Package 'PrInDT'

May 8, 2026

Type Package

Title Prediction and Interpretation in Decision Trees for Classification and Regression

Version 2.0.2

Description Optimization of conditional inference trees from the package 'party' for classification and regression.
For optimization, the model space is searched for the best tree on the full sample by means of repeated subsampling. Restrictions are allowed so that only trees are accepted which do not include pre-specified uninterpretable split results (cf. Weihs & Buschfeld, 2021a). The function PrInDT() represents the basic resampling loop for 2-class classification (cf. Weihs & Buschfeld, 2021a). The function RePrInDT() (repeated PrInDT()) allows for repeated applications of PrInDT() for different percentages of the observations of the large and the small classes (cf. Weihs & Buschfeld, 2021c). The function NesPrInDT() (nested PrInDT()) allows for an extra layer of subsampling for a specific factor variable (cf. Weihs & Buschfeld, 2021b). The functions PrInDTMulev() and PrInDTMulab() deal with multilevel and multilabel classification. In addition to these PrInDT() variants for classification, the function PrInDTreg() has been developed for regression problems. Finally, the function PostPrInDT() allows for a posterior analysis of the distribution of a specified variable in the terminal nodes of a given tree.
In version 2, additionally structured sampling is implemented in functions PrInDTCstruc() and PrInDTRstruc(). In these functions, repeated measurements data can be analyzed, too. Moreover, multilabel 2-stage versions of classification and regression trees are implemented in functions C2SPrInDT() and R2SPrInDT() as well as interdependent multilabel models in functions SimCPrInDT() and SimRPrInDT(). Finally, for mixtures of classification and regression models functions Mix2SPrInDT() and SimMixPrInDT() are implemented. Most of these extensions of PrInDT are described in Buschfeld & Weihs (2025Fc).
References:
--
Buschfeld, S., Weihs, C. (2025Fc) ``Optimizing decision trees for the analysis of World Englishes and sociolinguistic data'', Cambridge Elements.
-- Weihs, C., Buschfeld, S. (2021a) ``Combining Prediction and Interpretation in Decision Trees (PrInDT) - a Linguistic Example'' <doi:10.48550/arXiv.2103.02336>;
-- Weihs, C., Buschfeld, S. (2021b) ``NesPrInDT: Nested undersampling in PrInDT'' <doi:10.48550/arXiv.2103.14931>;

-- Weihs, C., Buschfeld, S. (2021c) ``Repeated undersampling in PrInDT (RePrInDT): Variation in undersampling and prediction, and ranking of predictors in ensembles" <doi:10.48550/arXiv.2108.05129>.

License GPL-2

Encoding UTF-8

Imports graphics, MASS, party, splitstackshape, stats, stringr, utils, gdata

RoxygenNote 7.3.2

Depends R (>= 2.10)

NeedsCompilation no

Maintainer Claus Weihs <claus.weihs@tu-dortmund.de>

Author Claus Weihs [aut, cre],
Sarah Buschfeld [aut],
Niklas Nitsch [ctb]

Repository CRAN

Date/Publication 2025-09-11 09:30:02 UTC

Contents

C2SPrInDT . . . 3
data_land . . . 5
data_speaker . . . 6
data_vowel . . . 6
data_zero . . . 7
Mix2SPrInDT . . . 8
NesPrInDT . . . 10
OptPrInDT . . . 13
participant_zero . . . 15
PostPrInDT . . . 15
PrInDT . . . 16
PrInDTAll . . . 20
PrInDTAllparts . . . 21
PrInDTCstruc . . . 23
PrInDTMulab . . . 27
PrInDTMulabAll . . . 29
PrInDTMulev . . . 31
PrInDTMulevAll . . . 33
PrInDTreg . . . 34
PrInDTregAll . . . 36
PrInDTRstruc . . . 37
R2SPrInDT . . . 41
RePrInDT . . . 43
SimCPrInDT . . . 45

SimMixPrInDT 47
SimRPrInDT 50

Index **53**

C2SPrInDT *Two-stage estimation for classification*

Description

The function C2SPrInDT applies two-stage estimation for finding an optimal model for relationships between the two-class factor variables specified as column indices of 'datain' in the vector 'inddep' and all other factor and numerical variables in the data frame 'datain' by means of 'N' repetitions of random subsampling with percentages 'percl' for the large classes and 'percs' for the small classes. One percentage of observations for each dependent variable has to be specified for the larger and the smaller class. For example, for three dependent variables, 'percl' consists of three percentages specified in the order in which the dependent variables appear in 'inddep'.
The dependent variables have to be specified as dummies, i.e. as 0 for 'property absent' or 1 for 'property present' for a certain property the dependent variable is representing.
The indices of the predictors relevant at 1st stage modeling can be specified in the vector 'indind'. If 'indind' is not specied, then all variables in 'datain' which are not specified in 'inddep' are used as predictors at 1st stage. At 2nd stage, all such variables are used as predictors anyway.
The optimization citerion is the balanced accuracy on the full sample. The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
C2SPrInDT(datain,ctestv=NA,conf.level=0.95,percl,percs,N=99,indind=NA,inddep,
                                  minsplit=NA,minbucket=NA)
```

Arguments

`datain`	Input data frame with class factor variables and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
`ctestv`	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.

conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
percl	list of undersampling percentages of larger class (numerical, > 0 and <= 1): one per dependent class variable in the same order as in 'inddep'
percs	list of undersampling percentage of smaller class (numerical, > 0 and <= 1); one per dependent class variable in the same order as in 'inddep'
N	no. of repetitions (integer > 0); default = 99
indind	indices of independent variables at stage 1; default = NA (means all independent variables used)
inddep	indices of dependent variables
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.6.1, for further information.

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.

Value

models1 Best trees at stage 1

models2 Best trees at stage 2

classnames names of classification variables

baAll balanced accuracies of best trees at both stages

Examples

```
data <- PrInDT::data_land # load data
dataclean <- data[,c(1:7,23:24,11:13,22,8:10)]  # only relevant features
indind <- c(1:9) # original predictors
inddep <- c(14:16) # dependent variables
dataland <- na.omit(dataclean)
ctestv <- NA
perc <- c(0.45,0.05,0.25)   # percentages of observations of larger class,
# 1 per dependent class variable
perc2 <- c(0.75,0.95,0.75)  # percentages of observations of smaller class,
# 1 per dependent class variable
outland <- C2SPrInDT(dataland,percl=perc,percs=perc2,N=19,indind=indind,inddep=inddep)
outland
plot(outland)
```

data_land *Landscape analysis*

Description

The use of language(s) on public signs and categorization criteria.

Usage

```
data_land
```

Format

A data frame with 149 observations and 28 columns

coder who coded the sign: Factor (3 levels "C","E","S") (anonymized)

researcher who took a photograph of the sign: Factor (2 levels "L","S") (anonymized)

sign where the sign was found: Factor (11 levels "digi","door","graf",...)

type.of.sign kind of sign: Factor (5 levels "com","commem","infra","reg","trans")

permanent was the sign permanent? Factor (2 levels "no","yes")

proper.noun kind of proper noun on the sign: Factor (9 levels "bn","bn+","cn",...)

no.languages number of languages on sign: Factor (4 levels "1","2","3","4+")

French French on sign? Factor (2 levels "0","1")

Dutch Dutch on sign? Factor (2 levels "0","1")

English English on sign? Factor (2 levels "0","1")

Italian Italian on sign? Factor (2 levels "0","1")

Spanish Spanish on sign? Factor (2 levels "0","1")

German German on sign? Factor (2 levels "0","1")

Indian,Mandarin,Greek,Indonesian,Portuguese,Libanese,Japanese,Russian,Danish, Norwegian,Hebrew,Catalan 12 infrequent languages: Factor (2 levels "0","1")

bn.unclear brand name unclear: Factor (2 levels "0","1")

multilingual.type type of multilingualism on sign: Factor (6 levels "0","1","2","3","4","uc")

location location of sign: Factor (2 levels "M","P") (anonymized)

Source

Sarah Buschfeld, TU Dortmund

data_speaker *Subject pronouns and a predictor with one very frequent level*

Description

Usage of subject pronouns and its predictors; speaker level "adult" very frequent.

Usage

```
data_speaker
```

Format

A data frame with 3370 observations and 6 columns

class subject pronoun realized? Factor (2 levels "zero","realized")

AGE age: Numerical (in months)

ETHN_GROUP ethnic group: Factor (3 levels "C","I","n_a") (anonymized)

MLU mean length of utterance: Factor (5 levels "1","2","3","adult","OL")

PRN_TYPE pronoun type: Factor (5 levels "dem","it_con","it_ex","it_ref","refer")

SPEAKER speaker: Factor (2 levels "adult","child")

Source

Sarah Buschfeld, TU Dortmund

data_vowel *Vowel length*

Description

Vowel length and categorization criteria.

Usage

```
data_vowel
```

Format

A data frame with 82 observations and 22 columns

Nickname nickname: Factor (43 levels "Nick1","Nick2",...,"Nick43") (anonymized)

LiBa linguistic background: Factor (2 levels "mono","multi")

MLU mean length of utterance: Factor (3 levels "1","2","3")

phone_label phone label: Factor (2 levels "fleece","kit")

lexeme lexeme: Factor (14 levels "bee","cheek","cheese","chicken", ...)

phone_left_1_duration duration of phone to the left of the vowel: Numerical (in msec)

phone_right_1_duration duration of phone to the right of the vowel: Numerical (in msec)

word_duration duration of word: Numerical (in msec)

vowel_minimum_pitch minimum pitch of vowel: Numerical (in Hertz)

vowel_maximum_pitch maximum pitch of vowel: Numerical (in Hertz)

vowel_intensity_mean mean intensity of vowel: Numerical (in decibel)

f1_fifty first formant F1 at midpoint of vowel (50%): Numerical (in Hertz)

f2_fifty second formant F2 at midpoint of vowel (50%): numercial (in Hertz)

target vowel length: Numerical (in msec)

cons_class_l class of consonant to the left of the vowel: Factor (6 levels "l","r","tsh",...)

cons_class_r class of consonant to the right of the vowel: Factor (7 levels "?"(glottal stop),"empty","nas",...)

ETH ethnic group: Factor (6 levels "C1a","C1b","C1c","C2a","C2b","C2c") (anonymized)

SEX gender: Factor (2 levels "female","male)

AGE age: Numerical (in months)

syllables number of syllables in lexeme: integer (1,2)

speed speed of speech: Numerical (word duration / syllables; in msec)

country country: Factor (2 levels "E","S") (anonymized)

Source

Sarah Buschfeld, TU Dortmund

`data_zero` *Subject pronouns*

Description

Usage of subject pronouns and its predictors.

Usage

```
data_zero
```

Format

A data frame with 1024 observations and 7 columns

real subject pronoun realized? Factor (2 levels "zero","realized")

AGE age: Numerical (in months)

LiBa linguistic background: Factor (3 levels "mono","multi", NA)

ETH ethnic group: Factor (6 levels "C1a","C1b","C1c","C2a","C2b","C2c") (anonymized)

SEX gender: Factor (2 levels "female","male")

MLU mean length of utterance: Factor (4 levels "1","2","3","OL")

PRN pronoun: Factor (7 levels "I","you_s","he","she","it","we","they")

Source

Sarah Buschfeld, TU Dortmund

`Mix2SPrInDT` *Two-stage estimation for classification-regression mixtures*

Description

The function Mix2SPrInDT applies 'N' repetitions of subsampling for finding an optimal subsample to model the relationship between the dependent variables specified in the sublist 'targets' of the list 'datalist' and all other factor and numerical variables in the corresponding data frame specified in the sublist 'datanames' of the list 'datalist' in the same order as 'targets'.
The function is prepared to handle classification tasks with 2 or more classes and regression tasks. At first stage, the targets are estimated only on the basis of the exogenous predictors. At second stage, summaries of the predictions of the endogenous features from the first stage models are added as new predictors.
Subsampling of observations and predictors uses the percentages specified in the matrix 'percent', one row per estimation taks. For classification tasks, percentages 'percl' and 'percs' for the larger and the smaller class have to be specified. For regression tasks, 'pobs' and 'ppre' have to specified for observations and predictors, respectively.
For generating summaries, the variables representing the substructure have to be specified in the sublist 'datastruc' of 'datalist'.
The sublist 'summ' of 'datalist' includes for each discrete target the classes for which you want to calculate the summary percentages and for each continuous target just NA (in the same order as in the sublist 'targets'). For a discrete target in this list, you can provide a sublist of classes to be combined (for an example see the below example).
The optimization citerion is the balanced accuracy for classification and R2 for regression, both evaluated on the full sample.
The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
Mix2SPrInDT(datalist,ctestv=NA,N=99,percent=NA,conf.level=0.95,minsplit=NA,minbucket=NA)
```

Arguments

`datalist`	list(datanames,targets,datastruc,summ) Input data: For specification see the above description
`ctestv`	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
`N`	Number of repetitions of subsampling for predictors (integer); default = 99
`percent`	matrix of percent spefications: For specification see the above description; default: 'percent = NA' meaning default values for percentages.
`conf.level`	(1 - significance level) in function `ctree` (numerical, > 0 and <= 1); default = 0.95
`minsplit`	Minimum number of elements in a node to be splitted; default = 20
`minbucket`	Minimum number of elements in a node; default = 7

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.6.1, for further information.

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name` where 'name' is the output data frame of the function.

Value

models1 Best trees at stage 1

models2 Best trees at stage 2

depnames names of dependent variables

nmod number of models of tasks

nlev levels of tasks

accAll accuracies of best trees at both stages

Examples

```
# zero data
datazero <- PrInDT::data_zero
datazero <- na.omit(datazero) # cleaned full data: no NAs
names(datazero)[names(datazero)=="real"] <- "zero"
CHILDzero <- PrInDT::participant_zero
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
##
# multi-level data
datastrat <- PrInDT::data_zero
datamult <- na.omit(datastrat)
# ctestv <- NA
datamult$mult[datamult$ETH %in% c("C1a","C1b","C1c") & datamult$real == "zero"] <- "zero1"
datamult$mult[datamult$ETH %in% c("C2a","C2b","C2c") & datamult$real == "zero"] <- "zero2"
datamult$mult[datamult$real == "realized"] <- "real"
datamult$mult <- as.factor(datamult$mult) # mult is new class variable
datamult$real <- NULL # remove old class variable
CHILDmult <- CHILDzero
##
# vowel data
data <- PrInDT::data_vowel
data <- na.omit(data)
CHILDvowel <- data$Nickname
data$Nickname <- NULL
syllable <- 3 - data$syllables
data$syllabels <- NULL
data$syllables <- syllable
data$speed <- data$word_duration / data$syllables
names(data)[names(data) == "target"] <- "vowel"
datavowel <- data
##
# function preparation and call
datanames <- list("datazero","datamult","datavowel")
targets <- c("zero","mult","vowel")
datastruc <- list(CHILDzero,CHILDmult,CHILDvowel)
summult <- paste("zero1","zero2",sep=",")
summ <- c("zero",summult,NA)
datalist <- list(datanames=datanames,targets=targets,datastruc=datastruc,summ=summ)
percent <- matrix(NA,nrow=3,ncol=2)
percent[1,] <- c("percl=0.075","percs=0.9") # percentages for datazero
# no percentages needed for datapast
percent[3,] <- c("pobs=0.9","ppre=c(0.9,0.8)") # percentages for datavowel
out2SMix <- Mix2SPrInDT(datalist,ctestv=ctestv,N=19,percent=percent,conf.level=0.99)
out2SMix
plot(out2SMix)
```

NesPrInDT *Nested PrInDT with additional undersampling of a factor with two unbalanced levels*

Description

Function for additional undersampling of the factor 'nesvar' with two unbalanced levels to avoid dominance of the level with higher frequency. The factor 'nesvar' is allowed not be part of the input data frame 'datain'. The data of this factor is given in the vector 'nesunder'. The observations in 'nesunder' have to represent the same cases as in 'datain' in the same ordering.
PrInDT is called 'repin' times with subsamples of the original data so that the level with the larger frequency in the vector 'nesunder' has approximately the same number of values as the level with the smaller frequency.
Only the arguments 'nesvar', 'nesunder', and 'repin' relate to the additional undersampling, all the other arguments relate to the standard PrInDT procedure.
As in PrInDT, the aim is to optimally model the relationship between the two-class factor variable 'classname' and all other factor and numerical variables in the data frame 'datain' by means of 'N' repetitions of undersampling. The trees generated by PrInDT can be restricted by excluding unacceptable trees which include split results specified in the character strings of the vector 'ctestv'.
The probability threshold 'thres' for the prediction of the smaller class may be specified (default = 0.5).
Undersampling may be stratified in two ways by the feature 'strat'.
The results are evaluated on the full sample and on the subsamples of 'nesunder'. The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Reference

Weihs, C., Buschfeld, S. 2021b. NesPrInDT: Nested undersampling in PrInDT. arXiv:2103.14931

Usage

```
NesPrInDT(datain,classname,ctestv=NA,N,plarge,psmall=1.0,conf.level=0.95,
       thres=0.5,stratvers=0,strat=NA,seedl=TRUE,nesvar,nesunder,repin,
       minsplit=NA,minbucket=NA)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
N	Number of repetitions (integer > 0)
plarge	Undersampling percentage of larger class (numerical, > 0 and <= 1)
psmall	Undersampling percentage of smaller class (numerical, > 0 and <= 1); default = 1
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
thres	Probability threshold for prediction of smaller class; default = 0.5

stratvers	Version of stratification; = 0: none (default), = 1: stratification according to the percentages of the values of the factor variable 'strat', > 1: stratification with minimum number 'stratvers' of observations per value of 'strat'
strat	Name of one (!) stratification variable for undersampling (character); default = NA (no stratification)
seedl	Should the seed for random numbers be set (TRUE / FALSE)? default = TRUE
nesvar	Name of factor to be undersampled (character)
nesunder	Data of factor to be undersampled (integer)
repin	Number of repetitions (integer) for undersampling of 'nesvar'
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify `windows(record=TRUE)` before `plot(name)` to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

undba balanced accuracies on undersamples

imax indices of best trees on undersamples

undba3en balanced accuracies of ensembles of 3 best trees on undersamples

accF balanced accuracies on full sample

accE balanced accuracy on full sample of best ensemble of 3 trees from undersampling

maxt indices of best trees on full sample

treesb 3 best trees of all undersamples of 'nesunder'; refer to an individual tree as `treesb[[k]]`, k = 1, ..., 3*repin

Examples

```
# data input and preparation --> data frame with
#   class variable, factors, and numericals (no character variables)!!
data <- PrInDT::data_speaker
data <- na.omit(data)
nesvar <- "SPEAKER"
N <- 49  # no. of repetitions in inner loop
plarge <- 0.06 # sampling percentage for larger class in nesunder-subsample
```

```
psmall <- 1 # sampling percentage for smaller class in nesunder-subsample
nesunder <- data$SPEAKER
data[,nesvar] <- list(NULL)
outNes <- NesPrInDT(data,"class",ctestv=NA,N,plarge,psmall,conf.level=0.95,nesvar=nesvar,
  nesunder=nesunder,repin=5)
outNes
plot(outNes)
hist(outNes$undba,main=" ",xlab = "balanced accuracies of 3 best trees of all undersamples")
```

OptPrInDT *Optimisation of undersampling percentages for classification*

Description

The function OptPrInDT applies an iterative technique for finding optimal undersampling percentages 'percl' for the larger class and 'percs' for the smaller class by a nested grid search for the use of the function PrInDT for the relationship between the two-class factor variable 'classname' and all other factor and numerical variables in the data frame 'data' by means of 'N' repetitions of undersampling. The optimization citerion is the balanced accuracy on the validation sample 'valdat' (default = full sample 'data'). The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The inputs plmax and psmax determine the maximal values of the percentages and the inputs distl and dists the the distances to the next smaller percentage to be tried.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.
The parameter 'steps' controls, how many of the 3 possible optimization steps should be carried out; default=3.

Usage

```
OptPrInDT(data,classname,ctestv=NA,N=99,plmax=0.09,psmax=0.9,
              distl=0.01,dists=0.1,conf.level=0.95,minsplit=NA,minbucket=NA,
              valdat=data,steps=3)
```

Arguments

data	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx.

	Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
N	Number (> 7) of repetitions (integer)
plmax	Maximal undersampling percentage of larger class (numerical, > 0 and <= 1); default = 0.09
psmax	Maximal undersampling percentage of smaller class (numerical, > 0 and <= 1); default = 0.9
distl	Distance to the next lower undersampling percentage of larger class (numerical, > 0 and < 1); default = 0.01
dists	Distance to the next lower undersampling percentage of smaller class (numerical, > 0 and < 1); default = 0.1
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7
valdat	validation data; default = data
steps	number of optimization steps = 1, 2, 3; default = 3

Details

See help("RePrInDT") and help("PrInDT") for further information.

Standard output can be produced by means of `print(name$besttree)` or just `name$besttree` as well as `plot(name$besttree)` where 'name' is the output data frame of the function.

Value

besttree best tree on full sample

bestba balanced accuracy of best tree on full sample

percl undersampling percentage of large class of best tree on full sample

percs undersampling percentage of small class of best tree on full sample

Examples

```
datastrat <- PrInDT::data_zero
data <- na.omit(datastrat) # cleaned full data: no NAs
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
# call of OptPrInDT
```

```
out <- OptPrInDT(data,"real",ctestv,N=24,conf.level=0.99,steps=1) # unstratified
out # print best model and ensembles as well as all observations
plot(out)
```

`participant_zero` *Participants of subject pronoun study*

Description

Participants (random excerpt according to data_zero)

Usage

```
participant_zero
```

Format

A dataset with 1024 observations and 1 column

participant participant of study (P1 - P51)

Source

Sarah Buschfeld, TU Dortmund

`PostPrInDT` *Posterior analysis of conditional inference trees: distribution of a specified variable in the terminal nodes.*

Description

The conditional inference tree 'ct' is analyzed according to the distribution of a variable 'var' in its terminal nodes.
In the case of a discrete variable 'var', the appearance of the different levels is considered for each terminal node.
In the case of a continuous variable 'var', means and standard deviations of 'var' or the target variable are considered for each terminal node.
In particular, this function can be used for the posterior analysis of a tree regarding the distribution of a variable not present in the tree.

Usage

```
PostPrInDT(datain, ct, target, var, vardata, vt)
```

Arguments

datain	input data frame with the observatios of all variables used in the model
ct	conditional inference tree to be analyzed
target	name of target variable of 'ct' (character)
var	name of variable of interest (character)
vardata	observations of 'var'
vt	type of variables: 'dd' for discrete target (classification) and discrete variable 'var', 'dc' for discrete target (classification) and continuous 'var', 'cd' for continuous target (regression) and discrete 'var', and 'cc' for continuous target (regression) and continuous 'var'.

Value

None: Relevant output is produced by the function.

Examples

```
data <- PrInDT::data_zero
data <- na.omit(data)
outAll <- PrInDTAll(data,"real")
PostPrInDT(data,outAll$treeAll,"real","ETH",data$ETH,vt="dd")
PostPrInDT(data,outAll$treeAll,"real","AGE",data$AGE,vt="dc")
datareg <- PrInDT::data_vowel
outregAll <- PrInDTregAll(datareg,"target")
PostPrInDT(datareg,outregAll$treeAll,"target","Nickname",datareg$Nickname,vt="cd")
PostPrInDT(datareg,outregAll$treeAll,"target","AGE",datareg$AGE,vt="cc")
```

PrInDT *The basic undersampling loop for classification*

Description

The function PrInDT uses ctrees (conditional inference trees from the package "party") for optimal modeling of the relationship between the two-class factor variable 'classname' and all other factor and numerical variables in the data frame 'datain' by means of 'N' repetitions of undersampling. The optimization citerion is the balanced accuracy on the validation sample 'valdat' (default = full input sample 'datain'). The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The undersampling percentages are 'percl' for the larger class and 'percs' for the smaller class (default = 1).
A probability threshold 'thres' for the prediction of the smaller class may be specified (default = 0.5).
Undersampling may be stratified in two ways by the option 'stratvers'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

In the case of repeated measurements ('indrep=1'), the values of the substructure variable have to be given in 'repvar'. Only one value of 'classname' is allowed for each value of 'repvar'. If for a value of 'repvar' the percentage 'thr' of the observed occurence of a value of 'classname' is not reached by the number of predictions of the value of 'classname', a misclassification is detected.

Usage

```
PrInDT(datain,classname,ctestv=NA,N,percl,percs=1,conf.level=0.95,thres=0.5,
      stratvers=0,strat=NA,seedl=TRUE,minsplit=NA,minbucket=NA,repvar=NA,indrep=0,
         valdat=datain,thr=0.5)
```

Arguments

`datain`	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
`classname`	Name of class variable (character)
`ctestv`	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
`N`	Number (> 2) of repetitions (integer)
`percl`	Undersampling percentage of larger class (numerical, > 0 and <= 1); if percs = default, default of percl = percentage of large class resulting in the same number of observations as in the small class
`percs`	Undersampling percentage of smaller class (numerical, > 0 and <= 1); default = 1
`conf.level`	(1 - significance level) in function `ctree` (numerical, > 0 and <= 1); default = 0.95
`thres`	Probability threshold for prediction of smaller class (numerical, >= 0 and < 1); default = 0.5
`stratvers`	Version of stratification; = 0: none (default), = 1: stratification according to the percentages of the values of the factor variable 'strat', > 1: stratification with minimum number "stratvers" of observations per value of "strat"
`strat`	Name of one (!) stratification variable for undersampling (character); default = NA (no stratification)

seedl	Should the seed for random numbers be set (TRUE / FALSE)? default = TRUE
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7
repvar	Values of variable defining the substructure in the case of repeated measurements, length = dim(valdat)[1] necessary; default = NA
indrep	Indicator of repeated measurements ('indrep=1'); default = 0
valdat	Validation data; default = datain
thr	threshold for element classification: minimum percentage of correct class entries; default = 0.5

Details

For the optimzation of the trees, we employ a method we call Sumping (Subsampling umbrella of model parameters), a variant of Bumping (Bootstrap umbrella of model parameters) (Tibshirani & Knight, 1999) which uses subsampling instead of bootstrapping. The aim of the optimization is to identify conditional inference trees with maximum predictive power on the full sample under interpretability restrictions.

References

– Tibshirani, R., Knight, K. 1999. Model Search and Inference By Bootstrap "bumping". Journal of Computational and Graphical Statistics, Vol. 8, No. 4 (Dec., 1999), pp. 671-686
– Weihs, C., Buschfeld, S. 2021a. Combining Prediction and Interpretation in Decision Trees (PrInDT) - a Linguistic Example. arXiv:2103.02336

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify `windows(record=TRUE)` before `plot(name)` to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

tree1st best tree on validation sample

tree2nd 2nd-best tree on validation sample

tree3rd 3rd-best tree on validation sample

treet1st best tree on test sample

treet2nd 2nd-best tree on test sample

treet3rd 3rd-best tree on test sample

ba1st accuracies: largeClass, smallClass, balanced of 'tree1st', both for validation and test sample

ba2nd accuracies: largeClass, smallClass, balanced of 'tree2nd', both for validation and test sample

ba3rd accuracies: largeClass, smallClass, balanced of 'tree3rd', both for validation and test sample

baen accuracies: largeClass, smallClass, balanced of ensemble of all interpretable, 3 best acceptable, and all acceptable trees on validation sample

bafull vector of balanced accuracies of all trees from undersampling

batest vector of test accuracies of all trees from undersampling

treeAll tree based on all observations

baAll balanced accuracy of 'treeAll'

interpAll criterion of interpretability of 'treeall' (TRUE / FALSE)

confAll confusion matrix of 'treeAll'

acc1AE Accuracy of full sample tree on Elements of large class

acc2AE Accuracy of full sample tree on Elements of small class

bamaxAE balanced accuracy of full sample tree on Elements

namA1 Names of misclassified Elements by full sample tree of large class

namA2 Names of misclassified Elements by full sample tree of small class

acc1E Accuracy of best tree on Elements of large class

acc2E Accuracy of best tree on Elements of small class

bamaxE balanced accuracy of best tree on Elements

nam1 Names of misclassified Elements by best tree of large class

nam2 Names of misclassified Elements by best tree of small class

lablarge Label of large class

labsmall Label of small class

valdat Validation set after edit

thr Threshold for repeated measurements

Examples

```
datastrat <- PrInDT::data_zero
data <- na.omit(datastrat) # cleaned full data: no NAs
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
N <- 41  # no. of repetitions
conf.level <- 0.99 # 1 - significance level (mincriterion) in ctree
percl <- 0.08  # undersampling percentage of the larger class
percs <- 0.95 # undersampling percentage of the smaller class
# calls of PrInDT
out <- PrInDT(data,"real",ctestv,N,percl,percs,conf.level) # unstratified
out # print best model and ensembles as well as all observations
plot(out)
out <- PrInDT(data,"real",ctestv,N,percl,percs,conf.level,stratvers=1,
              strat="SEX") # percentage stratification
out <- PrInDT(data,"real",ctestv,N,percl,percs,conf.level,stratvers=50,
              strat="SEX") # stratification with minimum no. of tokens
out <- PrInDT(data,"real",ctestv,N,percl,percs,conf.level,thres=0.4) # threshold = 0.4
```

PrInDTAll *Conditional inference tree (ctree) based on all observations*

Description

ctree based on all observations in 'datain'. Interpretability is checked (see 'ctestv'); probability threshold can be specified. The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Reference

Weihs, C., Buschfeld, S. 2021a. Combining Prediction and Interpretation in Decision Trees (PrInDT) - a Linguistic Example. arXiv:2103.02336

In the case of repeated measurements ('indrep=1'), the values of the substructure variable have to be given in 'repvar'. Only one value of 'classname' is allowed for each value of 'repvar'. If for a value of 'repvar' the percentage 'thr' of the observed occurence of a value of 'classname' is not reached by the number of predictions of the value of 'classname', a misclassification is detected.

Usage

```
PrInDTAll(datain, classname, ctestv=NA, conf.level=0.95, thres=0.5,
                minsplit=NA,minbucket=NA,repvar=NA,indrep=0,thr=0.5)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
thres	Probability threshold for prediction of smaller class (numerical, >= 0 and < 1); default = 0.5
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7
repvar	Values of variable defining the substructure in the case of repeated measurements, length = dim(datain)[1] necessary; default=NA
indrep	Indicator of repeated measurements ('indrep=1'); default = 0
thr	threshold for element classification: minimum percentage of correct class entries; default = 0.5

Details

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.

Value

treeall ctree based on all observations

baAll balanced accuracy of 'treeall'

interpAll criterion of interpretability of 'treeall' (TRUE / FALSE)

confAll confusion matrix of 'treeall'

acc1AE Accuracy of full sample tree on Elements of large class

acc2AE Accuracy of full sample tree on Elements of small class

bamaxAE balanced accuracy of full sample tree on Elements

namA1 Names of misclassified Elements by full sample tree of large class

namA2 Names of misclassified Elements by full sample tree of small class

lablarge Label of large class

labsmall Label of small class

thr Threshold for repeated measurements

Examples

```
datastrat <- PrInDT::data_zero
data <- na.omit(datastrat)
ctestv <- rbind('ETH == {C2a,C1a}','MLU == {1, 3}')
conf.level <- 0.99 # 1 - significance level (mincriterion) in ctree
outAll <- PrInDTAll(data,"real",ctestv,conf.level)
print(outAll) # print model based on all observations
plot(outAll) # plot model based on all observations
```

PrInDTAllparts *Conditional inference trees (ctrees) based on consecutive parts of the full sample*

Description

ctrees based on the full sample of the smaller class and consecutive parts of the larger class of the nesting variable 'nesvar'. The variable 'nesvar' has to be part of the data frame 'datain'. Interpretability is checked (see 'ctestv'); probability threshold can be specified. The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Reference

Weihs, C., Buschfeld, S. 2021b. NesPrInDT: Nested undersampling in PrInDT. arXiv:2103.14931

Usage

```
PrInDTAllparts(datain, classname, ctestv=NA, conf.level=0.95, thres=0.5,
       nesvar, divt,minsplit=NA,minbucket=NA)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
thres	Probability threshold for prediction of smaller class (numerical, >= 0 and < 1); default = 0.5
nesvar	Name of nesting variable (character)
divt	Number of parts of nesting variable nesvar for which models should be determined individually
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of `print(name)` or just `name` where 'name' is the output data frame of the function.

Value

baAll balanced accuracy of tree on full sample

nesvar name of nesting variable

divt number of consecutive parts of the sample

badiv balanced accuracy of trees on 'divt' consecutive parts of the sample

Examples

```
data <- PrInDT::data_speaker
data <- na.omit(data)
nesvar <- "SPEAKER"
outNesAll <- PrInDTAllparts(data,"class",ctestv=NA,conf.level=0.95,thres=0.5,nesvar,divt=8)
outNesAll
```

PrInDTCstruc *Structured subsampling for classification*

Description

The function PrInDTCstruc applies structured subsampling for finding an optimal subsample to model the relationship between the two-class factor variable 'classname' and all other factor and numerical variables in the data frame 'datain' by means of 'N' repetitions of subsampling from a substructure and 'Ni' repetitions of subsampling from the predictors. The optimization citerion is the balanced accuracy on the validation sample 'valdat' (default is the full input sample 'datain'). Other criteria are possible (cf. parameter description of 'crit'). The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The substructure of the observations used for subsampling in modelling is specified by the list 'Struc' which consists of the factor variable 'name' representing the substructure, the name 'check' of the factor variable with the information about the categories of the substructure, and the matrix 'labs' which specifies the values of 'check' corresponding to two categories in its rows, i.e. in 'labs[1,]' and 'labs[2,]'. The names of the categories have to be specified by `rownames(labs)`.
See parameter description of 'Struc' for its specification for 'vers="b"' and 'indrep > 0'.
The number of predictors 'Pit' to be included in the model and the number of elements of the substructure 'Eit' have to be specified (lists allowed), and undersampling of the categories of 'classname' can be controlled by 'undersamp=TRUE/FALSE'.
In structured subsampling, 'N' repetitions of subsampling of the variable 'name' with 'Eit' different elements of each category in 'check' are realized. If 'Eit' is a list, each entry is employed individually. If 'Eit' is larger than the maximum available number of elements with a certain value of 'check', the maximum possible number of elements is used.
Four different versions of structured subsampling exist:
a) just of the elements in the substructure (possibly with additional undersampling) with parameters 'N' and 'Eit',
b) just of the predictors with parameters 'Ni' and 'Pit',
c) of the predictors and for each subset of predictors subsampling of the elements of the substructure (possibly with additional undersampling) with parameters 'N', 'Ni', 'Eit', and 'Pit', and
d) of the elements of the substructure (possibly with additional undersampling) and for each of these subsets subsampling of the predictors with the same parameters as version c).
Sampling of the elements of the substructure can be influenced by using weights of the elements ('weights=TRUE') according to the number of appearances of the smaller class of 'classnames'. This way, elements with more realisations in the smaller class are preferred in modelling.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

The parameter 'indrep' indicates repeated measurement situations ('indrep > 0', only implemented for 'crit="ba"'); default = 0.
Repeated measurements are multiple measurements of the same variable taken on the same subjects (or objects) either under different conditions or over two or more time periods.
The variable with the repeatedly observed subjects (or objects) has to be specified by 'name' in 'Struc'. Only one value of 'classname' is allowed for each value of 'Struc$name'. In case of 'indrep > 0' it is automatically assumed that the same number of subjects (or objects) of the two classes under study have to be used for model building. Possible such numbers can be specified by 'Eit'.

For indrep=1, models are optimized for individual observations and the classes of the substructure elements are subsequently determined via the parameter 'thr'.
For indrep=2, models are optimized so that the criterion "ba" is maximal for the substructure elements taking 'thr' into account.
For indrep > 0 and 'valdat' unequal 'datain', the variable 'repvar' defines the substructure for 'valdat', length = dim(valdat)[1] necessary; default = NA'

Usage

```
PrInDTCstruc(datain,classname,ctestv=NA,Struc=NA,vers="d",weight=FALSE,
                Eit=NA,Pit=NA,N=99,Ni=99,undersamp=TRUE,crit="ba",ktest=0,
                stest=integer(length=0),conf.level=0.95,indrep=0,
                minsplit=NA,minbucket=NA,repvar=NA,valdat=datain,thr=0.5)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
Struc	= list(name,check,labs), cf. description for explanations; Struc not needed for vers="b"; for indrep=1, Struc = list(name);
vers	Version of structured subsampling: "a", "b", "c", "d", cf. description; default = "d"
weight	Weights to be used for subsampling of elements of substructure (logical, TRUE or FALSE); default = FALSE
Eit	List of number of elements of substructure (integers); default = c((Cl-4):Cl), Cl = maximum number of elements in both categories
Pit	List of number of predictors (integers) default = c(max(1,(D-3)):D), D = maximum number of predictors
N	Number of repetitions of subsampling from substructure (integer) default = 0 for vers="b", = 99 otherwise; if vers="b", any input is overwritten by the default
Ni	Number of repetitions of subsampling from predictors default = 0 for vers="a", = 99 for vers="b", = N otherwise; if vers="a", any input is overwritten by the default

undersamp	Undersampling of categories of 'classname' to be used ((logical, TRUE or FALSE) default = TRUE; if indrep=1 or vers="b", undersamp = FALSE
crit	Optimisation criterion: "ba" for balanced accuracy, "bat" for balanced accuracy on test sets, "ta" for test accuracy, "tal" for test accuracy of continuing parts of length 'ktest' in substructure elements 'stest'; default = "ba"
ktest	Length of continuing parts to be tested (for crit="tal"); default = 0
stest	Part of substructure to be tested (for crit="tal")(integer vector); default = integer vector of length 0
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
indrep	Indicator for repeated measurements, i.e. more than one observation with the same class for each element; for indrep=1, Struc=list(name) only; default = 0
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7
repvar	Values of variable defining the substructure corresponding to valdat in the case of repeated measurements, length = dim(valdat)[1] necessary; default = NA
valdat	validation data; default = datain
thr	threshold for element classfication: minimum percentage of correct class entries; default = 0.5

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.1, for further information.

Standard output can be produced by means of `print(name$besttree)` or just `name$besttree` as well as `plot(name$besttree)` where 'name' is the output data frame of the function.

Value

modbest Best tree

interp Number of interpretable trees, overall number of trees

dmax Number of predictors in training set for best tree

ntmax Size of training set for best tree

acc1 Accuracy of best tree on large class

acc2 Accuracy of best tree on small class

bamax Balanced accuracy of best tree

tamax Test accuracy of best tree

kumin Number of elements with misclassified parts longer than 'ktest' for best tree

elems Elements with long misclassified parts for best tree

mindlong Indices of long misclassified parts for best tree

ind1max Elements of 1st category of substructure used by best tree

ind2max Elements of 2nd category of substructure used by best tree

indmax Predictors used by best tree

bestTrain Training set for best tree

bestTest Test set for best tree

labs labs from Struc

lablarge Label of large class

labsmall Label of small class

vers Version used for structured subsampling

acc1E Accuracy of large class on Elements of best tree

acc2E Accuracy of small class on Elements of best tree

bamaxE Balanced accuracy of best tree on Elements

nam1 Names of misclassified Elements of large class

nam2 Names of misclassified Elements of small class

thr Threshold for element classification

Examples

```
data <- PrInDT::data_zero
data <- na.omit(data) # cleaned full data: no NAs
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
# substructure
name <- PrInDT::participant_zero
check <- "data$ETH"
labs <- matrix(1:6,nrow=2,ncol=3)
labs[1,] <- c("C1a","C1b","C1c")
labs[2,] <- c("C2a","C2b","C2c")
rownames(labs) <- c("Children 1","Children 2")
Struc <- list(name=name,check=check,labs=labs)
out <- PrInDTCstruc(data,"real",ctestv,Struc,vers="c",weight=TRUE,N=5,Pit=5,conf.level=0.99)
out
plot(out)
# indrep = 1
Struc <- list(name=name)
out <- PrInDTCstruc(data,"real",Struc=Struc,vers="c",Pit=5,Eit=5,N=5,crit="ba",indrep=1,
           conf.level=0.99)
```

PrInDTMulab *Multiple label classification based on resampling by* PrInDT

Description

Multiple label classification based on resampling by PrInDT. We consider two ways of modeling (Binary relevance modeling, dependent binary modeling) and three ways of model evaluation: single assessment, joint assessment, and true prediction (see the Value section for more information).
Variables should be arranged in 'datain' according to indices specified in 'indind', 'indaddind', and 'inddep'.
Please note that the dependent variables have to be specified as dummies, i.e. as 'absent' (value 0) or 'present' (value 1).
Undersampling is repeated 'N' times.
Undersampling percentages 'percl' for the larger class and 'percs' for the smaller class can be specified, one each per dependent class variable.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

References

- Buschfeld, S., Weihs, C. and Ronan, P.. 2024. Modeling linguistic landscapes: A comparison of St Martin's two capitals Philipsburg and Marigot Linguistic Landscape 10(3): 302–334. https://doi.org/10.1075/ll.23070.bus
- Probst, P., Au, Q., Casalicchio, G., Stachl, C., and Bischl, B. 2017. Multilabel Classification with R Package mlr. arXiv:1703.08991v2

Usage

```
PrInDTMulab(datain,classnames=NA,ctestv=NA,conf.level=0.95,percl,percs=1,N,
              indind=NA,indaddind=NA,inddep,minsplit=NA,minbucket=NA)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classnames	names of class variables (character vector); default = NA: from old version: superfluous now
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
percl	list of undersampling percentages of larger class (numerical, > 0 and <= 1): one per dependent class variable
percs	list of undersampling percentage of smaller class (numerical, > 0 and <= 1); one per dependent class variable

N	no. of repetitions (integer > 0)
indind	indices of independent variables
indaddind	indices of additional independent variables used in the case of dependent binary relevance modeling
inddep	indices of dependent variables
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify `windows(record=TRUE)` before `plot(name)` to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

accbr model errors for Binary Relevance (single assessment) - only independent predictors are used for modeling one label at a time, the other labels are not used as predictors. As the performance measure for the resulting classification rules, the balanced accuracy of the best model from PrInDT is employed for each individual label.

errbin combined error for Binary Relevance (joint assessment) - the best prediction models for the different labels are combined to assess the combined prediction. The 01-accuracy counts a label combination as correct only if all labels are correctly predicted. The hamming accuracy corresponds to the proportion of labels whose value is correctly predicted.

accdbr model errors for Dependent Binary Relevance (Extended Model) (single assessment) - each label is trained by means of an extended model which not only includes the independent predictors but also the other labels. For these labels, the truly observed values are used for estimation and prediction. In the extended model, other labels, which are not treated as dependent variables, can also be used as additional predictors.

errext combined errors for Dependent Binary Relevance (Extended Model) (joint assessment)

errtrue combined errors for Dependent Binary Relevance (True Prediction) - in the prediction phase, the values of all modeled labels are first predicted by the independent predictors only and then the predicted labels are used in the estimated extended model in a 2nd step to ultimately predict the labels.

coldata column names of input data

inddep indices of dependent variables (labels to be modeled)

treebr list of trees from Binary Relevance modeling, one tree for each label; refer to an individual tree as `treebr[[i]]`, i = 1, ..., no. of labels

treedbr list of trees from Dependent Binary Relevance modeling, one for each label; refer to an individual tree as `treedbr[[i]]`, i = 1, ..., no. of labels

Examples

```
data <- PrInDT::data_land # load data
dataclean <- data[,c(1:7,23:24,11:13,22,8:10)]  # only relevant features
indind <- c(1:9) # original predictors
indaddind <- c(10:13) # additional predictors
inddep <- c(14:16) # dependent variables
dataclean <- na.omit(dataclean)
ctestv <- NA
N <- 21  # no. of repetitions
perc <- c(0.45,0.05,0.25)   # percentages of observations of larger class,
#                             1 per dependent class variable
perc2 <- c(0.75,0.95,0.75)  # percentages of observations of smaller class,
#                             1 per dependent class variable
##
# Call PrInDT: language by language
##
outmult <- PrInDTMulab(dataclean,ctestv=NA,conf.level=0.95,percl=perc,percs=perc2,N=N,
                  indind=indind,indaddind=indaddind,inddep=inddep)
print(outmult)
plot(outmult)
```

`PrInDTMulabAll` *Multiple label classification based on all observations*

Description

Multiple label classification based on all observations. We consider two ways of modeling (Binary relevance modeling, dependent binary modeling) and three ways of model evaluation: single assessment, joint assessment, and true prediction (see the Value section for more information).
Interpretability is checked (see ctestv).
Variables should be arranged in 'datain' according to indices specified in 'indind', 'indaddind', and 'inddep'.
Please note that the dependent variables have to be specified as dummies, i.e. as 'absent' (value 0) or 'present' (value 1).
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Reference

Probst, P., Au, Q., Casalicchio, G., Stachl, C., and Bischl, B. 2017. Multilabel Classification with R Package mlr. arXiv:1703.08991v2

Usage

```
PrInDTMulabAll(datain,classnames=NA,ctestv=NA,conf.level=0.95,indind=NA,
               indaddind=NA,inddep,minsplit=NA,minbucket=NA)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classnames	names of class variables (character vector)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
indind	indices of independent variables
indaddind	indices of additional predictors used in the case of dependent binary relevance modeling
inddep	indices of dependent variables
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify `windows(record=TRUE)` before `plot(name)` to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

accabr model errors for Binary Relevance (single assessment) - only independent predictors are used for modeling one label at a time, the other labels are not used as predictors. The classification rules are trained on all observations. As the performance measure for the resulting classification rules, the balanced accuracy of the models for each individual label is employed.

errabin combined error for Binary Relevance (joint assessment) - the best prediction models for the different labels are combined to assess the combined prediction. The 01-accuracy counts a label combination as correct only if all labels are correctly predicted. The hamming accuracy corresponds to the proportion of labels whose value is correctly predicted.

accadbr model errors in Dependent Binary Relevance (Extended Model) (single assessment) - each label is trained by means of an extended model which not only includes the independent predictors but also the other labels. For these labels the truly observed values are used for estimation and prediction. In the extended model, further labels, which are not treated as dependent variables, can be used as additional predictors.

erraext combined errors for Dependent Binary Relevance (Extended Model) (joint assessment)

erratrue combined errors for Dependent Binary Relevance (True Prediction) - in the prediction phase, the values of all modeled labels are first predicted by the independent predictors only (see Binary Relevance) and then the predicted labels are used in the estimated extended model in a 2nd step to ultimately predict the labels.

coldata column names of input data

inddep indices of dependent variables (labels to be modeled)

treeabr list of trees from Binary Relevance modeling, one tree for each label; refer to an individual tree as `treeabr[[i]]`, i = 1, ..., no. of labels

treeadbr list of trees from Dependent Binary Relevance modeling, one for each label; refer to an individual tree as `treeadbr[[i]]`, i = 1, ..., no. of labels

Examples

```
data <- PrInDT::data_land # load data
dataclean <- data[,c(1:7,23:24,11:13,22,8:10)]  # only relevant features
indind <- c(1:9) # original predictors
indaddind <- c(10:13) # additional predictors
inddep <- c(14:16) # dependent variables
dataclean <- na.omit(dataclean)
ctestv <- NA
##
# Call PrInDTAll: language by language
##
outmultAll <- PrInDTMulabAll(dataclean,colnames(dataclean)[inddep],ctestv,conf.level=0.95,
                    indind,indaddind,inddep)
outmultAll
plot(outmultAll)
```

`PrInDTMulev` *PrInDT analysis for a classification problem with multiple classes.*

Description

PrInDT analysis for a classification problem with more than 2 classes. For each combination of one class vs. the other classes a 2-class `PrInDT` analysis is carried out.
The percentages for undersampling of the larger class ('percl' in `PrInDT`) are chosen so that the resulting sizes are comparable with the size of the smaller classes for which all their observations are used in undersampling ('percs' = 1 in `PrInDT`).
The class with the highest probability in the K (= number of classes) analyses is chosen for prediction.
Interpretability is checked (see 'ctestv'). The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
PrInDTMulev(datain,classname,ctestv=NA,N,conf.level=0.95,seedl=FALSE,
                                    minsplit=NA,minbucket=NA)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
N	Number of repetitions (integer > 0)
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1) (default = 0.95)
seedl	Should the seed for random numbers be set (TRUE / FALSE)? default = FALSE
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of print(name) or just name as well as plot(name) where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify windows(record=TRUE) before plot(name) to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

class levels of class variable

trees trees for the levels of the class variable; refer to an individual tree as trees[[k]], k = 1, ..., no. of levels

ba balanced accuracy of combined predictions

conf confusion matrix of combined predictions

ninterp no. of non-interpretable trees

acc balanced accuracies of best models for individual classes

Examples

```
datastrat <- PrInDT::data_zero
data <- na.omit(datastrat)
ctestv <- NA
data$rel[data$ETH %in% c("C1a","C1b","C1c") & data$real == "zero"] <- "zero1"
data$rel[data$ETH %in% c("C2a","C2b","C2c") & data$real == "zero"] <- "zero2"
data$rel[data$real == "realized"] <- "real"
data$rel <- as.factor(data$rel) # rel is new class variable
```

```
data$real <- NULL # remove old class variable
N <- 51
conf.level <- 0.99 # 1 - significance level (mincriterion) in ctree
out <- PrInDTMulev(data,"rel",ctestv,N,conf.level)
out # print best models based on subsamples
plot(out) # corresponding plots
```

`PrInDTMulevAll` *Conditional inference tree (ctree) for multiple classes on all observations*

Description

ctree for more than 2 classes on all observations. Interpretability is checked (see 'ctestv'). The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
PrInDTMulevAll(datain,classname,ctestv=NA,conf.level=0.95,minsplit=NA,minbucket=NA)
```

Arguments

`datain`	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
`classname`	Name of class variable (character)
`ctestv`	Vector of character strings of forbidden split results; (see function `PrInDT` for details.) If no restrictions exist, the default = NA is used.
`conf.level`	(1 - significance level) in function `ctree` (numerical, > 0 and <= 1) (default = 0.95)
`minsplit`	Minimum number of elements in a node to be splitted; default = 20
`minbucket`	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.

Value

treeall ctree based on all observations

baAll balanced accuracy of 'treeall'

interpAll criterion of interpretability of 'treeall' (TRUE / FALSE)

confAll confusion matrix of 'treeall'

Examples

```
datastrat <- PrInDT::data_zero
data <- na.omit(datastrat)
ctestv <- rbind('ETH == {C2a,C1a}', 'MLU == {1, 3}')
data$rel[data$ETH %in% c("C1a","C1b","C1c") & data$real == "zero"] <- "zero1"
data$rel[data$ETH %in% c("C2a","C2b","C2c") & data$real == "zero"] <- "zero2"
data$rel[data$real == "realized"] <- "real"
data$rel <- as.factor(data$rel) # rel is new class variable
data$real <- NULL # remove old class variable
conf.level <- 0.99 # 1 - significance level (mincriterion) in ctree
outAll <- PrInDTMulevAll(data,"rel",ctestv,conf.level)
outAll # print model based on all observations
plot(outAll)
```

PrInDTreg *Regression tree resampling by the PrInDT method*

Description

The function PrInDTreg realizes regression tree optimization to identify the best interpretable tree; interpretability is checked (see 'ctestv').
The relationship between the target variable 'regname' and all other factor and numerical variables in the data frame 'datain' is optimally modeled by means of 'N' repetitions of subsampling.
The optimization criterion is the R2 of the model on the validation sample 'valdat'.
Default for the validation sample is the full sample 'datain'.
Multiple subsampling percentages of observations and predictors can be specified (in 'pobs' and 'ppre', correspondingly).
The trees generated from subsampling can be restricted by rejecting unacceptable trees which include split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.
Besides the maximal R2, the minimal MAE (Mean Absolute Error) is reported.

Usage

```
PrInDTreg(datain,regname,ctestv=NA,N,pobs=c(0.9,0.7),ppre=c(0.9,0.7),
            conf.level=0.95,seedl=TRUE,minsplit=NA,minbucket=NA,valdat=datain)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
regname	name of regressand variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
N	Number of repetitions (integer > 0)
pobs	Vector of resampling percentages of observations (numerical, > 0 and <= 1)
ppre	Vector of resampling percentages of predictor variables (numerical, > 0 and <= 1)
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
seedl	Should the seed for random numbers be set (TRUE / FALSE)? default = TRUE
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7
valdat	Validation data; default = datain

Details

For the optimzation of the trees, we employ a method we call Sumping (Subsampling umbrella of model parameters), a variant of Bumping (Bootstrap umbrella of model parameters) (Tibshirani & Knight, 1999). We use subsampling instead of bootstrapping. The aim of the optimization is to identify conditional inference trees with maximum predictive power on the full sample under interpretability restrictions.

Reference

Tibshirani, R., Knight, K. 1999. Model Search and Inference By Bootstrap "bumping". Journal of Computational and Graphical Statistics, Vol. 8, No. 4 (Dec., 1999), pp. 671-686

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify `windows(record=TRUE)` before `plot(name)` to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

meanint Mean number of interpretable trees over the combinations of individual percentages in 'pobs' and 'ppre'

ctmax best resampled regression tree on the validation data set

percmax Best model achieved for %observations

perfeamax Best model achieved for %predictors

maxR2 maximum R2 on the validation data set for resampled regression trees (for 'ctmax')

minMAE minimum MAE (Mean Absolute Error) on the validation data set for resampled regression trees (for 'ctmax')

interpmax interpretability of best tree 'ctmax'

ctmax2 second best resampled regression tree on the full data set

percmax2 second best model achieved for %observations

perfeamax2 second best model achieved for %features

max2R2 second best R2 on the validation data set for resampled regression trees (for 'ctmax2')

min2MAE second best MAE on the validation data set for resampled regression trees (for 'ctmax2')

interp2max interpretability of second-best tree 'ctmax2'

Examples

```
data <- PrInDT::data_vowel
data <- na.omit(data)
ctestv <- 'vowel_maximum_pitch <= 320'
N <- 30 # no. of repetitions
pobs <- c(0.70,0.60) # percentages of observations
ppre <- c(0.90,0.70) # percentages of predictors
outreg <- PrInDTreg(data,"target",ctestv,N,pobs,ppre)
outreg
plot(outreg)
```

PrInDTregAll *Regression tree based on all observations*

Description

Regression tree based on the full sample; interpretability is checked (see 'ctestv').
The relationship between the target variable 'regname' and all other factor and numerical variables in the data frame 'datain' is modeled based on all observations.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.
Besides the maximal R2, the minimal MAE (Mean Absolute Error) is reported.

Usage

```
PrInDTregAll(datain,regname,ctestv=NA,conf.level=0.95,minsplit=NA,minbucket=NA)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
regname	name of regressand variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

Standard output can be produced by means of print(name) or just name as well as plot(name) where 'name' is the output data frame of the function.

Value

treeall tree based on all observations

R2All goodness of fit of 'treeall' based on all observations

MAEAll MAE of 'treeall' based on all observations

interpAll criterion of interpretability of 'treeall' (TRUE / FALSE)

Examples

```
data <- PrInDT::data_vowel
data <- na.omit(data)
ctestv <- 'vowel_maximum_pitch <= 320'
outreg <- PrInDTregAll(data,"target",ctestv)
outreg
plot(outreg)
```

PrInDTRstruc *Structured subsampling for regression*

Description

The function PrInDTRstruc applies structured subsampling for finding an optimal subsample to model the relationship between the continuous variable 'regname' and all other factor and numerical variables in the data frame 'datain' by means of 'M' repetitions of subsampling from a substructure and 'N' repetitions of subsampling from the predictors. The optimization citerion is the goodness of fit R2 on the validation sample 'valdat' (default = 'datain'). The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The substructure of the observations used for subsampling is specified by the list 'Struc' which consists of the factor variable 'name' representing the substructure, the name 'check' of the factor variable with the information about the categories of the substructure, and the matrix 'labs' which specifies the values of 'check' corresponding to two categories in its rows, i.e. in 'labs[1,]' and 'labs[2,]'. The names of the categories have to be specified by `rownames(labs)`.
In structured subsampling, 'M' repetitions of subsampling of the variable 'name' with 'Mit' different elements of each category in 'check' are realized. If 'Mit' is a list, each entry is employed individually. If 'Mit' is larger than the maximum available number of elements with a certain value of 'check', the maximum possible number of elements is used.
The number of predictors 'Pit' to be included in the model and the number of elements of the substructure 'Mit' have to be specified (lists allowed).
The percentages of involved observations and predictors can be controlled by the parameters 'pobs' and 'ppre', respectively.
The parameter 'Struc' is needed for all versions of subsampling except "b". Four different versions of structured subsampling exist:
a) just of the elements in the substructure with parameters 'M' and 'Mit',
b) just of the predictors with parameters 'N' and 'Pit',
c) of the predictors and for each subset of predictors subsampling of the elements of the substructure with parameters 'M', 'N', 'Mit', 'Pit', 'pobs', and 'ppre', and
d) of the elements of the substructure and for each of these subsets subsampling of the predictors with the same parameters as version c).
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.
Besides the maximal R2, the minimal MAE (Mean Absolute Error) is reported.

Repeated measurements can also be handled by this function (indrep=1). They are multiple measurements of the same variable taken on the same subjects (or objects) either under different conditions or over two or more time periods.
The variable with the repeatedly observed subjects (or objects) is assumed to be 'name' in 'Struc'.
The measure MAE is split according to the values of 'name'.

Usage

```
PrInDTRstruc(datain,regname,ctestv=NA,Struc=NA,vers="d",M=NA,Mit=NA,N=99,Pit=NA,
          pobs=c(0.9,0.7),ppre=c(0.9,0.7),conf.level=0.95,minsplit=NA,minbucket=NA,
               valdat=datain,indrep=0)
```

Arguments

`datain`	Input data frame with continuous target variable 'regname' and the influential variables, which need to be factors or numericals (transform logicals

	and character variables to factors)
`regname`	Name of target variable (character)
`ctestv`	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
`Struc`	= list(name,check,labs), cf. description for explanations; Struc not needed for vers="b"
`vers`	Version of structured subsampling: "a", "b", "c", "d", cf. description; default = "d"
`M`	Number of repetitions of subsampling from substructure (integer) in versions "a" and "d"; default = 99
`Mit`	List of number of elements of substructure (integers); default = c((Cl-4):Cl), Cl = maximum number elements in both categories
`N`	Number of repetitions of subsampling from predictors (integer) in versions "b" and "c"; default = 99
`Pit`	List of number of predictors (integers) default = c(max(1,(D-3)):D), D = maximum number of predictors
`pobs`	Percentage(s) of observations for subsampling in versions "c" and "d"; default=c(0.9,0.7)
`ppre`	Percentage(s) of predictors for subsampling in versions "c" and "d"; default=c(0.9,0.7)
`conf.level`	(1 - significance level) in function `ctree` (numerical, > 0 and <= 1); default = 0.95
`minsplit`	Minimum number of elements in a node to be splitted; default = 20
`minbucket`	Minimum number of elements in a node; default = 7
`valdat`	Validation data; default = datain
`indrep`	Indicator for repeated measurements, i.e. more than one observation with the same class for each element; indrep=1: Struc=list(name) only; default = 0

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.4, for further information.

Standard output can be produced by means of `print(name$besttree)` or just `name$besttree` as well as `plot(name$besttree)` where 'name' is the output data frame of the function.

Value

outmax Best tree

interp Number of interpretable trees, overall number of trees

ntmax Size of training set for best tree

R2max R squared of best tree

R2sub Mean R squared of objects in substructure

MAEmax MAE (Mean Absolute Error) of best tree

MAEsub Mean MAE of objects in substructure

ind1max Elements of 1st category of substructure used by best tree

ind2max Elements of 2nd category of substructure used by best tree

indmax Predictors used by best tree

gmaxTrain Training set for best tree

labs labs from Struc

vers Version used for structured subsampling

lMit Number of different numbers of substructure elements

lPit Number of different numbers of predictors

M Number of repetitions of selection of substructure elements

N Number of repetitions of selection of predictors

indrep Indicator of repeated measurements: indrep=1

Examples

```
data <- PrInDT::data_vowel
data <- na.omit(data)
CHILDvowel <- data$Nickname
data$Nickname <- NULL
data$syllables <- 3 - data$syllables
data$speed <- data$word_duration / data$syllables  ## NEW NEW
names(data)[names(data) == "target"] <- "vowel_length"
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
name <- CHILDvowel
check <- "data$ETH"
labs <- matrix(1:6,nrow=2,ncol=3)
labs[1,] <- c("C1a","C1b","C1c")
labs[2,] <- c("C2a","C2b","C2c")
rownames(labs) <- c("children 1","children 2")
Struc <- list(name=name,check=check,labs=labs)
outstruc <- PrInDTRstruc(data,"vowel_length",ctestv=ctestv,Struc=Struc,vers="d",
                  M=3,Mit=21,N=9,pobs=c(0.95,0.7),ppre=c(1,0.7),conf.level=0.99)
```

```
outstruc
plot(outstruc)
```

R2SPrInDT *Two-stage estimation for regression*

Description

The function R2SPrInDT applies 'N' repetitions of subsampling for finding an optimal subsample to model the relationship between the continuous variables with indices 'inddep' and all other factor and numerical variables in the data frame 'datain'.
Subsampling of observations and predictors uses the percentages in 'pobs1' and 'ppre1', respectively, at stage 1, and the percentages 'pobs2' and 'ppre2' at stage 2, accordingly. The optimization criterion is the goodness of fit R2 on the full sample.
The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
R2SPrInDT(data,ctestv=NA,inddep,N=99,pobs1=c(0.90,0.70),ppre1=c(0.90,0.70),
              pobs2=pobs1,ppre2=ppre1,conf.level=0.95,minsplit=NA,minbucket=NA)
```

Arguments

data	Input data frame with continuous target variable 'regname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
ctestv	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
inddep	Column indices of target variables in datain
N	Number of repetitions of subsampling from predictors (integer) in versions "b" and "c"; default = 99
pobs1	Percentage(s) of observations for subsampling at stage 1; default=c(0.9,0.7)

ppre1	Percentage(s) of predictors for subsampling at stage 1; default=c(0.9,0.7)
pobs2	Percentage(s) of observations for subsampling at stage 2"; default=pobs1
ppre2	Percentage(s) of predictors for subsampling at stage 2; default=ppre1
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.6.1, for further information.

Standard output can be produced by means of print(name) or just name as well as plot(name where 'name' is the output data frame of the function.

Value

models1 Best trees at stage 1

models2 Best trees at stage 2

depnames names of dependent variables

R2both R2s of best trees at both stages

Examples

```
data <- PrInDT::data_vowel
data <- na.omit(data)
CHILDvowel <- data$Nickname
data$Nickname <- NULL
syllable <- 3 - data$syllables
data$syllabels <- NULL
data$syllables <- syllable
data$speed <- data$word_duration / data$syllables
names(data)[names(data) == "target"] <- "vowel_length"
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
inddep <- c(13,9)
out2SR <- R2SPrInDT(data,ctestv=ctestv,inddep=inddep,N=9,conf.level=0.99)
out2SR
plot(out2SR)
```

RePrInDT *Repeated* PrInDT *for specified percentage combinations*

Description

By the function RePrInDT, the function PrInDT is called repeatedly according to all combinations of the percentages specified in the vectors 'plarge' and 'psmall'.
The relationship between the two-class factor variable 'classname' and all other factor and numerical variables in the data frame 'datain' is optimally modeled by means of 'N' repetitions of undersampling.
The optimization citerion is the balanced accuracy on the validation sample 'valdat' (default = full input sample 'datain').
The trees generated from undersampling can be restricted by rejecting unacceptable trees which include split results specified in the character strings of the vector 'ctestv'.
The probability threshold 'thres' for the prediction of the smaller class may be specified (default = 0.5).
Undersampling may be stratified in two ways by the feature 'strat'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Reference

Weihs, C., Buschfeld, S. 2021c. Repeated undersampling in PrInDT (RePrInDT): Variation in undersampling and prediction, and ranking of predictors in ensembles. arXiv:2108.05129

Usage

```
RePrInDT(datain, classname, ctestv=NA, N, plarge, psmall, conf.level=0.95,
       thres=0.5, stratvers=0, strat=NA, seedl=TRUE,minsplit=NA,minbucket=NA,
       valdat=datain)
```

Arguments

datain	Input data frame with class factor variable 'classname' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
classname	Name of class variable (character)
ctestv	Vector of character strings of forbidden split results; (see function PrInDT for details.) If no restrictions exist, the default = NA is used.
N	Number of repetitions (integer > 0)
plarge	Vector of undersampling percentages of larger class (numerical, > 0 and <= 1)
psmall	Vector of undersampling percentages of smaller class (numerical, > 0 and <= 1)
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
thres	Probability threshold for prediction of smaller class (numerical, >= 0 and < 1); default = 0.5

stratvers	Version of stratification; = 0: none (default), = 1: stratification according to the percentages of the values of the factor variable 'strat', > 1: stratification with minimum number 'stratvers' of observations per value of 'strat'
strat	Name of one (!) stratification variable for undersampling (character); default = NA (no stratification)
seedl	Should the seed for random numbers be set (TRUE / FALSE)? default = TRUE
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7
valdat	Validation data; default = datain

Details

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.
The plot function will produce a series of more than one plot. If you use R, you might want to specify `windows(record=TRUE)` before `plot(name)` to save the whole series of plots. In R-Studio this functionality is provided automatically.

Value

treesb best trees for the different percentage combinations; refer to an individual tree as `treesb[[k]]`, k = 1, ..., length(plarge)*length(psmall)

acc1st accuracies of best trees on full sample

acc3en accuracies of ensemble of 3 best trees on full sample

simp_m mean of permutation losses for the predictors

Examples

```
datastrat <- PrInDT::data_zero
data <- na.omit(datastrat) # cleaned full data: no NAs
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}', 'MLU == {1, 3}')
N <- 51  # no. of repetitions
conf.level <- 0.99 # 1 - significance level (mincriterion) in ctree
psmall <- c(0.95,1)     # percentages of the small class
plarge <- c(0.09,0.1)  # percentages of the large class
outRe <- RePrInDT(data,"real",ctestv,N,plarge,psmall,conf.level)
outRe
plot(outRe)
```

SimCPrInDT *Interdependent estimation for classification*

Description

The function SimCPrInDT applies interdependent estimation (endogenous case) for finding an optimal model for relationships between the two-class factor variables specified as column indices of 'data' in the vector 'inddep' and all other factor and numerical variables in the data frame 'data' by means of 'N' repetitions of random subsampling with percentages 'percl' for the large classes and 'percs' for the small classes. One percentage of observations for each dependent variable has to be specified for the larger and the smaller class. For example, for three dependent variables, 'percl' consists of three percentages specified in the order in which the dependent variables appear in 'inddep'.
The dependent variables have to be specified as dummies, i.e. as 'property absent' (value 0) or 'property present' (value 1).
The optimization citerion is the balanced accuracy on the full sample.
In an additional step, the mean balanced accuracy over all class variables is optimized (joint optimization).
The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
SimCPrInDT(data,ctestv=NA,inddep,percl,percs,N=99,M,psize,conf.level=0.95,
                                    minsplit=NA,minbucket=NA)
```

Arguments

data	Input data frame with class factor variables and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)
ctestv	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
inddep	indices of dependent variables
percl	list of undersampling percentages of larger class (numerical, > 0 and <= 1): one per dependent class variable in the same order as in 'inddep'

percs	list of undersampling percentage of smaller class (numerical, > 0 and <= 1); one per dependent class variable in the same order as in 'inddep'
N	no. of repetitions of subsampling of observations (integer > 0); default = 99
M	no. of repetitions of subsampling of predictors
psize	no. of predictors in the subsamples of the predictors
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.6.2, for further information.

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name)` where 'name' is the output data frame of the function.

Value

models1 Best trees at stage 1

models2 Best trees at stage 2

models3 Best trees from mean maximization

classnames names of classification variables

baAll balanced accuracies of best trees at both stages

Examples

```
data <- PrInDT::data_land # load data
dataclean <- data[,c(1:7,23:24,11:13,22,8:10)]  # only relevant features
inddep <- c(14:16) # dependent variables
dataland <- na.omit(dataclean)
ctestv <- NA
perc <- c(0.45,0.05,0.25)   # percentages of observations of larger class,
# 1 per dependent class variable
perc2 <- c(0.75,0.95,0.75)  # percentages of observations of smaller class,
# 1 per dependent class variable
outland <- SimCPrInDT(dataland,percl=perc,percs=perc2,N=9,inddep=inddep,M=2,psize=10)
outland
plot(outland)
```

SimMixPrInDT *Interdependent estimation for classification-regression mixtures*

Description

The function SimMixPrInDT applies structured subsampling for finding an optimal subsample to model the relationship between the dependent variables specified in the sublist 'targets' of the list 'datalist' and all other factor and numerical variables in the corresponding individual data frame specified in the sublist 'datanames' of the list 'datalist' in the same order as 'targets'. The data frames have to be different of each other!
The function is prepared to handle classification tasks with 2 or more classes and regression tasks. At first stage, the targets are estimated based on the full sample of all exogenous variables and on summaries of the observed endogenous variables.
For generating summaries, the variables representing the substructure have to be specified in the sublist 'datastruc' of 'datalist'.
The sublist 'summ' of 'datalist' includes for each discrete target the classes for which you want to calculate the summary percentages and for each continuous target just NA (in the same order as in the sublist 'targets'). For a discrete target in this list, you can provide a sublist of classes to be combined (for an example see the below example).
At second stage, structured subsampling is used for improving the models from first stage. The substructure of the observations used for structured subsampling is specified by the list 'Struc' which here only consists of the name 'check' of the variable with the information about the categories of the substructure (without specification of the dataset names already specified in 'datanames', see example below), and the matrix 'labs' which specifies the values of 'check' corresponding to two categories in its rows, i.e. in 'labs[1,]' and 'labs[2,]'. The names of the categories have to be specified by `rownames(labs)`.
In structured subsampling, first 'M' repetitions of subsampling of the variable 'name' with 'nsub' different elements of each category in 'check' are realized. If 'nsub' is a list, each entry is employed individually. If 'nsub' is larger than the maximum available number of elements with a certain value of 'check', the maximum possible number of elements is used. Then, for each of the subsamples 'N' repetitions of subsampling in classification or regression with the specified percentages of classes, observations, and predictors are carried out.
These percentages are specified in the matrix 'percent', one row per estimation task. For binary classification tasks, percentages 'percl' and 'percs' for the larger and the smaller class have to be specified. For multilevel classification tasks, NA is specified (see the below example) since the percentages are generated automatically. For regression tasks, 'pobs' and 'ppre' have to be specified for observations and predictors, respectively.
The optimization citerion is balanced accuracy for classification and goodness of fit R2 for regression on the full sample, respectively. At stage 2, the models are optimized individually. At stage 3, the models are optimized on the maximum of joint elements in their substructures.
The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
SimMixPrInDT(datalist,ctestv=NA,Struc,M=12,N=99,nsub,percent=NA,conf.level=0.95,
                                        minsplit=NA,minbucket=NA)
```

Arguments

datalist	list(datanames,targets,datastruc,summ) Input data: For specification see the above description
ctestv	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
Struc	list(name,check,labs) Paprametes for structured subsampling, as explained in the desciption above.
M	Number of repetitions of subsampling of elements of substructure; default = 12
N	Number of repetitions of subsampling for predictors (integer); default = 99
nsub	(List of) numbers of different elements of substructure per subsample
percent	matrix of percent spefications: For specification see the above description; default: 'percent = NA' meaning default values for percentages.
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.6.2, for further information.

Standard output can be produced by means of `print(name)` or just `name` as well as `plot(name` where 'name' is the output data frame of the function.

Value

modelsF Best trees at stage 1 (Full sample)

modelsI Best trees at stage 2 (Individual optimization)

modelsJ Best trees at stage 3 (Joint optimization)

depnames names of dependent variables

nmod number of models of tasks

nlev levels of tasks

accAll accuracies of best trees at both stages

Examples

```
# zero data
datazero <- PrInDT::data_zero
datazero <- na.omit(datazero) # cleaned full data: no NAs
names(datazero)[names(datazero)=="real"] <- "zero"
CHILDzero <- PrInDT::participant_zero
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
##
# multi-level data
datastrat <- PrInDT::data_zero
datamult <- na.omit(datastrat)
# ctestv <- NA
datamult$mult[datamult$ETH %in% c("C1a","C1b","C1c") & datamult$real == "zero"] <- "zero1"
datamult$mult[datamult$ETH %in% c("C2a","C2b","C2c") & datamult$real == "zero"] <- "zero2"
datamult$mult[datamult$real == "realized"] <- "real"
datamult$mult <- as.factor(datamult$mult) # mult is new class variable
datamult$real <- NULL # remove old class variable
CHILDmult <- CHILDzero
##
# vowel data
data <- PrInDT::data_vowel
data <- na.omit(data)
CHILDvowel <- data$Nickname
CHILDvowel <- as.factor(gsub("Nick","P",CHILDvowel))
data$Nickname <- NULL
syllable <- 3 - data$syllables
data$syllabels <- NULL
data$syllables <- syllable
data$speed <- data$word_duration / data$syllables
names(data)[names(data) == "target"] <- "vowel"
datavowel <- data
##
# function preparation and call
# datalist and percent
datanames <- list("datazero","datamult","datavowel")
targets <- c("zero","mult","vowel")
datastruc <- list(CHILDzero,CHILDmult,CHILDvowel)
summult <- paste("zero1","zero2",sep=",")
summ <- c("zero",summult,NA)
datalist <- list(datanames=datanames,targets=targets,datastruc=datastruc,summ=summ)
percent <- matrix(NA,nrow=3,ncol=2)
percent[1,] <- c("percl=0.075","percs=0.9") # percentages for datazero
# no percentages needed for datapast
percent[3,] <- c("pobs=0.9","ppre=c(0.9,0.8)") # percentages for datavowel
# substructures
labs <- matrix(1:6,nrow=2,ncol=3)
```

```
labs[1,] <- c("C1a","C1b","C1c")
labs[2,] <- c("C2a","C2b","C2c")
rownames(labs) <- c("children 1","children 2")
Struc <- list(check="ETH",labs=labs)
outSimMix <- SimMixPrInDT(datalist,ctestv=ctestv,Struc=Struc,M=2,N=9,nsub=c(19,20),
                      percent=percent,conf.level=0.99)
outSimMix
plot(outSimMix)+
```

`SimRPrInDT` *Interdependent estimation for regression*

Description

The function `SimRPrInDT` applies structured subsampling for finding an optimal subsample to model the relationship between the continuous variables with indices 'inddep' and all other factor and numerical variables in the data frame 'datain'.
The substructure of the observations used for subsampling is specified by the list 'Struc' which consists of the factor variable 'name' representing the substructure, the name 'check' of the factor variable with the information about the categories of the substructure, and the matrix 'labs' which specifies the values of 'check' corresponding to two categories in its rows, i.e. in 'labs[1,]' and 'labs[2,]'. The names of the categories have to be specified by `rownames(labs)`.
In structured subsampling, first 'M' repetitions of subsampling of the variable 'name' with 'nsub' different elements of each category in 'check' are realized. If 'nsub' is a list, each entry is employed individually. If 'nsub' is larger than the maximum available number of elements with a certain value of 'check', the maximum possible number of elements is used. Then, for each of the subsamples 'N' repetitions of subsampling of 'ppre' percentages of the predictors are carried out.
Subsampling of observations can additionally be restricted to 'pobs' percentages.
The optimization citerion is the goodness of fit R2 on the full sample. At stage 2, the models are optimized individually. At stage 3, the mean of accuracies is optimized over all models.
Struc=NA causes random subsampling of observations instead of structured subsampling.
The trees generated from undersampling can be restricted by not accepting trees including split results specified in the character strings of the vector 'ctestv'.
The parameters 'conf.level', 'minsplit', and 'minbucket' can be used to control the size of the trees.

Usage

```
SimRPrInDT(data,ctestv=NA,Struc=NA,inddep,N=99,pobs=0.9,ppre=c(0.9,0.7),
                M=1,nsub=1,conf.level=0.95,minsplit=NA,minbucket=NA)
```

Arguments

`data`	Input data frame with continuous target variables with column indices 'inddep' and the influential variables, which need to be factors or numericals (transform logicals and character variables to factors)

ctestv	Vector of character strings of forbidden split results; Example: ctestv <- rbind('variable1 == {value1, value2}','variable2 <= value3'), where character strings specified in 'value1', 'value2' are not allowed as results of a splitting operation in variable 1 in a tree. For restrictions of the type 'variable <= xxx', all split results in a tree are excluded with 'variable <= yyy' and yyy <= xxx. Trees with split results specified in 'ctestv' are not accepted during optimization. A concrete example is: 'ctestv <- rbind('ETH == {C2a, C1a}','AGE <= 20')' for variables 'ETH' and 'AGE' and values 'C2a','C1a', and '20'; If no restrictions exist, the default = NA is used.
Struc	= list(name,check,labs), cf. description for explanations
inddep	Column indices of target variables in datain
N	Number of repetitions of subsampling (integer) of predictors; default = 99
pobs	Percentage(s) of observations for subsampling; default=0.9
ppre	Percentage(s) of predictors for subsampling; default=c(0.9,0.7)
M	Number of repetitions of subsampling of elements of substructure
nsub	(List of) numbers of different elements of substructure per subsample
conf.level	(1 - significance level) in function ctree (numerical, > 0 and <= 1); default = 0.95
minsplit	Minimum number of elements in a node to be splitted; default = 20
minbucket	Minimum number of elements in a node; default = 7

Details

See Buschfeld & Weihs (2025), Optimizing decision trees for the analysis of World Englishes and sociolinguistic data. Cambridge University Press, section 4.5.6.2, for further information.

Standard output can be produced by means of print(name) or just name as well as plot(name where 'name' is the output data frame of the function.

Value

modelsF Best trees at stage 1

modelsI Best trees for the different values of 'nsub' at stage 2

modelsJ Best trees for the different values of 'nsub' after mean optimization

Struc Used structure

msub Best numbers of elements in substructure: 2nd stage, 3rd stage

depnames names of dependent variables

R2All R2s of best trees at stages 1, 2, mean max.

Examples

```
data <- PrInDT::data_vowel
data <- na.omit(data)
CHILDvowel <- data$Nickname
data$Nickname <- NULL
syllable <- 3 - data$syllables
data$syllabels <- NULL
data$syllables <- syllable
data$speed <- data$word_duration / data$syllables
names(data)[names(data) == "target"] <- "vowel_length"
# interpretation restrictions (split exclusions)
ctestv <- rbind('ETH == {C2a, C1a}','MLU == {1, 3}') # split exclusions
# structure definition
name <- CHILDvowel
check <- "data$ETH"
labs <- matrix(1:6,nrow=2,ncol=3)
labs[1,] <- c("C1a","C1b","C1c")
labs[2,] <- c("C2a","C2b","C2c")
rownames(labs) <- c("children 1","children 2")
Struc <- list(name=name,check=check,labs=labs)
# column indices of dependent variables
inddep <- c(13,9)
outSimR <- SimRPrInDT(data,ctestv=ctestv,Struc=Struc,inddep=inddep,N=3,M=2,
                    nsub=c(19,20),conf.level=0.99)
outSimR
plot(outSimR)
```

Index

∗ **datasets**
data_land, 5
data_speaker, 6
data_vowel, 6
data_zero, 7
participant_zero, 15

C2SPrInDT, 3

data_land, 5
data_speaker, 6
data_vowel, 6
data_zero, 7

Mix2SPrInDT, 8

NesPrInDT, 10

OptPrInDT, *13*, 13

participant_zero, 15
PostPrInDT, 15
PrInDT, *10, 11*, *13*, 16, *20*, *22*, *27*, *30–33*, *35*, *37*, *43*
PrInDTAll, 20
PrInDTAllparts, 21
PrInDTCstruc, 23
PrInDTMulab, 27
PrInDTMulabAll, 29
PrInDTMulev, 31
PrInDTMulevAll, 33
PrInDTreg, *34*, 34
PrInDTregAll, 36
PrInDTRstruc, 37, *38*

R2SPrInDT, 41
RePrInDT, *43*, 43

SimCPrInDT, 45
SimMixPrInDT, 47
SimRPrInDT, *50*, 50