Package ‘fscaret’

April 1, 2014

Type Package

Title Automated caret feature selection

Version 0.8.5.5

Date 2014-03-27

Depends R (>= 3.0.0), caret, gsubfn, utils, parallel

Suggests ada, arm, Boruta, bst, C50, car, caTools, class, Cubist,e1071, earth (>= 2.2-3), elasticnet, ellipse, evtree,extraTrees, fastLCA, foba, gam, gbm (>= 2.1), glmnet (>= 1.8), grid, hda, HDclassif, HiDimDA, Hmisc, ipred, kernlab, kknn, klaR, kohonen, KRLS, lars, leaps, LogitReg, MASS, mboost, mda, mgev, mlbench, neuralnet, nnet, nodeHarvest, obliqueRF, pamr, partDSA, party (>= 0.9-99992), penalized, penalizedLDA, pls, pROC, proxy, qrmn, quantregForest, randomForest, RANN, relaxo, rFerns, rocc, rpart, rrcov, RRF, rrlda, RSNN, RWeka (>= 0.4-1), sda, SDDA, sparseLDA (>= 0.1-1), spls, stepPlr, superpc

Author Jakub Szlek <j.szlek@uj.edu.pl>, acknowledgments to Aleksander Mendyk, contributions from stackoverflow and r-help@r-project.org mailing list community.

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

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

NeedsCompilation no

Repository CRAN

Date/Publication 2014-04-01 11:52:45
R topics documented:

fscaret-package ......................................................... 2
dataPreprocess ......................................................... 3
dataset.test ............................................................. 5
dataset.train ............................................................ 7
fscaret ................................................................. 9
funcReg.all ............................................................ 12
impCalc ................................................................. 12
imputeMean ............................................................. 14
installPckg .............................................................. 16
MSE ................................................................. 16
regVarImp ............................................................... 17
requiredPackages ......................................................... 19
RMSE ................................................................. 20
timeout ................................................................. 21

Index 22

fscaret-package  Automated feature selection caret (fscaret)

Description

This package provide 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 also the number of cores used is set to 1.

Details

Package: fscaret
Type: Package
Version: 0.8.5.5
Date: 2014-03-27
License: GPL-2 | GPL-3

Author(s)

Jakub Szlek <j.szlek@uj.edu.pl>, acknowledgments to Aleksander Mendyk, contributions from stackoverflow and r-help@r-project.org mailing list community.
Maintainer: Jakub Szlek <j.szlek@uj.edu.pl>.
Acknowledgments:
This work was funded by Poland-Singapore bilateral cooperation project no 2/3/POL-SIN/2012

References


See Also

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

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

```r
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)

# 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

# Selected attributes after data set preprocessing
```
**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 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.train)

dataset.train

---

table

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 = FALSE, 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 (for v0.8 it is not available)
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 v0.8 it is not available)
no.cores Number of cores to be used for modeling, if NULL all available cores are used
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
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 <- 1
} 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","pca"), 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
impCalc

Description

Vector of all regression methods used in solving problems by caret

Usage

data(funcRegPred)

Format

vector

Examples

# Load library
library(fscaret)

# Load data set
data(funcRegPred)

# Print out object
funcRegPred

impCalc

Description

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. impCalc function shouldn’t be used alone unless user has trained models from caret package in RData files.

Usage

impCalc(skel_outfile, xTest, yTest, lk_col)
impCalc

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

Details

impCalc function lists RData files in 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

```R
# 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
suppress.output <- TRUE
skel_outfile <- paste("_default_",sep="")
mySystem <- .Platform$OS.type

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))

# 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)) {
  for(row in 1:(lk_row)) {
    trainMatryca_nr[row,col] <- as.numeric(trainDF[row,col])
  }
}

# Pointing standard data set train
xTrain <- data.frame(trainMatryca_nr[,lk_col])
yTrain <- as.vector(trainMatryca_nr[,lk_col])

#--------Scan dimensions of trainDataFrame1 [lk_row x lk_col]
lk_col_test = ncol(testDF)
lk_row_test = nrow(testDF)

testMatryca_nr <- matrix(data=NA,nrow=lk_row_test,ncol=lk_col_test)
row=0
col=0
for(col in 1:(lk_col_test)) {
  for(row in 1:(lk_row_test)) {
    testMatryca_nr[row,col] <- as.numeric(testDF[row,col])
  }
}

# Pointing standard data set test
xTest <- data.frame(testMatryca_nr[,lk_col])
yTest <- as.vector(testMatryca_nr[,lk_col])

# Calling lower function to create models to calculate on
myVarImp <- regVarImp(model, xTrain, yTrain, xTest,
  fitControl, myTimeLimit, no.cores, lk_col,
  suppress.output, mySystem)

myImpCalc <- impCalc(skel_outfile, xTest, yTest, lk_col)
**imputeMean**

**Description**
Secondary function imputes the mean to columns with NA data.

**Usage**
```r
impute.mean(x)
```

**Arguments**
- `x`: a vector to calculate mean

**Author(s)**
Jakub Szlek and Aleksander Mendyk

**Examples**
```r
library(fscaret)

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

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

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

# Print out input matrix
class(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
class(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**

```r
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

---

**MSE**

**Description**

Function calculates mean squared error as predicted vs. observed

**Usage**

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

**Arguments**

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

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

Examples

```r
# Hashed to comply with new CRAN check
#
# Load library
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
suppress.output <- TRUE

mySystem <- .Platform$OS.type

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))

# 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)) {
    for(row in 1:(lk_row)) {
        trainMatryca_nr[row,col] <- as.numeric(trainDF[row,col])
    }
}

# Pointing standard data set train
xTrain <- data.frame(trainMatryca_nr[,-lk_col])
yTrain <- as.vector(trainMatryca_nr[,lk_col])

#--------Scan dimensions of trainDataFrame1 [lk_row x lk_col]
lk_col_test = ncol(testDF)
lk_row_test = nrow(testDF)

testMatryca_nr <- matrix(data=NA,nrow=lk_row_test,ncol=lk_col)

row=0
col=0

for(col in 1:(lk_col_test)) {
    for(row in 1:(lk_row_test)) {
        testMatryca_nr[row,col] <- as.numeric(testDF[row,col])
    }
}

# Pointing standard data set test
xTest <- data.frame(testMatryca_nr[,-lk_col])
yTest <- as.vector(testMatryca_nr[,lk_col])

myVarImp <- regVarImp(model, xTrain, yTrain, xTest,
    fitControl, myTimeLimit, no.cores, lk_col, 
    suppress.output, mySystem)

summary(myVarImp)
print(myVarImp)

# Get back to previous working directory
setwd(myWD)

---

requiredPackages

**Description**

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

data(requiredPackages)

Format

vector

Examples

data(requiredPackages)

<table>
<thead>
<tr>
<th>RMSE</th>
<th>RMSE</th>
</tr>
</thead>
</table>

Description

Function calculates root mean squared error.

Usage

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 the cpu time spent on single model fitting. It simply sets the killing process of itself to sleep for chosen number of seconds. Shouldn’t be called from R console. The function is not used under Windows OS.

Usage

timeout(expr, seconds, my.pid)

Arguments

expr | Expression to be time limited  
seconds | Number of seconds  
my.pid | Process PID

Author(s)

Jakub Szlek and Aleksander Mendyk with contribution to nabble users

Examples

## The function is currently defined as
function (expr, seconds, my.pid)
{
    killer.pid <- system(intern = TRUE, paste(" (sleep", seconds,
        "; kill -9", my.pid, ")>/dev/null\n echo $!", sep = " "))
    on.exit(system(paste("kill", killer.pid, "> /dev/null 2>&1", 
        sep = " ")))
    withCallingHandlers(expr, interrupt = function(...) {
        stop("Timedout", call. = FALSE)
    })
}
Index

*Topic arith
  MSE, 16
  RMSE, 20
*Topic array
  fscaret, 9
*Topic datasets
  dataset.test, 5
dataset.train, 7
  funcReg.all, 12
  requiredPackages, 19
*Topic data
  installPckg, 16
*Topic design
  impCalc, 12
*Topic error
  timeout, 21
*Topic iteration
  fscaret, 9
*Topic logic
  imputeMean, 15
*Topic math
  imputeMean, 15
  MSE, 16
  RMSE, 20
*Topic methods
  fscaret, 9
  timeout, 21
*Topic models
  impCalc, 12
  regVarImp, 17
*Topic optimize
  fscaret, 9
*Topic package
  fscaret-package, 2
  installPckg, 16
*Topic robust
  dataPreprocess, 3
  regVarImp, 17
*Topic univar
  dataPreprocess, 3
dataset.test, 5
dataset.train, 7
  fscaret, 9
  fscaret-package, 2
  funcReg.all, 12
  funcRegPred (funcReg.all), 12
  impCalc, 12
  impute.mean (imputeMean), 15
  imputeMean, 14
  installPckg, 16
  MSE, 16
  predict, 3
  regVarImp, 17
  requiredPackages, 19
  rfeControl, 3
  RMSE, 20
  timeout, 21
  train, 3
  trainControl, 3