Politecnico di Torino

Department of Control and Computer Engineering
Master of Science in Computer Engineering

Master Thesis

Algorithms for interpretable machine learning

Author: Eliana Pastor
Supervisor: Prof. Elena Baralis

October 2017

Abstract

Machine learning classifiers are increasingly applied in every aspect of our society; medical diagnosis, loan granting, insurance and marketing are only examples. The point is that in high risk tasks accuracy cannot be considered anymore as the only important metric. Since the application of classification models can potentially negatively affect people lives, users want to understand how the model works or at least to understand why a particular prediction was made. Only if users are able to comprehend the reasons behind a prediction, they can choose if trust it or not, based on their prior domain knowledge. The problem is that most of the machine learning models adopted are increasingly accurate but at the same time they are seen as black boxes. For this reason, users have often to chose between more accurate but less interpretable models or more interpretable but less performing ones. This is not the best solution and so many algorithms have been developed for improving the comprehensibility of non-interpretable models. In this thesis, some of the already existing approaches are illustrated, focusing on their advantages and limitations. These methods can be categorized into two main groups: model-dependent approaches and model-agnostic ones. While the former ones are developed for making more understandable only particular models, the latter are instead applicable to any classifier.

We propose a new model-agnostic explanation method for explaining the individual prediction of any classifier. Inspiring for our work is the solution proposed by the authors Kononenko et al. and our method overcome the exponential time complexity of this solution. The explanation of a prediction is provided in form of contributions of features' values that shows what are the values that influence the most the prediction. The idea is to compute the contribution of features' values deleting one or more attributes' values at the time and seeing how the prediction changes: the greater is the change, the more this or those attributes are important for the prediction. Our idea, for overcoming the problem of the computation of the features' values power set is to use a local interpretable model. The local model is learned in the locality of the prediction that we want to explain and it has to mimic the behavior of the complex model. The local model, being interpretable, is able to highlight what are the subset of features' values that are relevant for a particular prediction made by a particular model. In this way, only the important subsets are considered for computing the contribution of features' values, instead of all the power set. In addiction, considering a local zone instead of the entire model makes the method extensible also to Big Data applications, where a global model is difficult to obtain.

To Matteo and my family.

Acknowledgements

I would like to thank my advisor Prof. Elena Baralis for her guidance and support.

A special thank goes to my parents and my sister Viviana for their constant belief and support. I will be always grateful to them.

I would like to thank my grandmother, who has always been there for me and made me who I am today.

A special acknowledgement to Matteo for his continued support and encouragement.

Finally, I would like to thank all my friends for their support.

Eliana Pastor, Torino, October 2017

Contents

Chapter 1

Introduction

Machine learning algorithms are increasingly applied in every aspect of our society as in medicine [21], finance and insurance. In these high risk applications, understanding how the model works or at least understanding why a particular decision is made is becoming increasingly important since this could potentially affects people's lives [21, 64]. Users cannot act based only on models outputs but they take an action according to a prediction made by a model only if they can comprehend it and if they trust it. Trust is a term strictly linked to the concept of interpretability: only if users are able to comprehend the internal working of the model or the reasons behind a particular prediction they can compare it with their prior knowledge of the problem and, if it is consistent with it, they can act on that basis [88]. In real world cases, since the models' application have a great impact, trusting a prediction is often considered more relevant that the prediction accuracy of the model itself. Often less accurate but more interpretable models are preferred to the performing yet hardly understandable ones [21]. Trust is not the only concepts strictly linked to interpretability. Understanding a model, but also single predictions, allows users to have insights on how the model works and so also on its problems. Only if the relationships between inputs and model's predictions are highlighted, domain experts can inspected them and potentially find wrong associations. The point is that the recognition of the model's issues could potentially allow to convert an unreliable model into a trustworthy one [88]. If the model's problems are known, domain experts can investigate and potentially solve them. Improving interpretability of machine learning models is therefore particularly important for the models' debugging. In addiction, the demand for more interpretability derives also from the demand for fairness [64]. The deployment of machine learning models in critical areas can potentially have a negative impact on people's lives. Ethical concerns on the access and on the use of sensitive information arises [41]. Models' decisions may reflect the discrimination and unfairness that exist in our society, since they depend on data that has been collected from it [41]. Only if experts are able to understand why a particular prediction is made, they can investigate if it is based on discriminatory or sensitive aspects and identify appropriate solutions.

The aim of this thesis is to address the problem of improving interpretability of classification algorithms and to propose a novel explanation method for explaining individual predictions of any classification models.

The study, after having illustrated the importance of interpretability, firstly explores what are the existing methods for improving interpretability of classification algorithms, highlighting advantages and limitations. In particular, these approaches will be described focusing on the distinction between model-dependent and model-agnostic solutions. We then propose a new method for explaining single predictions of any classifier. Our solution is model-agnostic [88], applicable for explaining the decision of any classifier without making any assumptions about the characteristics of the model whose prediction we want to explain. Our approach can be seen as an extension of the solution proposed by the author Kononenko et al.: the method we propose is able to overcome its exponential time complexity [108]. This is done learning a local model on the locality of the instance that we want to explain that highlights what are the relevant attributes for that particular prediction. The prediction's explanation indicates, for each attribute's value, what is its contribution to the prediction, with respect to a particular target class.

The thesis is organized as follow. The Chapter 2 is focused on the illustration of the importance of interpretability also through real-case examples. The problem of its definition and of how to measure it is then addressed. The Chapter 3 is divided in two parts. In the first, the so called interpretable classification models are briefly described, focusing on what forms of interpretability they are able to provide. In the second section instead, model-dependent solution for improving the interpretability of hardly interpretable models are described. These approaches are applicable only to the particular model for which they were implemented. In the Chapter 4 instead, model-agnostic solutions are presented. In particular, the first part of the section focuses on model-agnostic methods for explaining how an entire model works while the second on methods for explaining single predictions. In the Chapter 5, our novel approach for explaining the predictions of any classifier in a model-agnostic way is formally described. In Chapter 6, experimental designs and results are reported, based on artificial and real data sets. Finally, Chapter 7 draws conclusion and illustrates future works.

Chapter 2

Interpretability

The aim of this chapter is to provide a complete description of the interpretability concept in machine learning. The first section is focused on the importance that interpretability is increasingly assuming and this also serves as an assertion of why interpretability is the subject of this thesis. In the second section, the term interpretability is defined, focusing in particular on the problem of proposing an accepted definition and on the illustration of concepts that are strictly related to this term. Finally, the third section illustrates what are the possible ways for measuring the interpretability and advantages and limitations of the possible estimation approaches are highlighted.

2.1 The importance of interpretability

More and more data are collected and made available to be mined. These heterogeneous and high-dimensional data led to the development of machine learning algorithms always more accurate. Models are built to solve increasingly complex problems and they show excellent behaviours in terms of performance and accuracy. For these reasons, machine learning algorithms are nowadays applied in every field, from medicine, insurance, finance to the law one. They are applied in high risk task and particularly for support humans in decision-making. In these application, it is important not only to obtain accurate results: users want also to understand why the model has made a particular decision [21, 88]. The more a decision could affect significantly people's lives, the more important it is to comprehend what are the factors that lead to that particular decision. This is particularly true for medical diagnosis: doctors cannot simply act based on models' predictions, they have firstly to trust them. For trusting them, doctors have firstly to understand why these decisions were made and compare these reasons with their prior domain knowledge [88]. In high risk applications, interpretability is considered more important than the accuracy metric [21]. The point is that, as Ribeiro et al. note, “if the users do not trust a model or a prediction, they will not use it” [88]. Interpretability so is strictly linked to the concept of trust: only if users are able to interpret a model or a decision, they can analyze how the model works or at least what are the important factors for the decision and so then see if this is consistent with their prior domain knowledge or not. The relevant aspects highlighted by the model not only allow a more reasoned choice if trusting or not a decision, but also to discover new knowledge and debug the model, if wrong associations have been highlighted [64]. Discovery new information is one of the goal of the KDD but the discovered knowledge must be comprehensible in order to be used [39]. When the patterns discovered by the model are not consistent with experts’ domain knowledge, they can inspect them and may find that the model has some issues. In this case, the understanding of what are the problems allows experts to try to fix them and to support debugging phase [88]. Comprehending a model or a prediction allows also to understand if decisions are based on potential discriminatory aspects [37, 41]. Interpretability is strictly connected to fairness and ethics. Since machine learning algorithms are nowadays applied in every aspect of society, unfair models could greatly affects people’s lives, negatively influencing their equal participation to the community [60, 64].

In the next two sub-sections, two real examples of the importance of interpretability are presented. The first is a real case study for predicting the pneumonia risk; it shows how understanding a model is relevant particularly because linked with the trust and the debugging concepts [21]. The second illustrates that interpretability is demanded not only by machine learning experts and users but also by institutions [76].

2.1.1 Case study: pneumonia risk

This case study can show the importance of interpretability and of criteria connected to it. In particular, it shows an example of why, in high risk applications, interpretable but less accurate models are preferred to more accurate but opaque ones.

Cost-Effective HealthCare in mid 90’s funded a project in order to predict pneumonia risk, using machine learning. The goal was to estimate the POD, probability of death of patients with pneumonia. High risk patients will be treated in hospital while the others as outpatients. In this way the ones with an higher POD could be treated with all the attention they need but in this way it is also possible for the hospital to reduce health care cost.

The studies [25, 26] show that the most accurate results were achieved using a multitask neural network, with an Area Under the Curve (AUC) of 0.86. What is important to notice is that the medical experts decided to discard the neural network model and to use the logistic regression one, even if it had a worse accuracy. Logistic regression model showed in fact a strange correlation between asthma and POD. The same correlation was also found in [5] using a rule based approach. The unusual rule was:

$$\text{``hasAsthma}(x) \rightarrow \text{LowerRisk}(x)\text{''}$$

which is that if a patient x has asthma, he has lower risk of dying. The medical staff, following this rule, should not admit to the hospital this patient and should treat him as an outpatient. This rule seems very counterintuitive because patients affected by asthma are usually considered more weaker and treated with more care. As the authors Caruana et al. reveal, this more carefulness is the cause of this correlation and the trained model simply had learned and identified it. Patients with asthma usually in fact were directly admitted to the Intensive Care Unit and receive an aggressive care [21].

This example can so show how applying directly a decision support system in an high risk real application is extremely dangerous: if doctors had followed the model, the patients with asthma would be not hospitalized and the consequences of this choice for them would be fatal. It was so preferred to use in the practise less accurate but intelligible models. In fact even if neural networks show better results in term of accuracy they are black boxes, and they do not offer any satisfactory explanation of their behaviour. If the simpler models had learned that having asthma led to a less risk of dying in case of pneumonia, it is very likely that even neural networks would have learned it. Being more accurate, it is possible that the neural nets had learned also other strange correlations that could put in danger the life of patients and these, being hidden, could not be fixet [21].

Interpretability is deeply linked to *trust*, fundamental in the case of taking some actions based on a prediction [88]. If the final users do not trust a model or a prediction, they will not adopt it as a support for decision making.

At the same time understanding the reasons why a model can not be trusted gives the possibility to fix it. In the example shown, an unsound correlation between asthma and probability is found when using intelligible models. Experts in the domain model, in this case medical ones, can so reason on the origin of this association. Interpretable models or predictions can provide a sort of window into the data and researchers and v experts have the possibility to investigate and analyze them [21]. In some cases, this reasoning can led to the discovery of new pattern and knowledge, in others, as in this case, to the detection of spurious correlations. A spurious correlation, citing the definition proposed by Vogt, is “a situation in which measures of two or more variables are statistically related [...] but are not in fact causally linked-usually because the statistical relation is caused by a third variable” [106]. Once this not correct behaviour of the model is detected is possible to repair the model. The demand for interpretability also arises for *debugging* purposes. The correction of the model is possible only when problems are enlightened. The critical rules can be removed in a rule-based model, the weights of features belonging to a spurious correlation can be correct in a logistic regression model. The correction itself is obviously possible also in the case of non interpretable models: a neural network can be re-trained without the features with problems or, in this pneumonia case, modified in order to invert the priority for the patients affected by asthma [21]. The problem for the asthma can be solved but other problems that the neural network can have, due to its lack of interpretability, are not known and so they cannot be fixed.

Another aspect that Lipton emphasizes as a desiderata of interpretability is *causality* [64]. Through the use of supervised learning models researchers have the desire to discover new patterns, properties and hopefully to generate new hypothesis about the world. But correlation does not imply causation [79]. The association can exist for other causes not observed but that are actually responsible of it. An example is given in the case of study presented. The lower risk of dying of pneumonia in a patient affected by asthma is due to the special treatment that he received, not to its illness itself. The causality of lower risk is imputable to the direct hospitalization to the Intensive Care Unit, not to the asthma. Only using an intelligible model it is possible to discover associations, to reason about them and investigate on the origin of them [21].

2.1.2 “Right to explanation”

The European Union Parliament in April 2016 has approved the General Data Protection Regulation, GDPR, a set of regulations for ensuring personal data protection. It concerns the collection, storage and use of personal data, defined in Article 4 as "any information relating to an identifies or identifiable natural person" [76]. It will replace, in April 2018, date of its effectiveness, the Data Protection Directive, DPD, of the 1995. The GDPR has been described as a "Copernican revolution" for its attempt to "increase protection of fundamental rights", for its strong willingness to be effective and efficient form a legal point of view [57]. While the Data Protection Directive is a directive and so it has to "be binding, as to the result to be achieved, upon each Member State to which it is addressed, but shall leave to the national authorities the choice of form and methods", the General Data Protection Regulation is regulation "shall be binding in its entirety and directly applicable in all Member States" [77]. The other important improvement of this direction is that it has effect not only in the European states but it is applicable to any companies that dispose of EU residents' personal data [41].

The GDPR will have a great impact on the use of machine learning algorithms. This regulation determines how data should be managed, the "right to be informed" of the persons whose data are processed [110], the "right to explanation" [41] and focuses on ethical decision-making.

Articles 13-15 concern the right of the data subjects to be informed of the data collected, "the period for which the personal data will be stored, or if that is not possible, the criteria used to determine that period", "where the processing is based on point (f) of Article 6(1), the legitimate interests pursued by the controller or by a third party" and the purposes [76]. While for some researchers these articles are intended more as a "right to be informed" [110], for Goodman and Flaxman the GDPR will legally mandate a "*right to explanation*" [41] .

Article 15 paragraph 1, reported in figure 2.1.2, not only remarks that a data subject has the right to be informed of different aspects related to its data but also in case of "automated decision-making, including profiling," to receive "meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing". This last statement arises a question: what is it intended for information about the logic involved? For Wachter et al. from a legal point of view it can be seen as an *ex ante* explanation or as an *ex post* [110]. An ex ante explanation is an explanation that is provided before the actual use of personal data for the automated decision processing and refers to the system functionality, not to the decisions made. The ex post explanation can instead refer also to how decisions are made.

In both cases this is an hard task to be achieved: most of the machine learning

> Article 15
> Right of access by the data subject
>
> 1. The data subject shall have the right to obtain from the controller confirmation as to whether or not personal data concerning him or her are being processed, and, where that is the case, access to the personal data and the following information:
>
>   (a) the purposes of the processing;
>
>   (b) the categories of personal data concerned;
>
>   (c) the recipients or categories of recipient to whom the personal data have been or will be disclosed, in particular recipients in third countries or international organisations
>
>   (d) where possible, the envisaged period for which the personal data will be stored, or, if not possible, the criteria used to determine that period;
>
>   (e) the existence of the right to request from the controller rectification or erasure of personal data or restriction of processing of personal data concerning the data subject or to object to such processing;
>
>   (f) the right to lodge a complaint with a supervisory authority;
>
>   (g) where the personal data are not collected from the data subject, any available information as to their source;
>
>   (h) 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.

Figure 2.1: Paragraph 1 of Article 15. Excerpt from the General Data Protection Regulation [76].

algorithms lack in terms of interpretability. They are not able to provide neither information on how they work neither how the decision is taken. It is for this reason necessary for researches to work on the direction of providing a rigorous and unified definition of interpretability and new algorithms for interpretable machine learning. On the other hand, as Wachter et al. suggest, there are some improvement on the GDPR that can be made [110]. In particular they suggest to clarify Article 15(1)h and what “existence of”, “meaningful information”, “logic involved”, “significance”, and “envisaged consequences” mean. This is important to be clarified from a machine learning point of view: proving an explanation on how the system works is different, and often more hard [88], than providing a clarification on how a single decision is taken.

The GDPR raises another concern regarding discriminating machine learning models. The right to non-discrimination is one of the founding principles on which the European Union was built [41], clarified in Article 21 of the Charter of Fundamental Rights of the European Union. This principle is underlined also in the GDPR. Paragraph 71 of the recital, where the intentions of the regulation are clarified, reported in figure 2.1.2, obliges data controller to implement measures that "prevents, inter alia, discriminatory effects" on the basis of sensitive data [76].

> In order to ensure fair and transparent processing in respect of the data subject, taking into account the specific circumstances and context in which the personal data are processed, the controller should use appropriate mathematical or statistical procedures for the profiling, implement technical and organisational measures appropriate to ensure, in particular, that factors which result in inaccuracies in personal data are corrected and the risk of errors is minimized, secure personal data in a manner that takes account of the potential risks involved for the interests and rights of the data subject and that *prevents, inter alia, discriminatory effects on natural persons on the basis of racial or ethnic origin, political opinion, religion or beliefs, trade union membership, genetic or health status or sexual orientation, or that result in measures having such an effect.* Automated decision-making and profiling based on special categories of personal data should be allowed only under specific conditions.

Figure 2.2: Excerpt from the General Data Protection Regulation, Recital 71 [italics added] [76]

Profiling and decision-support algorithms are applied in every field and so also in public goods as public health, fair employment, safety and finance. It is responsibility of government to supervise and legislate in order to guarantee the right of non-discrimination.

Different issues arise. First, machine learning profiling is intrinsically discriminatory [41]. The aim of profiling is to categorize, to identify some common behaviours of a group that are different from behaviours of other groups; different decisions are then taken based on the characteristic of the group. The problem is that machine learning algorithms are based on data collected from our society. As Barocas and

Selbst argue, the data are intrinsically discriminatory because data reflect the society [10]. The society itself is unfair, unequal and consequently also data and machine learning based on them [78]. The demands for *fairness* lead to the demand of more interpretable models [64]. Only having more insights on how a certain prediction is made can help to understand what kind of unfairness is present on the model itself.

The GDPR regulates on profiling, in particular the processing of sensitive data, i.e. personal data revealing racial origins, religious beliefs, sexual orientation and so on. The authors Goofman and Flaxman suggest that there are two possible interpretations, a minimal and a maximal one [41].

In the minimal interpretation there are restrictions and actions to be made only if the algorithms use explicitly sensitive data. For example, an action for removing discrimination could be to delete those variable that are considered sensitive. The problem is that this solution is absolutely non effective. The sensitive variables can be correlated with other variables. For example, Calders and Verwer showed that the postal code may be highly correlated with ethnicity [18]. Deleting the ethnicity variable is useless in this case because postal code remains an excellent predictor for this attribute.

In the maximal interpretation also the correlate variables should not be considered. However, datasets are increasingly larger and complex and the detection of correlation is a difficult task and in some cases impractical [41]. On the other hand, with the elimination of variables correlated to the sensible ones also useful information will be deleted. The resultant model would be presumably ineffective.

It is clear that many questions and problems concerning fairness of machine learning algorithms are still to be resolved. Only with comprehensible explanations of how automated decisions are made it is possible to reason and work on the granting of the right to non-discrimination [41].

2.2 What is interpretability?

The demand for interpretability comes from the implications that the applications of algorithms have in every aspect of human life. In the previous section the reasons of the importance of interpretability in machine learning have been enlightened and some real examples have been presented. At this point it is thus important to understand what it is really intended with the term interpretability.

Answering to the question of “what is interpretability?” is an hard task. Several authors have tried, but there is still not a definite answer [37, 64, 14]. The problem is that interpretability is an ambiguous term and it comprises many others.

To interpret means "to clarify or explain the meaning of; elucidate", "to construe the significance or intention of" [32], "make understandable", "to translate", "to have or show one's own understanding of the meaning of; construe"[33]. "To interpret" so has different shadows but they are all related to the concept of explaining in understandable terms. An accepted definition of interpretability in the machine learning community is instead more arduous to be find. Finale and Been assert that a possible one could be "the ability to explain or to present in understandable terms to a human" [37]. The term is so linked to the human beings, to their capacity of comprehend. But what does it mean "to explain"? There are still open questions on what an explanation is, how explanations are generated, when an explanation is better than another [66].

Before trying to deal with these open problems, it is important to illustrate firstly the several terms that are used to refer to "interpretability". The authors Bibal and Frènay proposed a structure where all the synonyms and related terms of interpretability are presented, reported in figure 2.3 [14]. On the first level, there are the interchangeable terms for interpretability, on the second level the terms that rely on it.

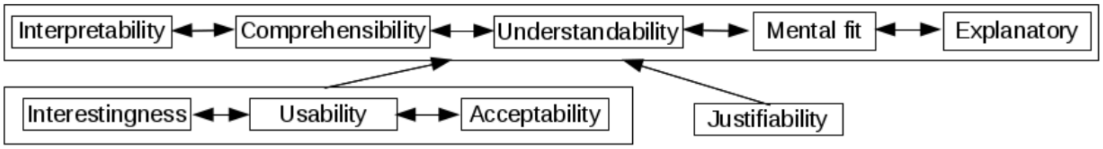


Figure 2.3: A ↔ B means that the two terms are equivalent, while A → B means that A is linked to B. [14]

Different authors use the term *comprehensibility* [7, 69, 52]. In the study of the author Askira-Gelman, emphasis is given to the importance of comprehensibility as a way to produce knowledge [7]. The discovery of new information can arise only when the pattern identified by machine learning algorithms are comprehensible. The other term used is *understandability* [74]. An interpretable model is a model that can be understood, in a reasonable amount of time [14]. It is an open question the quan-

tification of “the reasonable amount of time”. The point is that any model could be understood if the amount of time is infinite [14]. Other authors use the term *“mental fit”* [36, 112]. Feng and Michie proposed this term in contraposition to the term “data fit” [36]. While “data fit” can be seen as a synonym of accuracy or consistency [45], “mental fit” relates to the ability for a human to grasp the model. In particular for Feng and Michie classifiers should provide concepts that are meaningful to humans and “evaluable directly in mind” [36]. Characteristics of “mental fit” should be coverage, explainability and simplicity [103]. The importance of the explanation is also underline by Ustun and Rudin [101]. According to the authors, a model in order to be understandable should be *explanatory*, i.e. the relation between attributes and outcomes should be presented in an informative, meaningful and transparent way.
Another term is commonly used but not present in the summarizing figure proposed by Bibal and Frènay [14] is *intelligibility* [21, 67]. The word is strictly related to the human capability of understanding: a model is intelligible if it is interpretable by humans [21]. All these terms are used as synonyms of interpretability, even if each one has its own shadow.
As Lipton notes, “interpretability is not a monolithic concept”, it encloses different meanings. In addiction several terms are strictly connected with interpretability, even if each one represents a distinct idea. These concepts can be called *desiderata* [64, 37]: they are objectives of real-world applications. The intention is the optimization of these desiderata and authors argue that this is possible only through more interpretability.

In figure 2.4 these different terms linked with the concept of interpretability are displayed. Some of these concepts have already been introduced in the previous section for their importance in high risk applications and also because they are some of the reasons for the emerging demand of more interpretability. In the following part of this section a description of these desiderata is presented.

Trust: different researches argue that interpretability is one prerequisite for trust [88, 82]. Machine learning classifiers are increasingly applied but the choice of which classifier to utilize respect to one other strictly depends on the trust the humans place on it . “If the user do not trust a model or a prediction, they will not use it” [88]. This is particularly true for decision-making support in high risk applications as medical diagnosis, terrorism detection and finance. An example as been shown in the previous section: medical experts did not trust more accurate models as neural networks and they preferred more interpretable but less accurate one. The problem

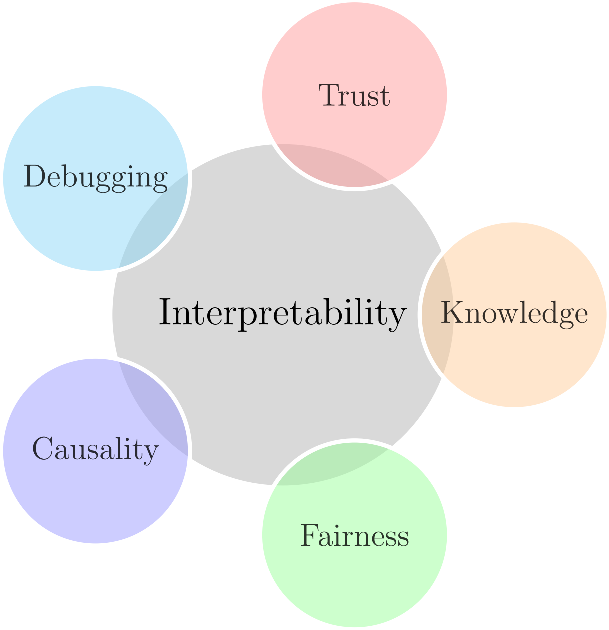


Figure 2.4: Concepts related to interpretability

in this case was that medical professionals were not able too understand how these complex models worked and how the decisions were made [21]. In order to trust models or a predictions they have to be interpretable.
In particular Ribeiro et al. propose two different definitions of trust: to trust a model and to trust a prediction [88]. *Trusting a model* means understanding how the complete model works and to apply it only if its behavior is reasonable in the domain of application. Trust a model is the confidence that the model will perform well in real scenarios [88]. Lipton highlights that this definition arise some questions [64]. When the behavior of the model is reasonable? It is simply confidence that a model will perform well? The problem is that accuracy is not a good indicator of a correct behavior [21]. Another definition of trusting a model is presented by Lipton [64]. A model can be trusted if it behaves like humans would do. So it is not strictly required to have models that outperform humans but they should reflect the behavior of humans. In particular the model can be considered trustworthy if it is accurate when humans are accurate and if it tends to make mistakes when humans make mistakes [64].

Trusting a prediction instead is the confidence in the individual prediction. In particular a user trusts a prediction if it is confident enough in it to take decisions based on it [88]. Trusting a single prediction is easier than trusting the entire model.

The reasons can be found on how human beings are [64]. Humans are capable of describing why they have taken particular decisions or actions, while the complete description of how their brain works is too complex to be given. This aspect is especially true when dealing with complex models. It is very difficult to catch on a single time the entire working mechanisms of a model. It is instead easier to understand why single predictions are made [64, 88]. The understanding of a prediction has not to be considered as less meaningful than the understanding of the the entire model. Trusting a prediction is what matter in decision-making. In addiction, explanations of single predictions can give insights in the comprehension of the entire model [88]. Through examples of how the model behaves we can decide if trusting the entire model or not [64]. Providing intelligible and clear explanations of single predictions is for these reasons important and new research has been done in this direction [90, 88].

Causality: through the application of machine learning algorithms, researches have the hope of finding new knowledge. Researches investigate patterns and correlations identified by models in order to generate hypotheses on the real world [64]. As already pointed in section 2.2, correlation does not mean causality. Correlation can be spurious and coming from unobserved variables that are actually responsible for the interrelation. On the other hand the ambition of finding true causality remains. Moreover, even in case of spurious correlation it is possible to actually find the cause of the relationship between the variables. As the example presented in section 2.1.1, the counterintuitive correlation between asthma and lower risk of dying of pneumonia enlightened by intelligible models let medical experts to investigate it [21]. The actual cause of this correlation was found in the better treatment that asthmatic patients usually received. Causality so it is strictly linked to interpretability: only through insights on model's working process it is possible to speculate on the patterns found and maybe find real causal relationships.

Knowledge: this term encloses different terms, even if they have different shadows : *scientific understanding* [37], *informativeness* [64], *interestingness* [14]. The human's goal is to acquire new knowledge. This is also the purpose of data mining: mining large amounts of data for extracting patterns and knowledge. The term "Knowledge discovery" was coined by Gregory Piatetsky-Shapiro and it has been defined as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [39]. A pattern becomes discovered knowledge if it has two characteristics: it is "interesting (according to a user-imposed interest measure)"

and “certain enough (again according to the users criteria)” [39]. The point is that the discovered patterns must be comprehensible. Only if the results achieved my machine learning algorithms are presented in an understandable way can be studied and investigated. If instead a model is a black boxes, as neural networks, support vector machines are, it is not possible to comprehend the actual working of the model neither why a single decision was taken. Seeing the importance of gaining new knowledge, research interest should be in the direction of implementing new interpretable and at the same time accurate algorithms but also finding ways to extract symbolic and comprehensible representations from black box models.
For some authors, as Lipton, supervised models should also provide information as support in decision-making applications and this is referred with the term *informativeness* [64]. The information represent a sort of argument of the taken decisions. The information can be a justification of the decision, enlightening the inner working of the model, or can be provided in form of examples. In particular in medical diagnosis, doctors use precedent studied cases to make decisions. If supervised models provide examples of why the decision is taken, that can be used by medical experts. They can confront the provided information with their precedent knowledge and decide if the prediction can be trusted or not [64].

Fairness: as already discussed in section 2.1.2, fairness and interpretability are connected: the demand for fairness lead to the demand for more interpretability [64]. The requirement of fairness in machine learning derives from the increasingly application of machine learning algorithms in public goods, such as public health, finance and job opportunities [41, 60, 64]. These algorithms have a great impact on human life. Their decisions control loans’ granting [117], hospital access [21], news provisioning [38]. For their influence in common good and in the society, these algorithms have been referred as social good algorithms [60]. The potential negative consequences of their use are increasingly studied [10, 31, 60]. The major concerns regard discrimination, social exclusion and privacy violation [60]. Private information can be accurately predicted computationally: from Facebook profiles and “Likes” and survey information it is possible to predict sensible data as sexual orientation, political ideologies, ethnic origin and drug consumption [55]. Resolving the discrimination and social exclusion presents in machine learning problems can be a very hard task. As already discussed in section 2.1.2, discrimination may be intrinsic in the model. The data are collected from the society and so data contain the inequality and discrimination actually present in the society [10]. Consequently, the machine learning models build on these data are unfair as the society is. Only through interpretable models is possible to have insights on results and investigate on possible discriminatory decisions [41, 60].

Debugging: if a model do not perform correctly, in order to fix it, it is decisive to understand why the model does not work. If the model is a black box, it is almost impossible to understand its working process and why it makes a particular prediction. As a consequence, it is also not possible to understand the causes of the incorrect behaviour and so to try to fix it. As examined in section 2.1.1, interpretable models instead give experts the possibility to comprehend why a decision was made and in case of incorrect behaviour to improve the model [21]. As Ribeiro et al. illustrate, explanations give insights on the model and this could be "helpful to convert an untrustworthy model into a trustworthy on" [88]. Another proof of this can be shown still through the pneumonia case study. The authors Caruana et al., applied the generalize addictive models with pairwise interaction (GA^{2}Ms) that are intelligible but at the same time high-accurate, on the pneumonia data set [21]. As discussed in section 2.1.1, the aim was to predict the risk of dying in patients affected by pneumonia. In case of high risk, the patient was hospitalized, while if at low risk the patient was treated as an outpatient. The GA^{2}Ms model exposed two other awkward correlations, shown in figure 2.5. Patients with chronic lung disease and history of chest pain had have lower risk of dying. These occurrences were studied and the explanation was found: as in patients affected by asthma, patients with lung and chest pain received usually a more attentive treatment and for this reason they were these correlations in the data set. Once this problematic was studied, it was possible also to fix the model, for example eliminating or editing these variables. This debugging was possible only because the model used was interpretable.

Now that the desiderata of interpretability research have been defined, there is still to make a distinction between local and global interpretability [88]. **Global interpretability** means to understand all the patterns present in the model. **Local interpretability** instead imply only the comprehension of a single prediction [37]. It is the understanding on "how the model behaves in the vicinity of the instance being predicted". Often the comprehension of the entire model is actually impossible due to the complexity of the model and of the domain of application itself. In addition in many cases the global understanding is unnecessary. In particular for what concerns

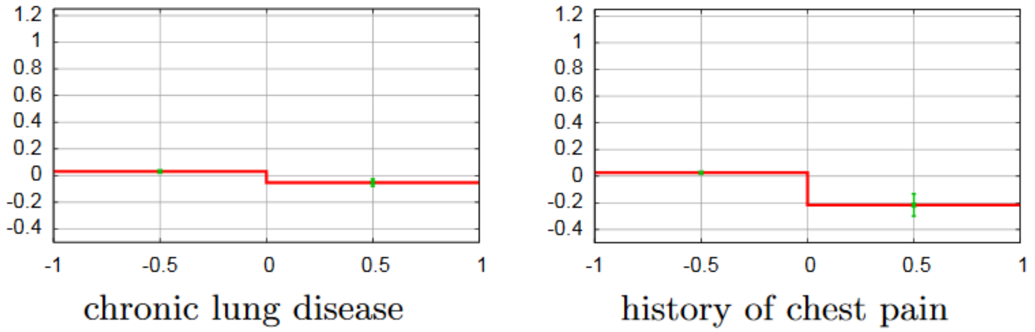


Figure 2.5: x-axis: feature, y-axis: probability of death (POD). The features "chronic lung disease" and "history of chest pain" are boolean and they assume value 1 in case of presence of the disease or -1, if not. In case of presence, the risk of death is lower, particularly for the "history of chest pain" term [21].

decision making support, it is sufficient to understand why the single decision was made for trusting it or not [88].

2.3 Measuring Interpretability

In the previous section, possible definitions of interpretability have been presented. At this point, it is important to discuss how interpretability can be measured. The ambiguous nature of interpretability is reflected also in its measurement. In fact, while for other machine learning metrics there are many precisely described and shared metrics, for interpretability there is not a unified way on how quantify it. As an example, for evaluating the performance of a supervised model different metrics can be used: classification accuracy, confusion metrics, sensitivity and F-measure are only some of them. The unsupervised learning evaluation is instead more complicated because there is not a comparison, we cannot test the obtained result with respect to "true one". Despite these problems, different metrics as been propose as the Sum of the Squared Error (SSE) [98], cohesion and separation and the silhouette coefficient [98].

Interpretability is difficult to quantify, also for its subjective nature. As expressed in section 2.2, something is interpretable if it is understandable for humans. The point is that humans have different background, prior experience, education and these differences reflect also on the different understanding of a model [49].

Before presenting possible interpretability measures, the target of the measure has to

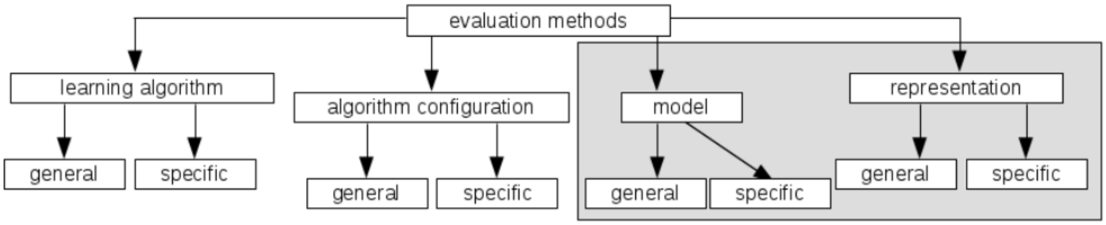


Figure 2.6: Adapted by [14] from [58]. In grey the authors have highlighted the targets where interpretability is evaluated.

be clarified. The authors Bibal and Frnay suggest that the interpretability can be measured with respect to models or to their representation, see figure 2.6 [14].

The intepretability of a whole model, for the authors, has to be measured quantitatively because models are analytical entities. This quantitive approach can be called *heuristic approach* [91].
The interpretability of a representation is instead measured through *user-based surveys*. Users evaluated model or predictions through their representation. An objection can be raised to the definition proposed by Bibal and Frénay of interpretability measure of the model. Some classifiers, as decision trees and rule-based classifiers, provide the representation of their internal working, through classification rules, trees or decision tables. In this case also a qualitative measure is possible. The model representations are shown to survey respondents and then, through an interview, it is possible to evaluate the qualitative global comprehensibility of the model [80]. Not all the classifiers are enough interpretable to provide a representation of their processing. In these cases the evaluation is not of the model itself but of its behaviour in particular cases. Lipton addresses this as post-hoc interpretability [64]. Post-hoc means “formulated after the fact”, and so, in the machine learning domain, after the training and the classification. With post-hoc interpretability the intention is not to clarify how a model works but to explain why a single prediction was made [64]. Now that these distinctions have been clarified, the problem of how to actually measure the interpretability of models in a quantitative way and of their representation will be described in the two following sections.

2.3.1 Heuristic approach

In the heuristic approach some heuristics are applied to measure model interpretability in a quantitive way [14, 91]. Heuristics are “set of rules for solving problems other than by algorithm" [33]. Heuristic strategies are applied for the good results obtained in past works: even if they are not the optimal method, they have enabled to achieve the expected result. For what concern measuring interpretability, different heuristic approaches can be used, they seem to work and at the moment they represent the only way. In fact, due to the problematic nature of the term interpretability, an algorithmic solution for its measure's computation still not exist.

The most used heuristic is the *size* of the model [7, 91]. For example, for decision trees the size is given by the number of nodes [34, 102], while for classification rules by the number of rules itself. The size is usually used also for heuristically measuring the complexity of a model [80]. It is frequently observed that interpretability is negatively correlated with complexity [4]. There is in fact a trade-off between interpretability and complexity: the more the model is complex, the less it is interpretable.

Some authors find justification of this heuristic in the limitations of human beings. The cognitive psychologist George A. Miller asserted that humans have great limitations on the amount of information their are able to process, imputable also to the limits of their short-memory [71]. The author theorized that humans are able to deal with seven, plus or minus two, abstract objects in their mind at the same time. For Cowan instead the mental storage capacity is of four entities [29]. Despite the disagreement of psychology experts, the common assumption is that human beings are able to take into account only a very limited amount of information at once. Humans cannot understand very complex problem in a single view.

This heuristic allows the comparison between models of the same type. It is also possible for instance to compare trees with propositional if-then rules and vice versa because they are logically equivalent and so it is possible to move from one form to the other [24, 49, 104]. But on most of the cases the comparison between models of different types is not possible. As an example, the number of nodes of a tree cannot be compared to the depth of a neural network, or to its neurons' number.

Moreover, other issues were highlighted by Freitas [40]. The author doubts the assumption that the smaller is the model, the more it is considered interpretable and so that, for this reason, with more propensity it will be used. In many cases, too simple models are rejected. Simple models are considered too elementary and so not able to catch the complexity of the problems they are suppose to solve. This is particularly true for medical applications, where trusting model's prediction is required. An example of this assertion was described by Elomaa [35]. Working on a data set of patients affected by Nephropathia epidemica, researches trained a tree classifiers and they obtained a 1-level tree. The only discriminate attribute was “fever”. Obviously this model is too simple and doctors cannot accept and use it. This case illustrates that not always at a small size corresponds a greater comprehensibility of the problem. For Elomaa “humans by nature are mentally opposed to too simplistic representations of complex relations” [35].

Sometimes complex models are preferable to simpler ones, because larger models can provide more information [4, 40]. The information can be relevant as a support for decision making or can also lead to discovery of new patterns.

In addition, it is important to underline the connection between time for analyzing the model and the comprehension of the model. Analyzing the model requires time and concentration, particularly if the model is complex and it incorporates many information. Interpretability is also correlated with efficiency: the user should have the possibility of grasping the model in a reasonable amount of time [14]. The problem is how to quantify “a reasonable amount” because it depends on the domain application: more complex problems obviously require more time than simpler ones. On the the other hand “it could be argued that any model could be understood given an infinite amount of time” [14]. Finally, the definition of “too complex” or “too large” depends on the users. It is a subjective matter: what is complex for one user can be simple for another one, due the different background and expertise [49, 80].

All these points illustrate many limitations of the size of the model as a measure for model’s interpretability. For this reasons other heuristics have been proposed.

Rüping suggests that a possible one could be splitting up the problem into sub-problems. In this way the obtained sub-problems would be less complex, still obtaining reasonable results [91].

Backhaus and Seiffert instead propose an approach that would resolve the problem of comparison between models of different type [8]. This approach is based on three criteria: “ability of the model to selected features from the input pattern, the ability to provide class-typical data points and information about the decision boundary directly encoded in model parameters” [8]. For example, multilayer perceptron (MLP) is graded 1 out of 3, because it satisfies only the third criteria: the decision boundaries are in fact directly encoded in the neurons [8]. The same grade is attributable to Support Vector Machines models (SVM) because the decision boundaries are stored in in support vectors and kernel [8]. In this way it is possible to compare different models. For these criteria, SVM and MPL have the same grade of interpretability. One limitation of this approach has been argued by Bibal and Frénay [14]. The comparison can be done only for models of different types but not of the same [14]. In addiction, this approach is only general because it is based only

on the characteristics of the machine learning algorithm applied. It does not take into account the resultant model obtained after the training phase.
Finally, the problem of all these existent heuristics is that these approaches do not consider semantic aspects [14, 40]. The criteria are only based on syntactical aspects of the model [40]. Semantics is instead decisive for the comprehension of the model.

2.3.2 User-based surveys

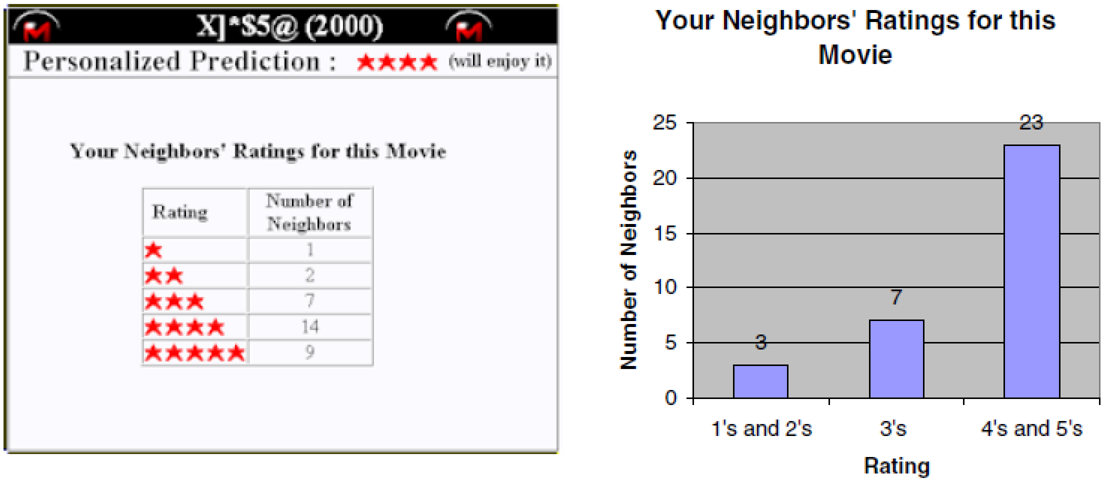
Seen the limitations of heuristic approaches, the preferable way for evaluating the interpretability is by using user-based surveys [14, 49, 80]. In this approach the evaluation is done based on model's or prediction's representation. One of the great difference and also virtue with respect to the heuristic approach is that using this approach it is possible to measure not only the interpretability of the entire model, but also of a single prediction. This is particularly relevant when the model is used as a support for decision-making. In these cases, understanding why a single decision is taken is extremely important. In addiction, for some applications it is also mandatory. In medical domain, doctors have to understand the model's prediction not only for the patients' interest. If they are sued for medical negligence doctors have to provide explanations of why they have made the decisions, even if provided by a model, as a their defense [89].
Interpretability is difficult to define and to measure also for its informal and subjective nature. For this reason, the user-based approach seems a more appropriate way for measuring it [14, 40]. This approach consists on presenting to users some representations and on asking questions about their interpretability. The interpretability so it is evaluated not for the model itself but for its representation.
Before describing how these surveys are designed, it has to be clarified what are the possible model representations.

Firstly, the representation can be of the model or of a single prediction. The *representation of a model* is possible only if the model is intelligible enough to provide information on its internal working process. Example of this kind of models are classification trees and rule-based systems [40, 80]. In these cases, the representation can be presented in a *graphical*, *textual* or *tabular form*. Classification decision trees are usually presented in a graphical structure. It consists on a set of nodes: the internal nodes specify the conditions to be tested while the leaves represent the value of the class label [84]. A single decision can be explained following the path from the root to the leaf. Rules are instead usually presented in a textual form that simply reports the extracted rules. Another possibility for representing these two models is through a decision table, a tabular representation that contains the complete set of conditional expressions [104]. All these three representations are logically equivalent and so it is possible moving from one representation to the other [49, 104]. Translating two different models into the same form of representation allows an easy comparison, especially for non-expert users. While machine learning experts are able to move from one form to the other on their own, for non-experts this could be difficult. For this reason, presenting the representations to be compared directly in the same form helps users in the comparison task.

The representation of a single prediction instead has the purpose of clarifying not how the entire model works but why the particular decision was made. Lipton calls this purpose *post-hoc interpretability* because the problem can be addressed only after that the model has been trained and the prediction made [64]. The prediction's representation can assume different forms and the choice of which representation to use depends on the machine learning algorithm and on the application domain. In the following part some of the section, possible types of prediction's representations are described. Lipton uses also the term explanation to indicate the representation [64]. The goal of these representations is in fact to explain the decision of a model.

Textual form or *"text explanation"* [64]: the prediction's explanation is presented in a textual form. This form is frequently used by recommender systems [116]. The system not only provide the recommendation but also a textual explanation of why it was recommended. Lipton underlines the importance of text explanation because it is similar to human behavior [64]. Humans usually explain why they have made a particular decision verbally [64]. This form is also applied in the medical domain. The text explanation is important not only as a support for doctors in the decision-making process but also for allowing a better communication between medical experts and patients [51].

Visualization [64]: the representation is a visualization of what the model has learned. This approach is frequently used for understanding neural networks processing. An example is the famous project of Google known as "Deep Dream" or "Inceptionism" [72]. The aim is to visualize what the neural network has learned on how to classify the training images. In order to understand what the model has learned for a specific class, an noisy image is presented and then it is altered in the direction of what the neural network studied as being of this class, through gradient descent [72]. The visualization form is also used in recommender systems. In movie recommender, the recommendation is usually done thanks to the previous movies and videos that the user has seen but also by looking at the behavior of users with similar characteristics, also called neighbors. Herlocker et al. present a visual explanation of a recommended movie showing the movie's rate of the neighbors [48]. The authors proposed different forms, as shown in figure 2.7. The user-based survey conducted shows that the standard bar histogram form is the form judged more comprehensible by the final users [48]. The visualization form is also used to underline what are the instance's attributes that have more influence in the prediction. In this direction, a great contribute was given by Kononenko et al. that proposed a horizontal bar histogram, where for each attribute its contribute in the prediction is shown [90, 95, 108].


XJ*$5@ (2000)

Personalized Prediction : ★★★★ (will enjoy it)

Your Neighbors' Ratings for this Movie

Rating	Number of Neighbors
★	1
★★	2
★★★	7
★★★★	14
★★★★★	9

Figure 2.7: Examples of visualizations for explaining a recommended movies [48]

Local explanation [64]: the explanation reflects how the model work in the vicinity of the instance we want to explain [88]. Ribeiro et al. train a interpretable classifier in the locality of the prediction they want to explain, using as training data the ones labeled by the black-box model they want to explain [88]. The representation depends then on if the data are structured or images. In particular for text classification the authors proposes a bar chart representation: for the relevant words its contribute is shown, differentiated with colors for the different class label [88]. For images instead, the representation highlights, in the image that has been classified and that they want to explain, only the parts that are meaningful for the prediction.

Explanation by example [64]: another way humans use to explain their decisions is by giving an example [64]. Humans use examples not only because it is a simply and a direct way for explaining, but also because examples reflect also how humans think: humans act and choose based on their experience. For instance, medical experts often use previous case studies as a support for their decisions and they refer to them as precedents [64]. Caruana et al. developed a method that provides which are the most similar cases, on the training set, of the instance that has to be explained [20]. This kind of explanation, that they call "case-based explanation", is typical of case-based algorithms, as K-Nearest neighbors (KNN). KNN finds, among the training set, the K cases that are closest to the instance that has to be classified, using a distance metric as for example the Euclidean Distance [28]. Caruana et al. implemented an algorithm to obtain case-base explanations also from not case-based learning methods [20]. Explanations by example are also provided by recommender systems: products can be recommended because similar to a product liked before.

Now that all these kinds of forms of representation have been described and exemplified, it is possible to now illustrate how the user-based surveys based are conducted and then to underline their pros and cons. As previously mentioned, user-based surveys are a way to measure the interpretability of models' and predictions' representations. Representations are presented to users, along with questions that have the aim to qualitative measuring the interpretability.

Different authors have investigated how this kind of surveys have to be designed [49, 80]. They all agree on the importance of the choice of which users will take part in the survey. As already mentioned, education, age, mother tongue, cultural background and past experience have a great impact on survey's results [13]. Intepretability has a subjective nature: what is understandable for one person, can be incomprehensible for another one and vice versa. For this reason, it is relevant the users' selection. Based on the application and on the survey designers, the respondent group can be only one and homogeneous [49] or split in different but homogeneous subgroups to allow also the comparison of the evaluated interpretability.

Once the test group has been determined, the survey can take place. In general, a short introduction and description of the representation used is given to the respondents [49]. After that, the experiment starts: a series of questions are presented and the users have to answer more precisely but quickly as they can. The type of questions changes not only for the different opinions that the survey designers have but also for the type of representation. If the representation regards the entire model is possible, as an example, to try to perform a classification of an instance. If instead the representation is only of the prediction, this cannot be done because the model's processing is not provided.

Piltaver et al. has describe precisely what kind of questions to for measuring the interpretability of model representation [80]. Their work was base only on classifications trees but they suggest that this could be extend also on other intelligible models' representations as classification rules. In the following part the type of question suggested by Piltaver er al. will be described, also underlining analogies and differences with other authors.

The authors Piltaver er al. divide the survey into six tasks [80]:

- *classify*: the user has to perform an instance's classification using the classifier's representation provided [80]. This question is also present in the experiment definition of Huysmans et al. [49]. In addiction, the authors ask the users to indicate how they were confident in the answer they have given [49].

- *explain*: the respondent has to indicate which are the most important attributes for a classified instance. In other word, the respondent has to understand which are the attributes whose change will also change the prediction to another class [80].

- *validate*: the model's representation and a statement regarding it are presented and the respondents have to indicate if the statement is proper or not. The statement can be for example a correlation and the users have to say if this can be find in the model or not [80]. Huysmans et al. present the statement as a logical question and the respondents have to answer it simply with a "YES" or "NO".

- *discover*: the respondent has to try to find an unusual correlation in the model [80]. As already illustrate in section 2.2, debugging is one of the desiderata of interpretability. If the model is intelligible enough to give insight of its internal working, it is possible to investigate the correlations the model has studied. In case of unusual correlations, the analysis is even more important and it can result in the discovery of new patterns or of errors in the model. In the latter case, after having understand the cause of the wrong correlation, it is possible to try to fix it [21].

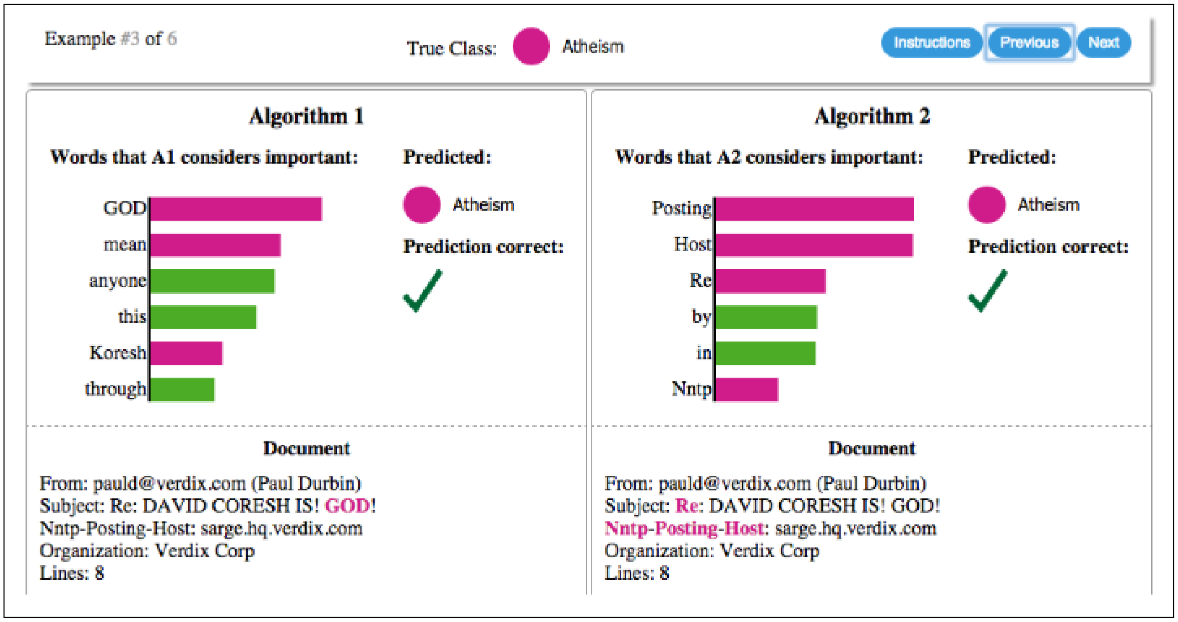
- *rate*: the respondent has to give its own opinion on the grade of interpretability of the representation provided [80].

- *compare*: the task is the comparison of two models. In the experiment of Piltaver et. al the comparison was between two classification trees. The comparison can also be between model's representations of different types: in this case participants have to indicate which is the type of representation that they find more comprehensible and the less one [49].

After answers have been collected, the following phase consists on the computation of some metrics in order to evaluate the interpretability. The *accuracy* is measured as the percentage of correctly answered questions [49, 80]. The *answer time* is the time it takes for a participant to answer a question [49]. The more is the time needed, the less the representation is understandable and consequently also the model itself [80]. Accuracy and time are strictly related: accuracy has to be evaluate taking into account also the time needed to answer. Otherwise, it "could be argued that any model could be understood given an infinite amount of time" [64] and so without the time constraint it could not be possible to measure and compare interpretability. Then also the *confidence* is evaluated, considering the personal judgment given by participants on how they were confident in answering the questions [49, 80].

Even if the user-based approach has been indicated as the more appropriate approach for evaluating intrepretability [14, 40], it presents many problems and limitations. This approach requires the active participation of the respondents; respondents, during the experiment, could lose motivation or get tired and this has a great impact on the survey validity [80]. Survey conductors should take this problem into account and for example carefully chose the order and the difficulty of the questions [80]. Surveys require much time to be designed and to be completed. The number of participants should be enough high to grant the achievement of statistically significant results [80]. As previously mentioned, participants have to be chosen with attention, in order to avoid bias inducted by the different background and experience. The choice can be made only having information on the participants, as their curriculum, their education and so on [49]. This arises concerns on ethics and privacy granting. In addiction, respondents should known the application domain on which the survey is done [80]. While for some domain no particular knowledge it is required, for others, as the medical or the finance ones, the participants should have a prior knowledge [80]. Contrarily, it would be complicate distinguishing case of non-interpretabily: the model could be non comprehended because it is actually hardly intelligible or because

the participants are not familiar with the classification domain.
With user-based surveys it is possible to compare representations but not to quantify interpretability [14]. Furthermore, as noted by Bibal and Frénay, the comparison is of representations, not of models or predictions themselves [14].



Algorithm 1

Words that A1 considers important:

Predicted: ● Atheism

Prediction correct: ✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Algorithm 2

Words that A2 considers important:

Predicted: ● Atheism

Prediction correct: ✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Figure 2.8: Explanations of an individual prediction for two different classifiers [88]. In magenta the words that are relevant for the attribution to the class "Atheism", in green for the class "Christianity". On the left, the explanation for the second classifier, based on the cleaned data, while the figure on the right refer to the first classifier [88].

The methodology described is relative to the comparison of model's representations. Ribeiro et al. proposed an experiment to compare single predictions of different classifiers [88]. They trained two SVM classifiers for text classification with the aim to determine if the document is about "Christianity" or "Atheism". The two classifiers were trained with two different data sets; the second one in particularly was a "cleaned" version were non generalizing features were removed [88]. The first classifier, even if it was based on unclean data, has a greater test accuracy with respect to the second one. During the survey, two different explanations, one for each classifier, of the same instance were shown to the users and the the participants had to indicated which were, in their opinion, the more faithful explanation [88]. The results shown that the users preferred the second classifiers. Through explanations, also non-machine learning experts were able to detect that the second model performed the classification relying on non pertinent attributes. As shown in figure 2.8 on the right, the document was assigned to the class “Atheism” for the words “Posting”, “Host”, “Re” and “Nntp” that obviously are not related to Atheism and Christianity. In the cleaned data set this problem was removed and the explanations reflects it. The participants preferred the second classifier because the explanation was more comprehensible and consistent with their prior knowledge. In addiction, this example shows the importance of considering the understandability of the model before the accuracy. In fact, if we only consider accuracy, the classifier to be used should be the first. For this reason, some authors suggest to firstly chose the models whose representations have the highest interpretability and then chose among them based on the accuracy metric [64, 112].

Chapter 3

Interpretability of Machine Learning Models

In the previous chapter, the importance of interpretability has been discussed, concentrating also on its definition and on how to measure it. This chapter's aim is to provide a brief illustration of machine learning algorithms, focusing on the intepretability aspect. In the first section, the defined-to-be-interpretable models will be described and this definition will be justified. The second section instead illustrates black box models, that means that their internal working is unknown. For these models, we also present model-dependent solutions for improving the interpretability. It means that these approaches are applicable only to a particular model with the aim of making it interpretable as a whole or at least for describing why it made a particular decision.

3.1 Explainable models

In this section, a briefly description of the defined-to-be-interpretable classification models will be presented. We use the word defined-to-be-interpretable because, as it will be more clear after the descriptions of these models, they are not equally interpretable and their true comprehensibility depends on different factors like their size, their complexity and also on the domain in which they are applied.

Classification trees

Classification tree models are known for their interpretable tree graphical representation. The common strategy of decision tree learners, as the C4.5 proposed by Quilan [84] or CART [16], is a top-down greedy approach. The predictors' space is split in sub-regions and a new instance is classified to the most common class value of the region in which that instance belongs. The split is made recursively, starting from the whole input space. At each step, it is chosen the best split, the one that minimize the heterogeneity of class values in each partition [47]. It corresponds to the selection of the best splitting attribute and value and this determines a new internal node of the tree. The recursion stops when there is no possible split left or if in that node a stop criteria is reached, like the minimum number of instances in a region or the maximum depth of the whole resultant tree. The best attribute and cutpoint for the split are determined using metrics that measure the impurity of a node, as the Gini index, the classification error rate and entropy [47]. The classification tree is so characterized by internal nodes and leaf nodes. Each interior node corresponds to one of the input attribute, the one chosen for the split and its structure depends on the attribute type and also on the number of edges [98]. They specify the condition to be tested. The leaves instead are the terminal nodes and represent the regions in which the recursive algorithm has partitioned the input space. Each leaf represent a value of the class, the most common class value of the instances of the training data set that fall back in that leaf.

Classification tree shows the complete internal working of the model. Users can get the *full picture* about the model and so they can understand how the model globally works [40]. Users can also understand why a single decision is made: starting from the root node, they can follow the path indicated by the internal node until a leaf node is reached [49]. This path represents a classification decision rule. The comprehension of the tree is not only facilitated by the graphical representation but also by the fact that in a tree generally only a subset of the attributes are considered, the ones that are considered as the relevant ones by model and present in the internal nodes [40]. In addiction, a great advantage of classification trees is that they have a hierarchical tree structure [40]. This allows to assigns a different importance to the different attributes, based on their distance from the root: in general it is considered more relevant for the classification task an attributes that is closer to the root [40]. This criteria however has been criticized and it is suggested to prefer others, based on the number of instances that are classified by this attribute. The idea is to consider relevant an attribute that is in the classification path of many instances and so that discriminate the most.

The problem is that in real applications decision trees can be so complex and large that can be hard for users to comprehend them. The comprehensibility is in fact strictly related to the size of the model [7, 40, 91]. The greater is the size, the less the model is considered understandable, since the size is a measure of model complexity. For classification tree models, the size can be estimated by the number of nodes or by its depth. Even if for some applications, like the medical one, larger trees are preferred, since too small models are considered too simple to capture the complexity of the problem [35], tree can be so big and complex that its understanding would require too much time and effort. The model understandability is in fact also linked to the time required to grasp the model. As Bibal and Frènay suggest, if we do not consider also the time, “it could be argued that any model could be understood given an infinite amount of time” [14]. A too large and complex model cannot be easily understood by us as humans for our inherent limitations. Some authors states that human beings are able to deal with seven, plus or minus two, entities at the same time while others limit this number to seven [71, 112].

It is for this reason that we introduce classification trees as defined-to-be interpretable models. Even if in general classification trees are considered comprehensible, in real applications this is not always true. In addiction, since interpretability is a subjective concept, some users consider tree less understandable than other representations, like rules or decision tables and so the so called interpretable models are so not equally interpretable [49]. Classification tree models, despite their great advantage of being in the general case interpretable, have the drawback of being typically less accurate than other classification models. This can be ascribed to the accuracy-interpretability trade-off: the more a model is comprehensible, the less is accurate [40]. For this reason, more accurate models, though less interpretable, are preferred than the underperforming classification trees for real applications.

Interpreting classification rules

Classification rules are of the form *IF (conditions) THEN class*, where conditions are a conjunction of attribute tests in the form: A_1=v_1 *and* A_2=v_2 *and ... and* A_k=v_k [98]. The set of conditions are the rule antecedent, while the class label is the rule consequent. There are different approaches for building classification rules, but we can classify them in direct methods and indirect methods [98]. The direct methods extract rules directly from the data [98]. An example of direct approach is the sequential covering methodology. The rule induction process for this approach consists on discovery rules one-at-a-time. At each stage, the most promising rule is selected, proceeding greedily and so finding the best choice only for that particular stage, not the optimal one. Once that the best local rule is selected, all the instances of the training set that are covered by these rules are removed [98, 111]. Examples of sequential covering algorithms are RIPPER [23], CN2 [22] and FOIL [83]. Even if these algorithms usually achieve equally or higher accuracy than traditional classification methods as C4.5 [85], their heuristic and greedy process do not guarantee to select the best set of rules [111].

For this reason, other solutions for extracting directly rules are preferred. In particularly, recently another approach has been proposed, known with the name of associative classification. Associative classifiers combine association mining and classification: the discovery of association patterns in a data set are used for generating classification rules that has to categorize new data, and so for performing the classification [97]. In general, an association rule assumes the form: A → B, where A and B are sets of item and an item is a pair *(attribute, value)* [1]. In associative classification, rules are used for classification purposes and so B is not a set of items but a class label [65]. Several techniques have been proposed for extracting the set of association rules and they distinguish themselves particularly on how they perform the rule pruning for extracting high-quality rules and on what rule mining algorithm (e.g. Apropri [2] or FP-Growth [44]) they use for finding the associations [97]. Examples of associative classifiers are DeEPS [61], CPAR [114], CMAR [62], CBA [65] and L^3 [9] and experiments show that they usually achieve better performances than traditional classification methods [97]. In addiction, associative classifiers tend to obtain better results also than traditional rule-based algorithms that are based on sequential covering [111, 114]. The reason is that sequential covering approaches rely on greedy techniques and so they consider the problem not globally, but they simply try to find the best solution only at each step; mined rules instead consider the correlations of different attributes [9, 111].

With indirect methods instead we refer to the approaches that extract rules not directly from data but from classification models [98], as the C4.5 algorithms that derive the rules from an unpruned tree. Recently, many algorithms for extracting rules from models have been developed, particularly from neural networks [6, 99]. The motivation is that is assuming, as already illustrated in section 2.1, increasingly importance the demand of interpretability also for the so called black box models.

Classification rules have in fact the great advantage of being considered highly understandable by humans. Rules are represented in a textual form and this is seen often as a drawback since it makes more difficult to get the "full picture" of the model [40]. It is argued that a graphical structure like the one of classification trees allows a more direct comprehension of the problem. However, as already mentioned, trees could be so complex and large that this is still not possible. The rules allows an easier comprehension of the problem locally or of why a particular decision is made. It is simpler in fact to inspect a single or subsets of rules and to investigate the problem locally rather than identify the paths from the classification tree [85]. The inspection of the relevant paths could require much time and effort if the decision tree is large and particularly in this cases a rule-based representation is preferred. The rule set comprehension is also problematic if its size is too big. The number of rules are considered as a measure of interpretability: too numerous rules are too difficult to be inspected and comprehended by humans. Classification rules, differently than trees, do not have a hierarchical positional representation and this makes more difficult the recognition of which attributes are more relevant in a rule [40]. This is instead simpler for trees: usually we assign a greater importance to the attributes that are closer to the root [40]. But as already mentioned, this is not usually the best approach for determining attributes' importance. So, for estimating the relevance of an attribute, it has been proposed to consider how many instances have been classified by rule that contains that attribute [40]. In this way, through the inspection of the rule that classifies a particular instance, it is possible not only to understand why that particular prediction is made but also to estimate the importance of each attribute present in that rule.

Naive Bayes

The Naive Bayes classifier is a probabilistic classifier that, as the name suggests, is based on the application of the Bayes's theorem but with the "naive" assumption of conditional independence of the attributes given the class [42, 46]. It is a probabilistic classifiers because, given the instance to classify with values V_i, it is able to returns the probability for each class C_k, computed in the following way:

$$P(C_k|\ V_1, V_2, ..., V_n) = P(C_i)\ \prod_i \frac{P(V_i|C_k)}{P(V_1,V_2,...,V_n)}$$

The instance is assigned to the most probable class, and so to the class with respect to which the term $P(C_i|\ V_1, V_2, ..., V_n)$ is greater. Since the class assignment is based on a maximization, the constant term $P(V_1, V_2, ..., V_n)$ can be omitted and so the class label y is determined as:

$$y = \arg \max_{k \in 1,2,..K} = P(C_k)\ \prod_i P(V_i|C_k)$$

The Naive Bayes model is considered interpretable because users can analyze the probability associated with all the attributed [40]. It is possible to compute the contributions of features' values easily and so understanding what are the most relevant features' values that are determinants for the prediction. During the training phase, all the terms $p(V_i \mid c_k)$ are computed. This information can then be used to calculate the contribution of each feature's value $\mathrm{V_i}$, based on the Bayes' Theorem:

$$p(c_k \mid V_i)=p(c_k)\,p(V_i \mid c_k)\,/p(V_i)$$

In this way, it is possible to analyze what are the features that influence mostly the prediction. Naive Bayes classifiers are greatly used in medical application, where explaining why a decision is made is required. In addiction, differently than classification trees or rules where only subsets of attributes are considered, Naive Bayes associates to each attribute a probability. As Lavrac noted, "one of the main advantages of this [Naive Bayes] approach, [...], is that all the available information is used to explain the decision" and this is considered extremely important for medical diagnosis and prognosis [59]. Medical experts prefer to have a more complete picture of the problem, which takes into account all the variables [53, 59]. Even if it argued that the probabilistic interpretation is important in medical practise, not all users are able to easily deal with probabilities. For this reason, some methods have been proposed for improving Naive Bayes interpretability. As an example, Kohavi et al. proposed an visualization method that highlights through rows of pie charts or through a bar representation the prior probabilities for the possible label values [12]. Kononenko and Kukar proposed to evaluate the contributions of individual feature value applying the logarithm to the model's equation [53, 54].

Interpreting Nearest Neighbors

K-Nearest Neighbors (KNN) is an instance based classifier since it is not learned a model neither abstractions derived from the instance are derived or maintained, but all the knowledge is embedded on the training instances themselves [3]. It is a very simple algorithm: the idea is to classify a new instance based on the most K similar instances of the training set [98]. The training phase consists only on the storing of the training data and no model is created. The classification phase instead consists on computing the K nearest neighbors to the instance x that we want to classify and these are used for assigning the class label; usually the class is assigned by majority vote and so the class label of x is the most common class among the K neighbors [98]. The K nearest instances are identified using a distance metric as the Euclidean, the Mahalanobis or the Manhattan distance. For improving the classification results, we can weigh the vote according to the distance with respect to the neighbors and thus assign a greater contribution to the nearest instances [98].

For what concerns its interpretabily, obviously this algorithm does not return an interpretable model, since it does not discover any abstract model. It instead can explain why a particular instance is assigned to a class. The form of explanation provided by the KNN has been defined *explanation by example* [64]. The explanation is given by the K neighbors themselves: the instance x has been classified in a particular way based on the most similar instances. So the K instances are examples of why that particular prediction is made. This reflects how we, as humans, often explain our decisions, that is by analogy [64]. For example, medical experts often take and justify their decisions through precedent case studies. Recommender systems also often use this from of explanation for supporting their recommendation; in movie recommendation, to a user it is presented not only the suggested movie but also its justification, for example saying that users with similar characteristics liked it or that this movie is similar to other movies for which the user has expressed a positive rating.

However, this form of interpretabily has some drawbacks. KNN returns the most similar instances but it does not highlight what are the most relevant attributes for the prediction. In the distance computation, usually all features have the same weight and so the distance is computed as all the features equally influence the prediction. Some solutions have been proposed, actually originally for improving the accuracy prediction, that weight the attribute so that the attribute weight is proportional to its relevance for the prediction [113]. Since no model is created, it is not possible to understand what is the global behavior. In addiction, since a single prediction is based only on its locality, what is important for a prediction can be totally different than what is relevant for another. A single rules or single paths instead can provide explanation for sets of prediction [40].

3.2 Model-dependent solutions

In the previous section, the so called interpretable model have been briefly described, focusing on what level of interpretability they can provide. In this section, some hardly interpretable models will be briefly described and some solutions for improving their interpretability will be illustrated. The approaches presented are *model-dependent*: a method is model-dependent if it applicable only for explaining a particular model.

Random Forests

Random forests are an ensemble learning method that is based on bagged trees but also on a random sub-sampling method [17, 47]. The bagging or bootstrap aggregation is a general-purpose procedure for reducing the variance of a model. With respect to classification, the idea is to select random samples with replacement from the training data set, B times; for each sample $b{=}1,\ldots,B$ we learn our model and obtaining the prediction $\hat{f}_b$ [47]. The prediction for a new instance x is made by taking the majority vote and so the class is the class that occurs more commonly between the B predictions $\hat{f}_b$. The random forest method is an improvement of bagged trees through an adjustment that reduce the correlation between trees [47]. The idea is to use at each node for the split, instead of all the features, a random subset of the features $m{\leq}p$, where p is the number of features; typically it is used $m{=}\sqrt{p}$ [47]. A random forest consists of a large number of trees and so it is not possible to understand its internal process and so it is considered a black box. One way of getting insights into a random forest was proposed by Breiman itself, in the paper that describes his Random Forests algorithm, based on the computation of the features importance [17].

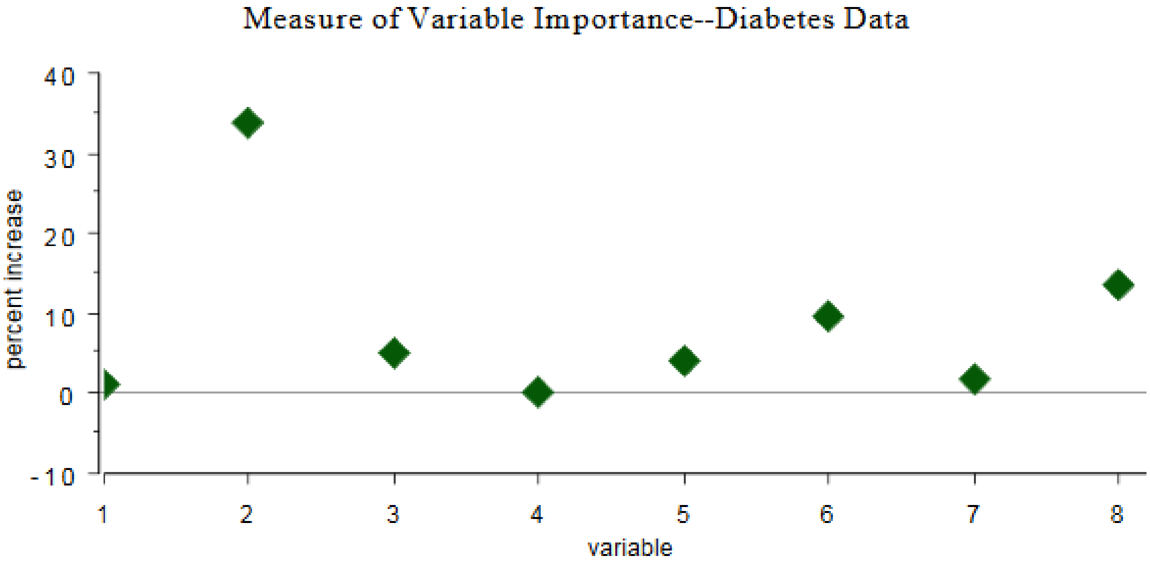


Figure 3.1: Example of visualization of feature importance for explaining Random Forest models proposed by Breiman [17]. The example shows the features' importance of the diabetes data set. For each possible variable, in the horizontal axis, there is the correspondent importance on the vertical axis. The most important variable is the variable "2", followed by variable "8" and "6".

After that the random forest is learned, the out-of-bag estimate is stored for each instance of data set. For determining the importance of a particular variable m-th, the values of the m-th feature are randomly permuted and the out-of-bag estimate is computed again. The importance of the m-th feature is computed comparing the out-of-bag estimate before and after the permutation [17]. This is repeated for each feature $m=1,..,M$ and these feature importance estimations can then be visualized as in figure 3.1: the greater is the importance value, the more the feature is important for the predictions. The point is that this approach is limited only to Random Forest because it is based on intrinsic characteristics of the algorithm.

Support Vector Machines

Support vector machines (SVM) are a supervised machine learning method based on finding the hyperplane or the set of hyperplanes that will separate better the data [27, 105]. Support vector machines, for classification applications, were firstly introduced as binary classifier and so for two-group classification problems [27] but their application can be extended also for multi-class problems using methods, that are based on bynary classification, as the one-versus-all (OVA) or the one-versus-one (OVO) approaches [47]. Suppose that $D=\{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2),..., (\boldsymbol{x_n}, y_n)\}$ is the training data set, where $x \in \Re^d$ and $y_i \in \{-1,1\}$ indicates the class to which x_i belongs, in a binary classification problem [47]. The idea is to choose, among all the possible hyperplanes in the form:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \alpha_i < \boldsymbol{x}, \boldsymbol{x_i} >$$

written in function of of inner products, the one that maximizes the distance between the two classes determined by the hyperplane itself, called *margin* [47]. This is resolved as a constrained optimization problem, because the margin is maximized subject to the constraint of $\boldsymbol{\alpha_i}$ being a normal vector to the hyperplane. In addiction, the margin is *soft*, that means that it is maximized but it can be subject to errors [27]. This is a relaxation of the problem, in order to allow the binary classifier to also deal with noisy or non linear separable data. The point is that if the data are hardly linearly separable, even this approach is not enough. For this reason, the solution proposed is to apply the so called *kernel trick* and so the hyperplane's function can be rewritten as [15]:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \alpha_i K(\boldsymbol{x}, \boldsymbol{x_i})$$

where possible examples of kernel function frequently used are the radial basis function, the hyperbolic tangent and the polynomial kernel. The great advantage of this approach is that, through kernels, data can be transformed into an higher dimensional space and the hyperplane in the enlarged space can be nonlinear in the original space, with a minimum effect on the computational time [47]. The name *support vector machines* derives from the support vectors, the samples closest to the the hyperplane that "support" the decision margin. Support vector classifiers often exhibit excellent predictive performance, but it is difficult to explain the results obtained. The SVM model could be presented through its support vectors, the subset of observation that determines the boundary, but it is only a reduction in the number of instances to consider [50]. Another possibility is to directly visualize the SVM model in the feature space, but the problem is that this is appropriate only if the space has no more than two or three dimensions; if greater, the visualization could be too complex to be understood and visualizing high-dimensional data is generally difficult.

Different solutions have been proposed for improving the interpretability of SVM models. Jakulin et al. propose to use nomograms for visualizing the contribution of feature values for SVM models. Nomograms are not a novelty: they were firstly introduced at the end of the 19th century by the mathematician Maurice d'Ocagne as a visualization technique for graphically representing any numerical relationship. Lubsen and coauthors proposed to use nomograms to visualize logistic regression models [68], while, more recently, the authors Možina et al. propose to visualize also the Naive Bayesian model in the form of a nomogram [73]. Nomograms enable the model visualization and to gain insight into the data, summarizing how the attributes influence the class probability [73]. So, in order to build the corresponding nomogram, a model has to predict the outcome probabilities. The point is that basic SVM models do not deal with probabilities properly by design. For this reason, it is needed to place a calibration phase before, that allows to better calibrate the model's probabilities and in the examples proposed by the authors Jakulin et al. the cross-calibration is used [50]. Since nomograms can be used to visualize Naive Bayes, Logistic Regression and also SVM models, they can be used also to compare models, but only limited to these type of classifiers. In figure 3.2, an example of comparison between the Naive Bayes and SVM nomograms is reported, using the log odds ratio scale, for the *Titanic* data set [50]. Each single attribute is presented in a distinct line in the nomogram that indicates the attribute importance for the particular model displayed through the nomogram [50]. The greater is the length of the line, the greater is the influence that the corresponding attribute has on the model's predictions. In the example, for the

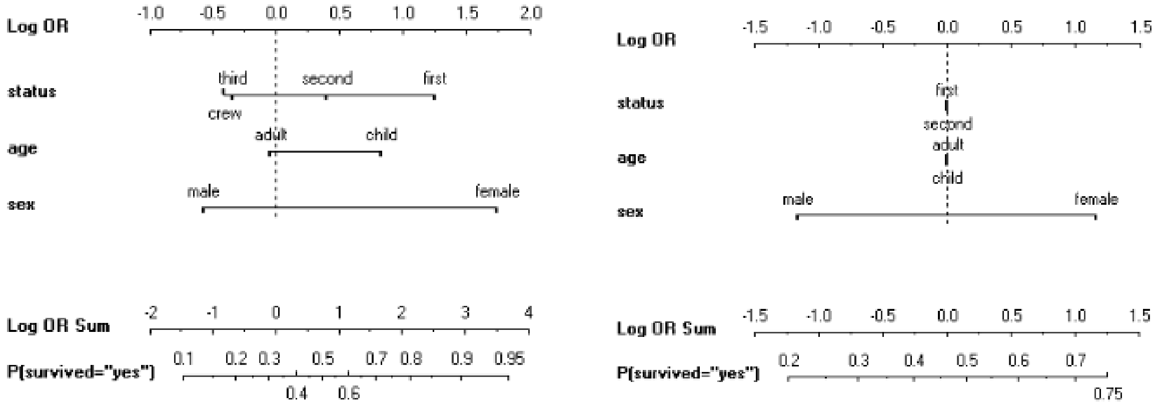


Figure 3.2: Example of comparison of a Naive Bayes nomograms, on the left, and of a SVM nomogram, for the *Titanic* dataset [50]

Naive Bayes classifier, the greater influence is given by the attribute *sex* and females have more chances of survival than males; being of the first class and a child increase the chances, but with less influence. For the SVM instead, on the right of the picture 3.2, the only important attribute is the *sex* attribute, while the others are considered irrelevant [50]. It so interesting to notice how the two models are different and we can state that the Naive Bayes is more complete since it considers more information. The comparison of models with nomograms so allows not only to visualize what the different models have captured but also a more weighted choice of which model to apply. Nomograms have the great advantage of being a visualization tool: they do not try to approximate the behavior of the SVM model, but they simply capture how the SVM model behaves and show it globally [50]. In addiction, particularly when the number of attributes is high, the visualization could be simplified showing only a part of the attributes, the ones that contribute the most and so that have the greater importance or ranked in order of importance [50]. The comparison between nomograms of different models, as already mentioned, is another great advantage but the problem is that nomograms can be applied for explaining few models, as the SVM, the Naive Bayes and Logistic regression ones. Moreover, nomograms present problems in handling redundant or highly correlated attributes and in these cases, it is difficult to distinguish the individual effect of each particular attribute [50]. The point is that the attributes' contributions are computed considering as the effects of the attributes are independent given the class [50].

Other solutions have been proposed for explaining SVM models. In particular, need to be mentioned some recent works that focuses on visualizing SVM, as the projection techniques proposed by Caragea et al. [19] and by Poulet [81]. Their idea is to project the high-dimensional feature space into lower dimensional subspaces; these projections are simpler to understand and can give insights on how the model works.

Artificial Neural Networks

Artificial neural networks (ANN) are a class of techniques inspired to the biological neural networks of the human brain. There are many various types of ANN in that differ for example for the topology, the number of hidden layers, the learning algorithm or the transfer function. Despite the differences, artificial neural networks are based on a set of connected units, called artificial neurons, and each connection is weighted, called synapse. Neurons in an ANN are usually structured in connected layers: input, one or more hidden layers and output layer. The activation of a neural network is based on the training phase. The weights are set at random and then iteratively modified until the resultant network minimizes the error at the output, estimated comparing the resulting outputs and the desired outputs [43]. After the learning phase, an ANN is able to represent an high-dimensional non-linear function [100]. The non-linearity into the network is introduced by the non-linear activation function, also called transfer function, that, given the inputs of a particular node, defines its output [43]. The point is that the information learned is stored in the weights and in the connections themselves that do not provide direct evidence of the what are the relationships between inputs and outputs that the network has learned and the comprehension of how the network works is made more complicated by its non-linearity. Despite the great performances that neural networks often achieve, they are seen as black boxes and this limits their use and acceptance.

Considering their great accuracy and advantages, a lot of work has been done in the direction of improving neural networks interpretability. In particular, many techniques have been proposed for extracting explicit rules from ANN. Rules are in fact considered as a good compromise, since their are able to capture the complexity of the problem but at the same time they still are understandable, as discussed in section 3.1. Need to be mentioned the work of Towell and Shavlik [99], that proposed the *MofN* method for extracting symbolic rules from trained neural network and the one of Setiono and Liu, that extract *NN rules* that should reflect how the neural network works [92]. For a more complete review of these techniques for extracting rules from trained ANN we refer the reader to the survey published by the authors Andrews et al., that overviews these methods, also proposing a taxonomy for classifying them and criteria for evaluating them [6]. Other solutions are instead directed on visualizing the neural network graph. As an example, Tzeng and Ma propose to "open the black box" in order to possibly find out dependencies between the inputs and the outputs of an ANN; in their work, they propose to, given a single data or a set of data, display the network as data going through the network [100].

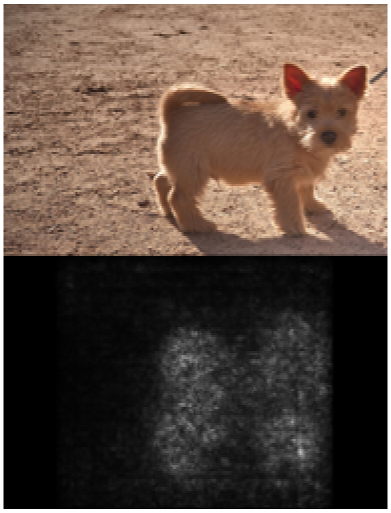


Figure 3.3: Example of image-specific class saliency map that highlight the parts of the image that are considered by the trained ConvNet discriminative for the prediction with respect to the top-1 predicted class [94].

Finally, much effort has been made for improving the understandability of image classification models, also due to the increasing application of deep Convolutional Networks (ConvNet) for image recognition. As an example Simonyan et al. propose an approach for visualizing, given a particular image and a target class, a *class saliency map* that underlines what are the important pixels of an image for that particular class assignation [94]. The image-specific class saliency map is computed using a single back-propagation pass through the trained ConvNet. The map extracted in this way highlights what are the pixels whose change affect the most the prediction for that class [94]. An example of image-specific class saliency map, provided by the authors in [94] is presented in figure 3.3, built using as target class the top-1 predicted class.

Chapter 4

Model-agnostic algorithms

In the previous chapter, some methods for improving machine learning interpretability have been presented. These methods are related only to particular machine learning classifiers and for this reason they are called **model-dependent explanation methods** [108]. The purpose of these methods is to improve the interpretability of hardly transparent model, providing more understandable representations. The limitation of these approaches is that each method can be applied only to a particular classifier. Distinct solutions have been proposed for distinct machine learning algorithms. As previously discussed, there are solutions for Random Forest classifiers, as the one proposed by Breiman [17], for Support Vector Machines and for Artificial Neural Networks. This chapter instead deals with model-independent methods [108], also referred with the term model-agnostic [88].

An approach is **model-agnostic** if it treats the machine learning model as a black box [88]. In terms of interpretability, model-agnostic approaches are able to provide explanations for any kind of classifier [88]. Model-agnostic methods have great advantages with respect to model-dependent ones. Ribeiro et al. summarize some of those into five desirable aspects [86]:

- *Model flexibility*: in real world applications, it often happens that less accurate but more understandable models are preferred to more accurate but less interpretable ones [21]. Accuracy itself is not considered as a sufficient metric for trusting a model: the users want to understand how the model works and this is particularly crucial for decision making [88]. The problem is that, in many cases, simple models are not enough to solve properly real-world task. The reality is complex and heterogeneous and interpretable algorithms are inherently inadequate to model it. For these reasons, there is a demand of complex but at the same time understandable solutions. The model-agnostic approach is the answer. The model is considered as a black box and an explanation of its internal working or of its prediction is given in the same way for any classifier [86]. In the choice of the model, the intrinsic interpretability of the algorithms is not anymore considered: what matters is the explanation given by the model-agnostic approach applied. In this way, there is a separation between the interpretability of a model and the model itself [86] and, as a consequence, there is more flexibility in choosing which to apply model. The model understanding and its accuracy can be considered independently and not anymore as a trade-off.

- *Explanation flexibility*: most of interpretable models are explained through defined and limited representations. For example, classification trees are usually explained through trees, rule-based classifiers through rules; in other cases the explanation is given through an example [20] or as line graphs [21]. The problem is that these representations of interpretable models are fixed: it is not possible to move from one form of representation to another. This possibility is important because each form of representation can highlight different information. As an example, in some cases it is preferable to point out which are the elements that influence the most a prediction, while in others which are the ones against it [86]. Using model-agnostic approaches, the explanations becomes separate from the model itself. In this way, the same model can be explained in different manners and it is possible to represent the explanations in the most suited form for each particular application [86].

- *Representation flexibility*: model-agnostic approaches allow also the flexibility in the representation itself. The features used for training the model can be different to the ones used for the explanation [86]. For example, for text classification, instead of using word embedding, the explanations can be in terms of words [86].

- *Lower cost to switch*: using a model-agnostic approach, there is a continuity of the explanations even if the model would be replaced. The explanation is separated from the model: the same form of explanation can be provided even if the model is switched with another one. This is true also in the case of new machine learning algorithms: it is not necessary to implement a new kind of explanation for new methods and the explanations' form can be still provided in the same way. This is convenient also for users because they do not have to lose time and make efforts to understand a new kind of explanation [108].

- *Comparing two models*: this is the great advantage of using model-agnostic explanations. The comparison of models in term of interpretability is one of the criteria for choosing the best model. Some authors emphasize the importance of interpretability and suggest, in order to chose the best model, to firstly select the more interpretable models and only then select the one with highest accuracy [112]. The problem is that, without using model-agnostic approaches, deciding which is the most interpretable model can be difficult. As already discussed in section 2.3.1, the comparison using heuristic approaches, that gives a quantitative measure of interpretability, is limited for models of the same type and even more challenging for models of the same types. As an example, it is difficult to define how to compare the number of nodes of a tree and the number of hyperplanes of a Support Vector Machine model [64]. Comparing the representations of two explanations is also problematic because they can be represented in different ways: for example it is not easy to say if an explanation by example, where the cases most similar to the instance to be explained are presented, is more or less interpretable than a bar plot that shows the contribution of each feature in the prediction. Interpretability is subjective and humans do not comprehend models' representations in the same way. A possible solutions is to evaluate the interpretability using user-based studies and to average the results [49, 80]. If instead we use a model agnostic approach, the models are seen as black boxes and the explanations can all be presented in the same and uniform form [108]. In addiction, some model-agnostic approaches supports the explanations of multiple models in the same plot and so a direct comparison between models is directly possible [56].

Model-agnostic algorithms can be applied to any type of classifier for obtaining a global understanding on how the model works or for providing an explanation of why a particular decision is made. In the following sections these two possible approaches will be illustrated, underlining advantages and limitation and providing a detailed description of some model-agnostic techniques.

4.1 Explaining the entire model

In this section, two examples for explaining the entire an entire model are illustrate. The idea of this kind of approaches presented is to *explain the model* and so to see how the model globally behaves. It is a more challenging task than explaining a single prediction. In addiction, understanding the model globally implies also to have a local comprehension and so to also understand why single predictions are made [88]. The opposite instead is not true. The first illustrated approach learns an interpretable model, in particular a classification tree on the prediction of the original model [30]. In the second solution instead, the inputs are perturbed and it is then observed how the predictions change [56]. The trend of each attribute is then plotted, showing for each value it can assume how it influences the prediction probability for a particular class. In the next sections, these two solutions will be briefly described.

4.1.1 CRAVEN

The authors Craven and Shavlik propose an algorithm for extracting comprehensible representation from a generic model f called TREPAN [30]. Given a generic model f, TREPAN extracts a decision tree that mimic globally the behavior of f. The authors' aim is to produce trees that are understandable and at the same time with an accuracy comparable to the one of the models from which they were extracted [30]. The application of this algorithm was described by the authors for explaining neural networks but, as also they note, TREPAN use the neural network as a black box and so it is applicable for explaining a generic model f [30]. This algorithm is based on queries to the model f and it is only interested in how f labels the instances. TREPAN uses the model for querying it and so it uses the model f as an "Oracle". The tree extraction algorithm is similar to tradition decision tree induction algorithms like C4.5 [85] and CART [16]. For learning the tree, as training data are used instances labeled by the model f. The first purpose of the model f seen as an Oracle is to determine the class label [30]. The second purpose of the Oracle is instead to select the best split for each internal node. The advantage of the tree extraction with respect to the usual one is that it does not use the training data but it queries the Oracle. What often happens in traditional tree induction algorithm is that with the increasing depth of the tree, in a node we can have too low training data and so the selection of the best split is based only on too few data. In the TREPAN implementation instead, if in a node there are too few examples, it can query the Oracle and obtain also new

labeled instances. The new instances have to follow firstly this particular constraint: their attributes values must be consistent to the ones present in the path from the root to the node that is considered for the splitting [30]. The values of the attributes that are not present in the path are instead selected randomly and the new instances generated are labeled by the model *f*.
Another difference from traditional tree extraction algorithm is that TREPAN employs a beam search method for growing the tree [30]. At each step, the best node is selected using the following evaluation function: *f(n) = reach (n) (1 - fidelity (n))*. The idea is to select the node that would potentially increase the fidelity to the model *f*. The term *reach(n)* is the fraction of instances that fall in that node *n*, while *fidelity(n)* is the estimation of how the resultant tree will be faithful to the model *f* if the node *n* is included. In particular the fidelity is the percentage of instances of the test set that are labeled by the tree in the same way by the model *f*. In addiction, while the traditional tree extraction algorithm based the split on a single feature, the algorithm proposed by Craven and Shavlik use a m-of-n test to partition the instance space [30]. A m-of-n expression is a boolean expression that is satisfied if at least m of its n conditions are satisfied.

TREPAN uses as stopping criteria both a local and a global one: a node becomes a leaf and so it is not further split if it covers only instances of one class with high probability or if the size of the tree would overcome a threshold value, indicated as a parameter by the user [30]. This parameter can control the tree comprehensibility, since the size of the node is an measure of heuristic measure of interpretability: the greater is the number of nodes, the less the tree is considered comprehensible, as already discussed in section 2.3.1. The authors compare the accuracy of the trees and neural networks from which they are extracted, for different data sets. They show that although the test-set accuracy of the trees is lower than that of the original network, it is still greater than the one of the decision trees learned from trained data using traditional tree classification algorithms. They also measured the fidelity of the extracted tree to the neural networks and they state that the extracted trees provide a close approximation [30]. The problem of this solution is that it is used a simple model, as the classification tree is, to mimic the behavior of the tree globally. We can argue that a simple model cannot capture the complexity of a sophisticate model as the neural networks are. In addiction, the authors state that the extracted trees are comprehensible as the tree learned by conventional classification tree algorithms. The point is that for measuring the complexity they use the size of the model, in particular the number of non-leaves node in the tree and the number of features used for the split.

As already mentioned, the size of a model is not a so good measure of interpretability: comprehensibility is strictly linked to humans and the same representation can be considered less or more understandable for different users.

4.1.2 Prospector: a visual inspection of black-box models

Prospector is a interactive visual analytics systems, developed by the authors Krause et al. for better understanding predictive models [56]. This system provide graphical representations of how features affect the predictions of a generic model f overall. This algorithm is based on the concept of partial dependence [47], that is a technique used for understanding the relationship between a feature and a prediction [56]. The idea is to consider the model f as a black-box and see how changes on an input feature affect the prediction. The partial dependence plot, *pdf*, is computed for one input at the time as follow [47, 56]:

$$pdf_a(v) = \frac{1}{N}\sum_{i}^{N} pred(x_i) with \quad x_{ia} = v$$

where a is the attribute respect which the partial dependence plot is computed, N is the number of instances and x is a particular instance. We change the value of the attribute a with the value v for each possible instance x_i and we see how the outcome probability, *pred*(x_i), changes, while the values of the other attributes are not varied [56]. The partial dependence plot of a particular attribute a is typically represented as a line graph where on the horizontal axis there is the possible values that it can assume and on the vertical axis the corresponding outcome probability. Prospector is applicable for visualizing the partial dependence plot for different models and so to easily compare them, since this algorithm treat the model as a black box [56]. In figure 4.1, the behavior of the same feature but for three different models is reported. The predictive models in the example had to predict the risk of developing diabetes [56]. The authors illustrate that in the medical domain, it is important to provide an explanation of why a particular decision is made. The general behavior of the features can be difficult to be inspected, particularly if the number of features is relevantly high. *Prospector* is able also to provide what are the the most relevant features for the prediction. For example for predicting diabetes, it return a summary of the top 5 features that contribute the most increasing the prediction of diabetes and the top 5 of the one that decrease instead the risk of diabetes.

The problem of this solution is that it relies on partial dependence that is computed for only one attribute at the time. The point is that in real-world applications,

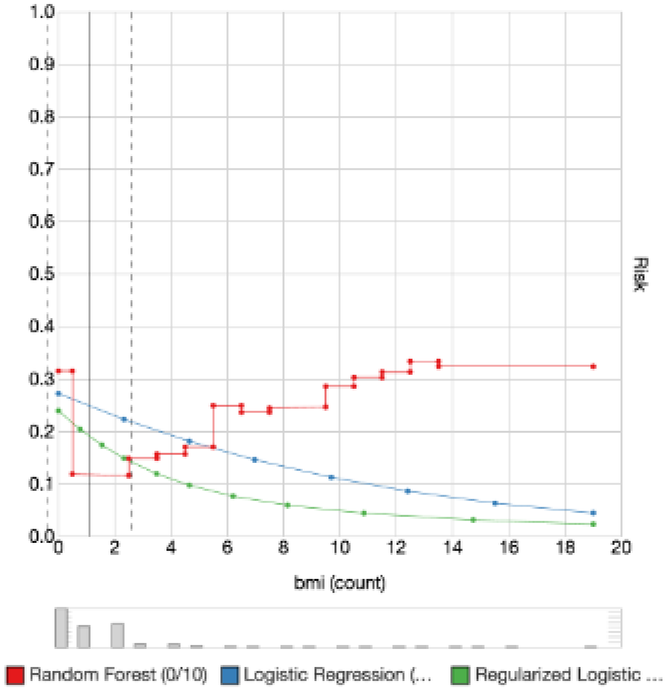


Figure 4.1: Partial dependence plot for comparing multiple models [56]. For each possible value of the bmi attribute, the corresponding outcome prediction is shown for each possible model.

attributes may be correlated and the prediction could be determined jointly by more of one attribute. Moreover, even if this solution is applicable for any classifier, it is not completely model agnostic. For obtain more accurate partial dependence plot in fact, *Prospector* uses information about inherent properties of the model it has to explain. In decision tree and random forest models for example, the predicted probability change only for specified thresholds, based on the condition of the internal nodes [56]. In this case, *Prospector* firstly extracts these thresholds from the rules in the internal nodes and then compute the partial dependence only for those values. So we can see that this approach is not purely model-agnostic, as it should treat the original model f without making any assumption about f [88].

4.2 Explaining single predictions

While model level explanation "aims to make transparent the prediction process of a particular model" [90], instance level explanation aims to to provide "a qualitative understanding of the relationship between the instance's components and the model's prediction" [88]. The intention is not to understand how the entire model works, but only to understand why a particular decision was made. This aim could be considered as less important or too limited. It is possible to show instead that explaining a prediction is what, in many applications, really matters and in some cases it is only convenient solution.

Explanations of an entire model, provided using an approach for improving the interpretability of the model or by the interpretable model itself, could be too complex to be really understood by humans. As an example, even if classification trees are considered interpretable, they may be so large and complex that it may not still be possible to comprehend the whole model. Humans are limited and they can deal with few entities at the same time [71, 112]. This problems are present even when model-agnostic approaches for explaining an entire model are used. As an example, *Prospector*, as illustrated in section 4.1.2, provides a graphical representation of the behaviour of all the features . If the number of features is high and the classification domain is complex, it is very difficult, even with this kind of explanation, to understand the model globally. A prediction's explanation instead regards only the locality of the instance to explain and only some features significantly contribute to it. For these reasons, its understanding is more simple.

In addition, a criticism can be advanced to those approaches that provide explanations for the entire model. Some of them, as the one described in section 4.1.1, try to learn an interpretable model on the predictions of the original one. The problem is that the interpretable model used is intrinsically more simple and less accurate. It can be considered over-simplistic the idea that, transforming a complex model into a simpler one, the same information that where encoded before will be present also in the simple and interpretable model.

Moreover, in many applications and in particular for decision making, users want to know why a particular prediction is made. This is particularly important for medical application because the model's decision can affect significantly people's lives. The authors Ribeiro et al. in their work propose a simple illustration for highlighting the importance of explaining individual predictions and it is reported in figure 4.2 [88]. Suppose we have a model that has to predict if a patient is ill or not. The explanation of why a particular prediction is made can help the doctor in the choice if trusting it or not. In this case, the model predicts that a particular patient has the flu and the explanation is in form of symptoms that are relevant for the prediction: in green the symptoms that lead to the prediction of flu and in pink the ones that are against it [88]. Doctors can inspect the relevant symptoms and they can decide if accepting or rejecting the prediction, confronting them with their prior knowledge of the disease [88].

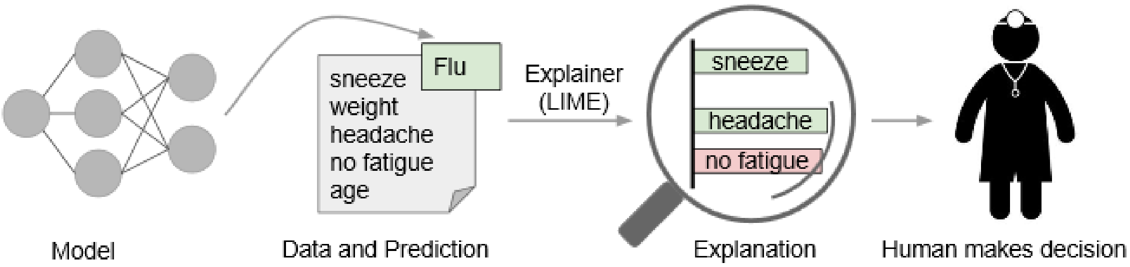


Figure 4.2: The explanation of a single prediction can help doctors in the decision of trusting the prediction or not [88]. They can analyze the relevant symptoms highlighted by the explanation and see if they are coherent with their prior knowledge or not.

The explanation of a single prediction can be easily understood by users, while the model's explanation could be too difficult to comprehend for its complexity.

In some cases, users do not need a global comprehension of the model. As an example, in movie recommendations systems, user want only to understand why a particular movie has been suggested and based on this explanation they can decide if accept the recommendation or not. In addition, it is not always possible to provide a global explanations also for legal and contractual reasons: some algorithms are proprietary and companies do not want that their internal working is provided.

In the following sections, two approaches for explaining individual predictions will be presented. These two solutions will be described in detail because they were inspiring for our work. It is important firstly to illustrate some key concepts that were pointed up by these two works, highlighting advantages and limitations.

4.2.1 LIME - Local Interpretable Model-Agnostic Explanations

Ribeiro et. al propose a method for explaining the prediction of any classifier, called LIME, Local Interpretable Model-Agnostic Explanations [88]. The great intuition of this work can be indeed summarized with the terms *local*. Some of the methods previously described try to build an interpretable model based on the predictions of the model they want to explain [30]. The problem is that in the general case the interpretable model learned cannot properly mimics the behavior of the model to be explained. Interpretable models are inherently simple, also for their capability to show their internal working in an understandable way. On the other hand, the models we want to explain are non-interpretable, they do not provide any understandable explanations of their working, precisely for their intrinsic complexity. A simpler model is not able to reflects all the heterogeneity of a more complex one.

Ribeiro et al. find a way to overcome this limitation. The problem is that the previous solution try to mimic the behavior of the entire model. The idea of the authors is to consider not globally all the predictions, but only the predictions that are local to the particular prediction they want to explain. They learn an interpretable model only locally around the prediction of interest [88]. They introduce the concepts of **locally faithful**: an explanation to be meaningful must correspond to how the model behaves in the vicinity of the instance being predicted. The model does not behave in the same way for all the predictions and this is a common situation in complex application. For Big Data problem, finding a global model is often unfeasible; the data are so sparse and heterogeneous that the model behaves very differently for different data. Describing a prediction only considering its locality can be the only possible solution in these cases. The authors underline that local fidelity does not imply global fidelity; the features that are relevant for a particular prediction may not be so important for the model as a whole and vice versa [88]. A globally faithful explanation is instead also locally faithful but obtaining it is problematic, in particular for complex models, as already described.

The method proposed by Ribeiro et al provides explanations of individual predictions for any classifier. Even if the explanations are for single predictions, the authors state that through representative single explanations it is possible to have also a global understanding on how the model works. This is important because, if users have insights on how the model works, they can decide if they can trust it or not. In addiction, if users do not trust the model because it presents some questionable behaviors, they can study these insights and debug the model [21, 88].

In order to understand how the explanation for a given instance is provided by LIME, some definitions have to be specified:

- g: the authors define g a model taken from a class of potential interpretable model G, that is $g \in G$ [88]. g can be for example a classification tree, a rule-based system, a linear model. These models are defined potentially interpretable because there is always to take into account the complexity-interpretability trade-off. As an example, even if a decision tree is usually referred as interpretable, if it is too large or if it has many nodes and so if it is very complex, it is actually non understandable by humans. g has for this reason to be simple enough to be comprehended and so its complexity has to minimized.

- $\Omega(g)$: this term is used as has a measure for complexity in order to evaluate the interpretability [88]. The more the model is complex, and so $\Omega(g)$ is high, the less the model is understandable. This way for measuring interpretability was already discussed in section 2.3.1. Interpretability is evaluated using a quantitative approach and it is estimated through the size of the model g. For a decision tree the size can be the number of nodes or its depth, for a linear model the number of non-zero elements, for a rule-based system the number of rules [14, 40].

- f: it is the model to be explained and it takes instances with d attributes and it provides for each one the class to which the instance belong, f: $\mathbb{R}^d \rightarrow \mathbb{R}$. $f(x)$ is the probability that the instance x belongs to the relevant class; the explanations are computed for each class separately in case of multi-class input data [88]. f can be any model because the solution proposed by Ribeiro et al. is model-agnostic.

- $\pi_x(z)$: used to express the locality between the instance x to be explained and a generic instance z [88]. It is a proximity measure: the greater is this term, the more z is near x.

- $\mathcal{L}(f, g, \pi_x)$: it is "a measure of how unfaithful g is in approximating f in the locality defined by π_{x}" [88]. The purpose is to minimize this terms because we want that g is able to mimic well the behavior of the model locally.

Once that all these terms have been described, it is possible to illustrate how the explanation provided by LIME is computed. The goal is to have a model g that at the same time approximate well the model f locally around the instance x to be explained but also that is interpretable enough to be understandable by humans [88]. So the point is to minimize at the same time $\mathcal{L}(f, g, \pi_x)$ and $\Omega(g)$. The explanation $\xi(x)$ of an instance x is computed by LIME in the following way [88]:

$$\xi(x) = \underset{g \,\in\, G}{\arg\min}\ \mathcal{L}(f,\ g,\ \pi_x) + \Omega(g)$$

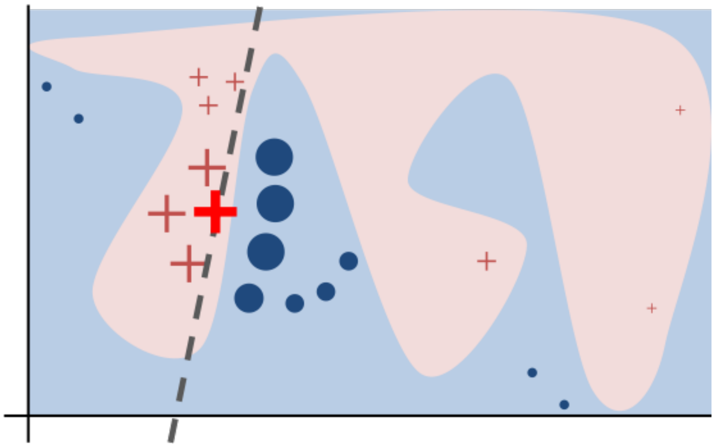


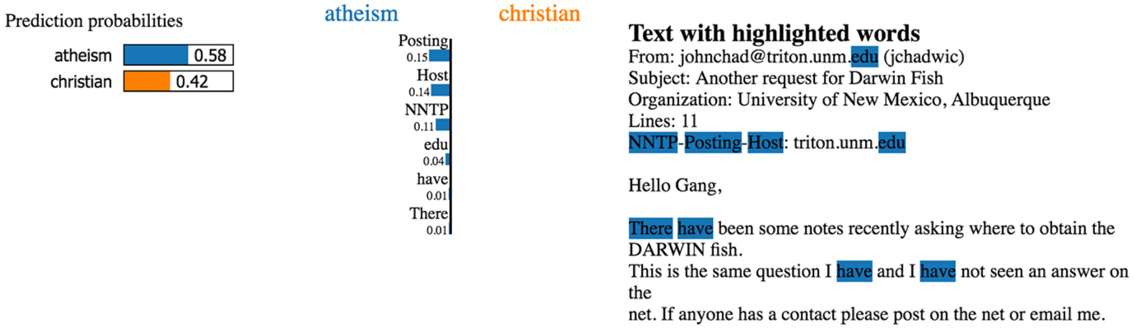
Figure 4.3: In blue and pink it is represented a black box model that we want to explain. The linear model, represented by a line, approximates it in the locality of the instance to be explained, represented by the bold red cross [88].

The interpretable model g learns the local behavior of f near the instance to be explained using as training set perturbed samples. The original instance x is perturbed randomly obtaining a set of perturbed instances. These instances as then labelled using the original model f. In order to consider also the locality, these samples are weighted using the measure π_{x}. The sample instances that are near the instance x will have an higher weight with respect to the far away instances [88].

Ribeiro et al. presented an illustration to describe what is the intuition of LIME [88], figure 4.3. In blue and pink there is the representation of the function f, considered as a black-box by the algorithms. The function f can be very complex, non-linear (as in the picture) and for this reason a simple interpretable model is not able to mimic it globally. The bold red cross represents the instance x that has to be explained. This instance x is then perturbed and the generated instances are then labelled using f and weighted based on their proximity to the original instance. In the picture, this is represented using cross and dots for representing the different class labels and the different size is for the different weight of each instance: the more a perturbed instance is near x, the more it is important. These perturbed instances are then used for training the interpretable model g, in the example a linear one represented through a dashed line. The interpretable model is able to simulate the behavior of the black-box model f only locally and can provide an explanation for the particular instance x.

As already mentioned, the explanation is faithful only locally around x, not globally. In addiction, this solution has problems if also in the locality of x is highly non-linear and in this case a linear or in general an interpretable model may not be able to learn the local behavior [88].

Ribeiro et al. evaluate the utility of the explanations provided by LIME also through user-based surveys [88]. User-based surveys, as already discussed in section 2.3.2, are considered as the best solution for evaluating the interpretability, because of its intrinsic subjective nature. In their experiments, the authors illustrate how the explanations can provide insights on the behavior of the model and on the data and that this can help also to debug the model itself.



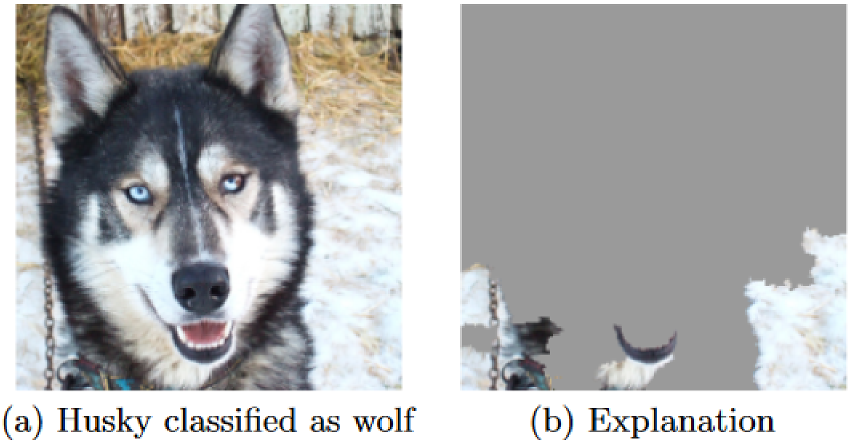
Text with highlighted words
From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11
NNTP-Posting-Host: triton.unm.edu

Hello Gang,

There have been some notes recently asking where to obtain the
DARWIN fish.
This is the same question I have and I have not seen an answer on
the
net. If anyone has a contact please post on the net or email me.

Figure 4.4: [87]

In a first experiment, they performed a text classification using as classifier a Support Vector Machine, SVM, with RBF kernel and as as data set a subset of the twenty newsgroup data set, accessible through the UCI Machine Learning Repository [63]. The classifier had to differentiate the documents that were related to “Christianity” and the ones to “Atheism”. Even if the classifier achieved an accuracy of 94%, through the inspection of the explanations provided by LIME, it was possible to notice that there was something wrong. An example of explanation is presented in figure 4.4. On the right there is the document to be explained, labelled by the classifier as referring to “Atheism” and in blue the words that are most relevant for this prediction are highlighted [87]. The explanation showed that the most relevant words were “Posting”, “Host”, “NNTP” and “edu” that have no connection with “Atheism” or “Christianity” and so it can be argued that they are not good discriminant features for the classification [88]. Once that this irregularity on the model’s behavior was highlighted, researches had the possibility to analyze it and investigate on its origin. They found that the problem was that these words were present in almost all the training data labelled as "Atheism" and so the classifier wrongly learned that these words were important for this class [88]. This is an example of how interpretability is important also because it allows the debugging of the model. The accuracy measure instead is not in general an indicative element of the behavior of the model in real-applications [88]. The explanations give insights on the model internal behavior and in case of problems experts have the opportunity to fix the them. In the example showed, the authors suggest as as solution to clean the dataset in order to remove the features that do not generalize well [88].


(a) Husky classified as wolf

(b) Explanation

Figure 4.5: On the left, an example of image provided during the experiment, representing an husky misclassified as a wolf. On the right, the corresponding explanation provided by LIME [88]. The explanation indicates as determining for the prediction of the image to the class "wolf" the snow in the background.

The authors made another experiment to show that explanations can lead to insights into classifiers. The authors' goal was to demonstrate that, through explanations, it was possible to find unwanted or spurious correlations. For this reason, they intentionally trained a bad classifier that had to differentiate between images with wolves and images with huskies. The authors on purpose trained the classifiers with images such that all the images of wolves presented snow in the background, while for the huskies the snow was not present [88]. During the experiment, they presented at first 10 images with the classifier's prediction. Among them, two were misclassified images: one of a wolf without snow in the background, the other of an husky with the snow. The participants had to indicate if they trust the model or not, why and are was the distinguishing characteristics between the two classes that, in their opinion, the classifier has been learned [88]. Among the 27 participants, 10 trusted the model and 12 indicated the snow as a potential distinguishing feature. In the second part of the experiment, the images were presented with their explanation; an example of them is proposed in figure 4.5. The explanation is built from the original image where only the parts considered determining for the class prediction are present [88]. After examining the explanation, only 3 of them still trusted the model and most of them, 25 of 27, recognized the snow as determining. This experiment shows that through explanations the users can understand why the predictions are made and they can decide if they can trust not only single predictions but also the model as a whole [88].

4.2.2 Feature Contribution

Kononenko et al. propose a solution for explaining a single prediction through the decomposition of it into the contributions of each attribute [90, 95]. Their intuitive idea is to estimate how each feature's value contributes to the prediction by deleting it and then observing if and how the prediction changes. If the prediction changes by deleting a particular feature's value, it means that this value for that feature is important for determining the prediction [90]. The importance of each feature's value is estimated by looking at how the probability of the class value changes if this feature is not taken into account. The more is the change, the more is the contribution of this feature's value.

More formally, the authors define f as a generic classification model that takes as input an instance and returns a numerical value that corresponds to the probability of the class value, $f\!: x \rightarrow f(x)$, [90]. The model can be any classifier because the solution proposed is model-independent. A_i is a generic attribute of the instance x for which we want to provide an explanation. The idea is to observe, for each attribute A_i with value a_k, how the prediction changes if we do not consider it [90]. The term $f(x \backslash A_i)$ is the probability of the class value for x without the knowledge of the value of A_i. In order to compute the importance of the value's a_k of A_i, we can see how the probability of the class value changes and so we have to compare $f(x)$ with $f(x \backslash A_i)$. The influence of each attribute's value is so estimated in the following way [90]:

$$predDiff_i(x) = f(x) \text{ - } f(x \backslash A_i)$$

The greater is the difference, the greater is the contribution of the attribute A_i with value a_k in the determination of the class value. The authors indicate different possibilities to actually compute the "prediction difference".

One possibility is to use directly the difference between probabilities [90]:

$$probDiff_i(x) = p(y \mid x) \text{ - } p(y \mid x \backslash A_i)$$

where y is the target class and $p(y \mid x)$ is the probability of the instance x to belong to the class y.
Another way proposed by the authors is based on the notion of information [93] and called information difference [90]. It is defined as:

$$infDiff_i(x) = log_2\ p(y \mid x) \text{ - } log_2\ p(y \mid x \backslash A_i)$$

Most of classification models returns for each instance to be classified not only the predicted label but also directly the probabilities of the instance to belong to each possible class. The term $p(y \mid x)$ in these cases can so be easily obtained. Other models instead do not provide directly the class membership probabilities; in these cases the $p(y \mid x)$ can be still computed using probability calibration [75, 115].

The computation of $p(y \mid x \backslash A_i)$ is instead more problematic [90]. We have to compute the probability of x to belong to the class y without the knowledge of the value of the attribute A_i. In other words, we have to delete the attribute and recompute the probability. One possibility is to remove the feature from the complete data set, re-train the whole model and classify the instance to be explained x without these feature, obtaining the probability [108]. The issue is that the re-train of the model has to be done for a number of times equals the number of attributes of the dataset. It is very problematic: the number of attributes can be high and the training of all these models can require too much time. In addiction, these new models are different to the one for which we want to provide the explanations and so we can argue that comparing the probabilities in this way is not adequate because the probabilities are obtained from two different models. The other possibility is to substitute the value a_k of the attribute A_i with an *unknown value* or a NA value, not available [108]. In this way, there is no need of re-training the model, adding inaccuracy. The problem is that only few models, as Naive Bayes, support the omission of features using these special values [108]. Consequently, this approach can be considered too limited and not general enough to be applied in a model-agnostic method.

For all these reasons, the authors propose a way for approximating the elimination of a feature value considering all the possible values that this feature can assume, weighted by the prior probability of the value [90]:

$$p(y \mid x \backslash A_i) = \sum_{s=1}^{m_i} p(\, A_i = a_s)\; p(\, y \,|\, x \leftarrow A_i = a_s) =$$

The term $p(\, A_\mathrm{i} = a_\mathrm{s})$ is the prior probability of the value. $p(\, y \,|\, x \leftarrow A_\mathrm{i} = a_\mathrm{s})$ instead is the probability for the class y of the instance x when the attribute $\mathrm{A_i}$ has value $\mathrm{a_s}$ [90]. This replacement is done for all the possible values that the attribute $\mathrm{A_i}$ can assume, $\mathrm{m_i}$ times. In case of continuous attributes, they are firstly discretized.

The prediction difference is computed for all the attributes $\mathrm{A_i}$ and in this way the contribution of each attribute $\mathrm{A_i}$ for the particular explanation of the instance x is obtained.

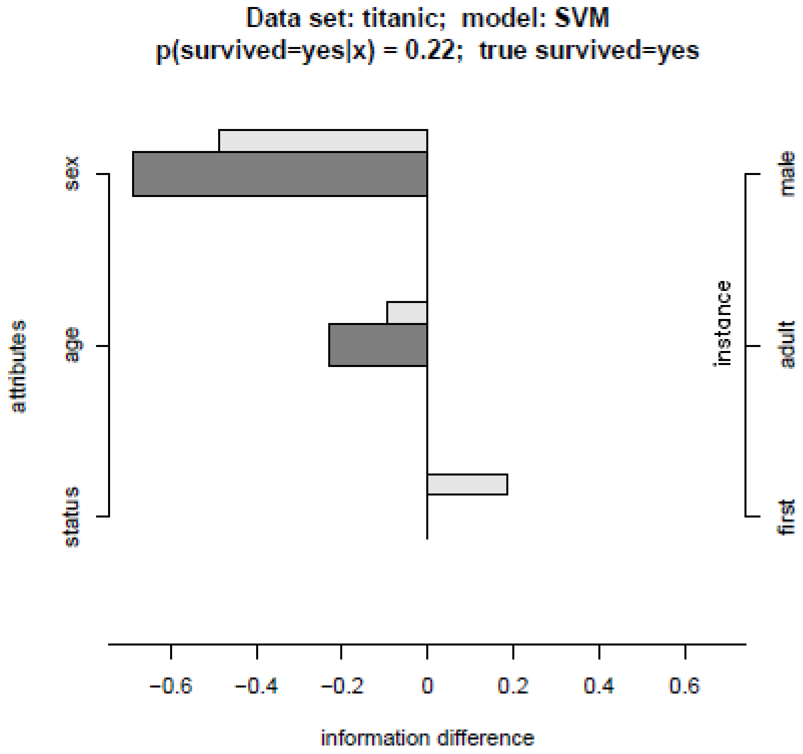


Figure 4.6: Example of explanation proposed by Kononenko et al. for a first class, adult, male passenger in the Titanic dataset, assigned by the SVM model to the class “survived=no” [90]. In dark grey, the contribution of each attribute’s value to the prediction, in light grey the average for all the instances with that particular value.

The authors propose also a visualization method to represent the explanation using a horizontal bar plot [90]. On the vertical axis there is the name of the attribute $\mathrm{A_i}$ and the value it assumes for the instance that has to be explained. On the horizontal axis instead there is, for each attribute, the contribution of that particular attribute and value to the prediction, computed as already described through the prediction difference. An example of explanation is shown in figure 4.6. The dataset used in this example is the "titanic" one, where the information of the passengers and if they survived or not are collected. The classifier has to predict the survival of a passenger, and so the labels are "yes" or "no" and the features are the status, that is the travelling class (first, second, third, crew), the age (adult or child) and sex (male or female). In the example, the explanation is for the prediction for one adult male passenger of the first class provided by a Support Vector Machine model. The SVM predicts a chance of survival of the 22%, p(survived=yes | x)=0.22, for this instance x and so the model assigns x to the class "no". In the figure 4.6, the contribution of each attribute's value is presented and the bar plot is built with respect to the class "yes", that we can call the target class. If an attribute's value contributes positively to the target class, the bar is on the right side, associated with a positive value; if instead the value has a negative influence to the target class, the bar is on the left side, with a negative value [90]. In the example, "sex=male" and "age=adult" have a negative influence, as highlighted by the darker bar. It means that the instance x is labeled with the class "survived=no" because the passenger is adult and male. The thinner bars, in light grey, represent the average contribution for the corresponding attribute value over all the training instances [90]. These contributions express the trend of attributes' values. In the example in figure 4.6, it is possible to note that "sex=male" and "age=adult" have in general a negative impact, while "status=first" has in general a positive one [90].

The approach illustrated allows to easily understand what is the contribution of each feature in the prediction. The graphical representation is simple and can be comprehended also by non machine learning experts. This is particularly important, as already mentioned, because explaining why a particular decision is made by a model can allows users to trust the prediction or not and so they can decide if apply the decision or not [88].

The problem of this solution is that the contributions to the explanations are computed considering only one attribute at the time. If instead it is the change of more of one attribute at time that determines a change also in the prediction, the methodology described is not able to evaluate the influence of attributes' values [90].

As an example, suppose that we have a data set with binary features and the class label is determined by the conjunction of two of them and so the class is equal to A_1 OR A_2 [108]. Suppose also that we want to explain the instance where A_1=1 and A_2=1 and so the class label is 1. In this case, the prediction do not change if we change only one attribute at the time because 1 OR 0 is still 1. The contribution of A_1=1 and A_2=1 is so estimated as null. This is obviously incorrect, their contribution should be not null and equal because they both contributes to the class assignation.

The authors present in their works different solutions to resolve this problem [108]. In particular, they propose a method that extends the computation of the contribution considering also the interaction that each attribute has with the others. The idea is so to change not only one attribute at the time and see how the prediction changes, but also change subsets of attributes and observe their influence [108]. The contribution of each attribute is computed taking into account all its possible interaction with the other attributes. The possible ways to compute the "prediction difference" presented previously consider only one attribute at the time. Kononenko et al. proposed for this reason another way to compute the contribution of each attribute. They define the prediction difference as [108]:

$$\Delta_Q = p(y \mid x_Q) \text{ - } p(y \mid x_\varnothing)$$

Q is a the subspace of the d-dimensional instance space, where only some of the features are considered and x is the instance that we want to explain. $p(y \mid x_\varnothing)$ is the prior probability for the class y, because it is the prediction using no features. $p(y \mid x_Q)$ is instead the predicted probability for the instance x to belong to the class y where we have knowledge of only a subset of the features, all the other features not included in Q are omitted. This term can be evaluated as previously described. The omission of the values not in Q can be computed re-training the model without considering them or substituting them with all the possible values that they can assume, weighted by their frequency. The prediction difference computed in this way takes into account the combined effect of the values of the attributes in Q. The authors also define the *interaction contribution* [108]:

$$I_Q = \Delta_Q - \sum_{W \subset Q} I_W$$

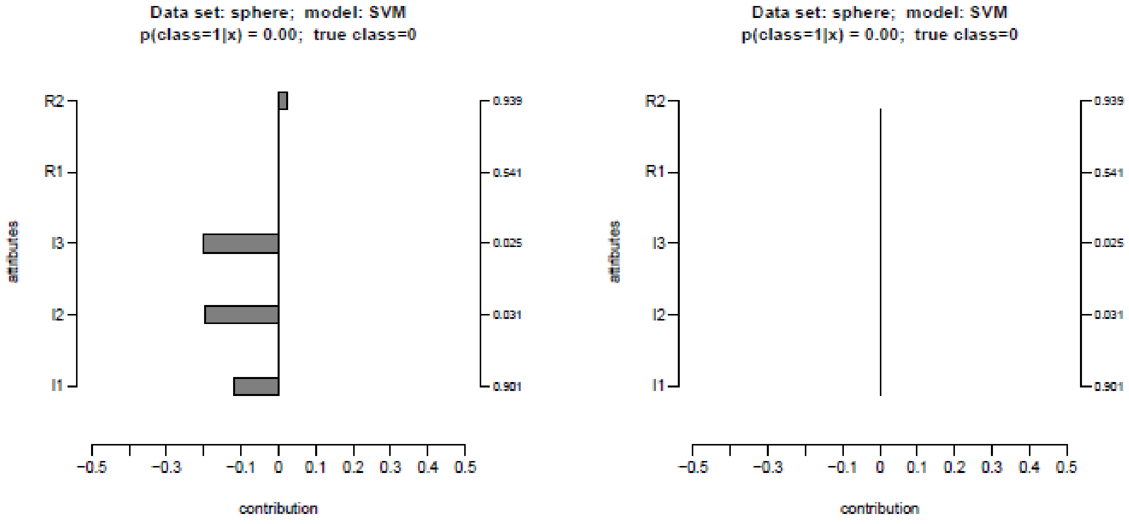
The point is that, in order to estimate the contribution of each subset, we have to not consider also the interactions of the subsets of it, otherwise we overestimate its contribution. Transforming this equation we can note that the prediction difference

Δ_Q is the sum of all the interaction contributions that derive from all the subsets of x. Finally, the contribution of the i-th feature's value is computed in the following way:

$$\pi_i = \sum_{W\subseteq\{1,2,...d\} \wedge i\in W} \frac{I_W}{|W|}$$

The contribution of an attribute's value is the sum of all the interactions in which the attribute is involved. The interaction in the sum is weighted by the number of features that are in the subset considered, W. The contribution π_i is computed for each attribute A_i. The greater is π_i, the more the value assumed by the attribute A_i in the instance that we want to explain is important in determining the class [108]. Using this approach to compute the contribution of each attribute is possible to overcome the problem of estimating the influence when more than one feature jointly determines the class.

In figure 4.7, there is a comparison, illustrated through a particular example, between the approach already described and the ones previously described that is not able to handle disjunctive concepts. The example shows the explanation provided by



(a) IME explanation

(b) EXPLAIN explanation

Figure 4.7: Explanations of a SVM prediction for an instance of the sphere data set. On the right there is the explanation provided by the methods that do not consider features' interactions and in fact it assigns null contribution to each attribute [90]. On the left there is the improved method that is able to handle also disjunctive concepts. It assigns a negative contribution to the relevant features I_1, I_2, I_3 [96, 108].

these two methods for a particular instance of the sphere data set. The sphere data set is an artificial data set designed by the Kononenko et al. to test and validate their explanation methods. The relationship between the attributes and the class label of these data sets are known and so it is possible to more easily confront the explanations provided with the "true explanation" [90]. The sphere data set in particular has five attributes: three are the coordinates in a three-dimensional space while two are instead random values [108]. The data set represents a sphere with radius 0.5 and centered in (0.5, 0.5, 0.5). The class label instead is 1 if the instance is within the sphere, 0 otherwise. In the example, the instance to explain is a point that is outside the sphere and the SVM classifier correctly assigned it to the class 0. On the right of the figure 4.7, there is the explanation provided by the method firstly illustrated: this approach consider only one attribute at the time and it is not able to correctly compute the contribution of the attributes' values. These contributions are in fact estimated as null. On the left instead it is presented the explanation provided by the method that consider also the interactions of more of one attributes. This method assigns a negative contributions to the three relevant attributes. There is also a small contribution of the random value "R1", indicating that the SVM model has capture also some noise [96]. This solution has however a great drawback. In order to compute the influence of features interaction, we have to examine how the prediction changes for each possible subset of the features' values [108]. This means that the power set of feature values has to be computed in order to estimate the contribution π_i of each attribute's value. This computation has an exponential time complexity and this is a great obstacle for its the application. The authors proposed different solutions to overcome this problem. Some are based on feature selection methods: the idea is to reduce the number of features in order to make applicable the computation of attributes' contributions [108]. They also propose an approximation algorithm based on sampling-based approximation [107, 95]. The issue of this solution is that the approximation is related to the data characteristic, as for example the features' variance, and not to the characteristic of the model itself.

Chapter 5

A novel explanation method

In this section, a new methodology to explain single predictions is presented. The solution proposed is model-agnostic and so it is able to explain the predictions of any classifier. This approach was inspired by some of the works described in the previous chapter. In particular, the incentive was to overcome the problem of the exponential time complexity of the solution proposed by the authors Kononenko et al. [108].

In their works, as already described in section 4.2.2, the starting point was to compute the influence that each value attribute, for a particular instance x to explain, has in determining the predicted class value. This is done by deleting one attribute's value at the time and then seeing how the prediction changes [90]. If the prediction changes, the attribute's value is relevant for the determination of the class. The problem is that in this way it is not possible to evaluate the attributes' contributions if it is the change of more of one attributes at the time that affects the prediction value. In order to resolve this issue, the authors proposed another methodology that is able to manage also disjunctive concepts. Their idea is to consider how the prediction changes when more of one attributes is deleted and then all these features interactions are combined to compute for each attribute's value its contribution [108]. The problem of this solution is that, in order to evaluate the features interaction, it has to be computed the power set of features values. It means that if the instance to explain is of dimension d, 2^d subsets have to be considered. The computation of the power set has an exponential time complexity and so the application of this method can be unfeasible in some practical uses where the attributes are many [108]. Kononenko et al. in more recent papers, proposed different solutions to overcome this problem with sampling-based approximation [107]. The approximations are quasi-random and adaptive, based on the characteristic of the data as feature's variance [95].

Our idea is, rather than approximating the features contribution through a sampling based only on data properties, to select only those subsets of feature's values that are relevant for the particular prediction that we want to explain. In this way, the drawback of the exponential time complexity due to the power set computation is overcome because only the subsets considered as determining for the prediction are considered. The relevant subsets for a prediction are estimated looking at the locality of the prediction itself. Inspiring is for us the work of Ribeiro et al., already described in section 4.2.1. They propose a method for explaining the prediction of any classifier, learning an interpretable model only on the locality of the prediction to be explained [88]. The local interpretable model obtained is then directly used to estimate what are the important factors that determine the prediction. The factors can be the features of tabular data, words for a text classification or parts of an image. In our work, the local interpretable model is instead learned in the locality of the prediction that we want to explain in order to obtain what are the subsets that are determining for the prediction. In this way, the problem of the power computation is overcome because we only considered the relevant subsets and only for them we compute their jointly influence to the prediction. The local model reflects how the model behaves in the vicinity of the prediction to be explained. Consider a local zone instead of the entire data in fact makes the method extensible also to Big Data applications where a global model is difficult to obtain. In Big Data problems in fact, data are so sparse, various and high-dimensional that a global model could be not only unfeasible to be comprehended but also learned. A local and interpretable model trained only on the locality of a particular prediction instead is a simpler problem and it allows to underline what are the important features that are relevant for that particular decision. In addiction, understanding why particular decisions are made could also provide insights on the internal working of the global model [88].

In the following section, a detailed description of our method is formally presented.

5.1 Definition of the explanation method

Let be χ a data set with d attributes $\{\mathrm{A}_1, \mathrm{A}_2, ..., \mathrm{A}_\mathrm{d}\}$ and n instances. f is a generic classification model whose predictions we want to explain. f can be any classifier because the explanation method we propose is **model-agnostic**. It means that it treats the model as a black box [88]. A model-independent approach has great advantages, as already discussed in chapter 4. We can explain the prediction of any classifier, without the need of knowing of what are the characteristics of the classification algorithm, how it is implemented or on what it bases the classification decisions. In addition, the explanations can be provided in the same form and this allows an easy comparison of how the decisions are made by different classifiers. The model *f* is able, given an instance $x \in \chi$, to assign it to a specific class $c \in$ C and should provide also class probabilities. The class probability is the probability that the instance x belongs to a particular class. For each class c *in* C, the model has to return *P(y=c | x)*, i.e. the probability that x belongs to the class c. So the model f can be any classifier with the limitation that it has to provide also the class probabilities. Many machine learning methods are naturally able to do it; they are called probabilistic classifiers. This kind of classifier not only outputs the class to which an instance belongs but also the probabilities for the instance to belong to each class $c \in$ C. Examples of naturally probabilistic classifiers are Naive Bayes, Multilayer Perceptron and logistic regression. Others machine learning classifiers, as Support Vector Machine, are not but there are some post-modeling methods that allow to obtain posterior probabilities. These methods are called probability calibration methods and they can be used not only for those models that do not support naturally probability prediction but also for better calibrate the probabilities provided, as the ones of Naive Bayes classifiers that are affected by the independence assumption [75, 109].

The model *f* so, naturally or through probability calibration, is a function f: $\chi \rightarrow$ $[0,1]^{\mathrm{k}}$ that for each instance $x \in \chi$ returns the probabilities to belong to each possible class, where k is the number of classes. In our work, we present the explanation for a single prediction with respect to only one class at the time and so we define *f(x)* as the probability of x to belong to the target class c, *P(y=c | x).*

The data set χ is used for training the model *f.* Given an instance x, our method provides an explanation that shows what are the features that influence the prediction for that specific model. In particular, we can distinguish between an instance that was used for training the model and a new one. It could be interesting not only to explain why a decision is made by the model for an new instance that we want to label but also to provide an explanation of an already learned instance that can show what the model has learned during the training from that particular instance.

Our goal is now to understand what are the subsets of features' values that are relevant for the prediction of the particular instance x. Considering only these subsets instead of the complete power set allows to overcome the problem of the exponential time complexity. For this reason, our idea is to train a local interpretable model that is able to directly output what are the relevant subsets. The interpretable model is trained only on the locality of the prediction of x. It has to mimic the behavior of the model f not globally but only around x.

The training data for learning the local model are generated considering the K instances that are nearest to the instance x that we want to explain. The locality is estimated using a distance metric, as the Euclidean, the Hamming or the Mahalanobis distance. The choice of the parameter K is important because it affects the generated model. It can be difficult to select this parameter a priori because it depends on the data. Some heuristics can be used to estimate K, similar to the ones used for estimate the parameter K of the k-Nearest Neighbors classifier [11]. In particular, for the tuning of the parameter we consider how the local model changes using different values of K. These K neighbors of the instance x are labeled by the model f. Differently from the work of Ribeiro et al., the instances in the locality of the prediction to be explained are not generated perturbing randomly x, but they are instances already learned by the model f because present in the training set [88].

These K neighbors of the instance x are then used for training the local model. The local model has to provide what are the important features' values that are determinant for the prediction. For this reason, the most appropriate classifiers are rule-based ones. This kind of classifiers extract classification rules from the training data and these are then used to classify unlabeled data. Association rules are in the form $X \to Y$, where X is a set of items, and Y, if the rules are used for classification purposes, is a class label [1]. An item is a pair (*attribute, value*) [43] and so these rules represent the association between pairs of (*attribute,value*) and a class label. This is exactly what we are looking for: we want to obtain what are the subsets of features' values that are associated with the class label. In our implementation, as rule-based system we use the classifier L^3, *Live and Let Live*, an associative classifier that is based on a lazy pruning approach [9]. L^3 is so trained with the K neighbors of the instance x that we want to explain. This local model is used not to classify data but only for obtaining the rules and, through them, the set of (*attribute=value*) that are considered relevant for the classification.

Learning a local model for obtaining the important subsets entails approximating the estimation of features' contribution. The point is that we try to capture the relevant associations learned by a generic model f through another classifier. The associative classifier could not be able to mimic the behavior of f well if also in the locality of x it is complex and highly non-linear [88]. However, we can argue that this approximation is anyway acceptable. In the work of Kononenko et al., the exponential time complexity is overcome through sampling-based approximation method [95, 107]. The problem is that the sampling is quasi-random or adaptive and it is based on a greedy approach, considering data characteristics, as the features' variance. In our work instead, the approximation is based on the behavior of the model f itself. We try to capture, through the associative classifier, what are the relevant features' values for f that determine the class label on the locality of x. So the approximation depends on what the model f has learned and not on general data properties.

Thus, through the local model we retrieve the subsets of significant features' values and we refer to these subset with B. We define S as the set of all pairs (*attribute, value*) that the instance x to be explained assumes. Instead of computing the contribution that each possible subsets of S has, we consider only the relevant ones highlighted by the local model. So, we deal with $|B|$ subsets instead of 2^{d}, that is the cardinality of the power set *P(S)*, with $B \subseteq$ *P(S)*. The local model should be able to recognize only the subsets that are determinant for that particular prediction and so should provide $B \subset$ *P(S)* and with $|B| \ll |$*P(S)*$|$. Obviously, this depend on many factors: the data, the number of features, the model *f*, the parameter K and on how the local model is able to mimic f in the locality of the instance x.

Once that the B subsets are provided, we can estimate the features' contribution. In order to compute them, in our work we use the definitions illustrated by the authors Kononenko et al., with few variations [90, 108]. The aim is to explain a single prediction of the model *f*. The explanation, in our approach, has to highlight what is the influence of each feature's value in the determination of the class. The idea of Kononenko et al., taken up in our work, is to estimate the influence changing one or more attributes at the time and then seeing how the prediction changes: the more the probability for a particular class changes, the more the values of the attributes changed are important for the prediction [90]. The effect is observed removing these attributes from the instance and confronting it with the prediction where all the attributes are considered. The authors Kononenko et al. defined this as prediction difference [90]:

$$\mathit{predDiff}_i(x) = f(x) \text{ - } f(x \backslash A_i)$$

where *f(x)* is the prediction for x to belong to a particular target class, when all the attributes' values are considered. This term is compared to *f(x*$\backslash A_i$*)* that is the prediction for x without the knowledge of the value of the attribute A_i. This difference is an indication of the importance of the value of A_i for the prediction for the instance x: a great difference means also a great importance of the value of A_i in the determination of the class [90].

We can extend this definition when also the elimination of more of one attribute's value is considered:

$$predDiff_{i,j,..,k}(x) = f(x) - f(x \backslash A_i,A_j,...,A_k)$$

where $A_i,A_j,...,A_k$ are the attributes whose values have been deleted from the instance x. $A_i,A_j,...,A_k$ are a subset of the set of features $A_1,A_2,...,A_d$ and if we refer to this subset with W the prediction difference can be rewritten as:

$$predDiff_W(x) = f(x) - f(x \backslash W)$$

The prediction difference can be evaluated in different ways [90]. In our work, it is directly computed considering the probabilities of belonging to a particular class y:

$$predDiff_W(x) = probDiff_W(x) = p(y \mid x) - p(y \mid x \backslash W)$$

where $p(y \mid x \backslash W)$ is the probability for the class value y of the instance x without the knowledge of the attributes' values in the subset W.

It is still to be clarified how to omit one or more values' attributes of x, in order to estimate $p(y \mid x \backslash W)$. There are several options, as illustrated by Kononenko et al. in their researches [90, 108].

The first way is to replace the values of the attributes $A_i,A_j,...,A_k$ that we want to omit with special unknown values as *NA*, i.e. not available, *don't care*, *don't know* [90]. The problem of this approach is that only few machine learning methods are able to deal with unknown values, as the Naive Bayes classifier; other methods instead do not handle naturally these special values. In our work, we do not use this approach for that very reason. It is not general but only applicable to few models. The explanation method that we propose has the aim to provide explanations for any classifier, using a model-agnostic approach, and so we do not want to restrict its application to only some specific models.

The second possibility is to re-train the whole model, using as training data the data set χ where the attributes $A_i,A_j,...,A_k$ in the subset W that we want to omit are removed [108]. After the training, we obtain a new model *f'* and we use it to estimate $p(y \mid x \backslash A_i,A_j,...,A_k)$. In this case, the new model already considers only the features that are not in the subset W and so we do not have to deal with special unknown values. This approach has however great disadvantages. Firstly, the model has to be re-trained for a number of times equal to the cardinality of B, that is the number of subsets considered relevant by the local model. The local model allows us to consider only the subsets B, the important ones for the prediction, and not all the entire power set. Even with this optimization, the number of times that the model has to be re-trained can be still too high. In addiction, the model re-training is a problem when many and high-dimensional data are involved, as in case in Big Data applications. For these drawbacks, this approach is not applied in our explanation method. Our goal is to provide a general solution, also applicable to Big Data problems.

The third approach, the one that is used in our work, consists on approximating the elimination of one or more attributes with a sort of weighted average value. This approach was proposed by Kononenko et al. for the omission of only one attribute at the time but it can be extended for more attributes [90]. Their idea was to estimate the elimination of an attribute A_i with value a_s with all the possible values that it can assume and weight the prediction with the prior probability that A_i assume that possible value [90]. The term $p(y \mid x \backslash A_i)$ can be computed in the following way [90]:

$$p(y \mid x \backslash A_i) = \sum_{s=1}^{m_i} p(\, A_i = a_s)\, p(\, y \,|\, x \leftarrow A_i = a_s)$$

$x \leftarrow A_\mathrm{i} = a_\mathrm{s}$ is the instance x where the value of the attribute A_i is replaced with the value a_s. So we estimate the probability for the class y of the instance x when the attribute A_i has value a_i. This is done for all the possible that the attribute A_i can assume, m_i times. These probabilities are weighted by $p(\, A_i = a_s)$ that is the prior probability that A_i assumes the value a_s.

This can be generalized for the omission of more or one attribute at the time. We have to consider the combination of all possible values that the attributes we want to omit can assume and then weight it. In this way, we are able to estimate the prediction difference also for subsets of attributes.

This approach can also be used for continuous attributes. In this case, we have at first to proceed with the discretization of the numerical attributes, splitting them into sub-intervals [90].

Using this approach, we are able to compute the prediction difference as a difference of probabilities when only one attribute is omitted, *probDiff*$_i(x)$, or more, *predDiff*$_{i,j,..,k}(x)$ = *probDiff*$_W(x)$, with A_i,A_j,...,A_k=W. We compute the prediction difference for each attributes and for each subset considered as important by the local model, the set of relevant subsets B.

Our goal is to estimate the contribution of each attribute in the determination of the class. The contribution of the attribute A_i is composed by the single contribution of the attribute, *probDiff$_i$(x)*, but also by all the contributions that derive on how this particular attribute's value interacts with other attributes and this can be derived from all the *probDiff$_W$(x)* where A_i is involved, $A_i \in W$. We have so to estimate how the attributes considered interact. The authors Kononenko et al. refer to this quantity as *interaction contribution* [108]. They define the quantity prediction difference as the sum of all these contributions I_H [108]:

$$predDiff_W = \sum_{H \subseteq W} I_H$$

The prediction difference for the set W is composed by all the interactions not only of the set W itself but also by the interactions of its subsets. The interaction contribution I_W is an estimation of how the features' values in the set W contribute together in the prediction. The authors derive the definition of I_W from the one of $predDiff_W$, recursively:

$$\begin{cases} I_W = predDiff_W - \sum_{H \subset W} I_H \\ I_{\{\}} = 0 \end{cases} \tag{5.1}$$

The idea is that, in the evaluation of the contribution of the interaction of the features' values in W, we don't have to take into account also the interactions of its subsets. If for example W=$\{A_1, A_2\}$, we want to estimate how the attributes A_1, A_2 jointly contribute to the prediction, $I_{\{1,2\}}$, and we do not have to consider how the attribute A_1 and A_2 alone determine the prediction and so we have to subtract the terms I_1 and I_2. The *single interaction contribution*, i.e. the interaction contribution of a single attribute A_i, is indicated with the term I_i and it is equal to the prediction difference *predDiff$_i$* itself because the contribution of an empty set is 0, $I_{\{\}}$ =*0*. The interaction contributions are estimated only for the subsets of the features space considered relevant. The local model trained in the locality of the instance x that we want to explain returns the set of subsets B, the subsets of features' values that are determinants for the prediction. In this way, we have to compute the interaction contributions of only $|B|$ instead of 2^{d}, that is the cardinality of the complete power set of features' values.

Once that the relevant interaction contributions of the subsets in B and all the single interaction contributions are computed, it remains to estimate for each attribute's value what is its contribution to the prediction. The authors Kononenko et al. define this quantity as π_i and it is the contribution of the i-th feature's value [108]. This quantity is computed considering all the interaction contributions in which the i-th feature's value takes part and these are weighted considering the number of features value involved in the interaction. Adapting the definition to our work, we compute π_i as follow:

$$\pi_i = I_i \ + \sum_{W \subseteq B \ \wedge \ i \in W \ \wedge \ |W|>1} \frac{I_W}{|W|}$$

We compute π_i for each attribute's value of the instance x to be explained. This values represents how the feature's value is relevant for determining the class: the greater is π_i, the more the value of the attribute A_i determines the class. These individual attribute contributions can be visualized through a bar plot representation, following the visualization method proposed by Kononenko et al., already presented in section 4.2.2 [90]. Each representation is build for explaining the prediction of a model f for particular instance x, with respect to a particular target class y. In the vertical axis, the attributes and their corresponding value are presented. In the horizontal axis instead there is the corresponding π_i contribution. A positive contribution means that the corresponding attribute's value has a positive influence on the determination of the class. A negative one instead means that it speak against the prediction for that particular target class.

The explanations provided by our method are dependent on the instance x that we want to explain, on the model f that make the prediction and also on the target class [90].

It is *instance dependent* because obviously we have different explanation results for different instances. The explanation has to report what are the feature's values that are relevant for the prediction of the particular instance that we want to explain.

It is *model dependent* because the explanation should reflect what are the important factors for the particular model f that made the prediction and so why that model has made that particular decision. Different models work and learn differently and so their explanations are different too.

Finally, our explanations are *class dependent* because the features' values that are important for a class may be irrelevant for another one, and vice versa. This is particularly true for multi-class problems, while for two class problems the influences are complementary. In this last case, the features' values that have a positive influence with respect to a class have instead a negative one with respect to the other, and vice versa.

5.2 An illustrative example

In this section, an illustrative example of how our explanation method works is presented. The data set used is the MONK's Problems Data Set and in particular the "Monk1" data set [63]. The data set is composed by 6 discrete attributes *a,b,c,d,e,f* and the class label can assume value 1 or 0. The relationship between the attributes and the class value is known: the class is 1 if $a{=}b$ or if $e{=}1$, 0 otherwise.

This data set is particular appropriate to be used not only as an explanatory example but also for checking if the explanations provided for the instances are coherent with the expected results. Since we know the true association between attributes and class, we can compare the explanation provided, related to a particular instance x and predicted by a particular model *f*, with the "true explanation".

As a first example, we train a multilayered feed-foward artificial neural network (ANN) using the *Monk1* data set. Let be x = (a=1, b=1, c=2, d=3, e=1, f=2) the instance that we want to explain. We know that the "true class" is equal to 1 because $a{=}b$ and $e{=}1$, and so both a, b and e in this case are important for the prediction. The ANN correctly predicts the class label as 1 with probability $p(class{=}1|x)$ equal to 0.999. In order to estimate what are the relevant subsets of features' values that are relevant for the ANN we need to train the local model in the locality of the instance x. So we select the K instances that are nearest to x, using in this case the Euclidean distance. As already discussed in the previous section, the parameter K is problematic, because its choice affects the resultant local model and so also the explanation. In this example, after a tuning phase, K is set to 25.

The K neighbors of x are used for training the associative classifier L^3. The local model has to highlight what are the important sets of *attribute=value* that are relevant for the prediction of the instance x. It has to mimic the behavior of the ANN only in the locality of the prediction.

The local model returns the following association rules:

$$\{\text{e=1}\} \rightarrow \text{class=1}$$

$$\{\text{a=1, b=1}\} \longrightarrow \text{class=1}$$

that is that if $e{=}1$ then the instance is assigned to the class 1 or also if a and b are both equal to 1. The local model reflects locally the ANN. These relationships are indeed the ones that, based on our knowledge of the *Monk1*'s problem, should determine the class. Now that the relevant subsets have been determined, we can compute the contribution of each attribute's value to the prediction.

As an example, the prediction difference for the attribute a with value 1 is computed as follow:

$$predDiff_a(x) = p(y \mid x) \text{ - } p(y \mid x\backslash a)$$

where the term $p(y \mid x\backslash a)$ indicate the prediction for x without the knowledge of the attribute a. As described in the previous section, the omission of one or more attributes is estimated through a weighted average. The value a in the data set Monk1 can assume 3 possible values: 1, 2 or 3. So the term $p(y \mid x\backslash a)$ is equal to:

$$p(y \mid x\backslash a) = \sum_{s=1}^{3} p(\, a = a_s)\; p(\, y \,|\, x \leftarrow a = a_s) =$$
$$p(\, a = 1)\; p(\, y \,|\, x \leftarrow a = 1) + p(\, a = 2)\; p(\, y \,|\, x \leftarrow a = 2) + p(\, a = 3)\; p(\, y \,|\, x \leftarrow a = 3)$$

where the $p(\, y \,|\, x \leftarrow a = a_s)$ is the prediction with respect to the class y for the instance x where the value of a is replaced with a_s, i.e. one of the possible values it can assume. For the computation of the contribution of the attribute a we have also to take into account the subsets highlighted by the local model in which a takes part: $\{a{=}1,\ b{=}1\}$. So we've at first to compute the prediction difference of $\{a{=}1,\ b{=}1\}$, considering that also b can assumes only the values 1, 2 and 3:

$$predDiff_{a,b}(x) = p(y \mid x) \text{ - } p(y \mid x\backslash a,b) = p(y \mid x) \text{ - } p(\, a = 1, b = 1)\,\cdot$$
$$\cdot\,(\, y \,|\, x \leftarrow a = 1, b = 1) + p(\, a = 1, b = 2)\; p(\, y \,|\, x \leftarrow a = 1, b = 2) + \ldots. +$$
$$+\, p(\, a = 3, b = 3)\; p(\, y \,|\, x \leftarrow a = 3, b = 3)$$

In this way we estimate how the prediction changes when both a and b are omitted. The interaction contribution of $\{a,\ b\}$ can so be calculated:

$$I_{\mathrm{a,b}} = predDiff_{\mathrm{a,b}} - \sum_{H \subset a,b} I_{\mathrm{H}} = predDiff_{\mathrm{a,b}} - I_{\mathrm{a}} - I_{\mathrm{b}}$$

where the single interaction contributions I_a and I_b are equal to $predDiff_a$ and $predDiff_b$ respectively. The quantity $predDiff_b$ can be computed as previously described, omitting b and considering all the possible values that b can assume and weight it by their prior probability.

Finally, the contribution of the attribute a with value *1* can be computed. This quantity is defined as π_a and it is measured as follow:

$$\pi_a = I_a \;+ \sum_{W \subseteq B \,\wedge\, i \in W \,\wedge\, |W| > 1} \frac{I_W}{|W|} = I_a + \frac{I_{a,b}}{2}$$

It is equal to the single interaction contribution I_a plus all the interaction contributions in which the attributes a takes part, weighted by the number of features involved in the interaction itself. The local model highlights as important only the subsets $\{a{=}1,\ b{=}1\}$ and $\{e{=}1\}$, that is $B = \{\{a, b\}, \{e\}\}$: so in the computation of π_a we have to consider also $I_{a,b}$, divided by 2 because two are the features in the corresponding subset.

The other π_i terms are computed analogously. For π_b, it is considered again $I_{a,b}$ while for the other terms, since the local model does not highlight any interaction for them, only the single contribution interaction is considered. For example, the term π_e is calculated as:

$$\pi_e = I_e \ + \sum_{W \subseteq B \,\wedge\, i \in W \,\wedge\, |W|>1} \frac{I_W}{|W|} = I_e = predDiff_e(x)$$

We obtain a d-dimensional vector of features' contributions, where d is the number of features. For each attribute's value, we have an estimation of its influence in the determination of the class for the particular instance x.

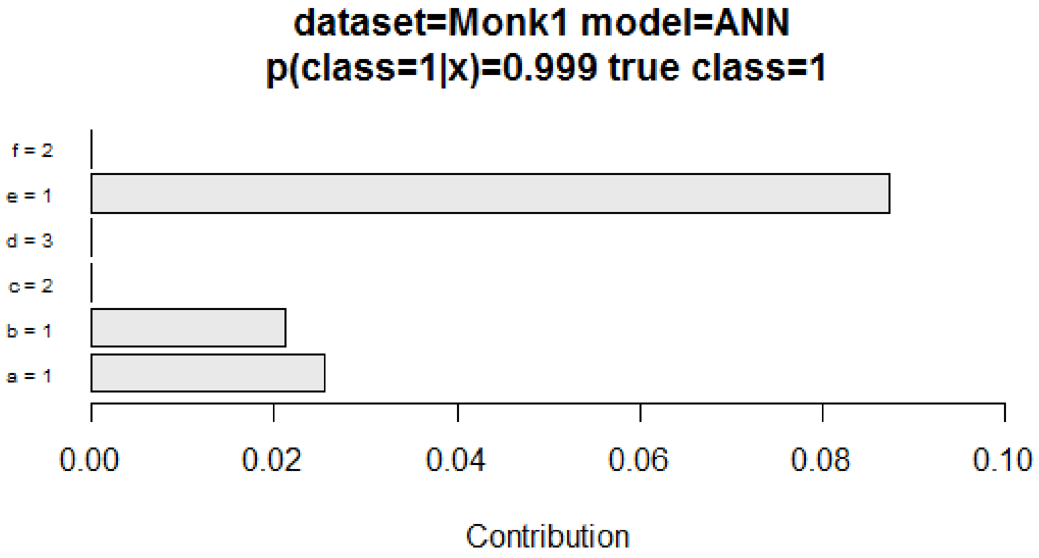


Figure 5.1: Explanation of a prediction of an artificial neural network from the Monk1 data set. This representation indicates as relevant the attribute e equals to 1 and with nearly the same importance a and b both equal to 1.

These contributions are plotted in an horizontal bar plot representation, following the visualization method proposed by the authors Kononenko et al. [90]. The representation is related to the particular prediction x of a particular model *f*, the ANN in this example, with respect to a particular class, the class 1.

In figure 5.1, the contributions are presented. As it is easy to notice, the major contribution to the prediction of x with respect to the class 1 is associated with *e=1*. A smaller and nearly equal importance is referred to *a=1* and *b=1*. Since we know the true relationship between attributes and class values, we can state that this explanation provided follows it. The explanation in fact is able to underline that *e=1* has a positive influence and the same, but lower, for both *a=1* and *b=1*. The other attributes' values instead do not influence the prediction. These last two contributions are not equal, differently than what we should expect. We can ascribe this difference to the different values distribution of a and b in the training data set.

As already discussed in the previous section, this explanation depends on the instance that we want to explain, on the model f and on the target class, that is, the class to which the contributions are calculated. If we explain the prediction for the same instance and still with respect to the class 1 but made by another model we can obtain a different result. The explanation in fact has to capture how the model behaves in the locality of the instance. Different models works differently and so it can be different not only the predicted class label but also what are the determining features' values that drive the prediction. An example of this statement can be provided showing the explanation of the same instance x and still built with respect to the class 1 but classified by the Naive Bayes classifier. In order to estimate the contributions π_i we proceed in the same way, as previously described. We compute firstly the K neighbors of the instance x and these are labelled by the Naive Bayes classifier. These K labeled instances are used as training set for the learning of the local model L^3. L^3 returns as relevant rules:

$$\{\text{e=1}\} \rightarrow \text{class=1}$$

$$\{\text{a=1, c=1, e=2}\} \longrightarrow \text{class=0}$$

The rules are different than the one returned for the artificial neural network. In particular, only the attribute e equal to 1 is considered determinant for the class 1. We compute the contributions and we plot them. The results are shown in figure 5.2. Also the Naive Bayes classifier assigns correctly the instance x to the class 1, but only because *e=1*. The attributes' values of a and b do not contribute positively to the assignation. Actually, they have a negative, but very small, influence.

This explanation highlights that the Naive Bayes classifier has not learned the association that if *a=b* then *class=1*. The explanation method in this case successfully reflects the model behavior. The Naive Bayes classifier is based on the Bayes' Theorem but with the assumption of independence between features [70]. Consequently, it is not able to learn the importance that a and b have together.

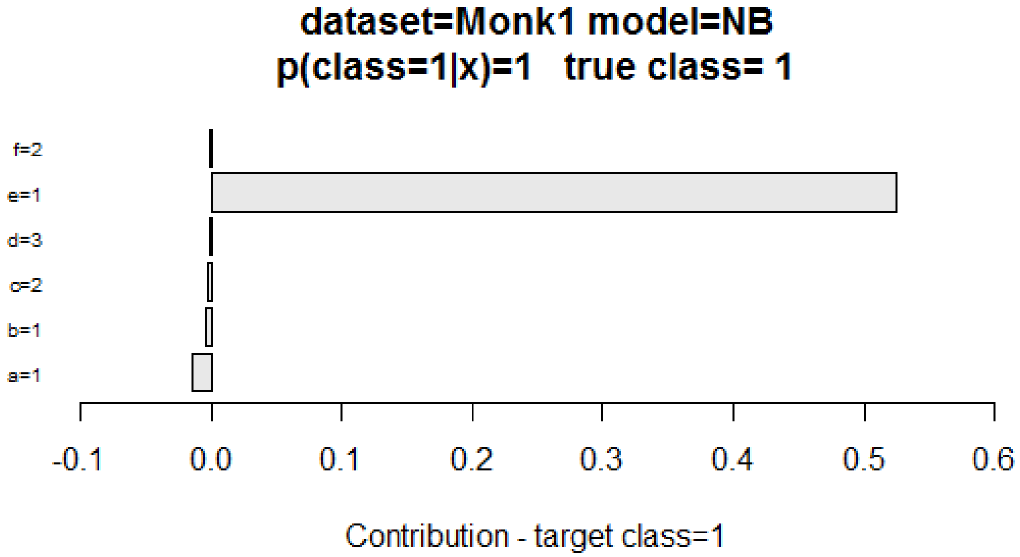


Figure 5.2: Explanation of a prediction from the Monk1 data set of a Naive Bayes classifier. This representation indicate as relevant the attribute e equals to 1.

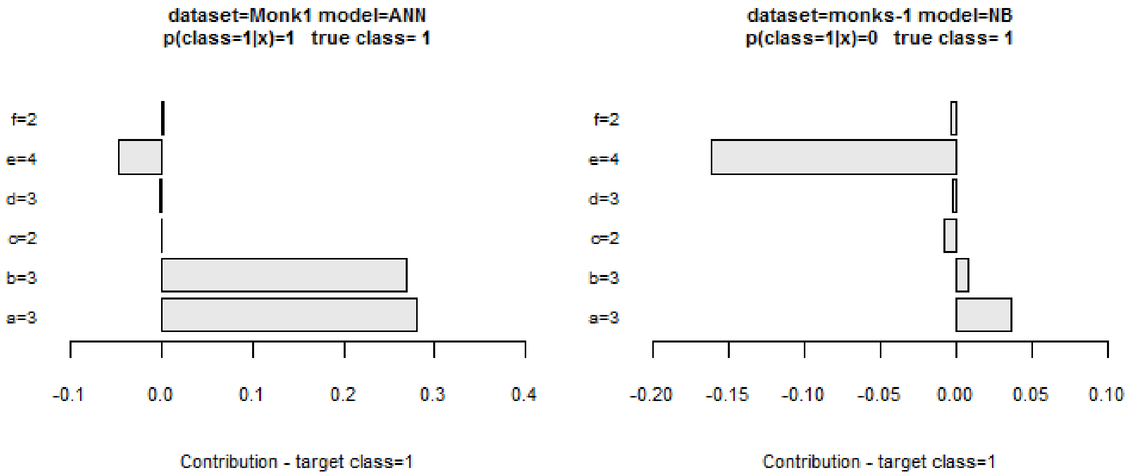
Providing the explanations in an uniform way allows us to directly compare the decisions of two or more classifiers. Each representation reflects why a particular decision has been made by the corresponding classifier. It is so possible to analyze what are the important factors considered as determinants and investigate if these follow what we expect. In this illustrative example, the comparison between the explanations provided and our domain knowledge is elementary because we know the "true explanation", i.e. the true association between features and class. In real application, the true correlations may be still unknown and in some cases machine learning algorithms are applied indeed with the hope to discover them. In these cases, the analysis of the relevant attributes' values has to be supported by experts in the domain knowledge.

Our explanations are model-agnostic and presented in the same way for every classifier. This allows an easy comparison, also for non-machine learning experts.

This is particularly important where machine learning algorithms are applied as a support for decision making, like for medical diagnosis and loan granting.

In addiction, comparing single predictions of two or more different models allows to select which is the best one [88]. The two examples, presented in figure 5.1 and 5.2, already show that the Naive Bayes classifier is not able to capture the true relationship between the features and the class.

Through other explanations of other instances, we can see if this insight is confirmed or not. In figure 5.3, the explanations for another instance are presented. The instance that we want to explain should be assigned to the class 1 because *a=b=3*. The comparison is still between the ANN and NB. In this case, the two models label differently the instance: the artificial neural network assigns correctly the instance to the class 1, while the Naive Bayes classifier misclassifies the instance, assigning it to the class 0. We train two local models, each one on the neighborhood of the instance: the neighbors of the first one are labeled by the ANN and the ones of the second are labeled by the NB model. The first local model highlights as relevant the subset {*a=3, b=3*} for the class 1, while the second returns as relevant only the term *e=4*. Then we compute the contributions and the results are presented in figure 5.3. On



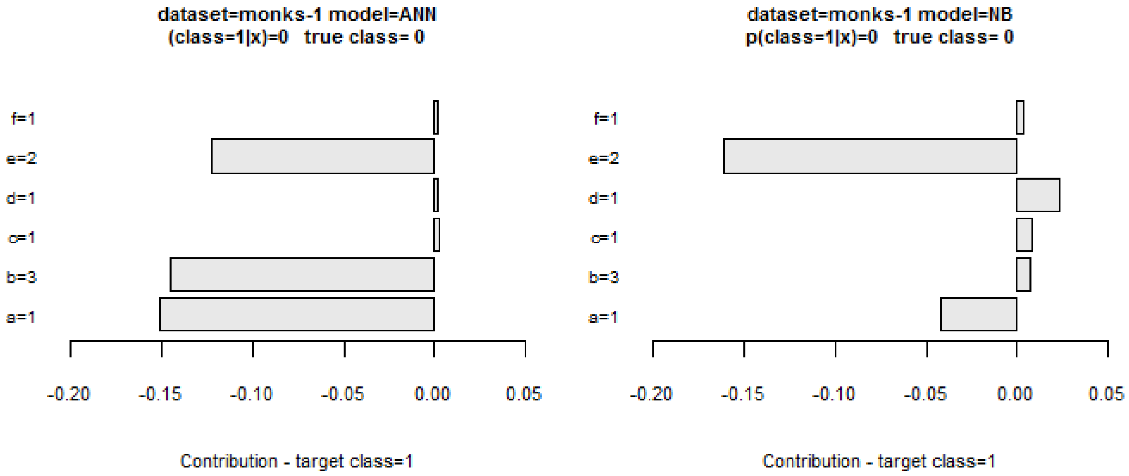
(a) ANN

(b) NB

Figure 5.3: Comparison of the explanations of the same instance for two different classifier, from the Monk1 data set. On the left, figure (a), the ANN correctly classifies the instance to the class 1, because *a=b=3*, while the term *e=4* has a negative influence. On the right, figure (b), the instance is misclassified and assigned to the class 0, because *e=4* has a great negative influence.

the left, there is the explanation of the ANN's prediction. The instance is assigned to the class 1 because *a=3* and *b=3*: these attributes' values have a positive contribution, nearly equal. The term *e=4* instead has a negative contribution; it means that this term is against the class assignment to 1. The explanation reveals that the ANN behavior reflects the one expected. The ANN assigns correctly the instance and because the *a* and *b* are equal. It also captures the negative influence of *e=4*: being different to 1, it is against the assignation to the class 1. On the right instead, there is the explanation for the prediction made by the Naive Bayes classifier. It assigns the instance to the class 0 because *e* is equal to 4. The NB model, as already revealed through the example shown in figure 5.2, is not able to capture the concept of equality, due to the conditionally independent assumption. For this reason, it misclassifies the instance and the prediction to 0 is made only because *e* assumes a value that is different to 1.

For completeness, we now show what are the explanations of these two model for the third possible case: both the two conditions for the assignment to the class 1 are not satisfied, that is *e*!=*1* and *a*!=*b*. The results are shown in figure 5.4, where



(a) ANN

(b) NB

Figure 5.4: Comparison of the explanations of the same instance for two different classifier, from the Monk1 data set. On the left, figure (a), the ANN correctly recognizes that the terms *a=1*, *b=1* and *e=2* have a negative influence with respect the assignation to the class 0 and so it assigns the instance to the class 0 because of these terms. On the right, figure (b), the Naive Bayes classifier assigns the instance to the class 0 principally because *e=2*.

the explanation are presented with respect to the class 1. Both the models assign correctly the instance to the class 0, but, as the figure shows, for different reasons. For the ANN, the terms *e=2*, *a=1* and *b=3* have a negative contributions with respect to the class 1. It means that these attributes' values are against the prediction to 1 and so they lead the assignment to 0. The Naive Bayes model instead classifies the instance to 0 mainly because *e=2*. This example still shows that the Naive Bayes has not learned the association "*a=b* then *1*". Only the term *a=1* has a negative influence and in addiction a positive one is assigned to the term *d=1*.

Chapter 6

Experiments

In this chapter, some experiments are presented with different aims. On the first part, the experiments are set up with the goal of testing our explanation method. The *validation* of explanation methodologies is very problematic: differently than the classifier's validation, we do not know the *true explanation.* For this reason, in the first part we present some artificial data set, already introduced by Robnik-Šikonja and Kononenko to test their explanation method [90]. Being artificial, we know the true relationship between attributes and class and so we can compare it with the one highlighted by our explanation method. In addiction, also if the true associations are known, we have to take into account the model characteristics: classifiers learn differently because they works differently. As already shown in the previous section in figure 5.3, the explanation of an prediction made by a Naive Bayes classifier does not reflect the true explanation. The problem, in this case, is not of the explanation method itself but of the Naive Bayes classifier: this classifier is not able to deal with equality concepts and this is reflected also in the explanation. Therefore, the explanations allow us to have insights of the internal working process of the model. In this way, it is also possible to understand if the model has learned incorrect associations and so decide to not *trust* it. Trusting a prediction is very important when the decision is applied in real-world applications [88]. If for example the machine learning model is used for medical diagnosis, the doctors follows the decision of the model only if they trust it [21]. As Ribeiro et al. noted, representative explanations of single prediction can allow the comprehension of how the model internally works and decide whether to trust the model as a whole or not [88]. In addiction, understanding the problems of a model allows its *debugging.* Once that the odd or wrong associations have been highlighted by the explanations, domain experts can investigate them and they can try to fix the model. In real-world applications, we do not know the true explanation. In these cases, the validation, the decision if the model can be trusted or not and the debugging phase all have to be guided by domain experts.

In the second sections, we apply our explanation method to some data sets available in the UCI machine learning repository [63]. The data sets chosen do not need an extended domain knowledge and so they are suitable to be presented, since their explanations do not need to be supported and validated by domain experts.

The experiments that will be presented in the next sections, both the ones based on the artificial data sets and the ones on the UCI data sets, illustrate a great advantage of our method: the *comparison* between different models. We can show the explanations of the same instance but predicted by different classifiers in an uniform way. This allows a direct comparison of what are the features' values that are relevant for one model and not for another one. Each explanation gives insights of the model internal working and so we can understand what each model has learned. The comparison between explanations by different model is extremely important for choosing which is the best model [88]. Investigating the associations highlighted by the explanation, we can decide which are consistent with the domain knowledge and so chose the model that seems more coherent. In the experiments presented, we will shows that the only classification accuracy is not a good parameter for the choice of the best model.

6.1 Experiments on artificial data sets

In this section, experiments on artificial data sets will be presented. The utilization of artificial data sets for evaluating a explanation methods was firstly proposed by the authors Kononenko et al. [90]. These data sets are on purpose designed for validating the explanations provided. In classification problems, for evaluating the performances of classification models, there are different and well known metrics that can be used. All these metrics are based on the knowledge of the true label. For learning a classifier, we have a set of labeled data: this can be split in training and set data or more sophisticated approaches can be used for training and testing the algorithm, as the cross-validation approach. In both cases, once that the training phase has been completed, we can test the performances of the model simply comparing the model's predictions for new data, that are in the test set, with the true class.

For evaluating the prediction's explanations, we do not have such true results, that are the true explanations. In almost every case, the true association between attribute values and class value is not known, we only know for each instance which is the associated label. Classification algorithms are actually in many cases applied on this purpose: the hope is that classification model is able not only to label correctly new data, but also to highlight the associations between data and label. But as already discussed, only interpretable machine learning algorithms are able to provide why they have made a particular decision and so to indicate associations. In addiction, "correlation does not imply causation" [79]: this mean that the revealed correlations may not indicate the true explanation.

For these reasons, the authors Kononenko et. al propose, in order to evaluate their explanation methods, to use artificial data set [90, 108]. These data sets are on purpose designed for the validation phase: since the relationship between attributes' values and class label is known, it is possible to compare the explanations with the true ones. The validation phase is also tricky in this case. If the explanation provided by a model f for a particular instance do not follow the true explanation, it does not straightforwardly imply that the our explanation method does not work properly. A good instance explanation should reflect why the model f has labeled in that particular way the instance. So we could have a good explanation also if it is not coherent with the "true explanation": the model could have learned the wrong associations or it could be not able to capture the true ones because of its intrinsic limitations. An example of this situation has been presented in the previous section: figures 5.2, 5.3 and 5.4 show that the Naive Bayes classifier is not able to deal with situations where the class label is determined jointly by two attributes, because the Naive Bayes algorithm is based on the independence assumption between attributes.

In the following part of the section, the artificial data sets used are described, already proposed by the author Kononenko et al. [90], and then we will show and compare single explanations provided by different models.

6.1.1 Cross data set

The cross data set is composed of four attributes. The first two attributes X and Y, are the relevant ones that define the class. These values assumes the form of a cross, with two possible class value: *Red* and *Blue*. The class is *Blue* if *(X-0.5)(Y-0.5)>0*, *Red* otherwise, as shown in figure 6.1. In addiction to these two relevant attributes, two random ones, *R1* and *R2* are added. The true explanation assigns equal importance to X and Y, while 0 to *R1* and *R2* because they are unrelated to the class [90].

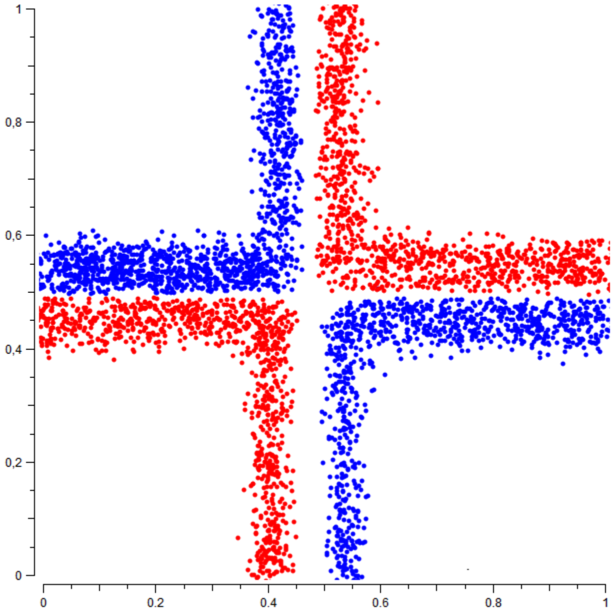


Figure 6.1: Visualization of the two relevant attributes X and Y in the *cross* data set.

The four attributes are continuous with values in the interval [0,1]. In our method, we approximate the omission of one or more attributes for an instance x replacing that or those attributes with all the possible values that they can assume, weighted by the values prior probability. When the attributes are continuous, we have firstly to discretize them. In this case, as also suggested by the authors Kononenko et al. [90], we use the equal-width discretization, setting to 2 the number of intervals, based on our knowledge of the problem. The classification and so also the explanations results strictly depend on the discretization chosen. In this case the choice is easy since we know how the features are distributed. In real-cases, when we do not have this kind of prior knowledge, it is suggested to use a fine grained discretization [90].

The explanations depend also on the features distribution: the replaced values that are used to approximate the elimination of attributes' values are in fact weighted by the value prior probability. We can show this comparing the explanations for two different data sets: one where the attribute value distributions are equal, the other not. We train an artificial neural network and a decision tree classifier on the cross data set where the distributions are equal. The decision tree classifier is interpretable because it generates the classification tree graph that shows how the decisions are made. In this case, it can be interesting comparing the explanation for a prediction made by the tree classifier and the classification tree. In figure 6.2, the explanations for a particular instance are presented. On the left, there is the explanation of the prediction made by the artificial neural network classifier (ANN), on the right the one of the classification tree, trained using as feature selection criteria the information gain. The two local model trained in the locality of the prediction with the K neighbors of the instance, labeled by the ANN and by the tree respectively, both returns as rule:

$$\{D\_X <= 0.5,\ D\_Y > 0.5\} \rightarrow \text{class=Blue}$$

Both the classifiers assign correctly the instance to the class *Blue* because of the values of the features D_X and D_Y, that equally contribute to the prediction, while the contribution of the 2 random value of D_R1 and D_R2 are correctly evaluated as null.

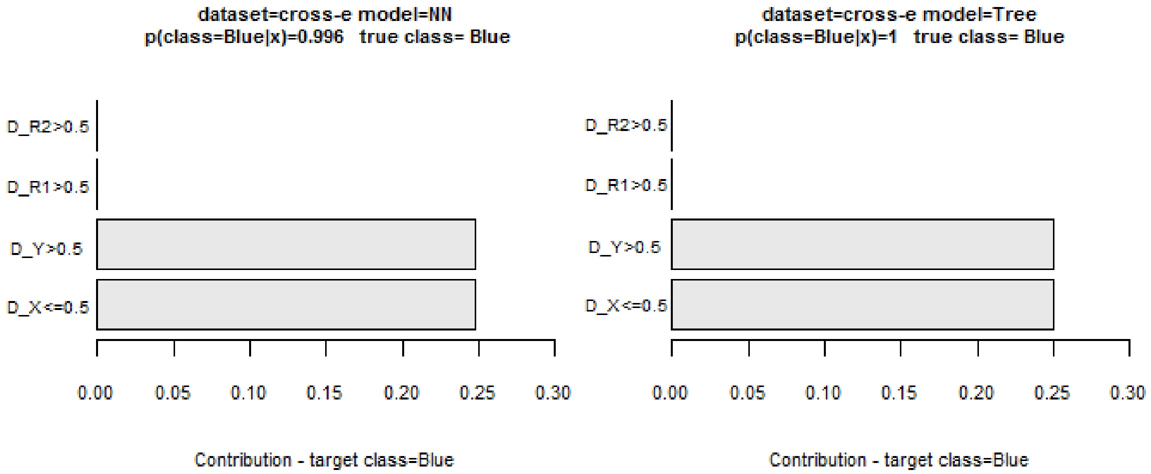


Figure 6.2: Comparison of two explanations for the cross data set - equal distribution. On the left, there is the explanation of the prediction of the artificial neural network, on the right of the classification tree classifier. They both correctly recognize as relevant the features $D\_X <= 0.5$ and $D\_Y > 0.5$.

In figure 6.3 instead, there are the explanation of the same instance but for the classifiers learned using the cross data set with not equal feature value distributions. The instance is still assigned to the class *Blue* because of the features $D\_X <= 0.5$ and $D\_Y > 0.5$, but in this case the contribution of $D\_X <= 0.5$ and $D\_Y > 0.5$ are not equal due to their different frequency in the data set. In addiction, for the artificial neural network there is a negative, even if almost insignificant, contribution of the random feature *R2*. The decision tree classifier instead is not affected by this problem. The explanation of the decision tree's decision in fact correctly reflects the behavior of the decision tree shown in figure 6.4: the class label is assigned considering only the attributes in the nodes of the tree, *D_X* and *D_Y*.

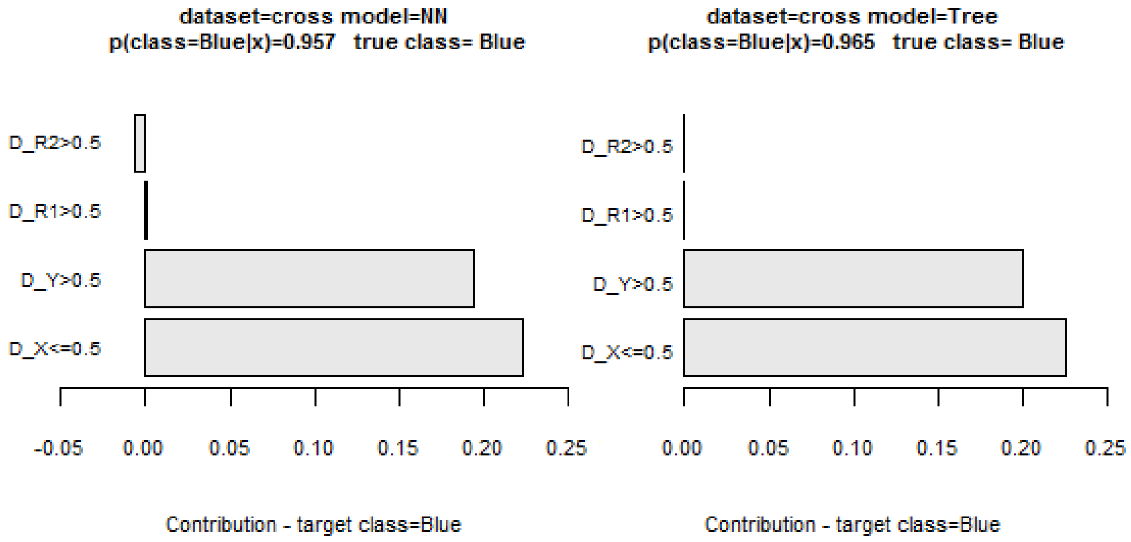


Figure 6.3: Comparison of two explanations for the cross data set with non equal distribution. On the left, there is the explanation of the prediction of the artificial neural network, on the right of the classification tree classifier. They both correctly recognize as relevant the features $D\_X <= 0.5$ and $D\_Y > 0.5$ but with non equal contribution, due to the fact that their values have not the same frequency in the cross data set.

6.1.2 Group data set

The group data set is composed by two relevant attributes, disposed as shown in figure 6.5, that define three different groups and that determine also the class label. The three class values are *Group1*, *Group2* and *Group3*, represented in the picture respectively in blue, red and green. As for the cross data set, we add two random attributes *R1* and *R2*. The true explanation credits equal importance to the two relevant features and zero to the random attributes. We need also in this case to discretize the attributes and we use a discretization with equal width of interval, splitting the values in 3 sub-intervals [90].

In figure 6.6, two explanations for the same instance from the group data set are compared. In this example, we applied our explanation method to the prediction of

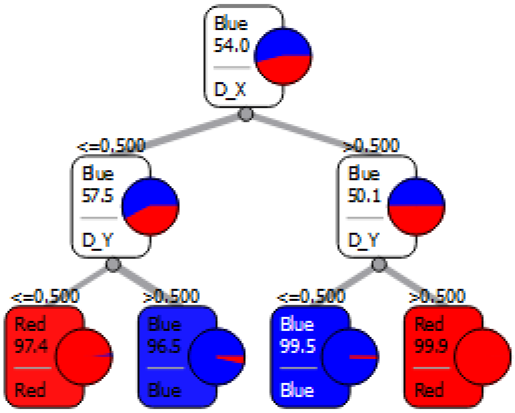


Figure 6.4: Visualization of the classification tree, learned from the cross data set.

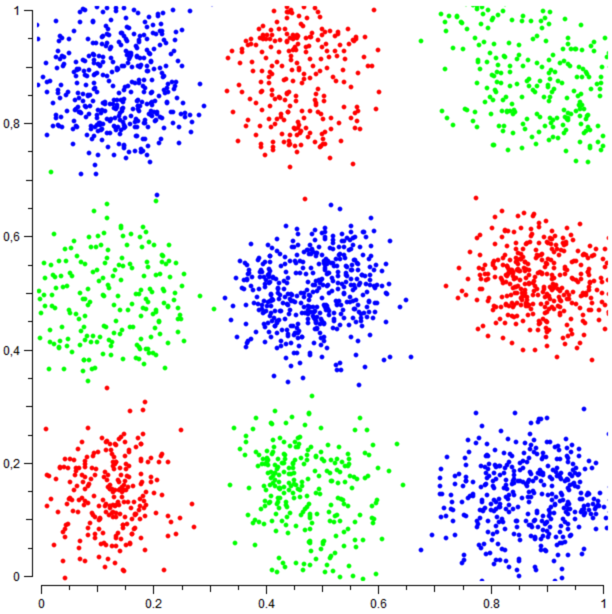


Figure 6.5: Visualization of the two relevant attributes X and Y in the *cross* data set.

the K-Nearest Neighbor classifier (KNN) and of the Random Forest one. Our method is in fact model-agnostic, i.e. a method applicable to explain the predictions of any classifier [88]. The local model trained using as training data set the neighbors of the instance labeled by KNN highlights the rule: $\{D\_X = (0.333{-}0.666], D\_Y > (0.333{-}0.666]\} \rightarrow$ *class=Blue*. The local model that has to mimic the local behavior of the Random Forest model returns instead $\{D\_X = (0.333{-}0.666], D\_Y > (0.333{-}0.666]\}$ $\rightarrow$ *class=Blue* and $\{D\_X = (0.333 - 0.666], D\_R2 < 0.333\} \rightarrow$ *class=Blue*.

Both the two classifier assigns correctly the instance to the class *Blue* but, as illustrate in figure 6.6, for different reasons. The explanation of the KNN's prediction correctly indicates as relevant the attributes' values $D\_X = (0.333 - 0.666]$ and $D\_Y > (0.333 - 0.666]$. The non equal contribution of the two attributes' values is imputable, as already said for the cross data set, to the different frequency of these values. The random forest classifier instead assigns the instance to the class *Group1* also because of the value of the random attribute D_R1 and the contributions of D_X and D_Y are remarkably different. The random forest so correctly classifies the instance but for the wrong reasons.

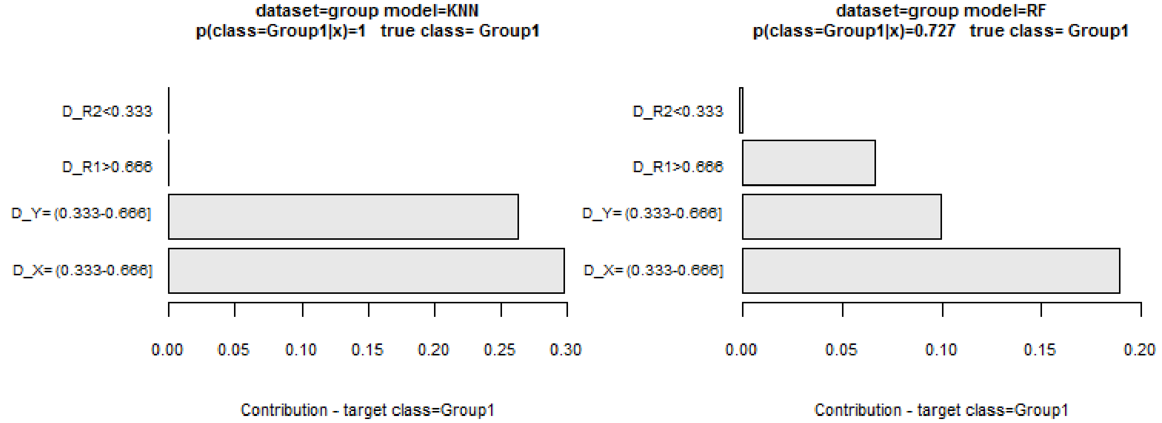


Figure 6.6: Comparison of two explanations for the group data set. On the left, there is the explanation of the prediction of the KNN classifier, on the right of the random forest (RF). The KNN correctly recognizes as relevant the features' values of D_X and D_Y, with slightly the same importance. The random forest classifier instead considers, incorrectly, as important also the value of the random feature D_R1.

6.1.3 Chess data set

The chess data set has two important attributes X and Y that form a 5x5 chessboard, with color *Blue* and *Red* that are the class values, as shown in figure 6.7. Then there the two random attributes *R1* and *R2*, not linked with the class value. The true explanation should assigns equal importance to X and Y and zero to the random values.

In figure 6.8, the comparison of two explanations of the same instance but predicted by two different classifiers is shown. The example shows that the accuracy metrics often cannot be considered as the only indicative metric for the choice of the

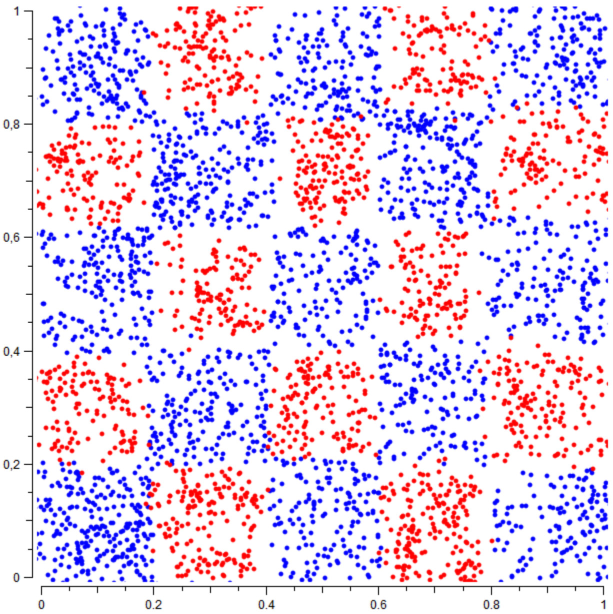


Figure 6.7: Visualization of the two relevant attributes X and Y in the *chess* data set.

better classifier [21, 88]. We trained an artificial neural network and a random forest classifier, using the chess data set. The random forest model (RF) outperforms the neural network. The classification accuracy, (CA), of the random forest, estimated through K-fold cross validation, is 0.90, while the AUC, that it Area Under the Curve is 0.95. For the artificial neural network instead the CA is equal to 0.76, while the AUC is 0.82. The accuracy metrics suggest that the random forest classifier is the best classifier and so that we would chose and apply it. However, if we look at the explanation provided for a particular prediction of the random forest model we can note that it assigns a great negative influence to the random value of the discretized attribute *D_R1*. It means that its value is against the assignation to the class *Blue* and that it lead the prediction to the class *Red*. Even if the RF classifies correctly the instance, it gives importance to the random value, while its contribution should be 0. If instead we look at the explanation for the same instance of the artificial neural network, we can see that it assigns correctly nearly the same contribution to the values of D_X and D_Y and null to the two random values. Since in this toy example we know the true explanation, we can state that the random forest model has learned also wrong associations. In real-word applications, we do not known the true explanation and so the comparison is not so simple. In these cases, the evaluation of the explanations provided has to be guided by domain experts. Explanations of single prediction can help not only in the choice of the best classifier but also for understanding what are the wrong associations and try to fix them [21, 88].

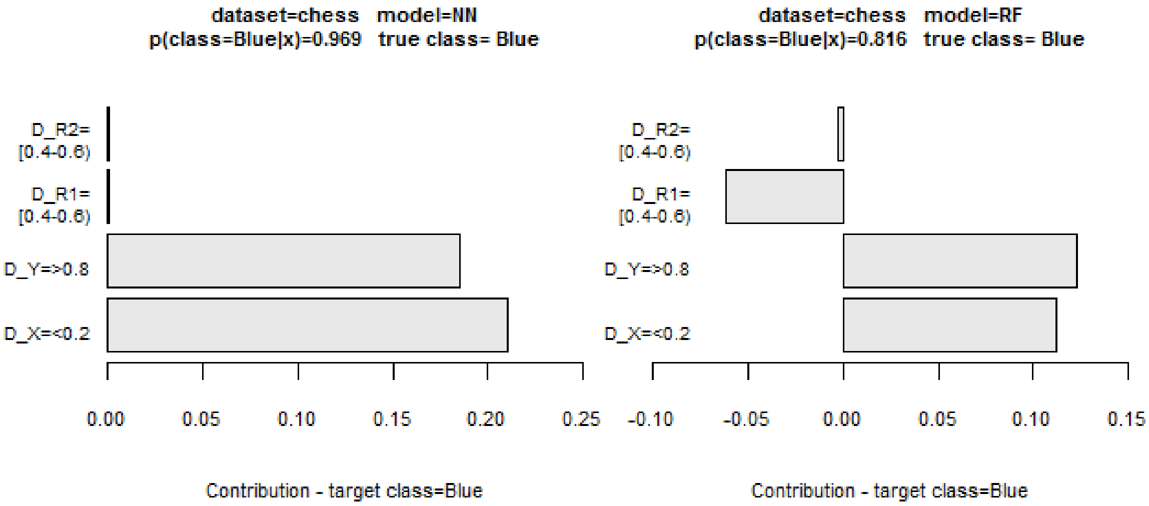


Figure 6.8: Comparison of two explanations for the chess data set. On the left, there is the explanation of the prediction of the artificial neural network (NN), on the right of the random forest (RF). The NN correctly recognizes as relevant the features' values of D_X and D_Y, where the non-equal importance is due to non-equal frequency of the attributes' values. The random forest classifier instead considers, incorrectly, as important also the value of the random feature D_R1, with a negative influence.

6.2 Experiments on data sets from the UCI machine learning repository

In this section, we applied our explanation method to data sets available in the UCI machine learning repository, in particular the *zoo* and the *adult* data sets [63]. We choose to experiments our method and to show the explanations for the prediction of these data sets because they are well-known and there is no need of an advanced domain knowledge for their comprehension. In real-world applications, the validation of the provided explanations is difficult, since we do not have the "true explanation", that is the true relationship between attributes and class values. In these cases, the associations highlighted by the explanations have to be analyzed by domain experts that have to validate them or not based on their prior knowledge. As an example, the authors Kononeko et al. applied their explanation method to a real-life oncology data set [108]. They provided the explanations for predictions of a random forest model to oncologists that had to confirm whether the explanations reflect their medical knowledge or not [108]. Our experiments instead are based on data sets for which non prior experience for their comprehension is required and so they are suitable to be presented as examples to all kind of users. The examples of explanations presented have the aim of underlining the importance of prediction interpretability. As already discussed in section 2.2, different concepts are strictly related to interpretability as trust, debugging and fairness. For *trusting* a prediction, users have firstly to understand why this particular decision is made. Only if they are able to comprehend the relationship between features and class prediction, they can decide whether to trust this prediction or not, based on their prior knowledge [88]. Interpretable prediction explanations allows also the *debugging* of models: explanations can show that a model has learned the wrong associations and experts can try to fix the model in order to improve the classifier [20, 88]. Finally, explanations can reveal discriminatory decisions. The data used for training the classification models are inherently unfair since data are collected from society, that is unequal and discriminatory and consequently also the models [37, 64]. Predictions' explanations can shows if the decisions are based on sensitive data, such as race or ethnic origin, political opinion or sexual orientation [41]. Only if these possible discriminatory factors are known is possible to manage the potential discriminatory effects in order to obtain *fair* classifiers.

Furthermore, the examples proposed highlight the power of our explanation method: the possibility to be applied for explaining the predictions of any classifier. This allows an easy *comparison* between different models and so a more weighted decision of which is the best one, based not only on accuracy metrics but also on what the models have learned.

In the following part of the sections, the UCI data sets chosen are briefly described and then the explanations for some of their instances, predicted by different classifiers, are shown.

6.2.1 Zoo data set

The *Zoo* data set consists on 101 animals from a zoo. There are 16 boolean-value attributes and one meta attribute that is the animal name, unique for each instance [63]. The class attribute, referred with the name "type", can assume 7 possible values: Mammal, Bird, Reptile, Fish, Amphibian, Insect and Invertebrate. The purpose is to classify the animals, based on their characteristics. We use this data set for training different classifiers, as Multilayer Perceptron (NN), Random Forest, Naive Bayes and also classification trees. Classification tree classifiers are known for being interpretable, as already discussed in section 3.1, even if users can have problems in the understanding of classification tree graphs if they are too complex. In this case however, we do not have this issue because the zoo classification problem is very simple and the classification tree graph can be easily comprehended. We provide also explanations of predictions made by the classification tree classifier because they can be easily compared with the true explanation provided by the graph. The explanation of a classification tree prediction in fact should report as important the attributes' values that are in the path from the root to the leaf, that represents a classification decision rule. Comparing the explanation provided by our method and the "true explanation" based on the classification path, we can validate our explanation method, at least when applied for decision tree classifiers.

The experiments presented aim at explaining the prediction of new instance but also at understanding what the model has learned for an instance that was present in the training set. The first target is the common one: given a new instance, not present in the training set, the model f has to classify it and we want also to understand why the model f has made that particular decision. However, it could also be interesting, given an instance of the training set, to understand what the model f has learned from it, what it considers important.

For the first purpose, suppose that we want to obtain the class of a new instance, that represents the characteristic of a zebra. Based on our prior knowledge, we know that the *zebra* instance should be classified as "*mammal*".

In figure 6.9, the explanations for the zebra prediction for the artificial neural network (NN) and the random forest classifier (RF) are presented. The explanation are computed with respect to the target class "*mammal*". Both the models assigns correctly the instance to the class "*mammal*". The two classifier both consider important that the animal produces milk, is haired and do not lay eggs. In particular the random forest classifier assigns lot importance also to "*toothed*=1" and so that it has a set of teeth.

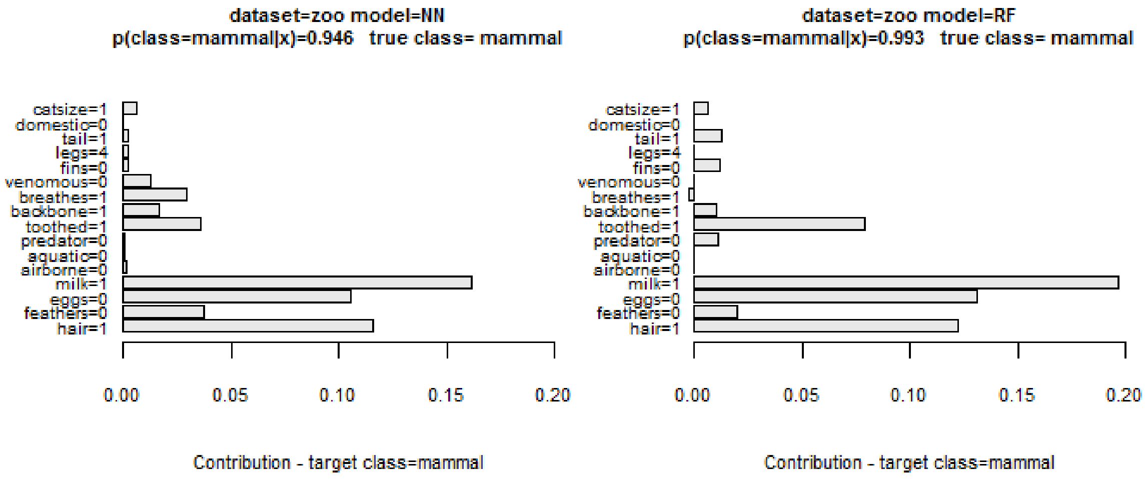


Figure 6.9: Two explanation of the instance *zebra*, not in the training set, from the zoo data set, computed with respect to the class "*mammal*". On the left, the prediction of the artificial neural network decision is explained, on the right the one of the random forest classifier.

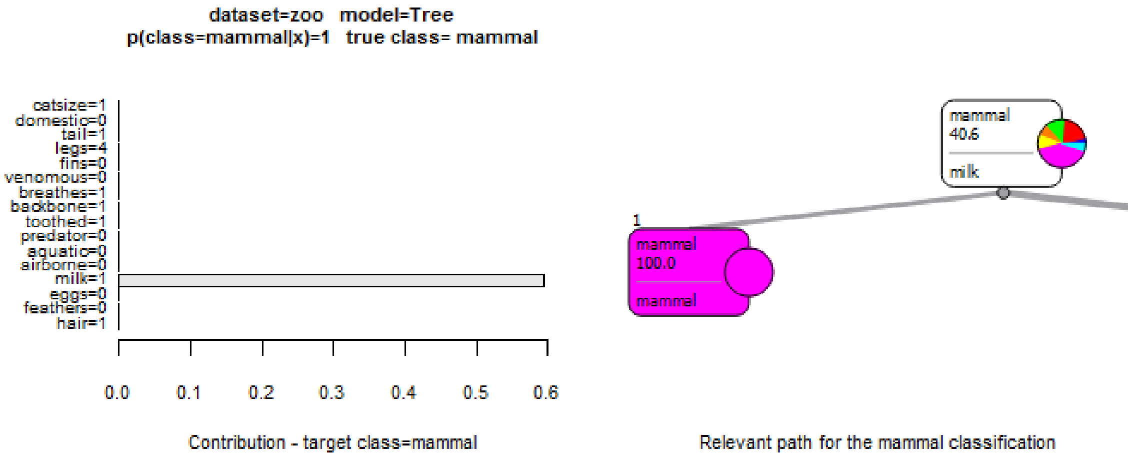


Figure 6.10: On the left, there is the explanation provided by our method for the prediction of the *zebra* instance made by the decision tree classifier. On the right, there is the true explanation, the path of the decision tree that determines the prediction.

In figure 6.10, the comparison between explanation of the classification tree decision for the *zebra* instance and the *true* one is presented. On the left, the explanation assigns as determinant for the prediction only the pair attribute-value *milk=1*. On the right, the relevant part of the classification tree for this particular prediction:

if the attribute assumes value 1, the instance is classified with the class *mammal*. The explanation provided by our method is coherent with the true one, provided by the decision tree itself. Comparing the explanation with our prior knowledge, we can say that also the simple decision tree model is able to capture the distinguishing characteristic of mammals, that is that females of all mammal species nourish their young with milk. The other too models, the NN and the RF, capture this too, but they also recognize other characteristics. It has also to be noted that, since the zoo data set is quite limited, being composed of only 101 instances, the construction of adequate local models is limited too. The local model in this case greatly depends on the parameter K for selecting the neighbors of the instance that we want to explain. For the choice of the parameter K we proceed through attempts and heuristics. In particular, with a too small value, the local model, based on the *Live and Let Live* algorithm (*L^3*), is not able to mimic locally the behavior of the model f [9]. If instead we choose a too large value, the local model is not local anymore and it should reflect how the model f works globally. As already mentioned, this is problematic: the interpretable model L^3 should mimic globally the behavior of the generic model f, that could be very complex and sophisticated.

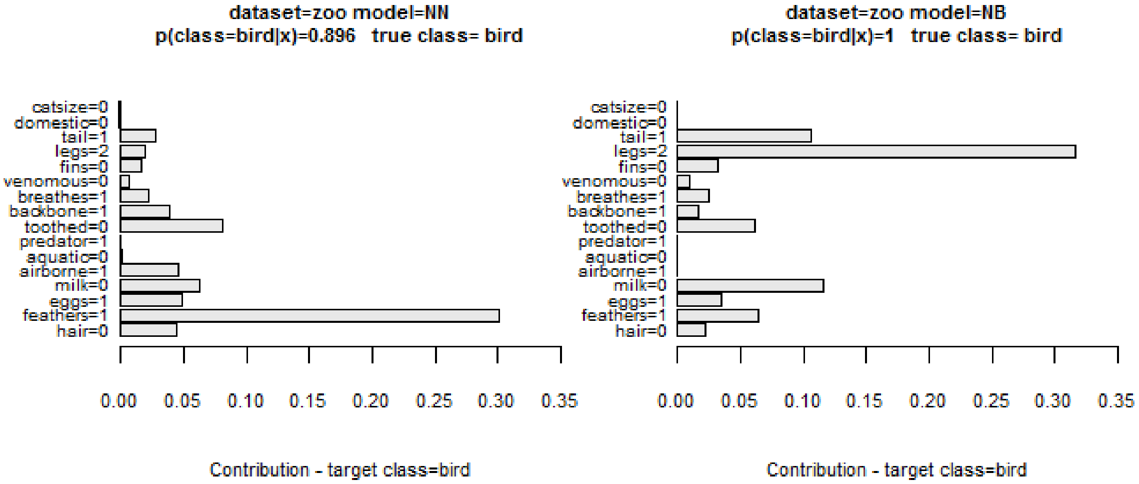


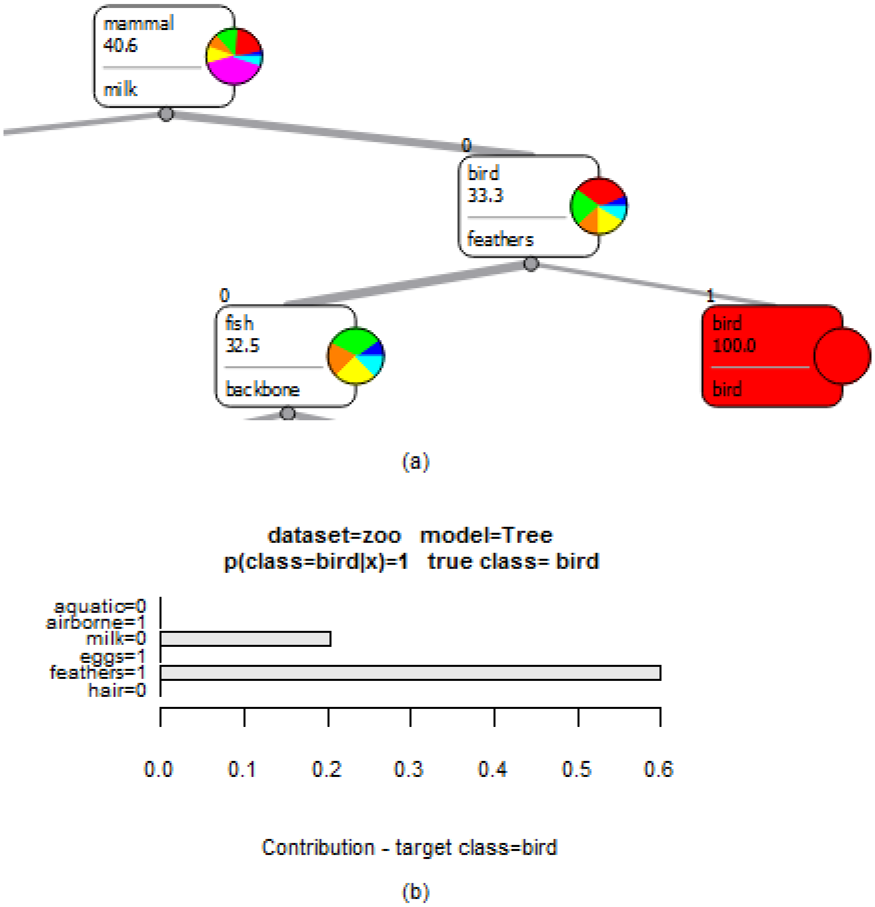
Figure 6.11: Two explanation of the instance *swallow*, not in the training set, from the zoo data set, computed with respect to the class "*bird*". On the left, the prediction of the artificial neural network decision is explained, on the right the one of the naive bayes classifier.

In figure 6.11, we present the explanations provide by the NN and the Naive Bayes classifier (NB) for a new instance, not present in the training set, that describes the characteristic of the *swallow*, that we know it should be classified as a *bird*. On the left, the explanation of the NN decision is presented, computed with respect to the class *bird*. The NN correctly assigns the instance to the class *bird*; determining are especially the term *feathers=1*, that is the fact that it is characterized by feathers but also the term *toothed=0*, that indicates that it is toothless. Also the NB classifiers correctly assigns the instance *swallow* but for different reasons: the great contributions are given by the term *legs=2*, followed by *tail=1* and *milk=0*. Comparing this with our elementary knowledge of animal classification, we can state that these terms do not seem the real determining characteristics of the class *bird*. This example shows how single explanation prediction can help users for selecting the best classifier. The first classifier seems more able to capture the animals' typical characteristics. In addiction, explanations allow the understanding of why a particular prediction is made and so users can decide if trusting it or not. In this example so, we tend to do not trust the Naive Bayes classifier and to prefer the artificial neural network.

Also for the *swallow* instance, we present, in figure 6.12, the comparison between our explanation of the decision tree classifier's prediction and of the true explanation, provided by the decision tree itself. The explanation correctly captures the determining path for the instance, assigning as important only the terms *feathers=1* and *milk=0*.

As already mentioned, it might be interesting to see what are the explanations for instances that where used for training the classifier. In this way, it is possible to understand what the model has learned for those particular instances. The following explanations are provided for the *tortoise* instance, present in training set.

The random forest classifier misclassifies the instance *tortoise* and assigns it to the class *bird* instead of the class *reptile*. In figure 6.13, the explanation of the random forest prediction is presented with respect to the true class *reptile*. Even if the absence of hair (*hair=0*) and the facts that the animal presented has a tail and it is not aquatic have a positive influence for the class *reptile*, since they have a positive contribution, the term *toothed=0* has a great negative influence. It means that this term leads the assignment to another class, not to the reptile one. It is for this reason interesting to show the explanation of this prediction with respect to the *bird* class, the class assigned by the RF, presented in figure 6.14. As it is possible to notice, the instance is assigned to the class *bird* indeed for the term *toothed=0*. The terms *feathers=0* and *airborne=0* instead are against the *bird* class. It can be explain as follows: the presence of feathers and the capability of an animal of flying (*airborne=1*) are considered by the random forest model as characteristics of the class



(a)

(b)

Figure 6.12: Comparison of the explanation provided by our method for the prediction of the *swallow* instance made by the decision tree classifier, figure (a), and the true explanation, the path of the decision tree that determines the prediction, figure (b).

bird. Since in this instance they are not present, they are against the assignment to the class *bird*. This example shows how explanations are important for understanding why classifiers uncorrectly classifies some instance.

The neural network classifier instead correctly classifies the instance *tortoise* and its explanation, with respect to the class *reptile* is shown in figure 6.15. It is interesting to notice that the terms *venomous=0*, *toothed=0* and *predator=0* have a negative influence. It could be explained in this way: the examples of reptile present in the training data set are species of snakes that have exactly the characteristic of being venomous, toothed and predator. The NN has learned these 3 typical features of the reptile class and so, since these are not present in the *tortoise* instance, their values have a negative contribution with respect to the class reptile.

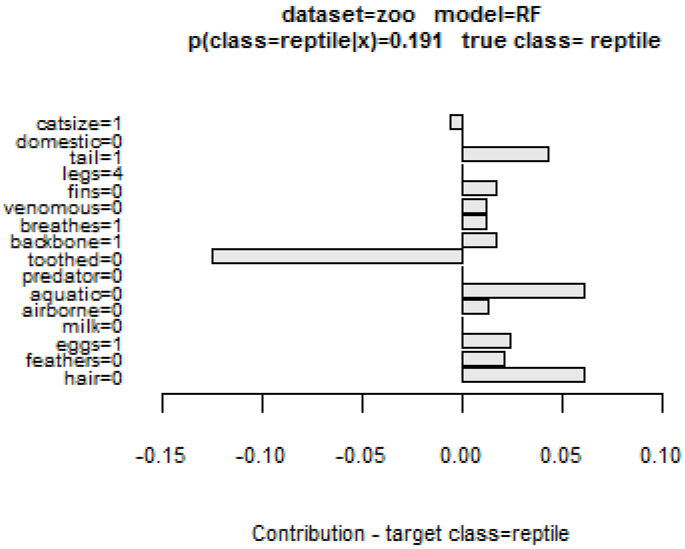


Figure 6.13: Explanation of the prediction for the *tortoise* instance made by the random forest classifier with respect to the reptile class. The RF misclassifies the instance, assigning it to the *bird* class. The instance is not assigned to the class *reptile* because the term *toothed=0* has a great negative influence.

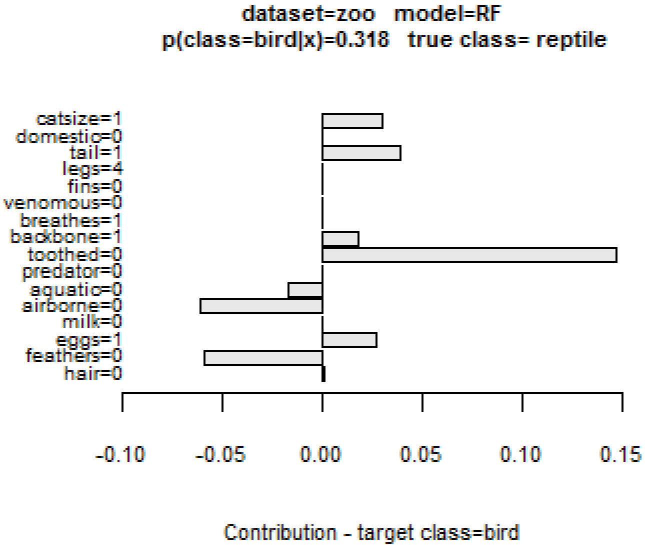


Figure 6.14: Explanation of the prediction for the *tortoise* instance made by the random forest classifier with respect to the bird class. The RF misclassifies the instance, assigning it to the bird class because the term *toothed=0* has a great positive contribution.

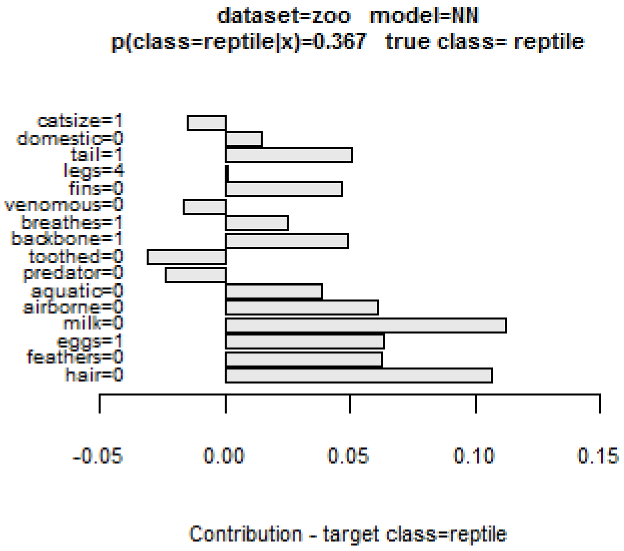


Figure 6.15: Explanation of the prediction for the *tortoise* instance made by the artificial neural network with respect to the *reptile* class.

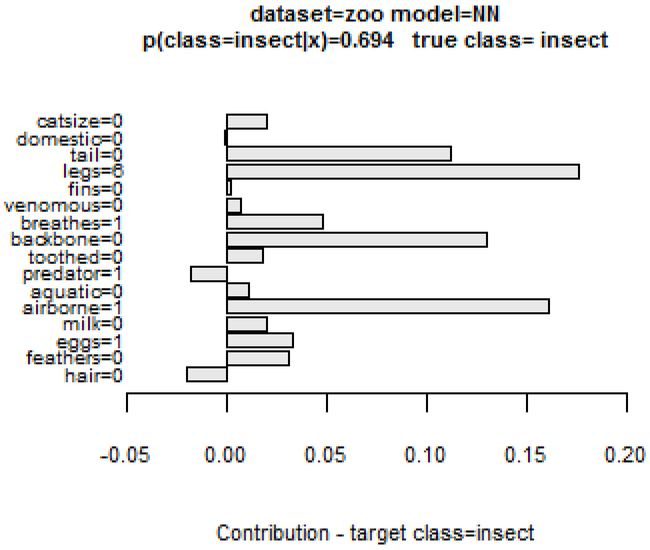


Figure 6.16: Explanation of the prediction for the *ladybird* instance made by the artificial neural network with respect to the *insect* class.

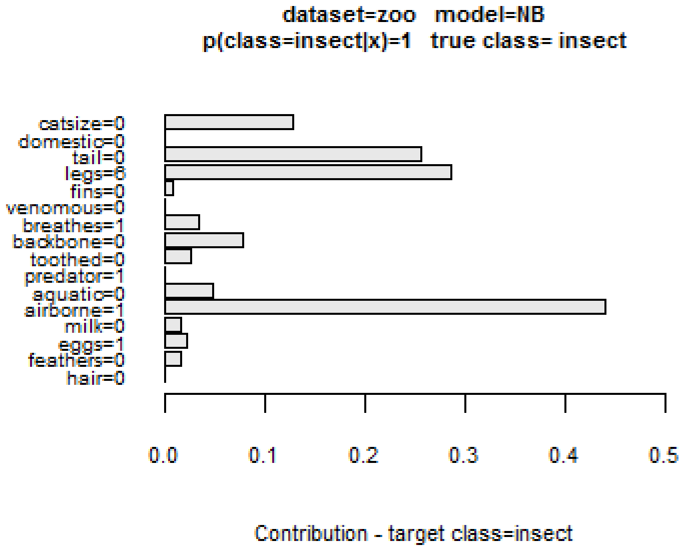


Figure 6.17: Explanation of the prediction for the *ladybird* instance made by the artificial neural network with respect to the *insect* class.

As last example from the zoo data set, we can show the explanation for an instance belonging to the *insect* class, the *ladybird* one. The instance is correctly assigned to the class *insect* both by the NN and by the NB classifiers and their explanation are presented in figure 6.16 and 6.17 respectively. They both consider as terms that contribute the most, positively, to the assignment to the class *insect* the number of legs equal to 6, the absence of tail, their flying capability (*airborne=1*). In addiction, the absence of backbone (*backbone=0*) for artificial neural network prediction is relevant for the *bird* class. The NN considers with a negative influence the term *hair=0* and at first glance it could be seen as a wrong association. Inspecting the training data set is instead possible to justify it. In the data set in fact, many of the insects present are haired, as the honeybee, the wasp and the moth. The NN has so learned that insects have often this characteristic. Since this is not present in the *ladybird* instance, the explanation assigns a negative contribution to the value *0* of the attribute *hair*.

6.2.2 Adult data set

The adult data set was extracted from the 1994 Census bureau database by Barry Becker and contains individuals annual income from various factors [63]. It is composed by 14 attributes, that describe the personal individual information as the education level, age, gender, occupation, relationship and gain. The class attribute is the annual income and it can assume two possible values: *<=50K* and *>50K*. The aim is to predict if a person makes over 50K a year or not, based on his personal information.

The 14 attributes are both categorical and continuous. In our explanation method, the contributions are computed considering all the possible values that an attribute can assume. For this reason, we first have to discretize them. As already mentioned, the discretization affected not only the classification models based on the discretized data but also the prediction explanation and so we should pay attention on this phase. The data set has also missing values; since not all machine learning algorithms are able to deal with them, we decide to remove the corresponding instances since our data set is large enough, but we could have used other techniques as imputation. We apply our explanation method for explaining the prediction of the following models: Naive Bayes, Random Forest, decision trees, multilayer perceptron. So we proceed as previously described: for explaining a particular prediction x of a generic model *f*, we firstly generate its K neighbors and then we train a local model that has to capture the behavior of f in the locality of x. We use the local model for approximating the computation of the attribute values contributions, since it should returns only the relevant pairs of attribute-value for that particular prediction. Suppose that we want to explain the prediction for a particular instance of the test set, labeled with *<=50K*:
age=56, workclass=Private, fnlwgt=128696, education=11th, education.num=7, marital.status=Married-civ-spouse, occupation=Tech-support, relationship=Wife, race=Black, sex=Female, capital.gain=0, capital.loss=0, hours.per.week=40, native-country=United-States.

In figure 6.18, the explanation of the decision tree classifier for this instance is presented. It correctly assigns this new instance to the class $<= 50K$ and it considers as relevant *occupation=Tech-support*, *relationship=Wife*, and *race=Black*. In this case we can easily validate the explanation provided by our explanation method comparing it with the classification tree. It highlights the following decision rule:

$$\text{if relationship=Wife \&\& occupation=Tech-support \&\& race=Black then}$$
$$\text{income} <= 50K(100.00\%)$$

So we can confirm that our explanation follows the true one.

We do not report the relevant part of the classification tree in a graphical representation because the tree is too large and complex that its representation is not

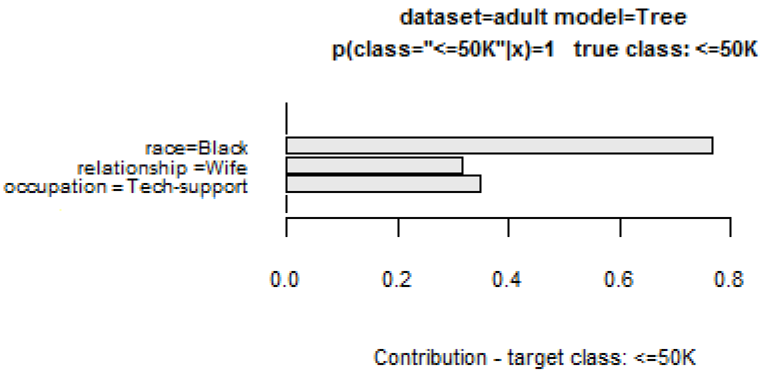


Figure 6.18: Explanation of the prediction for a particular instance from the *adult* data set of the decision tree model with respect to the $<= 50K$ class. Only the features values with non-null contribution are reported for clarity reasons.

suitable to be presented in a single image. As already mentioned in the section 3.1, even if classification trees are considered interpretable, in real applications they could be so large and complex that it could be difficult for us as humans to understand them as a whole [40, 71]. In cases like these, representing why a single prediction is made instead of all the internal working process of the model can allow users to really understanding the problem.

In addiction, this example shows that decisions can be taken also based on sensitive data as race. This sustains the concerns of many researches of the potential negative consequences of applying machine learning algorithms in fields as finance, public health and safety [10, 31, 60]. The problem is that machine learning models can take discriminatory decisions, based on sensitive data as racial or ethnic origin, sexual orientation, religious beliefs [41, 76]. Unfair models can greatly affect people's lives: from the denial of a loan granting, to the news or the job application are shown to them [18, 64]. Models are often inherently discriminatory since they are based on unfair data. Data in fact are collected from the society that is discriminatory, unequal and prejudiced [10, 41]. Only if we understand why a particular decision is made, we can realize if it is based on sensitive data or not and in the first case try to turn an unfair model into a fair one.

In figure 6.19, the explanation of the same instance but predicted by the random forest classifier is shown. The classifier assigns correctly the instance to the class *<=50K* but the greatest contribution is given by the low education level

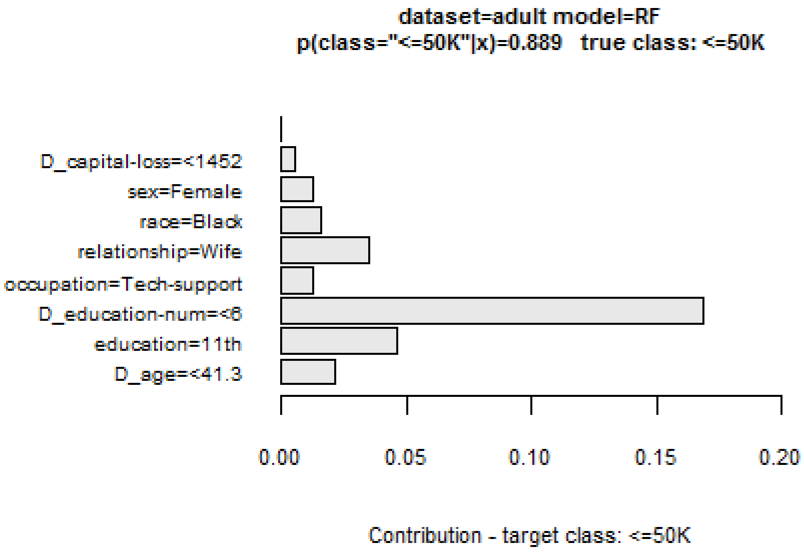


Figure 6.19: Explanation of the prediction for a particular instance from the *adult* data set of the random forest model with respect to the $<= 50K$ class. Only the features values with non-null contribution are reported for clarity reasons.

D_education<6). The terms *relationship=Wife* and *race=Back* are considered relevant but they contribute less to the prediction.

As a new example, from the test data set we chose at random a new instance labeled with $> 50K$. In figure 6.20, the explanation for the random forest prediction is reported and its representation is built with respect to the class $> 50K$. The bars on the right, with positive contribution, influence positively the prediction with respect to the class $> 50K$, while the bars on the left have a negative influence and so lead to the class $<= 50K$. The explanation shows that the instance is assigned to the class $> 50K$ mainly because the high education level, $D\_education > 11$ and *education=master*, and for being a husband, and so male and married. Comparing this explanation with the explanation of the other instance in figure 6.19 we can notice that the level of education, represented by the term *D_education*, is considered by the random forest model as a distinguishing attribute. So, through single explanations we can have insights on the determining attributes for the model as a whole.

In figure 6.21, the explanation of the same instance is presented but predicted by the artificial neural network. The education and the marital status are still considered determinants but the prediction is also greatly positively influenced by the occupation. The young age instead $D\_age < 41.3K$ is instead seen by the NN as a factor that lead to a lower income, $<= 50K$.

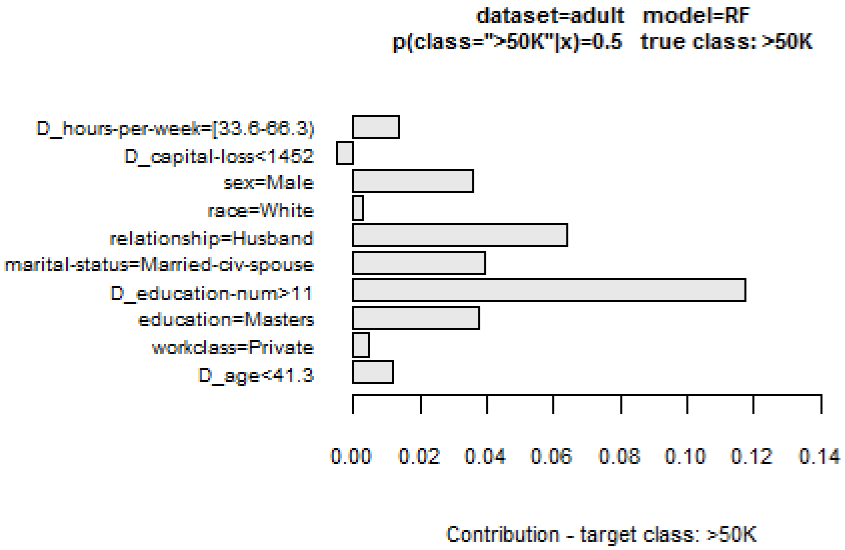


Figure 6.20: Explanation of the prediction for a particular instance from the *adult* data set of the random forest model with respect to the $> 50K$ class. Only the features values with a contribution greater than 0.01 are reported for clarity reasons.

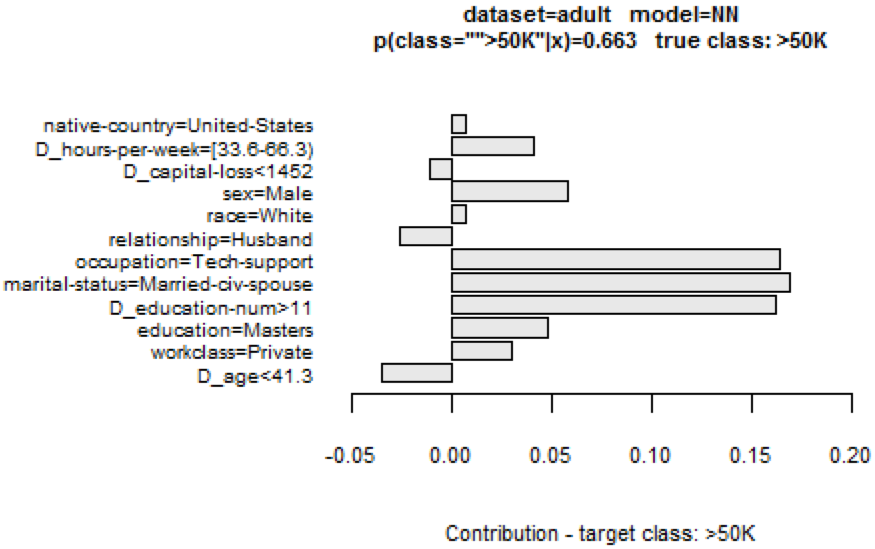


Figure 6.21: Explanation of the prediction for a particular instance from the *adult* data set of the artificial neural network model with respect to the $> 50K$ class. Only the features values with a contribution greater than 0.01 are reported for clarity reasons.

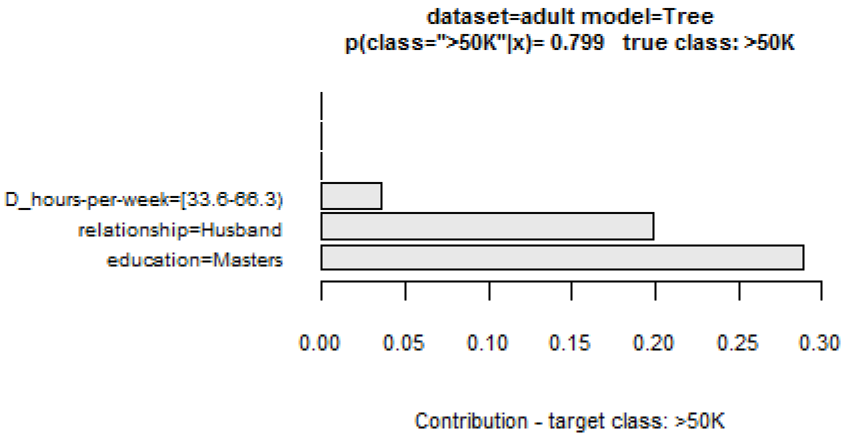


Figure 6.22: Explanation of the prediction for a particular instance from the *adult* data set of the classification tree model with respect to the $> 50K$ class. Only the features values with non-null contribution are reported for clarity reasons.

Finally, the explanation in figure 6.22 shows why the decision tree classifier assigns the instance to the class$<= 50K$. We can validate the explanation provided by our method comparing it with the true explanation, provided by the decision tree itself:

if relationship=Husband && education=Masters &&
D_hours-per-week=[33.667-66.334) then income$> 50K$(79.97%)

Our explanation highlights as important exactly the items that are in this rule and so we can validate it.

All these experiments show that providing explanations of single prediction in a uniform way can allow an easy comparison between different models. The representations' homogeneity is made possible by the characteristic of our explanation method of being model-agnostic, that is, that is applicable for explaining the prediction of any classifier *f*, without having to make any assumptions about f [88]. The restriction is only that the model f has to provide the class membership probabilities but, as already discussed in section 5.1, if f is not naturally a probabilistic classifiers we can use probability calibration methods in order to obtain them. Through an explanation, we can understand not what the model has learned a particular instance. Through significant explanations instead we can try to investigate what the model has learned and so they can give insights on the model behavior [88]. Comparing the explanations from different classifiers can help experts in the choice of the best classifier, because they can choose based not only on accuracy metrics but also on the understanding of how the models work.

Chapter 7

Conclusion

Machine learning algorithms are nowadays applied in every field, such as medicine, finance and marketing. Understanding why a machine learning model has made a particular prediction is becoming increasingly urgent since it could greatly affect people's lives. Especially for high risk tasks, interpretability is considered at least as important as the accuracy metrics [21]. The problem is that most of the machine learning models adopted are accurate but hardly interpretable. Moreover, in case of critical applications, less performing but more comprehensible models are often preferred [21]. For these reasons, many algorithms have been developed for improving the interpretability of already existing hardly-interpretable models. Some of these algorithms have been analyzed in this thesis.

A novel explanation method for explaining individual predictions of any classifier has been proposed. The solution is model agnostic, so it is independent of the type of classifier. The idea is to observe how the prediction changes if one or more attributes are changed at the time. The greater is the change, the more important these attributes are in determining the class. The work of Kononenko et al. has been the inspiring source for the presented method [90, 108]. Their solution is affected by the exponential time complexity due to the computation of the power set for the evaluation of features' contributions. Our method overcome this problem, learning a local model on the locality of the instance that we want to explain. The local model highlights only the attributes' values that are relevant for that particular prediction. In this way, only these important subsets of features' values are considered, instead of the complete power set. This allows us to overcome the exponential time complexity.

Our solutions, being model agnostic, has great advantages. Accuracy and interpretability represents a trade-off [40]: often, the greater is the accuracy, the lower the model is understandable. In real-world applications, the choice of which is the preferred metric could be difficult. Users need at the same time high performing models and to understand why particular decisions are made. Our explanation method allows to consider the model's accuracy and its interpretability distinctly. Being model-agnostic, it treats the model without making any assumptions about it and it provides what are the features' values that influence the predictions most. In this way, it is possible to comprehend the local behavior of the model beyond its intrinsic interpretability. Accuracy and interpretability can no longer be considered as a trade-off. This allows also a more weighted choice of which is the best classifier to apply. The explanations of single predictions of different classifiers can be investigated by domain experts. They can validate them or not by comparing them with their prior domain knowledge. For those models whose explanations are consistent, it can be selected the type of model with highest accuracy [14]. Finally, the proposed model-agnostic approach allows also an easy comparison between models. The explanations of single predictions provided by our method are in fact presented in a uniform way for any classification models. In this way, the predictions of the same instance, that are made by different classifiers, can be explained and compared. Each explanation highlights what are the important factors that determine that particular prediction, for that particular classifier. It is then possible to inspect what the different models have learned, comparing the explanations.

7.1 Future Work

The proposed explanation method is based on studying the local properties of a generic model with the aim of explaining its predictions. The local model is particularly effective and it is able to properly capture the behavior of the global model f in the locality of the instance to explain when the model is particularly complex and heterogeneous. As already mentioned, real-world data sets are not adequate to be used for introductive experiments, as the ones presented in the previous chapter. The point is that for comprehending the explanations of real-world data sets it is required also the presence of domain experts, that are able to validate or not the explanations. In our future work, our explanation method will be applied also for real world applications, with the assistance of domain experts. In addiction, using a local model for capturing the local behavior allows to explain the predictions also of Big Data models. In Big Data applications, data are so complex and heterogeneous that a global model could be difficult to obtain. Our intention is so to experiment our method also in a Big Data context.

List of Figures

Bibliography

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[2] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.

[3] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[4] Hiva Allahyari and Niklas Lavesson. User-oriented assessment of classification model understandability. In *11th scandinavian conference on Artificial intelligence*. IOS Press, 2011.

[5] R. Ambrosino, B. G. Buchanan, G. F Cooper, and M. J. Fine. The use of misclassification costs to learn rule-based decision support models for cost-effective hospital admission strategies. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 304308, 1995.

[6] Robert Andrews, Joachim Diederich, and Alan B Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373–389, 1995.

[7] I. Askira-Gelman. Knowledge discovery: comprehensibility of the results. In *Proceedings of the Thirty-First Hawaii International Conference on System Sciences*, volume 5, pages 247–255 vol.5. IEEE, Jan 1998.

[8] Andreas Backhaus and Udo Seiffert. Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size. *Neurocomputing*, 131:15–22, 2014.

[9] Elena Baralis, Silvia Chiusano, and Paolo Garza. A lazy approach to associative classification. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):156–171, 2008.

[10] Solon Barocas and Andrew D. Selbst. Big data's disparate impact. *California Law Review*, 105, 2016.

[11] GEAPA Batista and Diego Furtado Silva. How k-nearest neighbor parameters affect its performance. In *Argentine Symposium on Artificial Intelligence*, pages 1–12. sn, 2009.

[12] Barry Becker, Ron Kohavi, and Dan Sommerfield. Visualizing the simple bayesian classifier, 1997.

[13] Izak Benbasat and Ronald N Taylor. Behavioral aspects of information processing for the design of management information systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 12(4):439–450, 1982.

[14] Adrien Bibal and Benot Frenay. *Interpretability of Machine Learning Models and Representations: an Introduction*, pages 77–82. 2016.

[15] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[16] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[17] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[18] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, Sep 2010.

[19] Doina Caragea, Dianne Cook, and Vasant G. Honavar. Gaining insights into support vector machine pattern classifiers using projection-based tour methods. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 251–256, New York, NY, USA, 2001. ACM.

[20] Rich Caruana, Hooshang Kangarloo, JD Dionisio, Usha Sinha, and David Johnson. Case-based explanation of non-case-based learning methods. In *Proceedings of the AMIA Symposium*, page 212. American Medical Informatics Association, 1999.

[21] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1721–1730, New York, NY, USA, 2015. ACM.

[22] Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, Mar 1989.

[23] William W Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

[24] Robert M Colomb. Representation of propositional expert systems as partial functions. *Artificial Intelligence*, 109(1-2):187–209, 1999.

[25] G. Cooper, V. Abraham, C. Aliferis, J. Aronis, B. Buchanan, R. Caruana, M. Fine, J. Janosky, G. Livingston, T. Mitchell, S. Montik, and P. Spirtes. The use of misclassification costs to learn rule-based decision support models for cost-effective hospital admission strategies. *Journal of Biomedical Informatics*, page 304308, 2005.

[26] G. Cooper, C. Aliferis, R. Ambrosino, J. Aronis, B. Buchanan, R. Caruana, M. Fine, Glymour C., G. Gordon, B. Hanusa, J. Janosky, C. Meek, T. Mitchell, T. Richardson, and P. Spirtes. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9(2):107138, 1997.

[27] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[28] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[29] Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. 24:87–114; discussion 114, 03 2001.

[30] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, pages 24–30, Cambridge, MA, USA, 1995. MIT Press.

[31] Kate Crawford and Jason Schultz. Big data and due process: Toward a framework to redress predictive privacy harms. *BCL Rev.*, 55:93, 2014.

[32] Collins Dictionaries. Collins English Dictionary - Complete & Unabridged 10th Edition. Aug 2016.

[33] Webster's New World College Dictionaries. Websters new world college dictionary, 4th edition. 2010.

[34] Jeffrey S Dwoskin and Ruby B. Lee. Hardware-rooted Trust for Secure Key Management and Transient Trust. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS 2007)*, pages 389–400, Alexandria, VA, October 2007.

[35] Tapio Elomaa. In defense of C4.5: Notes on learning one-level decision trees. *ML-94*, 254:62, 2017.

[36] C. Feng and D. Michie. Machine learning, neural and statistical classification. chapter Machine Learning of Rules and Trees, pages 50–83. Ellis Horwood, Upper Saddle River, NJ, USA, 1994.

[37] Doshi-Velez Finale and Kim Been. Towards a rigorous science of interpretable machine learning. *arXiv*, 2017.

[38] Peter W Foltz and Susan T Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, 1992.

[39] William J. Frawley, Gregory Piatetsky-Shapiro, and Christopher J. Matheus. Knowledge discovery in databases: An overview. *AI Mag.*, 13(3):57–70, September 1992.

[40] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

[41] B. Goodman and S. Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *ArXiv e-prints*, June 2016.

[42] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[43] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[44] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000.

[45] David J. Hand. Construction and assessment of classification rules. 20(2):326–327, 1997.

[46] David J Hand and Keming Yu. Idiot's bayesnot so stupid after all? *International statistical review*, 69(3):385–398, 2001.

[47] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* . Springer, 2003.

[48] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

[49] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis. Support Syst.*, 51(1):141–154, April 2011.

[50] Aleks Jakulin, Martin Možina, Janez Demšar, Ivan Bratko, and Blaž Zupan. Nomograms for visualizing support vector machines. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 108–117. ACM, 2005.

[51] Holly B. Jimison, L.M. Fagan, R.D. Shachter, and E.H. Shortliffe. Patient-specific explanation in models of chronic disease. *Artificial Intelligence in Medicine*, 4(3):191 – 205, 1992. Therapy Planning and Monitoring.

[52] Y. Kodratoff. The comprehensibility manifesto. *KDD Nuggets*, (94:9), 1994.

[53] Igor Kononenko. Inductive and bayesian learning in medical diagnosis. *Applied Artificial Intelligence an International Journal*, 7(4):317–337, 1993.

[54] Igor Kononenko and Matjaz Kukar. *Machine Learning and Data Mining*. Elsevier, 2007.

[55] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.

[56] Josua Krause, Adam Perer, and Kenney Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5686–5697, New York, NY, USA, 2016. ACM.

[57] Christopher Kuner. *The European Commission's Proposed Data Protection Regulation: A Copernican Revolution in European Data Protection Law*, volume 7, pages 1–15. February 2012.

[58] Niklas Lavesson and Paul Davidsson. Evaluating learning algorithms and classifiers. *International Journal of Intelligent Information and Database Systems*, 1(1):37–52, 2007.

[59] Nada Lavra. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*, 16(1):3 – 23, 1999. Data Mining Techniques and Applications in Medicine.

[60] Bruno Lepri, Jacopo Staiano, David Sangokoya, Emmanuel Letouzé, and Nuria Oliver. *The Tyranny of Data? The Bright and Dark Sides of Data-Driven Decision-Making for Social Good*, pages 3–24. Springer International Publishing, Cham, 2017.

[61] Jinyan Li, Guozhu Dong, Kotagiri Ramamohanarao, and Limsoon Wong. Deeps: A new instance-based lazy discovery and classification system. *Machine Learning*, 54(2):99–124, Feb 2004.

[62] Wenmin Li, Jiawei Han, and Jian Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 369–376. IEEE, 2001.

[63] M. Lichmanm. UCI machine learning repository, 2013.

[64] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[65] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, pages 80–86. AAAI Press, 1998.

[66] Tania Lombrozo. The structure and function of explanations. *Trends in Cognitive Sciences*, 10(10):464–470, 2006.

[67] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 150–158, New York, NY, USA, 2012. ACM.

[68] Jacobus Lubsen, J Pool, E Van der Does, et al. A practical device for the application of a diagnostic or prognostic function. *Methods Archive*, 17:127–129, 1978.

[69] Mitja Luštrek, Matjaž Gams, Sanda Martinčić-Ipšić, et al. Comprehensibility of classification trees–survey design validation. In *6th International Conference on Information Technologies and Information Society-ITIS2014*, 2014.

[70] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI, 1998.

[71] George A Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[72] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog*, 20:14, 2015.

[73] Martin Možina, Janez Demšar, Michael Kattan, and Blaž Zupan. *Nomograms for Visualization of Naive Bayesian Classifier*, pages 337–348. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[74] Gholamreza Nakhaeizadeh and Alexander Schnabl. Development of multi-criteria metrics for evaluation of data mining algorithms. In *KDD*, pages 37–42, Newport Beach, CA, USA, 1997.

[75] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 625–632, New York, NY, USA, 2005. ACM.

[76] Council of the European Union and European Parliament. 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). volume L119, 2016.

[77] Conference of the Representatives of the Governments of the Member States. Consolidated versions of the treaty on european union and the treaty on the functioning of the european union. *Official Journal*, 2012.

[78] Devah Pager and Hana Shepherd. The sociology of discrimination: Racial discrimination in employment, housing, credit, and consumer markets. *Annual Review of Sociology*, 34(1):181–209, 2008.

[79] Karl Pearson. The grammar of science. *Nature*, 62(1594), May 1900.

[80] R Piltaver, M Luštrek, M Gams, and S Martinčić-Ipšić. Comprehensibility of classification trees–survey design. In *Proceedings of 17th International multi-conference Information Society*, pages 70–73, 2014.

[81] Francois Poulet. Svm and graphical algorithms: A cooperative approach. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pages 499–502, Washington, DC, USA, 2004. IEEE Computer Society.

[82] Pearl Pu and Li Chen. Trust building with explanation interfaces. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, IUI '06, pages 93–100, New York, NY, USA, 2006. ACM.

[83] J. R. Quinlan and R. M. Cameron-Jones. *FOIL: A midterm report*, pages 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[84] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[85] J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[86] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-Agnostic Interpretability of Machine Learning. *ICML Workshop on Human Interpretability in Machine Learning (WHI).*

[87] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Introduction to local interpretable model-agnostic explanations (LIME). https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime, August 2016.

[88] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM.

[89] G. Richards, V.J. Rayward-Smith, P.H. Sönksen, S. Carey, and C. Weng. Data mining for indicators of early mortality in a database of clinical records. *Artificial Intelligence in Medicine*, 22(3):215 – 231, 2001.

[90] M. Robnik-Šikonja and I. Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, May 2008.

[91] Stefan Rüping. Interpreting classifiers by multiple views. In *Proceedings of the Workshop on Learning with Multiple Views, 22th ICML*, IUI '06, pages 65–71, 2005.

[92] Rudy Setiono and Huan Liu. Understanding neural networks via rule extraction. In *IJCAI*, volume 1, pages 480–485, 1995.

[93] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(4):623–656, Oct 1948.

[94] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[95] E. Štrumbelj and I. Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41(3):647–665, Dec 2014.

[96] Erik Štrumbelj and Igor Kononenko. *Towards a Model Independent Method for Explaining Classification for Individual Instances*, pages 273–282. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[97] Yanmin Sun, Andrew KC Wong, and Yang Wang. An overview of associative classifiers. In *DMIN*, pages 138–143, 2006.

[98] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[99] Geoffrey G Towell and Jude W Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine learning*, 13(1):71–101, 1993.

[100] F-Y Tzeng and K-L Ma. Opening the black box-data driven visualization of neural networks. In *Visualization, 2005. VIS 05. IEEE*, pages 383–390. IEEE, 2005.

[101] B. Ustun and C. Rudin. Methods and Models for Interpretable Linear Classification. *ArXiv e-prints*, May 2014.

[102] Anneleen Van Assche and Hendrik Blockeel. *Seeing the Forest Through the Trees: Learning a Comprehensible Model from an Ensemble*, pages 418–429. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[103] van den Eijkel, Gerard C. *Rule Induction*, pages 195–216. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[104] Jan Vanthienen and Geert Wets. From decision tables to expert system shells. *Data & Knowledge Engineering*, 13(3):265–282, 1994.

[105] Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

[106] W.P. Vogt and R. Burke Johnson. *Dictionary of Statistics & Methodology: A Nontechnical Guide for the Social Sciences.* Sage Publications, 2011.

[107] E. Štrumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, March 2010.

[108] E. Štrumbelj, I. Kononenko, and M. Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data Knowl. Eng.*, 68(10):886–904, October 2009.

[109] Miha Vuk and Tomaz Curk. Roc curve, lift chart and calibration plot. *Metodoloski zvezki*, 3(1):89, 2006.

[110] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.

[111] Jianyong Wang and George Karypis. Harmony: Efficiently mining the best rules for classification. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 205–216. SIAM, 2005.

[112] C. Weihs and U. M. Sondhauss. *Combining Mental Fit and Data Fit for Classification Rule Selection*, pages 188–203. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[113] Dietrich Wettschereck, David W Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. In *Lazy learning*, pages 273–314. Springer, 1997.

[114] Xiaoxin Yin and Jiawei Han. Cpar: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 331–335. SIAM, 2003.

[115] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, volume 1, pages 609–616, 2001.

[116] Markus Zanker and Daniel Ninaus. Knowledgeable explanations for recommender systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 657–660. IEEE, 2010.

[117] Jozef Zurada and K Niki Kunene. Comparisons of the performance of computational intelligence methods for loan granting decisions. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–10. IEEE, 2011.