test explanation robustness to non-adversarial perturbations, as well as a novel validation assessment of the robustness estimation, to tackle the lack of a ground truth.

In practical applications, this approach aids practitioners in understanding the quality of an explanation in terms of its robustness, allowing questioning on the proposed results when the framework flags a datapoint as *uncertain*.

Advances with respect to state of the art approaches have been introduced with this work, specifically tackling the lack of comparable metrics (our approach is bounded therefore comparisons are possible both when changing the XAI method but also the model or the dataset). The robustness estimation leverages an innovative neighbourhood generating scheme which is on-manifold, a consideration which is often overlooked in other works, and constraints the robustness computation to homogeneous datapoints prediction-wise. Again, the approach allows for comparability and enhances the trustworthiness. Moreover, we have introduced a novel aggregation approach: despite being a powerful approach to deal also with the disagreement problem, aggregations are often investigated only in the context of adversarial attacks. Finally, the validation assessment of the robustness is a novel introduction in the literature, providing a solution to tackle the lack of a ground truth for the robustness in real use-cases, posing itself as an improvement over state of the art approaches.

In Chapter 7, we have investigated the reliability of explanations in unbalanced datasets. The analysis introduced a new metric, known as consistency, that - when used alongside the robustness score - helps investigating the reliability of the explanations for the minority class. Unbalanced datasets are frequently used in the insurance and financial fields (for example, in churn prediction or credit card frauds) and the reliability of explanations for the minority class is an important aspect which goes overlooked at times.

Practical applications such as highly unbalanced use-cases are not common in research papers, as the focus is more often put onto balanced scenarios or performance-based analysis. Real-world high-risk scenarios instead make frequent use of unbalanced datasets, as unpredictable or less frequent events are of particular interest. Our proposal is a first step towards a reliability analysis of explanations for the minority class (the one of interest), effectively addressing an open research question.

The Thesis further considered the scenario of multi-model aggregation, where we have presented two novel attribution approaches for k -Nearest Neighbours and Random Forest. The disagreement problem was addressed via an aggregation approach that considered the difference between the decision-making processes of multiple ML models as well as explanation and prediction disagreements between the model-explainer pairs. The usage of multiple models and associated explanations is beneficial from a practitioners point of view to detect commonly misused features or patterns in the data, while leveraging the predictive power of multiple models.

Our proposal introduced two new feature importance methods, tailored for two classes of models which often rely on other types of explanations (or are considered interpretable altogether). The variability and amount of the XAI approaches proposed up to now in the literature can be both beneficial and damaging. While on one hand it is possible to choose the best suited method for the problem at hand, on the other the lack of standardized criteria to validate explanations implies that the choices are often based on non-objective criteria. Our proposal introduces two new approaches in a class of explanations, that of feature attributions, which is more interpretable also for non-experts and that allows for comparisons among different ML models being explained.

Moreover, the aggregation of multi-model explanations investigates a mostly unbeaten research direction in the XAI field, especially when considering local explanations. The devised approach leverages previous results presented in this Thesis but is innovative due to its focus on reliability and robustness of the explanations.

The works presented in this Thesis investigated the trustworthiness and reliability of explanations, allowing practitioners to use AI/ML systems under the new EU legislative requirements. The usage of our tools aids both the practitioners in understanding the mechanics of the *black box* models which are used in production as well as protecting the rights of the end users, allowing for a fairer approach to AI tools.

10.1 Future Work

The advancements proposed in this Thesis could benefit from the following research directions for future works.

First of all, a promising research direction is the testing of the relationship between robustness and adversarial attacks, as we have focused on non-adversarial perturbations during the presented works. In particular, we wish to assess the defence ability of the ensemble, as aggregated explanations have proved to be more resilient to adversarial attacks in multiple contexts, and to validate whether the robustness estimator is able to detect attacks, as out-of-manifold perturbations are associated with lower robustness scores.

For the unbalanced dataset evaluation, future work will include the analysis of neural network performance and its impact on the reliability. Moreover, the approach could be extended to other classes of ML models in order to improve the performance on unbalanced datasets, adapting the choice of XAI approaches and validation metrics accordingly.

In the context of multi-model aggregations, the first natural extension of the work will aim at considering different classes of models and XAI approaches applied to them, spreading the possible usages and aggregation strategies according to the use-case at hand. In this regard, the role of aggregations will be further developed as it was shown in Chapter 9 that multiple aggregation strategies can produce equally satisfactory results.

On a more general note, explanation aggregations will be investigated also for different data-types and spanning from classification to regression tasks. The latter one requires minor adjustments in the neighbourhood generating process of the robustness analysis, specifically in the filtering step, as a sensibility threshold must be set to discriminate effectively between close predictions.

Finally, all the presented work would benefit from user-studies to investigate whether the proposed approaches increase the trustworthiness of the explanations (and of the AI/ML system) from a practical point of view.

Bibliography

- [1] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.
- [2] C. Agarwal, S. Krishna, E. Saxena, M. Pawelczyk, N. Johnson, I. Puri, M. Zitnik, and H. Lakkaraju. Openxai: Towards a transparent evaluation of model explanations. *Advances in neural information processing systems*, 35:15784–15799, 2022.
- [3] D. Alvarez-Melis and T. Jaakkola. On the robustness of interpretability methods, 06 2018.
- [4] D. Alvarez-Melis and T. S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, page 7786–7795, Red Hook, NY, USA, 2018. Curran Associates Inc. URL <https://hdl.handle.net/1721.1/137669>.
- [5] C. J. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel. Fairwashing explanations with off-manifold detergent. In *Proceedings of the 37th International Conference on Machine Learning*, ICML’20. JMLR.org, 2020.
- [6] D. W. Apley and J. Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):1059–1086, 06 2020. ISSN 1369-7412. doi: 10.1111/rssb.12377. URL <https://doi.org/10.1111/rssb.12377>.
- [7] L. Arrighi, L. Pennella, G. Marques Tavares, and S. Barbon Junior. Decision predicate graphs: Enhancing interpretability in tree ensembles. In *World Conference on Explainable Artificial Intelligence*, pages 311–332. Springer Nature Switzerland, 2024. ISBN 978-3-031-63797-1. doi: 10.1007/978-3-031-63797-1_16.
- [8] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

- [9] 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. doi: 10.1371/journal.pone.0130140.
- [10] H. Baniecki and P. Biecek. Adversarial attacks and defenses in explainable artificial intelligence: A survey. *Information Fusion*, 107:102303, 2024. ISSN 1566-2535. doi: 10.1016/j.inffus.2024.102303.
- [11] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020. ISSN 1566-2535. doi: <https://doi.org/10.1016/j.inffus.2019.12.012>. URL <https://www.sciencedirect.com/science/article/pii/S1566253519308103>.
- [12] U. Bhatt, A. Weller, and J. M. F. Moura. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI’20, 2021. ISBN 9780999241165. URL <https://dl.acm.org/doi/abs/10.5555/3491440.3491857>.
- [13] S. Bobek, P. Bałaga, and G. J. Nalepa. Towards model-agnostic ensemble explanations. In M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, editors, *Computational Science – ICCS 2021*, pages 39–51, Cham, 2021. Springer International Publishing. ISBN 978-3-030-77970-2.
- [14] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, and S. Rinzivillo. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 37(5):1719–1778, 2023.
- [15] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [16] N. O. Breuer, A. Sauter, M. Mohammadi, and E. Acar. Cage: Causality-aware shapley value for global explanations. In L. Longo, S. Lapuschkin, and C. Seifert, editors, *Explainable Artificial Intelligence*, pages 143–162, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-63800-8.
- [17] CAHAI. Possible elements of a legal framework on artificial intelligence, based on the Council of Europe’s standards on human rights, democracy and the rule of law., 2021. <https://rm.coe.int/cahai-2021-09rev-elements/1680a6d90d>.
- [18] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37456-2.
- [19] L. Cao. Ai in finance: Challenges, techniques, and opportunities. *ACM Comput. Surv.*, 55(3), Feb. 2022. ISSN 0360-0300. doi: 10.1145/3502289. URL <https://doi.org/10.1145/3502289>.

- [20] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intel-ligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Confer-ence on Knowledge Discovery and Data Mining*, KDD '15, page 1721–1730, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2788613. URL <https://doi.org/10.1145/2783258.2788613>.
- [21] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso. Machine learning inter-pretability: A survey on methods and metrics. *Electronics*, 8(8), 2019. ISSN 2079-9292. doi: 10.3390/electronics8080832. URL <https://www.mdpi.com/2079-9292/8/8/832>.
- [22] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Interpreting neural network classifications with variational dropout saliency maps. In *Proc. NIPS*, volume 1, pages 1–9, 2017.
- [23] M. Cinquini and R. Guidotti. Causality-aware local interpretable model-agnostic explanations. In *Explainable Artificial Intelligence*, pages 108–124, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-63800-8.
- [24] P. Cortez and M. J. Embrechts. Opening black box data mining models using sensitivity analysis. *IEEE SSCI 2011: Symposium Series on Computational Intelligence - CIDM 2011: 2011 IEEE Symposium on Computational Intelli-gence and Data Mining*, pages 341–348, 2011. <https://doi.org/10.1109/CIDM.2011.5949423>.
- [25] I. C. Covert, S. Lundberg, and S.-I. Lee. Understanding global feature con-tributions with additive importance measures. In *Proceedings of the 34th In-ternational Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- [26] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained neural networks. *Advances in Neural Information Processing Systems*, 8:24–30, 1996. <https://dl.acm.org/doi/10.5555/2998828.2998832>.
- [27] S. Dasgupta, N. Frost, and M. Moshkovitz. Framework for evaluating faithfulness of local explanations. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th In-ternational Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4794–4815. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/dasgupta22a.html>.
- [28] K. Dedja, F. K. Nakano, K. Pliakos, and C. Vens. Bellatrex: Building explana-tions through a locally accurate rule extractor. *IEEE Access*, 11:41348–41367, 2023. doi: 10.1109/ACCESS.2023.3268866.
- [29] K. Dhamdhere, M. Sundararajan, and Q. Yan. How important is a neuron? *CoRR*, abs/1805.12233, 2018. URL <http://arxiv.org/abs/1805.12233>.

- [30] A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel. Explanations can be manipulated and geometry is to blame, 2019. URL <https://arxiv.org/abs/1906.07983>.
- [31] A.-K. Dombrowski, C. J. Anders, K.-R. Müller, and P. Kessel. Towards robust explanations for deep neural networks. *Pattern Recognition*, 121:108194, 2022. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2021.108194>. URL <https://www.sciencedirect.com/science/article/pii/S0031320321003769>.
- [32] P. Domingos. Knowledge discovery via multiple models. *Intelligent Data Analysis*, 2(3):187–202, 1998. <https://doi.org/10.3233/IDA-1998-2303>.
- [33] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning, 2017. URL <https://arxiv.org/abs/1702.08608>.
- [34] M. Ebers. Regulating explainable ai in the european union. an overview of the current legal framework(s). 2021. <http://dx.doi.org/10.2139/ssrn.3901732>.
- [35] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.
- [36] M. Ester, H.-P. 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, page 226–231. AAAI Press, 1996.
- [37] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), 2016. <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.
- [38] European Commission. Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts (COM(2021) 206 final), 2021. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206>.
- [39] European Commission. Draft standardisation request to the european standardisation organisations in support of safe and trustworthy artificial intelligence., 2022. <https://ec.europa.eu/docsroom/documents/52376>.
- [40] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, 2017. doi: 10.1109/ICCV.2017.371.
- [41] A. A. Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, 2014.

- [42] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001. <https://doi.org/10.1214/aos/1013203451>.
- [43] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688, Jul. 2019. doi: 10.1609/aaai.v33i01.33013681.
- [44] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015. <https://doi.org/10.1080/10618600.2014.907095>.
- [45] J. Goldwasser and G. Hooker. Stabilizing estimates of shapley values with control variates. In L. Longo, S. Lapuschkin, and C. Seifert, editors, *Explainable Artificial Intelligence*, pages 416–439, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-63797-1.
- [46] A. Gosiewska and P. Biecek. Do not trust additive explanations. *arXiv: Learning*, 2019. URL <https://doi.org/10.48550/arXiv.1903.11420>.
- [47] A. H.-L. E. Group. Ethics guidelines for trustworthy ai, 2019. URL <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>. B-1049 Brussels.
- [48] R. Guidotti and S. Ruggieri. Ensemble of counterfactual explainers. In C. Soares and L. Torgo, editors, *Discovery Science*, pages 358–368, Cham, 2021. Springer International Publishing. ISBN 978-3-030-88942-5.
- [49] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems. *CoRR*, abs/1805.10820, 2018. URL <http://arxiv.org/abs/1805.10820>.
- [50] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), Aug. 2018. ISSN 0360-0300. doi: 10.1145/3236009. URL <https://doi.org/10.1145/3236009>.
- [51] R. Guidotti, A. Monreale, S. Matwin, and D. Pedreschi. Black box explanation by learning image exemplars in the latent feature space. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, page 189–205, Berlin, Heidelberg, 2019. Springer-Verlag. ISBN 978-3-030-46149-2. doi: 10.1007/978-3-030-46150-8_12. URL https://doi.org/10.1007/978-3-030-46150-8_12.
- [52] J. P. Göpfert, H. Wersing, and B. Hammer. Interpretable locally adaptive nearest neighbors. *Neurocomputing*, 470:344–351, 2022. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2021.05.105>. URL <https://www.sciencedirect.com/science/article/pii/S0925231221011097>.

- [53] R. Hamon, H. Junklewitz, I. Sanchez, G. Malgieri, and P. De Hert. Bridging the gap between ai and explainability in the gdpr: Towards trustworthiness-by-design in automated decision-making. *IEEE Computational Intelligence Magazine*, 17:72–85, 2022. <https://doi.org/10.1109/MCI.2021.3129960>.
- [54] A. Hedström, L. Weber, D. Bareeva, D. Krakowczyk, F. Motzkus, W. Samek, S. Lapuschkin, and M. M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *J. Mach. Learn. Res.*, 24(1), Jan. 2023. ISSN 1532-4435.
- [55] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020. doi: 10.1002/qj.3803.
- [56] X. Huang and J. Marques-Silva. The inadequacy of shapley values for explainability, 2023. URL <https://arxiv.org/abs/2302.08160>.
- [57] H. Hwang, A. Bell, J. Fonseca, V. Pliatsika, J. Stoyanovich, and S. E. Whang. Shap-based explanations are sensitive to feature representation. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency, FAccT '25*, page 1588–1601, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400714825. doi: 10.1145/3715275.3732105. URL <https://doi.org/10.1145/3715275.3732105>.
- [58] S. R. Islam, W. Eberle, and S. K. Ghafoor. Towards quantification of explainability in explainable artificial intelligence methods. *Proceedings of the 33rd International Florida Artificial Intelligence Research Society Conference, FLAIRS 2020*, page 75–81, 2020. <https://arxiv.org/abs/1911.10104>.
- [59] N. Jethani, M. Sudarshan, I. Covert, S.-I. Lee, and R. Ranganath. Fastshap: Real-time shapley value estimation, 2022. URL <https://arxiv.org/abs/2107.07436>.
- [60] A. Jha, J. K. Aicher, M. R. Gazzara, D. Singh, and Y. Barash. Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study. *Genome biology*, 21(1):149, 2020.
- [61] M. A. Kadir, A. Mosavi, and D. Sonntag. Evaluation metrics for xai: A review, taxonomy, and practical applications. In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*, pages 000111–000124, 2023. doi: 10.1109/INES59282.2023.10297629.
- [62] A. Kapishnikov, S. Venugopalan, B. Avci, B. Wedin, M. Terry, and T. Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5050–5058, June 2021.
- [63] B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In D. Lee, M. Sugiyama,

- U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf.
- [64] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.
- [65] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un)reliability of saliency methods, 2017. URL <https://arxiv.org/abs/1711.00867>.
- [66] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [67] P. Kolpaczki, G. Haselbeck, and E. Hüllermeier. How much can stratification improve the approximation of shapley values? In L. Longo, S. Lapuschkin, and C. Seifert, editors, *Explainable Artificial Intelligence*, pages 489–512, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-63797-1.
- [68] R. Konig, U. Johansson, and L. Niklasson. G-rex: A versatile framework for evolutionary data mining. In *2008 IEEE International Conference on Data Mining Workshops*, pages 971–974, 2008. doi: 10.1109/ICDMW.2008.117.
- [69] S. Krishna, T. Han, A. Gu, S. Wu, S. Jabbari, and H. Lakkaraju. The disagreement problem in explainable machine learning: A practitioner’s perspective. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. doi: 10.21203/rs.3.rs-2963888/v1.
- [70] G. Laberge, U. Aïvodji, S. Hara, F. Khomh, et al. Fool shap with stealthily biased sampling. *arXiv preprint arXiv:2205.15419*, 2022.
- [71] H. Lakkaraju, N. Arsov, and O. Bastani. Robust and stable black box explanations. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5628–5638. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/lakkaraju20a.html>.
- [72] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), 2019. doi: 10.1038/s41467-019-08987-4.
- [73] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. doi: 10.1109/TPAMI.2018.2858826.
- [74] S. Lipovetsky and M. Conklin. Analysis of regression in game theory approach. *Applied stochastic models in business and industry*, 17(4):319–330, 2001.

- [75] Z. C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16:31–57, 2018.
- [76] 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>.
- [77] S. M. Lundberg and S. I. Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-December: 4766–4775, 2017. <https://dl.acm.org/doi/10.5555/3295222.3295230>.
- [78] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, ..., and S.-I. Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020. <https://doi.org/10.1038/s42256-019-0138-9>.
- [79] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.
- [80] L. McInnes, J. Healy, N. Saul, and L. Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861.
- [81] S. Mishra, S. Dutta, J. Long, and D. Magazzeni. A survey on the robustness of feature importance and counterfactual explanations, 2023. URL <https://arxiv.org/abs/2111.00358>.
- [82] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K. R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. doi: 10.1016/j.patcog.2016.11.008.
- [83] 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>. URL <https://www.sciencedirect.com/science/article/pii/S1051200417302385>.
- [84] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller. *Layer-Wise Relevance Propagation: An Overview*, pages 193–209. 09 2019. ISBN 978-3-030-28953-9. doi: 10.1007/978-3-030-28954-6_10.
- [85] A. Mothilal, R. K. and Sharma and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020. <https://doi.org/10.1145/3351095.3372850>.

- [86] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlöterer, M. van Keulen, and C. Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, 55(13s), July 2023. ISSN 0360-0300. doi: 10.1145/3583558. URL <https://doi.org/10.1145/3583558>.
- [87] A. P. Nguyen and M. R. Martínez. On quantitative aspects of model interpretability. 2020. <https://arxiv.org/abs/2007.07584>.
- [88] L. H. B. Olsen and M. Jullum. Improving the weighting strategy in kernelshap, 2025. URL <https://arxiv.org/abs/2410.04883>.
- [89] A. C. Oztireli, M. Ancona, E. Ceolini, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. 2019. doi: 10.17863/CAM.47233. URL <https://www.repository.cam.ac.uk/handle/1810/300162>.
- [90] C. Panigutti, R. Hamon, I. Hupont, D. Fernandex Llorca, D. Fano Yela, ..., and E. Gomez. The role of explainable ai in the context of the ai act. *FAccT '23: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1139–1150, 2023. <https://doi.org/10.1145/3593013.3594069>.
- [91] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2, Part 2):3336–3341, 2009. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2008.01.039>. URL <https://www.sciencedirect.com/science/article/pii/S095741740800081X>.
- [92] M. Pawelczyk, K. Broelemann, and G. Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020, WWW '20*, page 3126–3132, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380087.
- [93] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720. URL <https://doi.org/10.1080/14786440109462720>.
- [94] C. Pirie, N. Wiratunga, A. Wijekoon, and C. F. Moreno-Garcia. AGREE: A feature attribution aggregation framework to address explainer disagreements with alignment metrics. In *ICCBR XCBR'23: Workshop on Case-Based Reasoning for the Explanation of Intelligent Systems at ICCBR2023*, volume 3438, pages 184–199, Aberdeen, Scotland, July 2023. CEUR Workshop Proceedings. URL https://ceur-ws.org/Vol-3438/paper_14.pdf.
- [95] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach. Face: Feasible and actionable counterfactual explanations. *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350, 2020. <https://doi.org/10.1145/3375627.3375850>.

- [96] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, pages 97—101, 2016. doi: 10.18653/v1/n16-3020.
- [97] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. *32nd AAAI Conference on Artificial Intelligence*, AAAI 2018:1527–1535, 2018. <https://doi.org/10.1609/aaai.v32i1.11491>.
- [98] L. Rieger and L. K. Hansen. A simple defense against adversarial attacks on heatmap explanations, 2020. URL <https://arxiv.org/abs/2007.06381>.
- [99] A. Rosenfeld. Better metrics for evaluating explainable artificial intelligence. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 1:45–50, 2021. <https://dl.acm.org/doi/abs/10.5555/3463952.3463962>.
- [100] S. Roy, G. Laberge, B. Roy, F. Khomh, A. Nikanjam, and S. Mondal. Why don't xai techniques agree? characterizing the disagreements between post-hoc explanations of defect predictions. In *2022 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 444–448, 2022. doi: 10.1109/ICSME55016.2022.00056.
- [101] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 2019. <https://doi.org/10.1038/s42256-019-0048-x>.
- [102] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [103] M. Sahakyan, Z. Aung, and T. Rahwan. Explainable artificial intelligence for tabular data: A survey. *IEEE Access*, 9:135392—135422, 2021. doi: 10.1109/ACCESS.2021.3116481.
- [104] W. Samek, A. Binder, S. Lapuschkin, and K.-R. Müller. Understanding and comparing deep neural networks for age and gender classification. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1629–1638, 2017. doi: 10.1109/ICCVW.2017.191.
- [105] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021. doi: 10.1109/JPROC.2021.3060483.
- [106] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247—278, 2021. doi: 10.1109/JPROC.2021.3060483.

- [107] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schutt, K.-R. Muller, and G. Montavon. Higher-Order Explanations of Graph Neural Networks via Relevant Walks . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(11):7581–7596, Nov. 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3115452. URL <https://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3115452>.
- [108] G. Schwalbe and B. Finzel. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 38(5):3043–3101, 2024.
- [109] A. D. Selbst and J. Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7-4:233–242, 2018. <https://doi.org/10.1093/idpl/ix022>.
- [110] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. doi: 10.1109/ICCV.2017.74.
- [111] S. M. Shankaranarayana and D. Runje. Alime: Autoencoder based approach for local interpretability, 2019. URL <https://arxiv.org/abs/1909.02437>.
- [112] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. *34th International Conference on Machine Learning, ICML 2017*, 7:4844–4866, 2017. <https://dl.acm.org/doi/10.5555/3305890.3306006>.
- [113] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. Not just a black box: Learning important features through propagating activation differences, 2017. URL <https://arxiv.org/abs/1605.01713>.
- [114] A. Shrikumar, J. Su, and A. Kundaje. Computationally efficient measures of internal neuron importance, 07 2018.
- [115] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.
- [116] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. *AIES 2020 - Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, page 180–186, 2020. doi: 10.1145/3375627.3375830.
- [117] K. Sokol and P. Flach. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT* '20*, page 56–67, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369367. doi: 10.1145/3351095.3372870. URL <https://doi.org/10.1145/3351095.3372870>.

- [118] M. Sonnewald and R. Lguensat. Revealing the impact of global heating on north atlantic circulation using transparent machine learning. *Journal of Advances in Modeling Earth Systems*, 13, 2021. doi: 10.1029/2021MS002496.
- [119] F. Sovrano, S. Sapienza, M. Palmirani, and F. Vitali. Metrics, explainability and the european ai act proposal. *J*, 5(1):126–138, 2022. ISSN 2571-8800. doi: 10.3390/j5010010. URL <https://www.mdpi.com/2571-8800/5/1/10>.
- [120] F. Sovrano, G. Vilone, M. Lognoul, and L. Longo. Legal xai: a systematic review and interdisciplinary mapping of xai and eu law, toawards a research agenda for legally responsible ai. 2025. doi: <http://dx.doi.org/10.2139/ssrn.5371124>.
- [121] T. Speith. How to evaluate explainability? - a case for three criteria. In *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, pages 92–97, 2022. doi: 10.1109/REW56159.2022.00024.
- [122] T. Speith. A review of taxonomies of explainable artificial intelligence (xai) methods. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT '22*, page 2239–2250, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3534639. URL <https://doi.org/10.1145/3531146.3534639>.
- [123] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net, 2015. URL <https://arxiv.org/abs/1412.6806>.
- [124] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41:647–665, 2014.
- [125] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *34th International Conference on Machine Learning, ICML 2017*, 7: 5109–5118, 2017. <https://dl.acm.org/doi/10.5555/3305890.3306024>.
- [126] S. Tan, R. Caruana, G. Hooker, and Y. Lou. Detecting bias in black-box models using transparent model distillation. 2018. <https://arxiv.org/pdf/1710.06169.pdf>.
- [127] Z. Tan, Y. Tian, and J. Li. Glime: General, stable and local lime explanation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 36250–36277. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/71ed042903ed67c7f6355e5dd0539eec-Paper-Conference.pdf.
- [128] The White House. Blueprint for an AI Bill of Rights: Making Automated Systems Work for the American People, 2022. <https://www.whitehouse.gov/ostp/ai-bill-of-rights/>.

- [129] S. Upadhyay, S. Joshi, and H. Lakkaraju. Towards robust and reliable algorithmic recourse, 2021. URL <https://arxiv.org/abs/2102.13620>.
- [130] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL <http://jmlr.org/papers/v9/vandermaaten08a.html>.
- [131] I. Vascotto, V. Blasone, A. Rodriguez, A. Bonaita, and L. Bortolussi. Assessing reliability of explanations in unbalanced datasets: a use-case on the occurrence of frost events. In P. Biecek, S. Nowaczyk, G. Kutyniok, and L. Longo, editors, *xAI-2025 Late-breaking Work, Demos and Doctoral Consortium Joint Proceedings*, volume 4017 of *CEUR Workshop Proceedings*, pages 73–80, 2025. URL https://ceur-ws.org/Vol-4017/paper_10.pdf.
- [132] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi. Aggregating explanations to increase trustworthiness: current directions and challenges. Abstract presented at Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy, 2025.
- [133] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi. Beyond single-model XAI: aggregating multi-model explanations for enhanced trustworthiness. In A. Følstad, D. Apostolou, S. Taylor, A. Palumbo, E. Tsalapati, G. Stamatiellos, and R. Catelli, editors, *Proceedings of TRUST-AI 2025 - the European Workshop on Trustworthy AI*, volume 4132 of *CEUR Workshop Proceedings*, pages 276–286, 2025. URL <https://ceur-ws.org/Vol-4132/short38.pdf>.
- [134] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi. When can you trust your explanations? A robustness analysis on feature importances. In R. Guidotti, U. Schmid, and L. Longo, editors, *Explainable Artificial Intelligence*, pages 225–249, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-032-08327-2. doi: https://doi.org/10.1007/978-3-032-08327-2_11.
- [135] E. Veledar, L. Zhou, O. Veledar, H. Gardener, C. M. Gutierrez, J. G. Romano, and T. Rundek. Synthesizing explainability across multiple ml models for structured data. *Algorithms*, 18(6), 2025. ISSN 1999-4893. doi: 10.3390/a18060368. URL <https://www.mdpi.com/1999-4893/18/6/368>.
- [136] 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>. URL <https://www.sciencedirect.com/science/article/pii/S1566253521001093>.
- [137] G. Vilone and L. Longo. Classification of explainable artificial intelligence methods through their output formats. *Machine Learning and Knowledge Extraction*, 3(3):615–661, 2021. ISSN 2504-4990. doi: 10.3390/make3030032. URL <https://www.mdpi.com/2504-4990/3/3/32>.
- [138] G. Visani, E. Bagli, and F. Chesani. Optilime: Optimized lime explanations for diagnostic computer algorithms. *ArXiv*, abs/2006.05714, 2020. URL <https://api.semanticscholar.org/CorpusID:219559031>.

- [139] A. White and A. D’Avila Garcez. Measurable counterfactual local explanations for any classifier. *Frontiers in Artificial Intelligence and Applications*, 325: 2529—2535, 2020. <https://doi.org/10.3233/FAIA200387>.
- [140] R. Yang, B. Wang, and M. Bilgic. Idgi: A framework to eliminate explanation noise from integrated gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23725–23734, June 2023.
- [141] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in neural information processing systems*, 32, 2019.
- [142] N. M. Zafar, M. R. and Khan. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint*, 2019. <https://arxiv.org/abs/1906.10263>.
- [143] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. doi: 10.1109/CVPR.2016.319.
- [144] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5), 2021. ISSN 2079-9292. doi: 10.3390/electronics10050593. URL <https://www.mdpi.com/2079-9292/10/5/593>.
- [145] Y. Zhou and G. Hooker. Interpreting models via single tree approximation. *arXiv preprint*, 2016. <https://arxiv.org/abs/1610.09036>.
- [146] L. Zou, H. L. Goh, C. J. Y. Liew, J. L. Quah, G. T. Gu, J. J. Chew, M. P. Kumar, C. G. L. Ang, and A. W. A. Ta. Ensemble image explainable ai (xai) algorithm for severe community-acquired pneumonia and covid-19 respiratory infections. *IEEE Transactions on Artificial Intelligence*, 4(2):242–254, 2023. doi: 10.1109/TAI.2022.3153754.