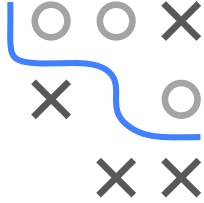
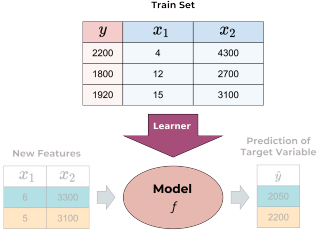
Introduction to Machine Learning



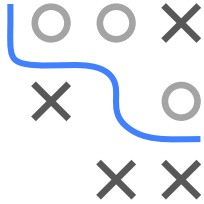
ML-Basics
Learner



Learning goals

- Understand that a supervised learner fits models automatically from training data
-

SUPERVISED LEARNING EXAMPLE



Imagine we want to investigate how working conditions affect productivity of employees.

- It is a **regression** task since the target *productivity* is continuous.
- We collect data about worked minutes per week (*productivity*), how many people work in the same office as the employee in question, and the employee's salary.

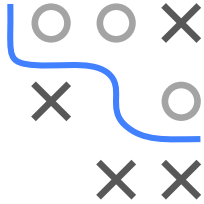
Features x		Target y
People in Office (Feature 1) x_1	Salary (Feature 2) x_2	Worked Minutes Week (Target Variable)
4	4300 €	2220
12	2700 €	1800
5	3100 €	1920

$n = 3$

$p = 2$

$x_1^{(2)}$ $x_2^{(1)}$ $y^{(3)}$

SUPERVISED LEARNING EXAMPLE



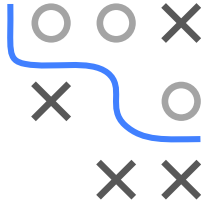
How could we construct a model from these data?

We could investigate the data manually and come up with a simple, hand-crafted rule such as:

- The baseline productivity of an employee with salary 3000 and 7 people in the office is 1850 minutes
- A decrease of 1 person in the office increases productivity by 30
- An increase of the salary by 100 increases productivity by 10

=> Obviously, this is neither feasible nor leads to a good model

IDEA OF SUPERVISED LEARNING

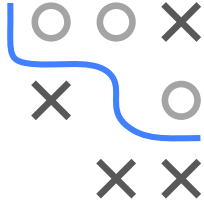


Goal: Automatically identify the fundamental functional relation in the data that maps an object's features to the target.

- **Supervised** learning means we make use of *labeled* data for which we observed the outcome.
- We use the labeled data to learn a model f.
- Ultimately, we use our model to compute predictions for **new** data whose target values are unknown.

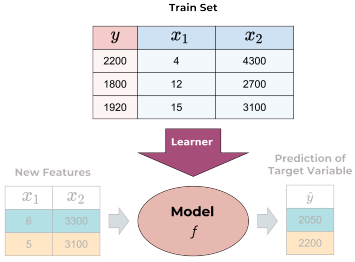
	x_1	x_2	Functional Relationship	y
Already seen Data	4	4300 €		2200
	12	2700 €		1800
	15	3100 €	f	1920
New Data	6	3300 €		???
	5	3100 €		???

LEARNER DEFINITION

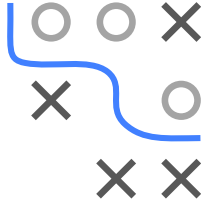


- The algorithm for finding our f is called **learner**. It is also called **learning algorithm** or **inducer**.
- We prescribe a certain hypothesis space, the learner is our means of picking the best element from that space for our data set.
- Formally, it maps training data $\mathcal{D} \in \mathbb{D}$ (plus a vector of **hyperparameter** control settings $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$) to a model:

$$\mathcal{I} : \mathbb{D} \times \boldsymbol{\Lambda} \rightarrow \mathcal{H}$$



LEARNER DEFINITION



As pseudo-code template it would work like this:

- Learner has a defined model space of parametrized functions $\mathcal{H}$.
- User passes data set $\mathcal{D}_{\text{train}}$ and control settings $\boldsymbol{\lambda}$.
- Learner sets parameters so that model matches data best.
- Optimal parameters $\hat{\boldsymbol{\theta}}$ or function $\hat{f}$ is returned for later usage.