
numerapi Documentation

Release 2.3.3

uuazed

Jul 22, 2020

Contents

1	Installation	3
2	Usage	5
2.1	Python module	5
2.2	Command line interface	6
3	API Reference	9
3.1	Contents	9
4	Indices and tables	17

build passing

Automatically download and upload data for the Numerai machine learning competition.

This library is a Python client to the Numerai API. The interface is programmed in Python and allows downloading the training data, uploading predictions, and accessing user, submission and competitions information. It works for both, the main competition and the newer Numerai Signals competition.

If you encounter a problem or have suggestions, feel free to open an issue.

CHAPTER 1

Installation

```
pip install --upgrade numerapi
```


Numerapi can be used as a regular, importable Python module or from the command line.

Some actions (like uploading predictions or staking) require a token to verify that it is really you interacting with Numerai's API. These tokens consists of a `public_id` and `secret_key`. Both can be obtained by login in to Numer.ai and going to Account -> Custom API Keys. Tokens can be passed to the Python module as parameters or you can be set via environment variables (`NUMERAI_PUBLIC_ID` and `NUMERAI_SECRET_KEY`).

2.1 Python module

2.1.1 Usage example - main competition

```
import numerapi
# some API calls do not require logging in
napi = numerapi.NumerAPI(verbosity="info")
# download current dataset
napi.download_current_dataset(unzip=True)
# get competitions
all_competitions = napi.get_competitions()
# get current leaderboard
leaderboard = napi.get_leaderboard()
# check if a new round has started
if napi.check_new_round():
    print("new round has started within the last 24hours!")
else:
    print("no new round within the last 24 hours")

# provide api tokens
example_public_id = "somepublicid"
example_secret_key = "somesecretkey"
napi = numerapi.NumerAPI(example_public_id, example_secret_key)
```

(continues on next page)

(continued from previous page)

```

# upload predictions
model_id = api.get_models()['uuazed']
napi.upload_predictions("preds.csv", model_id=model_id)
# check submission status
napi.submission_status(model_id)
# increase your stake by 1.2 NMR
napi.stake_increase(1.2)

# convert results to a pandas dataframe
import pandas as pd
df = pd.io.json.json_normalize(napi.daily_user_performances("uuazed"), sep="-")

```

2.1.2 Usage example - Numerai Signals

```

import numerapi

napi = numerapi.SignalsAPI()
# get current leaderboard
leaderboard = napi.get_leaderboard()

# setup API with api tokens
example_public_id = "somepublicid"
example_secret_key = "somesecretkey"
napi = numerapi.SignalsAPI(example_public_id, example_secret_key)

# upload predictions
model_id = api.get_models()['uuazed']
napi.upload_predictions("preds.csv", model_id=model_id)

# get daily performance as pandas dataframe
import pandas as pd
df = pd.DataFrame(napi.daily_user_performances("uuazed"))

```

2.2 Command line interface

To get started with the cli interface, let's take a look at the help page:

```

$ numerapi --help
Usage: numerapi [OPTIONS] COMMAND [ARGS]...

  Wrapper around the Numerai API

Options:
  --help  Show this message and exit.

Commands:
  account                Get all information about your account!
  check-new-round        Check if a new round has started within...
  competitions            Retrieves information about all...
  current-round          Get number of the current active round.
  daily-submissions-performances  Fetch daily performance of a user's...
  daily-user-performances  Fetch daily performance of a user.

```

(continues on next page)

(continued from previous page)

dataset-url	Fetch url of the current dataset.
download-dataset	Download dataset for the current active...
leaderboard	Get the leaderboard.
models	Get map of account models!
payments	List all your payments
profile	Fetch the public profile of a user.
stake-decrease	Decrease your stake by `value` NMR.
stake-drain	Completely remove your stake.
stake-get	Get stake value of a user.
stake-increase	Increase your stake by `value` NMR.
stakes	List all your stakes.
submission-filenames	Get filenames of your submissions
submission-ids	Get dict with username->submission_id...
submission-status	checks the submission status
submit	Upload predictions from file.
tournament-name2number	Translate tournament name to tournament...
tournament-number2name	Translate tournament number to tournament...
tournaments	Get all tournaments.
transactions	List all your deposits and withdrawals.
user	Get all information about you! DEPRECATED...
user-activities	Get user activities (works for all users!)
v1-leaderboard	Retrieves the leaderboard for the given...
version	Installed numerapi version.

Each command has it's own help page, for example:

```
$ numerapi submit --help
Usage: numerapi submit [OPTIONS] PATH

    Upload predictions from file.

Options:
  --tournament INTEGER  The ID of the tournament, defaults to 1
  --model_id TEXT       An account model UUID (required for accounts with
                        multiple models
  --help                Show this message and exit.
```


Checkout the [detailed API docs](#) to learn about all available methods, parameters and returned values.

3.1 Contents

3.1.1 Changelog

Notable changes to this project.

[2.3.3] - 2020-07-22

- get Numerai compute id if available and pass it along during predictions upload

[2.3.2] - 2020-07-21

- Signals: added `ticker_universe` to get the list of accepted tickers
- `submission_status` no longer needs (and accepts) a `submission_id`. It automatically uses the last submission associated with a model

[2.3.1] - 2020-05-06 - “Signals”

- fix Signals submission upload (#25)

[2.3.0] - 2020-05-06 - “Signals”

- added API for Numerai Signals
- refactor codebase

- more tests

[2.2.4] - 2020-05-11

- Remove required `model_id` annotation for submissions status lookups so that `None` can be passed
- Use consistent `modelId` in query spec
- Update doc examples

[2.2.2] - 2020-05-09

- fix `submission_status` for multi model accounts

[2.2.0] - 2020-04-17

- no more Python2 support
- added type hints
- add `get_account` to return private account information and deprecates `get_user` (#23)
- incorporates updates to the Numerai tournament API in anticipation of the rollout of a new account system with multi-model support (#23)

[2.1.6] - 2020-04-08

- add `rolling_score_rep` to `daily_user_performances` and `get_leaderboard`
- deprecate `reputation` in `daily_user_performances` and `get_leaderboard`

[2.1.5] - 2020-04-03

- added `payoutPending` and `payoutSettled` to `get_leaderboard` (#21)
- added `sumDeltaCorrelation`, `finalCorrelation`, `payoutPending` and `payoutSettled` to `daily_user_performances` (#21)

[2.1.4] - 2020-03-30 - “Spring cleanup”

- added “`sharpe`”, “`feature exposure`” and “`correlation with example predictions`” to `submission_status`
- remove deprecated `check_submission_successful`
- added `bio` and `totalStake` to `public_user_profile`
- remove deprecated `get_rankings`

[2.1.3] - 2020-03-30

- fix `get_user_activities`
- remove deprecated `get_staking_leaderboard`, `get_nmr_prize_pool`
- added `mmc` and `correlationWithMetamodel` to `daily_submissions_performances`

[2.1.2] - 2019-11-30

- fix staking after recent changes to the GraphQL backend

[2.1.1] - 2019-11-23

- add `round_details`, returning correlation scores of all users for the round

[2.1.0] - 2019-11-15

- add some more details to `get_leaderboard`
- adapt to changes in Numerai's staking API

[2.0.1] - 2019-10-28

- fix `stake_set`

[2.0.0] - 2019-10-23

- add v2 version of `get_leaderboard`
- add `stake_get` & `stake_set`
- add `stake_increase`, `stake_decrease` & `stake_drain`
- add `public_user_profile`
- add `daily_user_performances`
- add `daily_submissions_performances`
- remove v1 staking
- remove `get_staking_cutoff` - no longer relevant
- old `get_leaderboard` renamed to `get_v1_leaderboard`
- add v2-style staking to cli interface
- update documentation

[1.6.2] - 2019-07-31

- remove phone number and bonus fetching (#16)

[1.6.1] - 2019-07-12

- fix downloading dataset for tournaments > 1
- add `validationCorrelation` and `liveCorrelation` to all relevant places
- remove `validationAuroc` and `validationLogloss` from `submission_status`

[1.6.0] - 2019-07-10

- default to tournament 8 `katzuagi`
- update docstring
- added `reputationPayments`,

[1.5.5] - 2019-06-13

- include `otherUsdIssuances` and `phoneVerificationBonus` to `get_payments`
- add datetime information (`insertedAt`) to `get_transactions`

[1.5.4] - 2019-05-30

- return new `reputation` as announced by numerai on 2019-05-29 in `get_rankings`

[1.5.3] - 2019-05-23

- fix `setup.py` to make it work with the latest `twine` version

[1.5.2] - 2019-05-22

- add NMR returned information to `get_leaderboard` - useful for partial burns

[1.5.1] - 2019-04-14

- fix `get_staking_cutoff` for rounds ≥ 154

[1.5.0] - 2019-04-03

- tests: start testing the cli interface
- cli: fix `version` command on Python2.7
- added `liveAuroc` and `validationAuroc` to `get_leaderboard`
- added `liveAuroc` and `validationAuroc` to `get_staking_leaderboard`
- added `liveAuroc` and `validationAuroc` to `get_user_activities`
- added `validationAuroc` to `submission_status`
- added `ruleset` to `get_competitions`
- added `phoneNumber` and `country` to `get_user`
- remove consistency check from `test_check_submission_successful`

[1.4.6] - 2019-03-30

- remove total payments from leaderboard query (#13)
- fix `get_staking_leaderboard`

[1.4.5] - 2019-03-05

- `get_tournaments` now allows to filter for active tournaments only
- CLI: `tournaments` gained `active_only` / `all` flags to get all or only the active tournaments

[1.4.4] - 2019-02-17

- remove timeout completely to fix upload issues

[1.4.3] - 2019-02-17

- increase default timeout to 20s
- better error handling

[1.4.2] - 2019-02-10

- **`get_staking_cutoff` now gets the cutoff values via the api, instead of** doing it's own computation
- compatibility with `click` version ≥ 7.0

[1.4.1] - 2019-02-10

- handle connection errors more gracefully (#11)
- pin minimum version of `tqdm` to (hopefully) prevent an exception (#12)
- travis: test against Python 3.7

[1.4.0] - 2018-11-16

- added `burned` to `get_user_activities`
- docs: fixed typos + improved example
- `validation_logloss` -> `validationLogloss`, to follow numerai's docs
- remove everything `originality` related

[1.3.0] - 2018-08-09

- added `get_staking_cutoff` to compute staking cutoff for a given round and tournament.
- added `get_nmr_prize_pool` to get the NMR prize pool for a given round and tournament.

[1.2.1] - 2018-08-05

- removed `filename` from `get_user_activities`, no longer supported.
- rename `get_submission_filename` to `get_submission_filenames`
- `get_submission_filenames` now only works for the authorized user. It allows to get ones submission filenames, optionally filtered by `round_num` and tournament.

[1.2.0] - 2018-08-03

- added `get_rankings`, which gives access to numerai's global leaderboard
- added `get_user_activities`, that allows to see each user's submission and staking activity
- added `get_submission_filename` to get the submission filename for any user, tournament & round number combination
- added `prizePoolNmr`, `prizePoolUsd` and number of participants to the `get_competitions` endpoint
- ensure functionality of command line interface is in sync

[1.1.1] - 2018-06-06

- added `get_tournaments`
- added `tournament_name2number` and `tournament_number2name` to translate between tournament numbers and names

[1.1.0] - 2018-05-24

- added numerapi command line interface
- allow passing public ID and secret key via environment variables

[1.0.1] - 2018-05-17

- added `stakeResolution` information to `get_leaderboard`
- added badge for read the docs to README

[1.0.0] - 2018-04-25

- publish README as `long_description` on pypi
- fixed `get_transactions` after API change on Numerai's side
- added proper docstrings to all public methods, using Google Style as described at http://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html
- added examples for all public methods
- added documentation on readthedocs: <http://numerapi.readthedocs.io>

[0.9.1] - 2018-04-22

- add `tournamentId` to `get_stakes`
- fixed `stake` after API change on Numerai's side

[0.9.0] - 2018-04-13

- support tournament parameter for various endpoints. Numer.ai is planning to run more than one tournament at a time. This change makes numerapi ready for that.
- minor code cleanup

[0.8.3] - 2018-04-07

- don't query Numerai's API if the action requires an auth token, but there is none provided
- more & improved tests (test coverage now > 90%)
- consistency threshold moved to 58, following the latest rule change

[0.8.2] - 2018-03-09

- use `decimal.Decimal` instead of floats to avoid rounding errors (#3)
- optional flag to turn of tqdm's progress bars (#4)
- update `check_submission_successful` to recent rule changes (originality no longer required)
- update documentation

[0.8.1] - 2018-01-27

- import NumerAPI class to toplevel. now `from numerapi import NumerAPI` works
- added `get_dataset_url`
- more & improved tests

[0.8.0] - 2018-01-06

- added `check_new_round` to check if a new round has started
- added `check_submission_successful` to check if the last submission passes concordance, originality and consistency
- return proper Python data types, for example the NMR amounts are now floats and no longer strings
- show progress bar while downloading dataset
- general code cleanup & more tests

[0.7.1] - 2017-12-29

- fix import issues (py2 vs py3)

[0.7.0] - 2017-12-29

- convert datetime strings to proper Python datetime objects
- only append .zip to downloaded dataset if zip=True
- use round_number instead of date in default download filename
- setup travis to run test automatically
- run tests with different Python versions (2.7, 3.5 and 3.6)
- test coverage reports via codecov.io

[0.6.3] - 2017-12-20

- complete rewrite to adapt to Numerai's API switch to GraphQL
- update documentation and example
- added staking via API - stake
- added get_staking_leaderboard
- allow passing desired filename to data download
- allow custom API calls - raw_query
- started a test suite
- moved numerapi to it's new home (<https://github.com/uuazed/numerapi>)
- make numerapi available on pypi (<https://pypi.org/project/numerapi>)
- rename package from NumerAPI to all-lowercase numerapi

3.1.2 License

MIT License

Copyright (c) 2016

Permission **is** hereby granted, free of charge, to **any** person obtaining a copy of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies **or** substantial portions of the Software.

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`