Unboxing the Machine Learning Algorithms in Python
(taught by Dr. Christian Kauth)

Machine learning is all about algorithms and there are so many algorithms that it can feel overwhelming. Really, there are probably only a dozen algorithms that, if understood, will unlock the field for you.

This course unboxes some of the key machine learning algorithms and will give you a deep understanding of how the algorithms work, how they can be parameterized or modified to tune to a specific problem.
In this course, you will learn the intuitions behind different kinds of linear, non-linear and ensemble algorithms and implement some of them from scratch in Python. You will begin to understand how the mathematical description of the method relates to vectors and matrices of numbers that your code operates on. You will also learn how the models' parameters are used, their effects and even gain insights into how it could be further parameterized to specialize it for a problem. We will illustrate the procedure by working through some well-understood case study machine learning problems from the UCI Machine learning repository (http://archive.ics.uci.edu/ml/index.php).

Objectives

- To gain a deep understanding of how some key machine learning algorithms work.
- To understand how the mathematical description of the methods relate to vectors, matrices and parameters.
- To develop an understanding of how parameters and meta-parameters can be tuned.
- To code your own machine learning algorithms in Python.

Content

- How to evaluate machine learning algorithms (crash-course on model evaluation and base line models)
- How linear algorithms work and their implementation in Python (regression, multivariate linear regression, logistic regression, Perceptron algorithm)
- How non-linear algorithms work and their implementation in Python (Naive Bayes, k-Nearest Neighbors, Learning Vector Quantization, Backpropagation, Decision Trees)
- How ensemble algorithms work and their implementation in Python (Bootstrap Aggregation, Random Forest, Stacked Generalization)

Preconditions

- Basic fluency in the programming language "Python", as e.g. provided in the course "Introduction to Python for Predictive Modeling".
- Either a local development environment with Python, Jupyter and admin rights, or a Google account to access Google Colab.

Duration

- 1 day on Feb 9th (roughly 7*45 minutes each day)

Evaluation

- take home exam: project work to be solved in Python

ECTS

- 0.5