Package ‘fscaret’

March 4, 2015

Type Package
Title Automated Feature Selection from 'caret'
Version 0.9.3
Date 2015-03-3
Depends R (>= 3.1.0), caret, gsubfn, hmeasure, utils, parallel
Suggests ada, arm, Boruta, bst, C50, car, caTools, class, Cubist,
e1071, earth (>= 2.2-3), elasticnet, ellipse, evtree,
extraTrees, fastICA, foba, gam, gbm (>= 2.1), glmnet (>= 1.8),
hda, HDclassif, Hmisc, ipred, kernlab, kknn, klaR, kohonen,
KRLS, lars, leaps, LogicReg, MASS, mboost, mda, mgcv, mlbench,
nnet, neuralnet, nnet, nodeHarvest, obliqueRF, pamr, partDSA, party
(>= 0.9-99992), penalized, penalizedLDA, pls, pROC, proxy,
qrnn, quantregForest, randomForest, RANN, relaxo, rFerns, rocc,
rpart, rrcov, RRF, rrlida, RSNNS, RWeka (>= 0.4-1), sda,
sparseLDA (>= 0.1-1), spls, stepPlr, superpc
Maintainer Jakub Szlek <j.szlek@uj.edu.pl>
License GPL-2 | GPL-3
Description Automated feature selection using variety of models
provided by 'caret' package.
This work was funded by Poland-Singapore bilateral cooperation
project no 2/3/POL-SIN/2012.
NeedsCompilation no
Repository CRAN
Date/Publication 2015-03-04 16:03:57
Author Jakub Szlek [aut, cre],
Aleksander Mendyk [ctb]

R topics documented:

  fscaret-package .................................................. 2
classVarImp ............................................................ 3
dataPreprocess ...................................................... 4
Description

This package provides fast and automated feature selection based on caret package modeling methods. The main advantage of this extension is that it requires minimum user involvement. Also the variety of used methods in combination with the scaling according to RMSE or MSE obtained from models profit the user. The idea is based on the assumption that the variety of models will balance the roughness of calculations (default model settings are applied). On Windows OS the time limiting function is off, multicore functionality is enabled via parLapply() function of package ‘parallel’. Acknowledgments:

This work was funded by Poland-Singapore bilateral cooperation project no 2/3/POL-SIN/2012

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>fscaret</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.9.3</td>
</tr>
<tr>
<td>Date</td>
<td>2015-03-3</td>
</tr>
<tr>
<td>License</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

Author(s)

Jakub Szlek <j.szlek@uj.edu.pl> Contributions from Aleksander Mendyk, also stackoverflow and r-help@r-project.org mailing list community.
Maintainer: Jakub Szlek <j.szlek@uj.edu.pl>.
classVarImp

References


See Also

*train, trainControl, rfeControl* by Max Kuhn <Max.Kuhn at pfizer.com> and *predict* base utilities

---

classVarImp | classVarImp

Description

The function uses the caret package advantage to perform fitting of numerous classification models.

Usage

classVarImp(model, xTrain, yTrain, xTest,
             fitControl, myTimeLimit, no.cores,
             lk_col, supress.output, mySystem)

Arguments

- **model**: Chosed models as called from function fscaret(), argument Used.funcClassPred.
- **xTrain**: Training data set, data frame of input vector
- **yTrain**: Training data set, vector of observed outputs, must be in binary form 0/1.
- **xTest**: Testing data set, data frame of input vector
- **fitControl**: Fitting controls passed to caret function
- **myTimeLimit**: Time limit in seconds for single model fitting
- **no.cores**: Number of used cores for calculations
- **lk_col**: Number of columns for whole data set (inputs + output)
- **supress.output**: If TRUE output of models are supressed.
- **mySystem**: Called from fscaret() result of function .Platform$OS.type

Author(s)

Jakub Szlek and Aleksander Mendyk
dataPreprocess

dataPreprocess

dataPreprocess

dataPreprocess

Description

The functionality is realized in two main steps:

1. Check for near zero variance predictors and flag as near zero if:
   (a) the percentage of unique values is less than 20
   (b) the ratio of the most frequent to the second most frequent value is greater than 20,
2. Check for susceptibility to multicollinearity
   (a) Calculate correlation matrix
   (b) Find variables with correlation 0.9 or more and delete them

Usage

dataPreprocess(trainMatryca_nr, testMatryca_nr, labelsFrame, lk_col, lk_row, with.labels)

Arguments

trainMatryca_nr
  Input training data matrix

testMatryca_nr
  Input testing data matrix

labelsFrame
  Transposed data frame of column names

lk_col
  Number of columns

lk_row
  Number of rows

with.labels
  If with.labels=TRUE, additional data frame with preprocessed inputs corresponding to original data set column numbers as output is generated

Author(s)

Jakub Szlek and Aleksander Mendyk

References

Examples

library(fscaret)

# Create data sets and labels data frame
trainMatrix <- matrix(rnorm(150*120,mean=10, sd=1), 150, 120)

# Adding some near-zero variance attributes
temp1 <- matrix(runif(150,0.0001,0.0005), 150, 12)

# Adding some highly correlated attributes
sampleColIndex <- sample(ncol(trainMatrix), size=10)
temp2 <- matrix(trainMatrix[,sampleColIndex]*2, 150, 10)

# Output variable
output <- matrix(rnorm(150,mean=10, sd=1), 150, 1)

trainMatrix <- cbind(trainMatrix,temp1,temp2, output)

colnames(trainMatrix) <- paste("x",c(1:ncol(trainMatrix)),sep="")

# Subset test data set
testMatrix <- trainMatrix[sample(round(0.1*nrow(trainMatrix))),]

labelsDF <- data.frame("Labels"=paste("X",c(1:(ncol(trainMatrix)-1)),sep=""))

lk_col <- ncol(trainMatrix)
lk_row <- nrow(trainMatrix)

with.labels = TRUE

testRes <- dataPreprocess(trainMatrix, testMatrix, 
    labelsDF, lk_col, lk_row, with.labels)

summary(testRes)

# Selected attributes after data set preprocessing

testRes$labelsDF

# Training and testing data sets after preprocessing

testRes$trainMatrixca

testRes$testMatrixca
dataset.test

Example testing data set

Description

The data set after preprocessing, which resulted in 29 inputs. Original data set was obtained in literature survey with 298 inputs. Input: chemical descriptors and characteristics of 8 PLGA microparticles formulation. Output: mean particle size of PLGA microparticles Number of attributes 29, single output.

Usage

data(dataset.test)

Format

data.frame

Details

Literature survey yielded 68 formulations of PLGA microspheres with protein as active pharmaceutical ingredient. In vitro release profiles as well as formulation characteristics and composition were derived from articles. Chemical descriptors were obtained using Marvin ChemAxon software (cxcalc plugin). The final data base consisted of 298 inputs and single output mean particle size.

Source

Examples

library(fscaret)

data(dataset.test)

dataset.test

---

**dataset.train**  

*Example training data set*

### Description

The data set after preprocessing, which resulted in 29 inputs. Original data set was obtained in literature survey with 298 inputs. Input: chemical descriptors and characteristics of 8 PLGA microparticles formulation. Output: mean particle size of PLGA microparticles. Number of attributes 29, single output.

### Usage

`data(dataset.train)`

### Format

`data.frame`

### Details

Literature survey yielded 68 formulations of PLGA microspheres with protein as active pharmacutical ingredient. In vitro release profiles as well as formulation characteristics and composition were derived from articles. Chemical descriptors were obtained using Marvin ChemAxon software (cxcalc plugin). The final data base consisted of 298 inputs and single output mean particle size.

### Source


Examples

library(fscaret)

data(dataset.train)

dataset.train

fscaret  feature selection caret

Description

Main function for fast feature selection. It utilizes other functions as regPredImp or impCalc to obtain results in a list of data frames.

Usage

fscaret(trainDF, testDF, installReqPckg = FALSE, preprocessData = FALSE, with.labels = TRUE, classPred = FALSE, regPred = TRUE, skel_outfile = NULL, impCalcMet = "RMSE&MSE", myTimeLimit = 24 * 60 * 60, Used.funcRegPred = NULL, Used.funcClassPred = NULL, no.cores = NULL, method = "boot", returnResamp = "all", missData=NULL, supress.output=FALSE, ...)

Arguments

trainDF  Data frame of training data set, MISO (multiple input single output) type

testDF  Data frame of testing data set, MISO (multiple input single output) type

installReqPckg  If TRUE prior to calculations it installs all required packages, please be advised to be logged as root (admin) user

preprocessData  If TRUE data preprocessing is performed prior to modeling

with.labels  If TRUE header of the input files are read

classPred  If TRUE classification models are applied. Please be advised that importance is scaled according to F-measure regardless impCalcMet settings.

regPred  If TRUE regression models are applied
skel_outfile  Skeleton output file, e.g. skel_outfile=c("_myoutput_")
impCalcMet  Variable importance calculation scaling according to RMSE and MSE, for both please enter impCalcMet="RMSE&MSE"
myTimeLimit  Time limit in seconds for single model development
Used.funcRegPred  Vector of regression models to be used, for all available models please enter Used.funcRegPred="all"
Used.funcClassPred  Vector of classification models to be used, for all available models please enter Used.funcClassPred="all"
no.cores  Number of cores to be used for modeling, if NULL all available cores are used, should be numeric type or NULL
method  Method passed to fitControl of caret package
returnResamp  Returned resampling method passed to fitControl of caret package
missData  Handling of missing data values. Possible values: "delRow" - delete observations with missing values, "delCol" - delete attributes with missing values, "meanCol" - replace missing values with column mean.
supress.output  If TRUE output of modeling phase by caret functions are supressed. Only info which model is currently calculated and resulting variable importance.
...  Additional arguments, preferably passed to fitControl of caret package

Value

$modelPred  List of outputs from caret model fitting
$VarImp  Data frames of variable importance
$PPLabels  Data frame of resulting preprocessed data set with original input numbers and names
$PPTrainDF  Training data set after preprocessing
$PPTestDF  Testing data set after preprocessing

Note
Be advised when using fscaret function as it requires hard disk operations for saving fitted models and data frames. Files are written in R temp session folder, for more details see tempdir(), getwd() and setwd()

Author(s)
Jakub Szlek and Aleksander Mendyk

References
funcClass.all

Examples

library(fscaret)

# Load data sets
data(dataset.train)
data(dataset.test)

requiredPackages <- c("R.utils", "gsubfn", "ipred", "caret", "parallel", "MASS")

mySystem <- .Platform$OS.type

if(mySystem=="windows"){
  myCores <- 2
} else {
  myCores <- 2
}

myFirstRes <- fscaret(dataset.train, dataset.test, installReqPckg=FALSE,
  preprocessData=FALSE, with.labels=TRUE, classPred=FALSE,
  regPred=TRUE, skel_outfile=NULL,
  impCalcMet="RMSE&MSE", myTimeLimit=5,
  Used.funcRegPred=c("lm","pls","pcr"), Used.funcClassPred=NULL,
  no.cores=myCores, method="boot", returnResamp="all",
  supress.output=TRUE)

# Training data set after preprocessing
myFirstRes$PPTrainDF

# Testing data set after preprocessing
myFirstRes$PPTestDF

# Model predictions
myFirstRes$ModelPred

# Variable importance after scaling according to RMSE and MSE
myFirstRes$VarImp

# Reduced input vector (data set) after preprocessing
myFirstRes$PPLabels
**funcReg.all**  
*All regression methods used*

**Description**

Vector of all regression methods used in solving problems by caret

**Usage**

```r
data(funcRegPred)
```

**Format**

`vector`

**Examples**

```r
# Load library
library(fscaret)

# Load data set
data(funcClassPred)

# Print out object
funcClassPred
```
Examples

```r
# Load library
library(fscaret)

# Load data set
data(funcRegPred)

# Print out object
funcRegPred
```

Description

The `impCalc` function is designed to scale variable importance according to MSE and RMSE calculations. It also stores the raw MSE and RMSE derived from models. The `impCalc` function shouldn’t be used alone unless the user has trained models from the caret package in RData files.

Usage

```r
impCalc(skel_outfile, xTest, yTest, lk_col, labelsFrame, with.labels, regPred, classPred)
```

Arguments

- `skel_outfile`: Skeleton name of output file
- `xTest`: Input vector of testing data set
- `yTest`: Output vector of testing data set
- `lk_col`: Number of columns of whole data set
- `labelsFrame`: Labels to sort variable importance
- `with.labels`: Pass with.labels argument. It is advised to ALWAYS use labels as in some cases `VarImp` returns importance in descending values. If you insist turning with.labels FALSE, then make sure data base contains pure data and you read it (read.csv) to data.frame with option header=FALSE.
- `regPred`: Indicating if regression predictions are computed. Possible values TRUE/FALSE. If regPred is set TRUE, then classPred should be set FALSE.
- `classPred`: Indicating if classification predictions are computed. Possible values TRUE/FALSE. If classPred is set TRUE, then regPred should be set FALSE. Please be advised that importance is scaled according to F-measure.

Details

The `impCalc` function lists RData files in the working directory assuming there are only models derived by caret. In a loop function loads models and tries to get the variable importance.
Author(s)

Jakub Szlek and Aleksander Mendyk

Examples

# Hashed to comply with new CRAN check
library(fscaret)

# Read working directory
myWD <- getwd()

# Set working directory to tmp
setwd(tempdir())

# Load dataset
data(dataset.train)
data(dataset.test)

# Make objects
trainDF <- dataset.train
testDF <- dataset.test
model <- c("lm","pls","pcr")
fitControl <- trainControl(method = "boot", returnResamp = "all")
myTimeLimit <- 5
no.cores <- 2
supress.output <- TRUE
skel_outfile <- paste("_default_",sep="")
mySystem <- .Platform$OS.type
with.labels <- TRUE
redPred <- TRUE
classPred <- FALSE

if(mySystem=="windows"){
  no.cores <- 1
}

# Scan dimensions of trainDF [lk_row x lk_col]
lk_col = ncol(trainDF)
lk_row = nrow(trainDF)

# Read labels of trainDF
labelsFrame <- as.data.frame(colnames(trainDF))
labelsFrame <- cbind(c(1:ncol(trainDF)),labelsFrame)

# Create a train data set matrix
trainMatryca_nr <- matrix(data=NA,nrow=lk_row,ncol=lk_col)

row=0
col=0

for(col in 1:(lk_col)) {
Secondary function imputes the mean to columns with NA data.

Usage

    impute.mean(x)
installPckg

Arguments

x a vector to calculate mean

Author(s)

Jakub Szlek and Aleksander Mendyk

Examples

library(fscaret)

# Make sample matrix
testData <- matrix(data=rep(1:5),nrow=10,nrow=15)

# Replace random values with NA's
n <- 15
replace <- TRUE
set.seed(1)

rand.sample <- sample(length(testData), n, replace=replace)
testData[rand.sample] <- NA

# Print out input matrix
testData

# Record cols with missing values
missing.colsTestMatrix <- which(colSums(is.na(testData))>0)

for(i in 1:length(missing.colsTestMatrix)){

rowToReplace <- missing.colsTestMatrix[i]
testData[,rowToReplace] <- impute.mean(testData[,rowToReplace])
}

# Print out matrix with replaced NA's by column mean
testData

installPckg

Description

Function installs the packages that are listed in data(requiredPackages). The function is called within fscaret function. If argument "installReqPckg = TRUE" the function installs required packages.
Usage

installPckg(requiredPackages)

Arguments

requiredPackages
  Vector of packages to be installed

Details

Be advised setting "installReqPckg = TRUE" installs packages in your home directory (.R). To install packages for all users please login as root (admin).

Author(s)

Jakub Szlek and Aleksander Mendyk

Description

Function calculates mean squared error as predicted vs. observed

Usage

MSE(vect1, vect2, rows_no)

Arguments

vect1       Numeric vector of predicted values
vect2       Numeric vector of observed values
rows_no     Number of observations

Author(s)

Jakub Szlek and Aleksander Mendyk

Examples

## The function is currently defined as
function (vect1, vect2, rows_no)
{
  result = 0
  pred <- 0
  obs <- 0
  for (i in 1:rows_no) {

```
regVarImp

```r
result <- result + (vect1[i] - vect2[i])^2
}
result <- (result/rows_no)
return(result)
```

**Description**

The function uses the caret package advantage to perform fitting of numerous regression models.

**Usage**

```r
regVarImp(model, xTrain, yTrain, xTest, 
fitControl, myTimeLimit, no.cores, 
lk_col, supress.output, mySystem)
```

**Arguments**

- `model`  Chosed models as called from function fscaret(), argument Used.funcRegPred.
- `xTrain` Training data set, data frame of input vector
- `yTrain` Training data set, vector of observed outputs
- `xTest`  Testing data set, data frame of input vector
- `fitControl` Fitting controls passed to caret function
- `myTimeLimit` Time limit in seconds for single model fitting
- `no.cores` Number of used cores for calculations
- `lk_col` Number of columns for whole data set (inputs + output)
- `supress.output` If TRUE output of models are supressed.
- `mySystem` Called from fscaret() result of function .Platform$OS.type

**Author(s)**

Jakub Szlek and Aleksander Mendyk

**References**

### requiredPackages

Character vector of names of required packages to fully take advantage of fscaret

### Usage

```r
data(requiredPackages)
```

### Format

```
vector
```

### Examples

```r
data(requiredPackages)
```

---

### RMSE

Function calculates root mean squared error.

### Usage

```r
RMSE(vect1, vect2, rows_no)
```

### Arguments

- **vect1**: Numeric vector of predicted values
- **vect2**: Numeric vector of observed values
- **rows_no**: Number of observations

### Author(s)

Aleksander Mendyk
Examples

```r
## The function is currently defined as
function (vect1, vect2, rows_no)
{
    result = 0
    obs <- 0
    pred <- 0
    for (i in 1:rows_no) {
        result <- result + (vect1[i] - vect2[i])^2
    }
    result <- (result/rows_no)*0.5
    return(result)
}
```

Description

This function limits elapsed time spent on single model development. It uses low-level functions of parallel package and sets the fork process with time limit. If the result is not returned within set time, it kills fork. Function shouldn’t be called from R console. The function is not used under Windows OS. Only Unix-like systems have fork functionality.

Usage

```
timeout(..., seconds)
```

Arguments

- ... Expression to be time limited
- seconds Number of seconds

Author(s)

Original code by Jeroen Ooms <jeroen.ooms at stat.ucla.edu> of OpenCPU package. Modifications by Jakub Szlek and Aleksander Mendyk.

Examples

```r
## The function is currently defined as

function(..., seconds) {

    fork_to_check <- parallel::mcparallel(
    {eval(...)},
    silent = FALSE)

    # call mccollect to wait "seconds" for returning result of mcparallel.
```
my_result <- parallel::mccollect(fork_to_check, wait = FALSE, timeout = seconds)
# If my_result is returned kill fork
tools::pskill(fork_to_check$pid, tools::SIGKILL)
tools::pskill(-1 * fork_to_check$pid, tools::SIGKILL)

# kill the fork of forks if they were spawned
parallel::mccollect(fork_to_check, wait = FALSE)
# If the function mccollect had NULL (timedout), make stop
if (is.null(my_result))
  stop("Time limit has reached!")

# remove list format
my_result <- my_result[[1]]

# return result
return(my_result)
}
Index

*Topic arith
  MSE, 18
  RMSE, 20
*Topic array
  fscaret, 10
*Topic datasets
  dataset.test, 6
  dataset.train, 8
  funcClass.all, 13
  funcReg.all, 13
  requiredPackages, 20
*Topic data
  installPckg, 17
*Topic design
  impCalc, 14
*Topic error
  timeout, 21
*Topic iteration
  fscaret, 10
*Topic logic
  imputeMean, 16
*Topic math
  imputeMean, 16
  MSE, 18
  RMSE, 20
*Topic methods
  fscaret, 10
  timeout, 21
*Topic models
  classVarImp, 3
  impCalc, 14
  regVarImp, 19
*Topic optimize
  fscaret, 10
*Topic package
  fscaret-package, 2
  installPckg, 17
*Topic robust
  classVarImp, 3
  dataPreprocess, 4
  regVarImp, 19
*Topic univar
  dataPreprocess, 4
  classVarImp, 3
  dataPreprocess, 4
  dataset.test, 6
  dataset.train, 8
  fscaret, 10
  fscaret-package, 2
  funcClass.all, 12
  funcClassPred(funcClass.all), 13
  funcReg.all, 13
  funcRegPred(funcReg.all), 13
  impCalc, 14
  impute.mean(imputeMean), 16
  imputeMean, 16
  installPckg, 17
  MSE, 18
  predict, 3
  regVarImp, 19
  requiredPackages, 20
  rfeControl, 3
  RMSE, 20
  timeout, 21
  train, 3
  trainControl, 3