Package ‘regressoR’

June 29, 2023

Title Regression Data Analysis System

Type Package

Version 3.0.2

Depends R (>= 4.1)

Imports DT (>= 0.27), pls (>= 2.8-1), dplyr (>= 1.1.0), shiny (>= 1.7.4), golem (>= 0.3.5), rlang (>= 1.0.6), glmnet (>= 4.1-6), loadeR (>= 1.1.3), shinyjs (>= 2.1.0), traineR (>= 2.0.4), shinyAce (>= 0.4.2), echarts4r (>= 0.4.4), htmltools (>= 0.5.4), rpart.plot (>= 3.1.1), shinydashboard (>= 0.7.2), shinycustomloader (>= 0.9.0), shinydashboardPlus (>= 2.0.3)

Description Perform a supervised data analysis on a database through a 'shiny' graphical interface. It includes methods such as linear regression, penalized regression, k-nearest neighbors, decision trees, ada boosting, extreme gradient boosting, random forest, neural networks, deep learning and support vector machines.

License GPL (>= 2)

Encoding UTF-8

URL https://promidat.website/,https://github.com/PROMiDAT/regressoR

BugReports https://github.com/PROMiDAT/predictoR/issues

RoxygenNote 7.1.2

NeedsCompilation no

Author Oldemar Rodriguez [aut, cre],
Diego Jimenez A. [ctb, prg]

Maintainer Oldemar Rodriguez <oldemar.rodriguez@ucr.ac.cr>

Repository CRAN

Date/Publication 2023-06-29 16:40:02 UTC

R topics documented:

`app_server` *The application server-side*

Description

The application server-side

Usage

```
app_server(input, output, session)
```

Arguments

`input, output, session`
Internal parameters for shiny. DO NOT REMOVE.

as_string_c *as_string_c*

Description

creates a string representative of a vector

Usage

```
as_string_c(vect, quote = TRUE)
```

Arguments

vect	a vector with values
quote	a logical value. If TRUE, the values on the vector will be surrounded by quotes.

Examples

```
as_string_c(c("A", "B", "C"))
as_string_c(c(5, 6, 7))
as_string_c(c(5, 6, 7), quote = FALSE)
as_string_c(iris$Species)
```

boosting_importance_plot *boosting_importance_plot*

Description

generates the graph of variable importance.

Usage

```
boosting_importance_plot(
  model,
  titles = c("Importancia de Variables segun Influencia Relativa",
    "Influencia Relativa", "Variable")
)
```

Arguments

model	boosting model(gbm).
titles	Labels on the chart

calibrate_boosting *calibrate_boosting*

Description

helps to get the maximum of n.minobsinnode and bag.fraction values with which no error is generated in the model.

Usage

```
calibrate_boosting(data)
```

Arguments

data	the name of the learning data.

See Also

gbm

Examples

```
calibrate_boosting(iris)
```

coef_lambda *coef_lambda*

Description

get penalized regression coefficients.

Usage

```
coef_lambda(data, variable.pred, model, log.lambda = NULL)
```

Arguments

data	dataframe
variable.pred	the name of the variable to be predicted.
model	a penalized regression model(cv.glmnet).
log.lambda	numerical. Logarithm of lambda in case you don't want to use the optimal lambda.

datos.disyuntivos *Create disjunctive columns to a data.frame.*

Description

Create disjunctive columns to a data.frame.

Usage

```
datos.disyuntivos(data, var)
```

Arguments

`data`	a data.frame object.
`var`	the column name to apply disjunctive code.

Value

data.frame

Author(s)

Diego Jimenez <diego.jimenez@promidat.com>

Examples

```
datos.disyuntivos(iris, "Species")
```

disp_models *disp_models*

Description

this function generates the call code of the scatter function.

Usage

```
disp_models(prediction, model_name, var_pred)
```

Arguments

`prediction`	the name of the prediction object.
`model_name`	the name of the model.
`var_pred`	the name of the variable to be predicted.

Examples

```
disp_models("prediction.knn", "KNN", "Species")
```

dt_plot *dt_plot*

Description

makes the graph of the tree.

Usage

```
dt_plot(model)
```

Arguments

`model`	a decision trees model(rpart).

exe *exe*

Description

concat and execute a text in R.

Usage

```
exe(..., envir = parent.frame())
```

Arguments

`...`	one or more texts to be concatenated and executed.
`envir`	the environment in which expr is to be evaluated.

Value

the result of the execute.

Examples

```
exe("5+5")
exe("5","+","5")
exe("plot(iris$Species)")
```

extract_code *extract_code*

Description

gets the code of a function in text form.

Usage

```
extract_code(funcion, envir = parent.frame())
```

Arguments

`funcion`	the name of the function to be extracted.
`envir`	the environment in which expr is to be evaluated.

Examples

```
extract_code("cat")
extract_code("plot")

parse(text = extract_code("plot"))
```

e_coeff_landa *e_coeff_landa*

Description

Graph the coefficients and lambdas of a cv.glmnet model

Usage

```
e_coeff_landa(
  cv.glm,
  log.lambda = NULL,
  titles = c("Coeficientes", "Seleccionado", "Automatico")
)
```

Arguments

`cv.glm`	a cv.glmnet model.
`log.lambda`	number that specifies the logarithm of the selected lambda
`titles`	labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

See Also

`cv.glmnet`

`e_JS` *Eval character vectors to JS code*

Description

Eval character vectors to JS code

Usage

```
e_JS(...)
```

Arguments

`...`	character vectors to evaluate

Author(s)

Joseline Quiros <joseline.quiros@promidat.com>

Examples

```
e_JS('5 * 3')
```

`e_posib_lambda` *e_posib_lambda*

Description

Graph a cv.glmnet model

Usage

```
e_posib_lambda(
  cv.glm,
  log.lambda = NULL,
  titles = c("Error Cuadratico Medio", "Curva Inferior", "Curva Superior",
    "Seleccionado", "Automatico", "Coeficientes Distintos de Cero")
)
```

Arguments

`cv.glm`	a cv.glmnet model.
`log.lambda`	number that specifies the logarithm of the selected lambda
`titles`	labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

See Also

`cv.glmnet`

`general_indices` *general_indices*

Description

calculates indices to measure accuracy of a model.

Usage

```
general_indices(real, prediccion)
```

Arguments

`real`	the real values in traning-testing.
`prediccion`	the prediction values in traning-testing.

Value

a list with the Correlation, Relative Error, Mean Absolute Error and Root Mean Square Error.

Examples

```
real <- rnorm(45)
prediction <- rnorm(45)
model <- "KNN"
general_indices(real, prediction)
```

`importance_plot_rf` *importance_plot_rf*

Description

graphs the importance of variables for the random forest model according to the percentage increase in mean square error.

Usage

```
importance_plot_rf(
  model.rf,
  titles = c("Importancia de Variables Segun el Porcentaje de Incremento del MSE",
    "Aumento porcentual del error cuadratico medio", "Variable")
)
```

Arguments

`model.rf`	a random forest model.
`titles`	labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

See Also

`randomForest`

nn_plot *nn_plot*

Description

graph of the neural network.

Usage

```
nn_plot(model)
```

Arguments

model	a neural network model(neuralnet)

pairs_power *pairs_power*

Description

Generate a pair chart

Usage

```
pairs_power(data, decimals = 2)
```

Arguments

data	A DataFrame
decimals	Number of numbers after the decimal point.

plot_pred_rd *plot_pred_rd*

Description

graph of variance explained in the predictors according to components used.

Usage

```
plot_pred_rd(
  model,
  n.comp,
  titles = c("Varianza Explicada en Predictores", "Numero de Componentes",
    "Porcentaje de Varianza Explicada")
)
```

Arguments

`model`	a dimension reduction model.
`n.comp`	the optimum number of components.
`titles`	labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

`plot_real_prediction` *plot_real_prediction*

Description

scatter plot between the actual value of the variable to be predicted and the prediction of the model.

Usage

```
plot_real_prediction(
  real,
  prediction,
  model = "",
  titles = c("Predicciones vs Valores Reales", "Valor Real", "Prediccion")
)
```

Arguments

`real`	the real values in traning-testing.
`prediction`	the prediction values in traning-testing.
`model`	the name of the model of the scatter plot.
`titles`	Labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

plot_RMSE *plot_RMSE*

Description

graph the root mean square error of cross validation according to components used.

Usage

```
plot_RMSE(
  model,
  n.comp,
  titles = c("RMSE Segun Numero de Componentes", "Numero de Componente", "RMSE")
)
```

Arguments

model	a dimension reduction model.
n.comp	the optimum number of components.
titles	labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

plot_var_pred_rd *plot_var_pred_rd*

Description

graph of the variance explained in the variable to predict according to the components used.

Usage

```
plot_var_pred_rd(
  model,
  n.comp,
  titles = c("Varianza Explicada en Variable a Predecir", "Numero de Componente",
    "Porcentaje de Varianza Explicada")
)
```

Arguments

`model` a dimension reduction model.

`n.comp` the optimum number of components.

`titles` labels on the chart

Value

echarts4r plot

Author(s)

Ariel Arroyo <luis.ariel.arroyo@promidat.com>

`rd_model` *rd_model*

Description

generates a dimension reduction model.

Usage

```
rd_model(data, variable.pred, mode = 0, scale = TRUE)
```

Arguments

`data` dataframe

`variable.pred` the name of the variable to be predicted.

`mode` the method of dimension reduction is defined as mode=1 is the MCP, and mode=0 the ACP.

`scale` the scale parameter of the model.

See Also

`pcr`, `plsr`

`rd_prediction` *rd_prediction*

Description

generates the prediction of a dimension reduction model.

Usage

```
rd_prediction(model, test.data, ncomp = NULL)
```

Arguments

`model` dimension reduction model(pcr/plsr).

`test.data` dataframe.

`ncomp` a numerical value in case you don't want to use the optimum number of components.

`rd_type` *rd_type*

Description

returns the name of the method of dimension reduction.

Usage

```
rd_type(mode.rd = 0)
```

Arguments

`mode.rd` the method of dimension reduction is defined as mode=1 is the MCP, and mode=0 the ACP.

See Also

`pcr`, `plsr`

Examples

```
rd_type(1)
rd_type(0)
```

rlr_model *rlr_model*

Description

generates a penalized regression model.

Usage

```
rlr_model(data, variable.pred, alpha = 0, standardize = TRUE)
```

Arguments

`data`	dataframe
`variable.pred`	the name of the variable to be predicted.
`alpha`	the alpha parameter of the model.
`standardize`	the standardize parameter of the model.

See Also

`glmnet`, `cv.glmnet`

rlr_prediction *rlr_prediction*

Description

generates the prediction of the penalized regression model.

Usage

```
rlr_prediction(model, test.data, variable.pred, log.lambda = NULL)
```

Arguments

`model`	a penalized regression model(cv.glmnet).
`test.data`	dataframe.
`variable.pred`	the name of the variable to be predicted.
`log.lambda`	numerical. Logarithm of lambda in case you don’t want to use the optimal lambda.

rlr_type *rlr_type*

Description

returns the name of the penalty according to the alpha.

Usage

```
rlr_type(alpha_rlr = 0)
```

Arguments

`alpha_rlr`	the penalty is defined as alpha=1 is the lasso penalty, and alpha=0 the ridge penalty.

See Also

`glmnet`

Examples

```
rlr_type(1)
rlr_type(0)
```

rl_coeff *rl_coeff*

Description

get the information of the coefficients of the linear regression model

Usage

```
rl_coeff(modelo)
```

Arguments

`modelo`	linear regression model

run_app *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_app(...)
```

Arguments

`...`	A series of options to be used inside the app.

summary_indices *summary_indices*

Description

summarizes a variable by returning the minimum, first quartile, third quartile and maximum value.

Usage

```
summary_indices(data)
```

Arguments

`data`	a numeric vector.

Examples

```
summary_indices(iris$Sepal.Length)
```

Index