

Package ‘rchallenge’

May 17, 2015

Title A Simple Datascience Challenge System

Version 1.1

Description A simple datascience challenge system using R Markdown and Dropbox.
It requires no network configuration, does not depend on external platforms like e.g. Kaggle and can be easily installed on a personal computer.

URL <http://adrtod.github.io/rchallenge>

BugReports <https://github.com/adrtod/rchallenge/issues>

Depends R (>= 3.2.0)

Imports rmarkdown (>= 0.5.1), knitr (>= 1.6)

SystemRequirements pandoc (>= 1.12.3) -
<http://johnmacfarlane.net/pandoc>

License GPL-2

LazyData true

NeedsCompilation no

Author Adrien Todeschini [aut, cre],
Robin Genuer [ctb]

Maintainer Adrien Todeschini <adrien.todeschini@gmail.com>

Repository CRAN

Date/Publication 2015-05-17 16:38:17

R topics documented:

compute_metrics	2
countdown	3
data_split	3
german	4
get_best	4
get_data	5
glyphicon	6
html_img	6

last_update	7
new_challenge	7
new_team	8
plot_activity	9
plot_history	10
print_leaderboard	11
print_readerr	12
publish	12
store_new_submissions	13
update_rank_diff	14

Index 15

compute_metrics	<i>Compute metrics of the submissions in the history.</i>
-----------------	---

Description

Compute metrics of the submissions in the history.

Usage

```
compute_metrics(hist_dir = "history", metrics, y_test, ind_quiz, read_fun)
```

Arguments

hist_dir	string. directory where the history of the submissions are stored. contains one subdirectory per team.
metrics	named list of functions. Each function in the list computes a performance criterion and is defined as: <code>function(y_pred, y_test)</code>
y_test	character or numeric vector. the test set output.
ind_quiz	logical vector with the same length as y_test. <code>ind_quiz[i]=TRUE</code> if <code>y_test[i]</code> in the quiz subset.
read_fun	function that reads a submission file and returns a vector of predictions.

Value

`compute_metrics` returns a named list with one named member per team. Each member is a data.frame where the rows are the submission files sorted by date and the columns are:

date	the date of the submission
file	the file name of the submission
<metric name>.quiz	the score obtained on the quiz subset
<metric name>.test	the score obtained on the test set

countdown	<i>Countdown before deadline.</i>
-----------	-----------------------------------

Description

Countdown before deadline.

Usage

```
countdown(deadline, complete_str = intToUtf8(10004))
```

Arguments

deadline	POSIXct. deadline
complete_str	string. displayed when deadline is passed

data_split	<i>Split a data.frame into training and test sets.</i>
------------	--

Description

Split a data.frame into training and test sets.

Usage

```
data_split(data = get_data("german"), varname = "Class", p_test = 0.2,
           p_quiz = 0.5)
```

Arguments

data	data.frame
varname	string. output variable name
p_test	real. proportion of samples in the test set
p_quiz	real. proportion of samples from the test set in the quiz set

Value

list with members	
train	training set with output variable
test	test set without output variable
y_test	test set output variable
ind_quiz	indices of quiz samples in the test set

german	<i>German Credit Data.</i>
--------	----------------------------

Description

Data from Dr. Hans Hofmann of the University of Hamburg.

Usage

```
german
```

Format

A data.frame with 1000 rows and 21 variables

Details

These data have two classes for the credit worthiness: Good or Bad. There are predictors related to attributes, such as: checking account status, duration, credit history, purpose of the loan, amount of the loan, savings accounts or bonds, employment duration, Installment rate in percentage of disposable income, personal information, other debtors/guarantors, residence duration, property, age, other installment plans, housing, number of existing credits, job information, Number of people being liable to provide maintenance for, telephone, and foreign worker status.

This is a transformed version of the [GermanCredit](#) data set with factors instead of dummy variables

Source

UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Statlog+\(German+Credit+Data\)](https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data))

get_best	<i>Get the best submissions per team and per metric.</i>
----------	--

Description

Get the best submissions per team and per metric.

Usage

```
get_best(history, metrics = names(metrics), test_name = "quiz")
```

Arguments

history	list of the submissions history per team as returned by compute_metrics
metrics	character vector. names of the metrics
test_name	string. name of the test set used: "quiz" or "test"

Value

get_best returns a named list with one member per metric. Each member is a data.frame where the rows are teams in decreasing order of performance and the columns are:

team	name of the team
n_submissions	total number of submissions
date	the date of the best submission
file	the file name of the best submission
<metric name>.quiz	the score obtained on the quiz subset
<metric name>.test	the score obtained on the test set
rank	the rank of the team
rank_diff	the rank difference is set to 0 temporarily.

get_data	<i>Get dataset value.</i>
----------	---------------------------

Description

Get dataset value.

Usage

```
get_data(name = "german", package = "rchallenge", envir = environment(),
  ...)
```

Arguments

name	string. name of the dataset.
package	string. name of the package to look in for dataset.
envir	the environment where the data should be loaded.
...	additional arguments to be passed to data .

Value

The value of the dataset

See Also

[data](#), [base](#)

glyphicon *Path to glyphicon image file.*

Description

Path to glyphicon image file.

Usage

```
glyphicon(name, path = system.file("glyphicons", package = "rchallenge"))
```

Arguments

name string. name of the glyphicon.
path string. folder of search.

Value

the path to the file.

html_img *html code for an image.*

Description

html code for an image.

Usage

```
html_img(file, width = "10px")
```

Arguments

file string. image file.
width string. width of display.

last_update	<i>Formatted last update date before deadline.</i>
-------------	--

Description

Formatted last update date before deadline.

Usage

```
last_update(deadline, format = "%d %b %Y %H:%M")
```

Arguments

deadline	POSIXct. deadline
format	string. see format.POSIXct

new_challenge	<i>Install a new challenge.</i>
---------------	---------------------------------

Description

Install a new challenge.

Usage

```
new_challenge(path = ".", out_rmdfile = "challenge.rmd",
  recursive = FALSE, overwrite = recursive, quiet = FALSE,
  showWarnings = FALSE, template = c("en", "fr"), data_dir = "data",
  submissions_dir = "submissions", hist_dir = "history",
  install_data = TRUE, baseline = "baseline", add_baseline = install_data,
  clear_history = overwrite, title = "Challenge", author = "",
  date = "", email = "EDIT_EMAIL@DOMAIN.com",
  date_start = format(Sys.Date(), "%d %b %Y"),
  deadline = paste(Sys.Date() + 90, "23:59:59"),
  data_list = data_split(get_data("german")))
```

Arguments

path	string. install path of the challenge (should be somewhere in your Dropbox).
out_rmdfile	string. name of the output R Markdown file.
recursive	logical. should elements of the path other than the last be created? see dir.create .
overwrite	logical. should existing destination files be overwritten? see file.copy .
quiet	logical. deactivate text output.
showWarnings	logical. should the warnings on failure be shown? see dir.create .

template	string. name of the template R Markdown script to be installed. Two choices are available: "en" (english) and "fr" (french).
data_dir	string. subdirectory of the data.
submissions_dir	string. subdirectory of the submissions. see store_new_submissions .
hist_dir	string. subdirectory of the history. see store_new_submissions .
install_data	logical. activate installation of the data files of the template challenge.
baseline	string. name of the team considered as the baseline.
add_baseline	logical. activate installation of baseline submission files of the template challenge.
clear_history	logical. activate deletion of the existing history folder.
title	string. title displayed on the webpage.
author	string. author displayed on the webpage.
date	string. date displayed on the webpage.
email	string. email of the challenge administrator.
date_start	string. start date of the challenge.
deadline	string. deadline of the challenge.
data_list	list with members train, test, y_test and ind_quiz such as returned by the data_split function.

Value

The path of the created challenge is returned.

Examples

```
path <- tempdir()
wd <- setwd(path)
# english version
new_challenge()
# french version
new_challenge(template = "fr")
setwd(wd)
unlink(path)
```

new_team

Create new teams submission folders in your challenge.

Description

Create new teams submission folders in your challenge.

Usage

```
new_team(..., path = ".", submissions_dir = "submissions", quiet = FALSE,
  showWarnings = FALSE)
```

Arguments

... strings. names of the team subdirectories.

path string. root path of the challenge. see [new_challenge](#).

submissions_dir string. subdirectory of the submissions. see [new_challenge](#).

quiet logical. deactivate text output.

showWarnings logical. should the warnings on failure be shown? see [dir.create](#).

Value

The paths of the created teams are returned.

Examples

```
path <- tempdir()
wd <- setwd(path)
new_challenge()
new_team("team_foo", "team_bar")
setwd(wd)
unlink(path)
```

plot_activity

Plot the density of submissions over time.

Description

Plot the density of submissions over time.

Usage

```
plot_activity(history, baseline = "baseline", col = 1:length(history),
  alpha.f = 0.7, bw = 3600 * 24, by = 4, xlab = "Date",
  ylab = "Submissions density", bty = "l", fg = "darkslategray",
  col.axis = fg, col.lab = fg, text.col = fg, ...)
```

Arguments

history	list of the submissions history per team as returned by compute_metrics
baseline	string. name of the team considered as the baseline that will not be plotted.
col	colors of the teams.
alpha.f	factor modifying the opacity alpha of colors; typically in [0,1].
bw	real. the smoothing bandwidth to be used by density in seconds.
by	real. height of the interval between two teams in nb of submissions.
xlab,ylab	axis labels. see title .
bty,fg,col.axis,col.lab	graphical parameters. see par .
text.col	the color used for the legend text. see legend .
...	further parameters passed to plot function.

Value

NULL

See Also[density](#)

`plot_history`*Plot the history of the scores of each team over time.*

Description

The best score of each team has a bold symbol.

Usage

```
plot_history(history, metric, test_name = "quiz", baseline = "baseline",
  col = 1:length(history), pch = rep(21:25, 100), by = 0.05,
  xlab = "Date", ylab = "Score", bty = "l", fg = "darkslategray",
  col.axis = fg, col.lab = fg, text.col = fg, ...)
```

Arguments

history	list of the submissions history per team as returned by compute_metrics
metric	string. name of the metric considered
test_name	string. name of the test set used: "quiz" or "test"
baseline	string. name of the team considered as the baseline. Its best score will be plotted as a constant and will not appear in the legend.
col	colors of the teams

pch	symbols of the teams
by	real. interval width of grid lines
xlab, ylab	axis labels. see title .
bty, fg, col.axis, col.lab	graphical parameters. see par .
text.col	the color used for the legend text. see legend .
...	further parameters passed to plot function.

Value

NULL

print_leaderboard *Format the leaderboard in Markdown.*

Description

Format the leaderboard in Markdown.

Usage

```
print_leaderboard(best, metric, test_name = "quiz", ...)
```

Arguments

best	list of the best submissions per team and per metric as returned by get_best .
metric	string. name of the metric considered
test_name	string. name of the test set used: "quiz" or "test"
...	further parameters to pass to kableer

Value

print_leaderboard returns a character vector of the table source code to be used in a Markdown document.

Note

Chunk option results='asis' has to be used

See Also

[kable](#)

print_readerr	<i>Print read errors.</i>
---------------	---------------------------

Description

Print read errors.

Usage

```
print_readerr(read_err = list())
```

Arguments

read_err list of read errors returned by [store_new_submissions](#)

Value

NULL

publish	<i>Publish your challenge R Markdown script to a html page.</i>
---------	---

Description

Publish your challenge R Markdown script to a html page.

Usage

```
publish(input = "challenge.rmd", output_file = NULL,
        output_dir = file.path("~/Dropbox/Public"), quiet = FALSE, ...)
```

Arguments

input	string. name of the R Markdown input file
output_file	output file. If NULL then a default based on the name of the input file is chosen.
output_dir	string. output directory. default="~/Dropbox/Public" so that the rendered page can easily be shared on the web with Dropbox.
quiet	logical. deactivate text output.
...	further arguments to pass to render .

Value

The compiled document is written into the output file, and the path of the output file is returned.

See Also[render](#)**Examples**

```

path <- tempdir()
wd <- setwd(path)
new_challenge()
outdir = tempdir()
publish(output_dir = outdir, output_options = list(self_contained = FALSE))
unlink(outdir)
setwd(wd)
unlink(path)

```

store_new_submissions *Store new submission files.*

Description

store_new_submissions copies new files from the subdirectories of submissions_dir to the respective subdirectories of hist_dir. Each team has a subdirectory. The copied files in hist_dir are prefixed with the last modification date for uniqueness. A file is considered new if its name and last modification time is new, i.e not present in hist_dir. The files must match pattern regular expression and must not throw errors or warnings when given to the valid_fun function.

Usage

```

store_new_submissions(submissions_dir = "submissions", hist_dir = "history",
  deadline, pattern = ".*\\.csv$", valid_fun)

```

Arguments

submissions_dir	string. directory of the submissions. contains one subdirectory per team
hist_dir	string. directory where to store the history of the submissions. contains one subdirectory per team
deadline	POSIXct. deadline time for submissions. The files with last modification date after the deadline are skipped.
pattern	string. regular expression that new submission files must match (with ignore.case=TRUE)
valid_fun	function that reads a submission file and throws errors or warnings if it is not valid.

Value

store_new_submissions returns a named list of errors or warnings caught during the process. Members named after the team names are lists with members named after the file that throws an error which contain the error object.

update_rank_diff	<i>Update the rank differences of the teams.</i>
------------------	--

Description

Update the rank differences of the teams.

Usage

```
update_rank_diff(best_new, best_old)
```

Arguments

best_new	list of the best submissions per team and per metric as returned by get_best .
best_old	old list of the best submissions per team and per metric.

Value

update_rank_diff returns the input list best_new with an updated column rank_diff for each metric.

Index

*Topic **datasets**

german, 4

base, 5

compute_metrics, 2, 4, 10

countdown, 3

data, 5

data_split, 3, 8

density, 10

dir.create, 7, 9

file.copy, 7

format.POSIXct, 7

german, 4

GermanCredit, 4

get_best, 4, 11, 14

get_data, 5

glyphicon, 6

html_img, 6

kable, 11

last_update, 7

legend, 10, 11

new_challenge, 7, 9

new_team, 8

par, 10, 11

plot, 10, 11

plot_activity, 9

plot_history, 10

print_leaderboard, 11

print_readerr, 12

publish, 12

render, 12, 13

store_new_submissions, 8, 12, 13

title, 10, 11

update_rank_diff, 14