

# Workshop AI & Machine Learning



September 2025

ir. Johan Decorte

(johan.decorte@gmail.com)

# Wie ben ik?

- **Johan Decorte**
- **Burgerlijk ingenieur computerwetenschappen (KUL '86)**
- **25 jaar IT-carrière bij Siemens, Barco, Attentia, Agfa**
- **Momenteel:**
  - Docent (big) data en AI bij HOGENT
  - Voorzitter stuurgroep AI-strategie HOGENT
  - Trainer/consultant (big) data en AI
  - Medewerker aan deskundigenonderzoeken
- **Gepassioneerd door (big) data en de strategische meerwaarde van IT voor organisaties**

# Auteur van het boek (november 2021):

Elke twee jaar verdubbelt de hoeveelheid beschikbare gegevens in de wereld. Met krachtige en betaalbare computers en toegankelijke algoritmes kunnen we die data omvormen tot bruikbare inzichten over mensen, machines en processen. Het is niet alleen een nieuw businessmodel voor heel wat bedrijven: ook onze gezondheid en het klimaat kunnen er wel bij varen.

Maar elke technologie kan misbruikt worden, en big data zijn daarin geen uitzondering. Johan Decorte legt uit hoe datawetenschappers orde scheppen in de immense massa aan gegevens, belicht enkele van de belangrijkste toepassingsgebieden en werpt een blik op de gevaren ervan.

VIZIER

Johan Decorte

## BIG DATA

Een revolutie ontrafeld



ACADEMIA  
PRESS

# Training: [www.gorat.be](http://www.gorat.be)



## Databanken

cursus SQL

cursus t-SQL (Microsoft)

cursus SQL Server Performance  
Management (Microsoft)

cursus PL/SQL (Oracle)

cursus mySQL

cursus SQL Server

cursus Database Design



## AI, BI, DWH en NoSQL

cursus Machine Learning en Data Mining

Nieuw

cursus Generative AI in Python

Nieuw

cursus Power BI

Nieuw

cursus Datawarehouses, Business  
Intelligence en Datamining

cursus SQL Server Reporting Services

cursus SQL Server Business Intelligence

Nieuw

cursus NoSQL



## C++ en UML

cursus Python voor beginners

Nieuw

cursus C++

cursus C++ advanced

Nieuw

cursus C

cursus Design Patterns

cursus UML



## Business analyse, BPM en Agile

cursus Business Analyse

cursus Modelleertechnieken voor de  
business analyst

cursus BPMN 2.0

cursus Business Process Management

cursus Agile management & development

**Wie zijn jullie?**

# Cursusmaterialaal

[https://github.com/jdecorte/Workshop\\_AI\\_MachineLearning](https://github.com/jdecorte/Workshop_AI_MachineLearning)

# Tools

- Cursor (AI-fork van Visual Studio Code met Python-extensie)
- Python 3.12.x
- Python-libraries te installeren in de loop van de lessen

# Artificiële intelligentie

## **25 jaar geleden won IBM-supercomputer Deep Blue historische schaakpartij tegen wereldkampioen Kasparov**

10 februari 1996, dat is exact 25 jaar geleden en de dag waarop IBM-supercomputer Deep Blue in Philadelphia een historisch partijtje schaak won tegen Garry Kasparov. Uiteindelijk won de toen 32-jarige wereldkampioen een week later wél de match tegen Deep Blue met 4-2. In mei 1997 pakte Deep Blue II dan revanche en klopte grootmeester Kasparov met een verschil van één spel, opnieuw in een wedstrijd over zes partijen.

Joeri Vlemings 10-02-21, 13:14 Laatste update: 20:18





# AI: definitie

- Het automatiseren van activiteiten die we associëren met menselijk denken:
  - Beslissingen nemen
  - Problemen oplossen
  - Leren

en dit ***zonder expliciet te programmeren***.

*Expliciet programmeren*: stap voor stap (in een programmeertaal) beschrijven wat de computer moet doen.

AI: computer ontdekt *gaandeweg* wat er moet gebeuren → leerproces

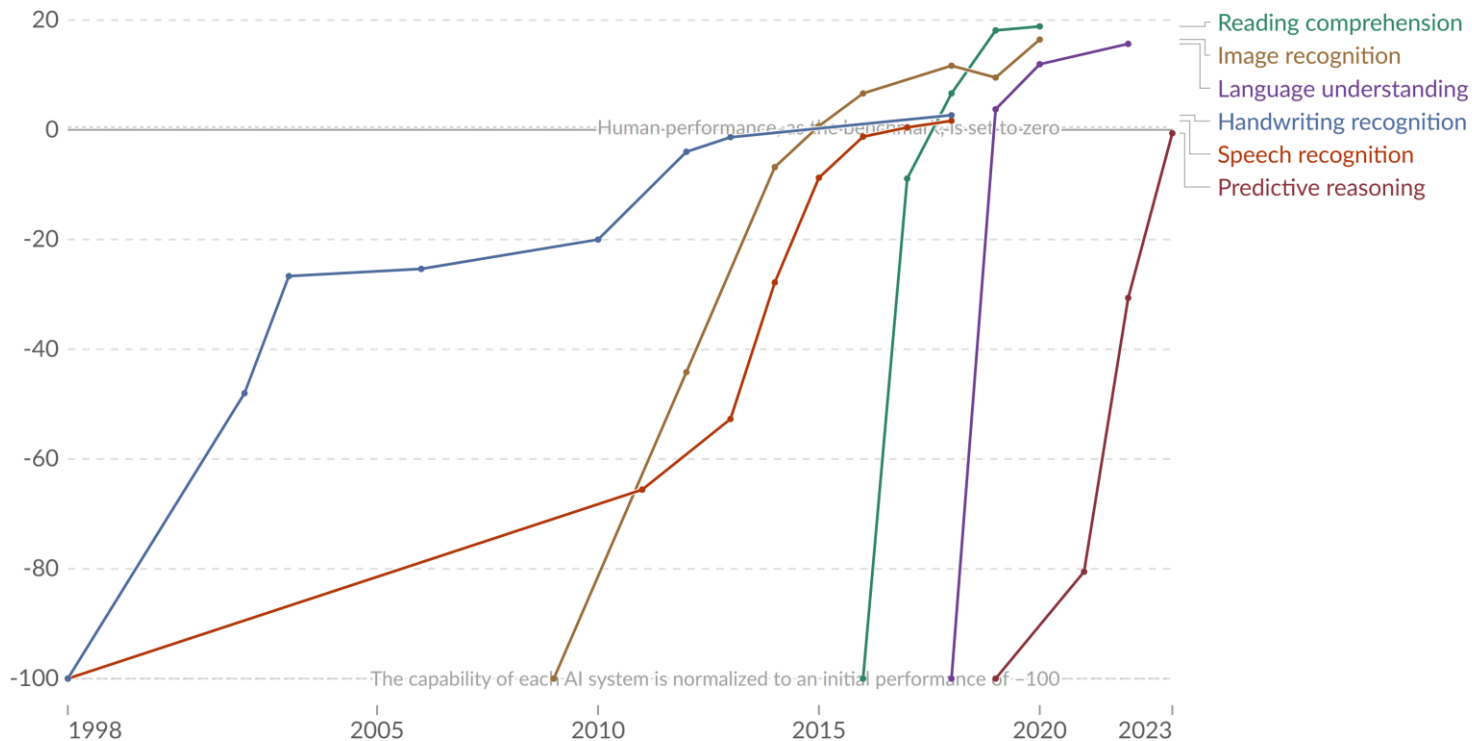
# AI = science fiction? Neen!

*Still in the early days of its expansion, but **AI's impact on so many different spheres of life is already obvious**. Artificial intelligence is the result of scientific and technological progress in the fields of computing and mathematics, **driven by big data and machine learning**, but also fed by insights in brain research, neuroscience and cognitive psychology. **Quantum computing** will soon dramatically move the frontiers of computer capabilities. And integration with **synthetic biology** is just around the corner.*

Dirk Van Damme, voormalig diensthoofd van het Centre for Educational Research and Innovation van de [OESO](#), 4 februari 2021

# Test scores of AI systems on various capabilities relative to human performance

Within each domain, the initial performance of the AI is set to -100. Human performance is used as a baseline, set to zero. When the AI's performance crosses the zero line, it scored more points than humans.



Data source: Kiela et al. (2023)

OurWorldinData.org/artificial-intelligence | CC BY

Note: For each capability, the first year always shows a baseline of -100, even if better performance was recorded later that year.

# Machine Learning

## 1 . The Machine Learning Landscape

*What Machine Learning is, what problems it tries to solve, and the main categories and fundamental concepts of its systems.*

# What is Machine Learning: definitions

*Machine Learning is the science (and art) of programming computers so they can learn from data.*

More general:

*Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.*

—Arthur Samuel, 1959

More engineering-oriented:

*A computer program is said to learn from experience  $E$  with respect to some task  $T$  and some performance measure  $P$ , if its performance on  $T$ , as measured by  $P$ , improves with experience  $E$ .*

—Tom Mitchell, 1997

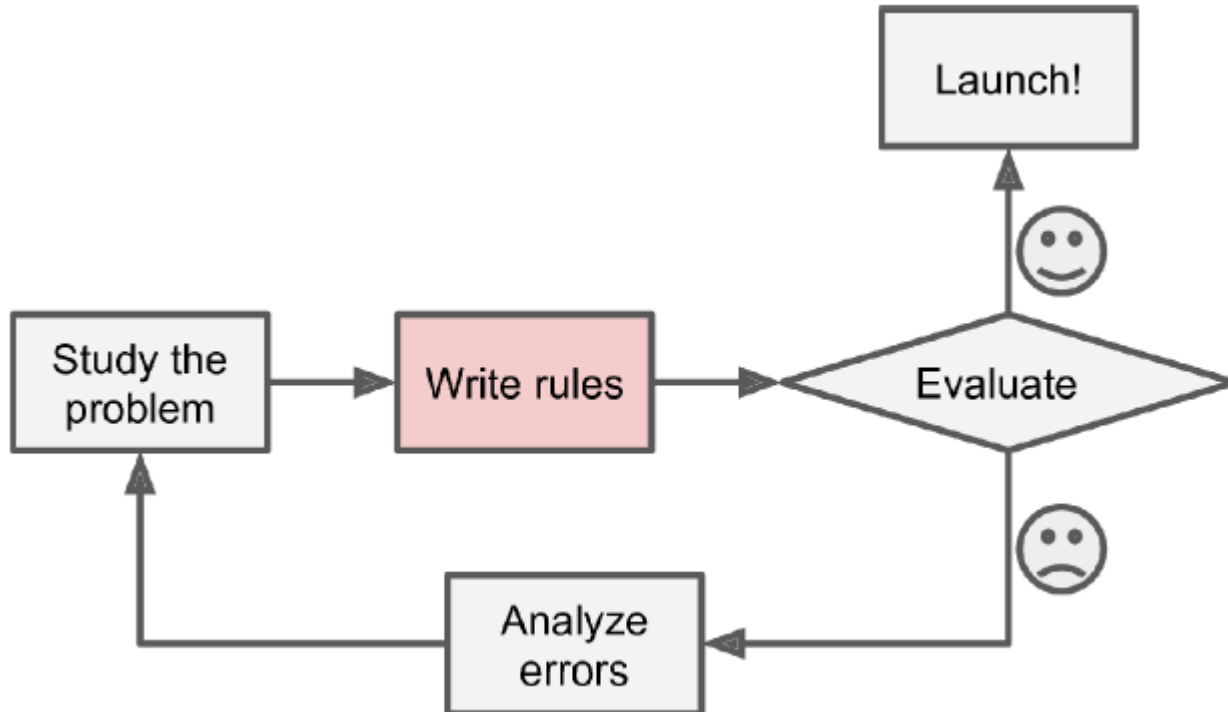
## Example: SPAM filter

- = Machine Learning program
- Given:
  - Examples of spam emails (flagged by users)
  - Examples of regular (nospam = “ham”) emails
- Learns to flag spam
- Training set = example the systems uses to learn

# Why use Machine Learning?

- Traditional approach for SPAM filter
  1. Find common words in spam emails: 4U, credit car, free, amazing, ...
  2. Flag all emails that contain (a combination of) these words as spam.
  3. Test and repeat 1+2 until it's good enough

# Why use Machine Learning?



*Figure 1-1. The traditional approach*



# Why use Machine Learning?

- Machine learning approach for SPAM filter
  - automatically learns which words and phrases are good predictors of spam by detecting unusually frequent patterns of words in the spam examples compared to the ham examples
  - The program is much shorter, easier to maintain, and most likely more accurate.

# Why use machine learning

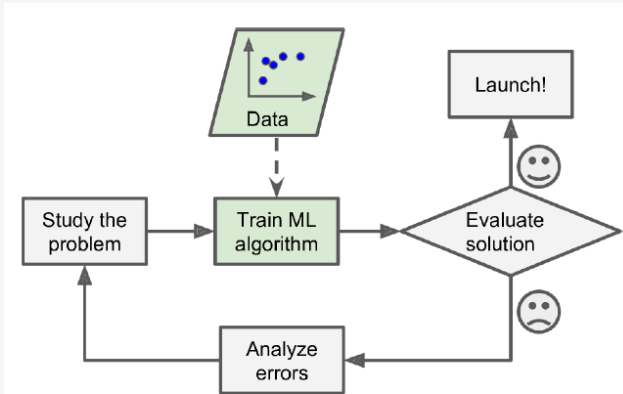


Figure 1-2. The Machine Learning approach

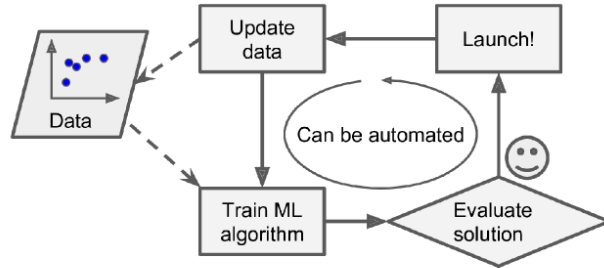
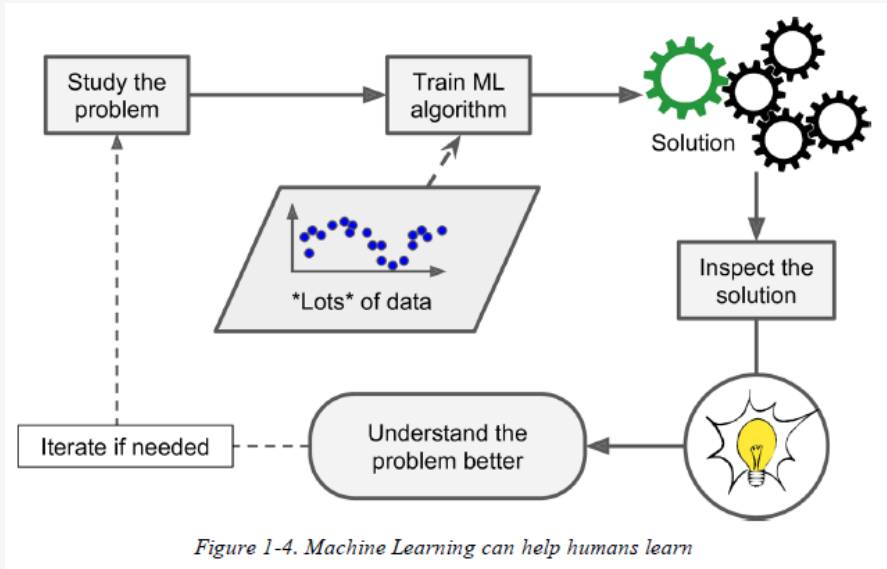


Figure 1-3. Automatically adapting to change

# Data mining



Once a spam filter has been trained on enough spam, it can easily be inspected to reveal the list of words and combinations of words that it believes are the best predictors of spam.

**Data mining = applying ML techniques to dig into large amounts of data to help discover patterns that were not immediately apparent.**

# Het data mining proces

- Analyseren data = **trainen** van een model
- Bij gesuperviseerd en niet-gesuperviseerd leren:
  - Op basis van een grote hoeveelheid (volume) goed uitgebalanceerde gegevens, wordt een statistisch model opgesteld dat nadien kan gebruikt worden om bijvoorbeeld:
    - Voorspellingen te maken van de prijs voor nieuwe diamanten (niet aanwezig in model)
    - Onbekende planten te determineren.
- **Laat de data spreken**
- Het model is maar zo goed als de data waarop het gebaseerd is (garbage in – garbage out)
- Data governance is noodzakelijk → draag zorg voor uw data!

# Examples of Applications

- Analyzing images of products on a production line to automatically classify them
- Detecting tumors in brain scans
- Automatically classifying news articles as sports, financial news, ...
- Automatically flagging offensive comments on discussion forums
- Summarizing long documents automatically
- Creating a chatbot or a personal assistant
- Forecasting your company's revenue next year, based on many performance metrics
- Making your app react to voice commands
- Detecting credit card fraud
- Segmenting clients based on their purchases so that you can design a different marketing strategy for each segment
- Representing a complex, high-dimensional dataset in a clear and insightful diagram
- Recommending a product that a client may be interested in, based on past purchases
- Building an intelligent bot for a game

# Types of Machine Learning systems

- Supervised/Unsupervised/Reinforcement Learning
- Batch and Online Learning
- Instance-Based vs. Model-Based Learning

# **Supervised/Unsupervised/Reinforcement Learning**

Classified according to the amount and type of supervision they get during training.

- Supervised learning
- Unsupervised learning
- Semisupervised learning
- Reinforcement Learning

# Supervised Learning: classification

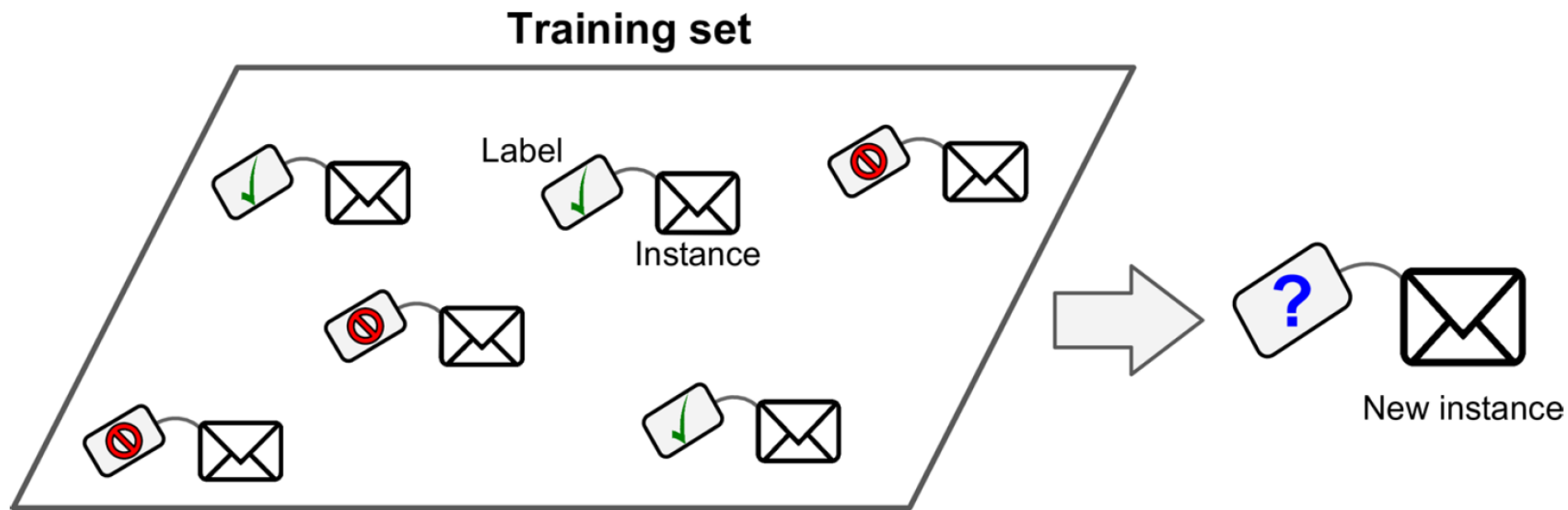


Figure 1-5. A labeled training set for spam classification (an example of supervised learning)



# Supervised Learning: regression

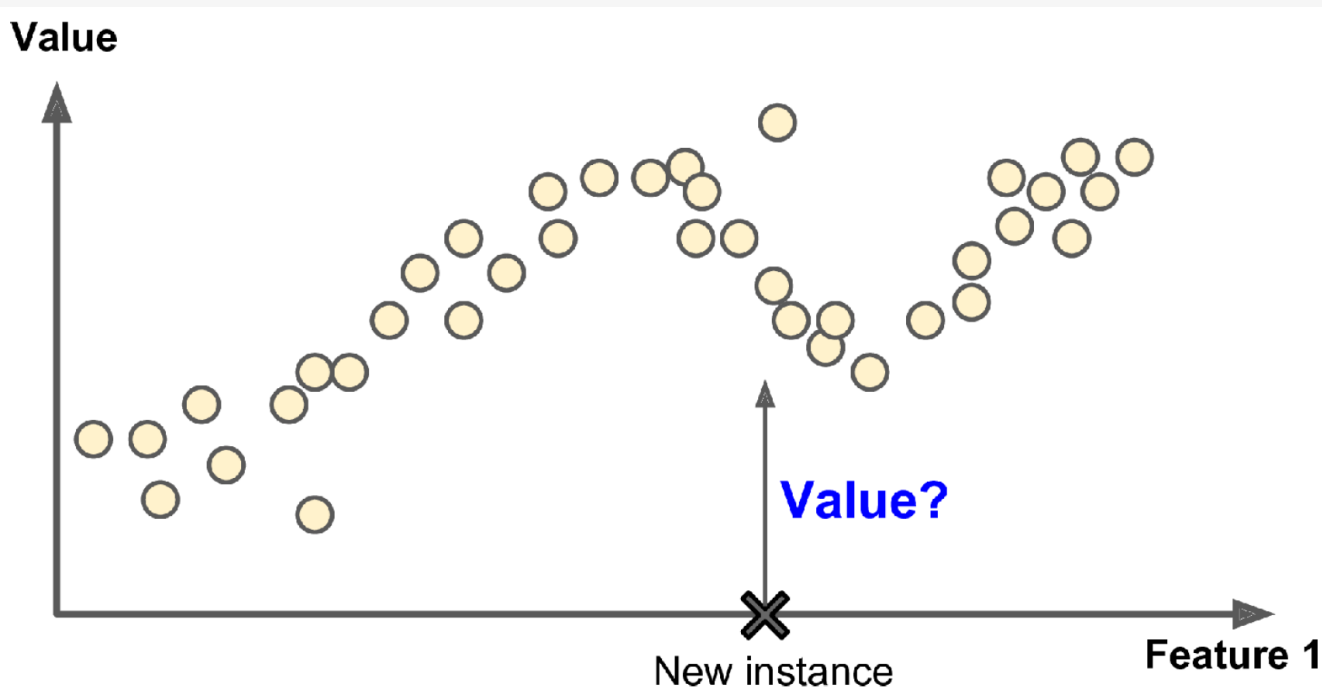


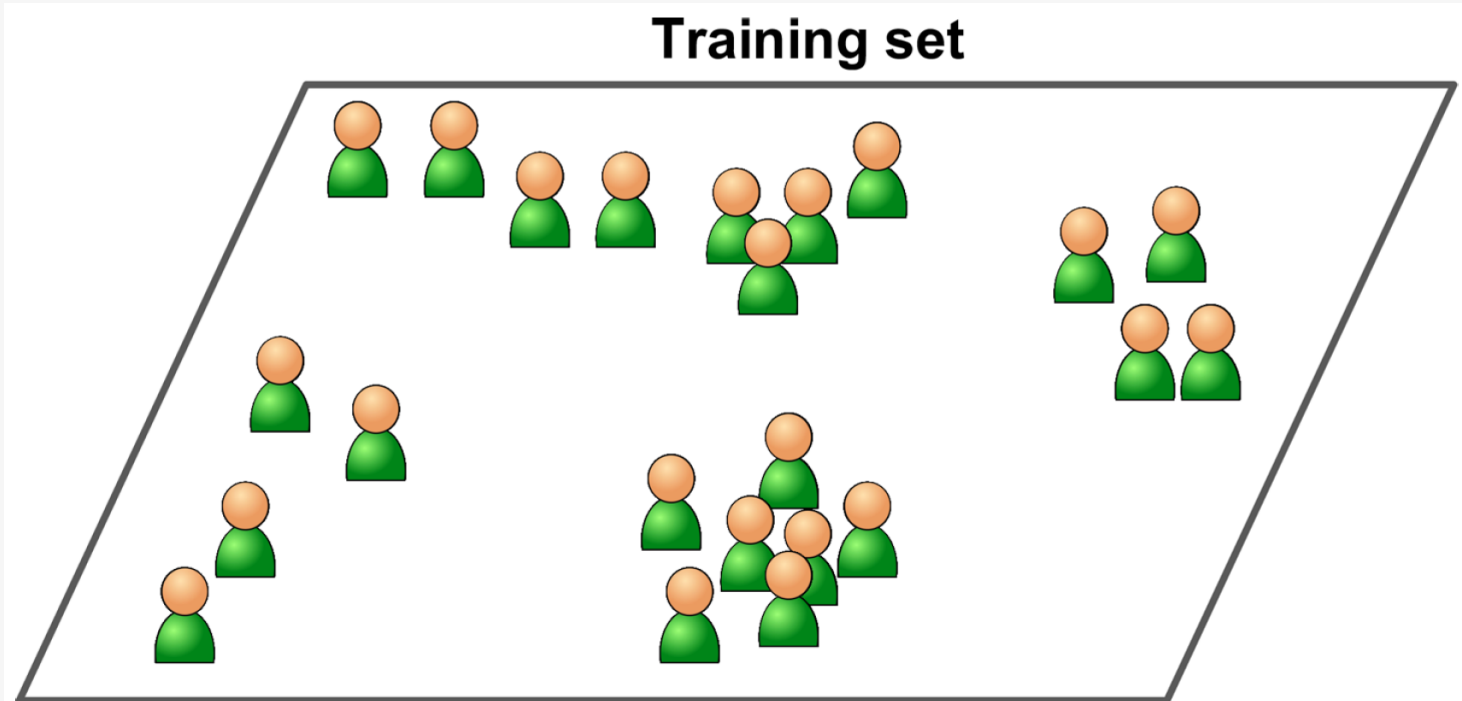
Figure 1-6. A regression problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)

# Supervised learning: algorithms

Some of the most important supervised learning algorithms

- k-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

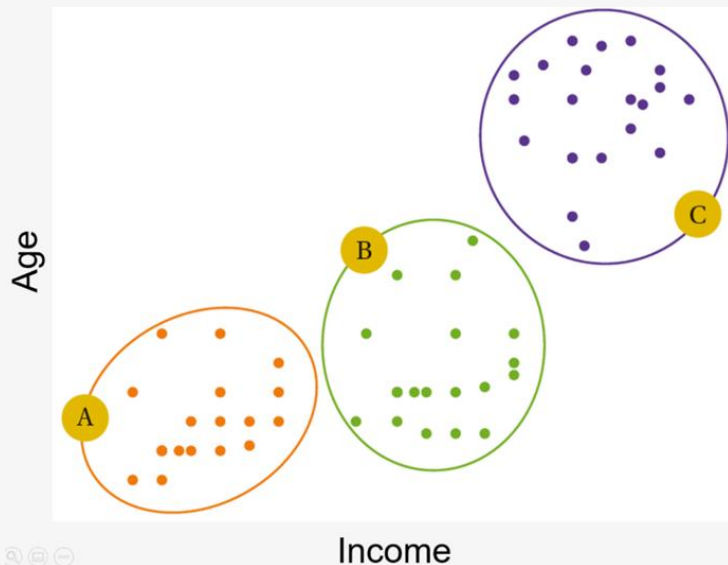
# Unsupervised learning



*Figure 1-7. An unlabeled training set for unsupervised learning*

# Unsupervised learning: algorithms

- Clustering
  - K-Means
  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)
- Anomaly detection and novelty detection
  - One-class SVM
  - Isolation Forest
- Visualization and dimensionality reduction
  - Principal Component Analysis (PCA)
  - Kernel PCA
  - Locally Linear Embedding (LLE)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Association rule learning (ex. Market Basket Analysis)
  - Apriori
  - Eclat



*Example of clustering*

# Unsupervised learning: anomaly detection



Figure 1-10. Anomaly detection

# Semisupervised learning

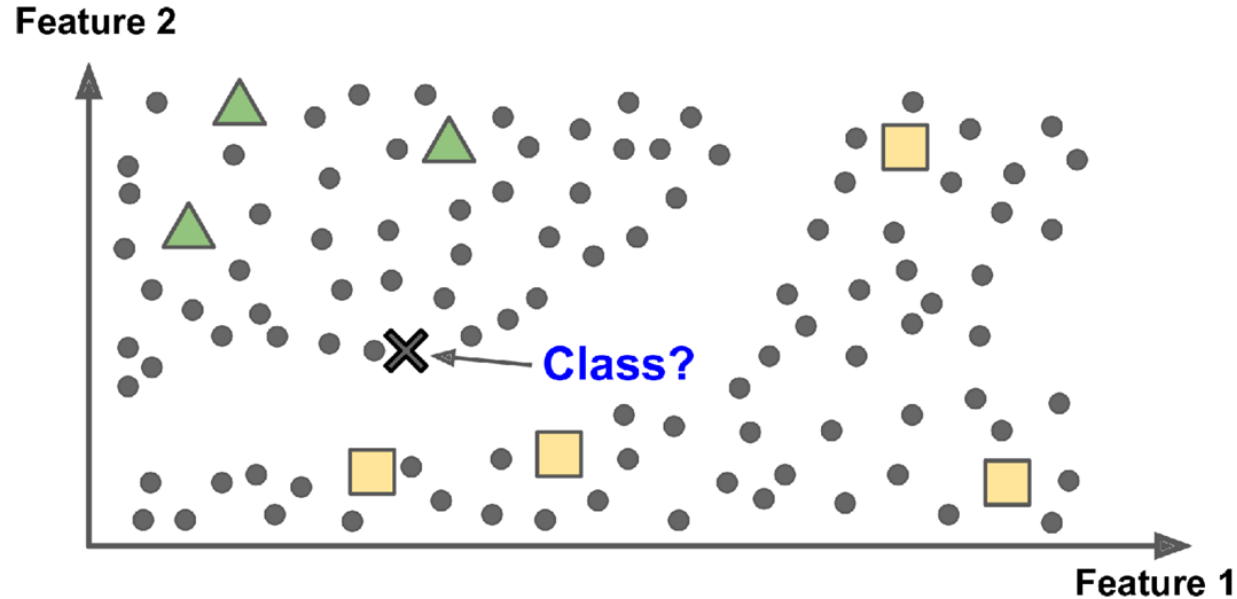


Figure 1-11. Semisupervised learning with two classes (triangles and squares): the unlabeled examples (circles) help classify a new instance (the cross) into the triangle class rather than the square class, even though it is closer to the labeled squares

# Reinforcement Learning

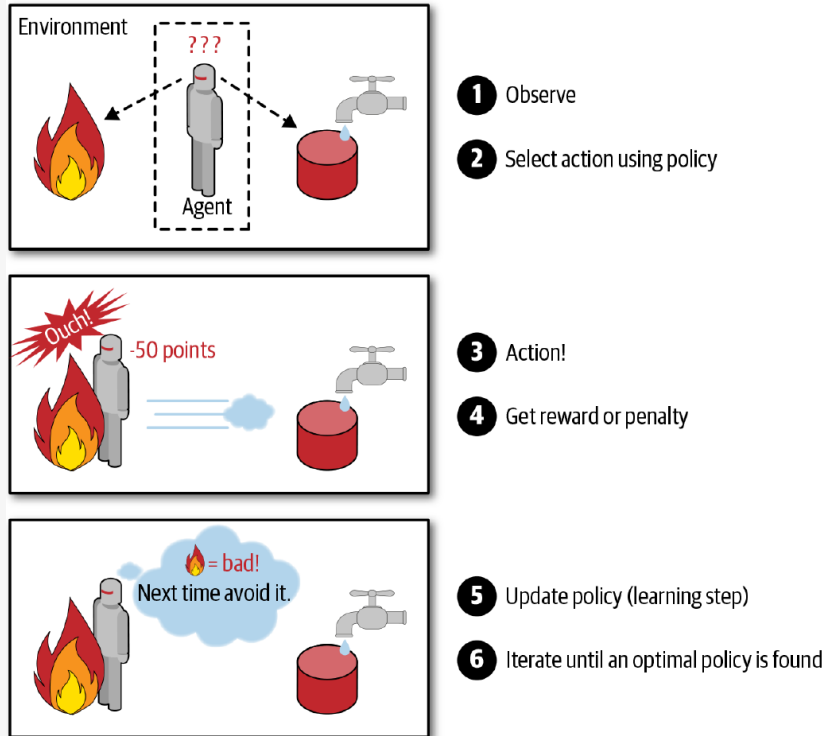
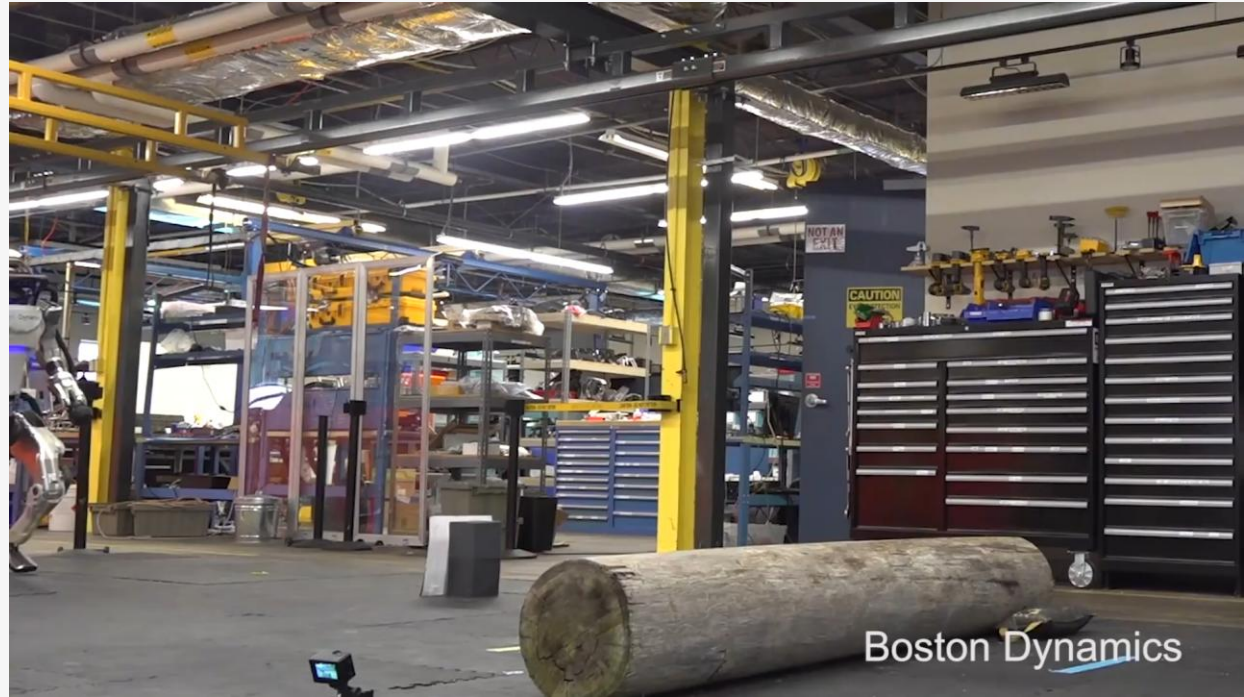


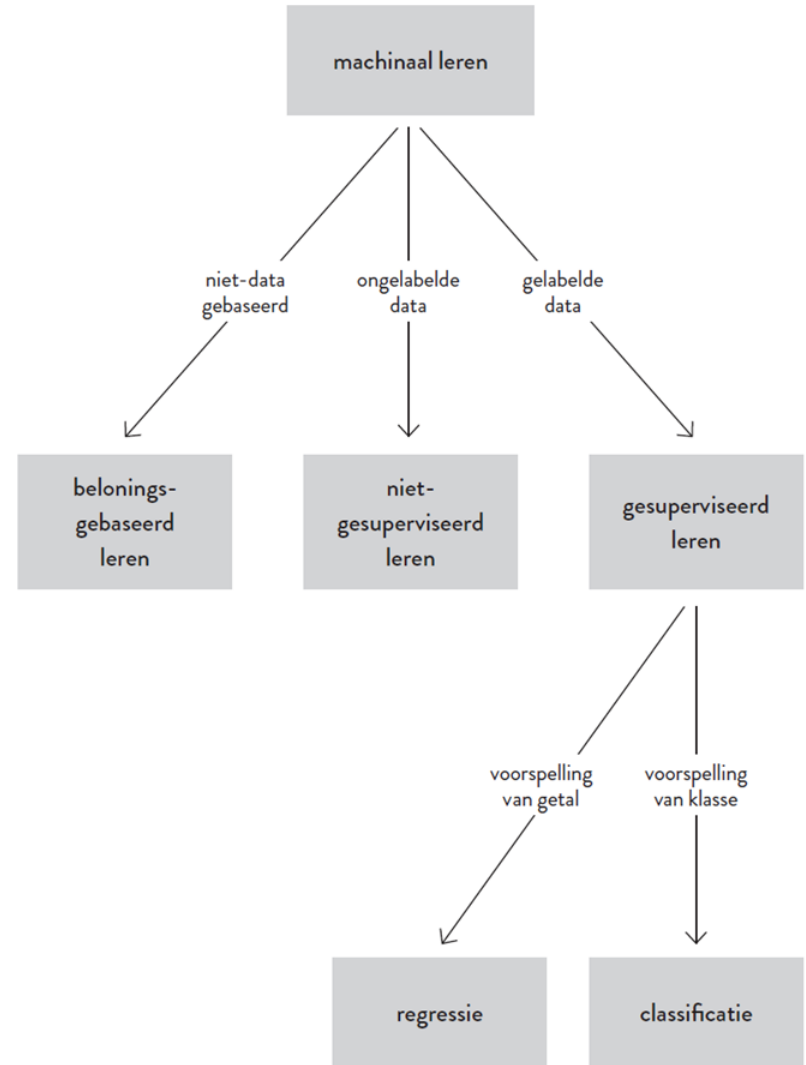
Figure 1-12. Reinforcement Learning

# Beloningsgebaseerd leren: voorbeelden





# Supervised/ Unsupervised/ Reinforcement Learning : samengevat



## Batch and Online Learning

- Classify Machine Learning systems based on whether or not the system can learn incrementally from a stream of incoming data

# Batch learning

- System is incapable of learning incrementally
- It must be trained using all the available data
- Takes a lot of time and computing resources  
→ done offline (“offline learning”)
- New data = retraining
- This process can be automated fairly easily

# Online learning

- System is trained incrementally by feeding it data instances sequentially, either individually or in small groups called *minibatches*.
- Each learning step is fast and cheap, so the system can learn about new data on the fly, as it arrives

# Online learning

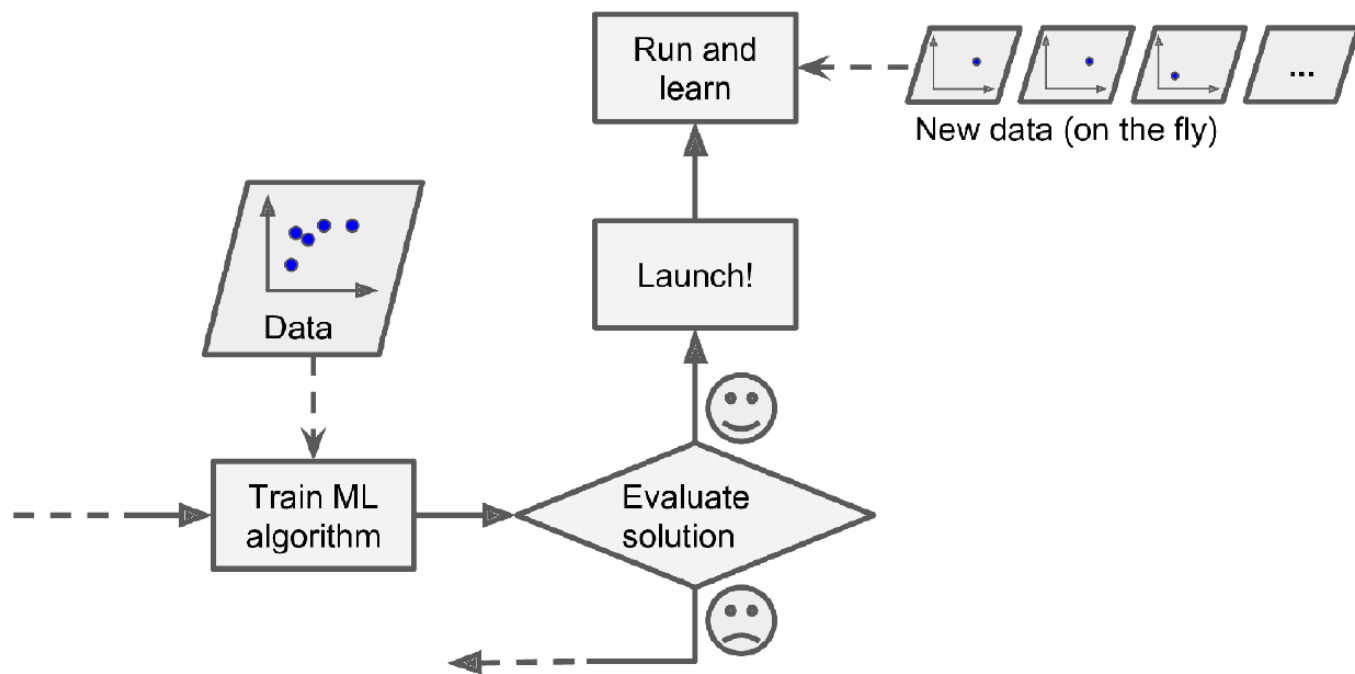


Figure 1-13. In online learning, a model is trained and launched into production, and then it keeps learning as new data comes in

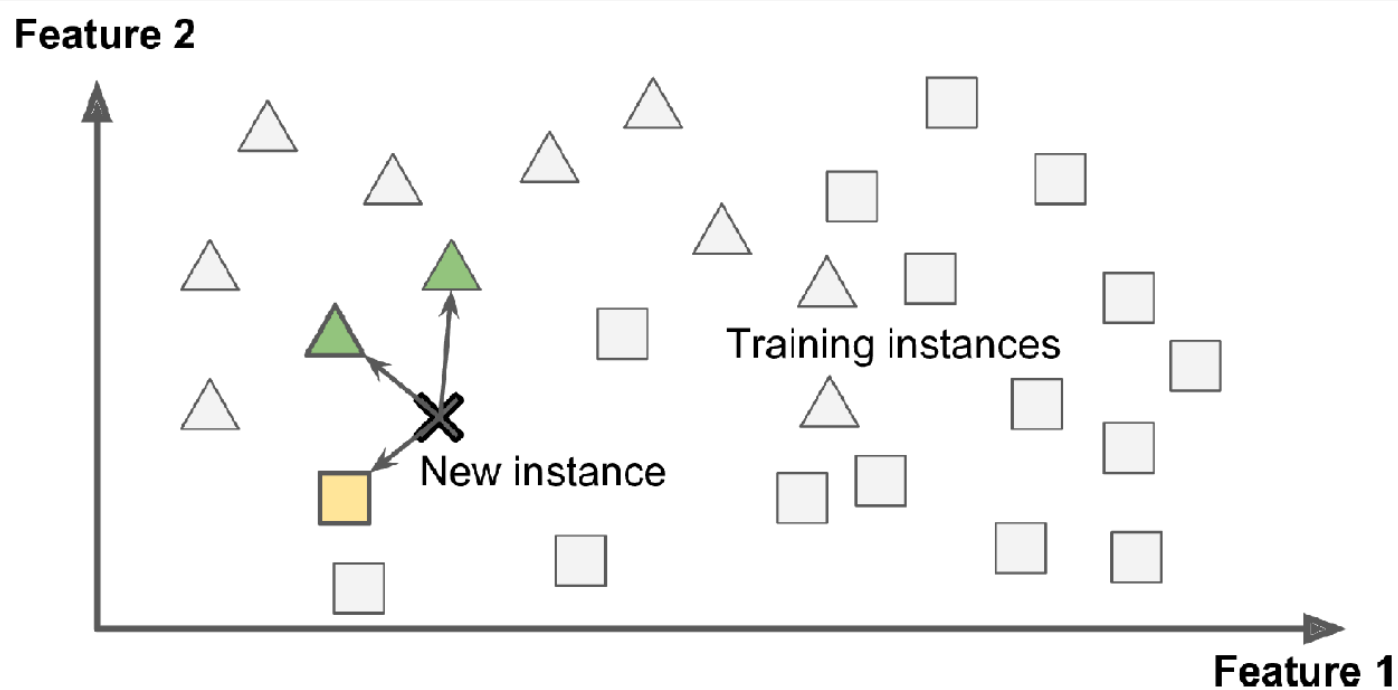
# Instance-Based vs. Model-Based Learning

- Categorize Machine Learning systems by how they *generalize*.
- Most Machine Learning tasks are about making predictions.
  - Given a number of training examples, the system needs to be able to make good predictions for examples it has never seen before. Having a good performance measure on the training data is good, but insufficient; the true goal is to perform well on new instances.
  - 2 main approaches to generalization: instance-based learning and model-based learning.

# Instance based learning

- Classify according to measure of similarity
- Example: flag an email as spam if it has many words in common with a known spam email.

# Instance based learning

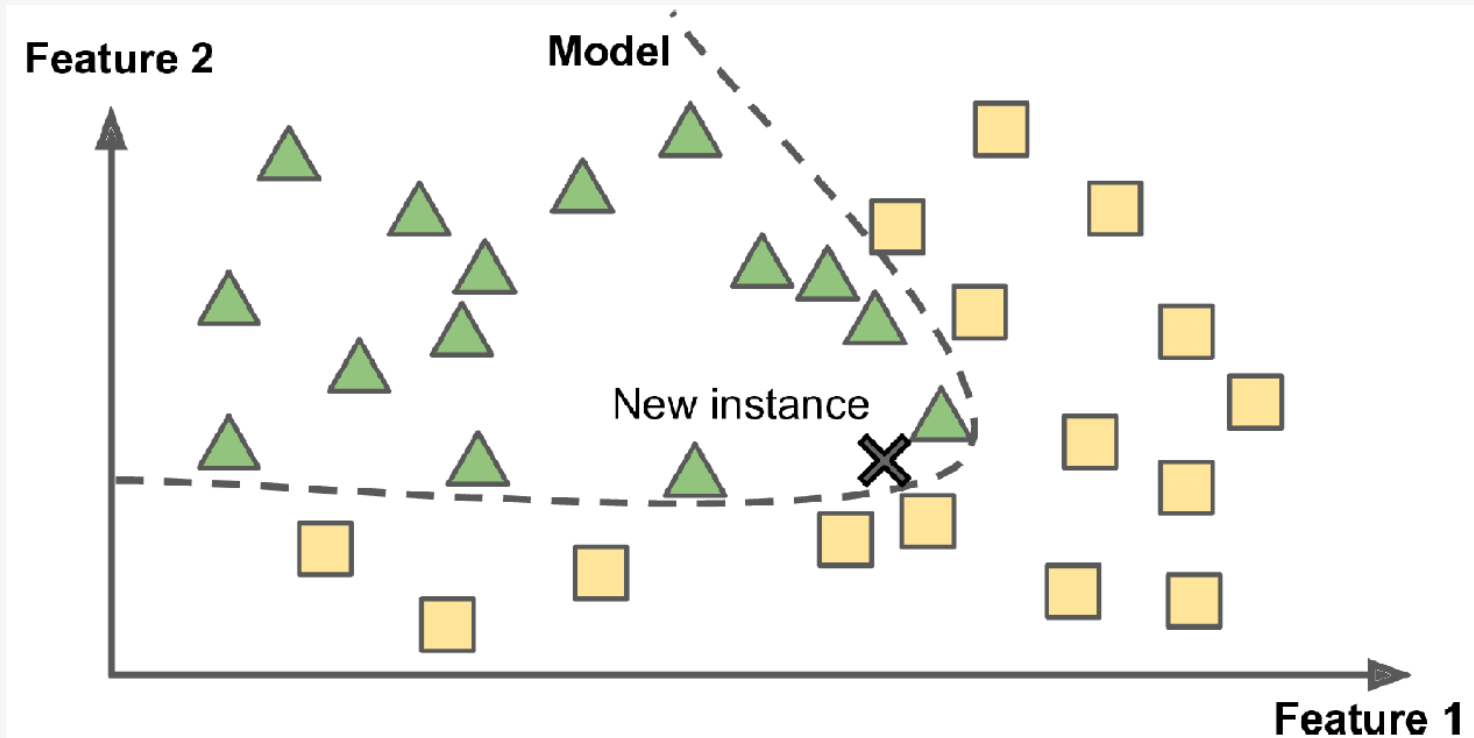




# Model-based learning

- Build a model of these examples and then use that model to make predictions.

# Model-based learning

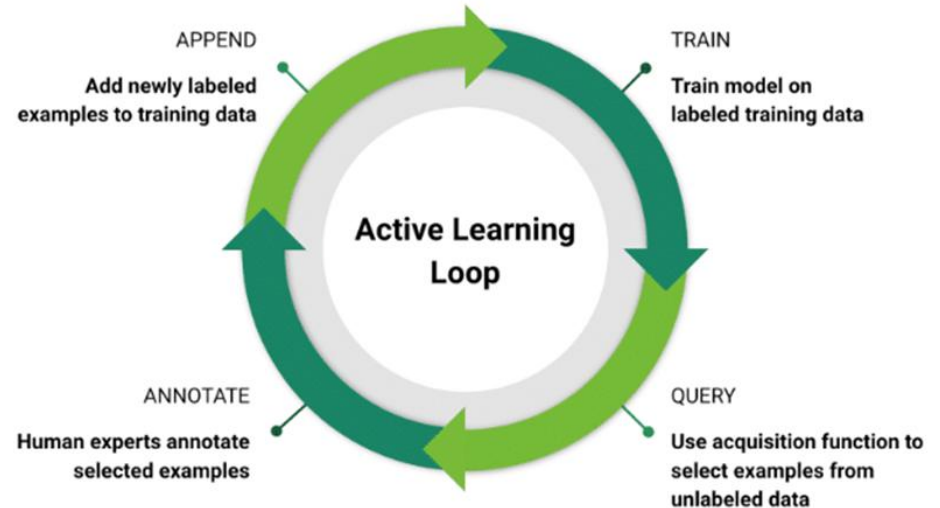


# Model Based Learning: example

- See [https://github.com/jdecorte/Workshop\\_AI\\_Machine\\_Learning/blob/main/0111-the\\_machine\\_learning\\_landscape.ipynb](https://github.com/jdecorte/Workshop_AI_Machine_Learning/blob/main/0111-the_machine_learning_landscape.ipynb)

# Active Learning

- Supervised Learning is duur omdat alle instanties moeten gelabeld zijn.
- Labeling wordt vaak uitbesteed: maatwerkbedrijven, Indië, ...
- Unsupervised Learning biedt vaak niet het gewenste resultaat .
- Oplossing: active learning
- Door slechts een beperkt aantal instanties te labelen wordt toch een vergelijkbaar resultaat bereikt.



# Data sources

- Columnar data
- Images - Video
- Audio
- Text

# Columnar Data: mammografie (1/2)

- Gegevens afgeleid uit 961 mammografieën.
- Kenmerken:
  - Age: patient's age in years (integer)
  - Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)
  - Margin: mass margin:
    - circumscribed=1
    - microlobulated=2
    - obscured=3
    - ill-defined=4
    - spiculated=5 (nominal)
  - Density: mass density
    - high=1 iso=2 low=3 fat-containing=4 (ordinal)
- Label: **Severity**:
  - benign=0 or malignant=1
  - benign: 516; malignant: 445
- BI-RADS assessment:
  - 1 to 5 (ordinal) = assessment by radiologist
- Bron: <https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass>

BIRADS	Age	Shape	Margin	Density	Severity
5	67	3	5	3	1
4	43	1	1	?	1
5	58	4	5	3	1
4	28	1	1	3	0
5	74	1	5	?	1
4	65	1	?	3	0
4	70	?	?	3	0
5	42	1	?	3	0
5	57	1	5	3	1
5	60	?	5	1	1
5	76	1	4	3	1
3	42	2	1	3	1
4	64	1	?	3	0
4	36	3	1	2	0
4	60	2	1	2	0
4	54	1	1	3	0
3	52	3	4	3	0
4	59	2	1	3	1
4	54	1	1	3	1
4	40	1	?	?	0
?	66	?	?	1	1
5	56	4	3	1	1
4	43	1	?	?	0
5	42	4	4	3	1
4	59	2	4	3	1
5	75	4	5	3	1
2	66	1	1	?	0
5	63	3	?	3	0
5	45	4	5	3	1
5	55	4	4	3	0
4	46	1	5	2	0
5	54	4	4	3	1
5	57	4	4	3	1
4	39	1	1	2	0
4	81	1	1	3	0
4	77	3	?	?	0
4	60	2	1	3	0
5	67	3	4	2	1
4	48	4	5	?	1
4	55	3	4	2	0
4	59	2	1	?	0
4	78	1	1	1	0
4	50	1	1	3	0
4	61	2	1	?	0

# mammografie (2/2)

- Gesuperviseerd leren: er is een label
- Trainingsfase = opstellen van het model:
  - Software zoekt naar patronen in de combinatie van kenmerken die leiden tot diagnose B/M
  - BIRADS-assessment wordt hierbij niet gebruikt!
  - **Correlatie, geen causaliteit:**
  - “Laat de data spreken”
- Gebruiksphase = gebruiken van het model:
  - Bij nieuwe beelden wordt aan het model gevraagd om een voorspelling te maken voor de diagnose
  - Meestal samen met een probabiliteit, vb. 70% kans goedaardig, 30% kans kwaadaardig.
- Voordelen:
  - Aanvulling op het oordeel van de radioloog
  - Evaluatie BIRADS-assessment
  - Bij grote zekerheid op goedaardigheid kan een ingrijpende biopsie vermeden worden.

# Beeldverwerking

- Elke instantie is een beeld
- Elke pixel is een feature
- Speciale technieken (convolutionele neurale netwerken) laten toe meerdere, aanliggende pixels tegelijk te bekijken om “grotere” features te maken
- bijv. gezichtsherkenning: neus, ogen, mond, ...



# Beeldverwerking: toepassing OCR

- OCR = optical character recognition
- Handschriftherkenning op gestandardiseerde formulieren

Cabinet medical din ambulatoriu de specialitate/spital.....

Medic .....

Specialitatea .....

**SCRISOARE MEDICALĂ**

Domnului/doamnei Dr. (adresa cabinetului medical) .....

Stimate(ă) coleg(ă), vă informăm că pacientul dumneavoastră .....

*Zaharia Delina* născut la data *13.09.1981*

CNP *Zaharia Delina* fost pacient în serviciul nostru la data de *12.06.2008*


Diagnosticul: *Străfu cerebeloasă*  
*Tetragigă atonică*

Tratament recomandat: *Hecevită tratament cronic + 20*  
*mg/ziune de câte alte persoane*

Data: *12/VI/2008*

Semnatura și ștampila medicului:

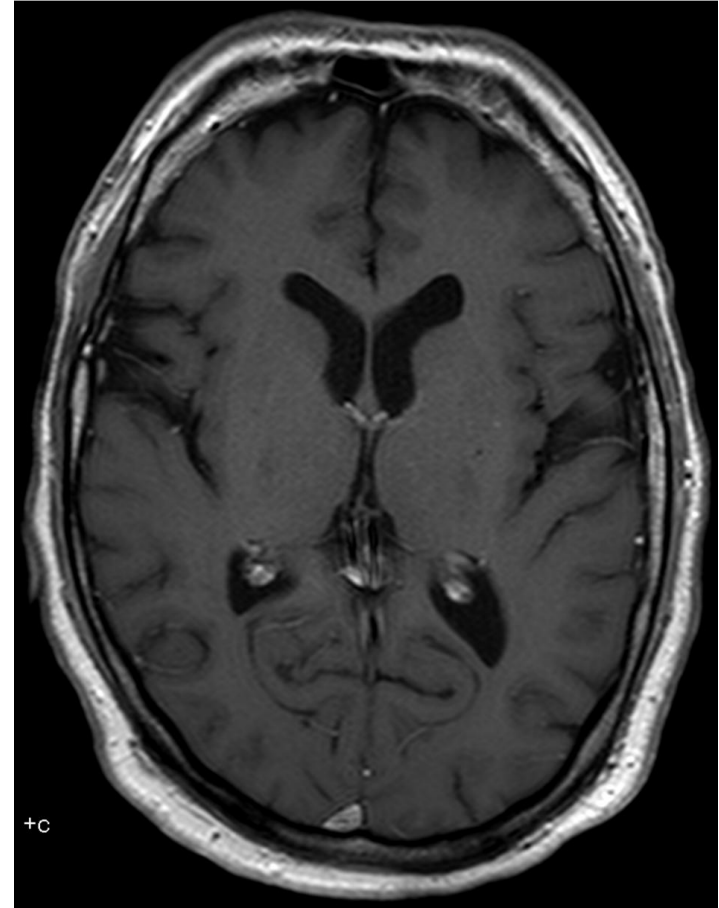
*DR. NICOLAI STELA*

 **Bayer HealthCare**  
Pharma

# Beeldverwerking: toepassing beeldclassificatie

Vb. hersenscan

- Is tumor goed- of kwaadaardig?
- I.p.v. via een (deels manueel) proces (cf. borstkanker) beelden te kwantificeren, kan men ook rechtstreeks met de foto's werken.



# Beeldverwerking: huidige status

- Resultaten benaderen het menselijke niveau
- Zeker bij sterk gestandaardiseerde toepassingen zoals ANPR, standaardformulieren, gestandaardiseerde beelden, ...
- Veel computerpower vereist
- Speciale processoren: GPU's, TPU's
- Voor standaardobjecten: gebruik pre-trained modellen of LMM's.

# Audio

- Ook geluidsfragmenten kun je *dataficeren*.
- Spotify deelt klanten op in groepen met gelijkaardige muzikale interesses (= unsupervised learning/clustering) op basis van genre, instrumenten, tempo, vrolijk-triest-weemoedig
- Werkt op audiosignaal zelf, niet op metadata
- Idem bij non-verbale communicatie bij telefoongesprek: wat zijn de emoties van de klant?

# NLP = natural language processing

- Kennisdomein rond begrijpen, manipuleren en genereren van menselijke taal.
- Een van de meest uitdagende domeinen in AI, meer nog dan beeldverwerking:
- Natuurlijke taal is allesbehalve wiskundig
- Regels hebben zeer veel uitzonderingen
- Ook teksten kunnen omgezet worden in tabelvorm!
- Text mining = data mining op basis van teksten

# NLP: enkele toepassingen

## Sentimentanalyse (*opinion mining*):

- Heeft een tekst (e-mail, beoordeling van een film, nota) een positieve of negatieve connotatie?
- Gesuperviseerd leren (classificatie): model trainen op basis van teksten met gekend sentiment
- Ofwel gebruik maken van voorgetrainde modellen, ofwel zelf trainen in specifieke context

## Spam-detectie bij e-mail, sms

- Gesuperviseerd leren (classificatie)

# NLP: enkele toepassingen (vervolg)

## Teksten samenvatten

- Niet-gesuperviseerd leren

## Text-scaling

- Bv.: In welke mate is een verkiezingsprogramma van een politieke partij pro of contra Europa?
- Gesuperviseerd leren (regressie): veel verkiezingsprogramma's met (door een mens toegekende) score nodig om model te trainen
- Bv. is eindverslag in bouwgeschil eerder in voordeel van aannemer of eerder in voordeel van bouwheer?

## Automatisch vertalen

## Text generation

- Teksten genereren op basis van gelijkaardige teksten
- = Generatieve AI
- Vb. ChatGPT (Generative Pretrained Transformer)

# NLP: enkele toepassingen (vervolg)

## Document clustering

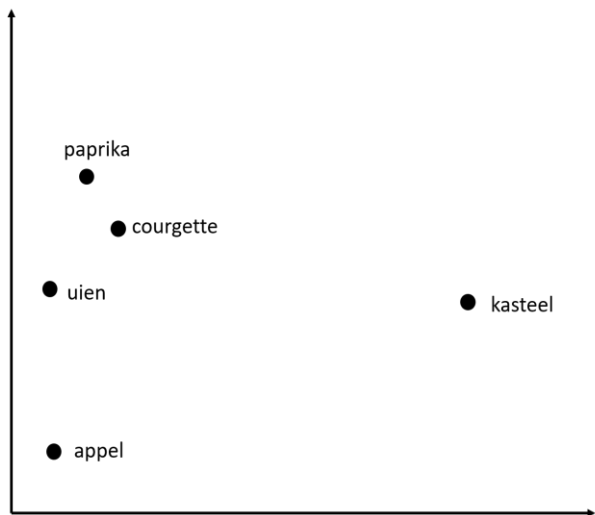
- Zoeken naar teksten over zelfde onderwerp of die gelijkaardig standpunt verdedigen.
- Ongesuperviseerd leren.
- We zoeken punten ("instanties") die op elkaar lijken op basis van (bv.) Euclidische afstand.
- Bepalen optimale aantal clusters en interpretatie van de clusters niet altijd evident.
- Hoe een document omzetten naar punten in een assenstelsel? Zie verder.



# Van teksten naar data: embeddings (1)

- Word en sentence embeddings: geavanceerde techniek ontwikkeld door o.a. Google (Word2Vec, transformers), Facebook (FastText) en Stanford University (Glove).
- Elk woord uit een voorgedefinieerde woordenschat (vb. 400.000 woorden van wikipedia) wordt afgebeeld op een punt in een (vb. 100-dimensionele) vectorruimte.
- Woorden die dicht bij elkaar staan komen meestal in elkaars buurt voor in teksten.
- Zo wordt een woord gezien in zijn context.

# Van teksten naar data: embeddings (2)



Welke woorden liggen dichtst bij 'paprika'?

- |              |                |
|--------------|----------------|
| 1. courgette | 6. chilipepers |
| 2. uien      | 7. aubergine   |
| 3. tomaten   | 8. tomaat      |
| 4. komkommer | 9. groenten    |
| 5. knoflook  | 10. gekookte   |

- Dit heeft het systeem volledig zelf geleerd.
- Toepassingen:
  - Als Irak dichterbij geweld dan Nederland → meer geweld in Irak

# Van teksten naar data: embeddings (3)

- Voor het bekomen van de feature matrix voor een zin of document wordt voor elke dimensie het gemiddelde genomen over alle woorden in de zin of het document.
- Aantal kolommen in feature matrix = aantal dimensies in vectorruimte  
→ meestal 100, 200, 400, ... .
- Domeinspecifieke (bv. Nederlandstalige juridische) teksten  $\neq$  wikipedia.
- Hertraining van word embeddings nodig om bv. Nederlandstalige, juridische termen af te beelden in de vectorruimte is noodzakelijk.
- Voorwaarde: voldoende vergelijkbare teksten (uit bv. deskundigenonderzoeken) van voldoende auteurs (schrijfstijlen!) digitaal beschikbaar.

# Main Challenges of Machine Learning

- Insufficient Quantity of Training Data
- Nonrepresentative Training Data
- Irrelevant Features
- Overfitting the Training Data
- Underfitting the Training Data

# Insufficient Quantity of Training Data

- It takes a lot of data for most Machine Learning algorithms to work properly.
- Even for very simple problems you typically need thousands of examples
- For complex problems such as image or speech recognition you may need millions of examples (unless you can reuse parts of an existing model).

# Non-representative Training Data

- Example: the set of countries we use for training the linear model is not perfectly representative; a few countries are missing.

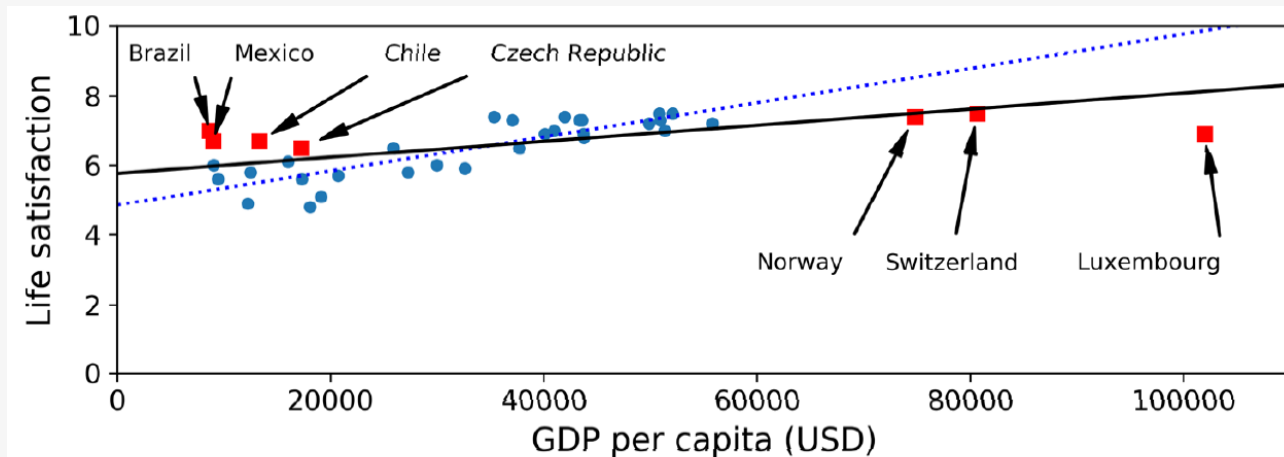


Figure 1-21. A more representative training sample

# Non-representative Training Data

- Too few data: *sampling noise*
- Very large samples can also be nonrepresentative if the sampling method is flawed → *sampling bias*

# Samling bias: voorbeelden

- Opsporen belastingfraude:
  - Wie maakt meeste kans op controle?
  - Gebaseerd op eerdere fraudegevallen  
→ alleen ontdekte fraude in dataset.
- Daderprofilering  
(o.a. gebruikt bij beslissing over voorlopige invrijheidsstelling)
  - Database van bekende daders en hun misdaden  
→ daderprofiel
  - Wat als in database bv. mensen van bepaalde etniciteit oververtegenwoordigd zijn?



# Poor-quality data

- Errors (ex. wrong classifications in training set), outliers, noise.
- First clean up the training data!
  - Some instances are clearly outliers
  - Some instances are missing a few features (e.g., 5% of your customers did not specify their age):  
decide whether you want to
    - ignore this attribute
    - ignore these instances
    - fill the missing values (e.g. median age)
    - train a model with the feature and one without

# Irrelevant features

- Garbage in/garbage out
- Feature engineering
  - Feature selection: selecting the most useful features to train on among existing features
  - Feature extraction: combining existing features to produce a more useful one
  - Creating new features by gathering new data

# Overfitting the Training Data

- Model performs well on the training data, but it does not generalize well.

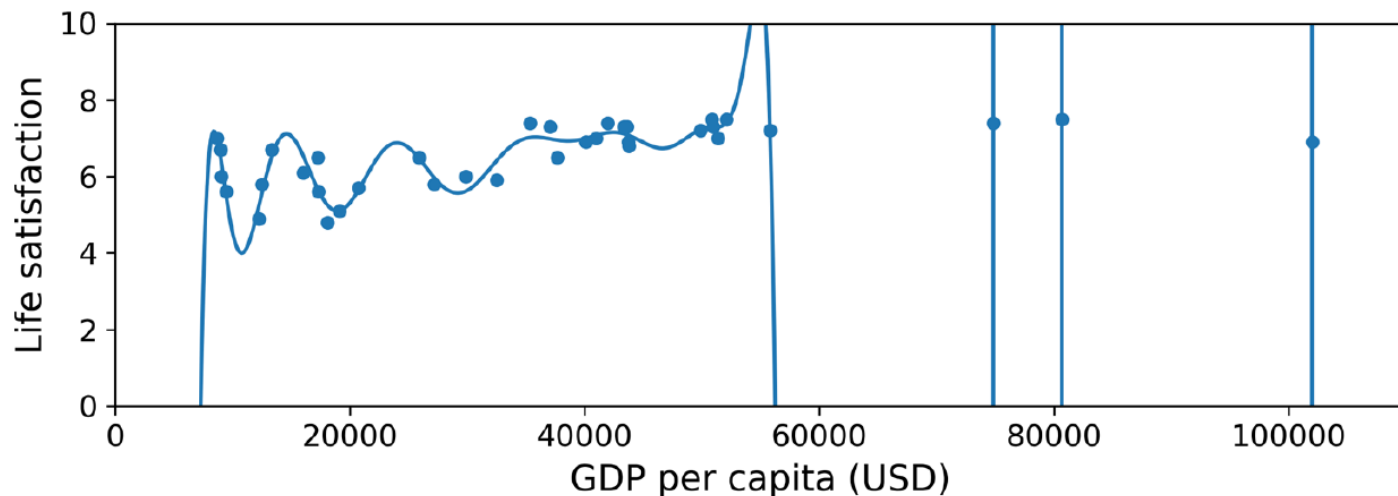
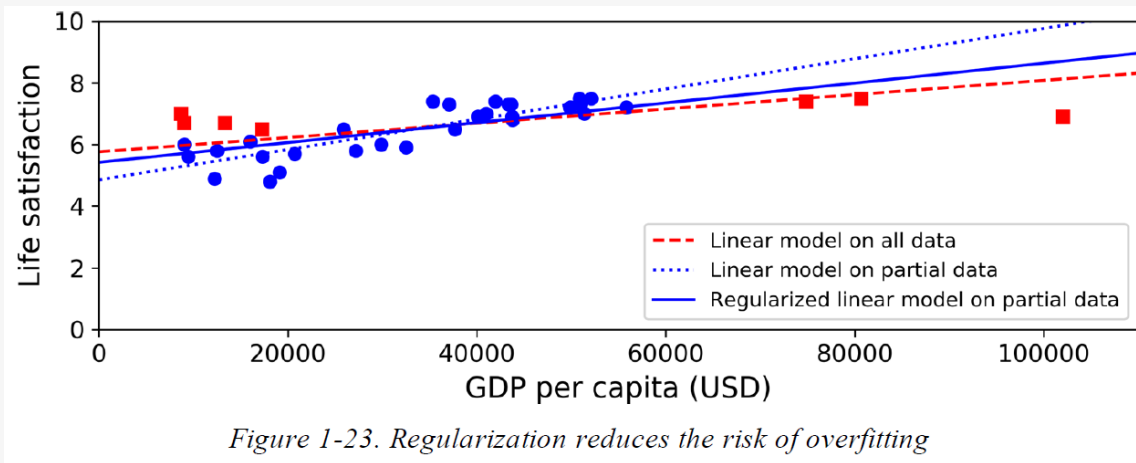


Figure 1-22. Overfitting the training data

# Overfitting the Training Data

- Overfitting happens when the model is too complex relative to the amount and noisiness of the training data.
- Possible solutions:
  - Simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data, or by constraining the model.
  - Gather more training data.
  - Reduce the noise in the training data (e.g., fix data errors and remove outliers).

# Avoid overfitting the Training Data by regularization

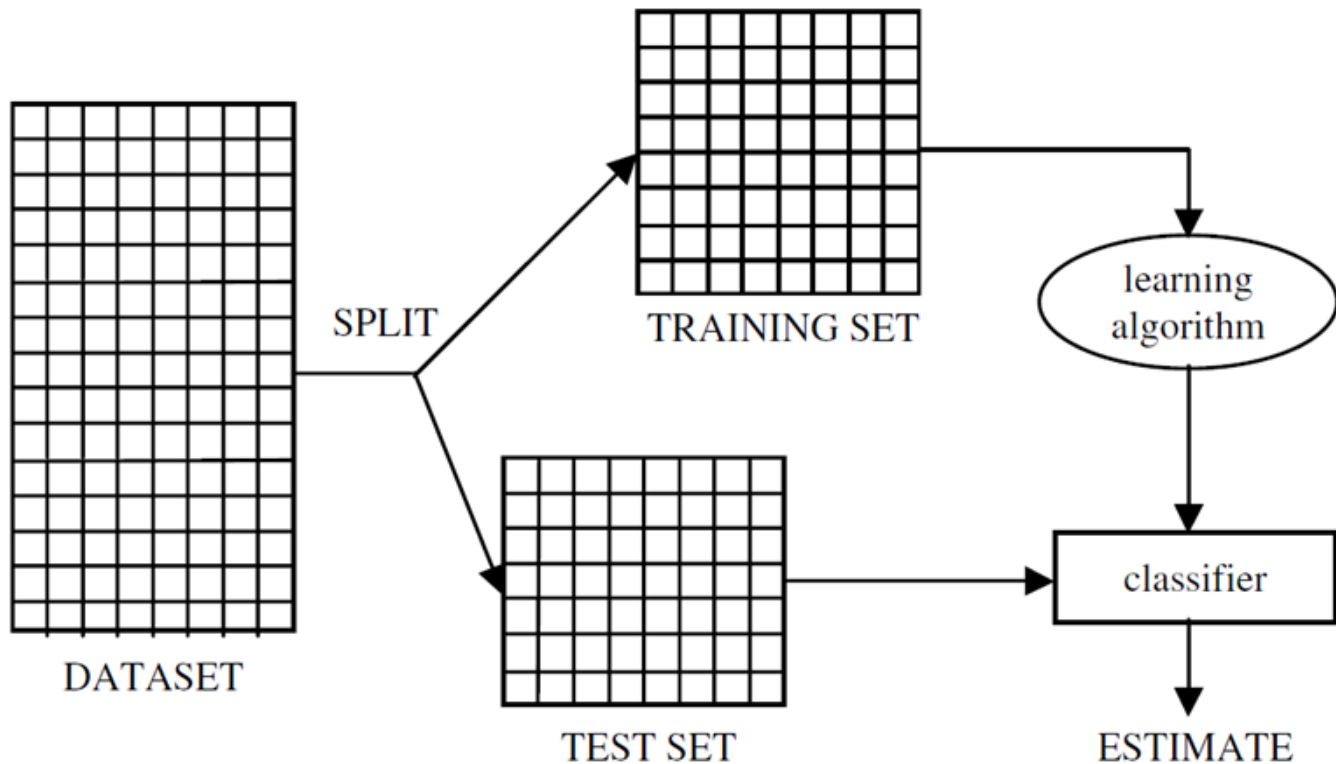


- Linear model has two parameters,  $\theta_0$  and  $\theta_1 \rightarrow 2$  degrees of freedom
- $\theta_1 = 0$ : 1 degree of freedom
- Regularization:  $\theta_1$  can vary but is forced to remain small = hyperparameter of learning algorithm

# Underfitting the Training Data

- Opposite of overfitting
- Main options for fixing this problem:
  - Select a more powerful model, with more parameters.
  - Feed better features to the learning algorithm (feature engineering).
  - Reduce the constraints on the model (e.g., reduce the regularization hyperparameter).

# Testing and Validating



## Tip:

It is common to use 80% of the data for training and hold out 20% for testing. However, this depends on the size of the dataset: if it contains 10 million instances, then holding out 1% means your test set will contain 100,000 instances, probably more than enough to get a good estimate of the generalization error.

# Hyperparameter Tuning and Model Selection

- Choosing between two types of model or different combinations of hyperparameters.
- If using same test set each time
  - optimizing for that particular test set.
  - “real” error might not be correct
- Solution:
  - use 3rd set: *holdout validation* (part of training set)
  - Train multiple models with various hyperparameters on reduced training set
  - Select best model according to validation set
  - Train this model on complete training set
  - Evaluate this model on test set



# Cross-validation

- If validation set is small → imprecise evaluations
- If validation set is too large → remaining training set too small → not ideal to compare candidate models
- Solution: cross validation
  - Many small validation sets
  - Each model is evaluated once per validation set after it is trained on the rest of the data
  - Average out all validations of a model
  - Drawback: training time is multiplied

# Artificial General Intelligence (AGI)

- Kunstmatige Algemene Intelligentie
- Voorgaande voorbeelden zijn zeer specifieke taken.
- Hypothetische intelligentie van een machine die de capaciteit heeft om elke intellectuele taak die een mens kan uitvoeren, te begrijpen of te leren.
- Zal AI het ooit overnemen en is de mens met uitsterven bedreigd?
  - Ja, volgens o.a.
    - Yuval Noah Harari (auteur *Sapiens*, *Homo Deus* en *21 lessen voor de 21e eeuw*)
    - Geoffrey Hinton (ex-Google), een van de grondleggers van AI-technologie
  - Neen, volgens o.a.
    - Yann LeCun (Chief AI Scientist by Meta), ir., ook een grondlegger van AI-technologie.
    - Andrew Ng (prof. AI in Stanford)

# Generatieve AI

- AI-systeem dat verschillende types nieuwe content kan creëren
  - Tekst
  - Beelden, video's
  - Programmacode
- Op basis van bestaande content
- Niveau van creativiteit en vrijheid kan gestuurd worden.

# Generatieve AI: eenvoudig voorbeeld

Genereer nieuwe foto's van persoon rond "zwaartepunt" van bestaande foto's

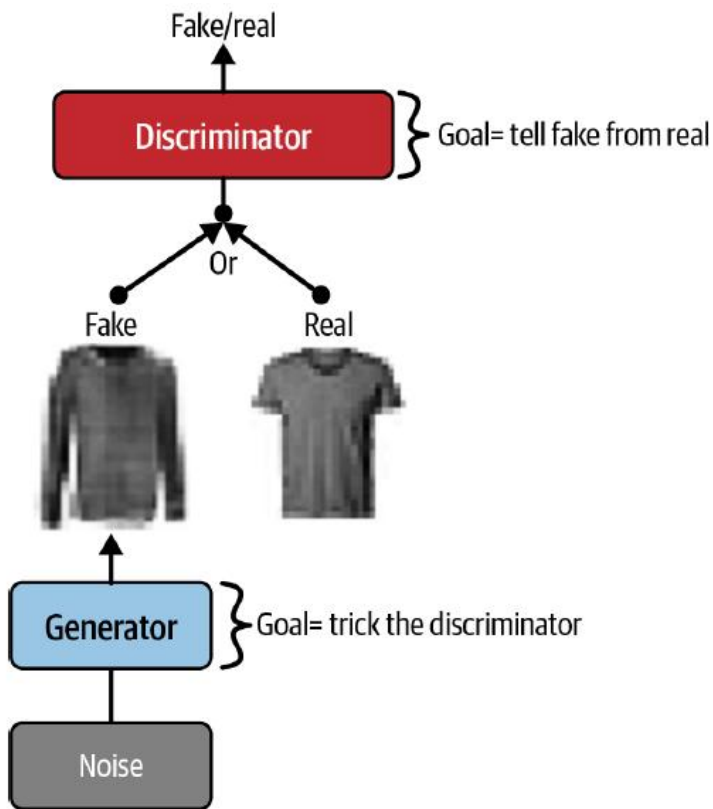


Bestaande foto's



Gegenereerde foto

# Generatieve AI: Generative Adversarial Networks (GANs)



- Generator en Discriminator worden afzonderlijk getraind.
- Doel is bereikt als Discriminator het onderscheid niet meer kan maken tussen *real* en *fake*.

# Generatieve AI: DALL-E

Genereer een beeld van een koalabeer op een motorfiets



# LLMs: Large Language Models

## Voorbeelden

- ChatGPT-4/5 (OpenAI, Microsoft): genereert tekst op basis van tekst en volgt een conversatie.
- Dall-E (OpenAI, Microsoft): genereert beeld op basis van tekst-input.
- Gemini (Google): cf. ChatGPT, zal worden geïntegreerd in Google Zoeken

## Wat?

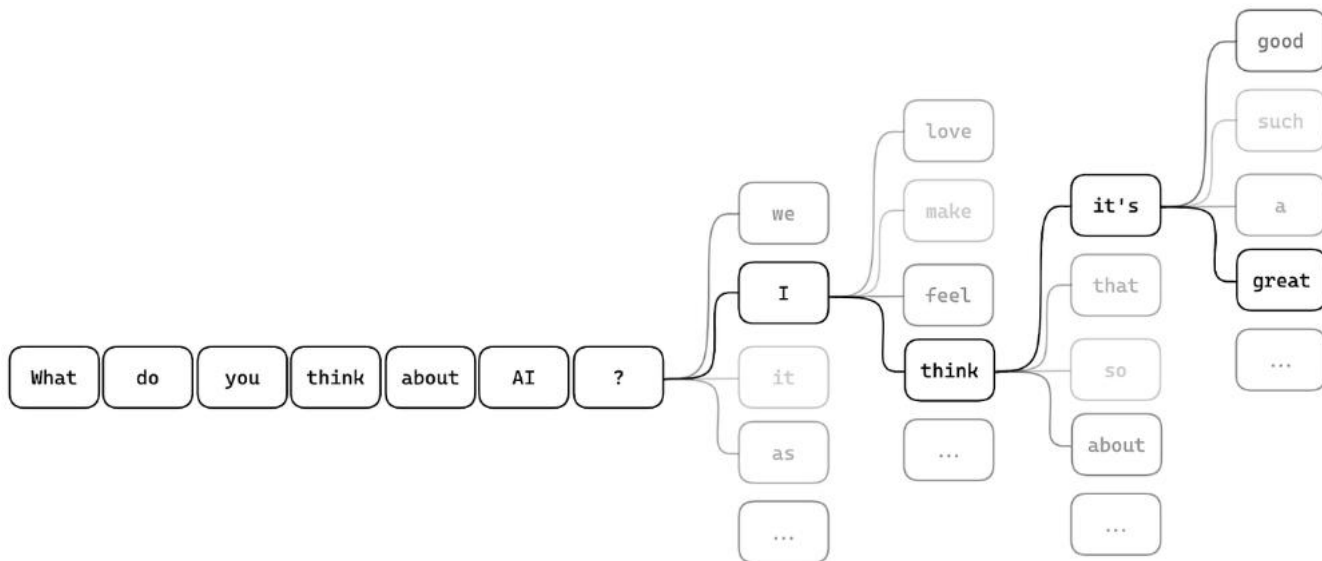
- ChatGPT = Generative Pretrained Transformer
- Teksten genereren op basis van gelijkaardige teksten
- Generatieve AI

## Waar zijn ze goed in?

- Begrijpen van tekst in verschillende talen
- Statistisch meest waarschijnlijke antwoord geven.

# LLMs: de basis

- Genereren de statistisch meest waarschijnlijke tekst.
- Genereren telkens het volgende woord in een zin.





# LLMs: teksten aanvullen

- Gmail

**Autocorrectie:**

- ☒ Autocorrectie aan  
☐ Autocorrectie uit

**Slim opstellen:**

(voorspellende schrijfsuggesties verschijnen tijdens het opstellen van een e-mail)

- ☒ Schrijfsuggesties staan aan  
☐ Schrijfsuggesties staat uit  
[Feedback over Smart Compose-suggesties](#)

**Smart Compose-personalisatie:**

(Smart Compose is gepersonaliseerd op basis van je schrijfstijl)

- ☒ Personalisatie aan  
☐ Personalisatie uit

**Gespreksweergave:**

(instellen of e-mails met hetzelfde onderwerp worden gegroepeerd)

- ☐ Gespreksweergave aan  
☒ Gespreksweergave uit

**Smart Reply:**

(Voorgestelde antwoorden tonen indien beschikbaar.)

- ☒ Smart Reply aan  
☐ Smart Reply uit

- Github copilot op Python-code in MS Visual Studio Code

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

```
rf = RandomForestRegressor(n_estimators=100, random_state=42)
```

← grijs = automatisch aangevuld

# LLMs: teksten aanvullen

- Microsoft 365 Copilot (Word, Excel, PowerPoint, Outlook, Teams)

*From Wikipedia, the free encyclopedia*

- Microsoft 365 Copilot is an **artificial intelligence assistant feature for Microsoft 365** applications and services
- Announced by Microsoft on March 16, 2023, the tool **builds on OpenAI's advanced GPT-4** large language models (LLMs) and incorporates Microsoft Graph in order to convert user text input into content in Microsoft 365 apps, such as Word, Excel, PowerPoint, Outlook, and Teams.
- Copilot is being marketed with a **focus on productivity for its users**, with 20 initial testers as of March 16, 2023.
- In **May 2023**, Microsoft expanded access to **600 customers** willing to pay for early access, with the office apps and services getting new Copilot features.
- Although there are public concerns about the chatbot, including hallucinations and racial or gender bias, experts believe that **Copilot may change the way that Microsoft users work and collaborate**.

- MS-Word

- According to Microsoft, Copilot can be used to **generate and edit text in Word documents based on user prompts**.
- Users can also ask Copilot to **push rewrite suggestions that strengthen the arguments of highlighted texts**.

# LLMs: belangrijkste functionaliteiten voor documentverwerking

- Samenvatten, vereenvoudigen
- Generatie van eerste draft op basis van input
- Stijl aanpassen: informeel, formeel, enthousiast, ...
- Chatgpt zelf uitproberen: <https://chat.openai.com/>

# LLMs: Foundation model

- Pretrained op massale (publieke?) data uit alle domeinen
- Reinforcement learning op basis van menselijke feedback om
  - Kwaliteit te verhogen
  - Toxische output te vermijden
- Toegankelijk via prompting

# LLMs: waarom disruptief?

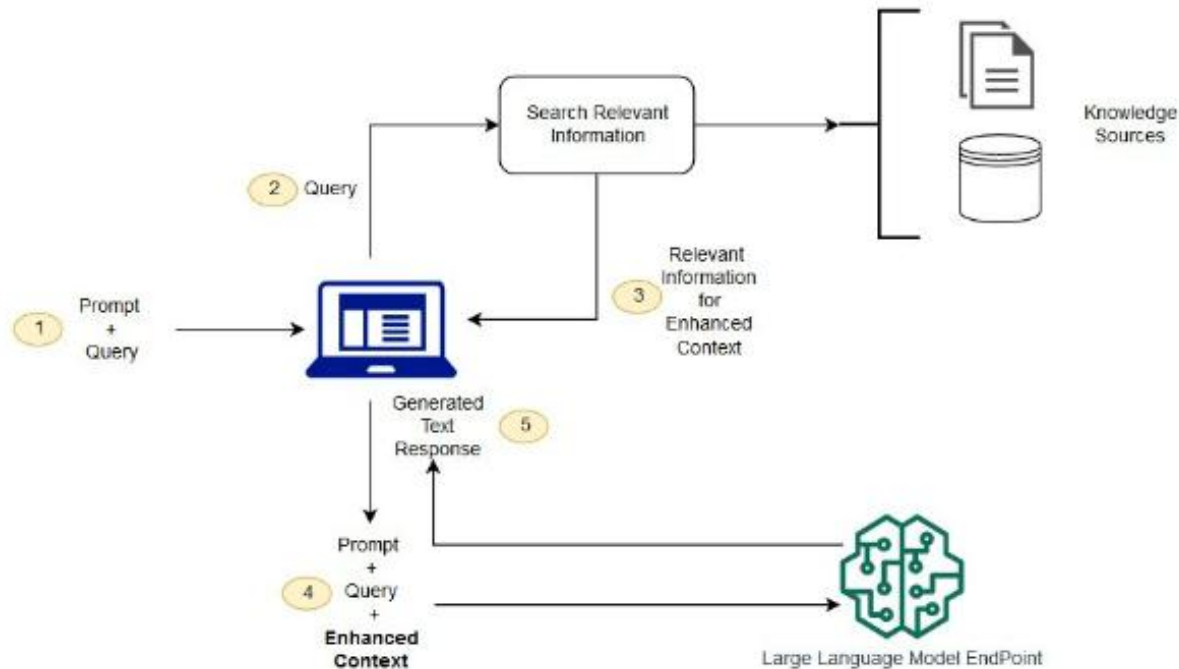
- Geschikt voor brede waaier aan taken, onmiddellijk bruikbaar.

*Fine-tuning voor specifiek domein ("corporate use") kan nuttig zijn, zie verder.*

- Gebruik via natuurlijke taal.
  - Zelf kunnen programmeren is niet meer noodzakelijk
  - Integratie in eigen software is mogelijk via API (Application Programming Interface)

# Corporate use: RAG

Domein-specifieke kennis toevoegen via  
*Retrieval Augmented Generation*



- Bronvermelding
- Veel minder kans op hallucinaties
- Eigen bronnen up-to-date houden