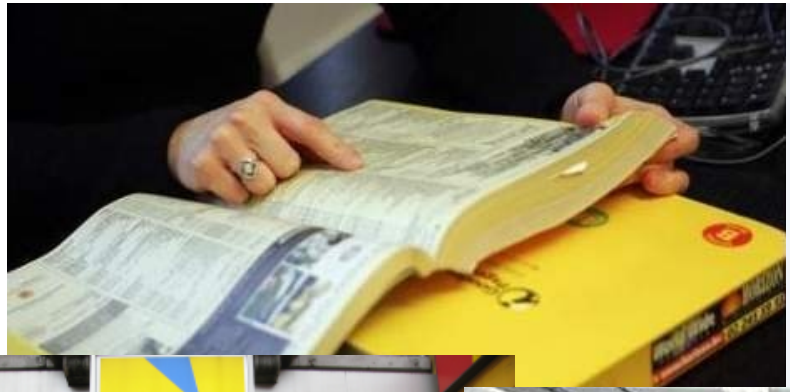
KU LEUVEN

Artificiële Intelligentie

joost.vennekens@kuleuven.be

Software is eating the world



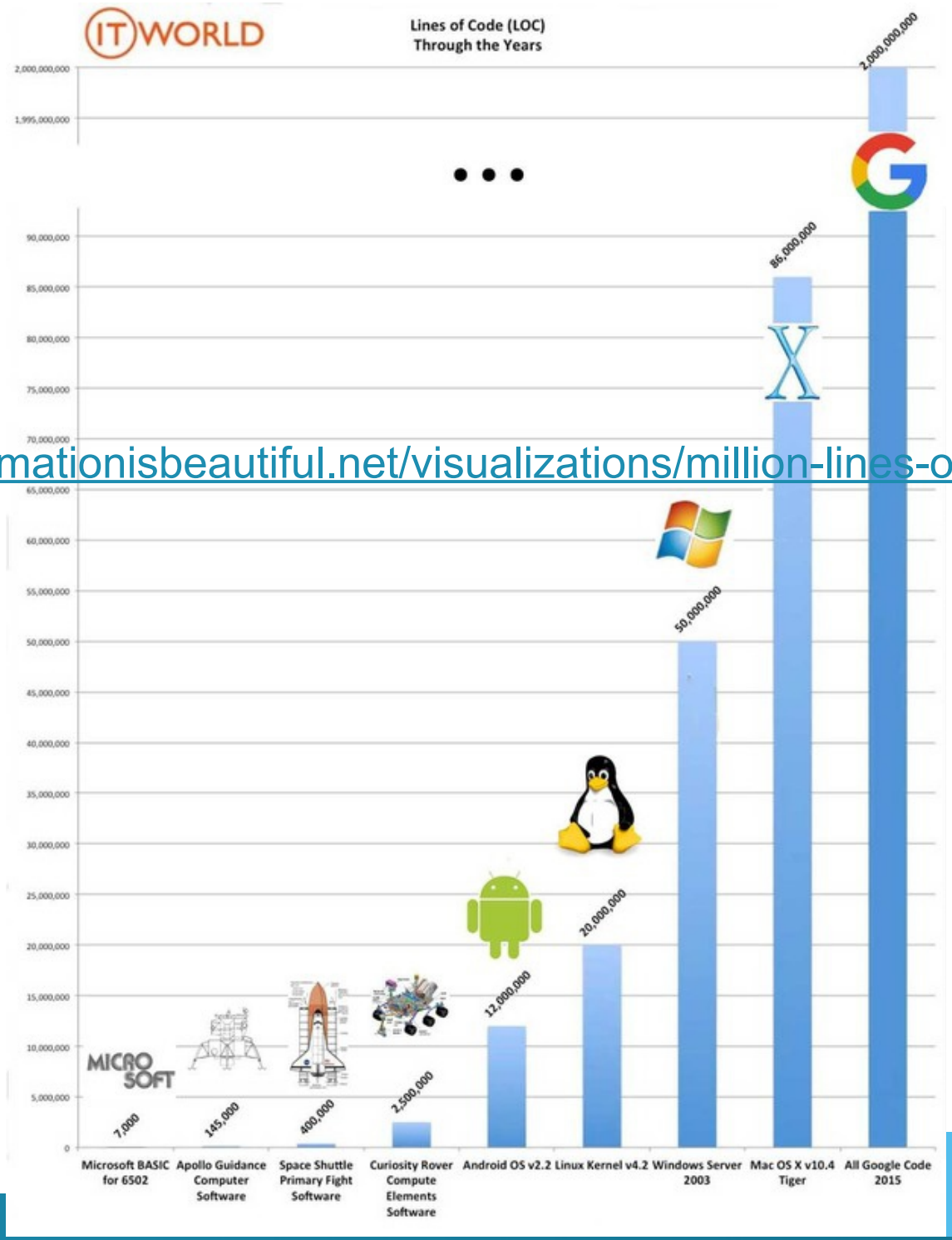
Belang van software

Company	Cap Rank on 2/3/17	Market Cap on 2/3/17
Apple	1	678.4
Alphabet	2	567.0
Microsoft	3	492.1
Berkshire Hathaway	4	404.5
Amazon.com	5	385.0
Facebook	6	377.5
Exxon Mobil	7	346.5
JPMorgan Chase	8	311.3
Johnson & Johnson	9	309.2
Wells Fargo	10	287.3

Grootste bedrijven in VS (miljard $)

http://hackernoon.com

Evolutie



https://informationisbeautiful.net/visualizations/million-lines-of-code/

www.itworld.com

Waar ligt de limiet?

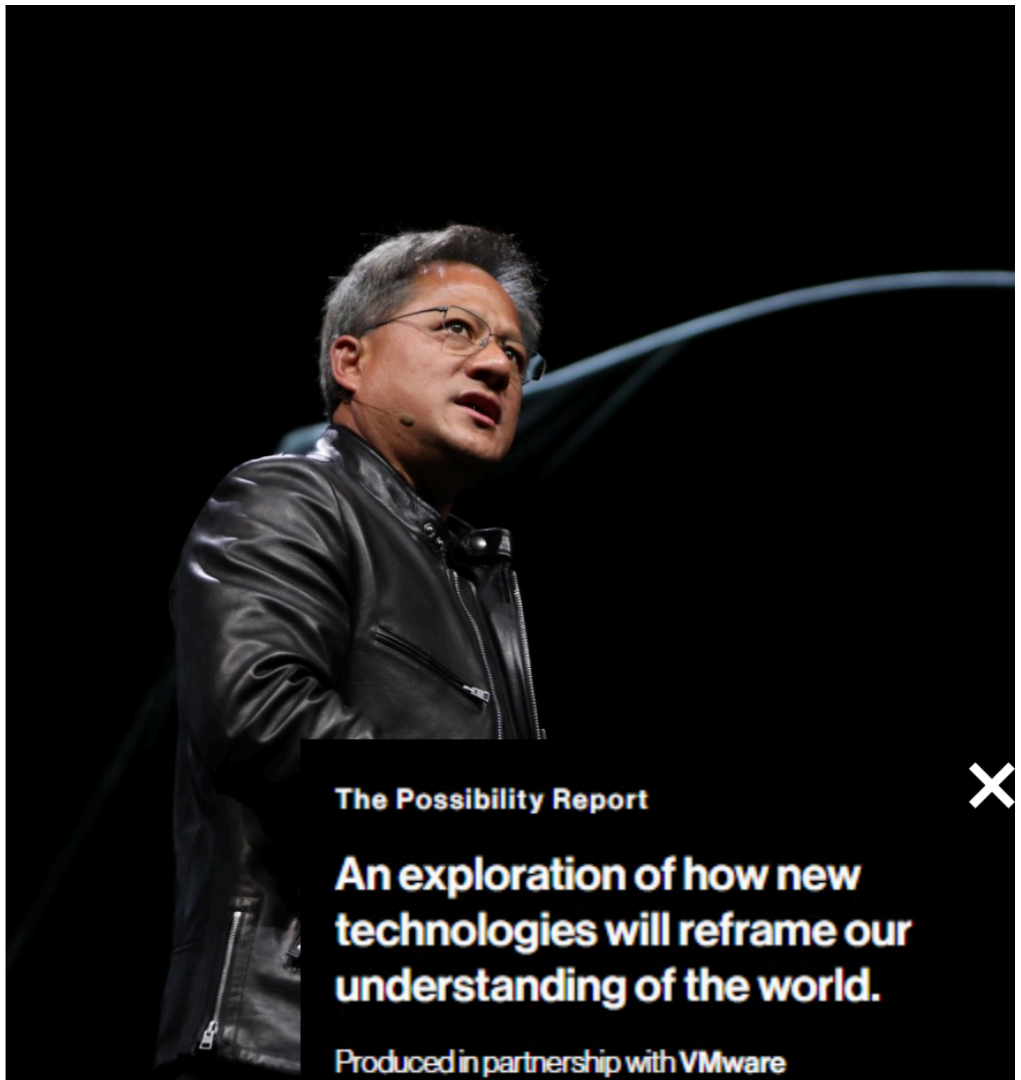
- We kunnen niet heel de wereld handmatig omzetten in software: te veel, te complex

Intelligent Machines

Nvidia CEO: Software Is Eating the World, but AI Is Going to Eat Software

Jensen Huang predicts that health care and autos are going to be transformed by artificial intelligence.

by Tom Simonite May 12, 2017



Wat is AI?

- “Standaard” definitie:
  - AI is het onderzoeksdomein dat zich bezighoudt met taken die vandaag de dag nog te moeilijk zijn voor een computer
  - (Merk op: tijdsgebonden!)

- Mijn lievelingsdefinitie:
  - Algemene patronen van menselijk denken
  - Complex gedrag met compacte, hoog-niveau instructies

Twee grote onderdelen in menselijk denken

Inductie

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}$$

Deductie

Twee grote topics in AI

- Inductie: Machine Learning, Data Mining

- Deductie:
  - In het algemeen: Kennisrepresentatie en redeneren
  - In specifieke gevallen:
    - Pad planning
    - Zoekalgoritmes
    - …

Planning van het vak

- Machine Learning
  - 2 x 1,5u hoorcollege
  - 3 x 1,5u practica

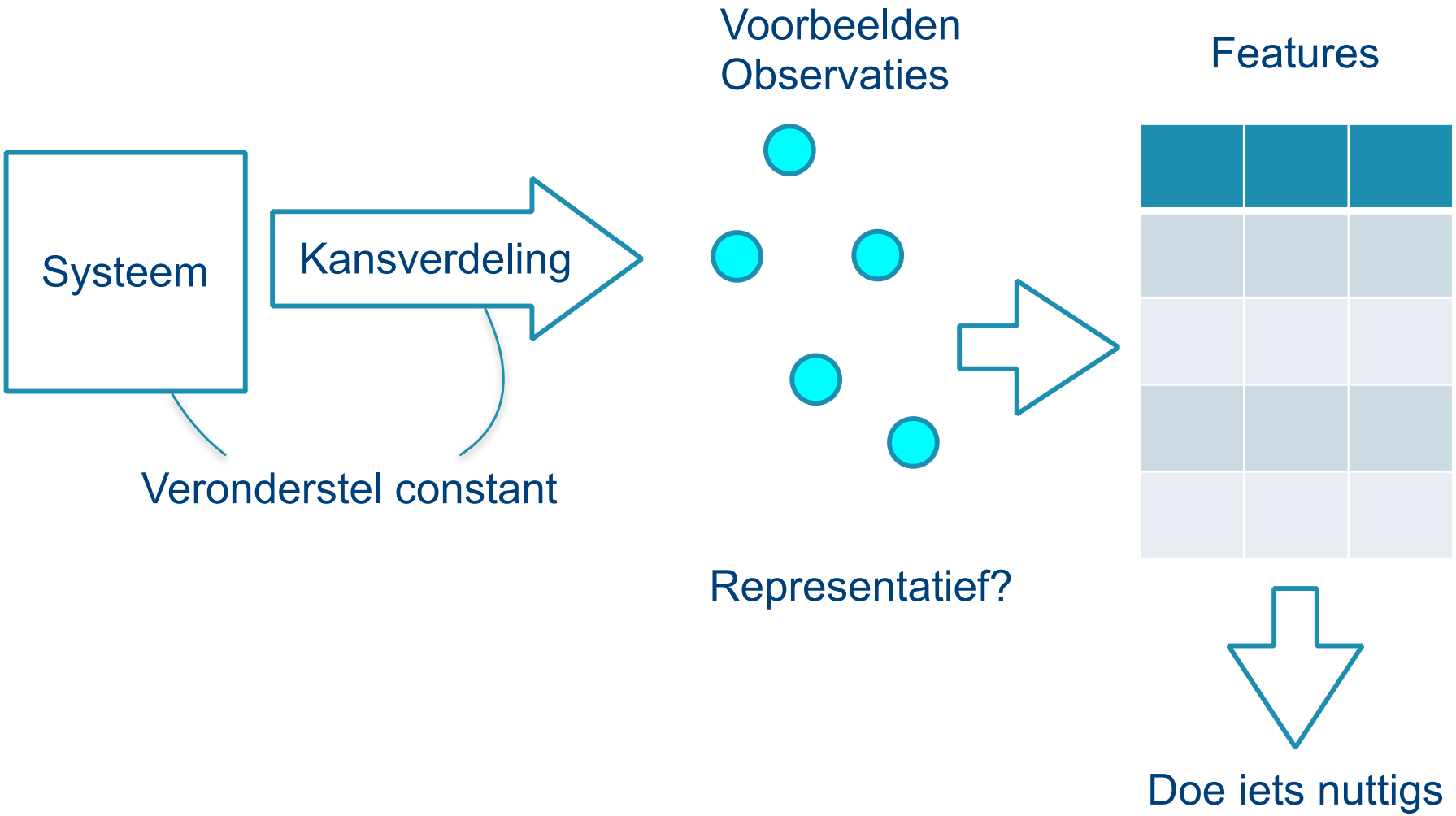
- Kennisrepresentatie en redeneren
  - 4 x 1,5u hoorcollege
  - 6 x 2u practica

- Zoekalgoritmes
  - 6 x 1,5u hoorcollege
  - 6 x 2u practica

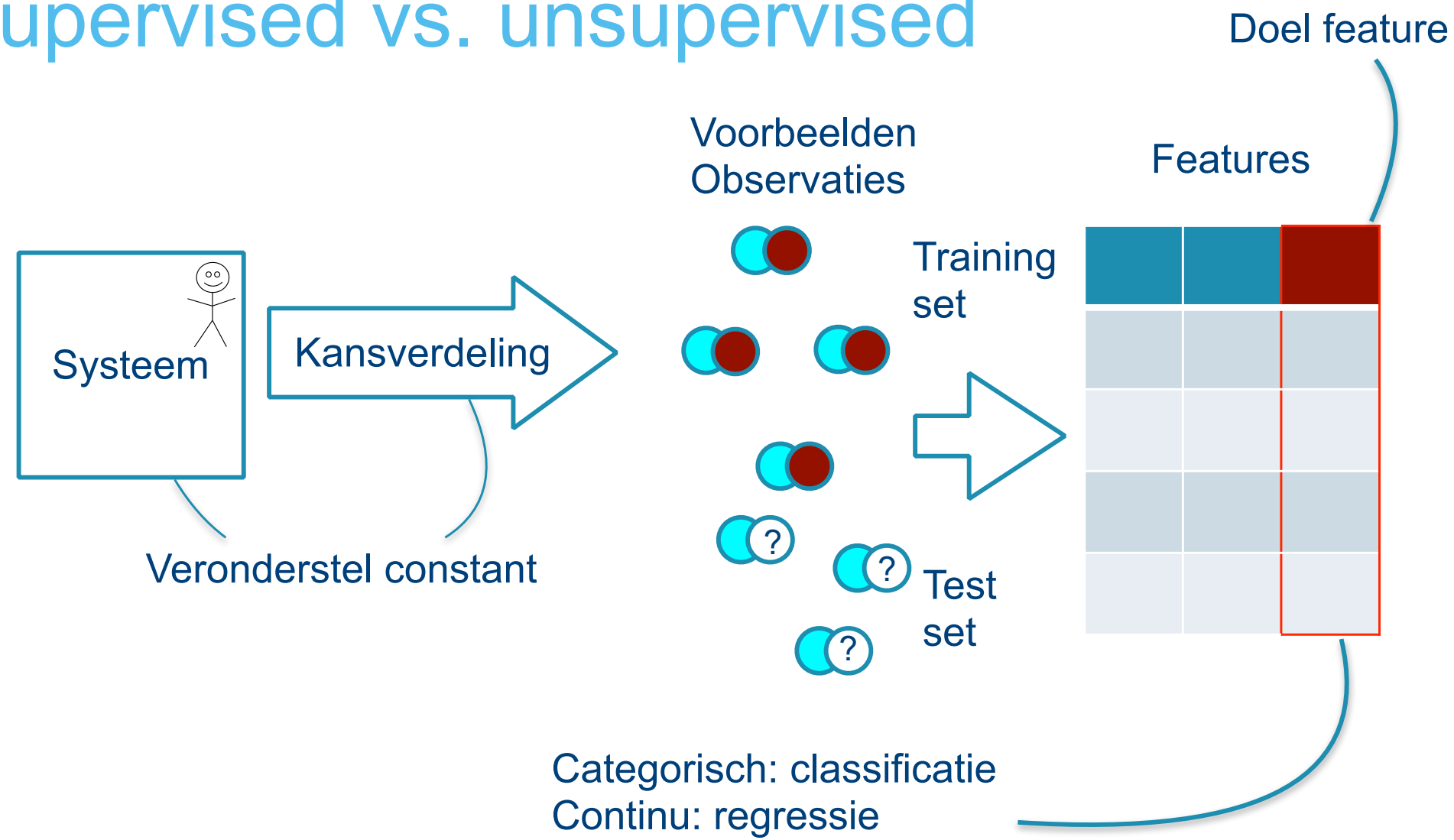
Inleiding: Verschillende soorten Machine Learning

KU LEUVEN

Machine Learning

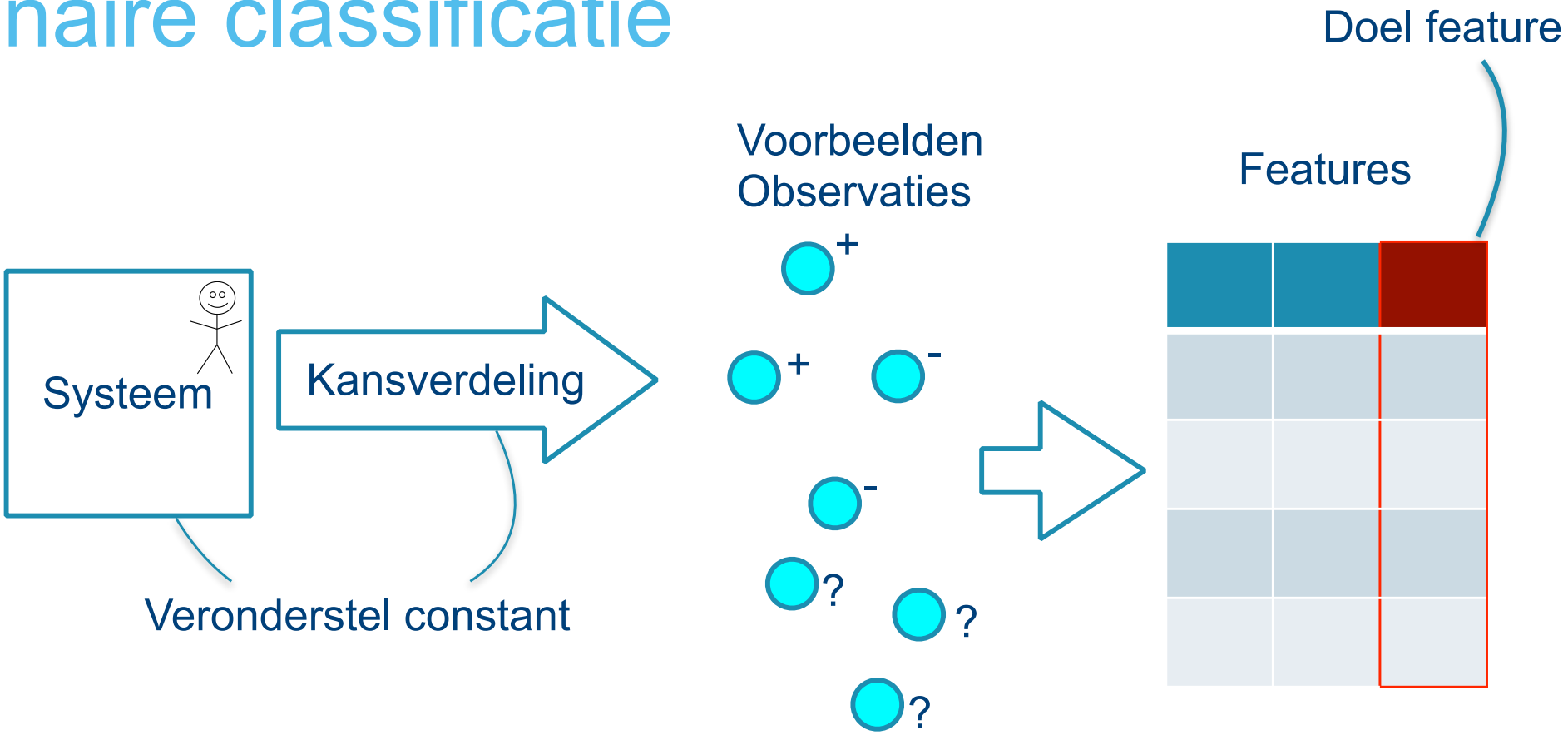


Supervised vs. unsupervised



KU LEUVEN

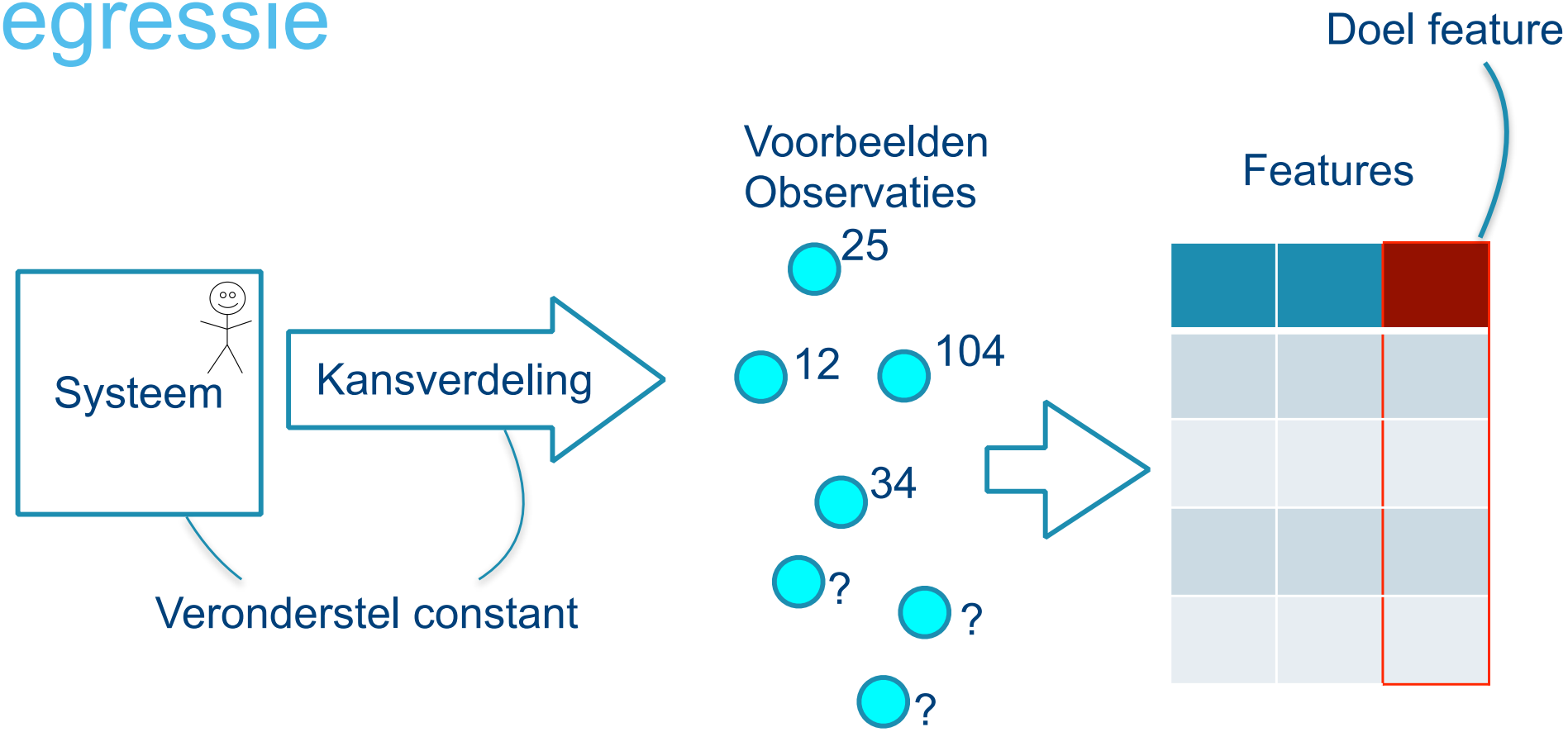
Binaire classificatie



$$f(x_1, \ldots, x_n) = y \in \{+, -\}$$

Neuraal netwerk, Bayesiaans netwerk, SVM, beslissingsboom (random forest), …

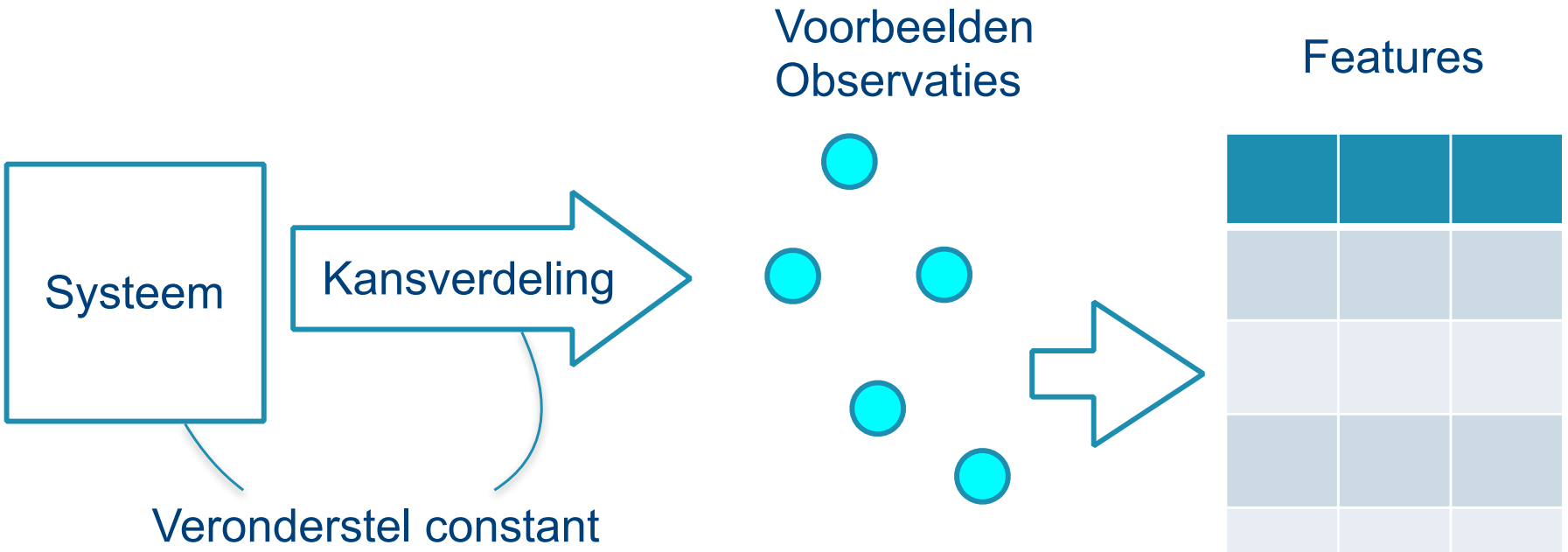
Regressie



$$f(x_1, \ldots, x_n) = y \in \mathbb{R}$$

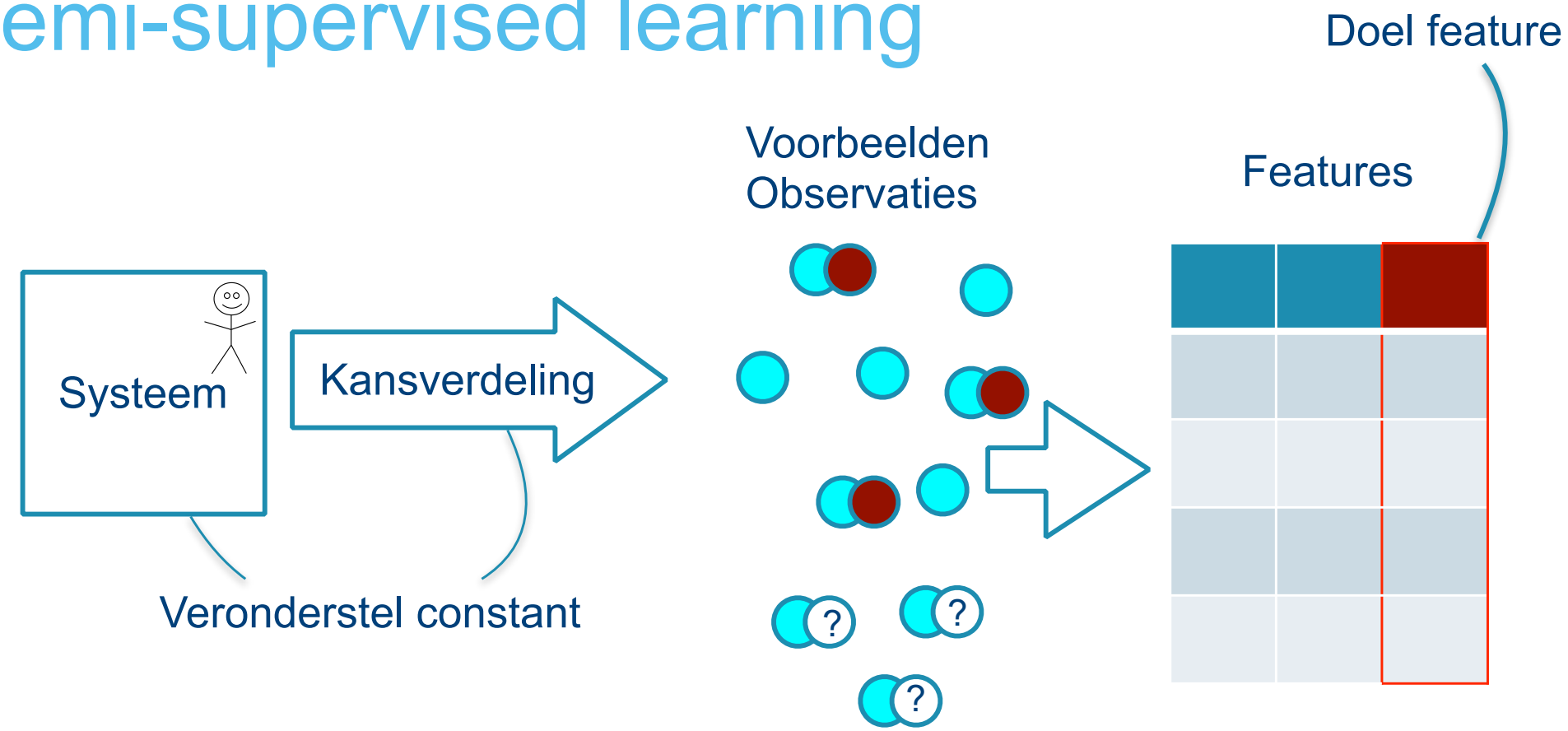
Lineaire regressie, Bayesiaans netwerk, neuraal netwerk, regressieboom, …

Unsupervised Learning

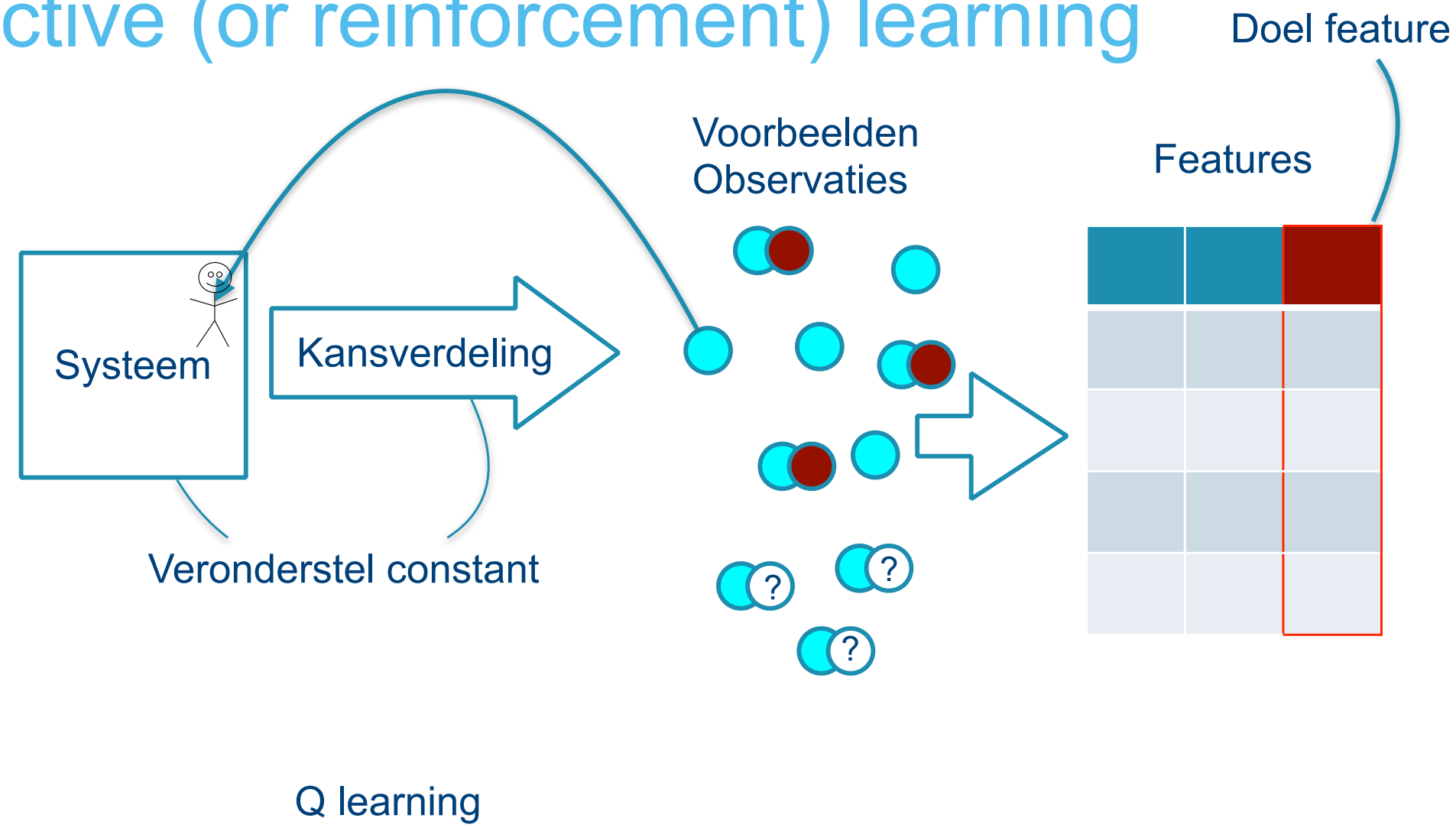


Clustering, pattern mining, Bayesian network learning…

Semi-supervised learning

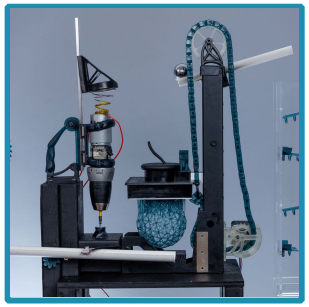


Active (or reinforcement) learning



Q learning

White box vs. black box



Voorbeelden
Observaties

Features

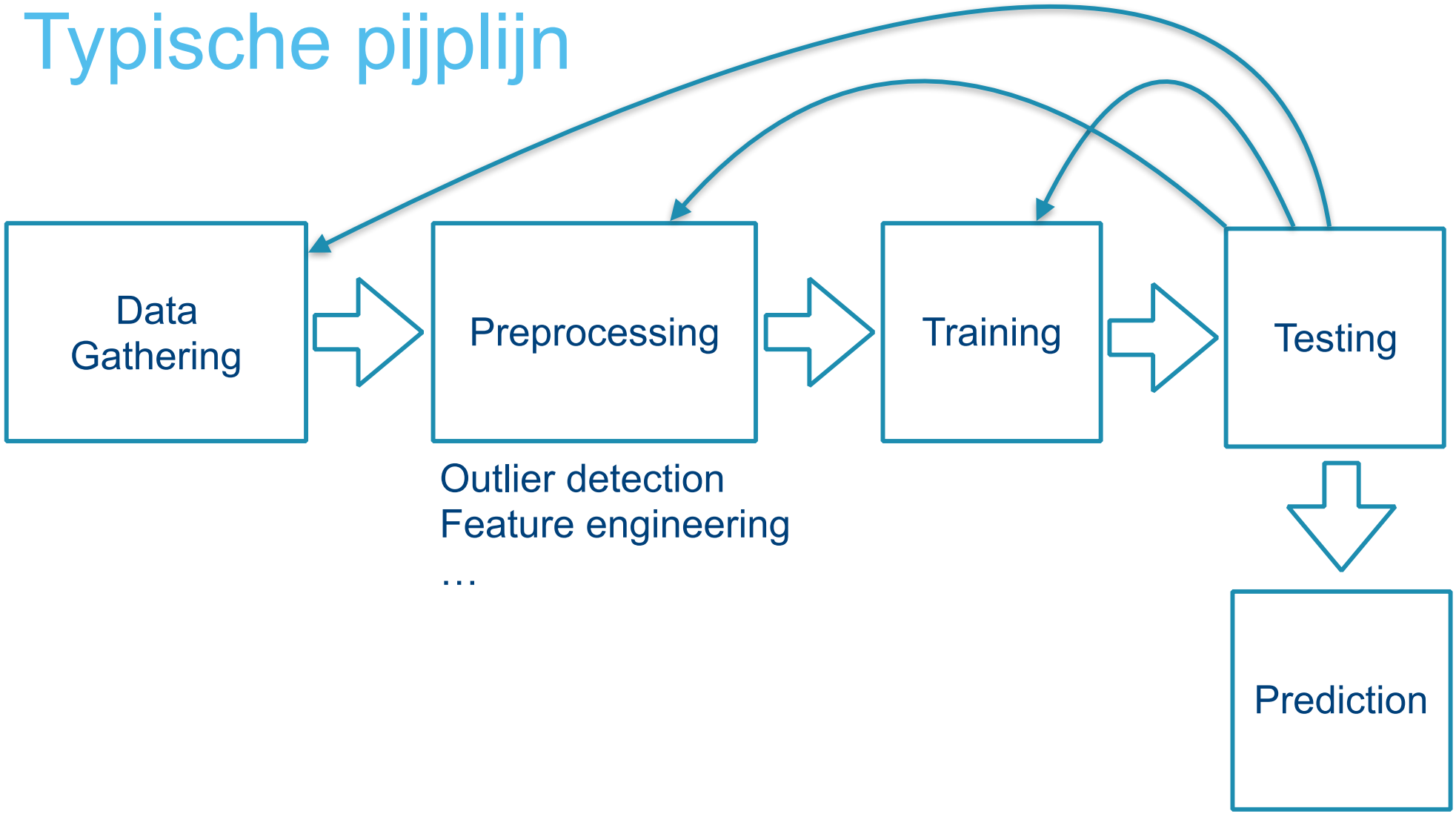
Lineaire regressie
Bayesiaans netwerk
Beslissingsboom
…

Neuraal netwerk
SVM
…

Interpreteerbaarheid
Domein kennis

Doe iets nuttigs

Typische pijplijn



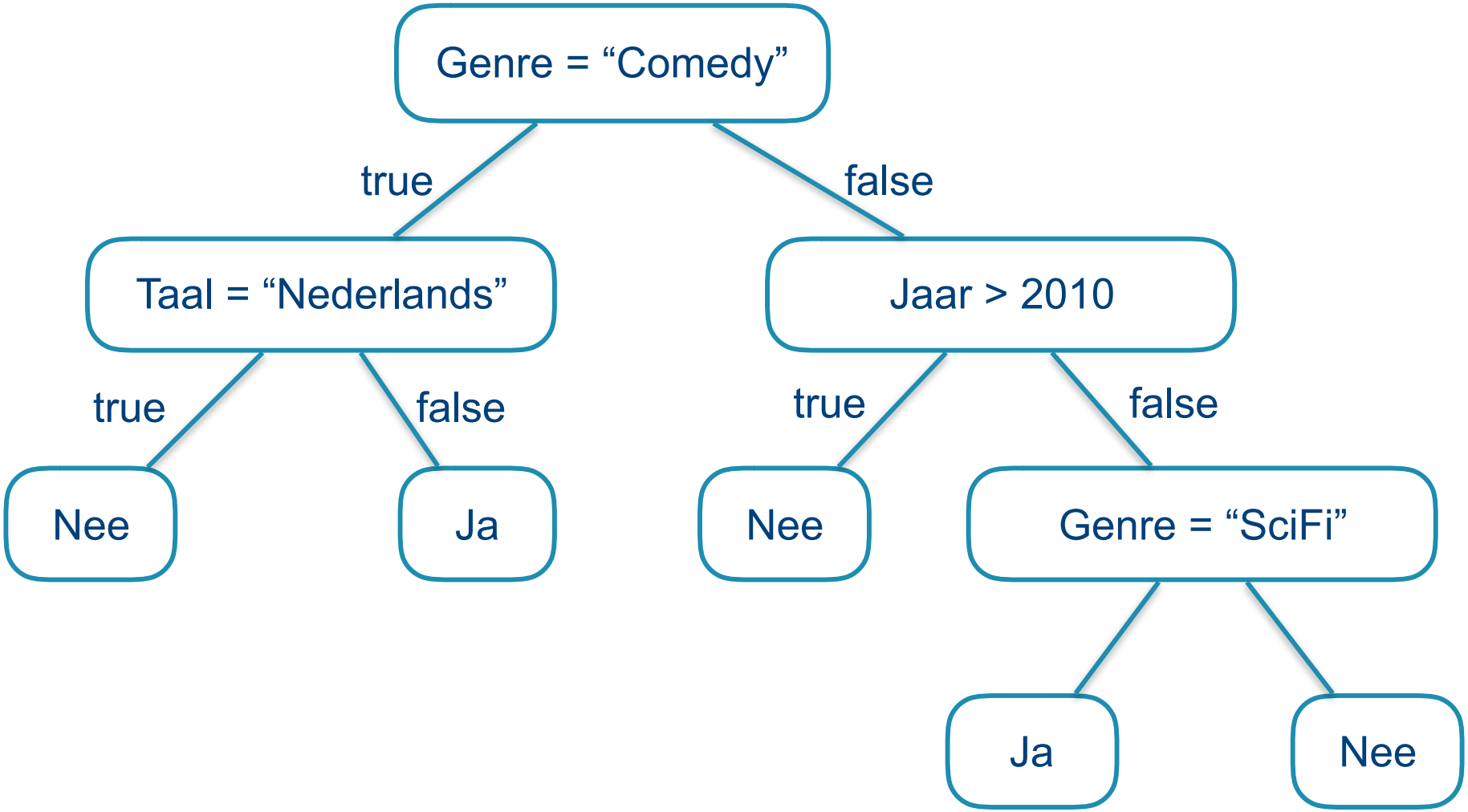
Concreet voorbeeld: beslissingsbomen

KU LEUVEN

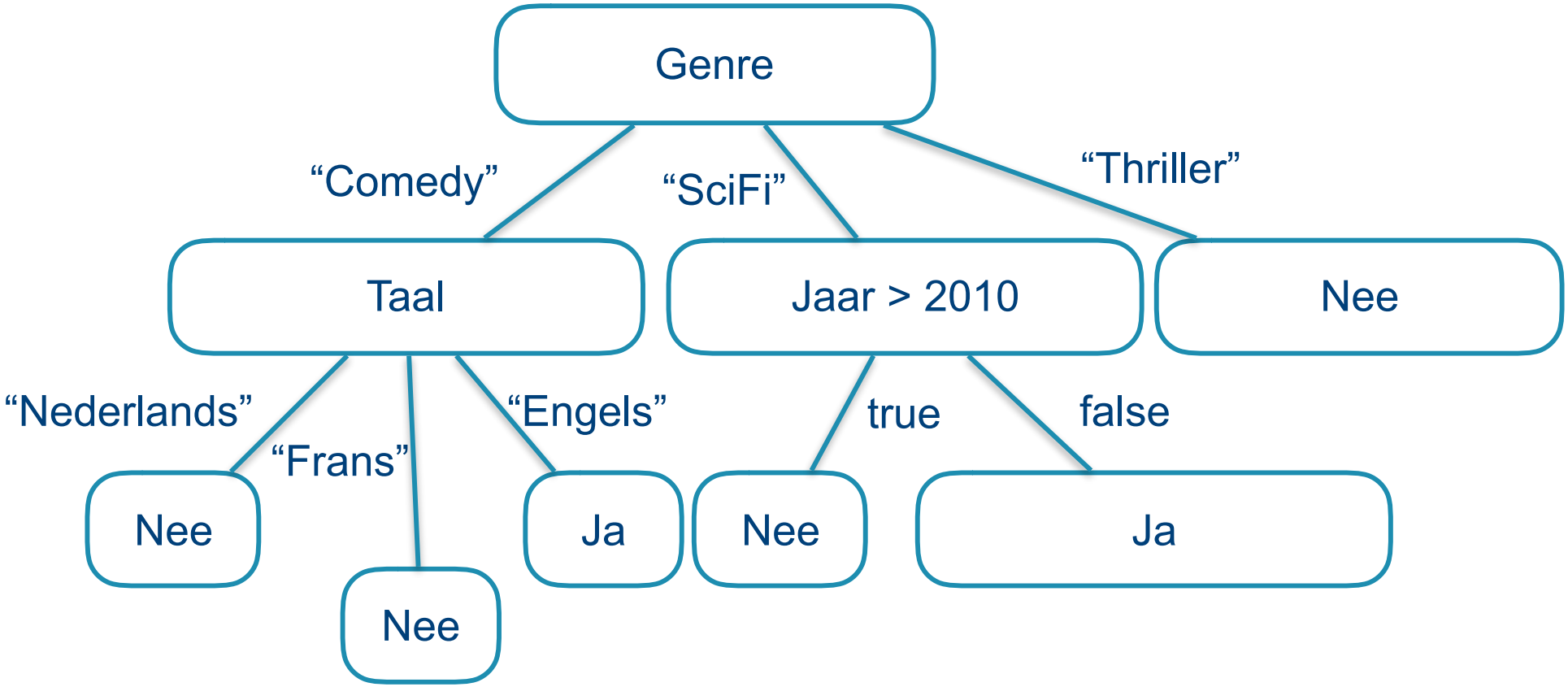
Voorbeeld

Genre	Jaar	Taal	Regisseur	Goed
Comedy	2017	Nederlands	Jan Verheyen	Nee
SciFi	2016	Engels	Gareth Edwards	Ja

Voorbeeld: binaire beslissingsboom



Voorbeeld: beslissingsboom



Kwaliteit meten

- Data = training set (70%) + test set (30%)

- Evalueer model op test set
  - True positives
  - False positives
  - True negatives
  - False negative

- Maatstaf: $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Leren van beslissingsboom

- Beslissingsboom met optimale accuracy op training set?
  - Niet te berekenen
- Greedy algoritme
  - Om zo hoog mogelijk te geraken
  - Kies telkens de weg die het steilste is
  - Eenvoudig, maar leidt tot lokaal ipv. globaal optimum

ID3 algoritme (andere: C4.5, CART)

- Begin met wortel
- TODO = { blad | blad **moet nog gesplitst worden** }
- Zolang als TODO =/= {}
  - Kies blad uit TODO
  - Kies **beste feature** om op te splitsen
  - Splits blad
  - TODO = { blad | blad **moet nog gesplitst worden** }

Voordeel van deze strategie: kleine bomen, Occam's razor

Welke bladeren splitsen?

- We laten training set indalen in boom
- In blad voorspellen we meerderheidsklasse
- In gemengd blad maken we sowieso fout
- Dus: splits tot 100% puurheid (?)

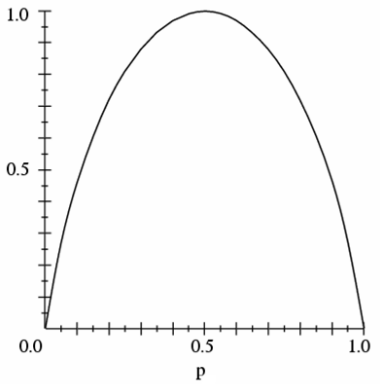
Welk criterium voor splitsing?

- Einddoel is pure bladeren
- Greedy strategie:
  Kies criterium dat puurheid meeste vergroot

Welk criterium voor splitsing?

- Hoe puurheid meten?

$$Entropy = \sum_{class\ c} p_c\ log_2 \frac{1}{p_c}$$



- Winst in puurheid: InformationGain(D,F)

$$Entropy(D) - \sum_{f \in dom(F)} \frac{|D[F=f]|}{|D|} Entropy(D[F=f])$$

ID3 algoritme

- Begin met wortel
- TODO = { blad | blad **is niet puur** } (???)
- Zolang als TODO =/= {}
  - Kies blad uit TODO
  - Kies **feature F met maximale Gain(D,F)** om op te splitsen, met D de data in blad
  - Splits blad volgens F

Voordeel van deze strategie: kleine bomen, Occam's razor

Gevaar voor overfitting

- Splitsen tot pure bladeren
  - Diepe bomen
  - Alle details van training set
  - Wat met ruis?

- Alternatief
  - Splits training set in training + validatie
  - Bereken of accuracy op **validatie** set nog steeds zou stijgen bij split
  - Nadeel: je verliest data
- Andere alternatieven: coverage of post-pruning

Wat met numerieke variabelen? (input)

- Continu of discreet met grote range (bv. jaartal)

- Op voorhand discretiseren
  - Nadeel: hoe grenzen kiezen?

- Dynamisch $x < n$ bepalen
  - Sorteer datapunten volgens numerieke variabele
  - Zoek overgangen n_1, n_2, … tussen klassen
  - Bereken test $x < n_i$ met grootste information gain

Gebruik van bomen

- Nadelen
  - Greedy betekent mogelijk niet-optimaal
  - Van gecorreleerde features wordt er typisch maar 1 gebruikt

- Voordelen
  - Eenvoudig te interpreteren
  - Efficient
  - Redelijk flexibel en robuust (bv. missing values)

Evaluatie van algemene techniek

- 10-fold crossvalidation
  - Meer robuuste evaluatie

- Kan ook gebruikt worden om beste boom te selecteren
  - Dwz. de boom met de beste accuracy op de *andere* 9 test sets

Uitbreidingen

- Random forests
  - Leer veel lichtjes verschillende bomen
  - Verschillende gecorreleerde features kunnen allemaal gebruikt worden
  - Robuuster tegen overfitting
  - Hoe? Idee van n-fold crossfold validation

- Boosting
  - Leer sequentie van kleine bomen, waarbij volgende boom focust op fouten van vorige

Case study: E'XperT project

- Agidens en Oiltanking Stolthaven Antwerp
- Problem
  - Too many alarms in control room
  - Distraction, large workload
- Solve with AI technology
- Support system for process engineer, instead of operator
  - Detect "interesting" alarms
  - Analyse why these alarms occur

General approach

- Datamining: detect general pattern in historical data
- Not for prediction, but for analysis by engineer
  - Readability of patterns is crucial
- Decision tree learner

Detect patterns in large set of examples

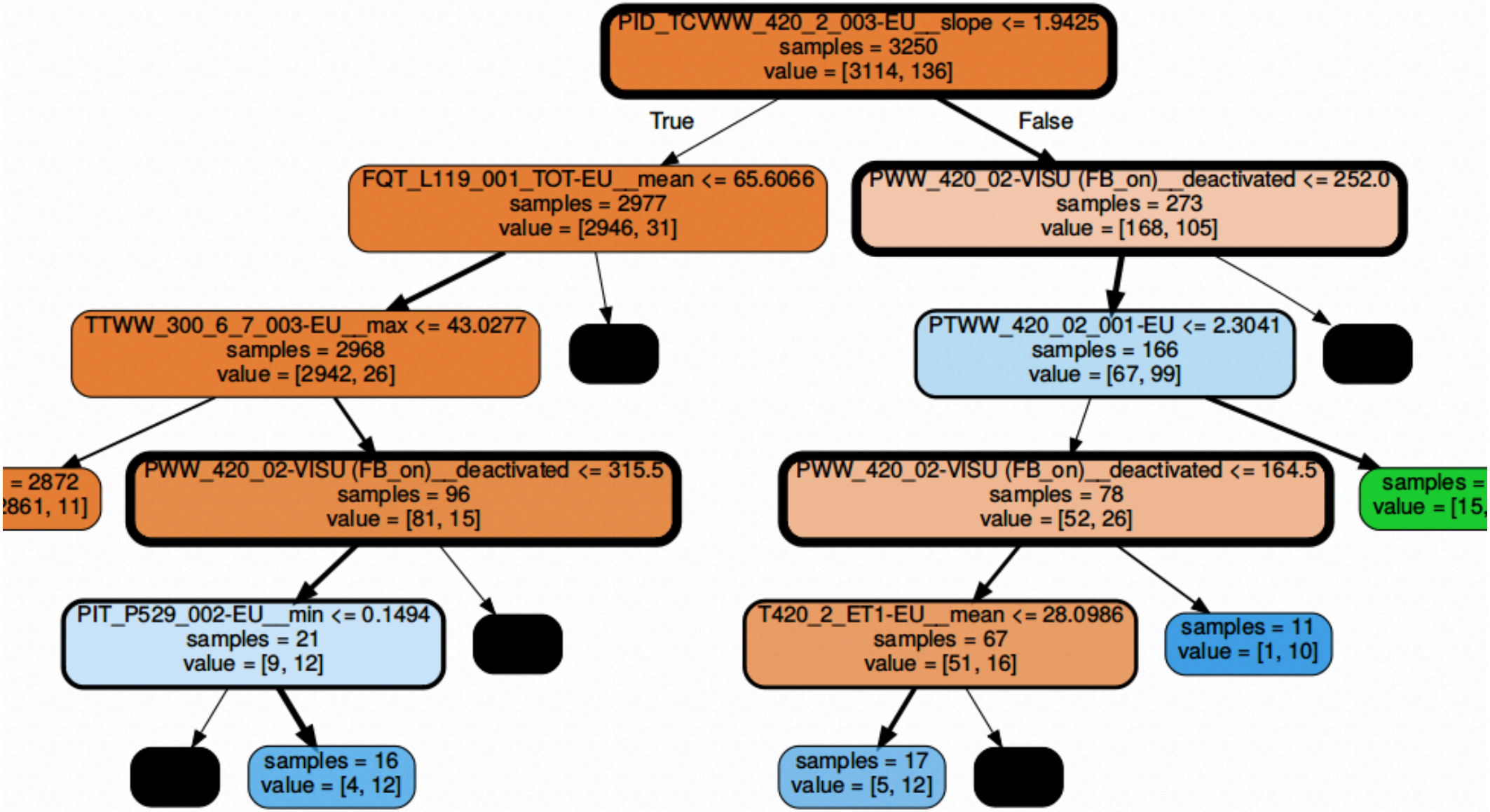
State at some point in time (including summary statistic of last 30min.)

Sensor 1	Sensor 2	Valve 1	…	Label
54.5	22	1	…	+
123.4	34	1	…	+
76.2	25	0	…	-
84.5	27	1	…	-
…	…	…	…	…

"Feature" of the terminal

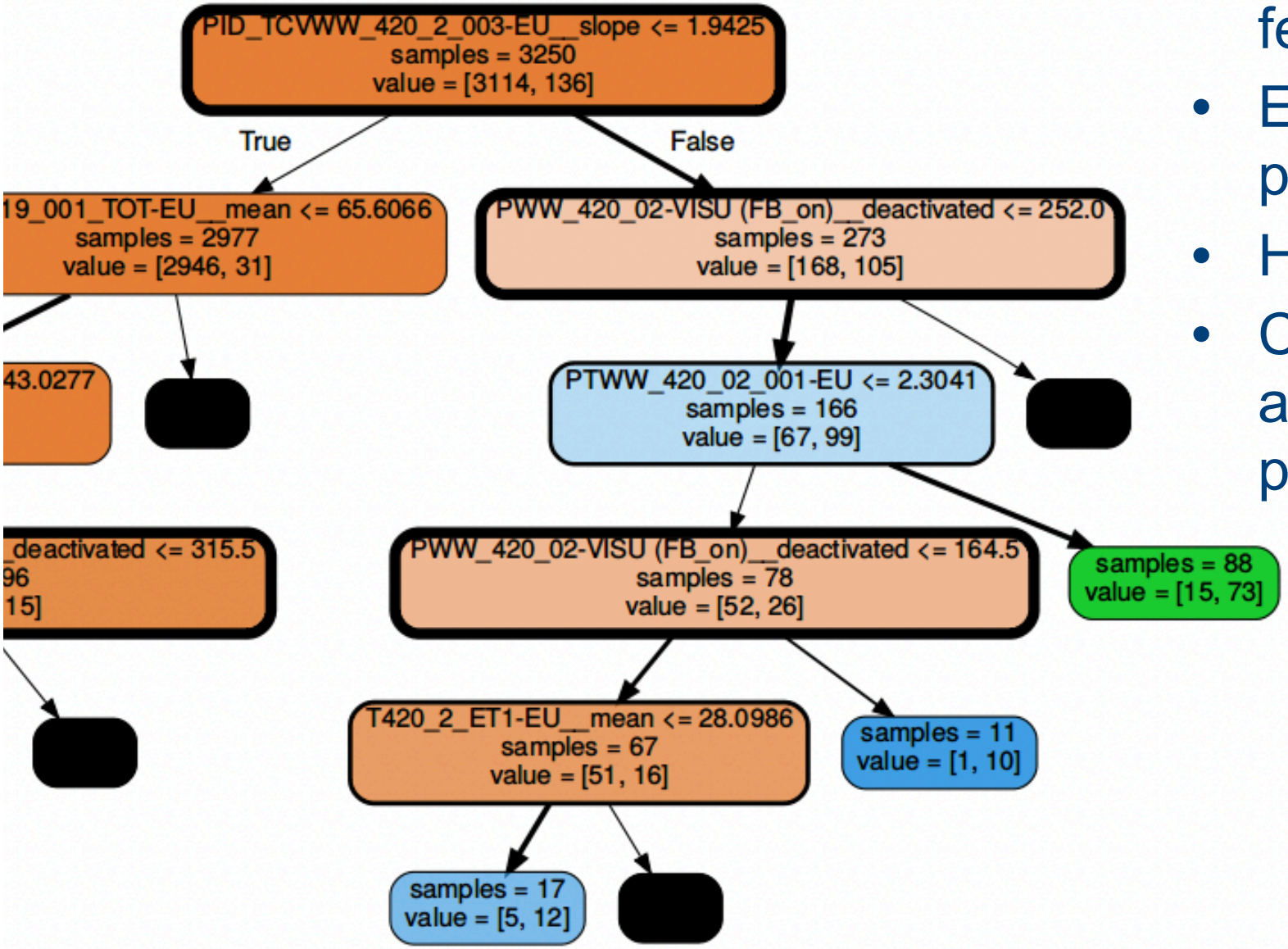
Whether the alarm that we are interested in occurred or not

Decision tree



KU LEUVEN

Visualisation of tree



- Border: importance of feature
- Edge: flow of positive examples
- Hiding of subtrees
- Color: ratio +/- and highlighting of positive leaves

Problems

- Still hard to read for domain experts
  - Even after multiple workshops

- Correlated features that are all relevant
  - Decision tree typically only shows one

Problems

- Still hard to read for domain experts
  - Even after multiple workshops
  - → **Text-based representation**

- Correlated features that are all relevant
  - Decision tree typically only shows one
  - → **Random forest**

Text-based representation for branch

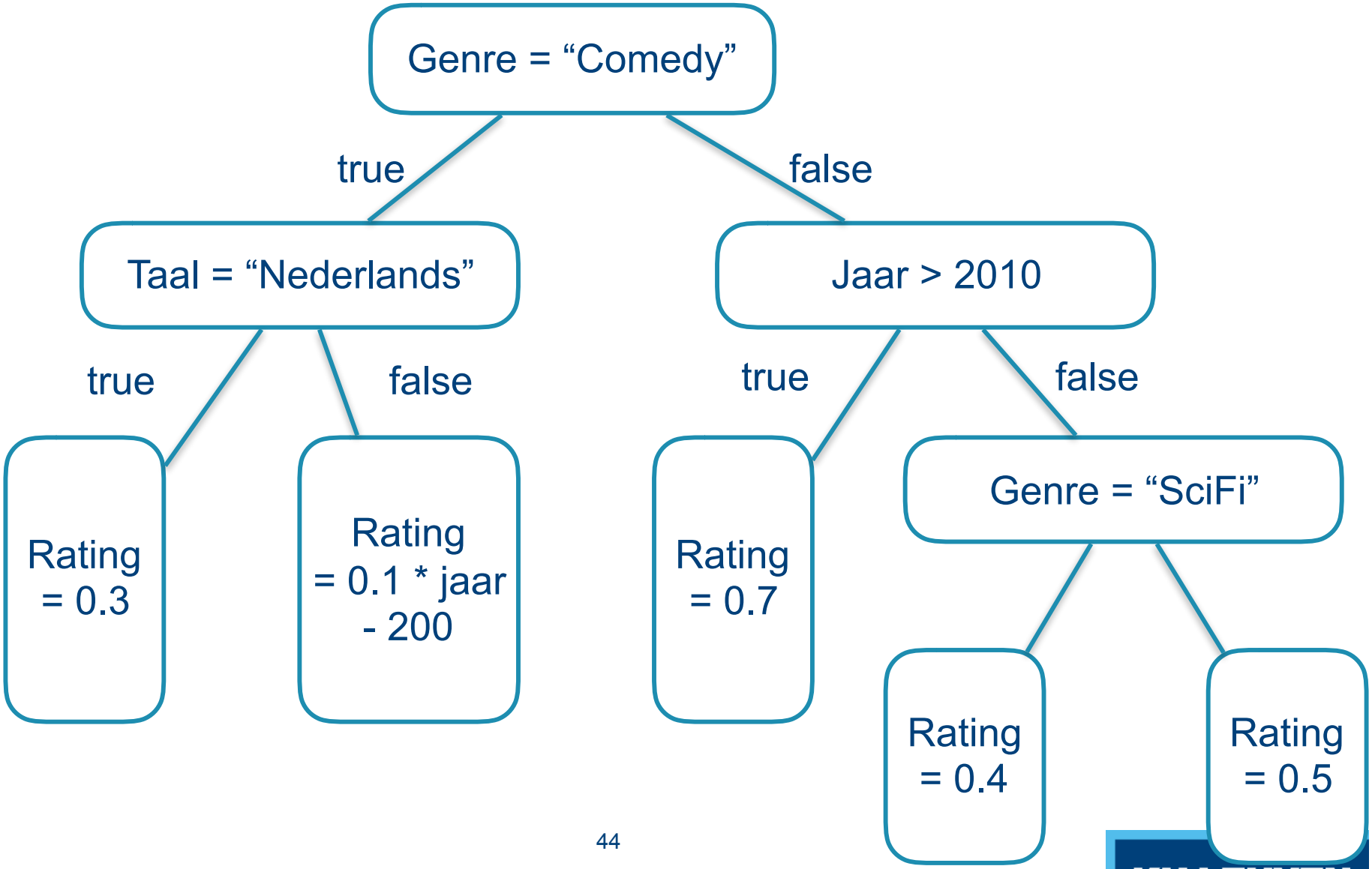
The alarm occurred in 73 out of 88 cases where all the following conditions are met: the last half hour the value of measurement_1 increased by more than 1.9; signal_1 became inactive less than 252 minutes ago; the value of measurement_2 lies between 2.3 and 2.9.

This describes 53% of the 136 occurrences of this alarm.

Regressie bomen

Genre	Jaar	Taal	Regisseur	Rating
Comedy	2017	Nederlands	Jan Verheyen	3.4
SciFi	2016	Engels	Gareth Edwards	6.7

Regression bomen



Regressie bomen leren

- Voorspelling in blad: ipv. meest voorkomende klasse, doe standaard LSE regressie

- Splitsen: ipv. information gain, gebruik:

$$SSE(D) = \sum_{d \in D} (d_f - \overline{D_f})^2$$

$$argmin_F \sum_{f \in dom(F)} SSE(D[F = f])$$

- Stoppen: SSE op testset verbetert niet meer

Regressie: neurale netwerken

One-hot encoding

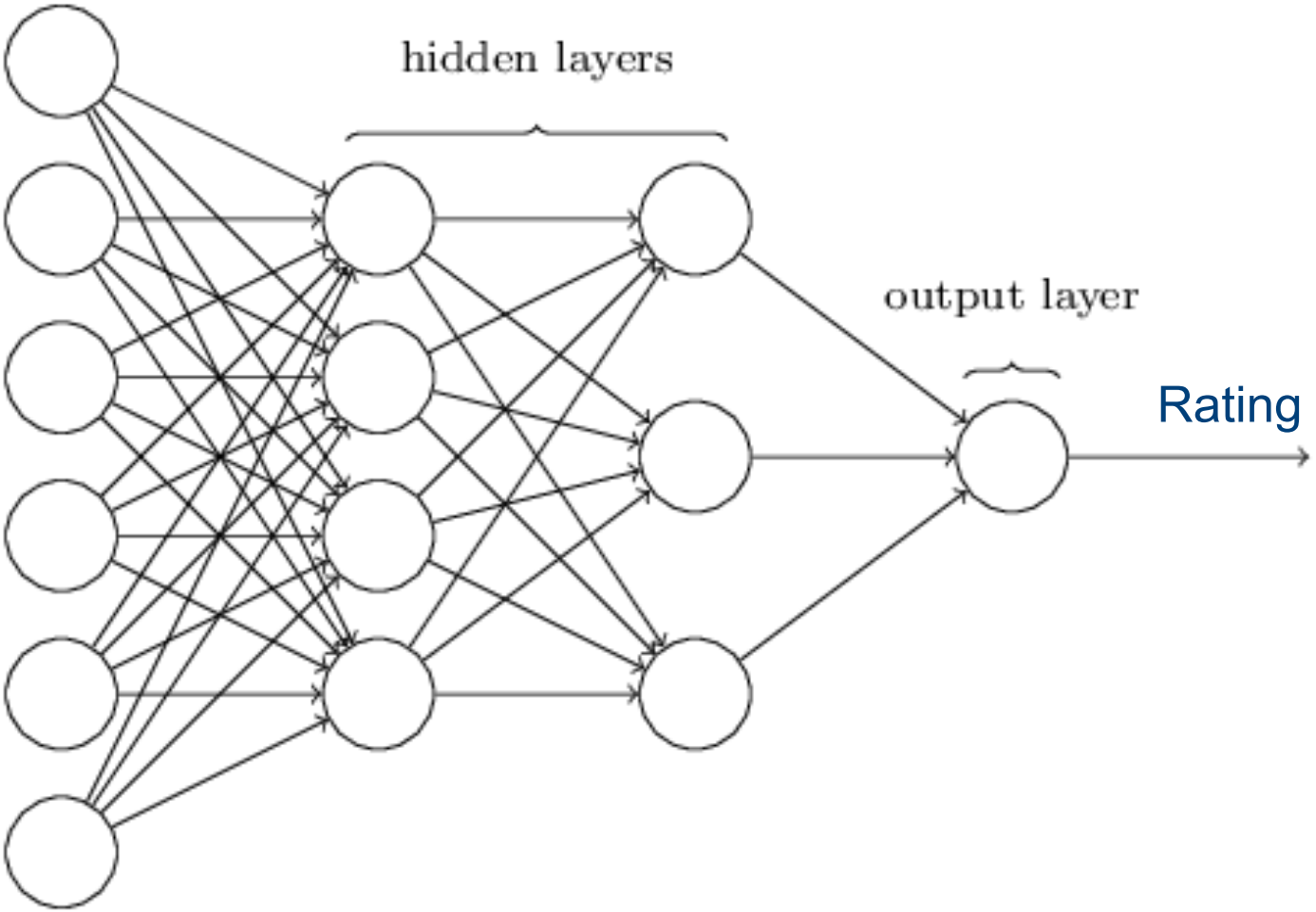
Jaar

Genre = “Scifi”

Genre = “Comedy"

Genre = “Thriller”

…



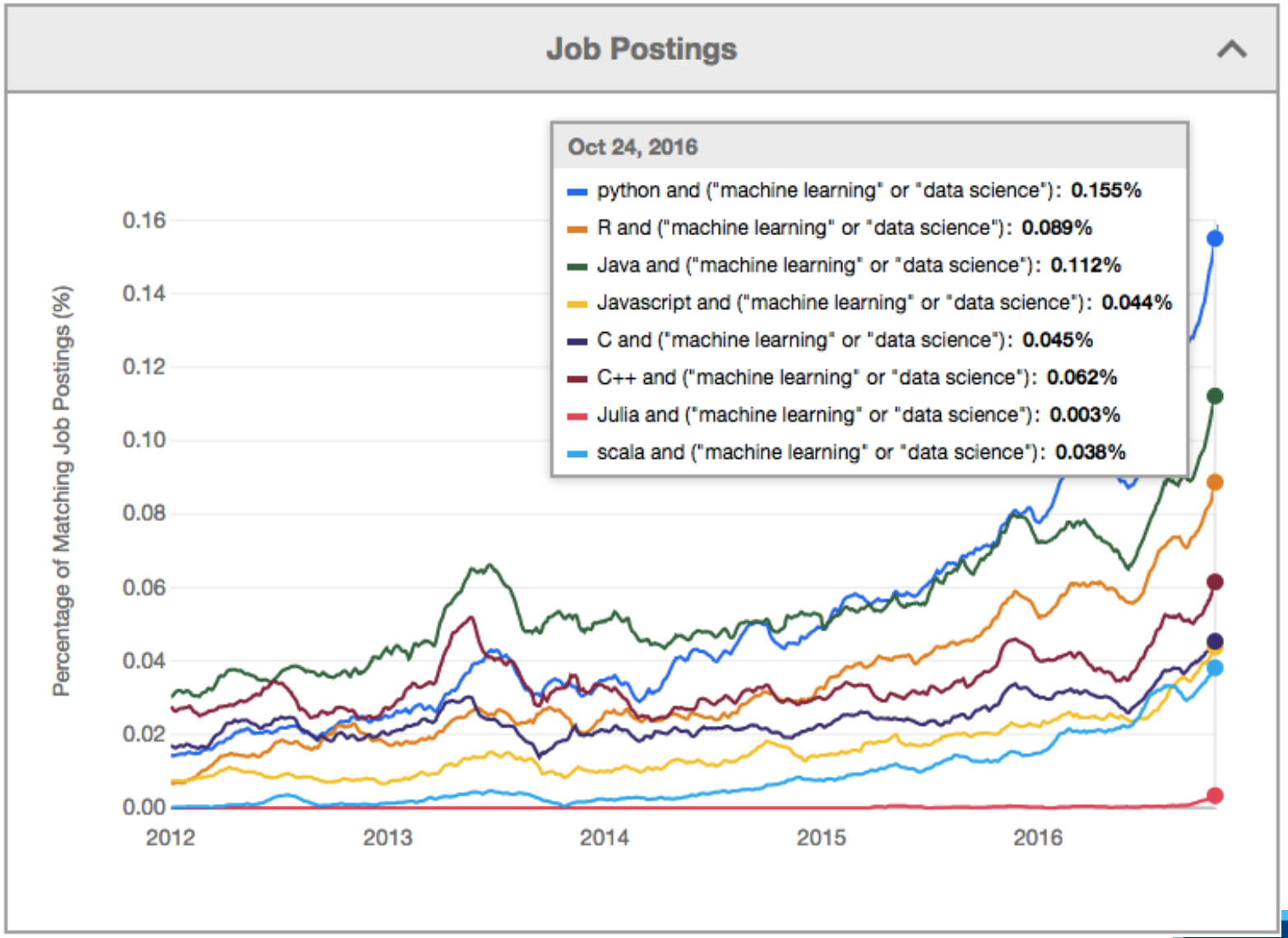
Voorbeeld: anomaly detection in SAP server logs (ism. Oxya)

Started	Server	Transaction	Program Function	T	Scr.	Wp	User	Response time (ms)	Time in WPs (ms)	Wait time (ms)	...	Tcode	CUA internal command	Bytes transf. (total)	Number of RFCs
2017-10-12 08:00:10	SAP133_PRD_00	NaN	<HANDLE RFC>	D	NaN	1.0	IBIDAN_Y	0.0	0.0	0.0	...	NaN	NaN	0.0	0.0
2017-10-12 08:00:24	SAP134_PRD_00	VL10E	RVV50R10C	D	120.0	2.0	CATRYS	741.0	741.0	0.0	...	VL10E	SICH_T	89794.0	0.0
2017-10-12 08:00:31	SAP134_PRD_00	ZSFI	ZBT_PP_INSP	D	9000.0	17.0	P150I07	579.0	24.0	0.0	...	ZSFI	%_GC 118 1	75182.0	0.0
2017-10-12 08:00:38	SAP133_PRD_00	VL02N	SAPMV50A	D	4004.0	19.0	WENGRZYL	56.0	21.0	0.0	...	VL02N	%_GC 105 25	0.0	0.0
2017-10-12 08:00:45	SAP134_PRD_00	ZPP01	ZBT_PP_SCHED_ALV	D	9100.0	0.0	XOLOCOTG	599.0	34.0	0.0	...	ZPP01	NaN	0.0	0.0

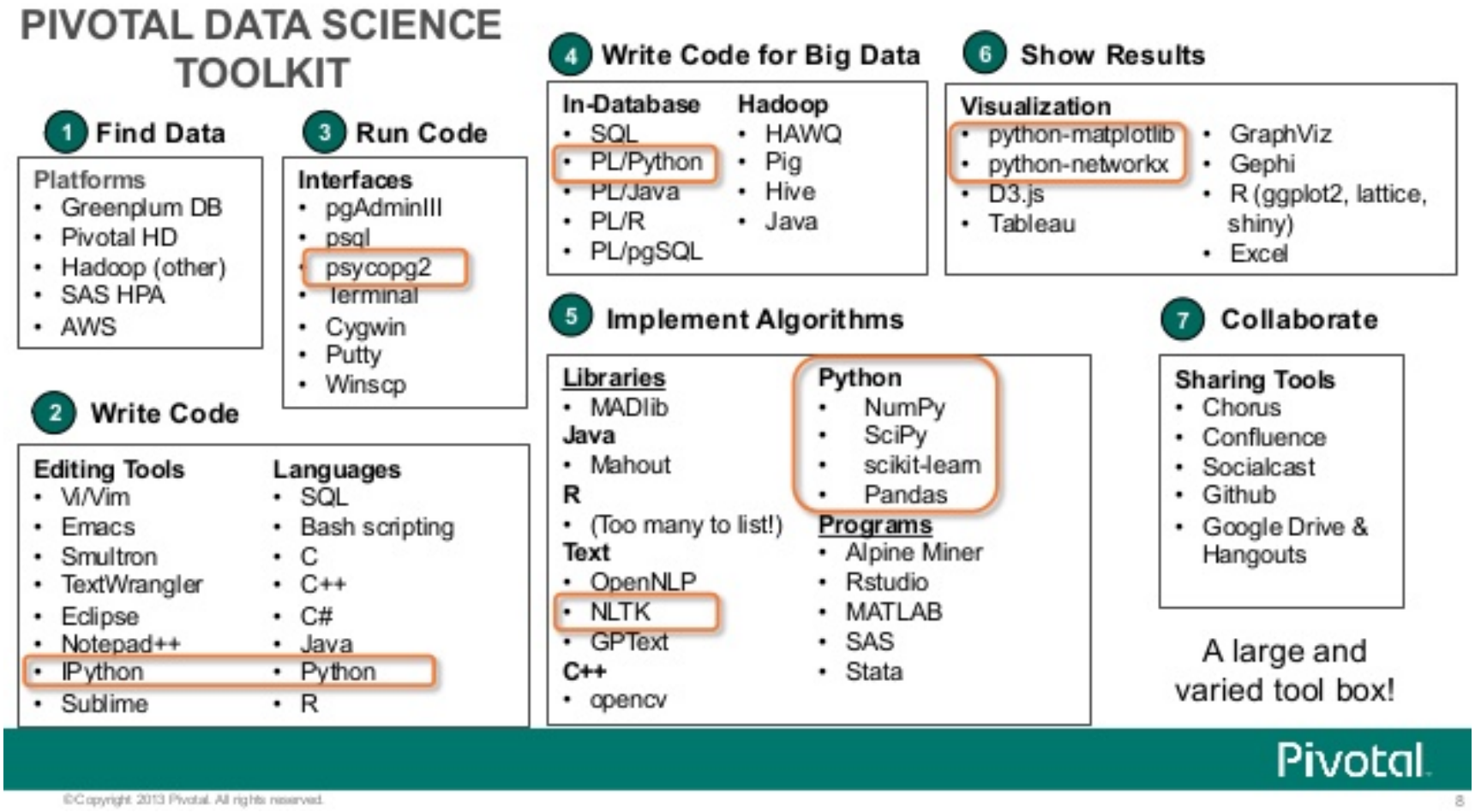
Praktisch: Data science in Python

KU LEUVEN

Data science: in Python



Python: (OO) bibliotheken



Enkele bibliotheken

- **numpy**: basis
- **matplotlib**: visualisatie
- **pandas**: manipulatie
  (inlezen, selecteren, transformeren, statistieken, …)
- **scikit-learn**: modellen op data passen

Proces

- Bekijken
- Inlezen
- Verkennen

} pandas

- Verschillende modellen passen
- Resultaten visualiseren

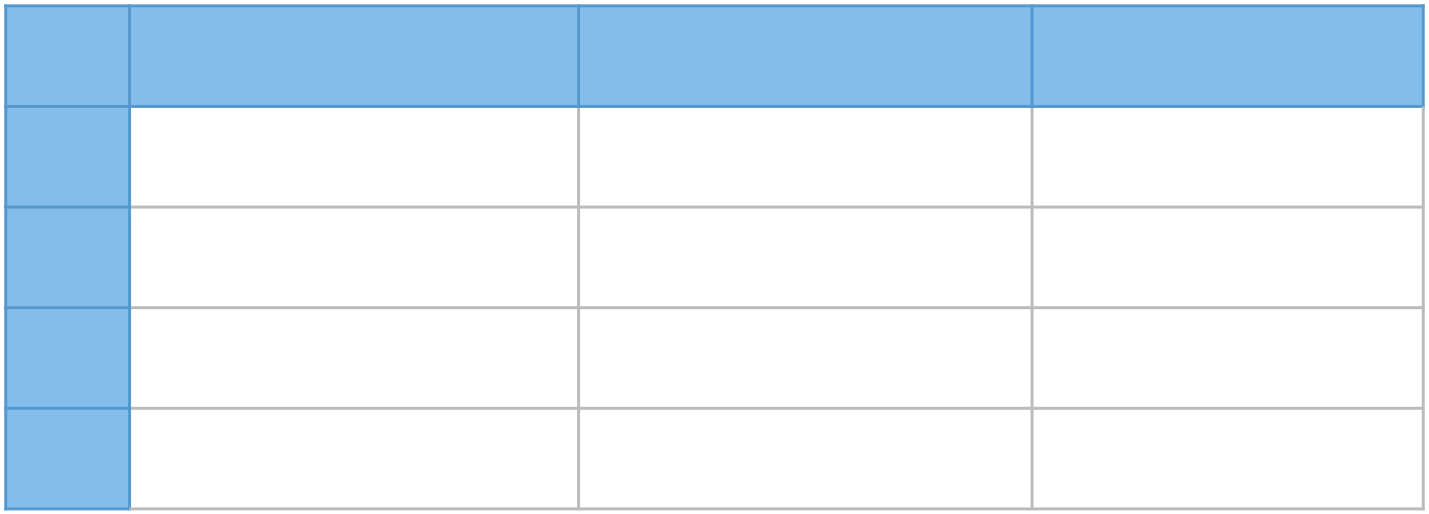
} scikit-learn

matplotlib

DataFrame

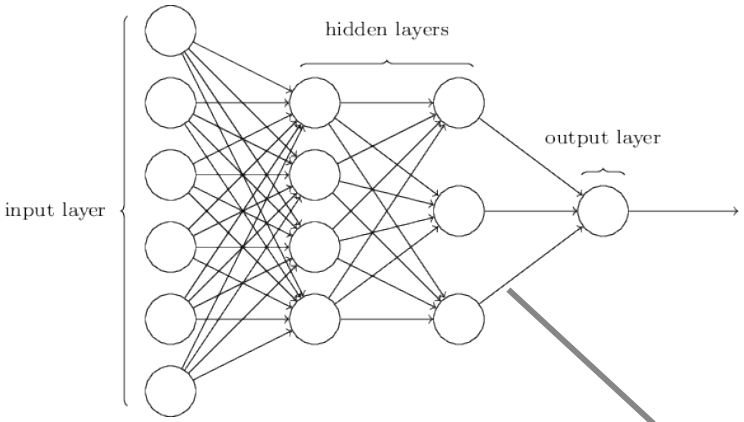
df.colums

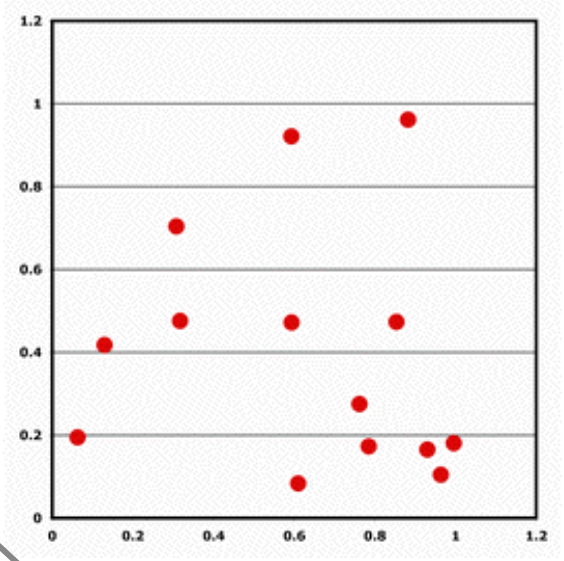
df.index

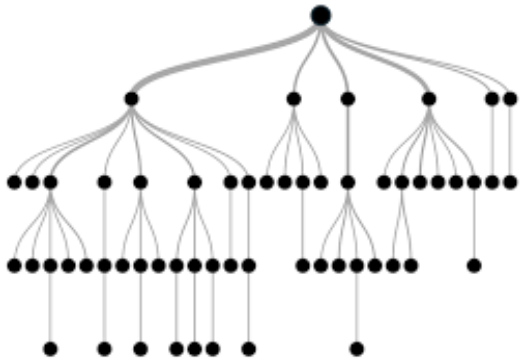


methodes om te bekijken, selecteren, visualiseren, …

SciKit-learn







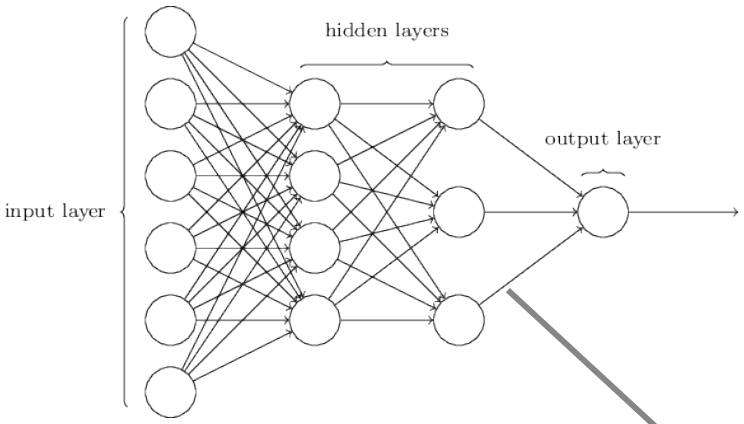
$$f(x) = ax + b$$

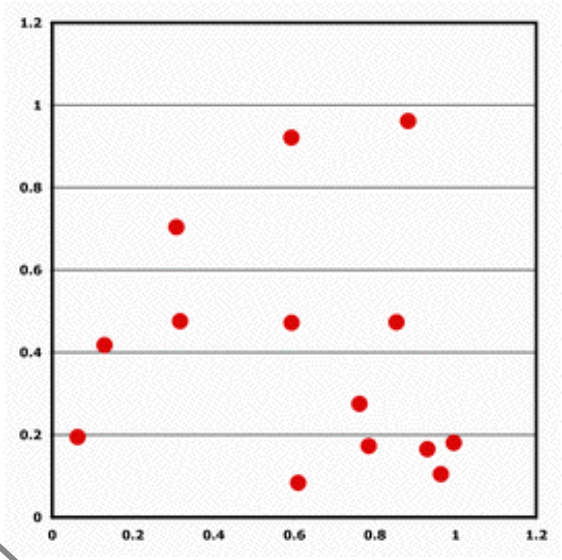
$$f(\cdot)$$

$$x \longrightarrow y$$

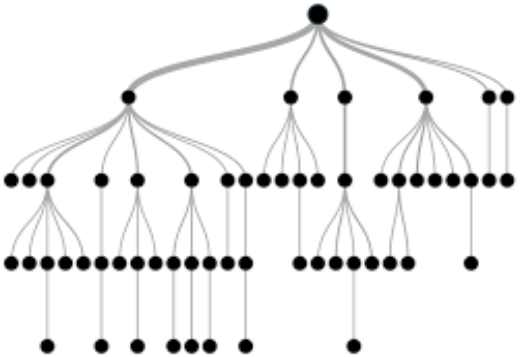
predictie

SciKit-learn





BaseEstimator



$f(x) = ax + b$

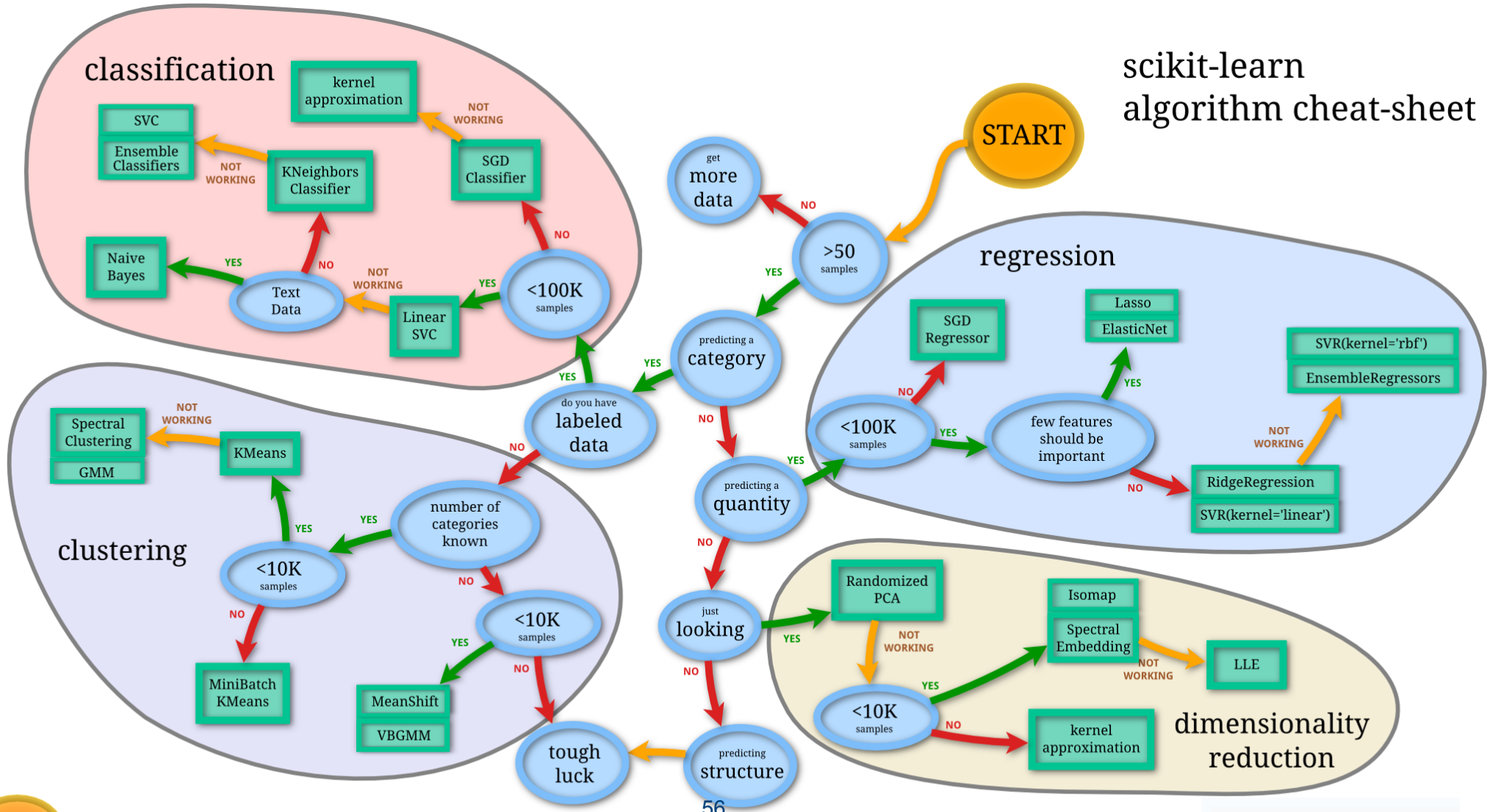
est.fit(x,y)

$f(\cdot)$

x → y

est.predict(x)

KU LEUVEN


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
do you have labeled data
YES
<100K samples
NO
Naive Bayes
YES
Text Data
NO
Linear SVC
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
NOT WORKING
kernel approximation
SGD Classifier
NO
classification
NO
predicting a quantity
YES
<100K samples
NO
SGD Regressor
YES
few features should be important
YES
Lasso
ElasticNet
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
regression
NO
just looking
YES
Randomized PCA
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
NO
number of categories known
YES
<10K samples
NO
MiniBatch KMeans
YES
<10K samples
NO
MeanShift
VBGMM
YES
KMeans
NOT WORKING
Spectral Clustering
GMM
clustering

Back

demo