Interpretability and Explainability in Machine Learning (IML)

1. Introduction to interpretability in machine learning

Pedro Delicado
Departament d'Estadística i Investigació Operativa and IMTech
Universitat Politècnica de Catalunya - Barcelona TECH

eBISS 2022. July 4-8, 2022, Cesena, Italy

Course material

- https://pedrodelicado.moodlecloud.com/
- Username: estudent
- Password: eBISS2022

Outline of the course

1. Introduction to interpretability in machine learning
2. Interpretability methods for specific models
3. Model-agnostic interpretability methods
    - Global methods
    - Local methods

Introduction to model interpretability and variable relevance

- In a stimulating and provocative paper, Breiman (2001)[1] shook the statistical community by making it to be aware that traditional Statistics was no longer the only way to learn from data:
  - *Data Modeling Culture* (traditional Statistics):
    - Linear regression, logistic regression, additive models, etc.
    - They allow to interpret how the response variable is associated with the input variables: **Transparent models**.
  - *Algorithmic Modeling Culture* (Machine Learning, Data Science):
    - Neural networks, support vector machines, random forest, etc.
    - They have extremely good predictive accuracy, and they usually outperform in this criterion statistical models.
    - Low interpretability: **Black boxes**.
- An apparent dichotomy: predictive capacity versus interpretability.
- Breiman claimed for procedures allowing better interpretation of the algorithmic models results, without giving up their predictive ability.

[1] Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science 16*, 199–231.

A real data example: Rent housing prices

- Data on rental housing in Spain, downloaded from Idealista.com on February 27th, 2018, by Alejandro German (Alex seralexger).
- Data available at `https://github.com/seralexger/idealista-data`
- Original data set: 67201 rows (advertisements) and 19 attributes. All cities in Spain.
- We have selected Madrid and Barcelona: 16480 rows.
- Training set 70%, test set 30%.
- Response variable: logarithm of the rental price.
- We work with 16 explanatory variables (some of them calculated from the original data).

```
##  [1] "price"                            "Barcelona"
##  [3] "categ.distr"                      "type.chalet"
##  [5] "type.duplex"                      "type.penthouse"
##  [7] "type.studio"                      "floor"
##  [9] "hasLift"                          "floorLift"
## [11] "size"                             "exterior"
## [13] "rooms"                            "bathrooms"
## [15] "hasParkingSpace"                  "isParkingSpaceIncludedInPrice"
## [17] "log_Days_since_first_activation"
```

```
## lm(formula = log(price) ~ ., data = rhBM.price[Itr, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72437 -0.17604 -0.02316  0.15692  1.45330
##
## Coefficients:                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           3.8169658  0.0344596 110.766  < 2e-16 ***
## Barcelona                             0.1126307  0.0052554  21.431  < 2e-16 ***
## categ.distr                           0.1169468  0.0033806  34.593  < 2e-16 ***
## type.chalet                          -0.0846942  0.0203106  -4.170 3.07e-05 ***
## type.duplex                          -0.0177992  0.0151519  -1.175  0.24013
## type.penthouse                        0.0428160  0.0101282   4.227 2.38e-05 ***
## type.studio                          -0.0762350  0.0139991  -5.446 5.27e-08 ***
## floor                                 0.0128181  0.0009696  13.220  < 2e-16 ***
## hasLift                               0.0480363  0.0118432   4.056 5.02e-05 ***
## floorLift                            -0.0013898  0.0044109  -0.315  0.75270
## log.size                              0.6186668  0.0090654  68.245  < 2e-16 ***
## exterior                             -0.0372539  0.0068935  -5.404 6.64e-08 ***
## rooms                                -0.0501949  0.0034204 -14.675  < 2e-16 ***
## bathrooms                             0.1431973  0.0047167  30.359  < 2e-16 ***
## hasParkingSpace                      -0.0074934  0.0129971  -0.577  0.56426
## isParkingSpaceIncludedInPrice        -0.0408757  0.0138863  -2.944  0.00325 **
## log_Days_since_first_activation       0.0418803  0.0018552  22.574  < 2e-16 ***
## ---
##
## Residual standard error: 0.2647 on 11519 degrees of freedom
## Multiple R-squared:  0.7602, Adjusted R-squared:  0.7599
## F-statistic:  2282 on 16 and 11519 DF,  p-value: < 2.2e-16
```

Neutral network fit

- Tuning parameters `size` and `decay` are chosen using `caret`.
- `size` in `c(10,15,20)`, `decay` in `c(0,.1,.3,.5)`.

```
# > nnet.logprice
#
# a 16-10-1 network with 181 weights
#
# inputs: Barcelona categ.distr type.chalet type.duplex type.penthouse type.studio
# floor hasLift floorLift log.size exterior rooms bathrooms hasParkingSpace
# isParkingSpaceIncludedInPrice log_Days_since_first_activation
#
# output(s): .outcome
# options were - linear output units  decay=0.5


# > 1-mean(nnet.logprice$residuals^2)
# [1] 0.8009131
```

Interpretable Machine Learning (IML), eXplainable Artificial Intelligence (XAI)

- Machine learning community has been worried about interpretability: *if the users do not trust a model or a prediction, they will not use it* (Ribeiro, Singh, and Guestrin 2016)
- In 2018 the General Data Protection Regulation of the European Union established the users' **right to explanation**: *when an algorithmic decision significantly affects a user, he or she has the right to ask for an explanation of such a decision*.
- A powerful research line has been developed: Interpretable Machine Learning, eXplainable Artificial Intelligence.
- A search query in the Web of Science (November 5th, 2021) with the terms "explainable artificial intelligence", "explainable machine learning" or "interpretable machine learning" found a total of 5673 publications, 51% of them published in 2020 or later.
- In Scopus, 7465 publications where found, 80% of which $\geq$ 2020.
- Several review papers (Barredo-Arrieta et al. 2020 is one of the most recent and extensive reviews).
- Three monographs: Molnar (2019), Biecek and Burzykowski (2021) and Masís (2021).

1. Introduction to model interpretability and variable relevance
   - Supervised learning
     - The regression problem
2. IML/XAI concepts

Supervised Learning (the prediction problem)

- Let $(\mathbf{X}, Y)$ be a r.v. with support $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \mathbb{R}$.
- General supervised learning or prediction problem:
  - Training sample: $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, i.i.d. from $(\mathbf{X}, Y)$.
  - The goal is to define a function (possibly depending on the sample) $h_S : \mathcal{X} \mapsto \mathcal{Y}$ such that for a new independent observation $(\mathbf{x}_{n+1}, y_{n+1})$, from which we only know $\mathbf{x}_{n+1}$, it happens that

$$\hat{y}_{n+1} = h_S(\mathbf{x}_{n+1}) \text{ is close to } y_{n+1} \text{ (in some sense).}$$

  - Function h_S is called generically prediction function (or classification function or regression function, depending on the case).
- We say that we have a problem of binary classification (or discrimination) when $\mathcal{Y} = \{0, 1\}$ (you can also use $\mathcal{Y} = \{-1, 1\}$).
- The problem of classification in K classes arises when $\mathcal{Y} = \{1, \dots, K\}$ (or $\mathcal{Y} = \left\{\mathbf{y} \in \{0, 1\}^K : \sum_{k=1}^K y_k = 1\right\}$).
- When $\mathcal{Y} = \mathbb{R}$ (or $\mathcal{Y}$ is an interval) we have a standard regression problem.

Loss function, risk, Bayes rule

- The (lack of) closeness between $h(\mathbf{X})$ and Y is usually measured by a **loss function** $L(Y, h(\mathbf{X}))$.
- For instance, the *squared error loss* is $L(Y, h(\mathbf{X})) = (Y - h(\mathbf{X}))^2$.
- $L(Y, h(\mathbf{X}))$ is a r.v., with expected value $R(h) = \mathbb{E}(L(Y, h(\mathbf{X})))$, called **risk** or **expected loss**, that only depends on h.
- **Decision problem:** To find the prediction function $h : \mathcal{X} \mapsto \mathcal{Y}$ that minimizes the expected loss.
- The optimal prediction function is **the Bayes rule**

$$h_B(\mathbf{x}) = \arg\min_{y \in \mathcal{Y}} \mathbb{E}(L(Y, y)|\mathbf{X} = \mathbf{x}).$$

The regression problem

- Let $(\mathbf{X}, Y)$ be a $(p+1)$-dimensional random variable. We consider the regression problem: To predict Y from known values of $\mathbf{X}$.
- The most common and convenient loss function is the squared error loss: $L(Y, h(\mathbf{X})) = (Y - h(\mathbf{X}))^2$.
- The expected loss is known as Prediction Mean Squared Error, (PMSE):
$$\text{PMSE}(h) = \mathbb{E}\left((Y - h(\mathbf{X}))^2\right).$$
- The Bayes rule in this case is
$$h_B(\mathbf{x}) = \arg\min_{y \in \mathcal{Y}} \mathbb{E}\left((Y - y)^2 | \mathbf{X} = \mathbf{x}\right) = \mathbb{E}(Y | \mathbf{X} = \mathbf{x}),$$
also known as regression function of Y over $\mathbf{x}$ and denoted by $m(\mathbf{x})$.

- Parametric regression models assume that $m(\mathbf{x})$ is known except for a finite number of unknown parameters,
$$m(\mathbf{x}) \equiv m(\mathbf{x}; \theta),\ \theta \in \boldsymbol{\Theta} \subseteq \mathbb{R}^q,$$
- For instance, the multiple linear regression model postulates that $m(\mathbf{x}) = \beta_0 + \mathbf{x}^{\mathsf{T}}\boldsymbol{\beta}_1$, with unknown parameters $\beta_0 \in \mathbb{R}$, $\boldsymbol{\beta}_1 \in \mathbb{R}^p$.
- The training sample, $S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, i.i.d. from $(\mathbf{X}, Y)$, is used to estimate the parameter θ.
- In this case $h_S(\mathbf{x}) = m(\mathbf{x}; \hat{\theta})$, where $\hat{\theta} = \hat{\theta}(S)$ is the estimation of θ from sample S.

Least squares estimation

- In this context the usual way to estimate θ is by least squares (LS):

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} (y_i - m(\mathbf{x}_i;\theta))^2.$$

- This is equivalent to the maximum likelihood estimation of θ if $(\mathbf{X}, Y)$ is assumed to have a joint normal distribution.
- In this case:
  - The regression function $m(\mathbf{x})$ is linear in $\mathbf{x}$.
  - It is equivalent to state the model as

$$Y = m(\mathbf{X}) + \varepsilon,$$

where ε is an additive noise normally distributed with zero mean and independent from $\mathbf{X}$, also normally distributed.

The nonparametric regression model

- We observe n pairs of data $(\mathbf{x}_i, y_i)$ coming from the nonparametric regression model

$$y_i = m(\mathbf{x}_i) + \varepsilon_i, \; i = 1, \ldots, n,$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are independent r.v. with

$$E(\varepsilon_i) = 0, V(\varepsilon_i) = \sigma^2 \text{ for all } i,$$

and the predicting variable values $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are known.
- The functional form of the regression function $m(\mathbf{x})$ is not specified.
- Certain regularity conditions on $m(\mathbf{x})$ could be assumed. For instance, it is usually assumed that $m(\mathbf{x})$ has continuous second derivatives.

What does it mean “to fit a nonparametric regression model”?

- To provide an estimator $\hat{m}(\mathbf{t})$ of $m(\mathbf{t})$ for all $\mathbf{t} \in \mathbb{R}^p$.
  - This implies to give an algorithm that computes $\hat{m}(\mathbf{t})$ for any input value $\mathbf{t} \in \mathbb{R}^p$.
    - Statistical nonparametric regression estimators: local averages (kernel regression, k nearest neighbors), local polynomial regression, spline smoothing, (generalized) additive models, CART (Classification and Regression Trees), ...
    - Machine learning prediction models: Neural networks, support vector machines, ensemble meta-algorithm (random forest, XGBost, ...), ...
    - In both cases, the algorithm uses the information contained in the observed sample S. The algorithm itself is $h_S(\mathbf{t}) = \hat{m}(\mathbf{t})$.
  - For the particular case of only one explanatory variable, usually the graphic of the pairs $(t_j, \hat{m}(t_j))$, $j = 1, \ldots, J$, is drawn, where t_j, $j = 1, \ldots, J$ is a regular fine grid covering the range of the observed values x_i, $i = 1, \ldots, n$.
- To give an estimator $\hat{\sigma}^2$ of the residual variance σ^2.

Example: k nearest-neighbors

- The k nearest-neighbor estimator of $m(\mathbf{t}) = E(Y|\mathbf{X} = \mathbf{t})$ is defined as

$$\hat{m}(\mathbf{t}) = \frac{1}{k} \sum_{i \in N_k(\mathbf{t})} y_i,$$

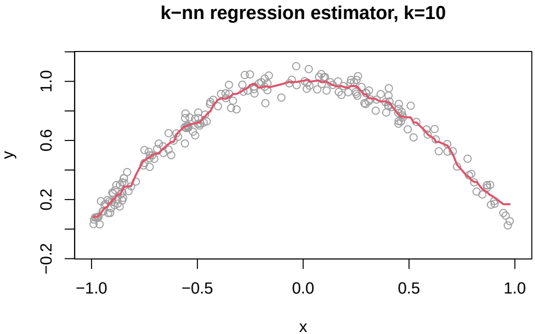
  where $N_k(\mathbf{t})$ is the neighborhood of $\mathbf{t}$ defined by the k closest points $\mathbf{x}_i$ in the training sample.
- *Closeness* is defined according to a previously chosen distance measure $d(\mathbf{t}, \mathbf{x})$, for instance, the Euclidean distance.

k-nn regression, in R

```
knn_regr<- function(x, y, t=NULL, k=3,
dist.method = "euclidean"){
  nx <- length(y)
  if (is.null(t)){
    t<- as.matrix(x)
  }else{
    t<-as.matrix(t)
  }
  nt <- dim(t)[1]
  Dtx <- as.matrix( dist(rbind(t,as.matrix(x)),
                         method = dist.method) )
  Dtx <- Dtx[1:nt,nt+(1:nx)]
  mt <- numeric(dim(t)[1])
  for (i in 1:length(mt)){
    d_t_x <- Dtx[i,]
    d_t_x_k <- sort(d_t_x,partial=k)[k]
    N_t_k <- unname( which(d_t_x <= d_t_x_k) )
    mt[i]=mean(y[N_t_k])
  }
  return(mt)
}
```

Example of *k*-nn regression

```
n <- 200; sd_eps <- .05
x <- sort(2*runif(n)-1)
mx <- 1-x^2
eps <- rnorm(n,0,sd_eps)
y <- mx+eps
plot(x,y,xlim=c(-1,1),ylim=c(-3*sd_eps,1+3*sd_eps),col=8)
k <- n/20
hat_mx <- knn_regr(x,y,k=k)
lines(x,hat_mx,col=2,lwd=2)
title(main=paste0("k-nn regression estimator, k=",k))
```



Attention! The borders are less and less clear:

- **Parametric models - Nonparametric models.**
  Example: lasso for $p >> n$.
  The estimation of parameter θ is done by penalized least squares:

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} (y_i - m(\mathbf{x}_i;\theta))^2 + \lambda\,\mathrm{Penalty}(\theta),$$

  for a pre-chosen $\lambda > 0$ and a given penalty function $\mathrm{Penalty}(\theta)$.
  Example: A one-hidden layer neural network is, in fact, a parametric regression model with a very large number of parameters (the connection weights).
- **Statistical models - Machine learning models.**
  Example: Random forests.

1. Introduction to model interpretability and variable relevance
   - Supervised learning
     - The regression problem
2. IML/XAI concepts

IML/XAI concepts

- Desirable properties for predictive models: transparency, interpretability, explainability.
- Not well defined concepts that are difficult to be measured.
    - Lipton 2018: *The term interpretability is ill-defined*.
    - Barredo-Arrieta et al. 2020: *The derivation of general metrics to assess the quality of XAI approaches remain as an open challenge*.

Transparency, interpretability, explainability

- An algorithm is said to be transparent if the mechanism by which it works can be understood by a human (Lipton 2018, Barredo-Arrieta et al. 2020).
- This definition encompasses different degrees of algorithm transparency, given the wide range of human expertise (Lipton 2018).
- The concept of interpretability is both *important and slippery*, as acknowledge by Lipton (2018), who mentions that a general *goal of interpretability might simply be to get more useful information from the model*.
- Barredo-Arrieta et al. (2020) consider than transparency and interpretability are synonymous in this context, and that what is relevant is what information you want to extract from a model and how to get it.
- Miller (2019) gives a slightly different sense to interpretability, equating this term to explainability with the meaning of *how well a human could understand the decisions in the given context*, that is, the ability of an algorithm to provide humans an explanation for any of its particular decisions.

- We can summarize telling that in the last years the quality perception of a prediction algorithm is no longer focused exclusively on the accuracy of predictions.
- In addition to that, the possibility of obtaining information on the performance of the algorithm, in both the *global* and *local* sense, is now appreciated.

Global versus local interpretability

- *Information about the global performance* refers to determining which is the role of each explanatory variable in the prediction process over the whole support of the explanatory variables.
    - **Global interpretability:** Measures of variable importance or relevance.
- On the other hand, the goal of understanding *local performance* is to provide a meaningful explanation of why the algorithm returns a certain prediction, given a particular combination of the predicting variables values.
    - **Local interpretability:** Why the prediction model does a particular prediction for a given individual?
- The *local* aspect of interpretability is directly related with the users' *right to explanation* advocated for by, for instance, the EU's GDPR.

Transparent models versus black-box modeels

According Barredo-Arrieta et al. (2020) and Maksymiuk, Gosiewska, and Biecek (2020), the prediction models can be classified as follows:

1. *Transparent models*, or *interpretable by design models*, or *white-boxes*, or *glass-boxes*.
   - Models that, by design, have an easily interpretable structure.
   - Linear models (LM, and generalized linear models, GLM), generalized additive models (GAM, including additive models), classification and regression trees (CART), decision rules, k-nearest neighbors, and Bayesian models (including Naïve Bayes prediction rules).
   - They offer sufficient interpretation and/or diagnostic tools, both numeric and graphic.
2. *Non-transparent models* or *black-box models*.
   - Their design does not provide a directly interpretable structure.
   - These models require additional interpretation tools.
   - All the prediction methods not explicitly mentioned before.

Non-transparent models can be divided into two subgroups:

(a) Models for which there exist *model-specific* methods for knowledge extraction:
   - Tree ensembles (including random forests and boosted methods).
   - Neural networks (NN, including deep learning based on multi-layer, recurrent or convolutional NN).
   - Support vector machines (SVM).

   Model-specific methods require full access to the model structure.

(b) The rest of the models:
   - Only *model-agnostic* methods are available for interpretation.
     - Do not need to know the internal structure of the prediction model to be explained.
     - Only requirement: the ability to evaluate the prediction model repeated times on data from the training or the test set, or on perturbations of them.
     - They can be applied to any predictive model, even to those having model-specific methods or those that are transparent models.

- All the interpretation methods applicable to non-transparent prediction models are globally known as *post-hoc interpretation methods*, a term that encompasses model-specific as well as model-agnostic methods.
- The results provided by these methods can be numerical and graphical, although most of the methods choose one or the other format.
- Finally, it is worth mentioning that most of the interpretation methods are heuristic, and only some of them are derived from a formal axiomatic statement.

Classification of models and interpretability tools

<table>
<tr><th>Transparent models</th><th>Black-boxes: Post-modeling interpretability</th></tr>
<tr><td>Linear model (LM)
GLM
GAM
CART
Rule based models
Naïve Bayes
k-nearest neighbours</td><td>

Model-specific methods:

- Tree ensembles
- Neural networks
- Support vector machines

Model-agnostic methods:

Global measures

- Variable importance by
  - Leave-one-covariate-out (LOCO)
  - Perturbing a variable in the test set: Random permutations, knockoffs, **Ghost-variables**, ...
- Variable importance based on Shapley's value
- Partial dependence plot (PDP)
- Accumulated local effects plot (ALE)

Local measures

- Local interpretable model-agnostic explanations (LIME)
- Local variable importance based on Shapley's value
- SHAP (SHapley Additive exPlanations)
- Break-down plots
- Individual conditional expectation (ICE) plot, or ceteris paribus plot

</td></tr>
</table>

Books on IML/XAI: Molnar (2019)

- Molnar (2019) offers a broad overview of techniques aimed at making machine learning models and their decisions interpretable.
- The concepts of interpretability are explored first.
- Then interpretable models (including linear and additive models, decision trees, and decision rules) are covered.
- Later, general model-agnostic methods for interpreting black-box models are introduced.
- An additional chapter is devoted to neural network interpretability.
- Three real data sets are used throughout the book to present the explained methods.
- At the end of the book, the R packages used for examples are listed, among which `iml` package should be highlighted (not for nothing Molnar is one of the authors of `iml`).

Books on IML/XAI: Biecek and Burzykowski (2021)

- In Biecek and Burzykowski (2021), the authors focus on model-agnostic techniques. They do not assume anything about the structure of the model.
- The only interaction allowed with the fitted model is its evaluation on a specific data set.
- There are two main parts in the book: one devoted to *instance-level exploration* (or local interpretability), and the other to *dataset-level exploration* (or global variable relevance).
- Every interpretability method in the book is introduced first at an intuitive level, and then its mathematical and computational aspects are presented in detail.
- The examples throughout the book are based on three real data sets.
- Additionally, a detailed full case study is presented in the last chapter of the book.
- All the methods presented in the book are available in both R (`DALEX` package) and Python (`dalex` library).
- The code for reproducing the examples is also available.

Books on IML/XAI: Masís (2021)

- Masís (2021) is structured in 3 sections, each including several chapters.
- Section 1 gives an introduction to machine learning interpretation, stating the key concepts and presenting several real data examples which are used through the book.
- In Section 2 the main interpretation methods are covered: global and local model-agnostic methods, counterfactual explanations, and visual interpretation methods for convolutional neural networks.
- Finally, Section 3 is devoted to more specific and technical interpretability issues.
- The book is practice oriented. Each chapter offers the reader the Python code to reproduce step-by-step the analysis and figures included in the book.
- A github repository (`https://github.com/PacktPublishing/Interpretable-Machine-Learning-with-Python`) contains the example code files for the book, ready to be downloaded.

Barredo-Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. (2020).
Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.
Information Fusion 58, 82–115.

Biecek, P. and T. Burzykowski (2021).
Explanatory model analysis: Explore, explain and examine predictive models.
Chapman and Hall/CRC.

Breiman, L. (2001).
Statistical modeling: The two cultures.
Statistical Science 16, 199–231.

Lipton, Z. C. (2018).
The mythos of model interpretability.
Queue 16(3), 31–57.

Maksymiuk, S., A. Gosiewska, and P. Biecek (2020).
Landscape of R packages for explainable artificial intelligence.
arXiv:2009.13248.

Masís, S. (2021).
Interpretable Machine Learning with Python.
Packt Publishing Ltd.

Miller, T. (2019).
Explanation in artificial intelligence: Insights from the social sciences.
Artificial Intelligence 267, 1 – 38.

Molnar, C. (2019).
Interpretable Machine Learning.
Lulu. com.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016).
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*, pp. 1135–1144. ACM.