



LUND  
UNIVERSITY

# EDAN96- Applied Machine Learning Lecture 1

---

MAJ STENMARK, 2023-10-30

1



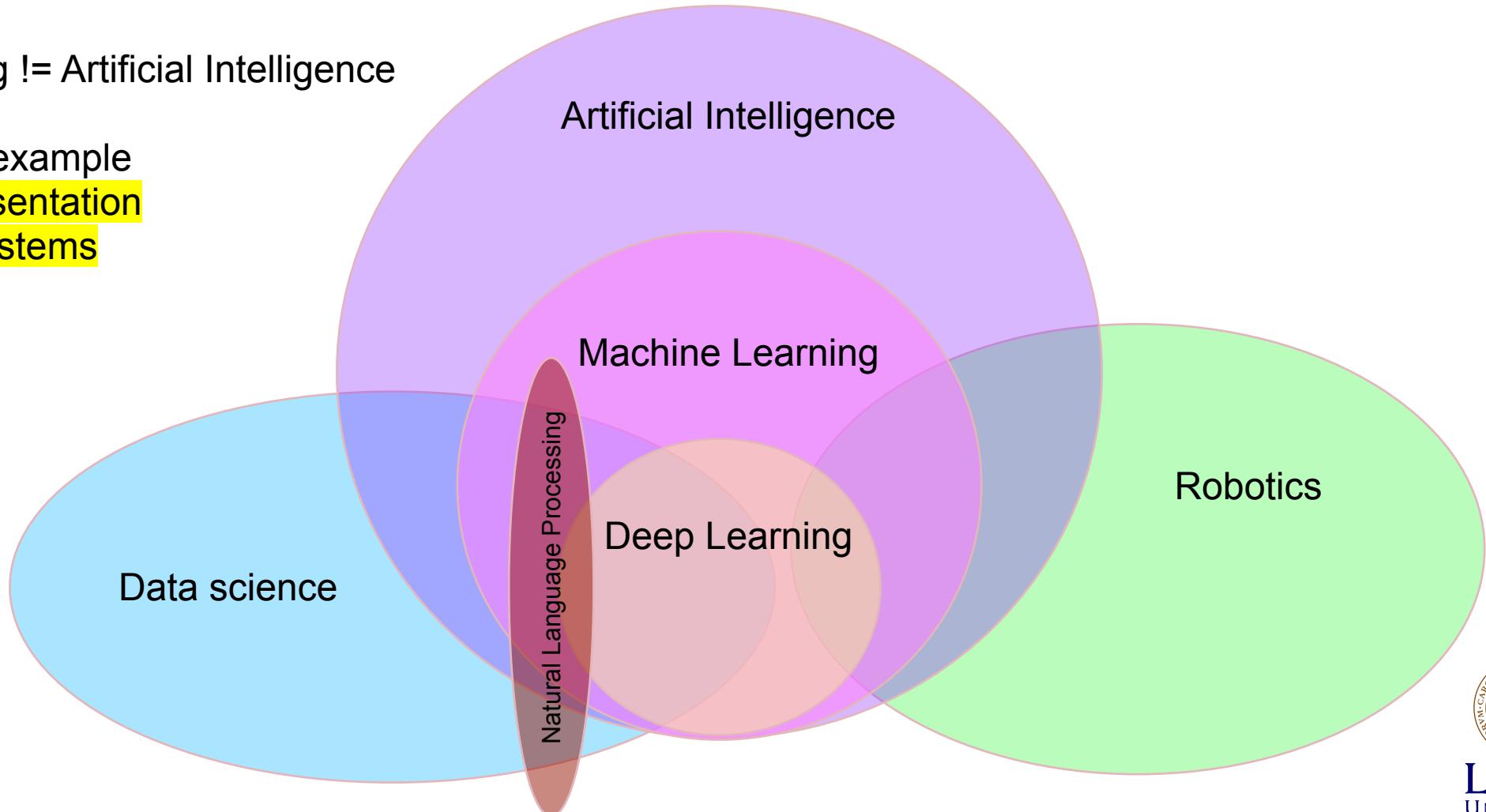
# Lecture 1

- Course info
- High-level intro to Machine Learning (ML).
  - Supervised/unsupervised learning
  - Problem classes
  - General workflow
  - Some evaluation metrics
- Programming assignment 1 (Lab 1)

# Machine Learning

Machine Learning != Artificial Intelligence

Non-ML AI is for example  
knowledge representation  
and reasoning systems



# LTH Course catalogue

EDAN20 Language Technology

EDAP01 Applied Artificial Intelligence

EDAN96 Applied Machine Learning

FMAN45 Machine Learning

EXTQ40 Introduction to Artificial Neural Networks and Deep Learning

STAN45 Data mining and visualization

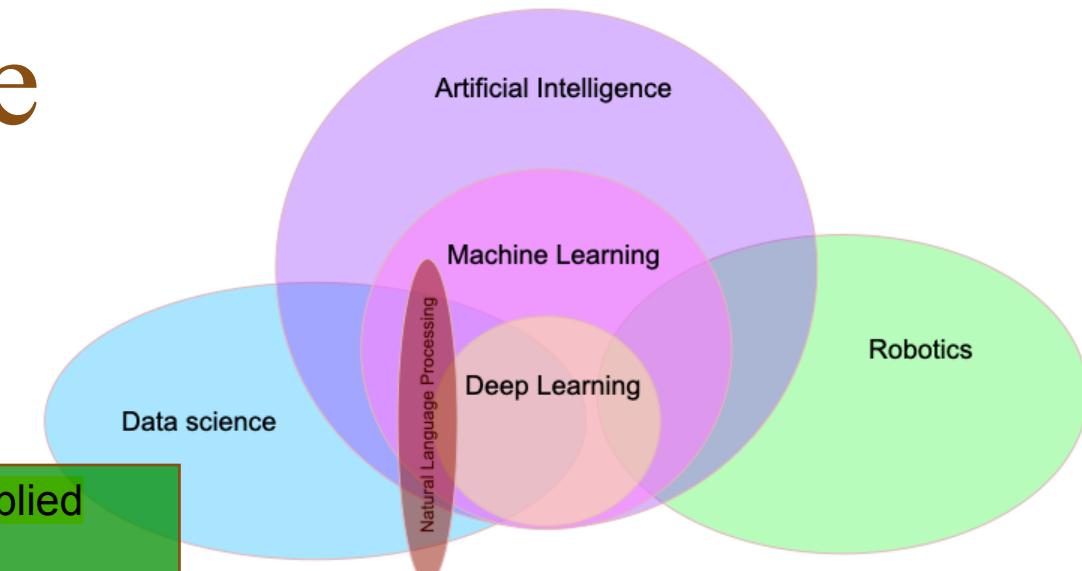


EDAP30 Advanced Applied Machine Learning

Overlaps!

FRTF20 Applied Robotics

EDAP20 Intelligent Autonomous Systems



Master's Thesis

# Applied Machine Learning Course content

## Aim

To give an *introduction* to fundamental methods and algorithms within Machine Learning

Specific Neural Networks are discussed in other courses: such as  
Transformers - Language Technology,  
Convolutional Neural Networks -  
Advanced Applied ML



List the topics in the course applied machine learning



Certainly, here are the topics covered in the course "Applied Machine Learning":

### 1. Fundamentals of Machine Learning:

- Unsupervised Learning
- Supervised Learning
- Classification
- Regression
- Probability Distributions
- Gradient Descent
- Model Selection and Cross-Validation
- Overfitting
- Generalized Linear Models
- Regression
- Kernel Methods
- Information Theory

### 2. Specific Machine Learning Topics:

- Principal Component Analysis
- Support Vector Machines
- Decision Trees
- Random Forests
- Ensemble Methods

### 3. Application-Related Topics (discussed at an overview level):

- Specific Neural Networks, including Convolutional Neural Networks and Recurrent Neural Networks
- Autoencoders
- Bayesian Classifiers

# Note: The cool stuff is advanced

- Alpha Go
- Github Copilot
- Large language models LLMs, e.g., ChatGPT
- Other generative AI models (such as text to image generation).
- Generative Adversarial Networks (GANs) for fake videos.

Prompt: “image for applied machine learning course”  
Style: steampunk  
Using [gencraft.com](https://gencraft.com)



Advanced architectures (designed for the application), large amounts of data, long process of training/tuning/evaluation and large servers.

# Course Content cont.

The course aims to give you an overview of, and explain different ML tools to enable you to

- decide whether the problem or task you face can be tackled using ML
- apply the right ML tool(s) for the right problem(s) in the correct way
- handle data in a suitable way
- reflect upon the consequences of applying these tools
- control the amount of “black magic” in your work

Prompt: “a hand held hammer hitting a nail in a wall”

Style: Cartoon

Using: [gencraft.com](http://gencraft.com)



# Course Syllabus

- 14 lectures, Mondays 3-5pm and Wednesdays 1-3pm. The lectures are given by 4 different teachers.
- 2 sets of tutorial material (Python / Numpy and Linear Algebra / Matrix operations)
- 7 lab sessions with 2 instances each, Fridays, 8-10 and 3-5pm
- **5 programming assignments** to be worked on in **pairs** and presented in the corresponding lab sessions (assignments 4 and 5 run over two sessions each),
- **3 homework (examination) assignments** (connected to three of the programming assignments) that require handing in an **individually written report** for a pass!
- 7.5 hp / ECTS credits for **passing all assignments** (grade 3 (pass) on 3,4,5/U scale)
- **1 optional written exam** on January 12, to get (higher) grade on the 3,4,5/U scale.

# Compulsory items

- *Lab sessions* are weakly compulsory: you need to present the solution to the assignments within the lab session instances you signed up for, if not another agreement has been made explicitly. You should try to follow the schedule as closely as possible, but “retake presentations” in later sessions are acceptable within manageable limits - priority is given to students following the schedule.
- Assignments 1,4, and 5 have (roughly) corresponding reports that are the main basis for examination (but you need to pass ALL 5 assignments). Failing to attend without notification or failing to hand in the respective report by the given deadline (approximately 10 days after you should pass the practical part) might result in not being able to pass the course in this instance.

# Cheating with AI?

- Unless specifically asked, do not use AI to solve your assignments. Specifically, do not use large language models such as ChatGPT to write your homework assignments.
- OK to use spell - and grammar checkers.

Prompt: “a student that is cheating with the help of AI”  
Style: Cartoon  
Using: [gencraft.com](https://gencraft.com)



# Course Canvas

Canvas and Discord.

Discord for discussions - help each other!

Discord is also used to queue for the labs (more about that later!)

The screenshot shows the Course Canvas interface for the module EDAN96. The left sidebar includes links for Home, Syllabus, Announcements, Assignments, Quizzes, Modules, Discussions, People, Grades, Pages, Files, Outcomes, BigBlueButton, Collaborations, Rubrics, and Settings. The main content area is titled 'Recent announcements' and lists various items with checkboxes and ellipsis buttons. The items include:

- General information and material, introduction and summary (Maj Stenmark)
- Discord server
- Literature list
- Contact information
- Lecture 1: Introduction
- Lecture 1.5: Linear algebra recap
- "Lab 0": Python intro
- Python\_introduction.ipynb
- Lab instructions quiz  
2 Nov | 5 pts
- Programming assignment 1, Data inspection and evaluation tools  
3 Nov | 10 pts
- Homework assignment 1 (Evaluation tools)  
16 Nov | 7 pts
- MSc projects / research areas with potential projects in the RSS group
- Previous exams

Buttons at the top right include 'Collapse all', 'View progress', 'Publish All', '+ Module', and a three-dot menu.

# This week

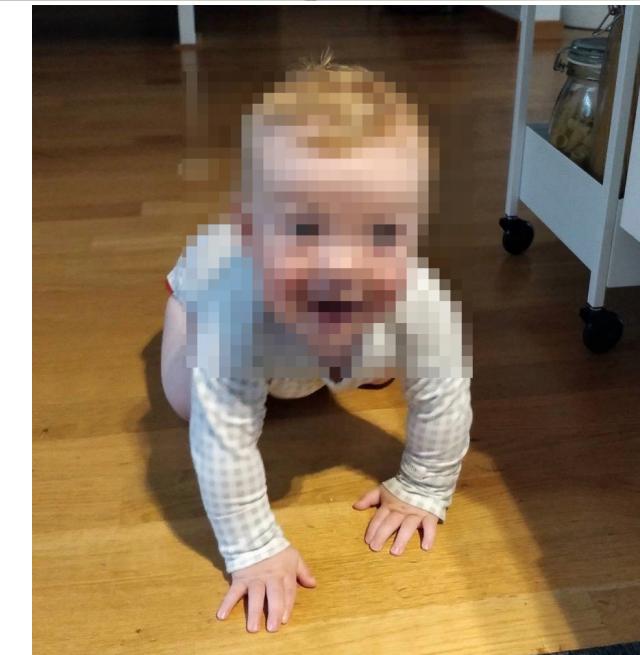
- Sign up for a lab session by WEDNESDAY, Nov 1st, 3:00 pm.

<https://sam.cs.lth.se/LabsSelectSession?occasionId=847>

- Do the “[Lab instructions quiz](#)” in Canvas prior to coming to the first lab (by Nov 2nd).
- Find a lab partner in Discord.
- First lab on Friday! The programming assignments takes quite some time, do most of the work ahead of the lab session.
- “**Resurs**” time in the schedule: online office hours. Questions will be answered by TAs in Discord *during the resurs times*.
- Lecture on Wednesday: Intro to Linear Algebra, Python, Numpy.
- Installation and intro to Python on Canvas, Lab 0.

# Who am I?

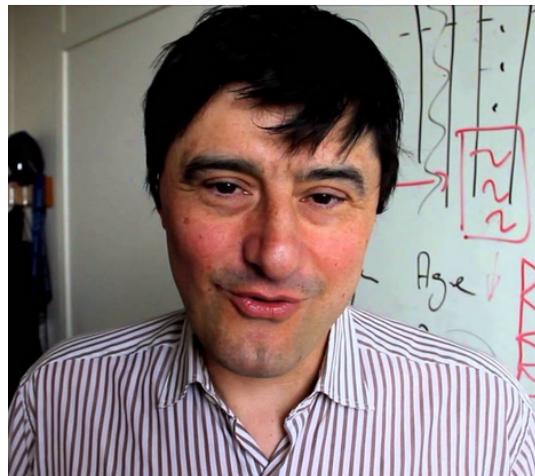
- Maj Stenmark, PhD
- Associate Senior Lecturer in Computer Science
- maj.stenmark@cs.lth.se
- Robotics and Semantic Systems group
- Research topic: Human-robot interaction
- Works Mondays and Wednesdays, on part-time parental leave.



LUND  
UNIVERSITY

# Who are we?

- 4 lecturers from the Robotics and semantic systems group Pierre Nugues, Elin A. Topp and Leonard Papenmeier.
- 14 TAs (students from last year, PhD students and post-doctoral researchers)



Pierre



Elin



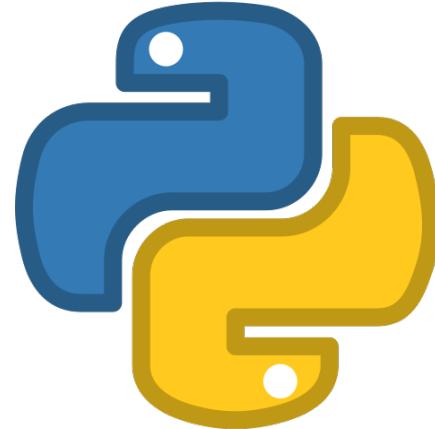
Leonard



LUND  
UNIVERSITY

# Who are you?

- 150 students from various programs.
- International Master's and exchange students.
- Second Java course as requirement or equivalent.
- No Python requirement.
- 120 credits within program, implicitly assuming some maths.



$$2 + 2 = ???$$

# Machine Learning - learns from data

ML is an area of (AI) research providing *powerful tools* that enable machines (computers) to find models describing *data* and the correlations between them, to

- *predict future outcomes or developments* given previous data (stock market, complex physical processes)
- *classify unknown input given known, classified data* (scene contains pedestrian or not)
- *identify structures* in unseen, unlabeled data (grouping people according to different attributes)
- *decide upon next steps or actions to take to maximise reward* (new measurement, robot action)
- *generate content*, such as text and images.

# Success stories Example 1

Silver et al, Nature, Oct 18, 2017:

## **“Mastering the game of Go without human knowledge”**

A long-standing goal of artificial intelligence is an algorithm that learns, *tabula rasa*, superhuman proficiency in challenging domains. Recently, AlphaGo became the first program to defeat a world champion in the game of Go. The tree search in AlphaGo evaluated positions and selected moves using deep neural networks. These neural networks were trained by supervised learning from human expert moves, and by reinforcement learning from self-play. Here we introduce an algorithm based solely on reinforcement learning, without human data, guidance or domain knowledge beyond game rules. AlphaGo becomes its own teacher: a neural network is trained to predict AlphaGo’s own move selections and also the winner of AlphaGo’s games. This neural network improves the strength of the tree search, resulting in higher quality move selection and stronger self-play in the next iteration. Starting *tabula rasa*, our new program AlphaGo Zero achieved superhuman performance, winning 100–0 against the previously published, champion-defeating AlphaGo.

...

<https://www.nature.com/articles/nature24270>

# Success stories Example 2

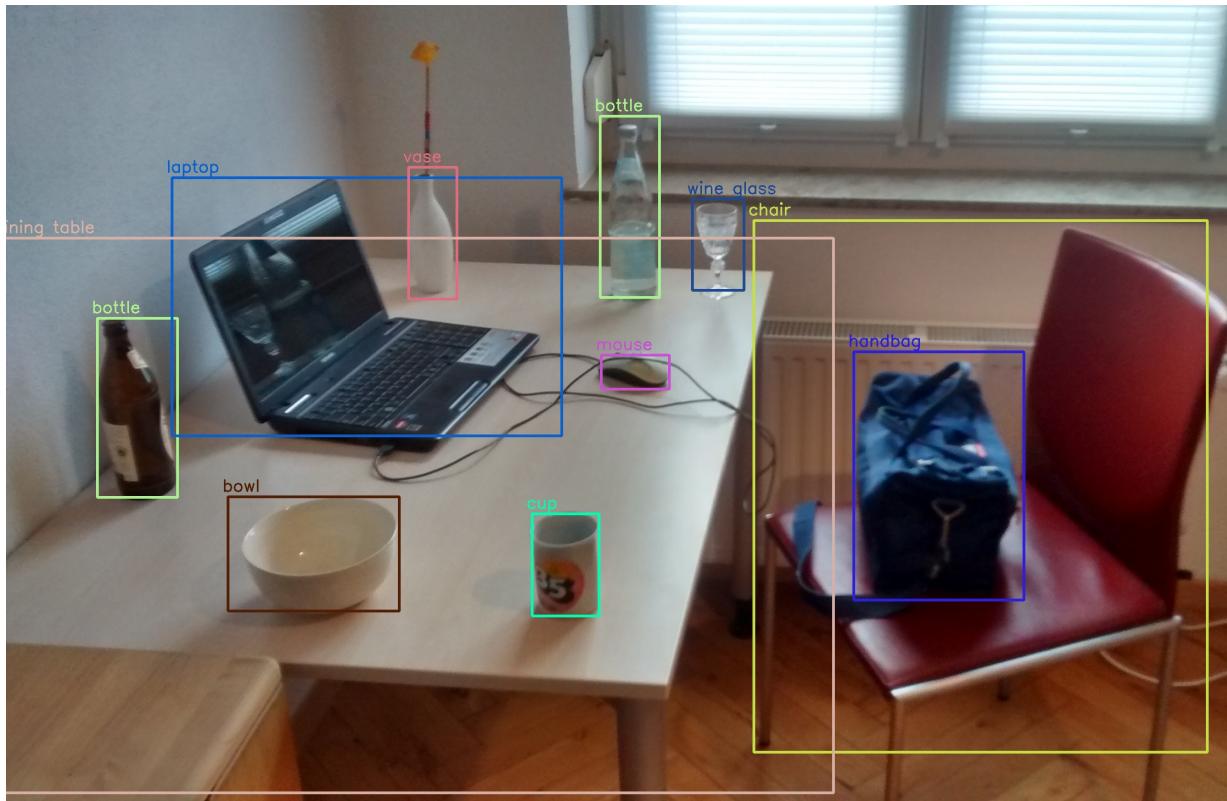


Image of YOLOv3 from Wikipedia by [MTheiler](#)

Object detection algorithms have improved significantly in recent years, e.g.,

You Only Look Once, YOLO  
(2015)

Single Shot MultiBox Detector, SSD (2015)



LUND  
UNIVERSITY

# Success stories Example 3



What does the GPT stand for in chatGPT



GPT stands for "Generative Pre-trained Transformer." It refers to the architecture and methodology used in training models like ChatGPT. The "Generative" part indicates that the model can generate text, the "Pre-trained" aspect means that it is trained on a large dataset before fine-tuning for specific tasks, and "Transformer" is the type of neural network architecture it's based on.



# Aaaaand fails

Nguyen, Yosinski, Clune, CVPR 2015 / AAAI Video Competition 2016:

**“Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images”**

...

Here we show a related result: it is easy to produce images that are completely unrecognizable to humans, but that state-of-the-art DNNs believe to be recognizable objects with 99.99% confidence (e.g. labeling with certainty that white noise static is a lion). Specifically, we take convolutional neural networks trained to perform well on either the ImageNet or MNIST datasets and then find images with evolutionary algorithms or gradient ascent that DNNs label with high confidence as belonging to each dataset class. It is possible to produce images totally unrecognizable to human eyes that DNNs believe with near certainty are familiar objects, which we call “fooling images” (more generally, fooling examples). Our results shed light on interesting differences between human vision and current DNNs, and raise questions about the generality of DNN computer vision.



# Confidently wrong



What new discoveries from the James Space Webb Telescope can I tell my 9 year old about?



Your 9-year old might like these recent discoveries made by The James Webb Space Telescope (JWST):

- In 2023, The JWST discovered a number of galaxies nicknamed "green peas." They were given this name because they are small and round, like peas.
- The telescope captured images of galaxies that are over 13 billion years old. This means that the light from these galaxies has been traveling for over 13 billion years to reach us.
- JWST took the very first pictures of a planet outside of our own solar system. These distant worlds are called "exoplanets". Exo means "from outside".

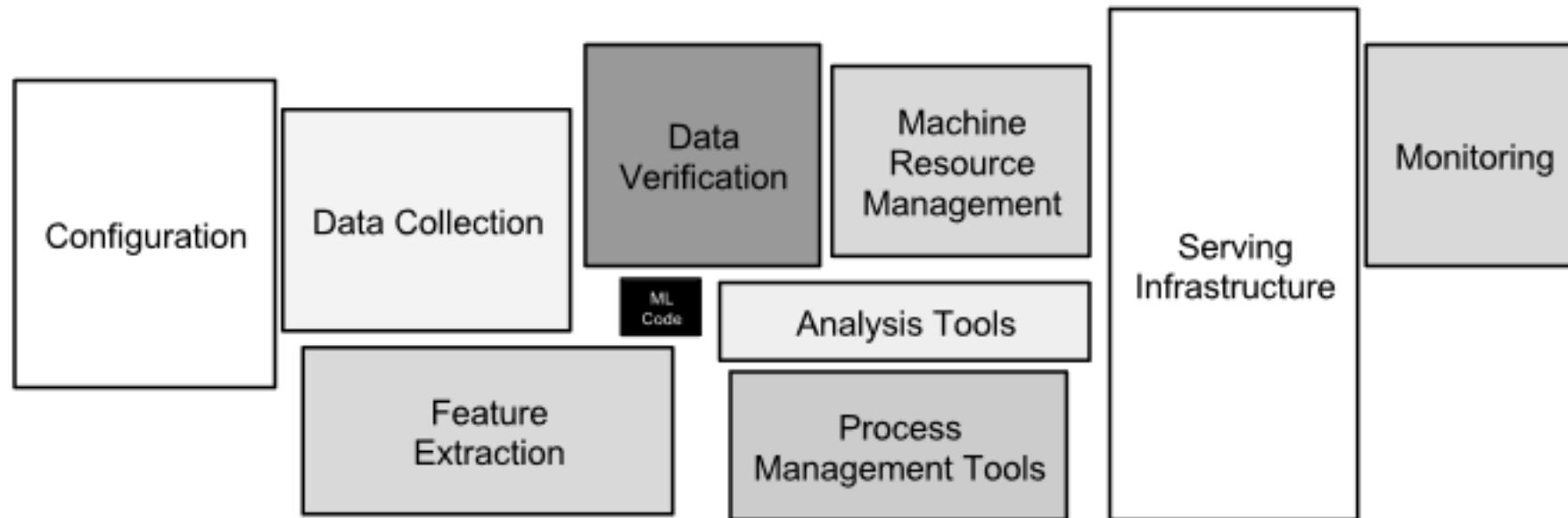
These discoveries can spark a child's imagination about the infinite wonders of the universe



Check it

⋮

# Technical depth of real world systems



# ML as a smörgåsbord

- Groups of algorithms / data (supervised, unsupervised, (semi)-supervised)
- Problem classes (finding structure, clustering, prediction, classification, action learning)
- Models (neural networks, statistical, probabilistic)

Categorising input data using a labeled dataset

# Supervised Learning

The model will take some input X and produce some output Y.

In the simplest setting X is a D-dimensional array (tensor) of numbers representing different features (attributes or variables). These can be nominal (numbers), or categorical.

If the output Y (response variable) is categorical we call it the class/target/label and if it is real valued it is regression.

**Supervised learning:** the training data consists of *labelled training examples* ( $x_i, y_i$ ) and the mapping from X to Y is learned.

Supervised learning is a type of machine learning in which an algorithm learns to map input data to a target output or label by being presented with a dataset containing input-output pairs. In supervised learning, the algorithm learns to make predictions or decisions based on input data, and it does so under the guidance or supervision of a labeled dataset

Regression is a statistical analysis used to model the relationship between dependent variables and independent variables.

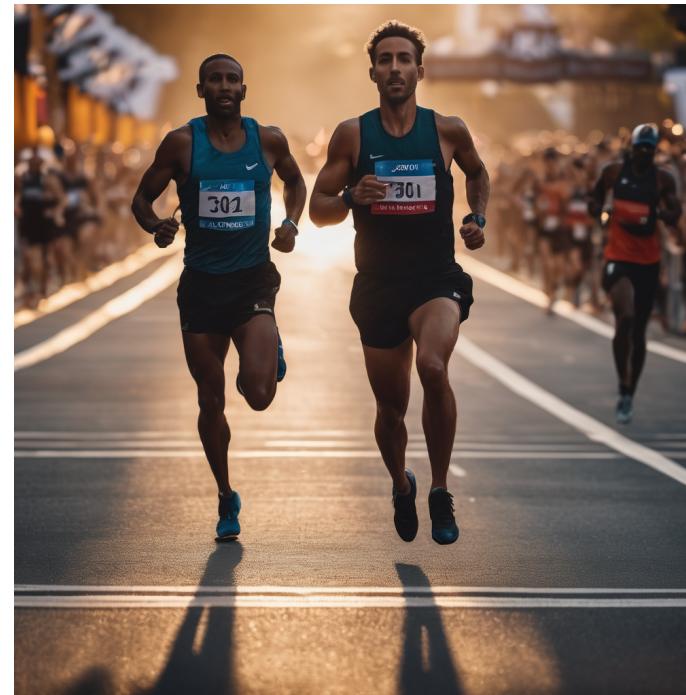
The goal is to quantify how changes in independent variables are associated with changes in the dependent one

# Supervised Learning: simple example

Simple example of input and output:

Given runners' 5k, 10k, and half marathon times, predict their marathon times.

Easier to just  
fit a curve ...



Prompt: "Marathon runners at the finish line"

Style: Realistic

Using: [gencraft.com](https://gencraft.com)



LUND  
UNIVERSITY

# Unsupervised Learning

- No labelled data.
- Discover pattern in the data, e.g., clusters.
- Example: in e-commerce cluster users into groups based on purchases and send targeted advertising to each group.
- Example method: *k-means clustering*.

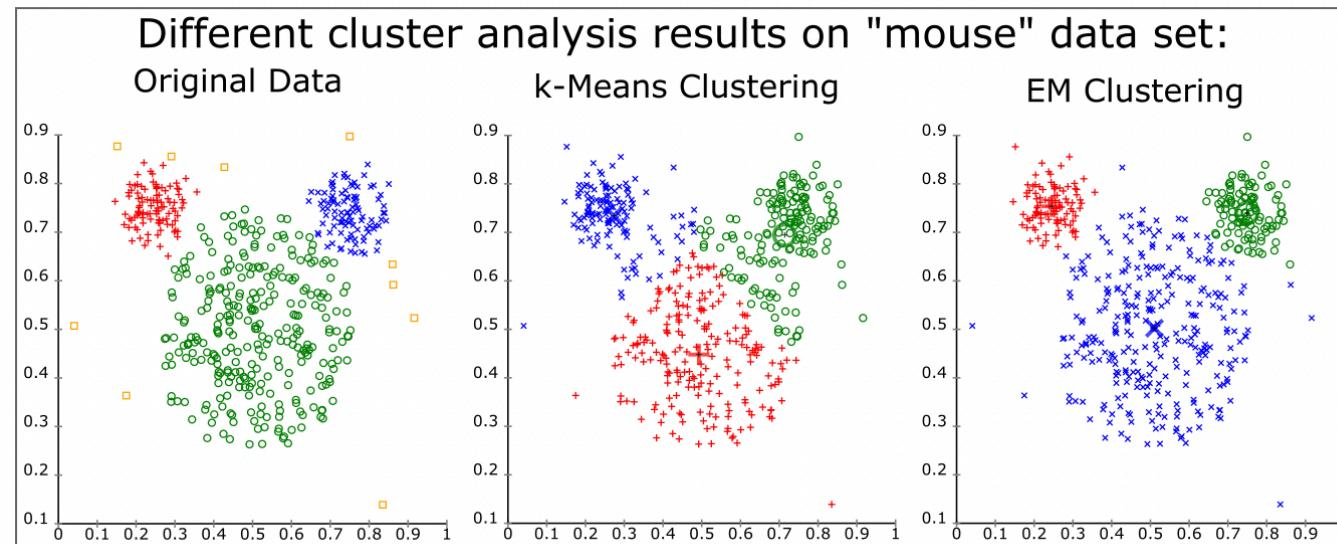
K-means clustering is an algorithm used to cluster similar data points into groups. The objective of K-means is to partition a dataset into K distinct (needs to specify K number of clusters), non-overlapping clusters, where each data point belongs to the cluster with the nearest mean (centroid).

A centroid is the central point within a cluster, i.e. the mean of the cluster and is essential for identifying natural groupings in data and making predictions based on central tendencies of these groups. They provide a way to summarise and represent clusters of datapoints.

Discovering patterns in data and label them

EM-clustering, Expectation-Maximization, is a probabilistic model-based clustering algorithm also used to partition data into clusters. It is used to estimate parameters in statistical models when you have missing or incomplete data.

Used when number of clusters is not known in advance. Thus, it requires a probabilistic model for data distribution such as GMM, and the performance depends on the initial parameterization. It is a soft clustering algorithm, meaning that data points can have partial membership in multiple clusters.



Neural networks is a network consisting of interconnected artificial neurons (nodes) organised in different layers. Input layer (feature/variable), Hidden layer (can be multiple, each node performs a weighted sum of its inputs and passes it to an activation function), Output layer produces the final result of the computations, consists of k neurons depending on the task, each node corresponds to a class.

# Problem classes: prediction

Weights and biases: adjust the strength of connections between neurons. The weights are multiplied by the inputs, and the biases are added before passing the result through an activation function. Learning in neural networks involves adjusting these weights and biases during training to minimize prediction errors.

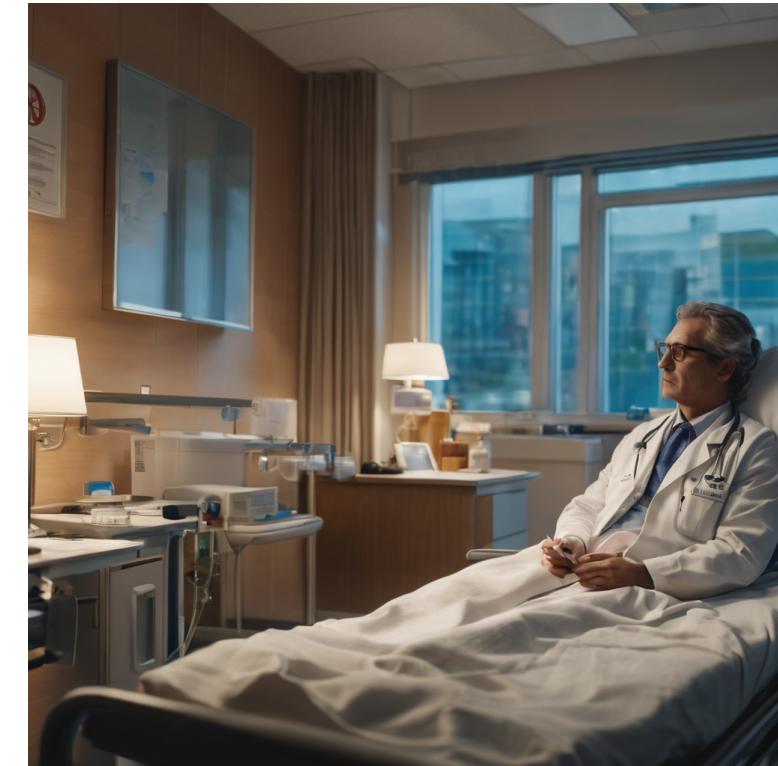
Likelihood of patient developing disease based on their medical history

Regression

Stock market development for automated trading.

Neural networks (LSTM)

Activation functions introduce non-linearity to the model. Ex. Sigmoid function, tanh, rectified linear unit ReLU, determine the output of a neuron based on its weighted sum of inputs



Prompt: "A patient in a hospital bed smoking a cigarette.  
Beside the bed a doctor with a chart."

# Problem classes: classification

Logistic regression is a statistical and machine learning model used for binary classification. It's a type of regression analysis that is well-suited for predicting the probability of a binary outcome (1/0, Yes/No, True/False) based on one or more predictor variables. While its name contains "regression," it is used for classification

## Binary:

Decide from an image if a mushroom is edible or not.  
Is the email spam or not?



## Multi-class/multi-label:

What species is the plant?  
What language is the document?

Logistic regression  
Neural networks

Prompt: "Poisonous mushroom"  
<https://stablediffusionweb.com>



LUND  
UNIVERSITY

Reinforcement learning focuses on how agents (person who interacts) or decision makers learn to make sequences of decision in an environment (system or world it interacts with) in order to achieve a specific. The learning involves learning through interaction with an environment and receiving feedback in the form of rewards or penalties. Rewards is a numerical value that the agent receives from the environment after taking an action in a specific state. The reward provides feedback about the desirability of the action. The goal is to find an optimal policy that maximises cumulative rewards over time. This involves a trade-off between exploration (trying new actions to discover potentially better ones) and exploitation (choosing actions that are currently believed to be the best).

# Problem classes: decision making

Given someone's income and current dept, should they be approved of a loan?

Decision trees

Where should a robot grip an object to pick it up?

E.g., Reinforcement Learning

Decision trees are used for classification and regression tasks, it's an algorithm that mimics the structure of a tree to make decisions or predictions. Each internal decision node of the tree represents a decision based on a feature (attribute), each branch represents a possible outcome of a decision, and each leaf node represents a class label or a numeric value (for regression). It uses splitting criterias such as Gini impurity (for classification) or mean squared error (for regression) to determine the best feature and threshold for splitting data at each decision node. To avoid overfitting, branches that do not greatly improve the model's performance on validation data are removed



Photo: Maj Stenmark

# Problem classes: generate content

Large language models: uses current sentence and user input to step-by-step generate (predict) new words.

**Neural networks (transformers)**

Generate images with diffusion models (similar to models for reducing blur in images).



# The general workflow

1. Prepare data e.g. reduce dimensions, consolidate data
2. Mathematical model To interpret the data, choose ML model e.g., neural network, decision tree etc.
3. Run learning algorithm to learn parameters Run it
4. Evaluate the results Precision and recall, confusion matrix

# Data: issues and tools

Values can be **nominal** or **categorical** (give each category a number).

High-dimensional: too much data for the model to handle.

Ex: Marathon time: we also know the runners' weight, age, sex, salary, nationality, eye color...

Issues:

- Unbalanced dataset
- Missing values
- “Wrong” values

Wrong values due to overfit model, and becomes too complex and generalises poorly to new unseen data

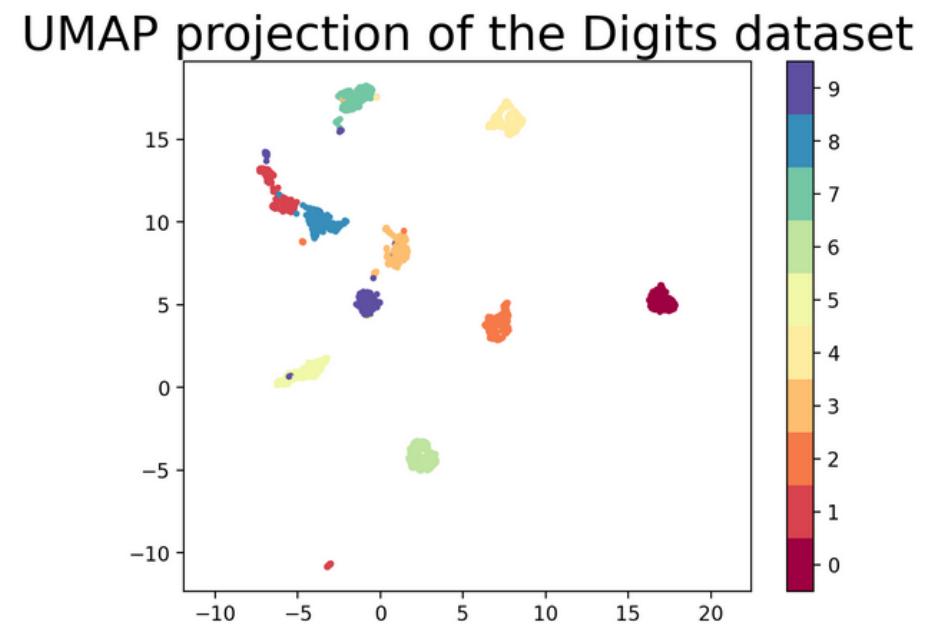
May not have enough data to cover all combinations of feature values -> can lead to missing values or incomplete representations of data

## The curse of dimensionality

as the number of features or dimensions in a dataset increases, the volume of the data space grows exponentially, and various issues related to data sparsity ((can be difficult to find meaningful patterns or relationships), computational complexity, and model performance can become more pronounced.

# Data: issues and tools

- Use **visualization** tools to understand data!  
matplotlib, sklearn built-in functions
- Tools to **reduce dimensionality**:
  - **PCA**
  - **UMAP** (Lab 1) *Uniform Manifold Approximation and Projection* for visualisation in 2D. **Keeps relative distances between samples.**



# Classic example

Given the weather data for a day, will my friend Aldo enjoy sport?

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport?
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes
unseen	Sunny	Warm	Normal	Strong	Cool	Change	?

Supervised!



<https://stablediffusionweb.com>

Prompt: "A windy day for playing badminton."

<https://stablediffusionweb.com/>

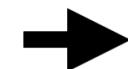
# Classic example *(cell)*

Each sample is a row in a matrix, each position in the row vector represents a property (attribute, feature).

Split your data into a training set and a test set, e.g., 80/20.

If you have target values (labels) use supervised method, otherwise unsupervised.

Sky = <Sunny = 0, Rainy = 1, Cloudy = 2>  
Air = <Warm = 0, Cold = 1>  
Humidity = <Normal = 0, High = 1>  
Wind = <Weak = 0, Strong = 1>  
Water temp = <Warm = 0, Cool = 1>  
Forecast = <Same = 0, Change = 1>  
  
Enjoy Sport = <No = 0, Yes = 1>



$$D = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

*the matrix*

$$d = (\text{Sunny } \text{ Warm } \text{ Normal } \text{ Strong } \text{ Cool } \text{ Change}) = (0 \ 0 \ 0 \ 1 \ 0 \ 1) \rightarrow ???$$

# Training loop supervised learning

We want to find a (mathematical) model for the arrow.

$$D = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

1. Choose models (neural network, decision tree, etc)
2. Initialize the model with initial estimates
3. Iteratively:
  - A. Calculate the error, using a *loss function*. E.g. mean squared error
$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$
  - B. Update estimates of the model.
4. Stop when loss is under desired threshold.
5. Evaluate the model on the test data.



# Evaluation tools

- Precision and recall
- F1-score is the harmonic mean

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

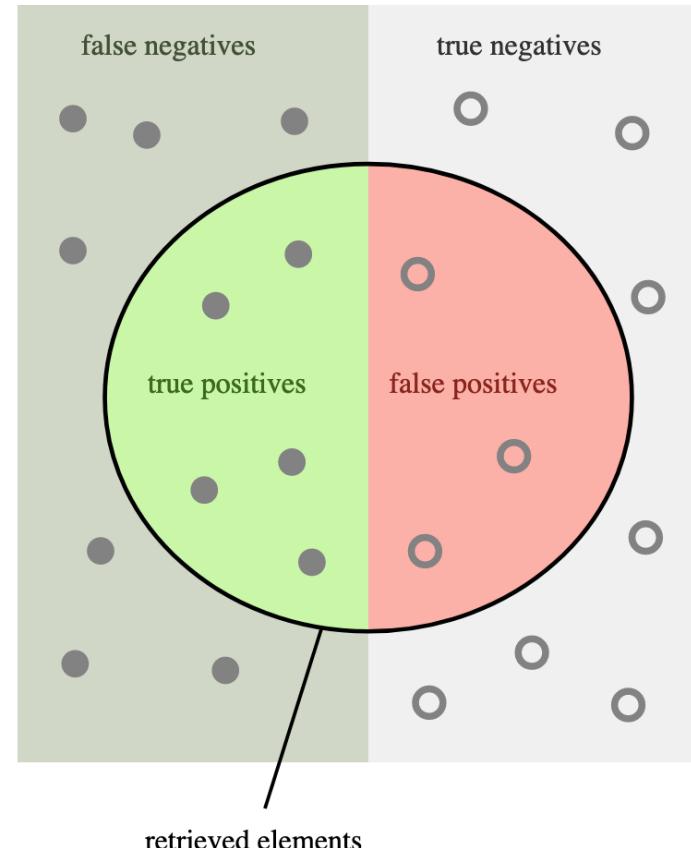
Precision and recall are used to assess the performance of a classification model, particularly in the context of binary classification problems e.g., spam or fraud detection, medical diagnosis.

Precision is a measure of the accuracy of the model's positive predictions = true positives. (Of all instances predicted as positive, how many of them were actually positive).

Recall is the sensitivity or true positive rate, measuring the model's ability to find all the positive instances in the dataset. Of all the actual positive instances, how many did the model correctly identify as positive?

Precision is useful when the cost of false positives is high, and you want to ensure that the positive predictions made by your model are highly accurate.

Recall is particularly important when the cost of false negatives (missed positive instances) is high, and you want to ensure that your model doesn't overlook positive cases.



Wikipedia: [https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)

How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$



# Evaluation tools

- Confusion matrix

Confusion matrix evaluates the performance of a classification model, especially in binary classification problems. It provides a breakdown of the model's prediction and the its accuracy

		Predicted condition	
Total population $= P + N$		Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Wikipedia: [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix)

True Positives (TP): These are cases where the model correctly predicted the positive class (e.g., correctly identified individuals with a disease).

True Negatives (TN): These are cases where the model correctly predicted the negative class (e.g., correctly identified individuals without the disease).

False Positives (FP): These are cases where the model incorrectly predicted the positive class (e.g., falsely identified individuals without the disease as having it).

False Negatives (FN): These are cases where the model incorrectly predicted the negative class (e.g., falsely identified individuals with the disease as not having it).



# Training loop unsupervised learning

1. Try to group the data into clusters.

2. Iteratively:

A. Calculate a *loss function*. E.g. measure the Euclidian distance of how much the center points of the clusters moved.

B. Continue until convergence.

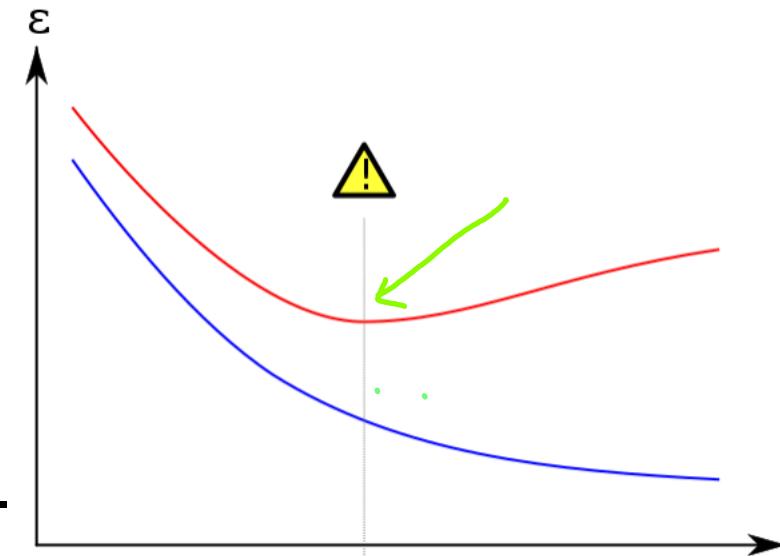
3. Methods to evaluate how well the clusters describe the data:  
*homogeneity, completeness, mutual information.*

4. (Evaluate the model on the test data.)

$$D = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} (1) \\ (1) \end{pmatrix}$$

# Pitfall: overfitting

- Your model learns the training data perfectly. Fails on unseen data.
- Training and test (validation) accuracy diverge at some point.
- What can you do?
  - Stop training when the two measures start diverging.
  - Reduce dimensionality of your model: the more complex, the more training data is needed.

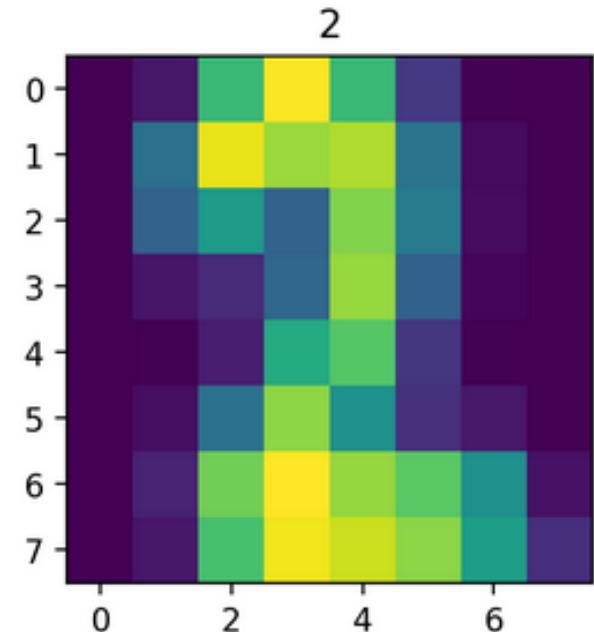


Wikipedia: Overfitting.svg by Gringer

# Programming assignment 1 (Lab 1)

**Available in Canvas, due in the first lab session**

- Python & SciKitLearn
- Some data inspection and manipulation
- Two methods: k-Nearest Neighbours classifier vs k-Means clustering on `sklearn.datasets.digits` (a VERY much simplified version of the MNIST handwritten numbers dataset)
- Evaluation tools
- Work in pairs!
- Discussion with TA required
- Running code is NOT ENOUGH, there are questions you NEED TO ANSWER!
- Connected to homework assignment 1, short report.

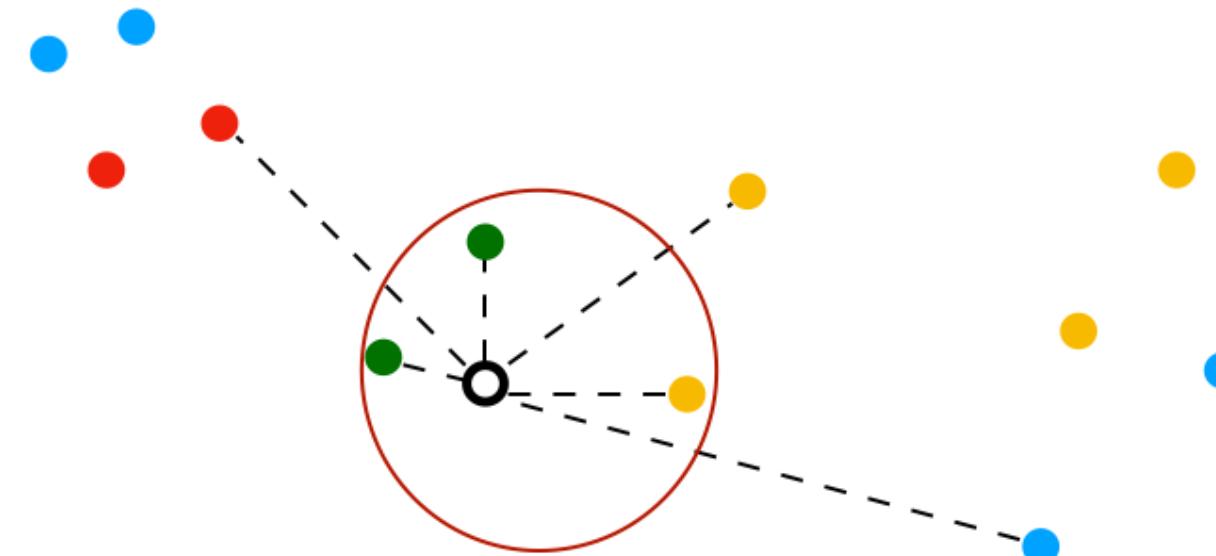


# k-Nearest Neighbours (kNN)

1. Input: points in space. Each point has a label.
2. Save the entire dataset as reference.
3. For each point  $p$  to classify: find the  $k$  number of closest points using some distance measure (e.g., Euclidian distance).
4. Give  $p$  the most common label of the  $k$  neighbours.

Break ties:

k odd or  
increase k by 1

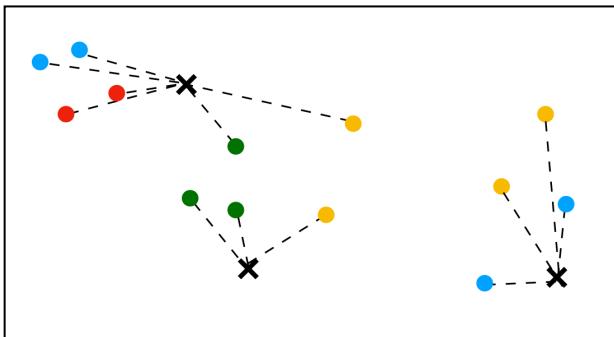


Unsupervised learning algorithm used to cluster data into group based on similarity.

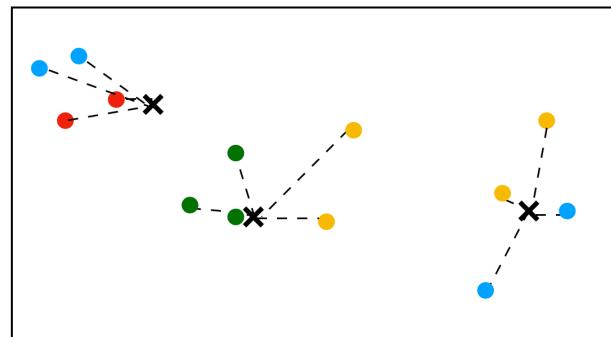
# k-means

1. Input: points in space.
2. Decide how many clusters k you want.
3. Place k centroids in the space (randomly)
4. Iteratively:
  1. Assign each point to the closest centroid. => move to closest centroid/cluster
  2. Recalculate the centroids as the average of the points in the cluster. => moves the centroids
  3. Repeat until the clusters do not change anymore. => Until convergence

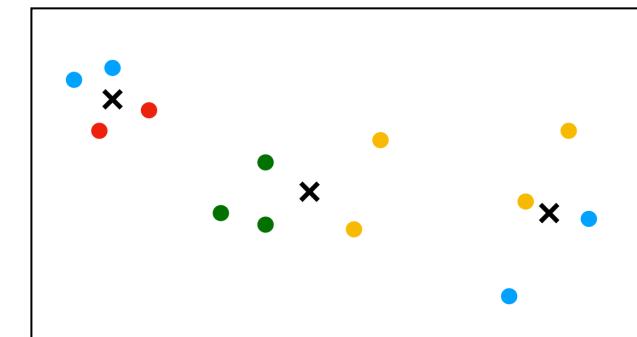
Random centroids. Assign points to closest



Move centroids. Assign points to closest again.



Move centroids. Stop when clusters do not change.



# Next lecture

- Linear Algebra, Python, Numpy
- ~~Pierre Nugues~~
- Extra material (recommended) on Canvas:
  - Lab 0: Python / numpy tutorial
  - Video: Linear Algebra / matrix operations pre-recorded lecture(s)
- Reading:
  - Lindholm et al, chapter 1,
  - Goodfellow chapters 2-3,
  - online material by Géron on probabilities and LA / matrix operations

# Summary

- Course info
- High-level introduction to machine learning.
- Tasks for the week:
  - Sign up for lab sessions by Wednesday, 3 pm.
  - Pass the Lab Instructions Quiz.
  - Get started with the Python tools
  - Work on Lab 1, present it Friday.