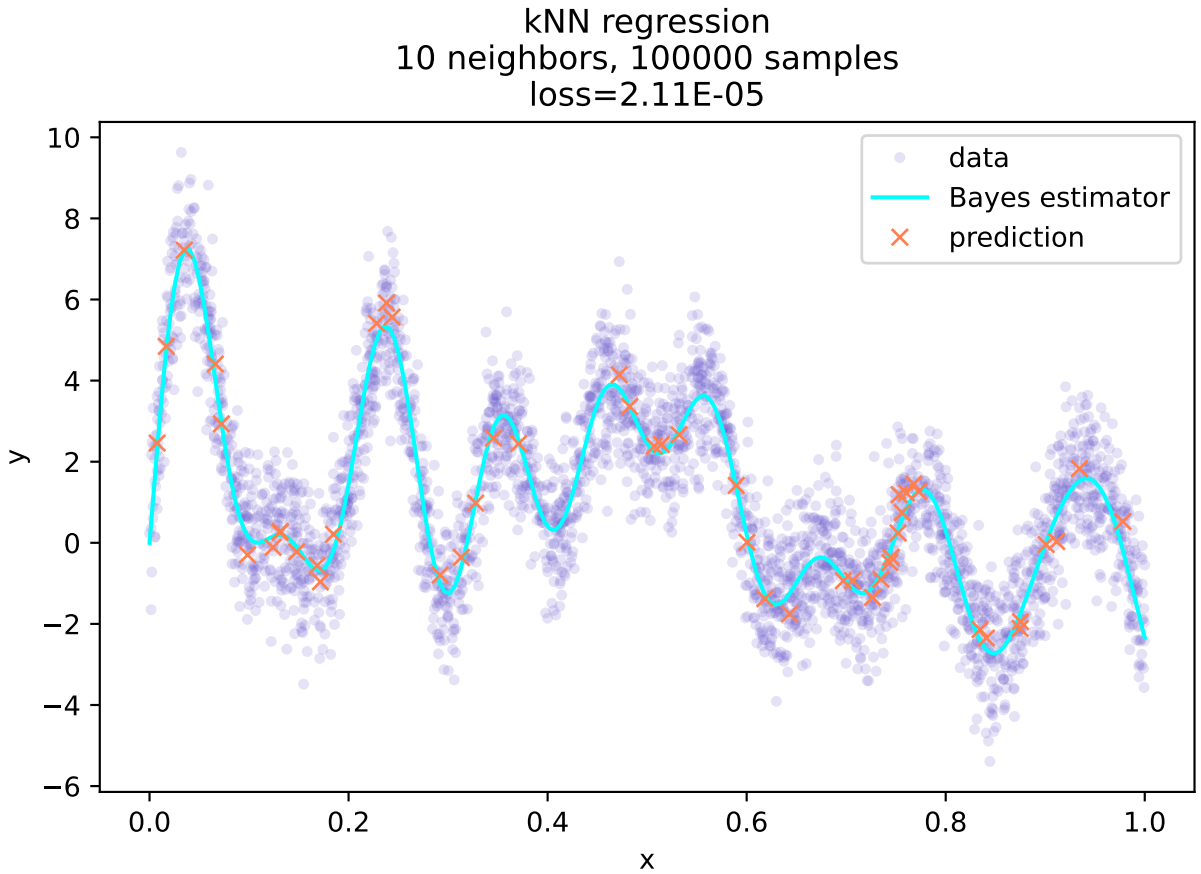
Machine learning I, supervised learning: overview



Overview of the module

Day 1 Definition of supervised learning, metrics, probabilities, risks, ordinary least squares, ridge regression.

Day 2 Gradient based algorithms, support vector machines, classification and regression trees, introduction to neural networks, sparsity.

Organization

- Presentations / discussions
- Coding exercises / paper+pen exercises
- Project : explained friday
- **questions :** please feel free to ask questions :
  - you can ask directly
  - or write them in the chat

more questions = more interesting / fun course !

Organization

- Please clone the following repository : `https://github.com/nlehir/MLI_SupervisedLearning.git`, that contains :
  - slides
  - exercises
  - other useful information

Practical aspects

- Python 3 `https://www.python.org/`
- Third-party libraries : see **requirements.txt**
- For installation, several options are available :
  - create a folder for the course and install libraries in a virtual environment (using e.g. pip) `https://docs.python.org/3/tutorial/venv.html`
  - install libraries globally on your machine, using e.g. pip (not recommended in the python community for production projects)
  - use docker (please see the README.md)

Virtual environment

```
..aph/AlgoGraph (-zsh)
..aph/AlgoGraph (-zsh)
(env) ➜  AlgoGraph git:(master) ✗ which python
/Users/nico/Desktop/enseignement/epitech/AlgoGraph/AlgoGraph/env/bin/python
(env) ➜  AlgoGraph git:(master) ✗ python --version
Python 3.9.0
(env) ➜  AlgoGraph git:(master) ✗ pip list
Package              Version
-------------------- -----------
anyio                3.6.1
appnope              0.1.3
argon2-cffi          21.3.0
argon2-cffi-bindings 21.2.0
asttokens            2.0.8
attrs                22.1.0
Babel                2.10.3
backcall             0.2.0
beautifulsoup4       4.11.1
bleach               5.0.1
certifi              2022.9.24
cffi                 1.15.1
charset-normalizer   2.1.1
contourpy            1.0.5
cycler               0.11.0
debugpy              1.6.3
decorator            5.1.1
defusedxml           0.7.1
entrypoints          0.4
executing            1.1.0
```

Organization

- **jupyter notebooks** are convenient python interpreters. Depending on your preference, you may use them or mere python scripts (I tend to prefer scripts but by copying and pasting, you are able to turn a script into a notebook and vice versa)