Package 'regfilter'

October 14, 2022

Version 1.0.2

Title Elimination of Noisy Samples in Regression Datasets using Noise Filters

Description Traditional noise filtering methods aim at removing noisy samples from a classification dataset. This package adapts classic and recent filtering techniques to be used in regression problems. To do this, it uses the approach proposed in Martin (2021) [<doi:10.1109/ACCESS.2021.3123151>]. Thus, the goal of the implemented noise filters is to eliminate samples with noise in regression datasets.

License GPL (>= 3)

URL https://github.com/juanmartinsantos/regfilter

Depends R (>= 3.2.0)

Imports e1071, FNN, gbm, modelr, nnet, randomForest, rpart

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Juan Martin [aut, cre],
José A. Sáez [aut],
Emilio Corchado [aut],
Pablo Morales [ctb] (Author of the NoiseFiltersR package),
Julian Luengo [ctb] (Author of the NoiseFiltersR package),
Luis P.F. Garcia [ctb] (Author of the NoiseFiltersR package),
Ana C. Lorena [ctb] (Author of the NoiseFiltersR package),
Andre C.P.L.F. de Carvalho [ctb] (Author of the NoiseFiltersR package),
Francisco Herrera [ctb] (Author of the NoiseFiltersR package)

Maintainer Juan Martin <juanmartin@usal.es>

Repository CRAN

Date/Publication 2022-03-10 11:00:08 UTC

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

R topics documented:

print.rfdata . . . 2
regAENN . . . 3
regBBNR . . . 5
regCNN . . . 7
regCVCF . . . 9
regDF . . . 11
regEF . . . 13
regENN . . . 15
regFMF . . . 17
regGE . . . 19
regHRRF . . . 21
regIPF . . . 23
regIRF . . . 25
regRND . . . 27
regRNN . . . 29
summary.rfdata . . . 30

Index **33**

`print.rfdata` *Print function for class rfdata*

Description

This methods displays the basic information about the noise filtering process contained in an object of class `rfdata`.

Usage

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

Arguments

`x`	an object of class `rfdata`.
`...`	other options to pass to the function.

Details

This function presents the basic information of the regression noise filter and the resulting noisy dataset contained in the object `x` of class `rfdata`. The information offered is as follows:

- the name of the noise filtering model.
- the parameters associated with the noise filtering model.
- the number of noisy and clean samples in the dataset.

Value

This function does not return any value.

See Also

summary.rfdata, regAENN, regENN, regGE, regEF

Examples

```
# load the dataset
data(rock)

# apply the regression noise filter
set.seed(9)
output <- regAENN(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# print the results
print(output)
```

---

regAENN *All-k Edited Nearest Neighbors for Regression*

Description

Application of the regAENN noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regAENN(x, y, t = 0.2, k = 5, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
k	an integer with the number of nearest neighbors to be used (default: 5).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

regAENN applies regENN from 1 to k throughout the dataset and removes those noisy samples considered by any regENN. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).
yclean	a double vector with the output regressand of clean samples (without errors).
numclean	an integer vector with the amount of clean samples.
idclean	an integer vector with the indices of clean samples.
xnoise	a data frame with the input attributes of noisy samples (with errors).
ynoise	a double vector with the output regressand of noisy samples (with errors).
numnoise	an integer vector with the amount of noisy samples.
idnoise	an integer vector with the indices of noisy samples.
filter	the full name of the noise filter used.
param	a list of the argument values.
call	the function call.

Note that objects of the class rfdata support print.rfdata and summary.rfdata methods.

References

I. Tomek, **An experiment with the edited nearest-neighbor rule**, *IEEE Transactions on Systems, Man, and Cybernetics*, 6:448–452, 1976. doi: 10.1109/TSMC.1976.4309523.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

regENN, regCNN, regGE, print.rfdata, summary.rfdata

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regAENN(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])
```

```
# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regAENN(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

---

regBBNR *Blame Based Noise Reduction for Regression*

Description

Application of the regBBNR noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regBBNR(x, y, t = 0.2, k = 5, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
k	an integer with the number of nearest neighbors to be used (default: 5).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification problems, *Blame Based Noise Reduction* (BBNR) removes a sample if it participates in the misclassification of another sample and if its removal does not produce the misclassification on another correctly classified sample. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class `rfdata`, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

S. Delany and P. Cunningham, **An analysis of case-base editing in a spam filtering system**, *in European Conference on Case-Based Reasoning*, 128-141, 2004. doi: 10.1007/9783540286318_11.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regCNN`, `regRNN`, `regENN`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regBBNR(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
```

```
out.frm <- regBBNR(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

---

regCNN *Condensed Nearest Neighbors for Regression*

Description

Application of the regCNN noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regCNN(x, y, t = 0.2, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

Condensed Nearest Neighbors (CNN) seeks to obtain a data subset that improves the quality of the original dataset. In classification problems, CNN performs a first classification and stores all the samples that are misclassified. Then, those stored samples are taken as a training set. The process stops when all the unstored samples are correctly classified. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).

`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

L. Devroye, L. Gyorfi and G. Lugosi, **Condensed and edited nearest neighbor rules.** *in: A Probabilistic Theory of Pattern Recognition*, 303-313, 1996. doi: 10.1007/9781461207115_19.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regRNN`, `regENN`, `regBBNR`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regCNN(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regCNN(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regCVCF *Cross-Validated Committees Filter for Regression*

Description

Application of the regCVCF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regCVCF(x, y, t = 0.2, nfolds = 10, vote = FALSE, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
nfolds	number of folds in which the dataset is split (default: 10).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification problems, *Cross-Validated Committees Filter* (CVCF) divides the dataset into nfolds cross-validation folds and builds a decision tree with C4.5 on each one. Using each classifier, a prediction of the whole dataset is obtained. Finally, a sample is considered as noisy using a voting scheme (indicated by the argument vote): if equal to TRUE, a consensus voting is used (in which a sample is removed if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is removed if it is misclassified by more than a half of the models). The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

S. Verbaeten and A. Van, **Ensemble methods for noise elimination in classification problems**, *in: International Workshop on Multiple Classifier Systems*, 317-325, 2003. doi: 10.1007/354044938-8_32.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regIPF`, `regIRF`, `regEF`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regCVCF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regCVCF(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regDF *Dynamic Filter for Regression*

Description

Application of the regDF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regDF(x, y, t = 0.2, nfolds = 10, m = 3, vote = FALSE, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
nfolds	number of folds in which the dataset is split (default: 10).
m	an integer in [1,9] with the number of algorithms in the ensemble (default: 3).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification, *Dynamic Filter* (DF) divides the dataset into nfolds cross-validation folds and obtains the prediction of 9 classifiers: SVM; k-NN with k = 3, 5 and 9; CART; C4.5; MLPN; *Random Forest* and *Naive Bayes*. Then, it selects one ensemble of size m with best predictions. Finally, a sample is considered as noisy using a voting scheme (indicated by the argument vote): if equal to TRUE, a consensus voting is used (in which a sample is removed if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is removed if it is misclassified by more than a half of the models). The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class `rfdata`, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

L. Garcia, A. Lorena and A. Carvalho, **A study on class noise detection and elimination**, *Brazilian Symposium on Neural Networks*, 13-18, 2012. doi: 10.1109/SBRN.2012.49.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regEF`, `regGE`, `regHRRF`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regDF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regDF(formula = perm ~ ., data = rock)
```

```
# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

---

regEF *Ensemble Filter for Regression*

Description

Application of the regEF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regEF(x, y, t = 0.2, nfolds = 10, vote = TRUE, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
nfolds	number of folds in which the dataset is split (default: 10).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification, *Ensemble Filter* (EF) divides the dataset into nfolds cross-validation folds. Then, a prediction is obtained for each one of the classifiers –C4.5, NN and LDA. Finally, a sample is considered as noisy using a voting scheme (indicated by the argument vote): if equal to TRUE, a consensus voting is used (in which a sample is removed if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is removed if it is misclassified by more than a half of the models). The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class `rfdata`, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

C. Brodley and M. Friedl, **Identifying mislabeled training data**, *Journal of Artificial Intelligence Research*, 11:131-167, 1999. doi: 10.1613/jair.606.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regDF`, `regCVCF`, `regIPF`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regEF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regEF(formula = perm ~ ., data = rock)
```

```
# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

---

regENN *Edited Nearest Neighbors for Regression*

Description

Application of the regENN noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regENN(x, y, t = 0.2, k = 5, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
k	an integer with the number of nearest neighbors to be used (default: 5).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification, *Edited Nearest Neighbors* (ENN) removes a sample if its class label is different from that of the majority of its nearest neighbors (k). The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).
yclean	a double vector with the output regressand of clean samples (without errors).

numclean	an integer vector with the amount of clean samples.
idclean	an integer vector with the indices of clean samples.
xnoise	a data frame with the input attributes of noisy samples (with errors).
ynoise	a double vector with the output regressand of noisy samples (with errors).
numnoise	an integer vector with the amount of noisy samples.
idnoise	an integer vector with the indices of noisy samples.
filter	the full name of the noise filter used.
param	a list of the argument values.
call	the function call.

Note that objects of the class rfdata support print.rfdata and summary.rfdata methods.

References

L. Devroye, L. Gyorfi and G. Lugosi, **Condensed and edited nearest neighbor rules.** *in A Probabilistic Theory of Pattern Recognition*, 303-313, 1996. doi: 10.1007/9781461207115_19.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

regAENN, regGE, regCNN, print.rfdata, summary.rfdata

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regENN(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regENN(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regFMF *Fusion of Multiple Filters for Regression*

Description

Application of the regFMF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regFMF(x, y, t = 0.2, vote = FALSE, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

Fusion of Multiple Filters for Regression (regFMF) is an adaptation of *Ensembles of label Noise Filters* (ENF) found in the field of classification, which creates an ensemble with the AENN, DF and HARF filtering techniques. Then, each filter generates one vote per sample. A sample is considered as noisy using a voting scheme (indicated by the argument vote): if equal to TRUE, a consensus voting is used (in which a sample is removed if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is removed if it is misclassified by more than a half of the models). The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).

`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

L. Garcia, A. Lorena, S. Matwin and A. de Carvalho, **Ensembles of label noise filters: a ranking approach**, *Data Mining Knowledge Discovery*, 30:1192–1216, 2016. doi: 10.1007/s10618016-04759.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regDF`, `regHRRF`, `regAENN`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regFMF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regFMF(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regGE *Generalized Edition for Regression*

Description

Application of the regGE noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regGE(x, y, t = 0.2, k = 5, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
k	an integer with the number of nearest neighbors to be used (default: 5).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification, *Generalized Edition* (GE) is a generalization of ENN, which can relabel a sample if at least half of its nearest neighbors (k) have the same class label; otherwise it is removed. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).
yclean	a double vector with the output regressand of clean samples (without errors).
numclean	an integer vector with the amount of clean samples.
idclean	an integer vector with the indices of clean samples.
xnoise	a data frame with the input attributes of noisy samples (with errors).

`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

J. Koplowitz and T. A. Brown, **On the relation of performance to editing in nearest neighbor rules**, *Pattern Recognition*, 13:251-255, 1981. doi: 10.1016/00313203(81)901023.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regENN`, `regAENN`, `regRNN`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regGE(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regGE(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regHRRF *Hybrid Repair-Remove Filter for Regression*

Description

Application of the regHRRF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regHRRF(x, y, t = 0.2, vote = FALSE, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

HRRF is an adaptation of *Hybrid Repair-Remove Filter* (HRRF) found in the field of classification, which builds a classifier set using SVM, MLPNN, CART and k-NN (k= 1, 3 and 5) on the dataset. HRRF removes noisy samples depending on chosen *voting* scheme (indicated by the argument vote): if equal to TRUE,a consensus voting is used (in which a sample is removed if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is removed if it is misclassified by more than a half of the models). The process is repeated while the prediction accuracy (over the original dataset) of the ensemble increases. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

A. Miranda, L. Garcia, A. Carvalho and A. Lorena, **Use of classification algorithms in noise detection and elimination** *in Hybrid Artificial Intelligence Systems*, 417-424, 2009. doi: 10.1007/9783642023194_50.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regIPF`, `regEF`, `regFMF`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock) # data regresion

# usage of the default method
set.seed(9)
out.def <- regHRRF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regHRRF(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regIPF *Iterative Partitioning Filter for Regression*

Description

Application of the regIPF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regIPF(x, y, t = 0.4, nfolds = 10, vote = FALSE, p = 0.01, s = 3, i = 0.5, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
nfolds	number of folds in which the dataset is split (default: 10).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
p	a double in [0,1] with the minimum proportion of original samples that must be labeled as noisy (default: 0.4).
s	an integer with the number of iterations without improvement for the stopping criterion (default: 3).
i	a double in [0,1] with the proportion of good samples which must be retained per iteration (default: 0.5).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification, *Iterative Partitioning Filter* (IPF) builds a classifier with C4.5 on each fold (nfolds) to evaluate the whole dataset. The noisy samples are removed depending on the chosen voting scheme (indicated by the argument vote): if equal to TRUE, a consensus voting is used (in which a sample is removed if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is removed if it is misclassified by more than a half of the models). In addition, IPF integrates an iterative process that stops depending on the arguments p, s and i. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class `rfdata`, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

T. M. Khoshgoftaar and P. Rebours, **Improving software quality prediction by noise filtering techniques**, *Journal of Computer Science and Technology*, 22:387-396, 2007. doi: 10.1007/s11390-00790542

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regIRF`, `regCVCF`, `regFMF`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regIPF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
```

```
out.frm <- regIPF(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

`regIRF` *Iterative Robust Filter for Regression*

Description

Application of the regIRF noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regIRF(x, y, t = 0.2, ...)

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

Arguments

`x`	a data frame of input attributes.
`y`	a double vector with the output regressand of each sample.
`t`	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
`...`	other options to pass to the function.
`formula`	a formula with the output regressand and, at least, one input attribute.
`data`	a data frame in which to interpret the variables in the formula.

Details

In classification, *Iterative Robust Filter* (IRF) builds models with C4.5 from the dataset and removes misclassified samples until there are no more wrong classifications. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (`t`) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class `rfdata`, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).

yclean	a double vector with the output regressand of clean samples (without errors).
numclean	an integer vector with the amount of clean samples.
idclean	an integer vector with the indices of clean samples.
xnoise	a data frame with the input attributes of noisy samples (with errors).
ynoise	a double vector with the output regressand of noisy samples (with errors).
numnoise	an integer vector with the amount of noisy samples.
idnoise	an integer vector with the indices of noisy samples.
filter	the full name of the noise filter used.
param	a list of the argument values.
call	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

S. Verbaeten, **Identifying mislabeled training examples in ILP classification problems**, *Proc. Twelfth Belgian-Dutch Conference on Machine Learning*, 71-78, 2002.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regIPF`, `regCVCF`, `regFMF`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regIRF(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regIRF(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regRND *Regressand Noise Detection for Regression*

Description

Application of the regRND noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regRND(x, y, t = 0.2, nfolds = 5, vote = FALSE, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
nfolds	an integer with the number of folds in which the dataset is split (default: 10).
vote	a logical indicating if the consensus voting (TRUE) or majority voting (FALSE) is used (default: FALSE).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

Regressand Noise Detection (RND) is an adaptation of *Class Noise Detection and Classification* (CNDC) found in the field of classification. In a first step, CNDC builds an ensemble with SVM, *Random Forest*, *Naive Bayes*, k-NN and *Neural Network*. Then, a sample is marked as noisy using a voting scheme (indicated by the argument vote): if equal to TRUE, a consensus voting is used (in which a sample is marked as noisy if it is misclassified by all the models); if equal to FALSE, a majority voting is used (in which a sample is marked as noisy if it is misclassified by more than a half of the models). Then, the decision to remove a sample is made by a distance filtering. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

`xclean`	a data frame with the input attributes of clean samples (without errors).
`yclean`	a double vector with the output regressand of clean samples (without errors).
`numclean`	an integer vector with the amount of clean samples.
`idclean`	an integer vector with the indices of clean samples.
`xnoise`	a data frame with the input attributes of noisy samples (with errors).
`ynoise`	a double vector with the output regressand of noisy samples (with errors).
`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

Z. Nematzadeh, R. Ibrahim and A. Selamat, **Improving class noise detection and classification performance: A new two-filter CNDC model**, *Applied Soft Computer*, 94:106428, 2020. doi: 10.1016/j.asoc.2020.106428.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regENN`, `regAENN`, `regGE`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regRND(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regRND(formula = perm ~ ., data = rock[,])

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

regRNN *Reduced Nearest Neighbors for Regression*

Description

Application of the regRNN noise filtering method in a regression dataset.

Usage

```
## Default S3 method:
regRNN(x, y, t = 0.2, ...)

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

Arguments

x	a data frame of input attributes.
y	a double vector with the output regressand of each sample.
t	a double in [0,1] with the *threshold* used by regression noise filter (default: 0.2).
...	other options to pass to the function.
formula	a formula with the output regressand and, at least, one input attribute.
data	a data frame in which to interpret the variables in the formula.

Details

In classification, *Reduced Nearest Neighbors* (RNN) is an enhancement of CNN that includes one more step, which removes samples in the dataset that do not affect the performance of the k-NN classifier. The implementation of this noise filter to be used in regression problems follows the proposal of Martín *et al.* (2021), which is based on the use of a noise threshold (t) to determine the similarity between the output variable of the samples.

Value

The result of applying the regression filter is a reduced dataset containing the clean samples (without errors or noise), since it removes noisy samples (those with errors). This function returns an object of class rfdata, which contains information related to the noise filtering process in the form of a list with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).
yclean	a double vector with the output regressand of clean samples (without errors).
numclean	an integer vector with the amount of clean samples.
idclean	an integer vector with the indices of clean samples.
xnoise	a data frame with the input attributes of noisy samples (with errors).
ynoise	a double vector with the output regressand of noisy samples (with errors).

`numnoise`	an integer vector with the amount of noisy samples.
`idnoise`	an integer vector with the indices of noisy samples.
`filter`	the full name of the noise filter used.
`param`	a list of the argument values.
`call`	the function call.

Note that objects of the class `rfdata` support print.rfdata and summary.rfdata methods.

References

G. Gates, **The reduced nearest neighbor rule (Corresp.)**, *IEEE Transactions on Information Theory*, 18:431-433, 1972. doi: 10.1109/TIT.1972.1054809.

J. Martín, J. A. Sáez and E. Corchado, **On the regressand noise problem: Model robustness and synergy with regression-adapted noise filters.** *IEEE Access*, 9:145800-145816, 2021. doi: 10.1109/ACCESS.2021.3123151.

See Also

`regCNN`, `regBBNR`, `regENN`, `print.rfdata`, `summary.rfdata`

Examples

```
# load the dataset
data(rock)

# usage of the default method
set.seed(9)
out.def <- regRNN(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# show results
summary(out.def, showid = TRUE)

# usage of the method for class formula
set.seed(9)
out.frm <- regRNN(formula = perm ~ ., data = rock)

# check the match of noisy indices
all(out.def$idnoise == out.frm$idnoise)
```

---

summary.rfdata *Summary function for class rfdata*

Description

This methods displays a summary containing information about the noise filtering process contained in an `object` of class `rfdata`.

Usage

```
## S3 method for class 'rfdata'
summary(object, ..., showid = FALSE)
```

Arguments

object	an object of class rfdata.
...	other options to pass to the function.
showid	a logical indicating if the indices of noisy samples must be displayed (default: FALSE).

Details

This function presents a summary containing information of the regression noise filter and the resulting dataset contained in the object of class rfdata. The information offered is as follows:

- the function call.
- the name of the regressand noise filter.
- the parameters associated with the noise filter.
- the number of noisy and clean samples in the dataset.
- the indices of the noisy and clean samples (if showid = TRUE).

Value

A list including information related to the noise filtering process contained in the object object of class rfdata with the following elements:

xclean	a data frame with the input attributes of clean samples (without errors).
yclean	a double vector with the output regressand of clean samples (without errors).
numclean	an integer vector with the amount of clean samples.
idclean	an integer vector with the indices of clean samples.
xnoise	a data frame with the input attributes of noisy samples (with errors).
ynoise	a double vector with the output regressand of noisy samples (with errors).
numnoise	an integer vector with the amount of noisy samples.
idnoise	an integer vector with the indices of noisy samples.
filter	the full name of the noise filter used.
param	a list of the argument values.
call	the function call.

This list also includes the showid argument.

See Also

print.rfdata, regEF, regDF, regHRRF, regIRF

Examples

```
# load the dataset
data(rock)

# apply the regression noise filter
set.seed(9)
output <- regAENN(x = rock[,-ncol(rock)], y = rock[,ncol(rock)])

# print the results
summary(output)
```

Index

print.rfdata, 2, *4*, *6*, *8*, *10*, *12*, *14*, *16*, *18*, *20*, *22*, *24*, *26*, *28*, *30*, *31*

regAENN, *3*, 3, *16*, *18*, *20*, *28*
regBBNR, 5, *8*, *30*
regCNN, *4*, *6*, 7, *16*, *30*
regCVCF, 9, *14*, *24*, *26*
regDF, 11, *14*, *18*, *31*
regEF, *3*, *10*, *12*, 13, *22*, *31*
regENN, *3*, *4*, *6*, *8*, 15, *20*, *28*, *30*
regFMF, 17, *22*, *24*, *26*
regGE, *3*, *4*, *12*, *16*, 19, *28*
regHRRF, *12*, *18*, 21, *31*
regIPF, *10*, *14*, *22*, 23, *26*
regIRF, *10*, *24*, 25, *31*
regRND, 27
regRNN, *6*, *8*, *20*, 29

summary.rfdata, *3*, *4*, *6*, *8*, *10*, *12*, *14*, *16*, *18*, *20*, *22*, *24*, *26*, *28*, *30*, 30