Interpretability and Explainability in Machine Learning (IML)

2. Interpretability methods for specific models

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

MESIO UPC-UB Summer School
Barcelona, 20-23 June 2022

1. Introduction to model-specific interpretability methods

2. Interpretability in Random forests
   - A brief review of CART and Random Forests
     - Regression trees
     - Classification trees
     - Random forests
   - Variable Importance based on node impurity measures
   - Out-of-Bag Variable Importance

3. Interpretability in Neural networks
   - Neural Networks: A brief review
   - Interpretability in Neural Networks

1. Introduction to model-specific interpretability methods

2. Interpretability in Random forests
   - A brief review of CART and Random Forests
     - Regression trees
     - Classification trees
     - Random forests
   - Variable Importance based on node impurity measures
   - Out-of-Bag Variable Importance

3. Interpretability in Neural networks
   - Neural Networks: A brief review
   - Interpretability in Neural Networks

Introduction to model-specific interpretability methods

- Model-specific methods:
  - Interpretability methods developed for a particular prediction method.
  - They allow model exploration, validation, or visualization.
  - They require full access to the model structure.
  - Different prediction models have different model-specific interpretability methods, usually difficult to be compared.
- We are talking here about interpretability in
  - random forests,
  - neural networks.
- For support vector machines, we refer to Section 4.2.2 in Barredo-Arrieta et al. (2020).

1 Introduction to model-specific interpretability methods

2 Interpretability in Random forests
- A brief review of CART and Random Forests
  - Regression trees
  - Classification trees
  - Random forests
- Variable Importance based on node impurity measures
- Out-of-Bag Variable Importance

3 Interpretability in Neural networks
- Neural Networks: A brief review
- Interpretability in Neural Networks

Interpretability in Random forests

- Random forests are combinations of more simple models: classification and regression trees (CART).
- CART are usually considered *transparent models* because the prediction rules they encode are easily understood by non-expert users.
- Additionally, a simple importance measure for the input variables can be defined for CART:
    - At each split in the tree, the improvement in the split-criterion is the importance measure attributed to the splitting variable.
- In random forests, this importance measure is accumulated over all the trees in the forest separately for each variable.
- Breiman (2001) introduced an alternative way to measure the variable importance in random forests, combining the use of the *out-of-bag* samples as test samples, and the principle of randomly permuting the values of each predictor in a test sample to measure the decrease in accuracy.

1 Introduction to model-specific interpretability methods

2 Interpretability in Random forests
- A brief review of CART and Random Forests
  - Regression trees
  - Classification trees
  - Random forests
- Variable Importance based on node impurity measures
- Out-of-Bag Variable Importance

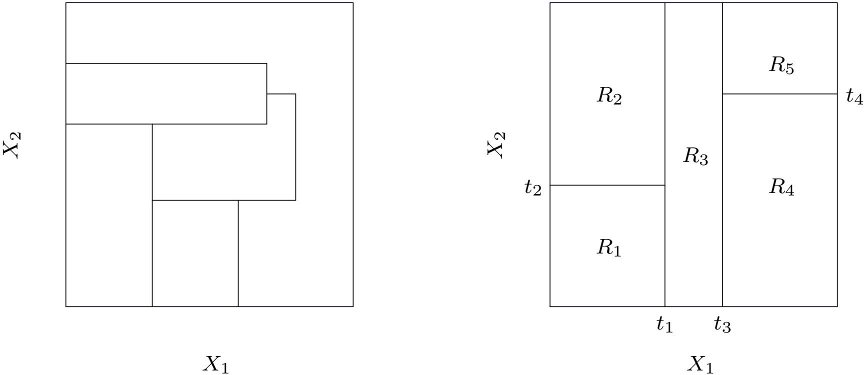
3 Interpretability in Neural networks
- Neural Networks: A brief review
- Interpretability in Neural Networks

A brief review of CART and Random Forests[1]

- Tree-based methods divide the feature space into a set of regions, and then fit a simple model (like a constant) at each one.
- They are conceptually simple yet powerful.
- Example:
  - Consider a regression problem with continuous response Y and inputs X_1 and X_2, each taking values in the unit interval.
  - Let $\{R_1, \dots, R_5\}$ be a partition of the unit square into 5 regions.
  - The corresponding regression model predicts Y with a constant c_m in region R_m, that is,

$$\hat{f}(x_1, x_2) = \sum_{m=1}^{5} c_m I_{R_m}(x_1, x_2).$$

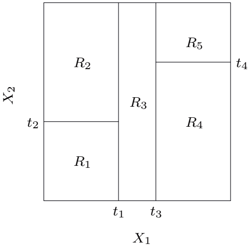
[1] We follow Section 9.2 and Chapter 15 in Hastie, Tibshirani, and Friedman (2009), and Chapter 8 in James, Witten, Hastie, and Tibshirani (2013).



Two examples of partitions of the unit square into 5 regions.

- In the left panel some of the regions are complicated to describe.
- In the right panel the rectangles $\{R_1, \ldots, R_5\}$ have been obtained by recursive binary partitions, that can be easily be represented by a binary tree.

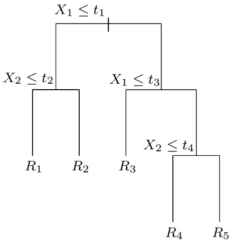
Source of graphics: Hastie, Tibshirani, and Friedman (2009).

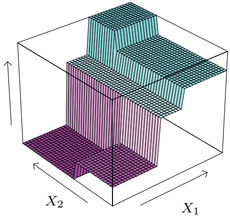


Bottom left panel shows the tree corresponding to the partition in the top left panel, and a perspective plot of the prediction surface

$$\hat{f}(x_1, x_2) = \sum_{m=1}^{5} c_m I_{R_m}(x_1, x_2).$$

appears in the bottom right panel.





Source of graphics: Hastie, Tibshirani, and Friedman (2009).

Regression trees

- Our data set consists of p inputs and a response, for each of n observations: $(\mathbf{x}_i, y_i)$, $i = 1, 2, \ldots, n$, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\mathsf{T}$.
- The algorithm needs to automatically decide on the splitting variables and split points.
- Suppose first that we have a partition into M regions $R_1, \ldots, R_M$, and we model the response as a constant c_m in each region:

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m I_{R_m}(\mathbf{x}).$$

- If we adopt as our criterion the minimization of the sum of squares, $\sum_{i=1}^{n}(yi - f(\mathbf{x}_i))^2$, then the best value for c_m is just the average (ave) of y_i in region R_m: $\hat{c}_m = \text{ave}(y_i|\mathbf{x}_i \in R_m)$.

Finding the best partition

- Finding the best binary partition in terms of minimum sum of squares is generally computationally infeasible.
- So a greedy strategy is adopted.
- Starting with all of the data, consider a splitting variable j and split point s, and define the pair of half-planes

$$R_1(j,s) = \{\mathbf{x} \in \mathbb{R}^p : x_j \leq s\} \text{ and } R_2(j,s) = \{\mathbf{x} \in \mathbb{R}^p : x_j > s\}.$$

- Then we seek the splitting variable j and split point s that solve

$$\min_{j,s} \left\{ \min_{c_1} \sum_{\mathbf{x}_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{\mathbf{x}_i \in R_2(j,s)} (y_i - c_2)^2 \right\}.$$

- For any choice j and s, the inner minimization is solved by

$$\hat{c}_1 = \text{ave}(y_i|\mathbf{x}_i \in R_1) \text{ and } \hat{c}_2 = \text{ave}(y_i|\mathbf{x}_i \in R_2).$$

- For each splitting variable, the determination of the split point s can be done very quickly and hence by scanning through all of the inputs, determination of the best pair (j, s) is feasible.
- Having found the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions.
- Then this process is repeated on all of the resulting regions.
- The squared-error impurity measure for the m-th region (or *node*) R_m, with N_m cases and average response $\hat{c}_m = \frac{1}{N_m}\sum_{\mathbf{x}_i \in R_m} y_i$, is defined as

$$Q_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \hat{c}_m)^2.$$

Classification trees

- Assume now that the target is a classification outcome taking values $1, \dots, K$.
- The only changes needed in the tree algorithm affect the criteria for splitting nodes and pruning the tree.
- In a node m, representing a region R_m with N_m observations, let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} I(y_i = k),$$

  the proportion of class k observations in node m.
- We classify the observations in node m to class $k(m) = \arg\max_k \hat{p}_{mk}$, the majority class in node m.

- The squared-error node impurity measure $Q_m(T)$ used for regression is not suitable for classification.
- Different measures $Q_m(T)$ of node impurity include the following:

Misclassification error:	$(1/N_m)\sum_{\mathbf{x}_i \in R_m} I(y_i \neq k) = 1 - \max_k(\hat{p}_{mk})$.
Gini index:	$\sum_{k \neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$.
Cross-entropy or deviance:	$\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$.

- Both the Gini index and cross-entropy are lower for the splits producing pure nodes, that are probably preferable.
- For this reason, either the Gini index or cross-entropy should be used when growing the tree.
- To guide cost-complexity pruning, any of the three measures can be used, but typically the misclassification rate is used.

Instability of Trees

- One major problem with trees is their high variance.
- Often a small change in the data can result in a very different series of splits, making interpretation somewhat precarious.
- The major reason for this instability is the hierarchical nature of the process: the effect of an error in the top split is propagated down to all of the splits below it.
- Instability is the price to be paid for estimating a simple, tree-based structure from the data.
- Bagging (Bootstrap aggregation; see, for instance, Section 8.7 in Hastie, Tibshirani, and Friedman 2009) averages B trees to reduce this variance.
- Random forest is another way to average trees to reduce instability, that in many problems outperforms bagging.

Random forests

- In random forests, a large amount of random trees is generated and then they are averaged.
- It is hopped to reduce variance without increments in bias.
- The first random ingredient: take a bootstrap sample, choosing with replacement n random elements from the original data set:

$$\{(\mathbf{x}_i^*, y_i^*), i = 1, \ldots, n\} \text{ randomly chosen from } \{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}.$$

- Several real data appear more than once in the bootstrap sample (around 2/3 of them).
- Other (around 1/3) do not belong to the bootstrap sample: they are called the *out-of-bag sample (OOB)*.
- This idea is shared with bagging (Bootstrap aggregation), but random forest try to reduce the correlation between the trees, without increasing the variance too much.
- This is achieved in the tree-growing process through random selection of the input variables.

Out-of-Bag Samples

- An important feature of random forests is its use of out-of-bag (OOB) samples:

  > *For each observation* $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ *in the training set, construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which* $\mathbf{z}_i$ *did not appear.*

- An OOB error estimate is almost identical to that obtained by *n*-fold cross-validation.
- Hence unlike many other nonlinear estimators, random forests can be fit in one sequence, with cross-validation being performed along the way.
- Once the OOB error stabilizes, the training can be terminated.

1 Introduction to model-specific interpretability methods

2 Interpretability in Random forests

- A brief review of CART and Random Forests
  - Regression trees
  - Classification trees
  - Random forests
- Variable Importance based on node impurity measures
- Out-of-Bag Variable Importance

3 Interpretability in Neural networks

- Neural Networks: A brief review
- Interpretability in Neural Networks

Variable Importance based on node impurity measures

- Let T be a tree found in the fitting process of a CART, and let $|T|$ be the number of terminal nodes in T.
- The total impurity measure of T is defined as

$$C(T) = \sum_{m=1}^{|T|} N_m Q_m.$$

- Assume that the j-th variable is used to split the node R_r of T is into two child-nodes, say $R_{r'}$ and $R_{r''}$, this way producing a new tree T' from T.
- The improvement in the impurity measure (also known as the improvement in the split-criterion) is

$$C(T) - C(T') = N_r Q_r - (N_{r'} Q_{r'} + N_{r''} Q_{r''}) .$$

- A simple importance measure for the input variables can be defined for **CART**:
  - At each split in the tree, the improvement in the split-criterion is attributed to the splitting variable as a partial measure of its importance.
  - The importance measure of a variable is the sum of the partial measures of importance corresponding to all splits defined by this variable.
- In **Random Forests**, this importance measure is accumulated over all the trees in the forest separately for each variable:
  - At each split in each tree, the improvement in the split-criterion is the importance measure attributed to the splitting variable, and is accumulated over all the trees in the forest separately for each variable.
  - The importance of each variable in the random forest is obtained by averaging the decrease in accuracy over all trees conforming the forest.

1 Introduction to model-specific interpretability methods

2 Interpretability in Random forests

- A brief review of CART and Random Forests
  - Regression trees
  - Classification trees
  - Random forests
- Variable Importance based on node impurity measures
- Out-of-Bag Variable Importance

3 Interpretability in Neural networks

- Neural Networks: A brief review
- Interpretability in Neural Networks

Out-of-Bag Variable Importance

Breiman (2001) introduced an alternative way to measure the variable importance (or prediction strength) in random forests, combining the use of the *out-of-bag* (OOB) samples as test samples, and the principle of randomly permuting the values of each predictor in a test sample to measure the decrease in accuracy.

- When the b-th tree is grown, the OOB samples are passed down the tree, and the prediction accuracy is recorded:

$$SSE_{oob}^{b} = \sum_{i \in oob_b} \left(y_i - T^b(\mathbf{x}_i) \right)^2 .$$

- Then the values for the j-th variable are randomly permuted in the OOB samples, and the accuracy is again computed:

$$SSE_{oob,\pi(j)}^{b} = \sum_{i \in oob_b} \left(y_i - T^b(\mathbf{x}_{i\{\pi(j)\}}) \right)^2 .$$

- The decrease in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable j in the random forest.

$$\mathrm{VI}_{oob}(j) = \frac{1}{B} \sum_{b=1}^{B} \left(SSE_{oob,\pi(j)}^{b} - SSE_{oob}^{b} \right)$$

- The randomization effectively voids the effect of a variable, much like setting a coefficient to zero in a linear model.
- This does not measure the effect on the prediction if this variable were not available, because if the model was refitted without the variable, other variables could be used as surrogates.

Practice:
Washington D.C. Bike Sharing Dataset

Follow the point
1. Fit a Random Forest
in the R-markdown file
`eBISS_IML_bike_sharing_data.Rmd`.

1 Introduction to model-specific interpretability methods

2 Interpretability in Random forests
- A brief review of CART and Random Forests
  - Regression trees
  - Classification trees
  - Random forests
- Variable Importance based on node impurity measures
- Out-of-Bag Variable Importance

3 Interpretability in Neural networks
- Neural Networks: A brief review
- Interpretability in Neural Networks

1. Introduction to model-specific interpretability methods

2. Interpretability in Random forests
   - A brief review of CART and Random Forests
     - Regression trees
     - Classification trees
     - Random forests
   - Variable Importance based on node impurity measures
   - Out-of-Bag Variable Importance

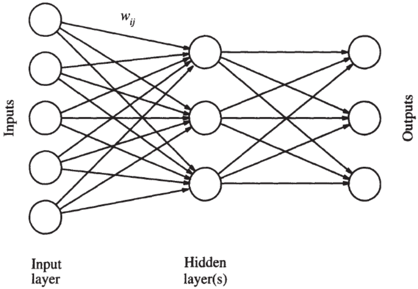
3. Interpretability in Neural networks
   - Neural Networks: A brief review
   - Interpretability in Neural Networks

A brief review of Neural Networks[2]

- Neural Networks (NN, or Artificial Neural Networks, ANN) are a class of Machine Learning models inspired in the human brain.
- They try to mimic with mathematical models the properties observed in the biological neural systems.
- Here we only deal with one-hidden-layer neural networks.

[2] We follow Chapter 11 in Hastie, Tibshirani, and Friedman (2009)

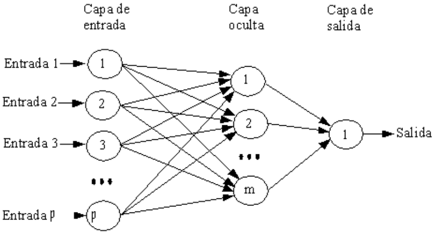
One-hidden-layer neural networks



- A one-hidden-layer neural network is a non-linear parametric regression model represented by the above directed graph.
- At each node **N** the inputs are additively combined, then they are transformed by an *activation function* σ and they result in

$$z_{\mathbf{N}} = \sigma\left(\sum_{\ell \in \text{Input de } \mathbf{N}} w_\ell x_\ell \right).$$

A neural network represented by the graph



corresponds to the following function from $\mathbb{R}^p$ to $\mathbb{R}$:

$$f(\mathbf{x}) = \sigma_2\left(w_0^{(2)} + \sum_{j=1}^{m} w_j^{(2)} \, \sigma_1\left(w_{0j}^{(1)} + \sum_{\ell=1}^{p} w_{\ell j}^{(1)} x_\ell \right)\right).$$

1. Introduction to model-specific interpretability methods

2. Interpretability in Random forests
   - A brief review of CART and Random Forests
     - Regression trees
     - Classification trees
     - Random forests
   - Variable Importance based on node impurity measures
   - Out-of-Bag Variable Importance

3. Interpretability in Neural networks
   - Neural Networks: A brief review
   - Interpretability in Neural Networks

Interpretability in Neural Networks

A useful tool for interpretability in NN is to look at the derivatives of the prediction function

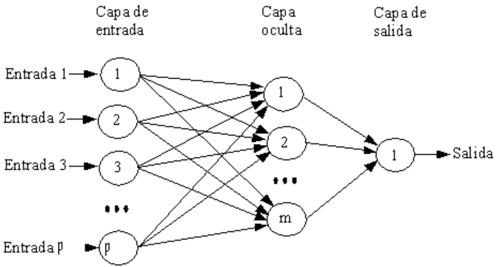
$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{m} \beta_j \, \sigma\left(\alpha_{0j} + \sum_{\ell=1}^{p} \alpha_{\ell j} x_\ell\right)$$

with respect to each variable x_ℓ, $\ell = 1, \ldots, p$:

$$\frac{\partial f}{\partial x_\ell} = \sum_{j=1}^{m} \beta_j \, \sigma'\left(\alpha_{0j} + \sum_{\ell=1}^{p} \alpha_{\ell j} x_\ell\right) \alpha_{\ell j}.$$

The gradient of f at $\mathbf{x}$ is often required:

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_p}\right).$$



$$f(\mathbf{x}) = \sigma_2\left(\beta_0 + \sum_{j=1}^{m} \beta_j\, \sigma_1\left(\alpha_{0j} + \sum_{\ell=1}^{p} \alpha_{\ell j} x_\ell\right)\right)$$

In general,

$$t_j = \alpha_{\cdot j}^{\mathsf{T}} \mathbf{x},\ j = 1, \ldots, m,\ y_j = \sigma(t_j),\ j = 1, \ldots, m,\ z = f(\mathbf{x}) = \beta^{\mathsf{T}} \mathbf{y}.$$

$$\nabla f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{t}} \frac{\partial \mathbf{t}}{\partial \mathbf{x}}.$$

These computations are easy to implement since partial derivatives can be computed using backpropagation, even in NN with complex architectures.

NN: Interpretation and explanation

- The tutorial paper of Montavon et al. (2018) is a good survey of interpretability for Neural Networks.
- Montavon et al. (2018) distinguishes between interpretation and explanation of a fitted neural network.
- For interpretation, activation maximization (and improvements) are the suggested techniques in Montavon et al. (2018).
- **Activation maximization** consists on searching for the input pattern (called *prototype*) that produces a maximum model response for a quantity of interest (for instance, the estimated probability to belong to one of the classes when the response is qualitative).
- So the found prototype indicates which characteristics in the data are mainly taken into account by the model.

NN Explanation

- Given a data point $\mathbf{x} \in \mathbb{R}^p$, the objetive is to explain why the NN produces the response prediction $f(\mathbf{x})$ for it.
- For explanation of NN decisions, sensitivity analysis and simple Taylor decomposition are considered in Montavon et al. (2018).
- **Sensitive analysis:** The goal is to identify the input feature along which the largest local variation is produced around a given data point $\mathbf{x}$.
- A possibility is to compute the relevance score at $\mathbf{x}$ for each feature h, that is the square of the partial derivative with respect to the h-th variable of the function codified by the NN.
- These computations are easy to implement since partial derivatives can be computed using backpropagation.

- **Simple Taylor decomposition:** The NN function is approached at a given data point **x** by the first order Taylor expansion, which is then interpreted as any linear estimator, providing an explanation of how the NN function varies around **x**.
- These procedures, as well as activation maximization, have limitations to show the general pattern of interactions among variables.

Interpretability in convolutional neural networks

- Montavon et al. (2018) acknowledges that interpreting deep neural networks remains a young and emerging field of research, and revises *backward propagation techniques* and *layer-wise relevance propagation*.
- The survey paper of Barredo-Arrieta et al. (2020) and Chapter 7 in Molnar (2019) offer information on this topic.
- The last part of this course is devoted to interpretability in convolutional neural networks.

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.

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

Hastie, T., R. Tibshirani, and J. Friedman (2009).
The elements of statistical learning: data mining, inference and prediction (2 ed.).
Springer.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013).
An Introduction to Statistical Learning with Applications in R.
Springer.

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

Montavon, G., W. Samek, and K.-R. Müller (2018).
Methods for interpreting and understanding deep neural networks.
Digital Signal Processing 73, 1–15.