Package ‘caretEnsemble’

September 20, 2023

Encoding UTF-8
Type Package
Title Ensembles of Caret Models
Version 2.0.3
Date 2023-09-19

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

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

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

Depends R (>= 3.2.0)

Suggests caTools, testthat, lintr, randomForest, glmnet, rpart, kernlab, nnet, e1071, ipred, pROC, knitr, mlbench, MASS, gbm, klaR, rmarkdown, usethis, sass

Imports methods, pbapply, ggplot2, digest, plyr, lattice, gridExtra, data.table, caret

License MIT + file LICENSE

VignetteBuilder knitr

RoxygenNote 7.2.3

LazyData true

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2023-09-20 13:00:02 UTC
R topics documented:

as.caretList .................................................. 3
as.caretList.default ........................................ 3
as.caretList.list .............................................. 4
autoplot ........................................................ 4
bestPreds ....................................................... 5
c.caretList ..................................................... 5
c.train .......................................................... 6
caretEnsemble ................................................ 7
caretList ....................................................... 8
caretModelSpec ............................................... 9
caretStack ..................................................... 10
check_bestpreds_indexes ..................................... 11
check_bestpreds_obs ......................................... 11
check_bestpreds_preds ....................................... 12
check_bestpreds_resamples .................................. 12
check_caretList_classes ..................................... 12
check_caretList_model_types ............................... 13
dotplot.caretStack .......................................... 13
extractBestPreds ............................................. 14
extractCaretTarget .......................................... 14
extractCaretTarget.default ................................. 15
extractCaretTarget.formula ................................. 15
extractModelName ............................................ 16
extractModelTypes ........................................... 16
extractModFrame .............................................. 17
extractModRes ................................................ 17
fortify .......................................................... 18
getBinaryTargetLevel ....................................... 18
getMetric ...................................................... 19
is.caretEnsemble ............................................ 19
is.caretList ................................................... 20
is.caretStack .................................................. 20
makePredObsMatrix .......................................... 20
methodCheck .................................................. 21
models.class .................................................. 21
multiResiduals ............................................... 21
plot.caretEnsemble .......................................... 22
plot.caretStack ............................................... 23
predict.caretList .......................................... 23
predict.caretStack .......................................... 24
print.caretStack ............................................. 25
residuals.caretEnsemble .................................... 26
setBinaryTargetLevel ....................................... 26
summary.caretEnsemble ..................................... 27
summary.caretStack .......................................... 27
trControlCheck .............................................. 28
as.caretList

Convert object to caretList object

Description

Converts object into a caretList

Usage

as.caretList(object)

Arguments

object R Object

Value

a caretList object

as.caretList.default

Convert object to caretList object - For Future Use

Description

Converts object into a caretList - For Future Use

Usage

## Default S3 method:
as.caretList(object)

Arguments

object R object

Value

NA
as.caretList.list  Convert list to caretList

Description

Converts list to caretList

Usage

## S3 method for class 'list'
as.caretList(object)

Arguments

object list of caret models

Value

a caretList object

autoplot  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

autoplot(object, which = c(1:6), mfrow = c(3, 2), xvars = NULL, ...)

Arguments

object a caretEnsemble object
which an integer index for which of the plots to print. DOES NOTHING.
mfrow an integer vector of length 2 specifying the number of rows and columns for plots
xvars a vector of the names of x variables to plot against residuals
... additional arguments to pass to autoplot
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

```r
## 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)
```

bestPreds

Extract the best predictions from a train object

Description

Extract predictions for the best tune from a model

Usage

bestPreds(x)

Arguments

x a train object

c.caretList

S3 definition for concatenating caretList

Description

take N objects of class caretList and concatenate them into a larger object of class caretList for future ensembling
Usage

```r
## S3 method for class 'caretList'
c(...)
```

Arguments

... the objects of class caretList or train to bind into a caretList

Value

a `caretList` object

Examples

```r
## Not run:
model_list1 <- caretList(Class ~ .,
data=Sonar, trControl = ctrl1,
tuneList = list(
  glm=caretModelSpec(method='glm', family='binomial'),
rpart=caretModelSpec(method='rpart')
),
metric='ROC')

model_list2 <- caretList(Class ~ .,
data=Sonar,
trControl = ctrl1,
tuneList = list(
  glm=caretModelSpec(method='rpart'),
rpart=caretModelSpec(method='rf')
),
metric='ROC')

bigList <- c(model_list1, model_list2)

## End(Not run)
```

---

**c.train**

*S3 definition for concatenating train objects*

Description

take N objects of class train and concatenate into an object of class caretList for future ensembling

Usage

```r
## S3 method for class 'train'
c(...)
```
caretEnsemble

Arguments

... the objects of class train to bind into a caretList

Value

a caretList object

Examples

## Not run:
rpartTrain <- train(Class ~ .,
data=Sonar,
trControl = ctrl1,
method='rpart')

rfTrain <- train(Class ~ .,
data=Sonar,
trControl = ctrl1,
method='rf')

bigList <- c(model_list1, model_list2)

## End(Not run)
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.

Examples

```r
## 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 Build a list of train objects suitable for ensembling using the caretEnsemble function.

Description

Create a list of several train models from the caret package Build a list of train objects suitable for ensembling using the caretEnsemble function.

Usage

caretList(
  ..., trControl = NULL,
  methodList = NULL,
  tuneList = NULL,
  continue_on_fail = FALSE
)
caretModelSpec

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)

`continue_on_fail`,

logical, should a valid caretList be returned that excludes models that fail, default is FALSE

Value

A list of `train` objects. If the model fails to build, it is dropped from the list.

Examples

```r
## 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

Generate a specification for fitting a caret model

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

Combine several predictive models via stacking

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

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.2859&rep=rep1&type=pdf
## check_bestpreds_indexes

### Check row indexes

Check that the row indexes from a caretList are valid

**Usage**

```r
check_bestpreds_indexes(modelLibrary)
```

**Arguments**

- `modelLibrary` a list of predictions from caret models

## check_bestpreds_obs

### Check observeds

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

**Usage**

```r
check_bestpreds_obs(modelLibrary)
```

**Arguments**

- `modelLibrary` a list of predictions from caret models
check_bestpreds_preds  

**Description**  
Check that a list of predictions from a caretList are valid

**Usage**  
check_bestpreds_preds(modelLibrary)

**Arguments**  
- `modelLibrary` a list of predictions from caret models

check_bestpreds_resamples  

**Description**  
Check that the resamples from a caretList are valid

**Usage**  
check_bestpreds_resamples(modelLibrary)

**Arguments**  
- `modelLibrary` a list of predictions from caret models

check_caretList_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

Validate a caretList

Usage

check_caretList_model_types(list_of_models)

Arguments

list_of_models  a list of caret models to check

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, excluding 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")
)
extractCaretTarget

```r
meta_model <- caretStack(models, method="lm", trControl=trainControl(method="cv"))
dotplot.caretStack(meta_model)
```

## End(Not run)

---

### extractBestPreds

**Extract the best predictions from a list of train objects**

**Description**

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

**Usage**

```r
extractBestPreds(list_of_models)
```

**Arguments**

- `list_of_models` an object of class caretList or a list of caret models

---

### extractCaretTarget

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

**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**

```r
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 ~ x1 + x2 + ...
- **data**: Data frame from which variables specified in formula are preferentially to be taken.
- **...**: ignored
**extractModelName**

*Extract the method name associated with a single train object*

**Description**

Extracts the method name associated with a single train object. Note that for standard models (i.e. those already prespecified by caret), the "method" attribute on the train object is used directly while for custom models the "method" attribute within the model$modelInfo attribute is used instead.

**Usage**

```r
extractModelName(x)
```

**Arguments**

- `x`: a single caret train object

**Value**

Name associated with model

---

**extractModelTypes**

*Extracts the model types from a list of train model*

**Description**

Extracts the model types from a list of train model

**Usage**

```r
extractModelTypes(list_of_models)
```

**Arguments**

- `list_of_models`: an object of class caretList
### extractModFrame

**Description**

Extract a dataframe of all predictors used in a caretEnsemble object.

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**

```r
extractModFrame(model)
```

**Arguments**

- `model`: a caretEnsemble to extract predictors from

**Value**

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

### extractModRes

**Description**

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

**Usage**

```r
extractModRes(ensemble)
```

**Arguments**

- `ensemble`: a caretEnsemble to make predictions from.
### fortify

<table>
<thead>
<tr>
<th>Description</th>
<th>Supplement the data fitted to a caret ensemble model with model fit statistics</th>
</tr>
</thead>
</table>

**Usage**

```r
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

---

### getBinaryTargetLevel

<table>
<thead>
<tr>
<th>Description</th>
<th>Return the configured target binary class level</th>
</tr>
</thead>
</table>

**Usage**

```r
getBinaryTargetLevel()
```

**Value**

Currently configured binary target level (as integer equal to 1 or 2)

**See Also**

- `setBinaryTargetLevel`
**getMetric**

*Extract accuracy metrics from a model*

**Description**

Extract accuracy metrics from a model
Extract accuracy metrics SDs from a model
Extract a model accuracy metric from a `train` object.
Extract the standard deviation from resamples for an accuracy metric from a model object.

**Usage**

```r
getMetric(x, metric, ...)
getMetricSD(x, metric, ...)
```

## S3 method for class 'train'
```r
getMetric(x, metric = NULL, ...)
```

## S3 method for class 'train'
```r
getMetricSD(x, metric, ...)
```

**Arguments**

- **x**: a `train` object
- **metric**: which metric to get
- **...**: passed through

**Value**

A numeric representing the metric desired metric.

---

**is.caretEnsemble**

*Check if an object is a caretEnsemble object*

**Description**

Check if an object is a caretEnsemble object

**Usage**

```r
is.caretEnsemble(object)
```

**Arguments**

- **object**: an R object
is.caretList  
Check if an object is a caretList object

Description
Check if an object is a caretList object

Usage
is.caretList(object)

Arguments
object an R object

is.caretStack  
Check if an object is a caretStack object

Description
Check if an object is a caretStack object

Usage
is.caretStack(object)

Arguments
object an R object

makePredObsMatrix  
Make a prediction matrix from a list of models

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**

*Check that the methods supplied by the user are valid caret methods*

**Description**

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

**Usage**

```r
methodCheck(x)
```

**Arguments**

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

---

**models.class**

*caretList of classification models*

**Description**

Data for the caretEnsemble package

**Author(s)**

Zachary Deane-Mayer <zach.mayer@gmail.com>

---

**multiResiduals**

*Calculate the residuals from all component models of a caretEnsemble.*

**Description**

This function calculates raw residuals for both regression and classification `train` objects within a `caretEnsemble`.

**Usage**

```r
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.

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

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

Arguments

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

Value

A plot

Examples

```r
## 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

Plot a caretStack object

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.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, newdata = NULL, ..., verbose = FALSE)
Arguments

object  an object of class caretList
newdata New data for predictions. It can be NULL, but this is ill-advised.
...  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,
  se = FALSE,
  level = 0.95,
  return_weights = FALSE,
  na.action = na.omit,
  ...
)

Arguments

object  a caretStack to make predictions from.
newdata a new dataframe to make predictions on
se  logical, should prediction errors be produced? Default is false.
level  tolerance/confidence level
return_weights  a logical indicating whether prediction weights for each model should be returned
na.action  the method for handling missing data passed to predict.train.
...  arguments to pass to predict.train.
print.caretStack

Details

Prediction weights are defined as variable importance in the stacked caret model. This is not available for all cases such as where the library model predictions are transformed before being passed to the stacking model.

Examples

```r
## 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

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

Arguments

- `x` An object of class caretStack
- `...` ignored

Examples

```r
## 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)
```
### residuals.caretEnsemble

_Calculate the residuals from a caretEnsemble._

**Description**

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

**Usage**

```r
## 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.

---

### setBinaryTargetLevel

_Set the target binary class level_

**Description**

For binary classification problems, ensemble stacks and certain performance measures require an awareness of which class in a two-factor outcome is the "target" class. By default, the first level in an outcome factor is used but this value can be overridden using `setBinaryTargetLevel(2L)`

**Usage**

```r
setBinaryTargetLevel(level)
```

**Arguments**

- `level`: an integer in \{1, 2\} to be used as target outcome level

**See Also**

- `getBinaryTargetLevel`
summary.caretEnsemble  Summarize the results of caretEnsemble for the user.

Description

Summarize a caretEnsemble

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

```r
## 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

*Check that the trainControl object supplied by the user is valid and has defined re-sampling indexes.*

**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 or "none", this function sets it to "final".

**Usage**

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

**Arguments**

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

---

### tuneCheck

*Check that the tuning parameters list supplied by the user is valid*

**Description**

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

**Usage**

```r
tuneCheck(x)
```

**Arguments**

- `x`: a list of user-supplied tuning parameters and methods
validateBinaryTargetLevel

Validate arguments given as binary target level

Description

Helper function used to ensure that target binary class levels given by clients can be coerced to an integer and that the resulting integer is in \{1, 2\}.

Usage

validateBinaryTargetLevel(arg)

Arguments

arg argument to potentially be used as new target level

Value

Binary target level (as integer equal to 1 or 2)

validateCustomModel

Validate a custom caret model info list

Description

Currently, this only ensures that all model info lists were also assigned a "method" attribute for consistency with usage of non-custom models.

Usage

validateCustomModel(x)

Arguments

x a model info list (e.g. getModelInfo("rf", regex=F)[[1]])

Value

validated model info list (i.e. x)
### 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**

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

**Arguments**

- `object`: a `caretEnsemble` to make predictions from.
- `...`: other arguments to be passed to `varImp`

**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**

```r
wtd.sd(x, w = NULL, na.rm = FALSE)
```

**Arguments**

- `x`: a vector of numerics
- `w`: a vector of weights equal to length of `x`
- `na.rm`: a logical indicating how to handle missing values, default = `FALSE`
Description

Index a caret list to extract caret models into a new caretList object

Usage

```r
## S3 method for class 'caretList'
object[index]
```

Arguments

- `object`: an object of class caretList
- `index`: selected index
Index

* data
  - models.class, 21
  - [.caretList, 31
  - as.caretList, 3
  - as.caretList.default, 3
  - as.caretList.list, 4
  - autoplot, 4
  - bestPreds, 5
  - c.caretList, 5
  - c.train, 6
  - caretEnsemble, 7, 8, 21, 27
  - caretEnsemble-package (caretEnsemble), 7
  - caretList, 3, 4, 6, 7, 8
  - caretModelSpec, 9
  - caretStack, 10, 24
  - check_bestpreds_indexes, 11
  - check_bestpreds_obs, 11
  - check_bestpreds_preds, 12
  - check_bestpreds_resamples, 12
  - check_caretList_classes, 12
  - check_caretList_model_types, 13
  - data.frame, 30
  - dotplot.caretStack, 13
  - extractBestPreds, 14
  - extractCaretTarget, 14
  - extractCaretTarget.default, 15
  - extractCaretTarget.formula, 15
  - extractModelName, 16
  - extractModelTypes, 16
  - extractModFrame, 17
  - extractModRes, 17
  - fortify, 18
  - getBinaryTargetLevel, 18
  - getMetric, 19
  - getMetricSD (getMetric), 19
  - is.caretEnsemble, 19
  - is.caretList, 20
  - is.caretStack, 20
  - makePredObsMatrix, 20
  - methodCheck, 21
  - models.class, 21
  - models.reg (models.class), 21
  - multiResiduals, 21
  - plot.caretEnsemble, 22
  - plot.caretStack, 23
  - predict.caretList, 23
  - predict.caretStack, 24
  - predict.train, 24
  - print.caretStack, 25
  - residuals.caretEnsemble, 26
  - setBinaryTargetLevel, 26
  - summary.caretEnsemble, 27
  - summary.caretStack, 27
  - train, 9, 19
  - trainControl, 9
  - trControlCheck, 28
  - tuneCheck, 28
  - validateBinaryTargetLevel, 29
  - validateCustomModel, 29
  - varImp, 30
  - varImp.caretEnsemble, 30
  - wtd.sd, 30
  - X.class (models.class), 21
  - X.reg (models.class), 21
  - Y.class (models.class), 21
  - Y.reg (models.class), 21