

[Table of contents \(../toc.ipynb\)](#)

## Additional Packages

This chapter will be a short collection of additional very useful Python packages, so that you know about them.

# Speed up Python code

One problem often discussed under more advanced Python users is that Python can be slow compared with compiled languages like C and C++. However, there are multiple options to speed up Python.



- [Cython \(https://cython.org/\)](https://cython.org/) is a packages which can compile Python code into C code. Add to this, the cython language can be used to write C code in a pythonic style.
- [ScyPy lectures notes: Interfacing C \(http://scipy-lectures.org/advanced/interfacing\\_with\\_c/interfacing\\_with\\_c.html\)](http://scipy-lectures.org/advanced/interfacing_with_c/interfacing_with_c.html) explain how to use native Python-C-API, CTypes, Swig, to include C code in your Python project.
- [Dask \(https://dask.org/\)](https://dask.org/) is a parallel computing package. With Dask, you can scale and distribute Python code on clusters for instance.
  - There is an extensive tutorial on github [https://github.com/dask/dask-tutorial \(https://github.com/dask/dask-tutorial\)](https://github.com/dask/dask-tutorial).
- [Numba \(https://numba.pydata.org/\)](https://numba.pydata.org/) translates some part of Python code and numpy code into fast machine code.

# Additional machine learning packages

Python is very often used for machine learning tasks. Hence, there are numerous top notch machine learning packages available.

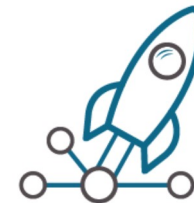
## Deep learning

Keras (<https://keras.io/>)  
is a user friendly library  
to build neural networks  
on top of Theano



(<http://www.deeplearning.net/software/theano/>) or TensorFlow  
(<https://www.tensorflow.org/>) deeplearning backends.

# Probalistic programming



- Bayesian estimation of model parameters, probalistic machine learning, Gaussian processes, ... All these tasks can be conducted with [PyMC3 package](https://docs.pymc.io/) (<https://docs.pymc.io/>).
  - Compared with common frequentist approach, Bayesian estimation places a prior distribution on unknown model paremeters and computes the a posterioy distribution of these parameters given some data.
  - Hence, beyond point estimation, a full distribution is result of Bayesian estimation.
  - Probalistic inference is introduced in this tutorial [https://docs.pymc.io/notebooks/api\\_quickstart.html](https://docs.pymc.io/notebooks/api_quickstart.html) ([https://docs.pymc.io/notebooks/api\\_quickstart.html](https://docs.pymc.io/notebooks/api_quickstart.html)).

# Statistical models

Finally, a great library for statistical modeling is [statsmodels](https://www.statsmodels.org/stable/index.html) (<https://www.statsmodels.org/stable/index.html>).

Statsmodels provides, linear regression, robust regression, statistical tests, recursive estimators, time series modelling, and much more.

