Interpretability

Jean-Michel Poggi

Master 2 Course in Statistics

Universidad de la República

Facultad de Ingeniería, Montevideo, Uruguay

February 2025

The presentation *"Random forests-based Variable importance measures"* from Pierre Geurts (Institut Montefiore, University of Liège, Belgium), ECAS-SFdS 2023 School, 8-13 September, 2023, inspired this course

Outline

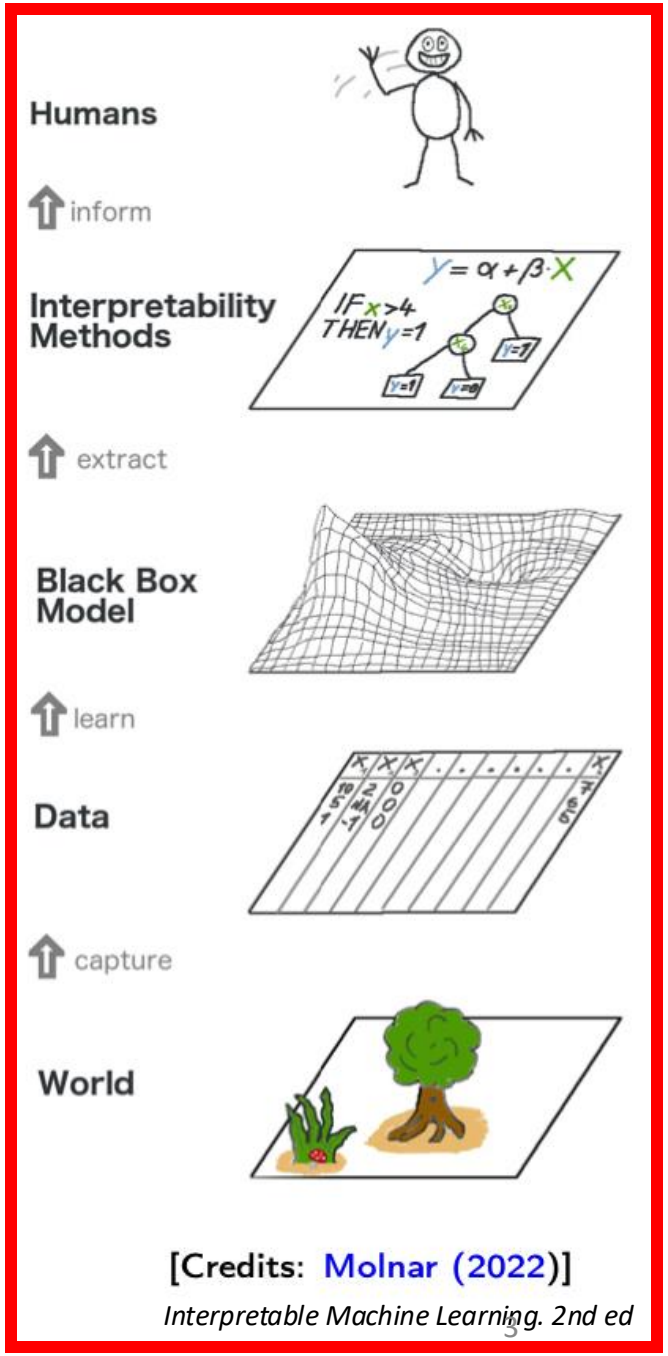
- Introduction
- Random Forests (RF)
- Importance scores: principles and usefulness
- Importance for RF, global measures
  - *Global MDI*
  - *Global MDA*
- Importance for RF, local measures
  - *Local MDI*
  - *Local MDA*
- Partial dependence plots
- Two packages: `pdp` and `randomForestExplainer`

Post-hoc interpretability methods

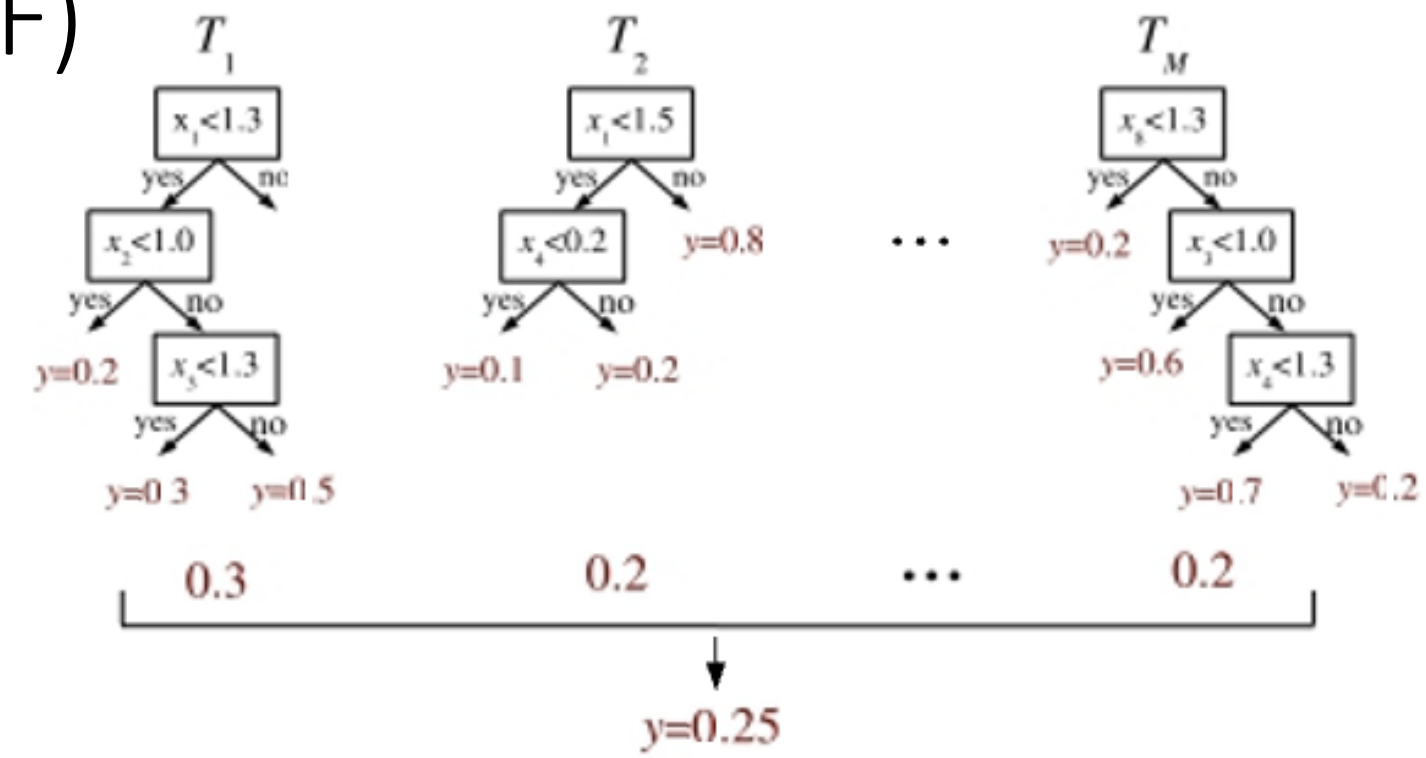
- Techniques to extract explanations from pre-trained black box models

- Explanations can take different forms:
  - variable importance
  - partial dependence plots
  - local linear model or tree, etc.

- Methods can be
  - specific to a model (adapted to a specific family of models)
  - agnostic (applicable to any black box model)

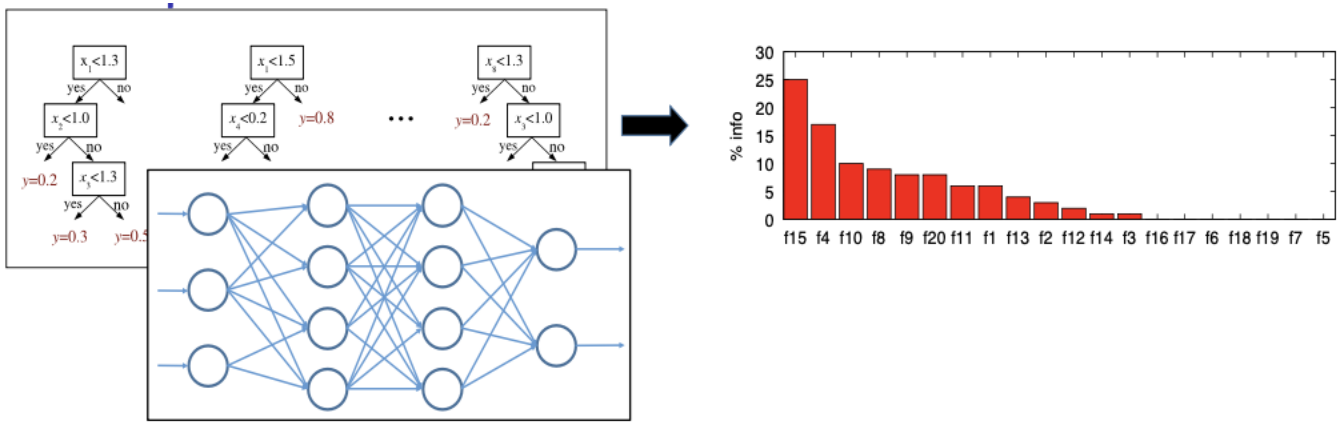


Random Forests (RF)



- Improve simple trees by reducing variance
- Breiman's random forests (2001) :
  - *Each tree is built from a bootstrap sample*
  - *The best split at each node is chosen from a number of inputs (mtry) selected (locally) at random*

Random forests: strengths and weaknesses

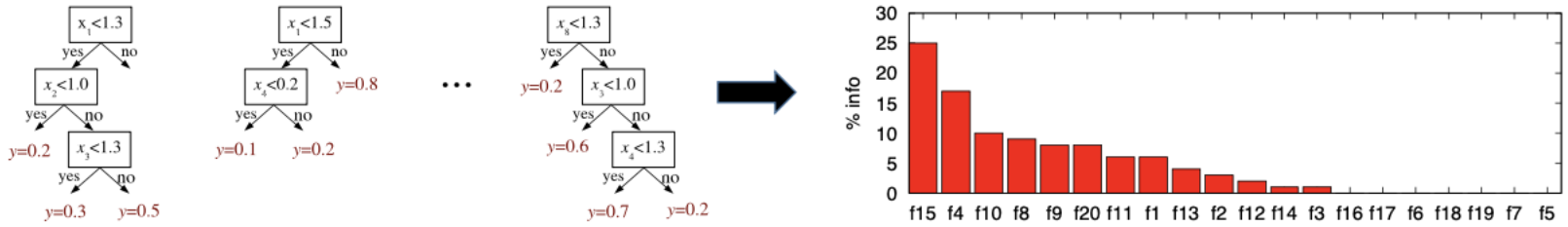


- Strengths:
  - Universal approximation
  - Robustness to outliers
  - Robustness to irrelevant variables (to a certain extent)
  - Invariant to input scale
  - Good computational efficiency and scalability
  - Very good predictive accuracy

- Weakness:
  - loss of interpretability compared with standard single trees

Importance scores: principle

- A numerical score that reflects the (relative) contribution of predictor variables to the model



- Used to evaluate the usefulness of a variable and to compare the contributions of two variables.

- Different categories of variable importance:
  - Model-specific or model-agnostic
  - Global (explaining the model as a whole) or local (explaining a specific prediction or set of predictions)

Importance of variables: what for?

- There is no single formal definition of the variable importance (VI), because the reasons for using them are diverse.

- Some common uses of VI measures:
  1. Inspection/debugging of an existing black box model (Model inspection)
  2. Find all the variables related to the response (sensitivity)
  3. Find the smallest subset of variables leading to optimal performance (variable selection for prediction)

- To assess the extent to which these objectives are being met, we need to understand the interaction between variable importance measures and the way in which the model is trained

Importance for RF
Global measures

Importance of variables for RF: why and how?

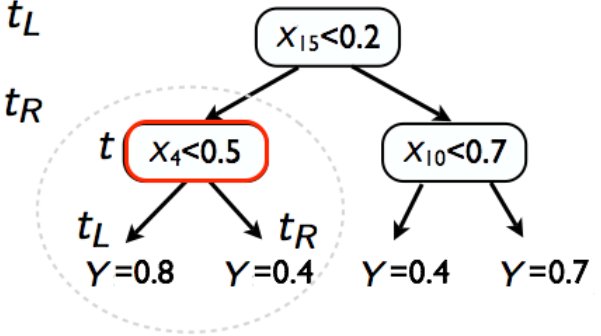
- RF (and more generally tree methods) are good candidates for deriving variable importances

- Node splitting is a variable selection mechanism

⋯	X_4	⋯	Y
⋯	⋯	⋯	⋯
⋯	-0.3	⋯	0.8
⋯	0.35	⋯	0.6
⋯	0.4	⋯	0.7
⋯	0.6	⋯	0.2
⋯	1.61	⋯	0.4
⋯	1.8	⋯	0.6
⋯	⋯	⋯	⋯

t, t_L, t_R

$$\Delta i(t) = \mathrm{Var}(t) - \frac{N_{t_L}}{N_t}\mathrm{Var}(t_L) - \frac{N_{t_L}}{N_t}\mathrm{Var}(t_R)$$
$$= 0.047 - \frac{3}{6}\cdot 0.01 - \frac{3}{6}\cdot 0.04 = 0.022$$



Importance of variables for RF: why and how?

Two main measures of importance:

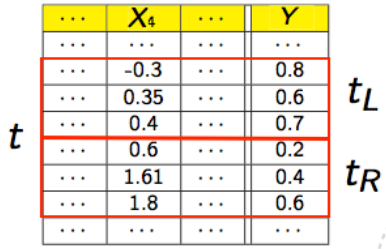
- Mean Decrease Impurity (MDI) :
  sum of the total reductions in the impurity at all the nodes of the tree where the variable is present (Breiman et al., CART, 1984)

- Mean Decrease Accuracy (MDA) :
  measure of the reduction in prediction error on OOB samples when the values of the variables are randomly permuted (Breiman, RF, 2001)

Global MDI

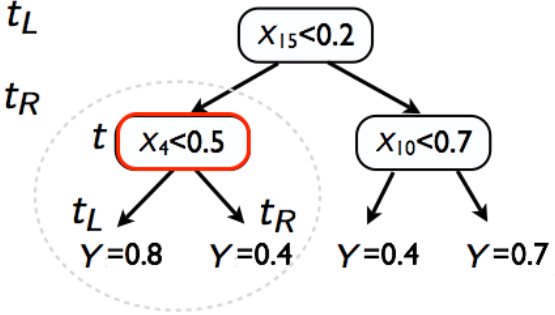
Mean Decrease Impurity (MDI)

- Originally proposed for single decision trees (Breiman et al., CART, 1984), it naturally applies to RF. Also known as the Gini Importance



⋯	X_4	⋯	Y
⋯	⋯	⋯	⋯
⋯	-0.3	⋯	0.8
⋯	0.35	⋯	0.6
⋯	0.4	⋯	0.7
⋯	0.6	⋯	0.2
⋯	1.61	⋯	0.4
⋯	1.8	⋯	0.6
⋯	⋯	⋯	⋯

$$\Delta i(t) = \mathrm{Var}(t) - \frac{N_{t_L}}{N_t}\mathrm{Var}(t_L) - \frac{N_{t_L}}{N_t}\mathrm{Var}(t_R)$$
$$= 0.047 - \frac{3}{6}\cdot 0.01 - \frac{3}{6}\cdot 0.04 = 0.022$$

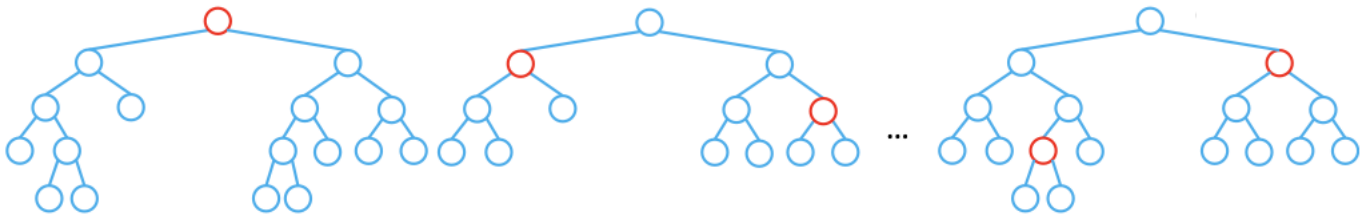


- RF model-specific and global
- Purely heuristic at the beginning, but there are theoretical justifications

Mean Decrease Impurity (MDI)

The idea is simple:

- sum the impurity reductions on all the nodes of a tree where the variable is used to split (in red)
- then average over all trees (en bleu)



$$VI_{\text{MDI}}(X_m) = \frac{1}{M}\sum_{T}\sum_{t\in T:v(t)=X_m} p(t)\Delta i(t)$$

Global MDA

Mean Decrease Accuracy (MDA)

- An alternative measure proposed by Breiman (2001) in his article on random forests
- Model independent (agnostic) and global
- Idea: measure the degradation of the model's performance when the variable is permuted

- There are different variants depending on how performance is estimated and how importances are aggregated across trees.
- Intuitive and heuristic, but there are also theoretical justifications
- More widely used than MDI

Mean Decrease Accuracy (MDA)

- Idea: measure the degradation of the model's performance when the variable is permuted

Original dataset $\mathcal{D}$

X_1	X_2	...	X_m	...	X_p	Y
1.4	3.4	...	8.3	...	5.8	2.3
9.5	8.0	...	6.9	...	7.8	5.4
5.2	9.8	...	0.9	...	1.9	6.3
0.7	7.6	...	4.5	...	6.4	3.3
4.7	2.0	...	2.1	...	4.5	9.9

$\sum_{(x,y)\in\mathcal{D}} \mathrm{L}(f(x), y)$

Permuted dataset $\tilde{\mathcal{D}}_m$

X_1	X_2	...	X_m	...	X_p	Y
1.4	3.4	...	0.9	...	5.8	2.3
9.5	8.0	...	4.5	...	7.8	5.4
5.2	9.8	...	2.1	...	1.9	6.3
0.7	7.6	...	6.9	...	6.4	3.3
4.7	2.0	...	8.3	...	4.5	9.9

$\sum_{(x,y)\in\tilde{\mathcal{D}}_m} \mathrm{L}(f(x), y)$

$$VI_{\mathrm{MDA}}(X_m; f, \mathcal{D}, R) = \left[\frac{1}{R}\sum_{r=1}^{R} \sum_{(x,y)\in\tilde{\mathcal{D}}_m^r} \mathrm{L}(f(x), y)\right] - \sum_{(x,y)\in\mathcal{D}} \mathrm{L}(f(x), y)$$

Mean Decrease Accuracy (MDA): how can it be improved?

- MDA for RF: choice of dataset
- Estimating the MDA on the learning set can lead to overestimated importances (overfitting)

- There are two ways of estimating it more accurately:
  - on an independent test set
  - with the OOB sample. In the latter case, the permutation is performed only on the OOB sample

Original OOB sample $\mathcal{D}$

X_1	X_2	...	X_m	...	X_p	Y
1.4	3.4	...	8.3	...	5.8	2.3
9.5	8.0	...	6.9	...	7.8	5.4
5.2	9.8	...	0.9	...	1.9	6.3
0.7	7.6	...	4.5	...	6.4	3.3
4.7	2.0	...	2.1	...	4.5	9.9

Permuted OOB sample $\tilde{\mathcal{D}}_m$

X_1	X_2	...	X_m	...	X_p	Y
1.4	3.4	...	8.3	...	5.8	2.3
9.5	8.0	...	4.5	...	7.8	5.4
5.2	9.8	...	0.9	...	1.9	6.3
0.7	7.6	...	6.9	...	6.4	3.3
4.7	2.0	...	2.1	...	4.5	9.9

Main features of MDI and MDA measures

- MDI :
  - RF model specific. Very closely connected to the tree construction algorithm. No additional computational cost
  - Suffers from certain known biases, probably more so than MDA

- MDA :
  - Model agnostic. Does not take into account tree specificities. Slower than MDI but still quick to calculate (no need to re-train a model)
  - Suffers from some known biases

- The two importances are mainly heuristic and lack a clear theoretical characterisation (although research is progressing)

Importance for RF
Local measures

Local (or individual) measures

- So far, we have only talked about global measures (explaining the model as a whole)
- The MDI or MDA of a variable is calculated for a previously estimated RF

- Local versions of these importance measures can be defined for
  - explain a specific prediction
  - explain a set of predictions

- We calculate the MDI or MDA of an observation

Local MDA

Local MDA

- MDA global

$$VI_{\text{MDA}}(X_m; f, \mathcal{D}, R) = [\frac{1}{R}\sum_{r=1}^{R}\sum_{(x,y)\in\tilde{\mathcal{D}}_m^r} \text{L}(f(x), y)] - \sum_{(x,y)\in\mathcal{D}} \text{L}(f(x), y)$$

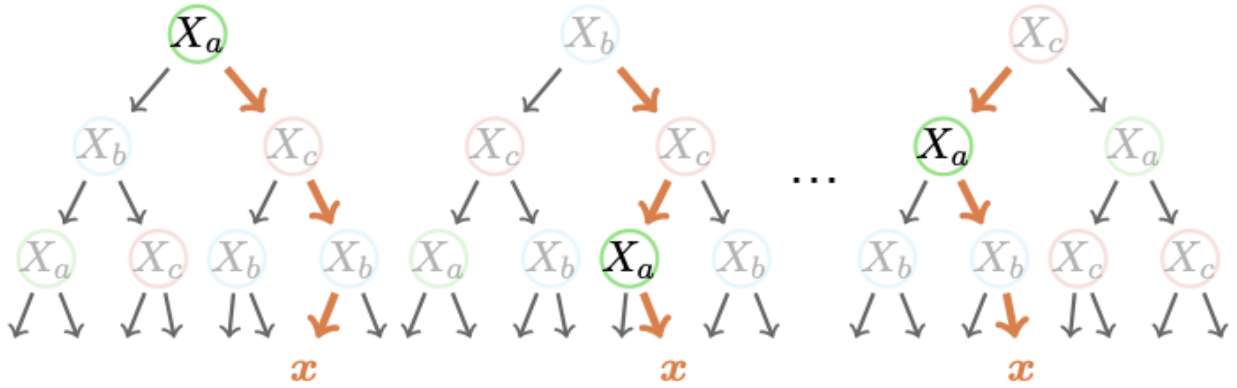
- MDA local

$$VI_{\text{MDA}}(X_m, x; f, \mathcal{D}, R) = [\frac{1}{R}\sum_{r=1}^{R} \text{L}(f(\tilde{x}_r), y)] - \text{L}(f(x), y),$$

- The sum of the individual MDAs over the learning sample = the overall MDA

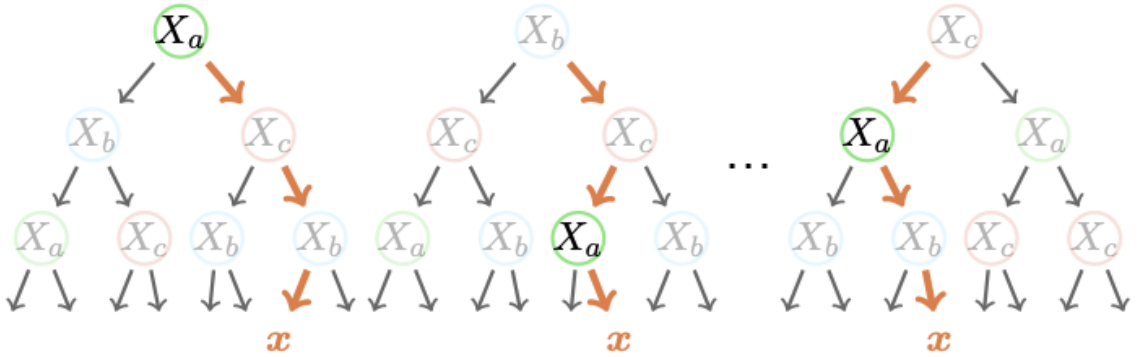
Local MDI

Local MDI



$$VI_{\text{MDI}}(X_a, x) = \frac{1}{M} \sum_{T} \sum_{t \in T : \nu(t) = X_a \wedge x \in t} i(t) - i(t_{x_a})$$

Local MDI



$$VI_{\text{MDI}}(X_a, x) = \frac{1}{M} \sum_{T} \sum_{t \in T : \nu(t) = X_a \wedge x \in t} i(t) - i(t_{x_a})$$

- The sum over the learning sample of the individual MDIs = the overall MDI

$$\sum_{(x,y) \in \mathcal{D}} VI_{\text{MDI}}(X_m, x) = VI_{\text{MDI}}(X_m)$$

Partial dependence plots (pdp)

Two packages:
`pdp` and `randomForestExplainer`

pdp

Package 'pdp'

October 14, 2022

Type Package

Title Partial Dependence Plots

Version 0.8.1

Description A general framework for constructing partial dependence (i.e., marginal effect) plots from various types machine learning models in R.

RF (in regression) on Boston data

The Boston Housing Dataset is a derived from information collected by the U.S. Census Service concerning housing in the area of Boston MA. The following describes the dataset columns:

- CRIM - per capita crime rate by town
- ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS - proportion of non-retail business acres per town.
- CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX - nitric oxides concentration (parts per 10 million)
- RM - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centres
- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per $10,000
- PTRATIO - pupil-teacher ratio by town
- LSTAT - % lower status of the population
- MEDV - Median value of owner-occupied homes in $1000's

RF (in regression) on Boston data

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Two importance measures

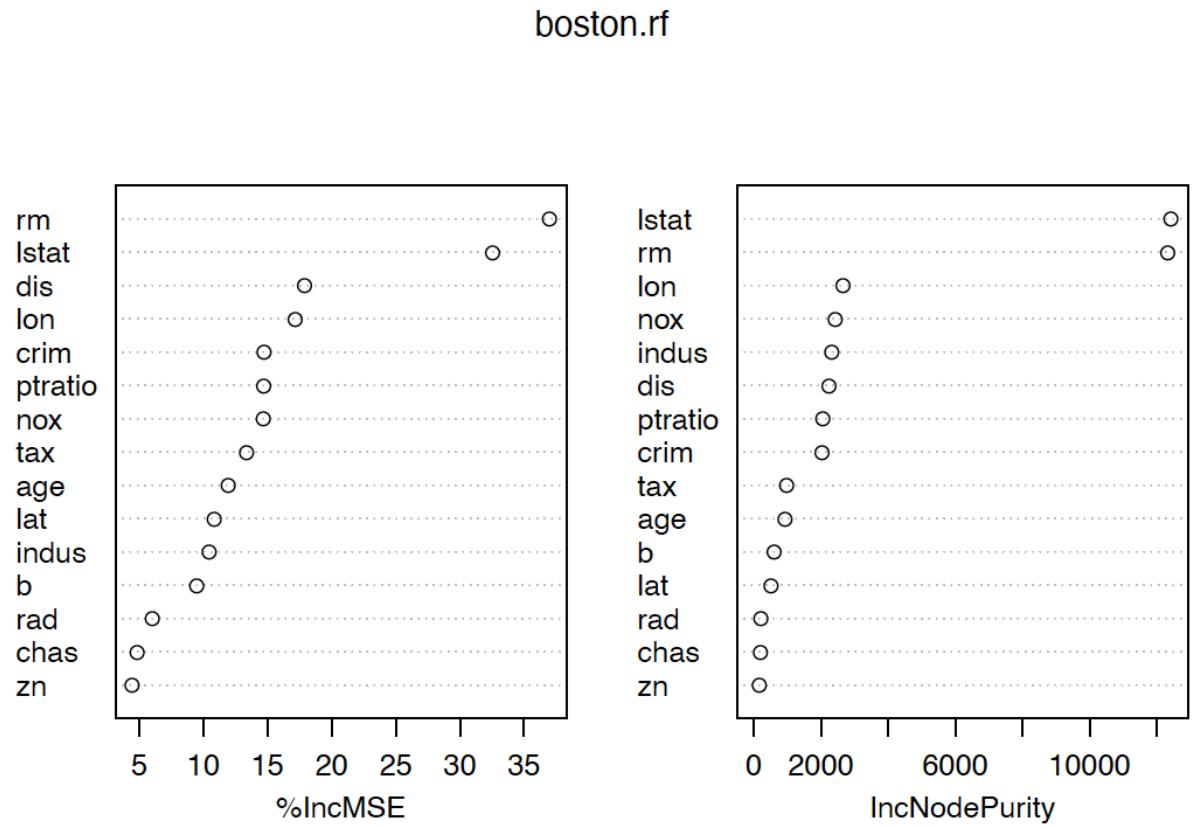


Figure 1: Dotchart of variable importance scores for the Boston housing data based on a random forest with 500 trees.

Partial dependence plots (pdp): principle

Let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_p\}$ represent the predictors in a model whose prediction function is $\widehat{f}(\boldsymbol{x})$. If we partition $\boldsymbol{x}$ into an interest set, z_s, and its compliment, $z_c = \boldsymbol{x} \setminus z_s$, then the "partial dependence" of the response on z_s is defined as

$$f_s(z_s) = E_{z_c}\left[\widehat{f}(z_s, z_c)\right] = \int \widehat{f}(z_s, z_c)\, p_c(z_c)\, dz_c, \tag{2}$$

where $p_c(z_c)$ is the marginal probability density of z_c: $p_c(z_c) = \int p(\boldsymbol{x})\, dz_s$. Equation (2) can be estimated from a set of training data by

$$\bar{f}_s(z_s) = \frac{1}{n}\sum_{i=1}^{n} \widehat{f}(z_s, z_{i,c}), \tag{3}$$

The marginal effects

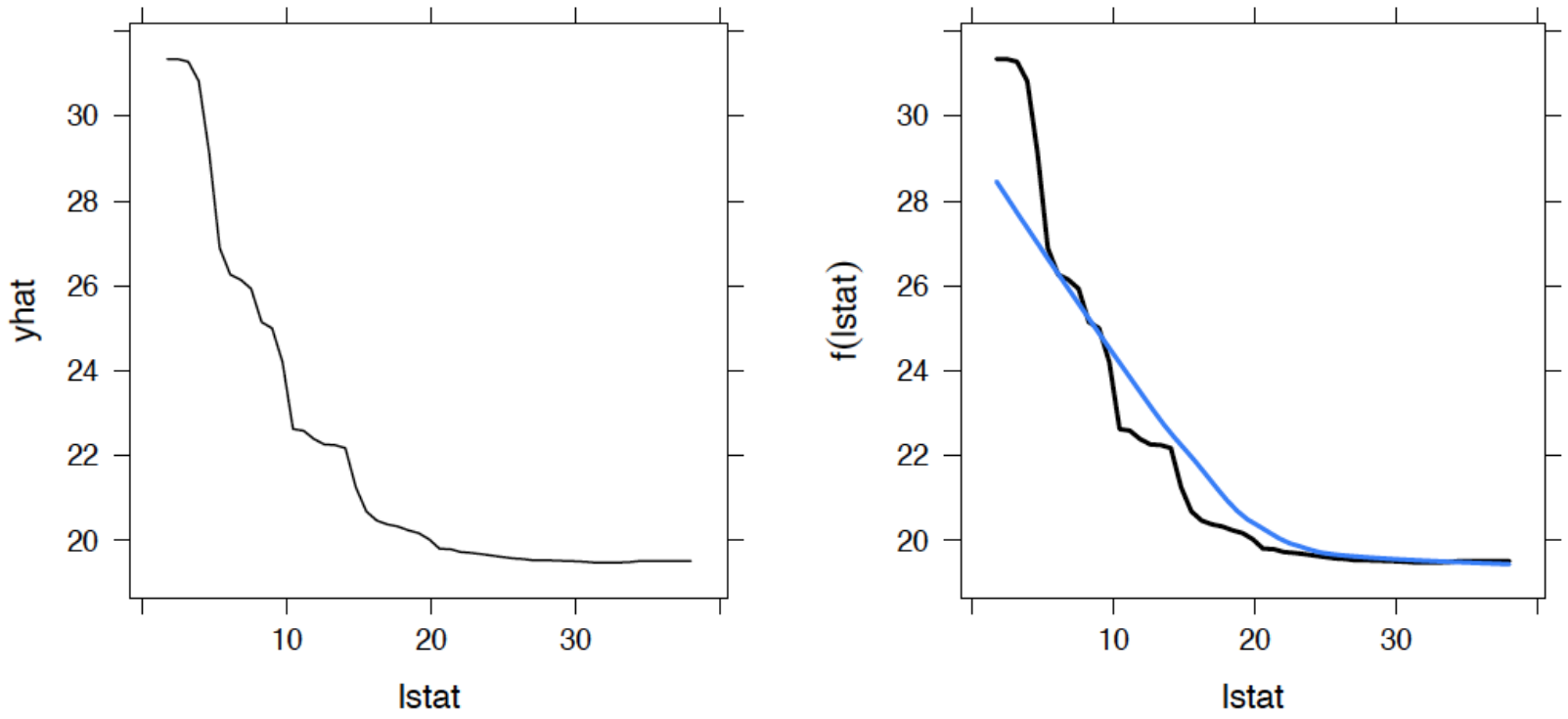


Figure 2: Partial dependence of `cmedv` on `lstat` based on a random forest. *Left*: Default plot. *Right*: Customized plot obtained using the `plotPartial` function.

Partial dependence plots

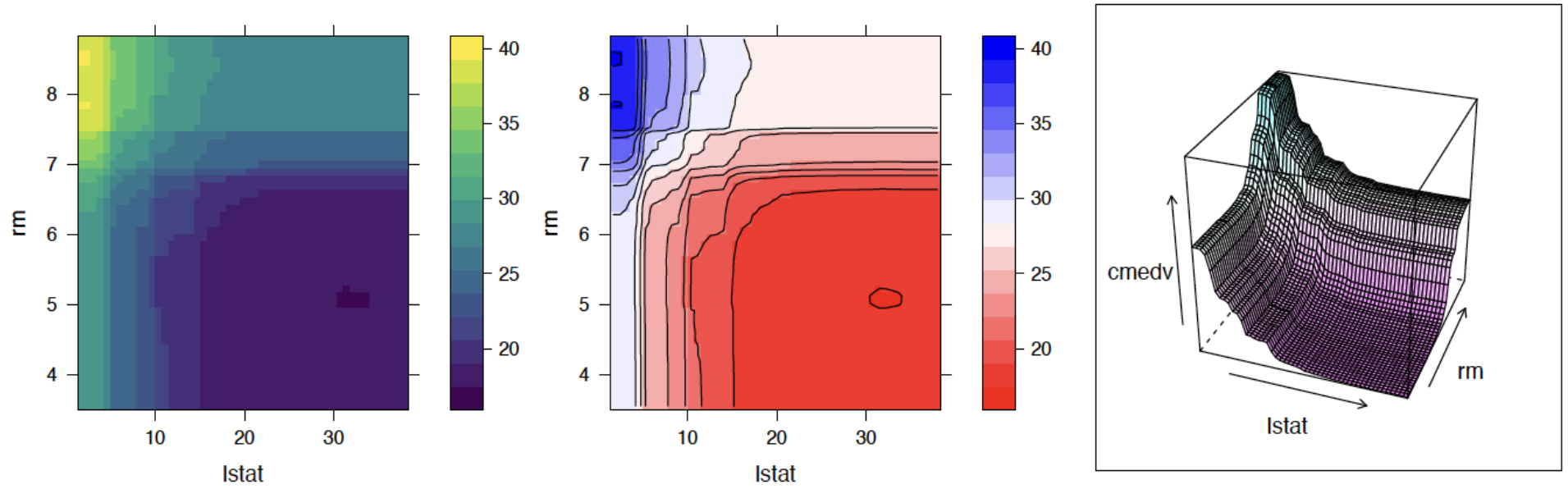


Figure 3: Partial dependence of `cmedv` on `lstat` and `rm` based on a random forest. *Left*: Default plot. *Middle*: With contour lines and a different color palette. *Right*: Using a 3-D surface.

randomForestExplainer

Package 'randomForestExplainer'

October 14, 2022

Title Explaining and Visualizing Random Forests in Terms of Variable Importance

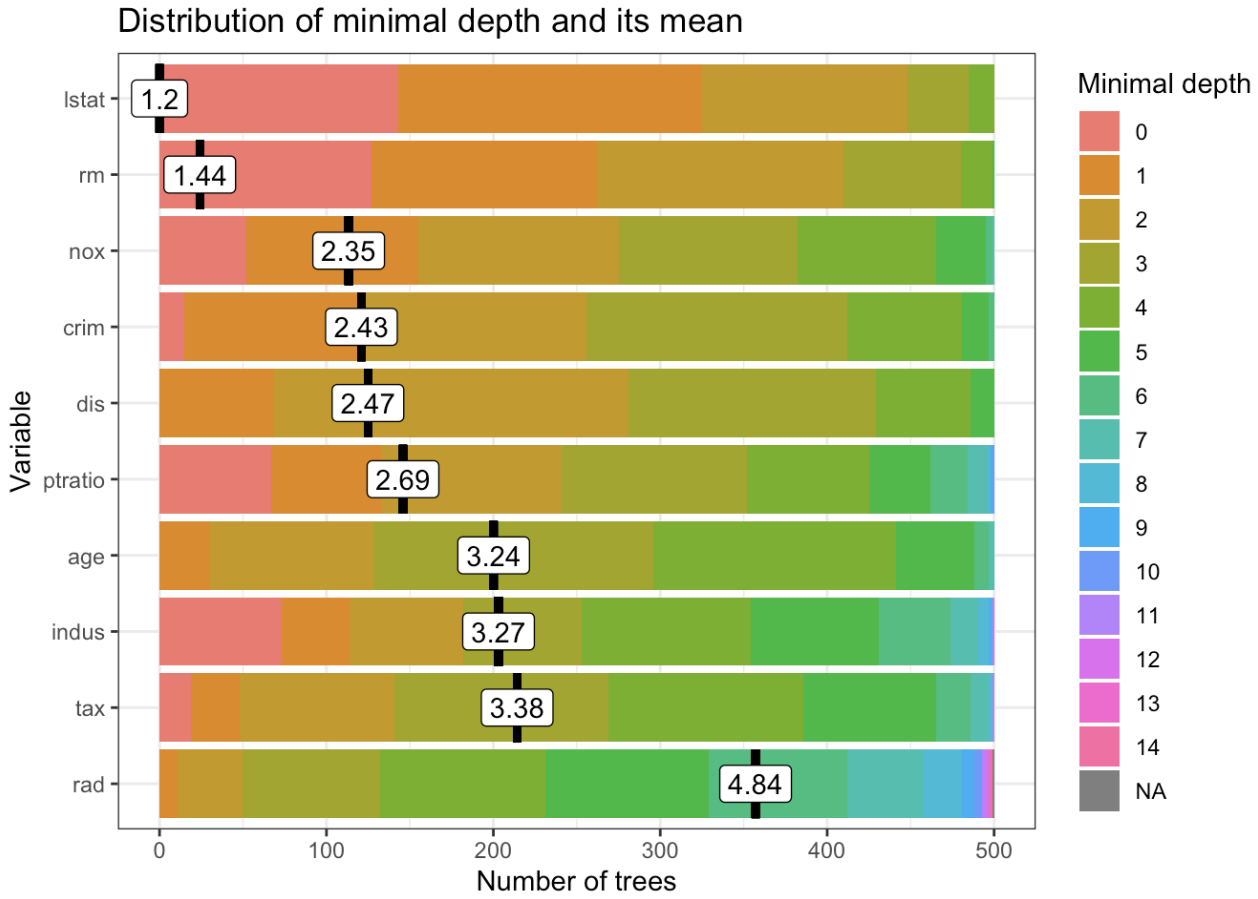
Version 0.10.1

Description A set of tools to help explain which variables are most important in a random forests. Various variable importance measures are calculated and visualized in different settings in order to get an idea on how their importance changes depending on our criteria (Hemant Ishwaran and Udaya B. Kogalur and Eiran Z. Gorodeski and Andy J. Minn and Michael S. Lauer (2010) <doi:10.1198/jasa.2009.tm08622>, Leo Breim

RF (in regression) on Boston data

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

The ‘minimum depth’ of a variable



The minimum depth of a variable in a tree is equal to the depth of the node that splits on that variable and is closest to the root of the tree.

The lower it is, the greater the number of observations divided into groups on the basis of this variable.

Importance measures

Importance measures

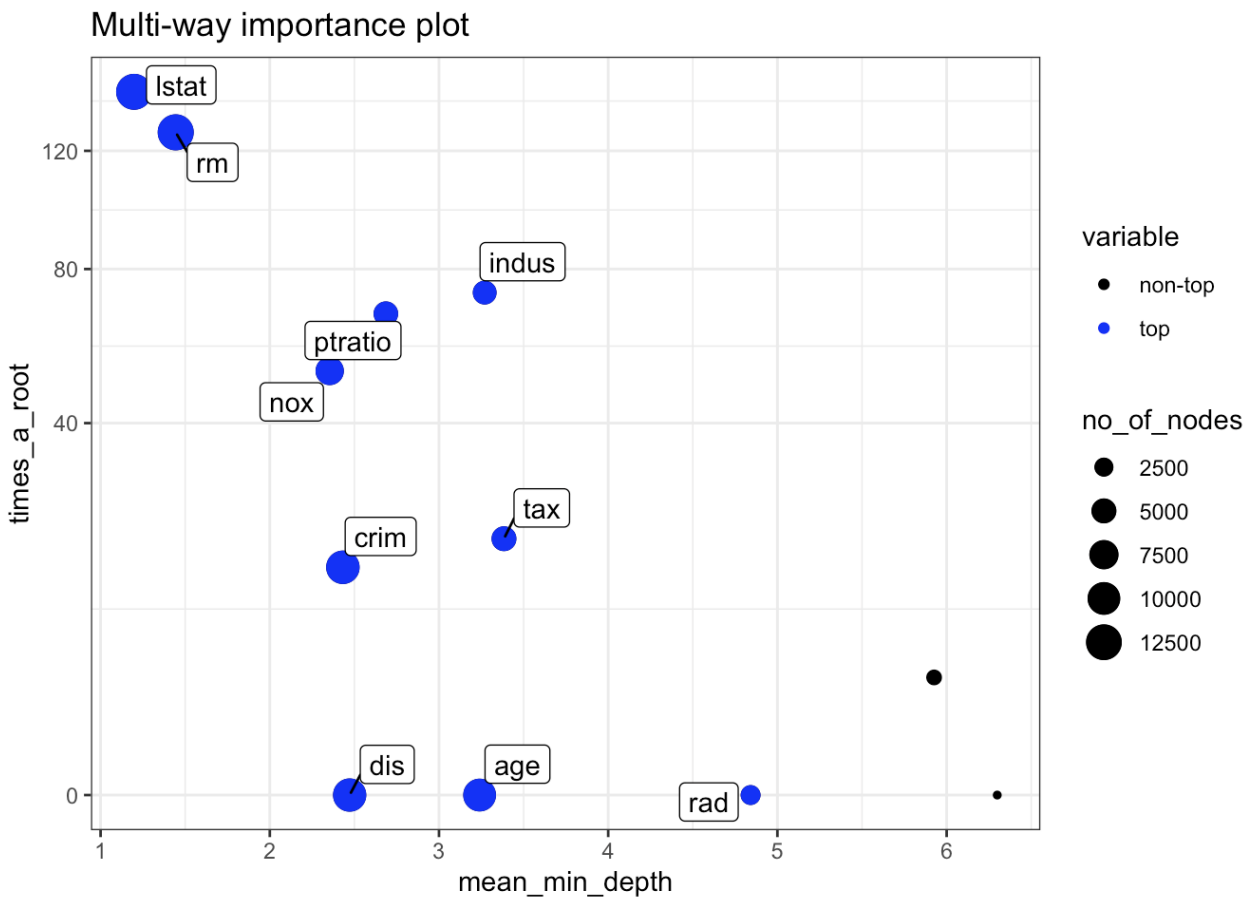
Below you can explore the measures of importance for all variables in the forest:

Show 25 entries Search:

	variable	mean_min_depth	no_of_nodes	mse_increase	node_purity_increase	no_of_trees	times_a_root	p_value
1	age	3.2400	10052	3.6901	1,065.3133	500	0	0.0000
2	chas	6.2991	890	0.4949	258.5535	427	0	1.0000
3	crim	2.4320	10591	8.9529	2,417.3690	500	15	0.0000
4	dis	2.4720	10426	7.3728	2,608.5888	500	0	0.0000
5	indus	3.2700	4431	6.3945	2,618.9056	500	73	1.0000
6	lstat	1.1980	12624	60.1779	12,829.4472	500	143	0.0000
7	nox	2.3540	6919	10.9807	2,996.0430	500	52	0.4489
8	ptratio	2.6860	4803	7.5179	2,712.0078	500	67	1.0000
9	rad	4.8414	2812	1.1381	301.6931	499	0	1.0000
10	rm	1.4440	12745	33.7212	12,302.2741	500	127	0.0000
11	tax	3.3840	4953	3.8943	1,276.1128	500	19	1.0000
12	zn	5.9259	1655	0.5324	260.3035	482	4	1.0000

Showing 1 to 12 of 12 entries Previous 1 Next

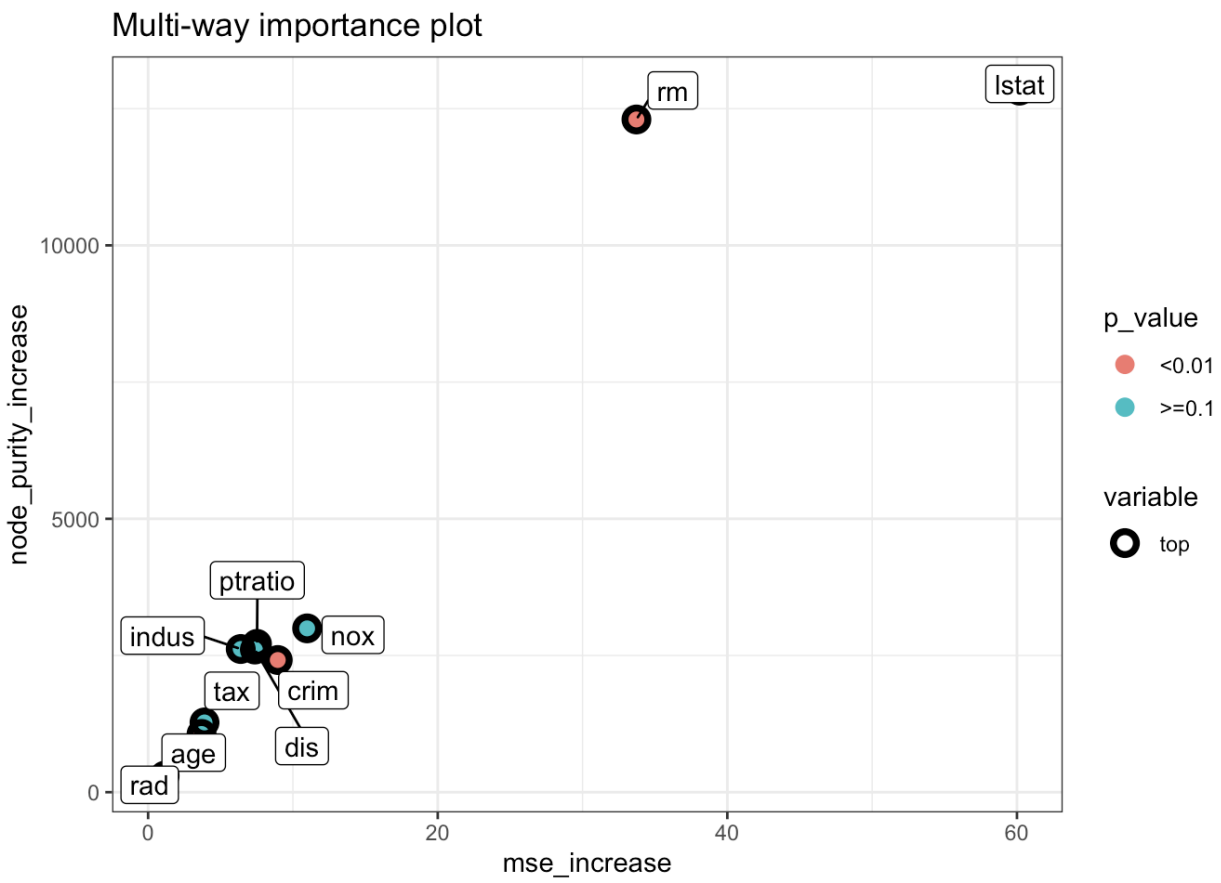
Multi-way importance plot (1)



This 1st multi-way importance plot focuses on three measures of importance that derive from the structure of the trees in the forest:

- the *average depth* of the first split on the variable
- the number of trees whose *root is split on the variable*
- the total number of nodes in the *forest that split on this variable*

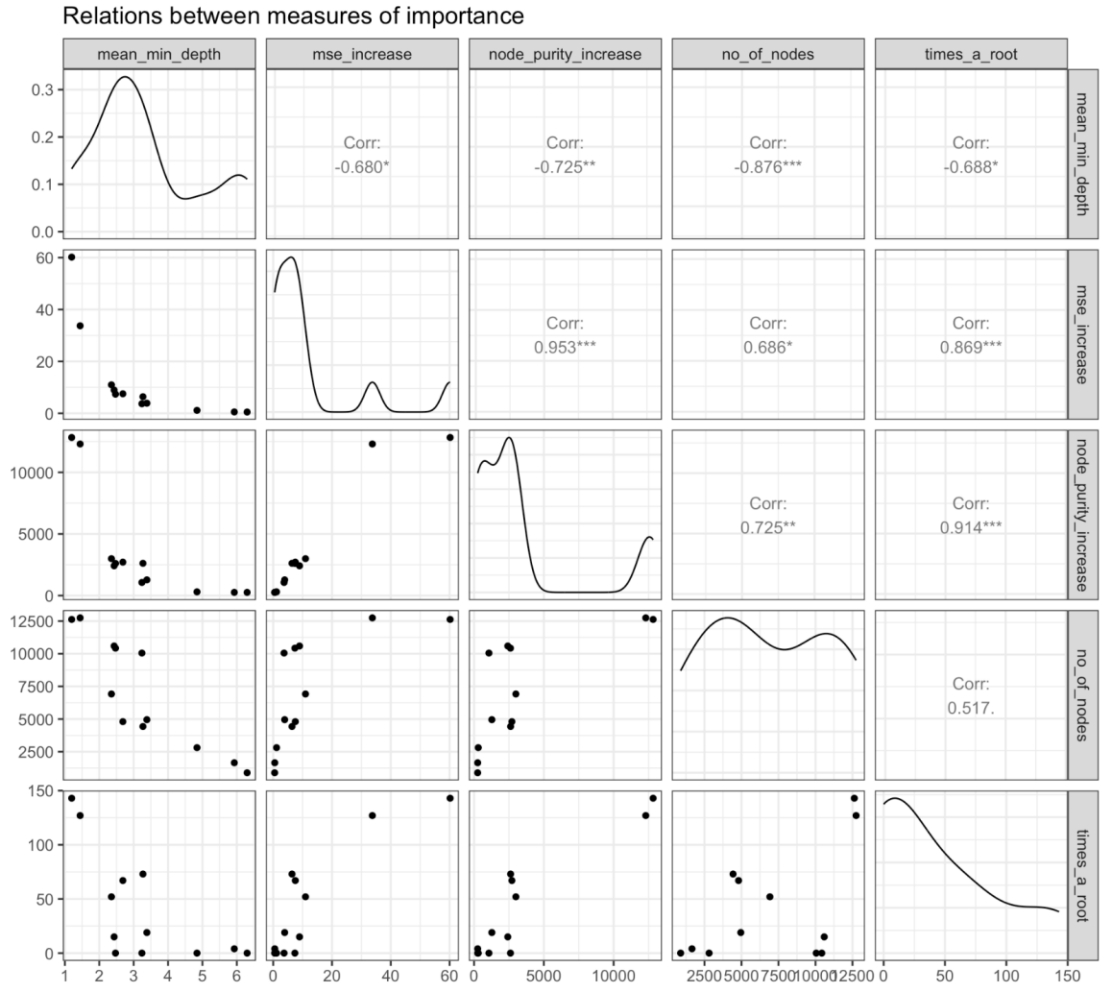
Multi-way importance plot (2)



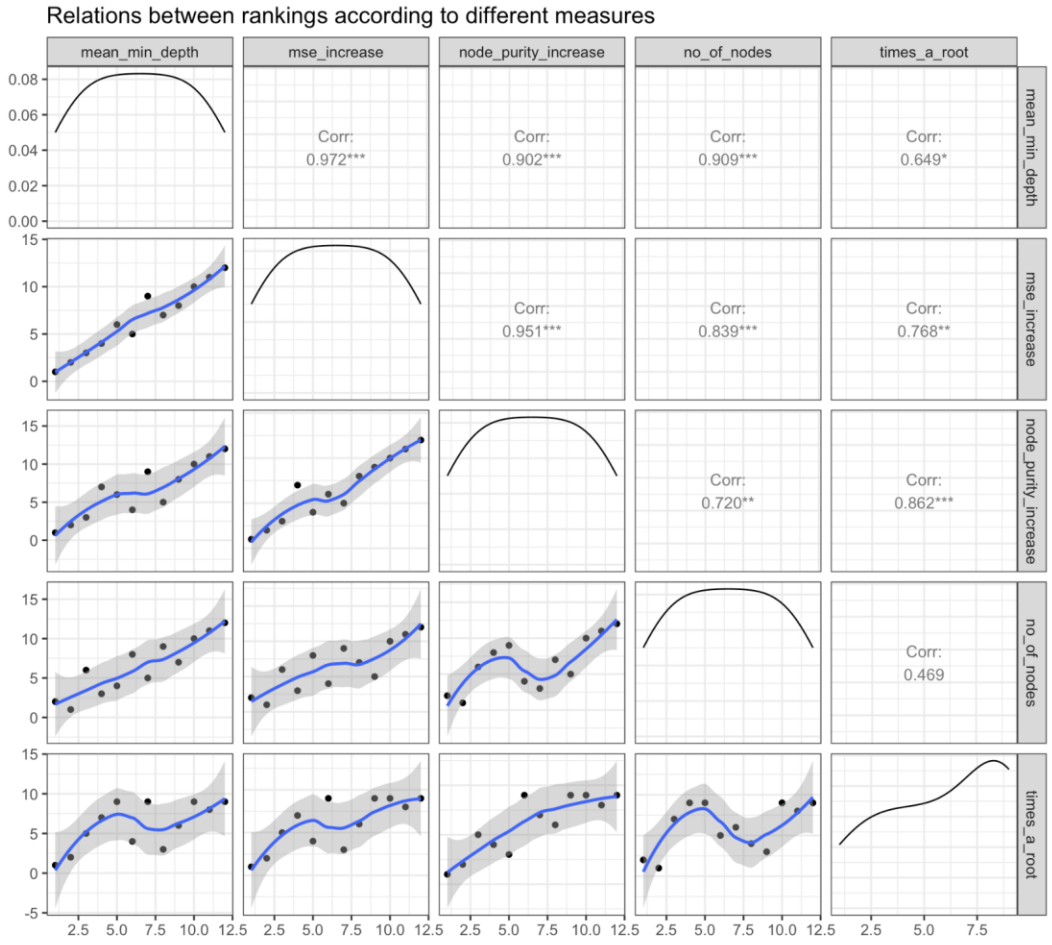
The 2nd multi-way importance plot shows two measures of importance which derive from the role played by a variable in the prediction

with *p-value information* based on a binomial distribution of the number of nodes split on the variable, assuming that variables are randomly selected to form splits
(i.e. if a variable is significant, this means that the variable is used for splitting more often than if the selection was random)

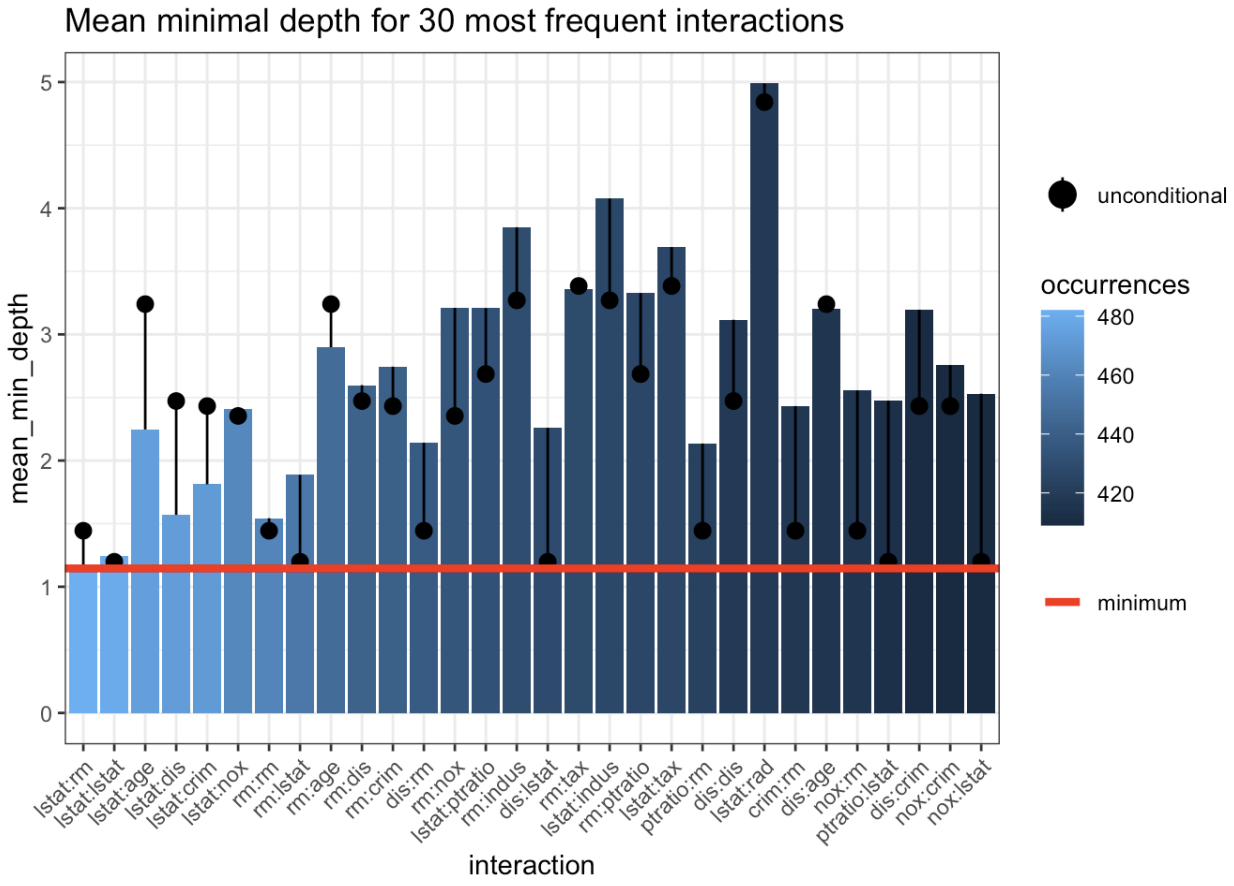
Compare VI measures using `ggpairs`



Compare different rankings

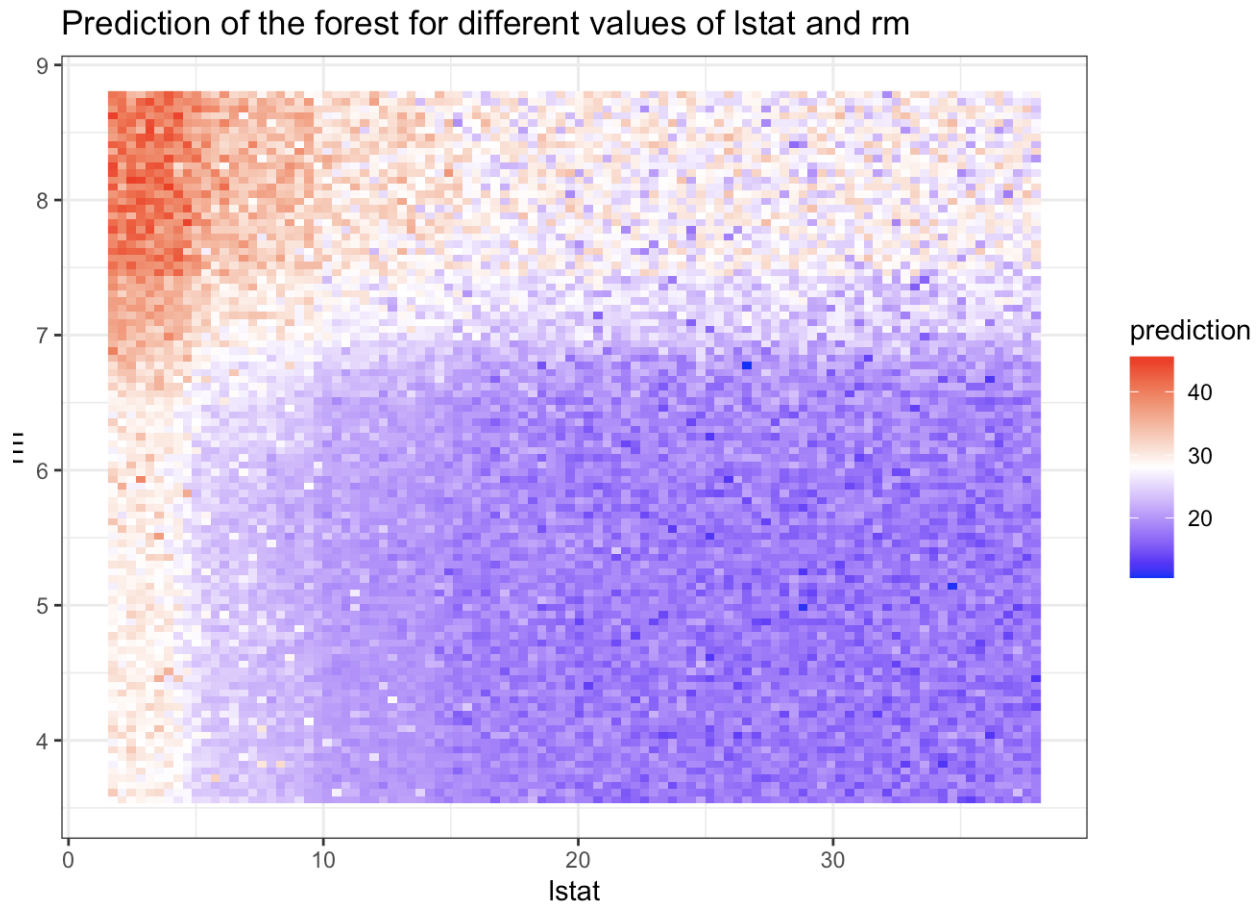


Interactions between variables



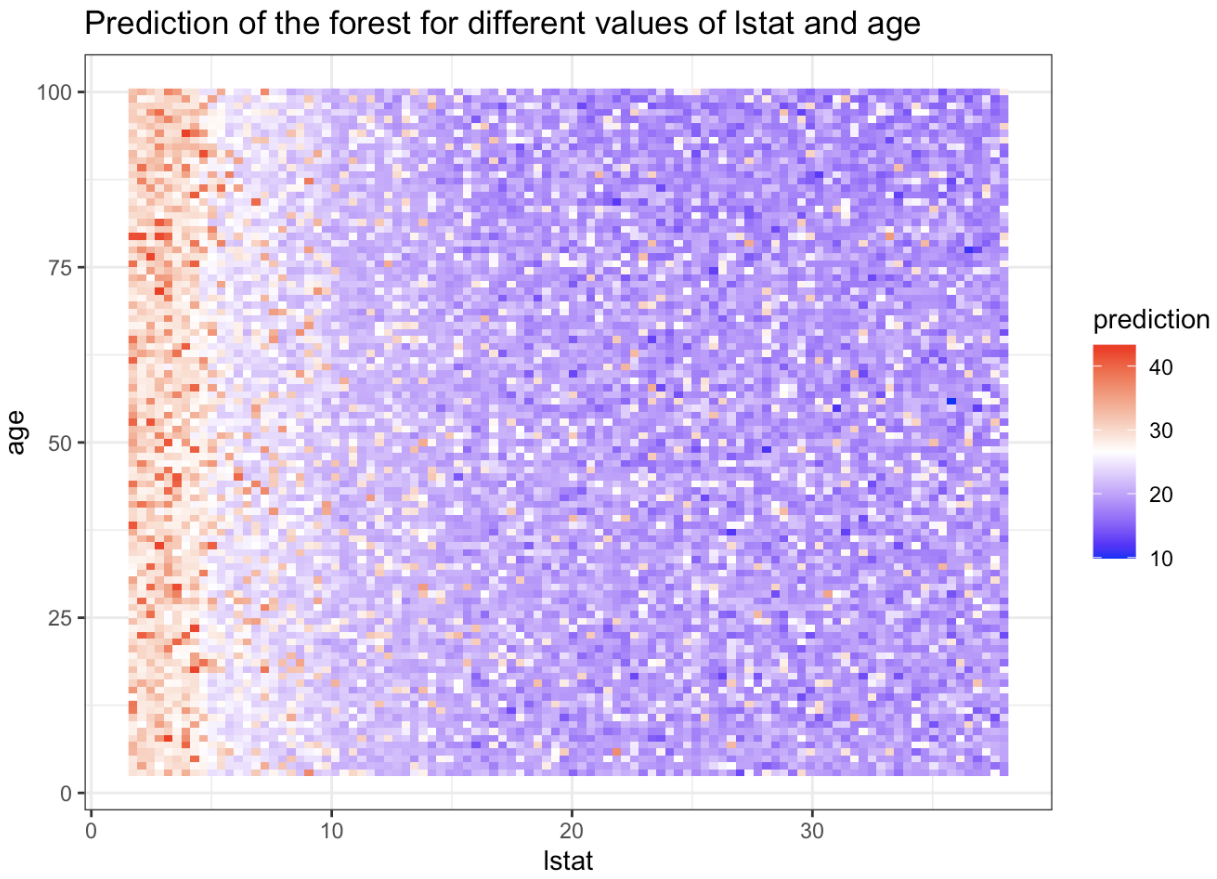
Once we have selected a set of the most important variables, we can study the interactions in relation to them, i.e. the splits appearing in the maximum sub-trees in relation to one of the selected variables

Forest forecast on a grid



To further study the most frequent interaction `lstat:rm`, we use the `plot_predict_interaction` function to plot the forest prediction on a grid of values for the components of each interaction

Forest forecast on a grid (2)



To further study another frequent interaction `lstat:age`, we use again the `plot_predict_interaction` function