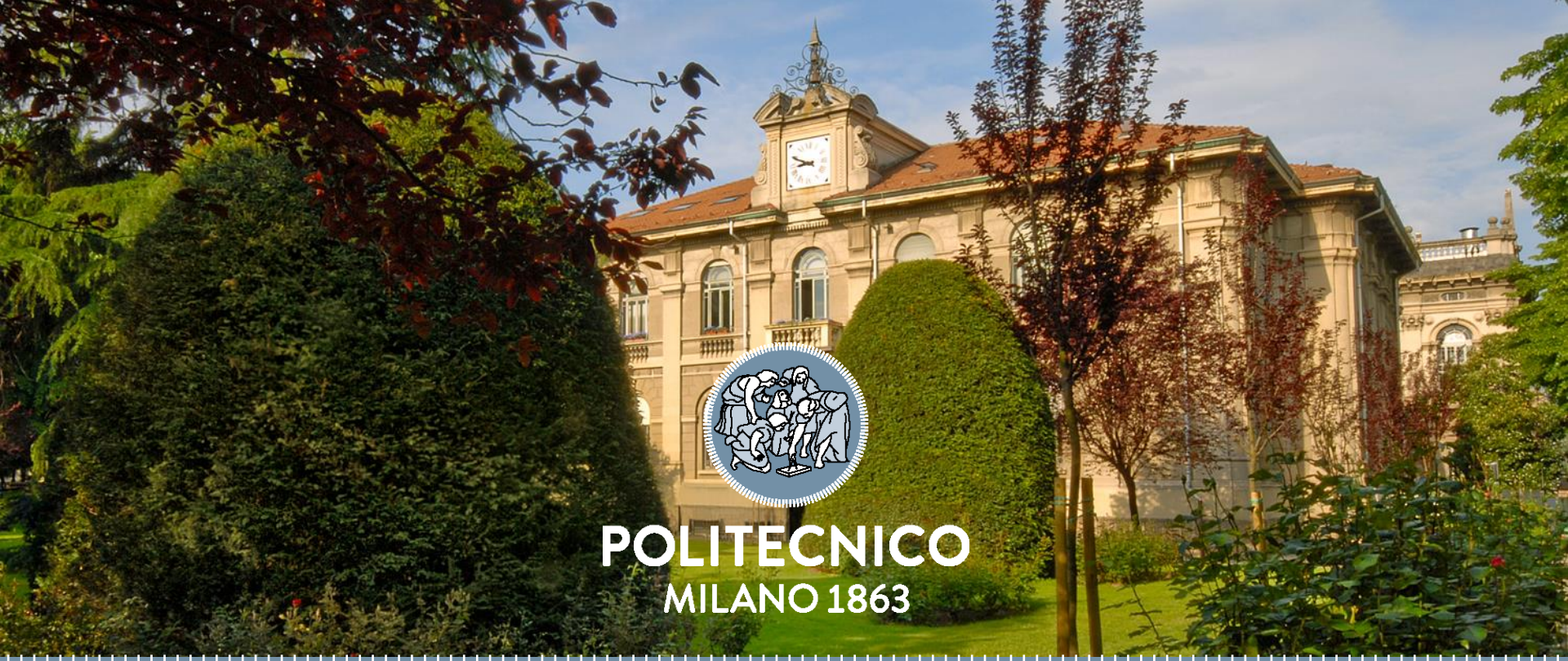

Recommender Systems

Introduction to practice sessions

A.Y. 2022/23

Hello!

Maurizio Ferrari Dacrema

Postdoc researcher

- Recommender Systems
- Quantum Machine Learning

maurizio.ferrari@polimi.it

What we do in the practice sessions

In the practice sessions we show you how to build a recommender system from scratch, evaluate it and understand its underlying behavior.

Programming language is Python

We will use prewritten and commented Jupiter notebooks

What we do NOT do in the practice sessions

We (mostly) do not implement models live, for a few reasons:

- Writing the core of the models is simple, writing all needed to fit and optimize them requires often too many steps to do clearly live
- Training and evaluation time of some models is too long
- We rely on few well known mathematical operations

What do we recommend for practice sessions?

Familiarize beforehand with Python and the libraries that we use (numpy, scipy, scikit-learn). Some basic materials are available.

Try to understand the models before attending the session, if possible, write them down using your favorite tool.

You are more likely to succeed and get the best of the practice sessions if you are prepared for them.

Why should you attend if all the materials are already online?

During the practice sessions we try to discuss the effects of certain choices made during the implementation of the model as well as interpret the model results and behavior.

Wider perspective than simply “show me the code”

Useful for those that want to participate in the challenge

The oral exam on practice sessions

If you take the oral exam on practice sessions you will be asked to show that you can understand and write in either Python code or pseudo code the core components of the models we see:

- Calculating similarity heuristics
- Generating recommendations given a model
- Fitting machine learning models with gradient descent

The oral exam on practice sessions

Furthermore, in terms of the way you write the code itself, it is:

Expected:

Intelligible and correct implementation

Some use of libraries and scipy sparse formats

You know how to implement a dot product without numpy

NOT expected:

Perfect syntax and perfect knowledge of all libraries

Highly optimized code

Course repository

The course repository contains all notebooks we will show in the practice sessions and the implementations of the models and tools you will need. Will be updated as the course progresses.

https://github.com/MaurizioFD/RecSys_Course_AT_PoliMi

Suggested resources

Python distribution: Anaconda

https://www.anaconda.com/products/individual

IDE: Pycharm Community

https://www.jetbrains.com/pycharm/download/

Suggested resources (lightweight)

Lightweight IDE: Visual Studio Code

https://code.visualstudio.com

Recommended IDE Extensions (install both)

Python

Pylance

Suggested resources for computation

If you need external computation resources you will be able to find some free ones on the cloud:

- Google Colab
- Google Cloud platform (gives you some initial credits)
- Kaggle notebooks (file system has limitations)

Let us know if you spot others you think we should add!