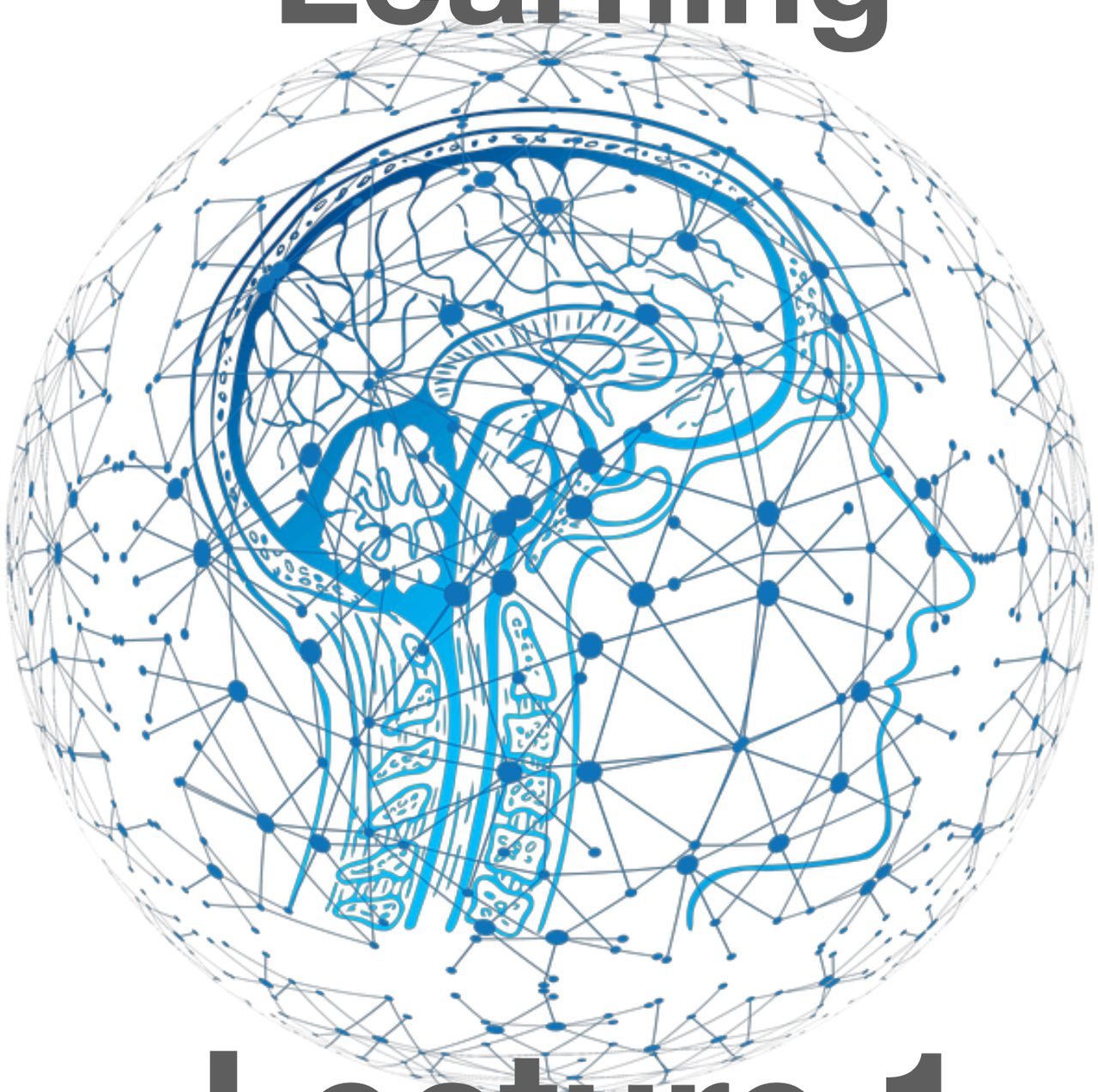
Course 395: Introduction to Machine Learning



Lecture 1

Dr Antoine Cully

Course 395:
Introduction to Machine Learning

- **Lecturers**:
  - Antoine Cully (a.cully@imperial.ac.uk)
  - Nuri Cingillioglu (nuri.cingillioglu13@imperial.ac.uk)
- **Goal** (Lectures): To present basic theoretical concepts and key algorithms that form the core of machine learning
- **Goal** (CBC): To enable hands-on experience with implementing machine learning algorithms (developed using Python)
- **Material**: Artificial Intelligence: a Modern Approach, Russel and Norvig (2009)
- **Acknowledgement**: This course is based on the lecture material from Dr Maja Pantic and Dr Aldo Faisal.

Course 395: Lectures

Lab sessions every week, starting next week.

- Week 2 (today!): Intro and Instance Based Learning (*A. Cully*)
- Week 3: Decision Trees & CBC Intro (*A. Cully*)
- Week 4: Evaluating Hypotheses (*A. Cully*)
- Week 5: Artificial Neural Networks I (*N. Cingillioglu*)
- Week 6: Artificial Neural Networks II (*N. Cingillioglu*)
- Week 7: Unsupervised Learning and Density Estimation (*A. Cully*)
- Week 8: Genetic Algorithms (*A. Cully*)

Course 395:
Computer-Based Courseworks

Lab sessions every week, starting next week.

• Week 2 (today!): Intro and Instance Based Learning (*A. Cully*)	**No CBC**
• Week 3: Decision Trees & CBC Intro (*A. Cully*)	**CBC 1**
• Week 4: Evaluating Hypotheses (*A. Cully*)	**CBC 1**
• Week 5: Artificial Neural Networks I (*N. Cingillioglu*)	**CBC 1**
• Week 6: Artificial Neural Networks II (*N. Cingillioglu*)	**CBC 2**
• Week 7: Unsupervised Learning and Density Estimation (*A. Cully*)	**CBC 2**
• Week 8: Genetic Algorithms (*A. Cully*)	**CBC 2**

Lab sessions on:
Wednesday 10am - 12pm **OR** Thursday 16pm - 18pm (alternative session)

Course 395: Grading

NOTE

CBC accounts for 33.3% of the final grade for the Machine Learning Exam.

$$final\_grade = 2/3\ exam\_grade + 1/3\ coursework\_grade$$

Piazza

Piazza is an on-line forum and is the first place where you can seek help for this course.

We encourage you to post questions on this forums (rather than sending e-mails) and to answer one-another's questions where appropriate.

You will be automatically enrolled onto the Piazza and you will receive an e-mail confirmation from Piazza (contact us asap if this is not the case)

Please note that Piazza is an external service, so does not run on college login credentials. For (hopefully obvious) security reasons, you should not use your college password for this service.

Piazza

To encourage student engagement in piazza we will leave **questions for 1 working day** so that other students have the opportunity to answer.

Coursework related questions are answered on Piazza up to two days before the coursework deadline, so that the answer will become available in time for submission.

We cannot provide answers to questions that directly solve part of the coursework or exam-style questions.

General questions will be fielded by GTAs from Monday to Friday during normal working hours.

Course 395:
Introduction to Machine Learning

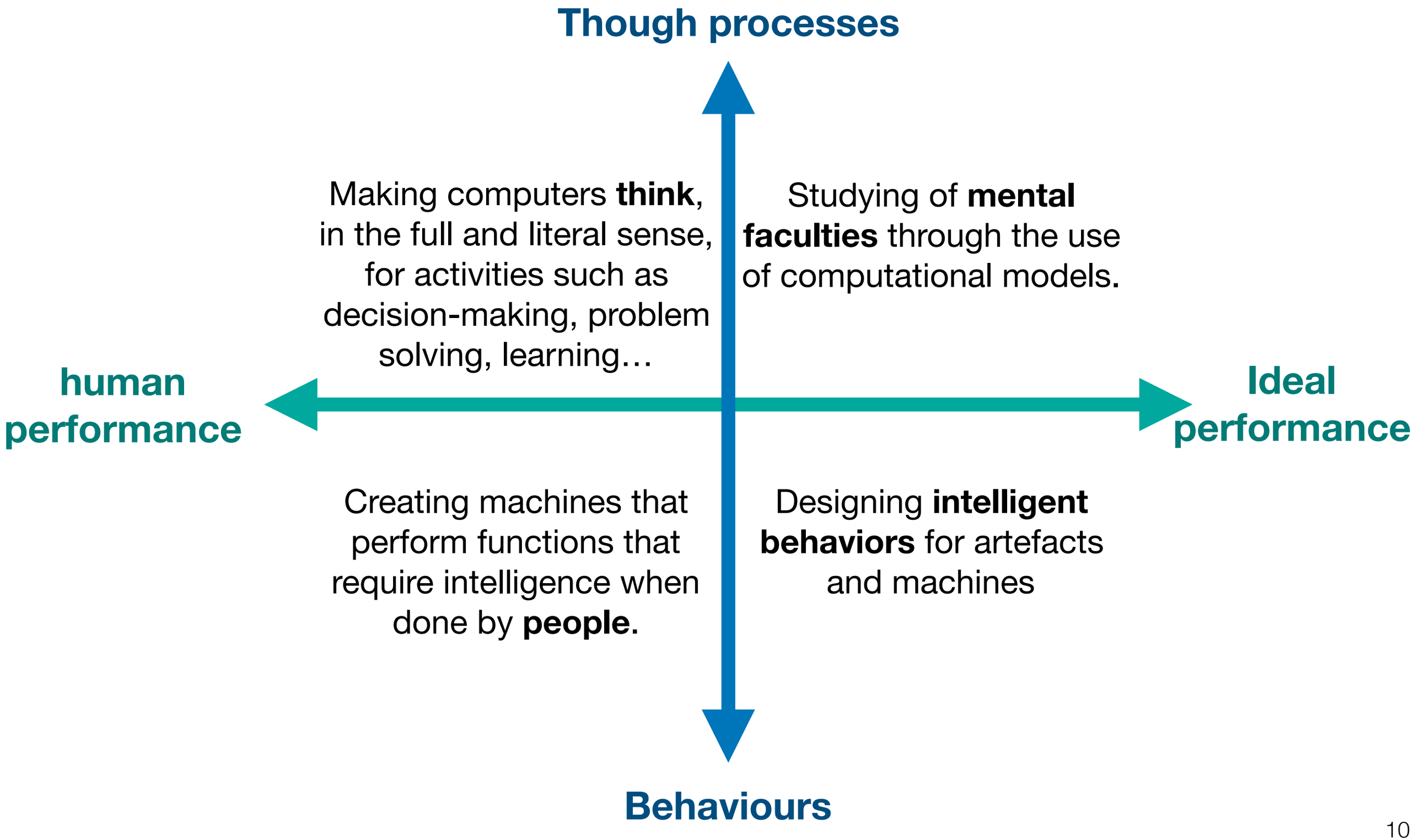
Lab sessions every week, starting next week.

- **Week 2 (today!): Intro and Instance Based Learning (*A. Cully*)**
- Week 3: Decision Trees & CBC Intro (*A. Cully*)
- Week 4: Evaluating Hypotheses (*A. Cully*)
- Week 5: Artificial Neural Networks I (*N. Cingillioglu*)
- Week 6: Artificial Neural Networks II (*N. Cingillioglu*)
- Week 7: Unsupervised Learning and Density Estimation (*A. Cully*)
- Week 8: Genetic Algorithms (*A. Cully*)

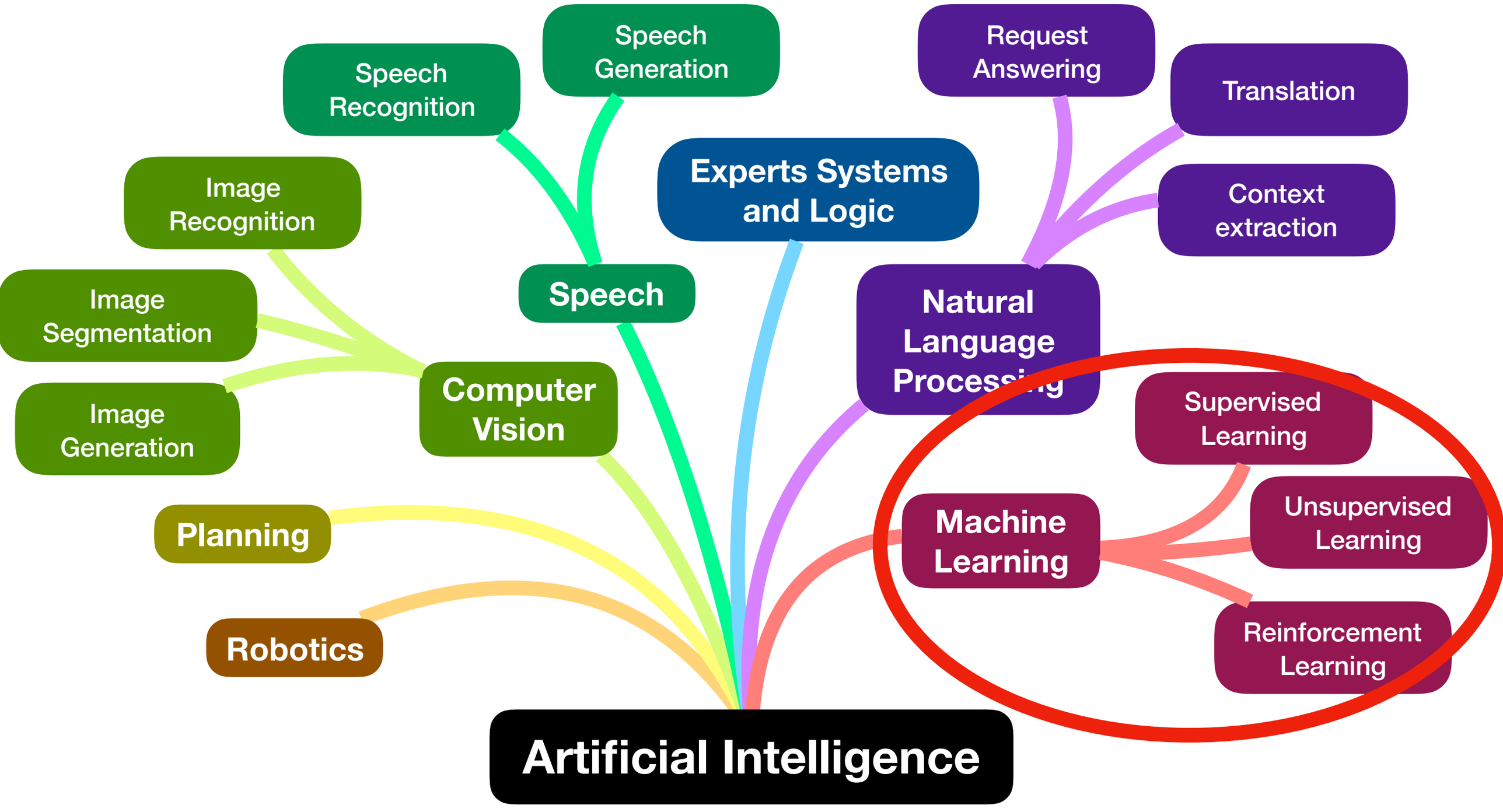
Lecture Overview?

- Introduction about AI and ML
    - Different types of ML approaches
    - Different types of ML problems
- Instance based learning
    - k-NN algorithm for classification
    - Distance-weighted k-NN
    - Curse of dimensionality
    - k-NN for regression
- Eager vs. Lazy learning

Artificial Intelligence



Artificial Intelligence



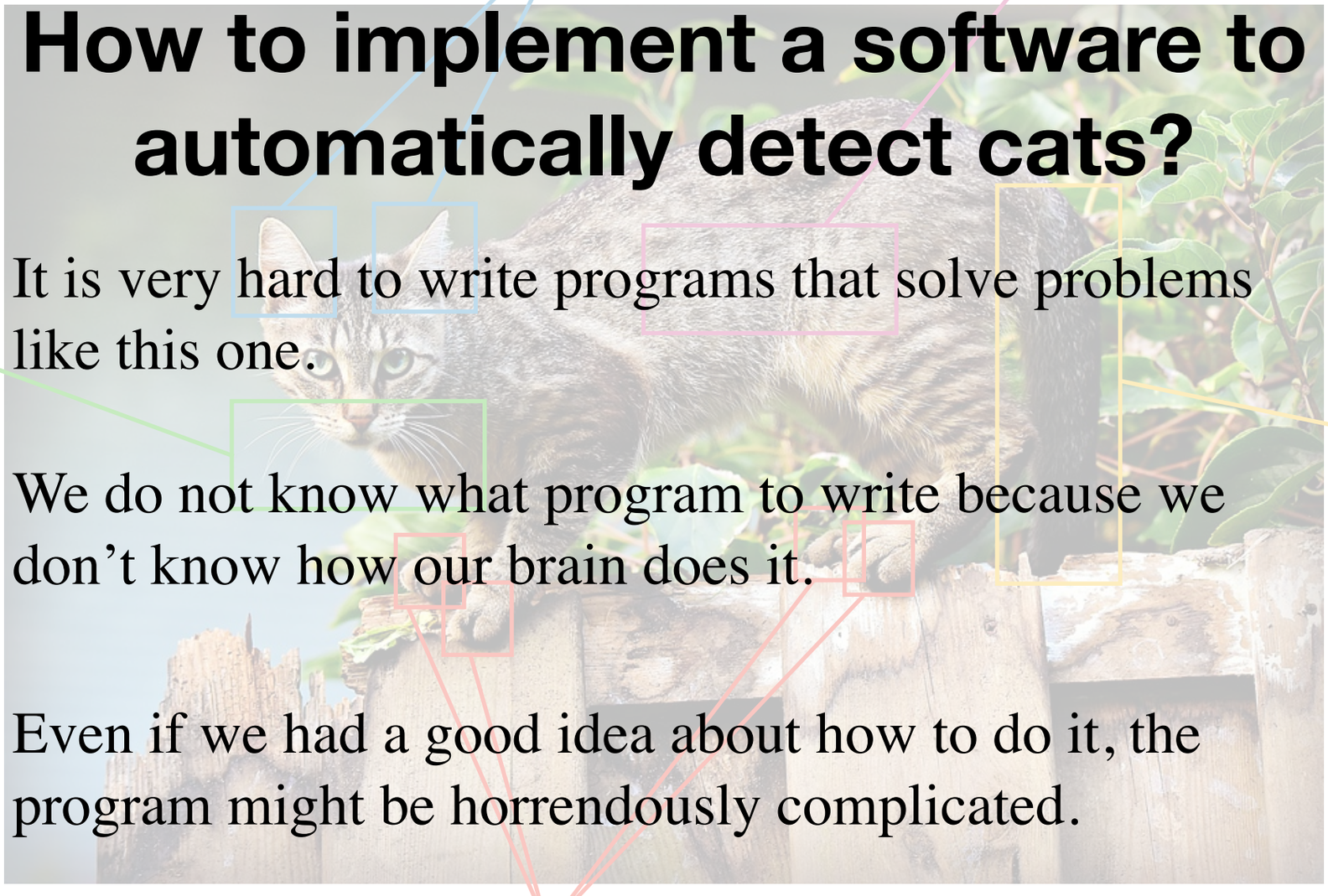
AI is the science of making machines do things that require intelligence if done by men (Minsky 1986)

Planning in real conditions

What about really real conditions?

Image recognition

an old topic in AI



How to implement a software to automatically detect cats?

It is very hard to write programs that solve problems like this one.

We do not know what program to write because we don’t know how our brain does it.

Even if we had a good idea about how to do it, the program might be horrendously complicated.

The machine learning approach

Instead of writing a program by hand, **we collect lots of examples that specify the correct output for a given input**. A machine learning algorithm then takes these examples and produces a program that does the job.

The program produced by the learning algorithm may look very different from a typical hand-written program. If we do it right, the program works for new cases as well as the ones we trained it on.

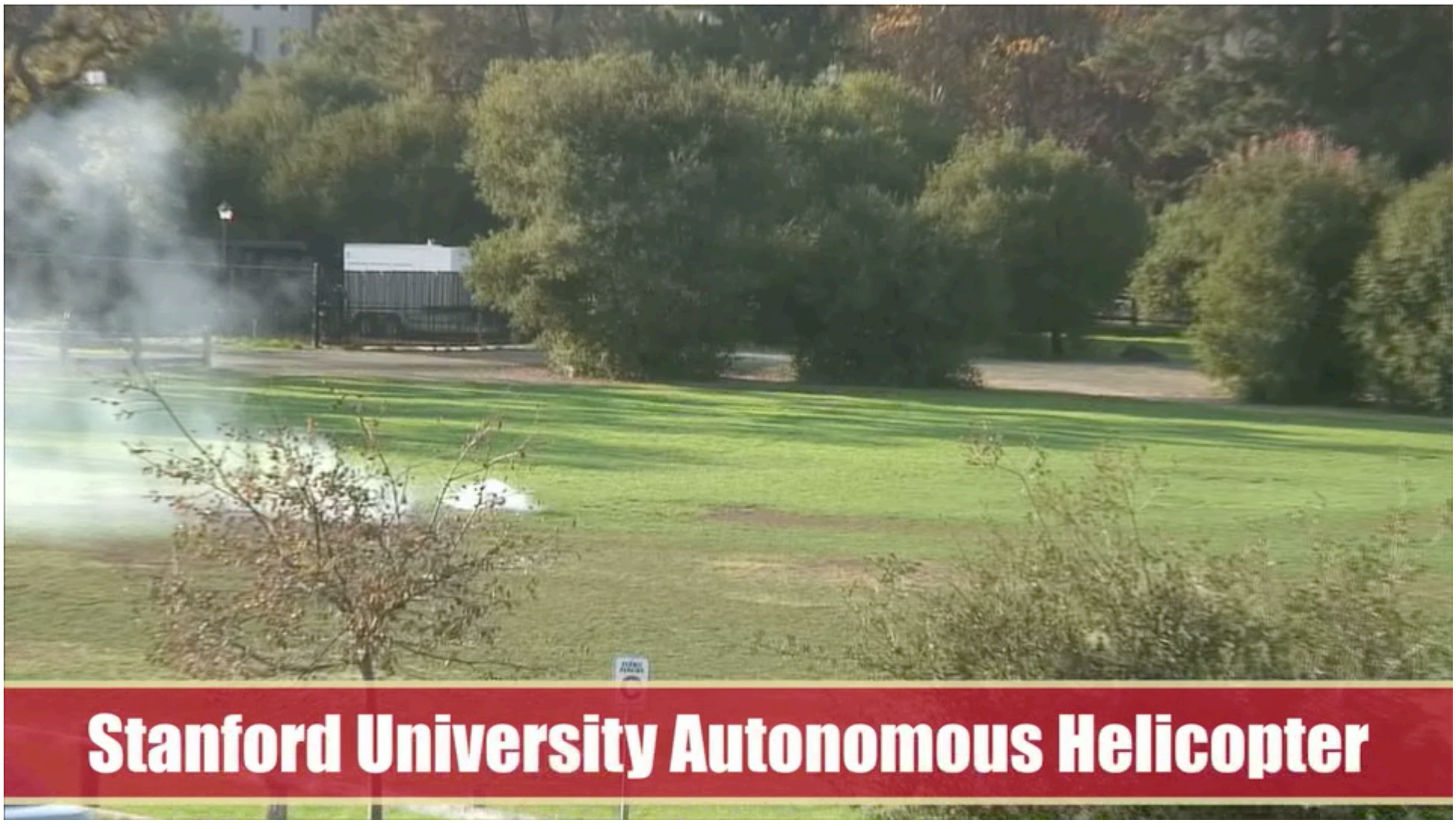
Machine Learning

"The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience."

Mitchell, T. M. (1997). *Machine learning*.

Examples

Mimicking pro-pilotes



Examples

Automatic Detection and Segmentation of Brain Lesions

Dr Ben Glocker

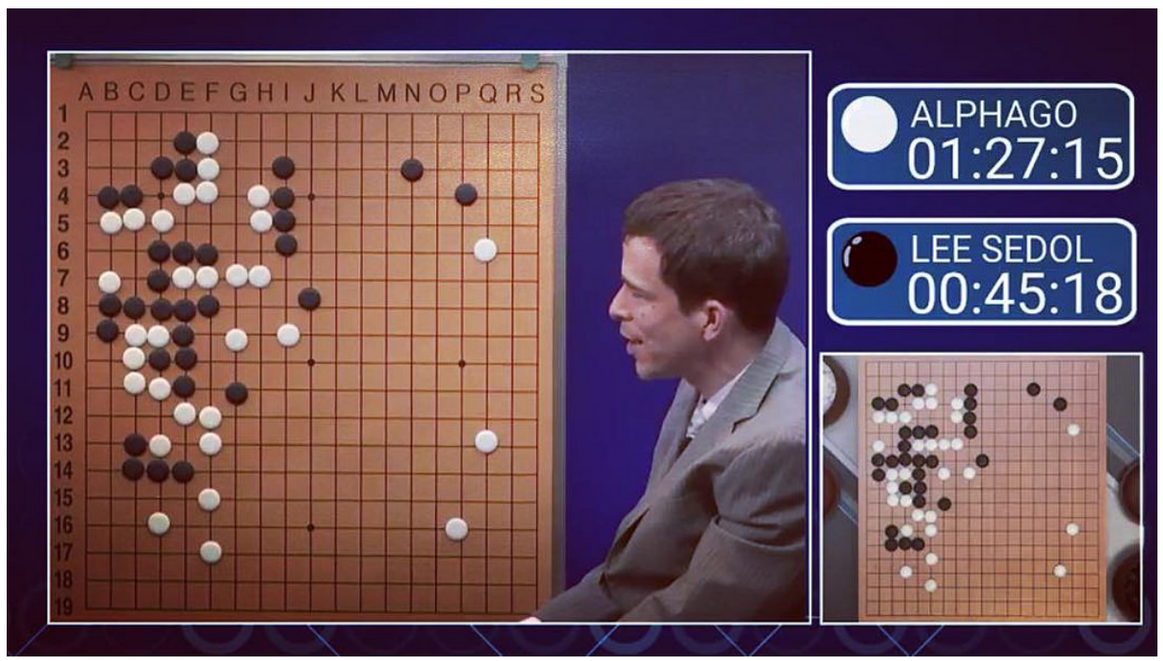
http://wp.doc.ic.ac.uk/bglocker/project/brain-lesions/

Examples

Automatic generation of super-human AI for games.

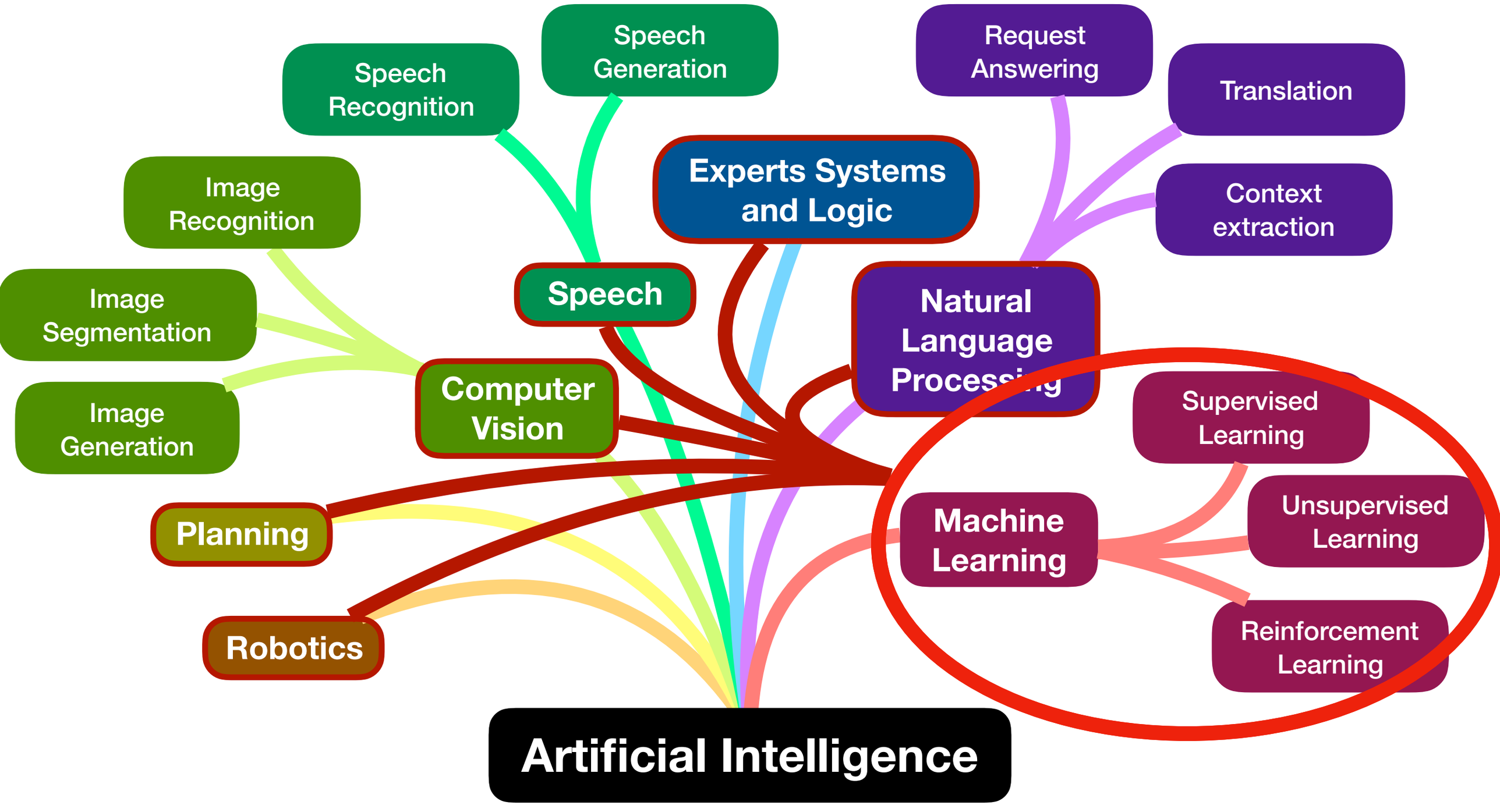
DQN from DeepMind

OpenAI Five (on Dota)



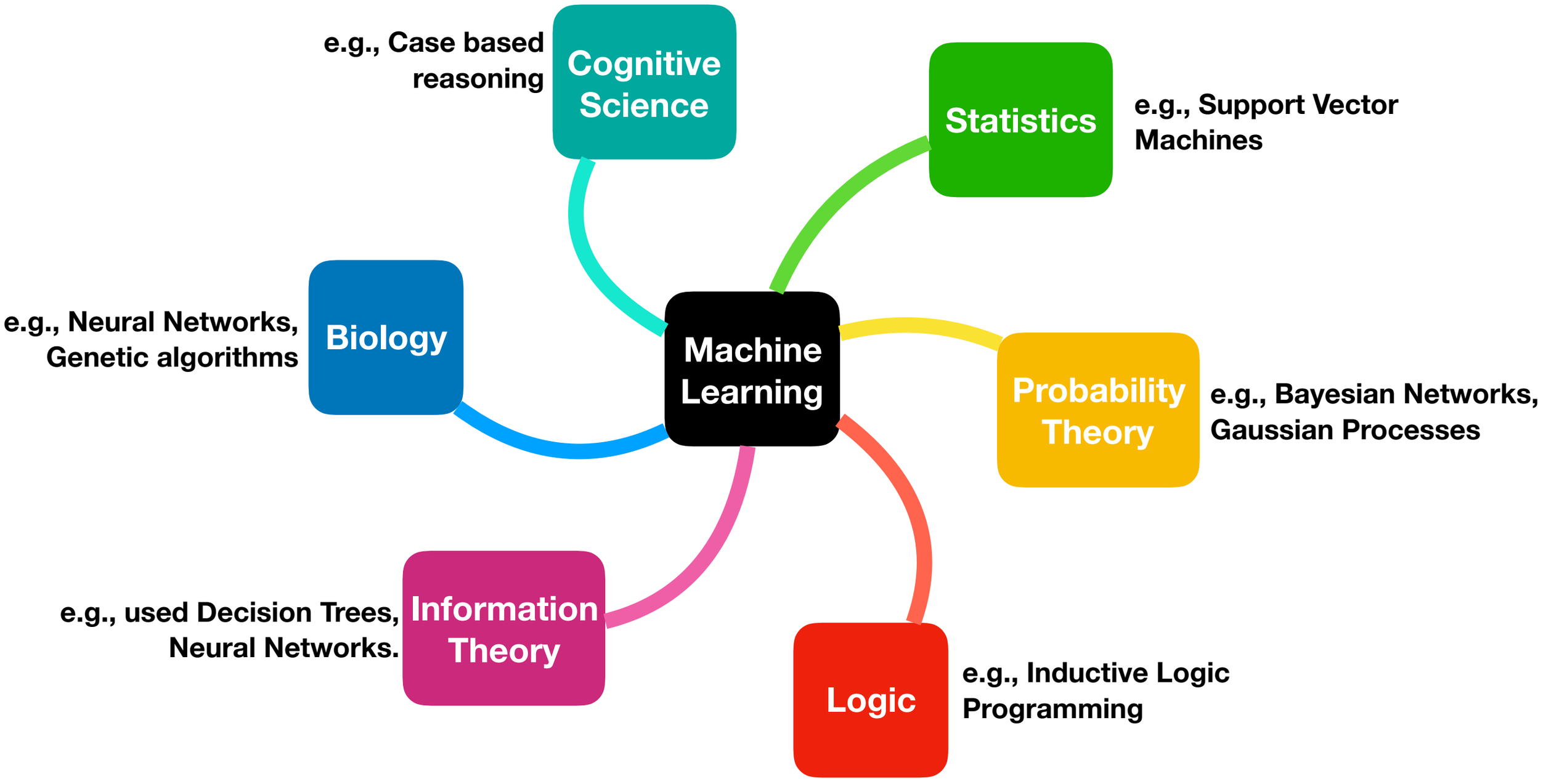
AlphaGo

Artificial Intelligence



AI is the science of making machines do things that require intelligence if done by men (Minsky 1986)

Machine Learning

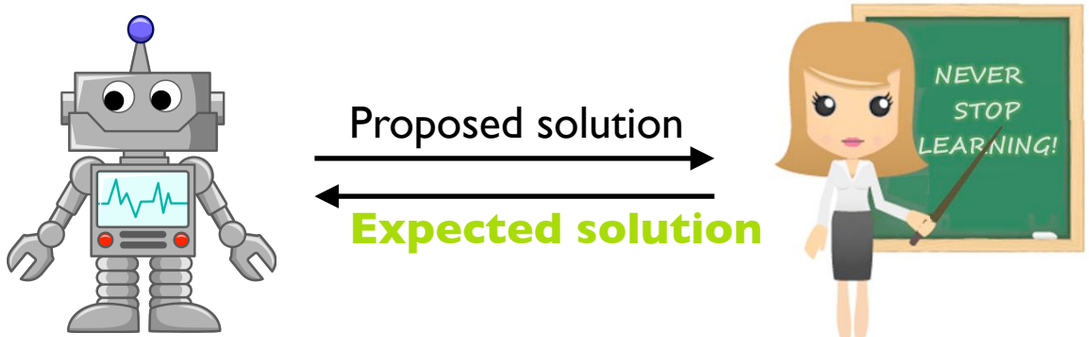


Everything is about Data in machine learning. This is why the "ML boom" is happening after the "Big Data boom"

Machine Learning

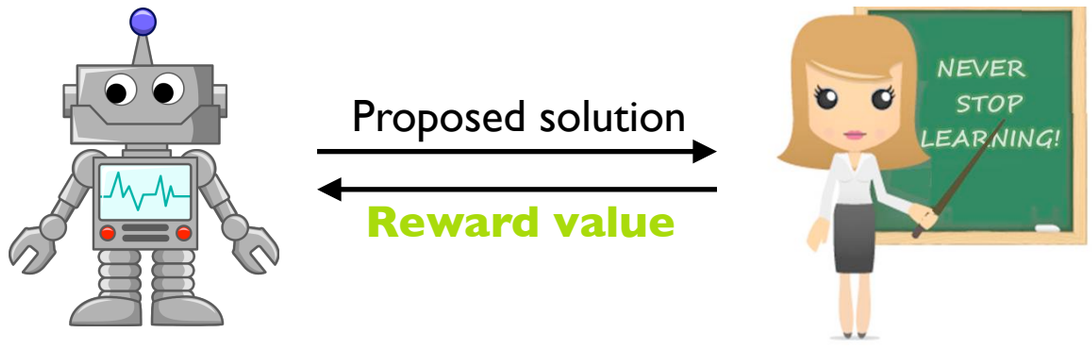
There are three main families of ML algorithms/approaches:

Supervised Learning



Unsupervised Learning

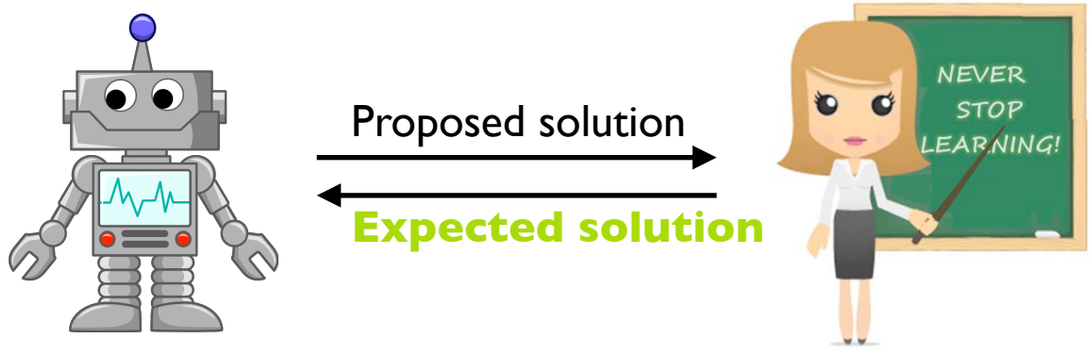
Reinforcement Learning



Supervised Learning

Definition: Supervised Learning

Learn an unknown mapping $f(X_i) = Y_i$ from training data given in the form of input x_i and output y_i pairs $D = \{(X_i, Y_i)\}_{i=1}^{N}$



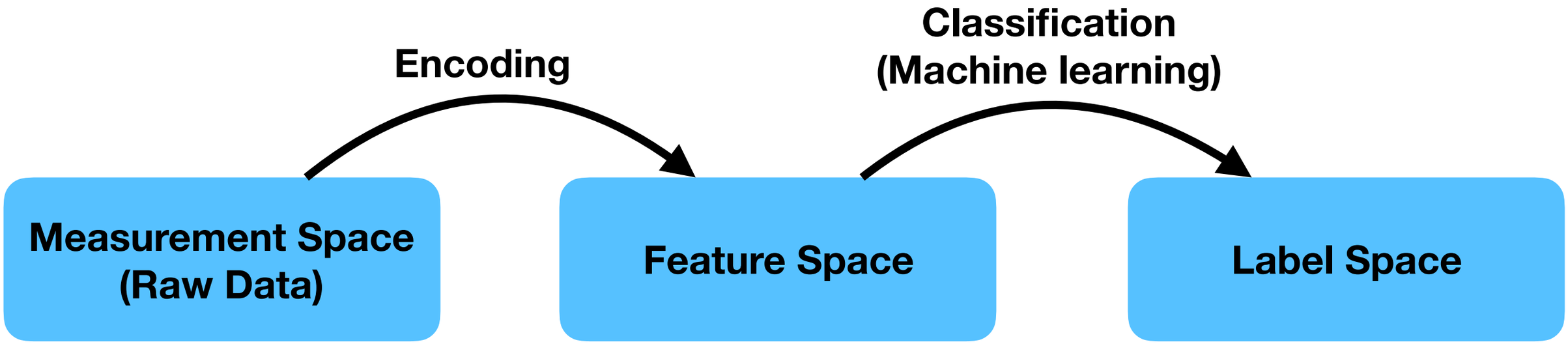
D is called the training set and N is the number of training examples/samples/ data points.

The x_i are in the feature space $\mathcal{X}$ and the $y_i \in \mathcal{Y}$ are in the label space.

In the simplest setting, each training input x_i is a D-dimensional vector of numbers. These are called features, attributes or covariates. In general, however, x_i could be a complex structured object (e.g. an image, a sentence, an email message, a time series, a molecular shape, a graph, etc).

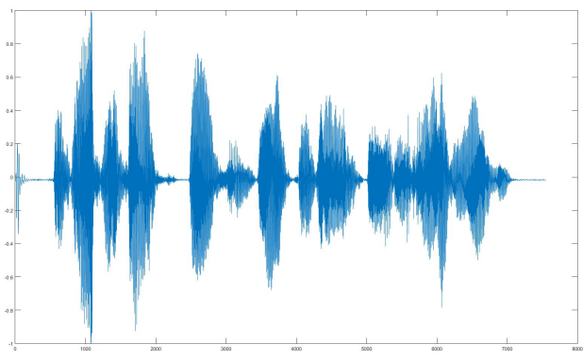
Spaces

(before the Deep Learning era)

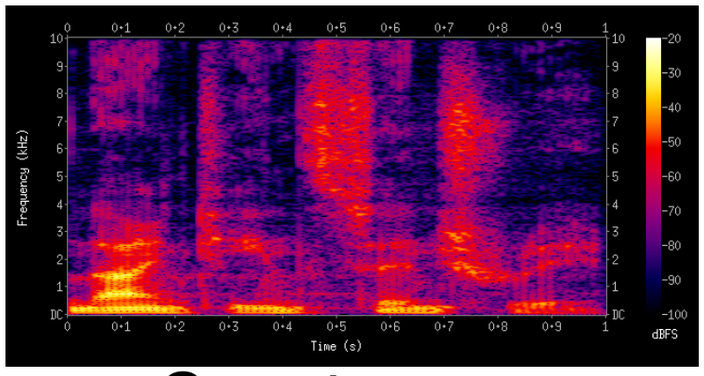


(usually of lower dimension)

Example:



Audio signal

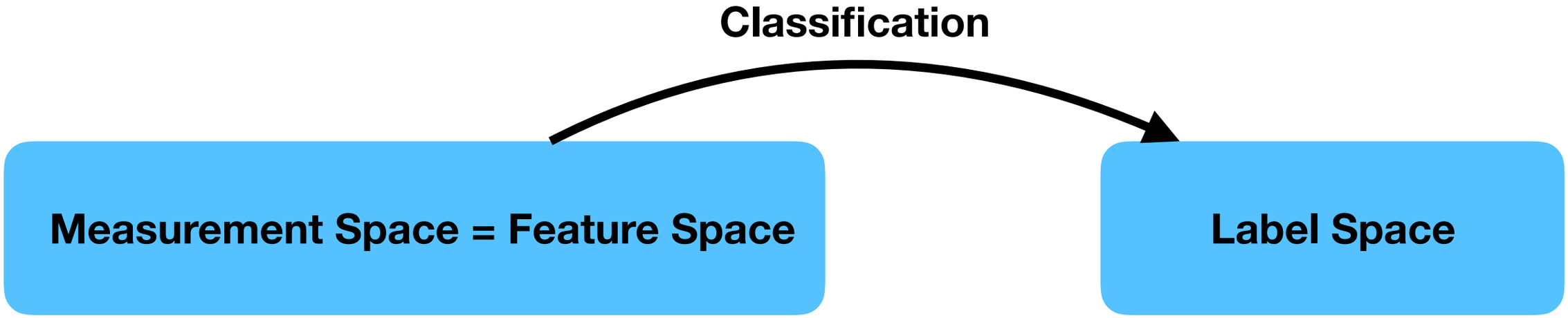


Spectrogram

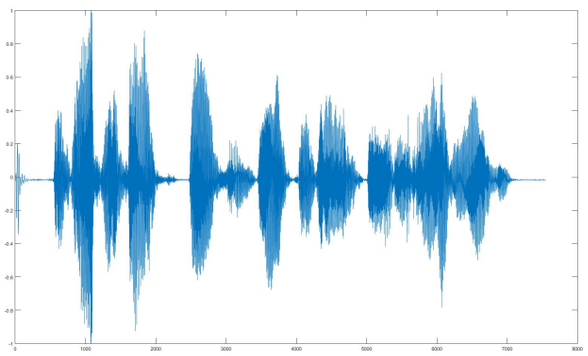
"nineteenth century"

Spaces

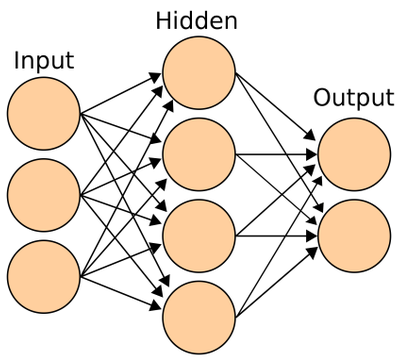
(After the Deep Learning era)



Example:



Audio signal



"nineteenth century"

Label Space

There are three main types of Label Spaces:

- **Categorical** (also called class): y_i is a categorical or nominal variable from some finite set $y_i \in \{1, ..., C\}$ (e.g. “Apple","Banana", …). In this cas, the supervised learning task is known as **classification** or **pattern recognition**.

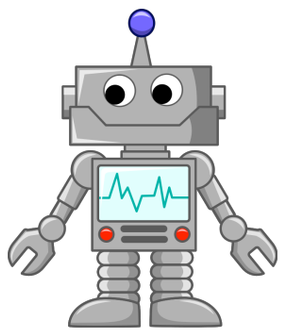
- **Real-valued scalar** (such as credit score). In this case the task is known as **regression** or **function approximation**.

- Another variant, known as **ordinal regression**, occurs where label space Y has some natural ordering, such as school grades I,II.1,II.2,etc. This is can be carefully approximated as a regression problem that is subsequently discretised.

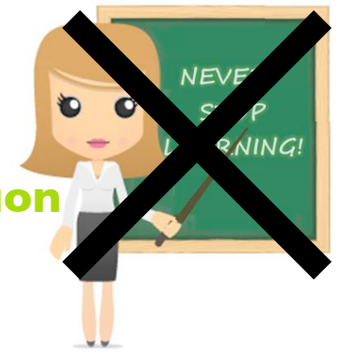
Unsupervised Learning

Definition: Unsupervised Learning

Discover an underlying/"hidden"/"latent" structure within data x



Self-organisation



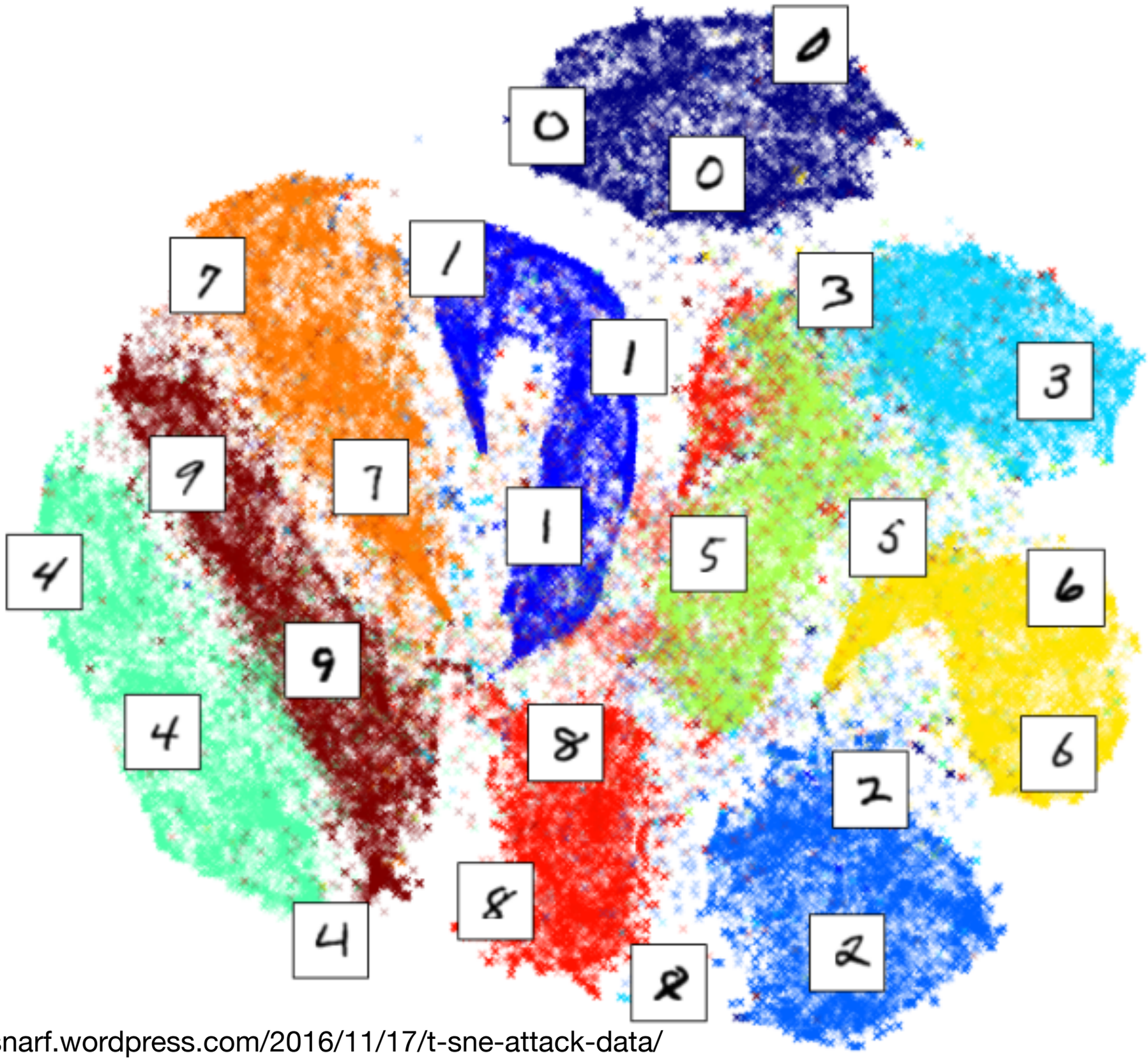
The aim is to find an underlying structure that explains the data *x* in a more efficient way.
We can think of this as reducing the amount of bits required to store the "important" features of the data set, akin to **lossy data compression**.

This is possible in broadly two ways by:

- reducing the dimensions in the data, **dimensionality reduction**
- assigning the data to automatically defined categorical labels, **clustering**.

MNIST dataset

Two-dimensional embedding of 70,000 handwritten digits with t-SNE

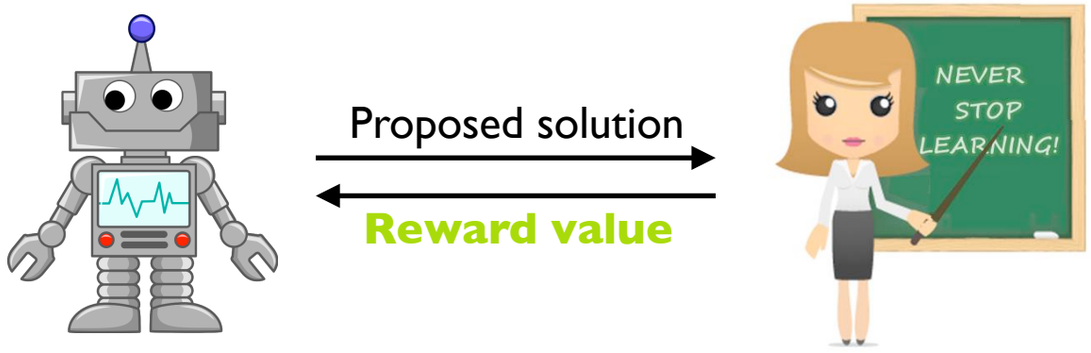


From https://bigsnarf.wordpress.com/2016/11/17/t-sne-attack-data/

Reinforcement Learning

Definition: Reinforcement Learning

Find which action to take in order to maximise the received rewards.

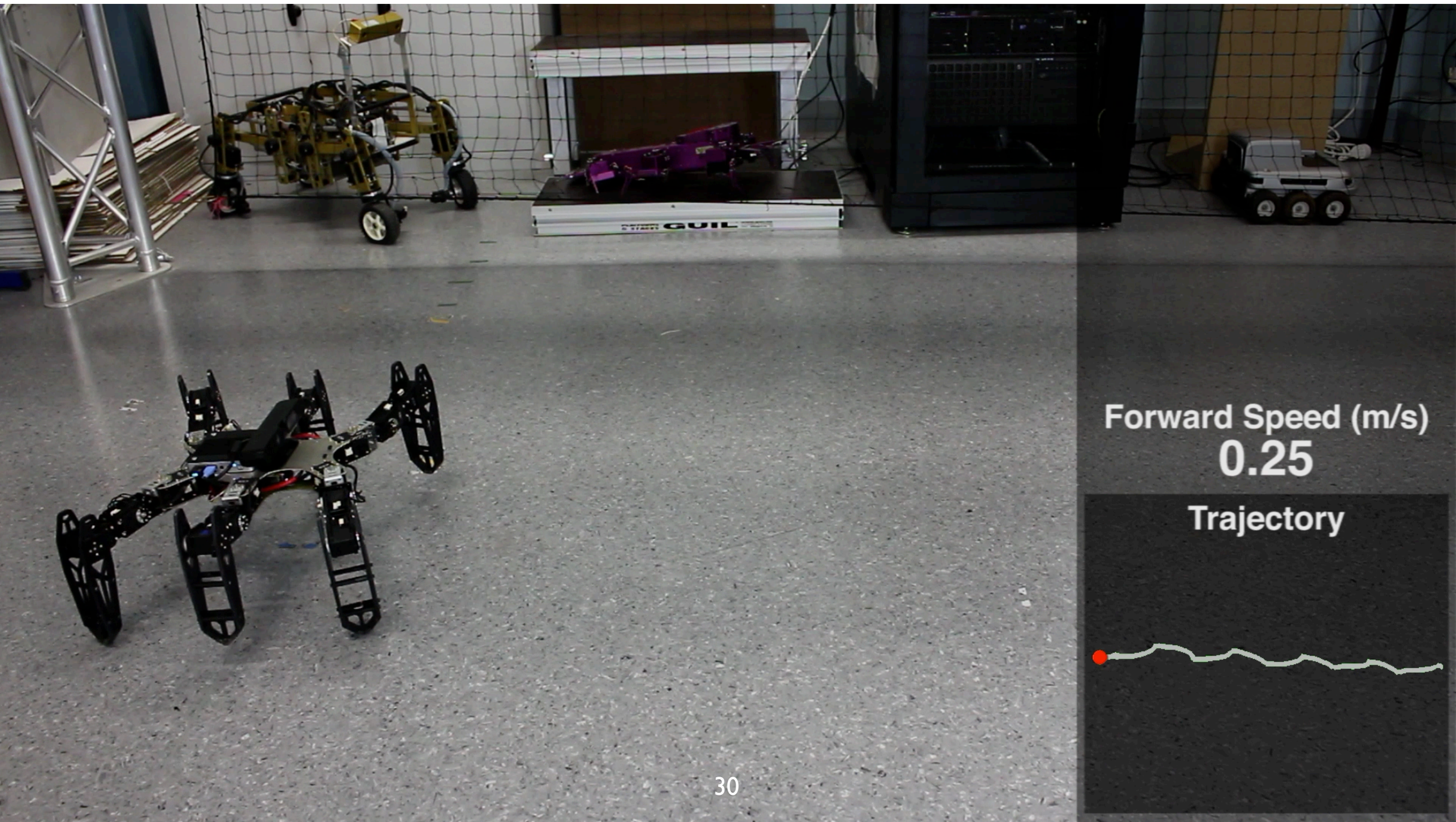


Main differences compared to the other approaches:

- The best (or correct) solution is not given to the agent, **only a reward signal**.
- Feedback is usually delayed, not instantaneous.
- Time really matters (data is sequential, non i.i.d data).
- Agent's decisions affect the subsequent data it receives.

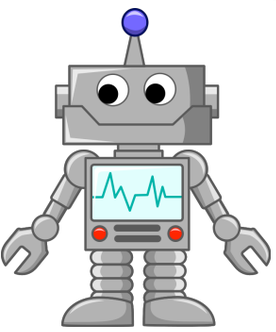
Reinforcement learning for adaptation

Cully et al. Nature 2015



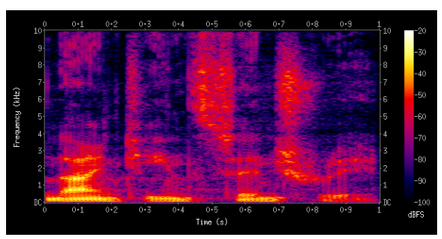
Machine Learning Sum up

Supervised Learning



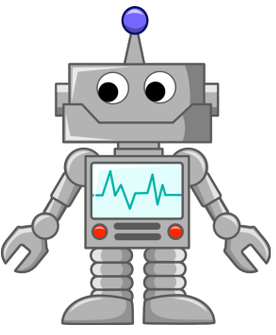
Proposed solution

Expected solution

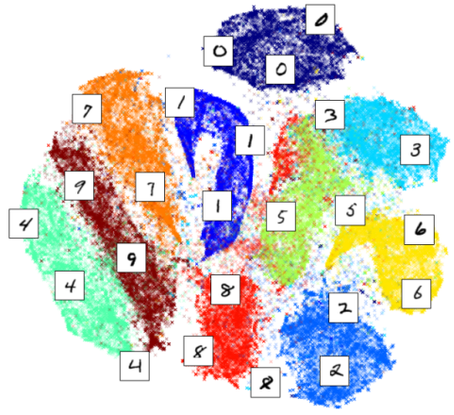


"nineteenth century"

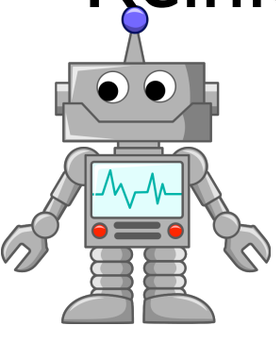
Unsupervised Learning



Self-organisation

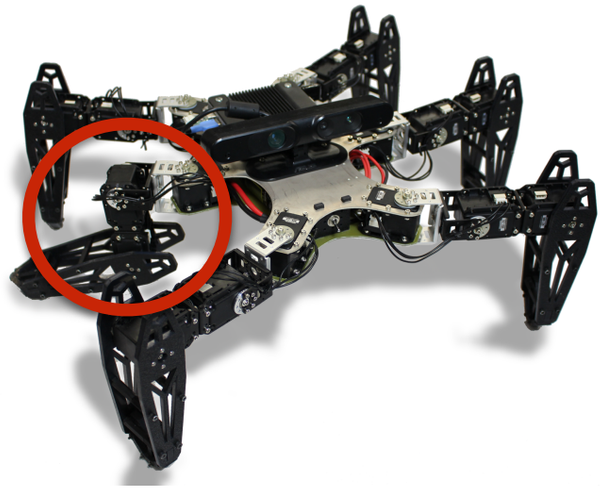


Reinforcement Learning



Proposed solution

Reward value

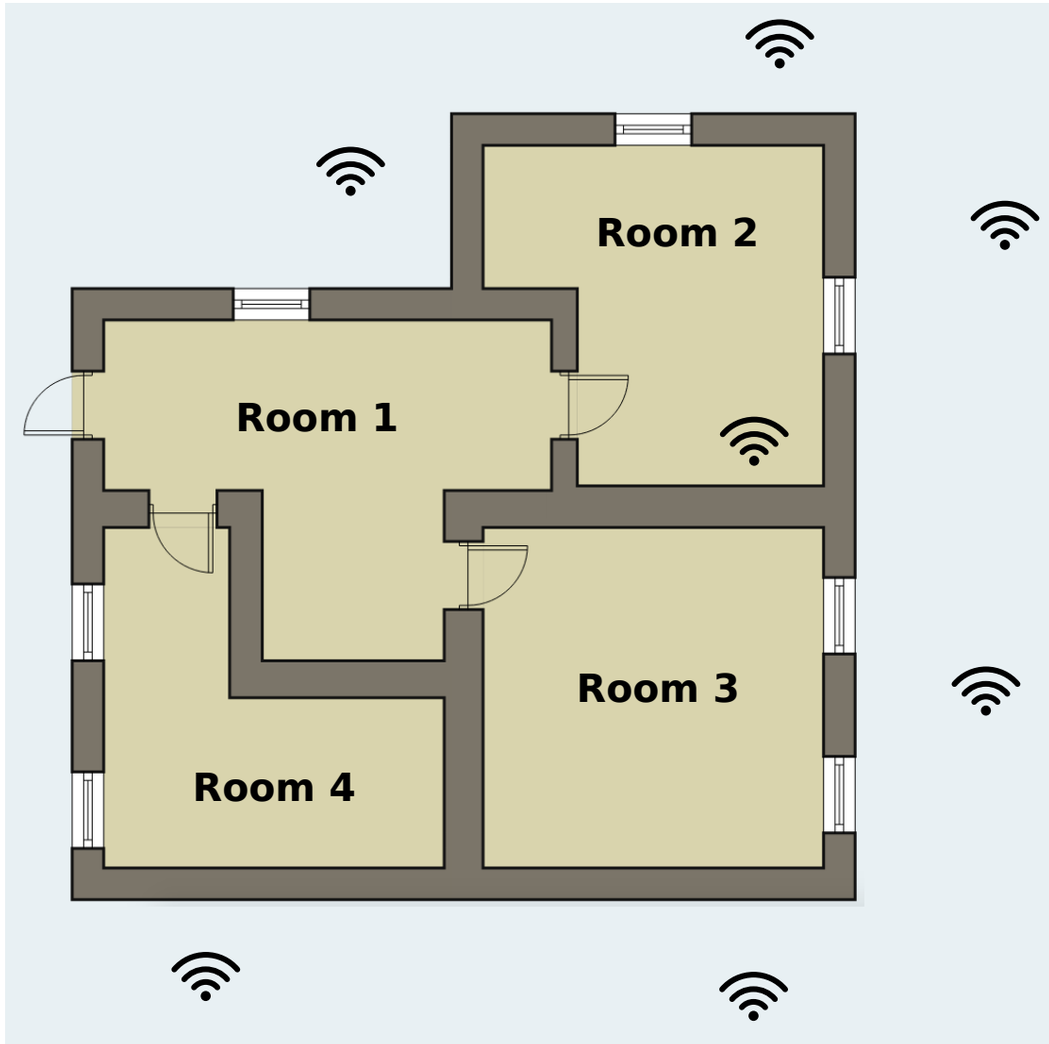


The main ML problems

- Classification
- Regression
- Clustering
- Dimensionality reduction
- Density Estimation
- Policy Search

Classification

Predict the right label for an unknown sample



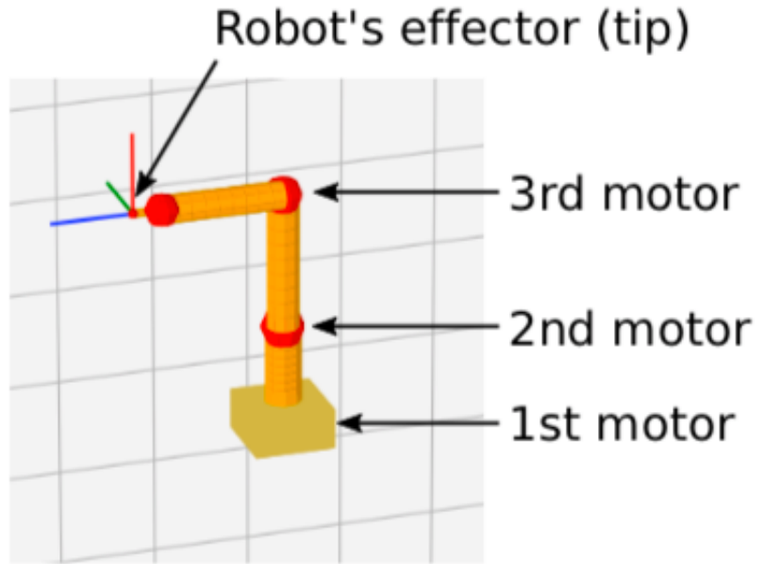
Objective of the first CBC

Appropriate ML approaches: Supervised learning & unsupervised learning

Regression

Approximate an unknown function

Real ABB IRB 120



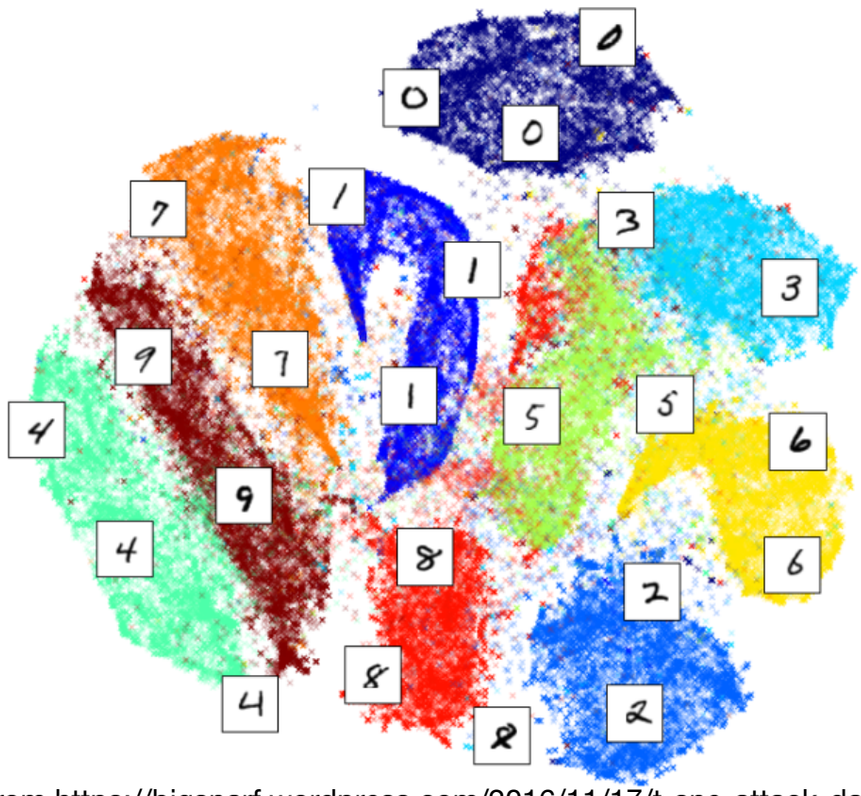
Simulated ABB IRB 120

Objective of the second CBC

Appropriate ML approaches: Supervised learning

Clustering

Group data in such a way that data points in the same group (called a **cluster**) are more similar to each other than to those in other clusters.



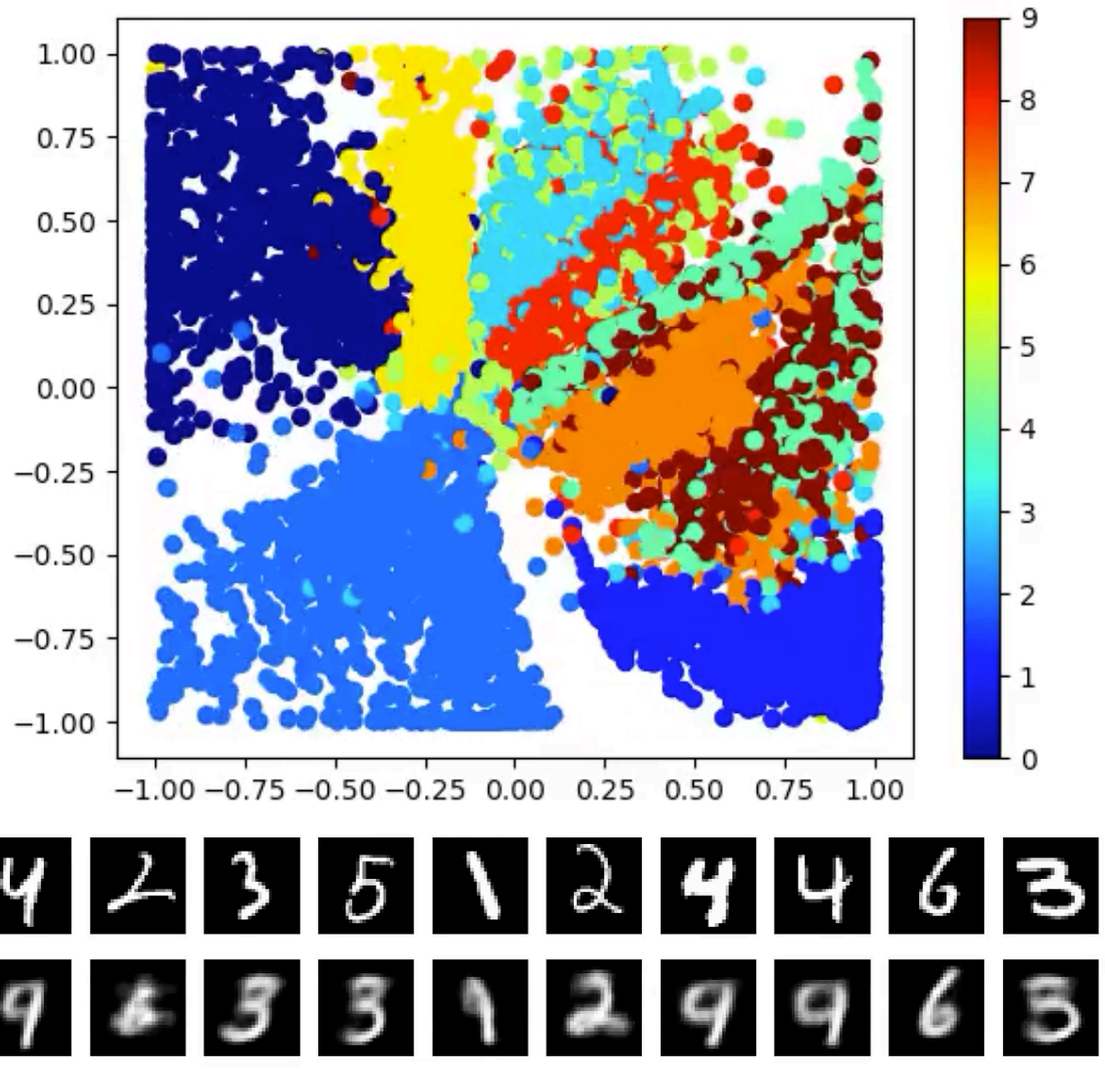
From https://bigsnarf.wordpress.com/2016/11/17/t-sne-attack-data/

This is a usual unsupervised learning approach to perform classification.

Appropriate ML approaches: Unsupervised learning

Dimensionality Reduction

Reduce the dimensionality of the observed data

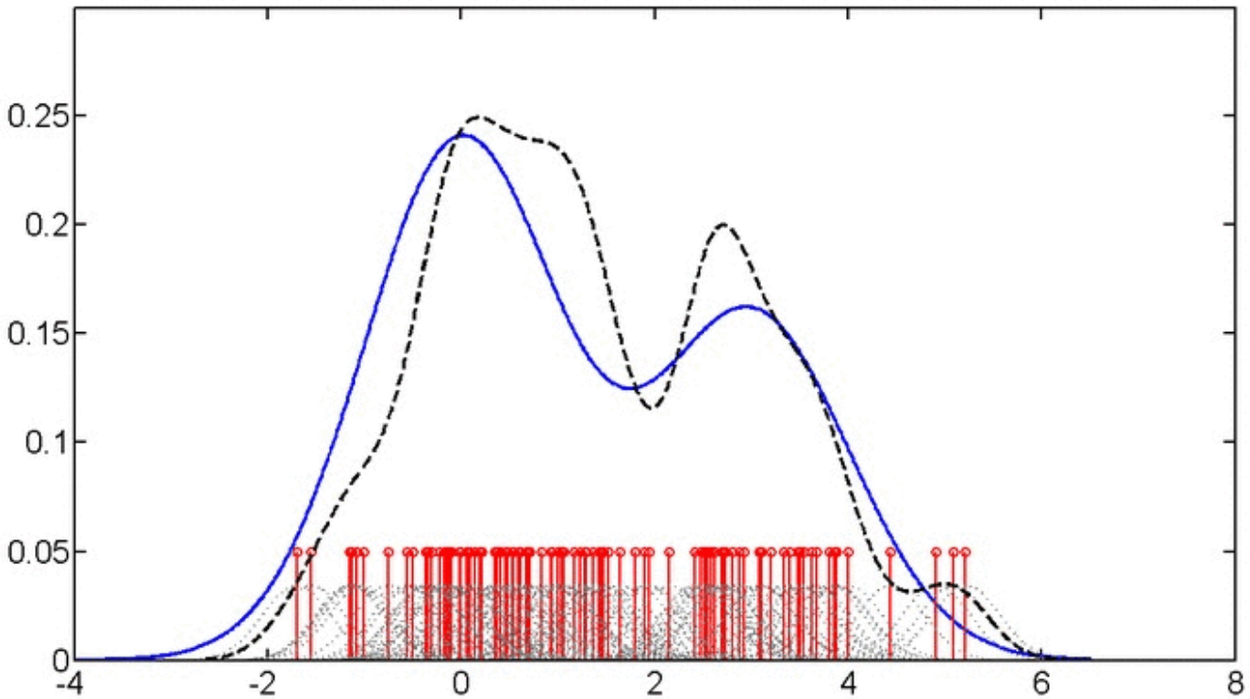


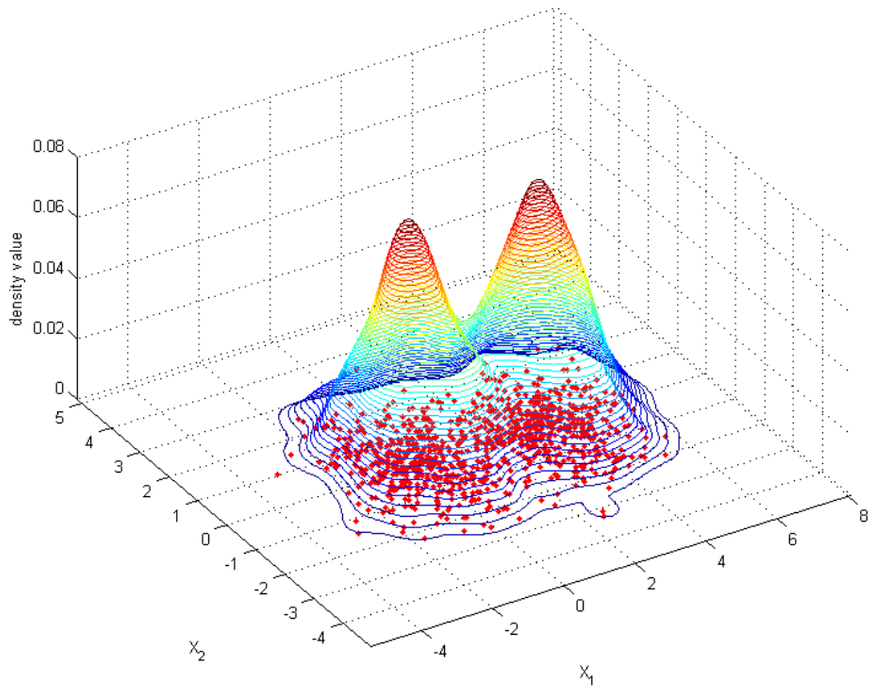
Appropriate ML approaches: Unsupervised learning

Not covered in this Introduction to ML

Density Estimation

Estimate unobservable underlying probability density function based on observed data.



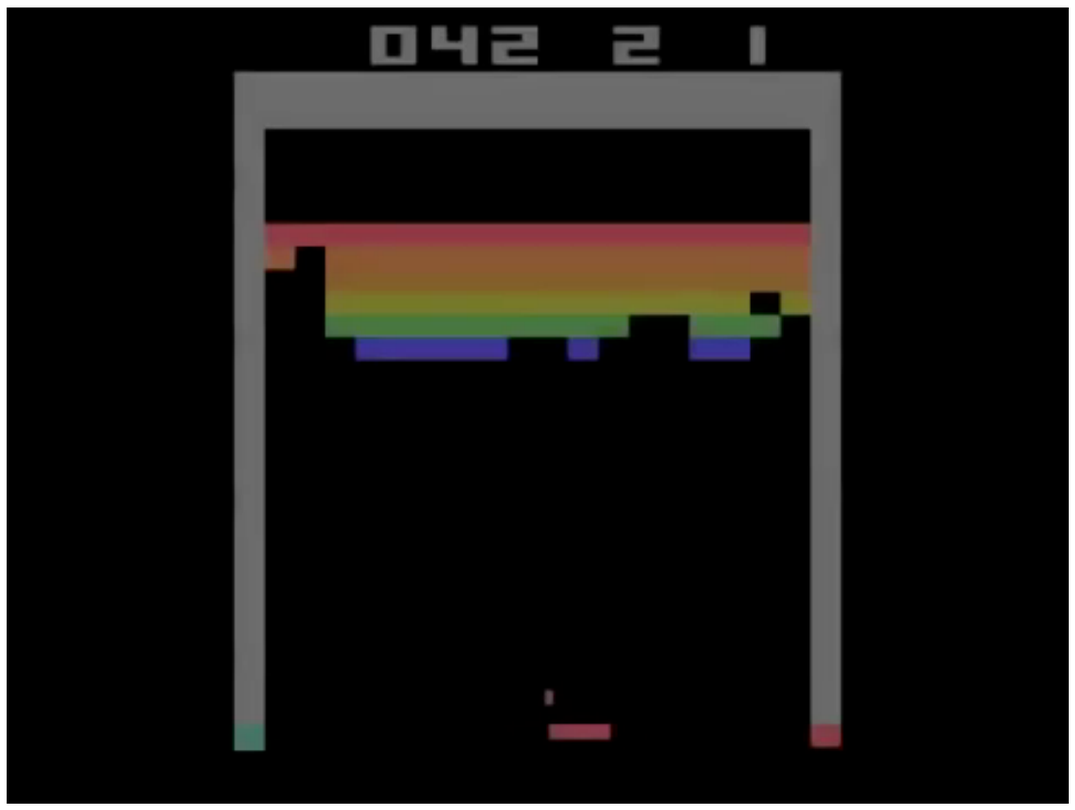


Appropriate ML approaches: Unsupervised learning

Policy Search

(or RL problems)

Find which action an agent should take, depending on its current state, to maximise the received rewards.



DQN from DeepMind

Kormushev et al., 2010

Appropriate ML approaches: Reinforcement learning

Not covered in this Introduction to ML

Sum-up

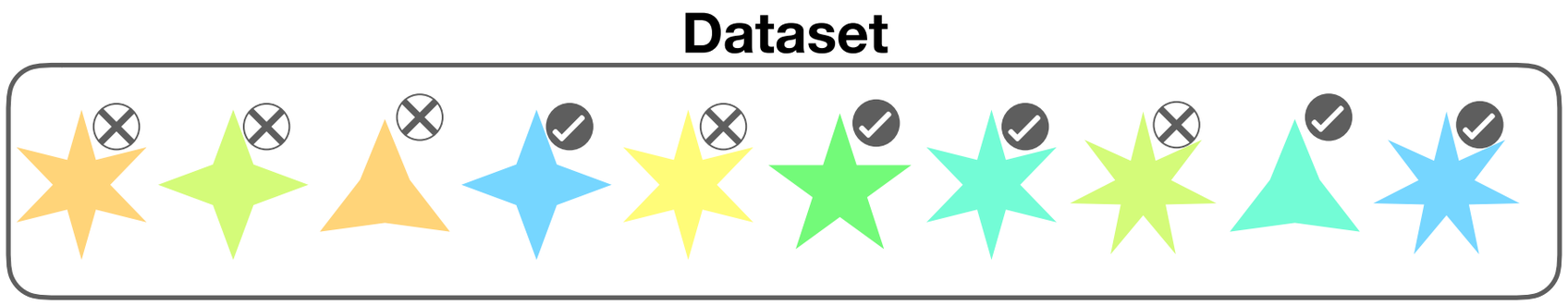
Different types of approach, for different types of problems.

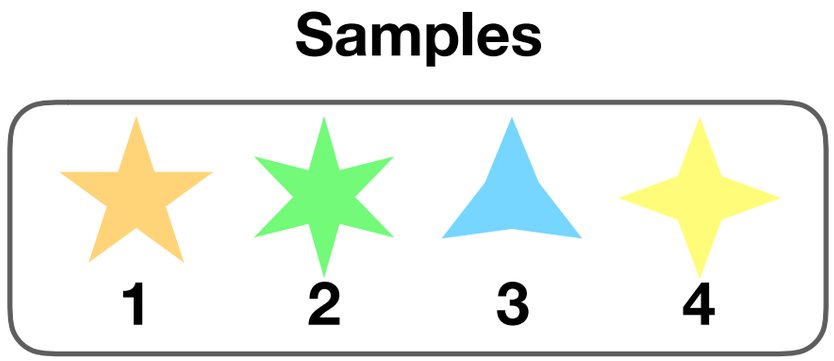
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

- Classification
- Regression
- Clustering
- Dimensionality reduction
- Density Estimation
- Policy Search

Instance Based Learning

Are you a good Machine Learner?



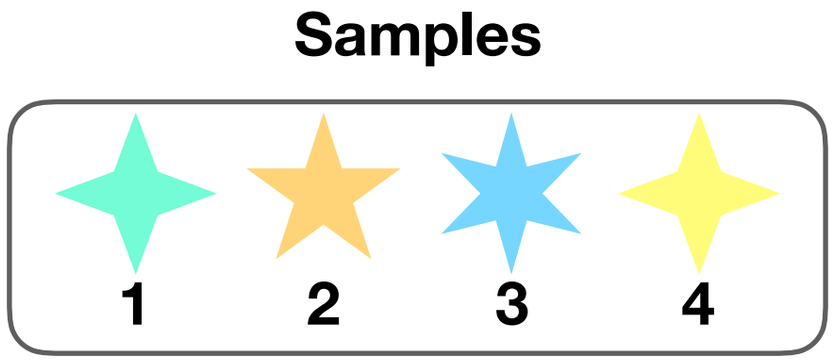


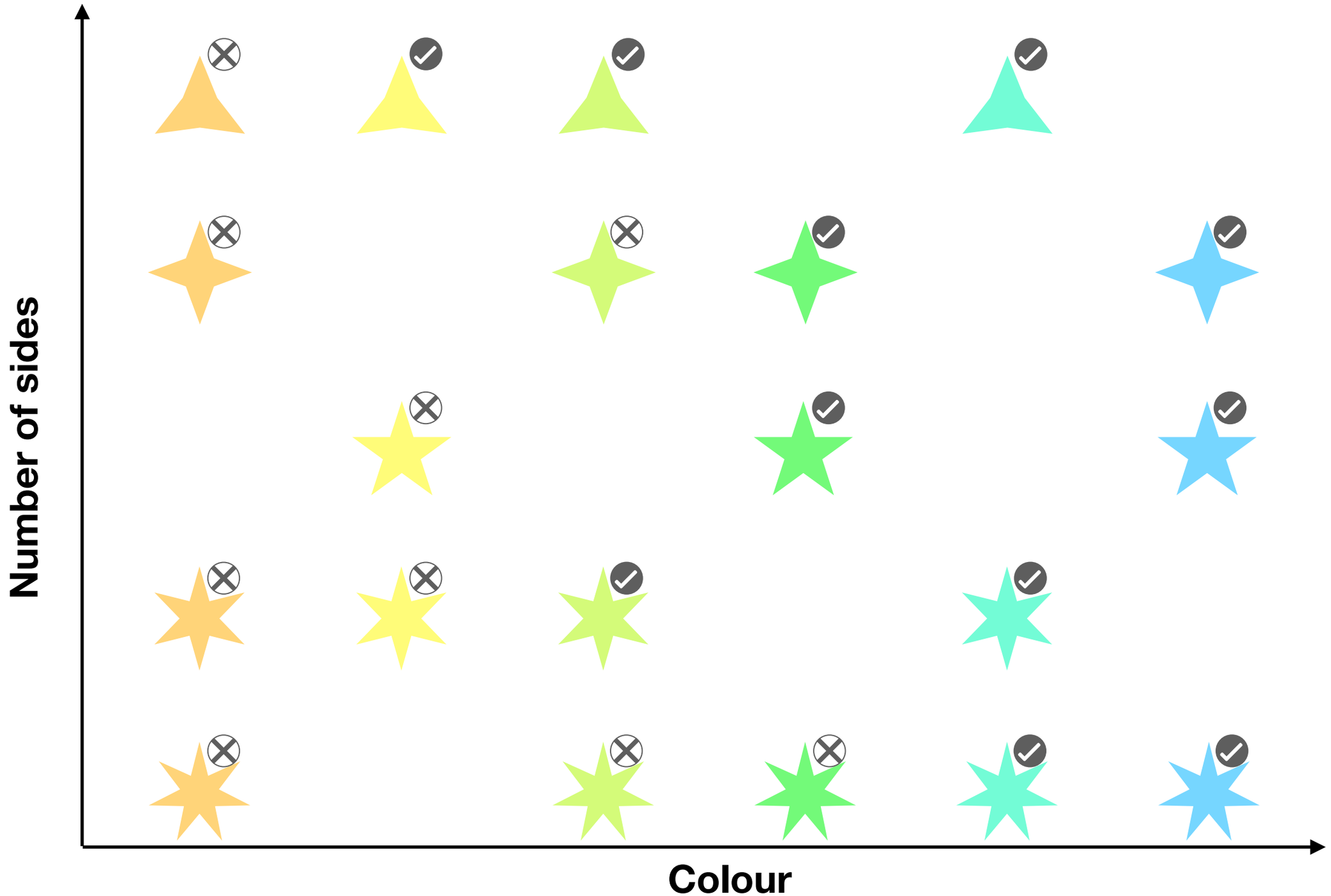
Are you a good Machine Learner?

Dataset

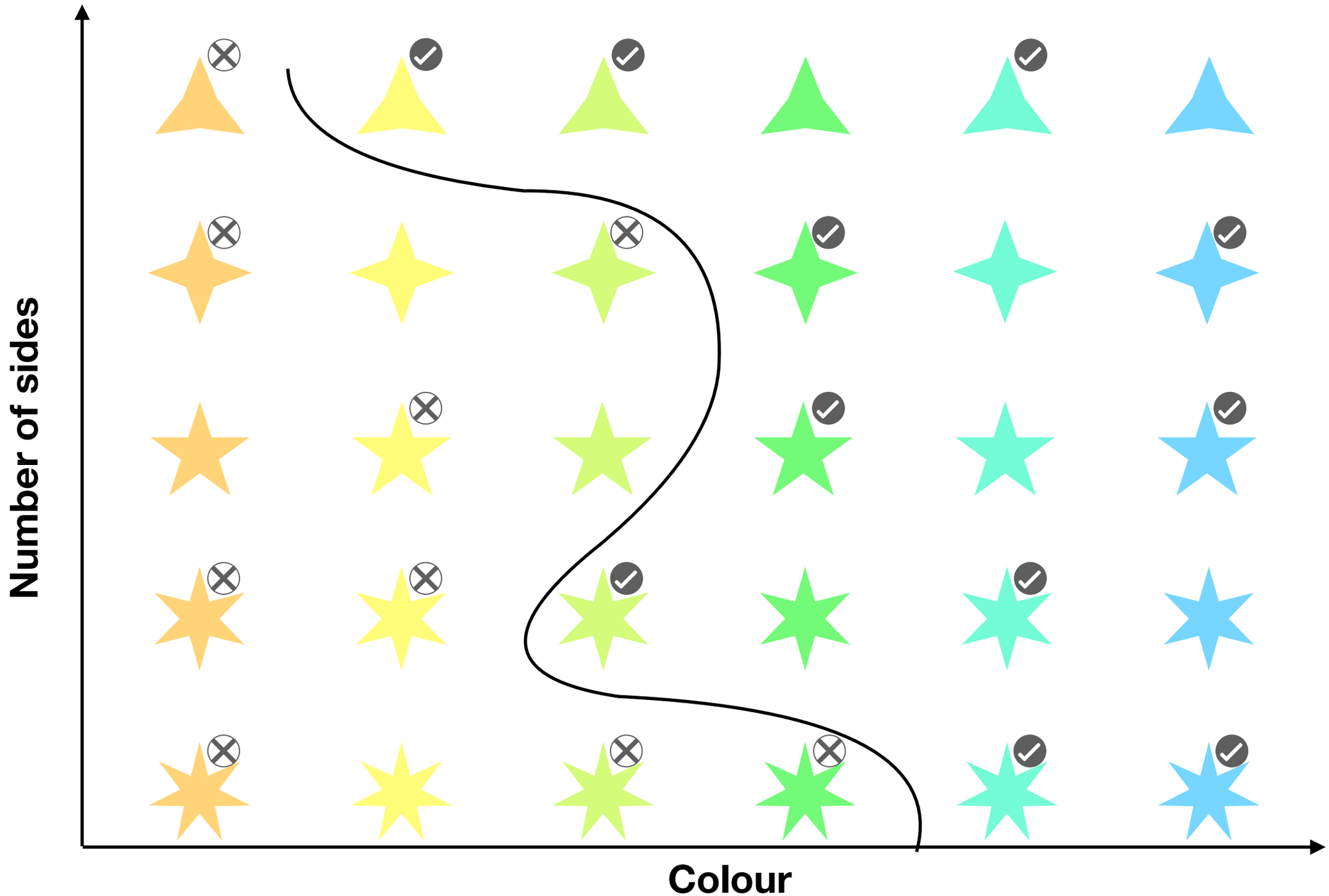
Samples

Are you a good Machine Learner?

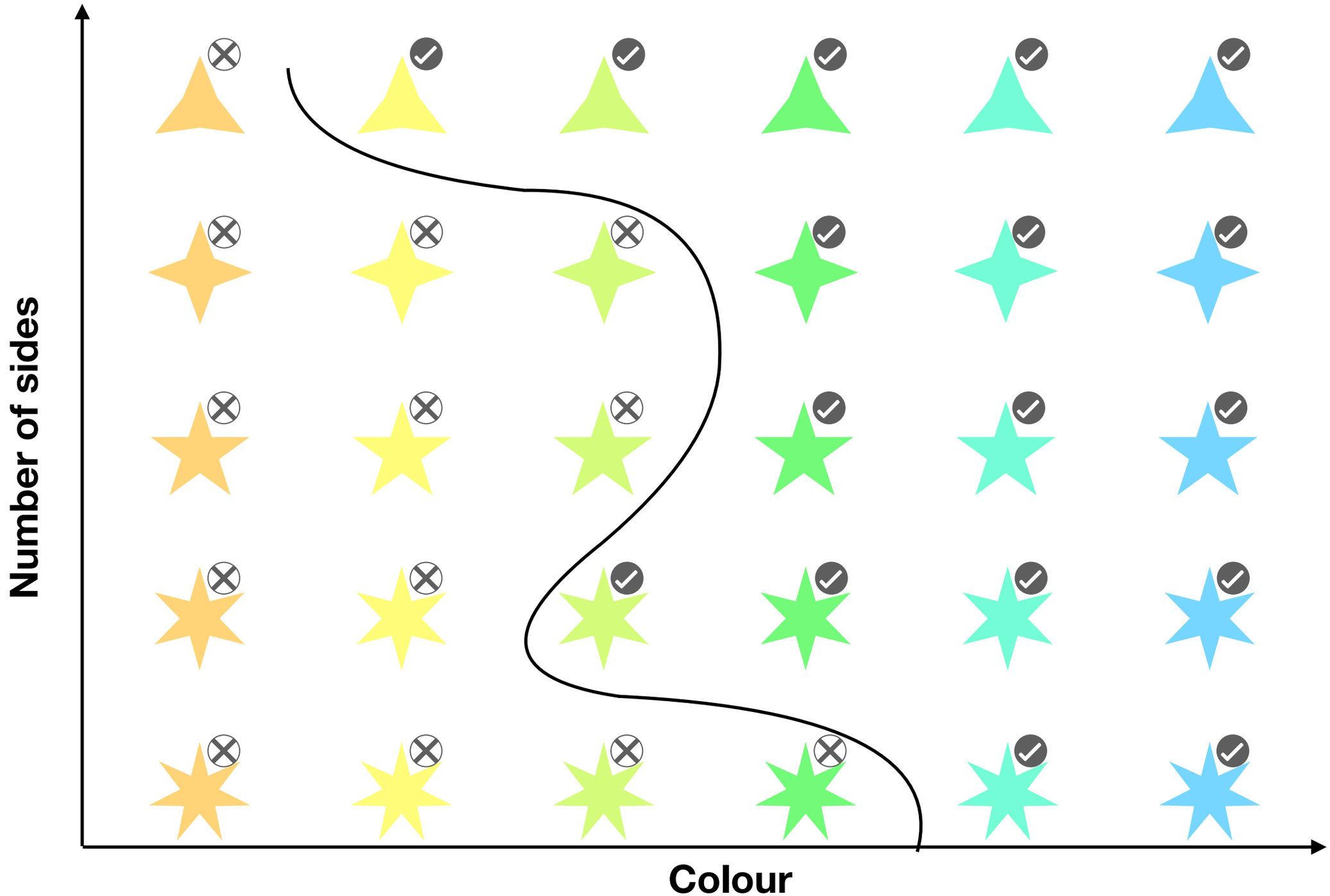




Are you a good Machine Learner?



Are you a good Machine Learner?



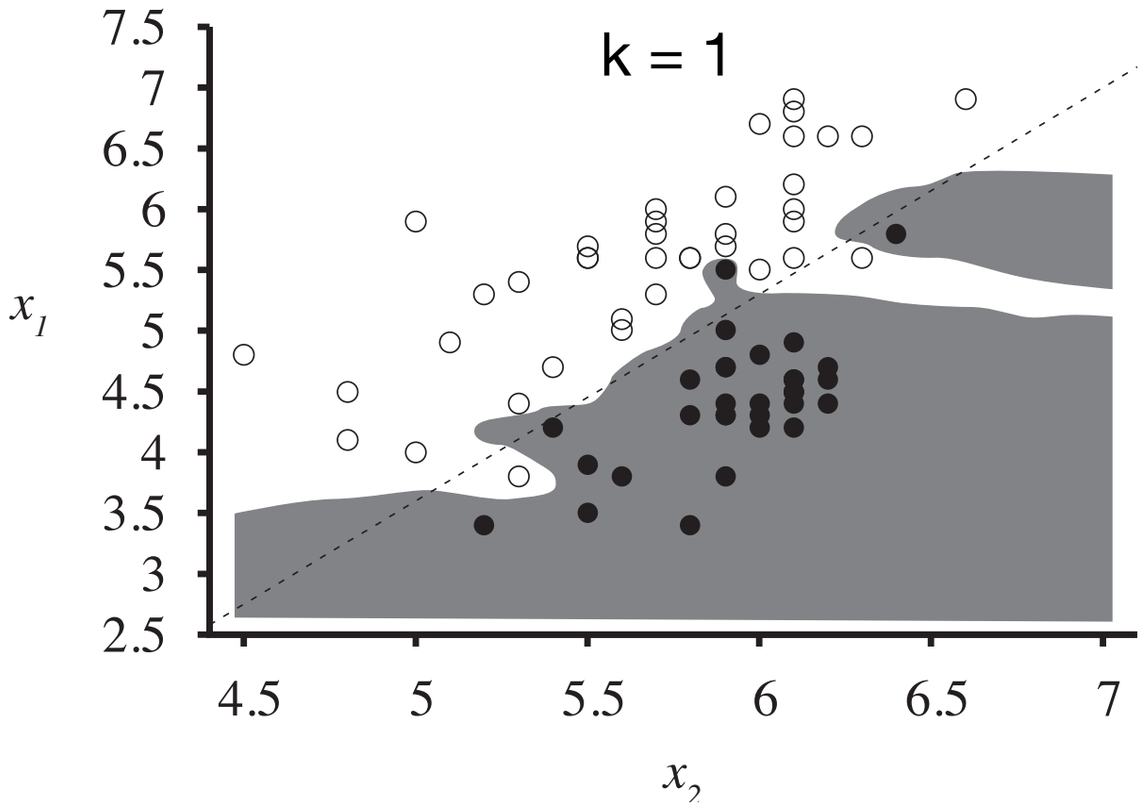
Are you a good Machine Learner?

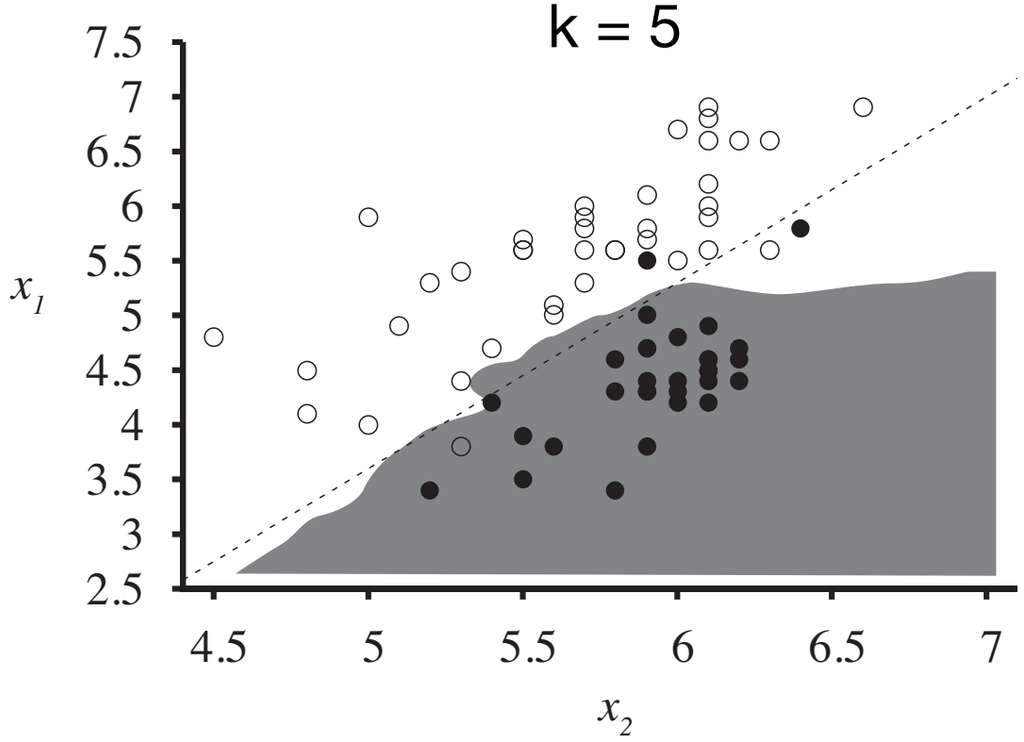
What did we learn?

- **Large amount of data** is required to make accurate predictions!
- Selection of the right **features and representation** is crucial!
- We can use **the nearest data points** to make predictions!

k-Nearest Neighbours

One way to implement a classifier is to consider the ***k* nearest neighbours** (in the feature space) of the current instance and assign the **class in the majority**.





The nearest neighbours of a query instance x_q are usually defined in terms of the Euclidean distance:

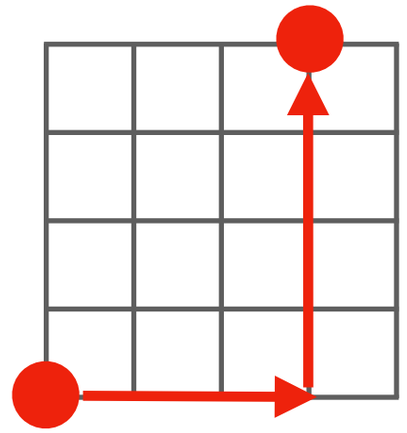
$$d(x_i, x_q) = \sqrt{\sum_g (a_g(x_i) - a_g(x_q))^2}$$

where the instances x_i belong to the dataset (previously observed data points) and all instances are described with a set of g = [1..p] features a_g

Different types of distance

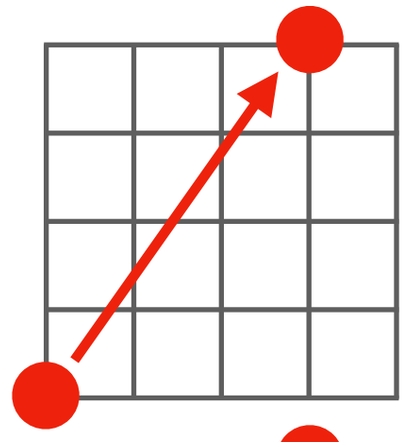
Manhanttan (L1-norm)

$$d(x_i, x_q) = \sum_g |a_g(x_i) - a_g(x_q)|$$



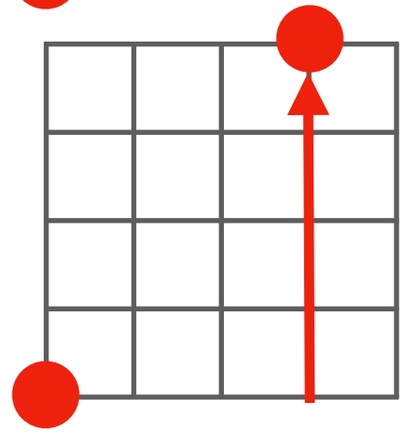
Euclidian (L2-norm)

$$d(x_i, x_q) = \sqrt{\sum_g (a_g(x_i) - a_g(x_q))^2}$$



Chebyshev (L-infinity-norm)

$$d(x_i, x_q) = \mathbf{max}_g |a_g(x_i) - a_g(x_q)|$$

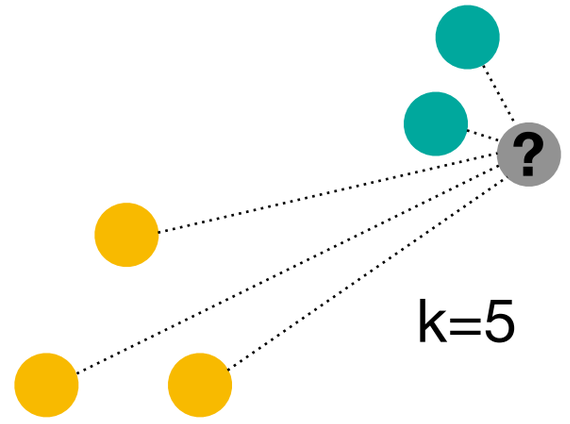


Many other distances exist.
For instance the Mahalanobis distance, to compute distance to a Gaussian distribution.
Or the Hamming distance to compute distance between binary (or symbolic) vectors.

k-Nearest Neighbours: Remarks

- Influence of k:
  - Small k => Good resolution of the borderline between the classes BUT highly sensitive to noise
  - Large k => Bad resolution of the borderline between the classes BUT quite robust to noise
- How to chose k? With a validation dataset (more in lecture 3)
- The k-NN algorithm is usually a quite powerful approach but: finding the k-NN might be slow if large dataset
  - Several approaches have been proposed to improve this:
    Using k-d Trees, Locality-sensitive hashing and hash tables, or generating prototypes with Learning Vector Quantisation.

Distance-weighted k-NN algorithm



A refinement of the *k*-NN algorithm: assign a weight w_r to each neighbour x_r of the query instance x_q based on the distance $d(x_r, x_q)$ such that $(d(x_r, x_q)\downarrow \leftrightarrow w_r\uparrow)$

Any measure favouring the votes of nearby neighbours will work

For instance:

$$w_r = \frac{1}{d(x_r, x_q)}$$

Inverse of the distance

$$w_r = \frac{1}{\sqrt{2 * \pi}} \exp(-d(x_r, x_q)^2/2)$$

Gaussian distribution

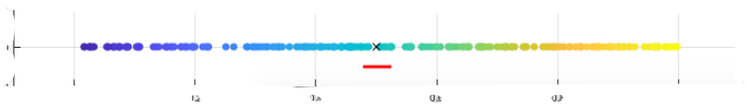
Distance-weighted *k*-NN algorithm: Remarks

- By the distance-weighted *k*-NN algorithm, the value of k is of minor importance as distant examples will have very small weight and will not greatly affect the classification.
- If k = n, where n is the total number of previously observed instances, we call the algorithm a global method. Otherwise, if $k < n$, the algorithm is called a local method.
- **Advantage** – Distance-weighted k-NN algorithm is robust to noisy training data: The classification is based on a weighted combination of all *k* nearest neighbours, effectively smoothing out the impact of isolated noisy training data.
- **Disadvantage** – All *k*-NN algorithms calculate the distance between instances based on all features → if there are many irrelevant features, instances that belong to the same class may still be distant from one another.
- **Remedy** – weight each feature differently when calculating the distance between two instances

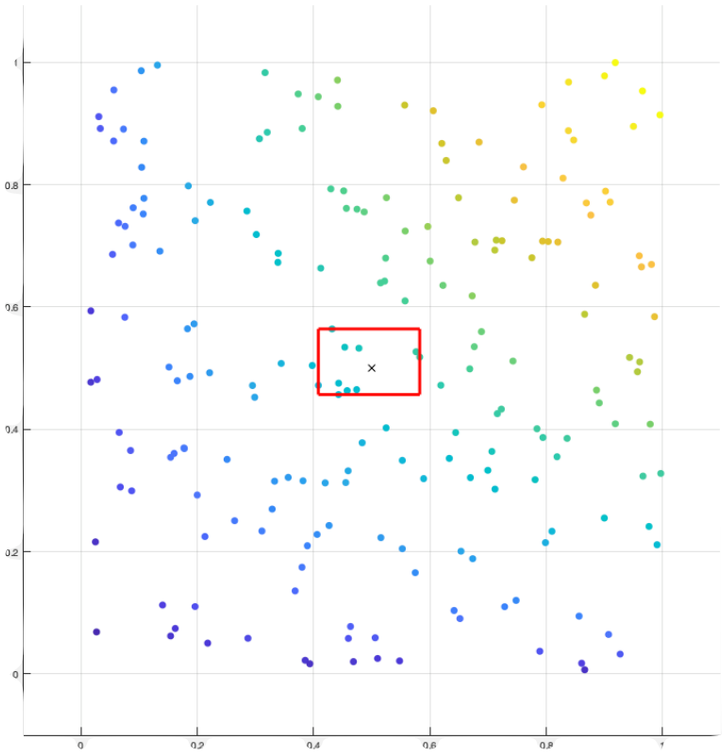
Curse of dimensionality

In high-dimensional feature spaces,
the nearest neighbours are usually not very near!

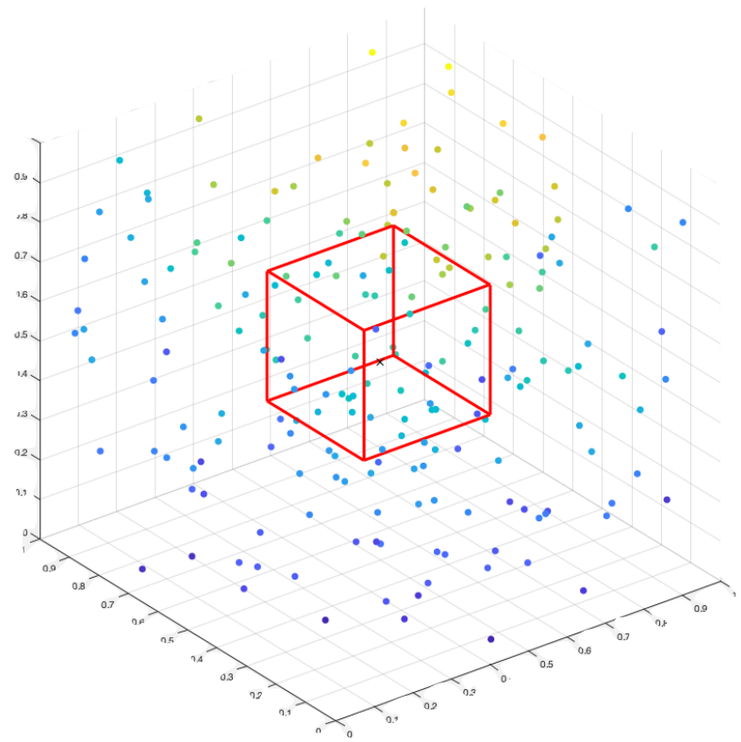
N=200 data points, k=10



Dim = 1
avg. length = 0.05

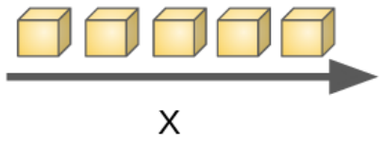


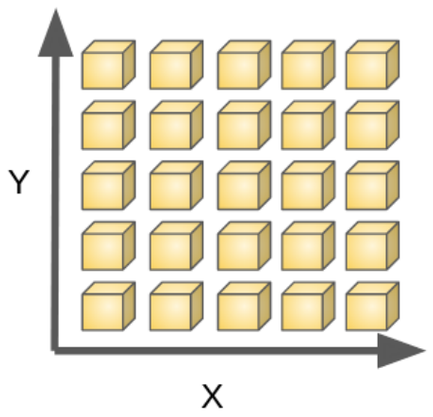
Dim = 2
avg. length = 0.14

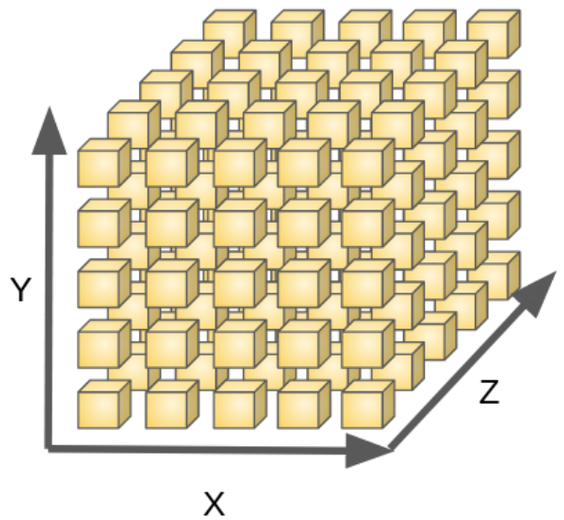


Dim = 3
avg. length = 0.33

Another way to visualise the curse of dimensionality is the number of points required to uniformly discretise the space with least n points on each dimensions.







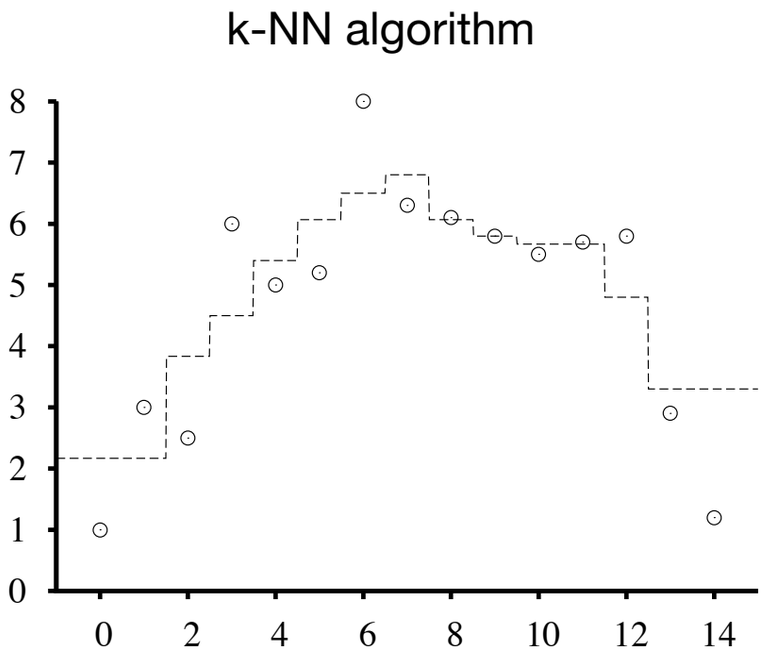
From https://medium.freecodecamp.org/

k-NN for regression

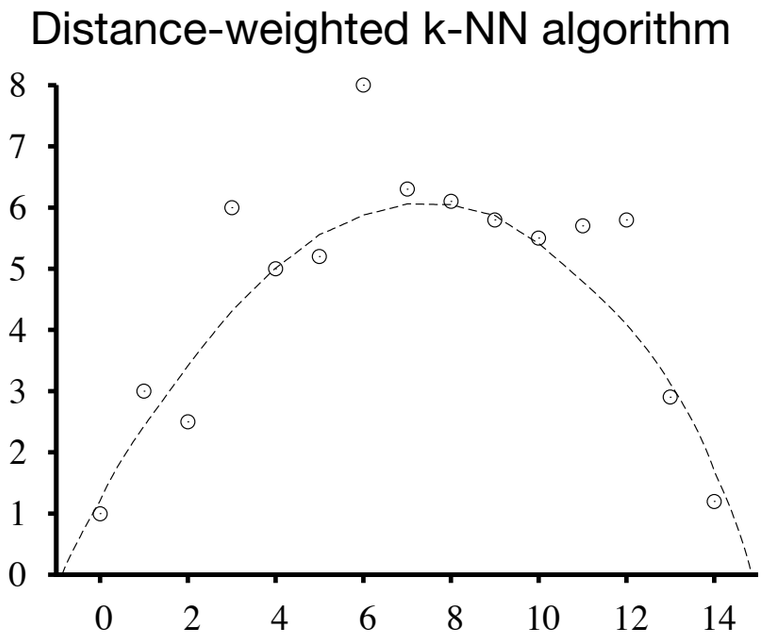
- Instead of using the class in the majority, we consider the **value in the majority**
- Distance-weighted *k*-NN algorithm can also be used for regression:

  The prediction of the algorithm is the weighted average value of the k nearest neighbours.

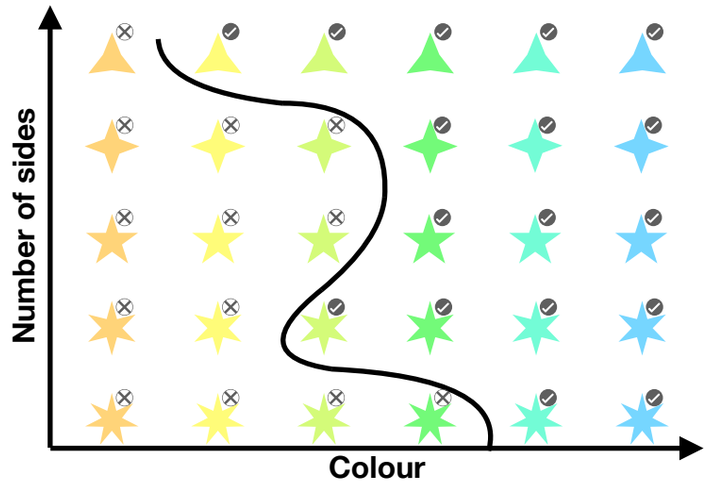
- This is also called Locally Weighted Regression



k=3

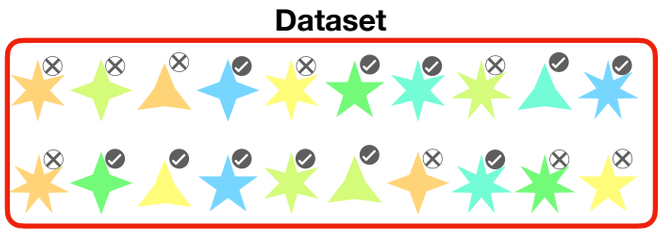


Lazy Learner
vs.
Eager Learner



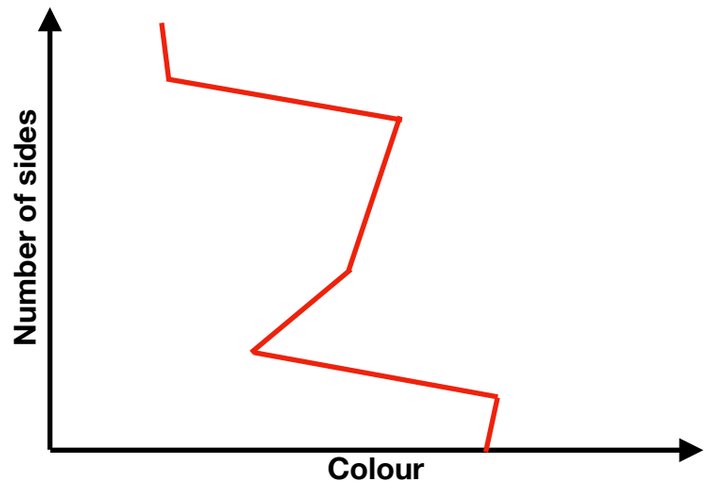
Definition: Lazy Learning

It stores the data and generalising beyond these data is **postponed** until an explicit request is made.



Definition: Eager Learning

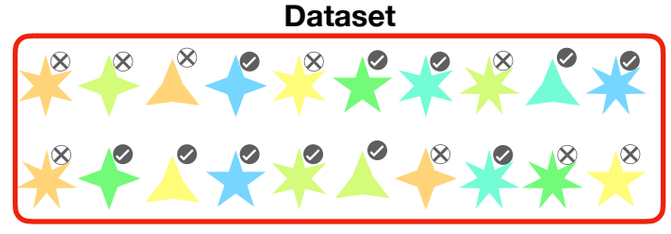
It constructs a general, explicit description of the target function based on the provided training examples

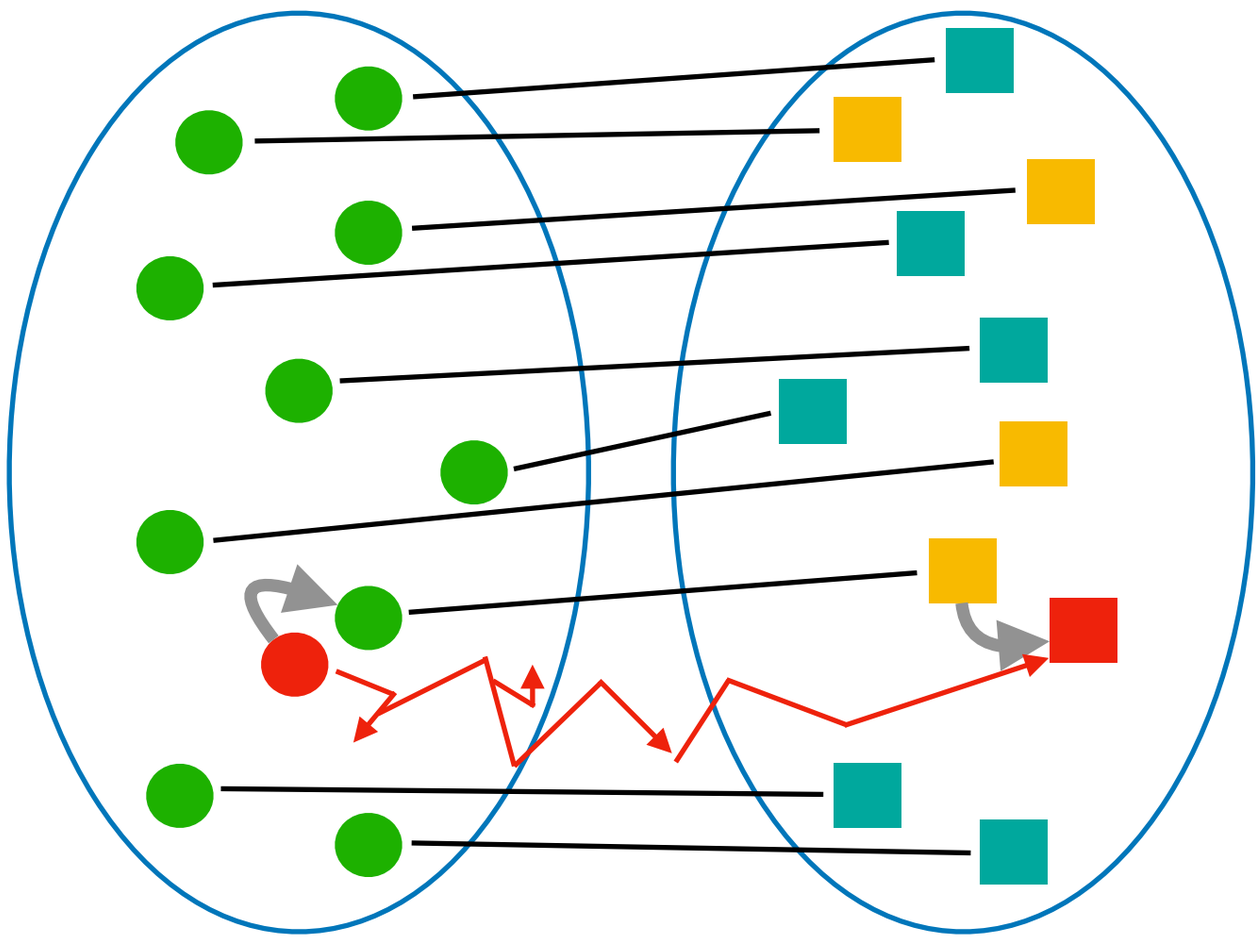


Lazy Learning

Definition: Lazy Learning

It stores the data and generalising beyond these data is **postponed** until an explicit request is made.





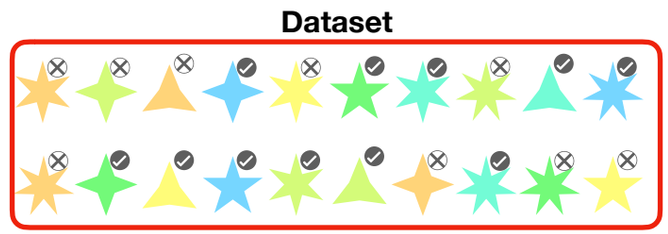
1. Search the memory for similar instances
2. Retrieve the related solutions
3. Adapt the solutions to the current instance
4. Assign the estimated solution to the current instance

- Example of eager learning: **k-NN**

Lazy Learning

Definition: Lazy Learning

It stores the data and generalising beyond these data is **postponed** until an explicit request is made.

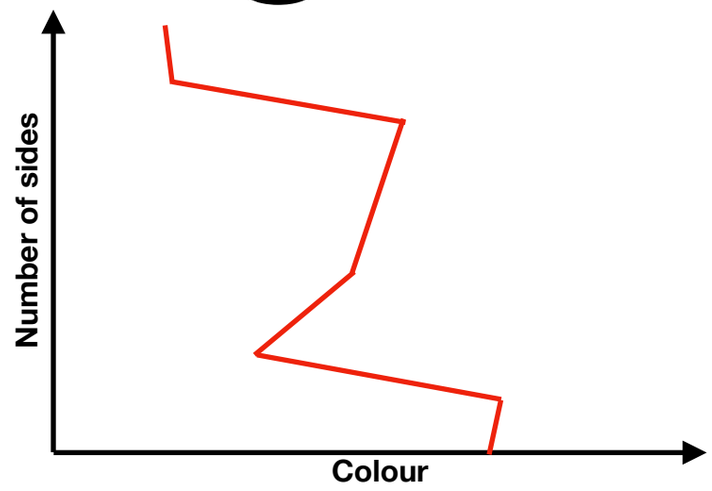


- It can construct a different approximation to the target function for each encountered query instance.
- It is very suitable for complex and incomplete problem domains, where a complex target function can be represented by a collection of less complex local approximations.
- The disadvantages with lazy learning include the large space requirement to store the entire training dataset. Usually long query time.
- Lazy learning is most useful for large datasets with few attributes.

Eager Learning

Definition: Eager Learning

It constructs a general, explicit description of the target function based on the provided training examples



- It uses the same approximation to the target function, which must be learned based on training examples and before input queries are observed.
- Better memory efficiency, and usually low query time.
- It usually deals better with noisy training datasets.
- The main disadvantage with eager learning is that it is generally unable to provide good local approximations in the target function.

- Example of eager learning: Artificial neural networks or decision trees (next week).

ML activities at imperial

Imperial College Machine Learning Initiative

http://www.imperial.ac.uk/machine-learning/

Machine Learning (ML) Tutorials

Web: http://wp.doc.ic.ac.uk/sml/teaching/ml-tutorials/

Target audience: PhD students, post-docs and academics with an interest in machine learning. Gives final-year and MSc students an outlook beyond the content level of the course at what is going on in ML.

Machine Learning Talks Mailing List

If you are interested in receiving emails regarding machine learning talks and seminars at or around Imperial College, please sign up here: https://mailman.ic.ac.uk/mailman/listinfo/ml-talks
Target audience: anybody interested in machine learning