

Package ‘caretEnsemble’

January 27, 2015

Type Package

Title Ensembles of Caret Models

Version 1.0.0

Date 2015-01-14

URL <https://github.com/zachmayer/caretEnsemble>

BugReports <https://github.com/zachmayer/caretEnsemble/issues>

Description Functions for creating ensembles of caret models: caretList, caretEnsemble, and caretStack. caretList is a convenience function for fitting multiple caret::train models to the same dataset. caretEnsemble will make a linear combination of these models using greedy forward selection, and caretStack will make linear or non-linear combinations of these models, using a caret::train model as a meta-model.

Depends R (>= 3.1.0), caret

Suggests testthat, randomForest, rpart, kernlab, nnet, e1071, ipred, pROC, knitr, mlbench, MASS, gbm, klaR

Imports caTools, pbapply, ggplot2, digest, plyr, grid, lattice, gridExtra

License MIT + file LICENSE

VignetteBuilder knitr

Author Zachary A. Mayer [aut, cre],
Jared E. Knowles [aut]

Maintainer Zachary A. Mayer <zach.mayer@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-16 22:19:29

R topics documented:

autoplot.caretEnsemble	3
caretEnsemble	4

caretList	5
caretModelSpec	6
caretStack	6
check_bestpreds_indexes	7
check_bestpreds_obs	8
check_bestpreds_preds	8
check_bestpreds_resamples	8
check_caretList_classes	9
check_caretList_model_types	9
dotplot.caretStack	9
extractBestPreds	10
extractCaretTarget	10
extractCaretTarget.default	11
extractCaretTarget.formula	11
extractModelTypes	12
extractModFrame	12
extractModRes	13
fortify.caretEnsemble	13
getMetric	14
getMetricSD	15
greedOptAUC	15
greedOptRMSE	16
makePredObsMatrix	17
methodCheck	17
multiResiduals	17
plot.caretEnsemble	18
plot.caretStack	19
predict.caretEnsemble	19
predict.caretList	20
predict.caretStack	21
print.caretStack	22
residuals.caretEnsemble	22
safeOptAUC	23
summary.caretEnsemble	24
summary.caretStack	24
trControlCheck	25
tuneCheck	25
varImp.caretEnsemble	26
wtd.sd	26

`autoplot.caretEnsemble`

*Convenience function for more in-depth diagnostic plots of
caretEnsemble objects*

Description

This function provides a more robust series of diagnostic plots for a `caretEnsemble` object.

Usage

```
## S3 method for class 'caretEnsemble'  
autoplot(object, which = c(1:6), mfrow = c(3, 2),  
         xvars = NULL, ...)
```

Arguments

<code>object</code>	a <code>caretEnsemble</code> object
<code>which</code>	an integer index for which of the plots to print
<code>mfrow</code>	an integer vector of length 2 specifying the number of rows and columns for plots
<code>xvars</code>	a vector of the names of x variables to plot against residuals
<code>...</code>	additional arguments to pass to <code>autoplot</code>

Value

A grid of diagnostic plots. Top left is the range of the performance metric across each component model along with its standard deviation. Top right is the residuals from the ensembled model plotted against fitted values. Middle left is a bar graph of the weights of the component models. Middle right is the disagreement in the residuals of the component models (unweighted) across the fitted values. Bottom left and bottom right are the plots of the residuals against two random or user specified variables.

Examples

```
## Not run:  
set.seed(42)  
models <- caretList(  
  iris[1:50,1:2],  
  iris[1:50,3],  
  trControl=trainControl(method='cv'),  
  methodList=c('glm', 'rpart'))  
ens <- caretEnsemble(models)  
autoplot(ens)  
  
## End(Not run)
```

`caretEnsemble`*Combine several predictive models via weights*

Description

Find a good linear combination of several classification or regression models, using either linear regression, elastic net regression, or greedy optimization.

Usage

```
caretEnsemble(all.models, optFUN = NULL, ...)
```

Arguments

<code>all.models</code>	an object of class <code>caretList</code>
<code>optFUN</code>	the optimization function to use
<code>...</code>	additional arguments to pass to the optimization function

Details

Every model in the "library" must be a separate `train` object. For example, if you wish to combine a random forests with several different values of `mtry`, you must build a model for each value of `mtry`. If you use several values of `mtry` in one `train` model, (e.g. `tuneGrid = expand.grid(.mtry=2:5)`), `caret` will select the best value of `mtry` before we get a chance to include it in the ensemble. By default, RMSE is used to ensemble regression models, and AUC is used to ensemble Classification models. This function does not currently support multi-class problems

Value

a `caretEnsemble` object

Note

Currently when missing values are present in the training data, weights are calculated using only observations which are complete across all models in the library. The optimizer ignores missing values and calculates the weights with the observations and predictions available for each model separately. If each of the models has a different pattern of missingness in the predictors, then the resulting ensemble weights may be biased and the function issues a message.

References

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.2859&rep=rep1&type=pdf>

Examples

```
## Not run:
set.seed(42)
models <- caretList(iris[1:50,1:2], iris[1:50,3], methodList=c('glm', 'lm'))
ens <- caretEnsemble(models)
summary(ens)

## End(Not run)
```

caretList
Create a list of several train models from the caret package

Description

Build a list of train objects suitable for ensembling using the [caretEnsemble](#) function.

Usage

```
caretList(..., trControl = trainControl(), methodList = NULL,
  tuneList = NULL)
```

Arguments

...	arguments to pass to train . These arguments will determine which train method gets dispatched.
trControl	a trainControl object. We are going to intercept this object check that it has the "index" slot defined, and define the indexes if they are not.
methodList	optional, a character vector of caret models to ensemble. One of methodList or tuneList must be specified.
tuneList	optional, a NAMED list of caretModelSpec objects. This much more flexible than methodList and allows the specification of model-specific parameters (e.g. passing trace=FALSE to nnet)

Value

A list of [train](#) objects

Examples

```
## Not run:
myControl <- trainControl(method='cv', number=5)
caretList(
  Sepal.Length ~ Sepal.Width,
  head(iris, 50),
  methodList=c('glm', 'lm'),
  trControl=myControl
)
```

```

caretList(
  Sepal.Length ~ Sepal.Width,
  head(iris, 50), methodList=c('lm'),
  tuneList=list(
    nnet=caretModelSpec(method='nnet', trace=FALSE, tuneLength=1)
  ),
  trControl=myControl
)

## End(Not run)

```

caretModelSpec	<i>Generate a specification for fitting a caret model</i>
----------------	---

Description

A caret model specification consists of 2 parts: a model (as a string) and the arguments to the train call for fitting that model

Usage

```
caretModelSpec(method = "rf", ...)
```

Arguments

method	the modeling method to pass to caret::train
...	Other arguments that will eventually be passed to caret::train

Value

a list of lists

Examples

```
caretModelSpec('rf', tuneLength=5, preProcess='ica')
```

caretStack	<i>Combine several predictive models via stacking</i>
------------	---

Description

Find a good linear combination of several classification or regression models, using either linear regression, elastic net regression, or greedy optimization.

Usage

```
caretStack(all.models, ...)
```

Arguments

all.models a list of caret models to ensemble.
... additional arguments to pass to the optimization function

Details

Check the models, and make a matrix of obs and preds

Value

S3 caretStack object

References

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.2859&rep=rep1&type=pdf>

Examples

```
## Not run:  
library('rpart')  
models <- caretList(  
  x=iris[1:50,1:2],  
  y=iris[1:50,3],  
  trControl=trainControl(method='cv'),  
  methodList=c('rpart', 'glm')  
)  
caretStack(models, method='glm')  
  
## End(Not run)
```

check_bestpreds_indexes

Check row indexes

Description

Check that the row indexes from a caretList are valid

Usage

```
check_bestpreds_indexes(modellibrary)
```

Arguments

modellibrary a list of predictins from caret models

`check_bestpreds_obs` *Check observeds*

Description

Check that a list of observed values from a `caretList` are valid

Usage

```
check_bestpreds_obs(modelLibrary)
```

Arguments

`modelLibrary` a list of predictins from caret models

`check_bestpreds_preds` *Check predictions*

Description

Check that a list of predictions from a `caretList` are valid

Usage

```
check_bestpreds_preds(modelLibrary)
```

Arguments

`modelLibrary` a list of predictins from caret models

`check_bestpreds_resamples`
Check resamples

Description

Check that the resamples from a `caretList` are valid

Usage

```
check_bestpreds_resamples(modelLibrary)
```

Arguments

`modelLibrary` a list of predictins from caret models

check_caretList_classes
Checks caretList model classes

Description

This function checks caretList classes

Usage

```
check_caretList_classes(list_of_models)
```

Arguments

list_of_models a list of caret models to check

check_caretList_model_types
Checks that caretList models are all of the same type.

Description

Checks that caretList models are all of the same type.

Usage

```
check_caretList_model_types(list_of_models)
```

Arguments

list_of_models a list of caret models to check

dotplot.caretStack *Comparison dotplot for a caretStack object*

Description

This is a function to make a dotplot from a caretStack. It uses dotplot from the caret package on all the models in the ensemble, plus the final ensemble model. At the moment, this function only works if the ensembling model has the same number of resamples as the component models.

Usage

```
## S3 method for class 'caretStack'  
dotplot(x, data = NULL, ...)
```

Arguments

x	An object of class caretStack
data	passed to dotplot
...	passed to dotplot

Examples

```
## Not run:
set.seed(42)
library('rpart')
models <- caretList(
  x=iris[1:100,1:2],
  y=iris[1:100,3],
  trControl=trainControl(method='cv'),
  methodList=c('rpart', 'glm')
)
meta_model <- caretStack(models, method='lm', trControl=trainControl(method='cv'))
dotplot.caretStack(meta_model)

## End(Not run)
```

extractBestPreds	<i>Extract the best predictions from a list of train objects</i>
------------------	--

Description

Extract predictions for the best tune from a list of caret models

Usage

```
extractBestPreds(list_of_models)
```

Arguments

list_of_models an object of class caretList

extractCaretTarget	<i>Extracts the target variable from a set of arguments headed to the caret::train function.</i>
--------------------	--

Description

This function extracts the y variable from a set of arguments headed to a caret::train model. Since there are 2 methods to call caret::train, this function also has 2 methods.

Usage

```
extractCaretTarget(...)
```

Arguments

... a set of arguments, as in the `caret::train` function

`extractCaretTarget.default`

Extracts the target variable from a set of arguments headed to the `caret::train.default` function.

Description

This function extracts the y variable from a set of arguments headed to a `caret::train.default` model.

Usage

```
## Default S3 method:  
extractCaretTarget(x, y, ...)
```

Arguments

x an object where samples are in rows and features are in columns. This could be a simple matrix, data frame or other type (e.g. sparse matrix). See Details below.

y a numeric or factor vector containing the outcome for each sample.

... ignored

`extractCaretTarget.formula`

Extracts the target variable from a set of arguments headed to the `caret::train.formula` function.

Description

This function extracts the y variable from a set of arguments headed to a `caret::train.formula` model.

Usage

```
## S3 method for class 'formula'  
extractCaretTarget(form, data, ...)
```

Arguments

form	A formula of the form $y \sim x_1 + x_2 + \dots$
data	Data frame from which variables specified in formula are preferentially to be taken.
...	ignored

extractModelTypes	<i>Extracts the model types from a list of train model</i>
-------------------	--

Description

Extracts the model types from a list of train model

Usage

```
extractModelTypes(list_of_models)
```

Arguments

list_of_models an object of class caretList

extractModFrame	<i>Extract a dataframe of all predictors used in a caretEnsemble object.</i>
-----------------	--

Description

This function constructs a dataframe consisting of the outcome and all of the predictors used in any of the models ensembled in a caretEnsemble object.

Usage

```
extractModFrame(model)
```

Arguments

model a caretEnsemble to extract predictors from

Value

A data.frame combining all of the variables used across all models.

extractModRes	<i>Extract the model accuracy metrics of the individual models in an ensemble object.</i>
---------------	---

Description

Extract the model accuracy metrics of the individual models in an ensemble object.

Usage

```
extractModRes(ensemble)
```

Arguments

ensemble a caretEnsemble to make predictions from.

fortify.caretEnsemble	<i>Supplement the data fitted to a caret ensemble model with model fit statistics</i>
-----------------------	---

Description

This function constructs a dataframe consisting of the outcome, all of the predictors used in any of the models ensembled in a caretEnsemble object, and some model fit statistics.

Usage

```
## S3 method for class 'caretEnsemble'
fortify(model, data = NULL, ...)
```

Arguments

model a caretEnsemble to extract predictors from
 data a data set, defaults to the data used to fit the model
 ... additional arguments to pass to fortify

Value

The original data with extra columns for fitted values and residuals

`getMetric`*Extract a model accuracy metric from an S3 object.*

Description

Extract a model accuracy metric from an S3 object.

Extract a model accuracy metric from a `train` object.

Extract the AUC metric from a `train` object.

Extract the RMSE metric from a model object.

Usage

```
getMetric(x, metric)
```

```
## S3 method for class 'train'  
getMetric(x, metric = c("AUC", "RMSE"))
```

```
getAUC(x)
```

```
getRMSE(x)
```

Arguments

`x` an object with model performanc metrics

`metric` a character, either "RMSE" or "AUC" indicating which metric to extract

Value

A numeric representing the metric desired metric.

A numeric for the AUC of the best model

A numeric for the RMSE of the best model

Note

AUC extracted from a train object is for all resamples pooled, not the average of the AUC for each resample.

RMSE extracted from a train object is for all resamples pooled, not the average of the RMSE for each resample. All missing values are ignored.

getMetricSD	<i>Extract the standard deviation from resamples for an accuracy metric from a model object.</i>
-------------	--

Description

Extract the standard deviation from resamples for an accuracy metric from a model object.

Usage

```
getMetricSD(x, metric, which = c("all", "best"))
```

Arguments

x	an object with model performanc metrics
metric	a character, either "RMSE" or "AUC" indicating which metric to extract
which	a character, either "all" or "best", default is best, see details

Details

Which allows the user to select whether to generate a standard deviation for the performance metric across all values of the tuning parameters and resamples, or only for resamples under the best tuning parameter. Missing values are ignored.

Value

A numeric for the standard deviation of the selected metric across tuning parameters and resamples in the original object.

greedOptAUC	<i>Greedy optimization of the area under the curve</i>
-------------	--

Description

This algorithm optimizes the area under the curve for classification models

Usage

```
greedOptAUC(X, Y, iter = 100L)
```

Arguments

X	the matrix of predictors
Y	the dependent variable
iter	an integer for the number of iterations

Details

If the optimization fails to produce an error term better than the best component model, a message is returned and the best optimization after N iterations is returned.

Value

A numeric of the weights for each model.

Examples

```
x <- matrix(runif(10), ncol=2)
y <- sample(c('Y', 'N'), 5, replace=TRUE)
greedOptAUC(x, y, iter = 2L)
```

greedOptRMSE

Greedy optimization of the reduced mean square error

Description

This algorithm optimizes the RMSE for regression models

Usage

```
greedOptRMSE(X, Y, iter = 100L)
```

Arguments

X	the matrix of predictors
Y	the dependent variable
iter	an integer for the number of iterations

Details

If the optimization fails to produce an error term better than the best component model, a message is returned and the best optimization after iterations is returned.

Value

A numeric of the weights for each model

Examples

```
x <- matrix(runif(10), ncol=2)
y <- runif(5)
greedOptRMSE(x, y, iter = 2L)
```

makePredObsMatrix	<i>Make a prediction matrix from a list of models</i>
-------------------	---

Description

Extract obs from one models, and a matrix of predictions from all other models, a helper function

Usage

```
makePredObsMatrix(list_of_models)
```

Arguments

list_of_models an object of class caretList

methodCheck	<i>Check that the methods supplied by the user are valid caret methods</i>
-------------	--

Description

This function uses modelLookup from caret to ensure the list of methods supplied by the user are all models caret can fit.

Usage

```
methodCheck(x)
```

Arguments

x a list of user-supplied tuning parameters and methods

multiResiduals	<i>Calculate the residuals from all component models of a caretEnsemble.</i>
----------------	--

Description

This function calculates raw residuals for both regression and classification train objects within a [caretEnsemble](#).

Usage

```
multiResiduals(object, ...)
```

Arguments

object a caretEnsemble to make predictions from.
... other arguments to be passed to residuals

Value

A data.frame in the long format with columns for the model method, the observation id, yhat for the fitted values, resid for the residuals, and y for the observed value.

plot.caretEnsemble *Plot Diagnostics for an caretEnsemble Object*

Description

This function makes a short plot of the performance of the component models of a caretEnsemble object on the AUC or RMSE metric

Usage

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

Arguments

x a caretEnsemble object
... additional arguments to pass to plot

Value

A plot

Examples

```
## Not run:  
set.seed(42)  
models <- caretList(iris[1:50,1:2], iris[1:50,3], methodList=c('glm', 'rpart'))  
ens <- caretEnsemble(models)  
plot(ens)  
  
## End(Not run)
```

plot.caretStack	<i>Plot a caretStack object</i>
-----------------	---------------------------------

Description

This is a function to plot a caretStack.

Usage

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

Arguments

x	An object of class caretStack
...	passed to plot

Examples

```
## Not run:
library('rpart')
models <- caretList(
  x=iris[1:100,1:2],
  y=iris[1:100,3],
  trControl=trainControl(method='cv'),
  methodList=c('rpart', 'glm')
)
meta_model <- caretStack(models, method='rpart', tuneLength=2)
plot(meta_model)

## End(Not run)
```

predict.caretEnsemble	<i>Make predictions from a caretEnsemble. This function passes the data to each function in turn to make a matrix of predictions, and then multiplies that matrix by the vector of weights to get a single, combined vector of predictions.</i>
-----------------------	---

Description

Make predictions from a caretEnsemble. This function passes the data to each function in turn to make a matrix of predictions, and then multiplies that matrix by the vector of weights to get a single, combined vector of predictions.

Usage

```
## S3 method for class 'caretEnsemble'
predict(object, keepNA = TRUE, se = FALSE,
        return_weights = FALSE, ...)
```

Arguments

object	a <code>caretEnsemble</code> to make predictions from.
keepNA	a logical indicating whether predictions should be made for all cases where sufficient data exists or only for complete cases across all models. When TRUE this does not predict for missing values. When FALSE, missing values are overwritten with predictions where possible.
se	logical, should prediction errors be produced? Default is false.
return_weights	a logical indicating whether prediction weights for each model for each observation should be returned
...	arguments (including newdata) to pass to predict.train. These arguments must be named

Value

If `return_weights = TRUE` a list is returned with a data.frame slot for predictions and a matrix slot for the model weights. If `return_weights = FALSE` a data.frame is returned for the predictions.

Examples

```
## Not run:
set.seed(42)
models <- caretList(iris[1:50,1:2], iris[1:50,3], methodList=c('glm', 'lm'))
ens <- caretEnsemble(models)
cor(predict(ens, newdata=iris[51:150,1:2]), iris[51:150,3])

## End(Not run)
```

predict.caretList *Create a matrix of predictions for each of the models in a caretList*

Description

Make a matrix of predictions from a list of caret models

Usage

```
## S3 method for class 'caretList'
predict(object, ..., verbose = FALSE)
```

Arguments

object	an object of class caretList
...	additional arguments to pass to predict.train. Pass the newdata argument here, DO NOT PASS the "type" argument. Classification models will return probabilities if possible, and regression models will return "raw".
verbose	Logical. If FALSE no progress bar is printed if TRUE a progress bar is shown. Default FALSE.

predict.caretStack *Make predictions from a caretStack*

Description

Make predictions from a caretStack. This function passes the data to each function in turn to make a matrix of predictions, and then multiplies that matrix by the vector of weights to get a single, combined vector of predictions.

Usage

```
## S3 method for class 'caretStack'
predict(object, newdata = NULL, ...)
```

Arguments

object	a caretStack to make predictions from.
newdata	a new dataframe to make predictions on
...	arguments to pass to predict.train .

Examples

```
## Not run:
library('rpart')
models <- caretList(
  x=iris[1:100,1:2],
  y=iris[1:100,3],
  trControl=trainControl(method='cv'),
  methodList=c('rpart', 'glm')
)
meta_model <- caretStack(models, method='lm')
RMSE(predict(meta_model, iris[101:150,1:2]), iris[101:150,3])

## End(Not run)
```

```
print.caretStack      Print a caretStack object
```

Description

This is a function to print a caretStack.

Usage

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

Arguments

x	An object of class caretStack
...	ignored

Examples

```
## Not run:  
library('rpart')  
models <- caretList(  
  x=iris[1:100,1:2],  
  y=iris[1:100,3],  
  trControl=trainControl(method='cv'),  
  methodList=c('rpart', 'glm')  
)  
meta_model <- caretStack(models, method='lm')  
print(meta_model)  
  
## End(Not run)
```

```
residuals.caretEnsemble  
      Calculate the residuals from a caretEnsemble.
```

Description

This function calculates raw residuals for both regression and classification caretEnsemble objects.

Usage

```
## S3 method for class 'caretEnsemble'  
residuals(object, ...)
```

Arguments

object a caretEnsemble to make predictions from.
... other arguments to be passed to residuals

Value

A numeric of the residuals.

safeOptAUC *Safe optimization of the AUC*

Description

This algorithm optimizes the AUC for regression models to avoid ensembling where the ensemble model fits worse than any component model

Usage

```
safeOptAUC(X, Y, iter = 100L)
```

Arguments

X the matrix of predictors
Y the dependent variable
iter an integer for the number of iterations

Details

This optimizer uses a stopping criterion that if the optimized model has an AUC that is worse than any individual model, it continues optimizing until this is no longer the case. If it fails to surpass any component model it issues a warning and weights the best model 1 and all other models 0.

Value

A numeric of the weights for each model

Examples

```
x <- matrix(runif(10), ncol=2)  
y <- sample(c('Y', 'N'), 5, replace=TRUE)  
safeOptAUC(x, y, iter = 2L)
```

summary.caretEnsemble *Summarize the results of caretEnsemble for the user.*

Description

Summarize the results of caretEnsemble for the user.

Usage

```
## S3 method for class 'caretEnsemble'
summary(object, ...)
```

Arguments

object a [caretEnsemble](#) to make predictions from.
 ... optional additional parameters.

Examples

```
## Not run:
set.seed(42)
models <- caretList(iris[1:50,1:2], iris[1:50,3], methodList=c('glm', 'lm'))
ens <- caretEnsemble(models)
summary(ens)

## End(Not run)
```

summary.caretStack *Summarize a caretStack object*

Description

This is a function to summarize a caretStack.

Usage

```
## S3 method for class 'caretStack'
summary(object, ...)
```

Arguments

object An object of class caretStack
 ... ignored

Examples

```
## Not run:
library('rpart')
models <- caretList(
  x=iris[1:100,1:2],
  y=iris[1:100,3],
  trControl=trainControl(method='cv'),
  methodList=c('rpart', 'glm')
)
meta_model <- caretStack(models, method='lm')
summary(meta_model)

## End(Not run)
```

trControlCheck	<i>Check that the trainControl object supplied by the user is valid and has defined re-sampling indexes.</i>
----------------	--

Description

This function checks the user-supplied trainControl object and makes sure it has all the required fields. If the resampling indexes are missing, it adds them to the model. If savePredictions=FALSE, this function sets it to TRUE.

Usage

```
trControlCheck(x, y)
```

Arguments

x	a trainControl object.
y	the target for the model. Used to determine resampling indexes.

tuneCheck	<i>Check that the tuning parameters list supplied by the user is valid</i>
-----------	--

Description

This function makes sure the tuning parameters passed by the user are valid and have the proper naming, etc.

Usage

```
tuneCheck(x)
```

Arguments

x	a list of user-supplied tuning parameters and methods
---	---

`varImp.caretEnsemble` *Calculate the variable importance of variables in a `caretEnsemble`.*

Description

This function wraps the `varImp` function in the `caret` package to provide a weighted estimate of the importance of variables in the ensembled models in a `caretEnsemble` object. Variable importance for each model is calculated and then averaged by the weight of the overall model in the ensembled object.

Usage

```
## S3 method for class 'caretEnsemble'
varImp(object, scale = TRUE, weight = TRUE, ...)
```

Arguments

<code>object</code>	a <code>caretEnsemble</code> to make predictions from.
<code>scale</code>	should importance values be scaled 0 to 100?
<code>weight</code>	should a model weighted importance be returned?
<code>...</code>	other arguments to be passed to <code>varImp</code>

Value

A `data.frame` with one row per variable and one column per model in object

`wtd.sd` *Calculate a weighted standard deviation*

Description

Used to weight deviations among ensembled model predictions

Usage

```
wtd.sd(x, weights = NULL, normwt = FALSE, na.rm = FALSE)
```

Arguments

<code>x</code>	a vector of numerics
<code>weights</code>	a vector of weights equal to length of <code>x</code>
<code>normwt</code>	a logical indicating whether the weights should be normalized to 1
<code>na.rm</code>	a logical indicating how to handle missing values, default = FALSE

Index

autoplot.caretEnsemble, 3

caretEnsemble, 4, 4, 5, 17, 20, 24

caretList, 5

caretModelSpec, 6

caretStack, 6, 21

check_bestpreds_indexes, 7

check_bestpreds_obs, 8

check_bestpreds_preds, 8

check_bestpreds_resamples, 8

check_caretList_classes, 9

check_caretList_model_types, 9

data.frame, 26

dotplot.caretStack, 9

extractBestPreds, 10

extractCaretTarget, 10

extractCaretTarget.default, 11

extractCaretTarget.formula, 11

extractModelTypes, 12

extractModFrame, 12

extractModRes, 13

fortify.caretEnsemble, 13

getAUC (getMetric), 14

getMetric, 14

getMetricSD, 15

getRMSE (getMetric), 14

greedOptAUC, 15

greedOptRMSE, 16

makePredObsMatrix, 17

methodCheck, 17

multiResiduals, 17

plot.caretEnsemble, 18

plot.caretStack, 19

predict.caretEnsemble, 19

predict.caretList, 20

predict.caretStack, 21

predict.train, 21

print.caretStack, 22

residuals.caretEnsemble, 22

safeOptAUC, 23

summary.caretEnsemble, 24

summary.caretStack, 24

train, 5, 14

trainControl, 5

trControlCheck, 25

tuneCheck, 25

varImp, 26

varImp.caretEnsemble, 26

wtd.sd, 26