NiaClass

Release 0.2.2

Luka Pečnik

Jul 25, 2025

USER DOCUMENTATION

1 Getting Started **3**
- 1.1 Examples . . . 3
- 1.2 Optimization Algorithms . . . 3

2 Installation **5**
- 2.1 Setup development environment . . . 5
  - 2.1.1 Requirements . . . 5
  - 2.1.2 Development dependencies . . . 5

3 Testing **7**

4 Documentation **9**

5 API **11**
- 5.1 `niaclass` . . . 11

6 Contributing to NiaClass **13**
- 6.1 Code of Conduct . . . 13
- 6.2 How Can I Contribute? . . . 13
  - 6.2.1 Reporting Bugs . . . 13
  - 6.2.2 Suggesting Enhancements . . . 13
  - 6.2.3 Pull requests . . . 13

7 Contributor Covenant Code of Conduct **15**
- 7.1 Our Pledge . . . 15
- 7.2 Our Standards . . . 15
- 7.3 Enforcement Responsibilities . . . 15
- 7.4 Scope . . . 16
- 7.5 Enforcement . . . 16
- 7.6 Enforcement Guidelines . . . 16
  - 7.6.1 1. Correction . . . 16
  - 7.6.2 2. Warning . . . 16
  - 7.6.3 3. Temporary Ban . . . 16
  - 7.6.4 4. Permanent Ban . . . 16
- 7.7 Attribution . . . 17

8 References **19**

Python Module Index **21**

Index **23**

NiaClass is a framework for solving classification tasks using nature-inspired algorithms. The framework is written fully in Python. Its goal is to find the best possible set of classification rules for the input data using the NiaPy framework, which is a popular Python collection of nature-inspired algorithms. The NiaClass classifier supports numerical and categorical features.

- **Free software:** MIT license
- **Github repository:** https://github.com/lukapecnik/NiaClass
- **Python versions:** 3.7.x, 3.8.x, 3.9.x

The main documentation is organized into a couple of sections:

- *User Documentation*
- *Developer Documentation*
- *About*

CHAPTER

ONE

GETTING STARTED

This section is going to show you how to use the NiaClass framework. First install NiaClass package using the following command:

```
pip3 install niaclass
```

After the successful installation you are ready to run your first example.

1.1 Examples

You can find usage examples here.

1.2 Optimization Algorithms

For the list of available optimization algorithms please see the NiaPy's documentation.

CHAPTER

TWO

INSTALLATION

2.1 Setup development environment

2.1.1 Requirements

- Poetry: https://python-poetry.org/docs/

After installing Poetry and cloning the project from GitHub, you should run the following command from the root of the cloned project:

```
$ poetry install
```

All of the project's dependencies should be installed and the project ready for further development. **Note that Poetry creates a separate virtual environment for your project.**

2.1.2 Development dependencies

List of NiaClass' dependencies:

Package	Version	Platform
niapy	^2.0.4	All
pandas	^1.5.0	All
numpy	^1.26.0	All
scikit-learn	^1.2.0	All

List of development dependencies:

Package	Version	Platform
coveralls	^3.0.1	Any
Sphinx	^3.5.1	Any
sphinx-rtd-theme	^0.5.1	Any
autoflake	^1.4	Any
black	^21.5b1	Any
pre-commit	^2.13.0	Any
pytest	^6.2	Any

CHAPTER

THREE

TESTING

Before making a pull request, if possible provide tests for added features or bug fixes.

We have an automated building system which also runs all of provided tests. In case any of the test cases fails, we are notified about failing tests. Those should be fixed before we merge your pull request to master branch.

For the purpose of checking if all test are passing localy you can run following command:

```
$ poetry run coverage run --source=niaclass -m unittest discover -b
```

If all tests passed running this command it is most likely that the tests would pass on our build system too.

CHAPTER

FOUR

DOCUMENTATION

To locally generate and preview documentation run the following command in the project root folder:

```
$ poetry run sphinx-build ./docs ./docs/_build
```

If the build of the documentation is successful, you can preview the documentation in the docs/_build folder by clicking the `index.html` file.

CHAPTER

FIVE

API

This is the NiaClass API documentation, auto generated from the source code.

5.1 niaclass

class niaclass.**NiaClass**(*pop_size=90*, *num_evals=5000*, *score_func_name='accuracy'*, *algo='FireflyAlgorithm'*, ***kwargs*)

Bases: `object`

Implementation of NiaClass classifier.

Date:
2021

Author:
Luka Pečnik

Reference:
The implementation is based on the following article: Iztok Fister Jr., Iztok Fister, Dušan Fister, Grega Vrbančič, Vili Podgorelec. On the potential of the nature-inspired algorithms for pure binary classification. In. Computational science - ICCS 2020 : 20th International Conference, Proceedings. Part V. Cham: Springer, pp. 18-28. Lecture notes in computer science, 12141, 2020

License:
MIT

Attributes:
__pop_size (int): Number of individuals in the fitting process. __num_evals (int): Maximum evaluations in the fitting process. __score_func_name (Optional(str)): Used score function. __algo (str): Name of the optimization algorithm to use. __rules (Dict[any, Iterable[_Rule]]): Best set of rules found in the optimization process. __history (Iterable[_Rule]): History of all identified rules.

fit(*x*, *y*)

Fit NiaClass.

Arguments:
x (pandas.core.frame.DataFrame): n samples to classify. y (pandas.core.series.Series): n classes of the samples in the x array.

Returns:
None

predict(*x*)

Predict class for each sample (row) in x.

Arguments:
x (pandas.core.frame.DataFrame): n samples to classify.

Returns:
Iterable[any]: n predicted classes.

`print_all_rules()`

Print the rules found during the optimization process.

`print_best_rules()`

Print the best set of rules found during the optimization process.

CHAPTER

SIX

CONTRIBUTING TO NIACLASS

First off, thanks for taking the time to contribute!

6.1 Code of Conduct

This project and everyone participating in it is governed by the *Contributor Covenant Code of Conduct*. By participating, you are expected to uphold this code. Please report unacceptable behavior to lukapecnik96@gmail.com.

6.2 How Can I Contribute?

6.2.1 Reporting Bugs

Before creating bug reports, please check existing issues list as you might find out that you don't need to create one. When you are creating a bug report, please include as many details as possible in the issue template.

6.2.2 Suggesting Enhancements

Open new issue using the feature request template.

6.2.3 Pull requests

Fill in the pull request template and make sure your code is documented.

CHAPTER

SEVEN

CONTRIBUTOR COVENANT CODE OF CONDUCT

7.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

7.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

7.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

7.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

7.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement at lukapecnik96@gmail.com. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

7.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

7.6.1 1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

7.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

7.6.3 3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

7.6.4 4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

7.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at https://www.contributor-covenant.org/faq. Translations are available at https://www.contributor-covenant.org/translations.

CHAPTER

EIGHT

REFERENCES

[1] Iztok Fister Jr., Iztok Fister, Dušan Fister, Grega Vrbančič, Vili Podgorelec. [On the potential of the nature-inspired algorithms for pure binary classification](http://www.iztok-jr-fister.eu/static/publications/267.pdf). In. Computational science - ICCS 2020 : 20th International Conference, Proceedings. Part V. Cham: Springer, pp. 18-28. Lecture notes in computer science, 12141, 2020

PYTHON MODULE INDEX

n

niaclass, **??**

INDEX

M

module
    niaclass, 1

N

niaclass
    module, 1