Package 'readMLData'

May 9, 2026

Type Package

Title Reading Machine Learning Benchmark Data Sets in Different Formats

Version 0.9-7

Date 2015-01-13

Author Petr Savicky

Maintainer Petr Savicky <savicky@cs.cas.cz>

Description Functions for reading data sets in different formats for testing machine learning tools are provided. This allows to run a loop over several data sets in their original form, for example if they are downloaded from UCI Machine Learning Repository. The data are not part of the package and have to be downloaded separately.

Imports XML

License GPL-3

URL http://www.cs.cas.cz/~savicky/readMLData

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-13 12:10:48

Contents

`readMLData-package` *Reading data from different sources in their original format.*

Description

The package contains functions, which allow to maintain and use a structure describing a collection of machine learning datasets and read them into R environment using a unified interface, see function `prepareDSList()` and `dsRead()`.

Details

The data are not part of the package. The package requires to receive a path to a local copy of the data and their description. The description of the data sets consists of a directory, which contains an XML file `contents.xml` and subdirectory "scripts", which contains an R script for each data set, which reads the data set into R. File `contents.xml` contains information on all the data sets. In particular it contains their names for local identification, their public names, and the names of files representing the data set. The name of the script for reading a data set is derived from its identification name. The complete list of the fields in `contents.xml` may be obtained using `getFields()`.

For the simplest use of the package for reading the data sets, the functions `prepareDSList()` and `dsRead()` are sufficient. The remaining functions are useful for including further data sets to the description. Use `help(package=readMLData)` or `library(help=readMLData)` to see the list of functions.

The list of fields, which should be included in `"contents.xml"`, consists of the fields with either `usage=="obligatory"` or `usage=="optional"` in the table produced by `getFields()`. Fields with `usage=="additional"` and `usage=="computed"` are included automatically by the function `prepareDSList()`.

An example of the description directory describing three UCI data sets is in `exampleDescription` subdirectory of the installed package. The data themselves are in `exampleData` subdirectory. See http://www.cs.cas.cz/~savicky/readMLData/ for description files of further data sets from UCI Machine Learning Repository.

Author(s)

Petr Savicky

References

UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/.

Additional resources for the CRAN package readMLData, http://www.cs.cas.cz/~savicky/readMLData/.

`analyzeData` *Determine the type of values in each column of a data frame.*

Description

For each column, its class and the number of different values is determined. For numeric columns, also the minimum and maximum is computed.

Usage

```
analyzeData(dat)
```

Arguments

`dat`	A data frame.

Value

A data frame with columns `"class"`, `"num.unique"`, `"min"`, `"max"`, which correspond to properties of columns of `dat`. The rows in the output data frame correspond to the columns of `dat`.

Author(s)

Petr Savicky

See Also

`readMLData`.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
dat <- dsRead(dsList, "glass")
analyzeData(dat)
```

checkConsistency *Checks consistency of the data frame* dsList.

Description

Checks consistency of the parameters specified for each dataset in the dsList data frame created by prepareDSList().

Usage

```
checkConsistency(dsList, outputInd=FALSE)
```

Arguments

dsList	Data frame as created by prepareDSList().
outputInd	Logical. Determines, whether the output should be a vector of indices of the data sets with conflicts.

Value

Depending on outputInd, either a vector of indices of data sets with a conflict between the specified parameters or NULL invisibly.

Author(s)

Petr Savicky

See Also

readMLData.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
checkConsistency(dsList)
```

checkType *Compares the type of columns stored in* dsList *and in a data set itself.*

Description

Compares types.

Usage

```
checkType(dsList, id, dat=NULL)
```

Arguments

dsList	Data frame describing the data sets as produced by prepareDSList().
id	Numeric or character of length one. Index or the identification of a data set.
dat	An optional data frame as read by dsRead(dsList, id, keepContents=TRUE).

Value

The name of the tested data set and the result of the test is printed. If errors are found, a more detailed message is printed. The output value is TRUE or FALSE invisibly according, whether the types are correct or not.

Author(s)

Petr Savicky

See Also

readMLData.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
checkType(dsList, 1)
```

dsDownload *Run an external tool to download a data set.*

Description

The function allows to run an external download tool with arguments read from a file in a data folder.

Usage

```
dsDownload(dsList, id, command, fileName)
```

Arguments

dsList	Data frame as created by prepareDSList().
id	Name of the data set in dsList$identification or the index of the row in dsList corresponding to the data set.
command	Character. A command line web downloding tool, for example "wget".
fileName	Character. A name of the file in the data directory, which contains the URL of the data on the web.

Details

If no data set or more than one data set corresponding to id is found, a corresponding error message is printed.

Value

Function has no value. The protocol generated by the specified tool is printed.

Author(s)

Petr Savicky

See Also

readMLData.

Examples

```
## Not run:
  pathData <- getPath("exampleData")
  pathDescription <- getPath("exampleDescription")
  dsList <- prepareDSList(pathData, pathDescription)
  dat <- dsDownload(dsList, "glass", "wget", "links.txt")

## End(Not run)
```

dsRead *Loading machine learning data from a directory tree using a unified interface.*

Description

The function allows to read data sets included in the description in the data frame `dsList` into R environment using a unified interface.

Usage

```
dsRead(dsList, id, responseName = NULL, originalNames=TRUE,
deleteUnused=TRUE, keepContents=FALSE)
```

Arguments

`dsList`	Data frame as created by `prepareDSList()`.
`id`	Name of the data set in `dsList$identification` or the index of the row in `dsList` corresponding to the data set.
`responseName`	Character. The required name of the response column in the output data frame created from the data set.
`originalNames`	If TRUE, the original names of columns are used, if they are present in the description XML file.
`deleteUnused`	Logical. Controls, whether the columns containing case labels or other columns not suitable as attributes, are removed from the data.
`keepContents`	Logical. If `TRUE`, then `deleteUnused` parameter is ignored and no columns are converted to factors.

Details

The function uses `dsList$avaiable` to determine, whether the files for the required data set is present in the local directory `dsList$pathData`. If not, a corresponding error message is printed. See `prepareDSList()` and `getAvailable()`.

Value

A data frame containing the required data set, possibly transformed according to the setting of the parameters `responseName`, `originalNames`, `deleteUnused`. If an error occurred, the function outputs `NULL`.

Author(s)

Petr Savicky

See Also

`readMLData`, `prepareDSList`, `getAvailable`.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
dat <- dsRead(dsList, "glass")
dim(dat)
```

---

dsSearch *Search a dataset by string matching against the names stored in* dsList.

Description

The function allows string matching against some of the fields "identification", "fullName", "dirName", "files" of the structure describing the data sets.

Usage

```
dsSearch(dsList, id, searchField=c("identification", "fullName", "dirName", "files"),
        searchType=c("exact", "prefix", "suffix", "anywhere"), caseSensitive=FALSE)
```

Arguments

dsList	Data frame as created by prepareDSList().
id	Character of length one or numeric of length at most nrow(dsList). If character, then it is used as a search string to be matched against the names of datasets. If numeric, it is used as indices of data sets in dsList.
searchField	Character. Name of a column in dsList to be searched.
searchType	Character. Type of search.
caseSensitive	Logical. Whether the search should be case sensitive.

Details

The parameter searchField determines, which column of dsList is searched, parameters searchType and caseSensitive influence the type of search. These three parameters are ignored, if id is numeric.

Regular expressions are not used. Matching with searchType="exact" is done with ==, searchType="prefix" and searchType="suffix" are implemented using substr(), searchType="anywhere" is implemented using grep(, fixed=TRUE).

Value

Data frame containing the indices and identification of the matching data sets and the value of the search field, if applicable.

Author(s)

Petr Savicky

See Also

readMLData.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
dsSearch(dsList, "ident", searchField="fullName", searchType="anywhere")
```

---

dsSort *Sort the rows of a data frame.*

Description

Sort the rows of a data frame lexicographically. This allows to compare two data sets as sets of cases disregarding their order.

Usage

```
dsSort(dat)
```

Arguments

dat	a dataframe.

Details

The function calls `order()` with the columns of `dat` as the sorting criteria.

Value

Data frame, whose rows are reordered by the sorting.

Author(s)

Petr Savicky

See Also

readMLData.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
dat <- dsRead(dsList, "glass")
sorted <- dsSort(dat)
```

`getAvailable` *Checks consistency of the data frame* `dsList`.

Description

Checks whether all the files of a specified data set are accesible in a local directory.

Usage

```
getAvailable(dsList, id=NULL, asLogical=FALSE)
```

Arguments

`dsList`	Data frame as created by `prepareDSList()`.
`id`	Character or numeric vector. A character vector should contain names matching the names `dsList$identification`. Numeric vector should consist of the indices of the rows in `dsList` corresponding to the data set. If `id=NULL`, then all data sets are checked.
`asLogical`	Logical, whether the output should be a logical vector of the same length as `id` or a character vector containing the identification of the available data sets.

Details

The test is not completely reliable, since it only verifies that the files with the required file name are accessible. If the files require some transformations after download and these are not performed, the data set is still reported as available. The test uses file names specified in `contents.xml` file. If these names are by mistake different from the files actually read in the reading scripts, then the test may also yield an incorrect result.

Value

Logical vector of the length `length(id)` specifying for each component of `id` the result of the check or a character vector containing the identification of the available data sets.

Author(s)

Petr Savicky

See Also

`readMLData`.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
getAvailable(dsList)
```

`getFields` *Prints the information on the fields in the data frame* `dsList` *describing the data sets.*

Description

The data frame `dsList` contains names of the data sets, the names of the directories, the files, which belong to each of the data sets, and some other information. The function returns a table describing the fields and their usage.

Usage

```
getFields()
```

Value

Table containing the names, types and usage of the fields expected in `dsList`.

Author(s)

Petr Savicky

See Also

`readMLData`.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
getFields()
```

getPath *Determine the path to package example directories.*

Description

Appends the path to the directory of an installed package and a name of its subdirectory.

Usage

```
getPath(dirName)
```

Arguments

dirName	Character. Name of the example subdirectory of an installed package. This is currently exampleDescription or exampleData.

Value

Character string, which is a full path to the required example directory in an installed package.

Author(s)

Petr Savicky

See Also

prepareDSList

getType *Determines the type vector for an input data set.*

Description

The type information is derived from the contents of individual columns of an input data frame.

Usage

```
getType(dat)
```

Arguments

dat	A data frame.

Value

A character vector of length ncol(dat) containing "n" for numerical columns, the number of different values for character or factor columns, and "o" otherwise.

Author(s)

Petr Savicky

See Also

readMLData.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
dat <- dsRead(dsList, "annealing")
getType(dat)
```

---

prepareDSList — *Prepares a data frame* dsList*, which describes the data contained in a local data description directory.*

Description

The data frame dsList is needed to read the data contained in a directory tree below dsList$pathData using dsRead(). The directory pathDescription is expected to contain the file contents.xml and subdirectory scripts with R scripts for reading the data sets.

Usage

```
prepareDSList(pathData, pathDescription)
```

Arguments

pathData	Character. A path to the required data directory.
pathDescription	Character. A path to a directory containing description of the required data, in particular the file "contents.xml".

Details

The character "~" expands to your home directory.

The directory pathData need not contain all the data sets included in pathDescription/contents.xml. The function getAvailable() is called and its output is stored in column availability of the output data frame, which is logical and specifies for each data set, whether it is or is not present.

See http://www.cs.cas.cz/~savicky/readMLData/ for description files of some of the data sets from UCI Machine Learning Repository. See the help page readMLData for more information on the structure of the description files.

Value

Data frame with columns `pathData`, `pathDescription`, and other as listed by `getFields()`. The output data frame can be used as `dsList` parametr of functions `dsSearch()`, `dsRead()`, `checkConsistency()`, `checkType()`.

Author(s)

Petr Savicky

See Also

`readMLData`, `getAvailable`, `checkConsistency`.

Examples

```
pathData <- getPath("exampleData")
pathDescription <- getPath("exampleDescription")
dsList <- prepareDSList(pathData, pathDescription)
```

---

xml *Handling XML files.*

Description

Input and output of a data set description from and to a XML file. These functions are not inteded for direct use by the user for reading the data sets. The function `readDSListFromXML()` is called from `prepareDataDir()`. The function `saveDSListAsXML` is used for preparing the file `contents.xml` in the data set description directory.

Usage

```
readDSListFromXML(filename)
saveDSListAsXML(dsList, filename)
```

Arguments

`dsList`	A data frame created by `prepareDataDirectory()`.
`filename`	The name of an XML file to be used.

Value

`saveDSListAsXML()` returns the filename of the created file. `readDSListFromXML()` returns a data frame with the description of the data sets.

Author(s)

Petr Savicky

See Also

readMLData.

Index