Package 'Rdrools'

December 8, 2018

Type Package

Version 1.1.1

Date 2018-12-05

Title A Rules Engine Based on 'Drools'

Description
An interface for using the popular Java based Drools, which is a Business Rule Management System (See <https://www.drools.org> for more information). This package provides data scientists an intuitive interface to execute business rules on datasets for the purpose of analysis or designing intelligent systems, while leveraging the Drools rule engine. Rules written in DRL format accepted natively by Drools can also be executed through an R interface. Credits to Mu Sigma for their continued support throughout the development of the package.

Depends R (>= 3.0.0), rJava, Rdroolsjars (>= 1.0.0)

Imports magrittr, dplyr, purrr, tibble, rlang

Suggests knitr, testthat, DT, lubridate, ggplot2, rmarkdown

SystemRequirements Java (>= 7.0)

License Apache License 2.0

Encoding UTF-8

LazyLoad yes

LazyData yes

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Ashwin Raaghav [aut],
SMS Chauhan [aut],
Naren Srinivasan [aut],
Dheekshitha PS [aut],
Zubin Dowlaty [aut],
Mayukh Bose [aut],
Arushi Khattri [aut],
Mu Sigma, Inc. [cre]

Maintainer ``Mu Sigma, Inc." <ird.experiencelab@mu-sigma.com>

Repository CRAN

Date/Publication 2018-12-08 15:00:13 UTC

R topics documented:

`class` *Sample data of students' grades*

Description

A dataset containing students' grades in different subjects

Usage

```
class
```

Format

A data frame with 15 observations and 5 variables

id ID of the student

name Name of the student

class Subject for which the grade is specified

grade Grade in the subject

email E-mail ID of the student

`executeRulesOnDataset` *Run a set of rules on a dataset*

Description

The `executeRulesOnDataset` function is an intuitive interface to execute rules on datasets, which is explicitly designed for data scientists. As input to this function rules are defined using the typical language of data science with verbs such as filter, group by and aggregate. Rules can be specified in a .csv file, loaded into the R session and passed to this function

Usage

```
executeRulesOnDataset(dataset, rules)
```

Arguments

`dataset`	a data frame on which the defined set of rules should be applied
`rules`	a data frame in which rules are defined

Details

The vignette provides further details on the format of the rules file

Value

A list of input, intermediate output and output per rule

1. **input**: a tibble containing the rules defined by the user
2. **intermediateOutput**: is an empty tibble when there is no group by condition specified, and a tibble with the aggregated value for each group when it is
3. **output**: a tibble with 3 columns:
   (a) **Group**: represents group name when there is a group by condition specified and the row number in the case that there is no group by condition
   (b) **Indices**: the row numbers corresponding to each group when there is a group by specified and row numbers of the data frame in the case that there is no group by condition
   (c) **IsTrue**: flag indicating if the data point/ set of data points satisfies the rule or not. Returns TRUE if the point or group satisfies the rule and FALSE if not.

Author(s)

Dheekshitha PS < Dheekshitha.PS@mu-sigma.com>

Naren Srinivasan <Naren.Srinivasan@mu-sigma.com>

Mayukh Bose <Mayukh.Bose@mu-sigma.com>

See Also

runRulesDrl, Rdrools

Other Interface Functions to Drools: rulesSessionDrl, runRulesDrl

Examples

```
## Not run:
  library(Rdrools)
  data("iris")
  data("irisRules")
  executeRulesOnDataset(iris, irisRules)

## End(Not run)
```

---

irisRules *Sample rules for iris dataset*

Description

A dataset containing the sample rules to be applied on the iris dataset, where each row represents a rule

Usage

```
irisRules
```

Format

A data frame with 7 rows and 6 variables:

Filters Filters to be applied in a specific rule

GroupBy Grouping conditions to be appled in a specific rule

Column The variable on which the function is to applied in a specific rule

Function Function to be applied on the specific variable in a rule

Operation The comparison operation to be performed to check the rule

Argument The value agains which the comparison operation is to be performed

Rdrools *Rdrools*

Description

An interface for using the popular Java based Drools, which is a Business Rule Management System (See <https://www.drools.org> for more information). This package allows data scientists to run rules on datasets provided through a CSV file using common data science verbs such as Filter, Group By and Aggregate. Additionally, rules defined directly in the DRL format accepted by the Drools engine can also be run. Credits to Mu Sigma for their continued support throughout the development of the package.

rules *Sample rules in DRL format*

Description

A list of character strings, providing an example for rules in the DRL format accepted by the Drools engine

Usage

```
rules
```

Format

An object of class `character` of length 26.

rulesSessionDrl *Creates a session of the rules engine*

Description

The rulesSession creates a session that interfaces between R and the Drools engine. The session is utilized by the runRulesDrl function for executing a data frame against a set of rules

Usage

```
rulesSessionDrl(rules, input.columns, output.columns)
```

Arguments

`rules`	a character vector consisting of lines read from a rules file of format *.drl* (Drools rules file). This character vector is eventually collapsed into a character vector of length 1, so the way you read the file could potentially be just about anything
`input.columns`	a character vector of a set of input column, for example. `input.columns<-c('name', 'class', 'grade`
`output.columns`	a character vector of a set of expected output columns, for example. `output.columns<-c('address', 's`

Details

An active drools rules session. This promotes re-usability of a session, i.e. you can utilize the same session repetitively for different data sets of the same format.

Value

rules.session.object Returns a session to the rules engine

Note

Please have a look at the examples provided in the 'examples' section of the `Rdrools`. A sample data set and a set of rules have been supplied help you understand the package usage.

Author(s)

Ashwin Raaghav <ashwin.raaghav@mu-sigma.com>

SMS Chauhan <sms.chauhan@mu-sigma.com>

See Also

`runRulesDrl`, `Rdrools`

Other Interface Functions to Drools: `executeRulesOnDataset`, `runRulesDrl`

Examples

```
library(Rdrools)
data(class)
data(rules)
input.columns<-c('name', 'class', 'grade', 'email')
output.columns<-c('address', 'subject', 'body')
rulesSession<-rulesSessionDrl(rules, input.columns, output.columns)
output.df<-runRulesDrl(rulesSession, class)
```

`runRulesDrl` *Apply a set of rule transformations to a data frame*

Description

This function is the core of the Rdrools package. Rules are applied on an input data frame and the results are returned as the output of the function. The columns on which the rules need to be applied have to be provided explicitly. Additionally, the new columns that would be created based on the rules have to be provided explicitly as well. The rules engine picks up a row from the data frame, applies the transformation to it based on rules provided and saves the result in an output data frame.

Usage

```
runRulesDrl(rules.session, input.df)
```

Arguments

`rules.session`	a session of the rules engine created using the `rulesSessionDrl` function
`input.df`	a data frame consisting of a set of rows you wish to transform, and columns you wish to use in the transformation

Details

If you are not familiar with the Drools file format, please have a look at the references provided in the `Rdrools`. More details on how conflicting rules are resolved using either salience or the Reete algorithm are also present in the references.

Transformation policy Transformations are applied row by row, iteratively. That is to say, all inputs required for a rule transformation should be present in columns as a part of that row itself. Each row should be considered independent of another; all input values required for a transformation should be available in that row itself. The expectation from rules engines are often misplaced.

Column Mismatch Please make sure that the list of output columns provided through the `output.columns` parameter is exhaustive. Any additional column which is created through the rules transformation but is not present in the list would inhibit proper functioning. In most cases, an error should be thrown.

Value

output.df a data frame which is the result of transformations applied to the input data frame(`input.df`), the columns being the list provided through the `output.columns` parameter in `rulesSessionDrl`

Author(s)

Ashwin Raaghav <ashwin.raaghav@mu-sigma.com>

SMS Chauhan <sms.chauhan@mu-sigma.com>

See Also

runRulesDrl, Rdrools

Other Interface Functions to Drools: executeRulesOnDataset, rulesSessionDrl

Examples

```
library(Rdrools)
data(class)
data(rules)
input.columns<-c('name', 'class', 'grade', 'email')
output.columns<-c('address', 'subject', 'body')
rulesSession<-rulesSessionDrl(rules, input.columns, output.columns)
output.df<-runRulesDrl(rulesSession, class)
```

---

transactionData *Sample transaction data*

Description

A dataset containing banking transactions, with variables such as number of transactions, transaction value, and so on

Usage

```
transactionData
```

Format

A data frame with 39999 rows and 16 variables:

---

transactionRules *Sample rules for the sample transaction dataset (transactionData)*

Description

A dataset containing the sample rules to be applied on the transaction dataset, where each row represents a rule

Usage

```
transactionRules
```

Format

A data frame with 7 rows and 6 variables:

Filters Filters to be applied in a specific rule

GroupBy Grouping conditions to be appled in a specific rule

Column The variable on which the function is to applied in a specific rule

Function Function to be applied on the specific variable in a rule

Operation The comparison operation to be performed to check the rule

Argument The value agains which the comparison operation is to be performed

Index