

# MACHINE LEARNING

## WHAT IS MACHINE LEARNING?

Sebastian Engelke

MASTER IN BUSINESS ANALYTICS



**UNIVERSITÉ  
DE GENÈVE**

# What is machine learning?

Machine learning aims at programming a computer to learn from information (**Data**) to perform a task (**Prediction, etc.**) in an optimal way in terms of a performance measure (**Loss Function**).

Recent success of **machine learning** due to advances in its main ingredients:

- (1) **Data availability**: huge growth due to automation, digitalization, web applications.
- (2) **Computing power**: significant increase thanks to GPU computing, parallelization, etc.
- (3) **The learning algorithms**: theoretical developments of new statistical models and training methods.

# The statistical framework

## Types of variables:

- ▶ **quantitative** variables take values in an ordered set, typically the real numbers  $\mathbb{R}$ ;
- ▶ **qualitative** variables (also called **categorical** or **factor**) take values in a finite set  $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_q\}$  without ordering, e.g., {Yes, No}, {blue, red, green, yellow}, etc.

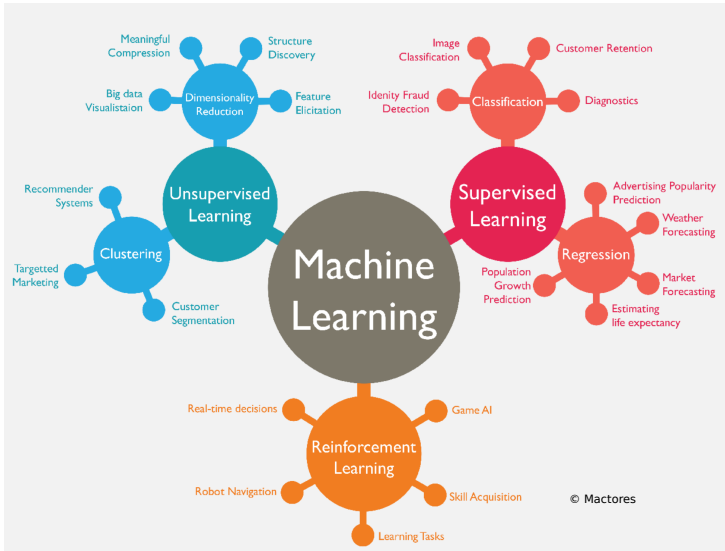
## Supervised learning

- ▶ We observe **training data**  $(x_i, y_i)$ ,  $i = 1, \dots, n$  where
  - ▶ the inputs  $x_i \in \mathbb{R}^p$  are called **predictors**, **covariates** or **features**;
  - ▶ the outputs  $y_i$  are called **responses** (in this course we assume they are univariate).
- ▶ The goal is to **predict** the (unknown) response  $y_0$  for a new (known) predictor  $x_0$ .
- ▶ **Regression**: the responses  $y_i \in \mathbb{R}$  are quantitative.
- ▶ **Classification**: the responses  $y_i \in \mathcal{G}$  are qualitative.

## Unsupervised learning

- ▶ We only observe multivariate data  $x_i \in \mathbb{R}^p$ ,  $i = 1, \dots, n$  but without a particular response.
- ▶ The goal is to **summarize**, **understand**, **interpret** and **visualize** the data.

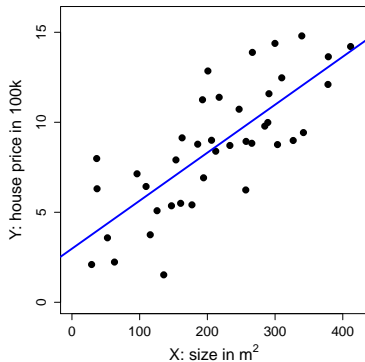
# Machine Learning: an overview



## Regression: simple linear model

**Data:** Observations  $(x_1, y_1), \dots, (x_n, y_n)$

**Goal:** Predict unknown outcome  $y_0$  for new predictor  $x_0$ .



► Linear regression

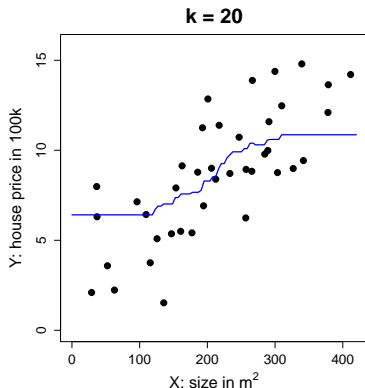
$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_0$$

- The numbers  $\hat{\beta}_0, \hat{\beta}_1 \in \mathbb{R}$  are estimated model parameters.

## Regression: the kNN method

**Data:** Observations  $(x_1, y_1), \dots, (x_n, y_n)$

**Goal:** Predict unknown outcome  $y_0$  for new predictor  $x_0$ .



► The k-Nearest-Neighbor (kNN) method

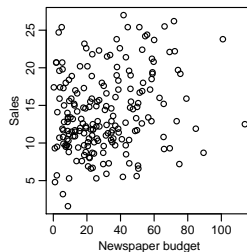
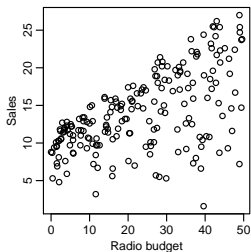
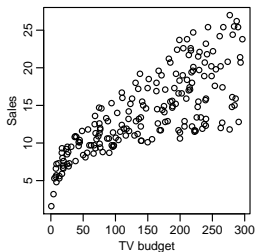
$$\begin{aligned}\hat{y}_0 &= \frac{1}{k} \sum_{i: x_i \in N_k(x_0)} y_i \\ &= \text{ave}\{y_i : x_i \in N_k(x_0)\},\end{aligned}$$

$N_k(x_0)$  are the  $k$  closest points  $x_i$  to  $x_0$ .

► The number  $k$  is a tuning parameter.

## Business application: advertising

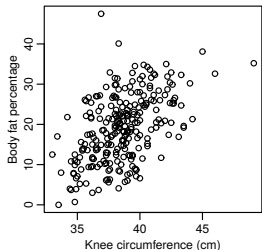
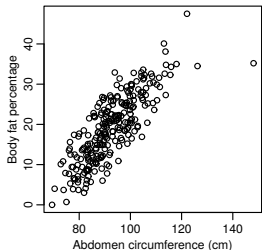
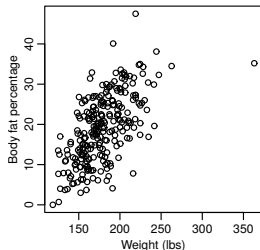
The **Advertising** data set contains **sales**, in thousands of units, as a function of **TV**, **radio**, and **newspaper** budgets, in thousands of dollars, for 200 different markets.



## Medical application: body fat

Data set **bodyfat** (R library **mfp**). The data set contains body fat estimates (**siri**) for 252 men with measurements of different body attributes. The first 10 measurements and a plot of the responses versus some of the predictors:

siri	age	weight	height	neck	chest	abdo	hip	thigh	knee	ankle	biceps	forearm	wrist
12.3	23	154.25	67.75	36.2	93.1	85.2	94.5	59.0	37.3	21.9	32.0	27.4	17.1
6.1	22	173.25	72.25	38.5	93.6	83.0	98.7	58.7	37.3	23.4	30.5	28.9	18.2
25.3	22	154.00	66.25	34.0	95.8	87.9	99.2	59.6	38.9	24.0	28.8	25.2	16.6
10.4	26	184.75	72.25	37.4	101.8	86.4	101.2	60.1	37.3	22.8	32.4	29.4	18.2
28.7	24	184.25	71.25	34.4	97.3	100.0	101.9	63.2	42.2	24.0	32.2	27.7	17.7
20.9	24	210.25	74.75	39.0	104.5	94.4	107.8	66.0	42.0	25.6	35.7	30.6	18.8
19.2	26	181.00	69.75	36.4	105.1	90.7	100.3	58.4	38.3	22.9	31.9	27.8	17.7
12.4	25	176.00	72.50	37.8	99.6	88.5	97.1	60.0	39.4	23.2	30.5	29.0	18.8
4.1	25	191.00	74.00	38.1	100.9	82.5	99.9	62.9	38.3	23.8	35.9	31.1	18.2
11.7	23	198.25	73.50	42.1	99.6	88.6	104.1	63.1	41.7	25.0	35.6	30.0	19.2

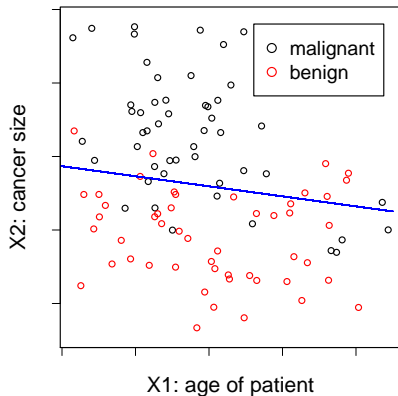




## Classification: simple linear regression

**Data:** Observations  $(x_1, y_1), \dots, (x_n, y_n)$ , where  $y_i \in \mathcal{G} = \{0, 1\}$

**Goal:** Predict unknown class  $y_0$  for new predictor  $x_0$ .



- Directly apply **linear regression** treating the classes as quantitative values 0 and 1. The prediction is

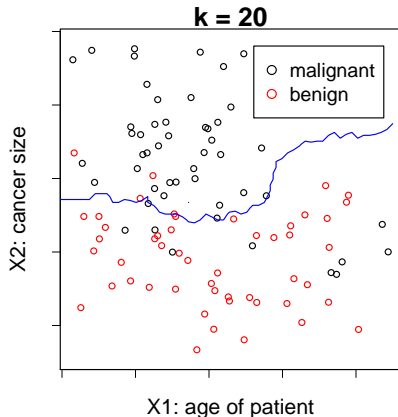
$$\hat{y}_0 = \begin{cases} 1, & \text{if } \hat{\beta}_0 + \hat{\beta}^T x_0 \geq 1/2, \\ 0, & \text{otherwise.} \end{cases}$$

- The numbers  $\hat{\beta}_0, \hat{\beta}_1 \in \mathbb{R}$  are estimated model parameters.

## Classification: the kNN method

**Data:** Observations  $(x_1, y_1), \dots, (x_n, y_n)$ , where  $y_i \in \mathcal{G} = \{0, 1\}$

**Goal:** Predict unknown class  $y_0$  for new predictor  $x_0$ .



- The **k-Nearest-Neighbor (kNN)** method predicts the majority of classes in the neighborhood

$$\hat{y}_0 = \begin{cases} 1, & \text{if } \frac{1}{k} \sum_{i: x_i \in N_k(x_0)} \mathbf{1}\{y_i = 1\} > 1/2, \\ 0, & \text{otherwise.} \end{cases}$$

$N_k(x_0)$  are the  $k$  closest points  $x_i$  to  $x_0$ .

- The number  $k$  is a tuning parameter.

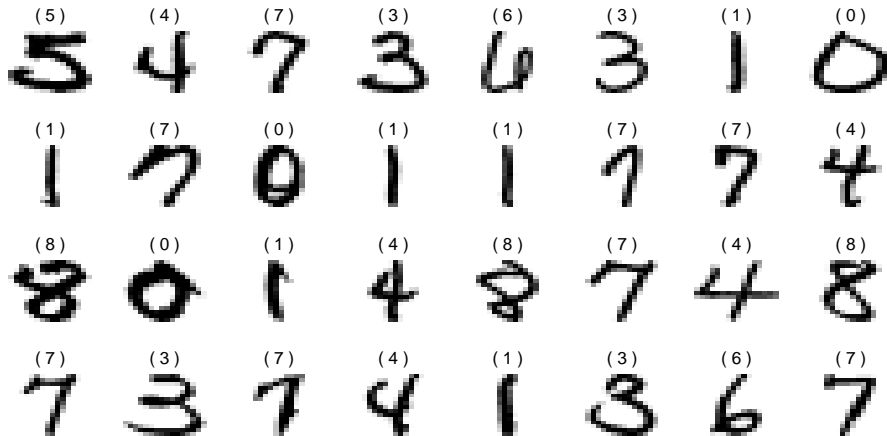
## Example: spam classification

The **spam** data set (R library **kernlab**). 4601 emails classified as one of two classes, i.e.,  $G \in \{\text{spam}, \text{nonspam}\}$ , based on and 57 predictors indicating the frequency of certain words and characters in the e-mail. A subset of the data:

	type	george	free	credit	money	hp	business	your	!	capitalTotal
1222	spam	0.00	0.00	5.19	0.64	0.00	1.29	1.29	0.09	135.00
1712	spam	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	186.00
2635	nonspam	0.66	1.33	0.00	0.22	3.34	0.00	0.44	0.37	411.00
4176	nonspam	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.04	97.00
928	spam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.40	495.00
4129	nonspam	0.00	0.00	0.00	1.16	0.00	0.00	1.16	0.49	34.00
4341	nonspam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.77	18.00
3036	nonspam	0.00	0.00	0.00	0.00	0.00	0.00	2.32	0.00	37.00
2890	nonspam	0.00	0.00	0.00	0.00	1.72	0.00	2.58	0.11	58.00
284	spam	0.00	0.40	1.81	0.60	0.00	1.61	2.62	1.45	513.00
946	spam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	339.00
811	spam	0.00	0.00	1.02	0.00	0.34	0.00	0.68	0.90	1330.00
3153	nonspam	0.38	0.00	0.00	0.00	0.90	0.00	0.00	0.00	1232.00
1763	spam	0.00	0.00	0.00	0.00	0.00	0.00	11.11	0.00	4.00
3532	nonspam	2.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	46.00
2283	nonspam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.00
3291	nonspam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
4547	nonspam	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00
1742	spam	0.00	0.00	0.00	0.00	0.00	0.00	0.47	0.69	239.00
3563	nonspam	4.16	0.00	0.00	0.00	8.33	0.00	0.00	0.00	30.00

## Example: handwritten characters/digits recognition

Data set from [\[ESL\]](#) homepage: handwritten digits ( $16 \times 16$  grey scale images). There are ten classes, namely,  $G = \{0, 1, \dots, 9\}$ .



## Example: CIFAR-10 data set

The **CIFAR-10 data set** contains 60'000 image of size  $32 \times 32$ . There are ten classes, namely,  $G = \{\text{airplane}, \text{automobile}, \dots, \text{truck}\}$ .

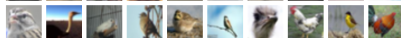
**airplane**



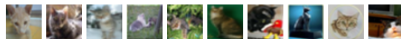
**automobile**



**bird**



**cat**



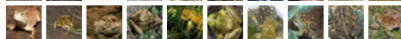
**deer**



**dog**



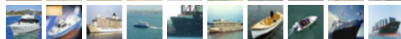
**frog**



**horse**



**ship**



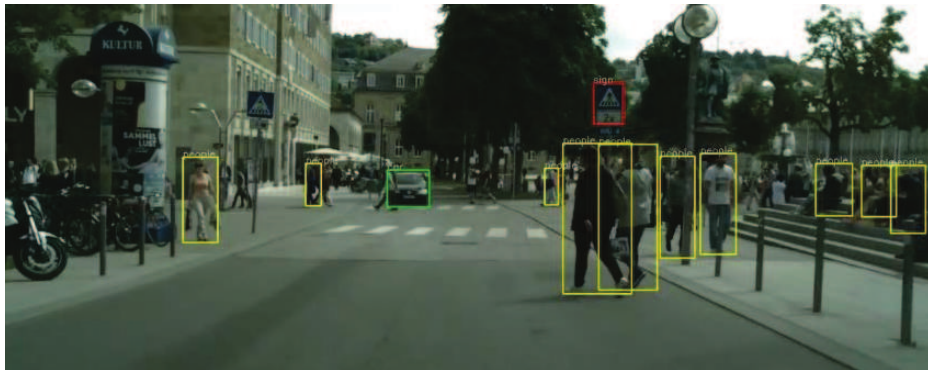
**truck**



## Example: smart cars

Smart cars classify objects according to

$G \in \{\text{cars, pedestrian, buildings, road signs, traffic lights, ...}\}$  and predict how these objects will move, and take decisions based on these predictions.



CES 2016: NVIDIA DRIVENet Demo - Visualizing a Self-Driving Future (part 5)  
([www.youtube.com/watch?v=HJ58dbd5g8g](http://www.youtube.com/watch?v=HJ58dbd5g8g))

## What is the right model?

*All models are wrong, some are useful.*

[George Box; Statistician]

*Everything should be made as simple as possible. But not simpler.*

[Albert Einstein; Physicist]

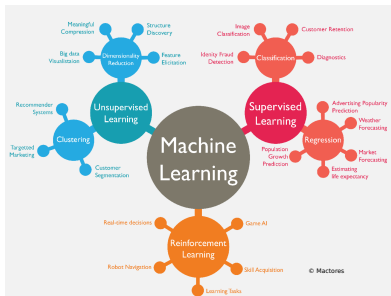
# In this course

## (1) Statistical concepts

- Modeling, inference, prediction.
- Model assessment and selection: training versus test error, cross-validation, etc.

## (2) Machine learning methods

- Regression: linear models, basis functions, regularization for high-dimensional data.
- Classification: linear methods, trees and random forest, support vector machines.
- Recent developments: convolutional (deep) neural networks, reinforcement learning, etc.



## (3) Applications

- Analysis of simulated and real data sets.
- Image recognition

## (4) Practical skills

- Programming with python.
- Working with data and interpretation of the results.



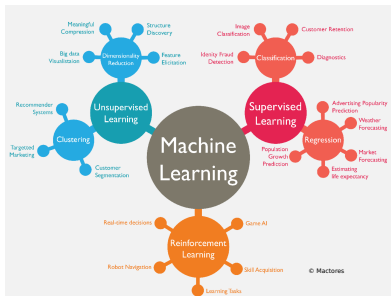
# In this course

## (1) Statistical concepts

- Modeling, inference, prediction.
- Model assessment and selection: training versus test error, cross-validation, etc.

## (2) Machine learning methods

- Regression: linear models, basis functions, regularization for high-dimensional data.
- Classification: linear methods, trees and random forest, support vector machines.
- Recent developments: convolutional (deep) neural networks, reinforcement learning, etc.



## (3) Applications

- Analysis of simulated and real data sets.
- Image recognition

## (4) Practical skills

- Programming with python.
- Working with data and interpretation of the results.