DATA SCIENCE

Dr. Anthony M.

CUK

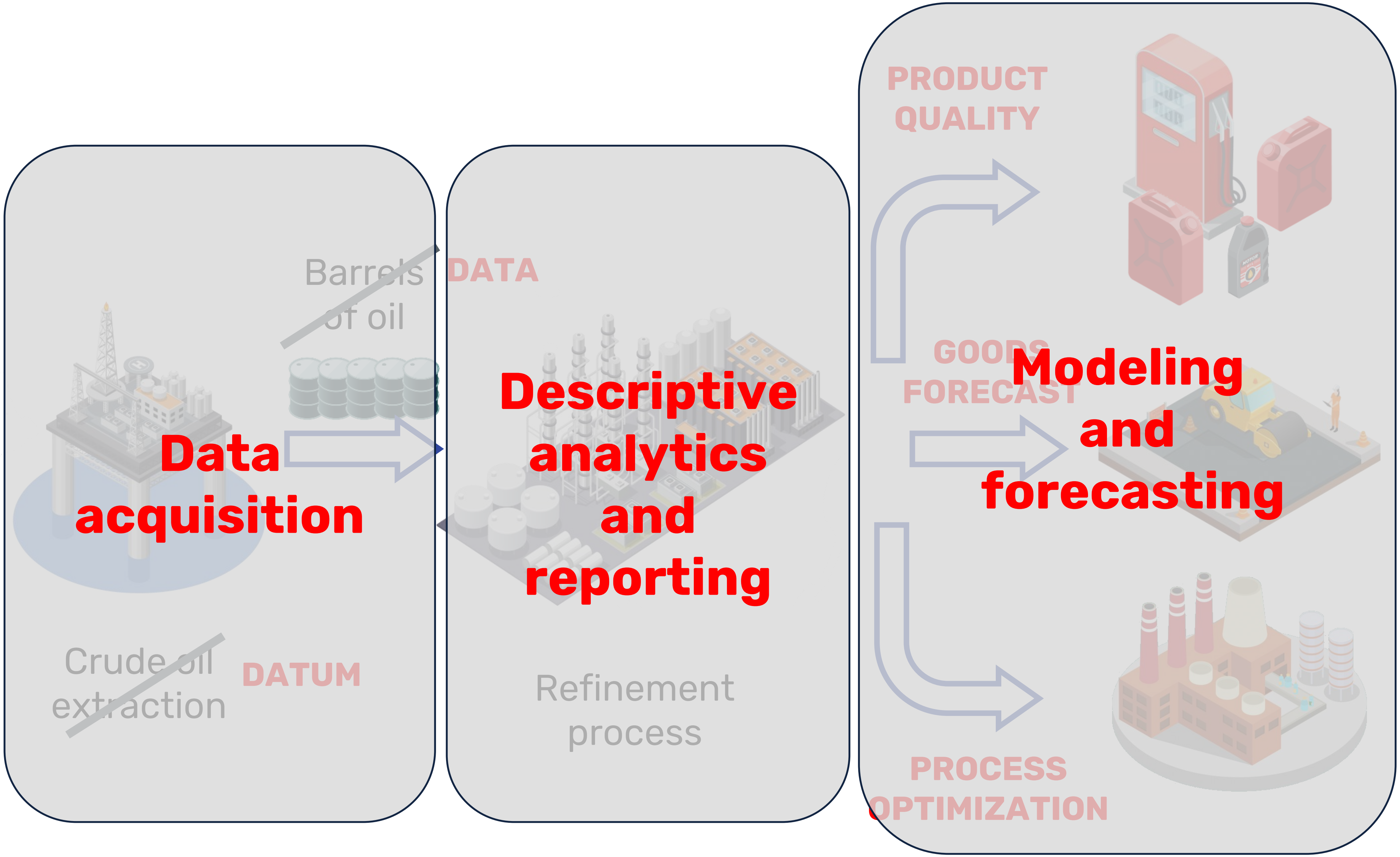
Introduction to Data Science

Course Information

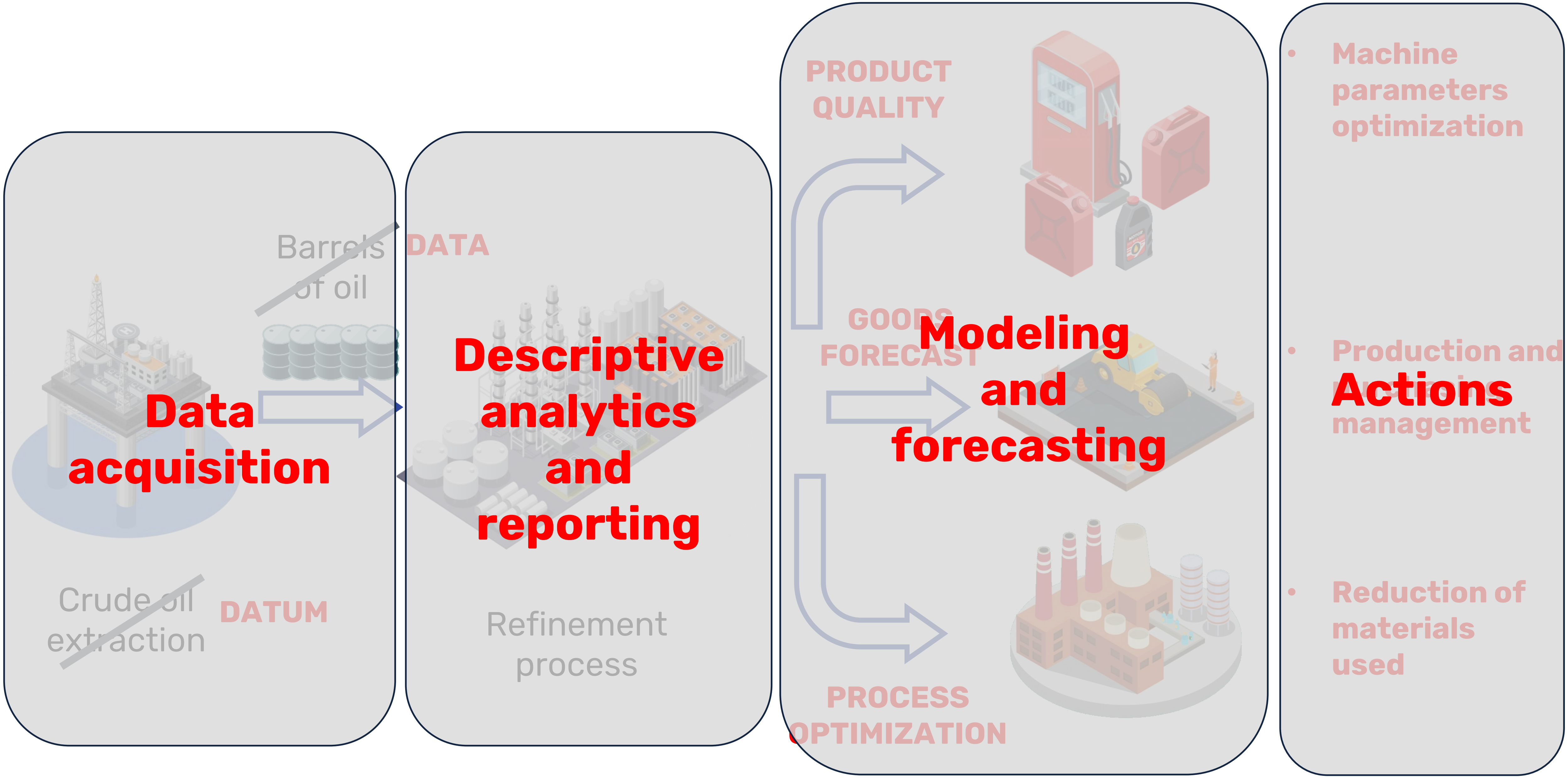
1. Assignments - 20%
2. CATs -30%
3. Exam -50%

Syllabus

1. Introduction to data science
2. Exploratory data analysis
3. Linear regression
4. Logistic regression
5. Overfitting and regularization
6. Validation and cross-validation
7. Decision trees
8. Neural networks
9. Convolutional neural networks
10. Clustering methods
11. Output-error method for system identification



- **Machine parameters optimization**
- **Production and purchasing management**
- **Reduction of materials used**



Data acquisition
~~Barrels of oil~~
DATA
~~Crude oil extraction~~
DATUM
Descriptive analytics and reporting
Refinement process
Modeling and forecasting
PRODUCT QUALITY
GOODS FORECAST
PROCESS OPTIMIZATION
Actions
Machine parameters optimization
Production and
management
Reduction of materials used

What is data science?

Data science is a set of fundamental principles, processes and techniques that guide the extraction of knowledge from data with the goal of **improving decision-making**

It is an interdisciplinary academic field that is based on:

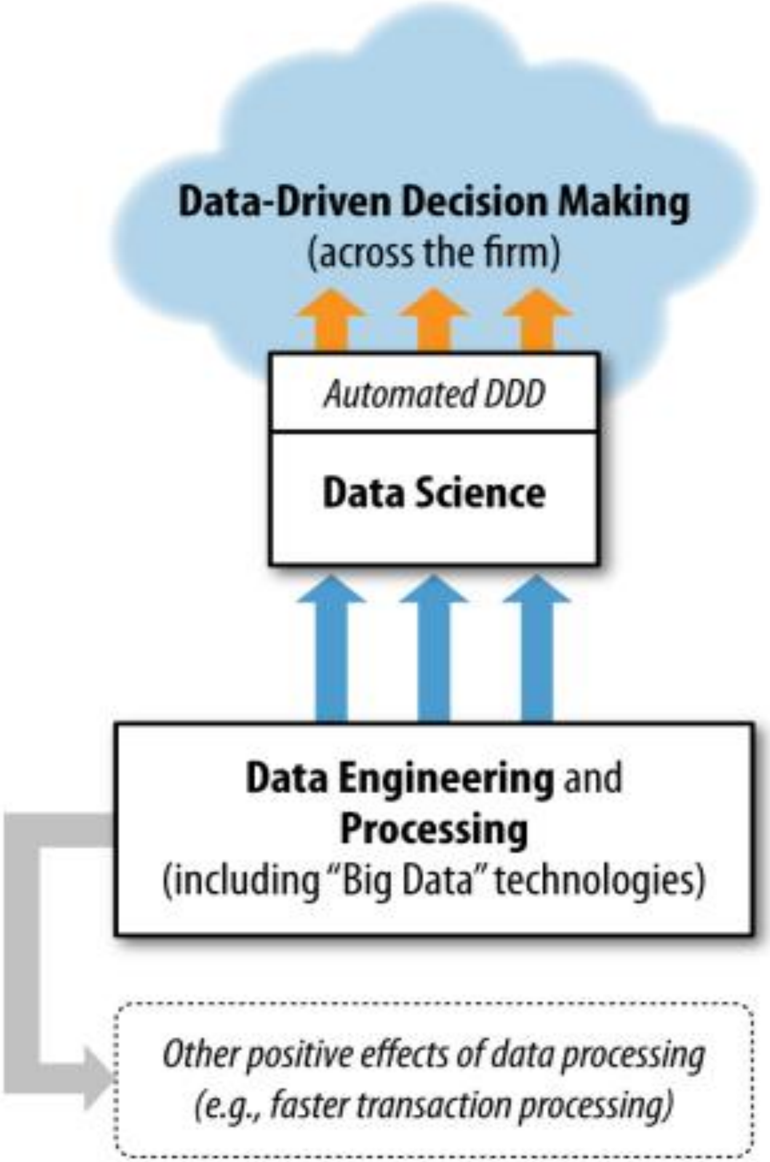
- Mathematics
- Statistics
- Machine learning and artificial intelligence
- Specialized programming

Data mining is the extraction of knowledge from data, via technologies that incorporate data science principles

The data-driven company

Data-driven decision-making (DDD) refers to the practice of basing decisions on the analysis of data, rather than purely on intuition [1, 2]

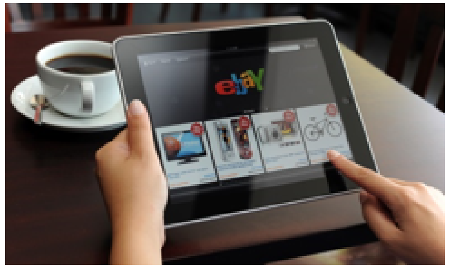
- Some decisions can be made **automatically** (finance, recommendations)
- **Data engineering and processing** support many data-oriented business tasks but do not necessarily involve extracting knowledge or data-driven decision making
- Data, and the capability to extract useful knowledge from data, should be regarded as **key strategic asset**
  - ✓ Need to invest to acquire the right data (even lose money)
  - ✓ Understand data science **even if you will not do it**



Picture taken from [1]

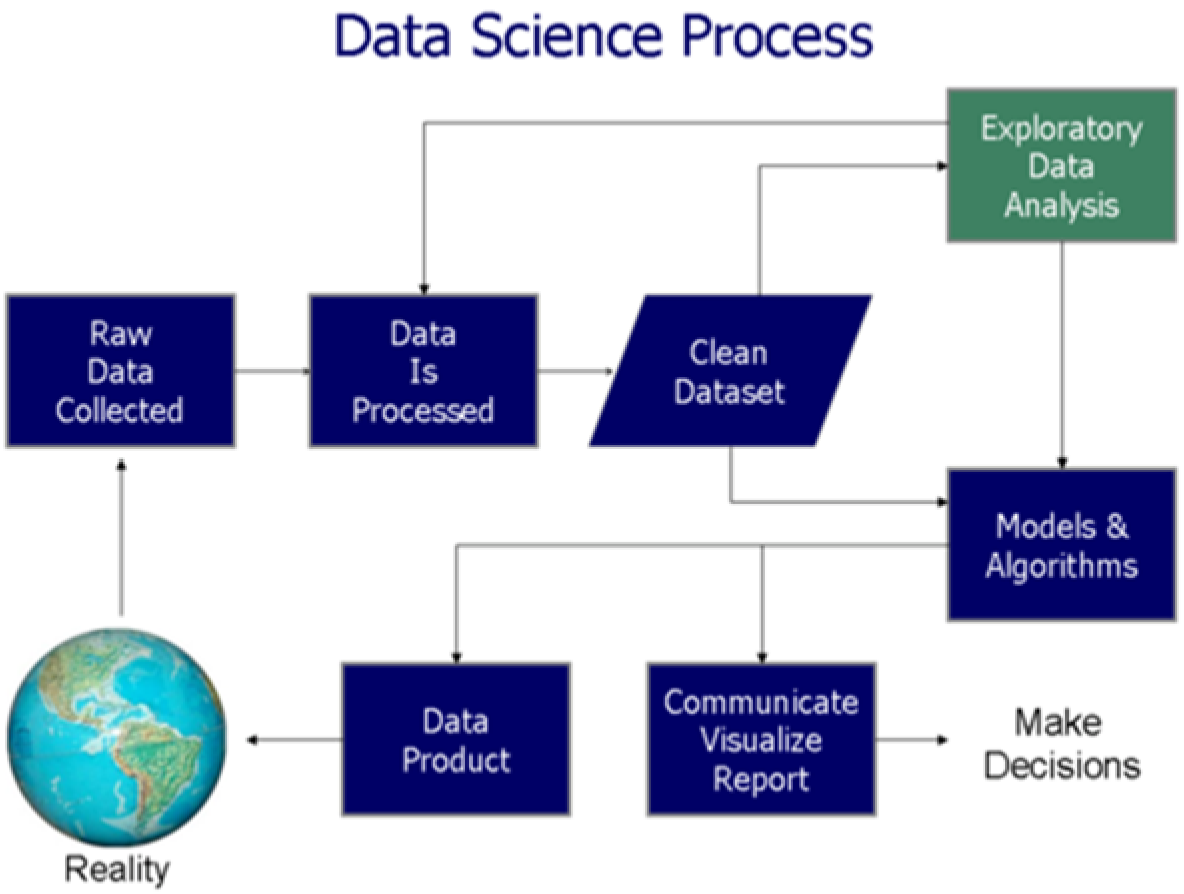
Data All Around

- **Lots of data is being collected and warehoused**
  - Scientific Experiments
  - Internet of Things
  - Web data, e-commerce
  - Financial transactions, bank/credit transactions
  - Online trading and purchasing
  - Social Network
  - etc

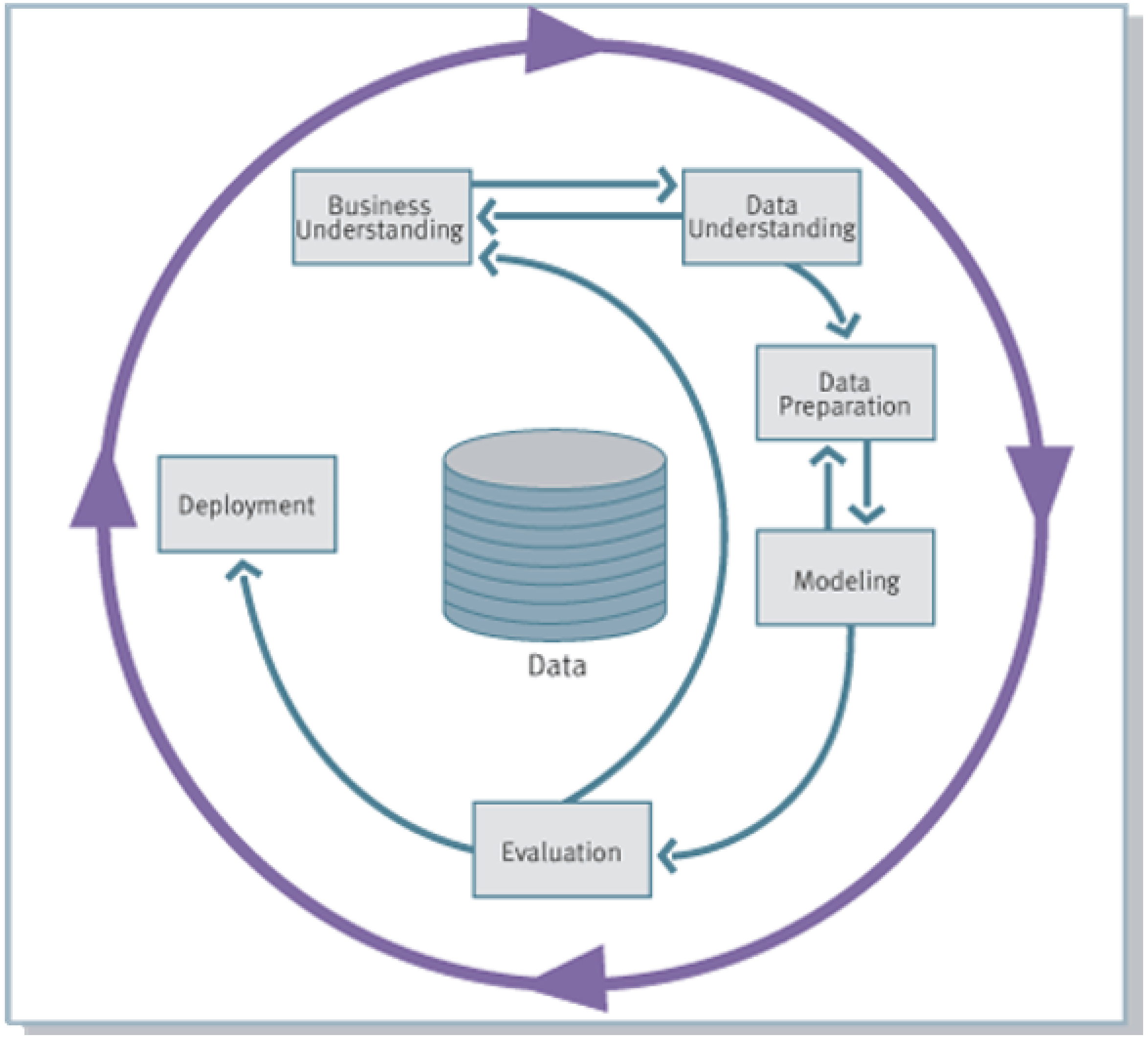


Data Science Process

Data science process flowchart (O'Neil and Schutt)



CRISP-DM (Cross Industry Standard Process for Data Mining)



Outline

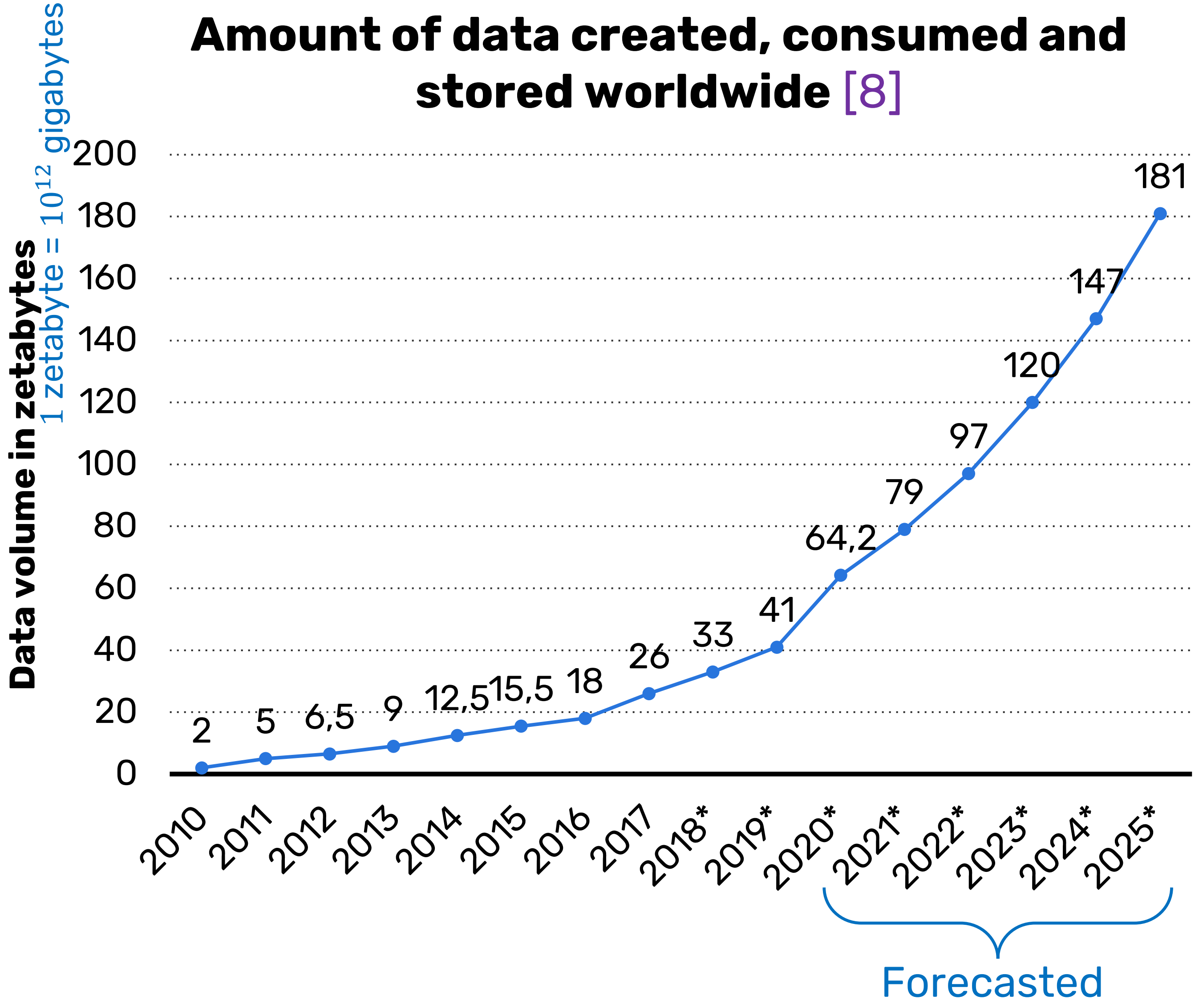
1. Course introduction
2. Data science and the data-driven company
3. **Data and its types**
4. What we are going to do with data (supervised and unsupervised learning)
5. Static and dynamical models in supervised learning
6. From business problems to data science tasks
7. The data mining life cycle (CRISP-DM)

What are data?

We refer to **data** as any piece of information that has been collected and stored in a computer

Examples:

- Sensor measurements
- Customer information
- Transaction history
- Social media posts
- …



Types of data: structured vs unstructured

Structured data

Data that are organized following a predefined scheme and stored in tabular formats (excel sheets, SQL databases…)

House area [feet^2]	**# bedrooms**	**Price** [k$]
523	1	115
645	1	150
708	2	210
⋮	⋮	⋮

Unstructured data

Data that can have an internal structure but do not follow a predefined data model or scheme

Audio files

Text files

Video files

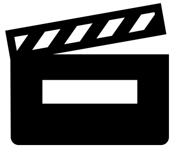
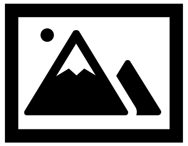


Image files



Types of data: quantitative vs qualitative

Ordinal qualitative data
can be ordered. Other examples:
low/high income, age ranges…

Nominal qualitative data
cannot be ordered

Runner name	**Sex**	**Placement**	**Time** [seconds]
Orlando Dillon	M	First	14.75
Izabella Kent	F	Second	15.01
Sophia Sanders	F	Third	15.33
⋮	⋮	⋮	

Qualitative (or categorical) data

assume non-numerical values, typically
belonging to pre-defined categories

Quantitative (or continuous) data

assume numerical values

Data are dirty

Common data problems:

- Missing values
- Unlikely values (outliers)
- Inconsistent formats
- ...

House area [feet^2]	**# bedrooms**	**Completion date**	**Price** [k$]
523	1	23/06/1998	115
645	1	01/07/2000	0.001
708	unknown	19/01/1980	210
1034	3	31-Jan-2001	unknown
unknown	4	17/12/2005	355
2545	unknown	14/02/1999	440
⋮	⋮	⋮	⋮

Typically, data must be cleaned before usage (**data cleaning**)

Outline

1. Course introduction
2. Data science and the data-driven company
3. Data and its types
4. **What we are going to do with data (supervised and unsupervised learning)**
5. Static and dynamical models in supervised learning
6. From business problems to data science tasks
7. The data mining life cycle (CRISP-DM)

What are we going to do with data?

In this course, we will use data for:

- **Descriptive analysis** and **visualization**

- **Supervised learning** (in particular, regression and classification)

- **Unsupervised learning** (in particular, clustering and dimensionality reduction)

Supervised vs unsupervised learning

Many data science tasks can be tackled either by supervised or unsupervised learning methods

- **Supervised learning**: predict the values of one or more **dependent variables** (**output(s)**) based on the values of one or more **independent variables** (**input(s)**)

$$\boldsymbol{\varphi} \longrightarrow \boldsymbol{y}$$

Inputs (Features) → **Outputs (Targets)**

Typically, we will focus on supervised learning problems with **<u>only one</u> output**

- **Unsupervised learning**: there are **<u>no</u> outputs**! The goal may be to discover groups of similar entities within the data or to project the data from a high-dimensional space (#**inputs** > 3) down to two or three dimensions for the purpose of visualization

Data science tasks

- **Regression***: predict the values assumed by the **continuous** **output(s)** from the **input(s)**

Example:
- ➢ Predict the **prices** of houses based on their **area**
- ➢ Predict the **prices** of houses based on their **area** and **number of bedrooms**

House area [feet^2]	**# bedrooms**	**Price** [k$]
523	1	115
645	1	150
708	2	210
⋮	⋮	⋮

$\varphi \in \mathbb{R}$

$y \in \mathbb{R}$

$\boldsymbol{\varphi} \in \mathbb{R}^{2\times 1}$

*: covered in this course : supervised : unsupervised

Data science tasks

- **Classification***: predict the values assumed by the **categorical** **output(s)** from the **input(s)**

Example: ➢ Develop an application that recognizes cats in **images**

Image	Label
	Cat
	Not cat
	Cat
	Not cat

Input: an image

$\varphi =$  $\in \mathbb{N}^{W \times H \times D}$

Images are basically matrices of numbers that describe color intensity

Output: the class label

$$y \in \{\text{Cat}, \text{Not cat}\}$$

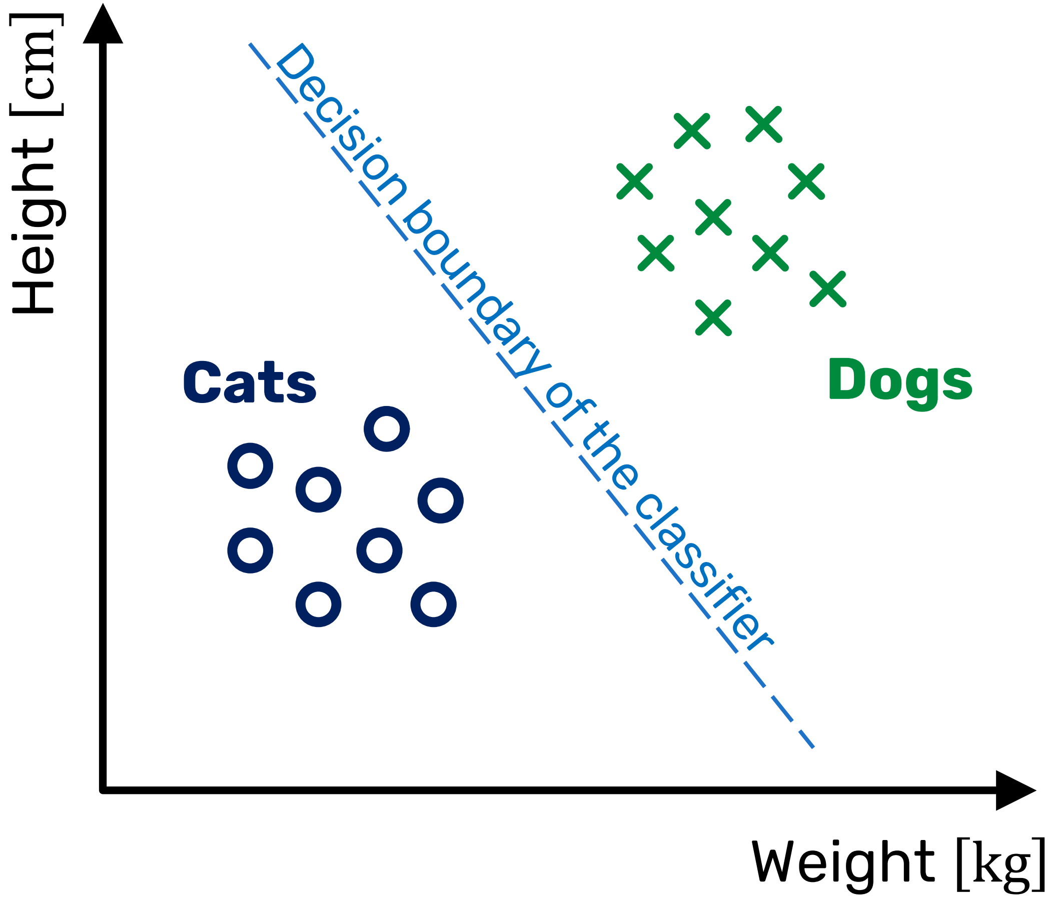
(single output)

*: covered in this course : supervised : unsupervised

Data science tasks

- **Classification***: predict the values assumed by the **categorical** **output(s)** from the **input(s)**

Example: ➢ Distinguish cats from dogs based on their **height** and **weight**



$\boldsymbol{\varphi} \in \mathbb{R}^{2\times1}$

(height and weight of the animal)

Output: the class label

$y \in \{\mathrm{cat}, \mathrm{dog}\}$

(single output)

*: covered in this course : supervised : unsupervised

Data science tasks

- **Causal modeling**: identify which **inputs** (**causes**) actually influence the **outputs** (**effects**) and, possibly, to what extent

  Example: ➢ Did a particular marketing campaign influence the consumers to purchase our product?

  Causal modeling typically involves substantial investments in data, such as randomized controlled experiments (**A/B tests**) and sophisticated methods for drawing causal observation data (**"counterfactual" analysis**)

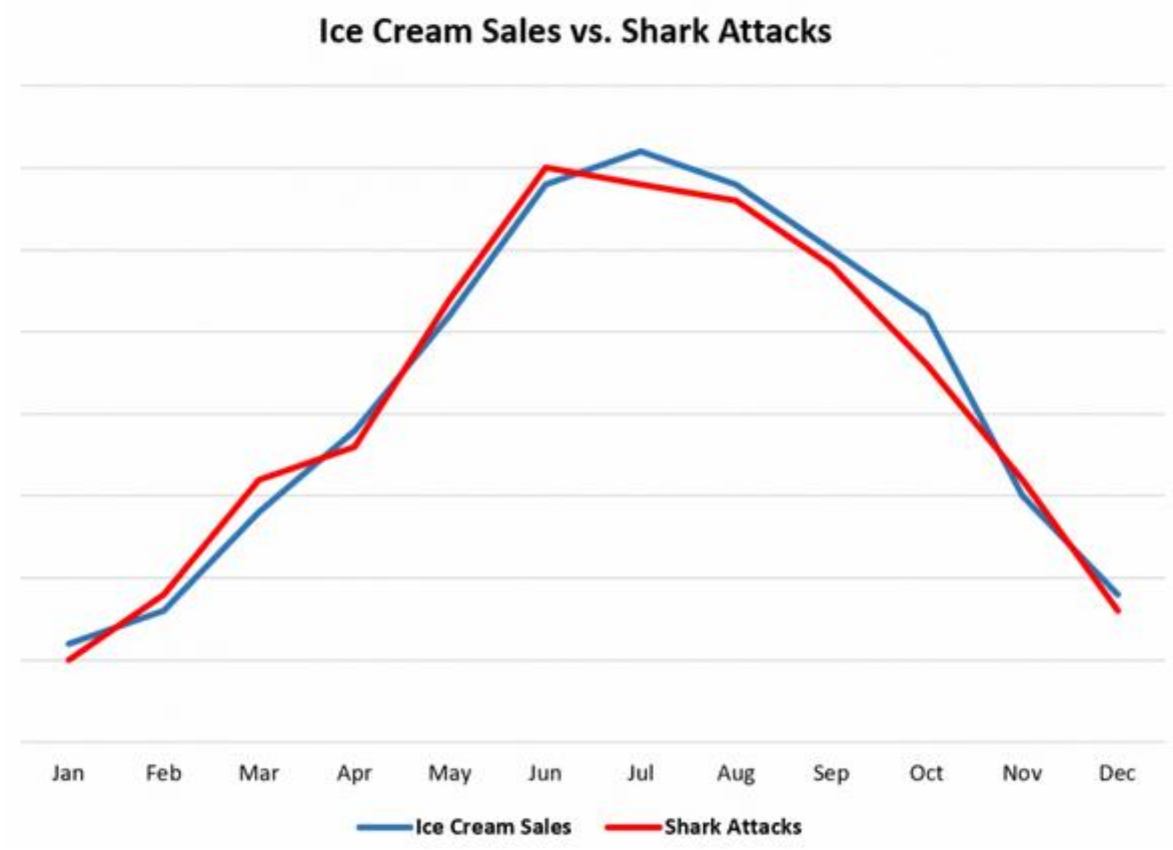
  What would be the difference in sales if we used an advertisement instead of another?

  Technical note: regression and classification are based on correlation, causal modeling is based on causality

*: covered in this course : supervised : unsupervised

Data science tasks

- **Causal modeling**: identify which **inputs** (**causes**) actually influence the **outputs** (**effects**) and, possibly, to what extent



Picture taken from [9]

Correlation does not imply causation!

If we take a look at the data representing monthly ice cream sales and monthly shark attacks around the United States each year, we can see that the two variables are highly correlated

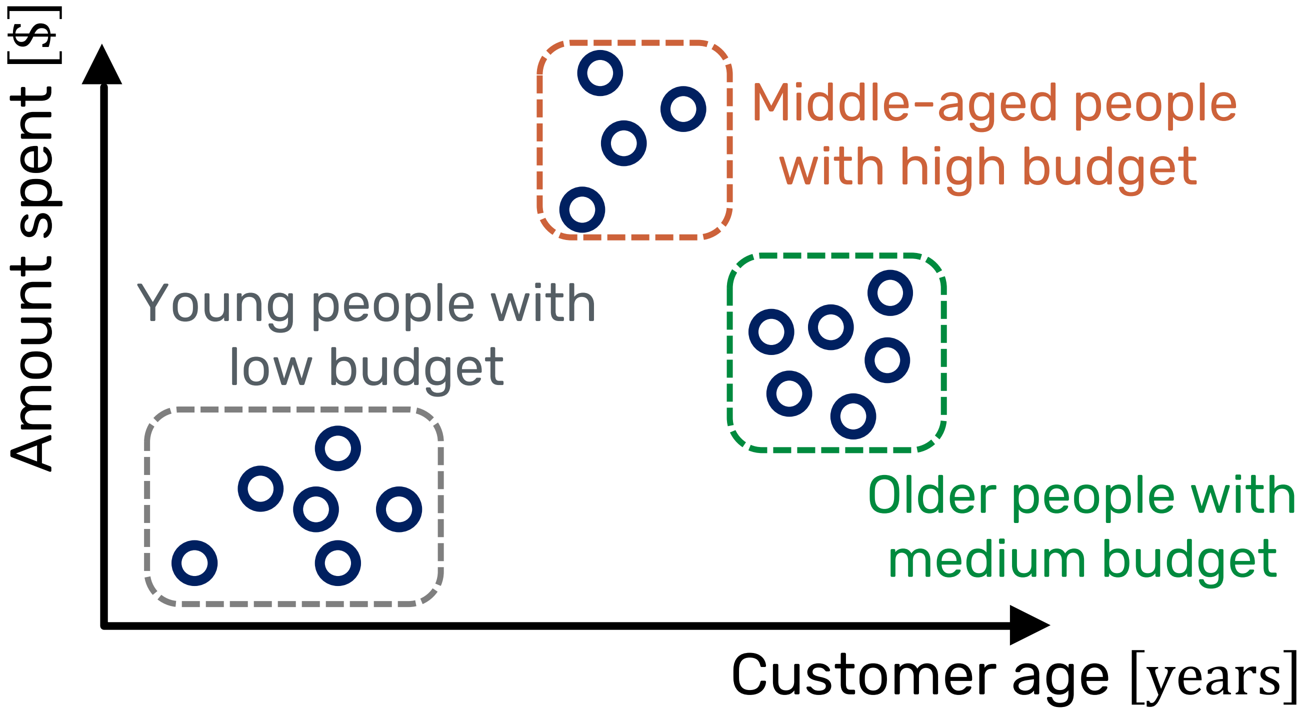
- Does this mean that consuming ice cream causes shark attacks? No! The more likely explanation is that more people consume ice cream and get in the ocean when it's warmer outside, explaining the high correlation

*: covered in this course : supervised : unsupervised

Data science tasks

- **Clustering***: organize the data into different groups based on their similarity

Example: ➢ Understand which types of customers are similar to each other by grouping individuals according to several **characteristics** → personalized marketing campaigns



$\boldsymbol{\varphi} \in \mathbb{R}^{2\times 1}$
(customer age and amount spent)

Output: none

*: covered in this course | : supervised | : unsupervised

Data science tasks

- **Co-occurrence grouping**: find associations between different entities (characterized by a set of **features**) based on transactions involving them

Example: ➢ What items are commonly purchased together? (**market basket analysis**)

Clustering looks at the similarity between entities based on their features, co-occurrence grouping considers the similarity of entities based on their appearing together in transactions (e.g., "a keyboard is not similar to a mouse, although they are typically bought together")

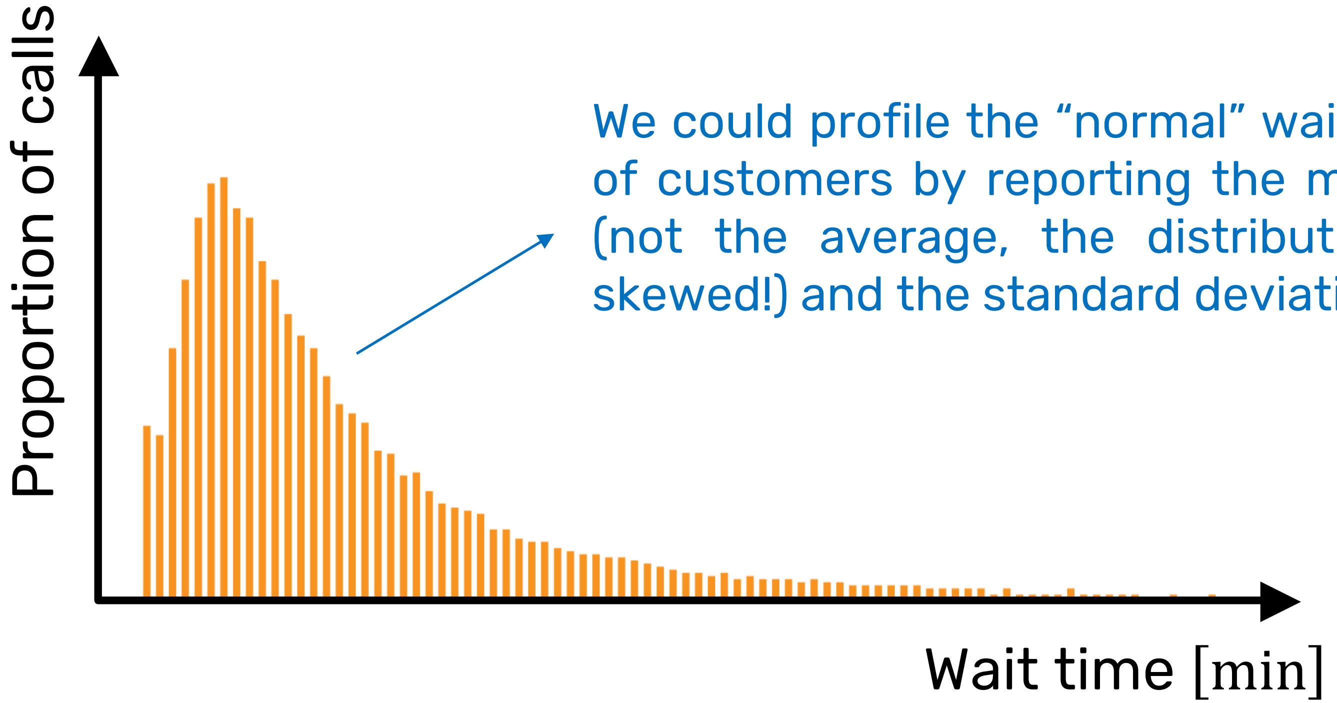
*: covered in this course | : supervised | : unsupervised

Data science tasks

- **Profiling**: find the typical behavior of an individual, group or population

Example:
➢ What is the typical credit card usage of a customer segment?

➢ Profile the typical wait time of customers who call into a call center



We could profile the "normal" wait time of customers by reporting the median (not the average, the distribution is skewed!) and the standard deviation

$\boldsymbol{\varphi} \in \mathbb{R}$
(wait time)

Output: none

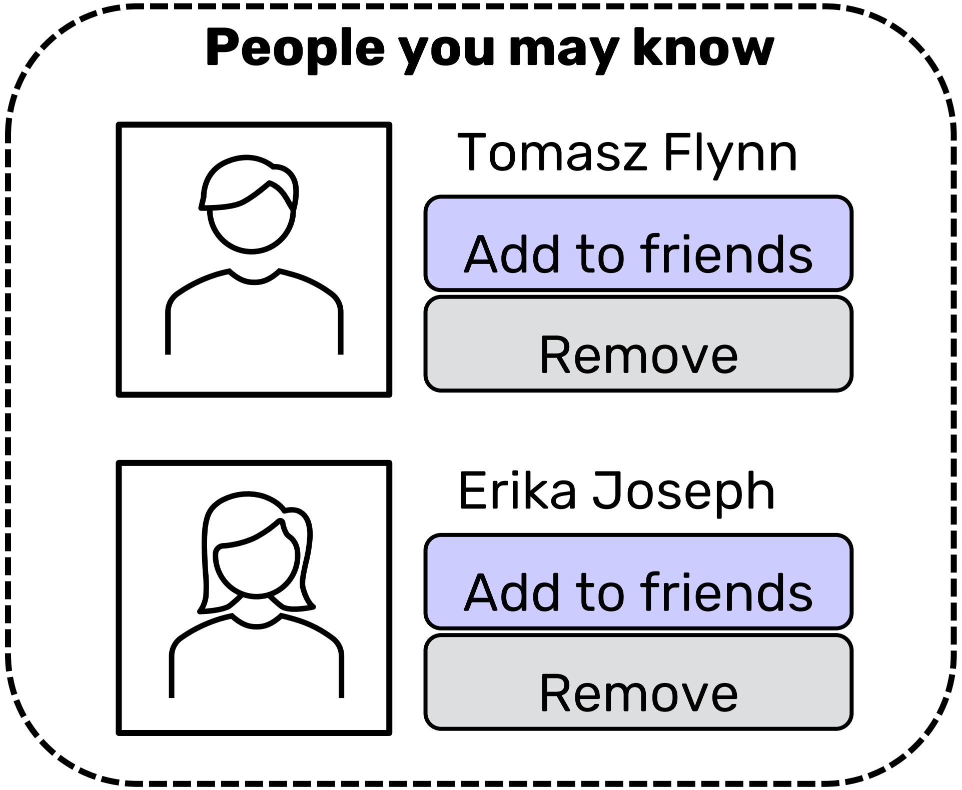
Picture taken from [1]

*: covered in this course : supervised : unsupervised

Data science tasks

- **Link prediction**: predict connections between entities in a network, usually by suggesting that a link should exist, and possibly also estimating the strength of the link

Example: ➢ Friend recommendations in social networks



*: covered in this course : supervised : unsupervised

Data science tasks

- **Dimensionality reduction***: take a large dataset (many **inputs** and, possibly, many **outputs**) and replace it with a smaller dataset, retaining as much information as possible

Example: ➢ Represent a collection of movies in a two-dimensional space (Netflix Prize)

Picture taken from [1]

Inputs:

- Movie title
- Year of release
- User id
- User rating
- Rating date

Output: none (in this example)

*: covered in this course : supervised : unsupervised

Data science tasks

- **Similarity matching**: find similar entities based on data known about them

Example: ➢ Recommendation systems

Inputs:

- Song titles
- Song genres
- Audio signals
- ⋮
- User ratings
- ⋮

Clustering is used for exploratory data analysis ("can we partition the data into different groups of similar entities?"), similarity matching has the specific goal of finding similar entities

Output: none (in this example)

*: covered in this course : supervised : unsupervised

Data science tasks vs algorithms

Data science task

(the problem that we are trying to solve, what we are trying to do)

Regression, classification, …

$\neq$

Algorithm (or method)

(how we solve it, a sequence of operations to follow)

Neural networks, KNN, K-means clustering, …

- Different data science tasks can be solved by the same algorithms

  K-means clustering can be used both for clustering and similarity matching

- Different algorithms can solve the same data science task

  A regression problem can be solved by the linear regression method, neural networks and KNN

In this course, we will study methods for solving different data science tasks

Syllabus

1. **Introduction to data science**
2. **Exploratory data analysis**
3. **Recap of statistics**
4. **Maximum likelihood estimation**
5. **Linear regression** (regression)
6. **Logistic regression** (classification)
7. **Bias-variance trade-off**
8. **Overfitting and regularization**
9. **Validation and cross-validation**

9. **Decision trees** (regression and classification)
10. **Neural networks** (regression, classification, dimensionality reduction…)
11. **Convolutional neural networks** (regression, classification, …)
12. **Clustering methods** (clustering)
13. **Principal component analysis** (dimensionality reduction)
14. **Output-error method for system identification** (regression)

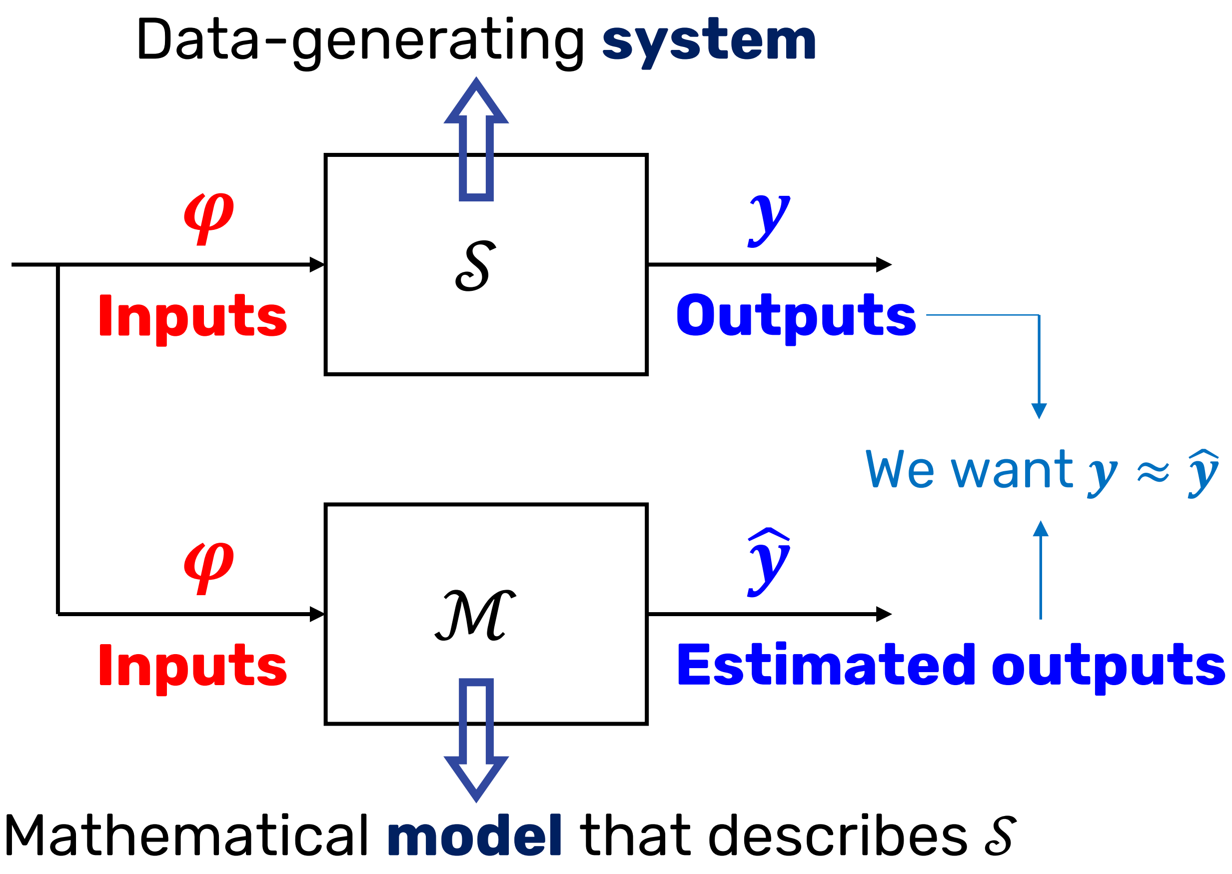
: supervised : unsupervised

Outline

1. Course introduction
2. Data science and the data-driven company
3. Data and its types
4. What we are going to do with data (supervised and unsupervised learning)
5. **Static and dynamical models in supervised learning**
6. From business problems to data science tasks
7. The data mining life cycle (CRISP-DM)

Models in supervised learning

Most supervised learning methods rely on mathematical **models** that describe the relationship between the **inputs** and the **outputs**

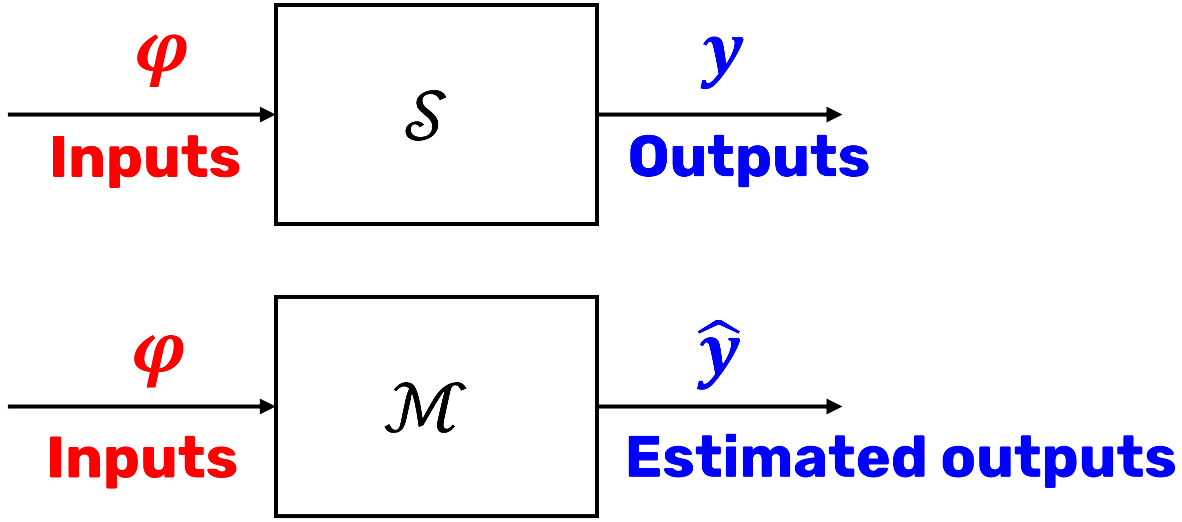


Supervised learning methods

estimate $\mathcal{M}$ from data

Models in supervised learning

We view both $\mathcal{S}$ and $\mathcal{M}$ as mathematical functions that map **inputs** (**features**) to **outputs** (**targets**)



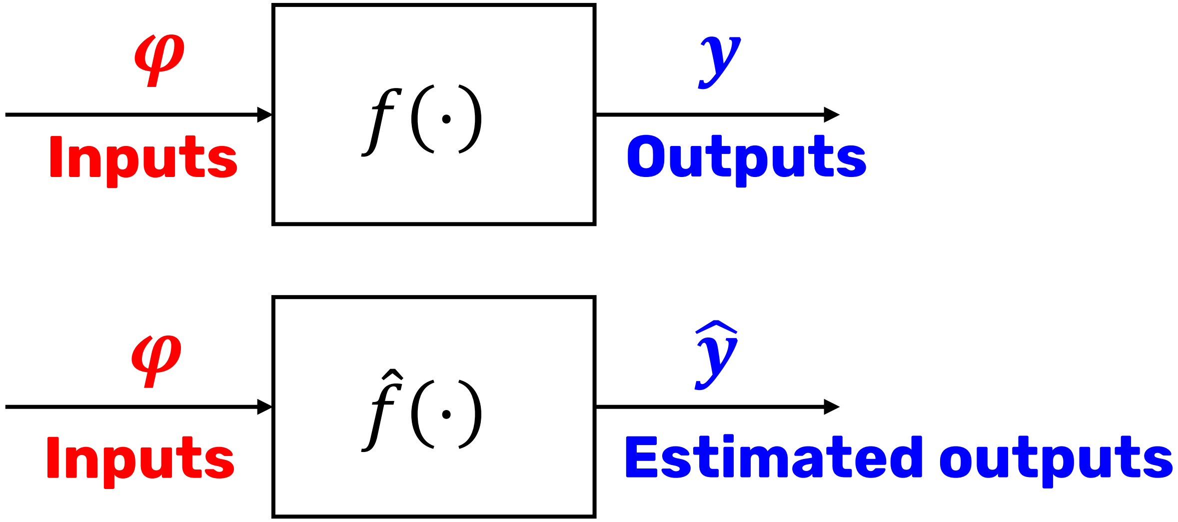
$$\equiv \qquad \boldsymbol{y} = f(\boldsymbol{\varphi})$$

$$\equiv \qquad \hat{\boldsymbol{y}} = \hat{f}(\boldsymbol{\varphi})$$

The goal of supervised learning methods is to learn a function $\hat{f}(\cdot)$ that approximates $f(\cdot)$ well **<u>on the whole domain</u>** of $\boldsymbol{\varphi}$

Models in supervised learning

We view both $\mathcal{S}$ and $\mathcal{M}$ as mathematical functions that map **inputs** (**features**) to **outputs** (**targets**)



$$\equiv \quad \boldsymbol{y} = f(\boldsymbol{\varphi})$$

$$\equiv \quad \hat{\boldsymbol{y}} = \hat{f}(\boldsymbol{\varphi})$$

The goal of supervised learning methods is to learn a function $\hat{f}(\cdot)$ that approximates $f(\cdot)$ well **on the whole domain** of $\boldsymbol{\varphi}$

Dataset notation

Before moving on, we introduce the following notation that we will use for any dataset

House area [feet^2]	# bedrooms	Price [k$]
⋮	⋮	⋮
		115
645	1	150
708	2	210
⋮	⋮	⋮

$\boldsymbol{\varphi}(i) = \begin{bmatrix} 523 \\ 1 \end{bmatrix}$

$y(i) = 115$

We refer to each row of the dataset as an **observation**

i-th observation (in this case it represents a house but, in general, it can be any entity)

$(\boldsymbol{\varphi}(i), y(i))$

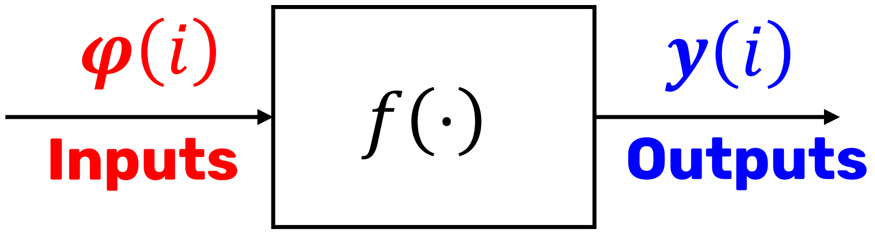
We denote the dataset as $\mathcal{D} = \{(\boldsymbol{\varphi}(1), y(1)), \ldots, (\boldsymbol{\varphi}(N), y(N))\}$

$= \{(\boldsymbol{\varphi}(i), y(i))\}_{i=1}^{N}$

(N observations in total)

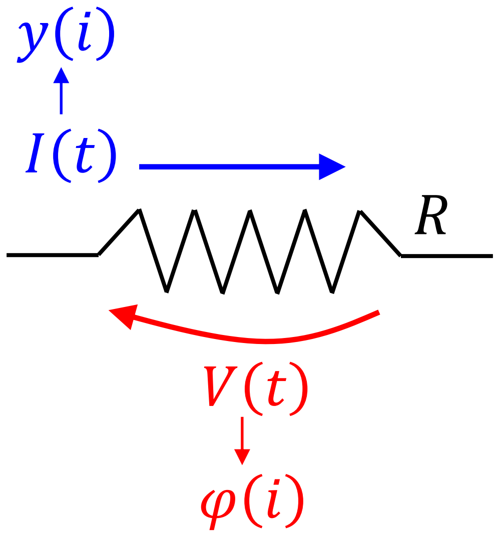
Static systems (and models)

A system whose **outputs** can be determined directly from the **inputs** is said to be a **static system** ("memoryless" system)



Example: Ohm's law

$$I(t) = \underbrace{\frac{V(t)}{R}}_{f(\varphi(i))}$$



The output $I(t)$ at time t only depends on the input $V(t)$ at the same time instant

We can view each voltage/current measurement by itself (i.e. as an observation $(\varphi(i), y(i))$ in its own right), we do not need to consider $V(t)$ and $I(t)$ as signals

"The time t can be omitted"

Static systems (and models)

Static systems need **not** describe **only** physics phenomena

House area [feet^2]	**# bedrooms**	**Price** [k$]
523	1	115
645	1	150
708	2	210
$\vdots$	$\vdots$	$\vdots$

$f(\cdot)$: mapping from house area and # bedrooms to price

Image	**Label**
	Cat
	Not cat
	Cat
	Not cat

$f(\cdot)$: mapping from image to label

Learning static systems

In the regression setting, the simplest model that can be used to describe static systems (**<u>but also dynamical systems</u>**!) is the **linear model**

$$\underset{1\times 1}{y(i)} = \theta_0 + \theta_1\varphi_1(i) + \cdots + \theta_{d-1}\varphi_{d-1}(i) + \epsilon(i) = \sum_{j=0}^{d-1} \theta_j\,\varphi_j(i) + \epsilon(i)$$

$$= \underset{1\times d}{\boldsymbol{\varphi}(i)^\top}\,\underset{d\times 1}{\boldsymbol{\theta}} + \underset{1\times 1}{\epsilon(i)}$$

i –th observation

- $\varphi_0 = 1$
- $\boldsymbol{\varphi}(i) = [\varphi_0 \quad \varphi_1(i) \quad \cdots \quad \varphi_{d-1}(i)]^\top \in \mathbb{R}^{d\times 1}$
- $\boldsymbol{\theta} = [\theta_0 \quad \theta_1 \quad \cdots \quad \theta_{d-1}]^\top \in \mathbb{R}^{d\times 1}$
- $y(i) \in \mathbb{R}$

- The vector $\boldsymbol{\theta}$ is called **parameters vector** → to be found by minimizing a cost function
- The vector $\boldsymbol{\varphi}(i)$ is called **features vector** for the i-th observation → attributes of entities
- The quantity $\epsilon(i)$ is the **error** due to not perfect explanation of $y(i)$ using $\boldsymbol{\varphi}(i)$

Learning static systems

To **“learn”** means to **estimate the values** of the parameters in $\boldsymbol{\theta} = [\theta_0 \quad \theta_1 \quad \cdots \quad \theta_{d-1}]^\top$

Key idea: find the values of $\boldsymbol{\theta}$ that **minimize** a “cost” (or “loss”), i.e. an “error” or “something bad” → it is good to minimize something bad

- This is achieved through **optimization**

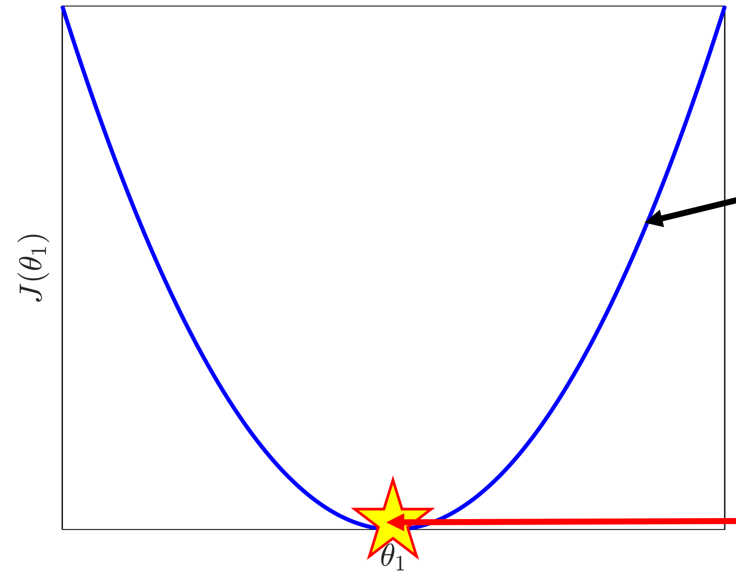
A typical cost in the regression setting is the following

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}(y(i) - \boldsymbol{\varphi}(i)^\top\boldsymbol{\theta})^2 = \frac{1}{N}\sum_{i=1}^{N}\epsilon(i)^2$$

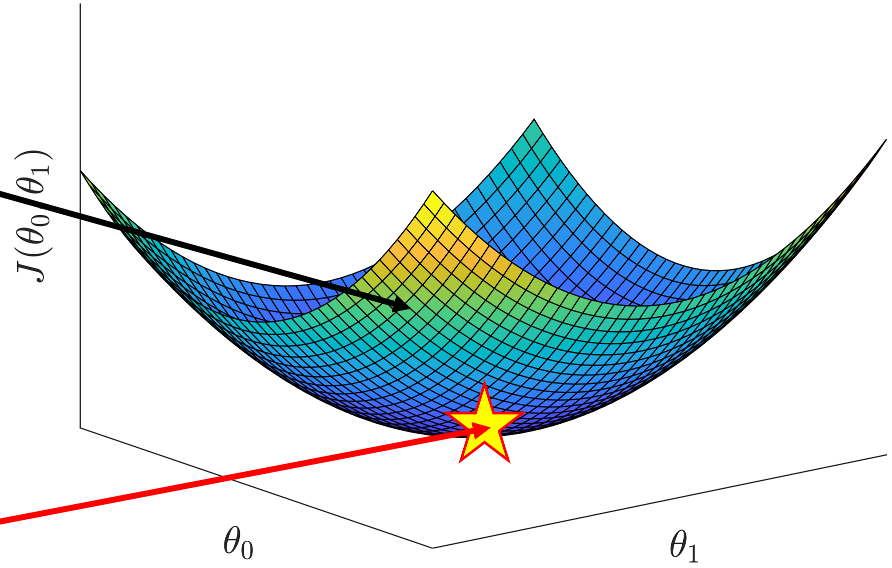
With this cost, we are **minimizing the sum of the squared errors** between the observed outputs (i.e. those reported in our dataset) and the outputs estimated by the linear model

Learning static systems

Scalar (single) parameter θ



Multiple parameters θ



Cost function

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \epsilon(i)^2$$

Minimizer of the cost function:

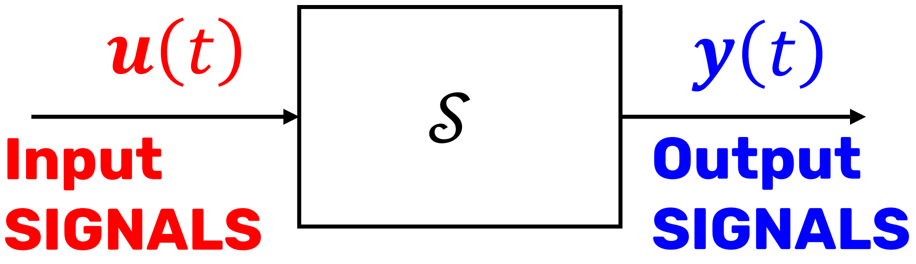
$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

This rationale is followed by the **linear regression method**

$$\hat{y}(i) = \hat{f}\big(\boldsymbol{\varphi}(i)\big) = \boldsymbol{\varphi}(i)^\top \widehat{\boldsymbol{\theta}}$$

Dynamical systems (and models)

A system whose **outputs** (at a certain time instant) cannot be determined directly from the **inputs** (at the same time instant) is said to be a **dynamical system**



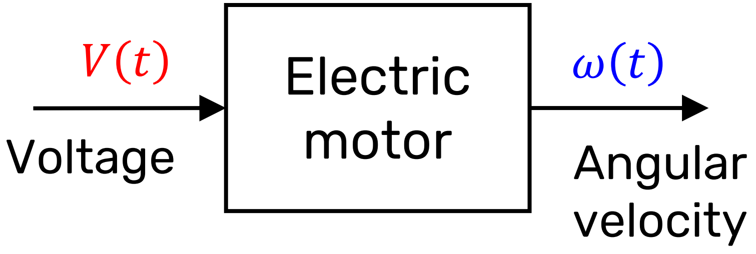
Dynamical models are mathematical models that describe the future evolution of the variables involved as a **function of their past trend**

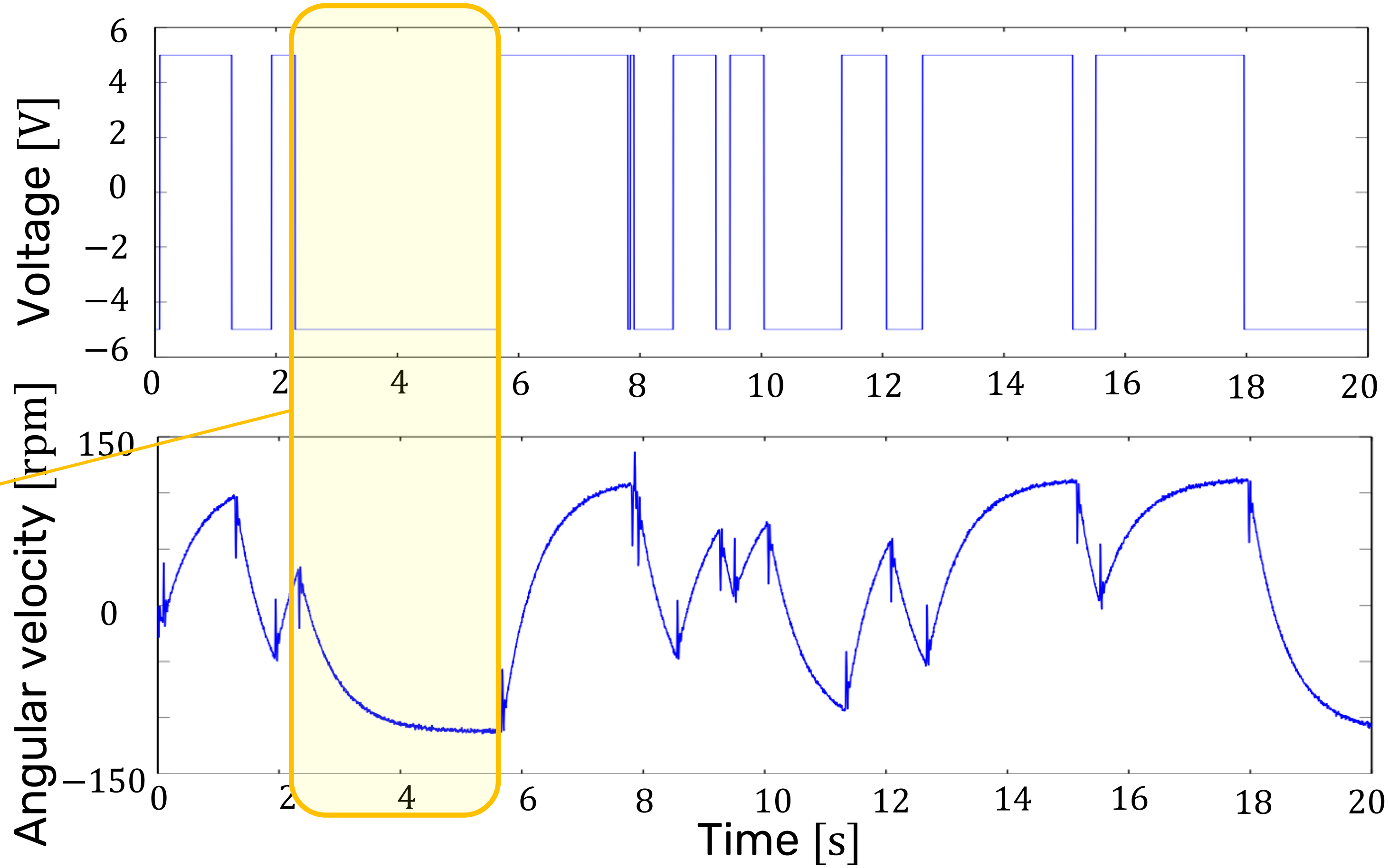
Dynamical systems usually involve the **time**: the **outputs** $y(t)$ at a certain time t **depend on the outputs at previous times**

This dependency on the past endows the model with a **"memory"** (i.e. the dynamics)

Dynamical systems (and models)

This dependency on the past endows the model with a **"memory"** (i.e. the dynamics)





We are dealing with a dynamical system because, although **the input is constant, the output keeps evolving**

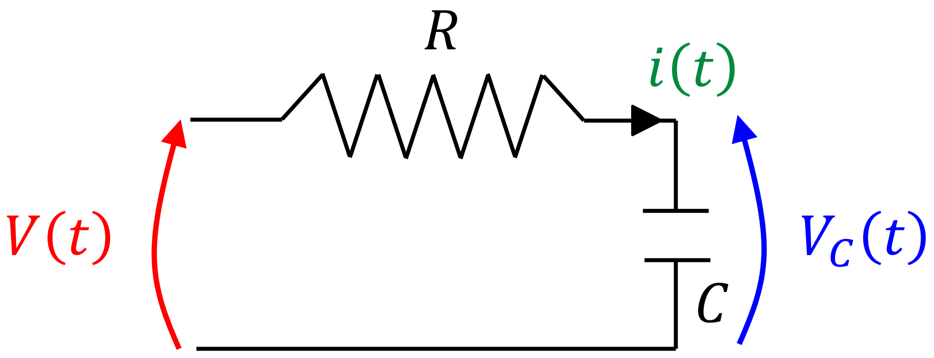
Dynamical systems (and models)

Dynamical systems can be defined in **continuous-time** or in **discrete-time**

Physics phenomena are (inherently) continuous

- In this case, the system is described by **differential equations**

Example: resistor-capacitor circuit (continuous-time)



$$\boxed{i(t) = C\dot{V}_C(t)} \qquad \dot{V}_C(t) = \frac{dV_C(t)}{dt}$$

$$V(t) = R \cdot \boxed{i(t)} + V_C(t)$$

$$\dot{V}_C(t) + \frac{1}{RC}V_C(t) = \frac{1}{RC}V(t)$$

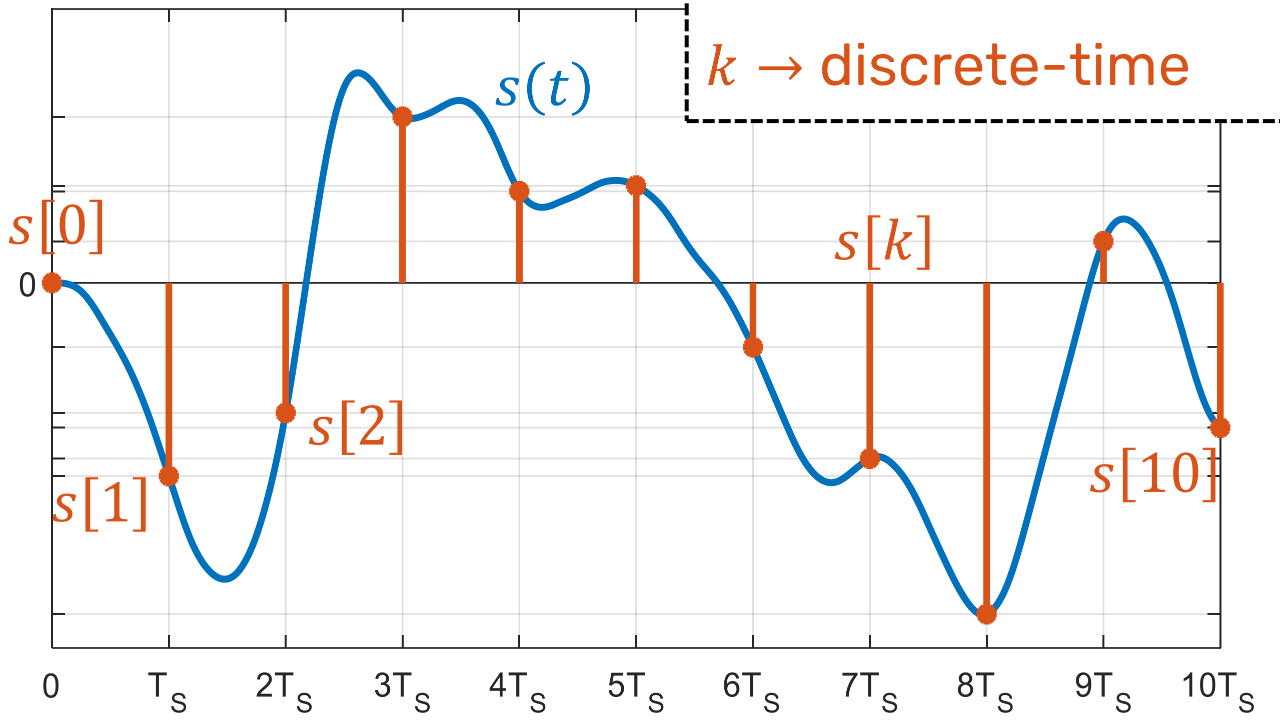
Dynamical systems (and models)

However, computers can only manage a **finite amount of data**. Thus, signals $s(t)$ should be **sampled** at a sampling time T_s so that we can store a finite amount of data corresponding to the time instants $kT_s,\ k = 1, \dots, N$, i.e.

t → continuous-time

k → discrete-time

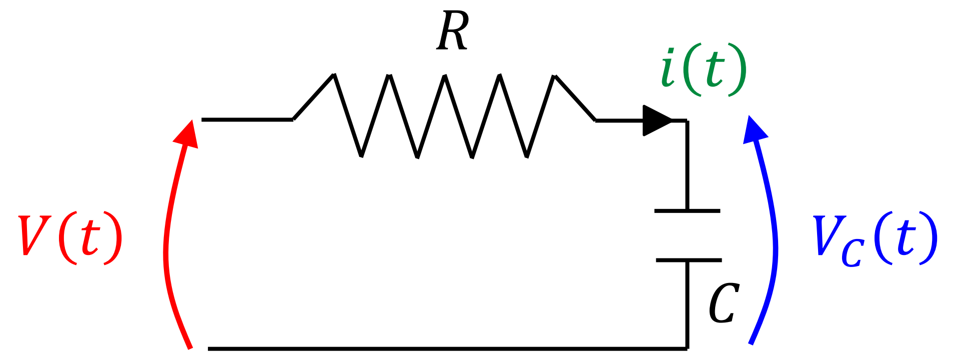
$$s(0), s(T_s), s(2T_s), s(3T_s), \dots$$



In the following, for discrete-time systems, we will use the notation $s[k]$ with the meaning of $s(kT_s)$ (i.e. the measurement of $s(\cdot)$ at the time kT_s)

Dynamical systems (and models)

Example: resistor-capacitor circuit (continuous-time → discrete-time)



$$\dot{V}_C(t) + \frac{1}{RC} V_C(t) = \frac{1}{RC} V(t)$$

Numerical differentiation

$$\dot{V}_C(kT_s) \approx \frac{V_C\big((k+1)T_s\big) - V_C(kT_s\,)}{T_s}$$

$$t = kT_s$$

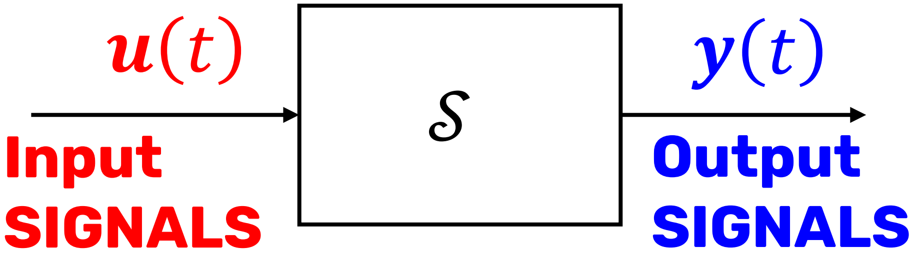
$$s[k] = s(kT_s)$$

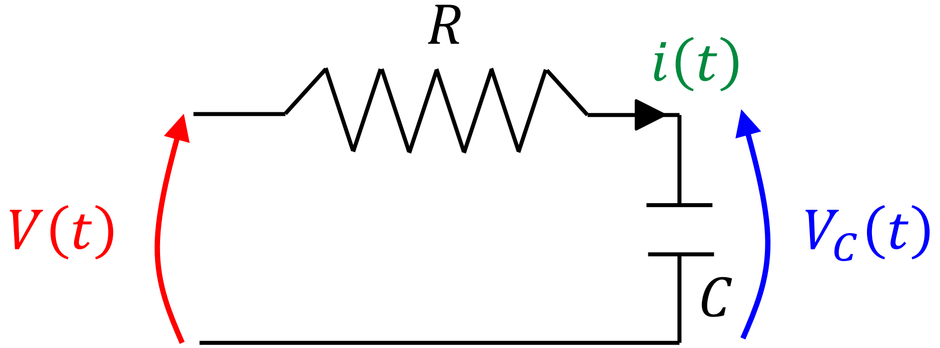
$$\frac{V_C[k+1] - V_C[k]}{T_s} + \frac{1}{RC} V_C[k] = \frac{1}{RC} V[k]$$

Shift back by 1 step and re-organize equation

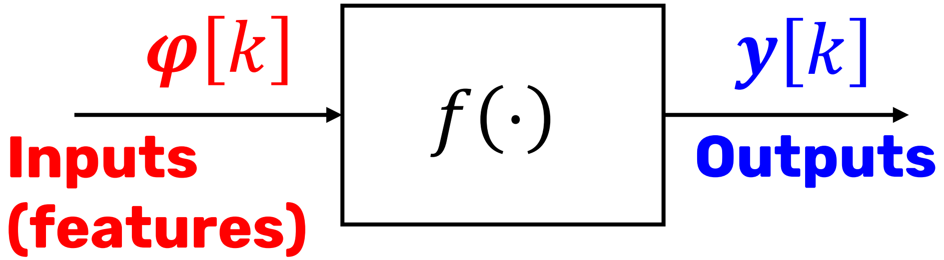
$$V_C[k] = \left(1 - \frac{T_s}{RC}\right) V_C[k-1] + \frac{T_s}{RC} V[k-1]$$

From signals to feature vectors





$$\dot{V}_C(t) + \frac{1}{RC} V_C(t) = \frac{1}{RC} V(t)$$



$$V_C[k] = \left(1 - \frac{T_s}{RC}\right) V_C[k-1] + \frac{T_s}{RC} V[k-1]$$

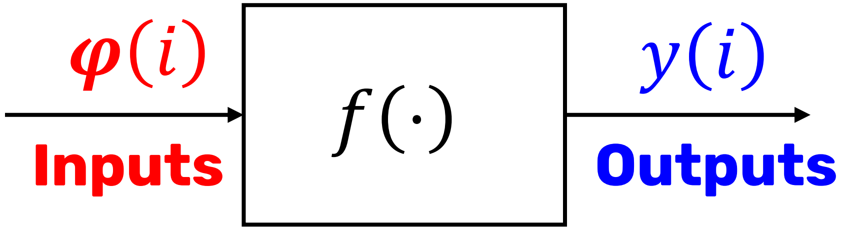
$$\Vert$$

$$y[k] = f(\boldsymbol{\varphi}[k]) = \boldsymbol{\varphi}[k]^\top \boldsymbol{\theta}$$

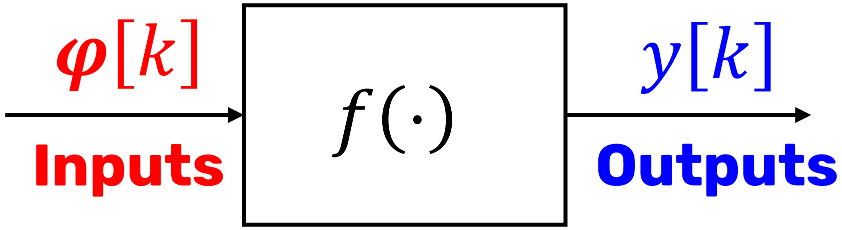
- $\boldsymbol{\varphi}[k] = [V_C[k-1] \quad V[k-1]]^\top$
- $\boldsymbol{\theta} = \left[1 - \frac{T_s}{RC} \quad \frac{T_s}{RC}\right]^\top$
- $y[k] = V_c[k]$

Static vs dynamical systems

Static systems



Dynamical systems



- For **static systems**, we will index the observations with the index i
- For **dynamical systems**, we will index the observations with the index k

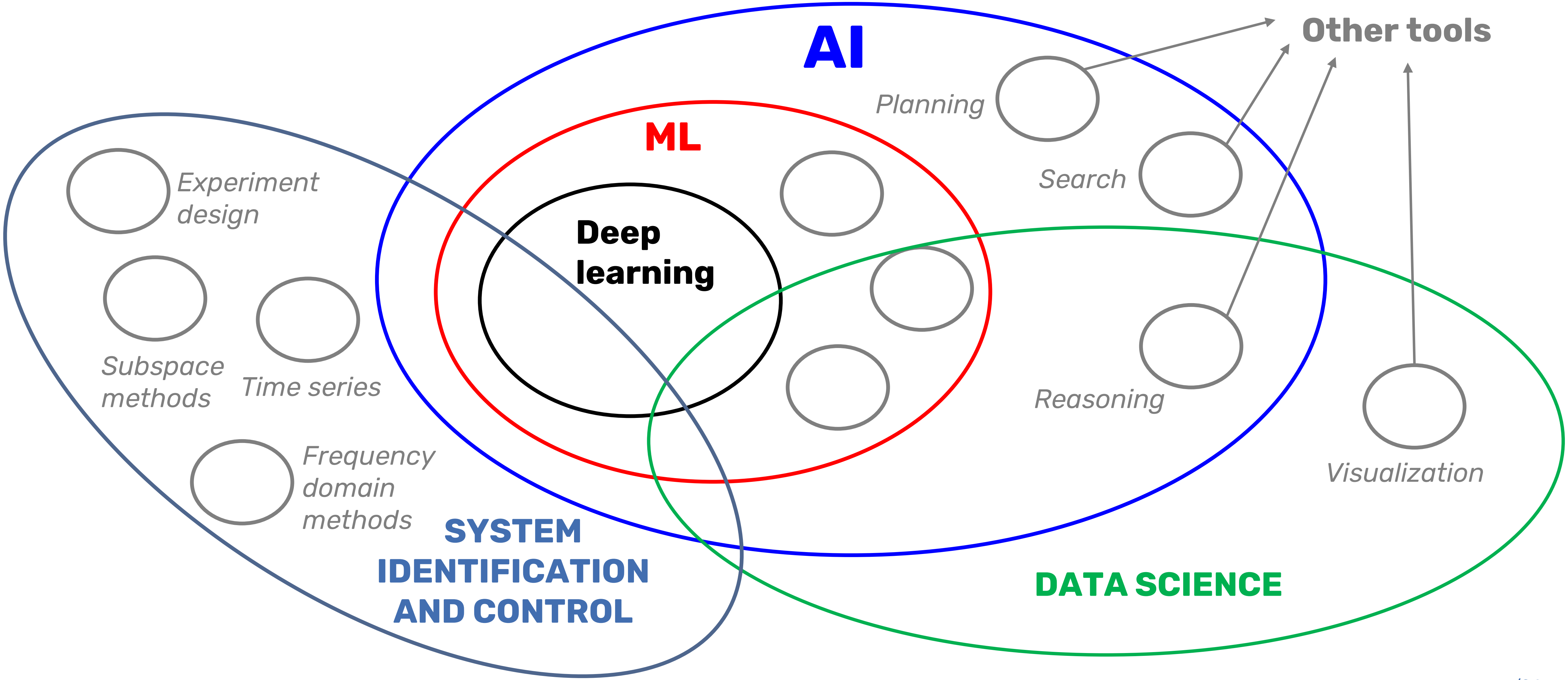
k can be interpreted as the k-th sampling step

In either case, our aim will be **to learn $f(\cdot)$ from data**

- In the static case, we talk about (model) **"learning"**
- In the dynamical case, we talk about (system) **"identification"**

Both are supervised learning tasks!

Machine Learning (ML), Artificial Intelligence (AI), Data Science and System Identification



Why do we need models?

All in all, we need a model to **better understand the phenomena** that are of our interest.

<u>Models are useful for</u>:

- **Decision-making:** suppose that we are testing a new vaccine. We have two groups of people. We give the vaccine to the first group (test group) and a placebo to the second one (control group). Then, we measure some variables from the patients. How can we determine if the vaccine was effective or not?

- **Communication:** a model allows to communicate to third parties the main insights and results of your analysis

Why do we need models?

All in all, we need a model to **better understand the phenomena** that are of our interest.

Models are useful for:

- **Prediction:** forecast the values that the output variables will assume based on the values assumed by the inputs variables and on which we have no data about

House area [feet^2]	# bedrooms	Price [k$]
523	1	115
645	1	150
708	2	210
⋮	⋮	⋮

How much does a $600\ \text{feet}^2$ house with 2 bedrooms cost?

Why do we need models?

All in all, we need a model to **better understand the phenomena** that are of our interest.

Models are useful for:

- **Inference**: understand how changes in the inputs affect the outputs

House area [feet^2]	**# bedrooms**	**Price** [k$]
523	1	115
645	1	150
708	2	210
⋮	⋮	⋮

- Does increasing house area increase the house price (and by how much)?
- Is # bedrooms actually associated with the price of a house?

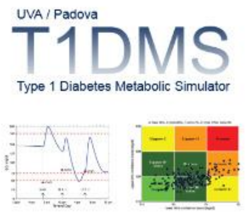
Prediction vs inference: prediction is not necessarily concerned with the structure of the model $\hat{f}(\cdot)$ and its complexity ($\hat{f}(\cdot)$ can be seen as a black-box) while inference uses the model to understand the relationship between each input and each output

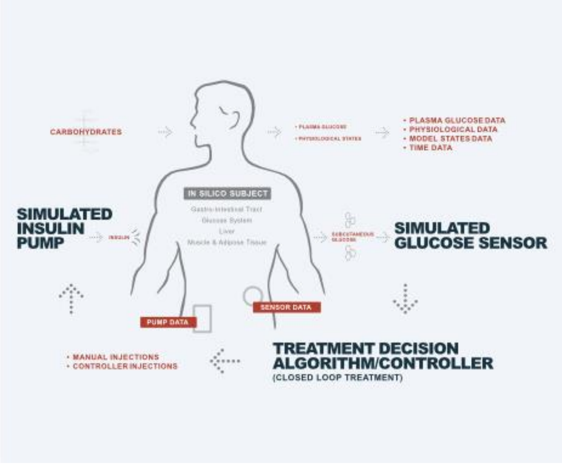
Why do we need models?

All in all, we need a model to **better understand the phenomena** that are of our interest.

Models are useful for:

- **Simulation:** we can simulate, with a computer, the response (outputs) of a model due to certain inputs. By looking at the model's response, we can get a better grasp of the modeled system



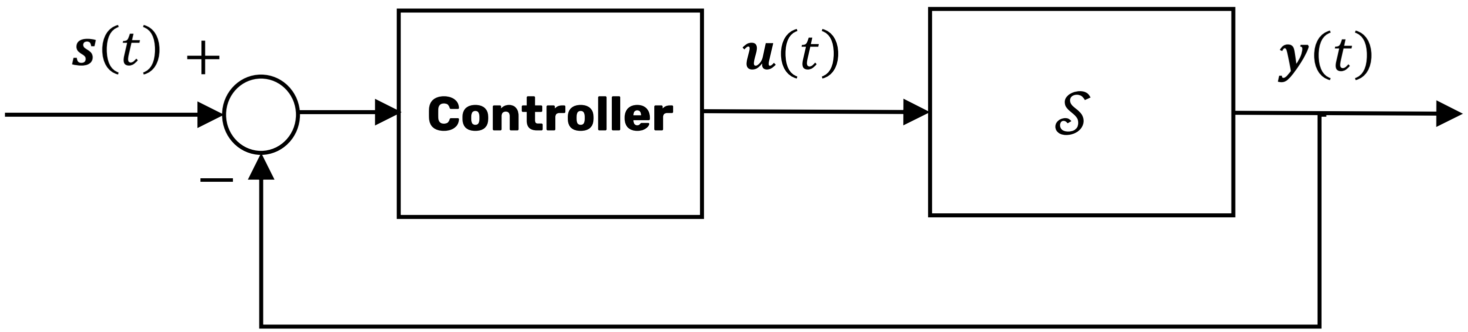


Why do we need models?

All in all, we need a model to **better understand the phenomena** that are of our interest.

Models are useful for:

- **Control:** often, in control engineering, we need a model of a system to design a controller that limits the deviation of the controlled variables $\boldsymbol{y}(t)$ from the reference variables $\boldsymbol{s}(t)$ (setpoints)

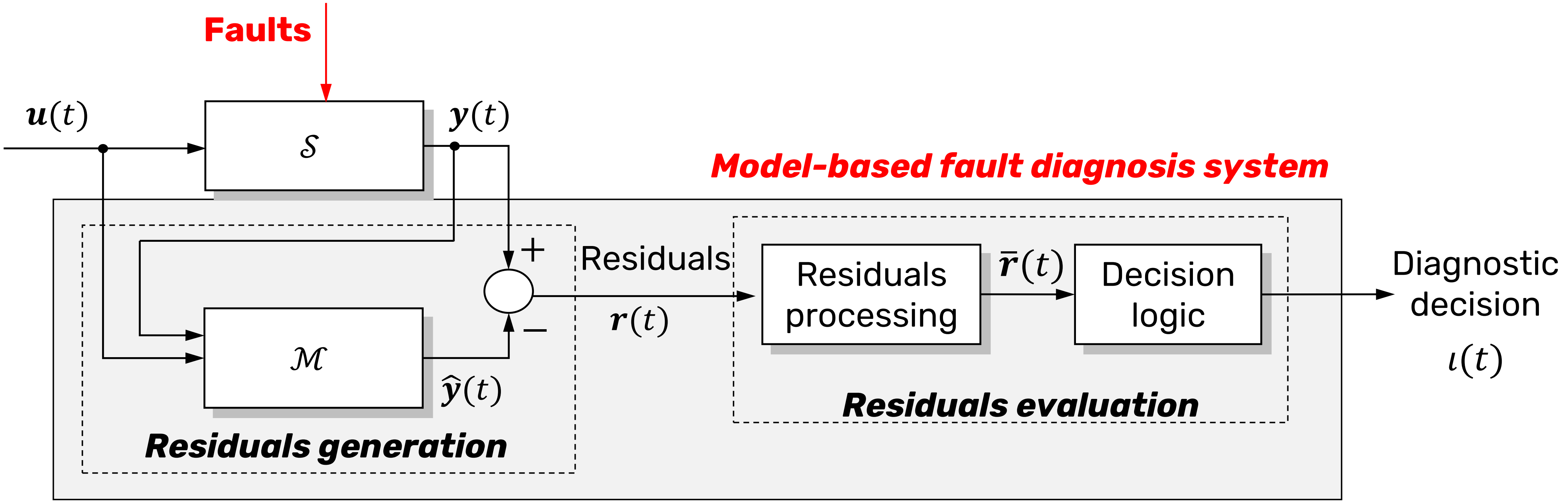


Why do we need models?

All in all, we need a model to **better understand the phenomena** that are of our interest.

Models are useful for:

- **Fault diagnosis:** we can check the presence of faults by comparing signals that come from the real system with those simulated by the estimated model



Outline

Business problems as data science tasks

Each data-driven project is **unique**. First and foremost, **decompose** the business problem into data science subtasks that can be solved by **existing methods**

Business problem

decompose

Data science (sub)task(s)

- Regression
- Classification
- Causal modeling
- Clustering
- Co-occurrence grouping
- Profiling
- Link prediction
- Dimensionality reduction
- Similarity matching

solve

Machine learning engineers focus on these aspects

Algorithm(s)/ method(s)

analyze

Analyze the results

(to derive insights and drive business-related decisions)

Data scientists focus on these aspects

Business problems as data science tasks

- Spam e-mail detection system Classification
- Credit approval Classification
- Fraud detection Profiling
- Recognize objects in images Classification
- Find the relationship between house prices and house sizes Regression
- Predict the stock market Regression

- Market segmentation Clustering
- Market basket analysis Co-occurrence grouping
- Language models (word2vec) Similarity matching
- Social network analysis Link prediction
- Low-order data representations Dimensionality reduction
- Movies recommendation Similarity matching
- A/B testing Causal modeling