Machine Learning

Rudolf Mayer

mayer@ifs.tuwien.ac.at

October 9th, 2019

Outline

- Organisational Questions?
- Introduction
- Perceptron
- k-NN
- Types of data & data preparation (intro)
- Performance evaluation (intro)

Course support

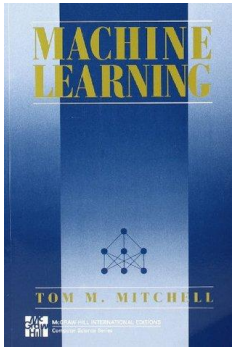
- Everyone registered in TISS ? (http://tiss.tuwien.ac.at)
- Register for the course in TUWEL to access material
  - https://tuwel.tuwien.ac.at/course/view.php?idnumber=184702-2019W
    - linked from TISS (automatic access when applied to lecture in TISS)

- TUWEL used for
  - Course materials (lecture slides, further readings)
  - Lab assignments
  - Forum
    - TISS forum etc. will **NOT** be used !!

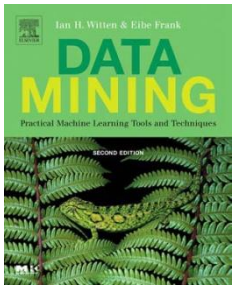
- Evaluation
  - 3 (+1) exercises
  - Written exam

Literature

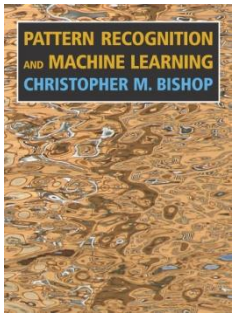
- Books
  - Tom Mitchell, “Machine Learning”



  - Ian Witten, “Data Mining: Practical Machine Learning Tools and Techniques” (WEKA Authors)



  - Christopher Bishop, “Pattern Recognition and Machine Learning”

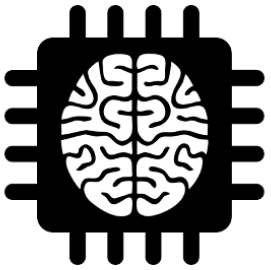


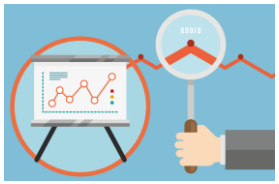
Outline

- Introduction
- Perceptron
- k-NN
- Types of data & data preparation (intro)
- Performance evaluation (intro)

Defining the area

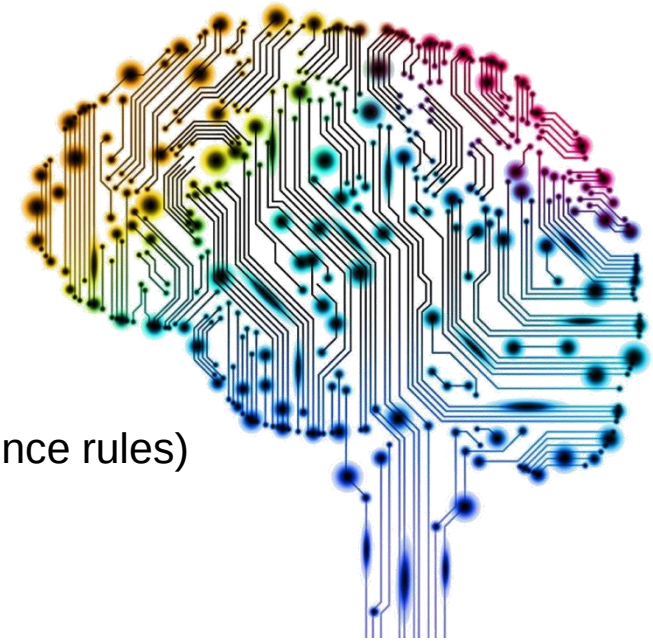
- Some related terms
  - Data Mining
  - Predictive analytics
  - Knowledge discovery
  - Data Science
  - Statistics
  - Cluster Analysis
  - Artificial Intelligence
  - Reasoning / deduction
  - Regression
  - Classification
  - Supervised/ unsupervised learning
  - ...





Artificial intelligence

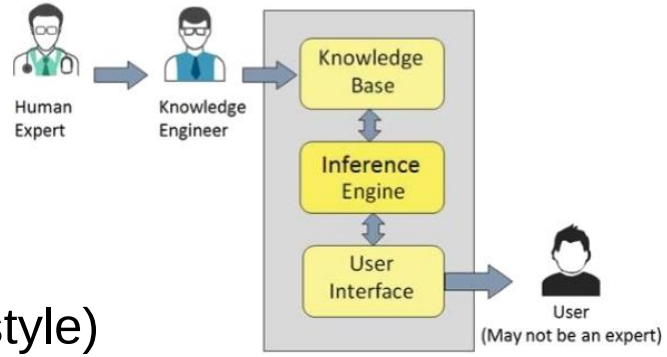
- Deals with intelligence of machines
  - *System that perceives its environment & takes actions that maximise its chances of success*



- Areas/problems of AI
  - Deduction, reasoning, problem solving
    - Often rule-based; manually created
  - Knowledge representation (reasoning)
    - Reasoning: generation of new knowledge (inference rules)
  - ***Machine learning***
  - Planning / scheduling
  - Natural language processing
  - …

Rule-based systems

- System that uses rules to make deductions or choices
  - E.g. domain-specific expert system



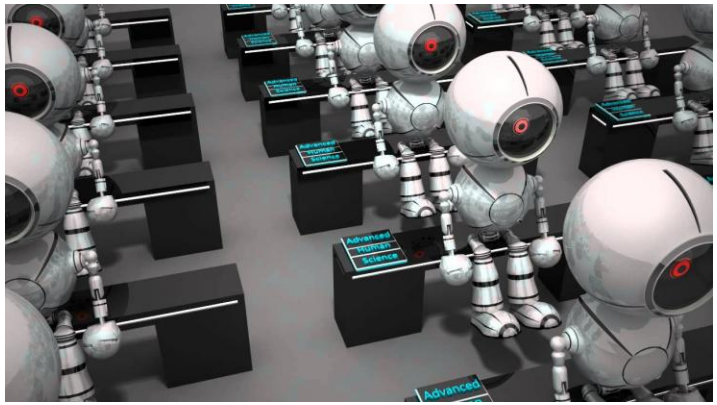
- Two components
  - Knowledge base: facts & rules (if → then style)
    - Knowledge representation (language)
    - Example rule: *IF (hot & smoke) THEN fire*
  - Inference engine: applies rules to deduce new facts
    - Forward chaining: assert new facts
    - Backward chaining: start with goal → determine which facts need to asserted
- Cf Ontologies & reasoning
- Rules often manually specified (by expert)
  - Expensive, incomplete
  - ***Does not scale well*** → enter Machine Learning?

Machine learning

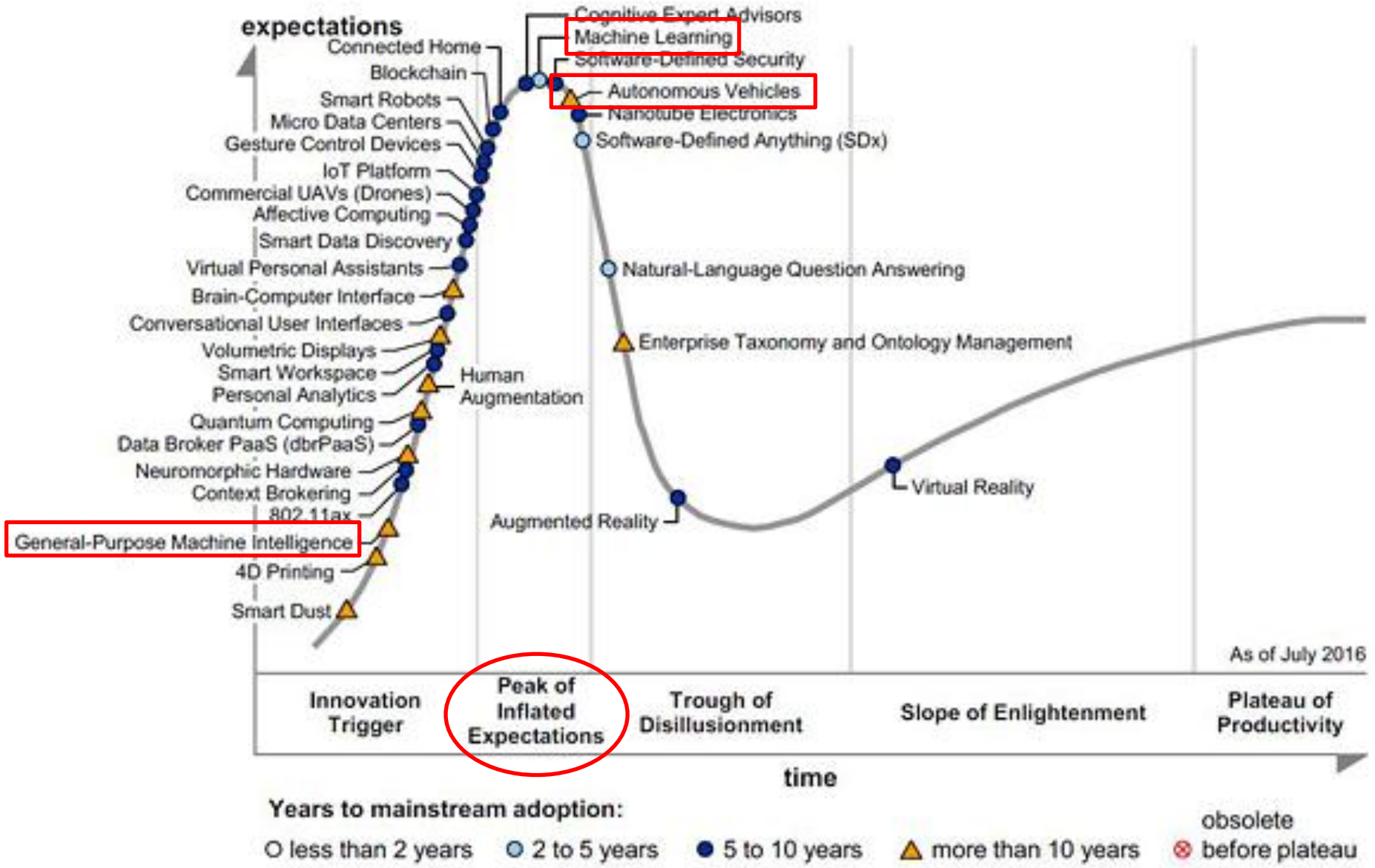
- Studies computer algorithms that can ***learn*** from data and make ***predictions*** on data

- Automatic methods – no human assistance in learning (i.e. no specification of rules!)

  – Human assistance generally required for
    - Defining problem
    - Gathering and assessing data

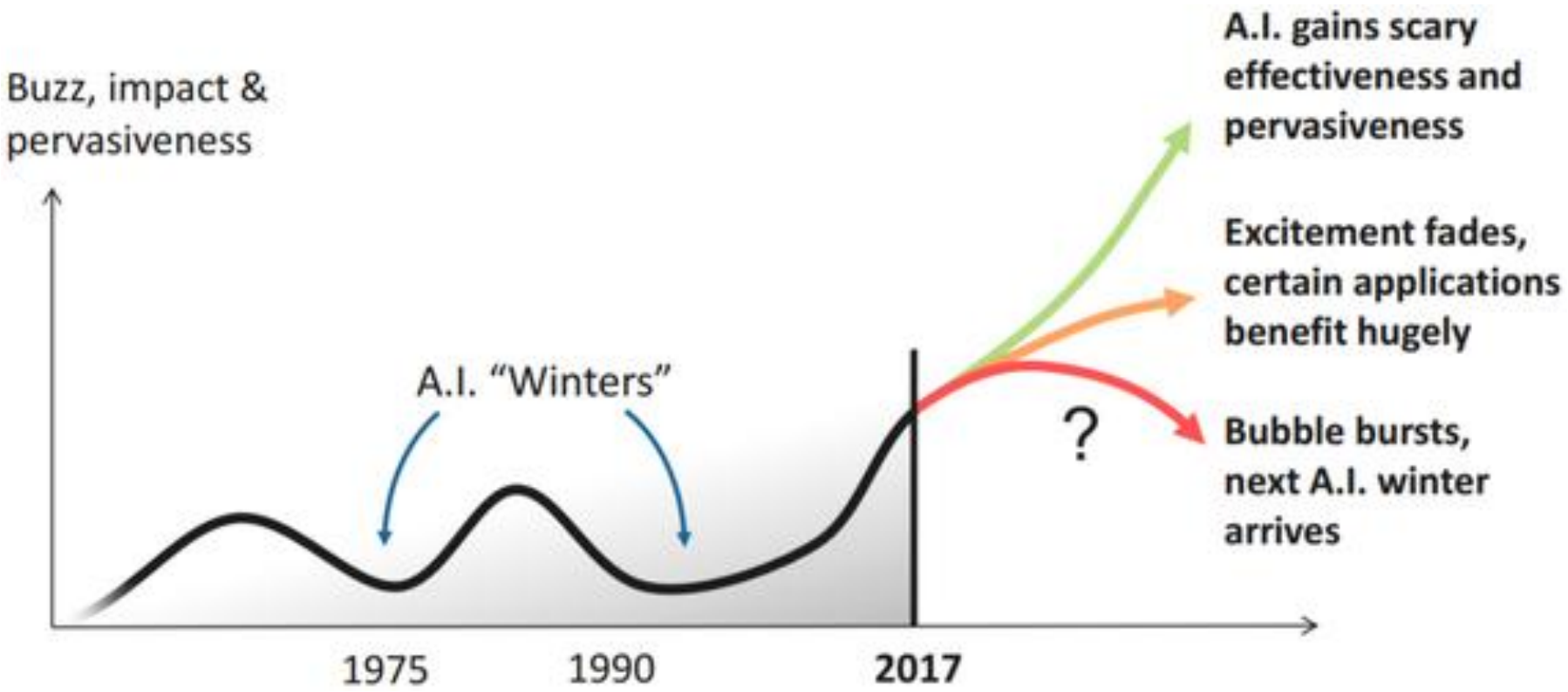


Machine learning – why?



Source: Gartner (July 2016)

Machine learning – hype?



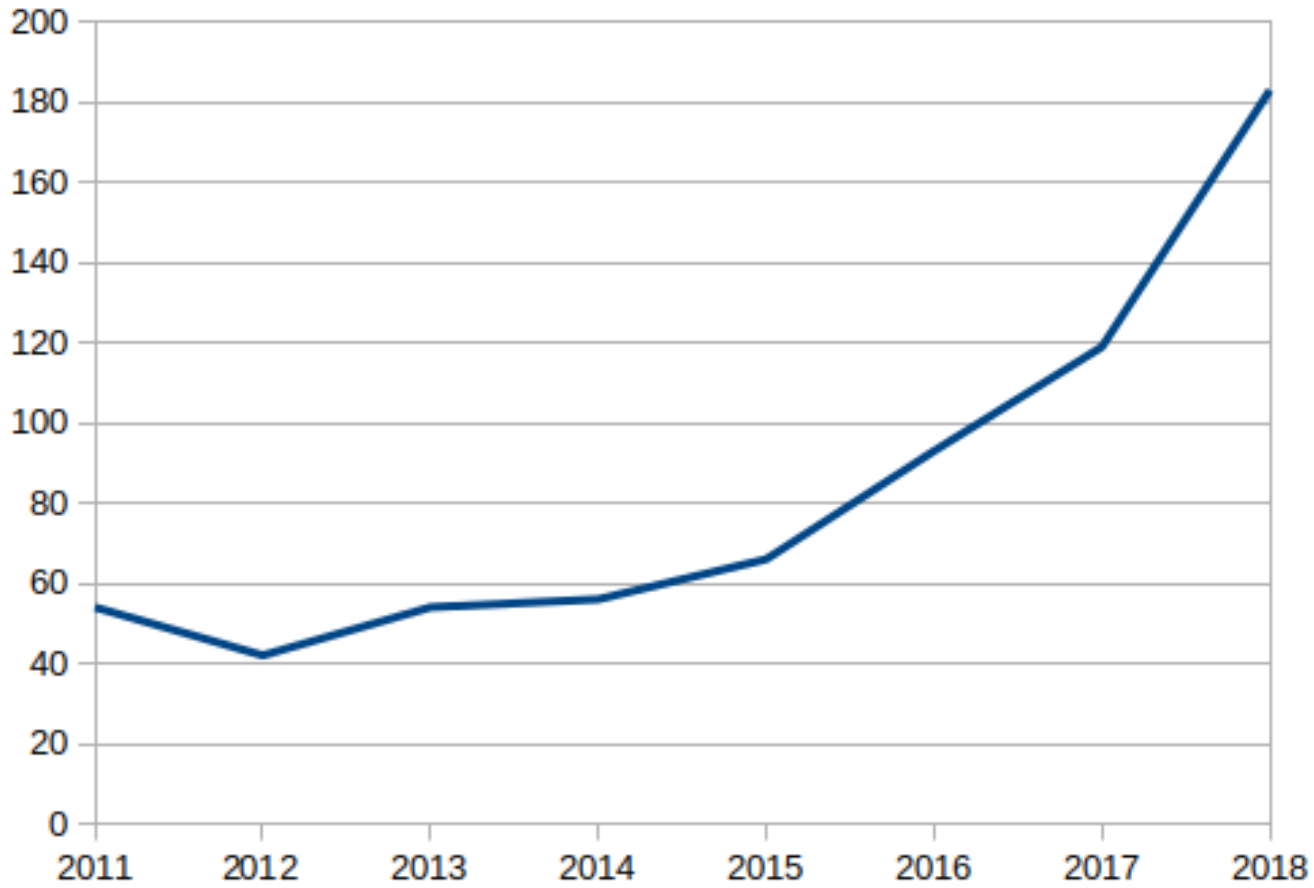
Machine learning – Hype

Machine Learning
The next revolution or just another hype?

- A little bit of both ...

Machine learning – really popular?

- Number of students enrolled for the ML course @ TU Wien



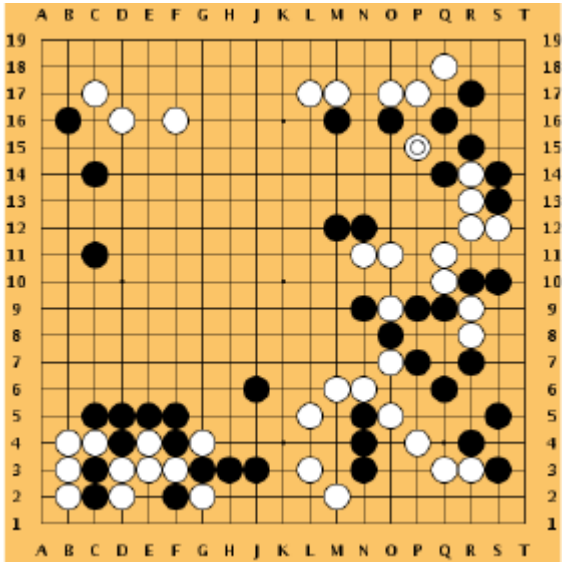
ML success stories: skin cancer diagnosis

- Deep Learning used to diagnose skin cancer from images
- Learns from examples
- Matched performance of certified dermatologists

Nature 542, 115–118 (02 February 2017), doi:10.1038/nature21056

ML success stories: AlphaGo

- DeepMind's AlphaGo beats Lee Sedol, one of the best Go players (March 2016)
  - Uses combination of Monte Carlo Simulation & Deep Learning



ML success stories: AlphaGo Zero

- AlphaGo Zero: not using any human generated *example* game data (October 2017)
  - Learns by playing "against itself" (from successful games)
  - Reaches level of 2016 AlphaGo (Lee Sedol) in 3 days
  - Implemented on very special hardware
    - Google's TPU (Tensor Processing Unit)

science ORF.at

Fernsehen TVthek Radio Debatte Österreich Wetter IPTV Sport Archiv

Go-Champion ohne menschliches Zutun

Das Aufsehen war groß, als AlphaGo 2016 der Sedol im Brettspiel Go besiegte. Forscher, dar österreichischer Informatiker, haben die Softv Ohne menschliches Zutun lernte sie das Spiel Vorgängerversion.

AlphaGo hat sich heuer auch mit einem glatten vermutlich weltbesten Go-Spieler Ke Jie durch Erfolgen wurde ersichtlich, wie weit die Entwic Künstlicher Intelligenz (KI) bereits fortgeschritt stammt von der britischen Firma DeepMind, di Jahren vom US-Konzern Google übernommen wurde.

Die Studie

„Mastering the game of Go without human knowledge",
Nature, 18.10.2017

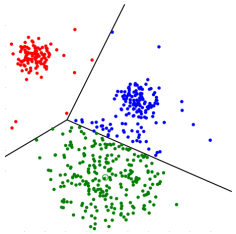
Besonders überraschend waren die Fähigkeiten von AlphaGo, weil das rund 3.000 Jahre alte asiatisch [illegible] dem menschlichen Geist exklusiv zugebilligt [illegible] Möglichkeiten für Züge, ist nämlich viel Intuition, kreatives Denken und Lernfähigkeit gefragt.

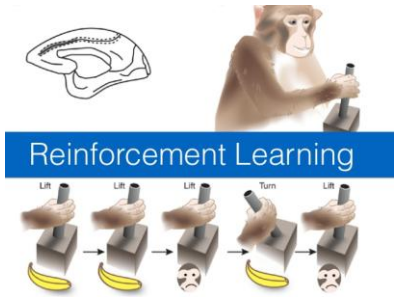
FACULTY OF INFORMATICS

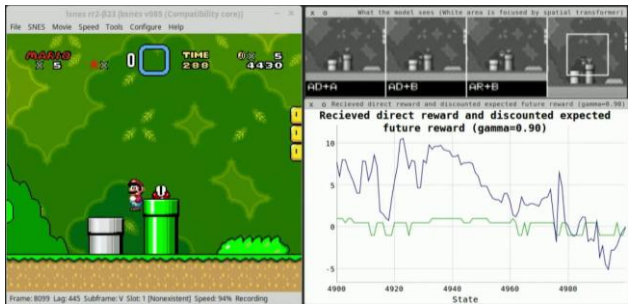
ML success stories: image manipulation

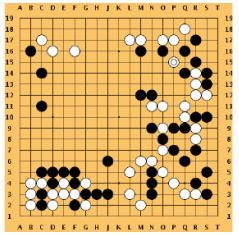
Machine learning: disciplines

- ML can take several forms, depending on the data and the task
- Common sub-disciplines
  - Unsupervised Learning
    - Clustering, collaborative filtering, outlier detection, ...
  - Supervised Learning
    - Classification, Regression, ...
  - Reinforcement Learning



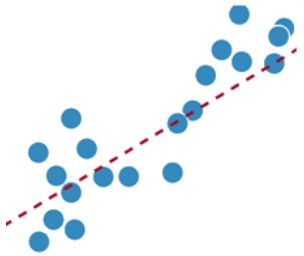




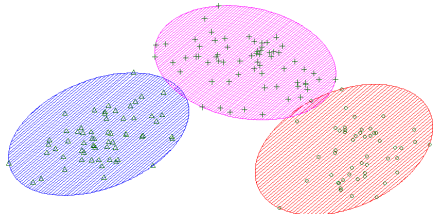


Defining the area

- Statistics
  - Collection, interpretation and presentation of data
  - Descriptive: summarise data (mean, standard deviation, probability density, correlation analysis)
  - Predictive modelling: e.g. regression

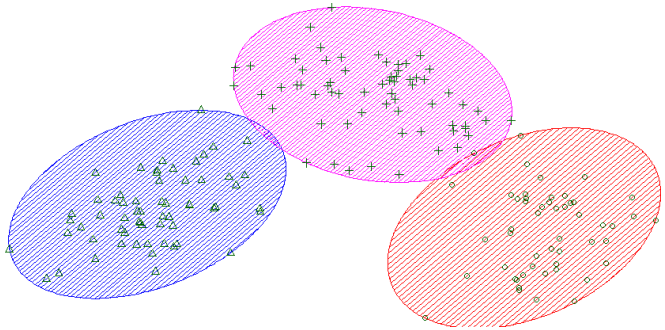


- Data Mining
  - Discovering patterns in large datasets
    - Cluster analysis, outlier detection, association rule mining



Unsupervised Learning

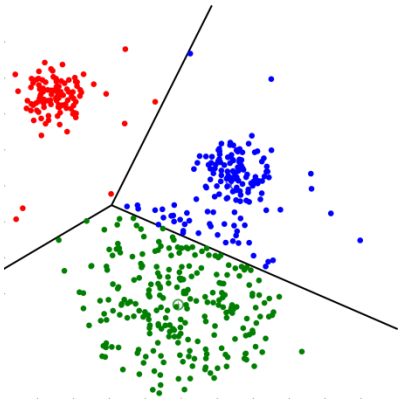
- Unsupervised learning
  - Data not ***labelled***
  - No information on which and how many classes or other structures, …
  - Goal:
    - Find structures (e.g. clustering)
    - Association rules learning
    - Find outliers (anomalies)

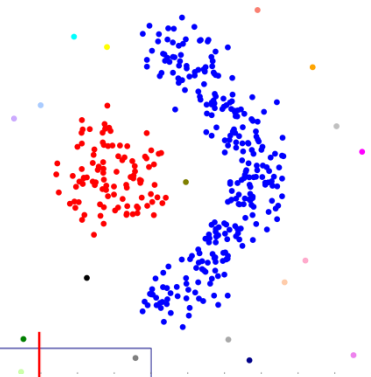


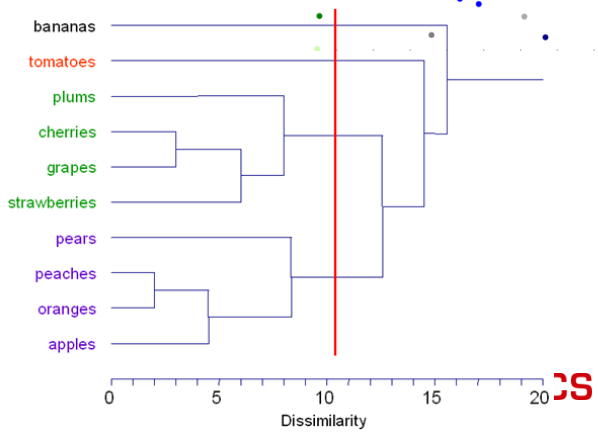
  - Most often associated with ***Data Mining***
- Covered in other lectures
  - Business Intelligence, Selbstorganisierende Systeme, ..

Cluster analysis

- Explorative method
- Find groups of similar objects
- Applications: e.g. market segmentation
- Several popular algorithms
  - K-means clustering (centroid based)
  - DBSCAN (density-based)
  - Linkage (hierarchical)
    - single, complete, average, Ward, ..







Cluster analysis

- Animal data set
  - Describes animal by some characteristics
  - Instances: cow, duck, cat, bee, sparrow, ...
  - Characteristics
    - Size (tiny, small, medium, big)
    - Number of legs (2, 4, 6, 8)
    - Feathers (yes/no)
    - Eggs (yes/no)

	size	*legs*	*feathers*	*eggs*
duck	small	2	yes	yes
dog	medium	4	no	no
spider	tiny	8	no	yes
ladybird	tiny	6	no	yes
cow	large	4	no	no
bee	tiny	6	no	no
sparrow	small	2	yes	yes

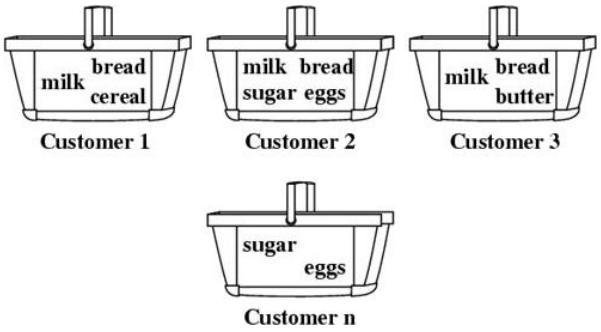
- Goal: find groups of related animals:
  - Mammals: cat, cow
  - Birds: duck, sparrow
  - Insects: bee, ladybird
  - Invertebrate: spider

Association rule mining

- Discover relations between variables
  - E.g. market basket analysis: which items are frequently bought together

➔ Useful for marketing

  - Identify rules that are
    - Frequent (“support”)
    - Reliable (“confidence”)



  - Anecdotal example: beer & diapers

  - Related: collaborative filtering / recommender systems

amazon NETFLIX

Supervised Learning

- Supervised learning
  - Data labelled with actual ***output*** variable
  - Goal: correctly label ***unknown*** data

  - Sometimes equivalently used with “machine learning”
    - In some definitions, machine learning means both unsupervised and supervised learning
    - In other definitions, unsupervised learning mostly equals/is a subset of data mining; and rather seen as part of statistics

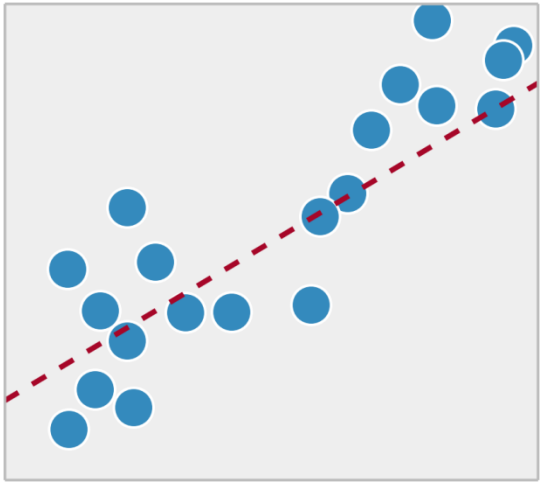
- *Types of supervised learning?*
  - Regression & Classification

Regression vs. Classification

- Regression tries to predict a ***continuous*** variable
  - Examples?

Regression vs. Classification

- Regression tries to predict a ***continuous*** variable
  - *Income of a house hold* (depending on education, ...)
  - *Temperature* (depending on wind, humidity...)
  - *Housing price* (depending on e.g. size, age..)
  - Methods: e.g. *linear regression*
    - Mostly associated with statistics
  - Many classification methods can also be used for regression

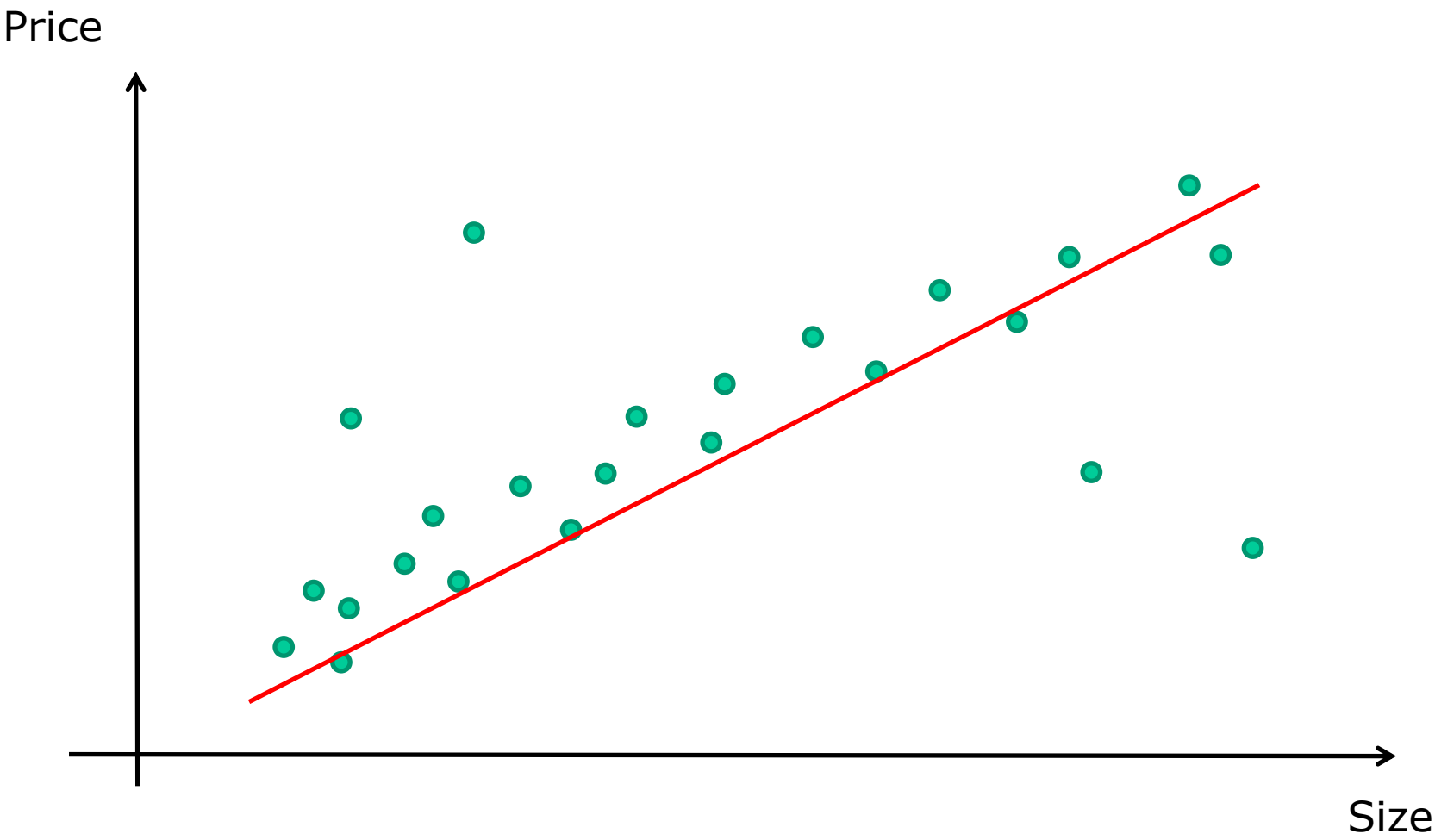


Regression example

- Data set about prices of houses (apartments)
- Input data e.g.
  - Size
  - Number of rooms
  - Size of garden/balcony/terrace
  - Year built / age of building
  - Location
- Goal: predict the expected price of the house in €
- Solution: multivariate regression
  - *If only one input: univariate regression*

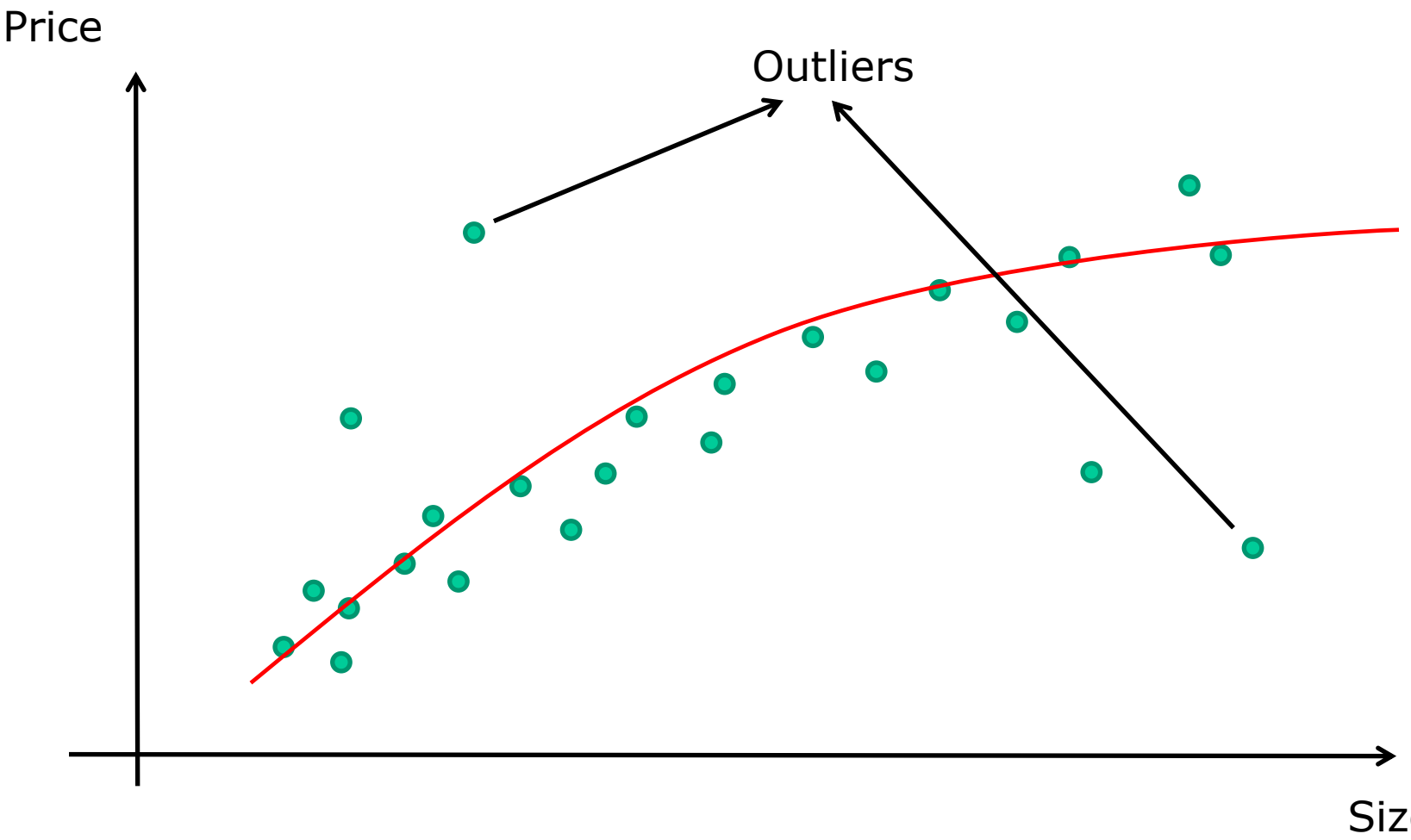
Regression example

Linear regression – find a "line" that describes the data



Regression example

Polynomial regression – find a "curve" that describes the data

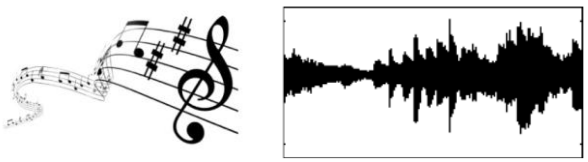


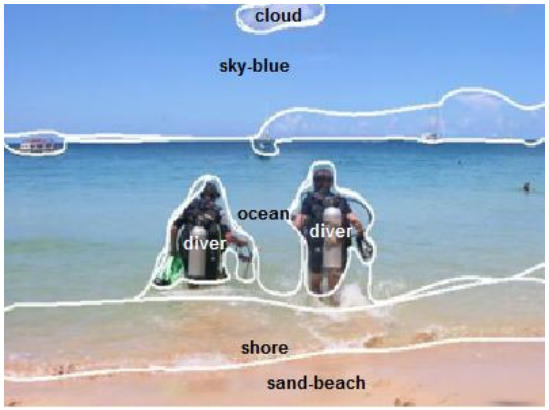
Regression vs. Classification

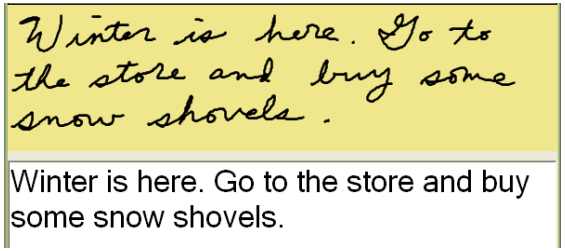
- If the output variable can take one of a predefined set of values: ***classification*** *(also: categorisation)*
  - Examples?

Regression vs. Classification

- If the output variable can take one of a predefined set of values: ***classification*** *(also: categorisation)*





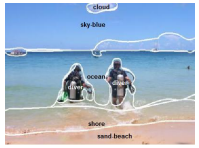


Regression vs. Classification

- If the output variable can take one of a predefined set of values: ***classification*** *(also: categorisation)*

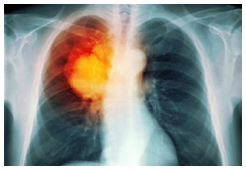
  - ***Information***: SPAM filtering

  - ***Image***: classification of hand-written letters for OCR; automatic labelling of images



  - ***Music***: classification of music into genres

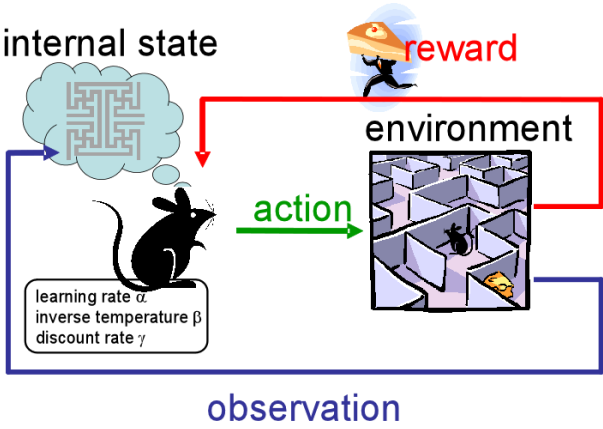
  - ***Medicine***: classification of whether a person has an illness, based e.g. on x-ray images, or other measurements



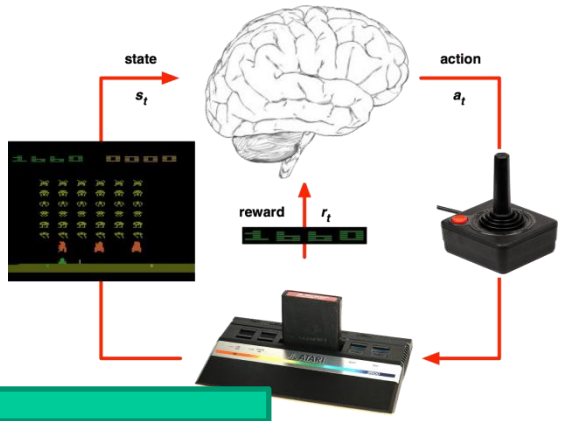
Reinforcement Learning

- Agent (e.g. computer program) takes actions in specified environment
  - Not explicitly presenting input/output pair
  - Reward (penalise) agent for actions

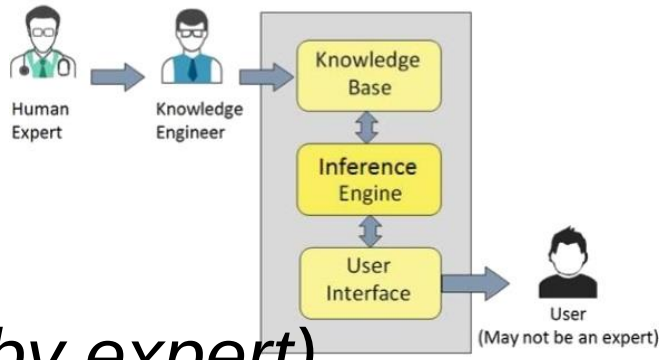
➔ Maximise cumulative reward



- Popular applications?

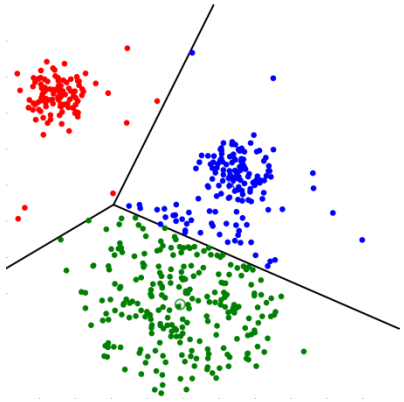


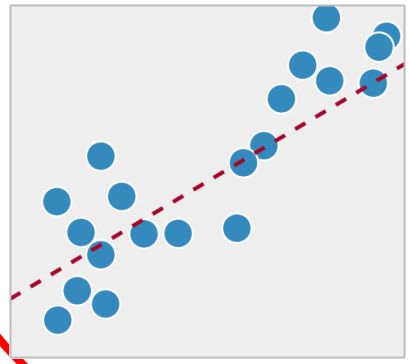
Rule-based systems

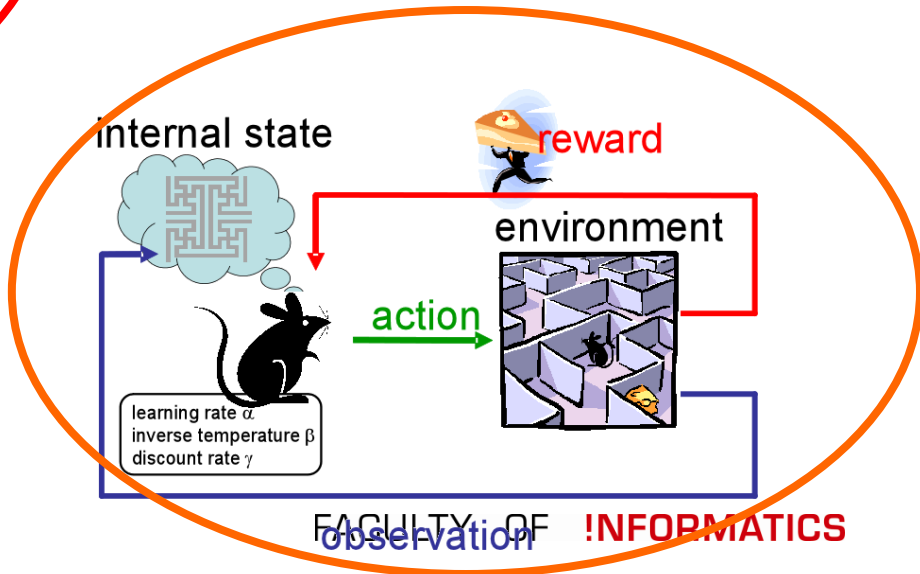


- *Rules often manually specified (by expert)*
- Alternative: learning rules
  - *Association rule mining – unsupervised*
  - Learning classifier systems
    - Can be supervised learning
    - Includes steps to discover new rules, selection of best rules, ...
- Related/similar approaches:
  - *Bayesian Networks*

Scoping the lecture







Classification: Setting

- Example: data set describing characteristics of patients
  - Gender
  - Age
  - Smoker yes/no
  - Eye colour

- Want to predict whether a person has lung cancer
  - Available: some data with a label / annotation (cancer yes(no)

Classification: Setting

Attributes / Variables / Features

Output Variable

Training Samples

gender	age	smoker	eye colour	lung cancer
male	19	yes	green	No
female	44	yes	grey	yes
male	49	yes	blue	yes
male	12	no	brown	no
female	37	no	brown	no
female	60	no	brown	yes
male	44	no	blue	no
female	27	yes	brown	no
female	51	yes	green	yes
female	81	yes	grey	no
male	22	yes	brown	no
male	29	no	blue	no

Ground Truth / Gold standard

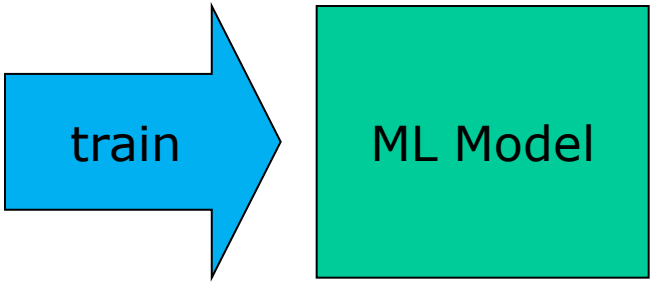
male	77	yes	grey	?
male	19	yes	green	?
female	44	no	grey	?

Prediction

Unlabelled samples

Classification: Setting

gender	age	smoker	eye colour	lung cancer
male	19	yes	green	No
female	44	yes	grey	yes
male	49	yes	blue	yes
male	12	no	brown	no
female	37	no	brown	no
female	60	no	brown	yes
male	44	no	blue	no
female	27	yes	brown	no
female	51	yes	green	yes
female	81	yes	grey	no
male	22	yes	brown	no
male	29	no	blue	no

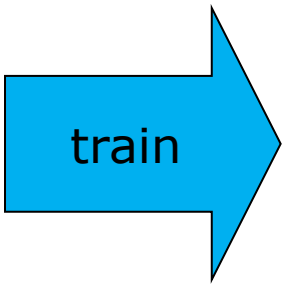


male	77	yes	grey	?
male	19	yes	green	?
female	44	no	grey	?

Classification: Setting

gender	age	smoker	eye colour
male	19	yes	green
female	44	yes	grey
male	49	yes	blue
male	12	no	brown
female	37	no	brown
female	60	no	brown
male	44	no	blue
female	27	yes	brown
female	51	yes	green
female	81	yes	grey
male	22	yes	brown
male	29	no	blue

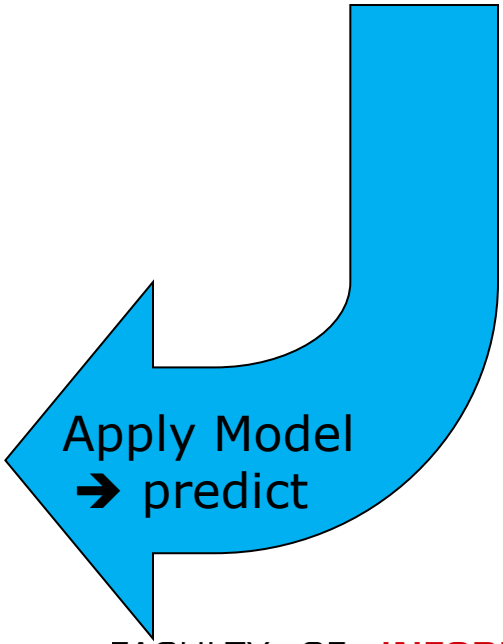
lung cancer
No
yes
yes
no
no
yes
no
no
yes
no
no
no



ML Model
(e.g. Neural Network)

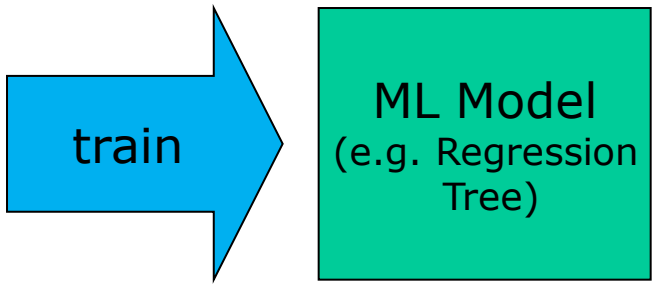
male	77	yes	grey
male	19	yes	green
female	44	no	grey

yes
no
yes

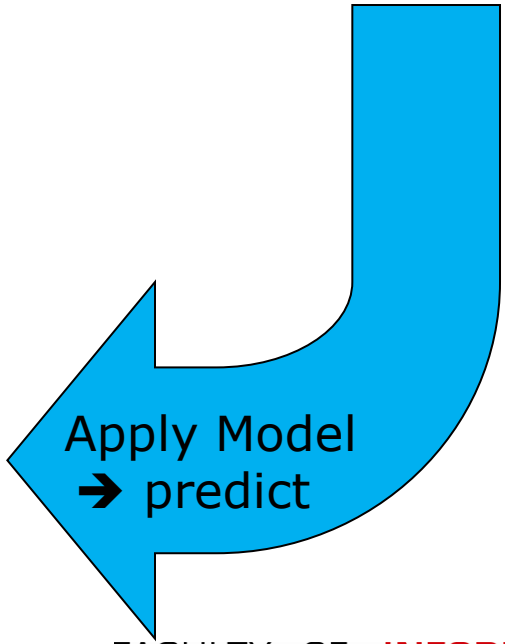


Regression: Setting

size	age	location	rooms	price
70	19	good	3	370000
60	44	good	2	165000
120	49	good	5	575000
90	12	bad	3	225000
80	37	bad	3	268000
45	60	bad	2	159000
52	44	bad	2	185000
85	27	good	4	472000
50	51	good	1	164000
35	81	good	1	155000
75	22	good	3	391000
95	29	bad	4	478000



35	77	good	4	274000
25	19	good	1	122000
85	44	bad	2	162000



Machine Learning: Setting

gender	age	smoker	eye colour	lung cancer
male	19	yes	green	No
female	44	yes	grey	yes
male	49	yes	blue	yes
male	12	no	brown	no
female	37	no	brown	no
female	60	no	brown	yes
male	44	no	blue	no
female	27	yes	brown	no
female	51	yes	green	yes
female	81	yes	grey	no
male	22	yes	brown	no
male	29	no	blue	no

male	77	yes	grey	?
male	19	yes	green	?
female	44	no	grey	?

- *Where do we get the data from?*

Machine Learning: Setting

gender	age	smoker	eye colour
male	19	yes	green
female	44	yes	grey
male	49	yes	blue
male	12	no	brown
female	37	no	brown
female	60	no	brown
male	44	no	blue
female	27	yes	brown
female	51	yes	green
female	81	yes	grey
male	22	yes	brown
male	29	no	blue

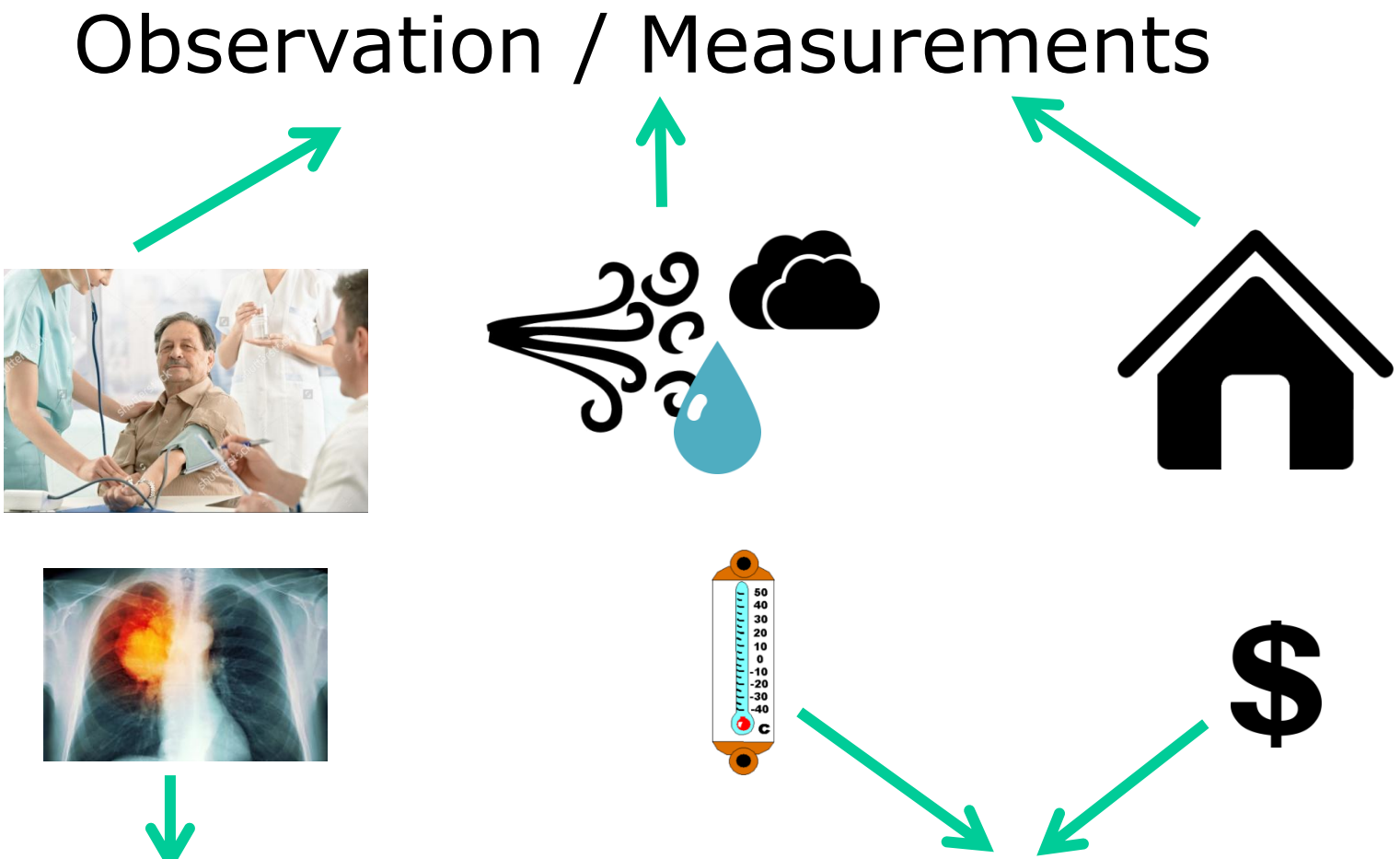
lung cancer
No
yes
yes
no
no
yes
no
no
yes
no
no
no

male	77	yes	grey
male	19	yes	green
female	44	no	grey

?
?
?

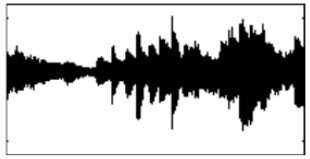
- Features:
  - Observation / measurement
  - *Extraction from media*
  - *(Transformation)*
- Labels (groundtruth):
  - Human experts
  - (Observation / measurement)

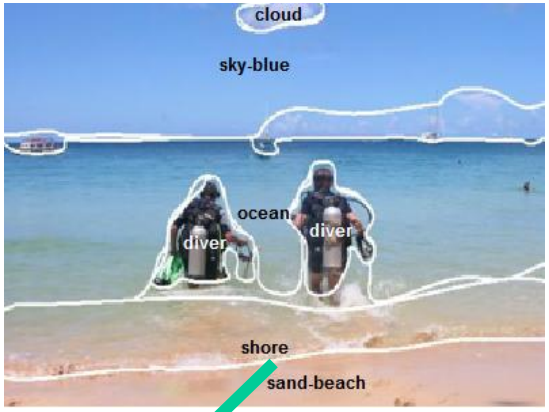
Machine Learning: Setting



Machine Learning: Setting

Feature Extraction





Human experts

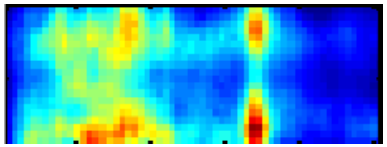
Machine Learning: Setting

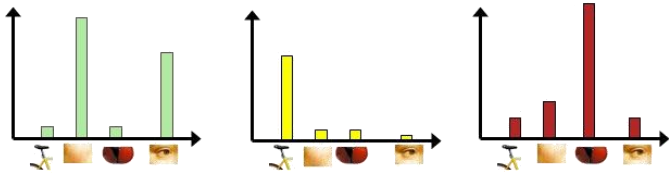
- Feature Extraction: description of complex content by derived, ***numeric*** values
  - Text: Bag of Words – counting term occurrences
  - Music: counting activity on various frequencies
  - Image: colour histograms, edge detection, “bag of visual words”, ...

- Important aspect of many data science applications
  - Overview/intro in this course
  - *More details e.g. Information retrieval (188.412), ...*

Machine Learning: Setting

	Word 1	Word 2	Word 3	Word 4	Word 5	...	Word n	∑
Doc 1	1	0	0	2	3		0	6
Doc 2	2	0	0	0	2		2	6
Doc 3	1	3	0	0	1		5	10
Doc 4	2	0	0	2	0		2	6
Doc 5	5	4	0	0	1		0	10
...							0	
Doc m	1	2	1	3	0		0	7

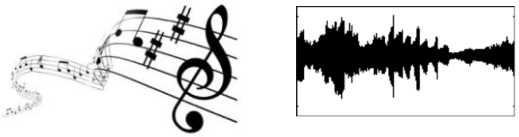


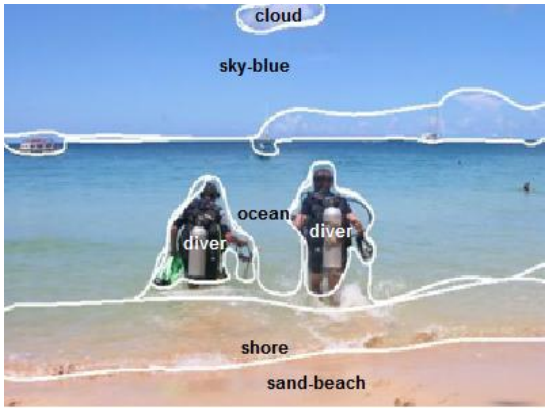


Text Feature Extraction

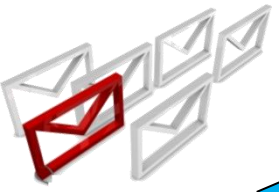
Music Feature Extraction

Image Feature Extraction





Machine Learning: steps



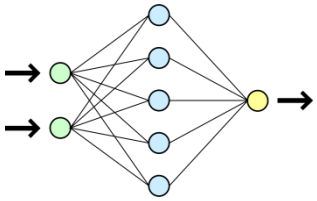
Ground Truth

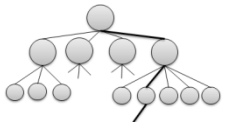
label

Data

extract/ transform

Features



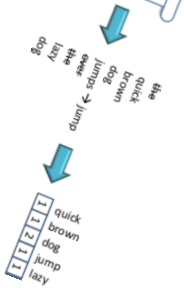


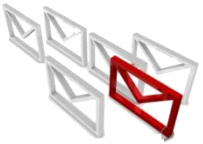
train

ML Model

Features

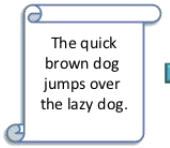
extract/ transform

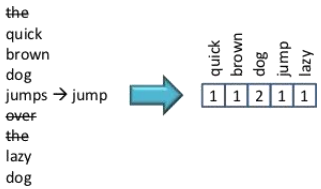




Unseen Data

apply





Prediction

Continuous (regression)

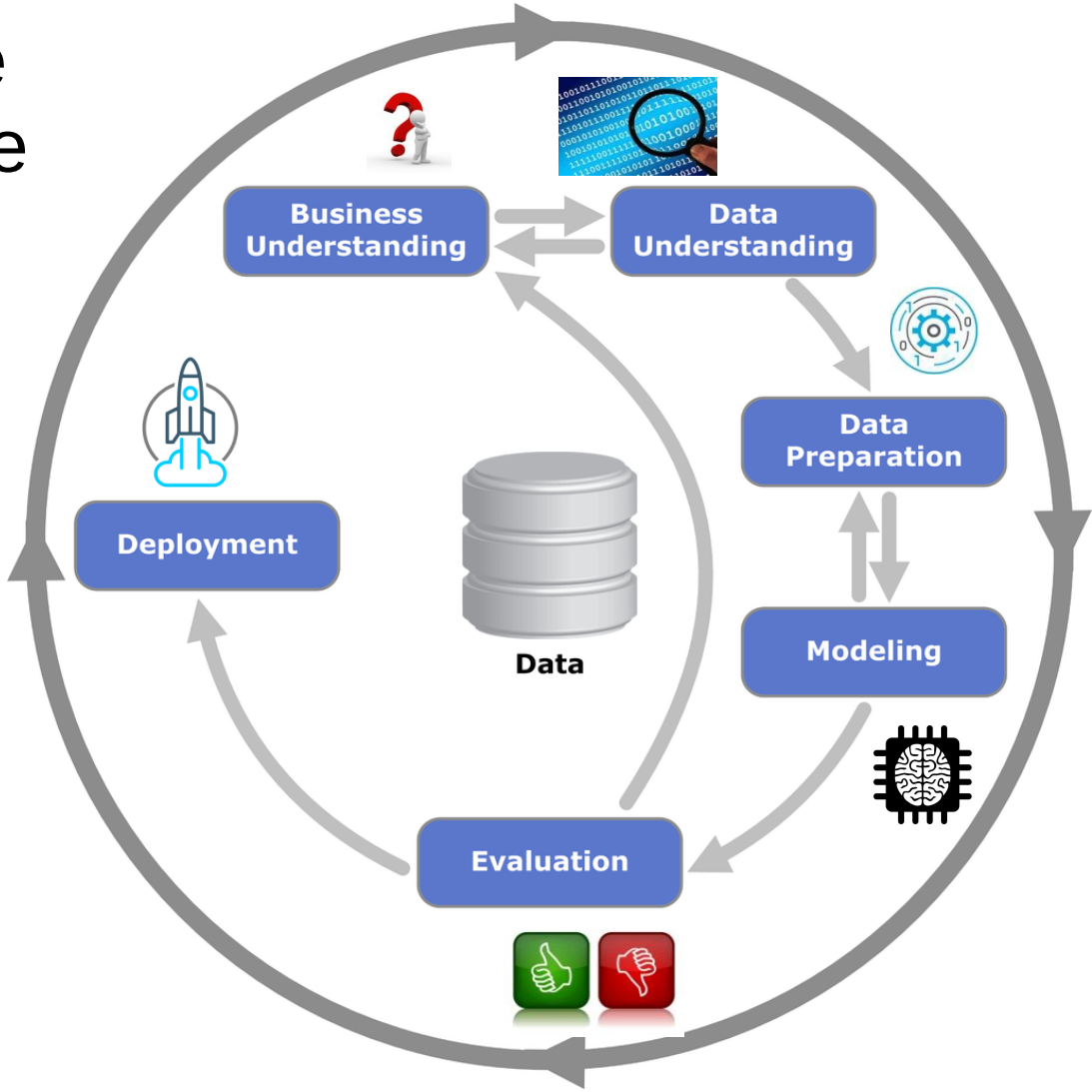
Binary (one out of two classes)

Multi-class (one out of n classes)

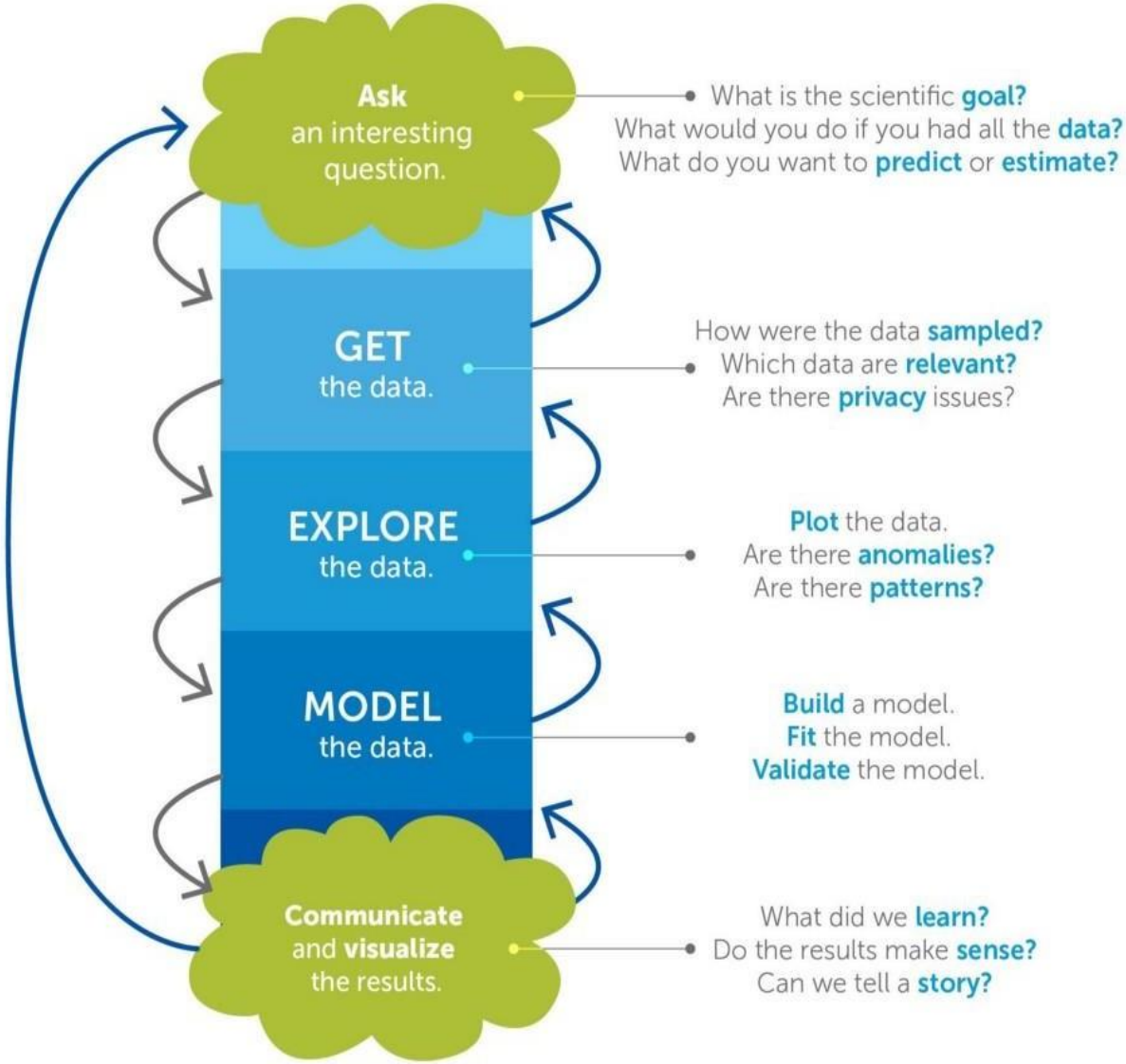
Multi-label (one or more out of n classes)

General analytical (data science) process

- E.g. CRISP Data Mining Model
- Modelling (machine learning) is only one of many important steps!



The Data Science Process



Derived from the work of Joe Blitzstein and Hanspeter Pfister, originally created for the Harvard data science course http://cs109.org/.

General analytical (data science) process

1. Identify problem / question
2. Identify & capture the available data

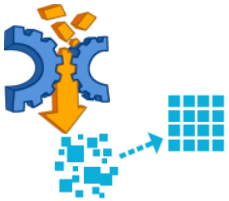
3. Prepare your data: clean & transform
4. Analyse your data ➔ actual machine learning

5. Create report with results, visualisation, insights
6. Embed results into business / decision making

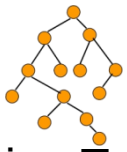
7. Plan for better data capturing in the future

Scoping the lecture

Data types & feature extraction



Data preparation & transformation



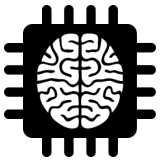
Decision Trees & Random Forests

Support Vector Machines

Neural Nets & Deep Learning

Ensemble learning

...

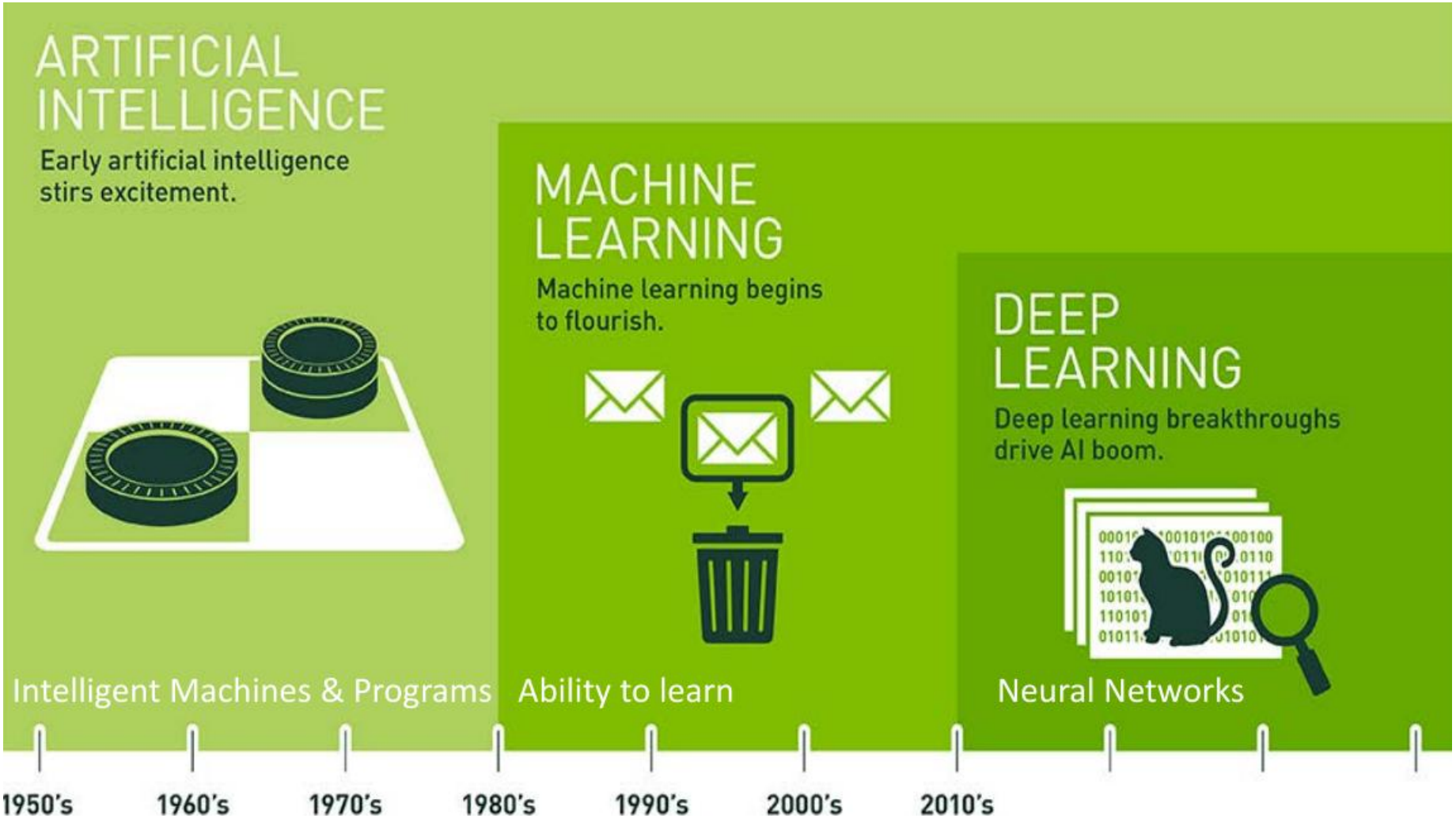


Evaluation

Model selection

Significance testing

AI & ML


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
Intelligent Machines & Programs
Ability to learn
Neural Networks
1950's
1960's
1970's
1980's
1990's
2000's
2010's

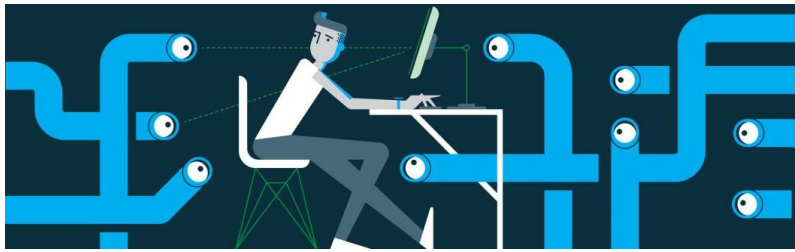
Scoping the lecture

- Matching your expectations?

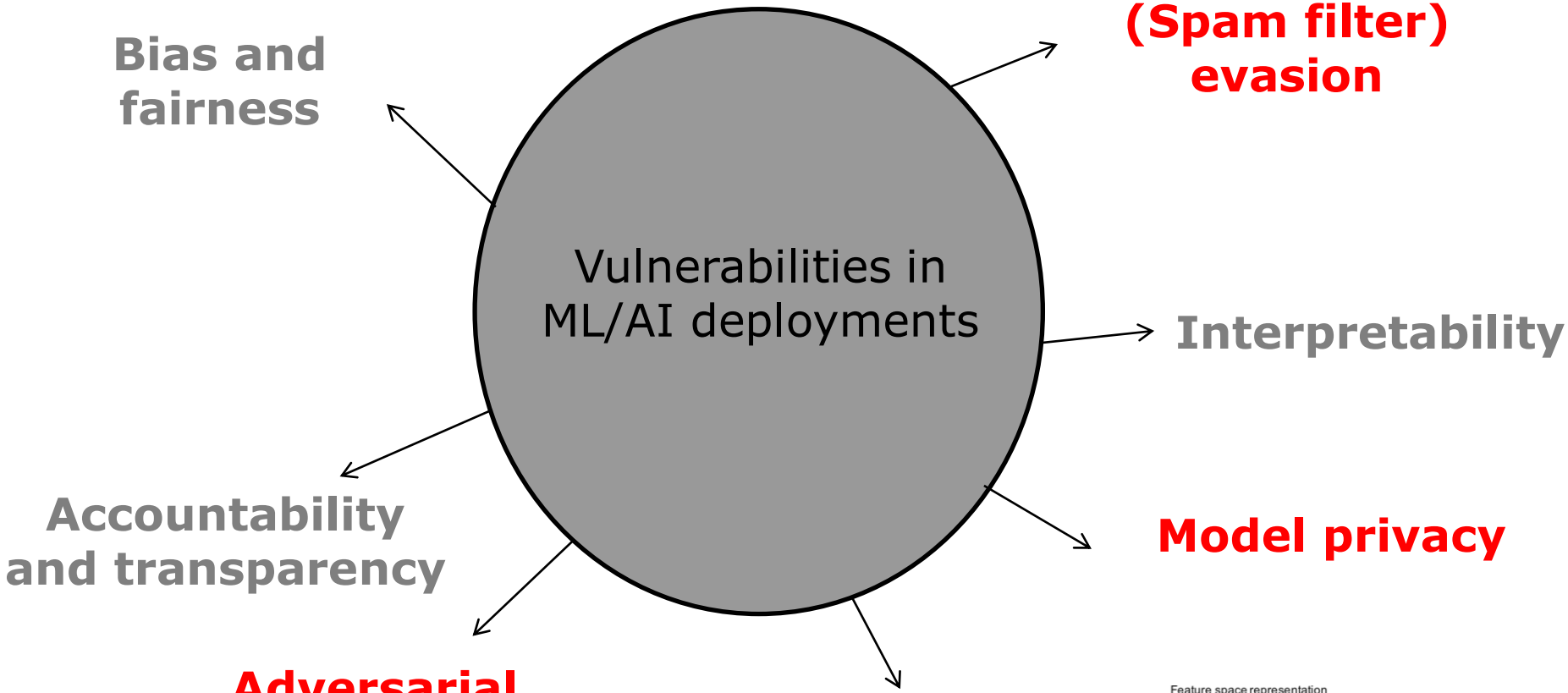
  - Anything not interesting at all?
  - Anything missing?

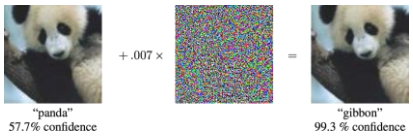
Other topics: Privacy

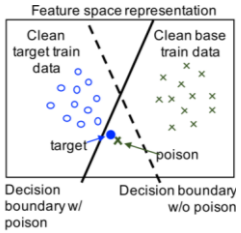
- Large amounts of personal data available
  - Collection & analysis often conflicting with data protection laws (more severe now: GDPR!)
- Especially with sensitive information
  - E.g. health data, financial data, ..
- Thus, often data needs to be anonymised before we can use it
  - E.g. generalisation of values (birthday ➔ month, year, decade)
  - Implications on the utility of the data & subsequent models...



Other topics: Vulnerabilities in ML

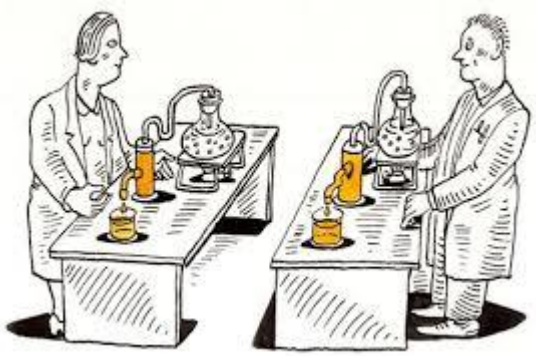




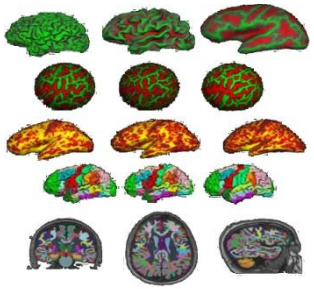


Other topics: Reproducibility

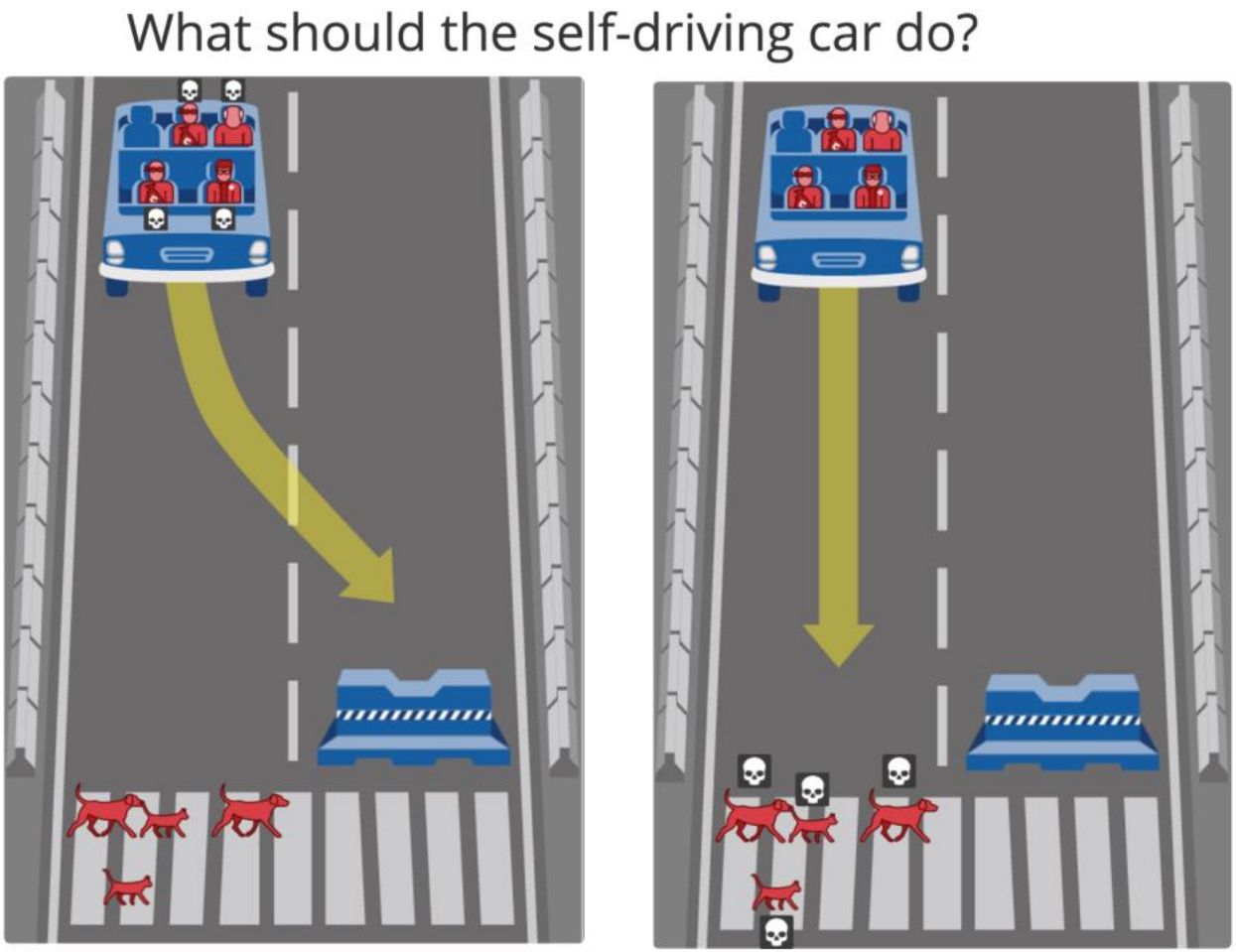
- For various reasons, we want to be able to achieve the same results as in a previous experiment (run)



- This is difficult, for various reasons
  - *Examples?*
  - Initialisation of algorithms (random number generation)
  - Order of data samples (e.g. for Perceptron)
  - Version of software used (bug fixes, improvements, etc..)
  - Version of operating system (e.g. a different C++ library..)
  - Any many more ...



Other topics: Ethics



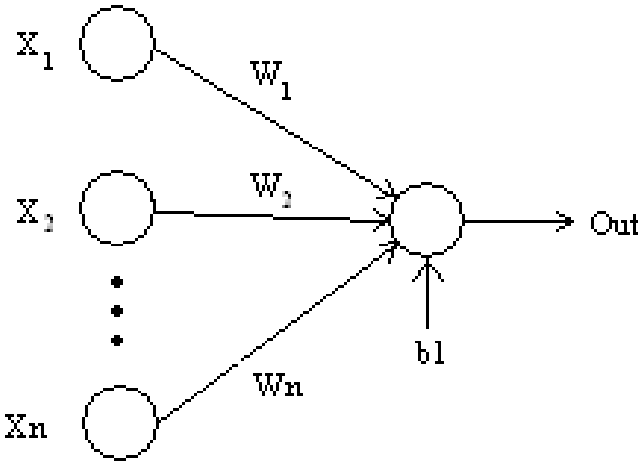
http://moralmachine.mit.edu/

Outline

- Introduction
- Perceptron
- k-NN
- Types of data & data preparation (intro)
- Performance evaluation (intro)

Perceptron

- Proposed in 1957
  - *Why is it relevant for you? ☺*
- Artificial (neural) network
  - only one neuron; “single-layer perceptron”
- Input: continuous values X (x_1, x_2, …, x_n)
- Output: activation a
  - Boolean value 0 / 1



Perceptron

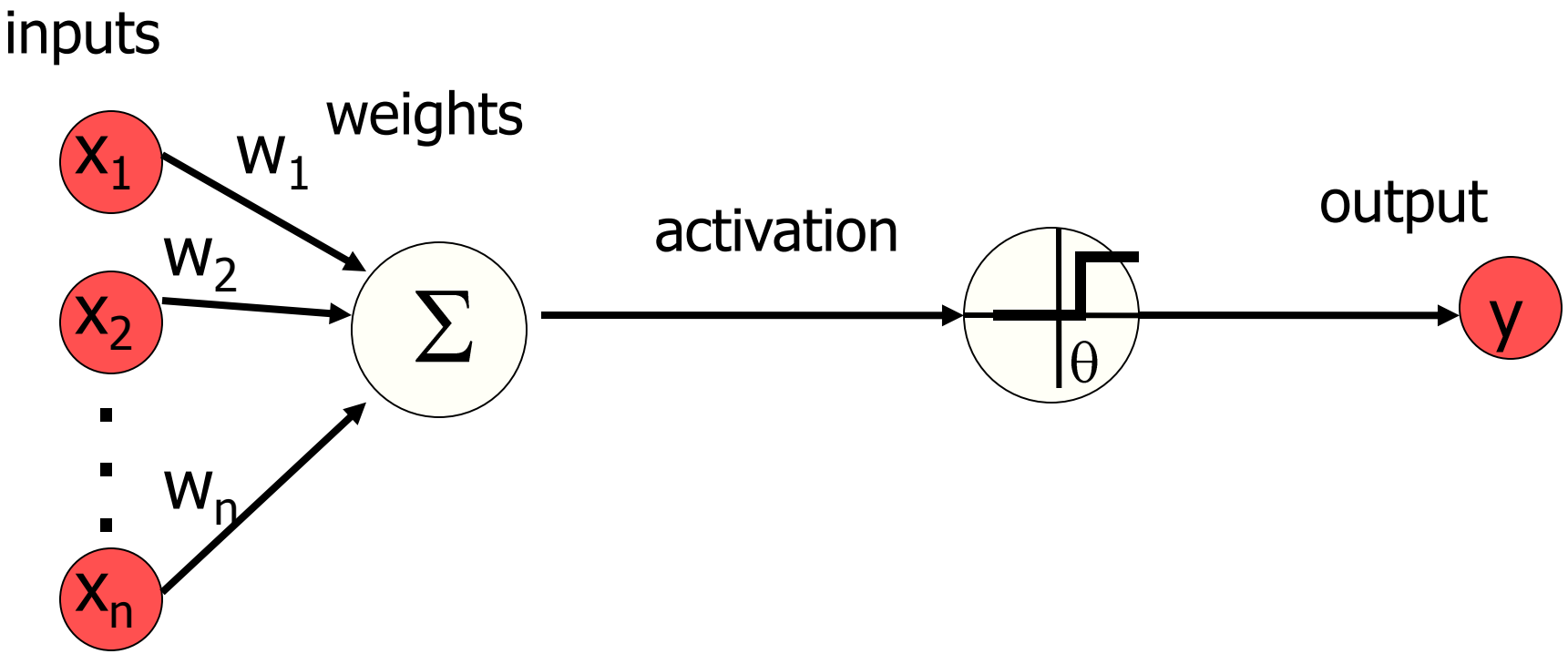
- Linear combination of inputs
  - Using weights *W* $a = \sum_{i=1}^{n} w_i x_i$

- Pass through threshold activation function with threshold θ

$$y = f(x) = \begin{cases} 1 \text{ if } a \geq \theta \\ 0 \text{ if } a < \theta \end{cases}$$

(Heaviside step function)

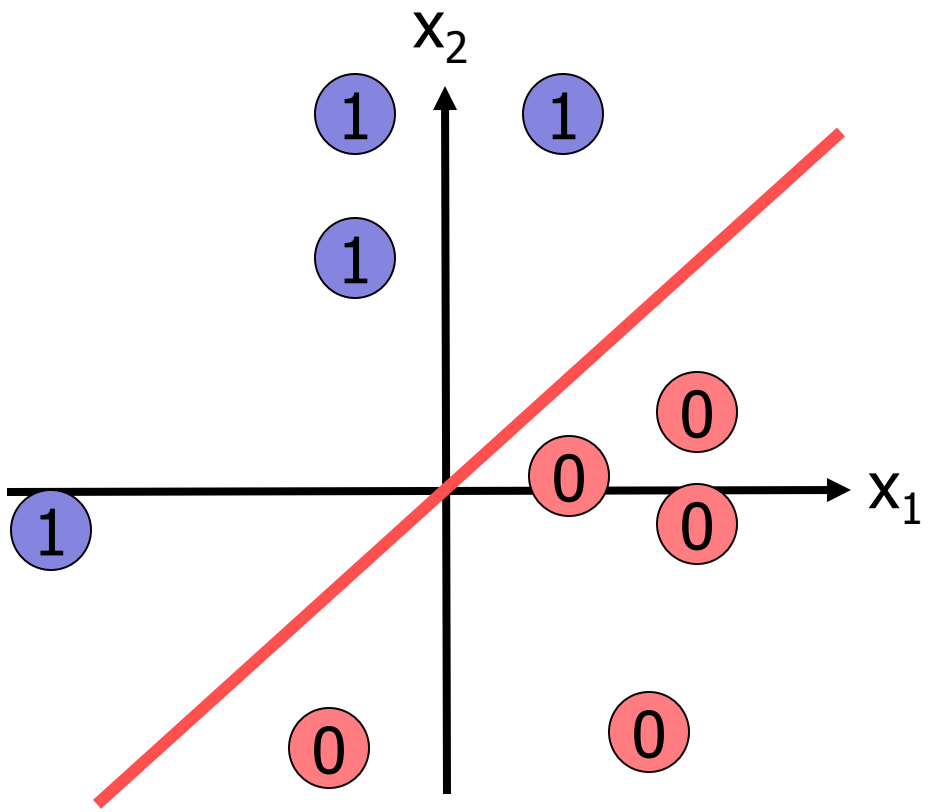
Perceptron


inputs
weights
x_1
w_1
w_2
x_2
w_n
x_n
Σ
activation
θ
output
y

Perceptron: simple example

- Two-dimensional data set
  - Variables x_1 & x_2
    - E.g. position on a map
- Weight vector also has two components
  - w_1 & w_2
- Data set can be easily visualised in Cartesian coordinate system
- Two classes: ***0*** and ***1***
  - E.g. **poor** and **rich** countries

Perceptron visualised



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Perceptron: learning

- Training the model: learning the weights from labelled samples (label: **y**)

  - Initialise weights

  - Repeat
    - Present training sample **x**
    - Predict sample label: **y'** = f(x)
    - Prediction correct? Compare **y** and **y'**
      - if **y' ≠ y** ➔ Compute new weights **w'** as **w'**=**w** + α (y-y') **x**

  - Until prediction correct (y'=y) for all samples

Perceptron: learning

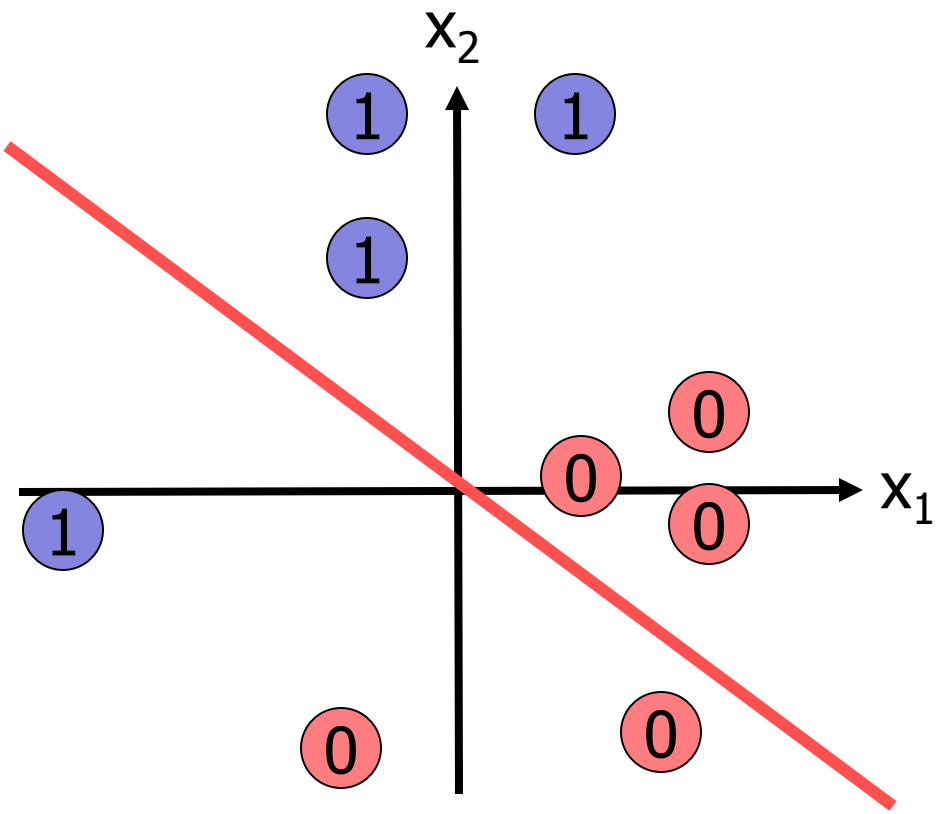
- Parameter α: *learning rate*
  - determines the magnitude of weight updates

- If output is correct (y=y')
  - weights **not** changed

- If output is incorrect (y ≠ y')
  - weights changed such that the output of the for the new weights **w'** is ***closer*** to the input x_i

Perceptron training

- t=0: random weights for initialisation



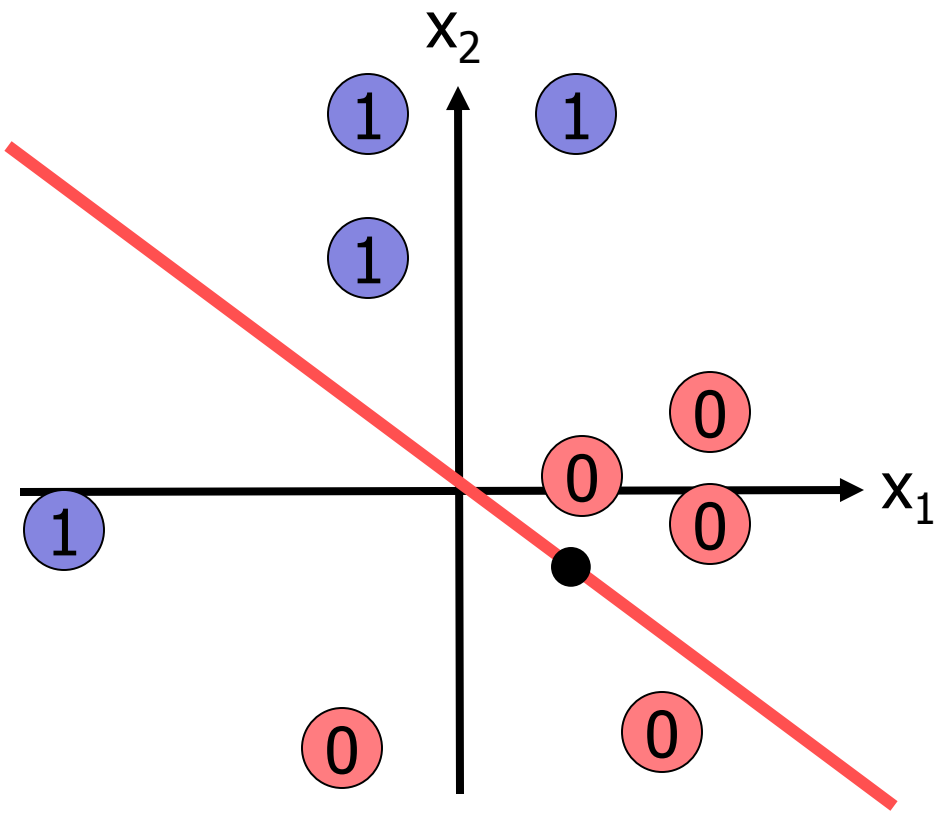
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(cf. $y = ax$)

Perceptron training

- t=0: random weights for initialisation



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Initial weight vector:

$W_1=1,\ W_2=1$

Point on decision line

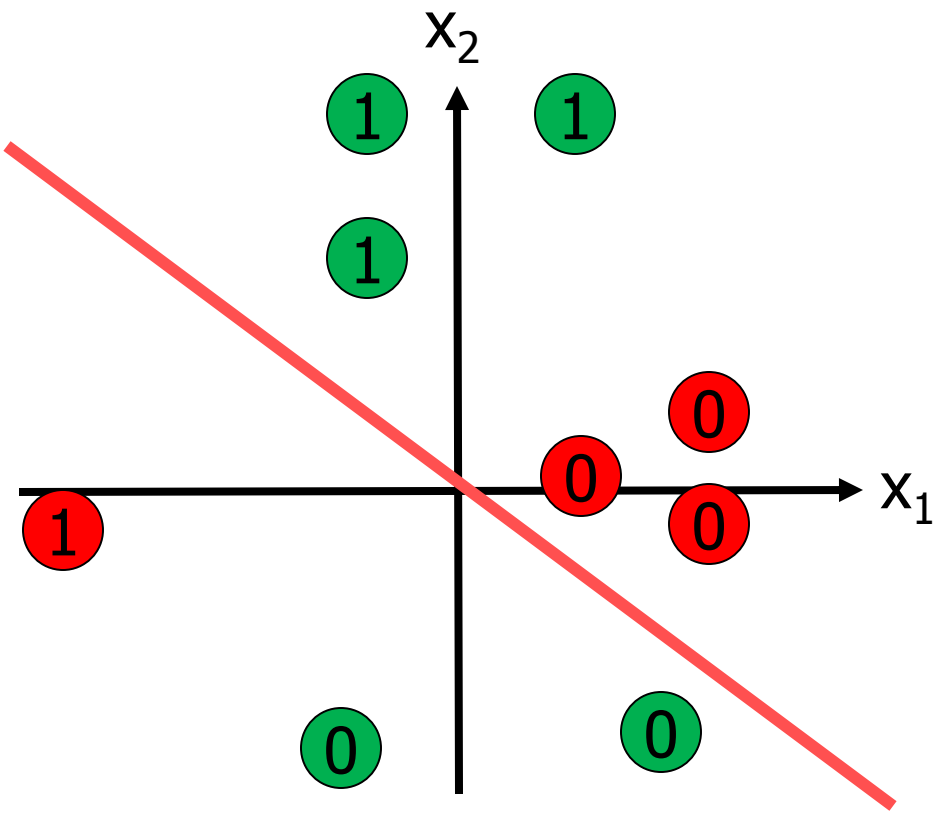
(1,-1)

➔ 1*1 +1*(-1) =

= 1 - 1 = 0 = θ

Perceptron training

- t=1: compute y' = f(x), compare to *y*



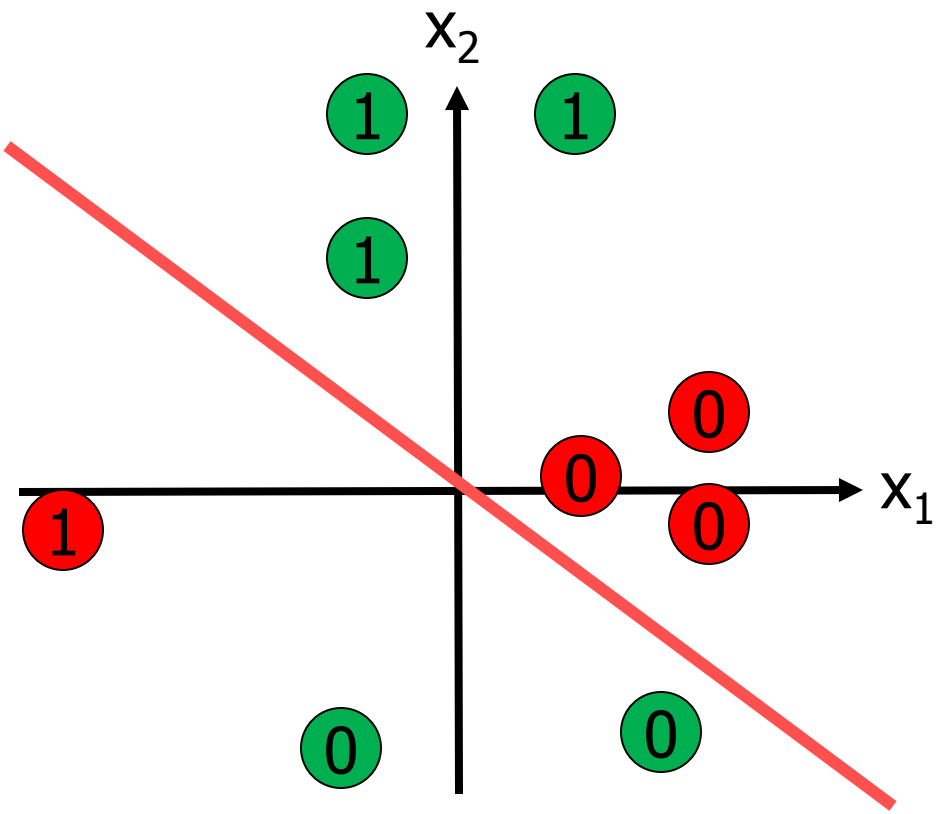
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=1: not all f(x) = y ➔ adapt weights (decision line)



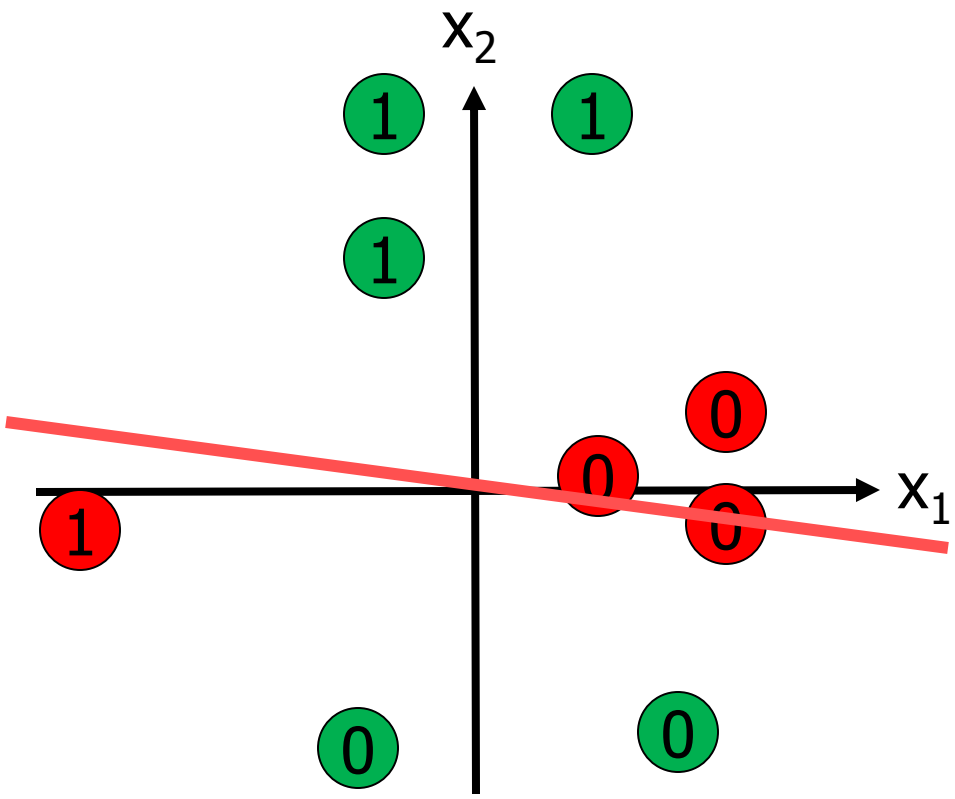
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=1: not all f(x) = y ➔ adapt weights (decision line)



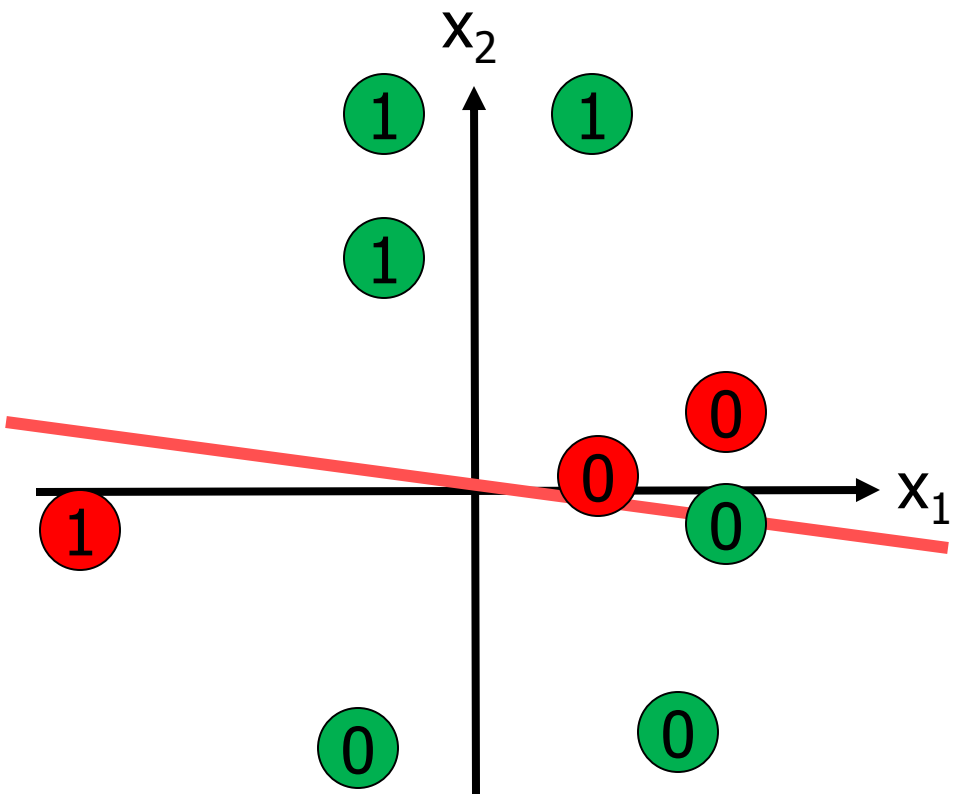
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=2: not all f(x) = y ➔ adapt weights (decision line)



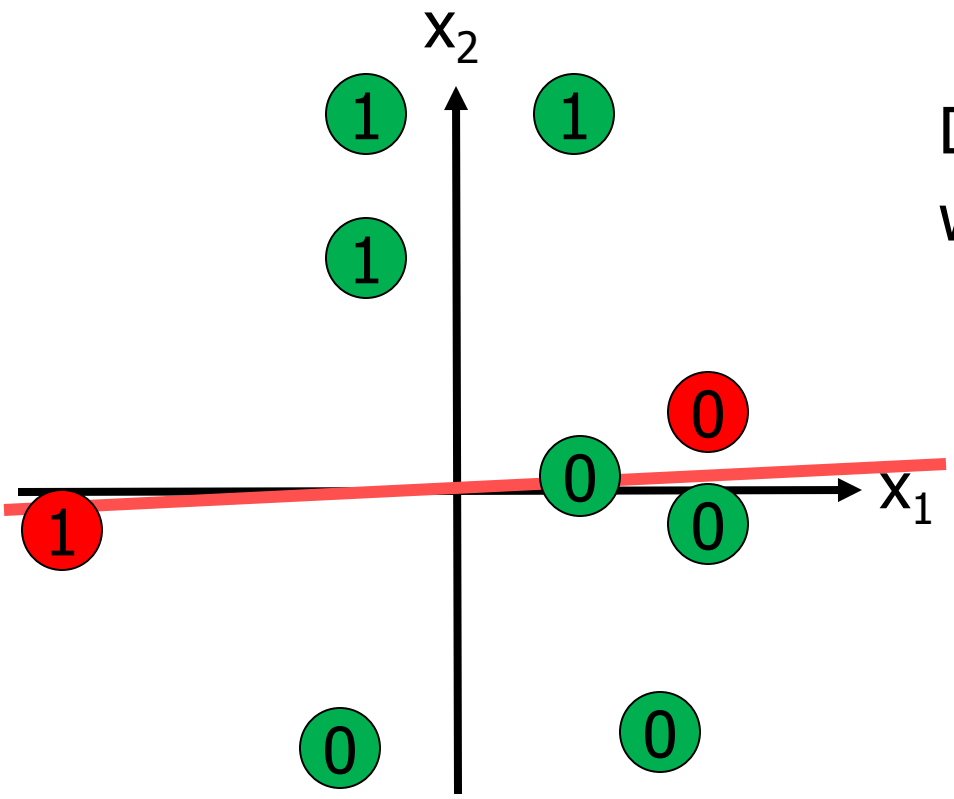
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=3: not all f(x) = y ➔ adapt weights (decision line)



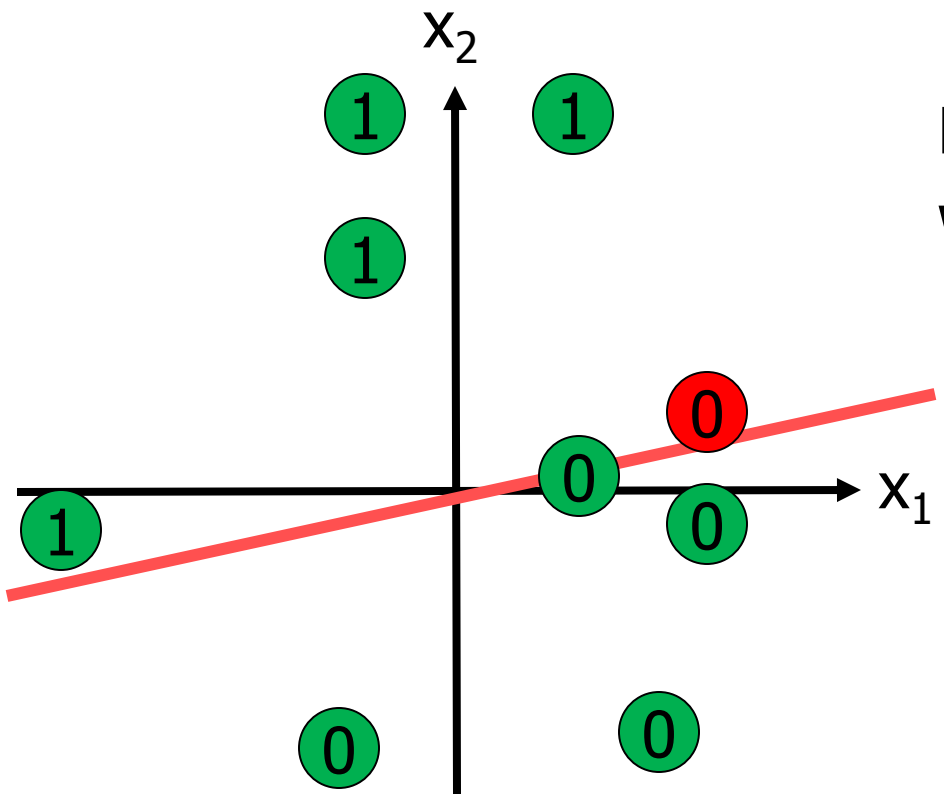
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=4: not all f(x) = y ➔ adapt weights (decision line)



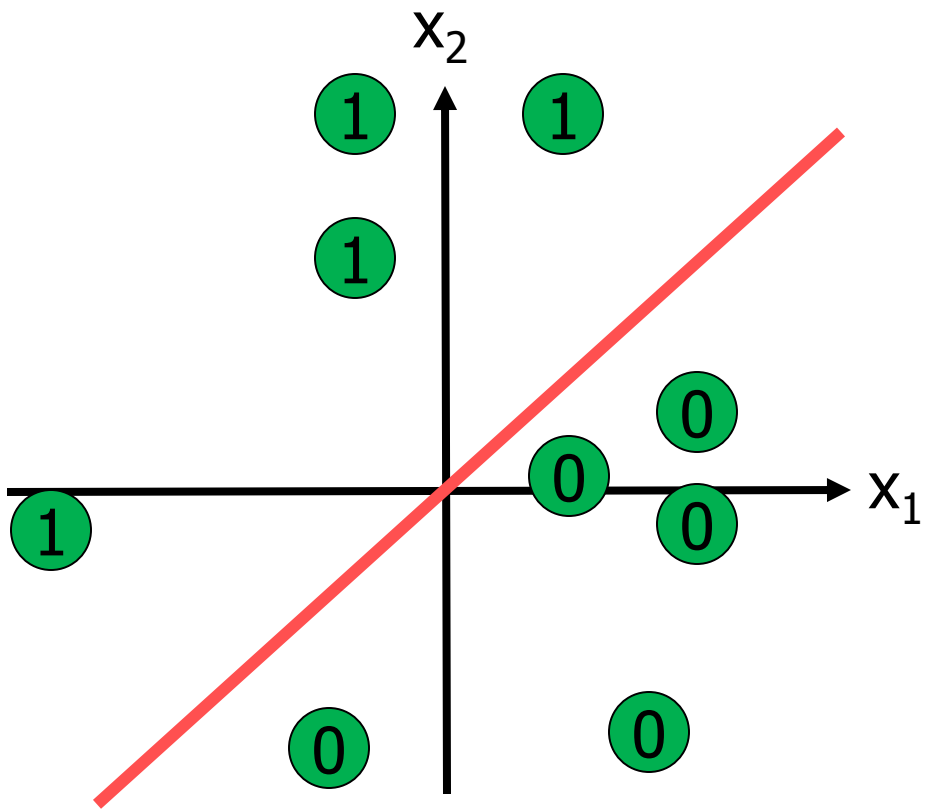
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = y ➔ final state



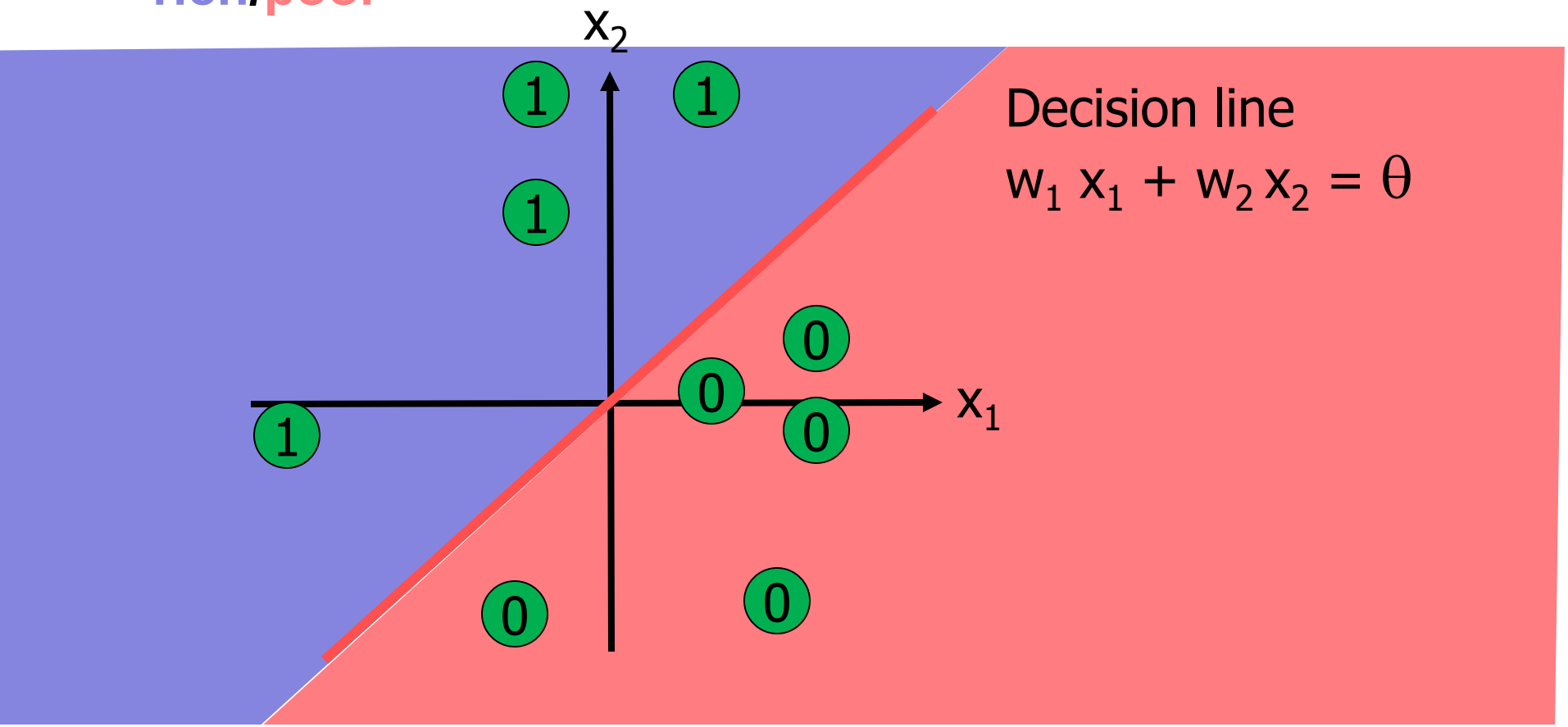
Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(green: correctly , red: wrongly classified)

Perceptron training

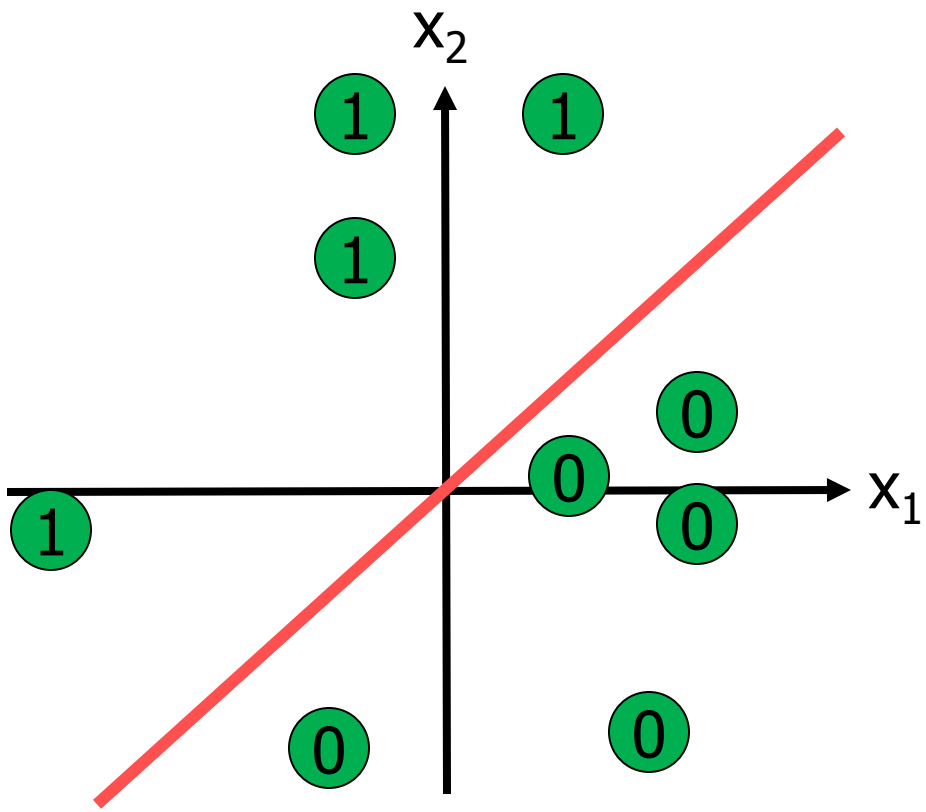
- t=5: all f(x) = y ➔ final state; classification in **rich**/**poor**



(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = y ➔ final state



Decision line

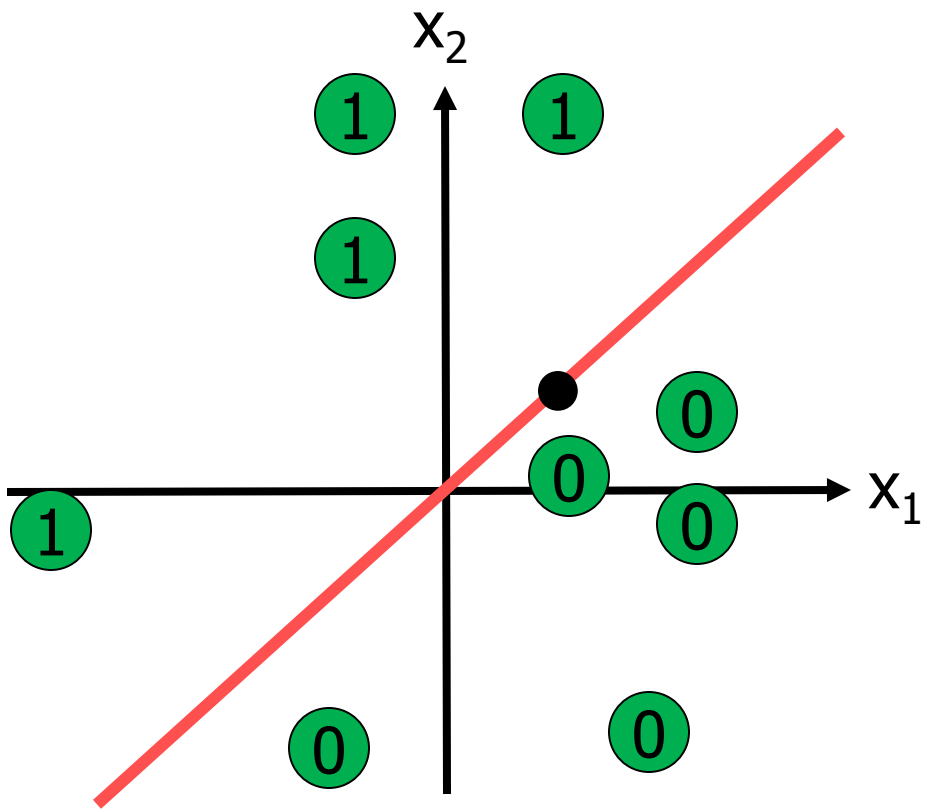
$w_1 x_1 + w_2 x_2 = \theta$

Weight vector ?

(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = y ➔ final state



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Weight vector ?

$w_1 = -1,\ w_2 = 1$

Point on decision line

(1,1)

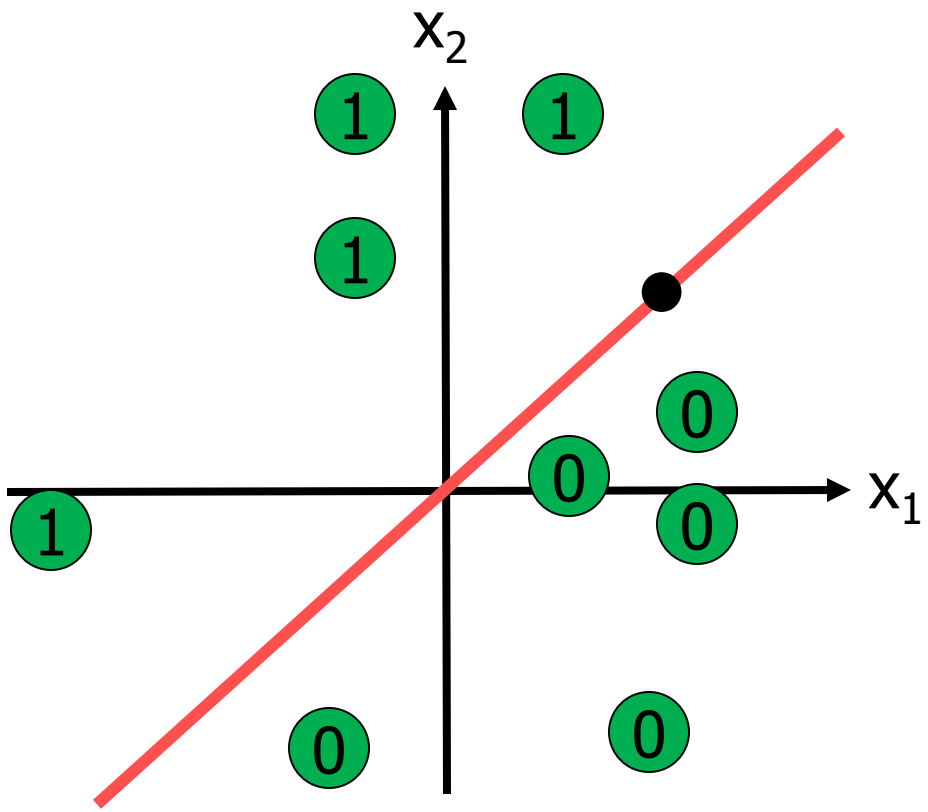
➔ (-1)*1 + 1*1 =

= -1 + 1 = 0 = θ

(green: correctly , red: wrongly classified)

Perceptron training

- t=5: all f(x) = y ➔ final state



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Weight vector ?

$w_1 = -1,\ w_2 = 1$

Point on decision line

(2,2)

➔ (-1)*2 + 1*2 =

= -2 + 2 = 0 = θ

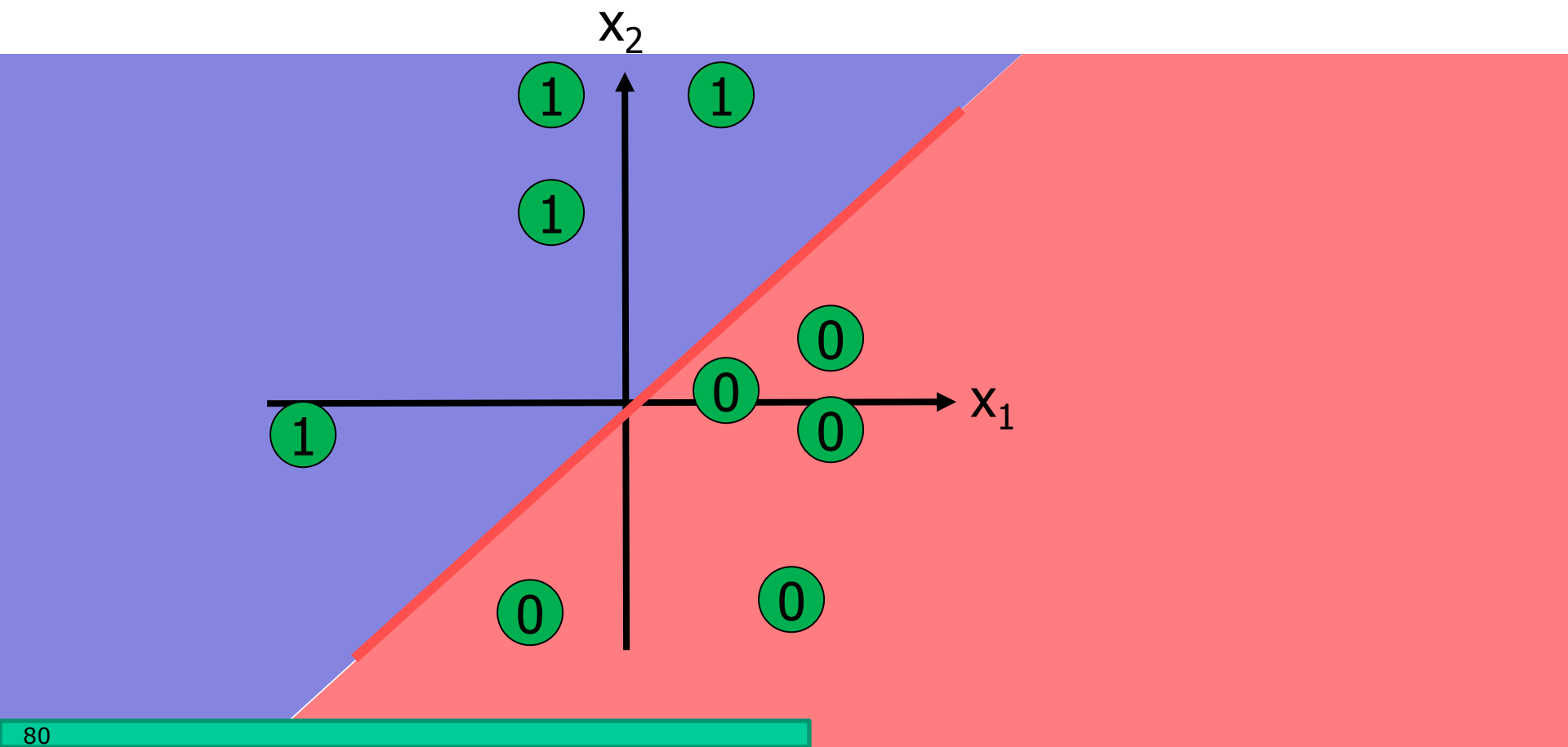
(green: correctly , red: wrongly classified)

Perceptron: properties

- Separates linearly
  - Converges to a stable state when data is *linear separable*

Perceptron: properties

- Separates linearly
  - Converges to a stable state when data is *linear separable*

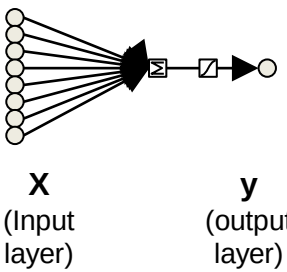


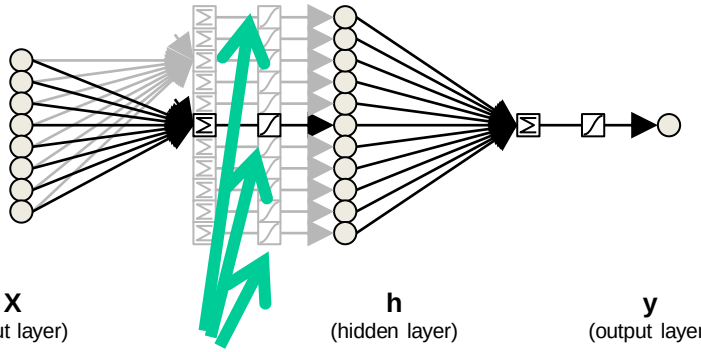
Perceptron: properties

- Very simple model
- Separates linearly
  - Converges to a stable state when data is *linear separable*
- Can predict binary decisions (true/false)
  - Can be extended for multi-class problems
  - *Training rule similar to other online learning algorithms, e.g. Self-Organising Maps*

- *Why is it relevant?*

Perceptron: why is it relevant?



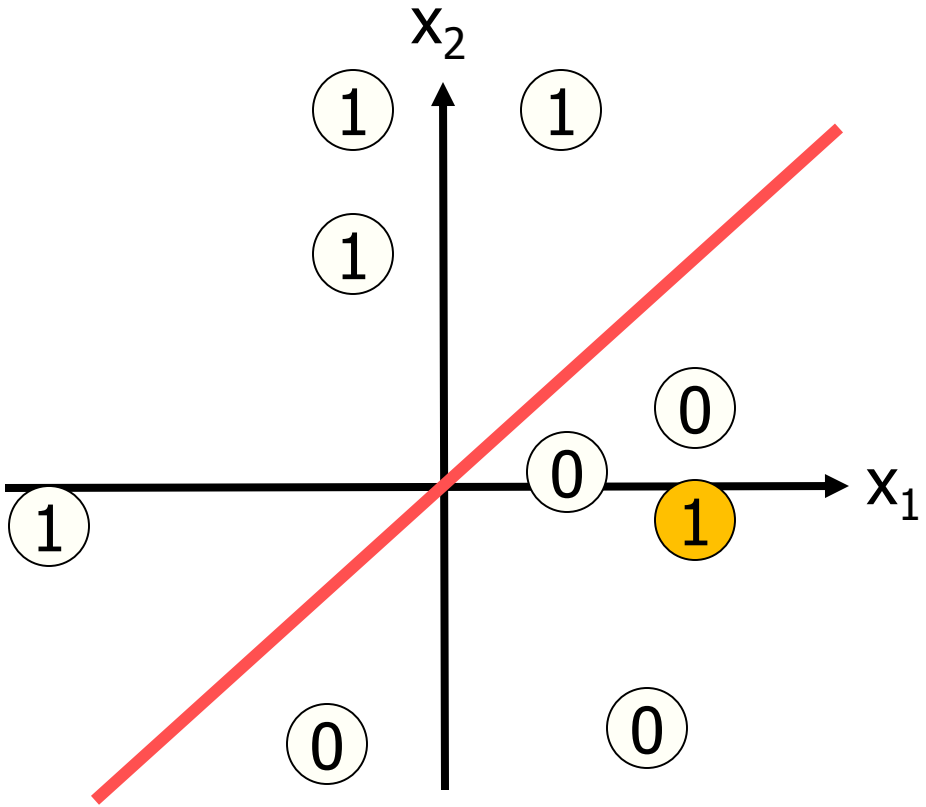


Perceptron demos

- http://ditam.github.io/demos/perceptron/perceptronDemo.html

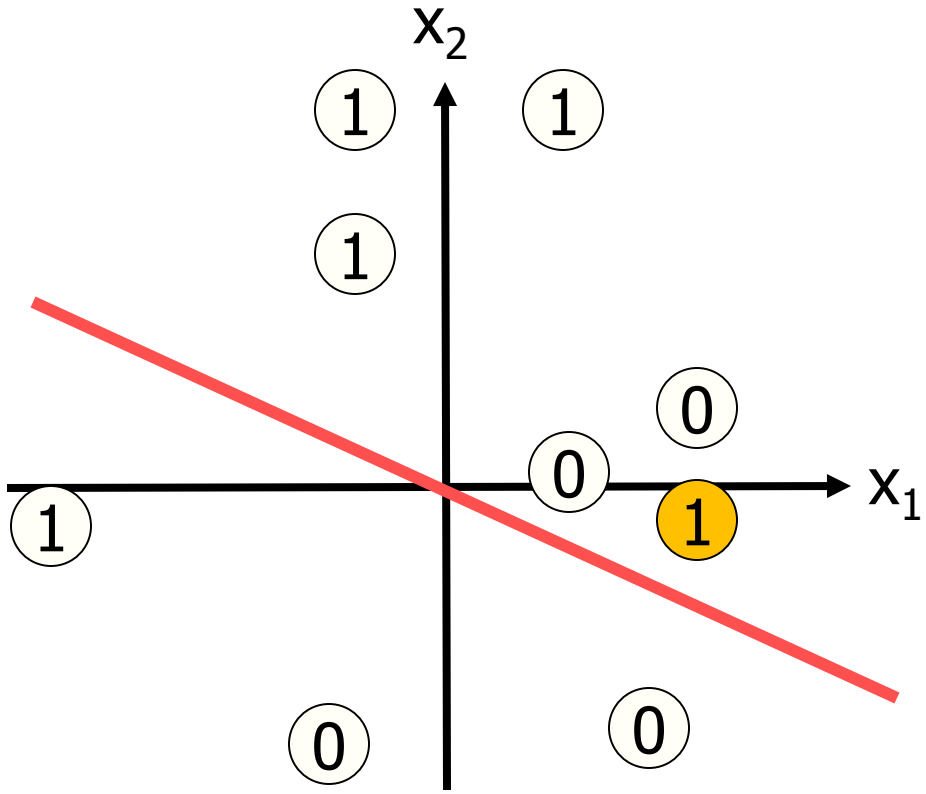
- https://www.cs.utexas.edu/~teammco/misc/perceptron/

Perceptron: Non-linear separable



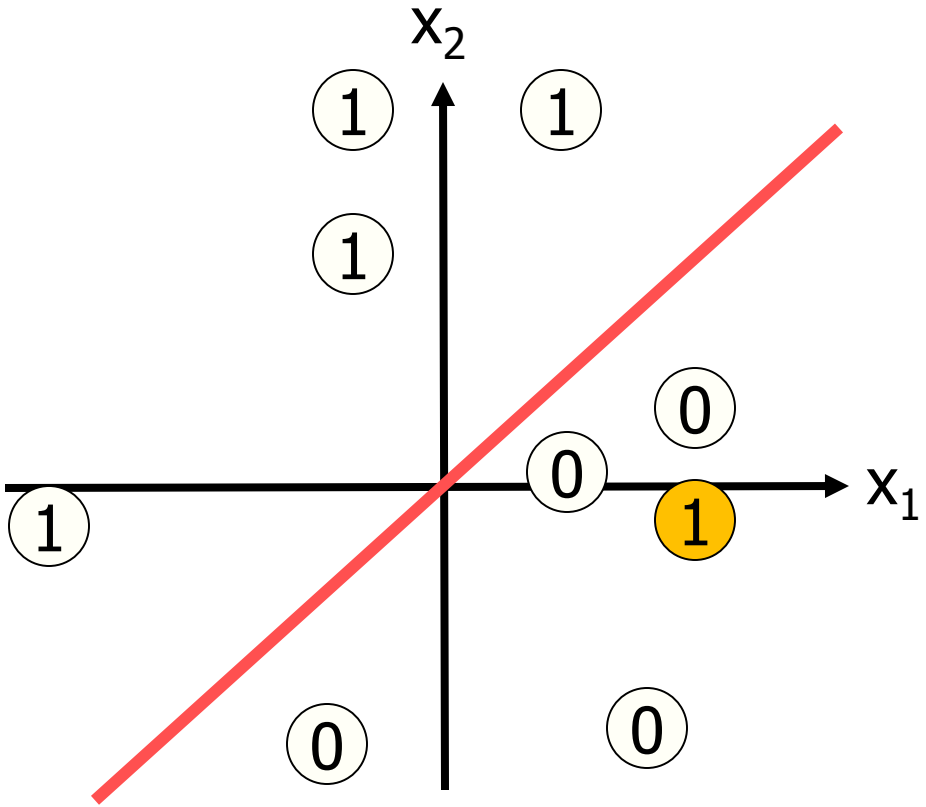
- *Perceptron will not converge to a stable state, but oscilate*

Perceptron: Non-linear separable



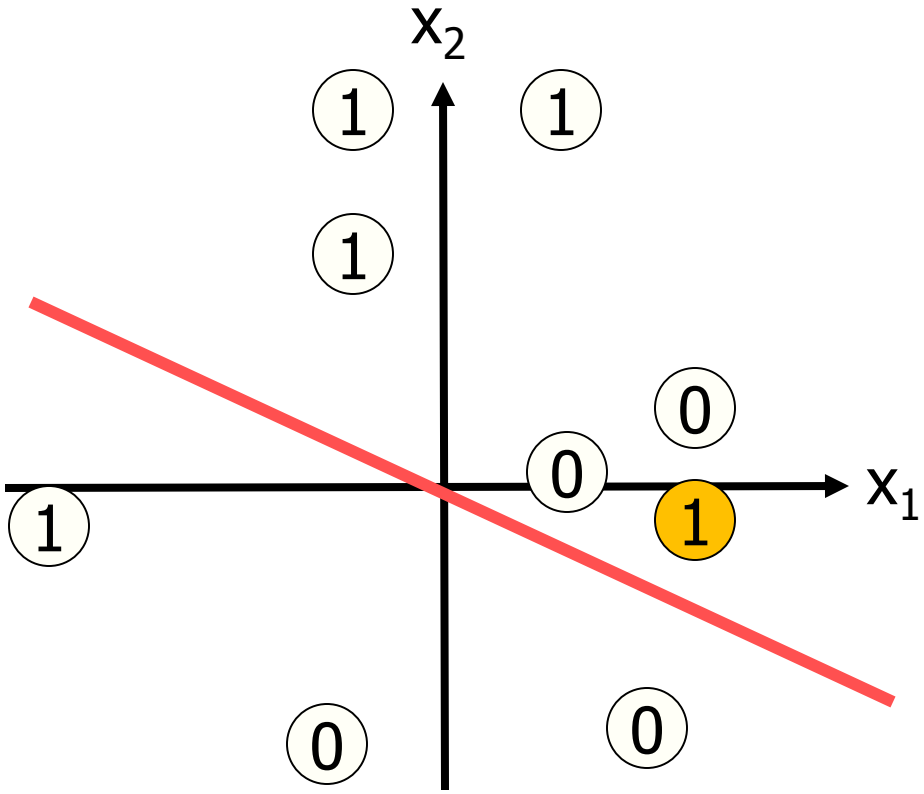
- *Perceptron will not converge to a stable state, but oscilate*

Perceptron: Non-linear separable



- *Perceptron will not converge to a stable state, but oscilate*
- *Solution?*

Perceptron: Non-linear separable



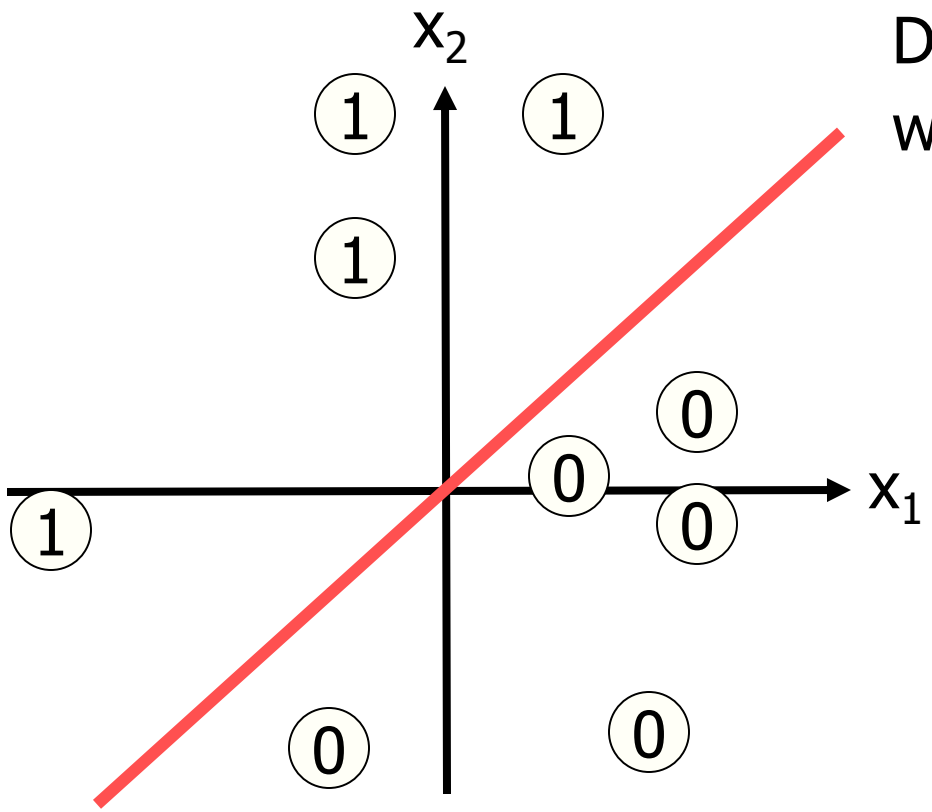
- *Perceptron will not converge to a stable state, but oscilate*

➔ *Need stopping criterion to end training*

Perceptron: Non-linear separable

- Stopping criteria for training
  - Stop after a maximum number of n training iterations
    - *How to set n?*
  - Stop if there is no more improvement since k iterations
    - Improvement measured e.g. as Accuracy (correctly classified samples)
  - ....
  - *Is that sufficient?*
- Pocket algorithm:
  - Keeps the best solution found so far
    - (e.g. the one with highest accuracy)
  - Returns that solution (instead of last state)

Perceptron: bias

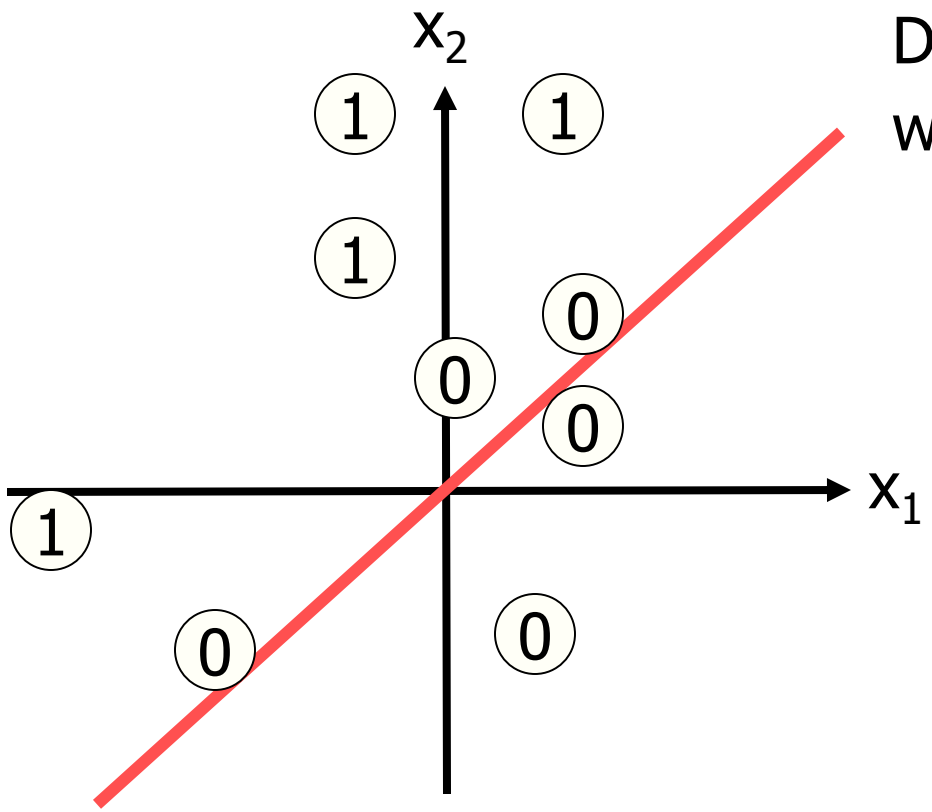


Decision line

$w_1 x_1 + w_2 x_2 = \theta$

(cf. $y = ax$)

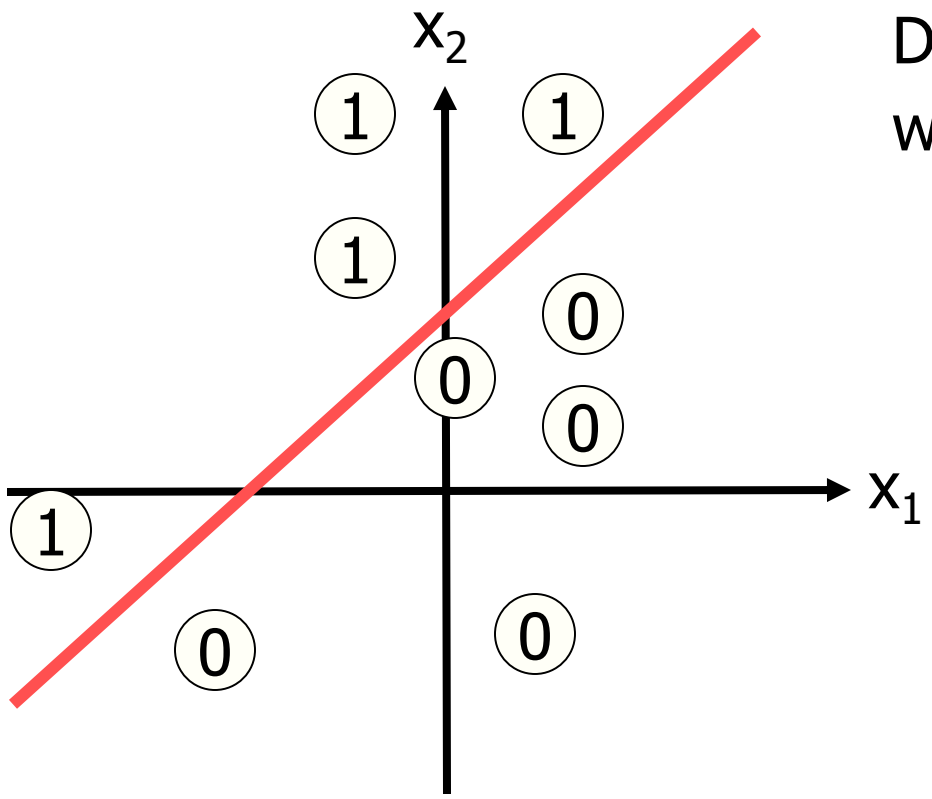
Perceptron: bias



Decision line

$w_1 x_1 + w_2 x_2 = \theta$

Perceptron: bias

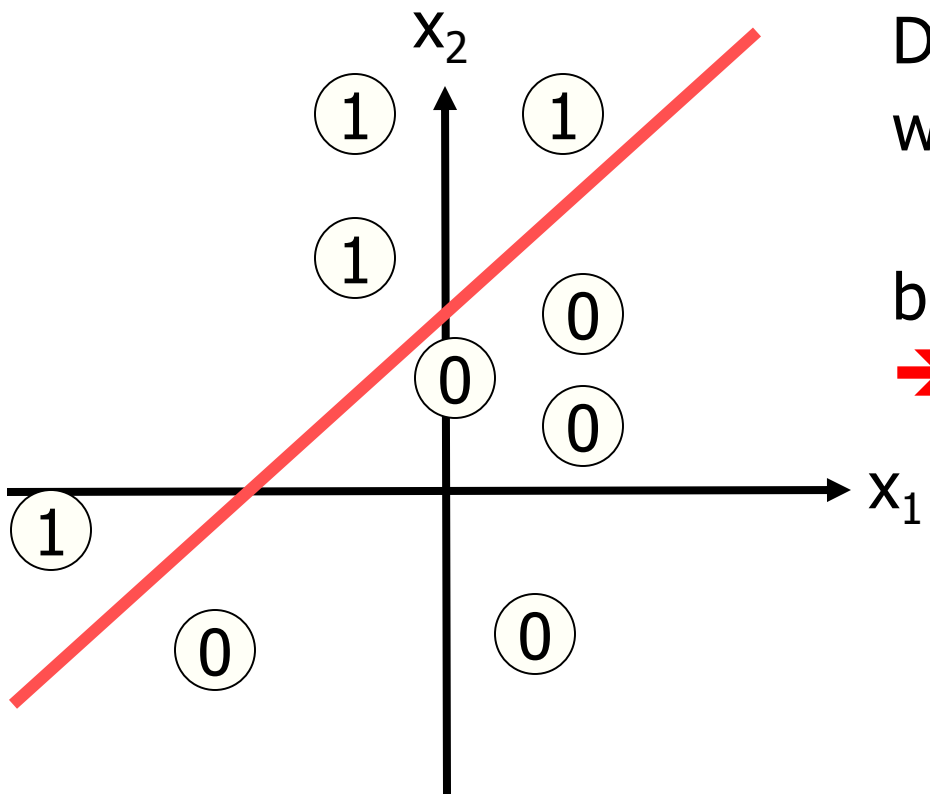


Decision line

$w_1 x_1 + w_2 x_2 + b = \theta$

(cf. $y = ax + b$)

Perceptron: bias



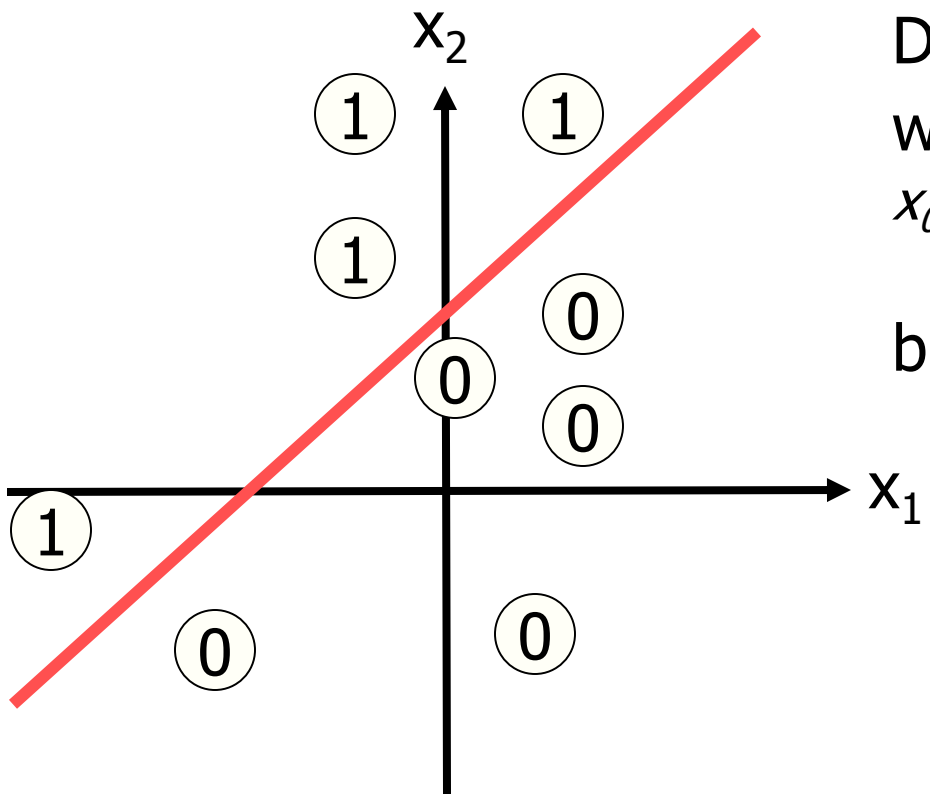
Decision line

$w_1 x_1 + w_2 x_2 + b = \theta$

bias b: constant

➔ how to determine?

Perceptron: bias



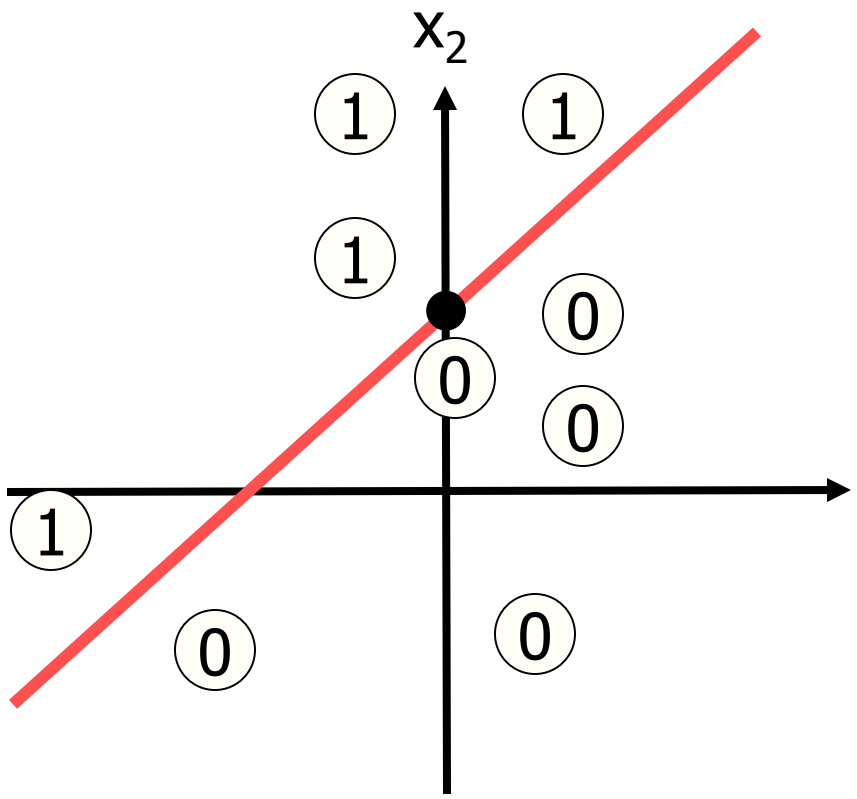
Decision line

$w_0x_0 + w_1 x_1 + w_2 x_2 = \theta$

$x_0 = 1$ (constant)

bias w_0: learned

Perceptron: bias



Decision line

$w_0x_0 + w_1 x_1 + w_2 x_2 = \theta$

$x_0 = 1$ (constant)

W_1=-1, W_2=1, b = -2

Point on decision line

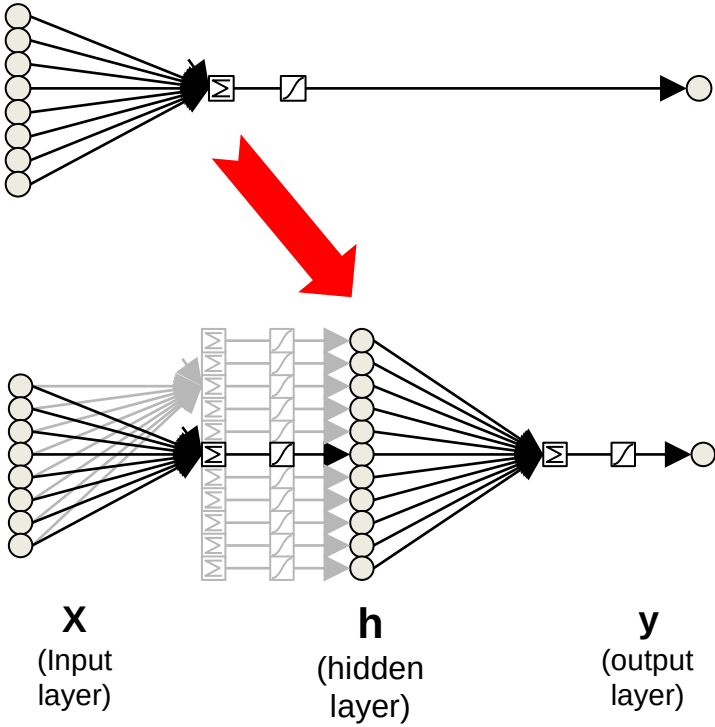
(0,2)

➔-2*1 + -(1)*0 + 1*2 =

= -2 + 0 + 2 = 0 = θ

Extensions to Perceptrons

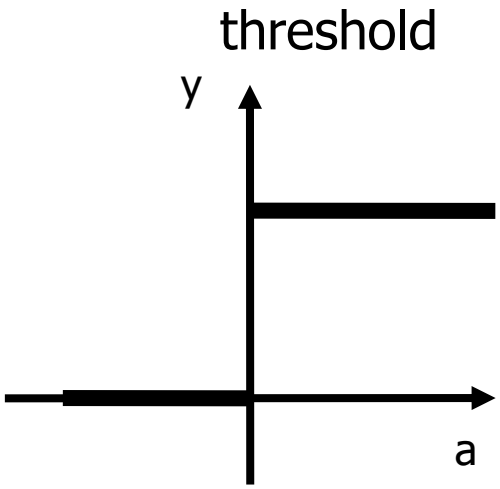
- Perceptron is a simple, one-neuron neural network
  - Basis for architectures that combine many basic neurons
    - Input layer & output layer
    - *Hidden layer(s)*
  - Initial motivation: model of human brain
    - Neurons connected via synaptic links

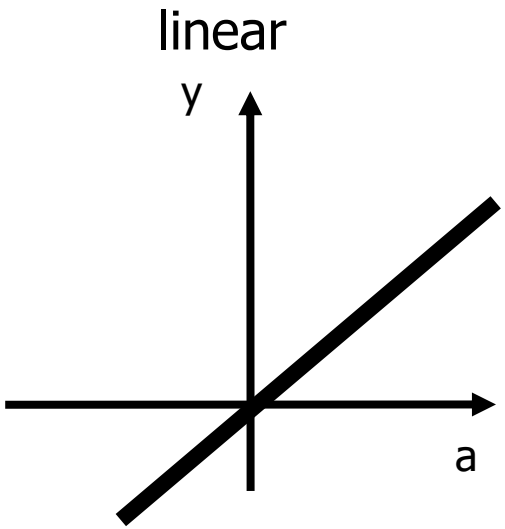


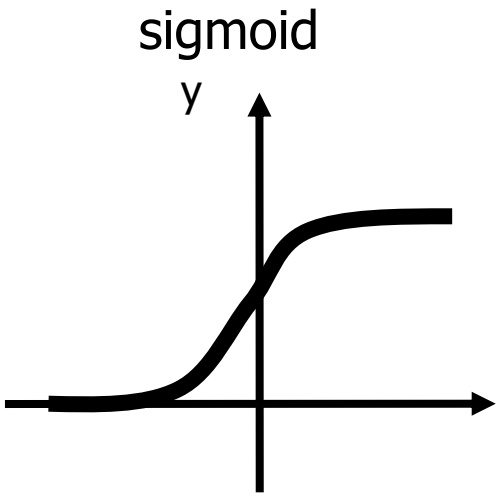
- Extensions for non-linear separable classes
  - Data space projections (Kernels), similar to SVM
    - More on that later ....

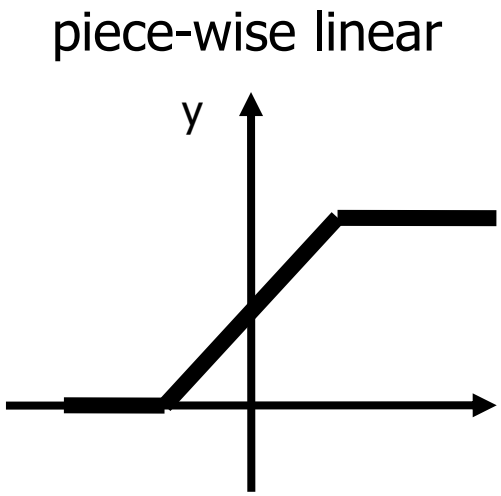
Perceptron: Activation Functions

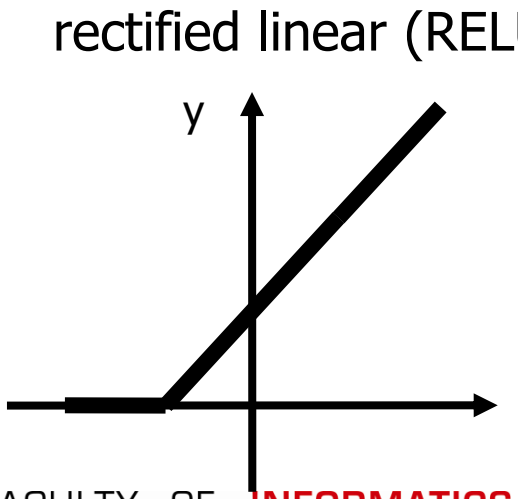
- Especially in Neural Networks, different activation functions are used











Outline

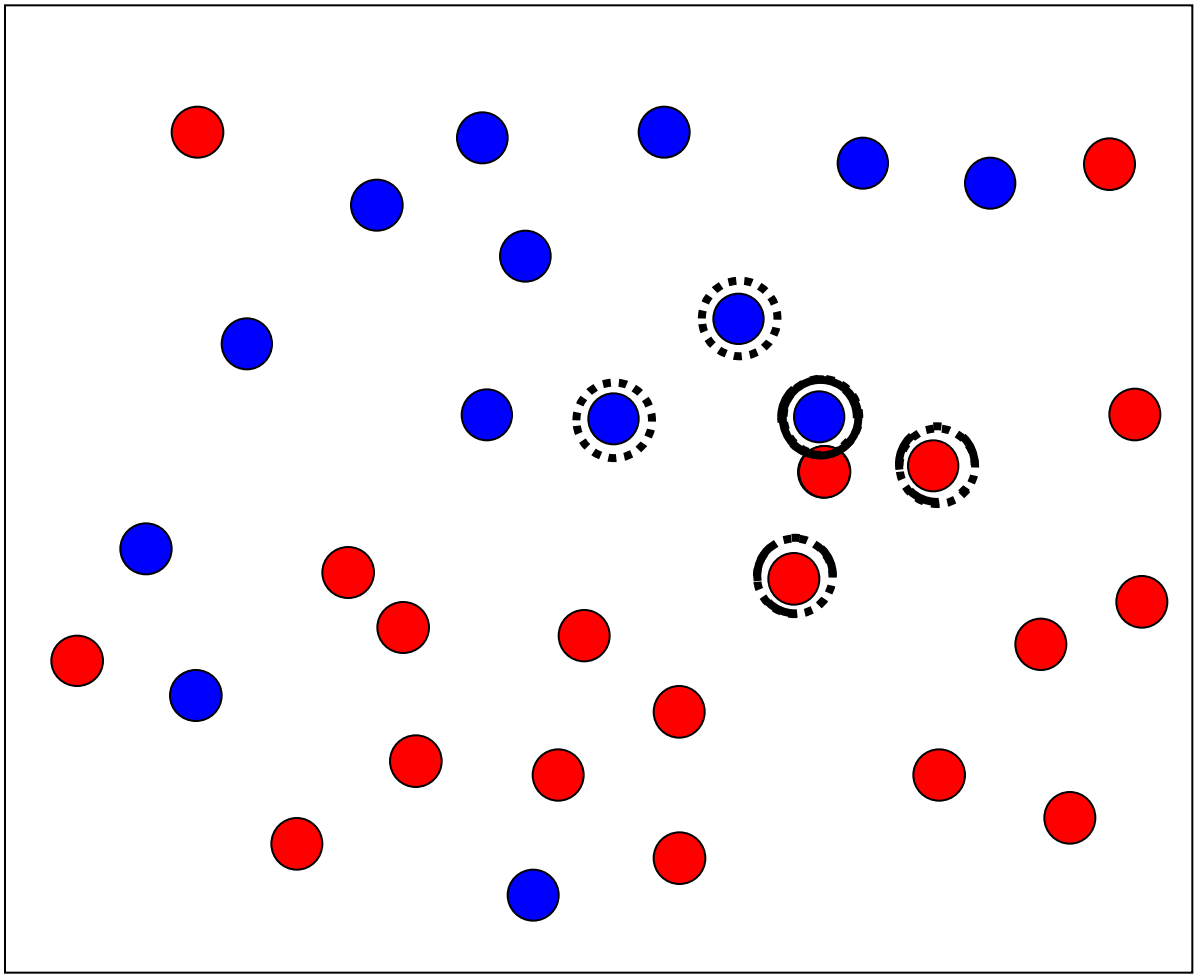
- Introduction
- Perceptron
- k-NN
- Types of data & data preparation (intro)
- Performance evaluation (intro)

k-nearest neighbour

- Simple algorithm, but well known
- Classify inputs based on *k* closest training examples
  - *k* to be chosen, high influence
  - Definition of “closest” – distance function

k-nn: Example

- 2-dimensional data, two classes (red & blue)



k-nn – Properties

- Very sensitive to local noise
  - Little abstraction / generalisation
- Larger values of *k* reduce the effect of noise
  - makes boundaries between classes less distinct
- How to determine *k* ?
  - Good values for *k* vary (a lot!) with the data
- *k*=1 is called “nearest neighbour algorithm”

k-nn – Properties

- *What is the maximum value for k?*

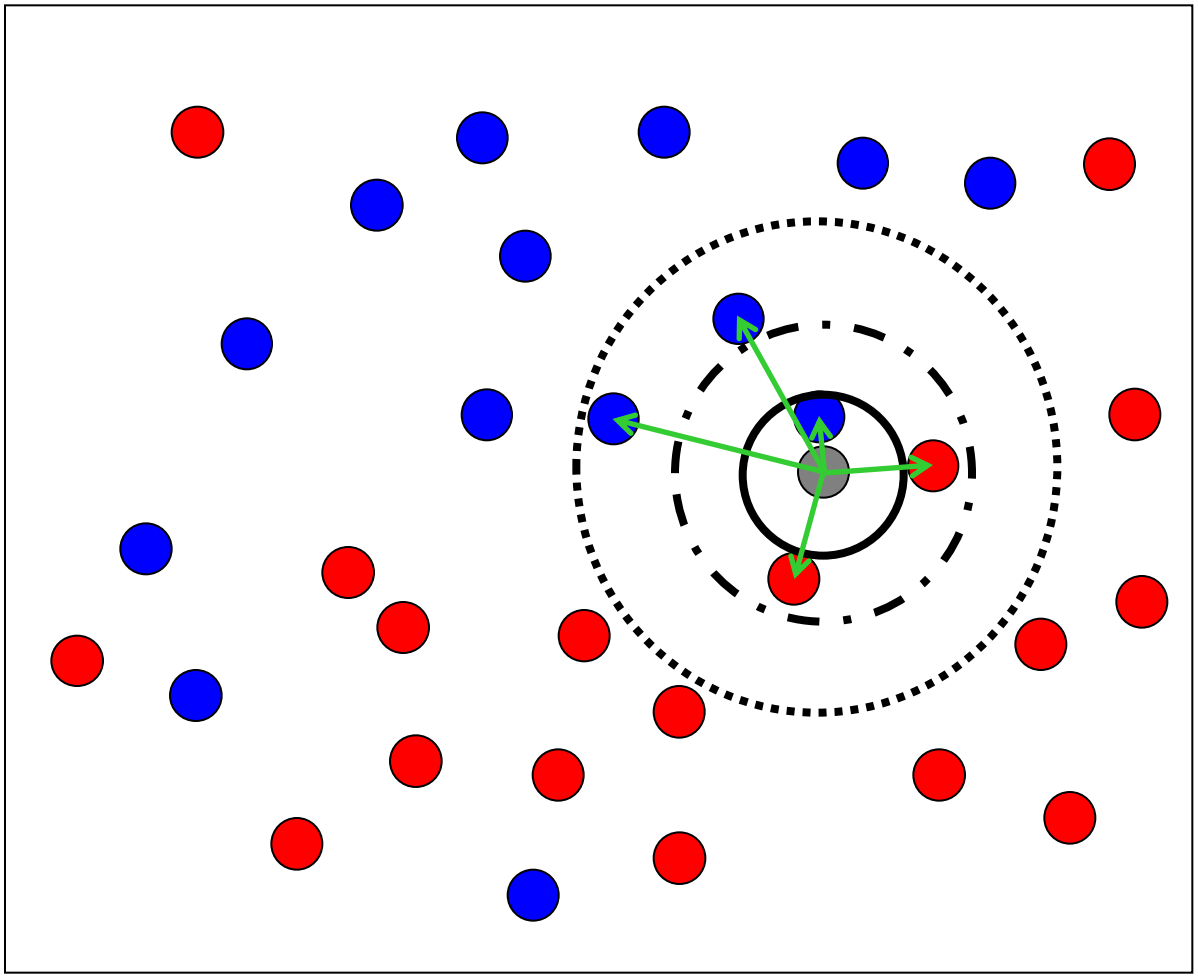
k-nn – Properties

- *What is the maximum value for k?*

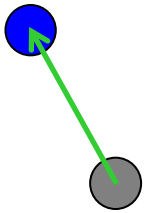
- *What happens if k = n (number of samples)*

k-nn: distance function

- In this example: Euclidean distance

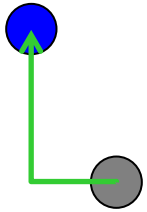


k-nn: distance function

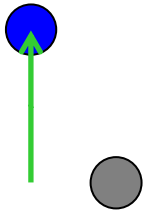


- Previous example: Euclidean distance

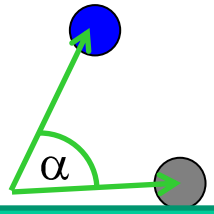
- Any other distance measure for numeric variables can be employed
  - Manhattan/City Block/L1
  - *(Generalised: Minkowski / L_n)*



  - Linfinity



  - Cosine



k-NN vs. Perceptron

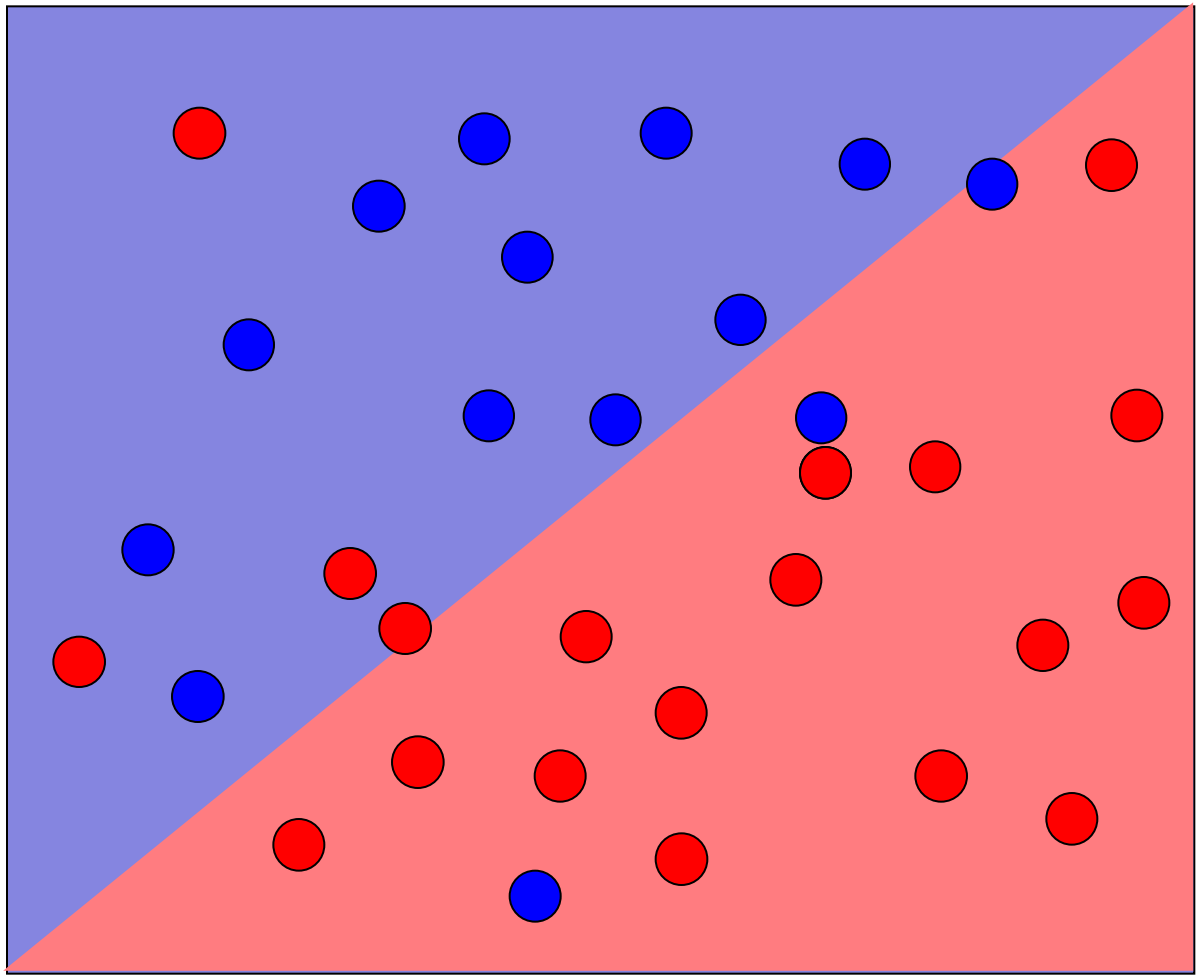
- Major differences between k-NN and perceptron algorithms?

- ***k-NN is a "Lazy learner"***
  - no model built beforehand
    ➔ computation is done at classification step
  - Opposite is called "eager learning"

- k-NN can learn complex (almost arbitrary) **decision boundaries**

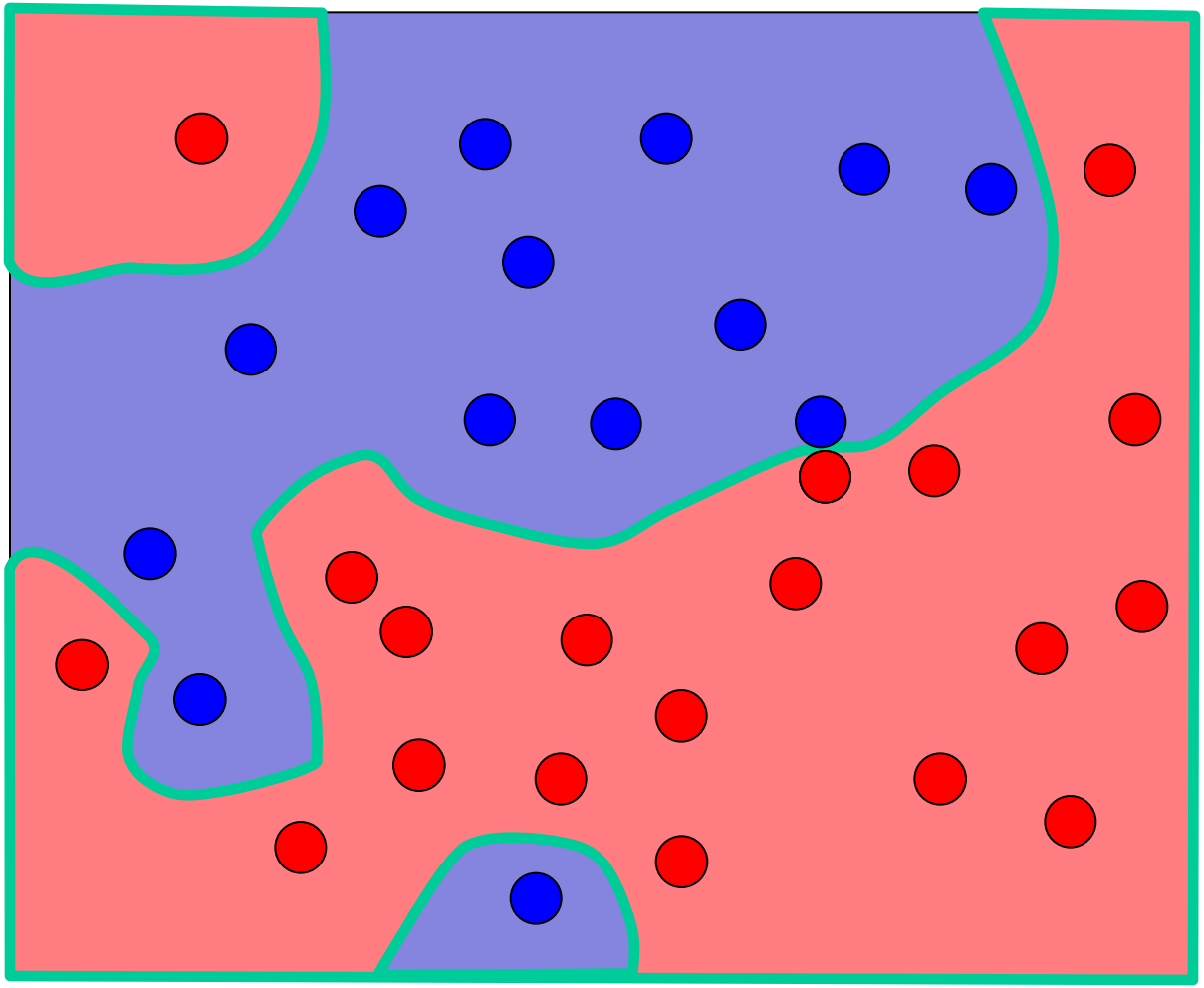
Perceptron: decision boundary

- *Only linear separation!*



k-nn: decision boundary

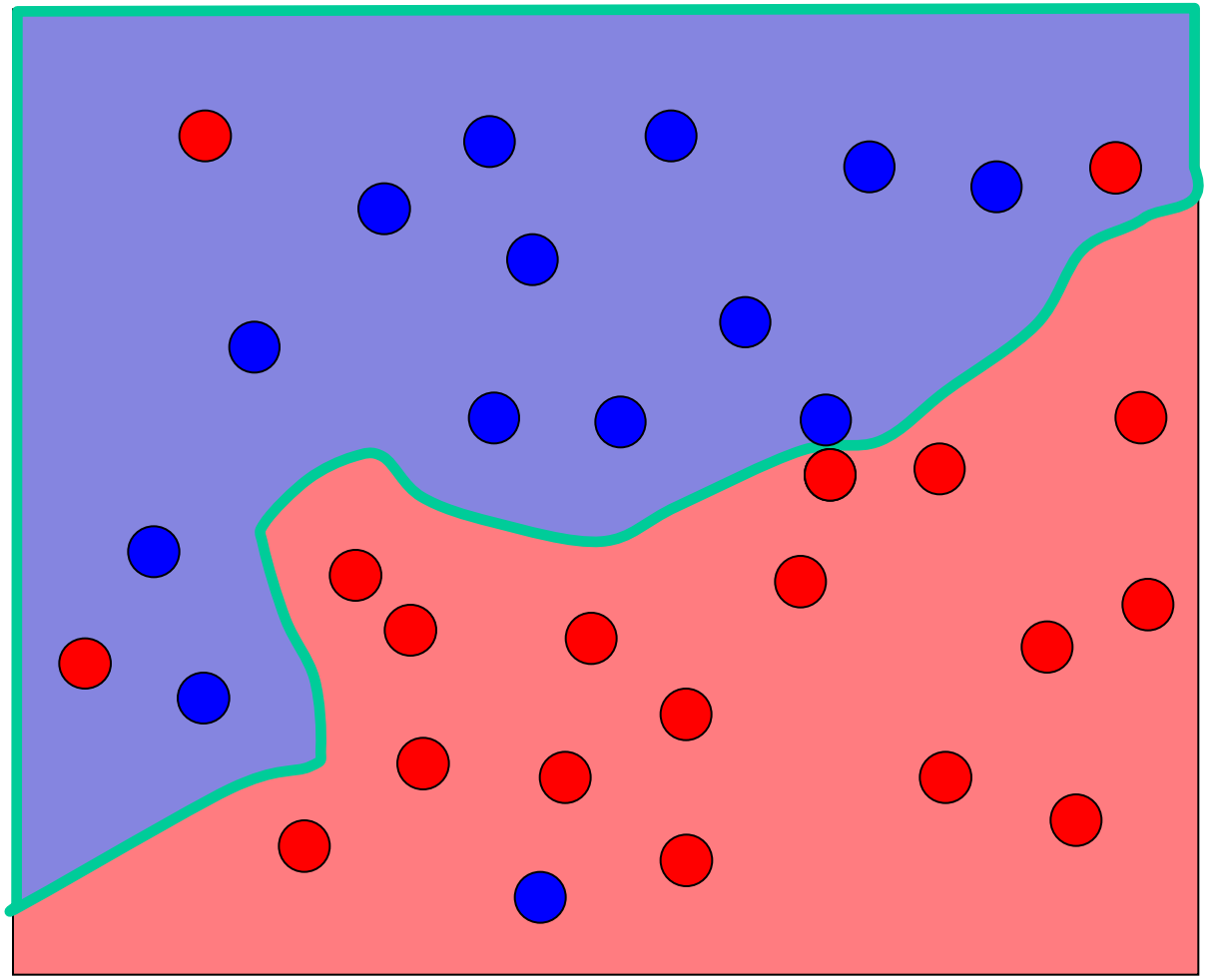
- *(Almost) Arbitrary boundary*



k = 1

k-nn: decision boundary

- *(Almost) Arbitrary boundary*



k = 3

k-nn: Properties

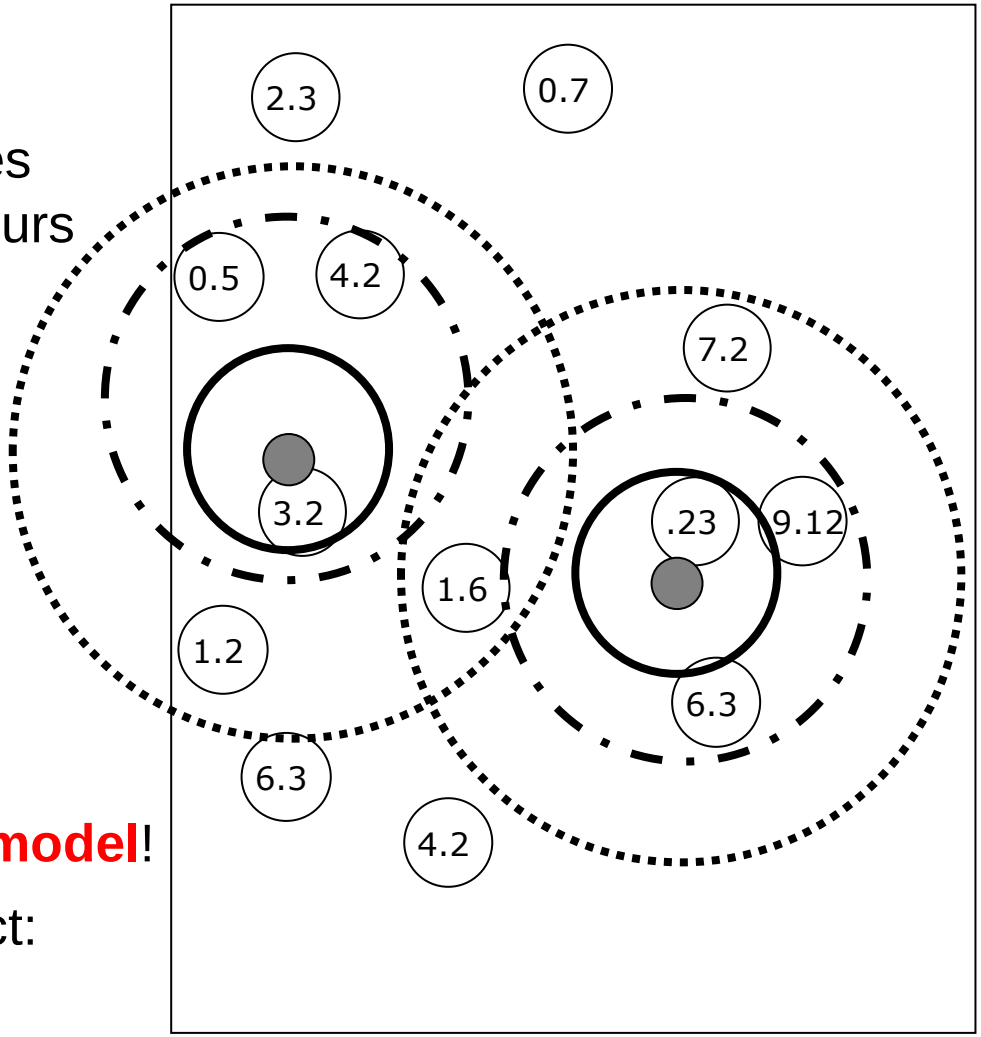
- Easy to understand and interpret
- Easy to implement
- No training time needed…

- Memory requirements depend on training data
- Becomes computationally expensive with many items to classify
  - Linear search: O (Nd), N = # samples, d = dimension
  - Several optimisations proposed

- *More details next lecture*

k-nn: regression

- *Can k-NN be used for regression?*
- How ?
  - Average of continuous values assigned to nearest neighbours
    - k=1: 0.23; k=3: 5.22; k=5: 4.89
  - Potentially weighted (rank, distance, ...)



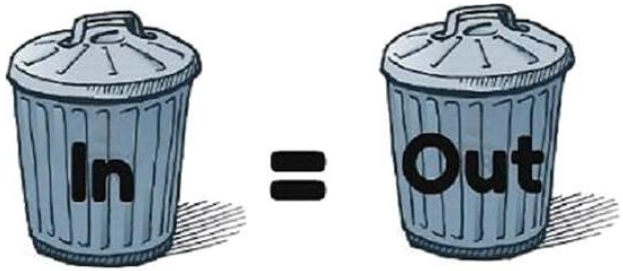
- N.b.: lazy learner:
  - we don't build a regression **model**!
  - For each value that to predict: do neighbour search !

Outline

- Introduction
- Perceptron
- k-NN
- Types of data & data preparation (intro)
- Performance evaluation (intro)

Data Preparation

- Vital step for machine learning (supervised and unsupervised)
- ML algorithm will ***always*** give you a model
  - Quality of that model depends highly on the quality of the input data
- "Garbage in" ➔ "Garbage out"



- One major goal of data preparation:
  - Eliminate "wrong influence" of variables

Data Types

- Possible attribute/feature/variable types (*"levels of measurement"*)
  - Nominal, ordinal, interval and ratio

- Nominal quantities
  - Values are distinct symbols
  - Example: attribute "eye colour"
    - Values themselves serve only as labels or names
    - Nominal comes from the Latin word for name
    - Values: "green","grey", "brown" and "blue"
  - No relation is implied among nominal values (no ordering or distance measure)
  - Only (in)equality tests can be performed

Data Types

- Ordinal quantities
  - Impose order (rank) on values
  - (In)equality, additionally comparison (<, >) and median
  - But: no distance between values defined
  - Example:
    - attribute “size” in animals data
    - Values: “large” > “medium” > “small”
    - Note: addition and subtraction don’t make sense

- Distinction between nominal and ordinal not always clear

Data Types

- Interval quantities
  - Interval quantities are not only ordered but measured in fixed and equal units
  - Degree of difference between measurements
  - Example 1: temperature
  - Example 2: date
  - Difference of two values makes sense
  - Ratio not (20 is not twice as hot as 10)
  - Sum or product doesn't make sense

Data Types

- Ratio quantities
  - The measurement scheme defines a zero point
  - Example: distance, mass, length,
  - Distance between an object and itself is zero
  - All mathematical operations are allowed
  - Meaningful to have ratios: “twice as long”

Data Preparation

gender	age	height	smoker	eye colour
male	19	170	yes	green
female	44	162	yes	grey
male	49	185	yes	blue
male	12	178	no	brown
female	37	165	no	brown
female	60	157	no	brown
male	44	190	no	blue
female	27	178	yes	brown
female	51	162	yes	green
female	81	168	yes	grey
male	22	184	yes	brown
male	29	176	no	blue

- Example data set from earlier (lung cancer)

- Potential issues ?
  - Missing values
  - Quantitative (continuous) data with different scales
  - Data type not fitting (categorical vs numerical) data

Scaling / Normalisation / Standardisation

- Different variables may exhibit significantly different value ranges
  - E.g. a length variable measured in cm, inch, or meters
  - Different types of measurements: length, speed, temperature, …
  - Different types of measuring devices capturing different value ranges
  - …
  - *Why is this a (potential) problem?*

Scaling / Normalisation / Standardisation

- Some ML algorithms rely on *measuring the (numeric) distance* between samples

- ➔ There should be no impact by the value range

  – higher values in one attribute / variable would have unproportional effect on measure distance
    ➔ would dominate distance metric
    ➔ might thus dominate learning

Scaling / Normalisation / Standardisation

- Remove effects of different value ranges in each attribute/variable

- Common method in statistics and machine learning
  - With many different notations
  - Scaling, Normalisation, Standardisation, ...
  - Often used interchangely, different for each field ...

Scaling / Normalisation / Standardisation

- Z-score standardisation / normalization

- Each variable *x* is converted to have a mean of 0 and a standard deviation of 1
  - subtracting the mean
  - dividing by standard deviation

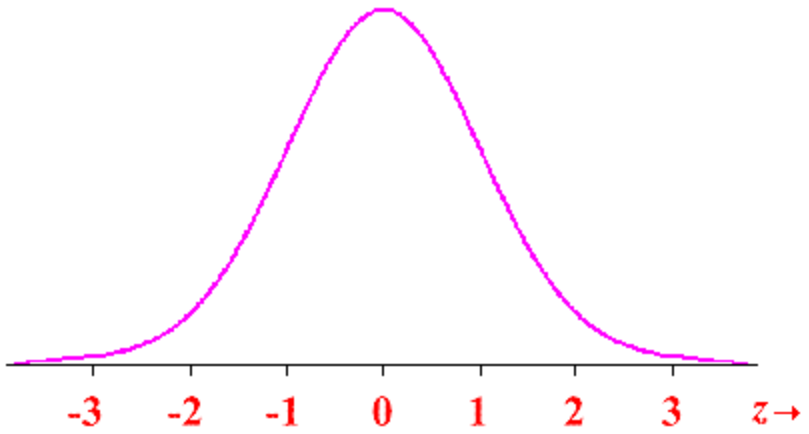
$$z_i = \frac{x_i - \mu}{\sigma}$$

Scaling / Normalisation / Standardisation

- Z-score standardisation / normalization

$$z_i = \frac{x_i - \mu}{\sigma}$$

- *What's the new value range after z-score?*



Scaling / Normalisation / Standardisation

- Min-Max scaling
  - Scale all variables to the same (fixed) range
  - Often between 0 and 1
  - Subtract minimum value for each variable
  - Divide by value range of each variable

$$z_i = \frac{x_i - \min(X)}{\max(X) - \min(X)}$$

  - *Multiply by new range (if different than 0..1)*

Scaling

- *Is scaling an issue for*
  - *k-nn?*
  - *Perceptron?*

Scaling / Normalisation / Standardisation

- Especially algorithms relying on distances
  - k-Nearest Neighbours
  - …
- Scaling not needed for algorithms that don't use distances, e.g.
  - Naïve Bayes
  - Decision trees
  - …

Scaling / Normalisation / Standardisation

- Especially algorithms relying on distances
  - k-Nearest Neighbours
  - ...

- *Caveat*
  - Many implementations already do this pre-processing implicitly
  - Check default settings carefully

Other data preparation aspects

- 1-hot encoding – special data types

- Missing values
  - For some samples, not all attribute values are known
    - Some ML algorithms can handle missing values
    - For others, we need a special treatment: data imputation

➔ *More in this in upcoming lectures*

Outline

- Introduction
- Perceptron
- k-NN
- Types of data & data preparation (intro)
- Performance evaluation (intro)

Evaluation

- When we use a machine learning model, we want to know how good it is (effectiveness)
  - To know how confident we can be in the predictions
  - To know which algorithm to use
  - ....
- ➔ *Model validation*

- Need to measure performance of an algorithm
  - Test on (labelled) data
  - Several different measures

- Orthogonal topic: efficiency, i.e. required runtime
  - *More on that later*

Evaluation (effectiveness)

- Binary classification (classes true/false)
- Table of confusion *(contingency table)*

		Actual value		
		true	false	
Test outcome	true	True positive (TP)	False positive (FP, Type I error)	***Precision*** $\frac{TP}{TP + FP}$
	false	False negative (FN, Type II error)	True negative (TN)	
		Recall Sensitivity $\frac{TP}{TP + FN}$		***Accur-acy*** $\frac{TP + TN}{\#\ samples}$

- *Examples of Type I & II errors?*
  - *Which is worse?*

Evaluation measures

	true	false
true	True positive (TP)	False positive (FP, Type I error)
false	False negative (FN, Type II error)	True negative (TN)

- Accuracy: # correctly predicted samples

$$\frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{\#\ samples}$$

- Precision $\frac{TP}{TP + FP}$ ➔ *How to optimise?*
- Recall $\frac{TP}{TP + FN}$ ➔ *How to optimise?*
- *More measures in later lectures ...*

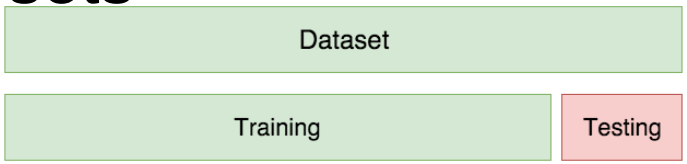
Evaluation: data

- Which data to do evaluation on?
- *The samples used for training?*
  - *Why not?*
  - Would not tell us how good the model works for unknown data
    - Which is however why we train a model in first place ...
  - *If we test on training data – we are biased*
    - Fully grown decision trees – will be 100% on training data
    - Perceptron on linear separable data: training data will be 100% correctly “predicted”
    - K-NN, Naïve Bayes: not necessarily 100% correct on training data. Still biased!
    - Similar for other algorithms, e.g. SVMs, ...
  - *Want to actually find out:* how will model perform on **unseen** data!

Training & Test set

- Really unseen data doesn't have labels
  - "Simulate" "unseen" data
  - ➔ Holdout method: keep part of your labelled data for testing

- Split data into training and test sets
  - E.g. ~80% training, 20% test
    - 66% - 33%
  - Sorted (linear), random, …

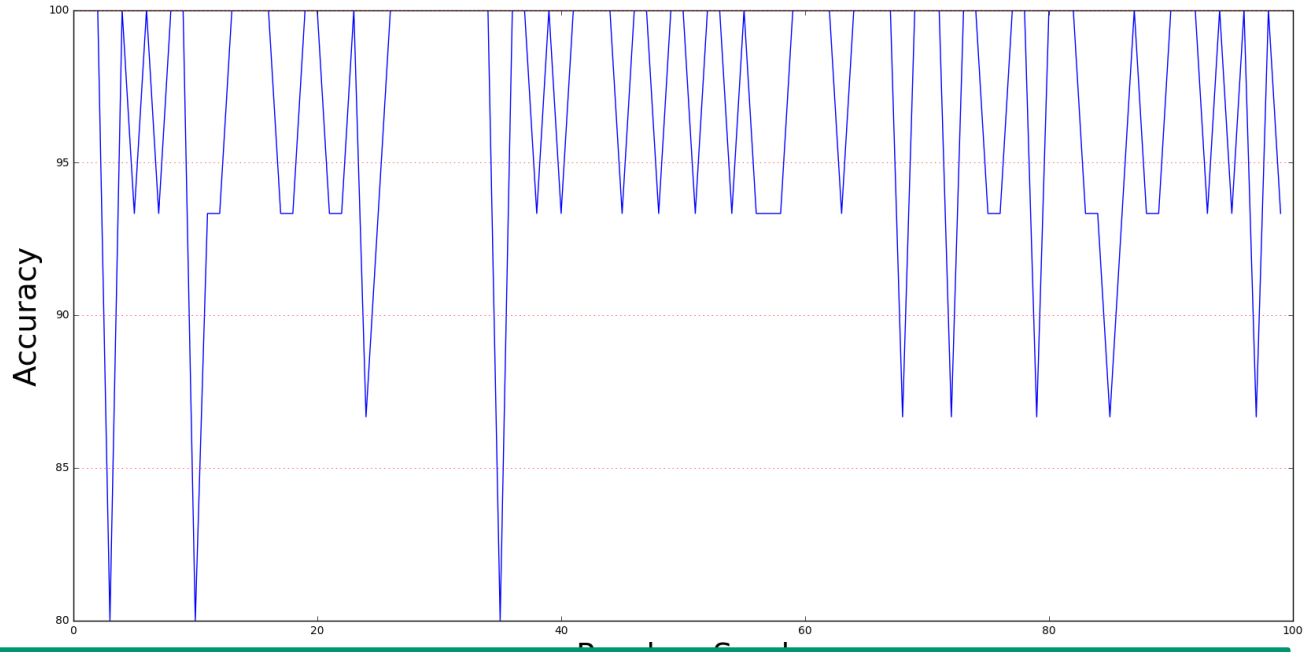


- Performance on test set is an estimate for generalisation ability of our model

- Results can vary a lot according to how the split is done

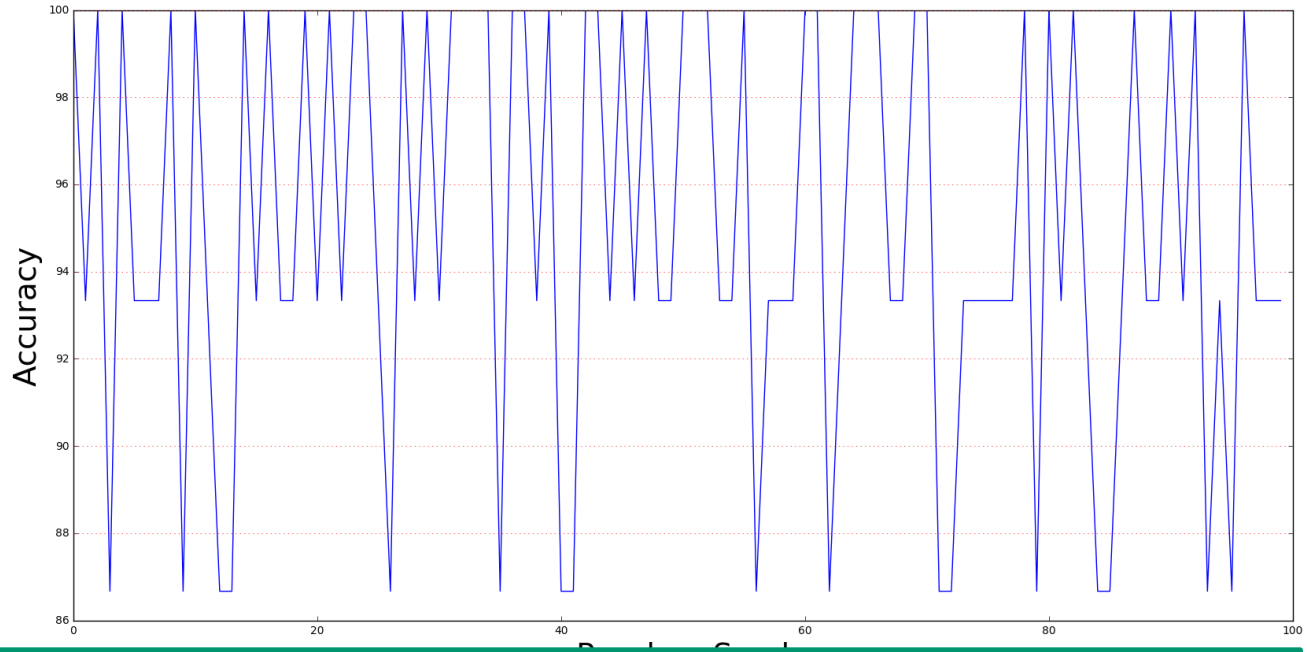
Holdout method

- Random influence in training/test split
  - Example: **Iris Data, *k-nn (k=15)***, random seed 1..100
    - Average Accuracy: 96.67 (+/- 9.8883) [range: 80-100]



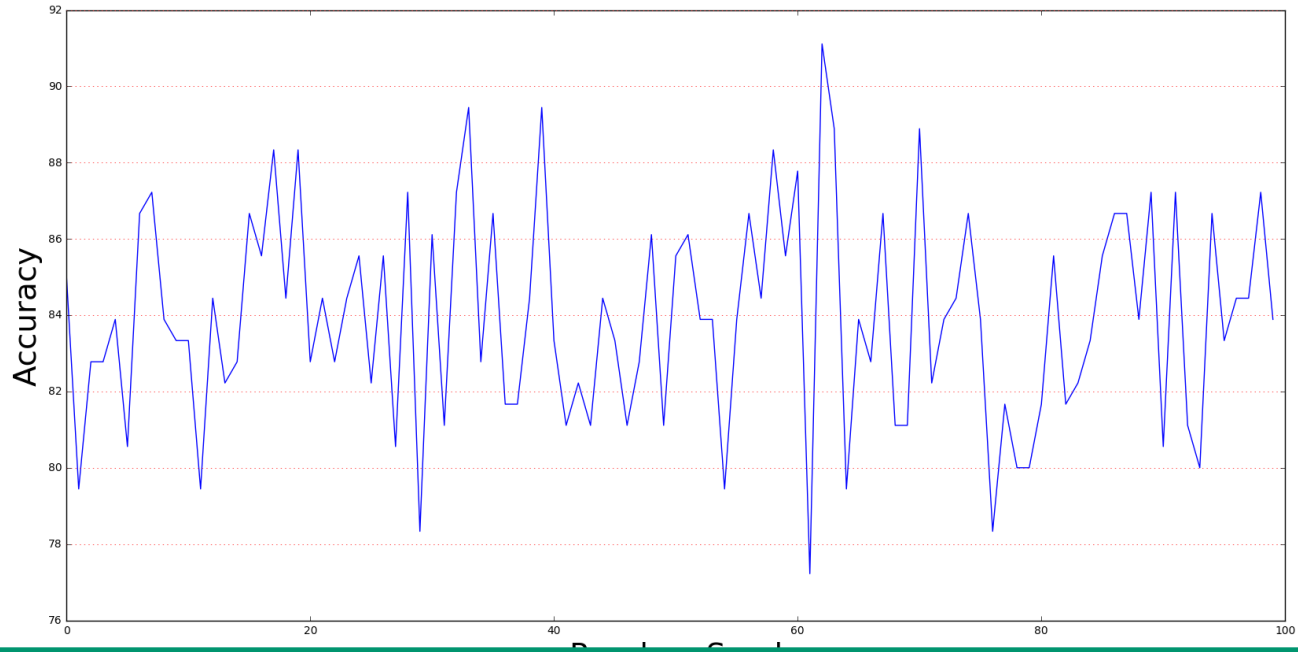
Holdout method

- Random influence in training/test split
  - Example: **Iris Data, *Naive Bayes***, random seed 1..100
    - Average Accuracy: 95.00 (+/- 9.6839) [range: 86.67-100]



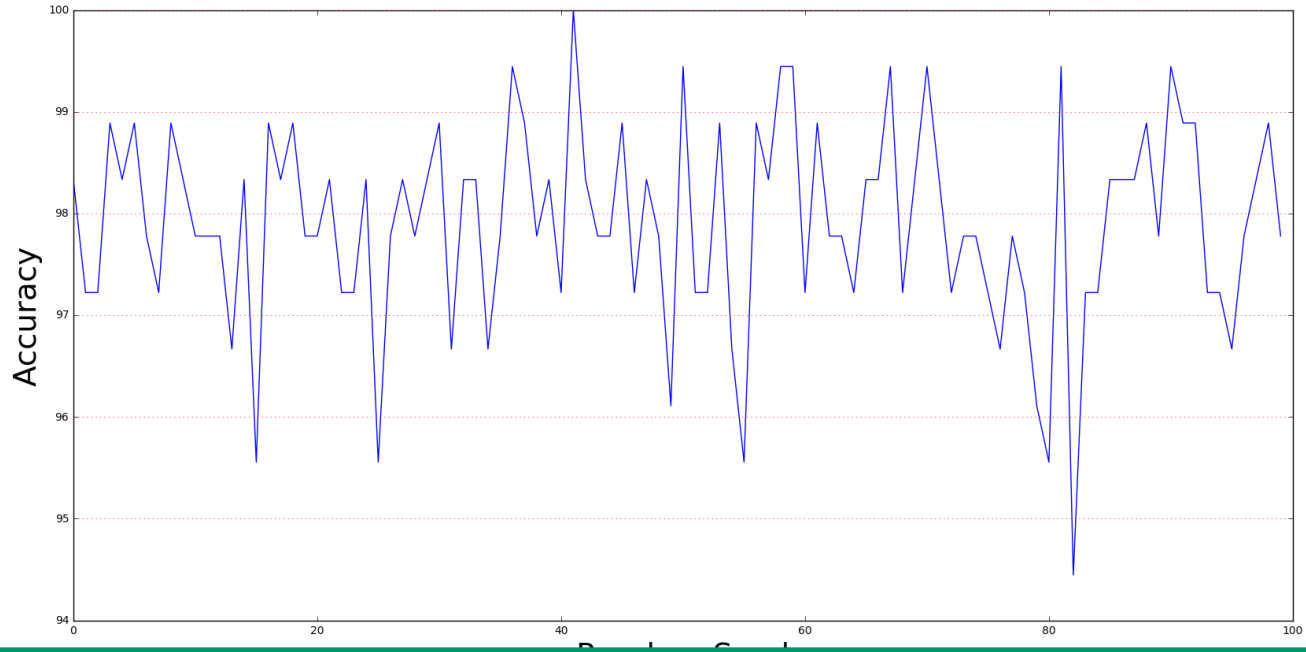
Holdout method

- Random influence in training/test split
  - Example: **Digit Data, *k-nn (k=15)***, random seed 1..100
    - Average Accuracy: 97.90 (+/- 2.0097) [range: 94.44-100]



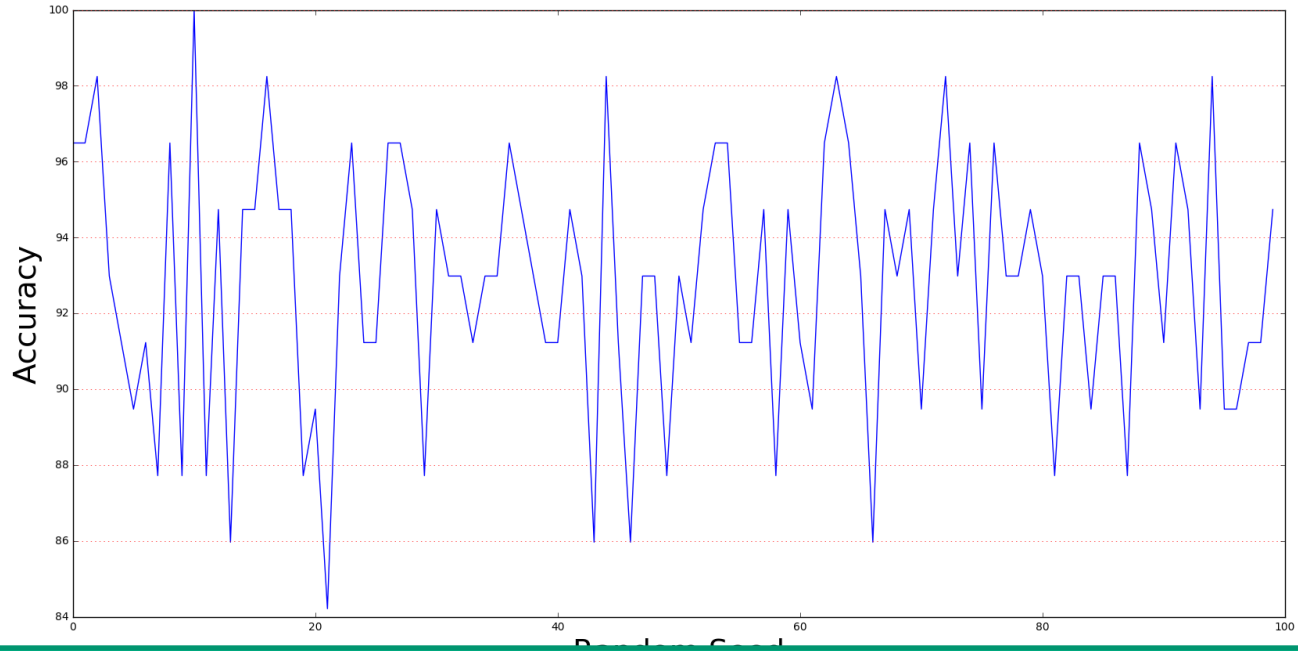
Holdout method

- Random influence in training/test split
  - Example: **Digit Data, *Naive Bayes***, random seed 1..100
    - Average Accuracy: 83.93 (+/- 5.6167) [range: 77.22-91.11]



Holdout method

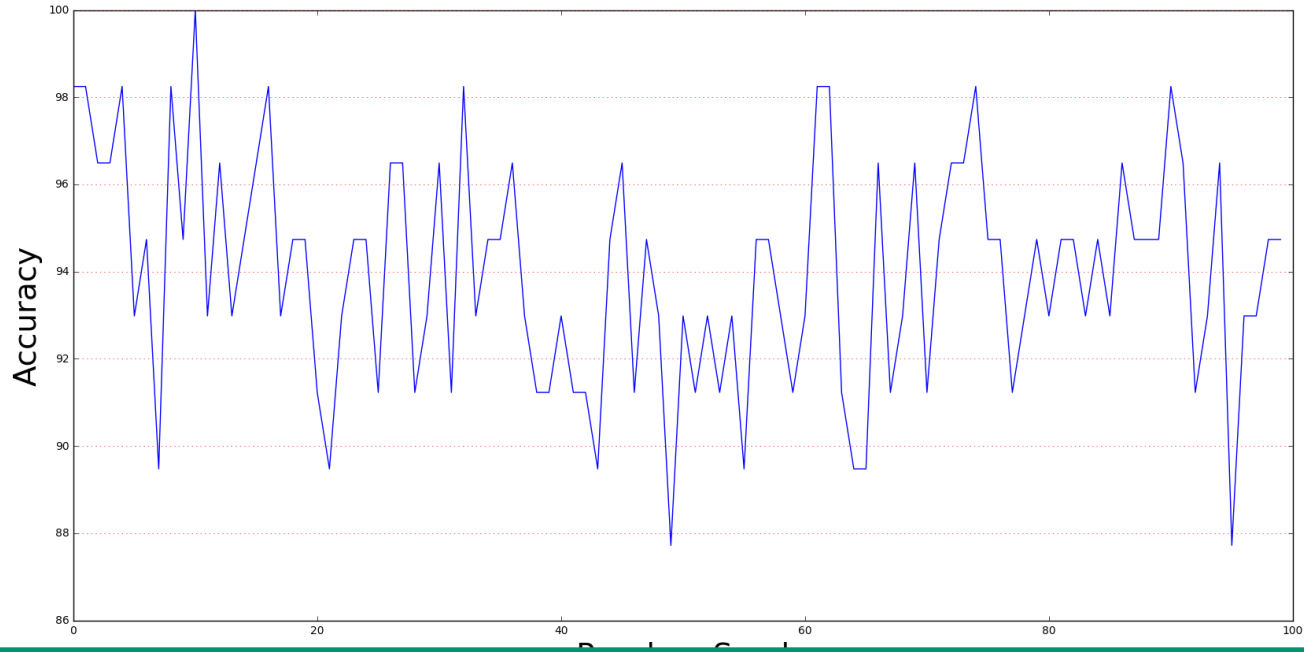
- Random influence in training/test split
  - Example: **Cancer Data, *k-nn (k=15)***, random seed 1..100
    - Average Accuracy: 92.81 (+/- 6.7401) [range: 84.21-100]



Holdout method

- Random influence in training/test split
  - Example: **Cancer Data, *Naive Bayes***, random seed 1..100
    - Average Accuracy: 93.96 (+/- 5.2355) [range: 87.719-100]

- Solution?



k-fold Cross validation

- Split data into e.g. 10 parts of equal sizes
  - This is called 10-fold cross validation
- Repeat 10 times:
  - use 9 parts for training (training set)
  - calculate performance on remaining part (test set)

- Estimate of performance is average (mean) of the validation set performances

k-fold Cross validation

- Estimate of performance is average (mean)
- In addition to mean, compute standard deviation
  - Indication on how stable the results are in the folds

    ➔ lower standard deviation is better ...

  - Standard deviation to be considered when comparing cross-validation performances from different classifiers

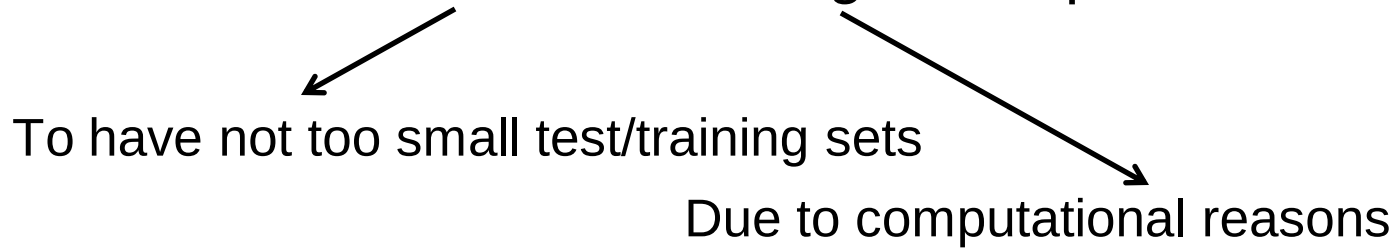
Classifier / Fold	1	2
1	91,80	86,7
2	82,30	87
3	84,40	87,1
4	93,00	85,7
5	81,60	86,8
6	87,40	86,4
7	82,40	87,2
8	92,10	86,5
9	91,90	86,5
10	87,40	86,5
Mean	87,4	86,6
Stdv	4,6	0,4

k-fold Cross validation

- Which classifier is better:
  - Average 87,4%, standard deviation 4,8%
    - *(87,4% 4,8)*
  - Average 86,6%, standard deviation 0,4%
    - *(86,6% 0,4)*
    - ➔ *More on that later: significance testing*

k-fold Cross validation

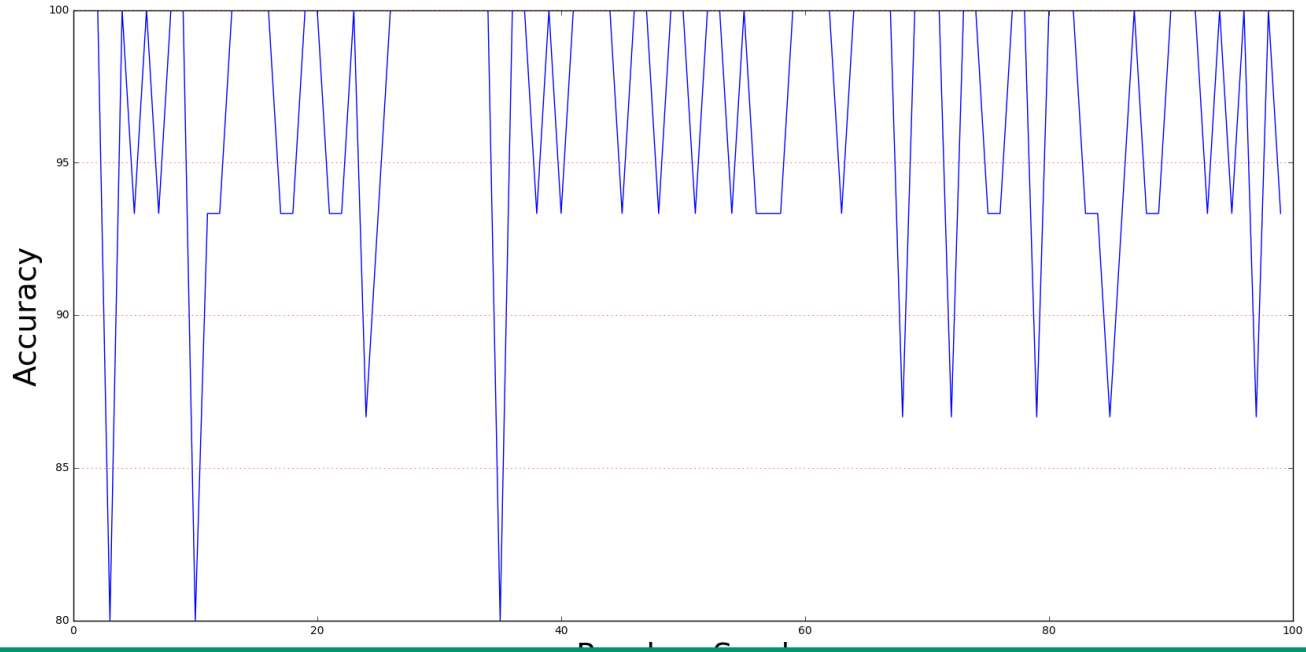
- Results obtained via cross-validation are generally much more reliable
  - Parameter of 10 often used
  - Fewer folds on *smaller and larger* sample size



- Number of folds increases runtime!
  - More-or-less linear with ***n***
  - Might not be that critical – why?
    - Can be parallelised

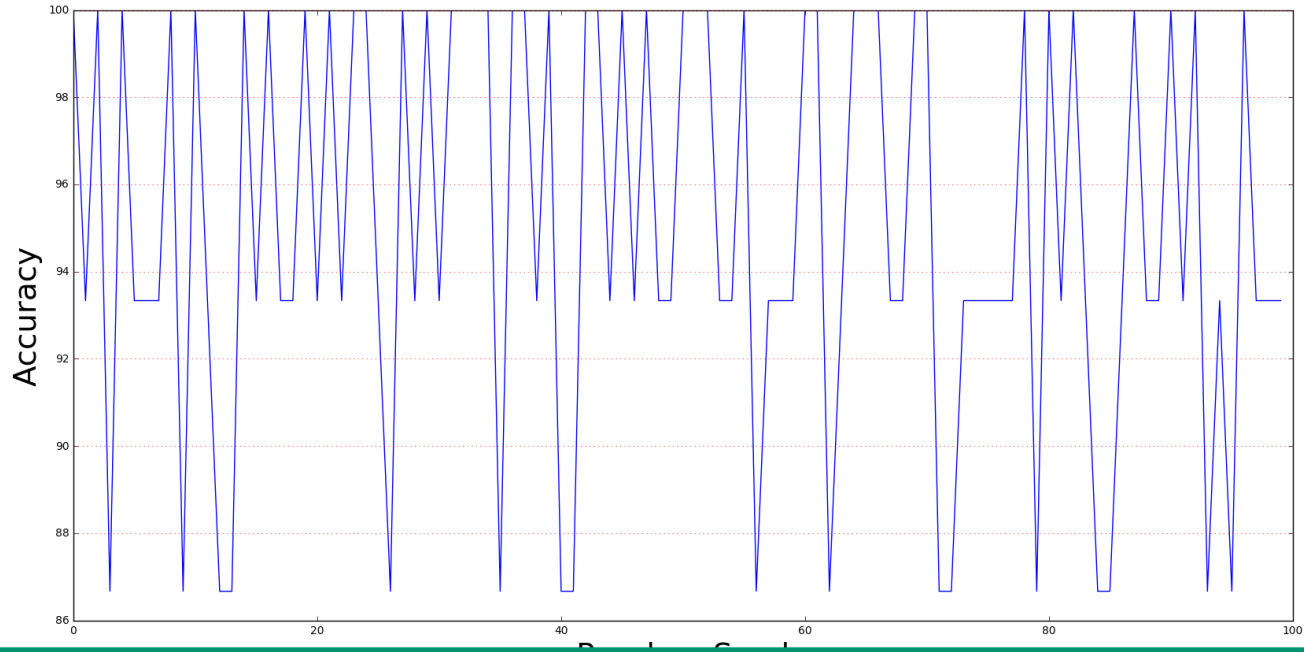
Cross validation

- Cross-validation
  - Smooth effects of random influence in training/test split
  - Example: **Iris Data, *k-nn (k=15)***, random seed 1..100
    - Average Accuracy: 96.67 (+/- 9.8883) [range: 80-100]
    - 10-fold CV: 96.67 (+/- 8.94)



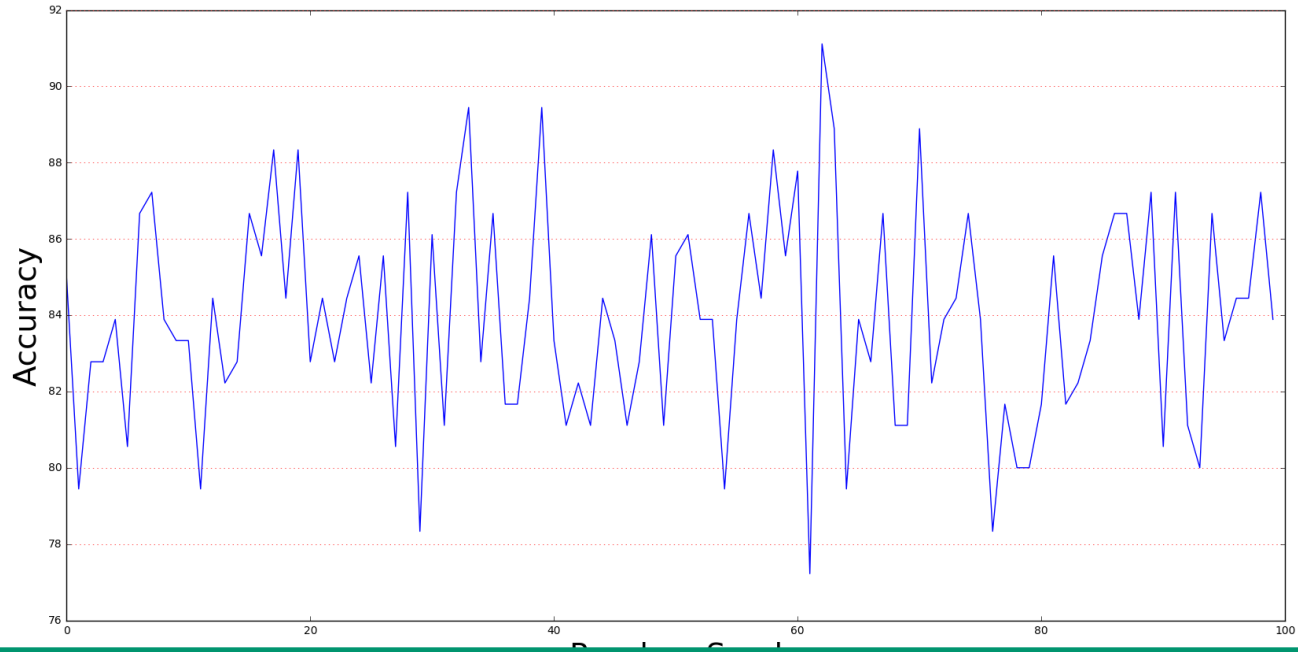
Cross validation

- Cross-validation
  - Smooth effects of random influence in training/test split
  - Example: **Iris Data, *Naive Bayes***, random seed 1..100
    - Average Accuracy: 95.00 (+/- 9.6839) [range: 86.67-100]
    - 10-fold CV: 95.33 (+/- 12.00)



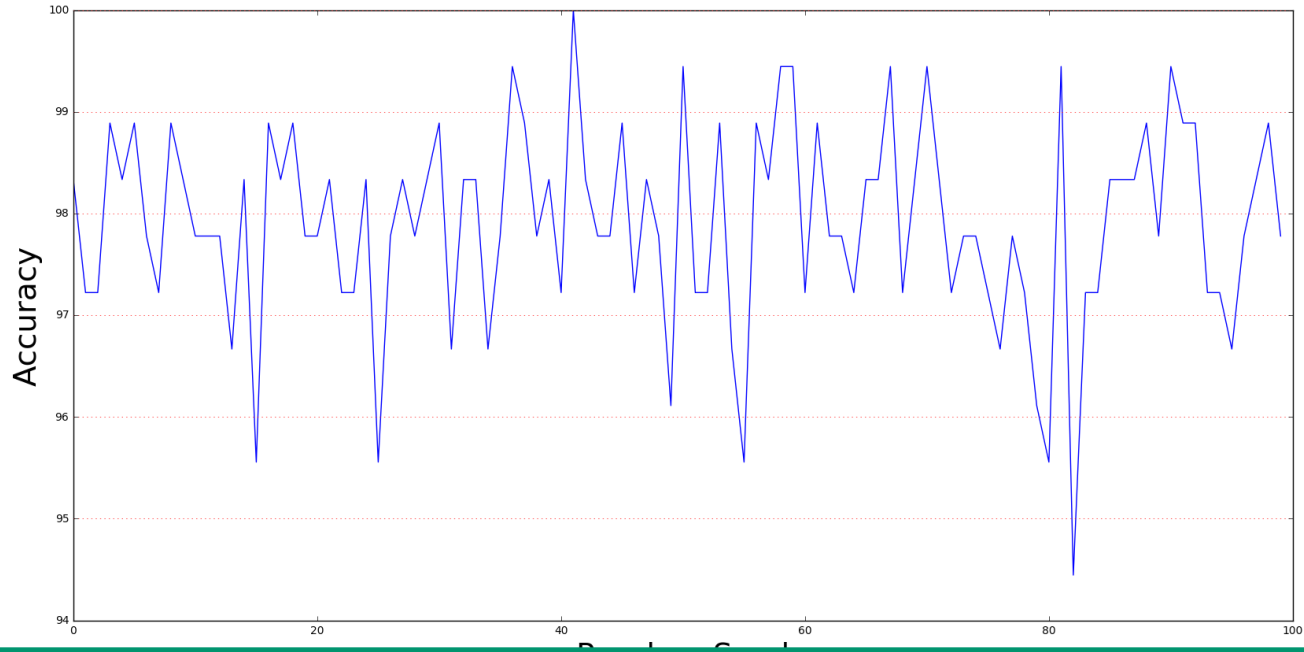
Cross validation

- Cross-validation
  - Smooth effects of random influence in training/test split
  - Example: **Digit Data, *k-nn (k=15)***, random seed 1..100
    - Average Accuracy: 97.90 (+/- 2.0097) [range: 94.44-100]
    - 10-fold CV: 97.99 (+/- 1.96)



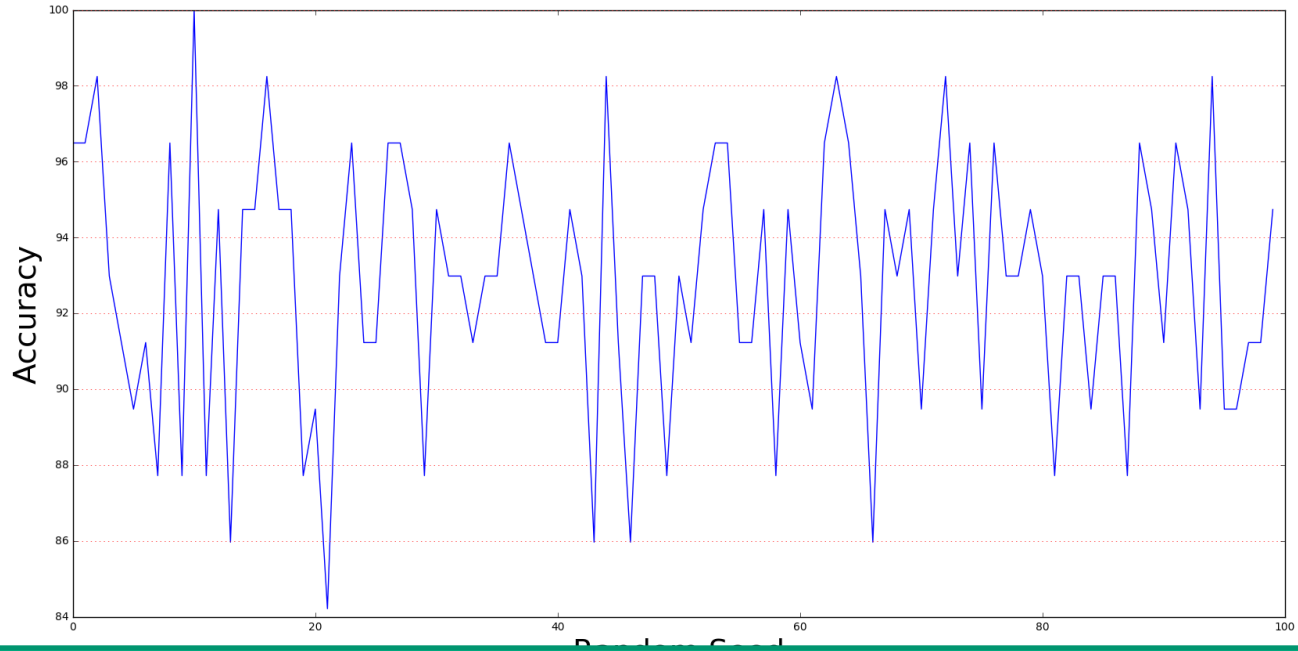
Cross validation

- Cross-validation
  - Smooth effects of random influence in training/test split
  - Example: **Digit Data, *Naive Bayes***, random seed 1..100
    - Average Accuracy: 83.93 (+/- 5.6167) [range: 77.22-91.11]
    - 10-fold CV: 84.03 (+/- 8.02)



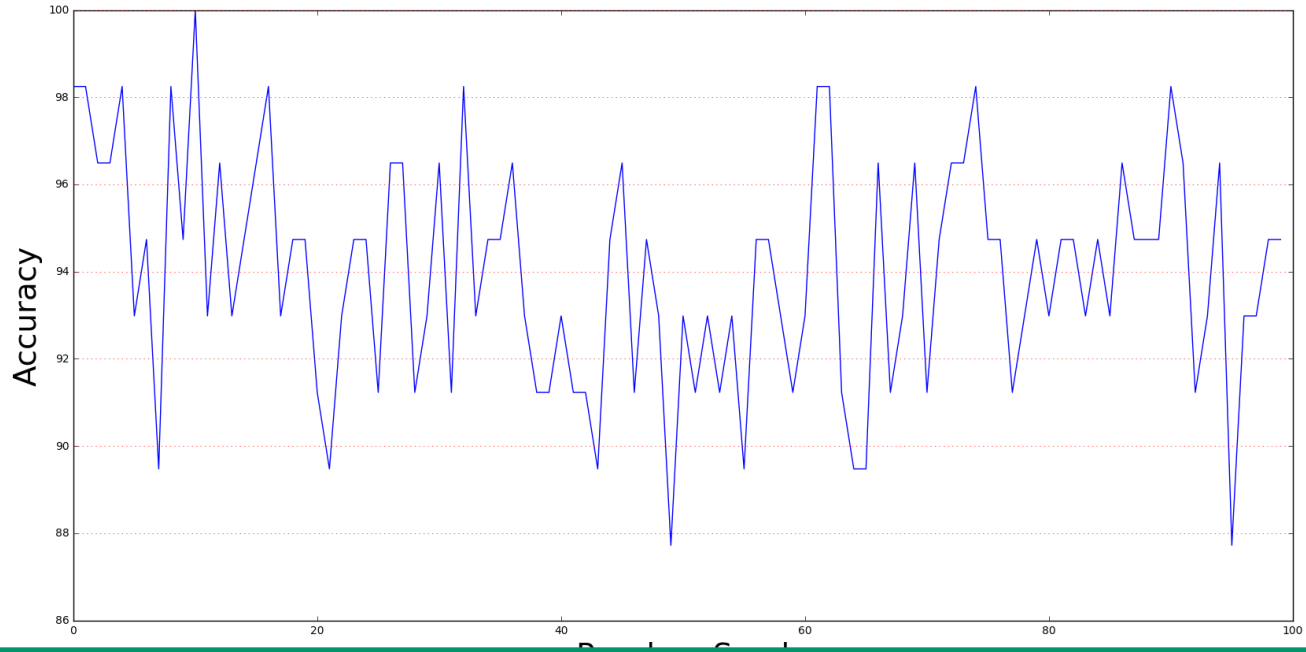
Cross validation

- Cross-validation
  - Smooth effects of random influence in training/test split
  - Example: **Cancer Data, *k-nn (k=15)***, random seed 1..100
    - Average Accuracy: 92.81 (+/- 6.7401) [range: 84.21-100]
    - 10-fold CV: 93.29 (+/- 8.1)



Cross validation

- Cross-validation
  - Smooth effects of random influence in training/test split
  - Example: **Cancer Data, *Naive Bayes***, random seed 1..100
    - Average Accuracy: 93.96 (+/- 5.2355) [range: 87.719-100]
    - 10-fold CV: 93.83 (+/- 4.33)



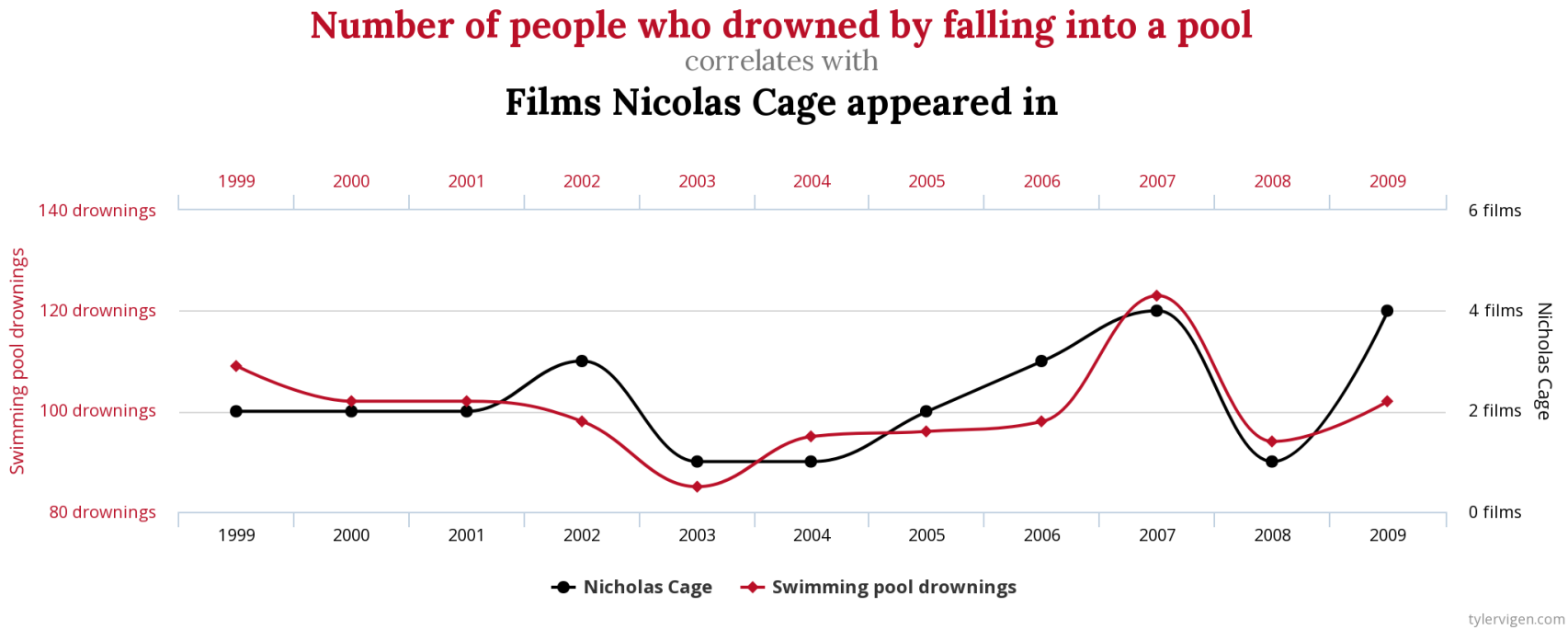
Evaluation: outlook

- Micro vs. macro averaging
  - Confusion matrix
  - Cost functions
- Other evaluation measures
- ROC curves
- Bootstrapping
- Significance testing

- Evaluation measures for regression

Meaningful?

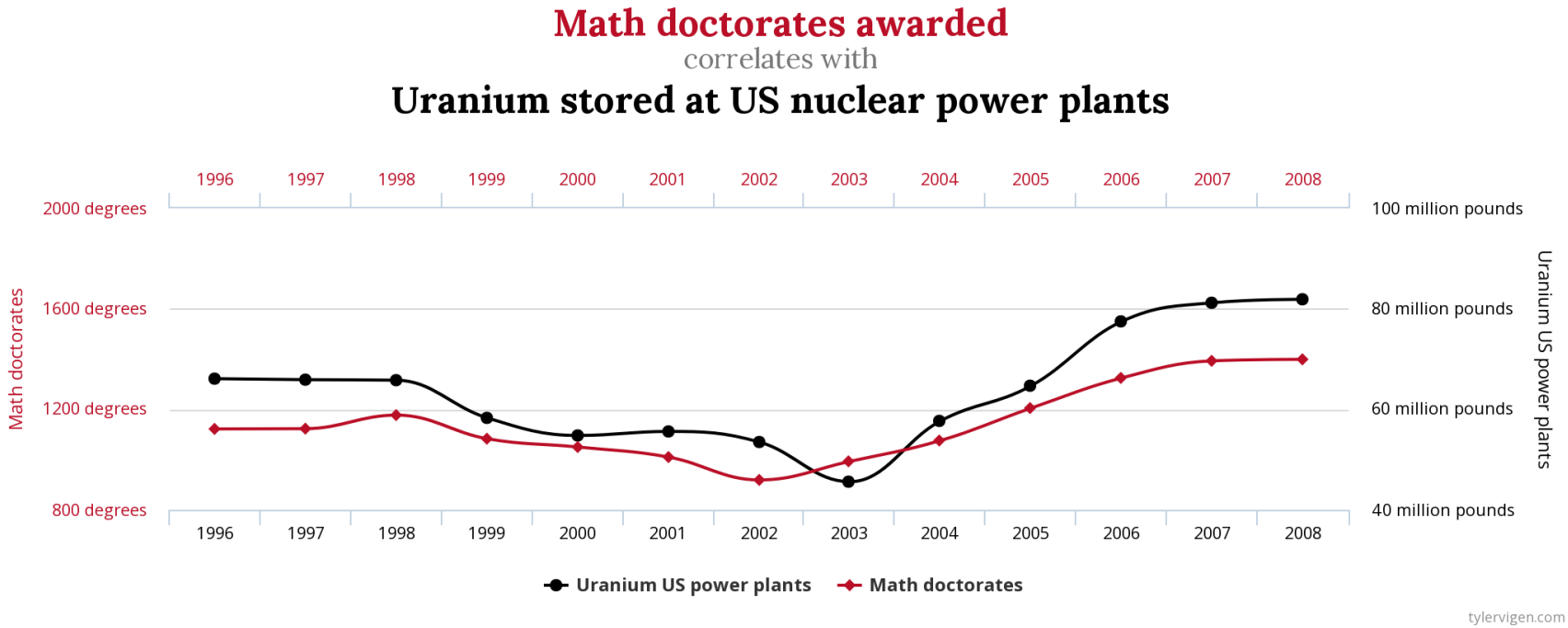
- Not every model is actually use/meaningful ☺



http://www.tylervigen.com/spurious-correlations

Meaningful?

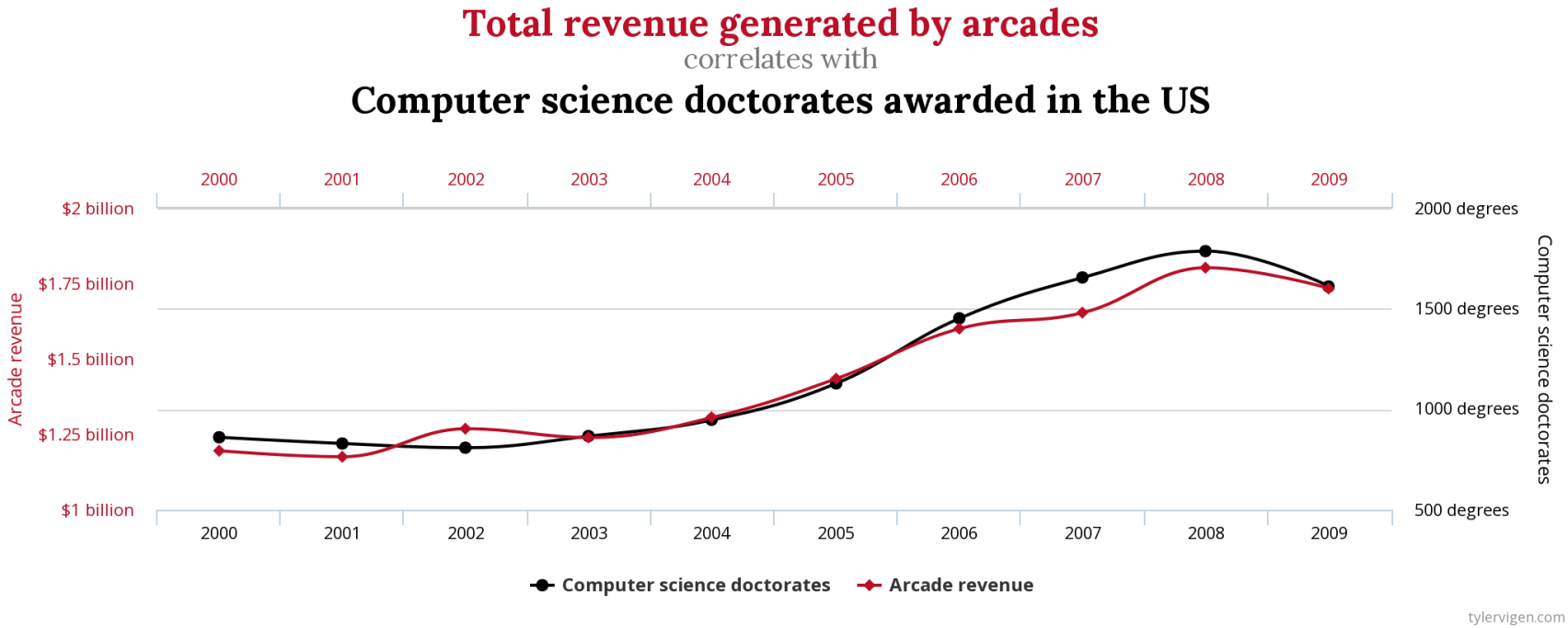
- Not every model is actually use/meaningful ☺



http://www.tylervigen.com/spurious-correlations

Meaningful?

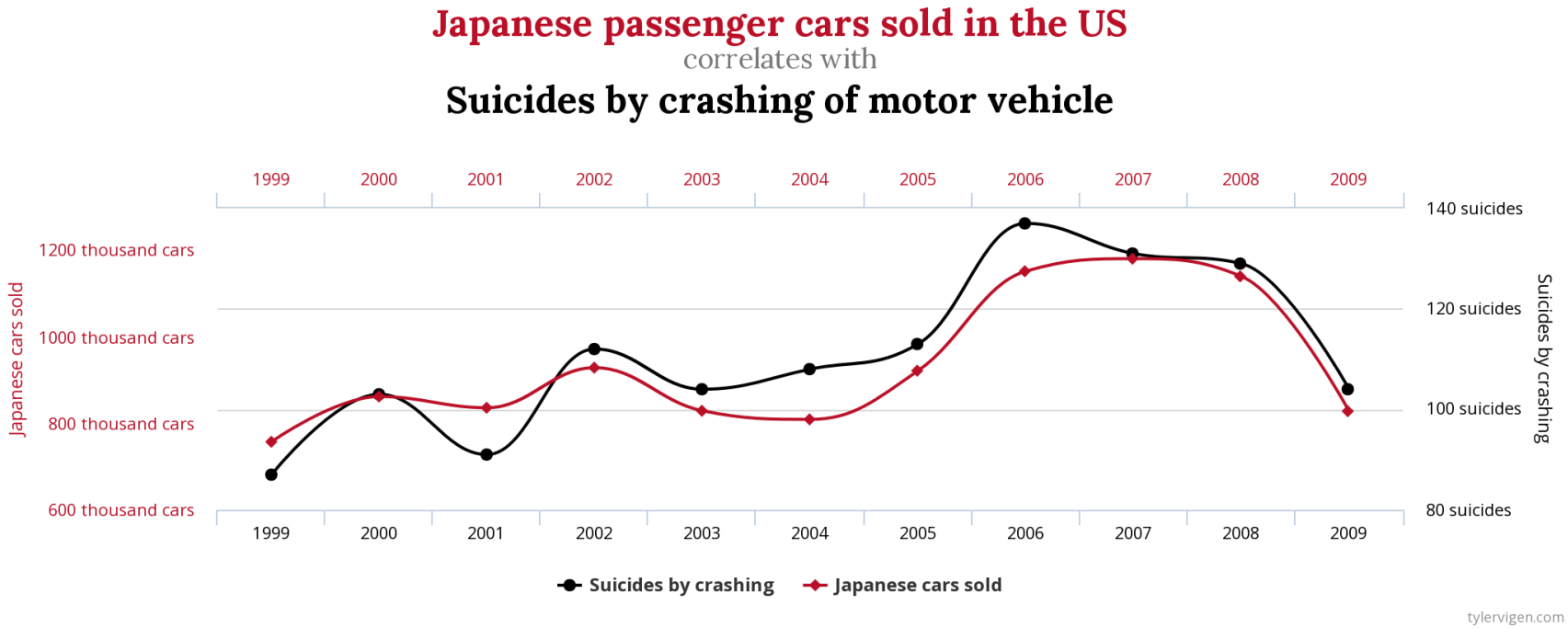
- Not every model is actually use/meaningful ☺



http://www.tylervigen.com/spurious-correlations

Meaningful?

- Not every model is actually use/meaningful ☺



http://www.tylervigen.com/spurious-correlations

Questions?

Upcoming topics:

- Decision trees
- Random Forests
- Support Vector Machines