# Machine Learning

## Computer Engineering

Fabio Vandin

October 2nd, 2023

# Machine Learning

6 credits:

- 48 hours in class lectures
  - Some hours will be in a lab: more decisions to come later
- **<u>102 hours individual study</u>**

Everything (lectures, exams, homeworks) in English!

**Note:** if you are enrolled in “ICT for Internet and Multimedia” or in “Control Systems Engineering”, you should follow a different section!

# Course Website

Course website: from https://stem.elearning.unipd.it/

**Register today if not done yet!**

# Lectures: When and Where

**Monday 10:30-12:30pm, room Ae:**

**How should we split the lecture?**

- 10:30-12:00pm; no break ☹

**Friday 10:30am-12:30pm, room Ae:**

**How should we split the lecture?**

- 10:30-12:00pm; no break ☹

# Lectures: When and Where

**Lectures are not live streamed.**

**Material from analogous lectures from 2021/2022 are available on the course website.**

**See the material posted on the website to see if there are differences in the material covered.**

# Labs and Homeworks

**Labs**:

- some (6?) lectures will be done in the lab, dates to be fixed


**Homeworks**:

- (up to) 3 homeworks
- will give up to 3 points for the final grade
- not compulsory but highly recommended
- will require to complete the code in Jupyter notebooks
- typical schedule:
  - day X:  homework released
  - day  X+14: deadline for homework submission

# Grading

**Written test (NEW FORMAT)**:

- Part 1: multiple choice questions, pass/fail [30 mins]
- Part 2: questions and exercises [1.5-2 hours]
  - **graded only if part 1 is passed**
  - see sample tests from previous years in elearning
  - will be graded on a scale from 0 to 30L.

**Homeworks [not compulsory]**: some (3?) homeworks. Up to 3 points as a bonus on the written test grade.

**Final Grade** = grades written test + homeworks

**Example**

24.5 written test + 2.66 homeworks = 27 final grade

**Note**: there may be an oral exam just to confirm the grade of the written exam.

# Final Exam: dates

1. **Tuesday, January 23rd, 2023**
   - time: 2pm
   - rooms: De, Ke, Ve

2. **Tuesday, February 13th, 2023**
   - time: 9:30am
   - room: De, Ke, Ve

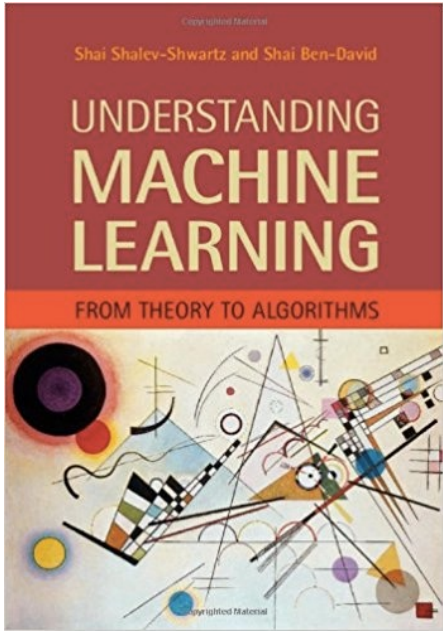
3. **Tuesday, July 2nd, 2023**
   - time: 9:30am
   - room: Ke

4. **Tuesday, September 10th, 2023**
   - time: 9:30am
   - room: Le

# Material

**Main Book**

- [UML] Shalev-Shwartz, S. and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.




Material in class will be related to the book as much as possible

**PDF available from the authors** http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/copy.html

# Material (2)

**Other Books (NOT Mandatory)**

- M. Hardt, B. Recht, *PATTERNS, PREDICTIONS, AND ACTIONS - A story about machine learning*. Princeton University Press, 2022 (available from the authors at https://mlstory.org/).

- T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning*. Springer, 2008.
  **PDF from authors online**
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006
- K.P. Murphy. *Machine Learning A Probabilistic Perspective*, MIT Press. 2012.
- Yaser S.Abu-Mostafa, M. Magdon-Ismail, H. Lin. *Learning from Data*. AMLBook, 2012.

Part of other books may be used: will provide handouts whenever possible…

**Additional material**: course website (*stem.elearning.unipd.it*)

- draft slides: provided sometime (…) before the lecture
- slides used in class: published after lectures
- links, etc.

**If you are missing background notions (probability, algebra): ask me for material**

# Programming and more

Language:  (https://www.python.org)

Some libraries: scikit-learn, numpy,…

Jupyter lab:

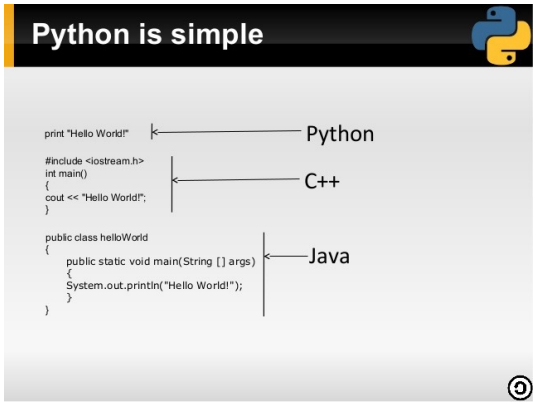
- https://jupyterlab.readthedocs.io/en/stable/getting_started/overview.html
- Allows for a mix of text and code;
- **Suggestion: install it through Anaconda** as suggested on Jupyter website, you get a lot of other packages/libraries for ML, visualization, etc.

  https://www.anaconda.com/download/

  (That's what we are going to use for homeworks, etc.)

# Why Python

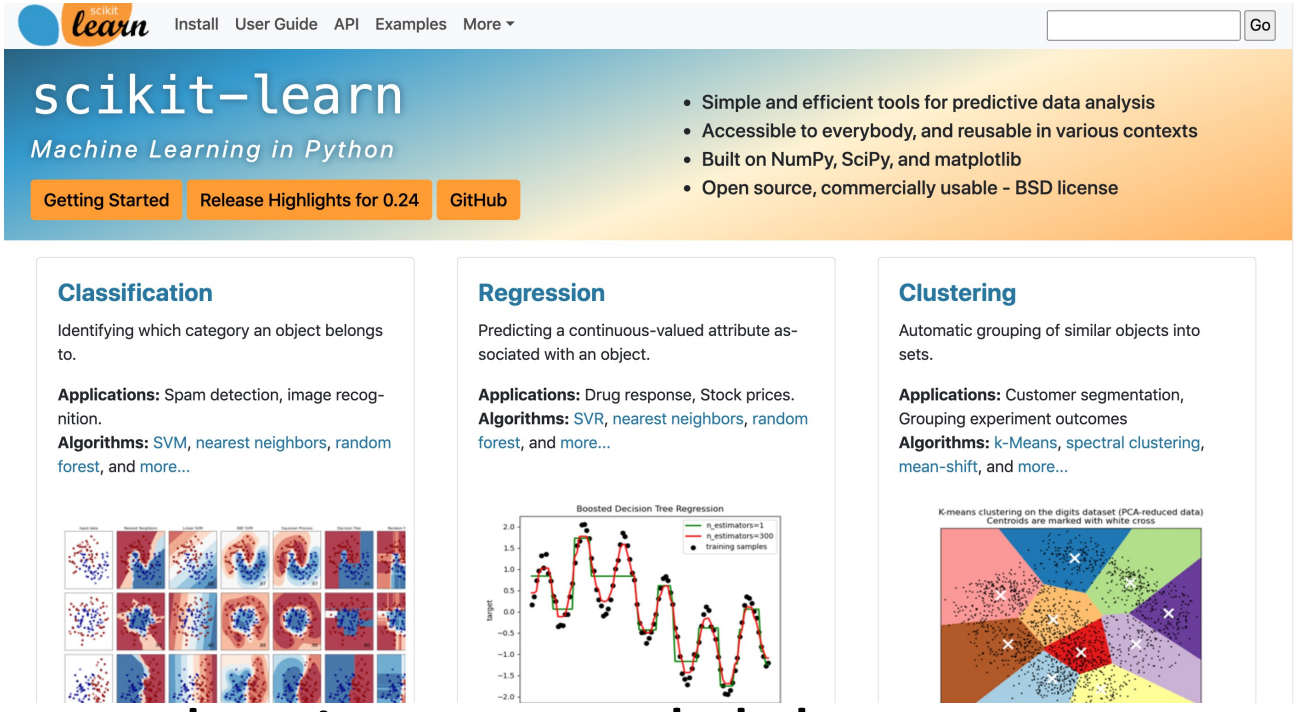
Easy! (...)




```python
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) / 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)

print quicksort([3,6,8,10,1,2,1])
```

A lot of support for ML!




Used in industry, academia, research labs...

# Homework 0

1. Go home and install Anaconda (version with Python 3.11):
   - https://www.anaconda.com/download/

2. Go through the following tutorial (it's for Python 3.7, but still useful):
   - http://cs231n.github.io/python-numpy-tutorial/

3. Read Chapter 1 and Chapter 2 from *Data Structures and Algorithms in Python* [Goodrich, Tamassia, Goldwasser]
   - available on course website

4. Get used to Jupyter lab/Jupyter notebooks

5. Go through the following tutorial running the Jupyter notebook and in script mode
   - https://github.com/marcc-hpc/python3-tutorial/blob/master/python3-tutorial.ipynb

# Lecturer

Fabio Vandin, Professor, DEI (Department of Information Engineering)

**Email**: fabio.vandin@unipd.it

**Website**: www.dei.unipd.it/~vandinfa

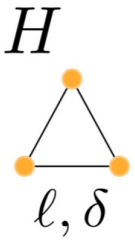
**Office**: 410 ($4^{th}$ floor, DEI/G), phone: 049-827-7946

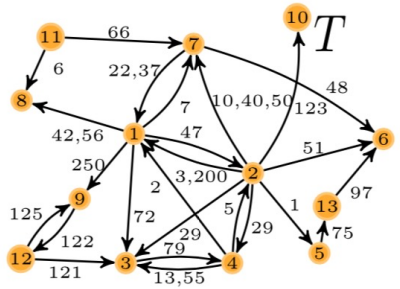
**Office hours:** Tuesday, 9-10am, by appointment (email before Monday at 5pm)

# Lecturer

**CV**

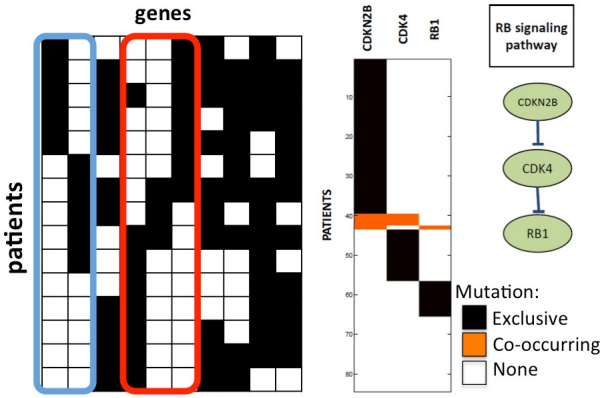
- Laurea Triennale in Computer Engineering (2004)
- Laurea Specialistica  in Computer Engineering (2006)
- Ph.D. at DEI (2010)
- 2010-May/2015: Researcher/Assistant Prof.
  - Brown University (USA)
  - University of Southern Denmark
- 2015-2020: Associate Prof. at DEI
- February 2020-now: Prof. at DEI




**Research Interests**:

- Methods: algorithms for machine learning, data mining, and big data
- Applications: Biology, Medicine, Social Networks, …



# Example: Ad Click Prediction

**Goal**: predict if a user will click on a given ad if the ad is shown to the user

**Why is it important?**

Social networks…

How would you solve the problem?

You have data about the user and about the ad

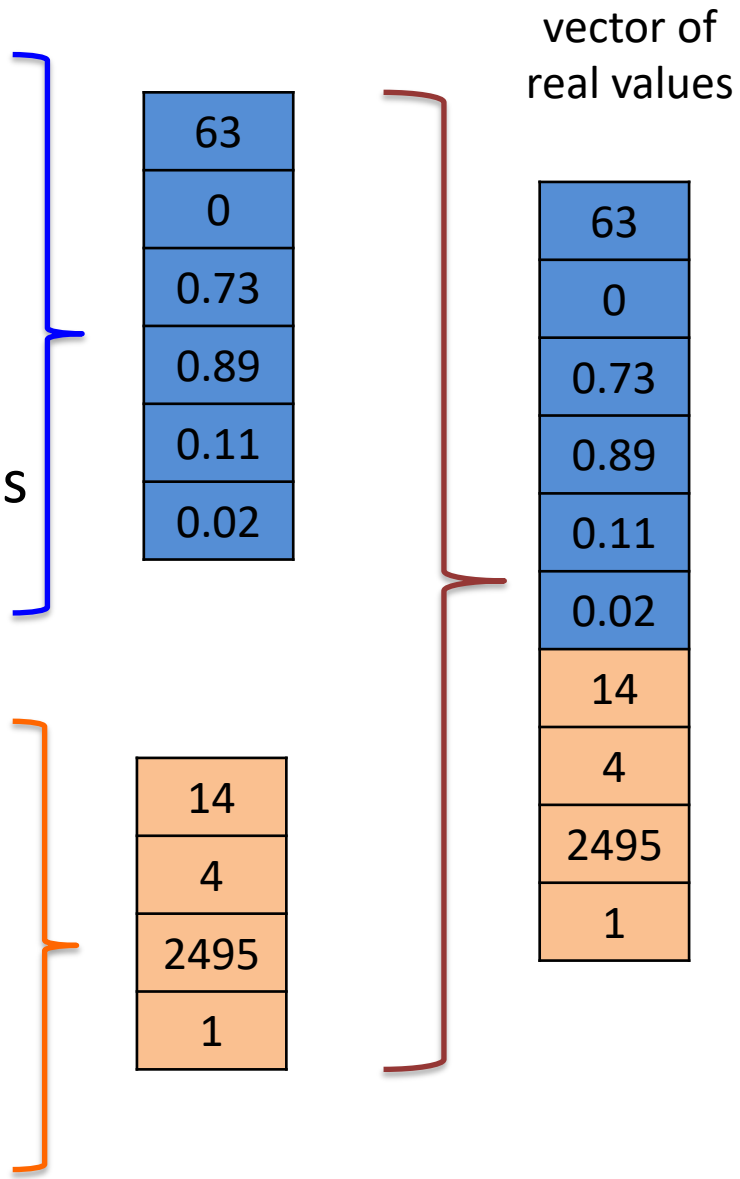
# Ad Click Prediction: Data

**User Data**:

- Age
- Gender
- How much he/she likes music
- How much he/she likes sports
- How much he/she likes books
- How frequently he/she clicks on ads
- …

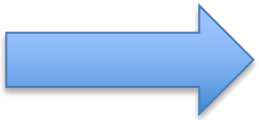
**Ad data**:

- Topic
- Colors
- Image (what is shown?)
- Does it contain the word “song”?
- …



# A Solution that is NOT Machine Learning

Somebody (an "expert") *manually* builds a *model/formula/algorithm* that decides if the user will click on the ad or not

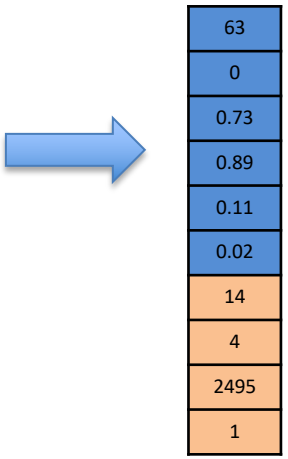


"If the age is <25
and
the gender is male
and
the ad is about soccer
then
he will click on the ad"

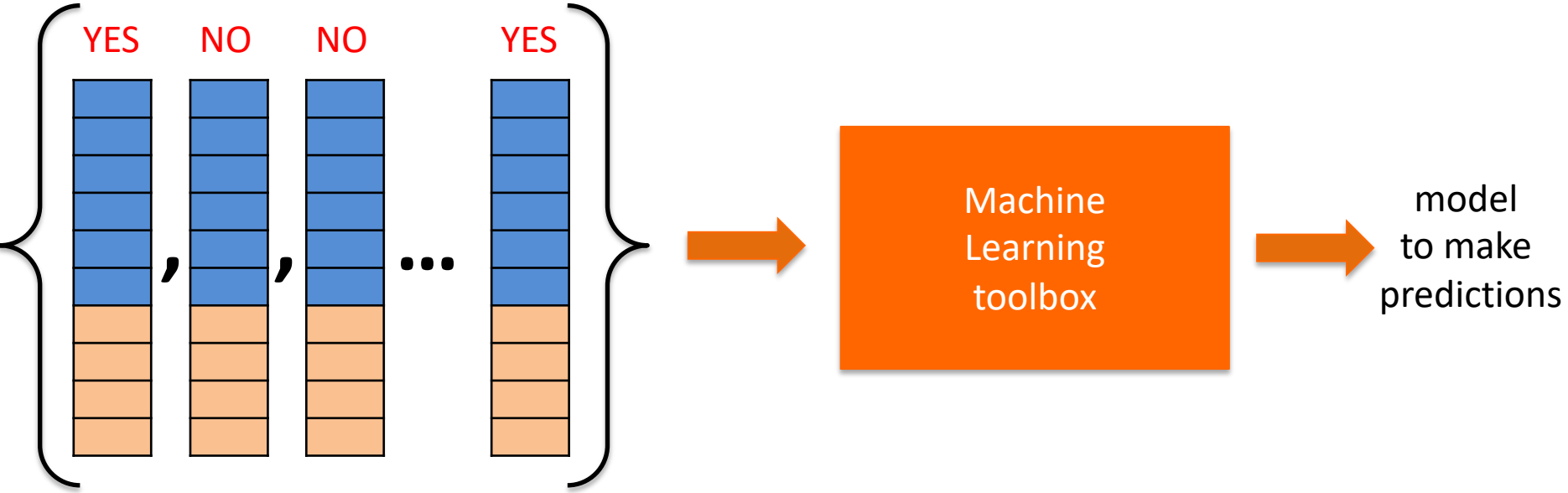
# A Machine Learning Solution

You use previous data about:

- user and ad that was displayed to user
- whether the user clicked on the ad (label: YES/NO)




You give a lot of such pairs (vector,label) to a machine learning system which then produces in output a *model* to make the prediction



# Learning: A  Difficult Problem

**Note**: coming up with a model that performs very well on the data I already have is very easy!

**How?**

The difficulty is about making predictions for data we have not seen!

# Questions?

# Example

Want to predict whether a student that just graduated from Computer Engineering will have a fun job or not based on some features:

- age at graduation
- LM final grade
- Machine Learning (ML) grade
- height

See Jupyter notebook.