DeepSurvK Documentation

Release 0.2.2

Arturo Moncada-Torres

Jul 25, 2022

CONTENTS:

CHAPTER

ONE

INSTALLATION

1.1 Stable release

To install DeepSurvK, run this command in your terminal:

```
$ pip install deepsurvk
```

If you want to install a specific version, you can run the command:

```
$ pip install deepsurvk==0.2.0
```

However, I recommend to always install the most recent (stable) release, since it probably has important bug fixes.

If you don't have pip installed, this Python installation guide can guide you through the process.

1.2 From sources

The sources for DeepSurvK can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/arturomoncadatorres/deepsurvk
```

Or download the tarball:

```
$ curl -OJL https://github.com/arturomoncadatorres/deepsurvk/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER

TWO

USAGE

To use DeepSurvK in a project:

```
import deepsurvk
```

CHAPTER

THREE

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at https://github.com/arturomoncadatorres/deepsurvk/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

3.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

3.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

3.1.4 Write Documentation

DeepSurvK could always use more documentation, whether as part of the official DeepSurvK docs, in docstrings, or even on the web in blog posts, articles, and such.

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/arturomoncadatorres/deepsurvk/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

3.2 Get Started!

Ready to contribute? Here's how to set up *deepsurvk* for local development.

1. Fork the *deepsurvk* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/deepsurvk.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv deepsurvk
$ cd deepsurvk/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 deepsurvk tests
$ python setup.py test or pytest
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

3.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/arturomoncadatorres/deepsurvk/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ pytest tests.test_deepsurvk
```

3.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER

FOUR

CREDITS

4.1 Development Lead

- Arturo Moncada-Torres <arturomoncadatorres@gmail.com>

4.2 Contributors

Special thanks to my students Foppe Crezee and Femke Janssen for being beta users of DeepSurvK.

CHAPTER

FIVE

HISTORY

5.1 0.2.2 (2021-08-10)

- Minor bug fixes, including missing data files in release.

5.2 0.2.1 (2021-07-12)

- Added randomized search (alternative to grid search) for hyperparameter optimization

5.3 0.2.0 (2020-11-05)

- Improved DeepSurvK parameter definition (from function arguments to dictionary)
- Added basic parameter optimization (grid search), including example notebook

5.4 0.1.2 (2020-10-08)

- Fixed import of packages

5.5 0.1.1 (2020-09-06)

- Fixed missing requirements

5.6 0.1.0 (2020-07-17)

- First release on PyPI.

CHAPTER

SIX

API REFERENCE

6.1 datasets

deepsurvk.datasets.**load_metabric**(*partition='complete'*, ***kwargs*)

Data from the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC), which uses gene and protein expression profiles to determine new breast cancer subgroups

It consists of clinical features of 1980 patients, of which 57.72% have an observed death due to breast cancer with a median survival time of 116 months. However, the file only contains data of 1904 patients.

For more information, see[1] as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

  Possible values are:

  - `complete` - The whole dataset (default)
  - `training` or `train` - Training partition as used in the original DeepSurv
  - `testing` or `test` - Testing partition as used in the original DeepSurv

- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

Returns

- X - Features
- Y - Target variable
- E - Event variable

Return type tuple of pandas DataFrames

[1] Curtis, Christina, et al. "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups." Nature 486.7403 (2012): 346-352.

References

deepsurvk.datasets.**load_rgbsg**(*partition='complete'*, ***kwargs*)

The training partition belongs to the Rotterdam tumor bank dataset[2]. It contains records of 1546 patients with node-positive breast cancer. Nearly 90% of the patients have an observed death time.

The testing partitiong belongs to the German Breast Cancer Study Group (GBSG)[3]. It contains records for 686 patients (of which 56 % are censored) in a randomized clinical trial that studied the effects of chemotherapy and hormone treatment on survival rate.

For more information, see[2] and[3], as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

  Possible values are:

  - `complete` - The whole dataset (default)
  - `training` or `train` - Training partition as used in the original DeepSurv
  - `testing` or `test` - Testing partition as used in the original DeepSurv

- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

Returns

- X - Features
- Y - Target variable
- E - Event variable

Return type tuple of pandas DataFrames

References

deepsurvk.datasets.**load_simulated_gaussian**(*partition='complete'*, ***kwargs*)

Synthetic data with a Gaussian (non-linear) log-risk function.

For more information, see[4] as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

  Possible values are:

[2] Foekens, John A., et al. "The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients." Cancer research 60.3 (2000): 636-643.

[3] Schumacher, M., et al. "Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German Breast Cancer Study Group." Journal of Clinical Oncology 12.10 (1994): 2086-2093.

[4] Katzman, Jared L., et al. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network." BMC medical research methodology 18.1 (2018): 24.

  - `complete` - The whole dataset (default)
  - `training` or `train` - Training partition as used in the original DeepSurv
  - `testing` or `test` - Testing partition as used in the original DeepSurv
- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

Returns

- X - Features
- Y - Target variable
- E - Event variable

Return type tuple of pandas DataFrames

References

`deepsurvk.datasets.`**`load_simulated_linear`**(*partition='complete'*, ***kwargs*)

Synthetic data with a linear log-risk function.

For more information, see[5] as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

  Possible values are:

  - `complete` - The whole dataset (default)
  - `training` or `train` - Training partition as used in the original DeepSurv
  - `testing` or `test` - Testing partition as used in the original DeepSurv
- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

Returns

- X - Features
- Y - Target variable

[5] Katzman, Jared L., et al. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network." BMC medical research methodology 18.1 (2018): 24.

- E - Event variable

Return type tuple of pandas DataFrames

References

deepsurvk.datasets.**load_simulated_treatment**(*partition='complete'*, ***kwargs*)

Synthetic data similar to the simulated_gaussian one, with an additional column representing treatment.

For more information, see[6] as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

  Possible values are:

  - `complete` - The whole dataset (default)
  - `training` or `train` - Training partition as used in the original DeepSurv
  - `testing` or `test` - Testing partition as used in the original DeepSurv

- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

Returns

- X - Features
- Y - Target variable
- E - Event variable

Return type tuple of pandas DataFrames

References

deepsurvk.datasets.**load_support**(*partition='complete'*, ***kwargs*)

Data from the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT), which studied the survival time of seriously ill hospitalized adults.

Originally, it consists of 14 clinical features of 9105 patients. However, patients with missing features were dropped, leaving a total of 8873 patients.

For more information, see[7] as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

[6] Katzman, Jared L., et al. "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network." BMC medical research methodology 18.1 (2018): 24.

[7] Knaus, William A., et al. "The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults." Annals of internal medicine 122.3 (1995): 191-203.

Possible values are:

- `complete` - The whole dataset (default)
- `training` or `train` - Training partition as used in the original DeepSurv
- `testing` or `test` - Testing partition as used in the original DeepSurv

- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

Returns

- X - Features
- Y - Target variable
- E - Event variable

Return type tuple of pandas DataFrames

References

`deepsurvk.datasets.`**`load_whas`**(*partition='complete'*, ***kwargs*)

Data from the Worcester Heart Attack Study (WHAS), which investigates the effects of a patient's factors on acute myocardial infraction (MI) survival.

It consists of 1638 observations and 5 features: age, sex, body-mass-index (BMI), left heart failure complications (CHF), and order of MI (MIORD).

For more information, see[8] as well as the accompanying README.

Parameters

- **partition** (*string*) – Partition of the data to load.

  Possible values are:

  - `complete` - The whole dataset (default)
  - `training` or `train` - Training partition as used in the original DeepSurv
  - `testing` or `test` - Testing partition as used in the original DeepSurv

- **data_type** (*string*) – Data type of the data.

  Possible values are:

  - `pandas` or `pd` or `dataframe` or `df`- pandas DataFrame (default)
  - `numpy` or `np` - NumPy array

  Note: NumPy is supported as an option, but DeepSurvK is built with pandas in mind.

[8] Hosmer Jr, David W., Stanley Lemeshow, and Susanne May. Applied survival analysis: regression modeling of time-to-event data. Vol. 618. John Wiley & Sons, 2011.

Returns

- X - Features
- Y - Target variable
- E - Event variable

Return type tuple of pandas DataFrames

References

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

INDEX