Variable Importance in Random Uniform Forests

Saïp Ciss

Variable Importance in Random Uniform Forests

Saïp Ciss*

January 18, 2015

Abstract

Random Uniform Forests (Ciss, 2015a) are an ensemble model that use many randomized and unpruned binary decision trees to learn data. It is a variant of *Random Forests* (Breiman, 2001) and in this article, we will focus on how *variable importance* is assessed in Random Uniform Forests. We provide many measures of variable importance and show how they can help to explain the data and how they can enhance prediction tasks. In Random Uniform Forests, the main purpose of Variable Importance is to assess *which, when, where and how covariates have influence* on the problem. We provide a description of measures of Variable Importance as they are defined in the model, with full comprehensive examples and many visualization tools. These ones may be viewed as the shadow of Variable Importance techniques and all tools discussed can be found in the *randomUniformForest* R package.

Keywords : Random Uniform Forests, Variable importance, statistical learning, machine learning, ensemble learning, classification, regression, R package.

*PhD. Université Paris Ouest Nanterre La Défense. Modal'X. saip.ciss@wanadoo.fr

1 Introduction

In the Machine Learning domain, Variable Importance takes an increasing part in the way of understanding models and explaining variables that rely to the problem one needs to solve. To take a parallel, in linear models especially when they come with statistical hypothesis, once the prediction task is achieved one needs to know how covariates are explaining the predictions; informations are usually got by assessing coefficients of the regression task. In real life problems, this is essential since one must provide clear informations to the man that pushes the button. Moreover, if shifts happen, letting predictions deviate from the true responses, the model should be able to provide, at least, a global explanation. For example, when detecting defaults in an industrial process, knowing which covariates (and their links with others) are leading the most to defaults is a companion task of the prediction one. Statistical hypothesis are valid but can be, more often than expected, a particular case of the real word. When going toward non linear and non parametric models, only a very few hypothesis are needed but validity must be found in the data. When the models are also random, concepts like Variable Importance depend from both model and data. The paradigm is to state that if predictions are consistent and if Variable Importance depends on both model and data, then Variable Importance will also be, at least, consistent.
For ensemble models like Bagging (Breiman, 1996) or Random Forests (Breiman, 2001), Breiman introduced measures to assess the importance of the covariates with respect to the problem, the model and the data. Friedman (2002) also provided an important measure, *partial dependence*, that allows to link, almost directly, predictions and any (or many) covariate(s). In this article, we extend these concepts and define them for Random Uniform Forests. Ensemble models are well suited for Variable importance measures for, at least, three reasons :
- they are, usually, random models. Hence what is first expected, is to also get random *relations* between covariates or between covariates and the target. If it does not happen, one may consider, with confidence, that relations lead to strong informations.
- Ensemble models use many base learners, hence data can be assessed over these ones, leading to enough information.
- They, usually, need a very few hypothesis about the data and can handle a huge number of variables in a same analysis.

We, first, provide a quick overview of Random Uniform Forests in section 2. In section 3, we describe the *global variable importance*. In section 4, we define and describe the *local variable importance* which leads to *partial importance* and *interactions* between covariates. Section 5 provides another alternative for getting *partial dependencies*. We give, in section 6, full comprehensive examples over two real datasets and conclude in section 7.

2 An overview of Random Uniform Forests

Random Uniform Forests are close to Breiman's Random Forests but they come with many differences at the theoretical and algorithmic levels. The most important one is the use of *random cut-points*. More precisely, Random Uniform Forests are designed to be

fairly simple and to let data speak for themselves with some kind of global optimization. They are also designed to be enough versatile to allow techniques like incremental learning, unsupervised learning or models like Bagging or ensemble of totally randomized trees.

Formally, a Random Uniform Forest is an ensemble of *random uniform decision trees*, which are unpruned and binary random decision trees that use the continuous Uniform distribution to be built. Let us consider $D_n = \{(X_i, Y_i), 1 \leq i \leq n\}$, corresponding to the observations and responses of a training sample, where (X, Y) is a $\mathbb{R}^d \times \mathcal{Y}$-valued random pair, with respect to the *i.i.d.* assumption. Let us suppose, for brevity, that $Y \in \{0, 1\}$ considering, then, the binary classification case. The decision rule of a random uniform decision tree can be written with the following lines :

$$g_{\mathcal{P}}(x, A, D_n) = g_{\mathcal{P}}(x) = \begin{cases} 1, \ if \ \sum_{i=1}^{n} \mathbf{I}_{\{X_i \in A, Y_i = 1\}} > \sum_{i=1}^{n} \mathbf{I}_{\{X_i \in A, Y_i = 0\}}, \ x \in A \\ \\ 0, \ otherwise. \end{cases}$$

A is the current terminal and optimal region (node), coming from the recursive partitioning scheme.
$g_{\mathcal{P}}$ is the decision rule of the tree.

For regression we have:

$$g_{\mathcal{P}}(x, A, D_n) = g_{\mathcal{P}}(x) = \frac{1}{\sum_{i=1}^{n} \mathbf{I}_{\{X_i \in A\}}} \sum_{i=1}^{n} Y_i \mathbf{I}_{\{X_i \in A\}}, \ x \in A.$$

To define each possible region, and then a terminal one, we need a variable and a *cut-point* (a threshold below and beyond which one region and its complementary ones are defined). We sates that A is an *optimal region* of the random uniform decision tree if :

$$for \ any \ A \in \mathcal{P}, \left\{ X_i^{(j^*)} \leq \alpha_{j^*} | D_n \right\}, \ 1 \leq j \leq d, 1 \leq i \leq n,$$
$$for \ any \ A^C \in \mathcal{P}, \left\{ X_i^{(j^*)} > \alpha_{j^*} | D_n \right\}, \ 1 \leq j \leq d, \ 1 \leq i \leq n,$$

where, for classification :
$\alpha_j \sim \mathcal{U}\left(\texttt{min}(X^{(j)}|D_n), \texttt{max}(X^{(j)}|D_n)\right)$ and $j^* = \underset{j \in \{1,\ldots,d\}}{\texttt{arg max}} \ \texttt{IG}(j, D_n)$,
and for regression :
$\alpha_j \sim \mathcal{U}\left(\texttt{min}(X^{(j)}|D_n), \texttt{max}(X^{(j)}|D_n)\right)$ and $j^* = \underset{j \in \{1,\ldots,d\}}{\texttt{arg min}} \ \texttt{L}_2(j, D_n)$,
with IG, the *Information Gain* function and $\texttt{L}_2$, an Euclidean distance function.

One can note that using the support of each candidate variable is not necessary, while more convenient, especially in classification. IG and $\texttt{L}_2$ are the criteria that allow to choose the best random optimal node at each step of the recursive partitioning. We get :

$$\texttt{IG}(j, D_n) = \texttt{H}(Y|D_n) - \left[\texttt{H}\left(\left(Y|X^{(j)} \leq \alpha_j\right)|D_n\right) + \texttt{H}\left(\left(Y|X^{(j)} > \alpha_j\right)|D_n\right)\right],$$

where H is the *Shannon entropy* (note that we use it with the natural logarithm),
and

$$\texttt{H}(Y|D_n) = -\sum_{c=0}^{1} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}_{\{Y_i = c\}} \ \texttt{log}\left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{I}_{\{Y_i = c\}} \right) \right\},$$

with, by definition, $0 \log 0 = 0$, so that $\mathtt{H}(Y) \geq 0$.

Let $n^{'} = \sum_{i=1}^{n} \mathbf{I}_{\left\{X_i^{(j)} \leq \alpha_j\right\}}$, then

$$\mathtt{H}\left(\left(Y|X^{(j)} \leq \alpha_j\right)|D_n\right) = -\frac{n^{'}}{n}\sum_{c=0}^{1}\left\{\frac{1}{n^{'}}\sum_{i=1}^{n}\mathbf{I}_{\{Y_i=c\}}\mathbf{I}_{\left\{X_i^{(j)} \leq \alpha_j\right\}} \log\left(\frac{1}{n^{'}}\sum_{i=1}^{n}\mathbf{I}_{\{Y_i=c\}}\mathbf{I}_{\left\{X_i^{(j)} \leq \alpha_j\right\}}\right)\right\},$$

and

$$\mathtt{H}\left(\left(Y|X^{(j)} > \alpha_j\right)|D_n\right) = \\ -\frac{n-n^{'}}{n}\sum_{c=0}^{1}\left\{\frac{1}{n-n^{'}}\sum_{i=1}^{n}\mathbf{I}_{\{Y_i=c\}}\mathbf{I}_{\left\{X_i^{(j)} > \alpha_j\right\}} \log\left(\frac{1}{n-n^{'}}\sum_{i=1}^{n}\mathbf{I}_{\{Y_i=c\}}\mathbf{I}_{\left\{X_i^{(j)} > \alpha_j\right\}}\right)\right\}.$$

For regression, we define $\mathtt{L}_2(j, D_n)$ by

$$\mathtt{L}_2(j, D_n) = \sum_{i=1}^{n}\left(Y_i\mathbf{I}_{\left\{X_i^{(j)} \leq \alpha_j\right\}} - \hat{Y}_A\mathbf{I}_{\left\{X_i^{(j)} \leq \alpha_j\right\}}\right)^2 + \sum_{i=1}^{n}\left(Y_i\mathbf{I}_{\left\{X_i^{(j)} > \alpha_j\right\}} - \hat{Y}_{A^C}\mathbf{I}_{\left\{X_i^{(j)} > \alpha_j\right\}}\right)^2,$$

with

$$\hat{Y}_A = \frac{1}{n'}\sum_{i=1}^{n} Y_i\mathbf{I}_{\left\{X_i^{(j)} \leq \alpha_j\right\}} \;\; and \;\; \hat{Y}_{A^C} = \frac{1}{n-n'}\sum_{i=1}^{n} Y_i\mathbf{I}_{\left\{X_i^{(j)} > \alpha_j\right\}}.$$

It remains to define the decision rule, $\bar{g}_{\mathcal{P}}^{(B)}$, of the Random Uniform Forest classifier :

$$\bar{g}_{\mathcal{P}}^{(B)}(x) = \begin{cases} 1, \; if \;\; \sum_{b=1}^{B}\mathbf{I}_{\{g_{\mathcal{P}}^{(b)}(x)=1\}} > \sum_{b=1}^{B}\mathbf{I}_{\{g_{\mathcal{P}}^{(b)}(x)=0\}} \\ \\ 0, \; otherwise. \end{cases}$$

And for regression :

$$\bar{g}_{\mathcal{P}}^{(B)}(x) = \frac{1}{B}\sum_{b=1}^{B} g_{\mathcal{P}}^{(b)}(x).$$

Random Uniform Forests inherit of all properties of Breiman's Random Forests, especially convergence and Breiman's bounds. Their main theoretical argument is to achieve low correlation of trees (or trees residuals) while not letting average variance increasing too much. While one may get more details (Ciss, 2015a), in this article the main argument that can be linked with the Variable Importance concepts relies on the functions `IG` and $\mathtt{L}_2$.

3 Global Variable Importance

As in Random Forests, we provide a similar measure for variable importance before going deeper in the assessment of variables.

We call it the *global variable importance*, measured directly using the optimization criterion. if $\texttt{VI}$ is the score of importance, we have for the j-th variable, $1 \leq j \leq d$, named j^* if it is optimal,

$$\texttt{VI}(j^*) = \sum_{b=1}^{B} \sum_{l=1}^{k_b} \texttt{IG}_{b,l}(j^*, D_n),$$

and, for regression

$$\texttt{VI}(j^*) = \sum_{b=1}^{B} \sum_{l=1}^{k_b} \texttt{L}_{2_{b,l}}(j^*, D_n),$$

where k_b is the number of regions for the b-th tree, $1 \leq b \leq B$.
We, then, define *relative influence* by computing $\texttt{VI}(j^*)/\sum_{j=1}^{d} \texttt{VI}(j)$ and report it as the *global variable importance* measure.

Variable importance is measured over all nodes and all trees, leading all variables to have a value, since cut-points are random. Hence each variable has equal chance to be selected but it will get importance only if it is the one that decreases the most the entropy at each node. In the case of regression, an important variable must really earn its place since the lowest value of the $\texttt{L}_2$ function is selected for each node, being more and more smaller as we get deeper in the tree. *Global variable importance produces the variables that lower the most the prediction error* but it tells us nothing about how one important variable affects the responses. We might want to know, for example, features that affect the most one class or features that lead to high realizations of Y, in case of regression. Or we might want to know interactions of variables.

4 Local Variable Importance

`Definition.` *A predictor is locally important at the first order if, for a same observation, and for all trees, it is the one that has the highest frequency of occurrence in a terminal node.*

Let us note $\texttt{LVI}^{(b)}(j, i)$, the local importance score of the i-th observation and the j-th variable of X, for the b-th tree,
$\texttt{LVI}(j, i)$, the score for all trees,
$\texttt{LVI}(j, .)$, the score of the j-th variable of X for all observations,
$R(j, \alpha_j) = \{X | X^{(j)} \leq \alpha_j\}$, a candidate terminal region whose all observations are below α_j for $X^{(j)}$,
$R^C(j, \alpha_j) = \{X | X^{(j)} > \alpha_j\}$, its complementary region.
To simplify, we rely on classification and define $\texttt{LVI}^{(b)}(j, i)$ by

$$\texttt{LVI}^{(b)}(j, i)$$
$$= \mathbf{I}_{\{\texttt{IG}_b(j^*, D_n) = \texttt{IG}_b(j, D_n)\}} \left(\mathbf{I}_{\left\{g_{\mathcal{P}}^{(b)}(X_i, R(j^*, \alpha_{j^*})) = g_{\mathcal{P}}^{(b)}(X_i, R(j, \alpha_j))\right\}} + \mathbf{I}_{\left\{g_{\mathcal{P}}^{(b)}(X_i, R^C(j^*, \alpha_{j^*})) = g_{\mathcal{P}}^{(b)}(X_i, R^C(j, \alpha_j))\right\}} \right).$$

The above relation states that we count when a variable falls into a terminal node (a leaf),

which tree and which observation are concerned. The first term of this expression tells us that the variable we are counting must be the optimal one for the node that leads to a leaf. The second and the last terms state that we are counting one of the two possible leaves, depending on where the tree drives the data. Hence, for the j-th variable we count it only when it is an optimal one for the leaf where falls the i-th observation. To be more clear, the `LVI` function is nothing other than a counting function for variables that immediately lead to a leaf and find these variables depends only to where an observation will fall. One can note that we do not use the training labels, since we want all the measures we will give to be only dependent to the forest itself. Knowing that, we can count for all trees the function $\texttt{LVI}(j,i)$, given by

$$\texttt{LVI}(j,i) = \sum_{b=1}^{B} \texttt{LVI}^{(b)}(j,i),$$

and for all trees and observations and for the j-th variable,

$$\texttt{LVI}(j,.) = \sum_{i=1}^{n} \texttt{LVI}(j,i).$$

Next step is to consider that the j-th variable can appear in second, third, or more, position among all others when counting all the scores and ordering them, for all trees and for any single observation. Note, here, that position and order designate the same term. For example, one variable may appear at the second position for half of the observations and at last position for the other half. We do not want to be too close to the observations, since in practice assessing the test set is as (if not more) important as assessing the training set. Hence, instead of just aggregating the local score by computing $\texttt{LVI}(j,.)$, we consider that a variable is locally important for all trees and all observations if it has a high rank. The procedure is defined with the following steps :
i) for each observation and variable, we compute $\texttt{LVI}(j,i)$ and report the name of the variable that has the highest number of occurrences, then the name of the one that has the second highest number of occurrences and so on.
ii) The first step gave us a table, with n observations and d columns. In each row we have the name of the variables ordered from variables that have the highest number of occurrences to the ones that have the lowest. Only the names of the variables are reported, hence each column is a meta-variable that we call *position* since it gives the rank of any variable for each observation.
iii) The third step is to count how many times a variable appears in position q, $1 \leq q \leq d$.

However, we want to know, if we choose the position q of the j-th variable what will be its score. We call this function the local importance score at the position q, $1 \leq q \leq d$, $\texttt{SLI}_q(j,.)$ for all observations and for the j-th variable. It is defined by

$$\texttt{SLI}_q(j,.) = \sum_{i=1}^{n} \mathbf{I}_{\left\{ j = \underset{u \in \{1,\dots,d\}}{\texttt{arg}} \ \texttt{LVI}_{(d-q+1)}(u,i) \right\}},$$

where $\texttt{LVI}_{(d-q+1)}$ is the $(d-q+1)$-th order statistics of the `LVI` function, ordering all variables for any single observation. Here, we state that if we choose the position q of the

j-th variable, we will get a score that matches all the cases (for all trees and all observations) where the j-th variable came at the position q. But we have d variables, so knowing the score of any of them does not tell us which is the most locally important since the positions we got were fixed and are not the true ones when considering all variables.

iv) Hence, the last step is to get all locally important variables ranked from the most influential to the less influential, allowing us, by the way, to define *interactions* between covariates. The true position (so its rank) of the j-th variable, for a fixed position q, is given by

$$q_j^* = \underset{u \in \{1,\dots,d\}}{\texttt{arg max}}\, \texttt{SLI}_q(u, .).$$

The computation of q_j^* implies that a variable can only achieve influence when comparing it to the influence of others for the same fixed position q. This is what defines implicitly the local importance in the same sense that relative influence is defined for the global variable importance. For example, if $q_j^* = 1$, meaning that we first define q to be 1 for all variables, then the j-th is the most locally important at the first order.

4.1 Partial importance

Partial importance is the way to assess the importance of a covariate when the label Y is fixed (or for realizations of Y above or below a threshold in the case of regression).

`Definition.` *A predictor is partially important if, for a same observation and one class, and at all the orders, it is the one that has the highest frequency of occurrence in a terminal node.*

Note that the above definition also matches the case of regression. The score of partial importance is given by

$$\texttt{SLI}(j, ., y) = \sum_{q=1}^{d} \left(\texttt{SLI}_q(j, .) | \bar{g}_{\mathcal{P}}^{(B)}(X) = y \right).$$

To understand partial importance one has to take account two points:
i) counting occurrences of a predictor is interesting but does just give frequency,
ii) for each observation met in a terminal node, we record the value estimated by the classifier. That gives us the intensity (or the label in case of classification).

Hence, we have for each observation the score of any covariate, at all positions, and the classifier estimate. Then, partial importance gives, for each class or for some responses above (or below) a threshold, the predictors which are the most relevant. Again, we compute relative score to be more close to the model. Clearly, partial importance tells what are the variables that are explaining the variability of the response (regression) or one class rather than another (classification). This gives us a new information with regard to global variable importance, that just tells us what are the relevant variables. One can note that interesting cases come with regression for which we can observe how covariates are explaining the vector of responses if we look any range of the latter.

4.2 Interactions

But we may want to know how covariates rely to the problem when we consider them all. For example, some variables could have a low relative influence on the problem but a strong effect on a more relevant covariate. Or this covariate could have many interactions with others, leading the variable to have influence. So, we need a definition for *interactions*.

`Definition.` *A predictor interacts with another one if, for a same observation, and for all trees, both have respectively the first and the second highest frequency of occurrence in a terminal node.*

If `VII` is the score of interactions, we have for $X^{(j)}$ and $X^{(j')}$,

$$\texttt{VII}_{(1,2)}(j, j') = \frac{\texttt{SLI}_1(j, .) + \texttt{SLI}_2(j', .)}{2n},$$

where the values 1 and 2 state for the first and second order and are given by first computing q_j^* and $q_{j'}^*$ to find which variables (two) come at first in the interaction order. Then, for the most important we report $\texttt{SLI}_1(j, .)$, where j is now fixed, and choose the j'-th variable for a fixed position $q = 2$ and for the remaining $d - 1$ variables. If we can not get the first and second order, then the interaction will be null for the two variables. Interactions act as a visualization tool in order to see the influence of all covariates in a unique manner.
The last step computes $\texttt{VII}_{(1,2)}(j, j')$ and we create a contingency table where all values of $\texttt{VII}_{(1,2)}$ and $\texttt{VII}_{(2,1)}$ will stand. That gives us the *interactions visualization tool* of all covariates. Moreover, we can merge covariates that have weak influence to have a granular view. Interactions are interesting since they give the big picture on how all variables rely to the problem for any pair of variables.
They also lead to the measure of *variable importance based on interactions*, given by

$$\texttt{VII}(j, .) = \frac{1}{2d} \sum_{j'=1}^{d} \texttt{VII}_{(1,2)}(j, j') + \frac{1}{2d} \sum_{j'=1}^{d} \texttt{VII}_{(2,1)}(j, j') + \texttt{VII}_{(1,2)}(j, j).$$

We state here that the variable importance based on interactions for the j-th variable is the measure on how this variable is having dependence with all others and how it is having influence on the local variable importance.

The point we introduced is that, for all measures that depend to the local variable importance, we did not want to average scores over observations, since the forest classifier refers to observations, dimension and trees. In order to get a more generic view, we rely on positions, meaning that each variable gets influence depending on the rank it obtains each time we compute a measure on a single observation, using all trees. The rank goes from 1 to the total number of variables and a high rank is simply a high number of occurrences. Then, the rank is generalized over all observations leading to the measures we get.

5 Partial dependencies

These are tools that allow to measure how influence of each covariate (or a pair of covariates) is affecting the response values, knowing the values of all others covariates. More clearly, a partial dependence plot is the marginal effect of a covariate over the response values. The idea of partial dependence came from Friedman (2002) who used it in *Gradient Boosting Machines (GBM)*; but it is implemented differently in Random Uniform Forests.
Let $\mathtt{pD}_q^{(j)}$ be the partial dependence at position q for the j-th variable. We have

$$\mathtt{pD}_q^{(j)}(Y,X) = \left\{ \left(\left(\bar{g}_{\mathcal{P}}^{(B)}(X_i, R(j,\alpha_j)), X_i^{(j)} \right) | \mathtt{SLI}_q(j,i) > 0 \right), 1 \leq i \leq n \right\},$$

where $\bar{g}_{\mathcal{P}}^{(B)}(X_i, R(j,\alpha_j))$ is the forest classifier evaluating each observation, knowing that the j-th variable is locally important for the current evaluated observation. The local importance is, so, designated by the $\mathtt{SLI}_q$ function for every point. What makes the difference with a simple look in a terminal node to get a prediction is that the covariate must have some influence for each observation. Hence both predicted value and observation will stress the partial dependence plot up to some point where values will no longer be available. To avoid getting a too weak relation between covariate and target, we define the partial dependence function at all orders. It is given by

$$\mathtt{pD}^{(j)}(Y,X) = \left\{ \left(\mathtt{pD}_1^{(j)}(Y,X), ..., \mathtt{pD}_q^{(j)}(Y,X), ..., \mathtt{pD}_d^{(j)}(Y,X) \right), 1 \leq q \leq d \right\}.$$

The function provides all the points needed in the range of the covariate and increases the randomness. One can note that each point $X_i^{(j)}$ will now have at most d values of the forest classifier for assessing the dependence. This has the advantage of getting more consistent and interpretable results. If covariate and target have no relation, plotted points will have an Uniform distribution on the map. Otherwise, one will get the shape of their relation. One of the interesting point with partial dependencies is their ability to allow extrapolation. This latter is known to be a limitation of Random (Uniform) Forests since their estimator can not produce unseen values beyond the range of Y in the training sample. One just has to compute a parametric model of the tails in the partial dependence function. Usually these will be linear with the covariate and extrapolation happens by choosing the right threshold beyond which a linear model will be a good approximation. If many covariates have influence, one can eventually compute one parametric model over a multi-dimensional partial dependence function. The main argument of Random Uniform Forests is that they allow to get more points for such objects like extrapolation.

We provided in the lines above many ways to assess variable importance and one can visualize most of them in just one screen to get the big picture. It leads to both feature selection and interpretation. This latter point is essential since, to our point of view, Random (Uniform) Forests are not a black box. We can then summarize main properties of Variable Importance in Random Uniform Forests.
i) All tools provided, except the global variable importance, work on both training and test samples. As a consequence, they do not use the labels (or responses) but only the

forest classifier.
ii) All tools are complementary. It means that when a feature is found to be important, an explanation can be found using each measure separately and/or combining them.
iii) The main purpose of variable importance is to assess *which, when, where and how covariates have influence* on the problem.
iv) In reference to Random Uniform Forests, we can define the importance of a variable with the following scheme : *importance = contribution + interactions, where contribution is the influence of a variable (relatively to the influence of all) on the prediction error and interactions is, at least, its influence on the others covariates.*

6 Experiments

In this section, we provide many visualization tools and measures of the methods provided below. To make them reproducible, all the R code is provided and we take for both classification and regression one real world dataset.

6.1 Classification

For classification, we chose the *Car evaluation* dataset freely available on the *UCI repository*. or in the *randomUniformForest* package. The dataset has 1728 rows, 6 attributes, all categorical, and 4 classes. The purpose of the task is to classify each car (given by a row) as *unacceptable (unacc), acceptable (acc), good or very good (vgood).* Variables are *buying, priceOfMaintenance, nbDoors, nbPersons, luggageBoot, safety.* In the context of Variable Importance we want to know how covariates lead to one class or another. One has to note that the randomUniformForest algorithm is stochastic. Hence, even with the same (data and) seed one will not be able to exactly reproduce results; however, since convergence happens, there will be only slight variations that will not change the analysis.

At first, let us (install and) load the package under the R software (>= 3.0.0).

```
# Install package
install.packages("randomUniformForest")

# load it
library(randomUniformForest)
```

Then, load the data, extract the labels column, take a random subset of the data and train the model. Note that one must take care of the categorical variables.

```
data(carEvaluation)
XY = carEvaluation
classColumn = ncol(XY)
Y = extractYFromData(XY, whichColForY = classColumn)$Y
X = extractYFromData(XY, whichColForY = classColumn)$X

# view a summary
str(X)
```

```
# or summarize
summary(X)
# see the distribution of the labels
table(Y)

# then train the model, using a subset (half) of the data
set.seed(2014)
n = nrow(X)
subsetIdx = sample(n, floor(n/2))
car.model.ruf = randomUniformForest(X, Y,
subset = subsetIdx, ntree = 500, categoricalvariablesidx = 1:6)
```

Note that we choose the number of trees in the forest, *ntree*, to be 500, and the number of selected features for each node is the default one, '$4/3 \times$ dimension'. This latter means that we select, randomly and with replacement, 8 variables that correspond to 8 candidate nodes from which, only one will be chosen (using the optimization criterion) to get the optimal random node at each step of the tree growth.
Using *categoricalvariablesidx = 1:6* forces the algorithm to consider all the variables as categorical, letting it use the engine that is dedicated to these variables. Accuracy may drop a little, in comparison to consider them as purely numeric, but Variable Importance assessment will be consitent with the whole process.

One may look the results of the evaluation, calling the trained model :

```
car.model.ruf
```

i) But, since we are interested by assessing covariates, let us look the *global Variable Importance* which comes with two blocks :

```
summary(car.model.ruf)
```

displays :

```
Variables summary:
           variables score class class.frequency percent percent.importance
1 priceOfMaintenance  3171   acc            0.51  100.00                 26
2             buying  2736   acc            0.50   86.27                 23
3            nbDoors  2335   acc            0.50   73.64                 19
4        luggageBoot  1962   acc            0.48   61.87                 16
5          nbPersons  1353 unacc            0.46   42.66                 11
6             safety   555 unacc            0.85   17.51                  5
```

This first table gives the score of global variable importance for all variables, the majority class and its frequency and the relative influence of each variable. The main point here is that *safety* which is the less influential, when considering all classes, is the most one, by far, when considering *unacceptable* cars. One can note that class frequencies do not need to sum up to 1, because each variable is considered separately from the others, getting first its score then looking the majority class. The link between classes and variables is

essential in Random Uniform forests. But, let us first look to the whole plot of global variable importance.

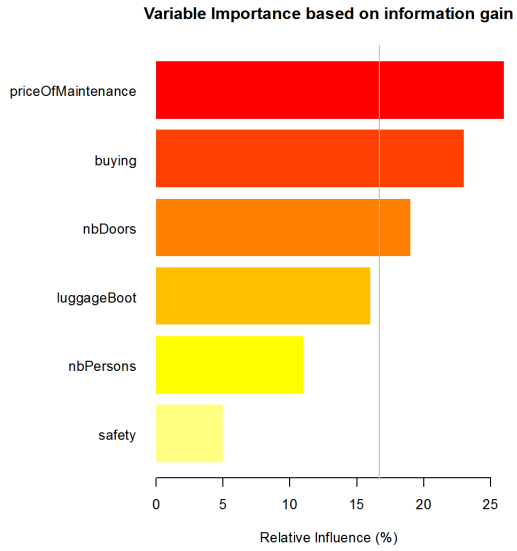

Figure 1: Global variance importance for (a subset of) the *Car evaluation* dataset

The plot is simply the extraction of the former table with the grey vertical line indicating where the importance of each variable would lie if all variables had the same influence. Comparing this plot with the one from Breiman's Random Forests would show *safety* and *nbPersons* as the most influential variables. If we take into account the class distribution, *unacceptable* cars are the most frequent (69% of the cases) and one would wonder, as it is shown in the table above, how matter the number of cases per class. In Random Uniform Forest, the point of view is taking account both prediction and class distribution aspects.
- *The global variance importance plot is a measure that gives the influence of the most predictive variables with little dependency to the class distribution.*
- *The global variance importance table nuances the measure by also getting how each variable is relying to a class, from the same predictive point of view.*
Hence, if we order the table by ordering variable according to the frequency of each class we get the same order than in the *randomForest* algorithm.

ii) The next step is to get the big picture. We have to call the *importance* function of the algorithm then simply plot the results. The function also applies to the test set.

```
car.importance.ruf = importance(car.model.ruf, Xtest = X[subsetIdx,],
maxInteractions = 6)
# that leads to many details which can be summarized with :
plot(car.importance.ruf, Xtest = X[subsetIdx,])
```

The commands below are the default ones, and can be refined to be more or less granular. In particular using a high value of *maxInteractions* lead to be as close as possible to the

details (the parameter q in the local variable importance). We get many plots (one will need to use the R menu, tiling windows vertically, to see them all) that correspond to the tools defined and described formerly.
The first one is the *interactions visualization tool* represented below :

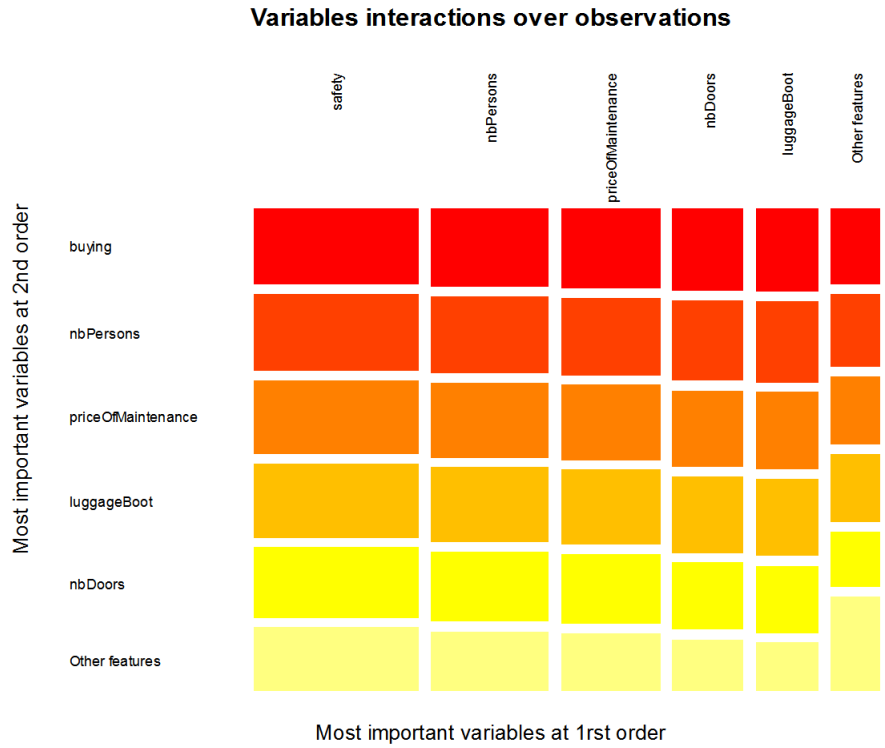

Figure 2: Interactions between covariates for (a subset of) the *Car evaluation* dataset.

The plot above provides the interactions at first and second order for all the covariates, according to the definition we gave for interactions. Its area is one. The first order states that the variables (ordered by decreasing influence) are the most important if a decision has to be taken by considering one, and only one, variable. *safety* comes at first, meaning that when evaluating a car it would be the first variable that would come with the evaluation. The second order states that if an unknown variable is already selected at the first order, then the second most important variable will be one of those in second order. At the second order *buying* (price) comes at first, meaning that if an unknown variable is selected to be the most important variable, *buying* would be the second most important.
To be more clear, *interactions* provide a table of ordered possibilities. First order gives the ordered possibilities of most important variables. Second order gives the ordered possibilities of second most important variables. Crossing a pair of variables gives their *relative co-influence* over all the possible co-influences. One can note that these measures are both model and data dependent. Hence, confidence in the measures relies directly to confidence in the predictions. One can also note that a meta-variable called *Others features* appears, meaning that we let the algorithm show the default view for visualization, grouping the variables that are less relevant.

iii) Interactions come with their Variable Importance measure that ranks variables by their relative co-influence. We represent it below :

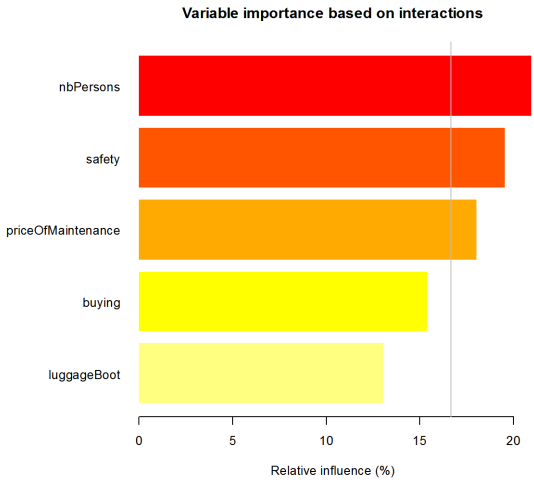

Figure 3: Variable Importance based on interactions for (a subset of) the *Car evaluation* dataset.

The graph above shows how each variable is represented when aggregating its co-influence with any other variable. *One important note is that the first variable is not necessary the most important but the one that has the most co-influence with others.*

iv) We also define the *Variable Importance over labels* which provides a view on how the aggregated interactions affect each class :

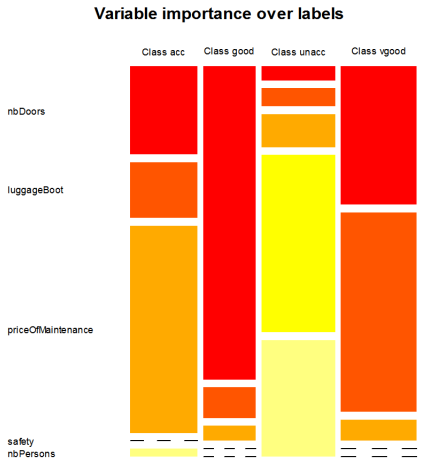

Figure 4: Variable Importance over labels based on interactions on each (fixed) class for the *Car evaluation* dataset.

safety is strongly linked with *unacceptable* cars which are, by far, the most important

class, so that, it comes at the top position of interactions or (global) variable importance of others algorithms. *Variable Importance over labels provides a local point of view : the class is fixed, meaning that one first takes a decision to fix the class by looking the variables that matter and act like constraints, then looks the important variables for each class.* Hence each variable has importance as if others classes did not exist. For example, very good (*vgood*) cars are the ones that offers enough number of doors and space in luggage boot. But before getting that, one has already take a decision with (eventually) others variables to state if a car is very good or not.
Here, *we are not interested by variables that lead a class to be chosen, but by variables that will matter within the class, once this latter chosen.* The order of variables gives their *aggregated rank* relatively to their rank in each class, without considering the importance of the class. For example, *nbDoors* appears in first because it is well placed within each class.*safety* comes in almost last position because it has an high rank only in one class, despite its importance. *An important note is that in the shown mosaic plot, the algorithm computes the values with respect to the displayed variables.* It means that we can also get informations by excluding variables. Here the plot is displayed as if the *buying* variable did not exist, but this latter would have been displayed if its aggregated rank had been high or if we had specified it.

v) Before calling the partial dependencies, one may review the plot above by asking the *partial importance*. In others words, if one decide to fix the class by itself, will one get the same informations about how variables matter within a class ? We need one line of R code per class :

```
car.partialImportance.ruf = partialImportance(X[subsetIdx,],
car.importance.ruf, whichClass = "good")
```

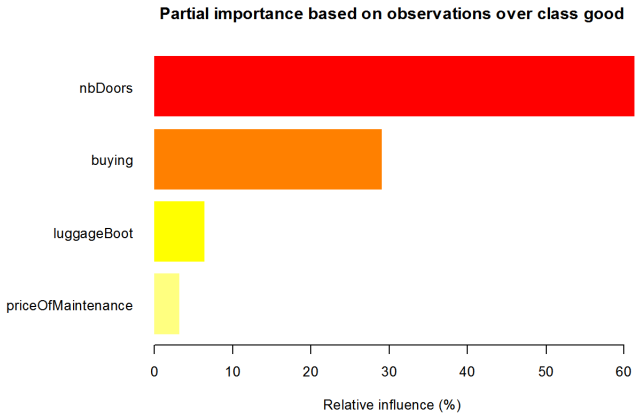

Figure 5: Partial Importance for the class *good* in (a subset of) the *Car evaluation* dataset.

In classification, *partial importance* is almost the same than *Variable Importance over labels* except that it overrides default parameters showing all variables for each asked class. We can see here than *buying* comes at second position when one decides to evaluate a car already considered as a *good* one.

vi) The analysis of Variable Importance can go further by calling *partial dependencies*. For brevity we call only the one for a single variable, here *safety*.

```
car.partialDependence.ruf = partialDependenceOverResponses(X[subsetIdx,],
car.importance.ruf, whichFeature = "safety", whichOrder = "all")
```

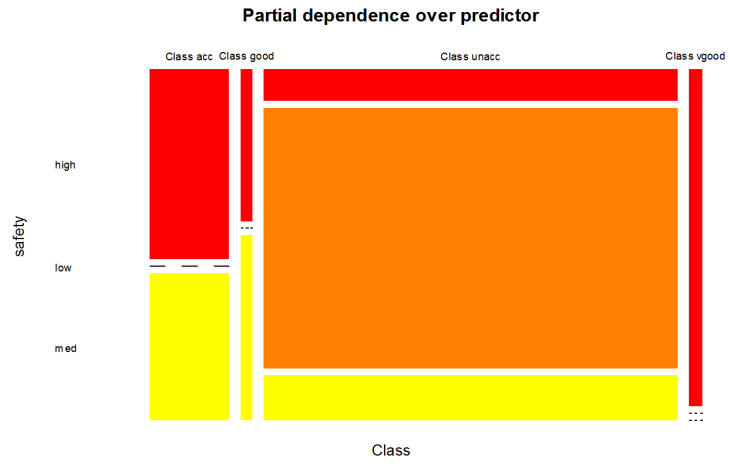

Figure 6: Partial dependence plot in (a subset of) the *Car evaluation* dataset.

The *Partial dependence plot* provides the highest level of granularity, since we get now within each variable, observing its marginal effect on each class. Above we can see that the variable is affecting, the most, *unacceptable* cars for *low* values of safety. As it may seem obvious, one has too remember that it is the result on how the algorithm is assessing the variable. Others classes do not accept any *low safety*, meaning that if another class is chosen or observed, It will have first passed the *safety* barrier. For *acceptable* or *good* cars a *medium safety* will be accepted while for a *very good* car, safety must be high and only high.

To summarize, Variable Importance in Random Uniform Forests goes from the higher level to the lower one of granularity. At first, we get *which* variables are important, nuanced by the weight of each class. Then, we find *what* make them influential, looking their interactions and the choice made to choose a variable at first, considering all the classes at once. Next step is to know *where* they get their influence, looking within each class once fixed. At last, we get *when* and *how* a variable is mattering by looking the partial dependence. All measures, except global variable importance, work on either the training or the test set.

6.2 Regression

For regression, the process is almost the same except some graphics that taking into account the continuous values. For the sake of brevity, we will not show the R commands or the plots that look like the ones in the classification example.

We chose the *Concrete compressive strength* dataset freely available on the *UCI repository* or in the *randomUniformForest* package. The dataset have 1030 rows and 9 attributes. The purpose of the task is to evaluate the compressive strength of Concrete ("the most important material in civil engineering"). The compressive strength depends on predictors, namely *Cement, Blast Furnace Slag, Fly Ash, Water, Superplasticizer, Coarse Aggregate, Fine Aggregate* and *Age*. In the context of Variable Importance we want to know what makes an efficient Concrete compressive strength.

i) We use the same analysis than in the classification case, retrieving a random subset (the seed is the same), training the model with default parameters (except the number of trees set to 500) then computing importance. The only difference with the classification case resides in the object of assessment. We will evaluate the test set, rather than the training one.
At first, we produce the *global Variable Importance* from which the whole analysis is driven:

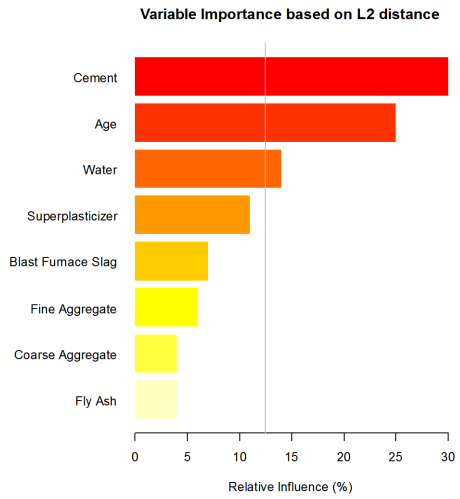

Figure 7: *global Variable Importance* for the *Concrete compressive strength* dataset

.

The *relative influence* given by the model is consistent with the one of GBM or Random Forests, except that both rank *Age* at the first position. Since, in Random Uniform Forests, cut-points are independent to responses this can be explained. Let us see how the details matter.

ii) We call the *interactions*, getting all possible ones and the resulting *Variable Importance based on interactions*. Recalling that we are assessing the test set, interactions show the variables that have the most co-influence with others (and not the ones that are the most important).

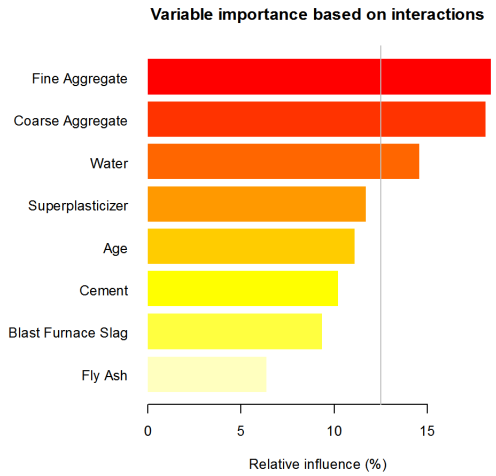

Figure 8: Variable Importance based on interactions for the *Concrete compressive strength* dataset.

We see *Coarse Aggregate* and *Fine Aggregate* as the variables that have the most interactions with the other ones. In practice, it means that if one wants to maximize the Concrete compressive strength these variables, while not being critical like *Cement* and *Age*, might lead to some explanation, for example, if the Concrete compressive strength is too low. Let us call the *Partial Importance* to assess that.

iii) Suppose that we want to know what leads to a high Concrete compressive strength and what leads to a low one. In the training set the unconditional mean of the Concrete compressive strength is 35.38 (*Mpa*, pressure unit) and its standard deviation is 16.86. Let us state that a high Concrete compressive strength must be more than the average + one standard deviation; a low one less than the average minus one standard deviation. We get :

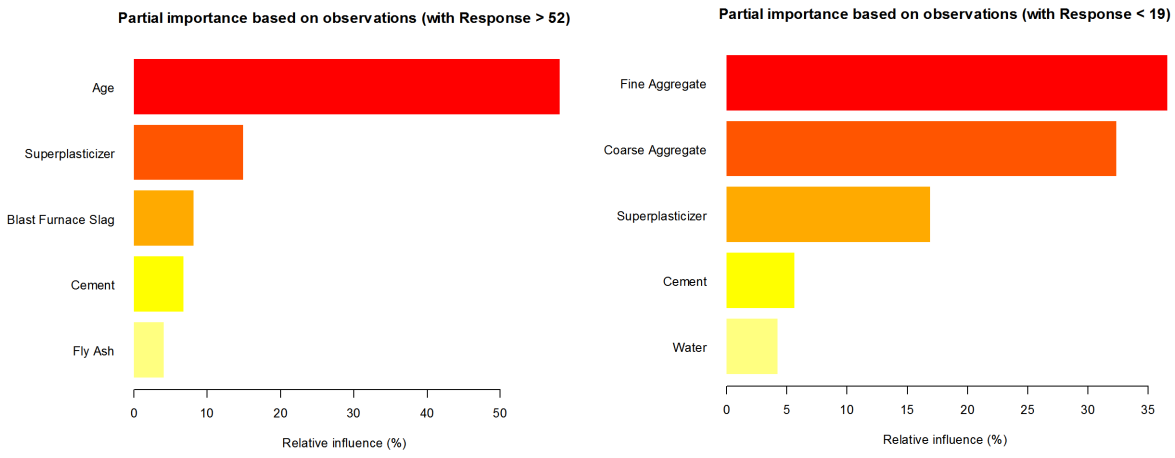

Figure 9: Partial Importance for the *Concrete compressive strength* dataset.

Interactions and global Variable Importance are now (partially) explained. *Coarse Aggregate* and *Fine Aggregate* are strongly involved in low Concrete compressive strength while *Age* (in days) is the main ingredient of an increasing one. However, we still don't

know how to explain the whole range of the latter. Since all covariates are part of the mixture to obtain Concrete one will need values to do the right choice.

iv) To get a point of view, we call the Variable Importance over each fixed variable, forgetting the others and looking how the response (Concrete compressive strength) is distributed over the fixed variable. Let us note that the results are the consequence of the design of the Local Variable Importance so that, a fixed variable will see its results to be dependent on what happened with the others when the whole forest was assessed.

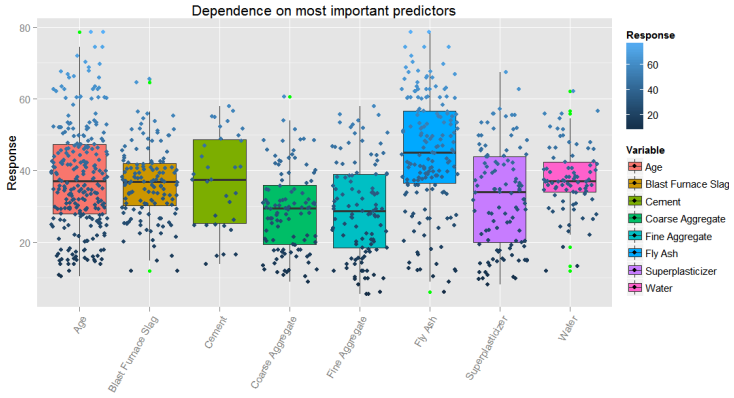

Figure 10: Dependencies for the *Concrete compressive strength* dataset.

Above, each variable is represented (in alphabetic order) and each boxplot is the distribution of the Concrete compressive strength over the variable. If, for example, we are interested by high values we can simply look what variables lead to these values and when to stop use the variables. More precisely, the plot above comes from the *Partial Importance* measure, considering each variable at all positions and the whole range of response values instead of a part. But, something is still missing. The distribution provides neither the direction nor the range of the potential predictor.

v) The last step of the analysis is, then, to assess dependent variable conjointly to any predictor, over the whole range of both variables. *Partial dependence* is the tool that provides these results. We can plot it for all the variables or for the ones that lead to the better (or an increasing) compression strength.

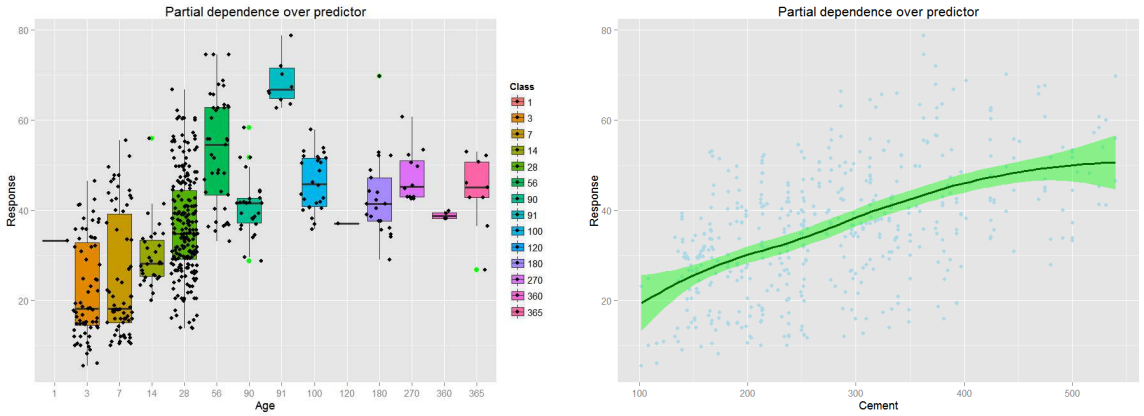

Figure 11: Partial Dependencies for the *Concrete compressive strength* dataset.

Partial dependencies show how the compressive strength is evolving, depending on a fixed variable and for all the possible known values of others variables. According to the definition given in section 5, the variable is fixed and we look in the whole forest what predictions are involved with this variable. For *Age*, we get an optimal value that maximizes the compressive strength, while it is an increasing function (on average) of the *Cement* values. Note that for *Age*, we choose to represent it as discrete values. Partial dependencies, in Random Uniform Forests, are designed to get the maximum number of points by using an additional layer on how the variable is assessed. This leads to get an highly flexible tool that provides interpolation, extrapolation (which requires to either train the model with a new paradigm or to combine parametric modeling of the tails and missing values imputation) and modeling dependencies (copula like model) for all the variables, while restricted to a pair in the current implementation.

Let us illustrate dependencies (using interpolation) of *Age* and *Cement* :

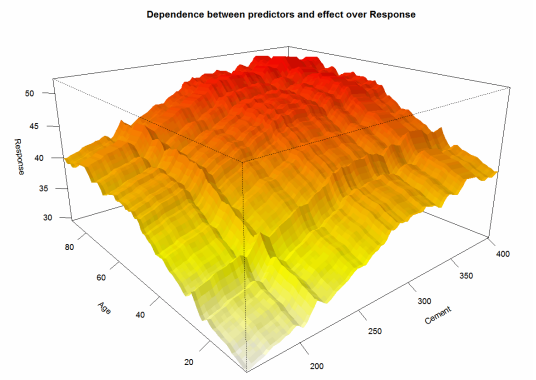

Figure 12: Partial dependencies of *Age* and *Cement* in the *Concrete compressive strength* dataset.

One can see how the dependence between the two variables lead, on average, to improve the compressive strength. Taking the variables independently would not lead to the same

improvements. Note that we removed outliers in order to get a consistent result. We could visualize all dependencies by pair of variables, but if the purpose of the task (other than prediction) is already known, one might focus on the most important variables to get the right direction, while the others would serve to limit variability.

7 Discussion

We provided, in this article, many analytical and visualization tools. Both types lead to a full analysis of Variable Importance. From what they are to how they act. We did not show all the possibilities of the tools provided, but examples shown go far in the analysis. To better understand Variable Importance one has to link it directly with predictions in the case of Random Uniform Forests. If they are pretty accurate then one may begin to have confidence to Variable Importance since all the measures are derived from either predictions or the learning process. While necessary, it is still not sufficient and the second strong guarantee relies on the stochastic nature of the model. One would, so, expect to get random results, implying uniform distribution of all the measures. As shown here, this does not happen, especially when learning many times the data, meaning that the main effect of Variable Importance is to show how influence is close or far to a model that would generate random influence. As statistical hypothesis are used in linear and parametric models, randomness is acting as the same level in Random Uniform Forests, separating noise to signal. Moreover, the main contribution has been to show how ensemble models were able to provide more details that any simple linear model with, at least, the same level of interpretation. *Global variable importance* was stated to describe *which* variables have, globally, the most influence on lowering the prediction error. *Local variable importance* describes *what* makes a variable to be an influential one, exploiting its *interactions* with the others. This leads to *Partial importance* which shows *when* a variable matters more. The last step of the Variable Importance analysis, *partial dependence*, defines *where* and/or *how* each variable is linked with the responses.

References

Bohanec M., Rajkovic, V., 1988. Knowledge acquisition and explanation for multi-attribute decision making. In *8th Intl Workshop on Expert Systems and their Applications*, Avignon, France. pages 59-78.

Bache, K., Lichman, M., 2013. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Breiman, L., Cutler, A. Variable Importance in Random Forests:
http://www.stat.berkeley.edu/∼breiman/RandomForests/cc_home.htmvarimp

Ciss, S., 2015a. Random Uniform Forests. <hal-01104340v2>

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29, 1189-1232.

Grömping, U., 2009. Variable Importance Assessment in Regression: Linear Regression versus Random Forest. *The American Statistician* 63, 308-319.

I-Cheng Yeh, 1998. "Modeling of strength of high performance concrete using artificial neural networks. *Cement and Concrete Research*, Vol. 28, No. 12, 1797-1808.

Hastie, T., Tibshirani, R., Friedman, J.J.H., 2001. *The elements of statistical learning.* New York: Springer.

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Shannon, C.E., 1949. *The Mathematical Theory of Communication.* University of Illinois Press.

Strobl, C., Boulesteix, A.-L., Zeileis, A., Hothorn, T., 2007. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* 8, 25.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., Zeileis, A., 2008. Conditional variable importance for random forests. *BMC Bioinformatics* 9, 307.

Vapnik, V.N., 1995. *The nature of statistical learning theory.* Springer-Verlag New York.

Zupan B., Bohanec, M., Bratko, I., Demsar, J., 1997. Machine learning by function decomposition. *ICML-97*, Nashville, TN. (to appear)