Package ‘Rnumerai’

August 21, 2021

Title Interface to the Numerai Machine Learning Tournament API

Version 2.1.3

Description Routines to interact with the Numerai Machine Learning Tournament
API <https://numer.ai>. The functionality includes the ability to automatically download the
current tournament data, submit predictions, and to get information for your
user. General ’GraphQL’ queries can also be executed.

Depends R (>= 3.1)

License GPL-3

Encoding UTF-8

URL https://github.com/Omni-Analytics-Group/Rnumerai

BugReports https://github.com/Omni-Analytics-Group/Rnumerai/issues

RoxygenNote 7.1.1

Imports httr, lubridate, dplyr, tidyr, ggplot2, purrr

NeedsCompilation no

Author Omni Analytics Group [aut],
Eric Hare [cre]

Maintainer Eric Hare <eric@omnianalytics.io>

Repository CRAN

Date/Publication 2021-08-21 00:20:02 UTC

R topics documented:

   account_info .......................................................... 2
current_round ......................................................... 3
download_data ....................................................... 3
get_api_key .............................................................. 4
get_models .............................................................. 4
get_password ............................................................ 5
get_public_id .......................................................... 5
get_valid_data ......................................................... 6
account_info

Get information about your account

Description

Get information about your account

Usage

account_info()

Value

A list containing information about account

Examples

```r
## Not run:
ainfo <- account_info()
names(ainfo)
ainfo$Latest_Submission

## End(Not run)
```
**current_round**

Get current round and it’s closing time

**Usage**

```r
current_round(tournament = "Nomi")
```

**Arguments**

tournament The name of the tournament, Default is Nomi and is not case-sensitive

**Value**

Returns the current round number and it’s closing times

**Examples**

```r
## Not run:
current_round()
## End(Not run)
```

---

**download_data**

Function to download the Numerai Tournament data

**Description**

Function to download the Numerai Tournament data

**Usage**

```r
download_data(location = tempdir(), tournament = "NOMI")
```

**Arguments**

location The directory path in which to store the data
tournament The name of the tournament, Default is Nomi and is not case-sensitive. Since at the moment the datasets are same for all tournaments this parameter can be left blank.

**Value**

A list containing the training and tournament data objects
get_models

Examples

## Not run:
## Directory where data files and prediction files to be saved
## Put custom directory path or use the current working directory
data_dir <- tempdir()

## Download data set for current competition
data <- download_data(data_dir,tournament="Nomi")
data_train <- data$data_train
data_tournament <- data$data_tournament

## End(Not run)

get_api_key Gets the Numerai API key

Description

Gets the Numerai API key

Usage

get_api_key()

Value

Your Numerai API key, if set

Examples

## Not run:
get_api_key()

## End(Not run)

get_models Get models associated with your account

Description

Get models associated with your account

Usage

get_models()
get_password

Value

A list containing information about the models

Examples

```r
## Not run:
models <- get_models()

## End(Not run)
```

get_password  Gets the Numerai Password

Description

Gets the Numerai Password

Usage

get_password()

Value

Your Numerai Password, if set

Examples

```r
## Not run:
get_password()

## End(Not run)
```

get_public_id  Gets the Numerai Public ID

Description

Gets the Numerai Public ID

Usage

get_public_id()

Value

Your Numerai Public ID, if set
Examples

## Not run:
get_public_id()

## End(Not run)

---

get_valid_data  *Get the valid dataset for a particular metric*

### Description
Get the valid dataset for a particular metric

### Usage

```r
get_valid_data(username, metric, merge = FALSE, round_aggregate = TRUE)
```

### Arguments

- `username`
  - A vector of one or more usernames

- `metric`
  - Based on the metric selected, get the correct data

- `merge`
  - If TRUE, merge the results into a single username

- `round_aggregate`
  - If TRUE, aggregate the submission data by round

---

leaderboard  *Get Current leaderboard*

### Description
Get Current leaderboard

### Usage

```r
leaderboard()
```

### Value
List containing leaderboard

### Examples

## Not run:
leaderboard()

## End(Not run)
performance_distribution

Get the performance of the user as a distribution

Description

Get the performance of the user as a distribution

Usage

performance_distribution(
    username,
    metric,
    merge = FALSE,
    round_aggregate = TRUE
)

Arguments

username  A vector of one or more usernames
metric    A statistic, as a character vector.
merge     If TRUE, combine the usernames into a single result
round_aggregate  If TRUE, aggregate the submission data by round

performance_over_time  Get the performance of the user over time

Description

Get the performance of the user over time

Usage

performance_over_time(
    username,
    metric,
    merge = FALSE,
    outlier_cutoff = if (round_aggregate) 0 else 0.0125,
    round_aggregate = TRUE
)
Arguments

username  A vector of one or more usernames
metric    A statistic, as a character vector.
merge     If TRUE, combine the usernames into a single result
outlier_cutoff The absolute value above which points will be displayed
round_aggregate If TRUE, aggregate the submission data by round

release_nmr  Release NMR

Description

Release NMR

Usage

release_nmr(value, model_id = NULL, mfa_code = "", password = "")

Arguments

value The amount of NMR to release
model_id The id of the model with which to stake
mfa_code The mfa code
password Your password

Value

The transaction hash for release request

Examples

## Not run:
release_tx_hash <- release_nmr(value = 1)

## End(Not run)
round_stats  

Get Information for a Round Number

Description
Get Information for a Round Number

Usage
round_stats(round_number, tournament = "Nomi")

Arguments
    round_number  Round Number for which information to fetch
    tournament    The name of the tournament, Default is Nomi and is not case-sensitive

Value
List containing general round information

Examples
    ## Not run:
    round_stats(round_number=177)
    ## End(Not run)

run_query  

Function to run a raw GraphQL query on the API interface

Description
Function to run a raw GraphQL query on the API interface

Usage
run_query(query, id = get_public_id(), key = get_api_key())

Arguments
    query         The graphQL query to run on the API as a string in single quotes
    id            The public id of the Numerai application
    key           The Numerai API key
**set_api_key**

**Value**

The parsed json content returned from the request

**Examples**

```r
## Not run:
## Run Custom GraphQL code from R
custom_query <- "query queryname {
rounds (number:82) {
closeTime
}
}" 
run_query(query=custom_query)$data 
## End(Not run)
```

---

**set_api_key**

Sets the Numerai API key

**Description**

Sets the Numerai API key

**Usage**

```r
set_api_key(key)
```

**Arguments**

```r
key
```

**Value**

A boolean TRUE if the key was successfully set

**Examples**

```r
## Not run:
set_api_key("abcdefghijklmnop") 
## End(Not run)
```
**set_password**

Sets the Numerai Password

**Description**

Sets the Numerai Password

**Usage**

```r
set_password(pass)
```

**Arguments**

- `pass` The Numerai Password

**Value**

A boolean TRUE if the password was successfully set

**Examples**

```r
## Not run:
set_password("abcdefghijklmnop")
## End(Not run)
```

---

**set_public_id**

Sets the Numerai Public ID

**Description**

Sets the Numerai Public ID

**Usage**

```r
set_public_id(id)
```

**Arguments**

- `id` The Numerai Public ID

**Value**

A boolean TRUE if the ID was successfully set
Examples

```r
## Not run:
set_public_id("abcdefghijklmnop")

## End(Not run)
```

---

**stake_nmr**

*Stake NMR*

**Description**

Stake NMR

**Usage**

```r
stake_nmr(value, model_id = NULL, mfa_code = "", password = "")
```

**Arguments**

- `value`: The amount of NMR to stake
- `model_id`: The id of the model with which to stake
- `mfa_code`: The mfa code
- `password`: Your password

**Value**

The transaction hash for stake made

**Examples**

```r
## Not run:
stake_tx_hash <- stake_nmr(value = 1)

## End(Not run)
```
status_submission_by_id

Get information about a submission from a submission id

Description
Get information about a submission from a submission id

Usage
status_submission_by_id(sub_id)

Arguments
sub_id The id of the submission

Value
A list containing information about the given submission id

Examples
## Not run:
status_submission_by_id(submission_id)

## End(Not run)

submit_predictions Function to submit the Numerai Tournament predictions

Description
Function to submit the Numerai Tournament predictions

Usage
submit_predictions(
    submission,
    location = tempdir(),
    tournament = "Nomi",
    model_id = NULL,
    prefix = tournament
)
### Arguments

- **submission**: The data frame of predictions to submit. This should have two columns named "id" & "prediction"
- **location**: The location in which to store the predictions
- **tournament**: The name of the tournament, Default is Nomi and is not case-sensitive
- **model_id**: Target model UUID (required for accounts with multiple models)
- **prefix**: The prefix to use for the submission csv file

### Value

The submission id for the submission made

### Examples

```r
## Not run:
submission_id <- submit_predictions(submission_data,tournament="Nomi")
## End(Not run)
```

---

### summary_statistics

Get the summary statistics for

### Description

Get the summary statistics for

### Usage

`summary_statistics(username, dates = NULL, round_aggregate = TRUE)`

### Arguments

- **username**: A vector of one or more usernames
- **dates**: A vector of one or more dates to consider. If NULL, use all data
- **round_aggregate**: If TRUE, aggregate the submission data by round
**user_info**  
*Get information about your username*

Description  
Get information about your username  

Usage  
`user_info(model_id = NULL)`  

Arguments  
- `model_id`: The id of the model  

Value  
A list containing information about user  

Examples  
```r  
## Not run:  
uinfo <- user_info()  
names(uinfo)  
uinfo$Latest_Submission  
## End(Not run)  
```  

---  

**user_performance**  
*Get User Performance*

Description  
Get User Performance  

Usage  
`user_performance(user_name = "theomniacs")`  

Arguments  
- `user_name`: UserName for which performance metrics to get  

Value  
Get User Performance
Examples

```r
## Not run:
user_performance(user_name="theomniacs")
## End(Not run)
```

user_performance_data  
*Get the performance of the user over time*

Description

Get the performance of the user over time

Usage

```r
user_performance_data(username, dates = NULL, round_aggregate = TRUE)
```

Arguments

- **username**: A vector of one or more usernames
- **dates**: A vector of one or more dates to consider. If NULL, use all data
- **round_aggregate**: If TRUE, aggregate the submission data by round
Index

account_info, 2

current_round, 3

download_data, 3

get_api_key, 4
get_models, 4
get_password, 5
get_public_id, 5
get_valid_data, 6

leaderboard, 6

performance_distribution, 7
performance_over_time, 7

release_nmr, 8
round_stats, 9
run_query, 9

set_api_key, 10
set_password, 11
set_public_id, 11
stake_nmr, 12
status_submission_by_id, 13
submit_predictions, 13
summary_statistics, 14

user_info, 15
user_performance, 15
user_performance_data, 16