Package 'tidyrules'

January 30, 2020

Type Package

Title Obtain Rules from Rule Based Models as Tidy Dataframe

Version 0.1.3

Maintainer Srikanth Komala Sheshachala <sri.teach@gmail.com>

Depends R (>= 3.6.0),

Imports tibble (>= 2.0.1), stringr (>= 1.3.1), magrittr (>= 1.5), purrr (>= 0.3.2), assertthat (>= 0.2.0), partykit (>= 1.2.2),

Suggests AmesHousing (>= 0.0.3), dplyr (>= 0.8), C50 (>= 0.1.2), Cubist (>= 0.2.2), rpart (>= 1.2.2), rpart.plot (>= 3.0.7), rsample (>= 0.0.2), testthat (>= 2.0.1), MASS (>= 7.3.50), mlbench (>= 2.1.1), knitr (>= 1.23), rmarkdown (>= 1.13), pander (>= 0.6.3),

Description Utility to convert text based summary of rule based models to a tidy dataframe (where each row represents a rule) with related metrics such as support, confidence and lift. Rule based models from these packages are supported: 'C5.0', 'rpart' and 'Cubist'.

URL https://github.com/talegari/tidyrules

BugReports https://github.com/talegari/tidyrules/issues

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Srikanth Komala Sheshachala [aut, cre], Amith Kumar Ullur Raghavendra [aut]

Repository CRAN

Date/Publication 2020-01-30 09:40:03 UTC

R topics documented:

`addBackquotes` *Add backquotes*

Description

(vectorized) Add backquotes when a string has a space in it

Usage

```
addBackquotes(string)
```

Arguments

`string`	character vector

Value

character vector

Examples

```
tidyrules:::addBackquotes(c("ab", "a b"))
```

package_tidyrules *About 'tidyrules' package*

Description

Obtain rules as tidy dataframes

Author(s)

Maintainer: Srikanth Komala Sheshachala <sri.teach@gmail.com>

Authors:

- Amith Kumar Ullur Raghavendra <amith54@gmail.com>

See Also

Useful links:

- https://github.com/talegari/tidyrules
- Report bugs at https://github.com/talegari/tidyrules/issues

positionSpaceOutsideSinglequotes *Position of space outside single quotes*

Description

(vectorised) Detect the position of space in a string not within a pair of single quotes

Usage

```
positionSpaceOutsideSinglequotes(string)
```

Arguments

string	A chracter vector

Value

A integer vector of positions

Examples

```
tidyrules:::positionSpaceOutsideSinglequotes(c("hello", "hel' 'o "))
```

removeEmptyLines *Remove empty lines*

Description

Remove empty strings from a character vector

Usage

```
removeEmptyLines(strings)
```

Arguments

strings	A character vector

Value

A character vector

Examples

```
tidyrules:::removeEmptyLines(c("abc", "", "d"))
```

strHead *Vectorized semantic equivalent of 'head' for a string*

Description

Picks the substring starting from the first character

Usage

```
strHead(string, n)
```

Arguments

string	string
n	(integer) Number of characters

Details

'n' can be in the interval [-len + 1, len] (both ends inclusive)

Value

A string

Examples

```
tidyrules:::strHead(c("string", "string2"), 2)
tidyrules:::strHead(c("string", "string2"), -1)
```

`strReplaceReduce` *Sequential string replace*

Description

Sequential string replace via reduce

Usage

```
strReplaceReduce(string, pattern, replacement)
```

Arguments

`string`	string
`pattern`	pattern
`replacement`	replacement

Value

character vector

Examples

```
tidyrules:::strReplaceReduce("abcd", c("ab", "dc"), c("cd", "ab"))
```

strSplitSingle *String split a string*

Description

and return a character vector (not a list)

Usage

```
strSplitSingle(string, pattern)
```

Arguments

string	A string
pattern	Passed as-is to 'stringr::str_split'

Value

A character vector

Examples

```
tidyrules:::strSplitSingle("abc,d", ",")
```

strTail *Vectorized semantic equivalent of tail for a string*

Description

Picks the substring starting from the first character

Usage

```
strTail(string, n)
```

Arguments

string	string
n	(integer) Number of characters

Details

'n' can be in the interval [-len + 1, len] (both ends inclusive)

Value

A string

Examples

```
tidyrules:::strTail(c("string", "string2"), 2)
tidyrules:::strTail(c("string", "string2"), -1)
```

`tidyRules` *Obtain rules as a tidy tibble*

Description

Each row corresponds to a rule. A rule can be copied into ‘dplyr::filter‘ to filter the observations corresponding to a rule

Usage

```
tidyRules(object, col_classes = NULL, ...)
```

Arguments

`object`	Fitted model object with rules
`col_classes`	Named list or a named character vector of column classes. Column names of the data used for modeling form the names and the respective classes for the value. One way of obtaining this is by running ‘lapply(data, class)‘.
`...`	Other arguments (currently unused)

Details

tidyRule supports these rule based models: C5, Cubist and rpart.

Value

A tibble where each row corresponds to a rule

Author(s)

Srikanth KS, <sri.teach@gmail.com>

`tidyRules.C5.0` *Obtain rules as a tidy tibble from a C5.0 model*

Description

Each row corresponds to a rule. A rule can be copied into ‘dplyr::filter‘ to filter the observations corresponding to a rule

Usage

```
## S3 method for class 'C5.0'
tidyRules(object, ...)
```

Arguments

`object`	Fitted model object with rules
`...`	Other arguments (See details)

Details

Optional named argument laplace(flag, default: TRUE) is supported. This computes confidence with laplace correction as documented under 'Rulesets' here: [C5 doc](https://www.rulequest.com/see5-unix.html)

Value

A tibble where each row corresponds to a rule. The columns are: support, confidence, lift, lhs, rhs, n_conditions

Author(s)

Srikanth KS, <sri.teach@gmail.com>

Examples

```
data("attrition", package = "rsample")
attrition <- tibble::as_tibble(attrition)
c5_model <- C50::C5.0(Attrition ~., data = attrition, rules = TRUE)
summary(c5_model)
tidyRules(c5_model)
```

`tidyRules.cubist` *Obtain rules as a tidy tibble from a cubist model*

Description

Each row corresponds to a rule. A rule can be copied into ‘dplyr::filter‘ to filter the observations corresponding to a rule

Usage

```
## S3 method for class 'cubist'
tidyRules(object, ...)
```

Arguments

`object`	Fitted model object with rules
`...`	Other arguments (currently unused)

Details

When col_classes argument is missing, an educated guess is made about class by parsing the RHS of sub-rule. This might sometimes not lead to a parsable rule.

Value

A tibble where each row corresponds to a rule. The columns are: support, mean, min, max, error, lhs, rhs and committee

Author(s)

Srikanth KS, <sri.teach@gmail.com>

Examples

```
data("attrition", package = "rsample")
attrition <- tibble::as_tibble(attrition)
cols_att <- setdiff(colnames(attrition), c("MonthlyIncome", "Attrition"))

cb_att <-
  Cubist::cubist(x = attrition[, cols_att],y = attrition[["MonthlyIncome"]])
tr_att <- tidyRules(cb_att)
tr_att
```

`tidyRules.rpart` *Obtain rules as a tidy tibble from a rpart model*

Description

Each row corresponds to a rule. A rule can be copied into ‘dplyr::filter‘ to filter the observations corresponding to a rule

Usage

```
## S3 method for class 'rpart'
tidyRules(object, ...)
```

Arguments

`object`	Fitted model object with rules
`...`	Other arguments (currently unused)

Details

NOTE: For rpart rules, one should build the model without **ordered factor** variable. We recommend you to convert **ordered factor** to **factor** or **integer** class.

Value

A tibble where each row corresponds to a rule. The columns are: support, confidence, lift, LHS, RHS

Author(s)

Amith Kumar U R, <amith54@gmail.com>

Examples

```
iris_rpart <- rpart::rpart(Species ~ .,data = iris)
tidyRules(iris_rpart)
```

varSpec *Get variable specification for a Cubist/C5 object*

Description

Obtain variable names, type (numeric, ordered, factor) and levels as a tibble

Usage

```
varSpec(object)
```

Arguments

object	Cubist/C5 object

Value

A tibble with three columns: variable(character), type(character) and levels(a list-column). For numeric variables, levels are set to NA.

Author(s)

Srikanth KS, <sri.teach@gmail.com>

Examples

```
data("attrition", package = "rsample")
attrition <- tibble::as_tibble(attrition)
cols_att <- setdiff(colnames(attrition), c("MonthlyIncome", "Attrition"))

cb_att <-
  Cubist::cubist(x = attrition[, cols_att],y = attrition[["MonthlyIncome"]])
varSpec(cb_att)
```

Index