Package: randomForestVIP (via r-universe)

October 26, 2024

Type Package

Title Tune Random Forests Based on Variable Importance and Plot Results

Version 0.1.3.9000

Description Functions for assessing variable relations and associations prior to modeling with a Random Forest algorithm (although these are relevant for any predictive model). Metrics such as partial correlations and variance inflation factors are tabulated as well as plotted for the user. A function is available for tuning the main Random Forest hyper-parameter based on model performance and variable importance metrics. This grid-search technique provides tables and plots showing the effect of the main hyper-parameter on each of the assessment metrics. It also returns each of the evaluated models to the user. The package also provides superior variable importance plots for individual models. All of the plots are developed so that the user has the ability to edit and improve further upon the plots. Derivations and methodology are described in Bladen (2022) <https://digitalcommons.usu.edu/etd/8587/>.

License GPL-3

URL https://github.com/KelvynBladen/randomForestVIP

Depends R (>= 4.0.0)

Imports car, caret, dplyr, gbm, ggeasy, ggplot2, gridExtra, methods, minerva, pdp, randomForest, rlang, stats, tidyr, trelliscopejs

Suggests EZtune, e1071, knitr, MASS, rmarkdown, rpart, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Repository https://kelvynbladen.r-universe.dev

RemoteUrl https://github.com/kelvynbladen/randomforestvip

RemoteRef HEAD

RemoteSha b9c5f647bc455cf3eff2eb6225d5acae83f7caa6

Contents

`boston` *Housing Values in Suburbs of Boston*

Description

The Boston data frame has 506 rows and 14 columns.

Usage

```
boston
```

Format

This data frame contains the following 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 nitrogen 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 mean of 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.

black $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town.

lstat lower status of the population (percent).

medv median value of owner-occupied homes in \$1000s.

Source

https://www.stats.ox.ac.uk/pub/MASS4/

---

caret_plot *Plot Caret Grid Search Hyper-parameter Tuning Results*

Description

This function uses caret grid training results to generate performance data.frames, heatmaps, and other plots for comparing the performance of models across their hyper-parameters and evaluating interactions between different model hyper-parameters.

Usage

```
caret_plot(
  x,
  sqrt = FALSE,
  marg1 = FALSE,
  marg2 = FALSE,
  col = NULL,
  row = NULL,
  facet = NULL
)
```

Arguments

`x`	An object of class train.
`sqrt`	Boolean value indicating whether assessment metrics should be adjusted via a square root transformation. Default is FALSE.
`marg1`	Boolean value indicating whether to aggregate performance down to 1 dimension of hyper-parameter and provide the corresponding data.frames and lineplots for assessment. Default is FALSE.
`marg2`	Boolean value indicating whether to aggregate performance down to 2 dimensions of hyper-parameter and provide the corresponding data.frames and heatmaps for assessment. Default is FALSE.
`col`	Name of the variable to plot on the columns of heatmaps. Only relevant for heatmaps of 3 or more dimensions. Default is NULL.
`row`	Name of the variable to plot on the rows of heatmaps. Only relevant for heatmaps of 3 or more dimensions. Default is NULL.
`facet`	Name of the variable to plot as the facets of heatmaps. Only relevant for heatmaps of 4 dimensions. Default is NULL.

Value

A list of caret training performance data.frames, heatmaps, and plots.

Examples

```
set.seed(123)
fit_control <- caret::trainControl(method = "cv", number = 10)
gbm_grid <- expand.grid(
  interaction.depth = c(1, 4), n.trees = c(15, 150),
  shrinkage = c(0.05, 0.1), n.minobsinnode = 10
)
x <- caret::train(factor(Species) ~ .,
  method = "gbm", tuneGrid = gbm_grid,
  trControl = fit_control, data = iris
)
p <- caret_plot(x, sqrt = FALSE, col = "n.trees", marg1 = TRUE, marg2 = TRUE)
p
```

ggvip *Variable Importance GGPlot*

Description

A ggplot of variable importance as measured by a Random Forest.

Usage

```
ggvip(x, scale = FALSE, sqrt = TRUE, type = "both", num_var)
```

Arguments

x	An object of class randomForest.
scale	For permutation based measures such as MSE or Accuracy, should the measures be divided by their "standard errors"? Default is False.
sqrt	Boolean value indicating whether importance metrics should be adjusted via a square root transformation. Default is True.
type	either 1 or 2, specifying the type of importance measure (1=mean decrease in accuracy or node impurity or mean decrease in gini). Default is "both".
num_var	Optional argument for reducing the number of variables to the top 'num_var'. Must be an integer between 1 and the total number of predictor variables in the model.

Value

A ggplot dotchart showing the importance of the variables that were plotted.

Examples

```
rf <- randomForest::randomForest(factor(Species) ~ .,
  importance = TRUE,
  data = iris
)
ggvip(rf, scale = FALSE, sqrt = TRUE, type = "both")
```

`lichen` *Lichen data from the Current Vegetation Survey*

Description

Data were collected between 1993 and 1999 as part of the Lichen Air Quality surveys on public lands in Oregon and southern Washington. Observations were obtained from 1-acre (0.4 ha) plots at Current Vegetation Survey (CVS) sites. Indicator variables denote the presences and absences of 7 lichen species. Data for each sampled plot include the topographic variables elevation, aspect, and slope; bioclimatic predictors including maximum, minimum, daily, and average temperatures, relative humidity precipitation, evapotranspiration, and vapor pressure; and vegetation variables including the average age of the dominant conifer and percent conifer cover. The data in lichenTest were collected from half-acre plots at CVS sites in the same geographical region and contains many of the same variables, including presences and absences for the 7 lichen species. As such, it is a good test dataset for predictive methods applied to the Lichen Air Quality data.

Usage

```
lichen
```

Format

A data frame with 840 observations and 40 variables. One variable is a location identifier, 7 (coded as 0 and 1) identify the presence or absence of a type of lichen species, and 32 are characteristics of the survey site where the data were collected.

There were 12 monthly values in the original data for each of the bioclimatic predictors. Principal components analyses suggested that for each of these predictors 2 principal components explained the vast majority (95.0%-99.5%) of the total variability. Based on these analyses, indices were created for each set of bioclimatic predictors. The variables with the suffix Ave in the variable name are the average of 12 monthly variables. The variables with the suffix Diff are contrasts between the sum of the April-September monthly values and the sum of the October-December and January-March monthly values, divided by 12. Roughly speaking, these are summer-to-winter contrasts.

The variables are summarized as follows:

LobaOreg Lobaria oregana (Absent = 0, Present = 1)

EvapoTransAve Average monthly potential evapotranspiration in mm

EvapoTransDiff Summer-to-winter difference in monthly potential evapotranspiration in mm

MoistIndexAve Average monthly moisture index in cm

MoistIndexDiff Summer-to-winter difference in monthly monthly moisture index in cm

PrecipAve Average monthly precipitation in cm

PrecipDiff Summer-to-winter difference in monthly precipitation in cm

RelHumidAve Average monthly relative humidity in percent

RelHumidDiff Summer-to-winter difference in monthly relative humidity in percent

PotGlobRadAve Average monthly potential global radiation in kJ

PotGlobRadDiff Summer-to-winter difference in monthly potential global radiation in kJ

AveTempAve Average monthly average temperature in degrees Celsius

AveTempDiff Summer-to-winter difference in monthly average temperature in degrees Celsius

MaxTempAve Average monthly maximum temperature in degrees Celsius

MaxTempDiff Summer-to-winter difference in monthly maximum temperature in degrees Celsius

MinTempAve Average monthly minimum temperature in degrees Celsius

MinTempDiff Summer-to-winter difference in monthly minimum temperature in degrees Celsius

DayTempAve Mean average daytime temperature in degrees Celsius

DayTempDiff Summer-to-winter difference in average daytime temperature in degrees Celsius

AmbVapPressAve Average monthly average ambient vapor pressure in Pa

AmbVapPressDiff Summer-to-winter difference in monthly average ambient vapor pressure in Pa

SatVapPressAve Average monthly average saturated vapor pressure in Pa

SatVapPressDiff Summer-to-winter difference in monthly average saturated vapor pressure in Pa

Aspect Aspect in degrees

TransAspect Transformed Aspect: TransAspect=(1-cos(Aspect))/2

Elevation Elevation in meters

Slope Percent slope

ReserveStatus Reserve Status (Reserve, Matrix)

StandAgeClass Stand Age Class (< 80 years, 80+ years)

ACONIF Average age of the dominant conifer in years

PctVegCov Percent vegetation cover

PctConifCov Percent conifer cover

PctBroadLeafCov Percent broadleaf cover

TreeBiomass Live tree (> 1inch DBH) biomass, above ground, dry weight

Source

Cutler, D. Richard., Thomas C. Edwards Jr., Karen H. Beard, Adele Cutler, Kyle T. Hess, Jacob Gibson, and Joshua J. Lawler. 2007. Random Forests for Classification in Ecology. Ecology 88(11): 2783-2792.

https://CRAN.R-project.org/package=EZtune/

`mtry_compare` *Mtry Tune via VIPs*

Description

A list of data.frames and useful plots for user evaluations of the randomForest hyperparameter mtry.

Usage

```
mtry_compare(
  formula,
  data = NULL,
  scale = FALSE,
  sqrt = TRUE,
  num_var,
  mvec,
  ...
)
```

Arguments

`formula` an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.

`data` an optional data frame containing the variables in the model. By default the variables are taken from the environment which randomForest is called from.

`scale` For permutation based measures such as MSE or Accuracy, should the measures be divided by their "standard errors"? Default is False.

`sqrt` Boolean value indicating whether importance metrics should be adjusted via a square root transformation. Default is True.

`num_var` Optional integer argument for reducing the number of plotted variables to the top 'num_var'. Should be an integer between 1 and the total number of predictor variables in the model or it should be a positive proportion of variables desired. If not provided, all variables are used.

`mvec` Optional vector argument for defining choices of mtry to have the function consider. Should be a vector of integers between 1 and the total number of predictor variables in the model. Or it can be a vector of proportions (between 0 and 1) of the number of predictor variables. If not provided, mvec is set to a vector of the lowest possible value, the default value, the highest possible value, and a middle value.

`...` Other parameters to pass to the randomForest function.

Value

A list of data.frames, useful plots, and forest objects for user evaluations of the randomForest hyperparameter mtry.

Examples

```
m <- mtry_compare(factor(Species) ~ ., data = iris, sqrt = TRUE)
m
```

`mtry_pdp_compare` *Mtry Tune via PDPs*

Description

This function builds randomForest algorithms, generates PDPs and combines them across different models. Outputs a list of data.frames and useful plots for user evaluations of the randomForest hyperparameter mtry. This also contains PDP-derived importance values for assessing effect of predictors on response.

Usage

```
mtry_pdp_compare(
  formula,
  data = NULL,
  mvec,
  var_vec,
  trim = 0.1,
  trellis = TRUE,
  which_class = 2L,
  prob = TRUE,
  ...
)
```

Arguments

`formula` an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.

`data` an optional data frame containing the variables in the model. By default the variables are taken from the environment which randomForest is called from.

`mvec` Optional vector argument for defining choices of mtry to have the function consider. Should be a vector of integers between 1 and the total number of predictor variables in the model. Or it can be a vector of proportions (strictly less than 1) of the number of predictor variables.

`var_vec` Optional vector argument for reducing the number of variables to consider and compare. Elements should be characters that match column names from the data used to generate the model x.

`trim` the fraction (0 to 0.5) of observations to be trimmed from each end of an individual PDP dataset before the trim-range is computed. The default of 0.1 will be used when values of trim outside that range are given.

`trellis` Logical indicating whether or not to generate trellis plots as output for comparing PDPs. Default is TRUE.

`which_class`	Integer specifying which column of the matrix of predicted probabilities to use as the "focus" class. Default is to use the first class. Only used for classification problems.
`prob`	Logical indicating whether or not partial dependence for classification problems should be returned on the probability scale, rather than the centered logit. If FALSE, the partial dependence function is on a scale similar to the logit. Default is TRUE.
`...`	Other parameters to pass to the randomForest function.

Value

A list of data.frames, useful plots, and forest objects for user evaluations of the randomForest hyperparameter mtry. This includes a list of partial dependence plots with adjusted y-axes so all PDPs are on an identical scale. This also contains comparative facet plots and PDP importance values for assessing true effect of predictors on response.

Examples

```
m <- mtry_pdp_compare(Petal.Length ~ ., data = iris)
m
```

`partial_cor` *Partial Correlations*

Description

A list of data.frames and useful plots for user evaluations of correlations and partial correlations of predictors with a given response.

Usage

```
partial_cor(formula, data = NULL, model = lm, num_var, ...)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	a data frame containing the variables in the model. By default the variables are taken from the environment which the model is called from.
`model`	Model to use for extraction partial correlations. Possible model choices are lm, rpart, randomForest, and svm. Default is lm.
`num_var`	Optional integer argument for reducing the number of variables to the top 'num_var'. Should be an integer between 1 and the total number of predictor variables in the model or it should be a positive proportion of variables desired.
`...`	Additional arguments to be passed to model as needed.

Value

A list of data.frames and useful plots for user evaluations of partial correlations.

Examples

```
pcs <- partial_cor(Petal.Length ~ ., data = iris[-5], model = lm)
pcs$plot_y_part_cors
```

---

pdp_compare *Small Multiple PDPs and Importance Metrics*

Description

This function takes a randomForest object, generates partial dependence plots for predictors and converts them to small multiples for appropriate comparison. Output is a list containing a comparative grid of PDPs, individual partial dependence plots, and PDP-derived importance values for assessing effect of predictors on response.

Usage

```
pdp_compare(
  x,
  var_vec,
  scale = FALSE,
  sqrt = TRUE,
  trim = 0.1,
  trellis = TRUE,
  which_class = 2L,
  prob = TRUE,
  ...
)
```

Arguments

x	An object of class randomForest.
var_vec	Optional vector argument for reducing the number of variables to consider and compare. Elements should be characters that match column names from the data used to generate the model x.
scale	For permutation based measures such as MSE or Accuracy, should the measures be divided by their "standard errors"? Default is FALSE.
sqrt	Boolean value indicating whether importance metrics should be adjusted via a square root transformation. Default is True.
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of an individual PDP dataset before the trim-range is computed. The default of 0.1 will be used when values of trim outside that range are given.

`trellis` Logical indicating whether or not to generate trellis plots as output for comparing PDPs. Default is TRUE.

`which_class` Integer specifying which column of the matrix of predicted probabilities to use as the "focus" class. Default is to use the first class. Only used for classification problems.

`prob` Logical indicating whether or not partial dependence for classification problems should be returned on the probability scale, rather than the centered logit. If FALSE, the partial dependence function is on a scale similar to the logit. Default is TRUE.

`...` Other parameters to pass to the partial function.

Value

A list of partial dependence plots with adjusted y-axes so all are on an identical scale. This list includes a comparative facet plot and pdp importance values for assessing true affect of predictors on response.

Examples

```
mtcars.rf <- randomForest::randomForest(formula = mpg ~ ., data = mtcars)
car_pd <- pdp_compare(x = mtcars.rf)
car_pd$full
car_pd$imp
gridExtra::grid.arrange(car_pd$wt, car_pd$disp,
  car_pd$hp, car_pd$cyl, nrow = 2)
```

robust_vifs *Non-linear Variance Inflation Factors*

Description

A list of data.frames and useful plots for user evaluations of the randomForest hyperparameter mtry.

Usage

```
robust_vifs(
  formula,
  data = NULL,
  model = randomForest,
  log10 = TRUE,
  num_var,
  ...
)
```

Arguments

`formula` an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.

`data` an optional data frame containing the variables in the model. By default the variables are taken from the environment which the model is called from.

`model` Model to use for extraction partial correlations. Possible model choices are rpart.

`log10` Applies a log10 transformation to VIFs when TRUE. Default is TRUE.

`num_var` Optional integer argument for reducing the number of variables to the top 'num_var'. Should be an integer between 1 and the total number of predictor variables in the model or it should be a positive proportion of variables desired.

`...` Additional arguments to be passed to models as needed.

Value

A list of data.frames and useful plots for user evaluations of VIFs.

Examples

```
rv <- robust_vifs(Petal.Length ~ ., data = iris[-5], model = lm)
rv
```

Index