



Applied Machine Learning

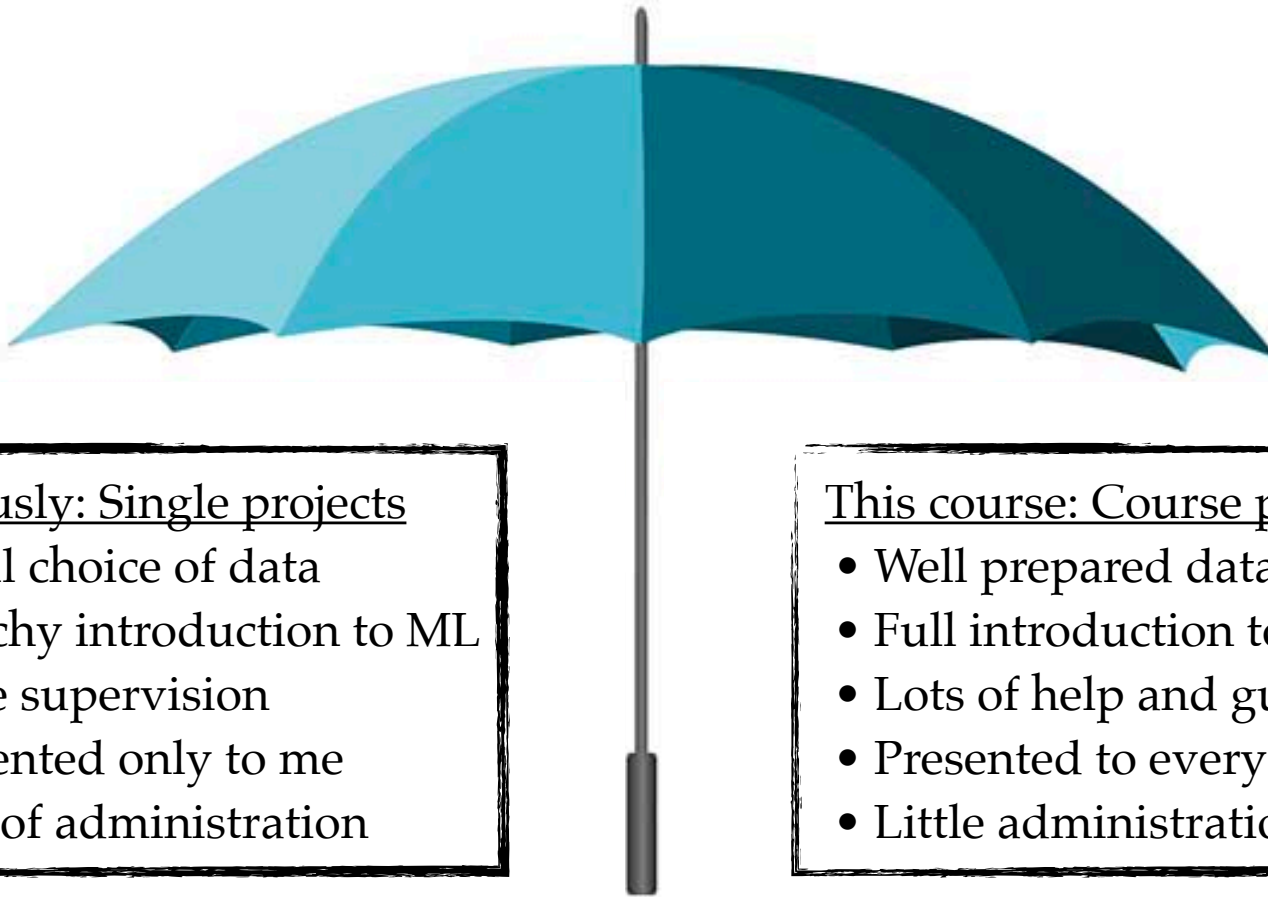
Course information 2022

Troels C. Petersen, Charles Steinhardt, Julius Kirkegaard,
Vadim Rusakov, Rajeeb Sharma & Azzurra d'Alessandro



This course is (partially) an
“umbrella course”
for doing projects with ML

An “umbrella course”



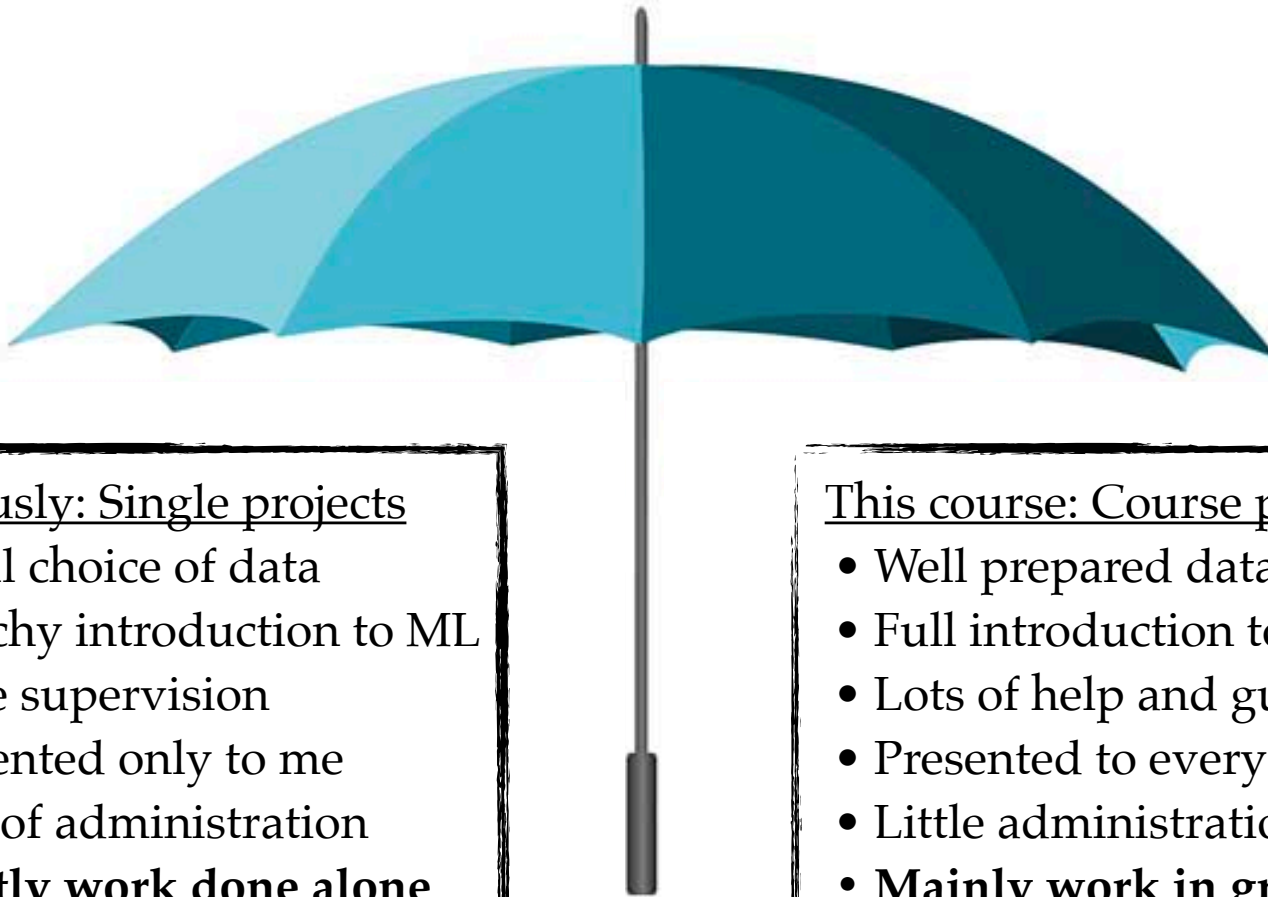
Previously: Single projects

- Small choice of data
- Sketchy introduction to ML
- Little supervision
- Presented only to me
- Lots of administration

This course: Course projects

- Well prepared data cases
- Full introduction to ML
- Lots of help and guidance
- Presented to everyone
- Little administration

An “umbrella course”



Previously: Single projects

- Small choice of data
- Sketchy introduction to ML
- Little supervision
- Presented only to me
- Lots of administration
- **Mostly work done alone**

This course: Course projects

- Well prepared data cases
- Full introduction to ML
- Lots of help and guidance
- Presented to everyone
- Little administration
- **Mainly work in groups**

General words on the course

We are back to normal times, which call for normal course running!

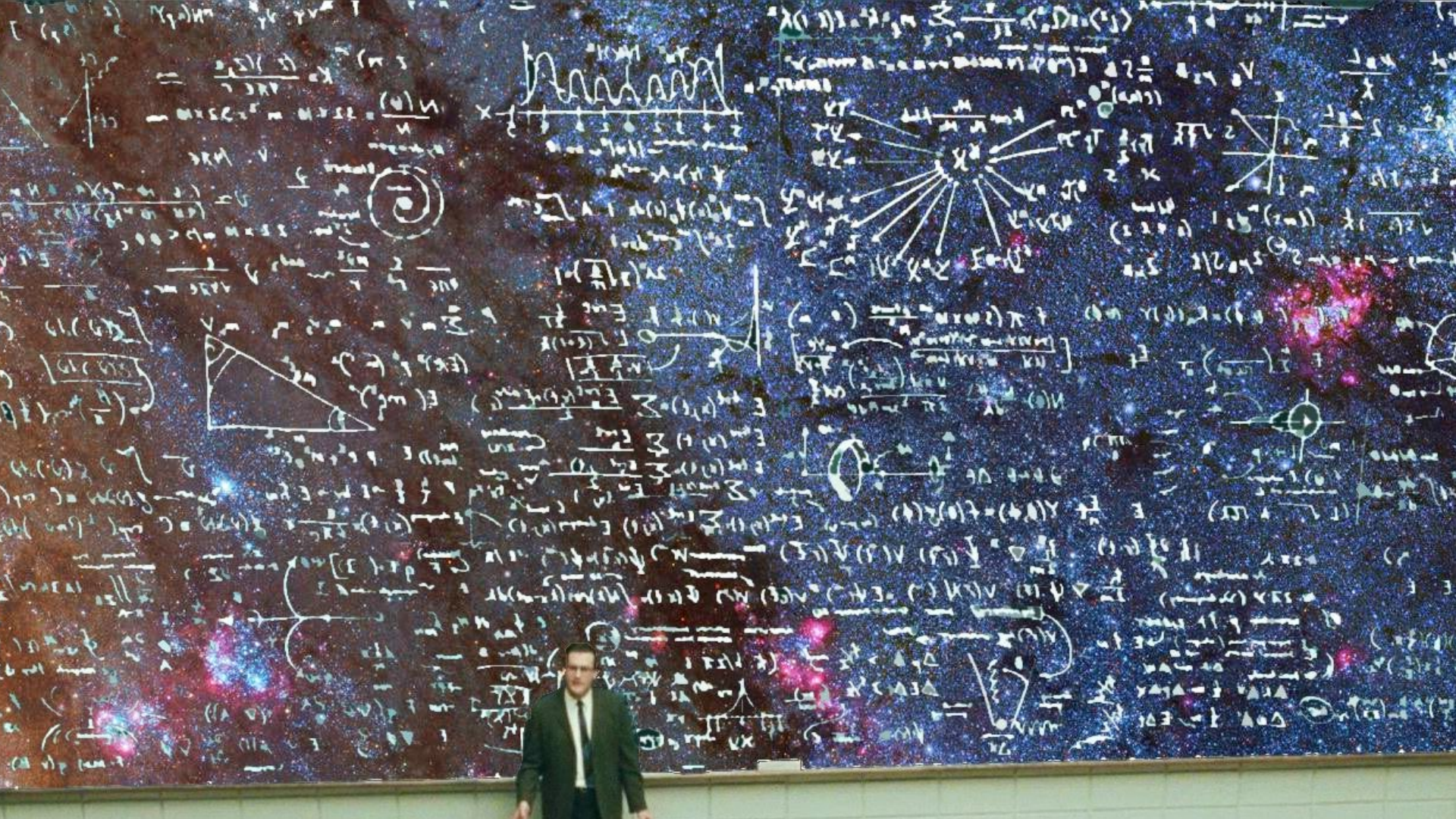
We will thus run the course **in person (only)**. However, most lectures will be linked to zoom videos from previous years. We are very happy to be back to normal, and look much forward to seeing you (for real).

The course require both self-disciplin and dedication.

We will of course do our best to inspire, help, debug (?), and promote collaboration, but it is of course up to you, how much you want to learn/benefit from this course.

Course work can/should be done in collaboration with fellow students.

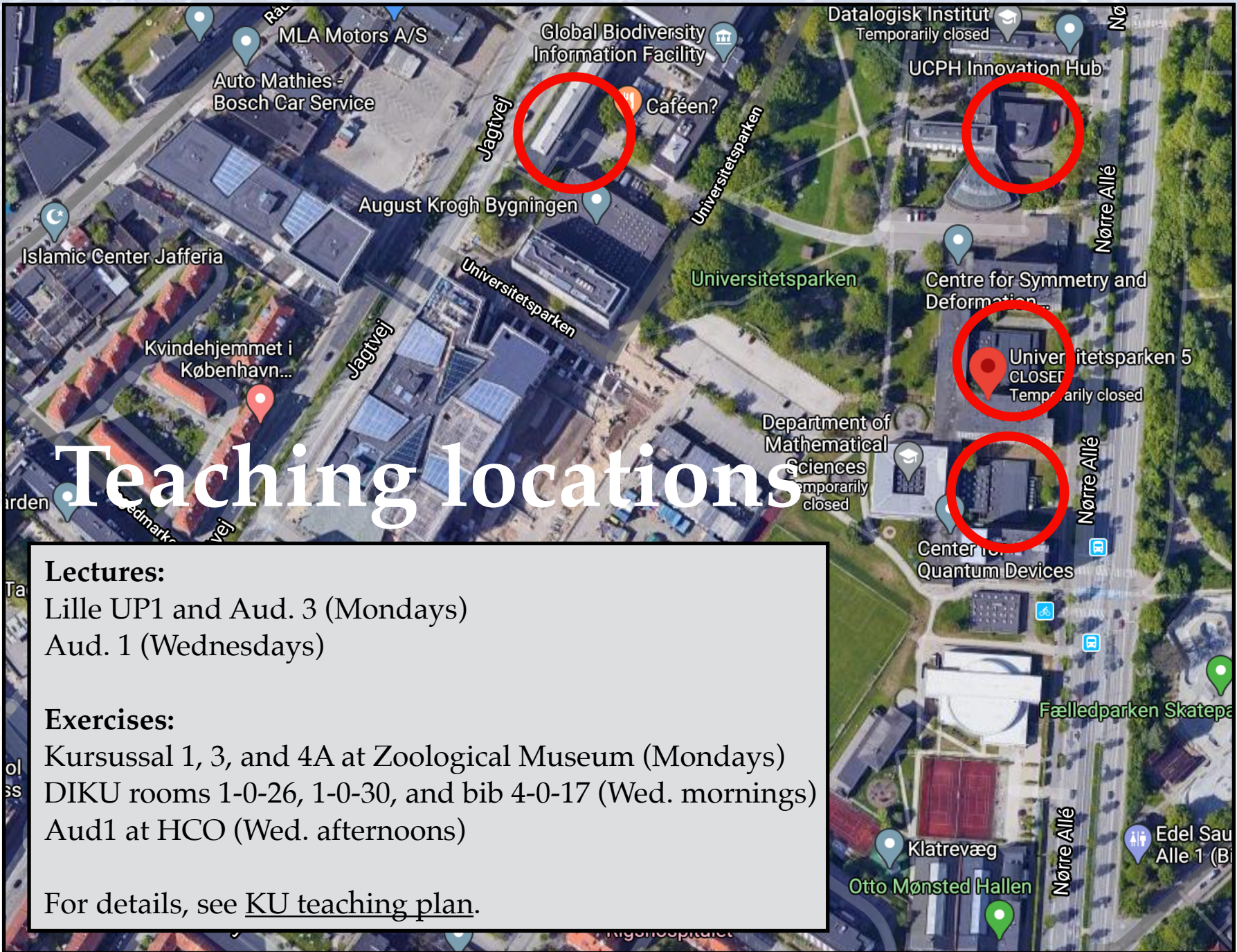
Please make small teams of peers, with whom you can discuss the many details of ML coding and the problems, challenges, and issues involved. This is you best way of discussing with peers, **learning most**, and not getting stuck.



You can not teach a person anything!

You can only help them discovering it in themselves..

[Galileo Galilei]



Teaching locations

Lectures:

Lille UP1 and Aud. 3 (Mondays)
Aud. 1 (Wednesdays)

Exercises:

Kursussal 1, 3, and 4A at Zoological Museum (Mondays)
DIKU rooms 1-0-26, 1-0-30, and bib 4-0-17 (Wed. mornings)
Aud1 at HCO (Wed. afternoons)

For details, see [KU teaching plan](#).

Teaching locations



4A

3

1

Entrance

Cafeen?

Exercises:
Kursussal 1, 3, and 4A at Zoological Museum (Mondays)

Additional locations

Troels' office
(building M, top floor)



Julius' office
(building K, 2. floor)



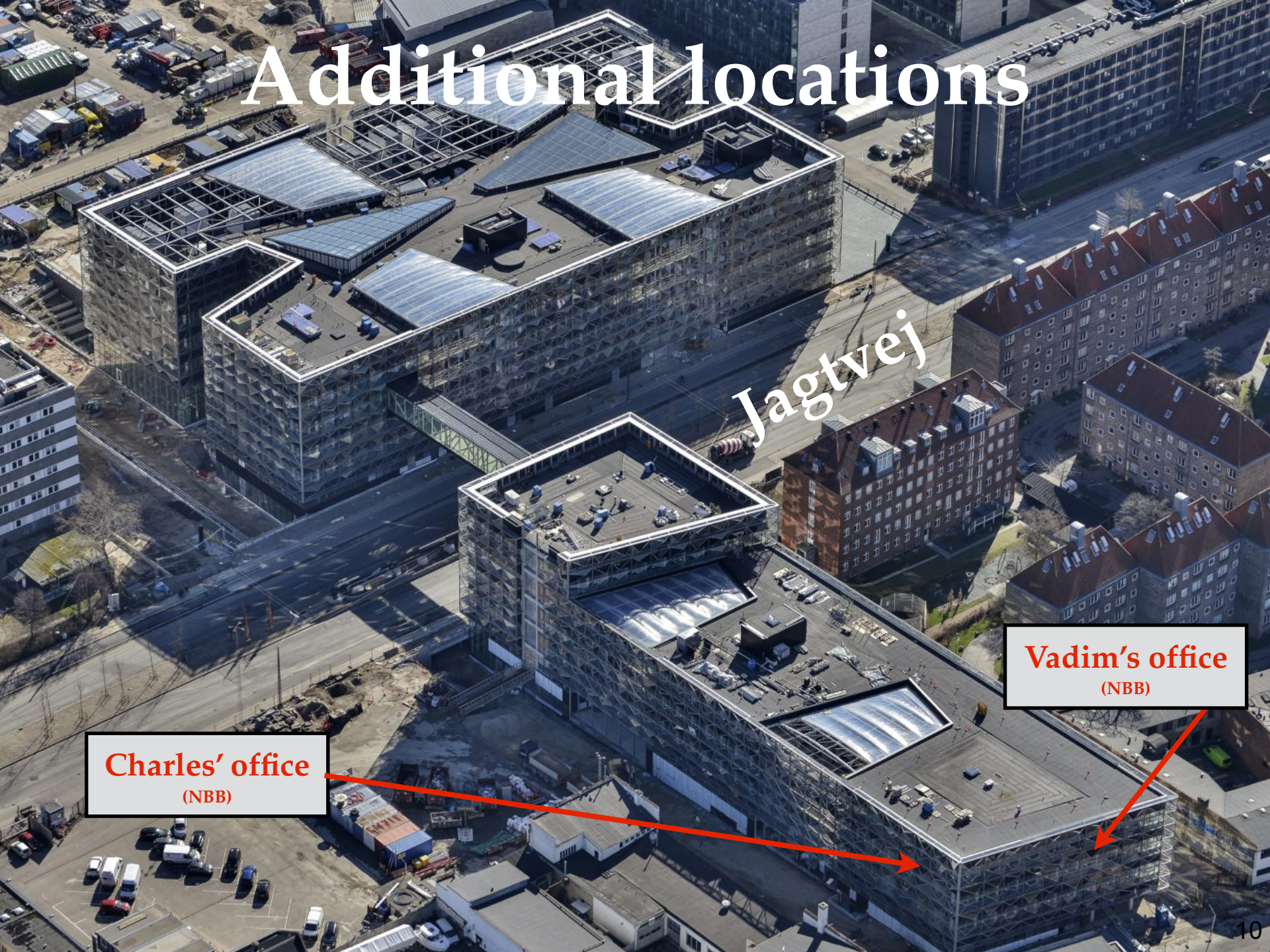
Blegdamsvej

Additional locations

Jagtvej

Vadim's office
(NBB)

Charles' office
(NBB)



Computers and software

We will program in Python. You may **choose as you wish**, but we highly recommend Python. We will only provide data, code snippets (in Python), and occasional code/solutions for inspiration.

We suggest that you use Jupyter Notebook, and run everything on your own laptop, possibly with ERDA as a backup. For GPU access, Google Colab is (so far) the only thing we can refer to. We also recommend that you use GitHub.

Data files will typically be provided in CSV and/or HDF5 format, but others might be used.

We will be using many additional Python packages, introduced along the way, and surely you have your own favourites. Use them happily (but knowingly).

Projects / Exam

This course is to some extent an umbrella course for projects using ML.

We will be doing two projects:

- An individual “initial project” on **common data** (2 weeks - 40% of your grade).
- A group “final project” on **data of your choice** (3 weeks - 60% of your grade).

The **initial project** will be the basic applications of ML (classification, regression, and clustering) to a data set, and we will evaluate your (algorithm’s) performance on a test set.

The **final project** will be your main task, and can be the application of ML on anything that you like. You will all be presenting your results to each other, so that also others may learn from what you did (and didn’t).

You can find much more information about both projects on the course webpage (so please go and read it at least once!):

- Initial project (to be submitted individually).
- Final project (to be submitted in groups).

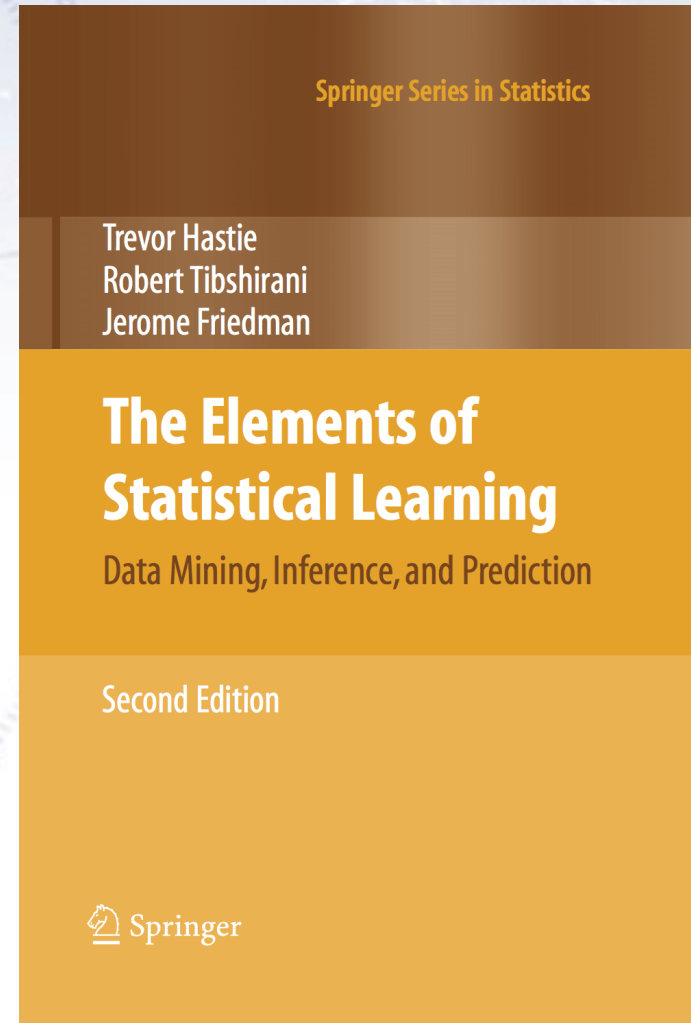
Literature

The main literature will be slides, notes, blogs, and links! However, we also wanted you to have a few more “solid” places to read comprehensively about ML.

“The Elements of Statistical Learning” (TEOSL) is a good read in PDF (though at times rather mathematical), and especially chapter 2 is a good introduction.

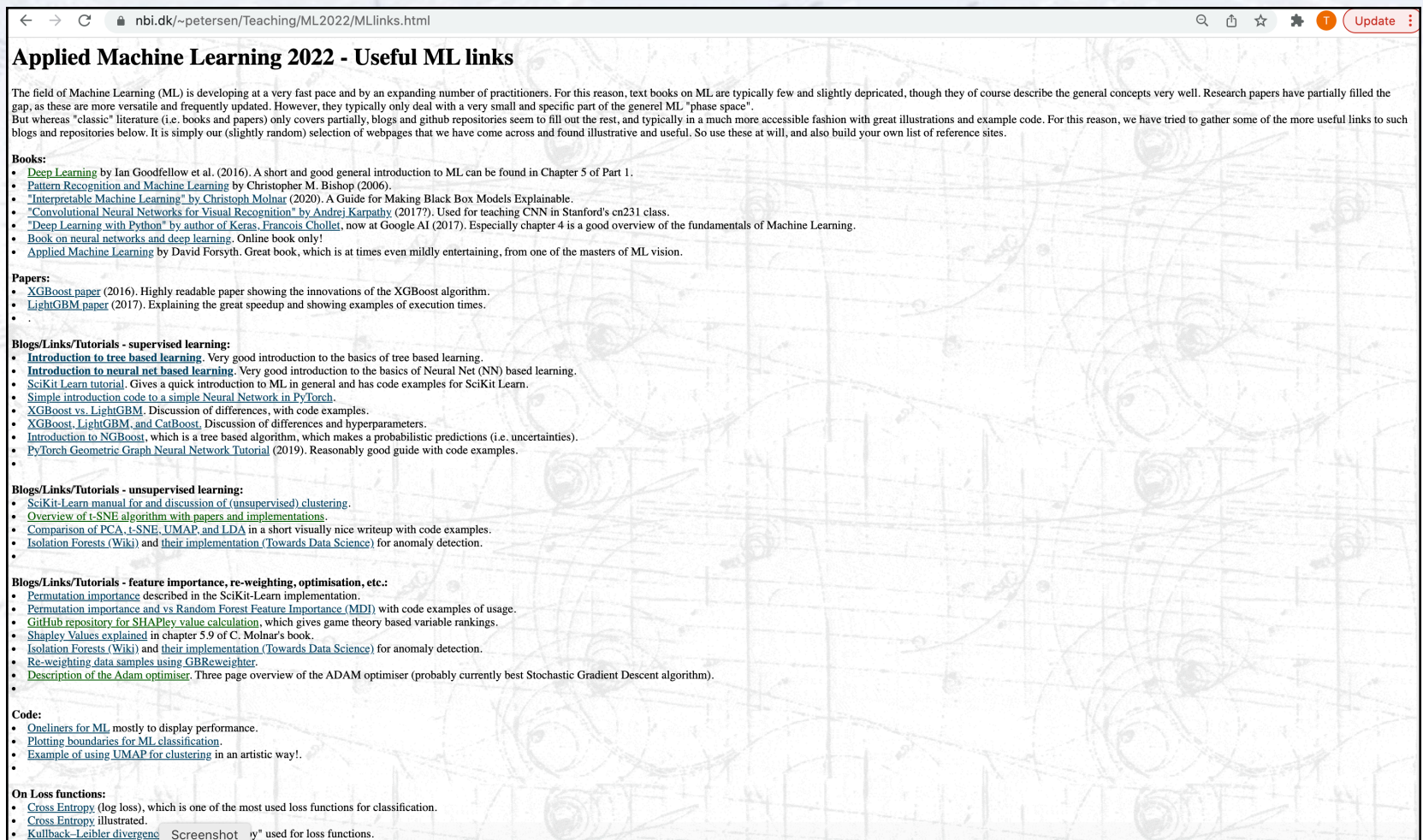
“Deep Learning” by Ian Goodfellow et al. in HTML is also good, and Chapter 5 of Part I gives a great overview of ML and its ingredients.

“Pattern Recognition and Machine Learning” by Christopher M. Bishop is also recommended, but it is not available on the web (for free).



Blogs as literature

In ML, blogs/articles/tutorials are a very common (and great) source of literature on ML. For this reason, we've made a list of links that we find good:



Applied Machine Learning 2022 - Useful ML links

The field of Machine Learning (ML) is developing at a very fast pace and by an expanding number of practitioners. For this reason, text books on ML are typically few and slightly deprecated, though they of course describe the general concepts very well. Research papers have partially filled the gap, as these are more versatile and frequently updated. However, they typically only deal with a very small and specific part of the general ML "phase space". But whereas "classic" literature (i.e. books and papers) only covers partially, blogs and github repositories seem to fill out the rest, and typically in a much more accessible fashion with great illustrations and example code. For this reason, we have tried to gather some of the more useful links to such blogs and repositories below. It is simply our (slightly random) selection of webpages that we have come across and found illustrative and useful. So use these at will, and also build your own list of reference sites.

Books:

- [Deep Learning](#) by Ian Goodfellow et al. (2016). A short and good general introduction to ML can be found in Chapter 5 of Part 1.
- [Pattern Recognition and Machine Learning](#) by Christopher M. Bishop (2006).
- ["Interpretable Machine Learning" by Christoph Molnar](#) (2020). A Guide for Making Black Box Models Explainable.
- ["Convolutional Neural Networks for Visual Recognition" by Andrej Karpathy](#) (2017?). Used for teaching CNN in Stanford's cs231 class.
- ["Deep Learning with Python" by author of Keras, Francois Chollet](#), now at Google AI (2017). Especially chapter 4 is a good overview of the fundamentals of Machine Learning.
- [Book on neural networks and deep learning](#). Online book only!
- [Applied Machine Learning](#) by David Forsyth. Great book, which is at times even mildly entertaining, from one of the masters of ML vision.

Papers:

- [XGBoost paper](#) (2016). Highly readable paper showing the innovations of the XGBoost algorithm.
- [LightGBM paper](#) (2017). Explaining the great speedup and showing examples of execution times.
- .

Blogs/Links/Tutorials - supervised learning:

- [Introduction to tree based learning](#). Very good introduction to the basics of tree based learning.
- [Introduction to neural net based learning](#). Very good introduction to the basics of Neural Net (NN) based learning.
- [SciKit Learn tutorial](#). Gives a quick introduction to ML in general and has code examples for SciKit Learn.
- [Simple introduction code to a simple Neural Network in PyTorch](#).
- [XGBoost vs. LightGBM](#). Discussion of differences, with code examples.
- [XGBoost, LightGBM, and CatBoost](#). Discussion of differences and hyperparameters.
- [Introduction to NGBBoost](#), which is a tree based algorithm, which makes a probabilistic predictions (i.e. uncertainties).
- [PyTorch Geometric Graph Neural Network Tutorial](#) (2019). Reasonably good guide with code examples.
- .

Blogs/Links/Tutorials - unsupervised learning:

- [SciKit-Learn manual for and discussion of \(unsupervised\) clustering](#).
- [Overview of t-SNE algorithm with papers and implementations](#).
- [Comparison of PCA, t-SNE, UMAP, and LDA](#) in a short visually nice writeup with code examples.
- [Isolation Forests \(Wiki\) and their implementation \(Towards Data Science\)](#) for anomaly detection.
- .


Blogs/Links/Tutorials - feature importance, re-weighting, optimisation, etc.:

- [Permutation importance](#) described in the SciKit-Learn implementation.
- [Permutation importance and vs Random Forest Feature Importance \(MDI\)](#) with code examples of usage.
- [GitHub repository for SHAPley value calculation](#), which gives game theory based variable rankings.
- [Shapley Values explained](#) in chapter 5.9 of C. Molnar's book.
- [Isolation Forests \(Wiki\) and their implementation \(Towards Data Science\)](#) for anomaly detection.
- [Re-weighting data samples using GBReweighter](#).
- [Description of the Adam optimiser](#). Three page overview of the ADAM optimiser (probably currently best Stochastic Gradient Descent algorithm).
- .

Code:

- [Oneliners for ML](#) mostly to display performance.
- [Plotting boundaries for ML classification](#).
- [Example of using UMAP for clustering](#) in an artistic way!
- .

On Loss functions:

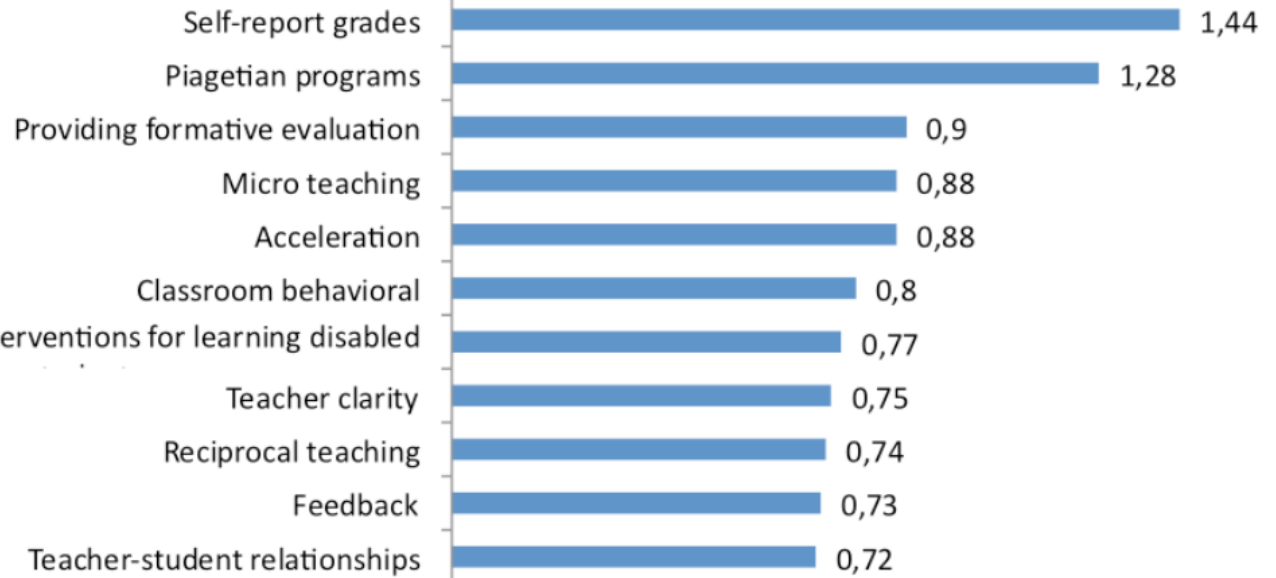
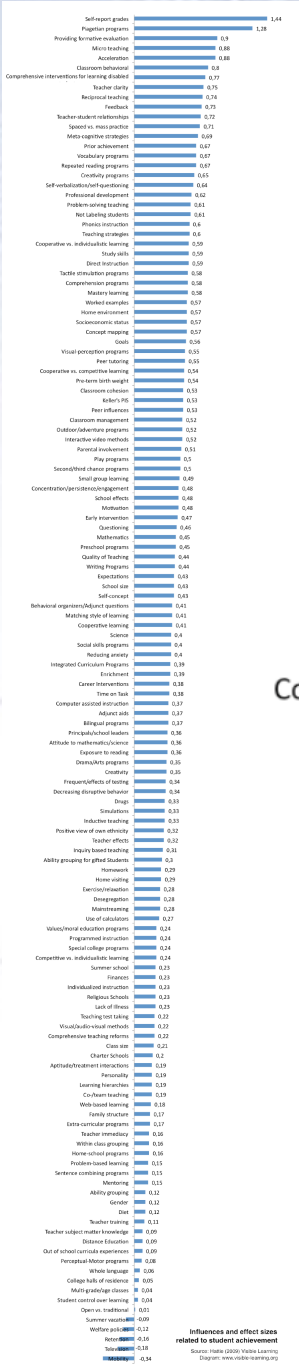
- [Cross Entropy](#) (log loss), which is one of the most used loss functions for classification.
- [Cross Entropy](#) illustrated.
- [Kullback-Leibler divergence](#) Screenshot  used for loss functions.



What influences learning?

What influences learning?

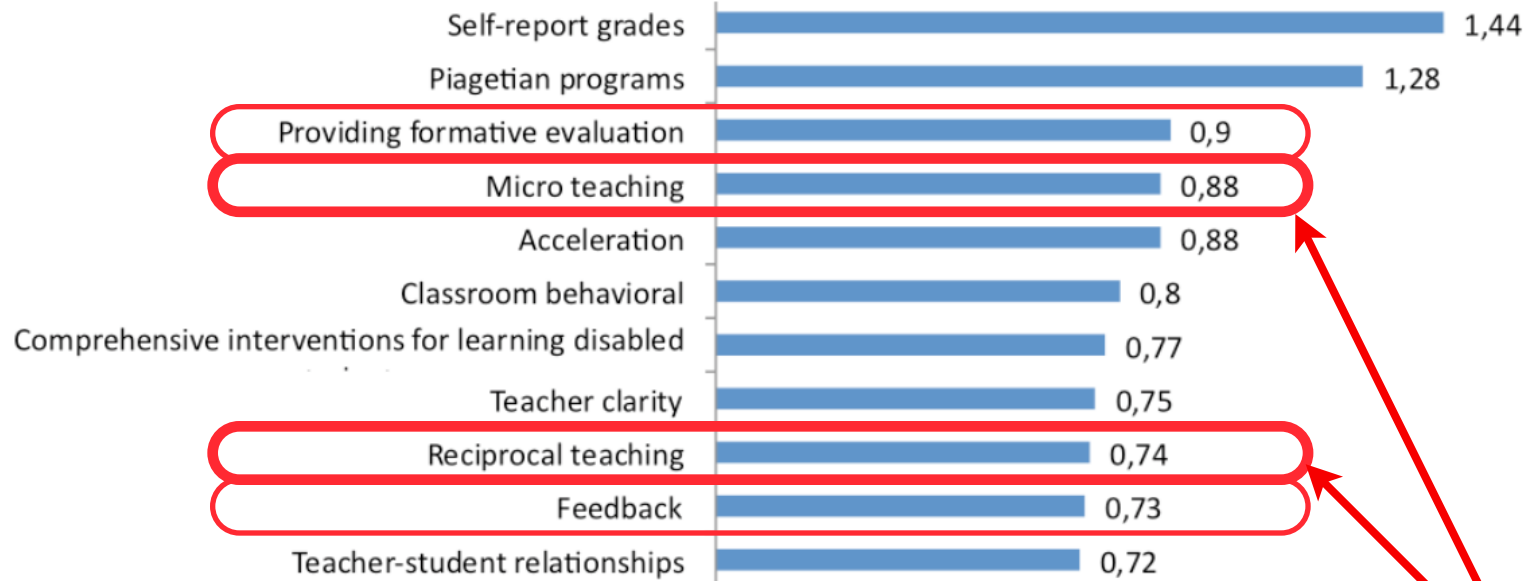
There are studies of this, one result shown below:



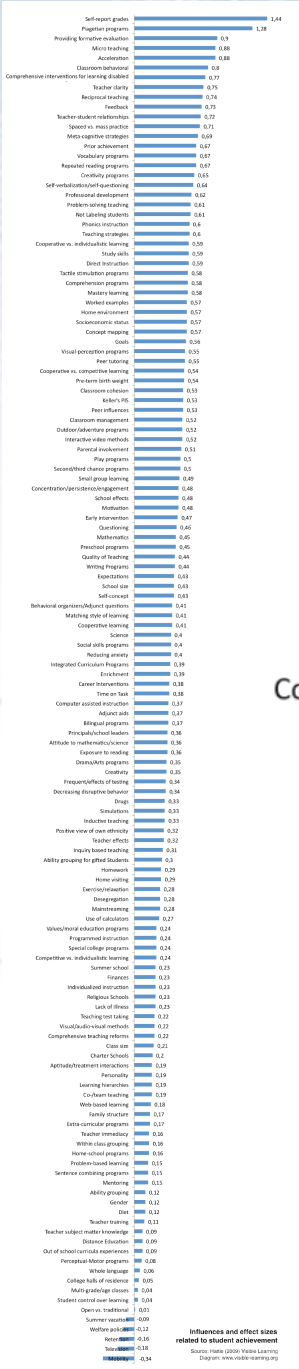
Influences and effect sizes related to student achievement
 Source: Hattie (2009) Visible Learning
 Dayani: www.visiblelearning.org

What influences learning?

There are studies of this, one result shown below:



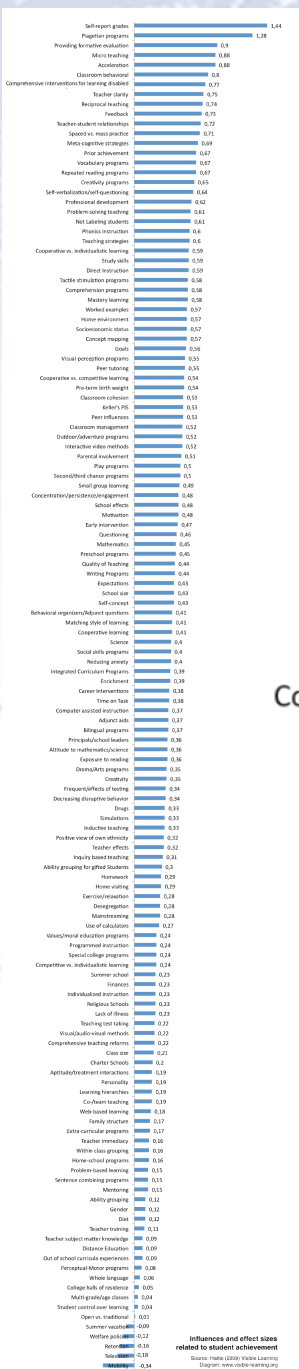
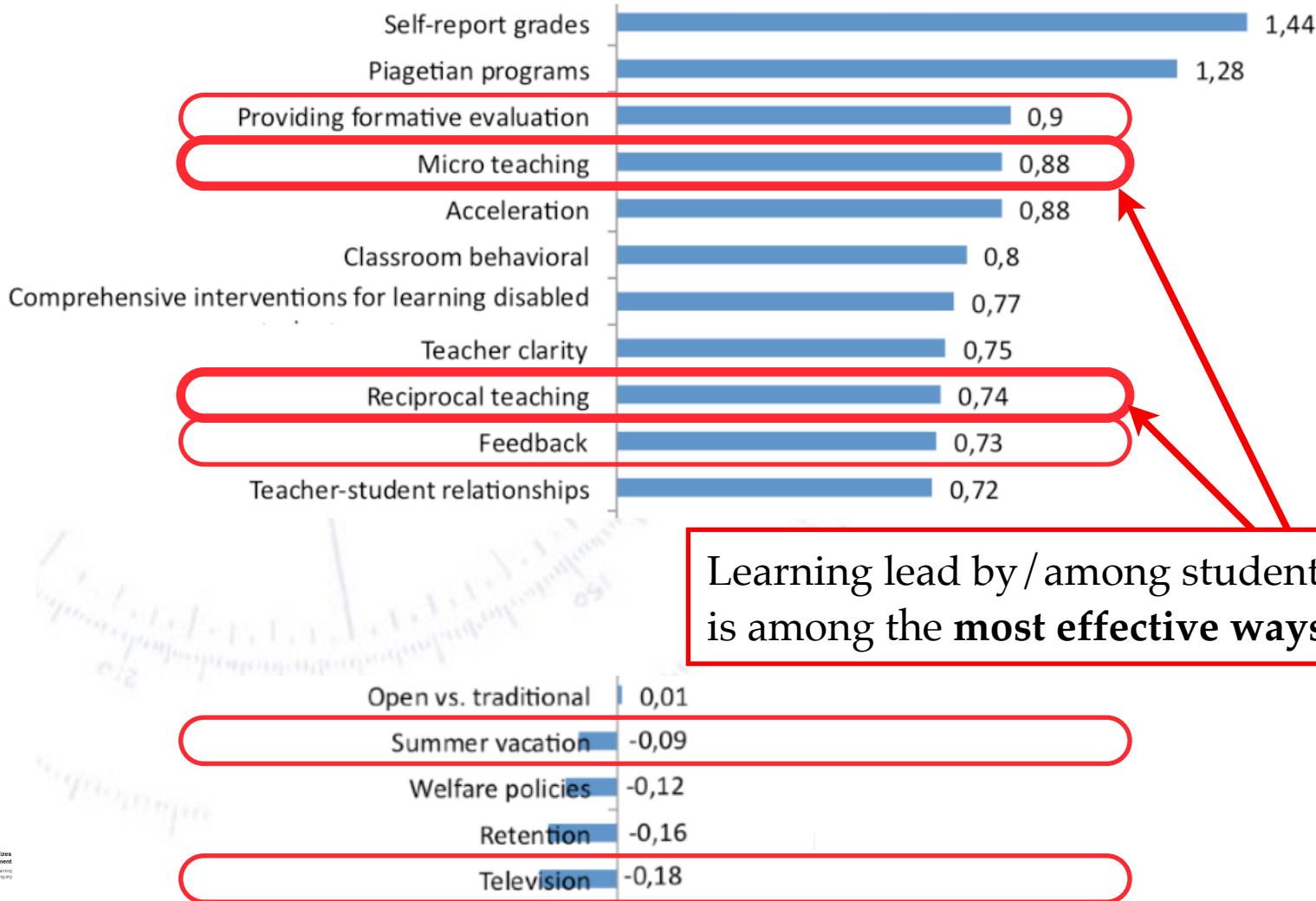
Learning lead by/ among students is among the **most effective ways!**



Influences and effect sizes related to student achievement
 Source: Hattie (2009) Visible Learning
 Dayani www.visible-learning.org

What influences learning?

There are studies of this, one result shown below:



Expectations

We want (read: insist) this course to be useful to all of you! Therefore, please give us feedback (the earlier the better), if you have anything to add/suggest/criticise/alter.

However, it is also a VERY independent course in the sense that it is up to YOU, how much you get out of it. Consider it as much a project as a course!

The aim is to make you knowledgeable about the basics of Machine Learning, and being able to apply it to (suitable) data.

Problems?

If you experience problems in relation to the course, whatever their origin and nature, then write us!

We may not be able to do anything about it, but if we don't know about your problems, then I most certainly can not do anything about them.

We consider ourselves fairly large, as long as I feel that this largeness is met by sincerity and will.

But... you need to write us in the first place! That is your responsibility.

Checklist

The following should be done by the end of the first week:

- Fill in questionnaire
- Ensure that Python runs with relevant packages
- Ensure that you got GitHub running
- **Absolutely ensure, that you have fellow students to work with!**

Lectures & Exercises

The lectures will (almost) always be:

Monday: 13:15-14:00

Wednesday: 9:15-10:00 and 13:15-14:00

Note: Wednesday 8:15-9:15 is “reserved” for your own work. We will not be checking if you do it (or other things) in that time range.

The exercises will (almost) always be:

Monday: 14:15-17:00

Wednesday: 10:15-12:00 and 14:15-17:00

Working on exercises

The exercises are meant to make you work through all the parts that make up “understanding and being able to use” Machine Learning.

If you don't understand the purpose or aim of the exercise with your peers, read through the exercise one more time (~5 minutes) and then ask us.

A. Work on exercise by yourself/in a group, preferably during exercise hours.

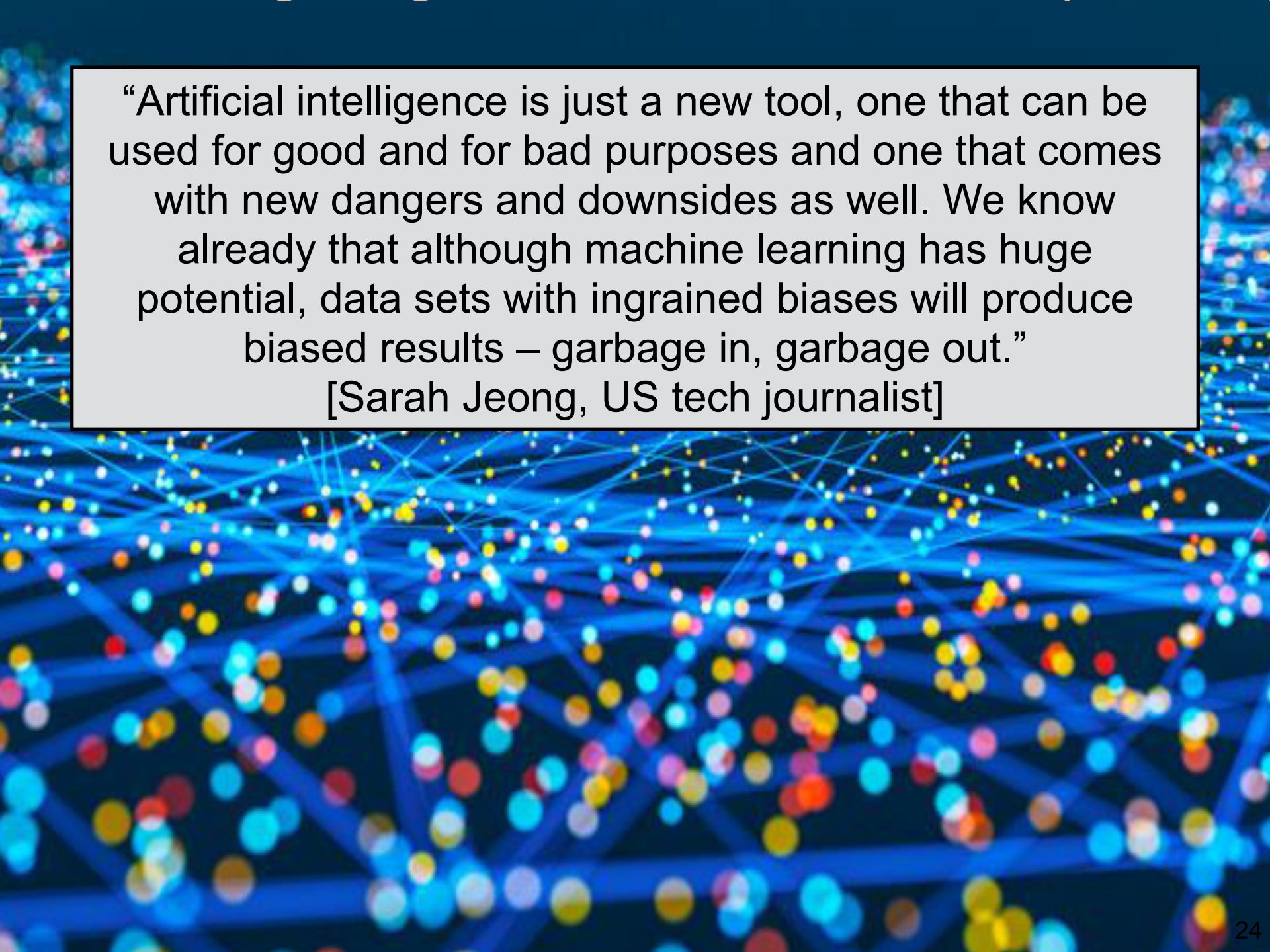
B. If you get stuck, discuss with your group/peers (for 5-10 minutes).
Of course also check “Dr. Google” for solutions!

C. If this doesn't solve it, call/write out in the chat and/or ask a TA.

D. If this doesn't solve it, we'll take it up in plenum.

E. If this doesn't solve it, we'll work out a solution for next class.

F. If this doesn't solve it, we'll write an ML paper on this interesting problem :-)

The background of the slide is a dense network graph. It consists of numerous small, multi-colored nodes (in shades of blue, yellow, red, and pink) connected by a web of thin, bright blue lines. The overall effect is a complex, interconnected web of data points, typical of a neural network or a large-scale data visualization.

“Artificial intelligence is just a new tool, one that can be used for good and for bad purposes and one that comes with new dangers and downsides as well. We know already that although machine learning has huge potential, data sets with ingrained biases will produce biased results – garbage in, garbage out.”
[Sarah Jeong, US tech journalist]