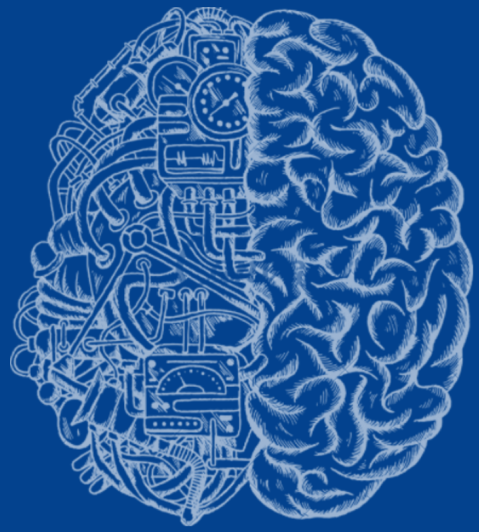
AARHUS UNIVERSITY

LESSON 1: Introduction



CARSTEN EIE FRIGAARD

SPRING 2025

"A COMPUTER PROGRAM IS SAID TO LEARN FROM EXPERIENCE E WITH RESPECT TO SOME CLASS OF TASKS T AND PERFORMANCE MEASURE P , IF ITS PERFORMANCE AT TASKS IN T , AS MEASURED BY P , IMPROVES WITH EXPERIENCE E." — MITCHELL (1997).

Agenda

1. ML crash-course
2. Kursus intro
   - Admin (eksamen, læringsmål, etc.)
   - Brightspace intro og online undervisning
3. Python/Anaconda intro
   - Opgave: `L01/modules_and_classes.ipynb`
4. ML intro
   - Opgave: `L01/intro.ipynb`

Underviser og hjælpelærer

Carsten Eie Frigaard:
kursusholder,
rum: E301/A,
email: `cef@ece.au.dk`

Henrik Daniel Kjeldsen
hjælpelærer,
email: `hdk@ase.au.dk`
KUN til opgaveretning!

Max Turunen
hjælpelærer,
email: `max@ece.au.dk`

Eksamen

Formel beskrivelse fra kursuskataloget

Prøveform

- Hjemmeopgave (?)

Bedømmelse

- Godkendt/Ikke godkendt, ingen censur.

Bemærkninger

- " I løbet af kurset skal et antal obligatoriske opgaver afleveres og der skal deltages i et antal obligatoriske præsentationer.

  Bedømmelsen af kurset sker på baggrund af én samlet vurdering af de afleverede opgaver og præsentationer, hvor der vil blive lagt vægt på, om den studerende opfylder punkterne i kvalifikationsbeskrivelsen.

  Bedømmelsen foretages kun af eksaminator (underviser). "

Reeksamen

- " Reeksamen: Næste ordinære eksamenstermin. Der skal afleveres nye opgaver og præsentationer. "

Eksamen

Afleveringer og evalueringer

Eksamensform, godkendelsesfag via:

- et sæt obligatoriske skriftlige gruppe-opgaver med afleveringsdeadlines (O1/O2/O3/O4),
- en poster-session, med aflevering af poster og *uformel* mundtlig præsentation af poster til andre studerende,
- en mundtlig gennemgang af den sidste afleveringsopgave (O4-rapport) med alle medlemmer i SWMAL gruppen, samt evaluering af hver gruppemedlems bidrag.

=> Endelig godkendelse af kurset sker på én samlet vurdering af de tre punkter ovenfor.

Opgaveafleveringer: O1, O2, O3 og O4

- O1: Opgavesæt fra L01+L02+.. (se Brightspace)
- O2: Opgavesæt fra ..
- O3: Opgavesæt fra ..
- O4: rapport, et mini-projekt:
  - For the final journal, you must design and implement a full machine learning system. You have relative free hands...

    Criterions [extract]:
    - Data must be split in a training-test set...
    - Your machine learning algorithm must be described in depth...
    - The system must be evaluated via a suitable performance metric...

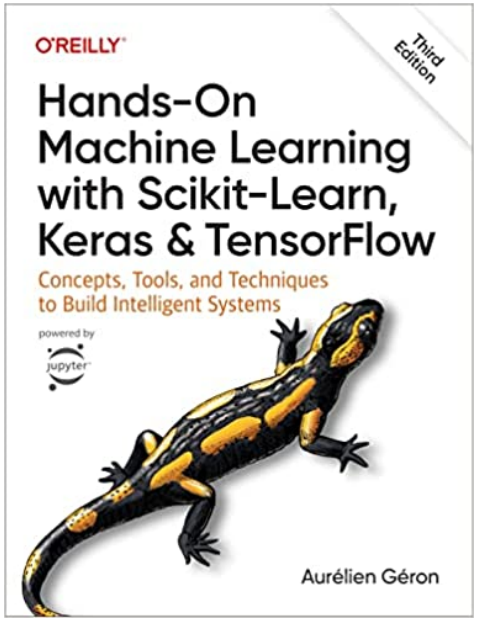
NOTE$_0$: Afleveringsformat i PDF eller Notebooks.
NOTE$_1$: O4 er blive specificeret på BS.

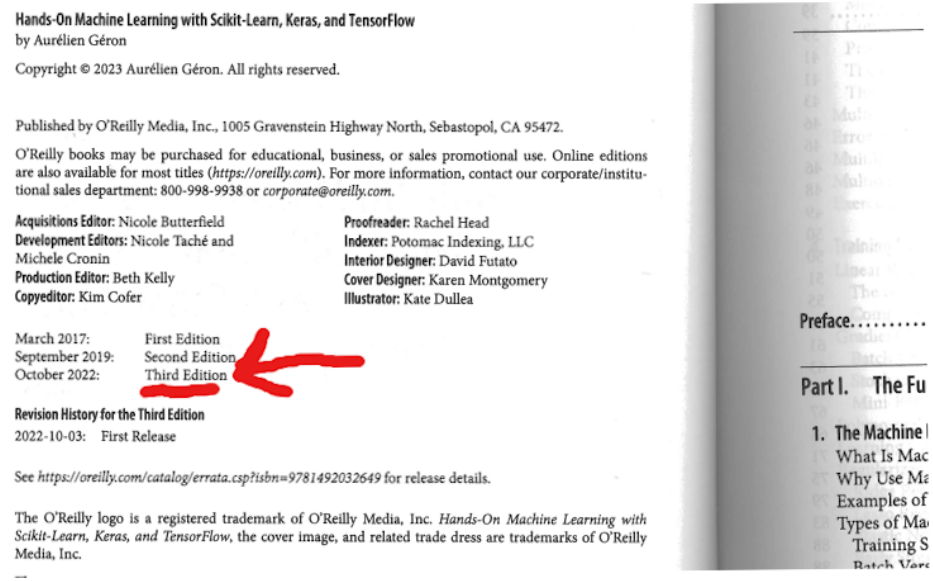
Læringsmål (SKIP slide!)

- SWMAL generelt:
  - **Redegøre** for de væsentligste begreber i machine learning terminologi samt principperne i en machine learning pipeline.
  - **Anvende** metoder til analyse af data, bl.a. med henblik på valg af machine learning model.
- SWMAL i relation til praktiske projekter:
  - **Anvende** udvalgte machine learning teknikker i praktiske opgaver og projekter.
  - **Anvende** udvalgte kodebiblioteker (frameworks) og udviklingsværktøjer til machine learning.
- ML Data og algoritmer:
  - **Beskrive** betydningen af datakvalitet i machine learning, samt anvende udvalgte databehandlings-teknikker til at forbedre kvaliteten af datagrundlaget.
  - **Sammenligne og vurdere** forskellige algoritmer og teknikkers anvendelighed i forbindelse med praktiske projekter.

Litteratur

"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" ,
Aurélien Géron, O'Reilly, 2022, 3.ed.




Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
by Aurélien Géron

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*https://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

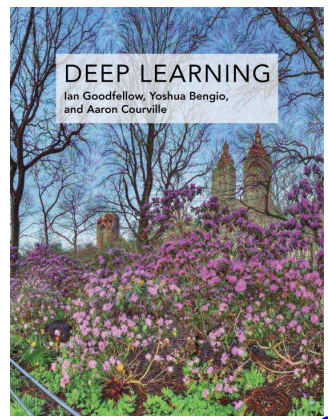
Acquisitions Editor: Nicole Butterfield
Development Editors: Nicole Taché and Michele Cronin
Production Editor: Beth Kelly
Copyeditor: Kim Cofer
Proofreader: Rachel Head
Indexer: Potomac Indexing, LLC
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Kate Dullea

March 2017: First Edition
September 2019: Second Edition
October 2022: Third Edition

Revision History for the Third Edition
2022-10-03: First Release

See *https://oreilly.com/catalog/errata.csp?isbn=9781492032649* for release details.

- [HOML]→ udtales som (Brian) Holm!
- Ref. til sidetal er for 3.Ed / *Third Release*.
- Plus yderligere materiale (brug links i BS).



Lektionsplan (fra BS)

Uge	Dato	Lektion	Emne	Opgave	Kommentar
05	28/01-2025	L01	Intro		
06	04/02-2025	L02	Klassifikation		
07	11/02-2025	L03	End-to-end ML	O1 (12/02)	CEF:vinterferie
08	18/02-2025	L04	Data analyse		
09	25/02-2025	L05	Træning		
10	04/03-2025	L06	Neurale net (NN)		
11	11/03-2025	L07	Convolutional NN (CNN)	O2 (12/03)	
12	18/03-2025	L08	Generalisering		
13	25/03-2025	L09	Regularisering og søgning		
14	01/04-2025	L10	Sprogmodeller (RNN, LLM, GPT)		
15	08/04-2025	L11	Avanceret CNN	O3 (09/04)	
16	15/04-2025				Påske, ingen UV
17	22/04-2025	L12	Framworks and Hardware (GPUs)		
18	29/04-2025	L13	O4 projekt		
19	06/05-2025	L14	O4 projekt		
20	13/05-2025	L15	O4 projekt aflevering	O4 (12/05)	Kursus-evaluering, O4-eval
21	20/05-2025				O4-eval

SWMAL Nomenklatur

[HOML]: Hands-On Machine Learning bog, aka (B.)Holm.

[GITHOML]: Git repository for [HOML].

[GITMAL]: Git repository for SWMAL kursus opgaver, (bruges meget kun lidt)

[G]: eller Grp, SWMAL gruppe, med tre studerende„ (evt. to/fire).

[SG]: SWMAL super-gruppe, ved nogle af opgaverne.

[O1]: opgavesæt 1, osv. (O2/O3/O4).

[L01]: Lektion 1, osv.

NOTE: se fuld liste på *'BS | Kursusinfo | Kursusforkortelser'*.

END Kursus intro/
BEGIN Python intro

python Introduction

- Python is an **interpreted** high-level programming language for general-purpose programming. Created by **Guido van Rossum** and first released in 1991, Python has a design philosophy that emphasizes **code readability**, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.
- Python features a **dynamic type system and automatic memory management**. It supports multiple programming paradigms, including **object-oriented**, **imperative**, **functional and procedural**, and has a large and comprehensive standard library.
- Python interpreters are available for many operating systems.

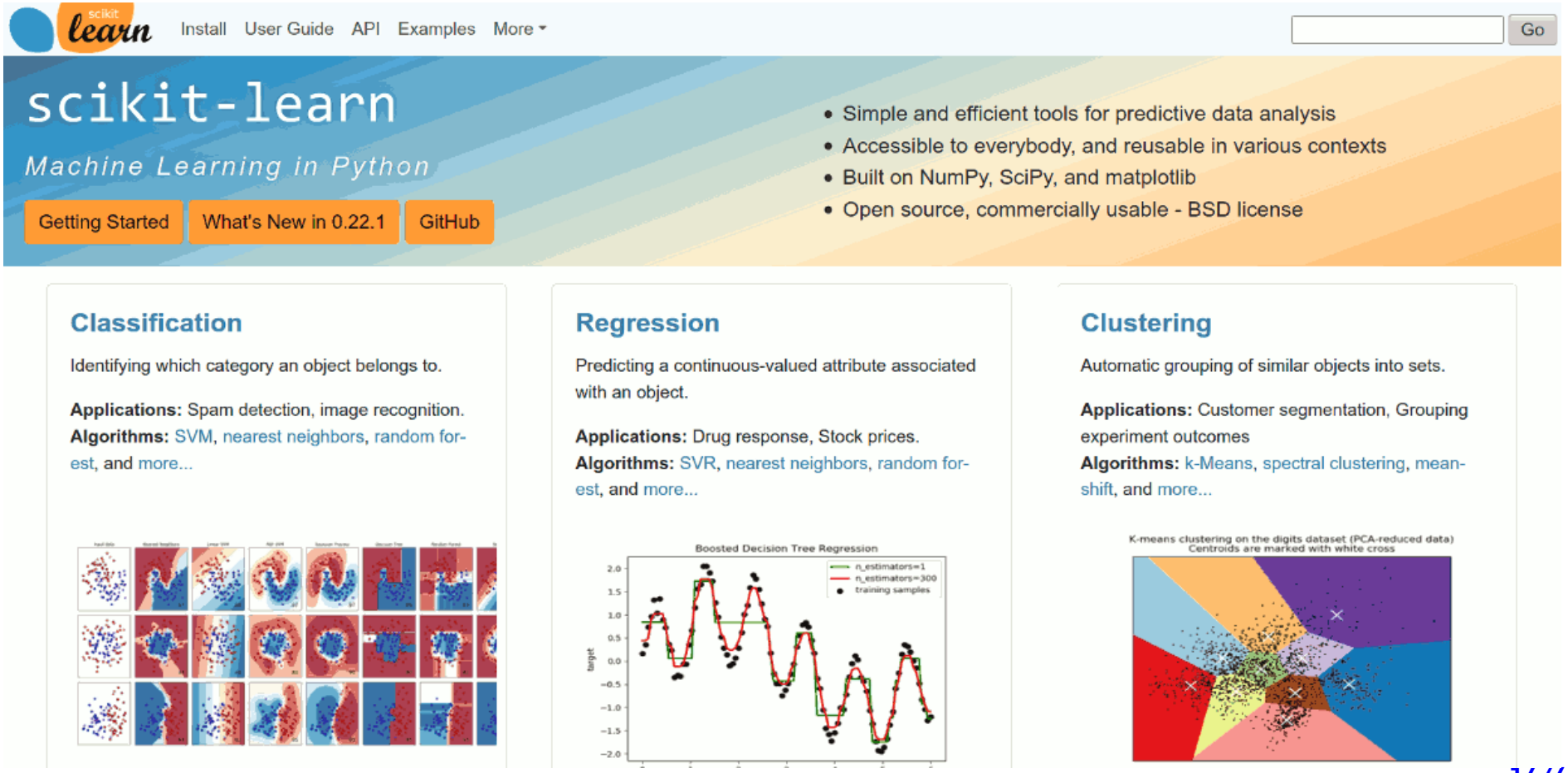
[https://en.wikipedia.org/wiki/Python_(programming_language)]

Anaconda and Jupyter Introduction

- **Anaconda**: a python distribution [https://www.anaconda.com].
- **Jupyter notebook**: interactive python development environment (GUI IDE), distributed with the Anaconda package.
- Jupyter is an anagram of: Julia, Python, and R.
- Jupyter notebook method:
  - √ polyglot environment, mixing source code, markdown test and formulas (LaTeX),
  - √ interactive/exploravt/trial-and-error environment,
  - ÷ not good at source-code level debugging.
- Other IDE's:
  - Spyder (Anaconda),
  - VSCode (Microsoft),
  - and many others...

Scikit-learn Introduction

- Scikit-learn: a framework (API + website) for machine Learning in python.
- `http://scikit-learn.org`
- `git@github.com:scikit-learn/scikit-learn.git`

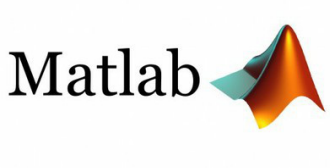


Vores videnskabelige framework

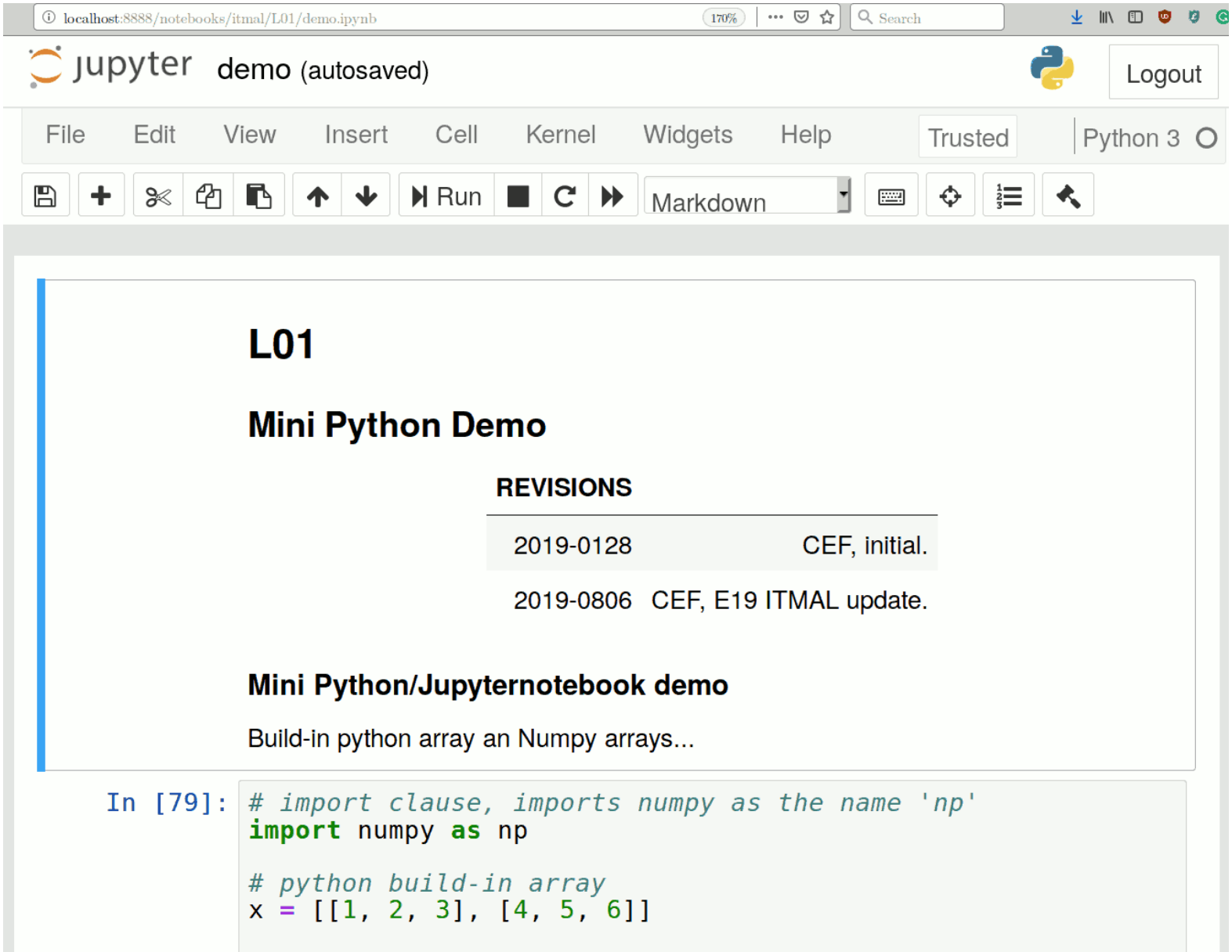
Sat sammen...

Gode hjælpe og dokumentations-systemer..

Alternativer kunne være...



Python (Anaconda) and Jupyter Notebook Demo

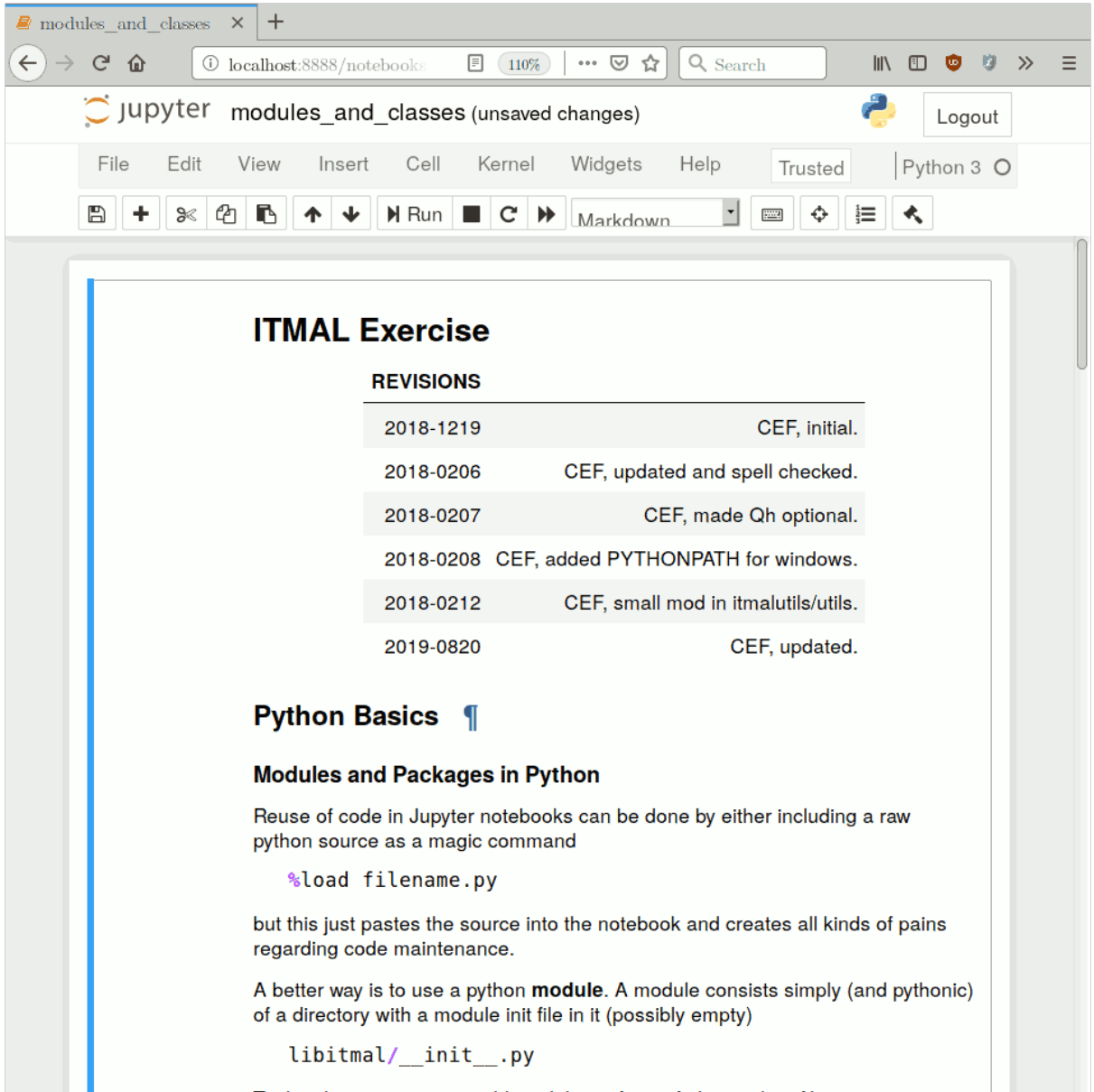


Anaconda and Jupyter Demo: Highlights...

- Polyglot miljø:
  - lidt ala Matlab IDE,
  - markdown (HTML+LaTeX)-og-Python-i-een = polyglot,
  - alt kører i browser, lokalt eller på server.
- Quickstart:
  - åbn via `http://localhost:8888` (efter launch),
  - ENTER på celle: editer celle,
  - CTRL+ENTER: kør celle,
  - SHIFT+TAB: hjælp på funktion,
  - TAB: tab-completion.
- Magics:
  - nulstil vars: `%reset -f`,
  - inline plots: `%matplotlib inline`.
- Hints:
  - Pas på globale vars (igen scopes ml. `.ipynb` celler),
  - Brug menu *'Help'* og
    find shortcuts i *'open command palette'*n,
  - Hvis du er C++ haj: alt er anderledes!

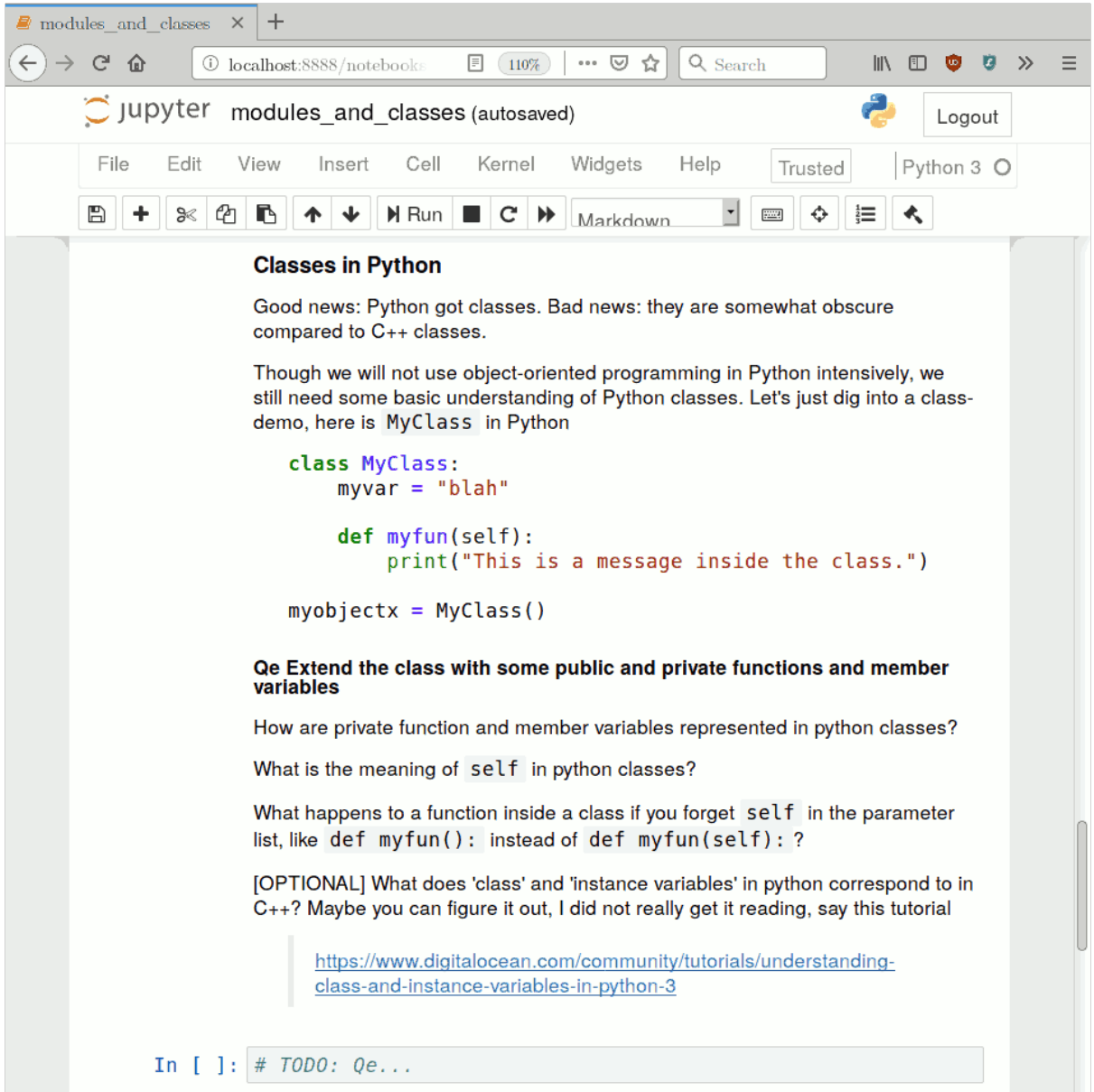
Q: L01/modules_and_classes.ipynb

Modules and Packages...

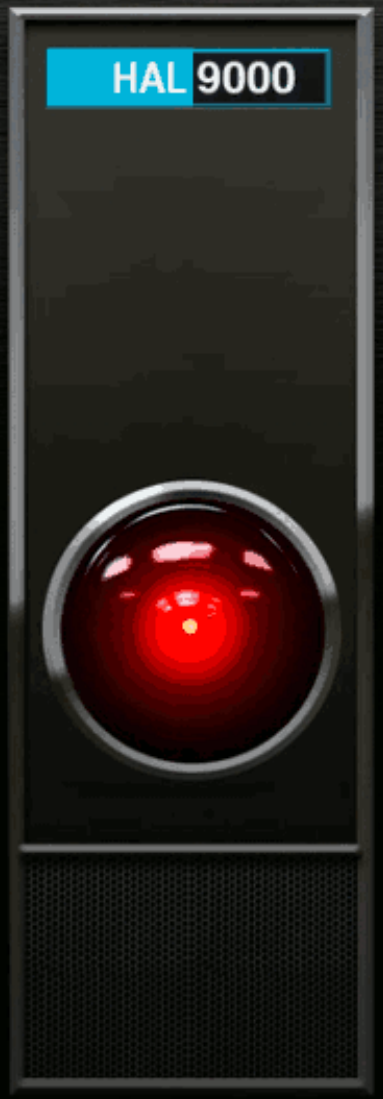


Q: L01/modules_and_classes.ipynb

Python classes...

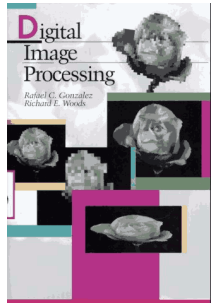


END Python intro/BEGIN ML intro



Klassisk maskinlæring årgang 1992

Pattern recognition, machine vision, neural networks...



Digital Image Processing, Gonzalez and Woods, 1992

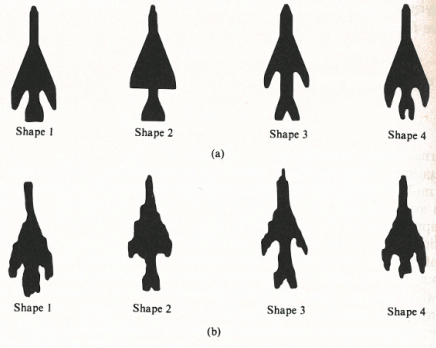


Figure 9.18 *(a) Reference shapes and (b) typical noisy shapes used in training the neural network of Fig. 9.19. (From Gupta et al. [1990].)*

Pattern vectors were generated by computing the normalized signatures of the shapes (see Section 8.1.3) and then obtaining 48 uniformly spaced samples of each signature. The resulting 48-dimensional vectors were the inputs to the three-layer feedforward neural network shown in Fig. 9.19. The number of neuron nodes in the first layer was chosen to be 48, corresponding to the dimensionality of the input pattern vectors. The 4 neurons in the third (output) layer correspond to the number of pattern classes, and the number of neurons in the middle layer was heuristically specified as 26 (the average of the number of neurons in the input and output layers). There are no known rules for specifying the number of nodes in the internal layers of a neural network, so this number generally is based either on prior experience or simply chosen arbitrarily and then refined by testing. In the output layer, the four nodes from top to bottom in this case represent classes ω_j, $j = 1, 2, 3$, and 4, respectively. After the network structure has been set, activation functions have to be selected for each unit and layer. All activation functions were selected to satisfy Eq. (9.3-50) so that, according to the earlier discussion, Eqs. (9.3-72) and (9.3-73) apply.

The training process was divided in two parts. In the first part, the weights were initialized to small random values with zero mean, and the network was

then trained with pattern vectors corresponding to noise-free samples like the shapes shown in Fig. 9.18(a). The output nodes were monitored during training. The network was said to have learned the shapes from all four classes when, for any training pattern from class ω_i, the elements of the output layer yielded $O_i \geq 0.95$ and $O_q \leq 0.05$, for $q = 1, 2, \ldots, N_Q$; $q \neq i$. In other words, for any pattern of class ω_i, the output unit corresponding to that class had to be high (≥ 0.95) while, simultaneously, the output of all other nodes had to be low (≤ 0.05).

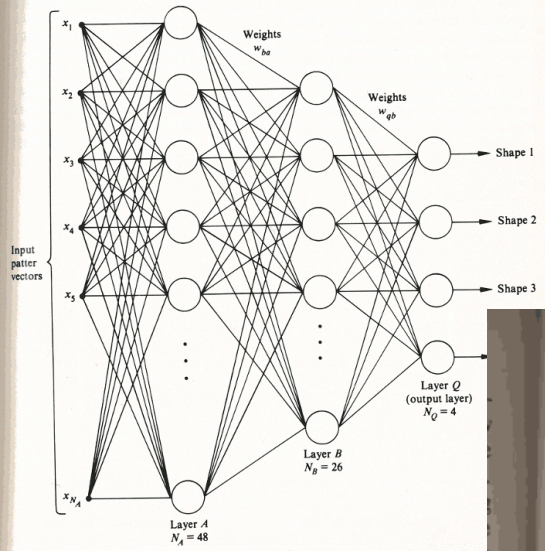


Figure 9.19 *Three-layer neural network used to recognize the shapes in Fig. 9.18*

- Indeholder allerede det 'meste' ML,
- ML "vintre og somre":
  90'erne=sommer,
  00'erne=vinter

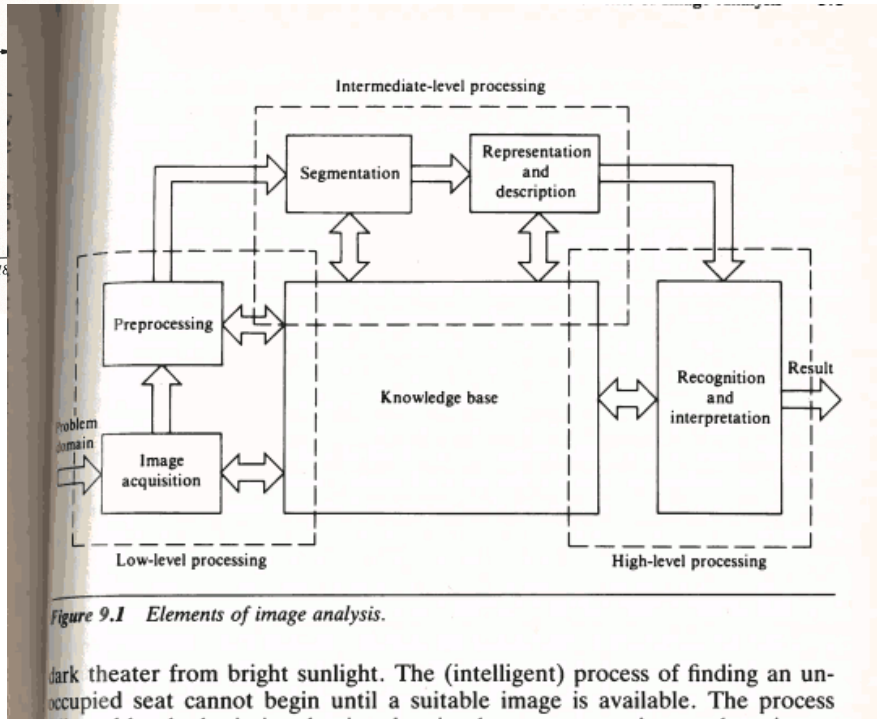


Figure 9.1 *Elements of image analysis.*

dark theater from bright sunlight. The (intelligent) process of finding an unoccupied seat cannot begin until a suitable image is available. The process

Moderne maskinlæring, renæssancen

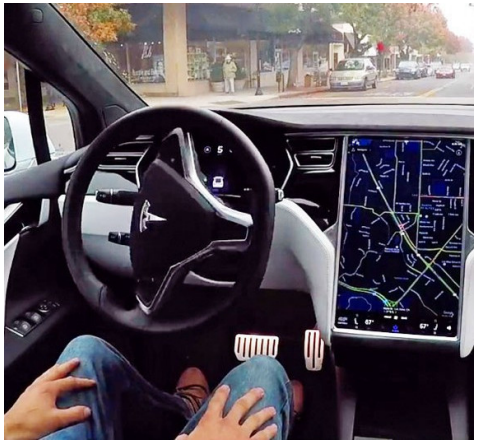
SciFi:

Real-tech examples OUT-DATED!

Real:

Apple's Siri

IBM's Watson

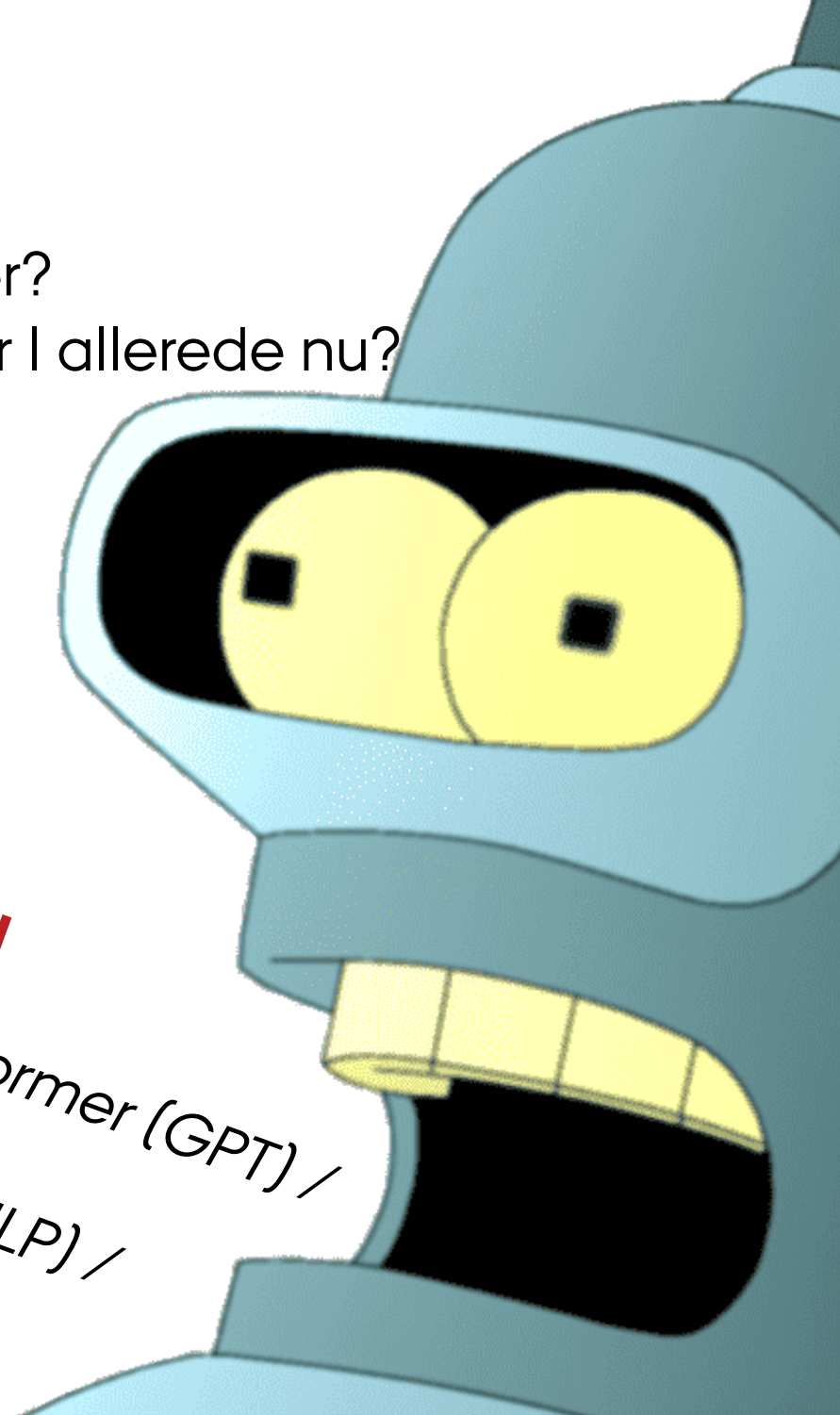


Tesla's selvkørende bil

Moderne maskinlæring

Diskussion: hvad er ML for Jer?

- hvad kender I af ML systemer?
- hvilke ML systemer anvender I allerede nu?
- ...og andre ML relaterede kommentarer!



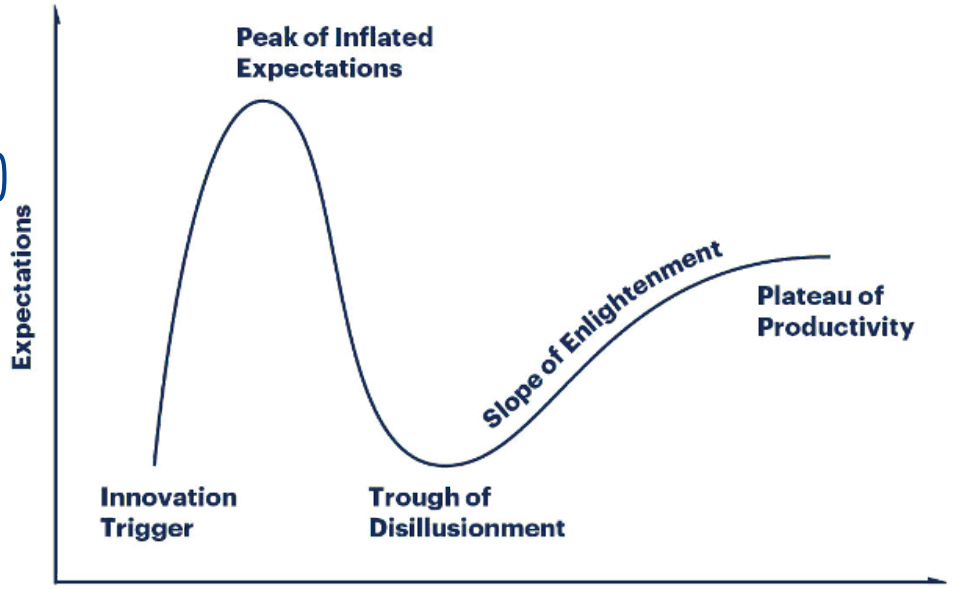
ChatGPT!

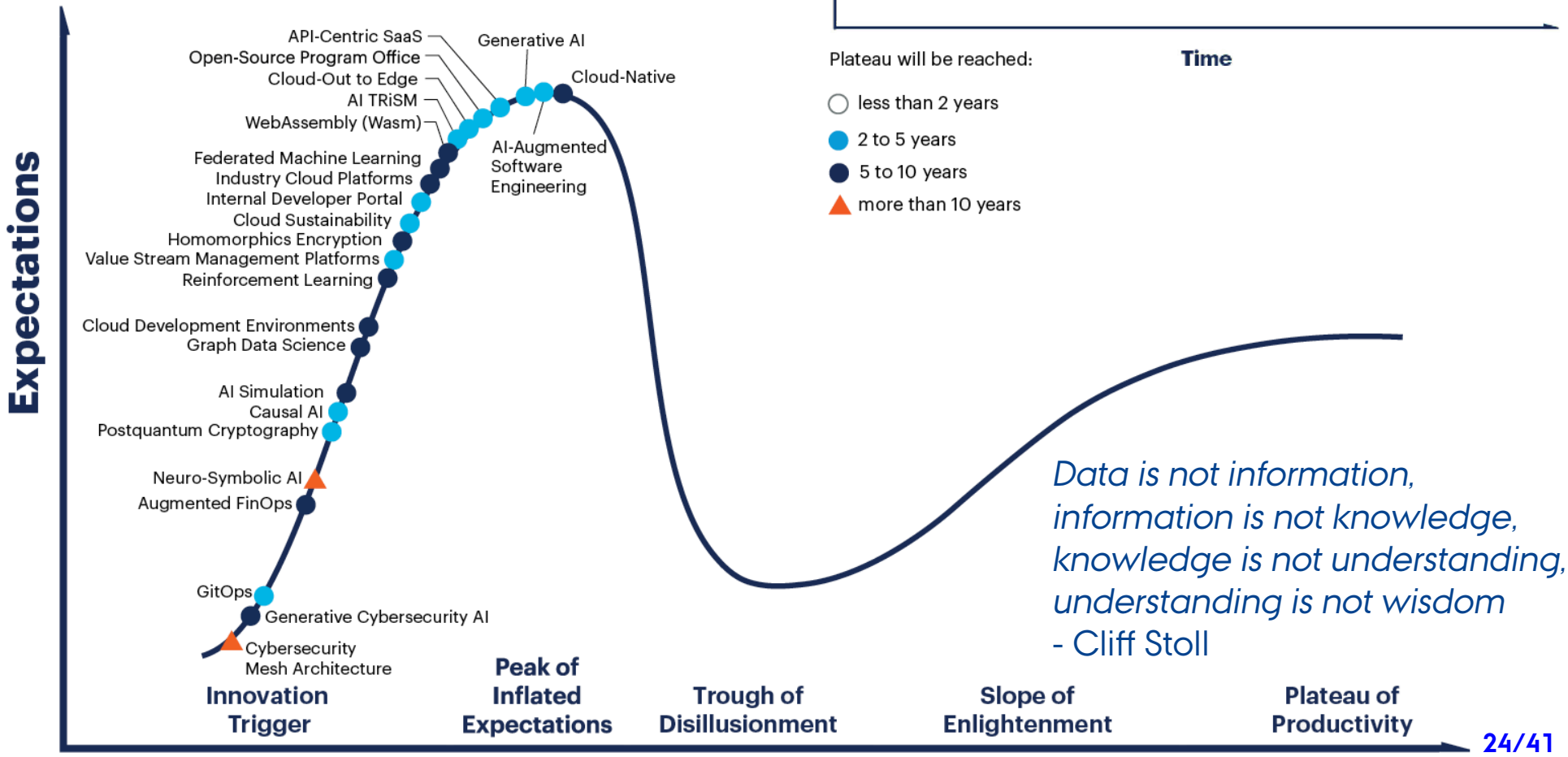
Generative Pre-trained Transformer (GPT) /
Large language model (LLM) /
Natural language processing (NLP) /
Deep learning algorithms (DL)
..and Soft Bullshit ?

Moderne maskinlæring

Forskel på ML og AI (artificial intelligence)

- Er det skrevet i PowerPoint
  så er det AI.
- Er det skrevet i Python
  så er det ML.





Data is not information,
information is not knowledge,
knowledge is not understanding,
understanding is not wisdom
- Cliff Stoll

BREAKING
NYHEDER SPORT UNDERHOLDNING

LIGE NU:

ALLE AI OVERSKRIFTER HAVDE MODALVERBER I NUTID (*skal give*)

MEN SÅ KOM DENNE.....

BREAKING NEWS

Første ML artikel i datid: *"Sådan gav ML resultater.."*

Fra klassisk til moderne maskinlæring

Stadig pattern recognition, machine vision, neural networks...

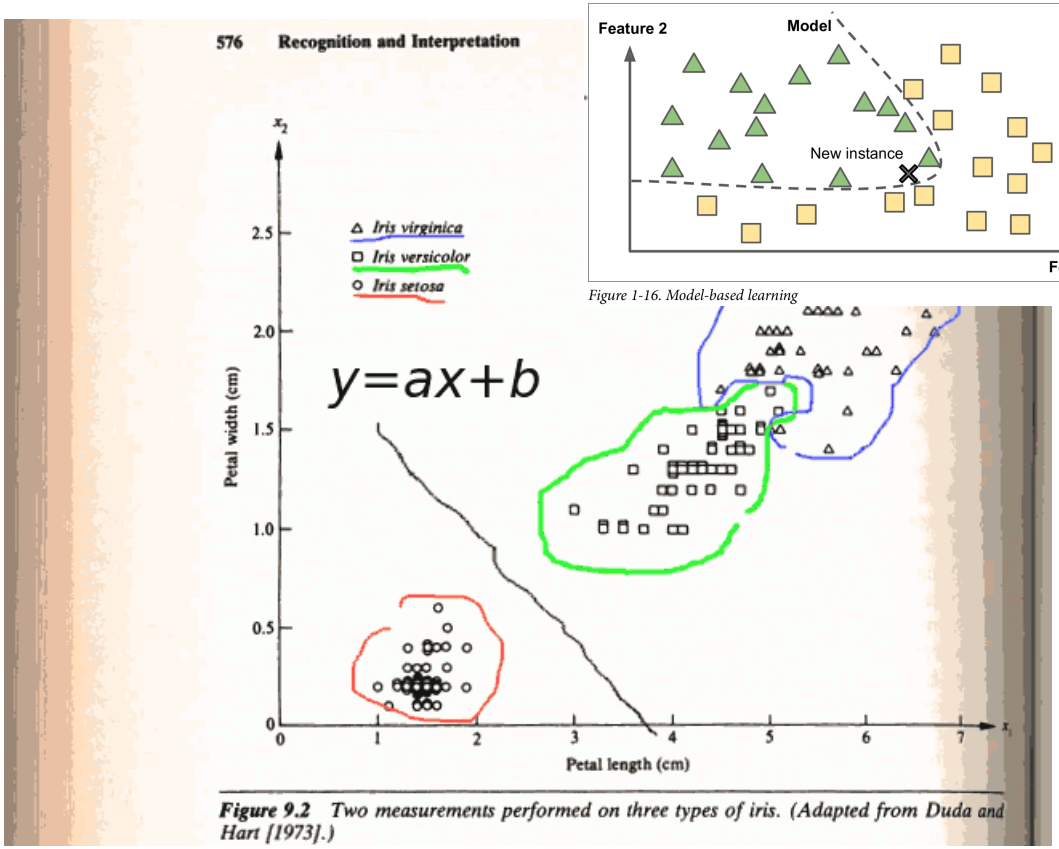
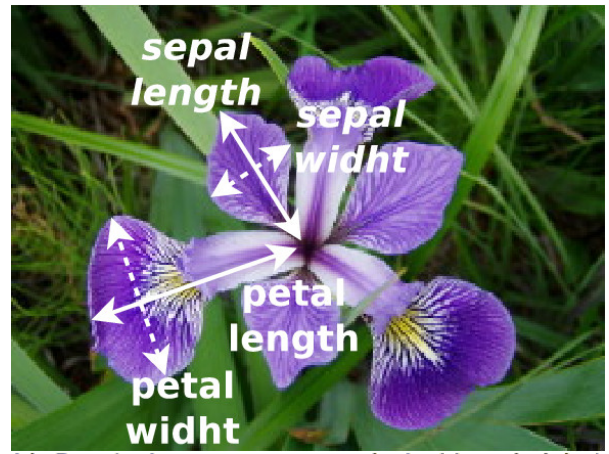
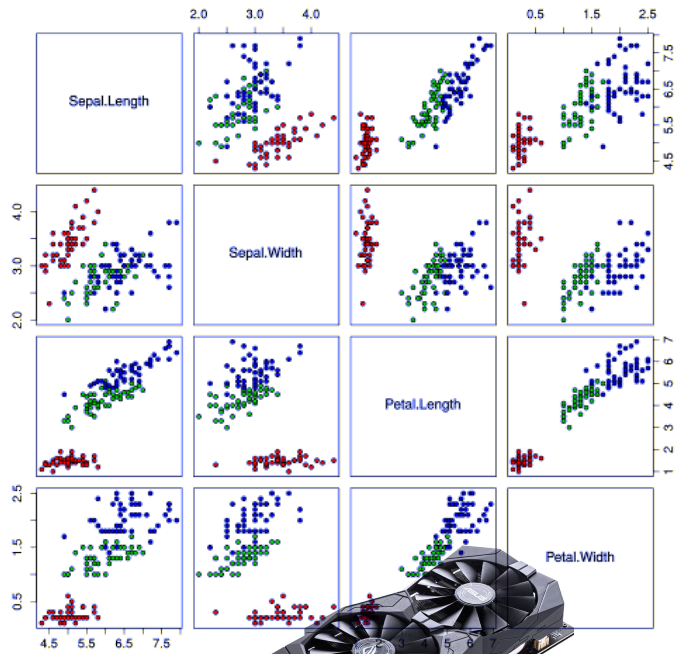


Figure 9.2 Two measurements performed on three types of iris. (Adapted from Duda and Hart [1973].)

Figure 1-16. Model-based learning



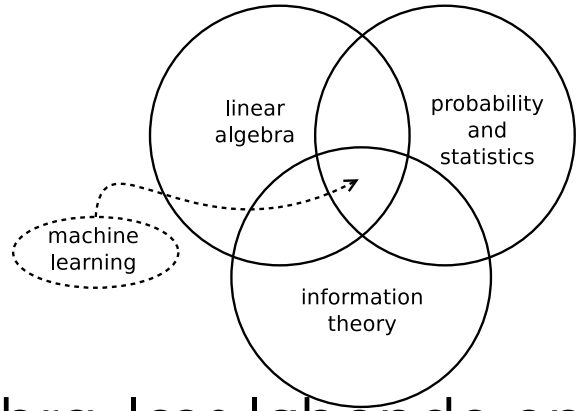
Iris Data (red=setosa,green=versicolor,blue=virginica)



- ikke-nyt$_1$: matematik,
- ikke-nyt$_2$: algoritmer,
- nyt$_1$: meget mere data og flere dimensioner, f.eks. 4D til 784D,
- nyt$_2$: hurtigere hardware (og parallelitet).

Machine learning baggrund

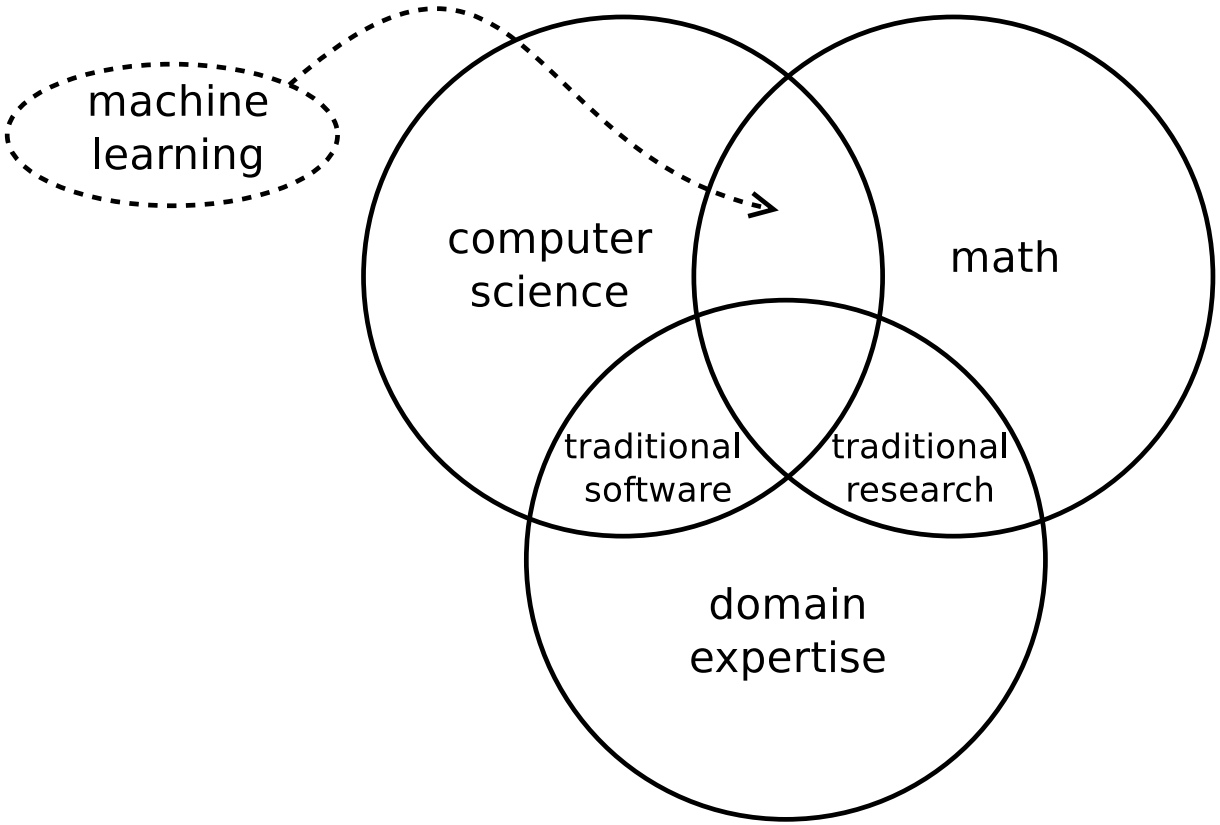
Machine learning: matematisk baggrund



- Lineær algebra, læs løbende op:
  - norm (afstand),
  - matrix algebra (mest multiplikation),
  - least-square closed solution, $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$,
  - nabla operator, $\nabla_{\mathbf{w}} = [\frac{\partial}{\partial\, w_1}, \frac{\partial}{\partial\, w_2}..]$,
- Sandsynlighedsregning, læs minimalt op på:
  - ~~multivariate~~ mean, variance,
  - ~~multivariate~~ Gaussisk distribution,
  - ~~(Bayes',)~~
- Informationsteori: vi navigere (mest) udenom entropi og andre informations-teori elementer i dette kursus.

Machine learning baggrund

Machine learning: ekspertise

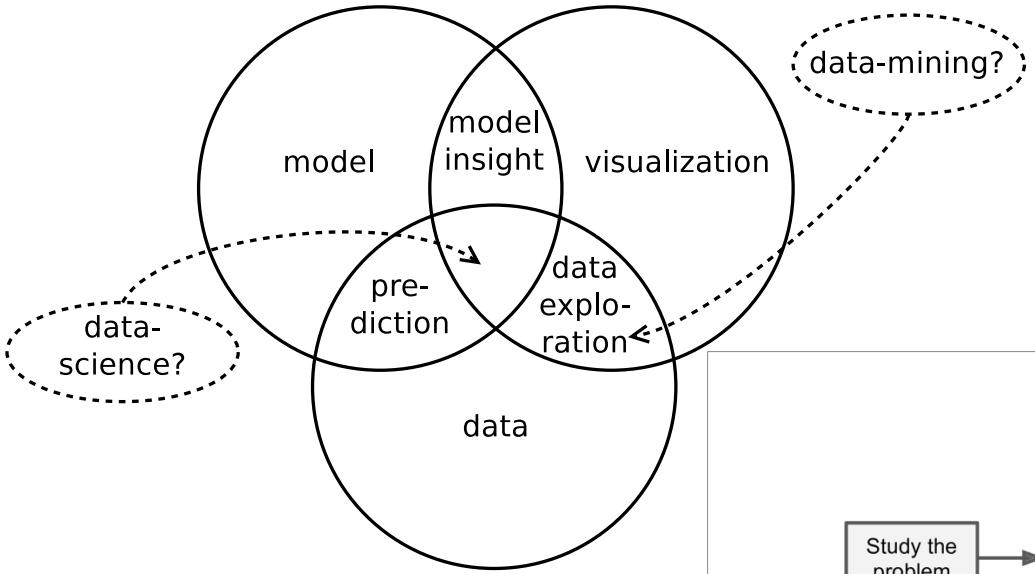


- ML ekspert er computer science og matematik ekspert.
- ML ekspert er IKKE (nødvendigvis) domæne ekspert!

NOTE:
[https://imarticus.org/what-are-the-skills-you-need-to-become-a-machine-learning-engineer/]

Machine learning baggrund

Machine learning: data science ekspert



- fra white-box domæne ekspert til black-box ML data scientist,
- stadig polytekniker:
  - math- og computer science,
  - pattern-recognition,
  - neurocompution,
  - datamining,
  - visualization,
  - etc..

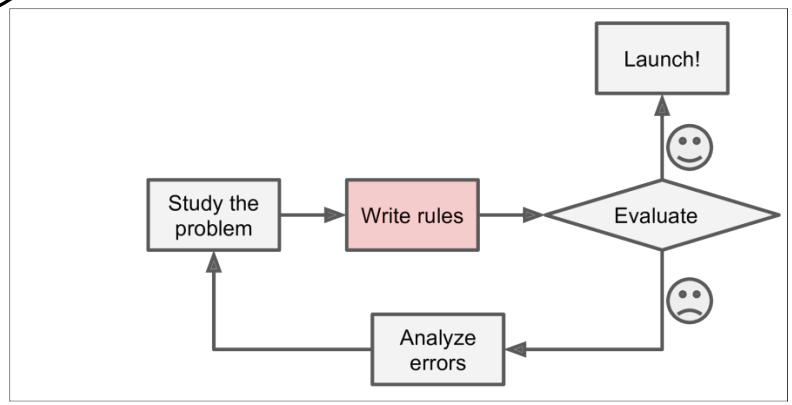


Figure 1-1. The traditional approach

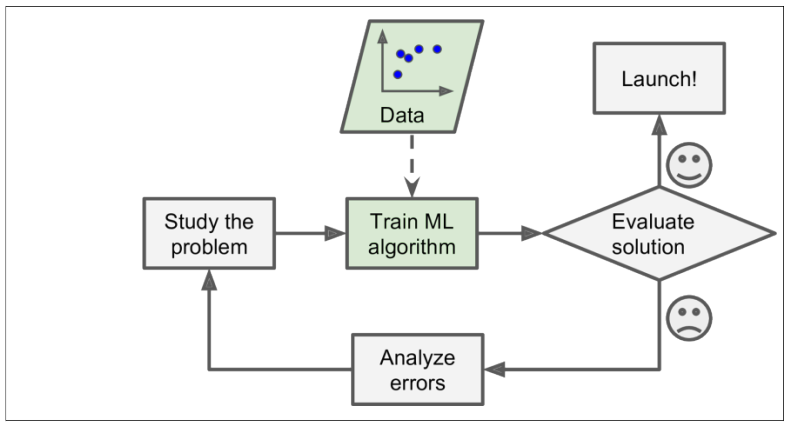
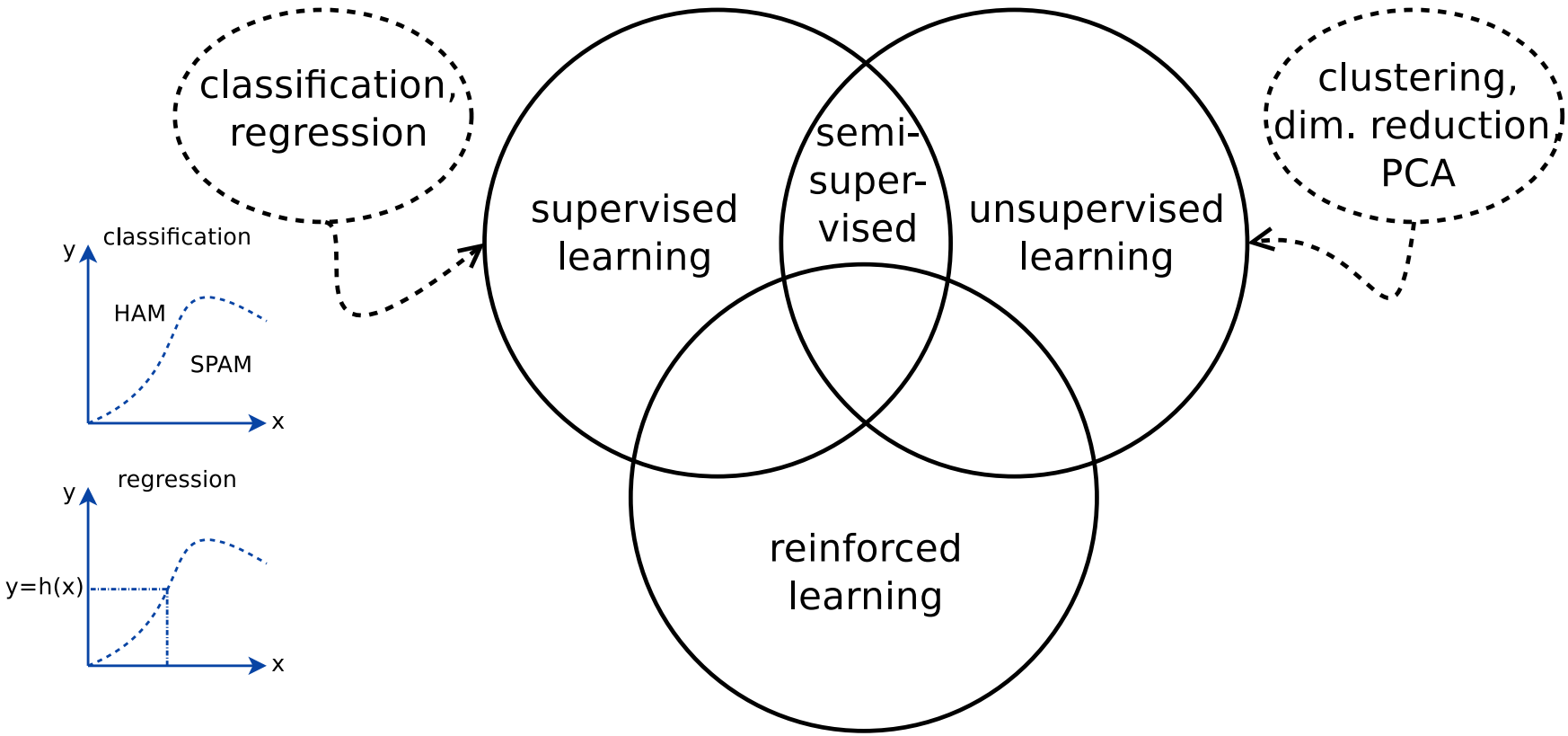


Figure 1-2. Machine Learning approach

Machine learning taksonomi

Machine learning læringstyper



Machine learning terminologi

X, **x**: input data matrix og vektor,

y, y: output data vektor og skalar,

θ or **w**: model parametre,

h: hypothesis funktion; typer af ML algos:

Bayes classifier, k-Nearest Neighbors, Linear Reg., Logistic Reg., SVM, Decision Trees, Random Forest, Neural Networks, k-Means, ...

$\mathbf{y}_{\text{true}}$: ground truth, til supervised learning,

$\mathbf{y}_{\text{pred}}$: predikteret værdi, aka $\hat{\mathbf{y}}$,

attribut: data type, f.eks. salgspris, dog anvendes 'feature' typisk i stedet for attribut!

λ, feature: data attribut plus value, f.eks. $\lambda_{\text{salgspris}} = \$42$,

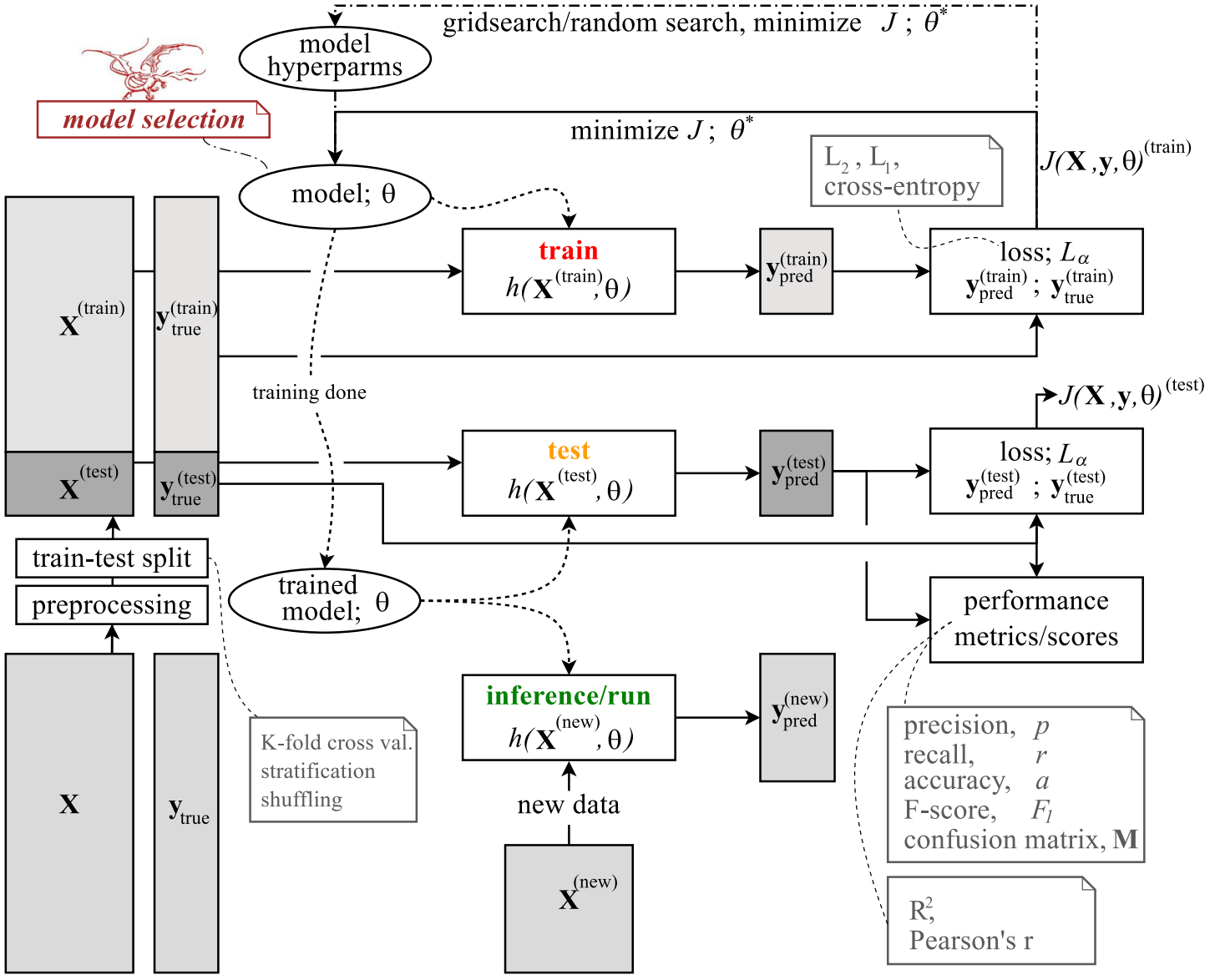
J, L, loss fun.: loss/cost/error/objective funktion, som minimeres via fitting, jo lavere jo bedre et fit,

score fun.: score/fitness/goodness funktion, jo højere

performance-metric: bruges typisk efter fit-minimeringingen til model inspektion og eftervalidering.

Supervised learning, blok diagram

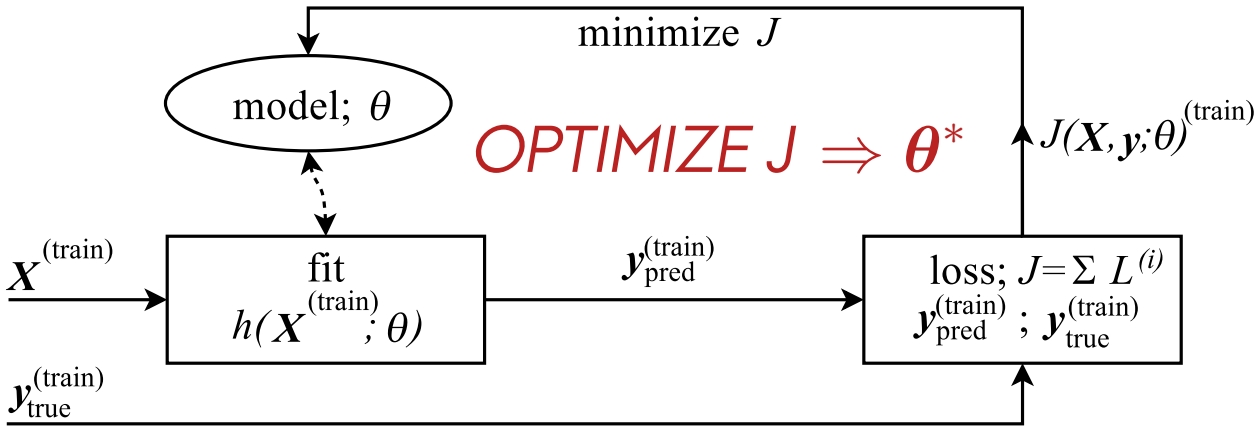
Sneak-preview af 'the full monty'...



NOTE: Kun et preview; vi går igennem detaljerne i figuren i de følgende lektioner.

Q: L01/intro.ipynb

ML supervised learning data flow model: i) Training (fit).



$\mathbf{X}^{(\text{train})}$: trænings matrix input data,

$\mathbf{x}^{(\text{train})}$: data input vector; $\mathbf{x} = [x_1, x_2, \cdots, x_d]$,

$\mathbf{y}_{\text{true}}^{(\text{train})}$: trænings input ground truth vektor,

$\mathbf{y}_{\text{pred}}^{(\text{train})}$: predikteret værdi for y, aka $\hat{y}$

$\boldsymbol{\theta}$: model parametre,

h : hypothesis funktion, aka. ML algoritmen,

$L^{(i)}$: loss funktion (individuel), $L^{(i)}(y_{\text{pred}}^{(i)}, y_{\text{true}}^{(i)})$

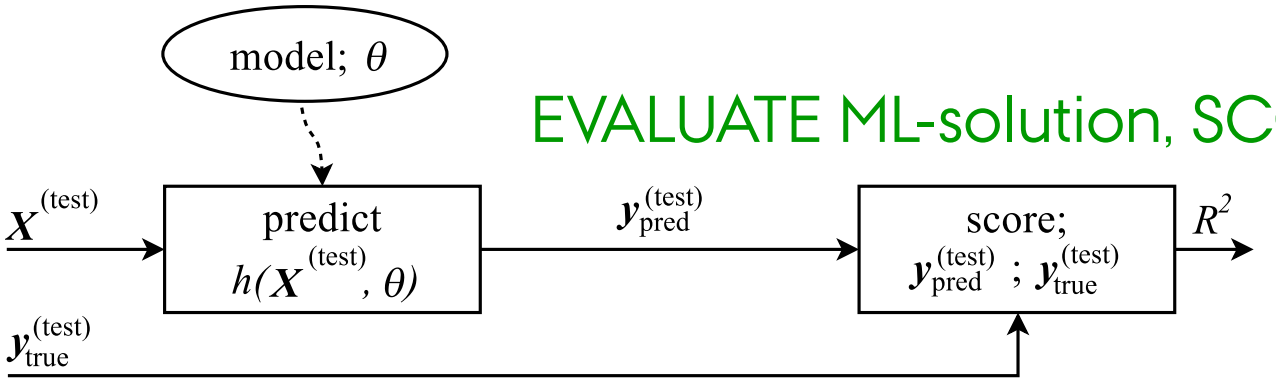
J : loss funktion (summeret), $J = \frac{1}{n} \sum_i L^{(i)}$.

NOTE: med **x** havende dimensionalitet d... mere om denne og loss funktioner i L02.

Extra slides..

Q: L01/intro.ipynb

ML supervised learning data flow model: ii) Testing (predict) + eval (score)



Øvelse:

- træn en lineær regression model,
  (Scikit-learn fit-predict interface),
- gå i detaljen med R^2 score funktionen,
  (NOTE: test data er lig train data for denne øvelse),
- check k-Nearest Neighbors modellen ud på data,
  sammenlign kNN-score med lineær regression-score.
- prøv en neuralt netværks-model på data
  (NOTE: den performer ekstrem dårligt!).

Q: `L01/intro.ipynb`

Opstart med Python, Scikit-learn og lidt matematik...

- Jupyter notebook: `intro.ipynb` [GITMAL].
- Scikit-learn `fit-predict` interface
- ML Algoritmer:
  - mange forskellige ML algoritmer, vi går pt. ikke i detaljen,
  - for denne opgave:

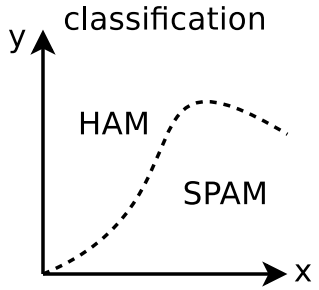
$$h(\mathbf{X}, \boldsymbol{\theta}) = \begin{cases} \text{Linear Regression.} \\ \text{k-Nearest Neighbors.} \\ \text{Neural-network (virker dårligt til data!).} \end{cases}$$

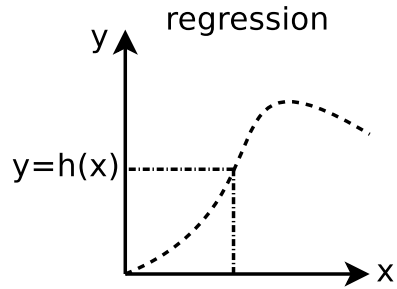
  - fokuserer på det overordnede ML flow.
- Loss og Scores funktioner
  - Loss: funktion, som ML algoritmen forsøger at minimere under `fit`.
  - Score: funktion, der fortæller noget om hvor godt et `predict` er, her afprøver vi R^2 (Coefficient of determination).

Et ML end-to-end projekt (SKIP slide!)

Emner fra [HOML] §2 'End-to-end Machine Learning'

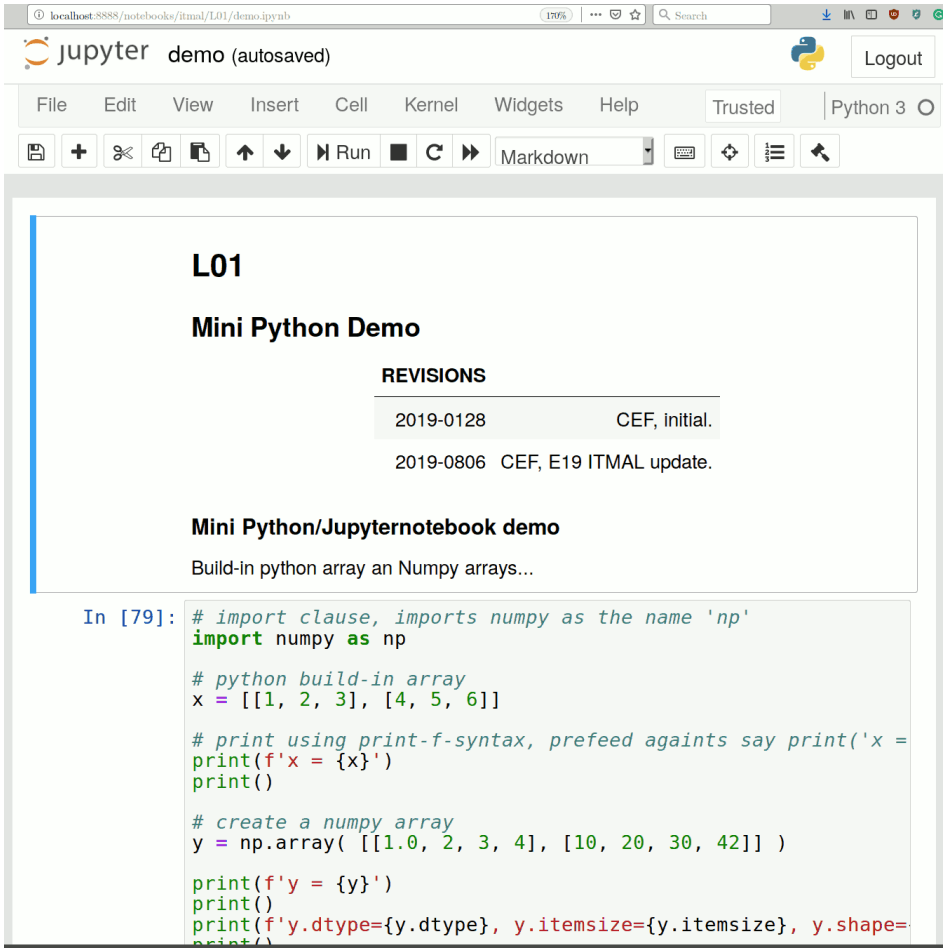
- Læringstyper:
  - supervised (mest om dette i SWMAL),
  - unsupervised, [semisupervised], [reinforced learning].
- Output klasser:
  - classification (ham/spam),
  - regression ($h(x) = y$),
- Læring via data:
  - batch læring (al data),
  - [inkrementel læring (on-the-fly)].
- Prediktions/generaliserings model:
  - model-based (pattern-detection, byg intern model),
  - [instance-based (lær al data udenad)],
- Typiske ML fejl klasser:
  - for lidt trænings data (small-data, brug cross-validation),
  - sampling noise, sampling bias (ved manglende stratificering),
  - outliers og dårlig data (i big-data),
  - model og algoritme fejl: underfitting/overfitting.

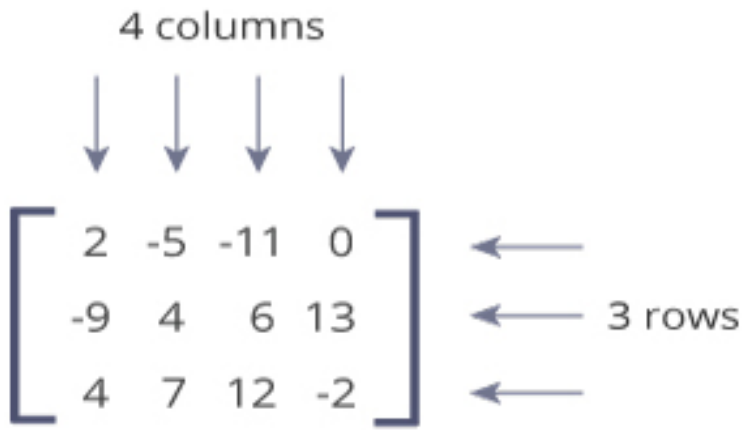




Fra MMLS (1.semester) til SWMAL

Matrix notation genopfriskning og intro til Design Matrix'en...

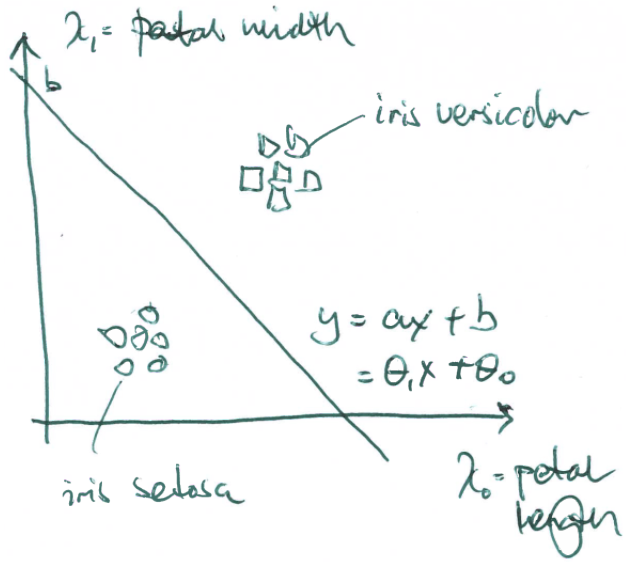




Q: `L01/intro.ipynb`

ML supervised learning data flow model: Iris classification example

Data, features λ, decision boundary:

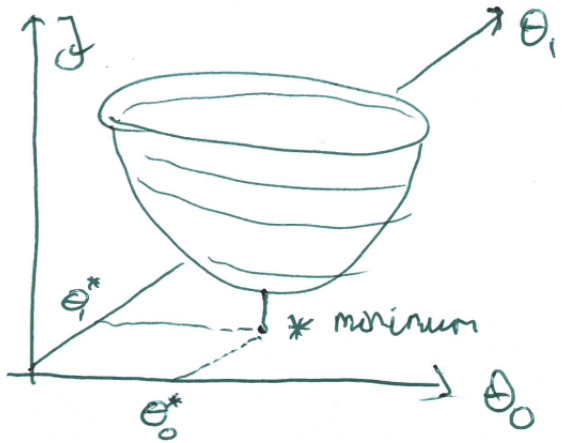


$\bar{x}^{(1)} = \begin{bmatrix} \lambda_0 \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} 1{,}2\,cm \\ 0{,}5\,cm \end{bmatrix}$ $\quad y^{(1)}_{true} = \text{'iris setosa'}$

$\bar{x}^{(2)} = \begin{bmatrix} 4{,}5\,cm \\ 1{,}2\,cm \end{bmatrix}$ $\quad y_{true} = \text{'iris versicolor'}$

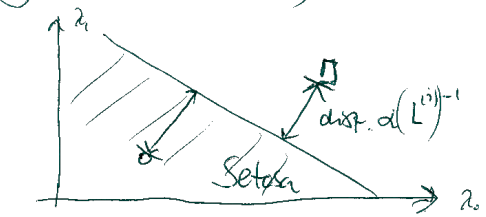
$\bar{\bar{X}} = \begin{bmatrix} \bar{x}^{(1)T} \\ \bar{x}^{(2)T} \end{bmatrix}$ $\quad \bar{y}_{true} = \begin{bmatrix} y^{(1)}_{true} \\ y^{(2)}_{true} \end{bmatrix}$ $\quad \bar{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$

Fit via cost function, J:



$$J = \sum_i L^{(i)} = L^{(1)}\left(h(\bar{x}^{(1)};\theta), y^{(1)}_{true}\right) + L^{(2)}\left(h(\bar{x}^{(2)};\theta), y^{(2)}_{true}\right) = L^{(1)}\left(y^{(1)}_{pred}, y^{(1)}_{true}\right) + L^{(1)}\left(y^{(2)}_{pred}, y^{(2)}_{true}\right)$$

$y^{(i)}_{pred} = \hat{y} = h(\bar{x}^{(i)};\theta)$

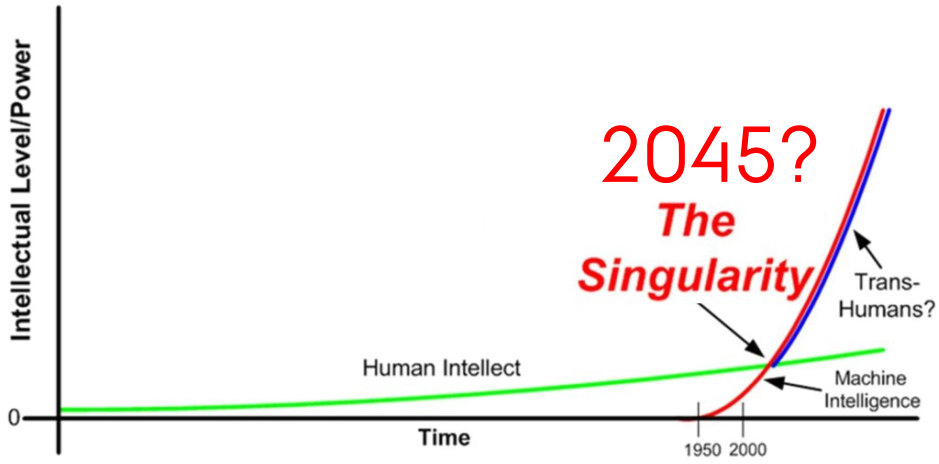


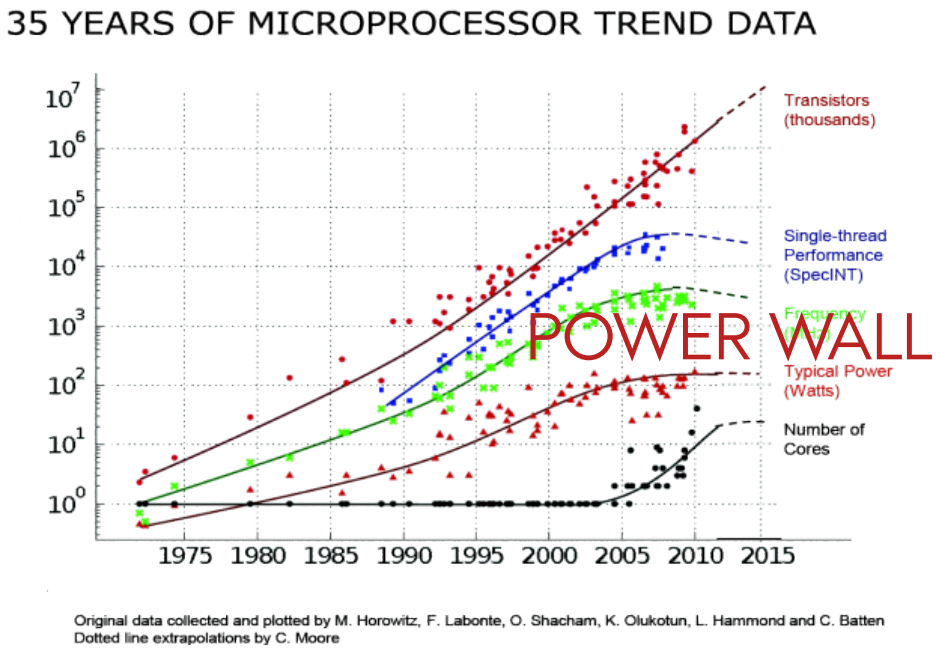
Moderne maskinlæring

Forskel på Sci-Fi og Science

- Har seks årtiers forskning i AI led til ingenting?
- Forstår vi hjernen, cognition, menneskelig intelligens?
- Vil eksponentiel udvikling give os ægte AI?

  (Ray Kurzweil, Singularity)
- Hvor er alle android, cyborg eller HAL-9000'erne i vores hverdag?





Sci: Dette kursus handler om Hands-on ML+python—ikke om Sci-Fi.

Moderne maskinlæring (SKIP slide!)

Eksponentiel udvikling og Moore's Lov

Software and computing

CPU scaling comes to the end of an era