

# \*\*\* Applied Machine Learning Fundamentals \*\*\*

## Machine Learning Introduction

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Winter term 2023/2024



Find all slides on [GitHub](#) (DaWe1992/Applied\_ML\_Fundamentals)

# Lecture Overview

<b>Unit I</b>	<b>Machine Learning Introduction</b>
<b>Unit II</b>	Mathematical Foundations
<b>Unit III</b>	Bayesian Decision Theory
<b>Unit IV</b>	Regression
<b>Unit V</b>	Classification I
<b>Unit VI</b>	Evaluation
<b>Unit VII</b>	Classification II
<b>Unit VIII</b>	Clustering
<b>Unit IX</b>	Dimensionality Reduction

# Agenda for this Unit

## ① Introduction

Motivation  
Definition of Machine Learning

## ② Problem Types in Machine Learning

Type of Training Information  
Availability of Training Examples  
Type of Target Variable

## ③ Key Challenges in Machine Learning

Generalization from Training Data  
Feature Selection / Feature Engineering

Performance Measurement  
Model Selection  
Computation

## ④ Machine Learning Applications

Natural Language Processing  
Computer Vision  
Robotics

## ⑤ Wrap-Up

Summary  
Self-Test Questions  
Lecture Outlook

## Section: Introduction



# Why Machine Learning?

- ‘*We are drowning in information and starving for knowledge.*’  
– *John Naisbitt*
- Era of big data:
  - In 2017 there are about **1.8 trillion** web-pages on the internet
  - **20 hours** of video are uploaded to YouTube every minute
  - Walmart handles more than **1 million** transactions per hour and has data bases containing more than **2.5 peta-bytes** ( $2.5 \times 10^{15}$ ) of information
- **No human being can deal with this data avalanche!**

# Why Machine Learning? (Ctd.)

*'I keep saying the sexy job in the next ten years will be **statisticians** and **machine learners**. People think I'm joking, but who would've guessed that computer engineers would've been the sexy job of the 1990s? The ability to take data - to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it - that's going to be a hugely important skill in the next decades.'*

*– Hal Varian, Chief Economist at Google, 2009*

# Definition of Machine Learning

- '*[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.*'  
– ***Arthur Samuel, 1959***
- '*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*'  
– ***Tom Mitchell, 1997***

# A more abstract Definition

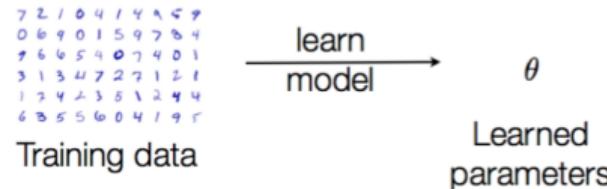
- Our task is to learn a mapping from input to output:  $h : \mathcal{I} \mapsto \mathcal{O}$
- Put differently, we want to predict the output from the input:

$$y = h(\mathbf{x}; \boldsymbol{\theta}) \quad \text{also: } y = h_{\boldsymbol{\theta}}(\mathbf{x})$$

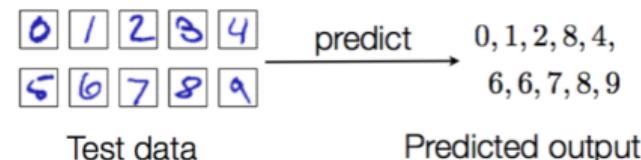
- $\mathbf{x} \in \mathcal{I}$  (Input)
- $y \in \mathcal{O}$  (Output)
- $\boldsymbol{\theta} \in \Theta$  (Parameters: What needs to be 'learned')

# General Paradigm

## Training



## Testing



Section:  
**Problem Types in Machine Learning**



# Type of Training Information

- **Supervised learning**
  - ‘Teacher’ provides **gold labels**
  - E. g. neural networks, decision trees, linear regression
- **Unsupervised learning**
  - Labels are **not** known during training
  - E. g. clustering, density estimation, association rule mining
- **Reinforcement learning**
  - Environment provides rewards for actions, but correct action is unknown
  - E. g. policy-iteration, Q-learning, SARSA
- **Semi-supervised learning** (partly labeled data)

# Type of Training Information (Ctd.)

## Supervised Learning

- A 'teacher' provides gold labels
- Neural networks, regression, ...

## Reinforcement Learning

- Feedback only
- No labels

## Semi-Supervised Learning

- Partly labeled data
- *not covered here*

## Unsupervised Learning

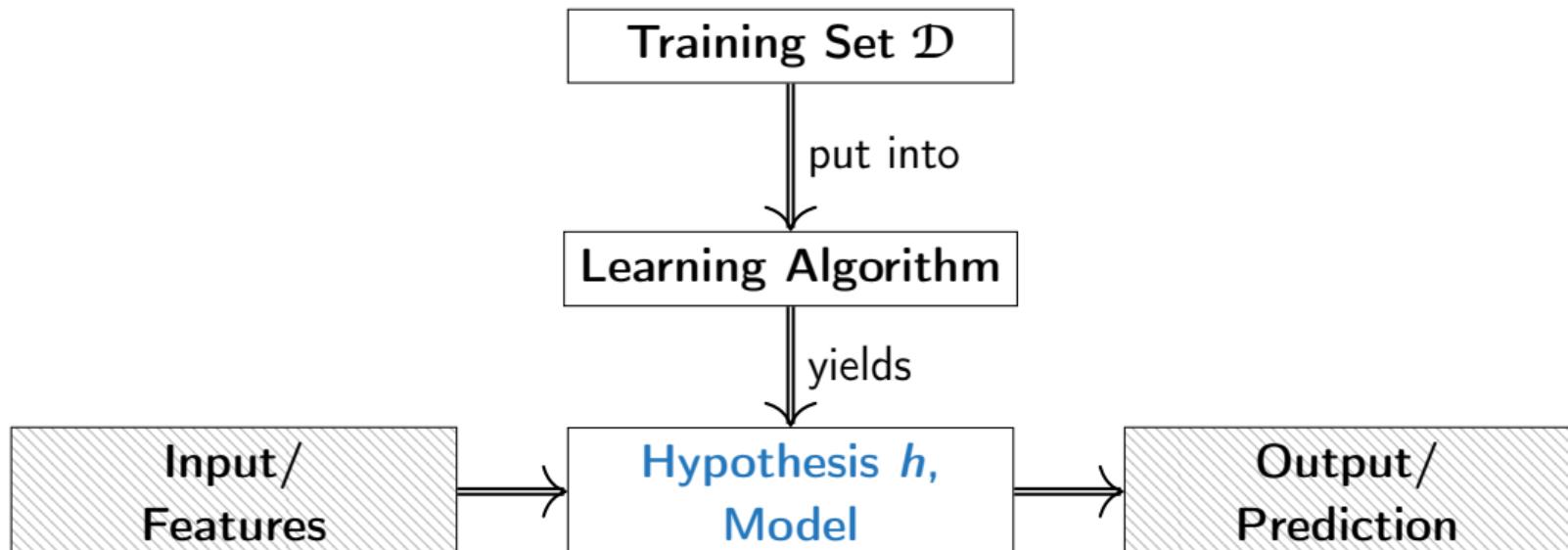
- No labels available
- Clustering, Apriori, ...

# Supervised Learning

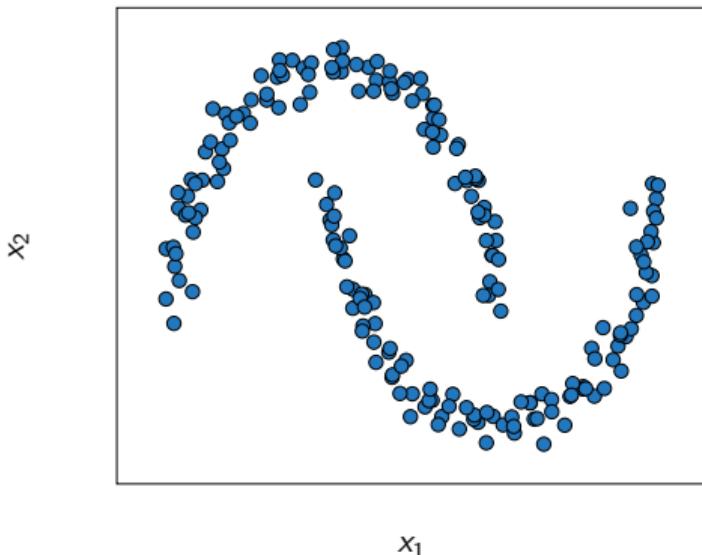
- A single row is called **example**
- An example without class label is called **instance**
- **Predictors:**
  - $\text{Outlook} \in \{\text{sunny}, \text{overcast}, \text{rainy}\}$
  - $\text{Temperature} \in \{\text{hot}, \text{mild}, \text{cool}\}$
  - $\text{Humidity} \in \{\text{high}, \text{normal}\}$
  - $\text{Wind} \in \{\text{weak}, \text{strong}\}$
- **Label:**
  - $\text{PlayGolf} \in \{\text{yes}, \text{no}\}$
  - Given a new instance, we want to predict its label
- **Label for the new instance???**

Outlook	Temperature	Humidity	Wind	PlayGolf
sunny	hot	high	weak	no
sunny	hot	high	strong	no
overcast	hot	high	weak	yes
rainy	mild	high	weak	yes
rainy	cool	normal	weak	yes
rainy	cool	normal	strong	no
overcast	cool	normal	strong	yes
sunny	mild	high	weak	no
sunny	cool	normal	weak	yes
rainy	mild	normal	weak	yes
sunny	mild	normal	strong	yes
overcast	mild	high	strong	yes
overcast	hot	normal	weak	yes
rainy	mild	high	strong	no
rainy	mild	normal	strong	???

# Supervised Learning: General Approach



# Unsupervised Learning



- There are **no** labels
- Try to find regularities in the data
- Examples for unsupervised learning:
  - Clustering
  - Density estimation
  - Dimensionality reduction

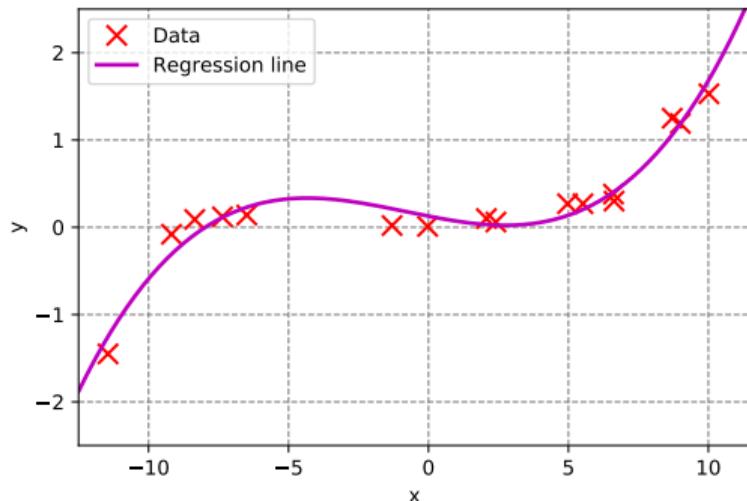
# Availability of Training Examples

- **Batch Learning**
  - The learner is provided with a fixed set of training examples
  - Cf. weather data set
  - E. g. neural networks, decision trees
- **Incremental / Online Learning**
  - Constant stream of training examples
  - The model is updated as new training examples arrive
  - E. g.  $k$ -nearest-neighbors
- Active Learning (*not covered*)

# Type of Target Variable: Regression

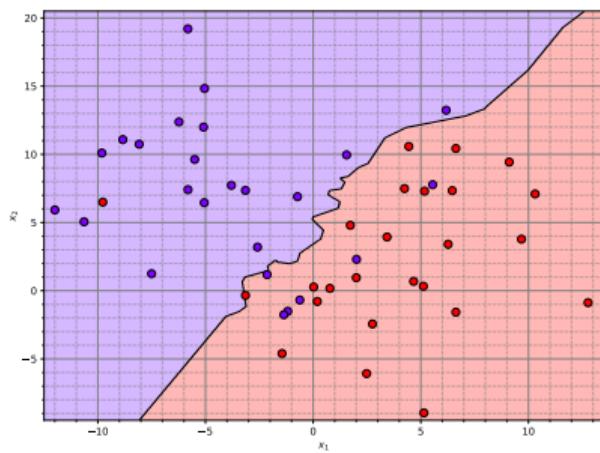
## Regression

- Learn a mapping into a continuous space
  - $\mathcal{O} = \mathbb{R}$
  - $\mathcal{O} = \mathbb{R}^3$
- E. g. curve fitting, financial analysis, housing prices, ...



# Type of Target Variable: Classification

## Classification



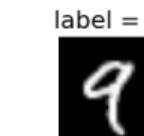
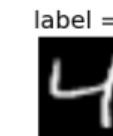
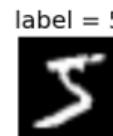
- Learn a mapping into a discrete space, e. g.
  - $\mathcal{O} = \{0, 1\}$  (binary classification)
  - $\mathcal{O} = \{0, 1, 2, 3, \dots\}$
  - $\mathcal{O} = \{\text{verb, noun, adverb, ...}\}$
- Examples:
  - Spam / no spam
  - Digit recognition
  - Part of speech tagging

Section:  
**Key Challenges in Machine Learning**



# Generalization from Training Data

- Learning does not mean memorizing the training data
- What if we see input that we **haven't seen before?**
- Example OCR (Optical Character Recognition):



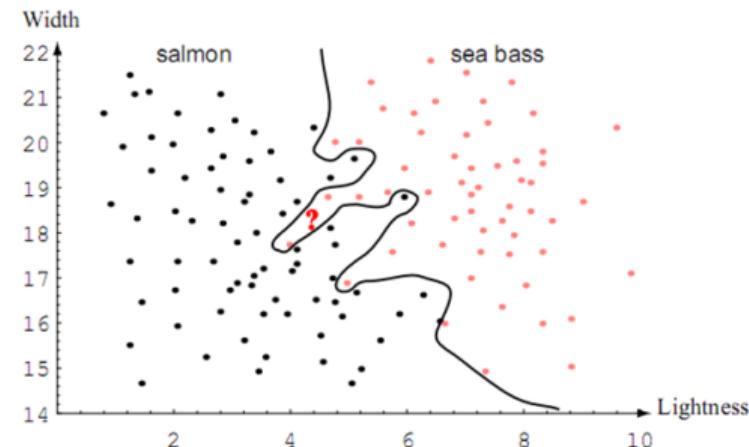
Hand-written digits from the MNIST data set

- Predict the character given the input image
- **People have different hand-writings**

# Generalization from Training Data (Ctd.)

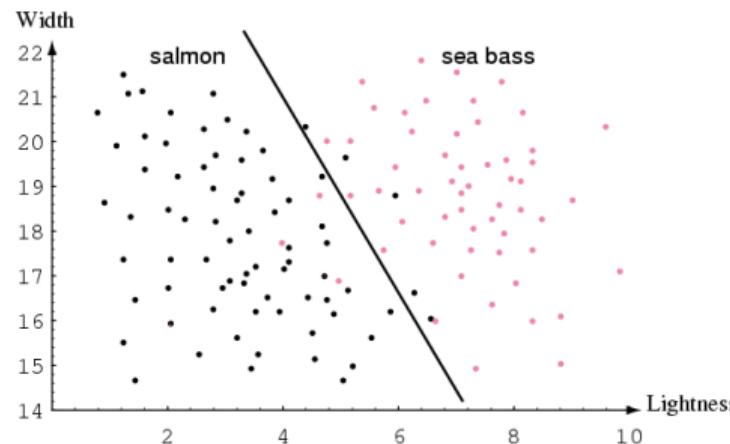
## What is the problem here?

- Complex decision boundary
- This leads to  **Overfitting** 
  - The model is too expressive...
  - ...and adapts to **idiosyncrasies** of the training data



**Solution:** Choose a simpler model (cf. **Occam's razor**)

# Generalization from Training Data (Ctd.)



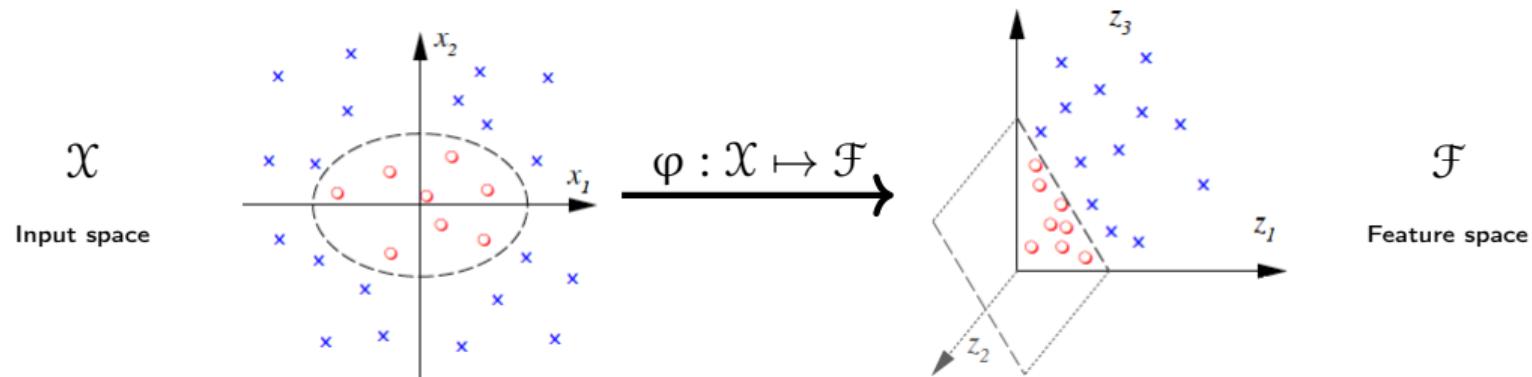
- Linear (less complex) model
- Allow for **misclassifications** of some training examples
- **Better generalization** to unseen instances

# A prominent Example of Overfitting



# Choosing the right Features

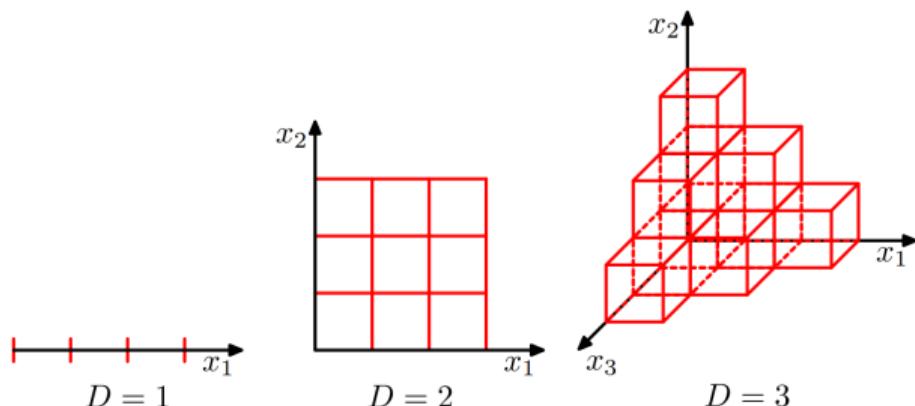
When stuck, move to a different perspective!



$$\varphi(x_1, x_2) \mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2) = (z_1, z_2, z_3)$$

# Choosing the right Features

But: Beware of the curse of dimensionality!



- Too many features significantly **slow down** the ML algorithm
- Need **exponential** amount of training data
- Dimensionality reduction

Image taken from [?], p. 35

# Performance Measurement

- How do we measure performance?
  - 99 % correct classification in speech recognition: What does it really mean?
  - *We understand the meaning of the sentence?, We understand every word?, For all speakers?*
- We need more **concrete numbers**:
  - % of correctly classified letters
  - Average distance driven (until accident, ...)
  - % of games won
  - % of correctly recognized words, sentences, etc.
- **Training vs. testing performance**

# Training vs. Testing Performance

- Evaluate on data which was **not used** for training (**out-of-sample**)

- Two-way split:

Train - Test

- Even better:

Train - Dev - Test

**Train:** Train model

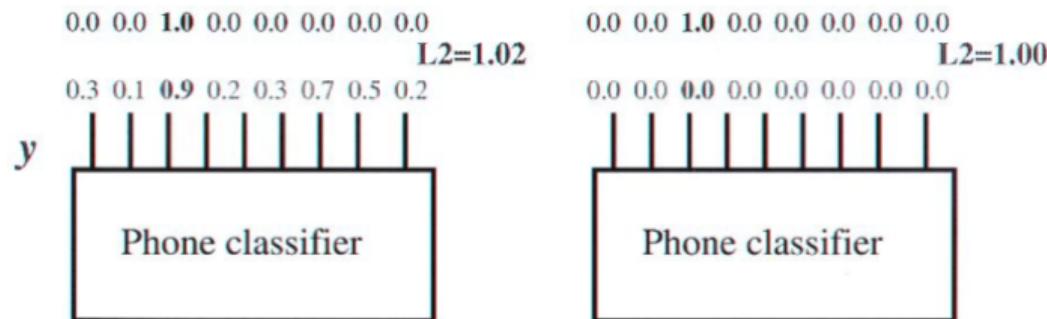
**Dev:** Tune hyper-parameters

**Test:** Test final model



# Performance Measurement (Ctd.)

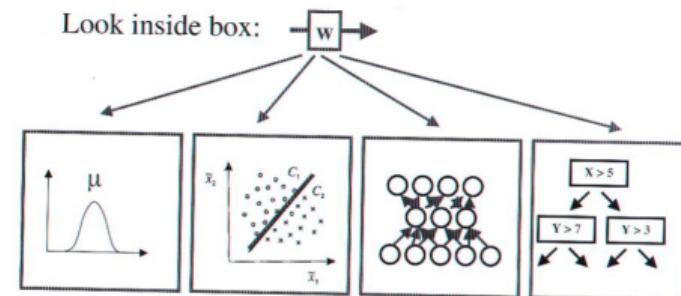
- We also need to define the right **error metric**:



- Which is better?
- Euclidean distance (L2-norm) might be useless

# Model Selection

- What is the **right model**?
- The learned parameters (here:  $w$ ) can mean a lot of different things:
- May characterize a family of functions
- May be parameters of a probability distribution
- $w$  may be a vector, adjacency matrix, graph, ...



# Computation

Even if the other problems are solved, **computation is usually quite hard**:

- Learning involves optimization of parameters
- Find / Search for best model parameters
  - GoogleNet has  $\approx$  6.5 million parameters
  - Often GPUs (**G**raphics **P**rocessing **U**nit) are needed
  - Google invented TPUs (**T**ensor **P**rocessing **U**nit)
- Often we have to deal with thousands, millions, ... of training examples
- Given a model, the prediction has to be computed efficiently

Section:  
**Machine Learning Applications**

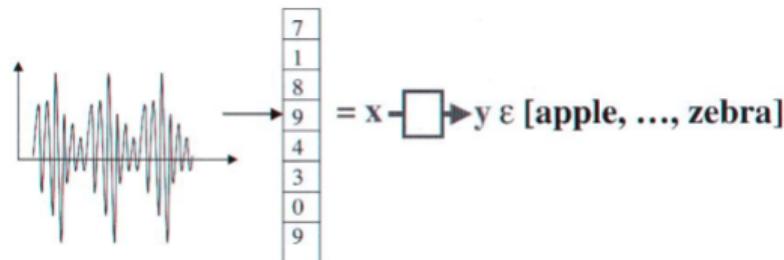


# Applications in Natural Language Processing

- **E-mail filtering:**

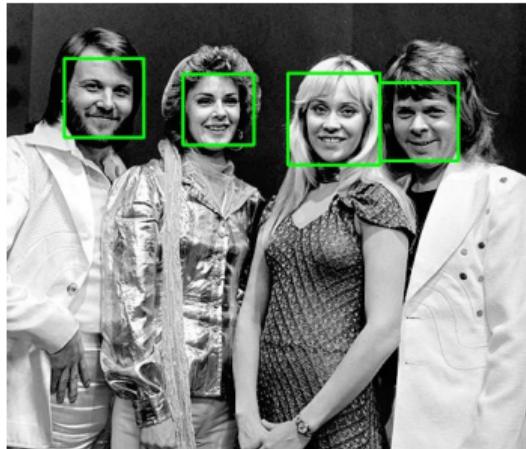
$$x \in [a-z]^+ \quad \rightarrow \quad y \in [\text{important, spam}]$$

- **Speech Recognition:**



# Applications in Computer Vision

Face detection:



Traffic sign detection:



# Applications in Computer Vision (Ctd.)

## Optical character recognition:

7 2 1 0 4 1 4 9 5 9  
0 6 9 0 1 5 9 7 3 4  
9 6 4 5 4 0 7 4 0 1  
3 1 3 4 7 2 7 1 2 1  
1 7 4 2 3 5 1 2 4 4  
6 3 5 5 6 0 4 1 9 5  
7 8 9 3 7 4 0 4 3 0  
7 0 2 9 1 7 3 2 9 7  
7 6 2 7 8 4 7 3 6 1  
3 6 9 3 1 4 1 7 6 9

Cf. Demonstration LeNet

# Applications in Robotics

**Robot control:**



**Autonomous driving:**



Section:  
**Wrap-Up**



# Summary

- Vast amounts of data are prohibitive for manual inspection
- Machine learning algorithms learn **without being explicitly programmed**
- **Important distinctions:**
  - Supervised learning ⇔ unsupervised learning
  - Classification ⇔ regression
- **Challenges:**
  - ① Generalization
  - ② Feature engineering
  - ③ Performance measurement
  - ④ Model selection
  - ⑤ Computation

# Self-Test Questions

- ① What is the difference between supervised and unsupervised learning?
- ② What is regression?
- ③ What does generalization mean?
- ④ '*The more features the better.*' Is this statement correct or not? Give reasons for your answer.
- ⑤ Why do we need train, dev and test sets?
- ⑥ True or false: 100 % train accuracy is desirable.
- ⑦ State some applications of machine learning.

# What's next...?

- Unit I** Machine Learning Introduction
- Unit II** Mathematical Foundations
- Unit III** Bayesian Decision Theory
- Unit IV** Regression
- Unit V** Classification I
- Unit VI** Evaluation
- Unit VII** Classification II
- Unit VIII** Clustering
- Unit IX** Dimensionality Reduction

**Thank you very much for the attention!**

**Topic:** \*\*\* Applied Machine Learning Fundamentals \*\*\* Machine Learning Introduction

**Term:** Winter term 2023/2024

**Contact:**

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

[daniel.wehner@sap.com](mailto:daniel.wehner@sap.com)

**Do you have any questions?**