Package 'OneR'

June 20, 2016

Type Package

Title Machine Learning classification algorithm with enhancements

Version 1.2

Date 2016-06-20

Author Holger von Jouanne-Diedrich

Maintainer Holger von Jouanne-Diedrich <holger.jouanne-diedrich@h-ab.de>

Description Implements the One Rule (OneR) Machine Learning classification algorithm with enhancements for sophisticated handling of missing values and numeric data together with extensive diagnostic functions.

License MIT + file LICENSE

URL http://vonjd.github.io/OneR/

BugReports http://github.com/vonjd/OneR/issues

LazyData TRUE

RoxygenNote 5.0.1

R topics documented:

bin *Binning function*

Description

Discretizes all numerical data in a dataframe into categorical bins of equal length or content.

Usage

```
bin(data, nbins = 5, labels = NULL, method = c("length", "content"),
  na.omit = TRUE)
```

Arguments

data	dataframe which contains the data.
nbins	number of bins (= levels).
labels	character vector of labels for the resulting category.
method	a character string specifying the binning method, see 'Details'; can be abbreviated.
na.omit	boolean value whether instances with missing values should be removed.

Details

Character strings and logical strings are coerced into factors. Matrices are coerced into dataframes. When called with a single vector only the respective factor (and not a dataframe) is returned. Method "length" gives intervals of equal length, method "content" gives intervals of equal content (via quantiles).

When "na.omit = FALSE" a new level "NA" is introduced into each factor.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR, optbin

Examples

```
data <- iris
str(data)
str(bin(data))
str(bin(data, nbins = 3))
str(bin(data, nbins = 3, labels = c("small", "medium", "large")))

## Difference between methods "length" and "content"
set.seed(1); table(bin(rnorm(900), nbins = 3))
set.seed(1); table(bin(rnorm(900), nbins = 3, method = "content"))
```

eval_model *Classification Evaluation function*

Description

Function for evaluating a OneR classification model. Prints prediction vs. actual in absolute and relative numbers. Additionally it gives the accuracy and error rate.

Usage

```
eval_model(prediction, actual)
```

Arguments

prediction	vector which contains the predicted values.
actual	dataframe which contains the actual data. When there is more than one column the last last column is taken. A single vector is allowed too.

Details

Invisibly returns a list with the number of correctly classified and total instances and a contingency table with the absolute numbers.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

Examples

```
data <- iris
model <- OneR(data)
summary(model)
prediction <- predict(model, data)
eval_model(prediction, data)
```

is.OneR *Test OneR model objects*

Description

Test if object is a OneR model.

Usage

```
is.OneR(x)
```

Arguments

x	object to be tested.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

Examples

```
model <- OneR(iris)
is.OneR(model) # evaluates to TRUE
```

---

maxlevels *Remove factors with too many levels*

Description

Removes all columns of a dataframe where a factor (or character string) has more than a maximum number of levels.

Usage

```
maxlevels(data, maxlevels = 20, na.omit = TRUE)
```

Arguments

data	dataframe which contains the data.
maxlevels	number of maximum factor levels.
na.omit	boolean value whether missing values should be treated as a level, defaults to omit missing values before counting.

Details

Often categories that have very many levels are not useful in modelling OneR rules because they result in too many rules and tend to overfit. Examples are IDs or names.

Character strings are treated as factors although they keep their datatype. Numeric data is left untouched.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR

Examples

```
df <- data.frame(numeric = c(1:26), alphabet = letters)
str(df)
str(maxlevels(df))
```

OneR *One Rule function*

Description

Builds a model according to the One Rule (OneR) machine learning classification algorithm.

Usage

```
OneR(data, formula = NULL, ties.method = c("first", "chisq"),
  verbose = FALSE)
```

Arguments

data	dataframe, which contains the data. When formula = NULL (the default) the last column must be the target variable.
formula	formula interface for the OneR function.
ties.method	a character string specifying how ties are treated, see 'Details'; can be abbreviated.
verbose	If TRUE prints rank, names and predictive accuracy of the attributes in decreasing order (with ties.method = "first").

Details

All numerical data is automatically converted into five categorical bins of equal length. Instances with missing values are removed. This is done by internally calling the default version of bin before starting the OneR algorithm. To finetune this behaviour data preprocessing with the bin or optbin functions should be performed.

When there is more than one attribute with best performance either the first (from left to right) is being chosen (method "first") or the one with the lowest p-value of a chi-squared test (method "chisq").

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

bin, optbin, eval_model, maxlevels

Examples

```
data <- optbin(iris)
model <- OneR(data, verbose = TRUE)
summary(model)
plot(model)
prediction <- predict(model, data)
eval_model(prediction, data)

## The same with the formula interface:
data <- optbin(iris)
model <- OneR(formula = Species ~., data = data, verbose = TRUE)
summary(model)
plot(model)
prediction <- predict(model, data)
eval_model(prediction, data)
```

---

optbin *Optimal Binning function*

Description

Discretizes all numerical data in a dataframe into categorical bins where the cut points are optimally aligned with the target categories, thereby a factor is returned. When building a OneR model this could result in fewer rules with enhanced accuracy.

Usage

```
optbin(data, formula = NULL, method = c("logreg", "naive"),
  na.omit = TRUE)
```

Arguments

data	dataframe which contains the data. When formula = NULL (the default) the last column must be the target variable.
formula	formula interface for the optbin function.
method	a character string specifying the method for optimal binning, see 'Details'; can be abbreviated.
na.omit	boolean value whether instances with missing values should be removed.

Details

The cutpoints are calculated by pairwise logistic regressions (method "logreg") or as the means of the expected values of the respective classes ("naive"). The function is likely to give unsatisfactory results when the distributions of the respective classes are not (linearly) separable. Method "naive" should only be used when distributions are (approximately) normal, although in this case "logreg" should give comparable results, so it is the preferable (and therefore default) method.

Character strings and logical strings are coerced into factors. Matrices are coerced into dataframes. If the target is numeric it is turned into a factor with the number of levels equal to the number of values. Additionally a warning is given.

When "na.omit = FALSE" a new level "NA" is introduced into each factor.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR, bin

Examples

```
data <- iris # without optimal binning
model <- OneR(data, verbose = TRUE)
summary(model)

data_opt <- optbin(iris) # with optimal binning
model_opt <- OneR(data_opt, verbose = TRUE)
summary(model_opt)

## The same with the formula interface:
data_opt <- optbin(formula = Species ~., data = iris)
model_opt <- OneR(data_opt, verbose = TRUE)
summary(model_opt)
```

---

plot.OneR *Plot Diagnostics for an OneR object*

Description

Plots a mosaic plot for the feature attribute and the target of the OneR model.

Usage

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

Arguments

x	object of class "OneR".
...	further arguments passed to or from other methods.

Details

If more than 20 levels are present for either the feature attribute or the target the function stops with an error.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR

Examples

```
model <- OneR(iris)
plot(model)
```

---

predict.OneR *Predict method for OneR models*

Description

Predict values based on OneR model object.

Usage

```
## S3 method for class 'OneR'
predict(object, newdata, ...)
```

Arguments

object	object of class "OneR".
newdata	dataframe in which to look for the feature variable with which to predict.
...	further arguments passed to or from other methods.

Details

newdata can have the same format as used for building the model but must at least have the feature variable that is used in the OneR rules. If cases appear that were not present when building the model the predicted value is UNSEEN.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR

Examples

```
model <- OneR(iris)
prediction <- predict(model, iris[1:4])
eval_model(prediction, iris[5])
```

---

print.OneR *Print OneR models*

Description

print method for class OneR.

Usage

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

Arguments

x	object of class "OneR".
...	further arguments passed to or from other methods.

Details

Prints the rules and the accuracy of an OneR model.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR

Examples

```
model <- OneR(iris)
print(model)
```

summary.OneR *Summarize OneR models*

Description

summary method for class OneR.

Usage

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

Arguments

object	object of class "OneR".
...	further arguments passed to or from other methods.

Details

Prints the rules of the OneR model, the accuracy, a contingency table of the feature attribute and the target and performs a chi-squared test on this table.

In the contingency table the maximum values in each column are highlighted by adding a '*', thereby representing the rules of the OneR model.

Author(s)

Holger von Jouanne-Diedrich

References

http://vonjd.github.io/OneR/

See Also

OneR

Examples

```
model <- OneR(iris)
summary(model)
```

Index