BACHELOR'S DEGREE FINAL PROJECT

Explainability and Causality in Machine Learning through Shapley values

Presented by:
Alba Carballo Castro

Supervised by:
DR. EMILIO CARRIZOSA PRIEGO

FACULTY OF MATHEMATICS
Statistics and Operational Research Department
Sevilla, June 2022

Abstract

Explainability and causality are becoming increasingly relevant in Machine Learning research. On the one hand, given the growing use of models in decision-making processes, the way in which they make predictions needs to be more thoroughly understood. On the other hand, a rising interest exists in formalising and introducing the causal relationships present in the real world into those same models. This work addresses both aspects through the use of Shapley values, a concept that is at the origin of SHAP, one of the most popular explainability techniques. Different methods for calculating Shapley values to explain predictions are introduced that take into account the dependence and the causal structure of the data. These methods are illustrated and compared through a series of experiments using a database whose causal structure is known. They show that differences can be observed when taking causality into account.

Resumen

La explicabilidad y la causalidad son áreas cada vez más relevantes en la investigación en Aprendizaje Automático. Por un lado, dado el creciente uso de los modelos en los procesos de toma de decisión, es necesario comprender mejor la forma en que realizan las predicciones. Por otro lado, existe un creciente interés por formalizar e introducir en esos mismos modelos las relaciones causales presentes en el mundo real. Este trabajo aborda ambos aspectos mediante el uso de los valores de Shapley, concepto que está en el origen de SHAP, una de las técnicas de explicabilidad más populares. Se exponen diferentes métodos de cálculo de valores de Shapley para explicar las predicciones que tienen en cuenta la dependencia y la estructura causal de los datos. Estos métodos se ilustran y comparan mediante una serie de experimentos que utilizan una base de datos cuya estructura causal se conoce. De ellos se pueden observar que existen diferencias cuando se tiene en cuenta la causalidad.

A Rosita, Javi y Mariloli
para que estéis siempre.

Contents

Chapter 1

Introduction

Since their origin in the mid-20th century, Machine Learning models allowing to make statistical associations and accurate predictions have experienced an unprecedented revolution. Over time, the path of research in this area has been to improve the speed, accuracy, and precision of these models to make them as useful as possible. However, new questions have recently started to be asked about the nature and future Machine Learning research [21].

On the one hand, making Artificial Intelligence interpretable (i.e. understanding why models make the decisions they do), and not only powerful and accurate in its predictions, has become a major concern. Since an increasing number of decision-making processes are starting to rely on AI, there exists a real need to justify the results that are obtained and make them comprehensible for humans [6]. By achieving this goal, we could envision reducing or even eliminating the presence of biases that could lead to discrimination in the real world.

On the other hand, research has also been recently focusing on the problem of formalising causality, that is, being able to formalise cause-effect relationships like finding out the consequences of actively fixing the value of a variable. This is essential, given that it brings the Machine Learning domain even closer to the way the real world works [25].

Although these two issues, interpretability and causality, may appear to be unrelated, the reality is that they have much in common. Models able to work with the causal structure of data will tend to be fairer, as we will be able to correct the biases present in the real world in them [14]. Moreover, the interpretability of the models may benefit from the knowledge of the underlying causal relationships between variables, resulting in better explanations.

In this work, we try to give a brief introduction to these two topics, showing how they can be connected through the use of Shapley values. These were firstly introduced in the context of Game Theory, but were later used as the basis for SHAP, a local interpretability technique that allows us to explain not only single instances but also complete models. Different methods for the computation of Shapley values in Machine Learning will be introduced, some of them taking into account causal relationships between variables.

The structure of this work is as follows: Chapter 2 features a brief theoretical introduction to the topic of explainability in Machine Learning, current formal approaches to the problem of causality and the Machine Learning models used later on. In Chapter 3, we present SHAP as an explainability technique based on the use of Shapley values. Later, we describe different approaches to the calculation of SHAP values that deal with feature dependency and causal structures. Finally, in Chapter 4 we introduce the computational framework for the practical implementation and experiments carried out in Chapter 5. In it, we compare the different methods to calculate SHAP values as well as the differences between models.

Chapter 2

Preliminars

In this chapter, we will introduce some of the theoretical concepts necessary to better understand this work. We will present the notion of explainability in Machine Learning, giving basic definitions and a classification of explanation methods. Then, the problem of causality and some theoretical concepts about graphs and causal models will be introduced. Finally, we will briefly describe some of the models that will be used in the experiments.

2.1 Explainable Machine Learning

Machine Learning models are being increasingly used to support decision-making processes in many different areas, ranging from Law to Medicine. In recent times, due to the interest in improving accuracy and speed of such models, they have become black-boxes incomprehensible for humans. But since humans are the users of these models and their predictions directly affect them, it is essential to have an explanation supporting them. In this context, XAI (eXplainable Artificial Intelligence, also often called Explainable Machine Learning), a set of algorithms aimed at explaining black-box models [21], is currently gaining interest.

The terms *explainability* and *interpretability* are often used interchangeably. However, some authors understand *interpretability* as the capacity to understand why a certain prediction or decision is made while *explainability* accounts for the ability of explaining the internal mechanics of a model in human terms. In this sense, explainable models are interpretable, since understanding the internal mechanics allows for an undestanding of predictions, but not the other way round [8].

The increasing need for explainability is also manifested by the fact that institutions such as the European Union have recently legislated on the use of algorithms in decision-making. In 2016, the EU introduced the General Data Protection Regulation (GDPR). This regulation went into effect in 2018, and it deals with concepts such as the right to non-discrimination, the right to explanation and the ethical design of algorithms. For a more detailed explanation, the reader is referred to [9].

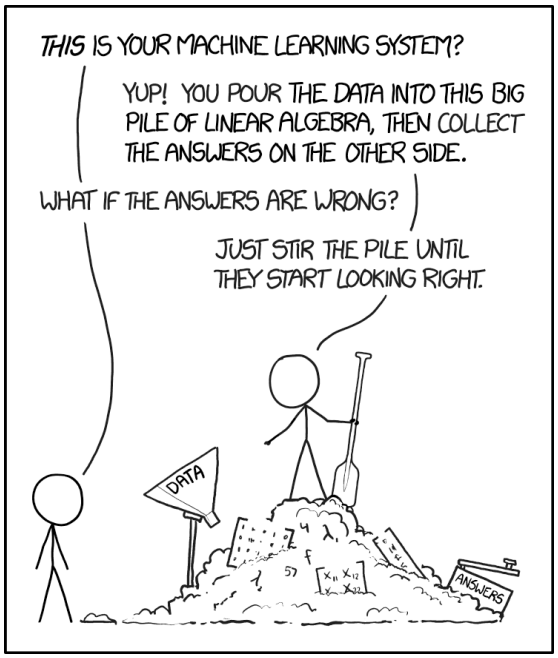

Figure 2.1: Humorous vignette on black-box models
Source: xkcd

The following definitions will be useful in the course of this work.

Definition 2.1.1 (Prediction model)**.** *For a Machine Learning model with n features we can define the* ***prediction model*** *as the function $f : \mathbb{R}^n \to \mathbb{R}$, yielding the value of the prediction in regression models or the probability of an individual belonging to a certain class in binary classification models.*

In binary classification problems, probabilities of belonging to one class or another are complementary and therefore it makes sense that the prediction model is a function in $\mathbb{R}$. In multiclass classification defining the prediction model becomes more complex as there are several probabilities to take into account. For that reason, throughout this work we will deal exclusively with binary classification models.

Definition 2.1.2 (Explanation model)**.** *For a Machine Learning model with n features we can define the* ***explanation model*** *as the function $g : \{0, 1\}^n \to \mathbb{R}$, corresponding to a model that allows to explain the predictions of a particular prediction model f.*

This notion of explanation model was firstly introduced in [17]. It receives as an input a simplified version of an observation x, known as simplified input (or interpretable input [26]) and denoted as x'.

Definition 2.1.3 (Coalition vector)**.** *In a feature model f with a set N of n features, the coalition vector $z' \in \{0,1\}^n$ for a subset $S \subseteq N$ represents whether a given feature is present in the coalition by setting $z'_i = 1$ if the i-th feature is present and $z'_i = 0$ if it is not [20].*

Coalition vectors are a type of simplified input. The correspondence between the original space of our prediction model and the simplified space of the explanation model can be made by means of the mapping function.

Definition 2.1.4 (Mapping function)**.** *Let f be a prediction model with n features and be x one of them. If x' is the simplified input corresponding to x, the mapping function $h_x : \{0,1\}^n \to \mathbb{R}^n$ is the one that transforms the simplified space of coalition vectors into the original space of features, making $x = h_x(x')$.*

In general, local methods try to guarantee that $h_x(z') \approx x$ when $z' \approx x'$ [17].

Definition 2.1.5 (Additive feature attribution methods)**.** *A method to explain a prediction model f with n features is said to be an additive feature attribution method if the explanation model g is a linear combination of binary variables*

$$g(z') = \phi_0 + \sum_{i=1}^{n} \phi_i z'_i \tag{2.1.1}$$

with $z' \in \{0,1\}^n$ and $\phi_i \in \mathbb{R},\ i = 0, \ldots, n$.

Finally, explanation methods can be classified as follows [5]:

1. According to *when* the explanation is made:
   - **Intrinsic**: explanations are produced alongside the predictions, since the method is built into the model.
   - **Post-hoc**: explanations are produced after the model is trained.
2. According to *what* is explained:
   - **Local**: individual, specific predictions are explained by measuring the contribution of each feature in a given model.
   - **Global**: the whole model is explained for a given dataset.

3. According to *how* the explanation is made:

    - **Model-agnostic**: the method is applicable to different Machine Learning models, since it does not inspect specific parameters.
    - **Model-specific**: the method is specifically built for a given Machine Learning model since it uses its internal structure and parameters to provide the explanations.

2.2 The problem of causality

At a time when data and the information they provide us with are taking on a central role, some scientists started to wonder if they could help establish cause-effect relationships, allowing to understand or even predict effects. The relevance that this issue has acquired is exemplified by the fact that the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel (commonly known as the Nobel Prize in Economics) was awarded in 2021 to Joshua Angrist and Guido Imbens for their contributions on the analysis of causality.

In Statistics, the field of causal inference is gaining importance, with scientists aiming to formally establish existent and predict future causal relationships. For this purpose, they are using different mathematical tools. In this section we will describe Judea Pearl's approach [25] to the problem of causality which combines a symbolic language, *do*-calculus, with the use of graphs.

2.2.1 Causal diagrams

Causal diagrams are a powerful tool in causal inference used to represent cause-effect relationships graphically but also to facilitate formalisation [23]. The main diagram used is the **directed acyclic graph (DAG)**.

Definition 2.2.1 (Directed acyclic graph)**.** *In causal inference, a directed acyclic graph (DAG) is a graph in which nodes represent variables and each arrow a cause-effect relationship between two of them. A DAG does not contain cycles, and it is only complete if variables that are common cause for any other two are included.*

It is important to note that DAGs are acyclic since variables should not cause themselves. This implies that bidirected arrows are not permitted, as two variables cannot cause each other [30].

There are three basic causal structures [25, 27]:

1. **Chain**: A $\rightarrow$ B $\rightarrow$ C. B is a mediator between the cause A and the effect C.
2. **Fork**: A $\leftarrow$ B $\rightarrow$ C. B is a common confounder for A and B.
3. **Collider** or **inverted fork**: A $\rightarrow$ B $\leftarrow$ C. Both A and C are causes for B.

Example 2.2.2 (DAG)**.** *In the following DAG we can see a chain (A $\rightarrow$ B $\rightarrow$ C), a fork (D $\leftarrow$ B $\rightarrow$ C) and a collider (A $\rightarrow$ E $\leftarrow$ C):*

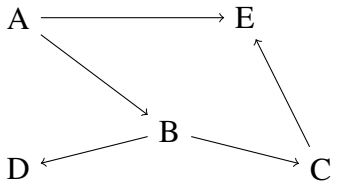

Figure 2.2: Directed acyclic graph

Another type of causal diagram that will be particularly useful throughout this work is the **causal chain graph**. They are useful when the complete causal structure is not known but only a partial ordering is available.

Definition 2.2.3 (Causal chain graph)**.** *A causal chain graph consists of chain components that contain a certain number of variables and are linked by directed edges without forming cycles [12]. In each chain component, the relationship between variables can be that of a common confounder, mutual interactions, etc.*

Example 2.2.4 (Causal chain graph)**.** *In a causal chain graph, only causal relationships between different chain components are known. The relationships between the variables of a given chain component are assumed. In this case, we assume a common confounder for A, B, C and mutual interactions for G, H, I.*

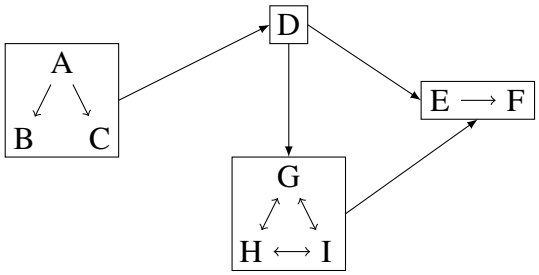

Figure 2.3: Causal chain graph

2.2.2 Pearl's *do*-calculus

The *do*-calculus is an axiomatic system developed by Judea Pearl in 1995 to enable identification of causal effects in models [24]. The basis of the *do*-calculus is the operator $do(X = x)$. It represents an intervention in which a function X of the model is replaced by a constant $X = x$, therefore actively setting the value of the variable.

The idea is to be able to manipulate expressions of the form $p(y \mid do(x), z, w)$ so that we arrive to classic probabilities without the *do*-operator. There are three basic rules that, applied successively, allow to do so. These rules are dependent the causal structure given by the diagrams, and in fact allow for them to be simplified.

We will denote $G_{\overline{X}}$ the operation of deleting edges from graph G that point to variable X, and $G_{\underline{X}}$ the one of deleting edges that come out from X. The rules are therefore the following [24, 25]:

1. **Rule 1** (Insertion/deletion of observations):

   $$p(y \mid do(x), z, w) = p(y \mid do(x), w)$$

   if the variables W and X block all possible paths from Y to Z in $G_{\overline{X}}$.

2. **Rule 2** (Action/observation exchange):

   $$p(y \mid do(x), do(z), w) = p(y \mid do(x), z, w)$$

   if the variables W and X block all possible paths from Y to Z in $G_{\overline{X}\underline{Z}}$.

3. **Rule 3** (Insertion/deletion of actions):

   $$p(y \mid do(x), do(z), w) = p(y \mid do(x), w)$$

   if the variables W and X block all possible paths from Y to Z in $G_{\overline{X}\overline{Z(W)}}$, where $G_{\overline{X}\overline{Z(W)}}$ accounts for the graph $G_{\overline{X}}$ withouth the edges that go into Z if Z is not an ancestor of W.

For further explanations on the *do*-calculus the reader is referred to [24].

2.3 Prediction models

In this section we will introduce briefly the different prediction models supported by the `shapr` package, as explained in Chapter 4. These models can be used both in classification and regression problems.

2.3.1 Generalized Linear Models

Generalized Linear Models (GLM) emerged as a generalisation of linear models. These models assume that the response variable $\mathbf{Y}$ follows a distribution in the exponential family. GLM have three components [19]:

- *Random component*: probability distribution of the response variable $\mathbf{Y}$.
- *Systematic component*: linear predictor produced by the covariates $(x_1, \ldots, x_n)$, noted as $\eta = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$.
- *Link function*: associates the random and systematic components, $E[\mathbf{Y}] = \eta$.

In GLM models, the linear relationship is not between the response variable and the covariates, but between the transformed response variable by means of the link function and the covariates.

Example 2.3.1 (Logistic regression)**.** *The logistic regression is an example of GLM which can be used in binary classification problems. The link function is* $\eta = \log\left(\frac{\pi}{1-\pi}\right)$ *where* $\pi = \mathbb{P}(\mathbf{Y} = 1 \mid \mathbf{X} = x)$ *and* $1 - \pi = \mathbb{P}(\mathbf{Y} = 0 \mid \mathbf{X} = x)$.

Generalized Additive Models with integrated smoothness estimation

Another example of GLM are Generalized Additive Models (GAM). In this case, the linear predictor is obtained by adding the result of applying smooth functions over the covariates and a conventional parametric component of the linear predictor (α):

$$\eta = \alpha + f_1(\mathbf{X}_1) + \cdots + f_n(\mathbf{X}_n) \tag{2.3.1}$$

Smooth functions over the covariates can be chosen freely by the user. However, there are selection techniques based on likelihood that penalize its maximization in order to avoid over-fitting.

2.3.2 Random Forests

Random Forests are a supervised learning algorithm that was firstly introduced by L. Breiman in [2] in 2001. The idea is to build a large number of non-correlated decision trees and then averaging the prediction of each of them, which will therefore be the Random Forest prediction. This is known as *ensemble learning*, since multiple algorithms are combined to obtain more accurate predictions.

The algorithm as described in [11] is the following:

Algorithm 1 Random Forest

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample of size M from the training data.
   (b) Create a decision tree T_b of size n_{min} (minimum node size) by repeating:
      i. Randomly select p variables from the n initial variables.
      ii. Pick the best separation variable among the p selected.
      iii. Split the node into two.
2. Output the ensemble of trees $\{T_b\}_1^B$
3. The prediction for an individual x will be $C_{rf}^B(x) = majorityvote\{C_b(x)\}_1^B$, where $C_b(x)$ is the class prediction for the b-th random forest.

Fitting B models using B different bootstrap samples and then averaging the prediction is a technique called *bootstrap aggregation* or *bagging*. Another characteristic of Random Forests is *feature randomness*. Randomly selecting a subset of variables from the original space of features allows for more independence across trees, since the separation variable at each step will be different.

Finally, this method can also be used for regression. In that case, the prediction for an individual x can be calculated as $f_{rf}^B = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

2.3.3 Extreme Gradient Boosting (XGBoost)

The eXtreme Gradient Boosting (XGBoost) algorithm is another supervised learning algorithm that can be used both in regression and classification problems. It gained notoriety as it was widely used and achieved great results in Machine Learning competitions [10].

The idea behind it is building a large number of trees using boostrap aggregation, but in this case trees are built sequentially trying to minimize errors from previous ones. This technique is known as tree *boosting*.

The error is minimized using a *gradient descent* optimization algorithm, with gives name to XGBoost. The adjective *extreme* comes from the fact that it is optimized to treat missing values and use few computational resources to provide accurate predictions. A detailed explanation of the algorithm can be found in [3].

Chapter 3

SHAP

SHAP (SHapley Additive exPlanations) is one of the most commonly used model agnostic techniques to explain predictions. In a Machine Learning model, it allows to quantify the contribution of a particular feature in the prediction for an observation by assigning it an importance value [17].

This measure of importance is calculated through the use of *Shapley values*, designed in the first instance to solve a problem in Game Theory. Later on, the underlying idea was applied in the area of Explainable Machine Learning.

3.1 Shapley values

Shapley values were proposed by Lloyd Shapley in 1953 in the context of Game Theory [28]. The purpose was to numerically determine the contribution or marginal gain of each player in a cooperative game.

Definition 3.1.1 (Characteristic function)**.** *Let $N = \{1, \ldots, n\}$ be a set of players. The function $v : 2^N \to \mathbb{R}$ defined over this set is known as characteristic function. It measures the value or contribution of a given subset S of N with $v(\emptyset) = 0$.*

The characteristic function can therefore measure the marginal gain of an individual player or a subgroup of them. As players may have overlapping skills, it is needed to average the contribution of a player in all possible formations, i.e. in all possible subsets of N. A Shapley value is therefore the calculation of the marginal value of all possible subsets weighted over all possible permutations.

Definition 3.1.2 (Shapley value)**.** *Let $N = \{1, \dots, n\}$ be a set of n players, S a subset with size s and v the characteristic function. The Shapley value of player i is defined as:*

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!} [v(S \cup \{i\}) - v(S)] \tag{3.1.1}$$

With n being the number of players, $S \subseteq N \setminus \{i\}$ indicates all the possible subsets in which we can add player i to the set. The term $s!$ accounts for all the possible permutations of players in S and $(n-s-1)!$ for those belonging to the residual players that are not in it. Finally, $n!$ is the total number of possible sets, taking order into account. It acts therefore as a normalizing constant as it is in the denominator of all weights.

Proposition 3.1.3 (Properties of Shapley values)**.** *Shapley values satisfy the following properties [12]:*

1. ***Symmetry**: If $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq N \setminus \{i, j\}$, then $\phi_i(v) = \phi_j(v)$.*
2. ***Null-player (Dummy)**: If $v(S \cup \{i\}) = v(S)$ for all $S \subseteq N \setminus \{i\}$, then $\phi_i(v) = 0$. A player that never contributes to the game either directly or indirectly has null Shapley value.*
3. ***Linearity**: If we have two possible value functions v_1 and v_2, then they satisfy $\phi_i(\alpha_1 v_1 + \alpha_2 v_2) = \alpha_1 \phi_i(v_1) + \alpha_2 \phi_i(v_2)$ for any $\alpha_1, \alpha_2 \in \mathbb{R}$.*
4. ***Efficiency**: The marginal gain of each player of the game equals the total contribution of the set of players: $v(N) = \sum_{i=1}^{N} \phi_i(v)$.*

3.1.1 Model explanation using Shapley values

In Machine Learning, Shapley values are at the core of some **local, post-hoc, model-agnostic interpretation methods**. Let us assume that we have a prediction model f over a set N of n features that we aim to explain. Shapley values allow to measure the contribution of a feature i to a prediction made for a specific observation $x = (x_1, \dots, x_n)$.

For linear models, Shapley values can be calculated as [17]

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!} [f_{S\cup\{i\}}(S \cup \{i\}) - f_S(S)] \tag{3.1.2}$$

where the characteristic function is set as $v(S) = f_S(S)$ and is the result of evaluating the model f_S trained only with the features present in the subset S.

In general, Shapley values measure the difference between the estimate of the model for features that are not in S and the averaged predicted value of the model. The estimate of the model over a subset of parameters is the result of integrating the parameters that are not in the subset, i.e. marginalizing over features that are not in the model. As a result we obtain the following expression of the characteristic function:

$$v(S) = \int f(x_1, \ldots, x_n) d\mathbb{P}_{i \notin S} - E_X[f(x)] \tag{3.1.3}$$

When substituting this characteristic function in Equation 3.1.1, since the expected value of the model $E_X[f(x)]$ will be the same for any subset S, it will cancel. The problem that remains then is how to approximate the integral of the parameters that are not in the subset.

3.2 SHAP

SHAP (SHapley Additive exPlanations) was introduced by Lundberg and Lee as a framework that allows prediction interpretation [17]. The idea is explaining the prediction of a given instance by calculating the marginal contribution of each feature. It is an additive feature attribution method.

Proposition 3.2.1. *If x is a given input and x' its coalition vector, SHAP verifies the following properties:*

1. ***Local accuracy**: The explanation model and the original model agree*

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{n} \phi_i x'_i$$

This property is consistent with the efficiency property of the Shapley values (3.1.3) by setting $\phi_0 = E[f(x)]$ and $x'_i = 1$ for all $i = 1, \ldots, n$ [20].

2. ***Missingness**: $x'_i = 0 \Rightarrow \phi_i = 0$, that is, features not present in a coalition do not have an impact in the explanation.*

3. ***Consistency**: If two models f and f' verify that for all $S \subseteq N$*

$$f'_x(S) - f'_x(S \setminus \{i\}) \geq f_x(S) - f_x(S \setminus \{i\})$$

then $\phi_i(f', x) \geq \phi_i(f, x)$, where $f_x(S) = f(h_x(z'))$ and $z' \in \{0, 1\}^n$ is the coalition vector for the subset S. This means that if the contribution of a feature increases when changing the model, then its Shapley value will either increase or stay the same.

Theorem 3.2.2. *The only possible explanation model g that satisfies the properties above is obtained by setting*

$$\phi_i(f) = \sum_{S \subseteq N \setminus \{i\}} \frac{s!(n-s-1)!}{n!}[f_x(S \cup \{i\}) - f_x(S)] \tag{3.2.1}$$

with $n = |N|$, $s = |S|$ and $f_x(S) = f(h_x(z'))$ as defined previously.

3.3 SHAP values

SHAP values were proposed in [17] and are the Shapley values of a conditional expectation function of the original model, that is, they approximate the integral in Equation 3.1.3 by the expected output of the model conditional on the values of the subset chosen. They set the characteristic function of Shapley values as

$$v(S) = f_x(S) = f(h_x(z')) = E[f(z) \mid z_S = x_S] \tag{3.3.1}$$

where $z_S = h_x(z')$. Since z' is the coalition vector for a subset $S \subseteq N$, the vector z_S in the original space will have missing values for the features that are not included in the coalition S. We are therefore approximating $f(z_S)$ by using $E[f(z) \mid z_S = x_S]$, the expected output of the model conditional on the feature values of the subset. This conditional expectations can be rewritten as:

$$\begin{aligned} v(S) = E[f(z) \mid z_S = x_S] = E[f(z_{\bar{S}}, z_S) \mid z_S = x_S] = \\ = \int f(z_{\bar{S}}, x_S) p(z_{\bar{S}} \mid z_S = x_S) dz_{\bar{S}} \end{aligned} \tag{3.3.2}$$

with $\bar{S} = N \setminus S$. From this we deduce that in order to compute SHAP values all the conditional distributions $p(z_{\bar{S}} \mid z_S = x_S)$ are necessary.

By assuming feature independence, we can approximate $p(z_{\bar{S}} \mid z_S = x_S) \approx p(z_{\bar{S}})$ and therefore

$$\begin{aligned} v(S) = f_x(S) = f(h_x(z')) = E[f(z) \mid z_S = x_S] = E[f(z_S, z_{\bar{S}}) \mid z_S = x_S] = \\ = \int f(z_{\bar{S}}, x_S) p(z_{\bar{S}} \mid z_S = x_S) dz_{\bar{S}} \approx \int f(z_{\bar{S}}, x_S) p(z_{\bar{S}}) dz_{\bar{S}} = \\ = E[f(z_{\bar{S}}, z_S)] \end{aligned} \tag{3.3.3}$$

If model linearity is assumed on top of feature independence, a further simplification is

$$v(S) = E[f(z) \mid z_S = x_S] \approx E[f(z_{\bar{S}}, z_S)] \approx f(E[z_{\bar{S}}], z_S) \tag{3.3.4}$$

It is important to note that a feasible computation of this SHAP vales assumes feature independence to the marginal expectation. This raises the problem of how to deal with features that are not independent, and whether supposing independence may result in erroneous explanations of the model.

3.4 Methods for feature dependency

Shapley values have proved to be a useful and solid tool to explain predictions of complex Machine Learning models. However, as mentioned before, they generally assume features to be independent, which may result in erroneous or counterintuitive explanations.

To deal with this problem, Aas, Jullum and Løland proposed different methods in [1] that allow to estimate directly the conditional probability $p(z_{\bar{S}} \mid z_S = x_S)$ without the need of assuming feature independence, and thus maintaining the dependence structure in data.

3.4.1 Multivariate Gaussian distribution method

This method assumes that feature vectors come from a multivariate Gaussian distribution $Z \sim \mathcal{N}_N(\mu, \Sigma)$, with N the number of features. The vector μ and the covariance matrix Σ can be estimated as the mean and covariance of the sample of the training data.

Under this assumption, the conditional distribution $p(z_{\bar{S}} \mid z_S = x_S)$ is also a multivariate Gaussian $\mathcal{N}_{|\bar{S}|}(\mu_{\bar{S}|S}, \Sigma_{\bar{S}|S})$. If we write μ and Σ as

$$\mu = (\mu_S, \mu_{\bar{S}}) \quad \text{and} \quad \Sigma = \begin{bmatrix} \Sigma_{SS} & \Sigma_{S\bar{S}} \\ \Sigma_{\bar{S}S} & \Sigma_{\bar{S}\bar{S}} \end{bmatrix}$$

then the parameters $\mu_{\bar{S}|S}$ and $\Sigma_{\bar{S}|S}$ can be obtained as indicated in [1]

$$\mu_{\bar{S}|S} = \mu_{\bar{S}} + \Sigma_{\bar{S}S}\Sigma_{SS}^{-1}(x_S - \mu_S) \tag{3.4.1}$$

$$\Sigma_{\bar{S}|S} = \Sigma_{\bar{S}\bar{S}} - \Sigma_{\bar{S}S}\Sigma_{SS}^{-1}\Sigma_{S\bar{S}} \tag{3.4.2}$$

By obtaining K samples from this distribution, both $p(z_{\bar{S}} \mid z_S = x_S)$ and the integral in Equation 3.3.2 can be approximated, thus obtaining the characteristic function:

$$v(S) = \frac{1}{K}\sum_{k=1}^{K} f(z_{\bar{S}}^k, x_S) \tag{3.4.3}$$

where $z_{\bar{S}}^k$, $k = 1, \ldots, K$ are the samples obtained and x_S are the values of the observation x for features included in S.

3.4.2 Gaussian Copula method

If features do not seem to follow a multivariate Gaussian distribution, the dependence structure can be accounted for by means of a Gaussian copula, with marginals represented by their empirical distributions.

Definition 3.4.1 (Copula)**.** *A n-dimensional copula is a function $C : [0,1]^n \to [0,1]$ that is the joint cumulative distribution function of a d-dimensional random vector and satisfies [22]:*

1. *If $u_k = 0$, $C(u_1, \ldots, u_{k-1}, 0, u_{k+1} \ldots, u_n) = 0$, $\forall k = 1, \ldots, n$.*
2. *$C(1, \ldots, 1, u_k, 1 \ldots, 1) = u_k$, $\forall k = 1, \ldots, n$.*
3. *C is a n-increasing function.*

Theorem 3.4.2 (Sklar, 1959)**.** *Let H be a joint distribution function with marginal distributions $F_1, \ldots, F_n$. Then there exists a n-copula C such that for all $u_1, \ldots, u_n \in \mathbb{R}^n$ [22]:*

$$H(u_1, \ldots, u_n) = C(F_1(u_1), \ldots, F_n(u_n))$$

By assuming a Gaussian copula, the expressions in Equations 3.4.1 and 3.4.2 can be used to approximate $p(z_{\bar{S}} \mid z_S = x_S)$ by generating samples through the following steps [1]:

1. If Z is the distribution of the features, approximating each marginal Z_j to a Gaussian $V_j = \Phi^{-1}(\hat{F}_j(Z_j))$, where Φ is the cumulative distribution function of a standard normal distribution and $\hat{F}_j$ the empirical cumulative distribution function of the marginal Z_j.
2. Assuming V follows a multivariate Gaussian, the previously described multivariate Gaussian distribution method can be used to obtain the desired samples for the margins distribution V_j.
3. Transforming back to the original distribution using that $\hat{Z}_j = \hat{F}_j^{-1}(\Phi(V_j))$.

As in the previous method, we can obtain K samples $z_{\bar{S}}^k$ from this distribution and approximate the characteristic function using the formula in Equation 3.4.3.

3.4.3 Empirical conditional distribution method

When neither the marginal distributions nor the dependence structure follow a Gaussian distribution, a different, non-parametric approach is needed. This method is based on the idea that if samples $(z_{\bar{S}}, z_S)$ have $z_S \approx x_S$, then they are informative about the conditional distribution $p(z_{\bar{S}} \mid x_S)$.

The method relies therefore in the calculation of distances between z_S and x_S, using a version of the *Mahalanobis distance*.

Definition 3.4.3 (Mahalanobis distance)**.** *The Mahalanobis distance between two vectors x and y respect to the covariance matrix Σ is:*

$$\rho(x, y) = \sqrt{(y - x)'\Sigma^{-1}(y - x)} \tag{3.4.4}$$

The steps are the following [1]:

1. Computing the distance between the instance to be explained x and all the m training instances z^i, $i = 1, \ldots, m$ using a scaled Mahalanobis distance:

$$\rho_S(x, z^i) = \sqrt{\frac{(z^i - x)'\Sigma_S^{-1}(z^i - x)}{|S|}} \tag{3.4.5}$$

   with Σ_S the covariance of the n training instances z^i.

2. Using these distances to compute weigths by means of a Gaussian kernel:

$$\omega_S(x, z^i) = e^{-\frac{\rho_S(x, z^i)^2}{2\sigma^2}} \tag{3.4.6}$$

   with σ a bandwith parameter that has to be specified.

3. With an increased order of the weights, $z^{[k]}$ refers to the k-th largest weight. The integral in Equation 3.3.2 can be approximated as:

$$v(S) = \frac{\sum_{k=1}^{K} \omega_S(x, z^{[k]}) f(z^{[k]}, x_S)}{\sum_{k=1}^{K} \omega_S(x, z^{[k]})} \tag{3.4.7}$$

   where $K \leq m$ can be adjusted so that only the K largest weights are taken into account.

The number K of samples can be also chosen so that a fraction η of the total weight is accounted for

$$K = \min_{L>0} \left\{ \frac{\sum_{k=1}^{L} \omega_S(x, z^{[k]})}{\sum_{i=1}^{m} \omega_S(x, z^i)} \right\}$$

Finally, the role of the bandwith parameter σ is that of adjusting the trade-off between variance and bias. A small value of σ will give as a result low bias and high variance since most weight will be put on the closest training observations. On the other hand, a big value will distribute more the weights resulting in high bias but low variance.

Authors in [1] suggest a method to choose σ using the corrected Akaike Information Criterion (AICc), a version proposed in [13] to deal with small sample sizes:

$$\text{AICc} = \text{AIC} + \frac{2(k+1)(k+2)}{n-k-2}$$

where n is the sample size and k the number of estimated parameters.

3.5 Methods for causal structures

In addition to feature dependency, there may also exist causal structures that affect model explanation. In this context, new frameworks have been proposed to incorporate these causal structures, leading to new definitions of different ways of computing Shapley values.

It is important to note that methods described below require additional information about the causal relationships between variables. This information is not easily available, and is usually obtained from empirical studies, although some researchers are currently focusing in the development of techniques that allow to find it directly from data [29].

3.5.1 Asymmetric Shapley values

A more general definition of Shapley values can be given by taking into account a particular ordering $\pi \in \Pi$ for the features, where Π is the set of all permutations of elements in N.

Definition 3.5.1 (Shapley value - General definition)**.** *Let $\Delta(\Pi)$ be the set of probability measures over Π so that $w \in \Delta(\Pi)$ is a distribution over the permutations, $w : \Pi \to [0, 1]$, that satisfies $\sum_{\pi\in\Pi} w(\pi) = 1$. The Shapley value of player i is defined then as:*

$$\phi_i^w(v) = \sum_{\pi\in\Pi} w(\pi)[v(\{j : \pi(j) \leq \pi(i)\}) - v(\{j : \pi(j) < \pi(i)\})] \tag{3.5.1}$$

with $\pi(j) < \pi(i)$ if feature j precedes i in the permutation π.

If the choice for the distribution over the permutations is the uniform $w(\pi) = \frac{1}{n!}$, we retrieve 3.1.1. It is important to note that property (1) of symmetry is lost for non-uniform distributions, although the rest apply [12].

Frye *et al.* [7] named these Shapley values for which symmetry is lost as *Asymmetric Shapley values*, and proposed them as a way to incorporate causal knowledge into model explainability. They argue that the property of symmetry may confuse causal relationships in the data, and that some choices of $w(\pi)$ incorporate causal understanding into the explanation.

If there are causal dependencies between features, this can be introduced by setting $w(\pi) = 0$ for those permutations in which a known ancestor i does not precede a descendant j. As such, the Asymmetric Shapley value of feature x_i will measure its effect by assuming feature x_j is unknown (since it is its descendant), while the Asymmetric Shapley value of feature x_j will assume its predecesor x_i has already been specified.

3.5.2 Causal Shapley values

Heskes *et al.* [12] try to provide explanation that takes into account causal dependencies through a different method. They introduce the notion of *Causal Shapley value* by choosing the following characteristic function:

$$v(S) = E[f(z) \mid do(z_S = x_S)] = \int f(z_{\bar{S}}, x_S)\, p(z_{\bar{S}} \mid do(z_S = x_S))\, dz_{\bar{S}} \qquad (3.5.2)$$

This equation is just the result of replacing $p(z_{\bar{S}} \mid z_S = x_S)$ in Equation 3.3.2 by $p(z_{\bar{S}} \mid do(z_S = x_S))$. The use of Pearl's *do-calculus* allows to measure the contribution of a given feature if we actively set its value compared to not knowing it.

The practical calculation of these causal Shapley values is made by means of the causal diagrams presented in Section 2.2.2. It is possible to retrieve the formulas for $p(z_{\bar{S}} \mid do(z_S = x_S))$ from both DAGs and causal chain graphs, since the latter can be considered as a DAG where instead of variables there are chain components. Details on these formulas can be found in [12].

3.6 Other methods

Lundberg and Lee [17] proposed numerous methods that supposed an improvement to previous approaches to explainability, such as Kernel SHAP, Deep SHAP, Linear SHAP and Max SHAP. We will present Kernel SHAP, Tree SHAP [16] and the *Monte-Carlo sampling* method.

3.6.1 Kernel SHAP

Kernel SHAP combines ideas from Shapley values and LIME (Local Interpretable Model-agnostic Explanations) [26], another local explainability method. LIME creates local linear explainable models by training a dataset of perturbed samples from the original instance. The originality of this method is that it is possible to retrieve the Shapley values from the coefficients of a weighted linear model.

The weights are obtained from a SHAP kernel proposed by Lundberg and Lee in [17]:

$$\pi_x(z') = \frac{n-1}{\binom{n}{|z'|}\,|z'|(n-|z'|)} \tag{3.6.1}$$

where $|z'|$ is the number of features present in the coalition vector z', i.e. the number of components of the vector that are equal to 1.

They prove then that if the explanation model g is trained by optimizing the loss function

$$L(f, g, \pi_x) = \sum_{z' \in Z} \left[f(h_x(z')) - g(z')\right]^2 \pi_x(z') \tag{3.6.2}$$

then the Shapley values can be recovered. Kernel SHAP algorithm is as follows:

Algorithm 2 Kernel SHAP

1. For $k = 1$ to K:
   (a) Sample a coalition vector $z'_k \in \{0,1\}^n$.
   (b) Convert z'_k to the original feature space by using h_x, with x the instance of interest, and then apply the model to get $f(h_x(z'_k))$.
   (c) Use the SHAP kernel (Equation 3.6.1) to calculate the weight $\pi_x(z'_k)$ for z'_k.
2. Fit the weighted linear model by optimizing the loss function described in Equation 3.6.2.
3. The coefficients of this linear model will be the Shapley values ϕ_k.

This method is an improvement since it offers a solution to the problems that arise from the exponential complexity of the Shapley values formula and the challenge of estimating SHAP values. However, it still assumes feature independence, and therefore presents the same issues when dealing with dependencies between variables.

3.6.2 Tree SHAP

Tree SHAP allows to compute SHAP values for decision tree models in a more computationally efficient way [16].

The characteristic function is defined by means of the conditional expectation instead of the marginal expectation. This results in a reduction the complexity of computing from $O(TLD^N)$ to $O(TL2^N)$, where T is the number of trees, L the maximum number of leaves in any tree, D the maximum depth and N the number of features.

The aim is to calculate $E[f(z) \mid z_S = x_S]$ in a tree model. The prediction for z can fall into different leaves since only known values for z are $z_S = x_S$. The idea is then the following [20]:

- If $S = N$ the set of all features, the expected value will be the expected prediction of the model, the only leave into which z can fall.
- If $S = \emptyset$, the expected value will be the weighted average of all leaves, where the weights are calculated by node size (number of training samples that fall in that node).
- If S contains some features, the expected value will be the weighted average of the leaves that can be reached following the decision path given by $z_S = x_S$.

This last calculation can be performed in polynomial time by calculating the outcome of considering all possible subsets at the same time, descending in the tree. A detailed explanation of the algorithm can be found in [16] and [18].

Finally, as Shapley values are additive, this method is also useful when working with tree ensemble models, since the SHAP values will be the average of the SHAP values for the single trees.

3.6.3 Monte-Carlo sampling

When dealing with large sets of features, problems of computational nature may arise. All possible subsets $S \subseteq N$ need to be evaluated, thus becoming an intractable problem. A possible solution lies in the use of the *Monte-Carlo sampling* algorithm [20], which allows to retrieve an approximation to the Shapley value of a feature i for a given individual x.

Algorithm 3 Monte-Carlo sampling method to approximate Shapley values

1. For $m = 1$ to M:
   (a) Obtain a second individual y by sampling randomly from the dataset.
   (b) Create two new instances from x and y by replacing a number of feature values by others from the individual y:

$$x^m_{+i} = (x_1, \ldots, x_{i-1}, x_i, y_{i+1}, \ldots, y_n)$$
$$x^m_{-i} = (x_1, \ldots, x_{i-1}, y_i, y_{i+1}, \ldots, y_n)$$

   In the first case, the original value of feature i for the individual x is kept while in the second it is also taken from the sampled individual y.
2. The Shapley value of feature i will therefore be the result of averaging the difference of those two instances:

$$\hat{\phi}_i = \frac{1}{M} \sum_{m=1}^{M} \big(f(x^m_{+i}) - f(x^m_{-i}) \big) \tag{3.6.3}$$

3.6.4 Feature importance

Shapley values as described in this section allow to calculate importance of a feature in a particular prediction (local interpretability). In order to estimate the importance of a feature for a Machine Learning method, it is possible to average Shapley values across instances.

If our prediction model has been trained over n features, the relevance of feature j can be calculated as follows

$$\Phi_j = \frac{1}{M} \sum_{i=1}^{n} |\phi^i_j| \tag{3.6.4}$$

where ϕ^i_j represents the Shapley value of feature j for the instance i.

This method would give a measure of **global interpretability**, that is, an explanation of the expected model behaviour [21].

Chapter 4

Computational implementation

In this chapter, the computational tools used to design and carry out the different experiments in Chapter 5 will be described. The packages for Shapley value computation, model training and plotting will be described.

4.1 Shapley values computation

One of the main packages for the calculation of Shapley values in R is `shapr`. It is based on the method for computation of Shapley values described in [1]. The package is composed of three functions [15]:

- `shapr`: gets Shapley weights for test data, creating an explainer object.
- `explain`: computes Kernel SHAP values for test data.
- `plot.shapr`: plots the individual prediction explanations.

The first step is training the model. Afterwards, the function `shapr()` can be used to obtain the Shapley weights for test data, that is, the predictions we want to explain. Arguments include the data, the trained model and optionally the number of combinations. By default, when omitting the argument for the number of combinations, it will calculate the weights for the 2^n possible combinations, where $n = |N|$ is the number of features considered. The data should have a numeric form, with categorical features encoded and converted into integer format.

This function currently supports explanation of the following models:

- Generalized linear models: `stats::glm`.
- Linear models: `stats::lm`.
- Random Forests through the package "ranger": `ranger::ranger`.
- Extreme Gradient Boosting (XGBoost): using both `xgboost::xgboost` or `xgboost::xgb.train`.
- Generalized additive models with integrated smoothness estimation using the function `mgcv::gam`.

After the explainer object is created in the previous step, individual predictions can be explained using the function `explain()`. The main arguments are test data corresponding to the individuals to be explained, the explainer object, the approach to be used and the prediction value for unseen data, which is usually the mean of the response.

There are different approaches to the computation of Shapley values as seen in Chapter 3. Some options are *empirical* for the empirical approach, *gaussian* for the multivariate Gaussian distribution approach and *copula* for the Gaussian copula approach, amongst others. If multiple individuals are explained, a different approach for each individual can be chosen by giving a vector of the length of the number of predictions to be explained as an argument.

4.1.1 Causal Shapley values

To compute causal Shapley values as described in Chapter 3, authors in [12] developed an extension of the `shapr` package that allows to do so. The code is available at their GitLab repository. This extension modifies both functions presented before introducing new arguments.

The causal structure of data, also known as *partial ordering* is needed to calculate the causal Shapley values. This partial ordering is given in the form of a list of vectors, each of them representing a chain component of the causal chain graph. The vectors themselves contain the numbers corresponding to the columns of the variables in the dataset that are included in the chain component.

The first function receives two new arguments: `shapr(..., asymmetric = FALSE, ordering = NULL)`. The argument for asymmetry allows to compute both symmetric or asymmetric Shapley values. The ordering argument receives the partial ordering list.

As for the `explain()` function, the approach *causal* is introduced in addition to those mentioned above. Additionally, three new arguments are to be set for the function: the previously seen for asymmetry and ordering and another for confounding. The *confounding* argument accounts for whether the variables in each of the chain components are confounded or not. It is therefore a vector of logicals, with the same length as the partial ordering list.

4.2 Data preparation, model training and plotting

Data preparation The dataset to be used in the experiments has been prepared using functions from the `tidyverse` package to merge, filter and declare types of variables. A descriptive analysis of the features has also been made using this package.

Partitioning data into train and test sets is possible using the library `caret` (Classification And REgression Training) and its function `createDataPartition()`. This function receives a vector with the response variable, the number of partitions to create and the percentage of training data, among other arguments. It returns a list of the row positions that will be included in the training data.

Random Forest Different libraries allow Random Forest implementation. The one which is currently suported by the `shapr` package is the library `ranger`. The function to train the Random Forest is `ranger()`. It receives as input the formula (description of the model to fit) and the training data.

Other arguments that can be set are the number of trees to calculate, whether sampling with replacement or not, weights for sampling or whether the forest to grow is a classification forest. More information is available on the online documentation.

XGBoost XGBoost models can be implemented through the `xgboost` package by using the function `xgb.train()` or the simpler version `xgboost()`. These functions receive the training data in matrix form as an input and a set of parameters.

Other arguments such as the maximum number of iterations, printing information about model performance or saving the model can also be set. A detailed description of the functions and its arguments can be found in the online documentation.

Plotting results Plots have been constructed using `ggplot2`, the package for graphical representation included in `tidyverse`. The code for the individual explanation plots and the sina plots has been adapted from the ones available in the *shaprr* package and its causal extension respectively.

Chapter 5

Experiments

In this chapter, a series of experiments will be performed on real data in order to illustrate the concepts introduced in this work. Data used has been obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI).

We will first describe the dataset and its causal structure, which we will compare with the structure of associations between variables. Later, we will contrast the different ways of calculating Shapley values described in Chapter 3. Finally, we will see if there are differences in Shapley values when working with different models. All the code used for the different experiments can be found on this GitHub repository.

5.1 ADNI database

Data used were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership. Its primary goal has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD).

Features considered are described in Figure 5.1. They are divided into three groups according to their type. The demographic variables account for the sex, years of education and age at baseline (at the beginning of the study) of the patients. Biomarkers are indicators in the human body that empirical studies have associated with the presence of Alzheimer's disease. Finally, the gene ApoE4 has also been linked to an increased risk of development of the disease.

The response variable will be the diagnosis of Alzheimer's disease. It is a categorical variable with three classes: *Alzheimer's disease (AD)*, *mild cognitive impairment (MCI)* and *cognitive normal (CN)*. For simplicity, a new variable DXB has been created by grouping the classes AD and MCI into a new one category *disease present (DP)*. This brings us to a binary classification problem, which will simplify the interpretation of the different Shapley values as mentioned in Section 2.1.

Label	**Description**	**Mean (SD)**	**Classes (%)**
Biomarkers			
FDG	Fudeoxyglucose	6.2 (0.77)	
ABETA	Amyloid beta	171.9 (53.39)	
PTAU	Phosphorilated tau	39.9 (22.83)	
Genetics			
APOE4	Number of apoliprotein alleles		0: no alleles (52%) 1: one allele (37%) 2: two alleles (11%)
Demographic variables			
PTGENDER	Sex		0: male (57%) 1: female (43%)
AGE	Age at baseline	73.1 (7.41)	
PTEDUCAT	Years of education	16.08 (2.79)	
Diagnosis (response variable)			
DX	Diagnosis of Alzheimer's Disease		0: AD (17%) 1: CN (24%) 2: MCI (59%)
DXB	Diagnosis of Alzheimer's Disease (binary classification)		0: DP (76%) 1: CN (24%)

Table 5.1: Continuous and categorical variables description

5.2 The "gold standard" graph

The selection of the ADNI dataset was motivated by the fact that some authors have proposed a causal association graph of some of the variables present in the dataset. As mentioned in Chapter 4, in order to calculate causal Shapley values, a partial ordering of the variables is required. In [29], authors proposed the "gold standard" graph, which is based on relationships between variables that are well established in literature. It is shown in Figure 5.2.

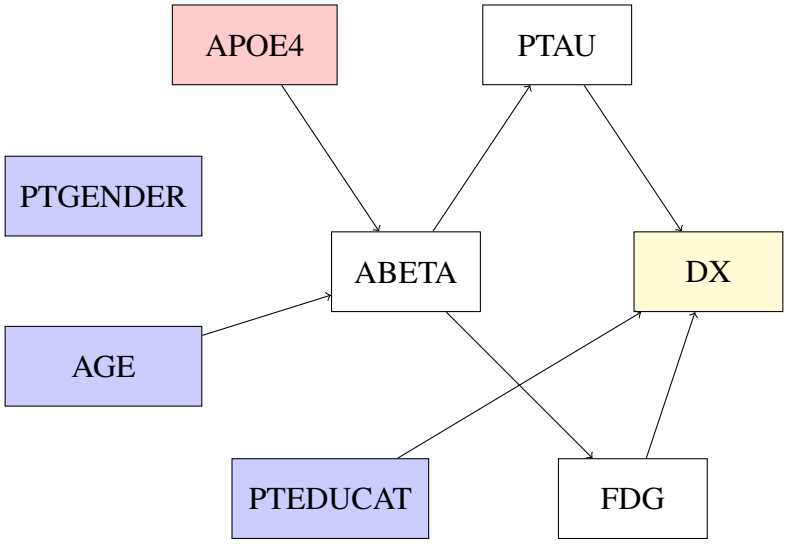

Figure 5.2: "Gold standard" graph

The "gold standard" graph allows therefore to establish the following partial ordering for the calculation of the Causal Shapley values.

$$(\{\text{APOE4, PTEGENDER, AGE, PTEDUCAT}\}, \{\text{ABETA}\}, \{\text{FDG, PTAU}\})$$

We can try to compare this causal structure with that of correlations. Since our dataset contains both categorical and numerical variables, we will better speak of association structure. The way we have calculated the association between variables according to their type is as follows:

- Between numerical variables: Pearson's correlation coefficient.
- Between numerical and categorical variables: ANOVA.
- Between categorical variables: Cramer's V [4], which is based on the χ^2 statistic.

These calculations allow us to construct a matrix of associations, which we can visualise with a type of graph known as network plot. It shows the different variables related by coloured nodes whose colour and intensity depends on the sign and the strength of the association. This network plot for our dataset is shown in Figure 5.3.

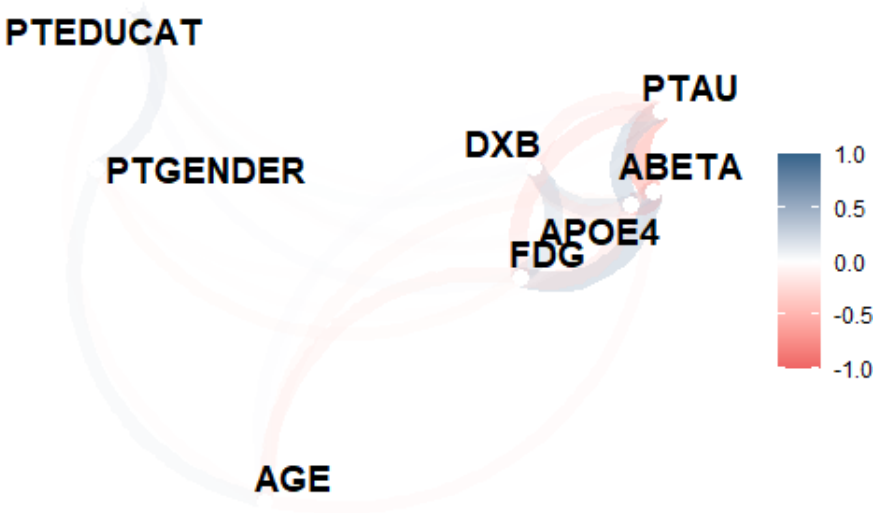

Figure 5.3: Network plot showing associations between variables

The plot shows that the associations between features are not very strong. The most strongly associated variables are grouped together, while the less related ones are further away.

It is observed that the group of variables APOE4, ABETA, PTAU and FDG is connected to each other and to the response variable DXB. In the "gold standard" graph, variables pointing to DXB were PTAU, FDG and PTEDUCAT, while APOE4 and ABETA are related to DXB through intermediate variables. This could explain why all this variables are close together except for the case of PTEDUCAT, which seems poorly associated with any other variable.

Lastly, for the variables PTGENDER and AGE, the "gold standard" graph shows that the variable PTGENDER is isolated, something similar to what happens in the network plot. The AGE variable is an ancestor of ABETA, whereas in this graph they are barely correlated.

5.3 Differences in methods

In this first experiment, we aim at showing whether Shapley values differ when using the different methods for feature dependency described in Section 3.4 and those that take into account the causal structure of data described in Section 3.5.

5.3.1 Feature dependence

We will analyse first the results of using the methods for feature dependency: multivariate Gaussian distribution, Gaussian copulas and empirical conditional distribution.

These three methods will be compared with the result of calculating the Shapley values using the causal method but assuming that all variables are confounded with each other. That is, there is no order or causal structure in the data. These Shapley values are called *Marginal Shapley values* in [12], and we will refer to them in this way.

Outcomes shown in Figure 5.5 belong to the XGBoost model. In the model evaluation metrics we get an accuracy of 77.1%, a sensitivity of 84.5% and a specificity of 45.5%. See also Table 5.4.

		Predicted	
		DP	CN
Actual	DP	120	18
	CN	22	15

Table 5.4: Confusion matrix for the XGBoost model

The summary plots show the results are through sina plots. In them, each dot represents the Shapley value of a given individual for the features in the y-axis. The color gradient indicates the original value for the feature. Results obtained with the first two methods for the two categorical variables (APOE4 and PTGENDER) do not have a strong theoretical support and therefore are not included, as it does not make sense to assume that they come from a Gaussian distribution or a Gaussian copula.

For the numerical variables, we see that results across methods are very similar. For variables such as FDG, PTAU and PTEDUCAT, it is not observed that Shapley values differ according to the original value of the variable. For ABETA and AGE variables it can be seen that small values of the original variable correspond to small Shapley values.

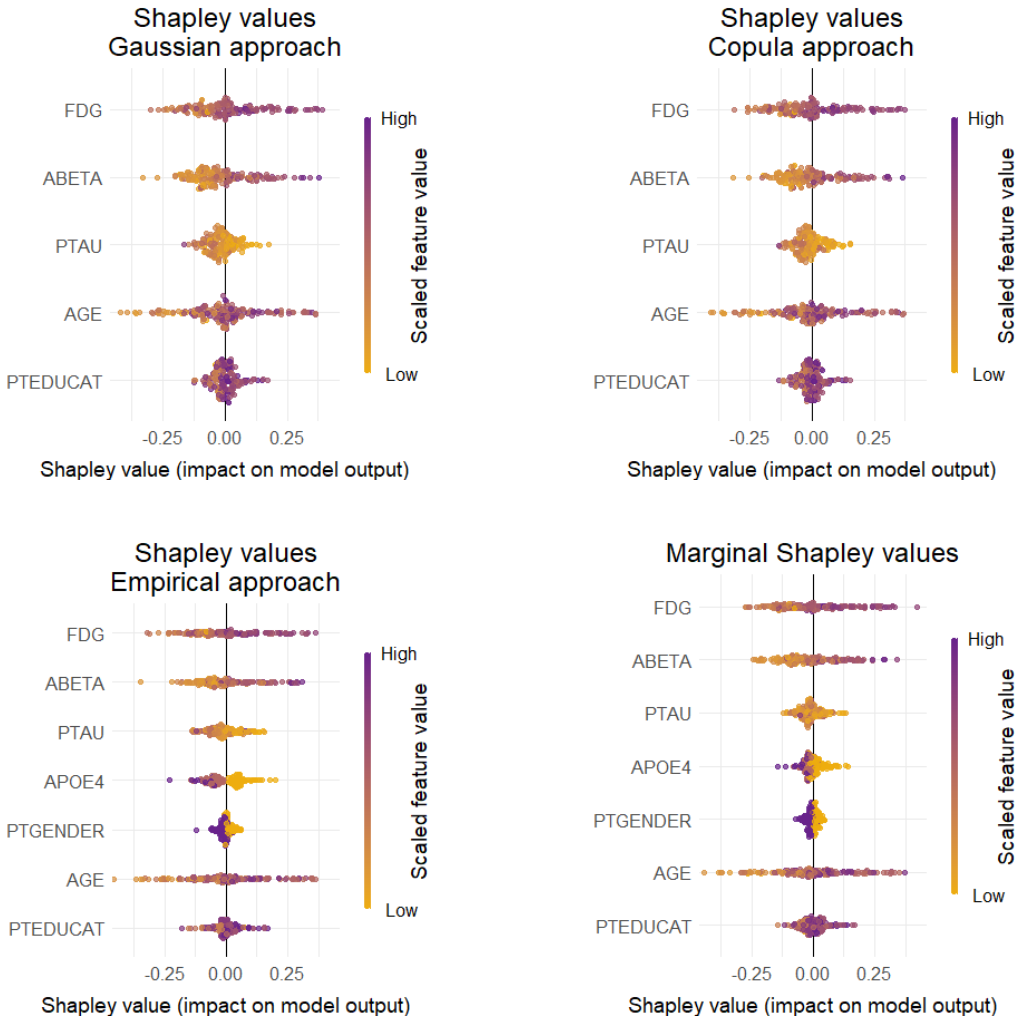

Figure 5.5: Sina plots for the feature dependence methods in the XGBoost model

For the categorical variables, as discussed before it does only make sense to compare the empirical approach and the Marginal Shapley values. We see that the three categories of the APOE4 variable are slightly separated in the empirical approach, though it is not the case in the Marginal Shapley values sina plot. As for PTGENDER, in both methods the two cases are moderately differentiated, with one category corresponding to a negative impact and the other to a positive one.

5.3.2 Causal structure

To deal with the causal structure, we will calculate four different Shapley values using the partial order introduced before when describing the "gold standard" graph. The sina plots with the results calculated for the same XGBoost model as before are shown in Figure 5.6.

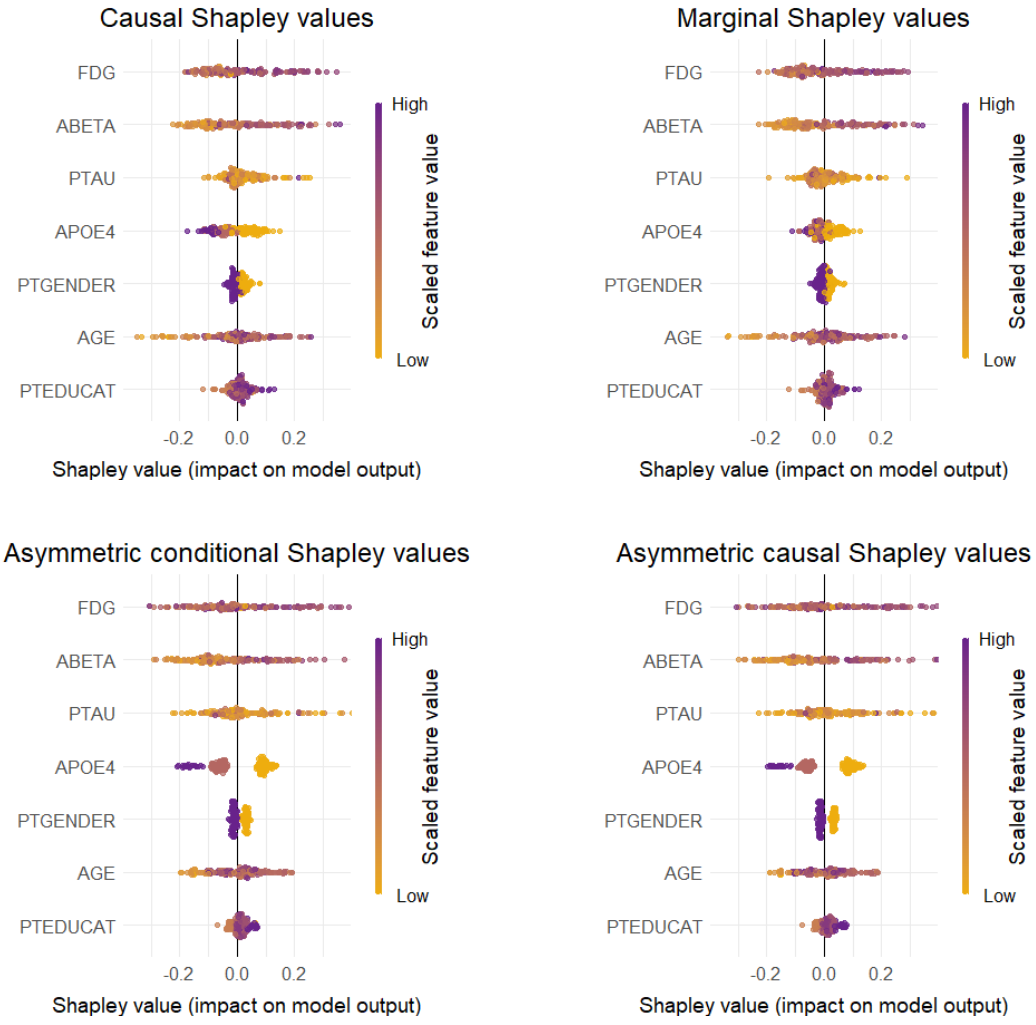

Figure 5.6: Sina plots for the different Shapley values in the XGBoost model

These Shapley values have been calculated setting the arguments of the different functions as follows:

- Symmetric Shapley values: without including the argument for asymmetry.
  - Symmetric causal Shapley values: causal approach and partial order.
  - Marginal Shapley values: causal approach with all variables confounded.
- Asymmetric Shapley values: including the argument for asymmetry.
  - Asymmetric Shapley values: causal approach with all variables confounded.
  - Asymmetric causal Shapley values: causal approach and partial order.

We can see that there are differences across the Shapley values calculated, especially for categorical features. Both asymmetric Shapley values clearly separate the categories of APOE4 and PTGENDER, whereas the symmetric ones do not do so well.

In the case of continuous features, there are hardly any differences between methods. In some of the variables, such as FDG and ABETA, low values of the feature might be associated to negative Shapley values and high values to positive in all methods. For the remaining variables no evident patterns are observed in any of the four types of Shapley values.

5.3.3 Analysis of individual explanations

Additionally, we can plot individual explanations using the four Shapley values dealing with causality for given individuals to further illustrate the experiment. We have chosen three different observations, one corresponding to the positive class (Disease Present, with a low predicted probability), other classified in the negative class (Cognitive Normal, high predicted probability) and the third having a predicted probability near 0.5.

The different plots show the Shapley value for each of the features alongside their value. Positive Shapley values are considered to push to increase the predicted value, that is, the predicted probability, whereas features with negative Shapley values will be the ones pushing to decrease the prediction.

For the first individual the model has predicted the presence of Alzheimer's disease. In all four Shapley value prediction explanations we can see that the variable that influences the decision the most is the age of the individual, which pushes to decrease the predicted probability, therefore contributing to the classification in the positive class.

As for the rest of the variables, it is here where we see differences across methods. In the symmetric one, the other variable that seems to have an influence in the prediction is ABETA, with a positive contribution. However, in both asymmetric methods this variable has nearly no influence, and in the case of the asymmetric Shapley value we see that instead of contributing to increase it does so to decrease the predicted probability.

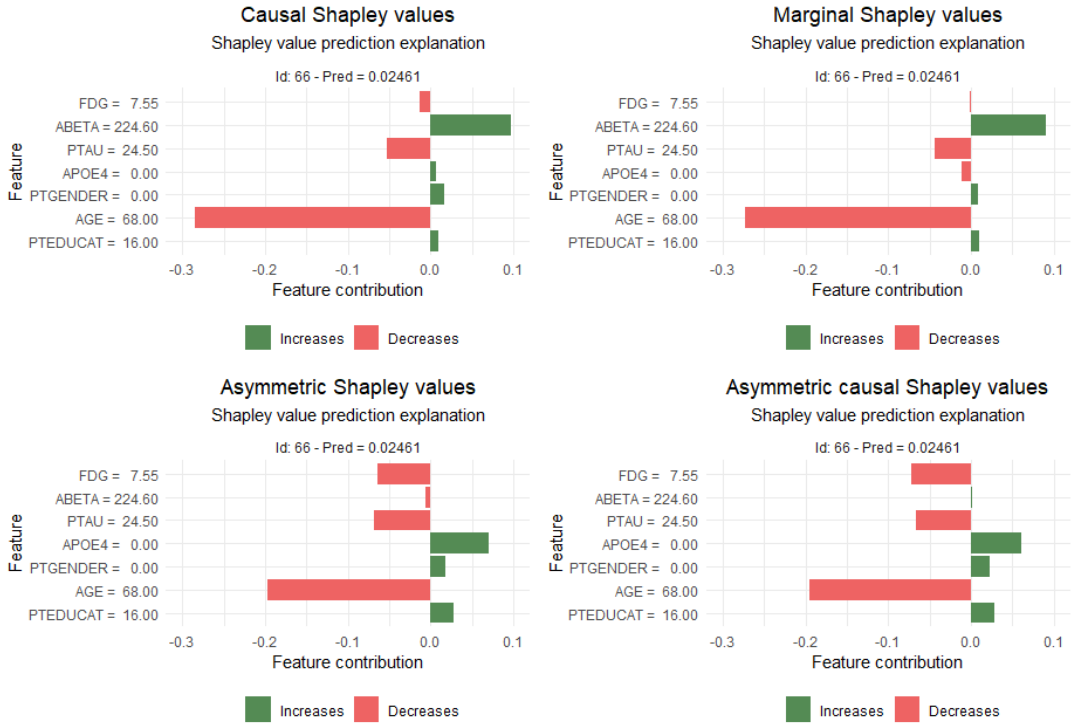

Figure 5.7: Individual plots for the different Shapley values in the XGBoost model

The second individual has a predicted class with no disease present since its predicted probability is nearly 1. In this case the variables that have the greatest influence is FDG, contributing towards an increment of the predicted probability.

For the rest of features, we see differences across methods. For the causal Shapley value, the second variable that influences the prediction the most is the age, as in the marginal Shapley values. For the asymmetric Shapley values we see that we see that both PTAU and AGE have an positive influence in the decision.

Finally, in the case of the asymmetric causal Shapley values both AGE a PTAU have a positive influence as in the precedent case, but it is also remarkable that ABETA pushes towards decreasing the predicted probability, something that does not happen in the rest of cases.

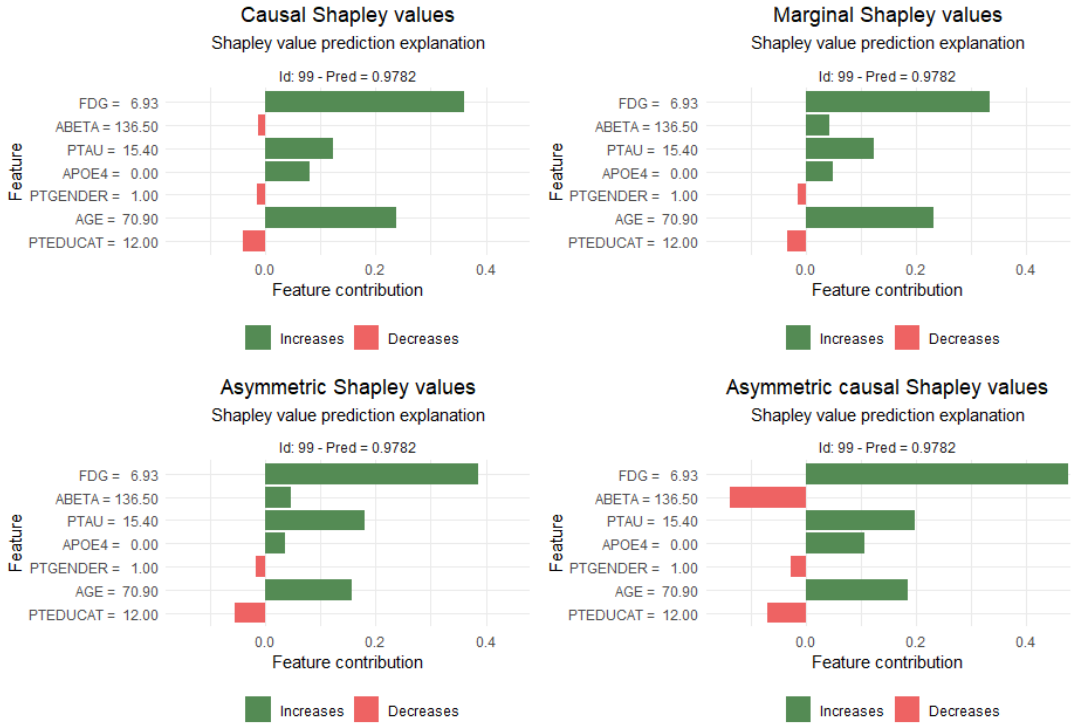

Figure 5.8: Individual plots for the different Shapley values in the XGBoost model

Finally, the third individual is assigned to the class with disease present, although it has been chosen because the predicted probability is close to the threshold, established in 0.5.

The variable that contributes the most towards the prediction is FDG in a positive way for all methods, but we find discrepancies on the contributions of the other features. In the causal Shapley value nearly no other variable has influenced the decision. As for the marginal Shapley value, age seems to be of some relevance. Differences are found in both asymmetric Shapley values, since PTAU pushes to decrease the predicted probability but seems to have no influence when using symmetric methods. We also see that there are discrepancies in the way the rest of the variables contribute across all methods, but since they are not very relevant this can be disregarded.

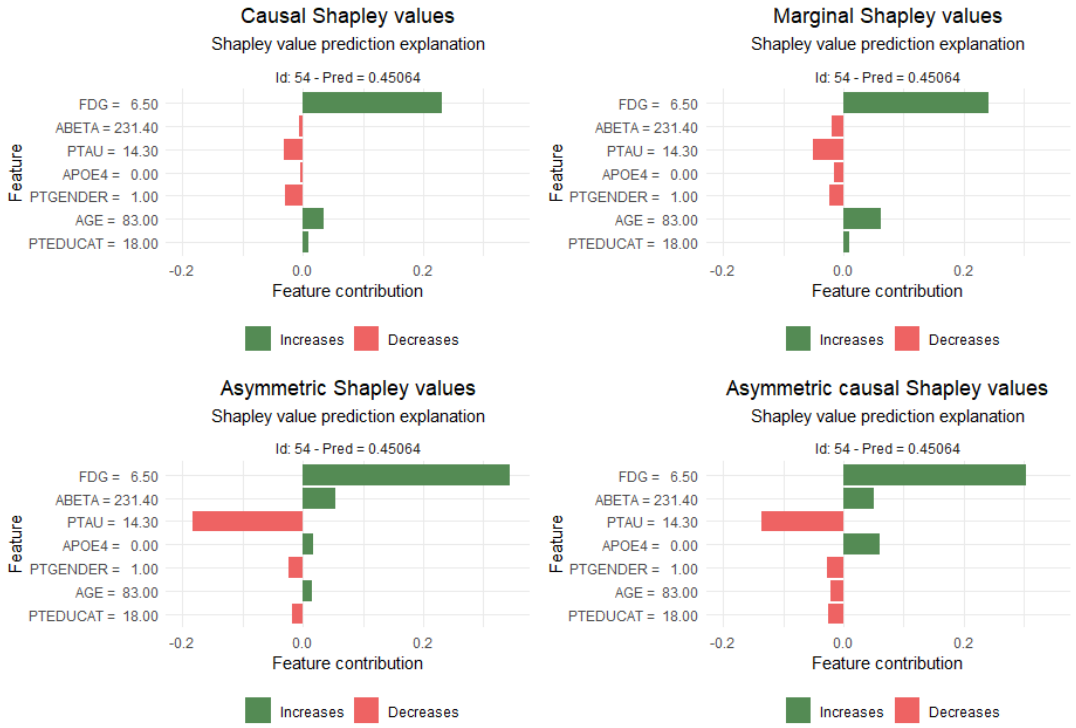

Figure 5.9: Individual plots for the different Shapley values in the XGBoost model

5.4 Differences in models

This second experiment aims at showing if there are differences in Shapley values across models. To this end, we will compare the four Shapley values dealing with the causal structure obtained before for the XGBoost model with the obtained for a logistic regression model, a type of Generalized Linear Model. The other methods are not presented since they posed computational problems.

		Predicted	
		DP	CN
Actual	DP	130	8
	CN	32	5

Table 5.10: Confusion matrix for the logistic model

In the model evaluation metrics for the logistic regression we get an accuracy of 77.1%, a sensitivity of 80.3% and a specificity of 38.5%. See also Table 5.10. The summary plots of the four Shapley values are shown in Figure 5.11

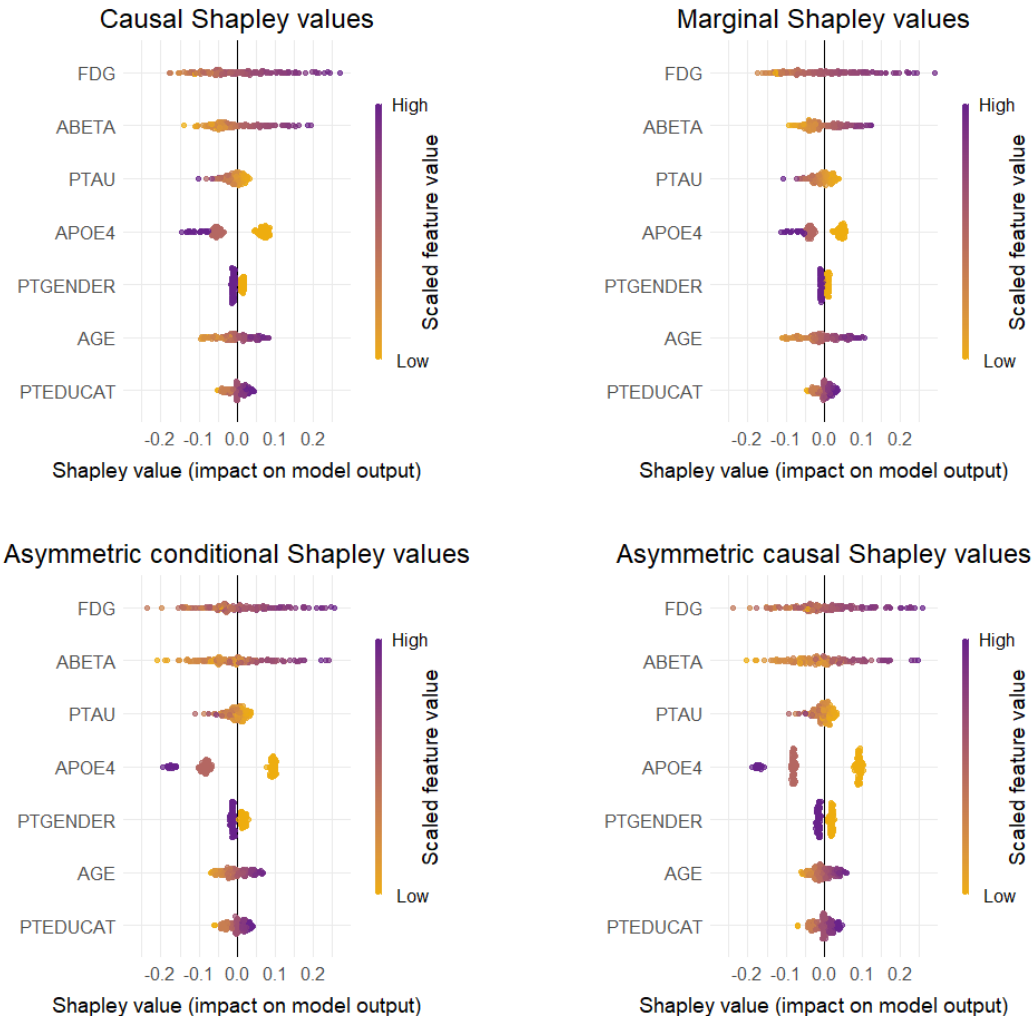

Figure 5.11: Sina plots for the different Shapley values in the logistic model

In general, we see that in this case a relationship between the Shapley values and those of the variables is best observed compared to what happened in the XGBoost model. For the categorical variables, both symmetric and asymmetric Shapley values separate the different categories, whereas this only happened in the asymmetric Shapley values of the XGBoost model. In the logistic model, the three categories of APOE4 are better differentiated in the asymmetric case. Additionally, the two genders are clearly separated in the PTGENDER variable, and all individuals corresponding to one of the categories obtain the same Shapley value in this variable.

As for the continuous variables, in the XGBoost model it was difficult to tell whether a given value of a feature would correspond to a low or high Shapley values. For some variables, such as PTAU or PTEDUCAT this was impossible. In the case of the logistic model, the PTAU variable is still difficult to read, whereas for the rest, it is intuited that low feature values will correspond to negative Shapley values and vice-versa. It is also observed that the variables were more spread out on the Shapley value axis for the XGBoost whereas in this case they are more clustered. This difference is particularly noticeable in the case of the AGE variable. Finally, within the logistic model there are no significant differences between Shapley values for continuous variables.

Chapter 6

Conclusions

In this work we have shown how Shapley values, which were born 70 years ago in the context of Game Theory, can be an useful tool when addressing a recent issue in Machine Learning, that of model explainability. They are thus the idea behind SHAP, a technique that allows local interpretability but whose results can also be extrapolated to explain models globally.

At first, SHAP values were developed assuming feature independence. In this text, we present the original technique, as well as later developments that focused on removing this assumption with the objective of getting accurate explanations efficiently. Three methods are described, which work with Gaussian distributions, Gaussian copulas and empirical conditional distributions - a non-parametric approach.

Additionally, we aimed at introducing the problem of causality. With models being increasingly used in different areas of society and taking part in decision-making processes, it is essential that they accurately reflect and model the reality of the world we live in. In that sense, we present two methods to calculate Shapley values when knowing the underlying causal struture of the variables, thus introducing causality into the realm of explainability. One of this methods relies on Pearl's axiomatic system, the *do*-calculus, whereas the other works with the symmetry property of Shapley values.

In order to illustrate these concepts, we chose the ADNI (Alzheimer's Disease Neuroimaging Initiative) dataset to carry out different experiments. The choice for this dataset was motivated by the fact that some authors had proposed a causal structure for its features, thus allowing to calculate Shapley values that need this information. We have compared seven different techniques for calculating Shapley values for the XGBoost model.

In general, it can be seen that the main discrepancies are found between symmetrical and asymmetrical Shapley values. These differences are best noticed when analysing prediction explanations for specific individuals, as it can be observed that the methods sometimes differ in highlighting the variables that contribute the most towards the prediction. Finally, it is also observed that the explanations change with the models. In this case, we have compared XGBoost with logistic regression, finding that the latter separates better the variables.

The question that arises now is how to go beyond. We have seen that taking into account the causal structure of the data changes explanations. The problem is that this causal relationships cannot be formally found directly from data so far. Some attemps have been made in this sense [29], but it seems there is still a long way to go in the quest of translating reality of the real world into that of formality.

Acknowledgements

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Bibliography

[1] Kjersti Aas, Martin Jullum, and Anders Løland. *Explaining individual predictions when features are dependent: More accurate approximations to Shapley values*. 2020. arXiv: 1903.10464 [stat.ML].

[2] Leo Breiman. "Random Forests". In: *Machine Learning* 45 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[3] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *CoRR* (2016). arXiv: 1603.02754.

[4] Harald Cramer. "Mathematical Methods of Statistics". In: Princeton, USA: Princeton University Press, 1946. Chap. The two-dimensional case, p. 282. ISBN: 0-691-08004-6.

[5] Mengnan Du, Ninghao Liu, and Xia Hu. *Techniques for Interpretable Machine Learning*. 2019. arXiv: 1808.00033 [cs.LG].

[6] Upol Ehsan et al. "Expanding Explainability: Towards Social Transparency in AI systems". In: *CoRR* (2021). arXiv: 2101.04719.

[7] Christopher Frye, Colin Rowat, and Ilya Feige. *Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability*. 2020. arXiv: 1910.06358 [stat.ML].

[8] Leilani H. Gilpin et al. *Explaining Explanations: An Overview of Interpretability of Machine Learning*. 2018. DOI: 10.48550/ARXIV.1806.00069.

[9] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI Magazine* 38.3 (2017), pp. 50–57. DOI: 10.1609/aimag.v38i3.2741.

[10] Yuval Greenfield. "Four ways teams win on Kaggle". In: *Towards Data Science* (2020). Accessed: 2022-05-07. URL: https://towardsdatascience.com/four-ways-teams-win-on-kaggle-50e62acb87f4.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. New York: Springer, 2009. ISBN: 978-0-387-84857-0. DOI: 10.1007/978-0-387-84858-7.

[12] Tom Heskes et al. *Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models*. 2020. arXiv: 2011.01625 [cs.AI].

[13] Clifford M. Hurvich and Chih-Ling Tsai. "Regression and time series model selection in small samples". In: *Biometrika* 76.2 (1989), pp. 297–307. DOI: 10.1023/A:1010933404324.

[14] Marina Krakovsky. "Solving for Why". In: *Communications of the ACM* 65.2 (Feb. 2022), pp. 11–13. DOI: 10.1145/3503777.

[15] Camilla Lingjærde, Martin Jullum, and Nikolai Sellereite. *shapr: Explaining individual machine learning predictions with Shapley values*. R package version 0.2.0. 2021. URL: https://CRAN.R-project.org/package=shapr.

[16] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. *Consistent Individualized Feature Attribution for Tree Ensembles*. 2019. arXiv: 1802.03888 [cs.LG].

[17] Scott M. Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 2017. arXiv: 1705.07874 [cs.AI].

[18] Scott M. Lundberg et al. "From local explanations to global understanding with explainable AI for trees". In: *Nat Mach Intell* 2 (2020), pp. 56–67. DOI: 10.1038/s42256-019-0138-9.

[19] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Second. United Kingdom: Chapman and Hall/CRC, 1989. ISBN: 9780412317606.

[20] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2019. URL: https://christophm.github.io/interpretable-ml-book/.

[21] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. *Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges*. 2020. arXiv: 2010.09337 [stat.ML].

[22] Roger B. Nelsen. *An Introduction to Copulas*. USA: Springer, Inc., 2006. ISBN: 978-0387-28659-4.

[23] Judea Pearl. "Causal diagrams for empirical research". In: *Biometrika* 82.4 (Dec. 1995), pp. 669–688. ISSN: 0006-3444. DOI: 10.1093/biomet/82.4.669.

[24] Judea Pearl. *The Do-Calculus Revisited*. 2012. arXiv: 1210.4852 [cs.AI].

[25] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. USA: Basic Books, Inc., 2018. ISBN: 046509760X.

[26] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *CoRR* (2016). arXiv: 1602.04938.

[27] Julia M. Rohrer. “Thinking Clearly About Correlations and Causation: Graphical Causal Models for Observational Data”. In: *Advances in Methods and Practices in Psychological Science* 1.1 (2018), pp. 27–42. DOI: 10.1177/2515245917745629.

[28] Alvin E. Roth. “Introduction to the Shapley value”. In: *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Ed. by Alvin E.Editor Roth. Cambridge University Press, 1988, pp. 1–28. DOI: 10.1017/CBO9780511528446.002.

[29] Xinpeng Shen et al. “Challenges and Opportunities with Causal Discovery Algorithms: Application to Alzheimer’s Pathophysiology”. In: *Science Reports* 10 (2020), p. 2975. DOI: 10.1038/s41598-020-59669-x.

[30] Thomas C Williams et al. “Directed acyclic graphs: a tool for causal studies in paediatrics”. In: *Pediatric Research* 84 (June 2018), pp. 487–493. DOI: 10.1038/s41390-018-0071-3.