Lecture 0: Introduction to the course

Andreas Wichert
Department of Computer Science and Engineering
Técnico Lisboa

Corpo docente –Tagus

- Andreas (Andrzej) Wichert – Teóricas
  - andreas.wichert@tecnico.ulisboa.pt
  - tel: 214233231
  - room: N2 5-7 (Taguspark)
  - http://web.tecnico.ulisboa.pt/andreas.wichert/

- Practical (Tagus)
  - Andreas (Andrzej) Wichert

Corpo docente –Tagus

- Practical (Tagus)
  - Maria Urze Osório
  - maria.osorio@tecnico.ulisboa.pt

- Practical (Tagus)
  - José Miguel Penedo Ramos
  - jose.miguel.ramos@tecnico.ulisboa.pt

Horário de dúvidas, Andreas Wichert

Tagus, room : N2 5-7

Monday, Tuesday, Wednesday: 15h-17h

(Seg, Ter, Qua, Qui: 15h-17h)

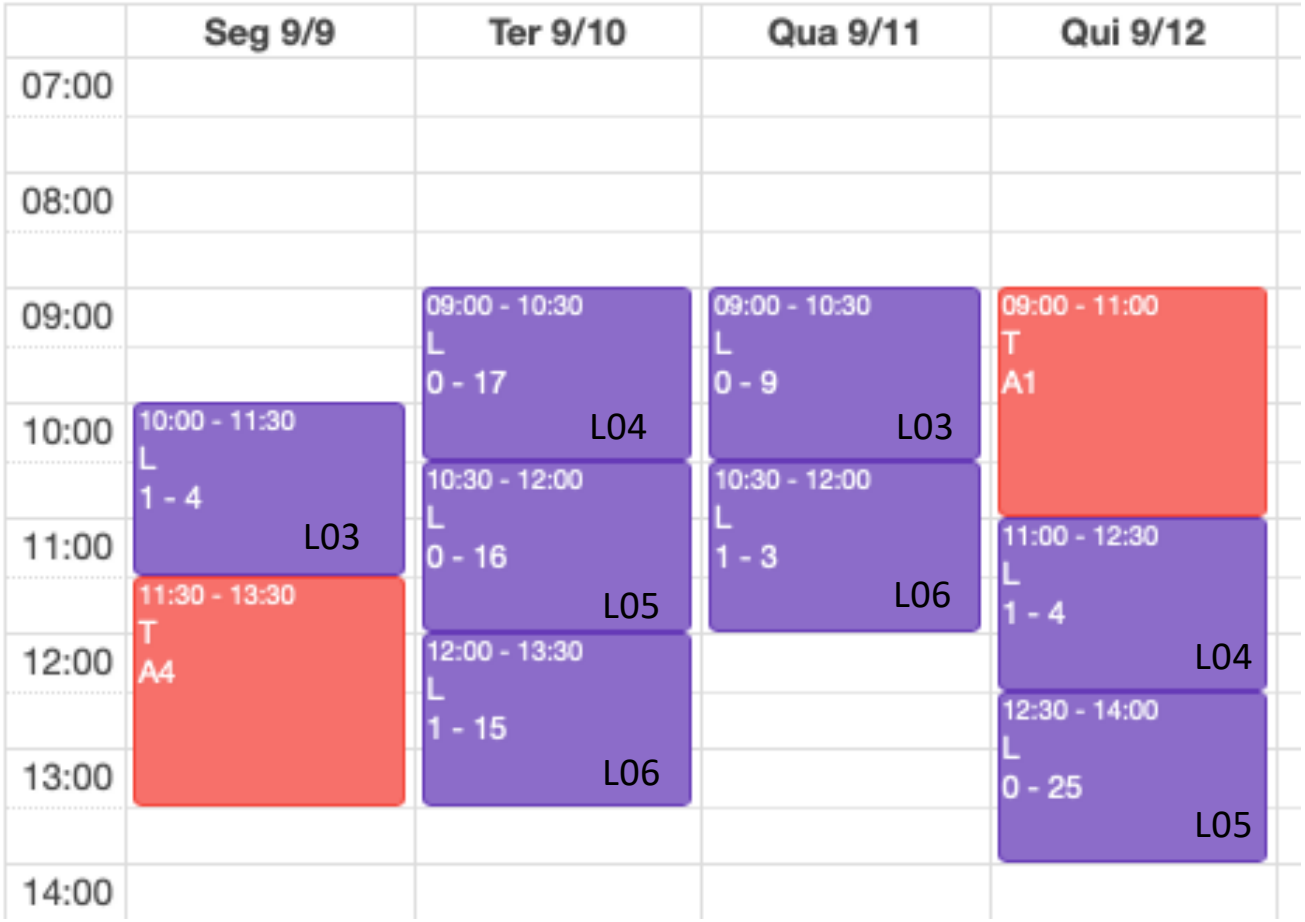
Organização das aulas

- Teóricas:
  - Matéria: slides (baseados no livros) + **livros**!

- Práticas
  - Exercícios Teoretical Exercises -> Exam
    - (Python: NumPy, Scikit-Learn)

  - Ínicio: **11th September**
  - Grupos: número de alunos **2**
  - **Inscription:** 11 September 14h...

Timetable - Rooms



Avaliação/Evaluation

- Regular period
  - Exam 50% (minimum 8)
  - 4 Home works 4 x 12.5%=50% (overall minimum 8)

- Special evaluation period
  - Exam 100% (minimum 8)

Homeworks

- **H1**: Decision Trees and Evaluation
  - statement release: September 18
  - deadline: Friday, September 25
- **H2**: Bayesian and Lazy Learning
  - statement release: September 25
  - deadline: Monday, October 7
- **H3**: Neural Networks and Regression
  - statement release: October 7
  - deadline: Friday, October 18
- **H4:** Clustering and PCA
  - statement release: October 18
  - deadline: Monday, October 28

Avaliação

Testes/Exames	Dia	Início	Fim	Período de inscrição	Salas	Cursos
Exame: 1ª Época	04/11/2024	13:00	15:00			LEIC-T
Exame: 2ª Época	06/02/2025	10:30	12:30			LEIC-T
Exame: Época Especial	24/07/2025	15:30	17:30			LEIC-T

Labs / Practical Classes

- Registration
  - Groups with **2** students (from the same campus)
  - Registration from 11th September 14h on Fénix

- First practical class 16th September
  - **Pen and paper**, we will use *some* examples on the computer as well
- We will use *Python: NumPy, Matplotlib*, *Seaborn*, *Scikit-Learn, Pandas*
  - Anaconda-Navigator or Conda
    - Install different Python libraries
  - Editor: Visual Studio Code
    - https : //code.visualstudio.com/doc

- We will use *Python: NumPy, Scikit-Learn* with *Jupyter* notebooks
  - https://scikit-learn.org/stable/index.html
  - http://www.numpy.org
- Anaconda-Navigator or Miniconda
  - Install different Python libraries, like NumPy, Scikit-Learn
  - Editor:
    - For example Visual Studio Code or another Editor that supports *Jupyter* notebooks
    - https : //code.visualstudio.com/
- Google Colab (Cloud Service, NO NEED TO INSTALL ANYTHING ☺)

Additional Libraries

- **Matplotlib:** Visualization with Python
- https://matplotlib.org
  - Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- **seaborn**: statistical data visualization
- https://seaborn.pydata.org
  - Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- Numpy and Matplotlib are (nearly) compatible to **MATLAB**

- Operatin with matrices and Vectors and Ploting results
- You can find many examples and descriptions at
  - https://www.mathworks.com/products/matlab.html
- Search with Google for Matlab etc
  - For example search for: matlab matrices
  - https://www.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html?s_tid=srchtitle_site_search_2_MATLAB%20MATRICES

- Python and all the librarys run as an interpreter (slow)
- However some **NumPy** routines are precompiled for your processor and run much faster
- Use matrix operationusing **NumPy** are **much faster** than the matrix operation using standart Python

- The task of the class is not to learn how to program *Scikit-Learn*
- Rather to understand the basic mathematical framework of the algorithms!

- There are many other frameworks
  - R, Weka, Shogun, Tensorflow, Keras, PyTorch, MatLab, Mathematica, etc…
- Software changes, you have constantly to adapt, *what is most popular today will be out in comming years*

Serge Gainsbourg (1928-1991)

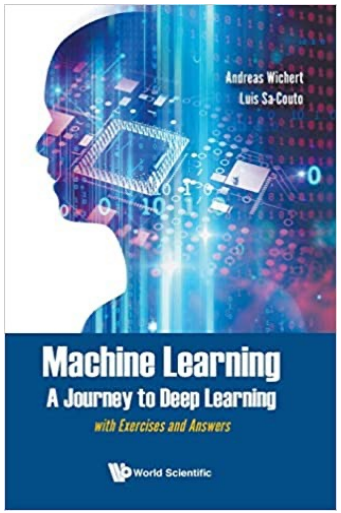
Qui est in ?
Qui est out ?
Qui est in ?
Qui est out ?

Tu aimes la nitroglycérine (in)
C'est au Bus Palladium que ça s'écoute (out)
Rue Fontaine, il y a foule
Pour les petits gars de Liverpool
Barbarella garde tes bottines (in)
Et viens me dire une fois pour toutes (out)
Que tu m'aimes ou sinon
Je te renvoie à ta science fiction

- You have to understand what is happening, do not use the algorithms as a **blackbox..**

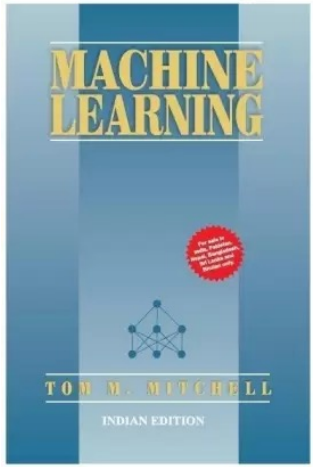
- The mathematical models do not change fast, most algorithms were developed 30 or 40 years ago (or even more), the corresponding knowlede stayes for ever!
  - Known algorithms are adapted and improved over the time, see backpropagation algorithm developed around 1984 that is the basis of deep learning
- A lot of progress was mostly made by faster and more powerful computers (see deep learning revolution)

Main literature



- **Machine Learning: A Journey to Deep Learning**

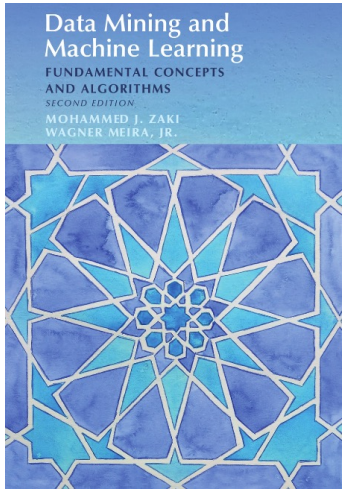
Andreas Wichert and Luis Sá Couto, World Scientific, 2021
https://www.worldscientific.com/worldscibooks/10.1142/12201



- **Machine Learning**

Tom Mitchell, McGraw Hill, 1997
http://www.cs.cmu.edu/~tom/mlbook.html

Secondery literature (Data Science)



- **Data Mining and Machine Learning: Fundamental Concepts and Algorithms**

  Mohammed J. Zaki and Wagner Meira Jr, Cambridge Univ. Press, 2nd Ed., 2020

  https://dataminingbook.info/

ML - the **Theoretical** Minimum:

- Lecture 1: Learning and Univariate Data Analysis
- Lecture 2: Decision Trees
- Lecture 3: Model Evaluation
- Lecture 4: Probability and Bayesian Classifier
- Lecture 5: K Nearest Neighbour
- Lecture 6: Linear Regression
- Lecture 7: Perceptron
- Lecture 8: Logistic Regression
- Lecture 9: Multilayer Perceptrons
- Lecture 10: Clustering
- Lecture 11: PCA
- Lecture 12: Deep Learning