Package 'MEml'

January 13, 2020

Title Mixed Effect Machine Learning

Version 0.0.1

Description Machine learning methods for longitudinal and clustered data based on standard generalized linear mixed effect model, regression trees, random forest, SVM, and generalized boosted machines.

Depends R (>= 3.1.2)

Imports PresenceAbsence, caret, rpart, lme4, rpart.plot, partykit,RRF,flexmix, Matrix, vcd, grid, plyr, inTrees, gbm, DMwR, kernlab

License GPL(>= 3)

LazyData true

NeedsCompilation no

RoxygenNote 7.0.2

Author Che Ngufor [aut, cre]

Maintainer Che Ngufor <Ngufor.Che@mayo.edu>

R topics documented:

`collect.garbage` *collect gabbage*

Description

Collects garbage until the memory is clean

Usage

```
collect.garbage()
```

`denormalize` *denomalize data*

Description

tranform normalized data back to original scale

Usage

```
denormalize(normalized, min, max)
```

Arguments

`normalized`	the normalized data frame/matrix
`min, max`	the min and max of each variable in the data. This can be otained by retrieving the attribute values of the normalized data, i.e. output of `normalize`

`glmerBoost` *Boosted generalized mixed-effect regression models (glmer).*

Description

Boosted generalized mixed-effect regression models (glmer).

Usage

```
glmerBoost(
  form,
  dat,
  lme.family = binomial,
  max.iter = 100,
  verbose = FALSE,
  coeflearn = "Breiman",
  ...
)

glmerBoost(
  form,
  dat,
  lme.family = binomial,
  max.iter = 100,
  verbose = FALSE,
  coeflearn = "Breiman",
  ...
)

glmerLogitBoost(form, dat, max.iter = 100, verbose = FALSE, ...)

glmerLogitBoost(form, dat, max.iter = 100, verbose = FALSE, ...)
```

Arguments

`form`	A formula describing fixed and random effects (as in lme4)
`dat`	Matrix/data frame
`lme.family`	family for glmer
`max.iter`	maximum iteration
`verbose`	print iteration outputs ?
`coeflearn`	Adaboot.M1 learning method
`...`	other arguments

Value

a list with items

`tree.fit`	fitted lme4 models
`fitted.probs`	fitted probabilites for final model

`fitted.class` fitted class labels for final model

`train.perf` various performance measures for final model on training set

`threshold` classification cut-off

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

Examples

```
## Not run:
set.seed(12345)
mod <- glmerBoost(form, data))

## End(Not run)
## Not run:
set.seed(12345)
mod <- glmerBoost(form, rand.form, data))

## End(Not run)
```

---

heart.valve *Aortic valve replacement surgery data from the* joineR *package*

Description

This is longitudinal data on an observational study on detecting effects of different heart valves, differing on type of tissue, implanted in the aortic position. The data consists of longitudinal measurements (three cardiac functions) from patients who underwent aortic valve replacement from 1991 to 2001 at the Royal Brompton Hospital, London, United Kingdom. The data was first reported in [1] where the authors used all patients during the 10 years period with at least a year of follow up with serial echocardiographic measurements and applied a linear mixed-effect model to predict left ventricular mass index (LVMI). Similarly, the data was used in [2] to predict longitudinal profile of LVMI categorized as high or normal using several patient baseline characteristics and laboratory variables. LVMI is considered increased if LVMI >134 g/m 2 in male patients and LVMI >110 g/m 2 in female patients, thus values in this range for both sex was considered as the positive class in MEml.

Usage

```
data(heart.valve)
```

Format

This is a data frame in the unbalanced format, that is, with one row per observation. The data consists in columns for patient identification, time of measurements, longitudinal multiple longitudinal measurements, baseline covariates, and survival data. The column names are identified as follows:

`num` number for patient identification.

`sex` gender of patient (`0` = Male and `1` = Female).

`age` age of patient at day of surgery (years).

`time` observed time point, with surgery date as the time origin (years).

`fuyrs` maximum follow up time, with surgery date as the time origin (years).

`status` censoring indicator (`1` = died and `0` = lost at follow up).

`grad` valve gradient at follow-up visit.

`log.grad` natural log transformation of `grad`.

`lvmi` left ventricular mass index (standardised) at follow-up visit.

`log.lvmi` natural log transformation of `lvmi`.

`ef` ejection fraction at follow-up visit.

`bsa` preoperative body surface area.

`lvh` preoperative left ventricular hypertrophy.

`prenyha` preoperative New York Heart Association (NYHA) classification (`1` = I/II and `3` = III/IV).

`redo` previous cardiac surgery.

`size` size of the valve (millimeters).

`con.cabg` concomitant coronary artery bypass graft.

`creat` preoperative serum creatinine (μmol/mL).

`dm` preoperative diabetes.

`acei` preoperative use of ace inhibitor.

`lv` preoperative left ventricular ejection fraction (LVEF) (`1` = good, `2` = moderate, and `3` = poor).

`emergenc` operative urgency (`0` = elective, `1` = urgent, and `3` = emergency).

`hc` preoperative high cholesterol (`0` = absent, `1` = present treated, and `2` = present untreated).

`sten.reg.mix` aortic valve haemodynamics (`1` = stenosis, `2` = regurgitation, `3` = mixed).

`hs` implanted aortic prosthesis type (`1` = homograft and `0` = stentless porcine tissue).

Source

Mr Eric Lim (`http://www.drericlim.com`)

References

Lim E, Ali A, Theodorou P, Sousa I, Ashrafian H, Chamageorgakis T, Duncan M, Diggle P, Pepper J. A longitudinal study of the profile and predictors of left ventricular mass regression after stentless aortic valve replacement. *Ann Thorac Surg.* 2008; **85(6)**: 2026-2029. Che Ngufor, Holly Van Houten, Brian S. Caffo , Nilay D. Shah, Rozalina G. McCoy Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c, in Journal of Biomedical Informatics, 2018

See Also

`mental`, `liver`, `epileptic`, `aids`.

LongiLagSplit *Train-Test longitudinal data split*

Description

Split longitudinal data into training and test set, where the training set lags the test set by an amount "lag". e.g predict an outcome at lag=x hospital visits in advanced. There must be at least 2 repeated measurements for each unit

Usage

```
LongiLagSplit(dat, id, rhs.vars, resp.vars, order.vars = "time", lag = 1)
```

MECTree *Mixed Effect Conditional Inference Trees*

Description

Trains a Mixed Effect Model for conditional inference trees

Usage

```
MECTree(
  X,
  Y,
  con.tree = FALSE,
  rhs.vars,
  rand.vars = "1",
  groups = NULL,
  tol = 1e-05,
  max.iter = 100,
  verbose = FALSE,
  likelihoodCheck = TRUE,
  glmer.Control = glmerControl(optimizer = "bobyqa"),
  nAGQ = 0,
  cv = TRUE,
  cpmin = 1e-04,
  minsplit = 50,
  minbucket = 10,
  no.SE = 1,
  mincriterion = 0.975,
  maxdepth = 30,
  stump = FALSE,
  ...
)
```

Arguments

X	data.frame with predictors
Y	binary response vector
con.tree	do conditional inference trees instead of rpart?
rand.vars	random effect variables
groups	character name of the column containing the group identifier
tol	convergence tolerance
max.iter	maximum number of iteration for the EM algorithm
verbose	logical for printing intermediate trees results
likelihoodCheck	logical: should the likelihood of random effect model be used to check for convergence?
glmer.Control	same as glmerControl - glmer controls, default to glmerControl(optimizer = "bobyqa")]
nAGQ	as in glmer, default to 10 with maximum likelihood or restricted maximum likelihood
cv	[TRUE] - Should cross-validation be used?
cpmin	[0.0MEBoostedTree001] - complexity parameter used in building a tree before cross-validation
no.SE	[0] - number of standard errors used in pruning (0 if unused)
...	Further arguments passed to or from other methods.
part.vars, rhs.vars	partitioning variables for MOB and predictors
reg.vars	regressors for MOB
initialRandomEffects	[0] a vector of initial values for random effects
initialProbs	[0.5] a vector of initial conditional probabilites of success
include.RE	include random effects in glmtree model part?
minsize, maxdepth, minsplit, minbucket, mincriterion, stump	see rpart
alpha	stability parameter in glmtree

Value

An object of class MECTree; a list with items

tree.fit	fitted classification trees model
glmer.fit	fitted mixed effect logistic regression model
logLik	log likelihood of mixed effect logistic regression
random.effects	random effect parameter estimates
form	modified formula for fitted classification trees model
rand.form	modified formula for random effects

glmer.Control	glmer controls
tree.control	rpart controls
glmer.CI	estimates of mixed effect logistic regression with approximate confidence intervals on the logit scale. More accurate values can be obtained by bootstrap
fitted.probs	fitted probabilites for final model
fitted.class	fitted class labels for final model
train.perf	various performance measures for final model on training set
threshold	classification cut-off

MEgbmRules *Mixed Effect GBM*

Description

Trains a Mixed Effect gradient boosted machine for longitudinal continuous, binary and count data. A rule based version or these methods using the inTree package is also implemented(see [1])

Usage

```
MEgbmRules(
  form,
  dat,
  groups = NULL,
  rand.vars = "1",
  para = NULL,
  tol = 1e-05,
  max.iter = 100,
  include.RE = FALSE,
  verbose = FALSE,
  maxdepth = 5,
  glmer.Control = glmerControl(optimizer = "bobyqa"),
  nAGQ = 0,
  likelihoodCheck = TRUE,
  K = 3,
  decay = 0.05,
  ...
)
```

Arguments

form	formula
dat	data.frame with predictors
groups	character name of the column containing the group identifier
rand.vars	random effect variables
para	named list of gbm training parameters
tol	convergence tolerance
max.iter	maximum number of iterations

include.RE	(logical) to include random effect Zb as predictor in gbm?
verbose	verbose for lme4
glmer.Control	glmer control
likelihoodCheck	(logical) to use log likelihood of glmer to check for convergence?
...	Further arguments passed to or from other methods.
type	of predictions of gbm to pass to lme4 as population estimates (these will be used as offset)

Value

An object of class MEgbm; a list with items

gbmfit	fitted gbm model
glmer.fit	fitted mixed effect logistic regression model
logLik	log likelihood of mixed effect logistic regression
random.effects	random effect parameter estimates
glmer.form	lmer4 formula
glmer.CI	estimates of mixed effect logistic regression with approximate confidence intervals on the logit scale. More accurate values can be obtained by bootstrap
fitted.probs	fitted probabilites for final model
fitted.class	fitted class labels for final model
train.perf	various performance measures for final model on training set
threshold	classification cut-off
predRules	fitted rules
Y.star	fitted transform outcome

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

References

Che Ngufor, Holly Van Houten, Brian S. Caffo , Nilay D. Shah, Rozalina G. McCoy Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c, in Journal of Biomedical Informatics, 2018

MEglm *Mixed Effect GLM*

Description

Trains a Mixed Effect generalized mixed effect models. This function just calls glmer from the Lme4 package.

Usage

```
MEglm(form, data, control, nAGQ = 0)
```

Arguments

form	formula
data	data.frame with predictors
nAGQ	see glmer
Control	glmer control

MEglmTree *Mixed Effect Model Based Trees*

Description

Trains a Mixed Effect Model Based Recursive partitioning and Mixed Effect classification trees for longitudinal continuous, binary and count data. These functions can be called directly and more efficiently by the functions MEml and MEml.lag

Usage

```
MEglmTree(
  X,
  Y,
  part.vars,
  reg.vars = "1",
  rand.vars = "1",
  groups = NULL,
  initialRandomEffects = rep(0, nrow(X)),
  initialProbs = rep(0.5, nrow(X)),
  tol = 1e-05,
  max.iter = 100,
  include.RE = TRUE,
  verbose = FALSE,
  likelihoodCheck = TRUE,
  glmer.Control = glmerControl(optimizer = "bobyqa"),
  nAGQ = 0,
  alpha = 0.05,
  minsize = 20,
```

```
  maxdepth = 50,
  para = NULL,
  ...
)
```

Arguments

X	data.frame with predictors
Y	binary response vector
part.vars, rhs.vars	partitioning variables for MOB and predictors
reg.vars	regressors for MOB
rand.vars	random effect variables
groups	character name of the column containing the group identifier
initialRandomEffects	[0] a vector of initial values for random effects
initialProbs	[0.5] a vector of initial conditional probabilites of success
tol	convergence tolerance
max.iter	maximum number of iteration for the EM algorithm
include.RE	include random effects in glmtree model part?
verbose	logical for printing intermediate trees results
likelihoodCheck	logical: should the likelihood of random effect model be used to check for convergence?
glmer.Control	same as glmerControl - glmer controls, default to glmerControl(optimizer = "bobyqa")]
nAGQ	as in glmer, default to 10 with maximum likelihood or restricted maximum likelihood
alpha	stability parameter in glmtree
minsize, maxdepth, minsplit, minbucket, mincriterion, stump	see rpart
...	Further arguments passed to or from other methods.
con.tree	do conditional inference trees instead of rpart?
cv	[TRUE] - Should cross-validation be used?
cpmin	[0.0MEBoostedTree001] - complexity parameter used in building a tree before cross-validation
no.SE	[0] - number of standard errors used in pruning (0 if unused)

Value

An object of class MECTree; a list with items

tree.fit	fitted classification trees model
glmer.fit	fitted mixed effect logistic regression model
logLik	log likelihood of mixed effect logistic regression

`random.effects`
: random effect parameter estimates

`form`
: modified formula for fitted classification trees model

`rand.form`
: modified formula for random effects

`glmer.Control`
: glmer controls

`tree.control`
: rpart controls

`glmer.CI`
: estimates of mixed effect logistic regression with approximate confidence intervals on the logit scale. More accurate values can be obtained by bootstrap

`fitted.probs`
: fitted probabilites for final model

`fitted.class`
: fitted class labels for final model

`train.perf`
: various performance measures for final model on training set

`threshold`
: classification cut-off

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

`MEmixgbm` *Mixture of Mixed Effect GBM*

Description

Trains a Mixed Effect gradient boosted machine where the random effects are assumed to follow a mixture of gaussian distribution.

Usage

```
MEmixgbm(
  form,
  dat,
  groups = NULL,
  rand.vars = "1",
  para = NULL,
  tol = 1e-05,
  max.iter = 100,
  include.RE = FALSE,
  verbose = FALSE,
  maxdepth = 5,
  glmer.Control = glmerControl(optimizer = "bobyqa"),
  nAGQ = 0,
  likelihoodCheck = TRUE,
  K = 3,
  krange = 2:5,
  decay = 0.05,
  ...
)
```

Arguments

groups	character name of the column containing the group identifier
rand.vars	random effect variables
para	named list of gbm training parameters
tol	convergence tolerance
max.iter	maximum number of iterations
include.RE	(logical) to include random effect Zb as predictor in gbm?
verbose	verbose for lme4
likelihoodCheck	(logical) to use log likelihood of glmer to check for convergence?
...	Further arguments passed to or from other methods.
X	data.frame with predictors
Y	binary response vector
gbm.dist	gbm loss function
lme.family	glmer control
type	of predictions of gbm to pass to lme4 as population estimates (these will be used as offset)

Value

An object of class MEgbm; a list with items

gbmfit	fitted gbm model
glmer.fit	fitted mixed effect logistic regression model
logLik	log likelihood of mixed effect logistic regression
random.effects	random effect parameter estimates
boost.form	gbm formula for fitted model
glmer.form	lmer4 formula
glmer.CI	estimates of mixed effect logistic regression with approximate confidence intervals on the logit scale. More accurate values can be obtained by bootstrap
fitted.probs	fitted probabilites for final model
fitted.class	fitted class labels for final model
train.perf	various performance measures for final model on training set
threshold	classification cut-off

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

References

Che Ngufor, Holly Van Houten, Brian S. Caffo , Nilay D. Shah, Rozalina G. McCoy Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c, in Journal of Biomedical Informatics, 2018

MEmixgbm2 *Mixture of Mixed Effect GBM (version 2)*

Description

Trains a Mixed Effect gradient boosted machine where the random effects are assumed to follow a mixture of gaussian distribution (I am not sure what the difference between MEmixgbm and MEmixgbm2 was!).

Usage

```
MEmixgbm2(
  form,
  dat,
  groups = NULL,
  rand.vars = "1",
  para = NULL,
  tol = 1e-05,
  max.iter = 100,
  include.RE = FALSE,
  verbose = FALSE,
  maxdepth = 5,
  glmer.Control = glmerControl(optimizer = "bobyqa", check.nobs.vs.nRE = "ignore
    check.nobs.vs.nlev = "ignore"),
  nAGQ = 0,
  likelihoodCheck = TRUE,
  K = 3,
  krange = 2:5,
  decay = 0.05,
  ...
)
```

MEml *Calls the MEml models: MEgbm, MEgbmrules, MErfrules, MEglmtree, MECTree, etc. The training and test data can be split into lagged training and testing as described in [1]*

Description

Calls the MEml models: MEgbm, MEgbmrules, MErfrules, MEglmtree, MECTree, etc. The training and test data can be split into lagged training and testing as described in [1]

Usage

```
MEml_lag(
  lag = NULL,
  classifier,
  dat,
  id,
```

```
  rhs.vars,
  resp.vars,
  order.vars,
  rand.vars = NULL,
  reg.vars = NULL,
  part.vars = NULL,
  para,
  max.iter = 10,
  seed = 1,
  return.model = TRUE
)

MEml(
  classifier,
  dat.trn,
  dat.tst,
  id,
  rhs.vars,
  resp.vars,
  rand.vars = NULL,
  reg.vars = NULL,
  part.vars = NULL,
  para,
  max.iter = 10,
  seed = 1,
  return.model = FALSE,
  ...
)

MEml2(classifier, data, id, resp.vars, rhs.vars, rand.vars = NULL, para, ...)
```

Arguments

`lag`	time lag between predictors and outcome: e.g if lag = 1, then we use predictors in current vist to predict outcome in the next visit.
`classifier`	character or character vector with names of classification models. See names(`Train.Test`()) for list of possible models.
`dat`	data frame with predictors and outcome
`id`	character name of the column containing the group identifier
`rhs.vars`	caracter vector of predictors
`order.vars`	order variables (usually time variable)
`rand.vars`	random effect variables
`reg.vars`	reg.vars regressors for MOB
`part.vars`	partitioning variables for MOB and predictors
`para`	named list of gbm training parameters
`max.iter`	maximum number of iterations
`return.model`	should the train model be return. Otherwise the return values is only the performance metrics

Details

1. `MEml_lag` Takes the full data set and calls `LongiLagSplit` to split data into lagged training and testing. `MEml_lag` also trains the MOB and CTree models (see [1]).
2. `MEml` is the same as `MEml_lag`, except that you pass in the training and test set. So you can call `LongiLagSlit` and pass the derived training and test sets to `MEml`.
3. `MEml2` is the same as `MEml`, except that you don't pass in the test set. Also, it is currently implemented only for the GLMER, MEgbm and MErf models.

Value

The train MEml model and performance matrics (as data frame) if return.model = TRUE

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

References

Che Ngufor, Holly Van Houten, Brian S. Caffo , Nilay D. Shah, Rozalina G. McCoy Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c, in Journal of Biomedical Informatics, 2018 #

Examples

```
## Not run:
# parameter list
para <- list(
method = "cv", # internal cross-validation method for parameter tuning. See caret package
tuneLength=3, # grid size for parameter search
number = 3, # number of internal cross-validation
n.trees=100, # number of trees in gbm
ntree = 50, # number of trees in random forest
mtry = 5, # mtry in random forest
interaction.depth=4,
shrinkage=0.01,
n.minobsinnode=10,
opt.para= TRUE, # perform parameter tuning through internal cross-validation
coefReg = 0.5,
coefImp=1,
include.RE = FALSE,
con.tree = FALSE,
max.iter = 10, alpha=0.05, minsize=20,maxdepth=30,
K = 3, decay = 0.05, tol= 1e-5,
seed = 1 # random seed
 )
data(heart.valve)
dat <- heart.valve
dat$id <- as.numeric(dat$id)  ## random effect grouping variable
resp.vars <- "inc.lvmi"
id <- "id"
## fixed effect variables
rhs.vars <- c("sex", "age", "time", "fuyrs", "grad", "log.grad", "bsa", "lvh", "prenyha",
              "redo", "size", "con.cabg", "creat", "dm", "acei", "lv", "emergenc",
              "hc", "sten.reg.mix", "hs")
order.vars = "time"
```

```
rand.vars= "time"  ## random effect variables
### split data into lagged training and testing
### predict two time points in advanced
dd <- LongiLagSplit(dat=dat, id=id, rhs.vars=rhs.vars,resp.vars=resp.vars,
                    order.vars=order.vars,lag= 2)
train <- dd$train
test <- dd$test
res <- MEml(classifier="GBM", dat.trn = train, dat.tst=test, id=id,
            rhs.vars=dd$rhs.vars, resp.vars=dd$resp.vars,
            rand.vars=rand.vars, para=para, max.iter = 10, seed = 1,
            return.model = FALSE)
res$GBM

## End(Not run)
```

`MErfRules` *Mixed Effect random forest*

Description

Trains a Mixed Effect random forest for longitudinal continuous, binary and count data. A rule based version or these methods using the `inTree` package is also implemented(see [1])

Usage

```
MErfRules(
  form,
  dat,
  groups = NULL,
  rand.vars = "1",
  para = NULL,
  tol = 1e-05,
  max.iter = 100,
  include.RE = FALSE,
  verbose = FALSE,
  maxdepth = 5,
  glmer.Control = glmerControl(optimizer = "bobyqa"),
  nAGQ = 0,
  likelihoodCheck = TRUE,
  K = 3,
  decay = 0.05,
  ...
)
```

Arguments

`form`	formula
`dat`	data.frame with predictors
`groups`	character name of the column containing the group identifier
`rand.vars`	random effect variables
`para`	named list of gbm training parameters

tol	convergence tolerance
max.iter	maximum number of iterations
include.RE	(logical) to include random effect Zb as predictor in gbm?
verbose	verbose for lme4
glmer.Control	glmer control
likelihoodCheck	(logical) to use log likelihood of glmer to check for convergence?
...	Further arguments passed to or from other methods.
type	of predictions of gbm to pass to lme4 as population estimates (these will be used as offset)

Value

An object of class MEgbm; a list with items

rf.fit	fitted random forest model
glmer.fit	fitted mixed effect logistic regression model
logLik	log likelihood of mixed effect logistic regression
random.effects	random effect parameter estimates
glmer.form	lmer4 formula
glmer.CI	estimates of mixed effect logistic regression with approximate confidence intervals on the logit scale. More accurate values can be obtained by bootstrap
fitted.probs	fitted probabilites for final model
fitted.class	fitted class labels for final model
train.perf	various performance measures for final model on training set
threshold	classification cut-off
predRules	fitted rules
Y.star	fitted transform outcome

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

References

Che Ngufor, Holly Van Houten, Brian S. Caffo , Nilay D. Shah, Rozalina G. McCoy Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c, in Journal of Biomedical Informatics, 2018

`MEsvm` *Mixed Effect support vector machine*

Description

Train a Mixed Effect support vector machine for binary outcome.

Usage

```
MEsvm(
  form,
  dat,
  groups = NULL,
  rand.vars = "1",
  para = NULL,
  tol = 1e-05,
  max.iter = 100,
  include.RE = FALSE,
  verbose = FALSE,
  maxdepth = 5,
  glmer.Control = glmerControl(optimizer = "bobyqa"),
  nAGQ = 0,
  likelihoodCheck = TRUE,
  K = 3,
  decay = 0.05,
  ...
)
```

Arguments

`form`	formula
`dat`	data.frame with predictors
`groups`	character name of the column containing the group identifier
`rand.vars`	random effect variables
`para`	named list of gbm training parameters
`tol`	convergence tolerance
`max.iter`	maximum number of iterations
`include.RE`	(logical) to include random effect Zb as predictor in gbm?
`verbose`	verbose for lme4
`likelihoodCheck`	(logical) to use log likelihood of glmer to check for convergence?
`...`	Further arguments passed to or from other methods.
`lme.family`	glmer control
`type`	of predictions of gbm to pass to lme4 as population estimates (these will be used as offset)

Value

An object of class MEgbm; a list with items

`svmfit`	fitted svm model
`glmer.fit`	fitted mixed effect logistic regression model
`logLik`	log likelihood of mixed effect logistic regression
`random.effects`	random effect parameter estimates
`svm.form`	svm formula for fitted model
`glmer.form`	lmer4 formula
`glmer.CI`	estimates of mixed effect logistic regression with approximate confidence intervals on the logit scale. More accurate values can be obtained by bootstrap
`fitted.probs`	fitted probabilites for final model
`fitted.class`	fitted class labels for final model
`fitted.decision`	fitted decision values for final model
`train.perf`	various performance measures for final model on training set
`threshold`	classification cut-off

Author(s)

Che Ngufor <Ngufor.Che@mayo.edu>

References

Che Ngufor, Holly Van Houten, Brian S. Caffo , Nilay D. Shah, Rozalina G. McCoy Mixed Effect Machine Learning: a framework for predicting longitudinal change in hemoglobin A1c, in Journal of Biomedical Informatics, 2018

`normalize` *normalize data*

Description

Normalize matrix or data frame to the min-max range of the variables.

Usage

```
normalize(x)
```

Arguments

`x`	matrix or data frame

Value

nomalized matrix/data frame with min and max of each variable as attributes

opt.thresh *Optimal threshold*

Description

Compute the the optimal classification threshold

Usage

```
opt.thresh(pred, obs)
```

Arguments

obs	ground truth (correct) 0-1 labels vector
prob	Predicted probabilities by a classifier
opt.method	optima classifcation threshold method see package PresenceAbsence. Default is the minRoc distance: i.e the threshold value at the minimum distance between the ROC curve and the to left hand corner (0,1)

Value

threhold

Examples

```
data(cars)
logreg <- glm(formula = vs ~ hp + wt,
              family = binomial(link = "logit"), data = mtcars)
prob <- logreg$fitted.values
opt.thresh(prob = pob, obs = mtcars$vs)
```

Performance.measures *Performance measures*

Description

Compute accuracy, AUC, sensitivity, specificity, positive predictive values, etc.

Usage

```
Performance.measures(pred, obs, threshold = NULL, prevalence = NULL)
```

Arguments

pred	predicted probabilities
obs	true class
threshold	class probability threshold (default to NULL)
prevalence	class prevalence in the population (default to NULL)

`Plot.MEglmTree` *Plot method for MEglmTree object Take a MEglmTree and plot the barplot of fitted classes or boxplot of fitted probability*

Description

Plot method for MEglmTree object Take a MEglmTree and plot the barplot of fitted classes or boxplot of fitted probability

Usage

```
## S3 method for class 'MEglmTree'
plot(
  x,
  main = NULL,
  terminal_panel = node_terminal,
  tp_args = list(),
  inner_panel = node_inner,
  ip_args = list(),
  edge_panel = edge_simple,
  ep_args = list(),
  drop_terminal = FALSE,
  tnex = 1,
  newpage = TRUE,
  pop = TRUE,
  gp = gpar(),
  ...
)

## S3 method for class 'MECTree'
plot(x, ...)

## S3 method for class 'MECTree'
print(x, ...)

heat.tree(tree, low.is.green = FALSE, ...)

.plot_node(
  node,
  xlim,
  ylim,
  nx,
  ny,
  terminal_panel,
  inner_panel,
  edge_panel,
  tnex = 2,
  drop_terminal = TRUE,
  debug = FALSE
)
```

```
NodeBoxplot(
  obj,
  fitted = TRUE,
  col = "black",
  fill = "lightgray",
  width = 0.5,
  yscale = NULL,
  ylines = 3,
  cex = 0.5,
  id = TRUE,
  mainlab = NULL,
  gp = gpar(),
  ...
)

NodeBarplot(
  obj,
  fitted = TRUE,
  col = "black",
  fill = NULL,
  beside = NULL,
  ymax = NULL,
  ylines = NULL,
  widths = 1,
  gap = NULL,
  reverse = NULL,
  id = TRUE,
  mainlab = NULL,
  gp = gpar(),
  ...
)
```

Arguments

`x`	fited MEglmTree or MECtree object.
`main, ep_args`	see partykid
`terminal_panel`	see partykit
`tp_args, ip_args, xlim, ylim`	see partykit
`inner_panel`	see partykit
`edge_panel`	see partykit
`drop_terminal`	see partykit
`tnex`	see partykit
`newpage, pop, gp, tree, low.is.green`	see partykit
`...`	Further arguments passed to partykit
`node`	see partykit

nx, ny	see partykit
debug	see partykit

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

`predict.glmerBoost` *Make predictions using a fitted MECtree model*

Description

Make predictions using a fitted MECtree model

Usage

```
predict.glmerBoost(object, newdata, type = c("prob", "class")[1], ...)

predict.glmerLogitBoost(object, newdata, type = c("prob", "class")[1], ...)
```

Arguments

object	Fitted model from MECtree.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
...	Further arguments passed to or from other methods.

Value

A list with items

prob	predicted class probabilities
class	predicted class memberships obtained by thresholding class probabilities at the prevalence rate of the positive class

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

predict.MECTree *predict MECtree*

Description

Make predictions for mixed effect conditional inference trees.

Usage

```
predict.MECTree(object, newdata, type = c("prob", "class")[1], ...)
```

Arguments

object	Fitted model from MECtree.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
...	Further arguments passed to or from other methods.

Value

A list with items

prob	predicted class probabilities
class	predicted class memberships obtained by thresholding class probabilities at the prevalence rate of the positive class

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

predict.MEgbmRules *Predict MEgbm Make predictions for Mixed Effect GBM.*

Description

Predict MEgbm Make predictions for Mixed Effect GBM.

Usage

```
predict.MEgbmRules(
  object,
  newdata,
  type = c("prob", "class")[1],
  allow.new.levels = FALSE,
  ...
)
```

Arguments

object	Fitted model.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
allow.new.levels	specifry if new levels of factor variables in the test set should be allowed. Default to FALSE.

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

`predict.MEglm` *Predict MEglm Make predictions for Mixed Effect logistic regression. This is just a wrapper for* `predict.merMod` *in lme4 .*

Description

Predict MEglm Make predictions for Mixed Effect logistic regression. This is just a wrapper for `predict.merMod` in lme4 .

Usage

```
predict.MEglm(
  object,
  newdata,
  type = c("prob", "class")[1],
  allow.new.levels = FALSE,
  ...
)
```

Arguments

object	Fitted model.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
allow.new.levels	specifry if new levels of factor variables in the test set should be allowed. Default to FALSE.

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

predict.MEglmTree *predict MEglmtree*

Description

Make predictions for mixed effect model based trees.

Usage

```
predict.MEglmTree(object, newdata, type = c("prob", "class")[1], ...)
```

Arguments

object	Fitted model from MEglmtree.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
...	Further arguments passed to or from other methods.

Value

A list with items

prob	predicted class probabilities
class	predicted class memberships obtained by thresholding class probabilities at the prevalence rate of the positive class

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

predict.MEmixgbm *predict MEmixgbm*

Description

Make predictions for mixed effect mixture of GBM.

Usage

```
predict.MEmixgbm(object, newdata, type = c("prob", "class")[1], ...)
```

Arguments

object	Fitted model from MEmixgbm.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
...	Further arguments passed to or from other methods.

Value

A list with items

prob	predicted class probabilities
class	predicted class memberships obtained by thresholding class probabilities at the prevalence rate of the positive class

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

---

predict.MEmixgbm2 *predict MEmixgbm2*

Description

Make predictions for mixed effect mixture of GBM (version 2).

Usage

```
predict.MEmixgbm2(object, newdata, type = c("prob", "class")[1], ...)
```

Arguments

object	Fitted model from MEmixgbm2.
newdata	A new input data frame.
type	of prediction: "prop" for probabilities and "class" for class labels.
...	Further arguments passed to or from other methods.

Value

A list with items

prob	predicted class probabilities
class	predicted class memberships obtained by thresholding class probabilities at the prevalence rate of the positive class

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

`predict.MErfRules` *Predict MErf Make predictions for Mixed Effect random forest.*

Description

Predict MErf Make predictions for Mixed Effect random forest.

Usage

```
predict.MErfRules(
  object,
  newdata,
  type = c("prob", "class")[1],
  allow.new.levels = FALSE,
  ...
)
```

Arguments

`object`	Fitted model.
`newdata`	A new input data frame.
`type`	of prediction: "prop" for probabilities and "class" for class labels.
`allow.new.levels`	specifry if new levels of factor variables in the test set should be allowed. Default to FALSE.

Author(s)

Che Ngufor Ngufor.Che@mayo.edu

Index