

---

# An implementation of survival trees

---

## 1 Description

The main objective in survival analysis [1, 2, 3] is the estimation of the distribution of the time it takes for an event to happen given the explanatory variables. This estimation is commonly accomplished from the analysis of survival data. The central role in survival analysis is played by the *time until event* which, depending on the context of application, is called survival time, failure time, or event time. Survival analysis provides models and statistical procedures to estimate the time until event.

Survival analysis is also used to study whether there are subgroups of patients that differ in survival, if the differences are due to the different treatments they receive, and whether it is possible can predict survival times. Recently, a methods from machine learning have adapted to work in the survival analysis scenario. Decision trees [4, 5, 6], random forests [7, 8, 9], and support vector machines [10] have investigated within this scope.

## 2 Objectives

The goal of the project is to implement a survival tree learning algorithm in Python. The program will receive as input survival dataset and should output the tree.

The student should: 1) Implement the program that learn the survival tree. 2) Generate test sets to validate the program. 3) Visualize the survival trees learned.

As in other projects, a report should describe the characteristics of the design, implementation, and results. A Jupyter notebook should include calls to the implemented function that illustrate the way it works.

## 3 Suggestions

- Read an introduction to survival analysis, e.g., <http://www.biecek.pl/statystykaMedyczna/Stevens>
- Read bibliography on survival trees [4, 5, 6].
- Read the following discussion on how to adapt an ordinary decision tree representation to learn survival trees: <https://www.benkuhn.net/survival-trees>
- Inspect available Python packages for survival analysis: scikit-survival <https://k-d-w.org/node/97> and lifelines <https://lifelines.readthedocs.io/en/latest/>
- Use survival data available for these implementations to test the implemented algorithm, ex: <https://nbviewer.jupyter.org/github/sebp/scikit-survival/blob/master/examples/>
- Implementations can use any other Python library.

## References

- [1] Taane G. Clark, Michael J. Bradburn, Sharon B. Love, and Douglas G. Altman. Survival analysis part i: basic concepts and first analyses. *The British Journal of Cancer*, 89(2):232, 2003.

- [2] David R Cox. Regression models and life-tables. In *Breakthroughs in statistics*, pages 527–541. Springer, 1992.
- [3] David G Kleinbaum and Mitchel Klein. *Survival analysis: a self-learning text*. Springer Science & Business Media, 2006.
- [4] Imad Bou-Hamad, Denis Larocque, and Hatem Ben-Ameur. A review of survival trees. *Statistics Surveys*, 5:44–71, 2011.
- [5] Matthias Schmid, Helmut Küchenhoff, Achim Hoerauf, and Gerhard Tutz. A survival tree method for the analysis of discrete event times in clinical and epidemiological studies. *Statistics in Medicine*, 35(5):734–751, 2016.
- [6] Asanao Shimokawa, Yohei Kawasaki, and Etsuo Miyaoka. Comparison of splitting methods on survival tree. *The International Journal of Biostatistics*, 11(1):175–188, 2015.
- [7] Ulla B Mogensen, Hemant Ishwaran, and Thomas A Gerds. Evaluating random forests for survival analysis using prediction error curves. *Journal of statistical software*, 50(11):1, 2012.
- [8] Marvin N Wright, Theresa Dankowski, and Andreas Ziegler. Random forests for survival analysis using maximally selected rank statistics. *CoRR*, abs/1605.03391, 2016.
- [9] Joeri Ruysinck, Joachim van der Herten, Rein Houthoofd, Femke Ongenae, Ivo Couckuyt, Bram Gadeyne, Kirsten Colpaert, Johan Decruyenaere, Filip De Turck, and Tom Dhaene. Random survival forests for predicting the bed occupancy in the intensive care unit. *Computational and mathematical methods in medicine*, 2016, 2016.
- [10] Sebastian Pölsterl, Nassir Navab, and Amin Katouzian. An efficient training algorithm for kernel survival support vector machines. *CoRR*, abs/1611.07054, 2016.