Principles of Data Science

Prof. Tias Guns
tias.guns@kuleuven.be

DTAI lab, KU Leuven
Data Analytics lab, VUB

Virtual Summer School “Combinatorial Optimization, Constraint Programming and Machine Learning”, Nov 2020

General outline

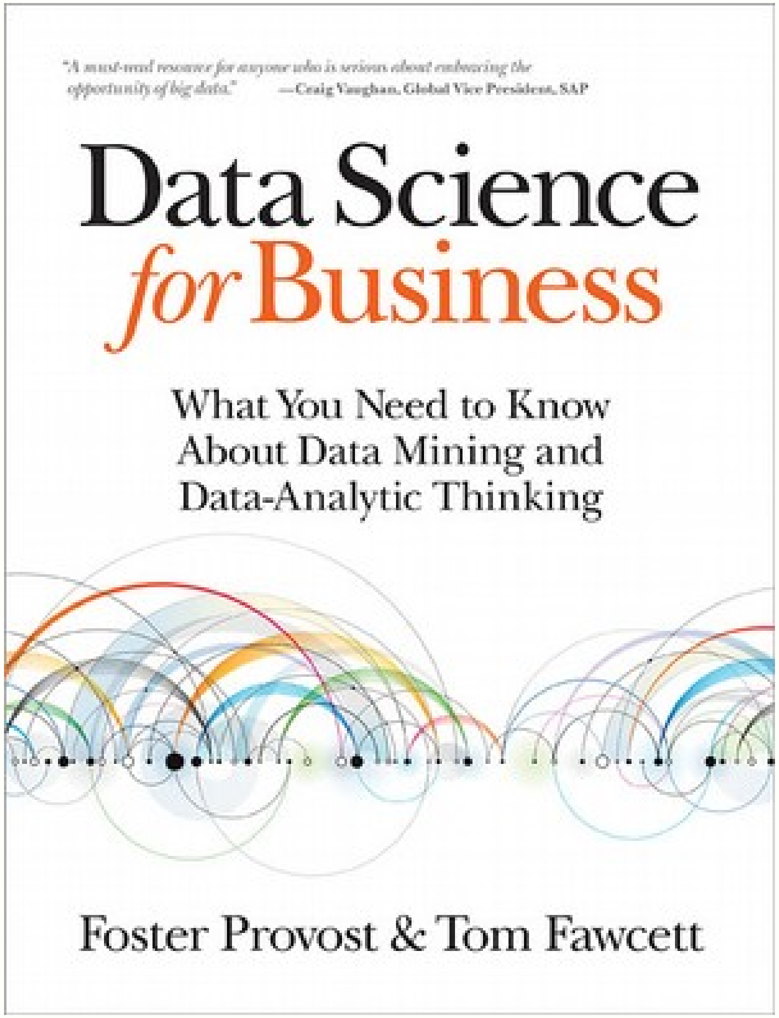
This session:

- ~60 min Principles of Data Science
- ~30 min Learning from user in CP
- ~30 min Practical (python notebook)

Afternoon session:

- ~20 min Learning from vision in CP
- ~40 min Learning from environment in CP
- ~30 min Practical (other python notebook)

Sources



Includes slides from
P. Adamopoulos (NYU) and
E. Ricci (Perugia)

and colleagues and
collaborators, including:

Hendrik Blockeel and
Luc De Raedt, KU Leuven

Paulo Frasconi, Uni Florence

Wouter Verbeke, VUB

What is data science?

“The previous new hype?”

Statistics...

Big data...

Machine Learning...

Data Analytics...

Data Mining...

Deep Learning...

Prescriptive analytics...

A.I...

Data Science...

What is Data Science?

No single definition

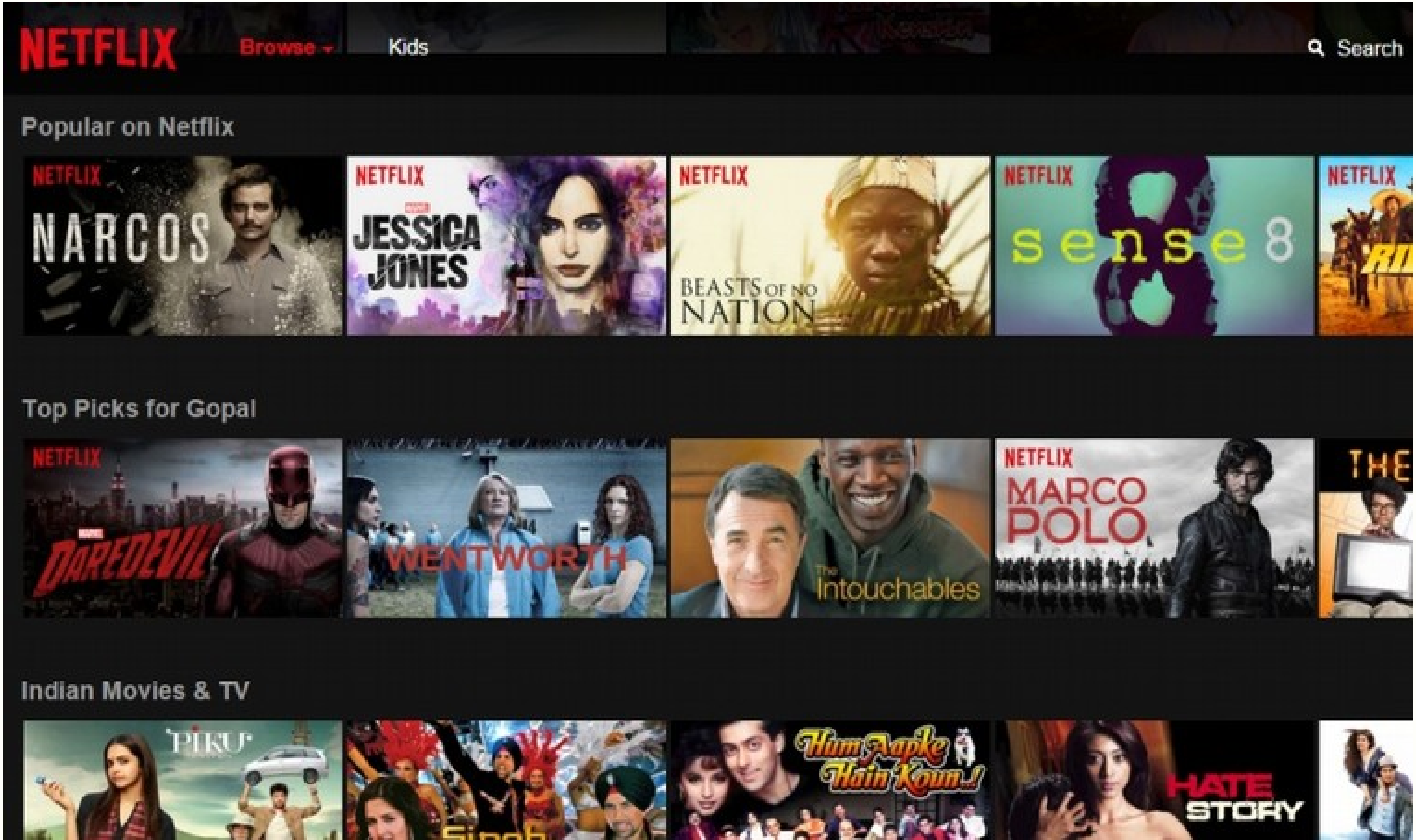
Components:

- Data-driven (the more the better: big data)
- Interdisciplinary (math, stat, CS, ...)
- Extract knowledge from observed data

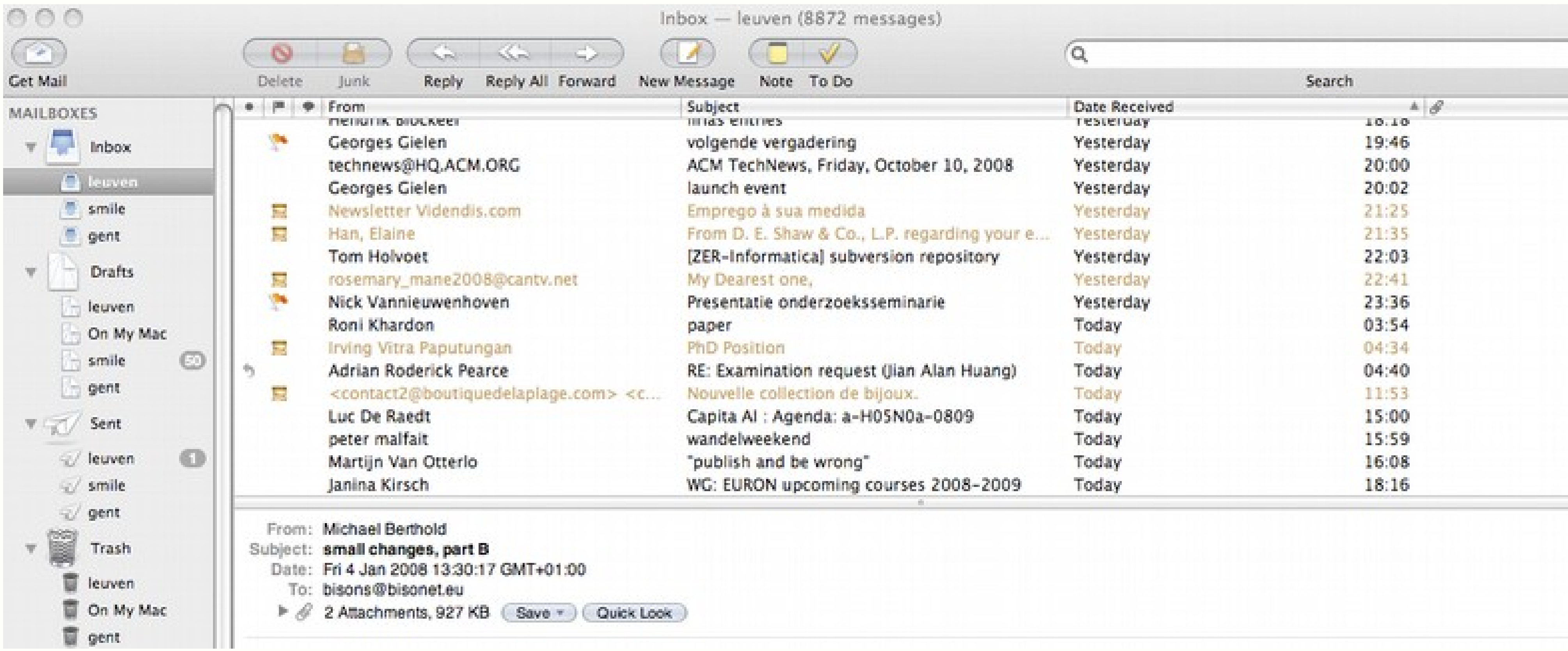
Success stories 1/3
automatic image captioning

Automatic Image Caption Generation
Sample taken from Andrej Karpathy, Li Fei-Fei

Success stories 2/3
product recommendation



Success stories 3/3
spam detection



[Example by Paolo Frasconi]

Common Data Mining Tasks

- Classification and class probability estimation
  - How likely is this consumer to respond to our campaign?
- Regression
  - How much will she use the service?
- Similarity Matching
  - Can we find consumers similar to my best customers?
- Clustering
  - Do my customers form natural groups?

Common Data Mining Tasks

Task	Supervised Methods	Unsupervised Methods
*Classification	✓	
*Regression	✓	
Causal Modeling	✓	
Similarity Matching	✓	✓
Link Prediction	✓	✓
Data Reduction	✓	✓
Clustering		✓
Co-occurrence Grouping		✓
Profiling		✓

Supervised = *labelled* data, target attribute (e.g. SPAM or not)
Unsupervised = *no labels* (e.g. customer records for profiling)

Terminology

Attributes

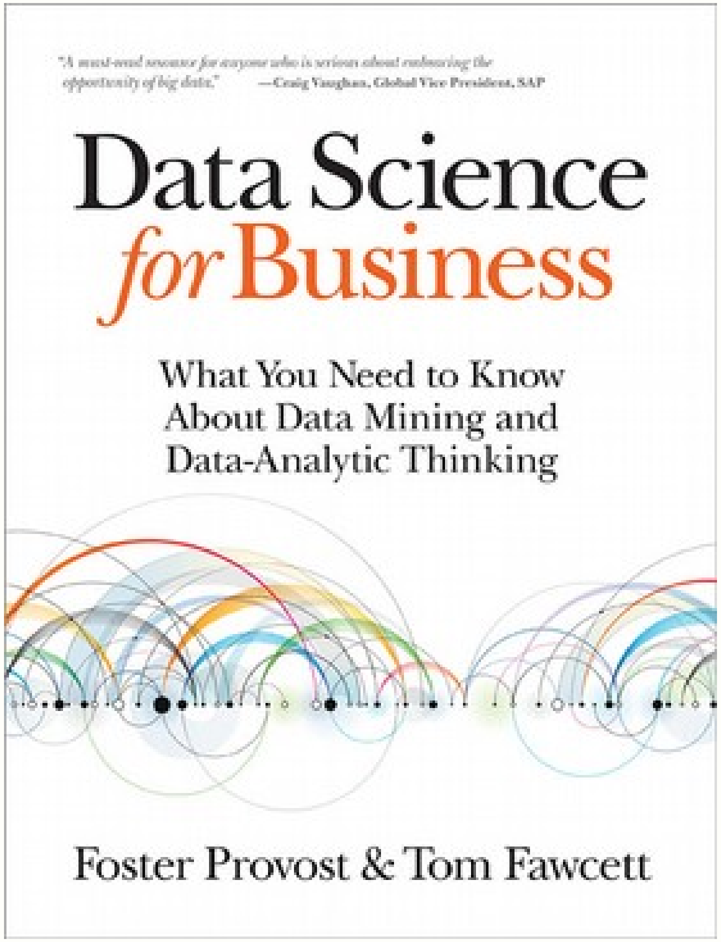
Target attribute

Name	Balance	Age	Employed	Write-off
Mike	$200,000	42	no	yes
Mary	$35,000	33	yes	no
Claudio	$115,000	40	no	no
Robert	$29,000	23	yes	yes
Dora	$72,000	31	no	no

This is one row (example).
Feature vector is: **<Claudio,115000,40,no>**
Class label (value of Target attribute) is **no**

Books

Most books: algorithmic or statistical focus



Focus on general principles

“In ten years’ time,
technologies will likely have changed
such that today’s choices seem quaint.”

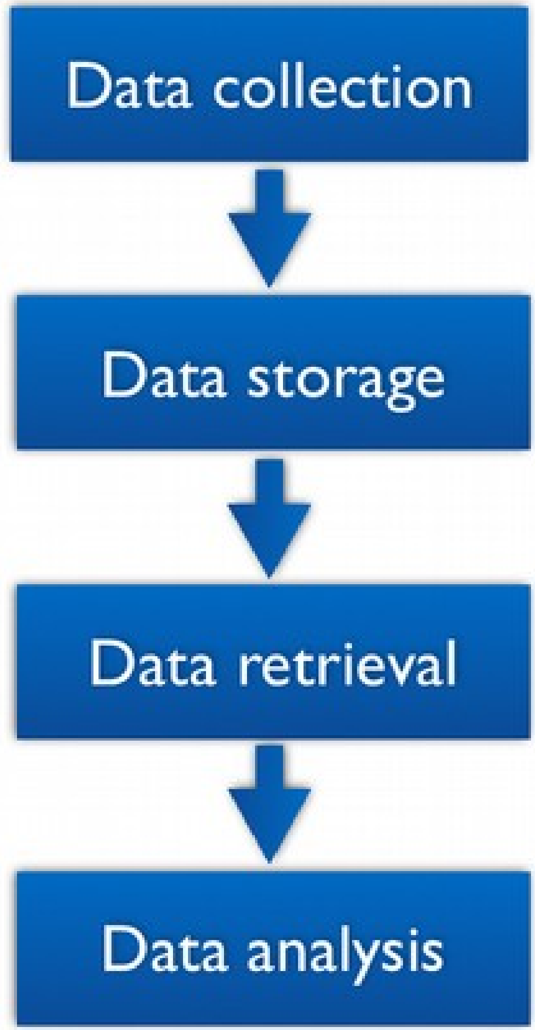
…

“general principles same for 20 years”

Principle 1:

Data Science is a process

From collection to utilization



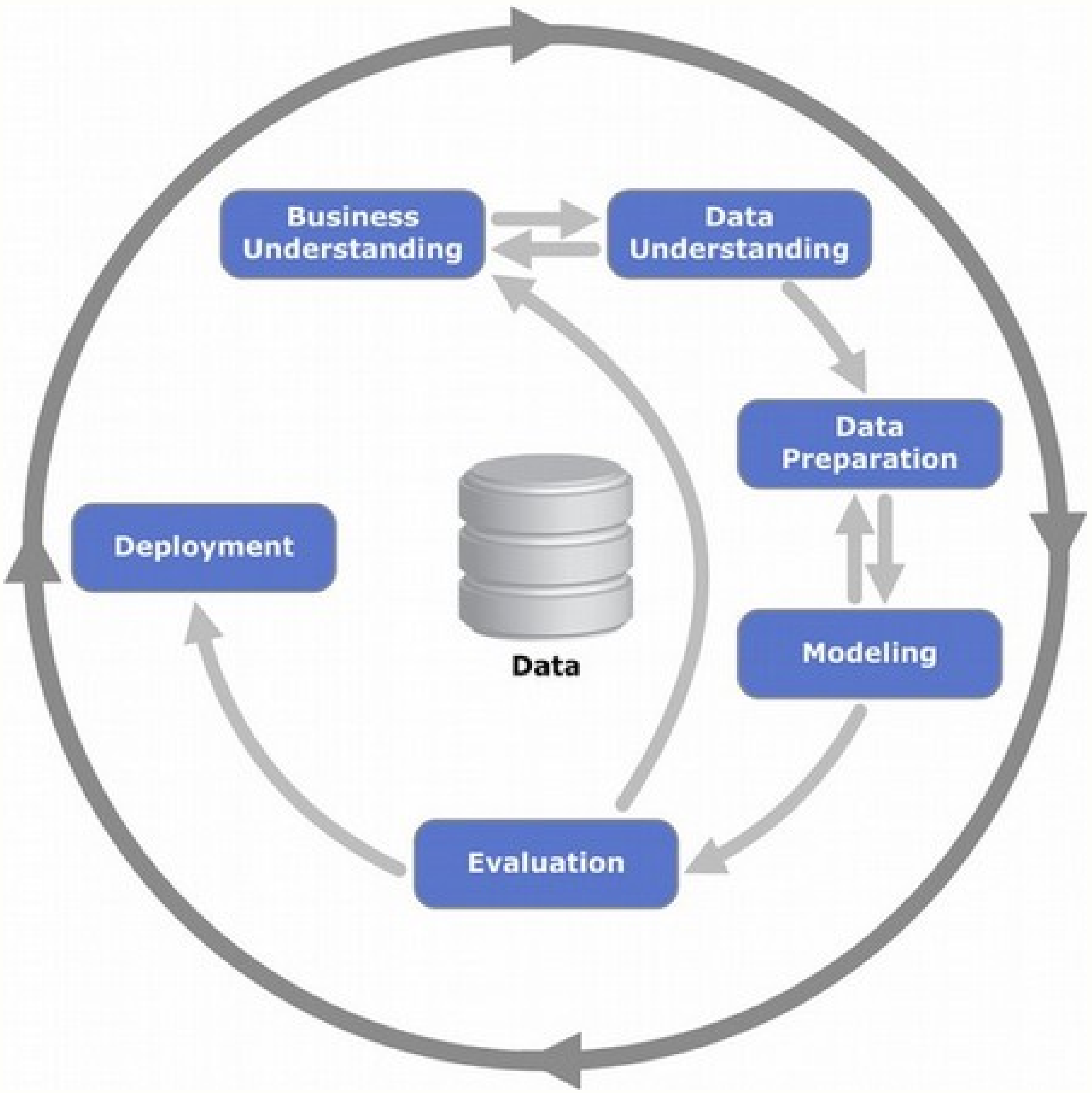
Chronology

	1950-	1970-	1990-	2000-
Collection	manual entry			Automatic collection (microarrays, ...) Collaborative collection
Storage	ad-hoc databases	relational databases	object-relational databases	social media, web, cloud
Retrieval	programming	query language	information retrieval	collaborative filtering, social methods
Analysis	statistics	machine learning	data mining	relational data mining, network analysis

"big data"

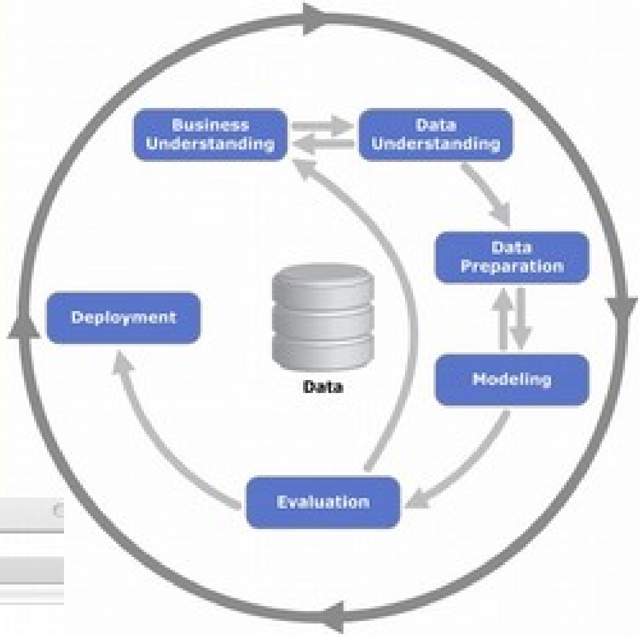
KU LEUVEN

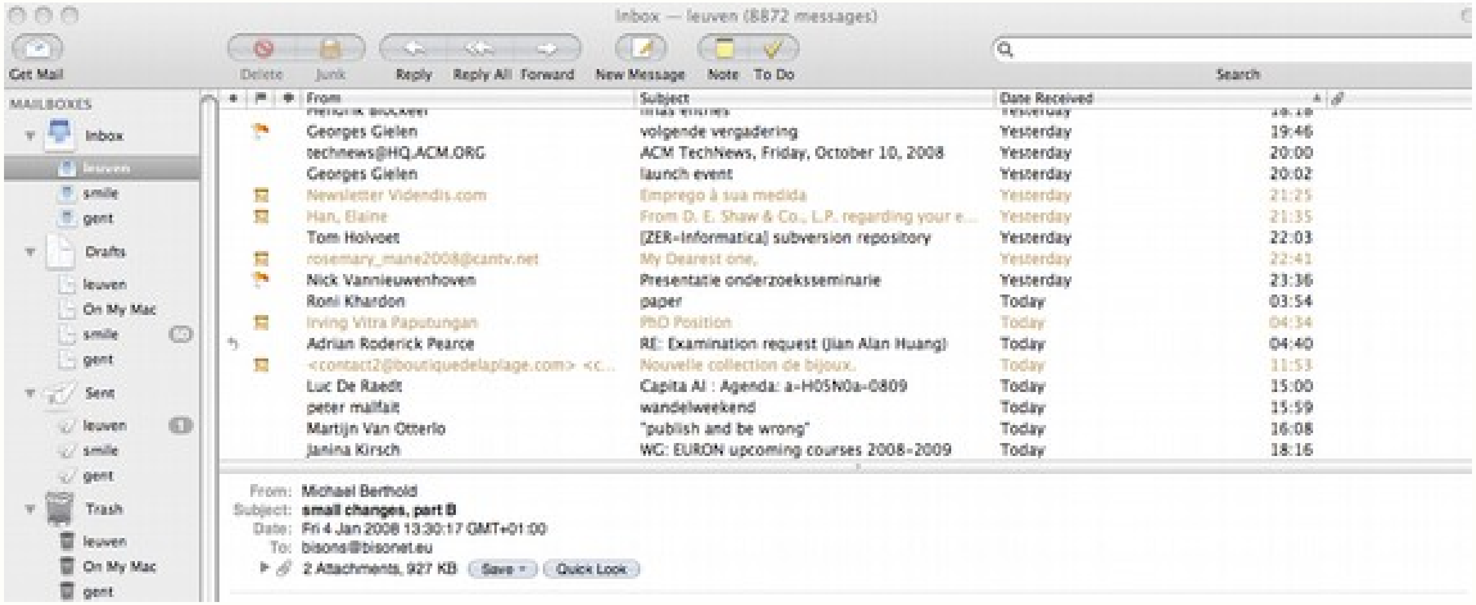
CRISP-DM process



An example

Business understanding





SPAM email reduces productivity,

automatically remove it

An example

Data understanding

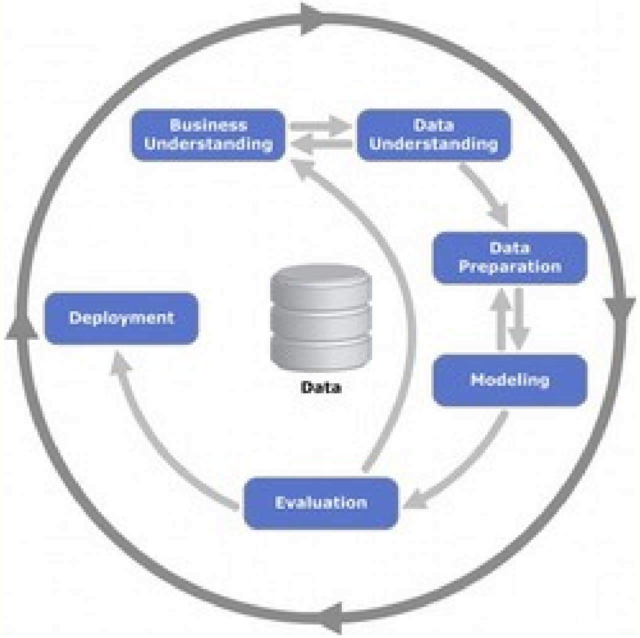
Given a text message, predict whether it is spam or not

→ text categorization, useful in general

→ we want a function from message to {0,1}

→ is called binary classification problem

An example



<u>Data preparation:</u> raw text

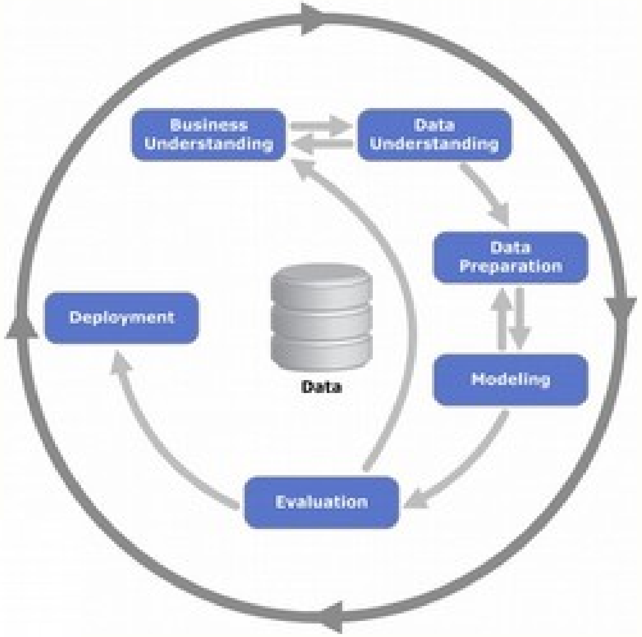
<u>Modeling:</u>

We could write a rule-based system, such as

if Title.contains("YOU HAVE WON!!!") then
  return Spam

Does it work well? → evaluate

An example



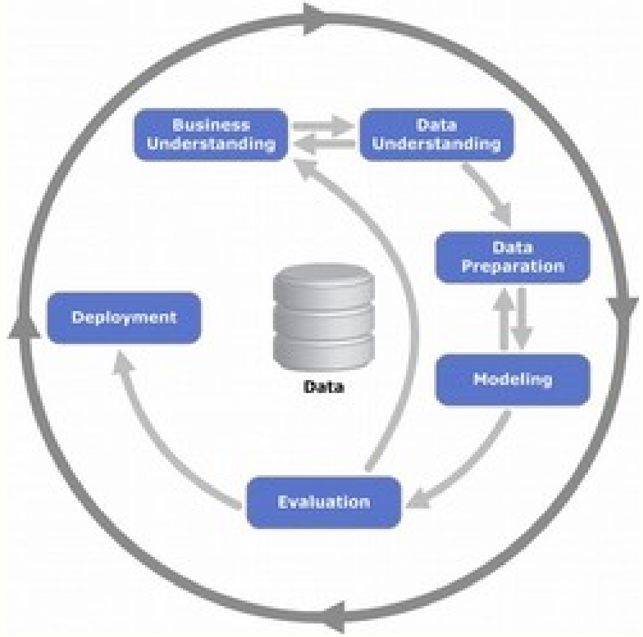
Evaluation

		Truth	
		Spam	Legitimate
Predicted as	Spam	150	30
	Legitimate	200	720

False positives

False negatives

An example

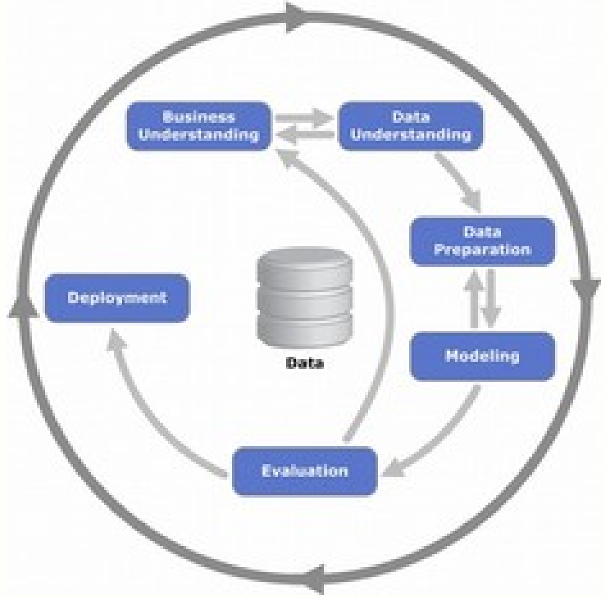


Evaluation to Business understanding:

- How do we find good rules?
  Knowledge elicitation or formalization may be difficult
- How do we define good? Will depend on user?

We need a system that can adapt: self-learning

An example, classificatier-based



- Data understanding: collect messages, in general and from the user, that are spam (negative) and legitimate (positive)
- Data preparation: bag-of-words representation
- Modeling: train a classifier (e.g. naïve bayes)
- Evaluate: on unseen emails
- Deploy: predict for new emails, retrain when user disagrees

Principle 2a:

Machine Learning is optimisation,

it optimises loss functions

A formal task description

Function approximation!!!

•**Given:**

- a space of possible instances X
- an unknown target function f: $X \rightarrow Y$
- a hypothesis space L containing functions $X \rightarrow Y$
- **a set of examples E = { (x, f(x)) | x $\in$ X }**
- a loss function *loss*(h,E) $\rightarrow \mathbb{R}$

•**Find:** h $\in$ L that minimizes *loss*(h,E)

model

"supervised"

Matches many (not all) tasks

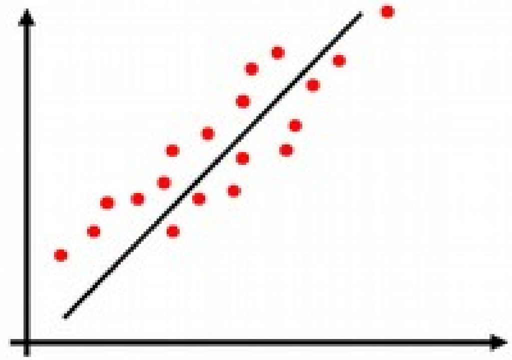
Attributes

Target attribute

Name	Balance	Age	Employed	Write-off
Mike	$200,000	42	no	yes
Mary	$35,000	33	yes	no
Claudio	$115,000	40	no	no
Robert	$29,000	23	yes	yes

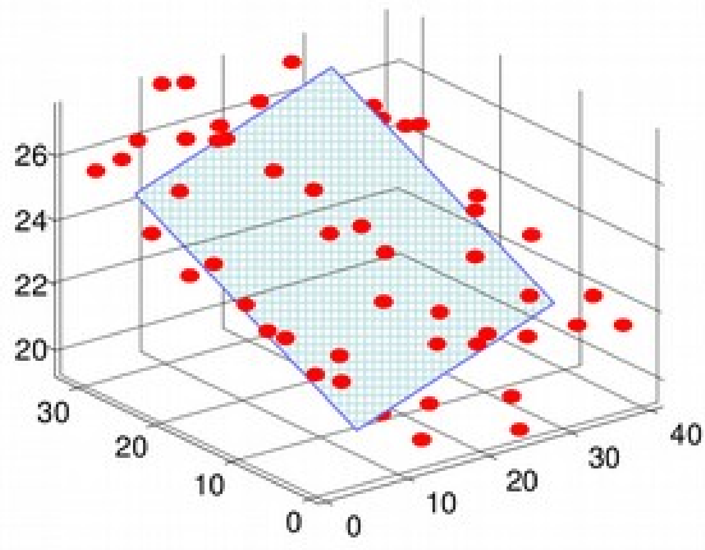
Linear Regression

Notations:



- Datapoints $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \quad \mathbf{x}_i \in R^d$
- Labels $\mathbf{y} = \{y_1, y_2, \ldots, y_n\} \quad \mathbf{y} \in R$
- Linear decision function $f(\cdot) : R^d \to R$

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

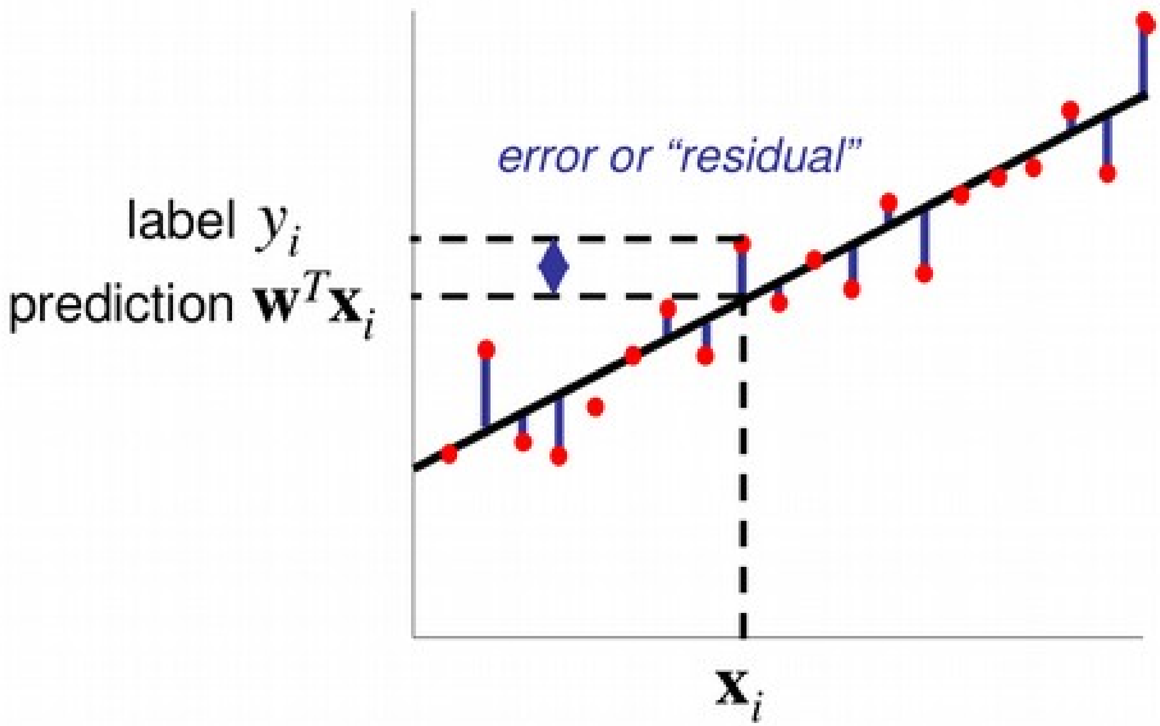


- Parameter vector $\mathbf{w}$

[Slide by E. Ricci, Analysis of Patterns, 2009]

Linear Regression

- Goal: find a linear function $\mathbf{X}\mathbf{w}$ that approximates the labels $\mathbf{y}$.
- For a new test point $\mathbf{x}$ the label y can be estimated as $\mathbf{w}^T\mathbf{x}$.

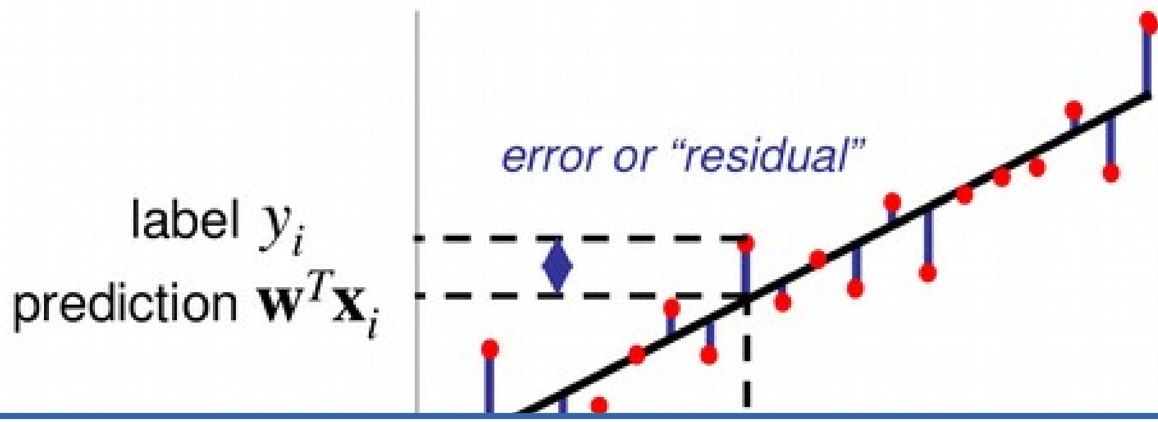


Sum Squared Error $\quad E = \sum_{i=1}^{n} \left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2 = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$

[Slide by E. Ricci, Analysis of Patterns, 2009]

Linear Regression

- Goal: find a linear function $\mathbf{X}\mathbf{w}$ that approximates the labels $\mathbf{y}$.
- For a new test point $\mathbf{x}$ the label y can be estimated as $\mathbf{w}^T\mathbf{x}$.



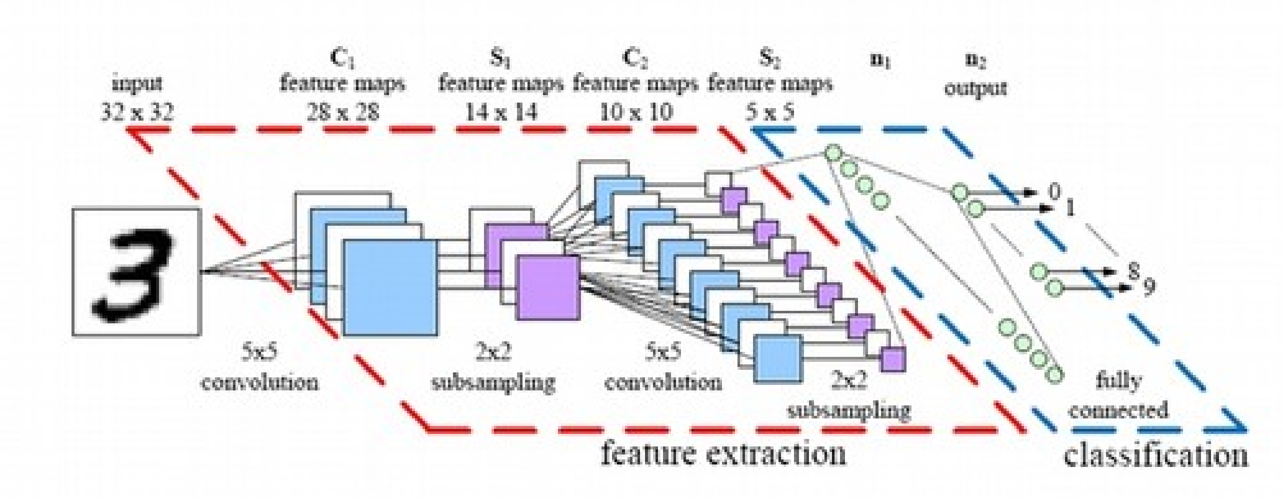
Loss function is Sum of Squared errors (L2 norm)
Is convex → optimise it (least squares regr.)

Sum Squared Error $E = \sum_{i=1}^{n} \left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2 = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$

[Slide by E. Ricci, Analysis of Patterns, 2009]

Deep learning

handwritten number recognition:



learning: (stochastic) gradient descent

Deep learning

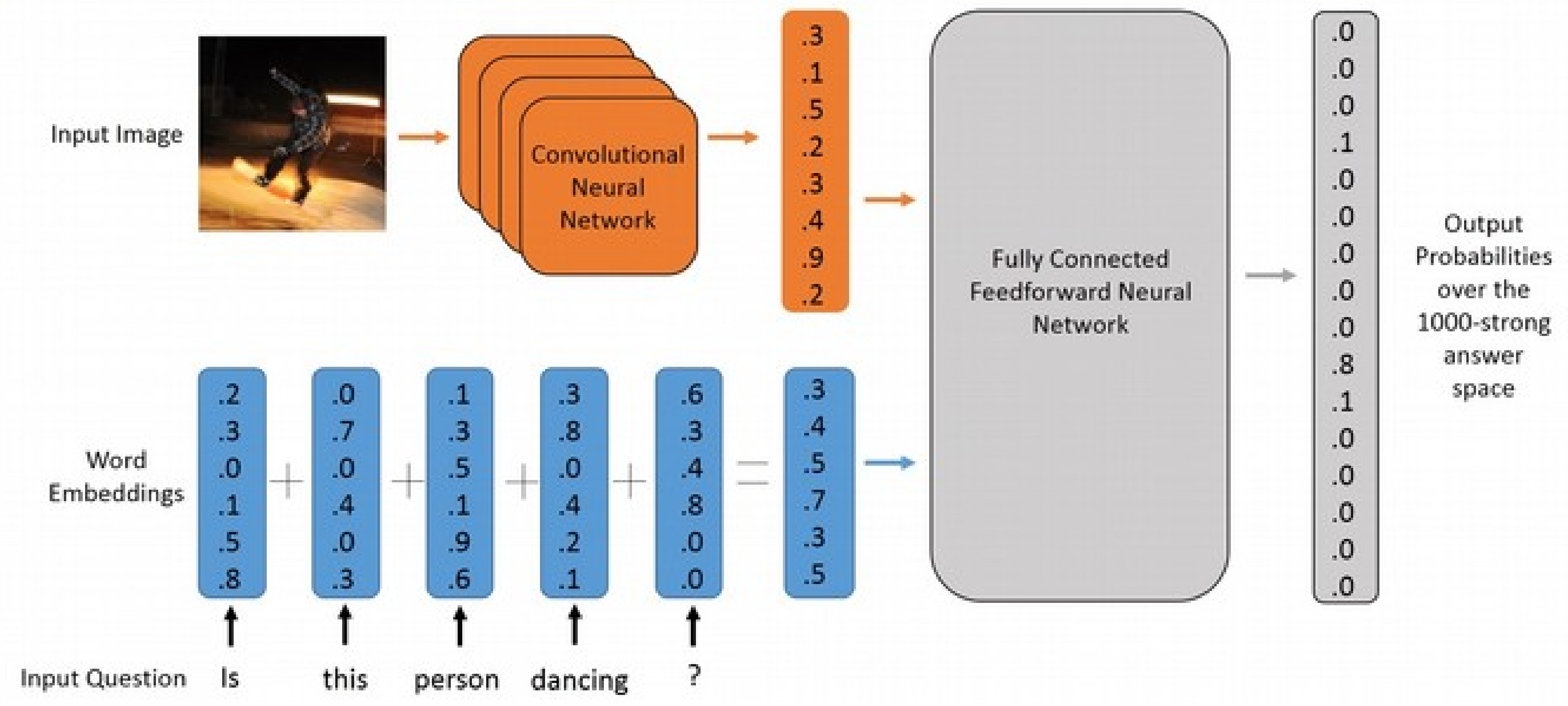
Optimisation through stochastic gradient descent

Algorithm 1: Stochastic gradient descent

Input : training data $\mathcal{D} = \{X, y\}_{i=1}^{n}$, learning rate γ

initialize θ (neural network weights)
for *epochs* **do**
    for *batches* **do**
        sample batch $(X, y) \sim \mathcal{D}$
        $\hat{y} \leftarrow g(z, \theta)$ (forward pass: compute predictions)
        Compute loss $L(y, \hat{y})$ and gradient $\frac{\partial L}{\partial \theta}$
        Update $\theta = \theta - \gamma \frac{\partial L}{\partial \theta}$ through backpropagation (backward pass)
    end
end

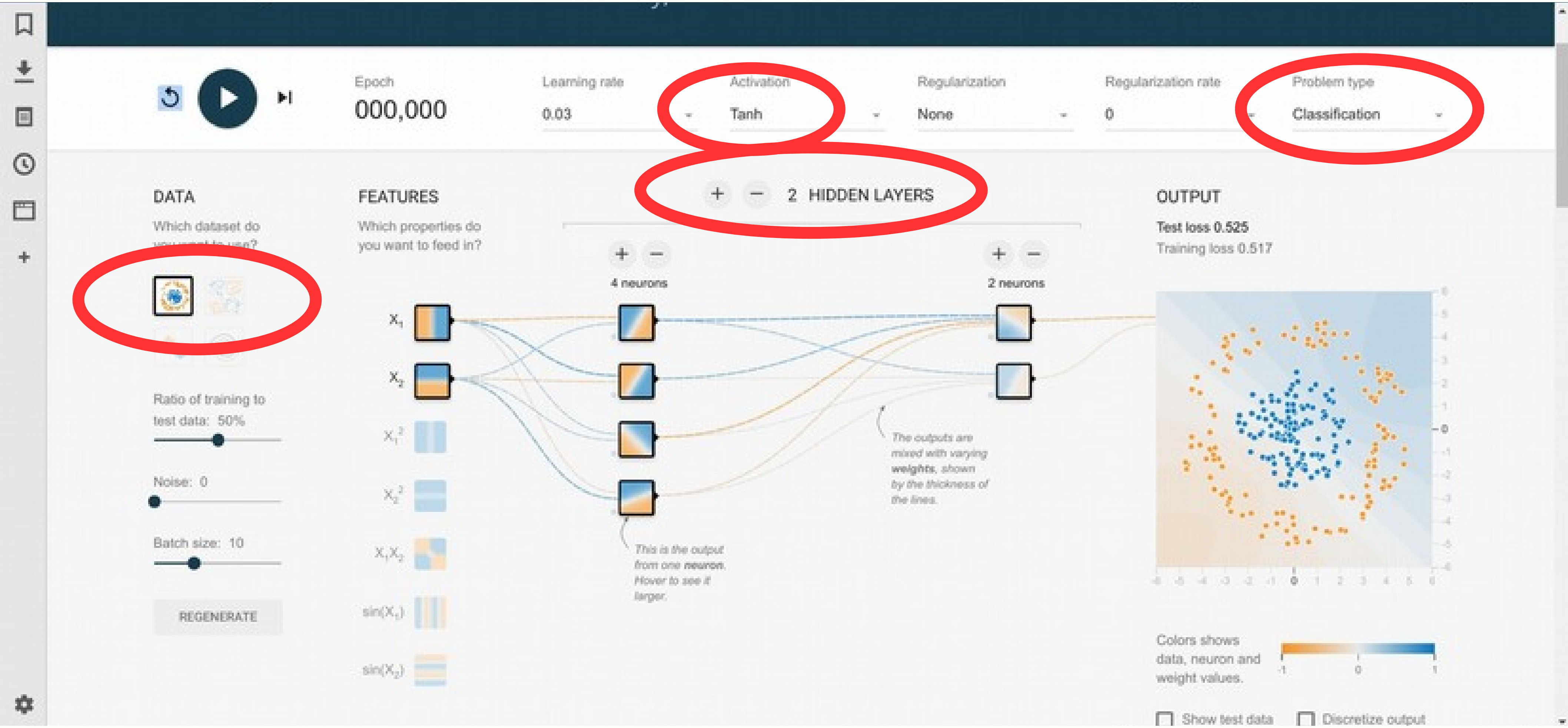
Deep learning



learning: (stochastic) gradient descent

https://playground.tensorflow.org/

Deep learning: importance layers, number neurons?



Principle 2b:

If you look too hard at a dataset,

you'll find things that <u>don't generalize</u> to unseen data

Principle 2b:

If you look too hard at a dataset,

you'll find things that <u>don't generalize</u>
to unseen data

Overfitting

A formal task description

Function approximation!!!

•**Given:**

- a space of possible instances X
- an unknown target function f: $X \rightarrow Y$
- a hypothesis space L containing functions $X \rightarrow Y$
- **a set of examples E = { (x, f(x)) | x $\in$ X }**
- a loss function *loss*(h,E) $\rightarrow \mathbb{R}$

•**Find:** h $\in$ L that minimizes *loss*(h,E)

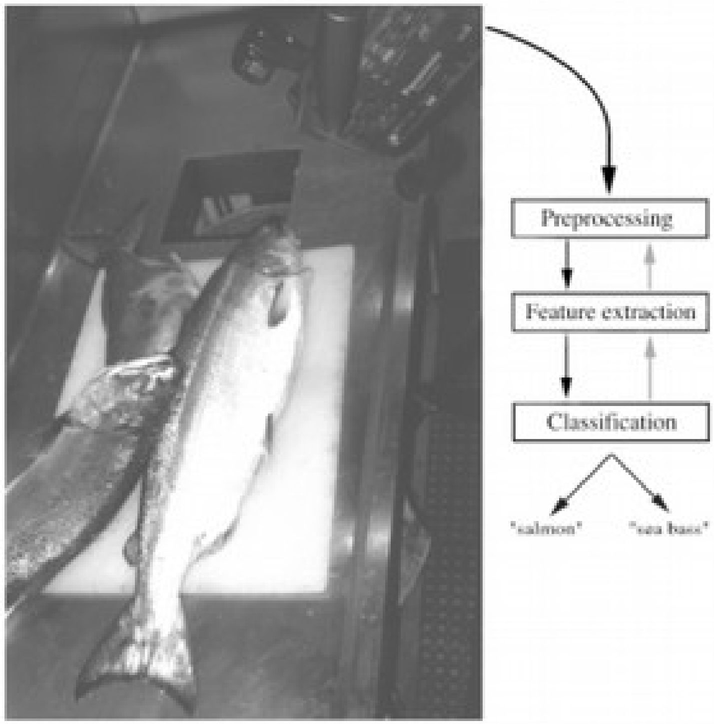
model

"supervised"

Matches many (not all) tasks

KU LEUVEN

A toy example



e.g. contour extraction

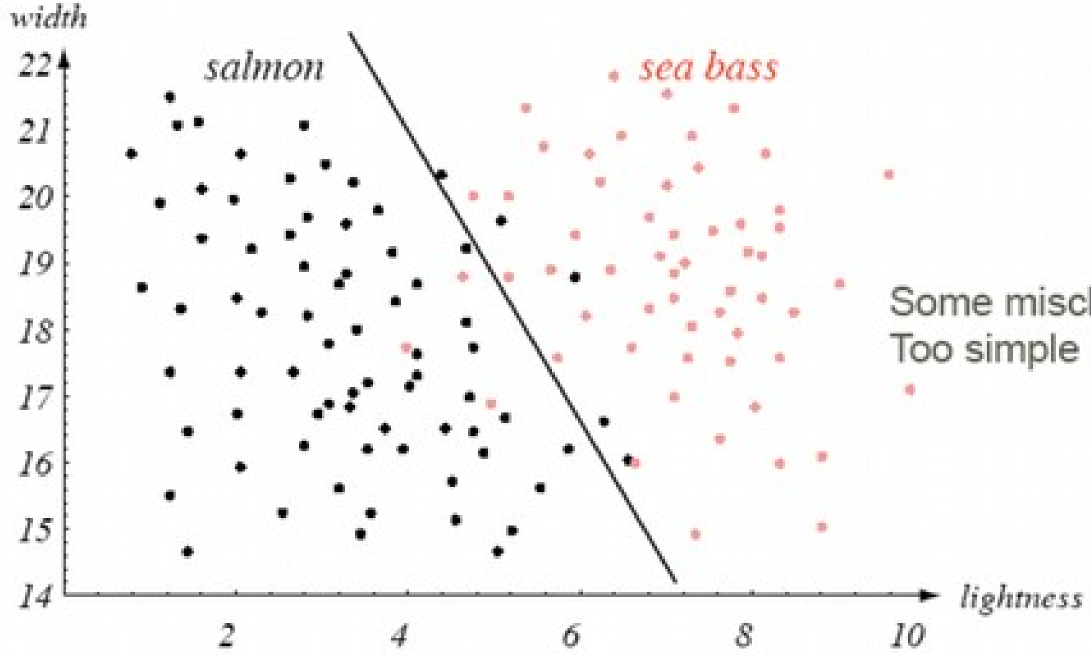
e.g. measure length and lightness

machine that makes decision given the above features

Example from: Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000

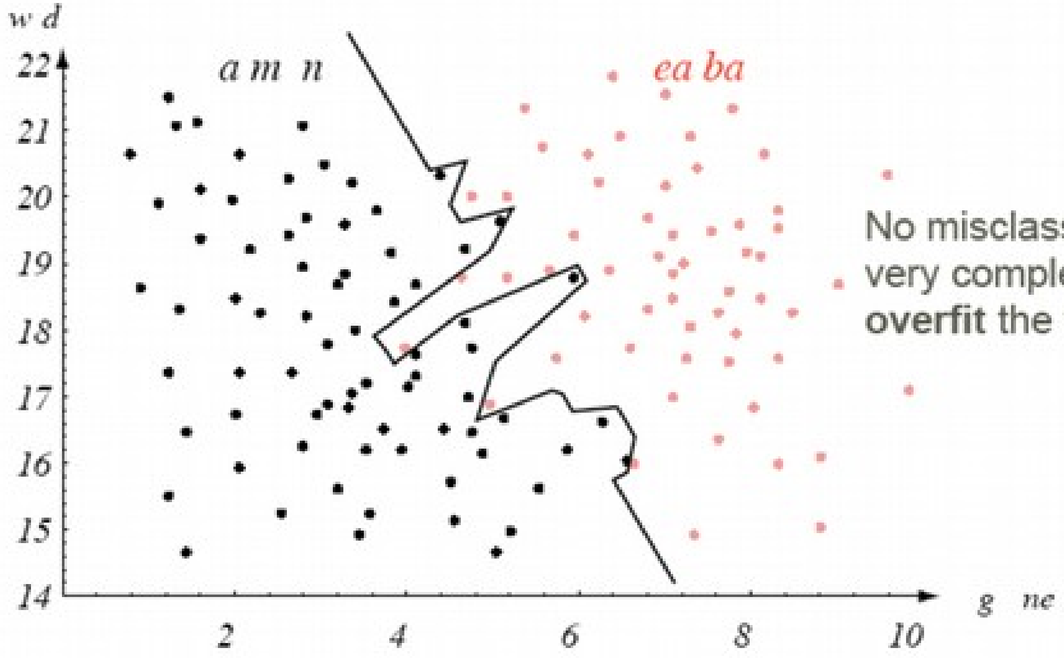
KU LEUVEN

Linear Separation



Some misclassified fishes
Too simple decision?

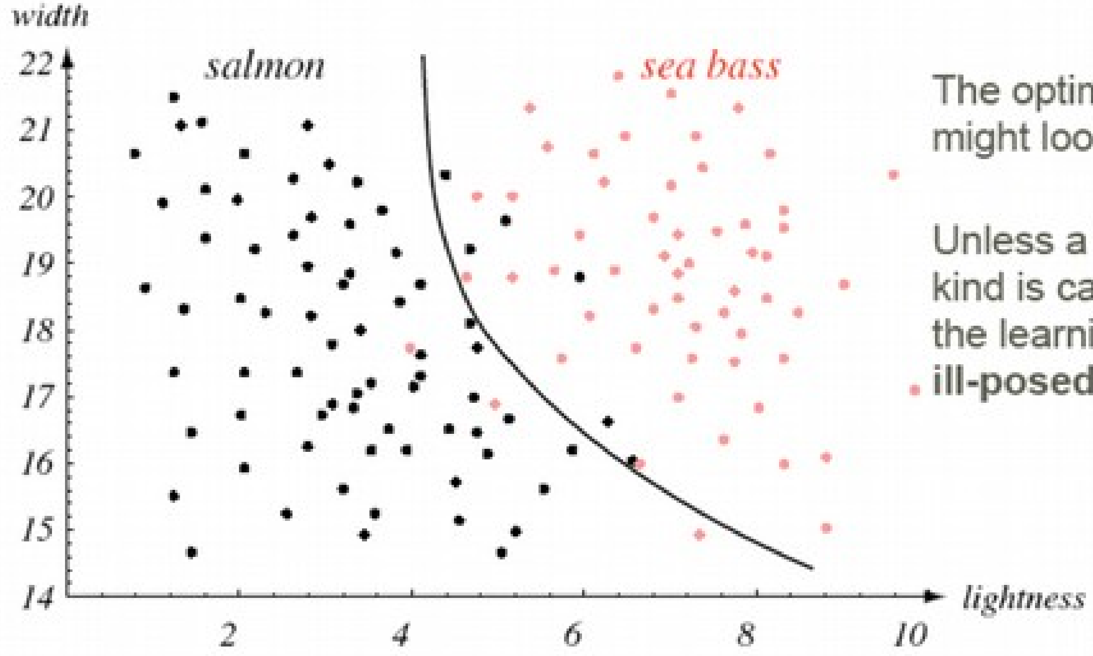
Overfitting



No misclassifications but...
very complex decisions can
overfit the training data

KU LEUVEN

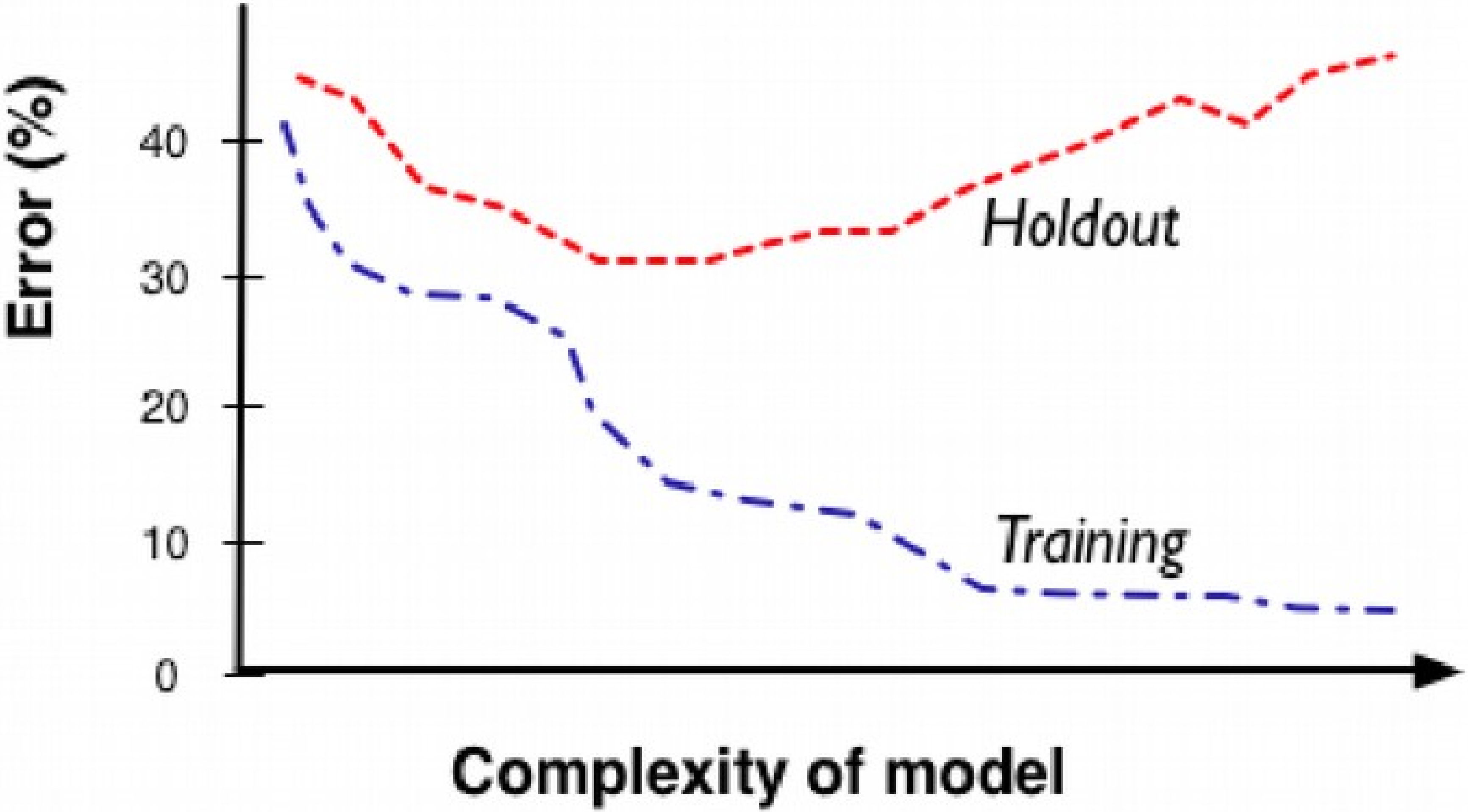
Reasonable Solution



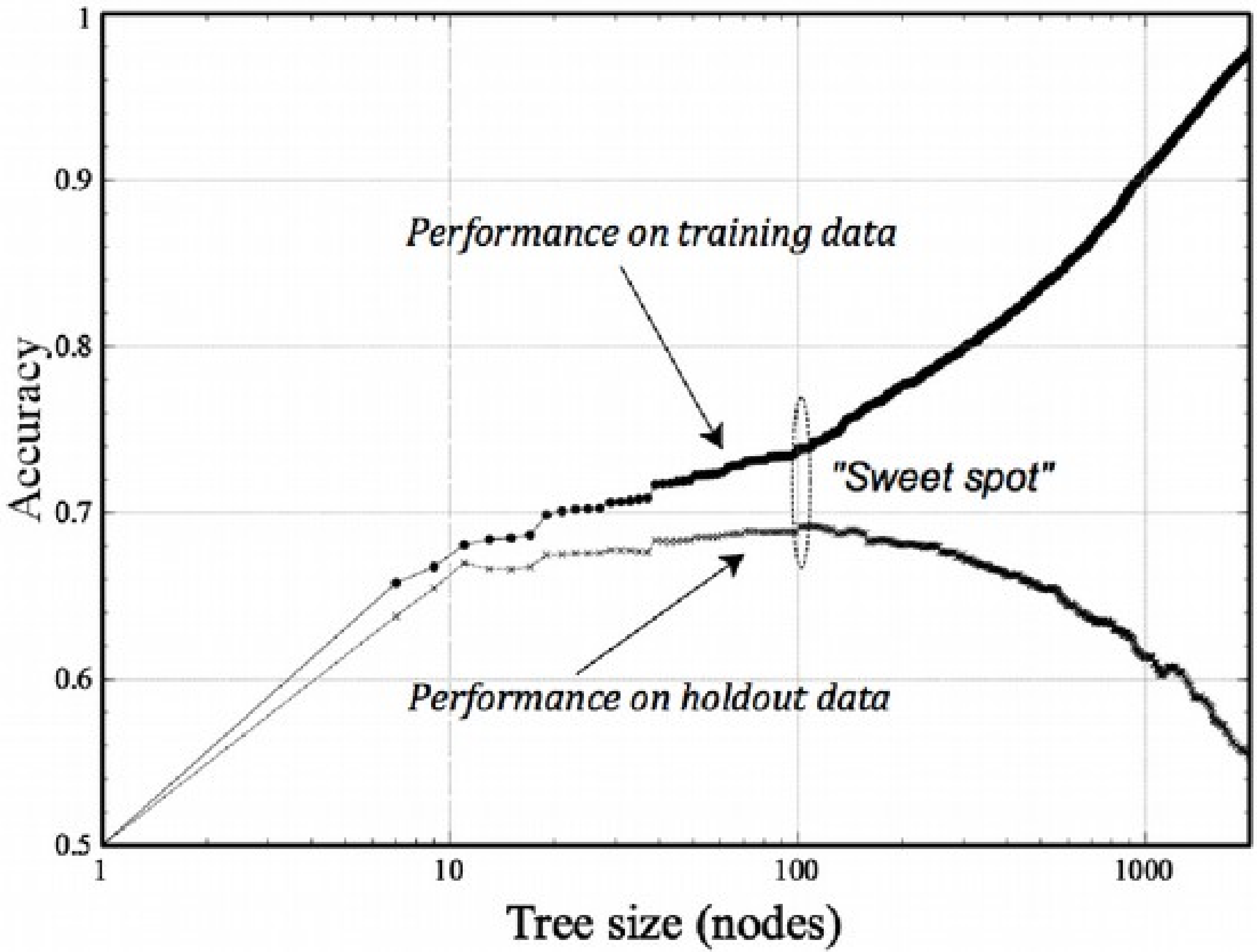
The optimal tradeoff might look like this

Unless a tradeoff of this kind is carefully defined the learning problem is **ill-posed!**

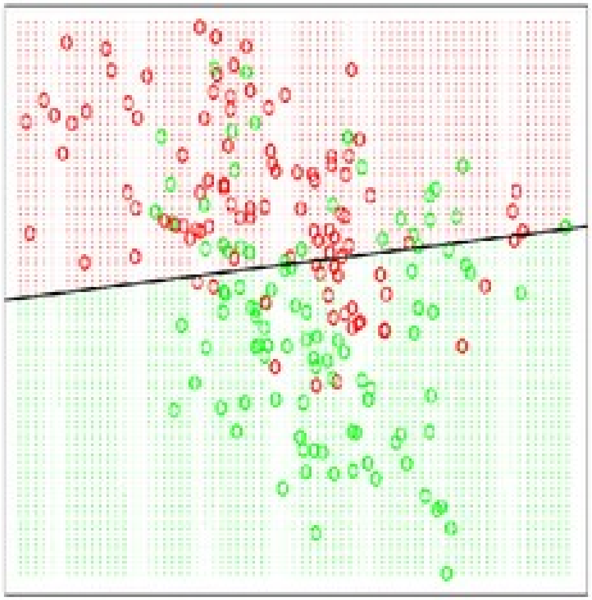
Fitting Graph



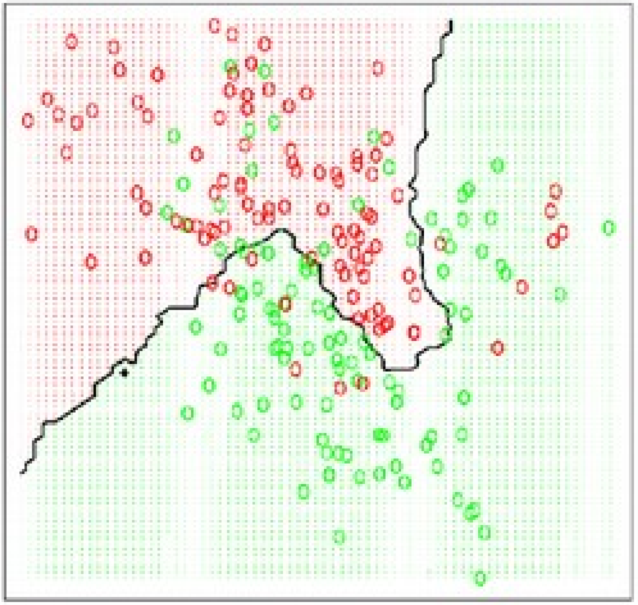
Over-fitting in tree induction



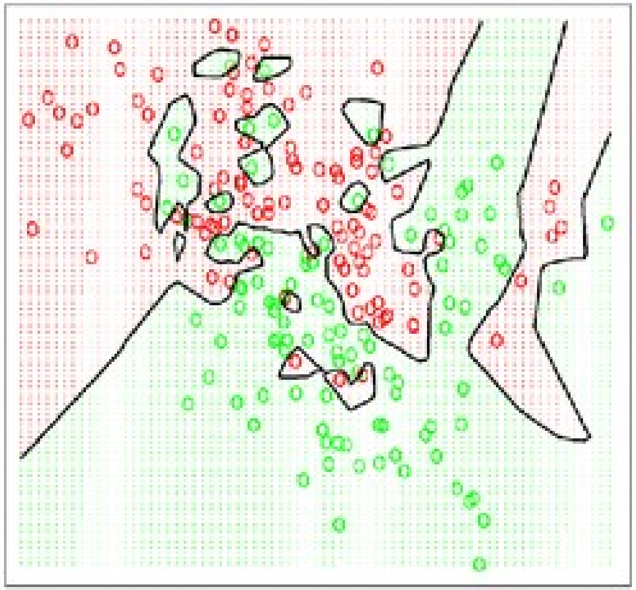
Need for holdout evaluation



Under-fitting



Good

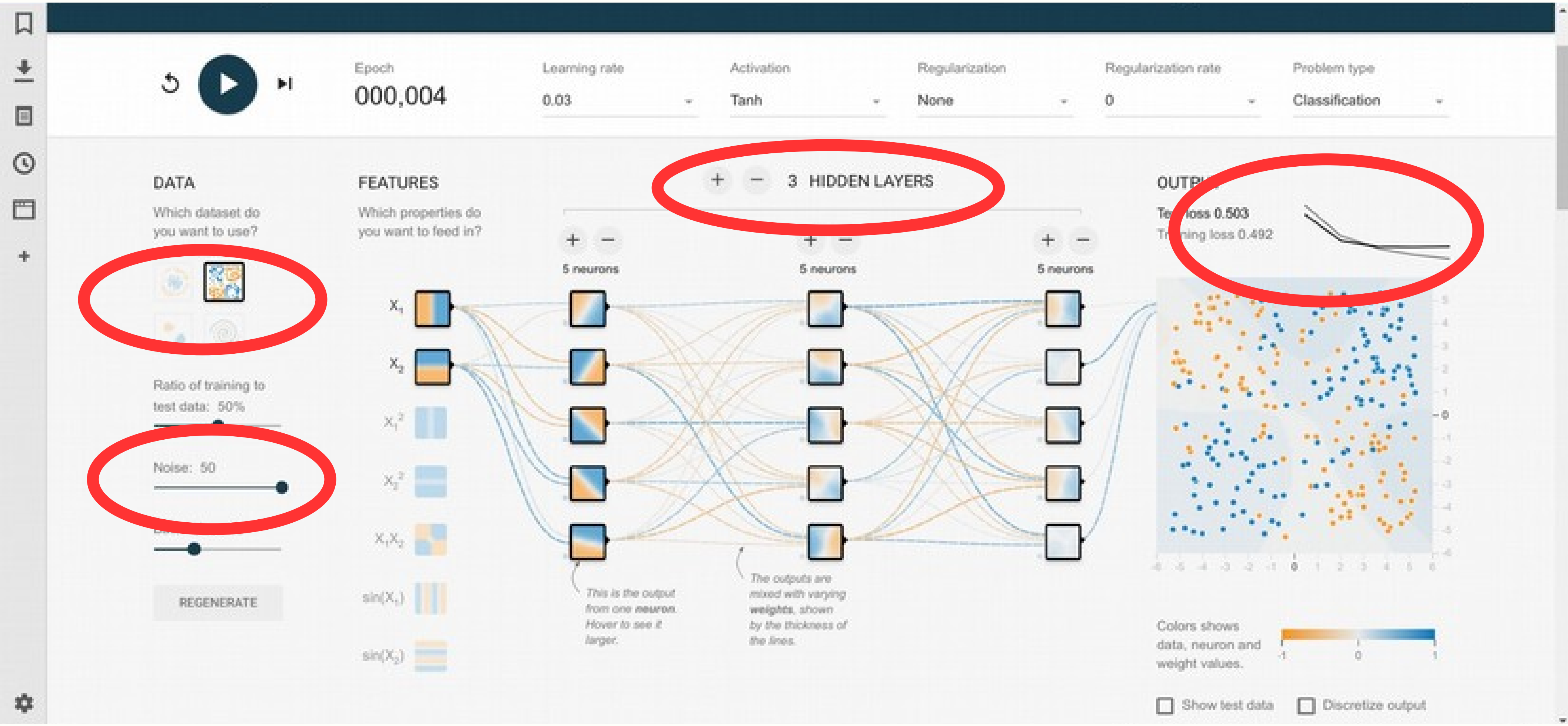


Over-fitting

- In sample evaluation is in favor or “memorizing”
- On the *training data* the right model would be best
- But on *new data* it would be bad

https://playground.tensorflow.org/

Overfitting: try different ways to make it happen

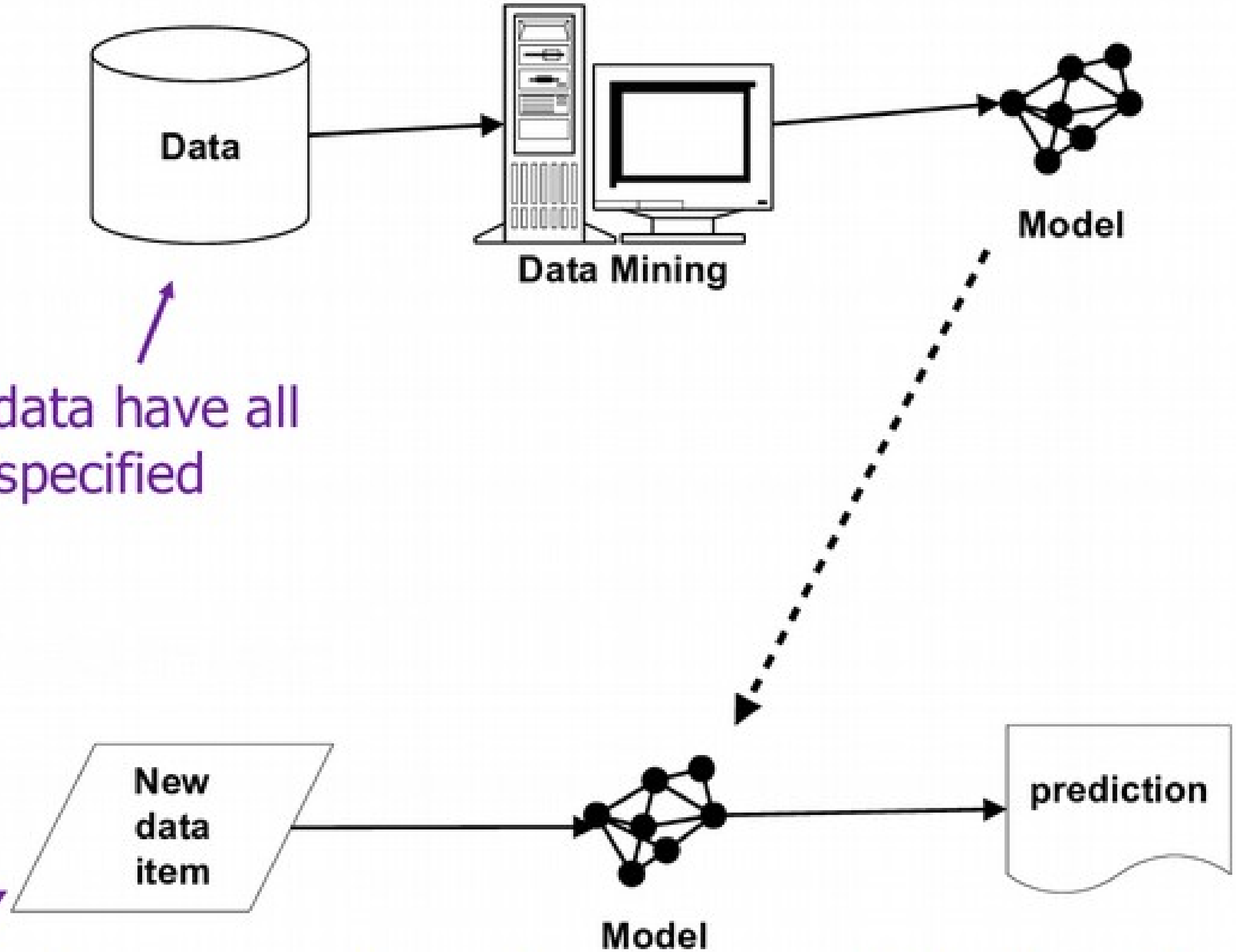


Principle 3:

Data science needs to be evaluated in the context of operation

Data Mining versus Use of the Model

"Supervised" modeling:



"Training" data have all values specified

New data item has some value unknown (e.g., will she leave?)

Pitfalls in DM

- Training data is not consistent with actual use
- Bad sample
- Bad features

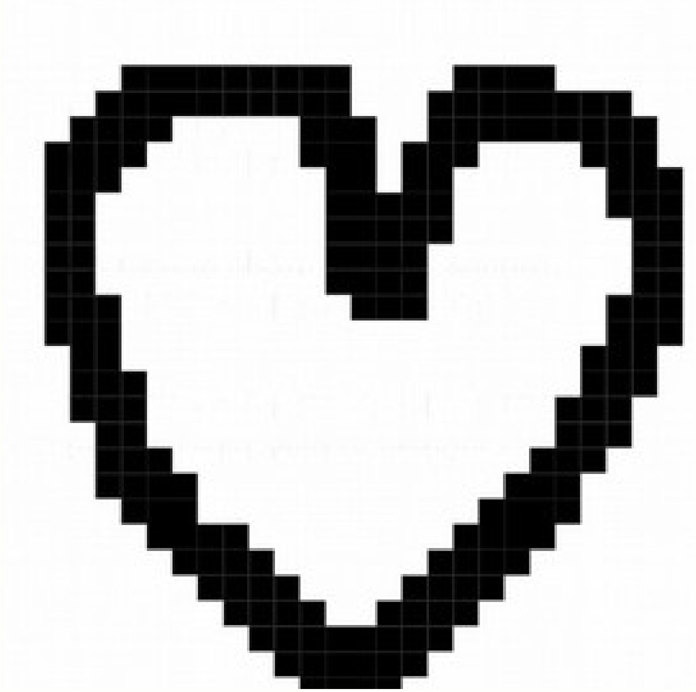
Ex: “survivorship issues”

Lending agency wants to use ML to screen applications and accept/reject them

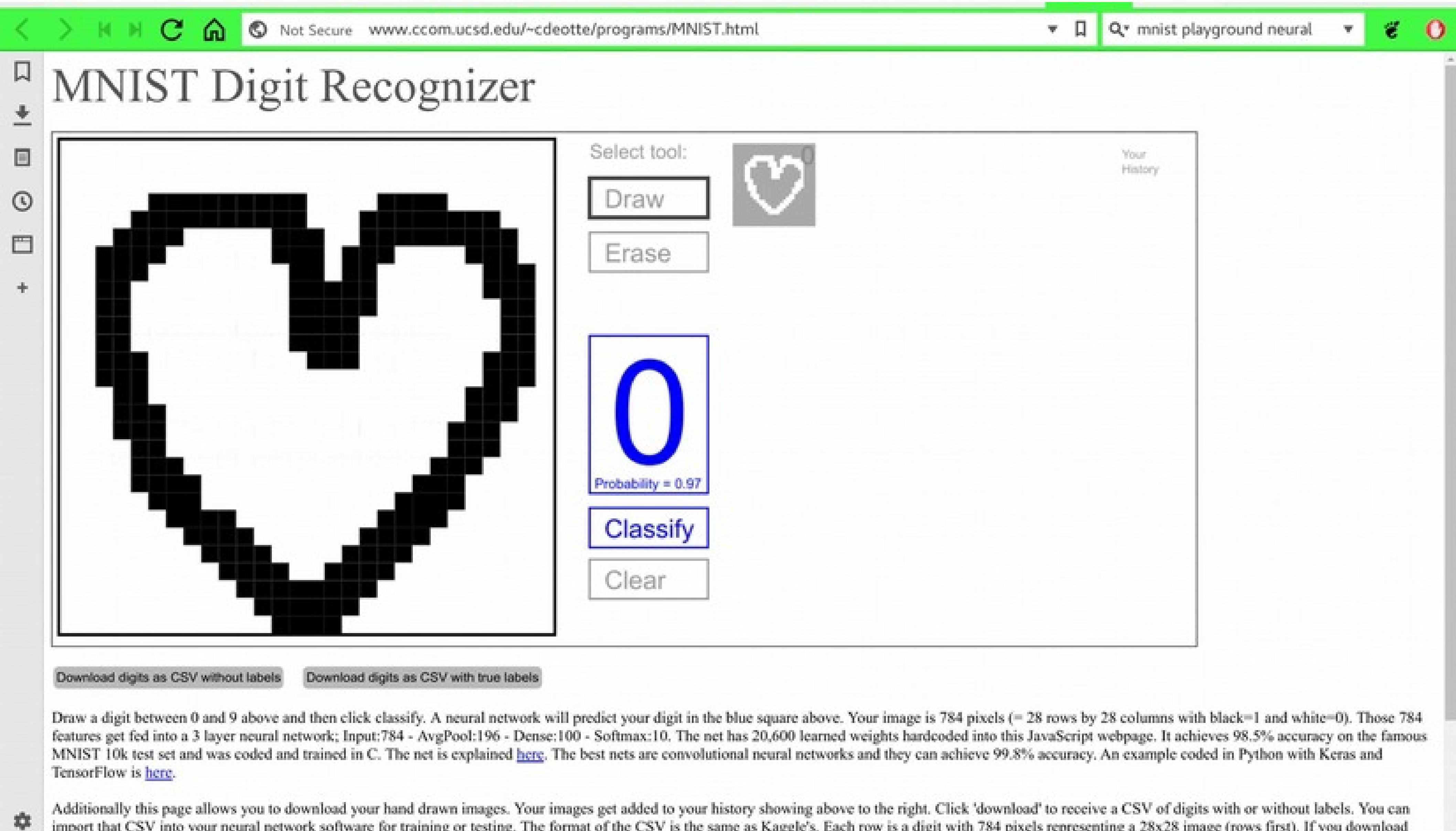
- data of accepted loans + outcome
- BAD: use this to learn an outcome predictive model

Pitfalls in DM

What number is this?



http://www.ccom.ucsd.edu/~cdeotte/programs/MNIST.html

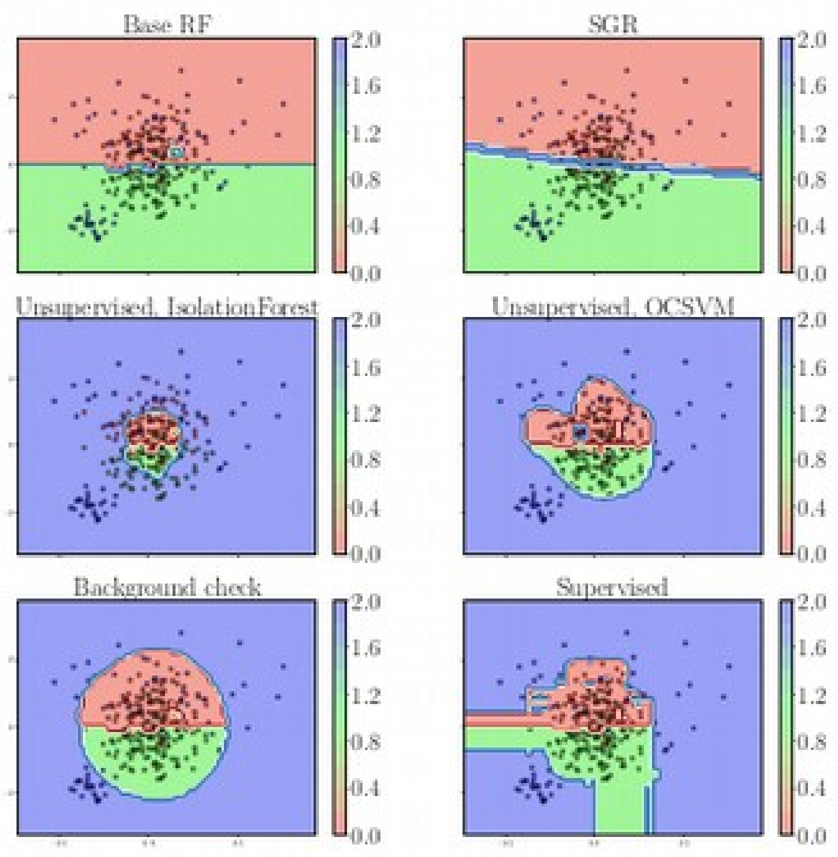


Pitfalls in DM

Can we do something about this?

Highly relevant topic; related to:

- out-of-distribution detection
- classification with a reject option



[Probability of default estimation, with a reject option. Coenen, Abdullah, Guns, DSAA20]

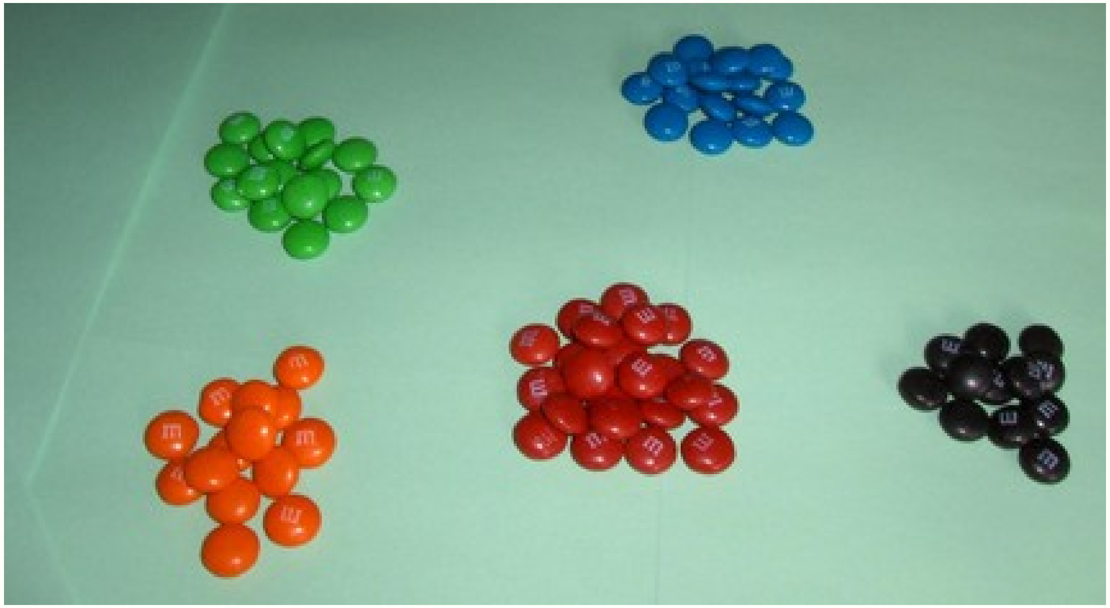
Principle 4:

Entities that are similar on some attributes

often are similar on unseen attributes

Similarity

- Ex: clustering



- Also solved with optimisation, e.g. min. distances to cluster center

Similarity

Key concept: <u>distance</u> between objects

Euclidean, manhatten, edit distances (strings), dynamic time warping (temporal sequences), …

Ex. group hand-written letters together

1) based on raw pixels (but: shift, scale)

2) based on learned representation (auto-encoding) https://cs.stanford.edu/people/karpathy/convnetjs/demo/autoencoder.html

Auto-encoding + clustering of representation

Principle 5:

To draw causal conclusions,

one must pay very close attention to the presence of (possibly unseen) confounding factors

Causality?

Machine models exploit correlation, NOT causality

Very tempting to inspect model and see
“what causes things to be true/false”

Causality?

Machine models exploit correlation, NOT causality

Very tempting to inspect model and see
“what causes things to be true/false”

E.g. coefficients of linear regression

$Y = 20*X_1 – 12*X_2 + 300*X_3 + 99*X_4 - 299*X_5$

Which feature has most impact?

Principles

1. Data Science is a process
2. ML is optimisation of loss functions
3. ML must generalize to unseen data
4. Evaluate data science in its operational context
5. Similar entities can have similar unseen attribs
6. Correlation, not causation

Popular practical tools

- Exploratory analysis of high dim. tabular data:
  tableau (web only, not open source)
- Classification and regression on tabular data:
  scikitlearn
- Non-linear regression on large tabular data:
  xdgboost
- Deep learning on sensory data (images, audio, ...):
  pytorch

Questions?

Slides available: http://homepages.vub.ac.be/~tiasguns/ (soon)

More playgrounds?

https://cs.stanford.edu/people/karpathy/convnetjs/