Machine Learning for Networking
ML4N

Luca Vassio
Gabriele Ciravegna
Zhihao Wang
Tailai Song

Recap – key concepts

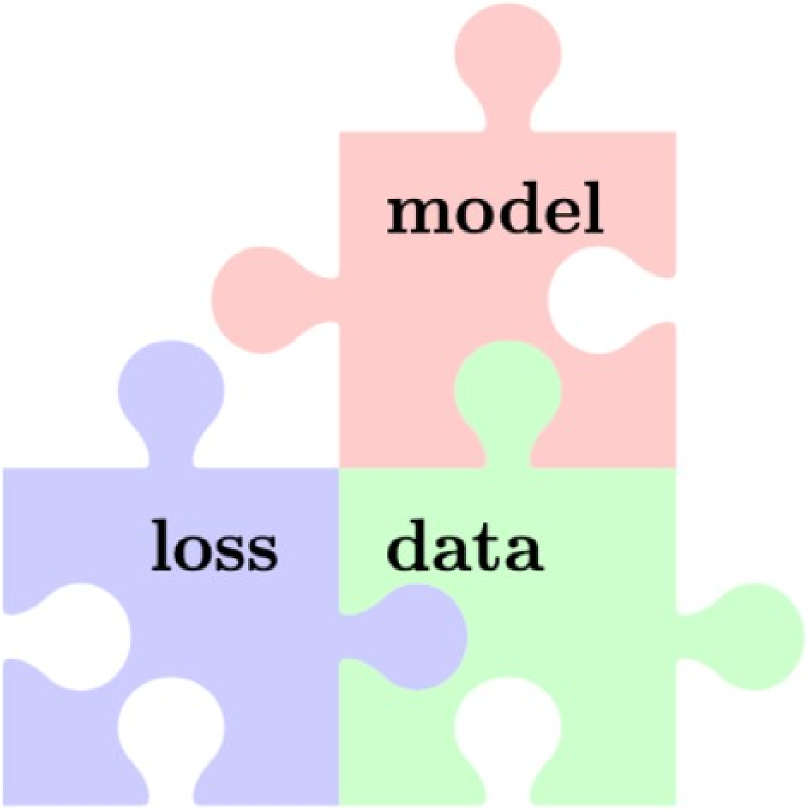
- Random variables and their samples
- Correlations of variables
- Similarity and distance between samples
- Data preprocessing
  - Feature normalization
  - Feature learning/dimensionality reduction (PCA,..)
  - ...

Learning goal

Become familiar with concepts of

- **Data** points
- **Model** or **hypothesis space**
- **Loss function**

The three components of ML

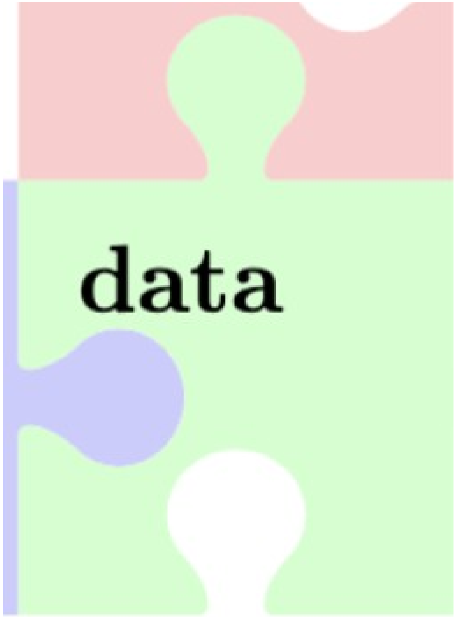


Data

Data

“For a lot of problems, we should shift our mindset toward **not just improving the code** but in a more systematic way of **improving the data**”

Andrew Ng

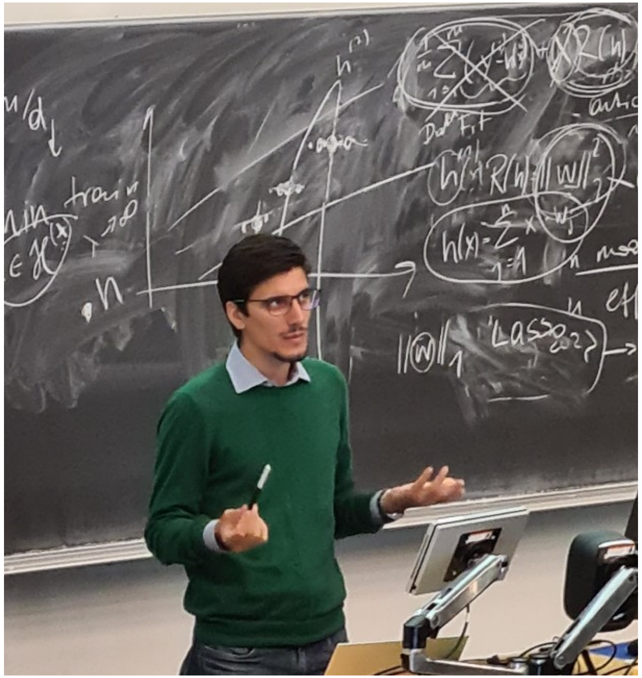


Data

- **Data** = set/collection of data points
- **Data points** = objects, records, cases, samples, entities, or instances
- Data points carry information = **features**

Data point

- A person

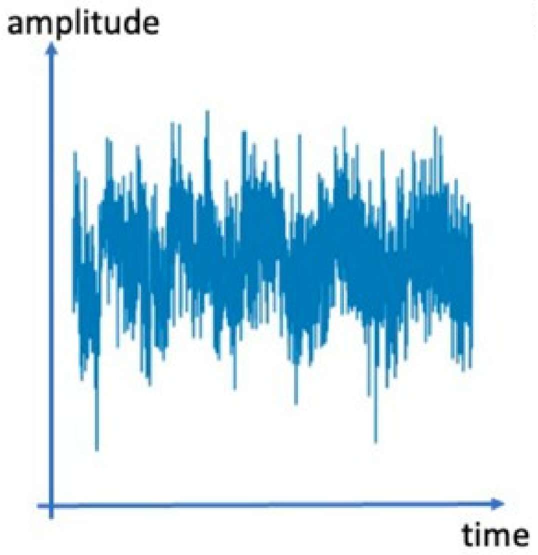


Data point

- An image

Data point

- A signal



Data point

- A server

Data point

- An ssh attack

```
root@kali:~# nmap -sV -p22 192.168.1.103
Starting Nmap 7.80 ( https://nmap.org ) at 2019-12-23 09:51 EST
Nmap scan report for literally.vulnerable (192.168.1.103)
Host is up (0.00060s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
MAC Address: 00:0C:29:E3:D3:A5 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```


https://www.hackingarticles.in/ssh-penetration-testing-port-22/

Data

Data points carry information

- **Features**
  - Low-level properties
  - Often easy to measure/compute
- **Labels**
  - High-level quantity of interest
  - Often difficult to measure/determine

- Distinction might be blurry sometimes

Data

features (pixel RGB values)

"Cat"	"Dog"	"Cat"	?

label

https://commons.wikimedia.org/

Features

- We mainly use **numeric features**
- Stack features into feature vector
- $x_1,\ldots,x_n$ to characterize a datapoint

Features of an image

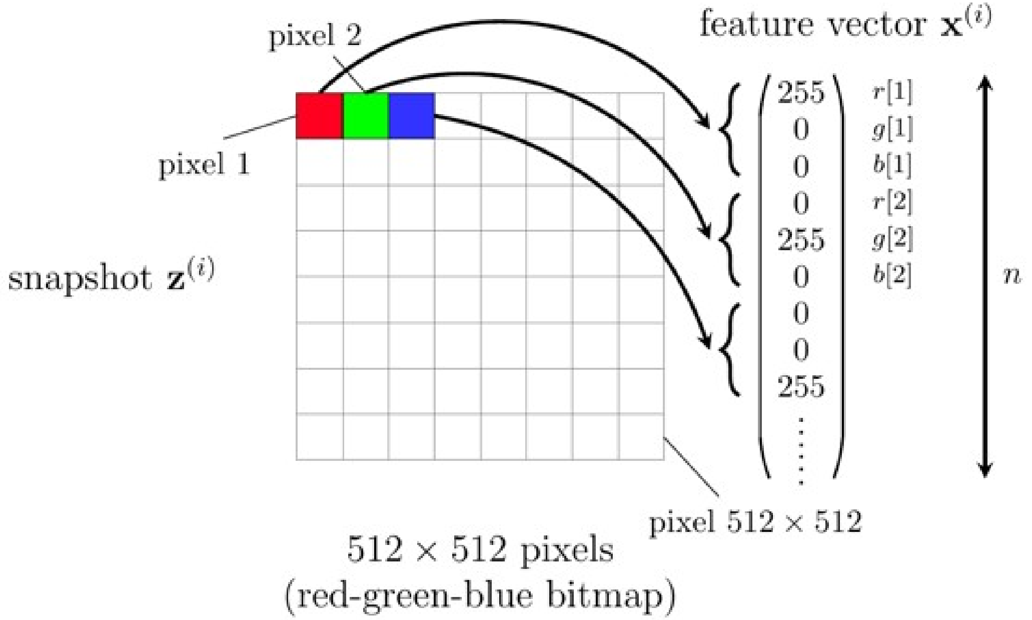


Figure 2.5: If the snapshot $\mathbf{z}^{(i)}$ is stored as a 512×512 RGB bitmap, we could use as features $\mathbf{x}^{(i)} \in \mathbb{R}^n$ the red-, green- and blue component of each pixel in the snapshot. The length of the feature vector would then be $n = 3 \times 512 \times 512 \approx 786000$.

Label

- Label is **design choice**
- YOU choose what to consider as **label** of a data point
- Also called output variable, target, response variable
- By choosing/defining label you **define the ML problem or learning task**

Label

- Label can be categorical or numerical
  - **Categorical: classification** task
    - If only 2 categories: **binary** classification
    - If more than 2 categories: **multi-class** classification
  - **Numerical: regression** task
  - If a data point have more than one label: **multi-label** problem
  - If a data point has more than one type of labels: **multi-task** learning

Multi-label Classification

= 1 or 0 if **car** present or not

= 1 or 0 if **person** present or not

= 1 or 0 if **tree** present or not

= 1 or 0 if a **cat** present or not

Raw data

- Tabular data

Year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	-1
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7

Points, features, label

features

label

data point

year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	-1
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7

data points, features and labels are design choices!

Number of points and features

number of features n

Year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	-1
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7
2020	1	18	00:00	19,6	65	-4,6	-7,3	-4,2
2020	1	19	00:00	0,7	81	-4,4	-8,8	-2,7
2020	1	20	00:00	2,8	79	-1,8	-10,5	1,2

number of data points, sample size m

Data

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}.$$

Data points characterized by features and label

- **Features** low-level properties

$$\mathbf{X} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\right)^T = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \ldots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \ldots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \ldots & x_n^{(m)} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

- **Labels** high-level properties (quantity of interest)

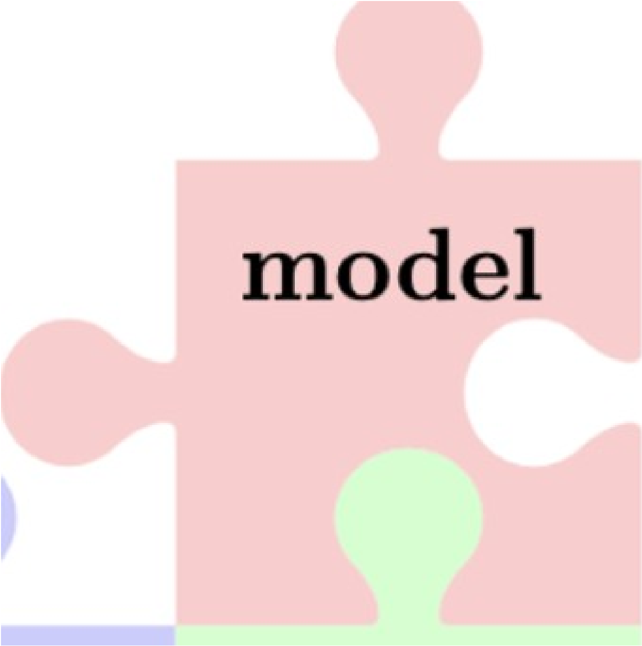
$$\mathbf{y} = \left(y_1, y_2 \ldots, y_m \right)^T \in \mathbb{R}^m$$

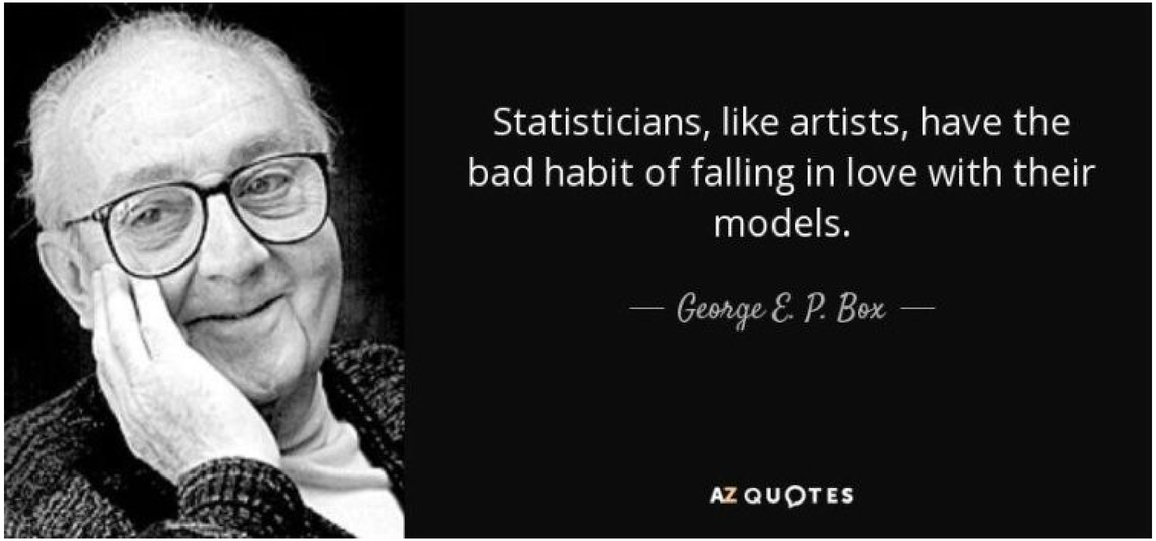
Number of features

- Use only **most relevant features but not fewer**
- **Missing relevant** features **bad for accuracy**
- **Using irrelevant** features **wastes computation** and might result in **overfitting**

Model

Model

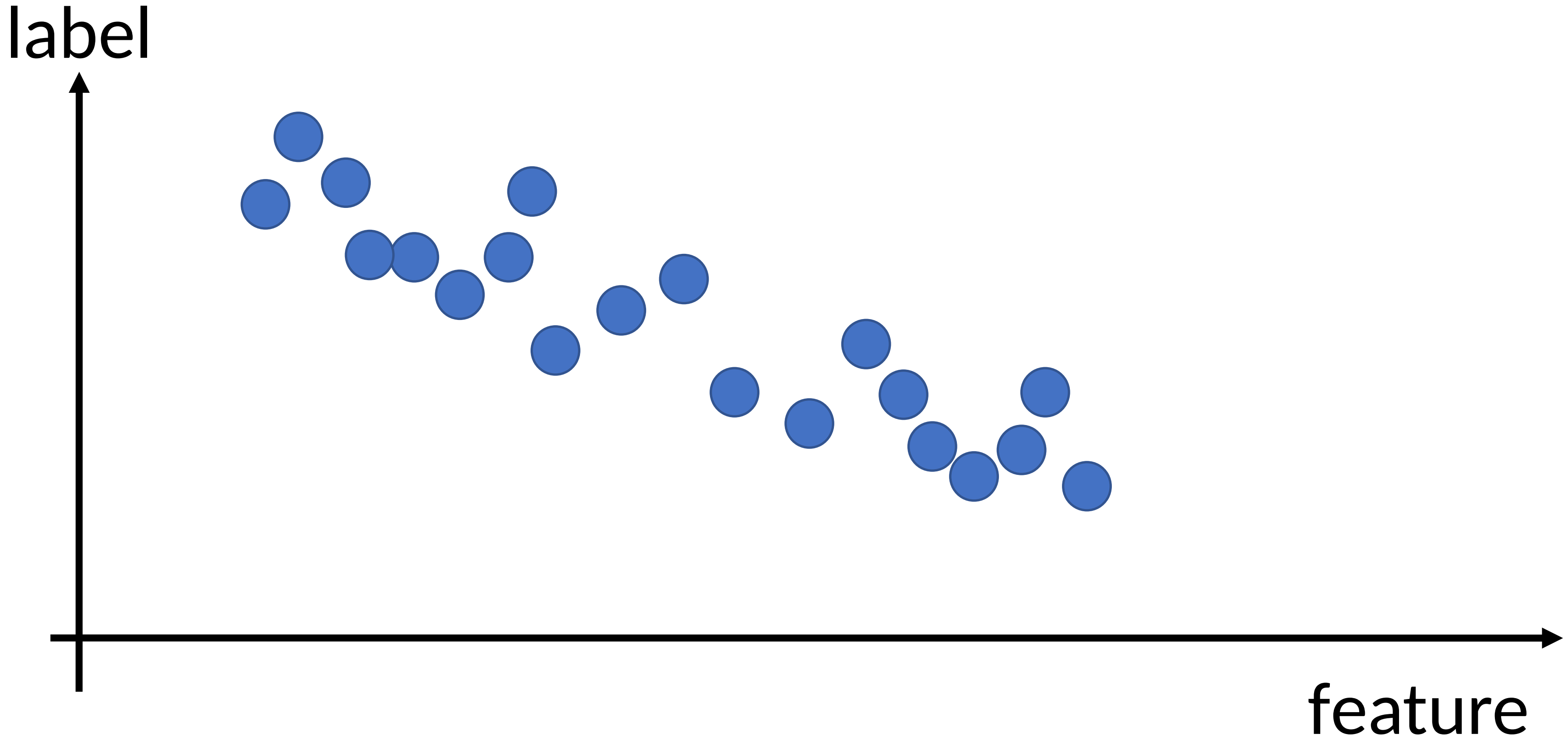

model


Statisticians, like artists, have the bad habit of falling in love with their models.
— George E. P. Box —
AZ QUOTES

Machine learning

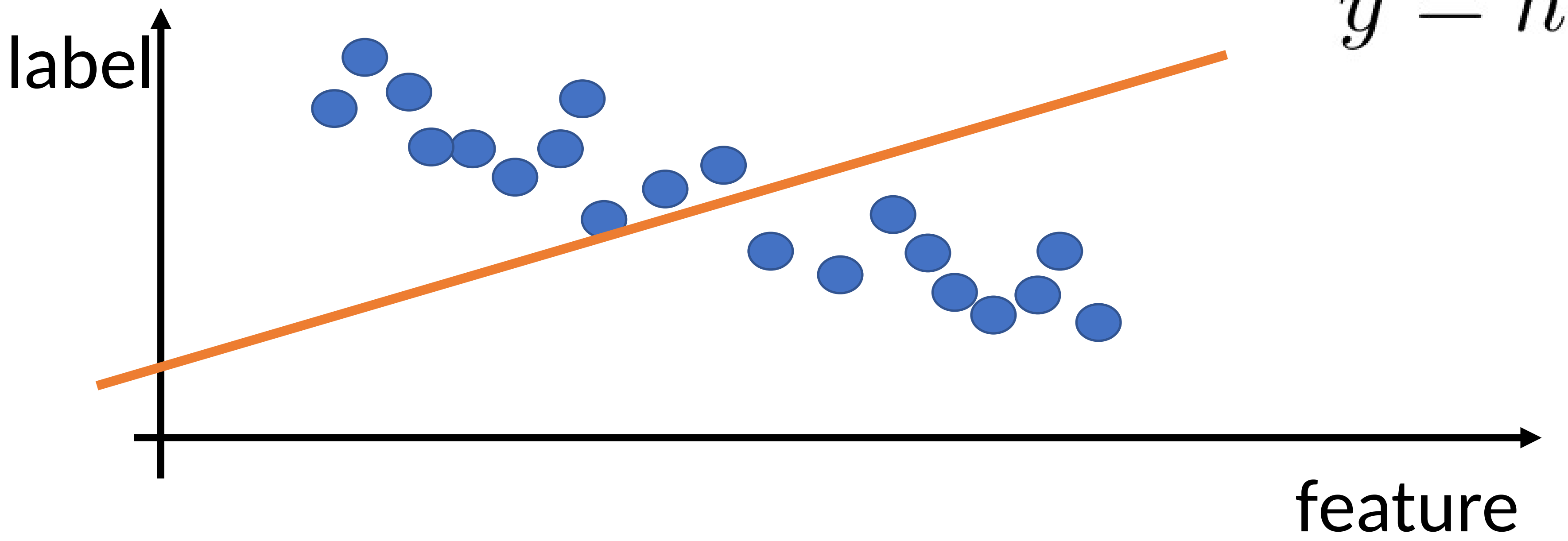
- Learn to **predict** the **label** y of a data point **from its features x**
- Learn a **hypothesis** $h \in \mathcal{H}$ such that h(**x**) ≈ y $h : \mathcal{X} \to \mathcal{Y}$

Data

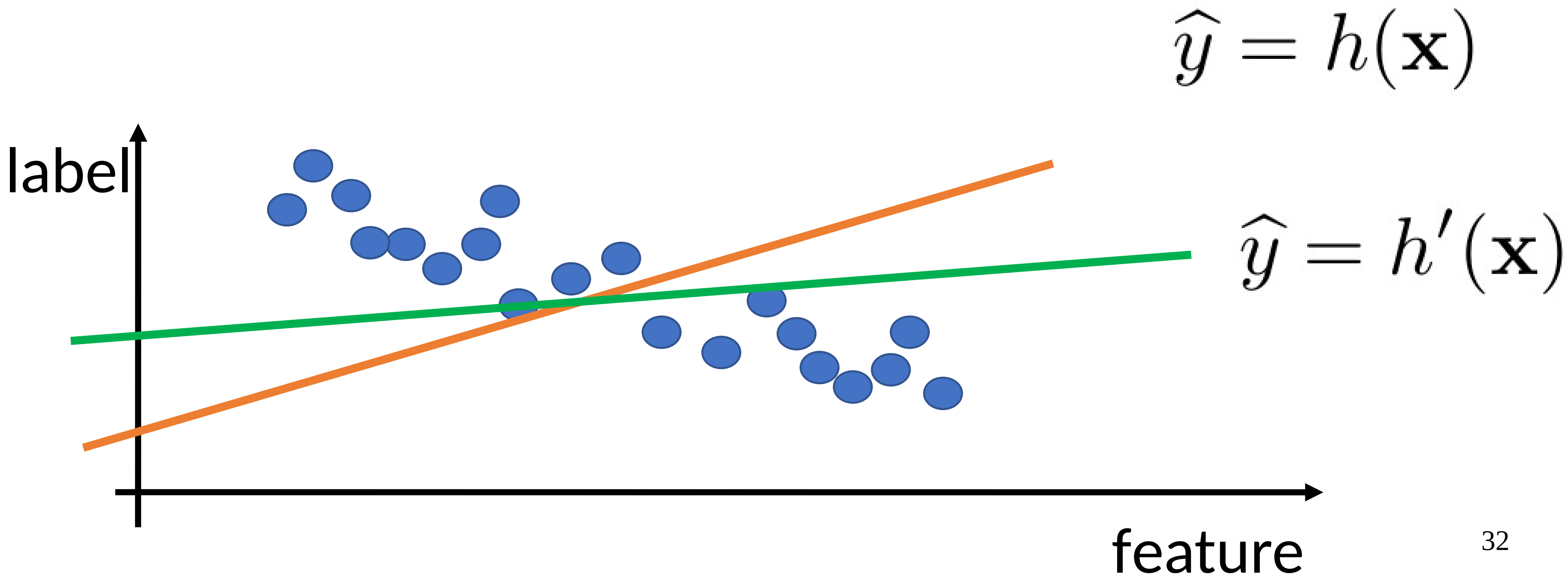


How to predict?

$$\widehat{y} = h(\mathbf{x})$$

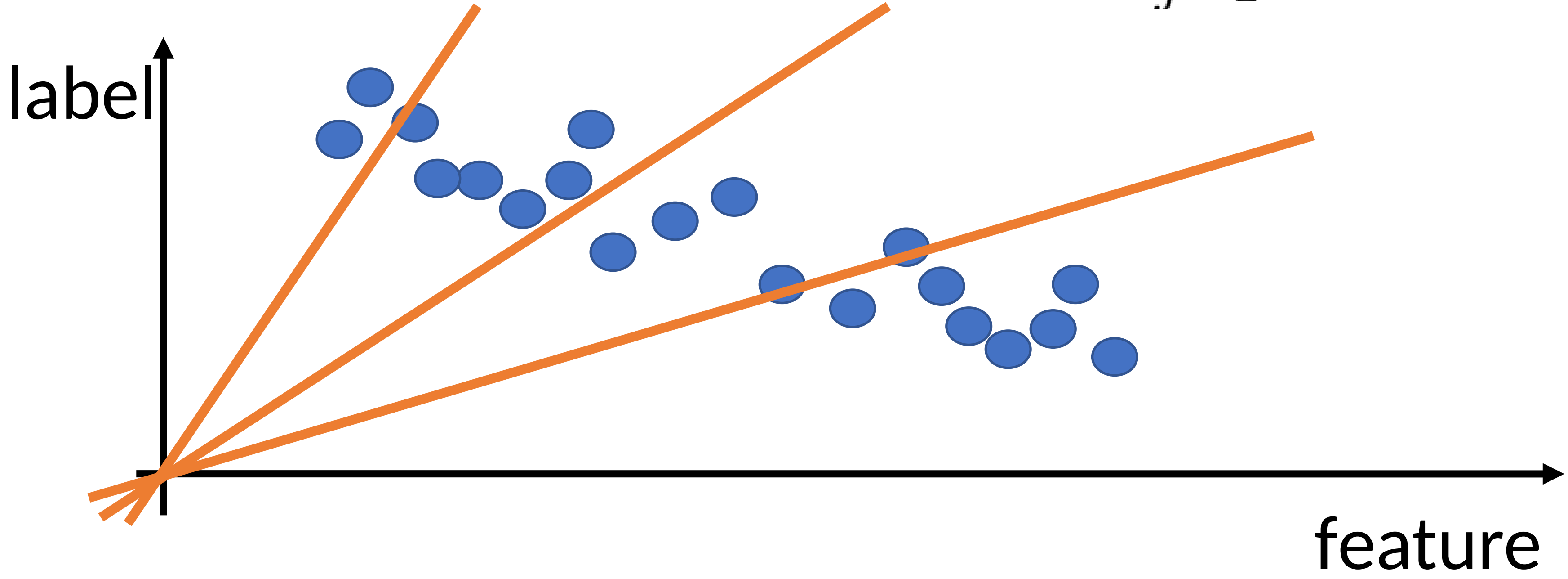


Model = several hypothesis



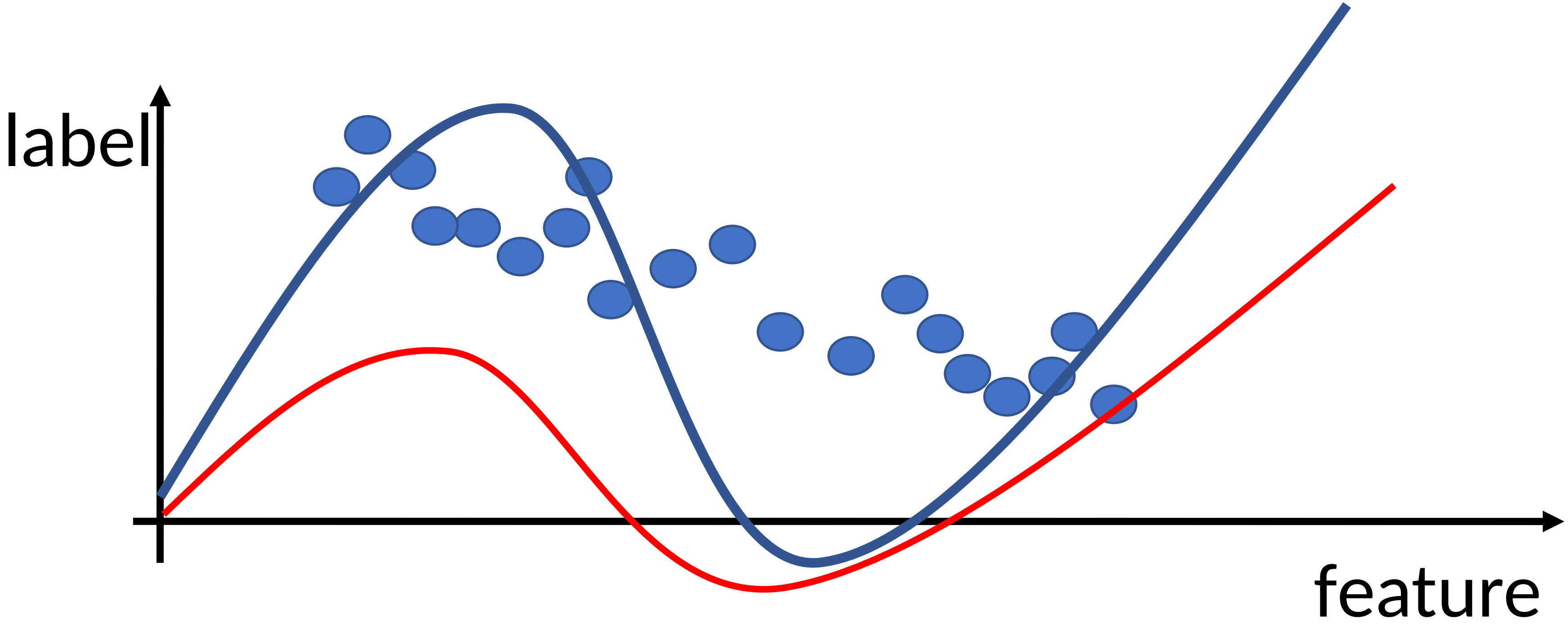
Linear model

$$h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{j=1}^{n} w_j x_j$$

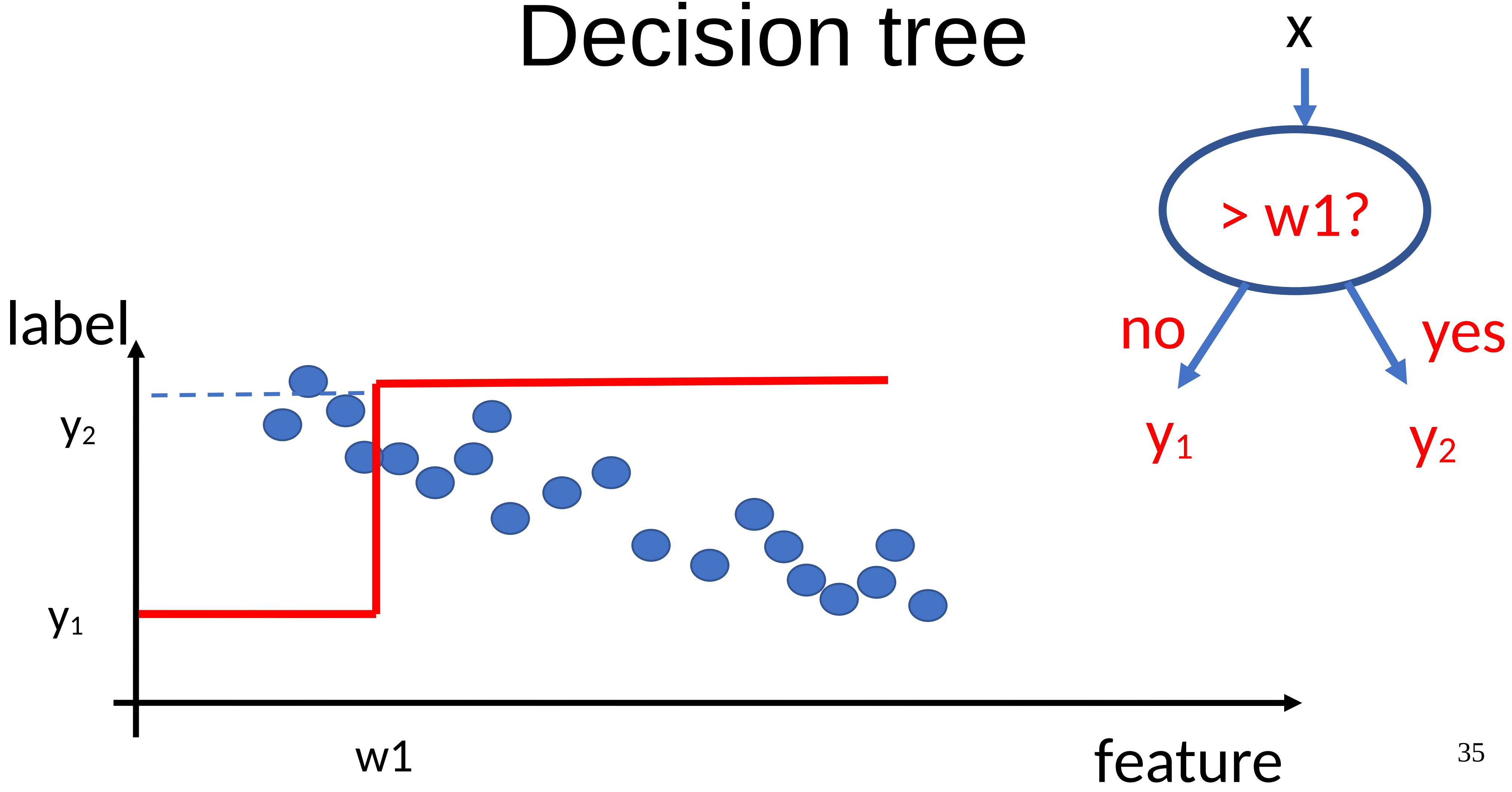


Polynomials

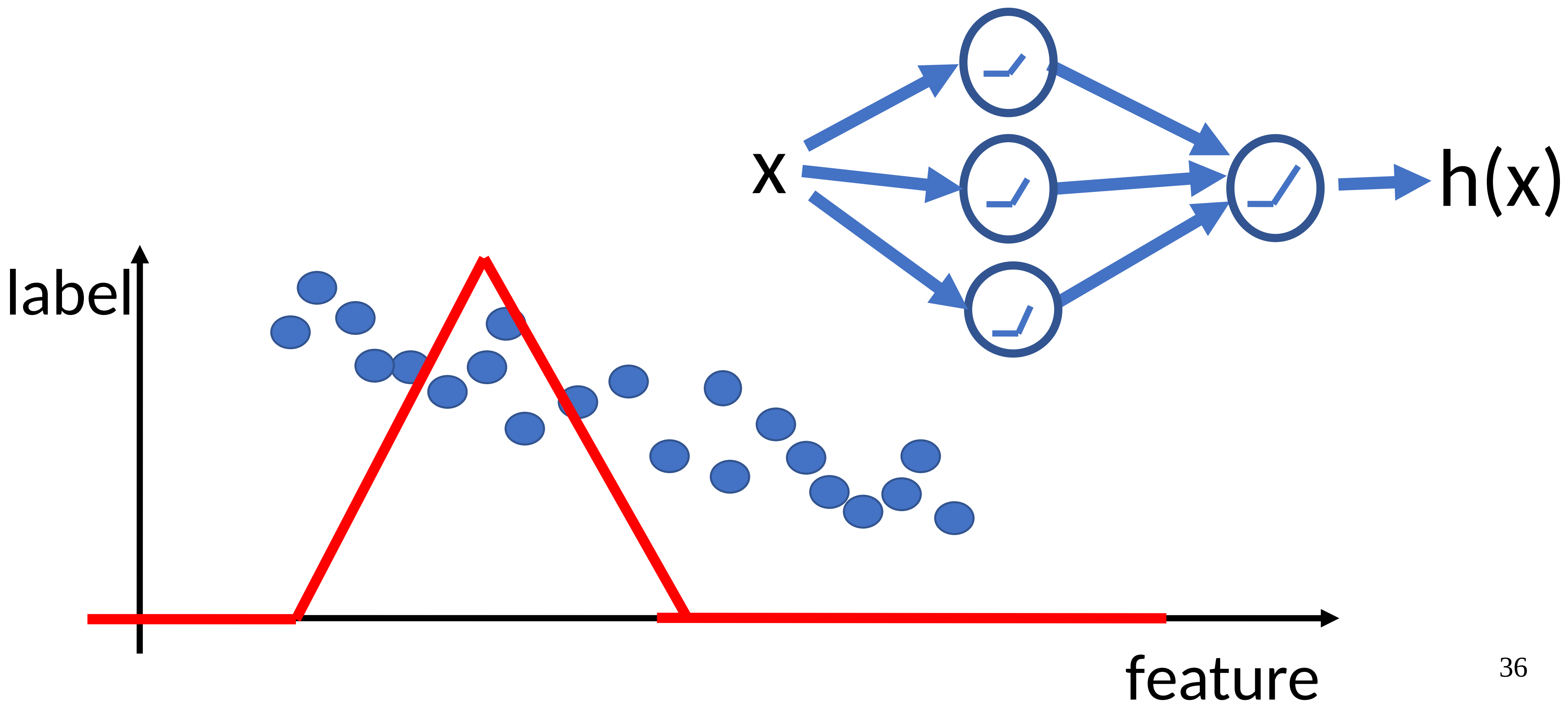
$$h^{(\mathbf{w})}(x) = \sum_{r=1}^{k} w_r x^{r-1}$$



Decision tree

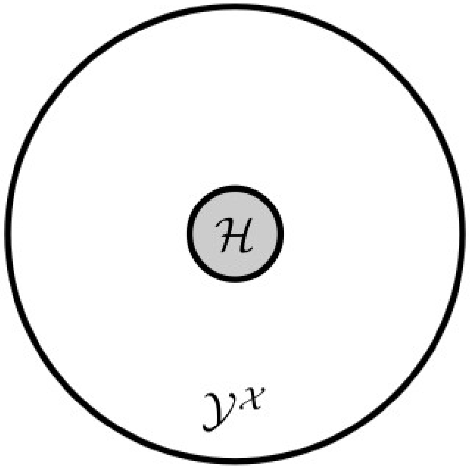


Artificial neural network



Size of hypothesis space

- $h : \mathcal{X} \to \mathcal{Y} \qquad h \in \mathcal{H}$
- The hypothesis space H is a (typically very small) subset of the (typically very large) set Y^X of all possible maps from feature space X into the label space Y.

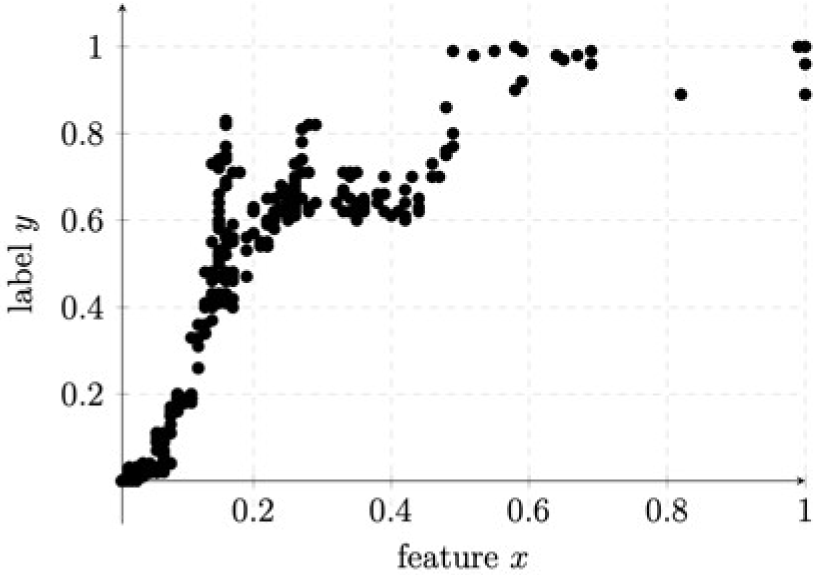


Which model to choose?

- **Large** to contain a good hypothesis

Sufficiently large

- Linear model might be **too small** for such data
- There is no straight line that fits well the
- Data points here need **larger models** that also contain non-linear maps

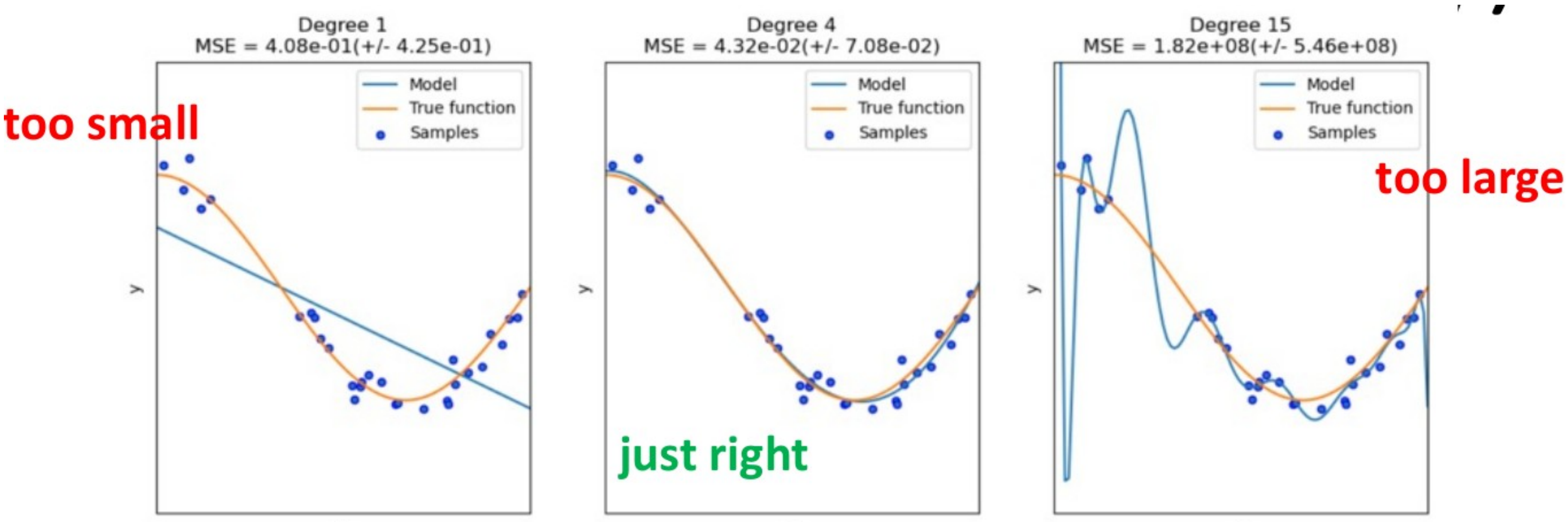


Which model to choose?

- **Large** to contain a good hypothesis
- **Small** to avoid **overfitting**
- **Small**/simple to fit **computational resources**

Sufficiently small

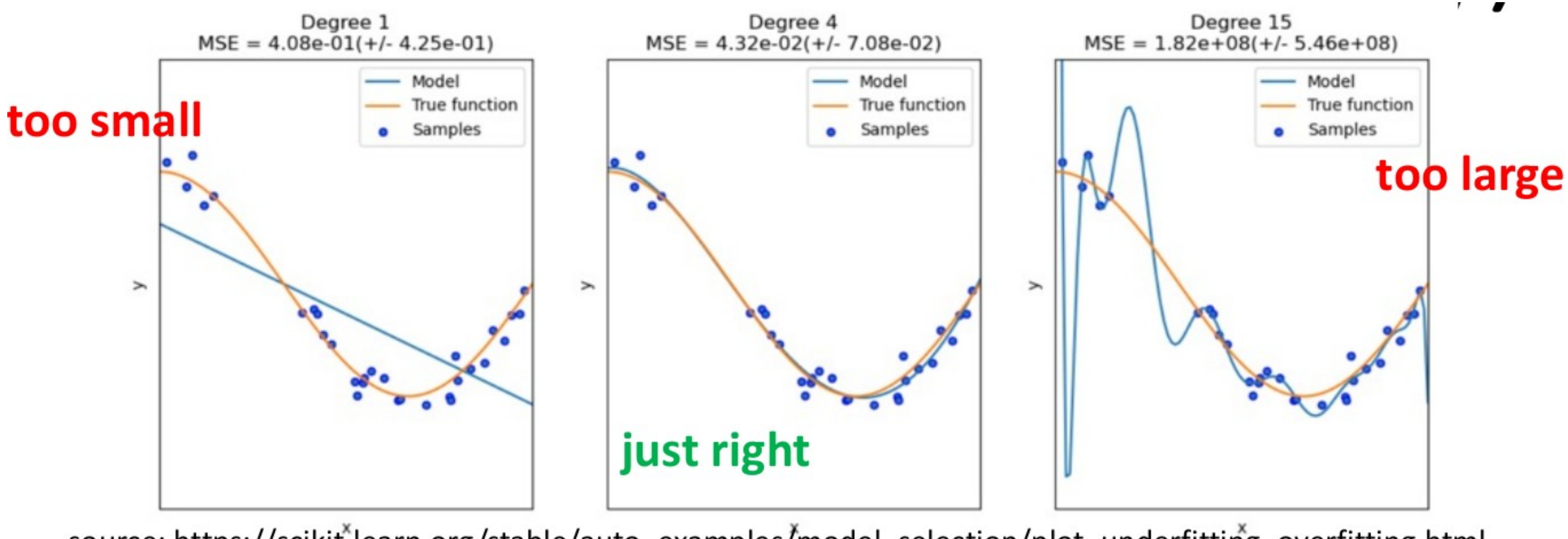
Statistically



source: https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

Sufficiently small

Statistically



source: https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html

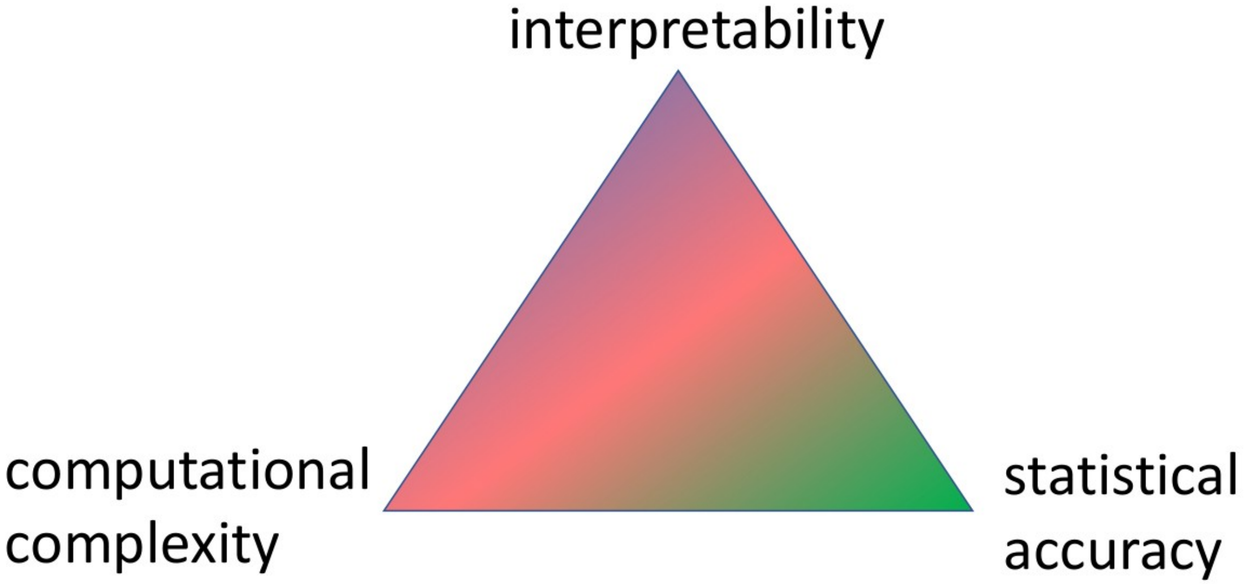
Overfitting: model fits well training data but does a very poor job oautside the training data

Sufficiently small

Computationally

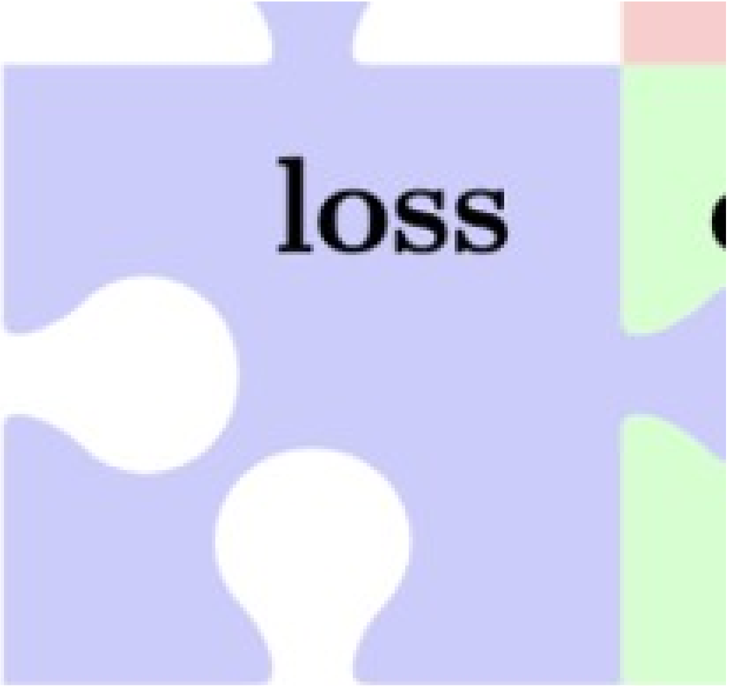
- hypothesis map h(x) easy to learn=train
- hypothesis map h(x) easy to evaluate

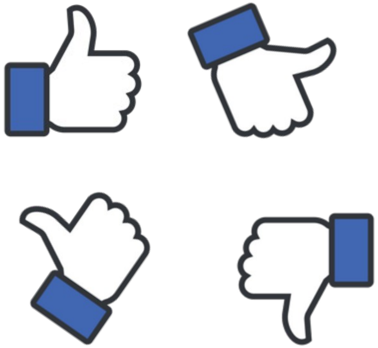
Design choice: model



Loss

Loss



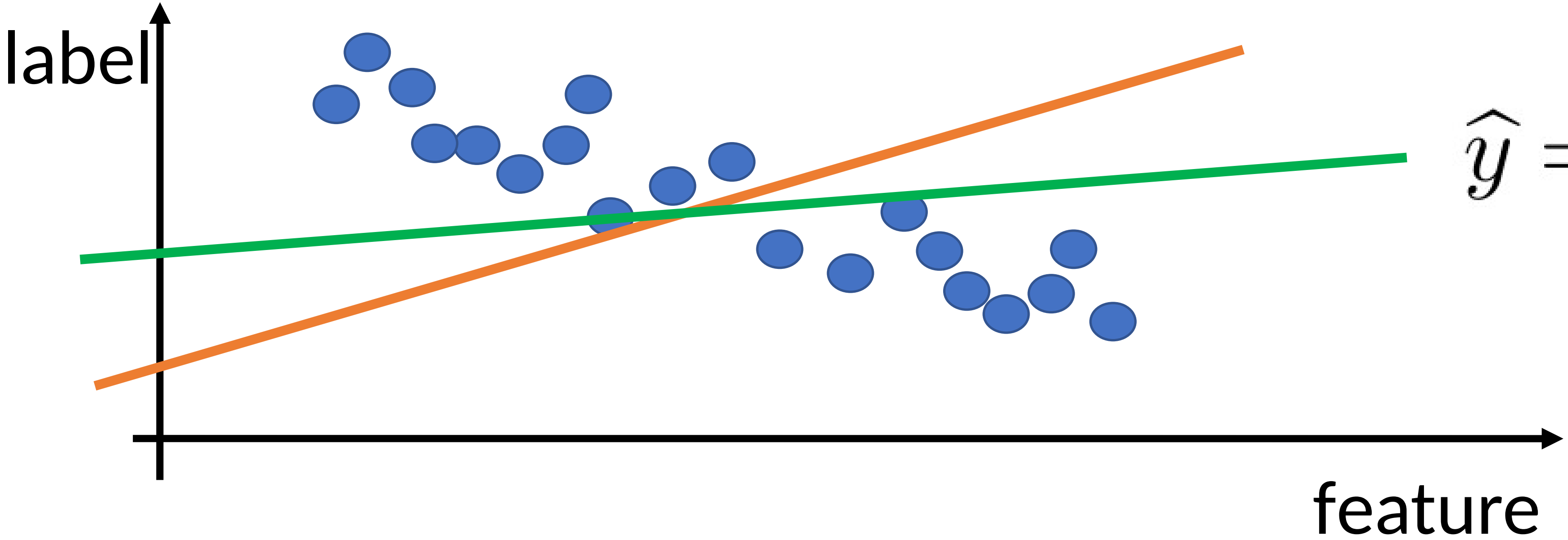


"Loss functions: because every machine needs a little heartbreak to learn."

ChatGPT

Which hypothesis is better

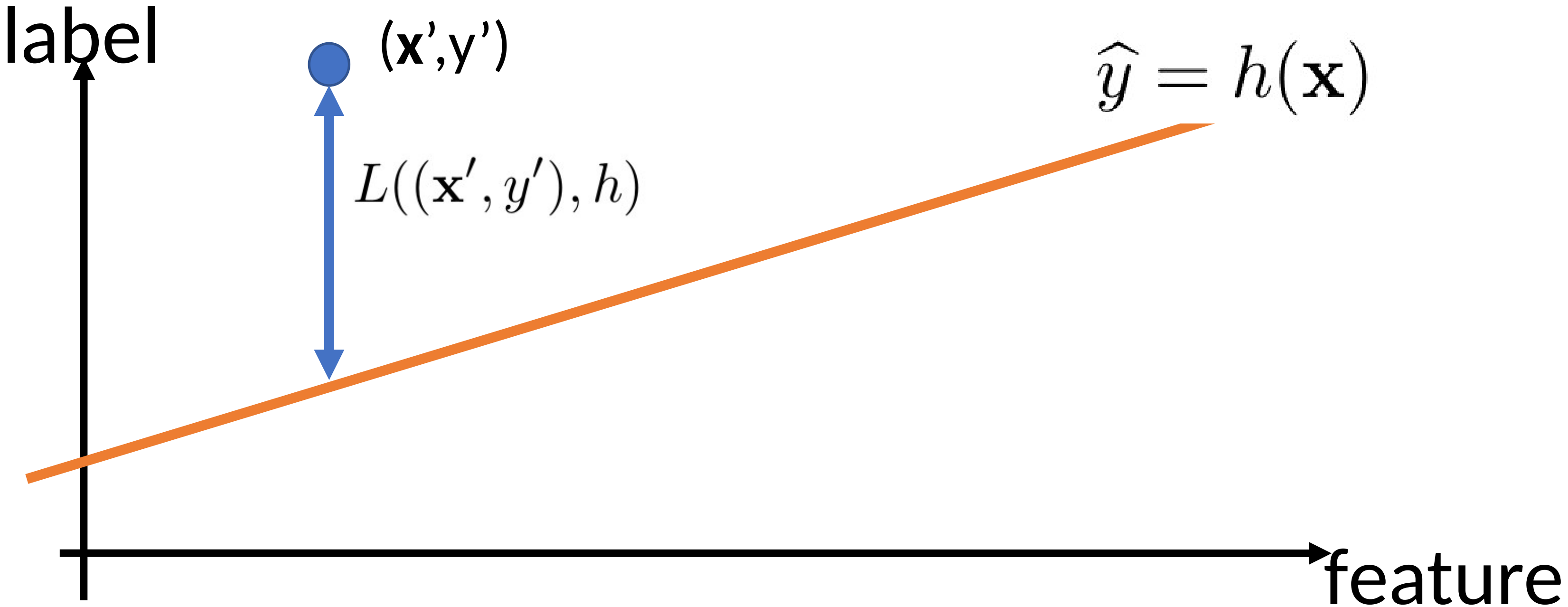
$$\widehat{y} = h(\mathbf{x})$$



$$\widehat{y} = h'(\mathbf{x})$$

A loss function

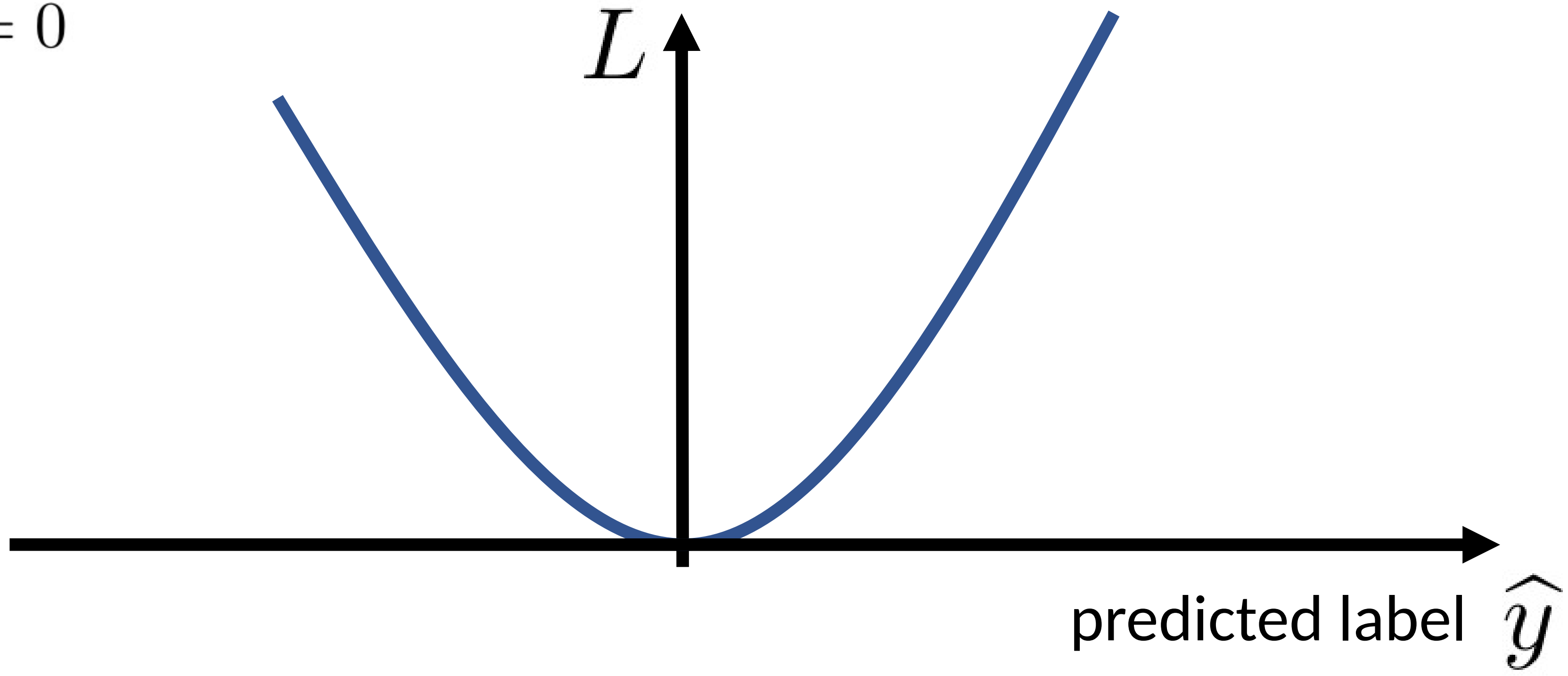
Quantitative measure of prediction error obtained when using hypothesis h to predict label y' of datapoint with features **x**'



Squared error loss

$$L := (\widehat{y} - y)^2$$

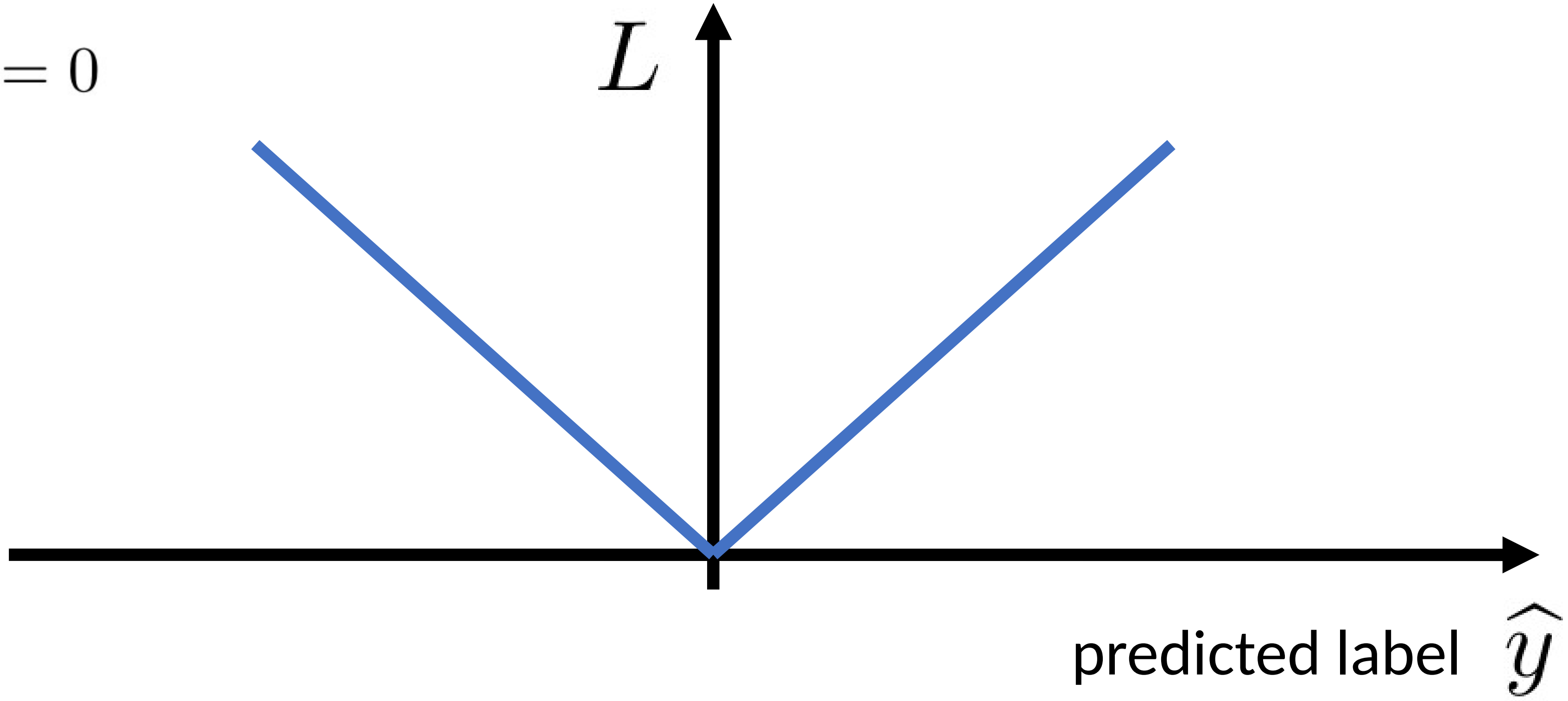
if $y = 0$



Absolute error loss

$$L := |\widehat{y} - y|$$

if $y = 0$



Loss Functions for Binary Classification

0/1 loss

label y = “cat”

h(x) = “dog”

Loss = 1

features **x** = pixels

Loss Functions for Binary Classification

0/1 loss

label y = “cat”

h(x) = “cat”

Loss = 0

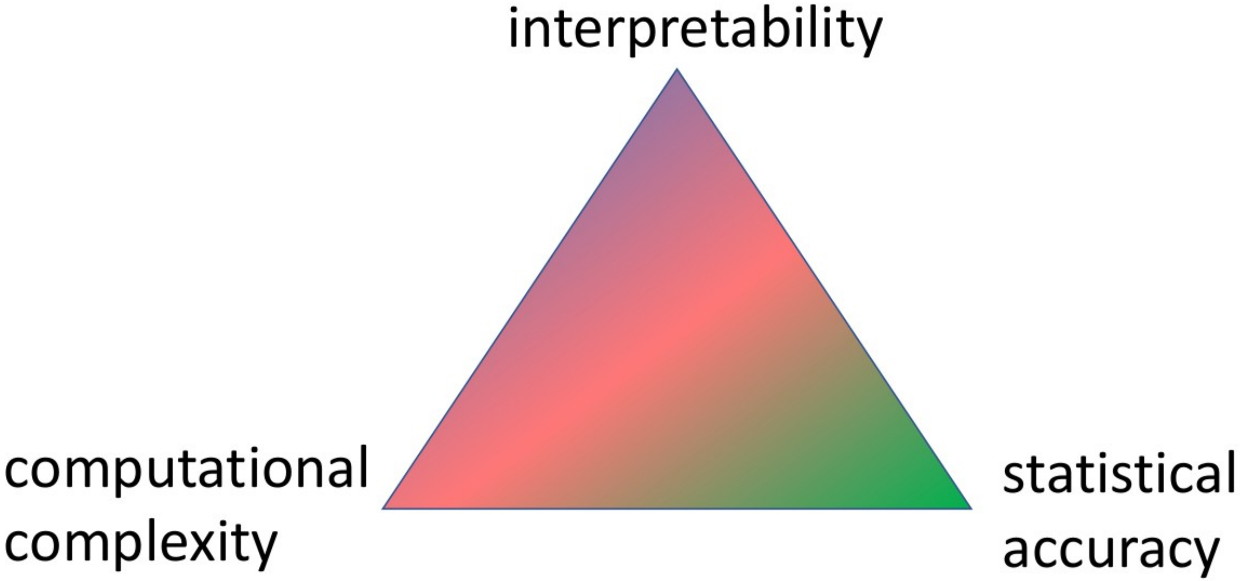
features **x** = pixels

Which loss function ?

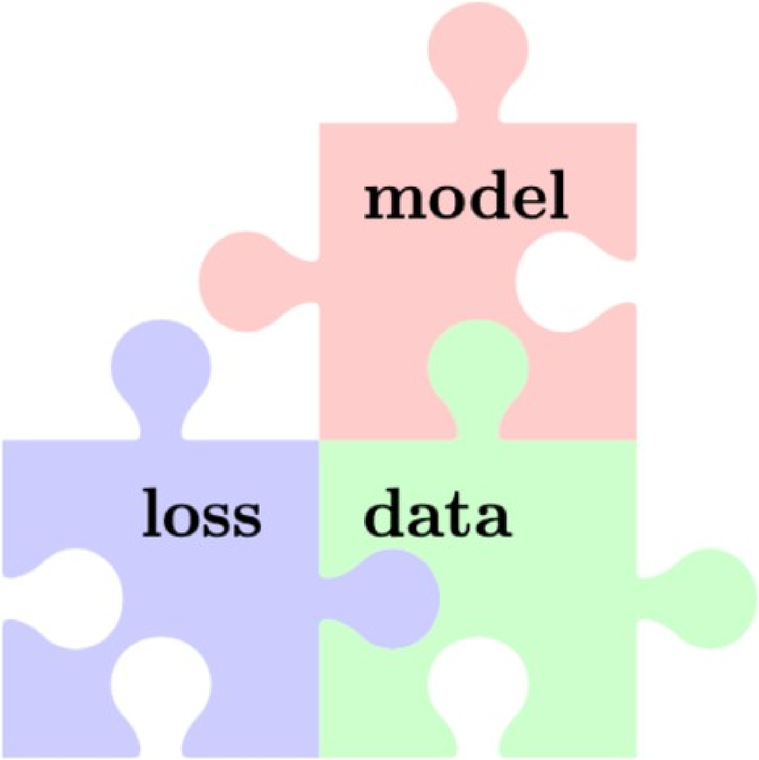
- **Statistical aspects** -- should favour “reasonable” hypothesis
- **Computational aspects** -- must be able to minimize them
- **Interpretation** -- what does log-loss = -3 mean ?

- ...choosing a suitable loss function is often **non-trivial** !

Design choice: loss



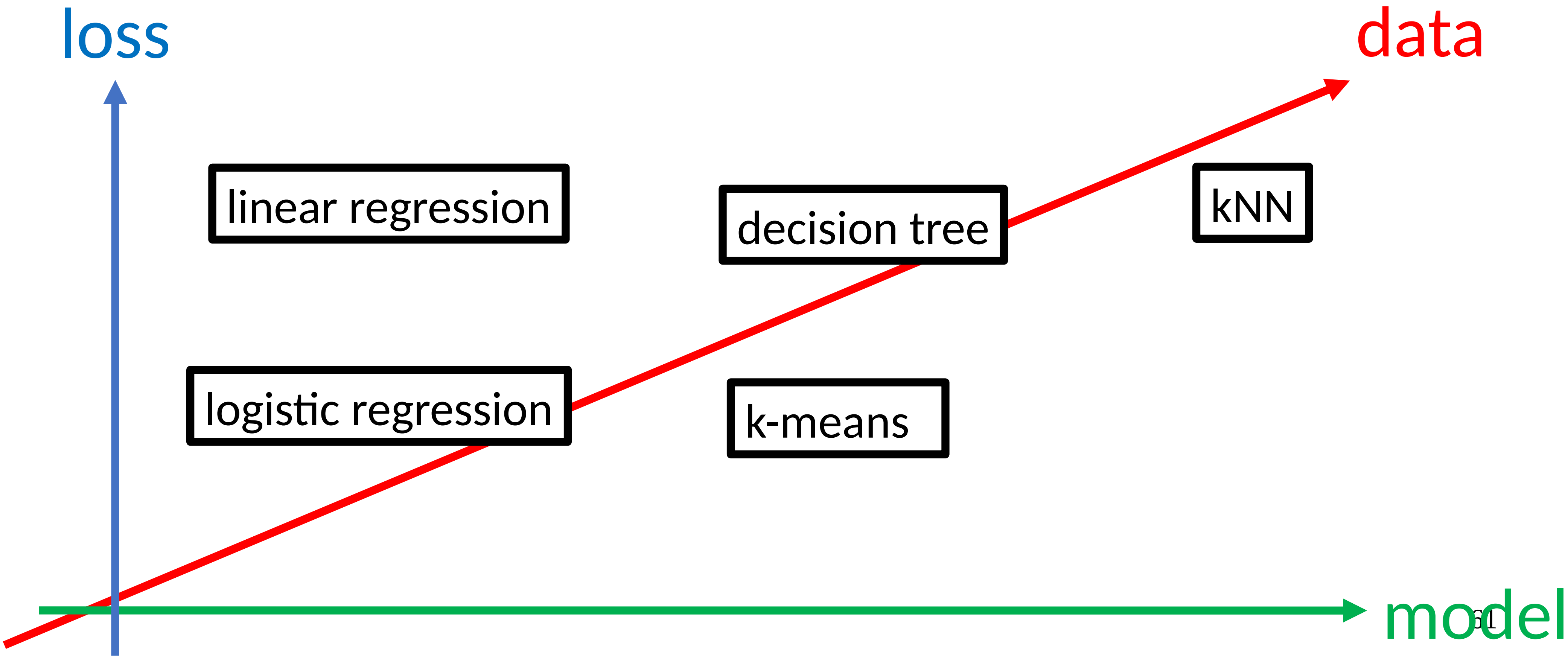
Three components of ML



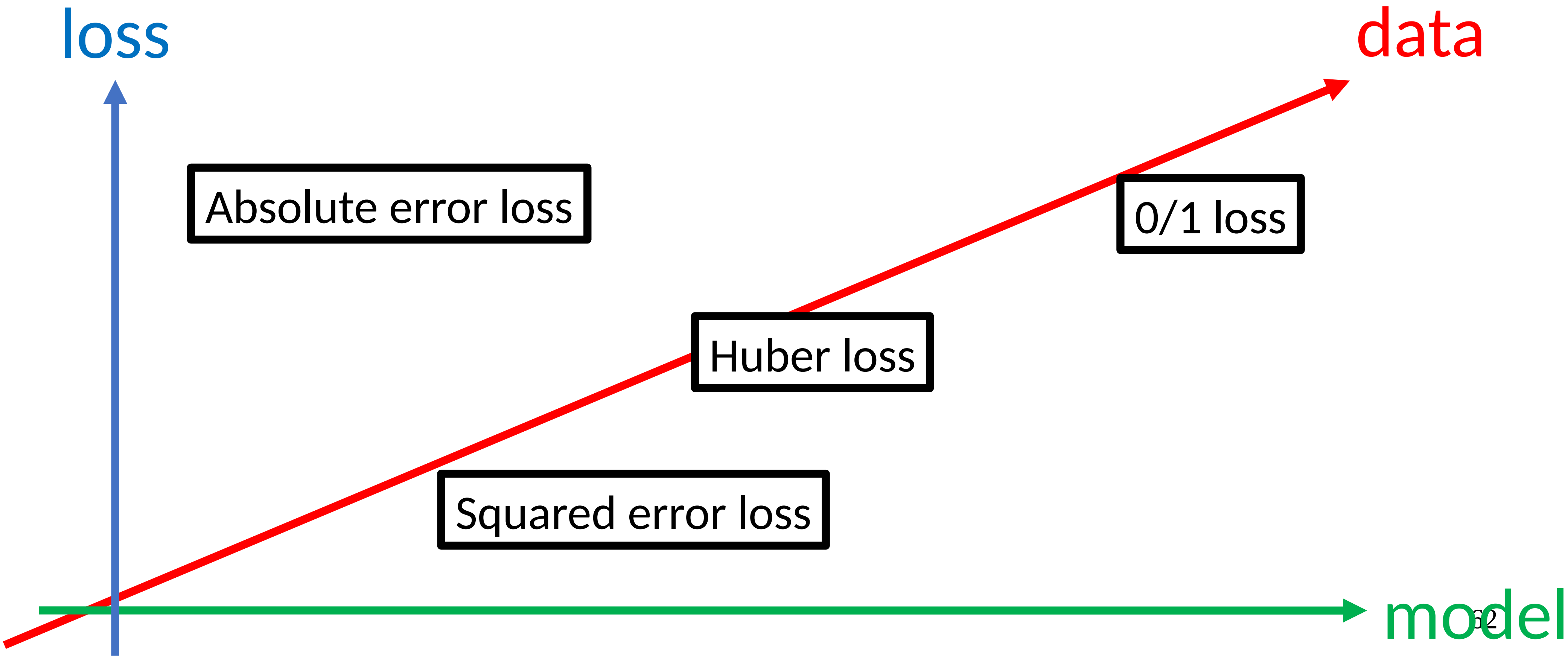
Landscape of ML Methods – data axis

Year	m	d	Time	precp	snow	airtmp	mintmp	maxtmp
2020	1	2	00:00	0,4	55	2,5	-2	4,5
2020	1	3	00:00	1,6	53	0,8	-0,8	4,6
2020	1	4	00:00	0,1	51	-5,8	-11,1	-0,7
2020	1	5	00:00	1,9	52	-13,5	-19,1	-4,6
2020	1	6	00:00	0,6	52	-2,4	-11,4	-1
2020	1	7	00:00	4,1	52	0,4	-2	1,3
2020	1	8	00:00	4,3	51	0,8	0,1	1,8
2020	1	9	00:00	-1	51	-0,6	-1,9	1,6
2020	1	10	00:00	-1	51	-6,2	-11	-1,4
2020	1	11	00:00	2,8	50	-4,8	-10,7	-2,1
2020	1	12	00:00	-1	53	-1,3	-3,5	0,9
2020	1	13	00:00	-1	53	-6,4	-12,9	-3,1
2020	1	14	00:00	9,7	52	-2,8	-9	-0,7
2020	1	15	00:00	-1	63	0,2	-0,7	0,6
2020	1	16	00:00	0,4	62	-3,9	-5,2	0,1
2020	1	17	00:00	2	62	-5,2	-8,4	-0,7

Landscape of ML Methods – model axis



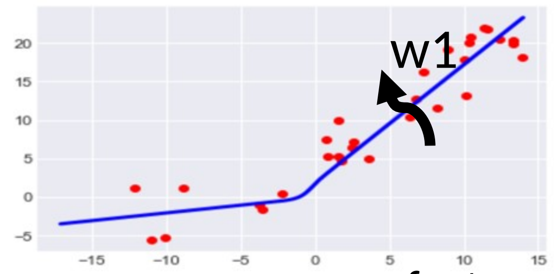
Landscape of ML Methods – loss axis



Three Views on Machine Learning

Data View

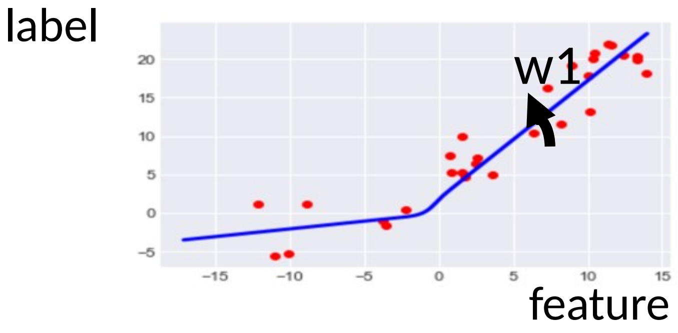
label



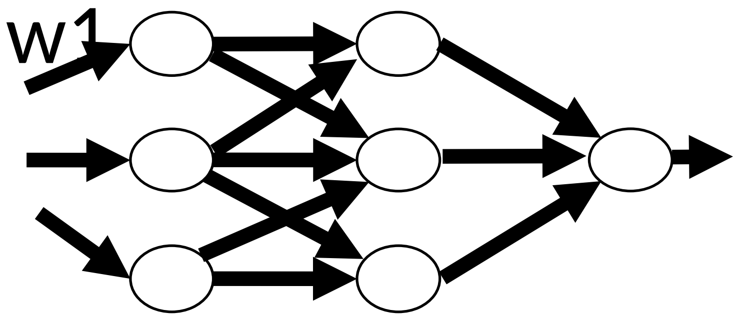
feature

Three Views on Machine Learning

Data View

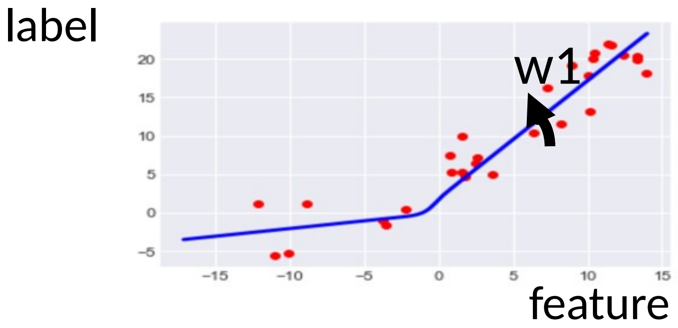


Model View

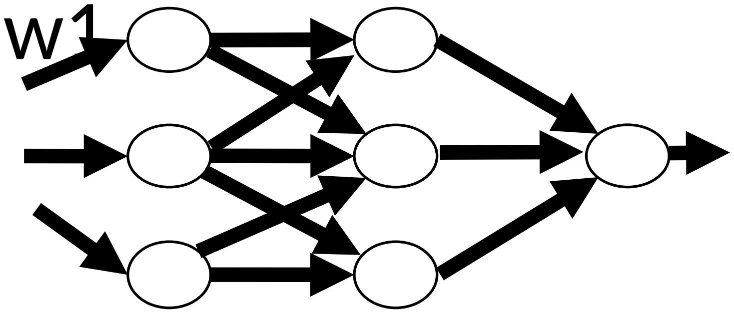


Three Views on Machine Learning

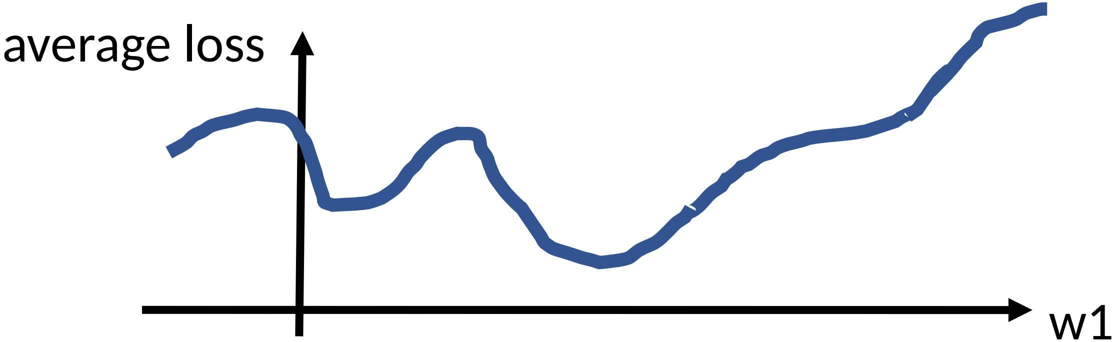
Data View



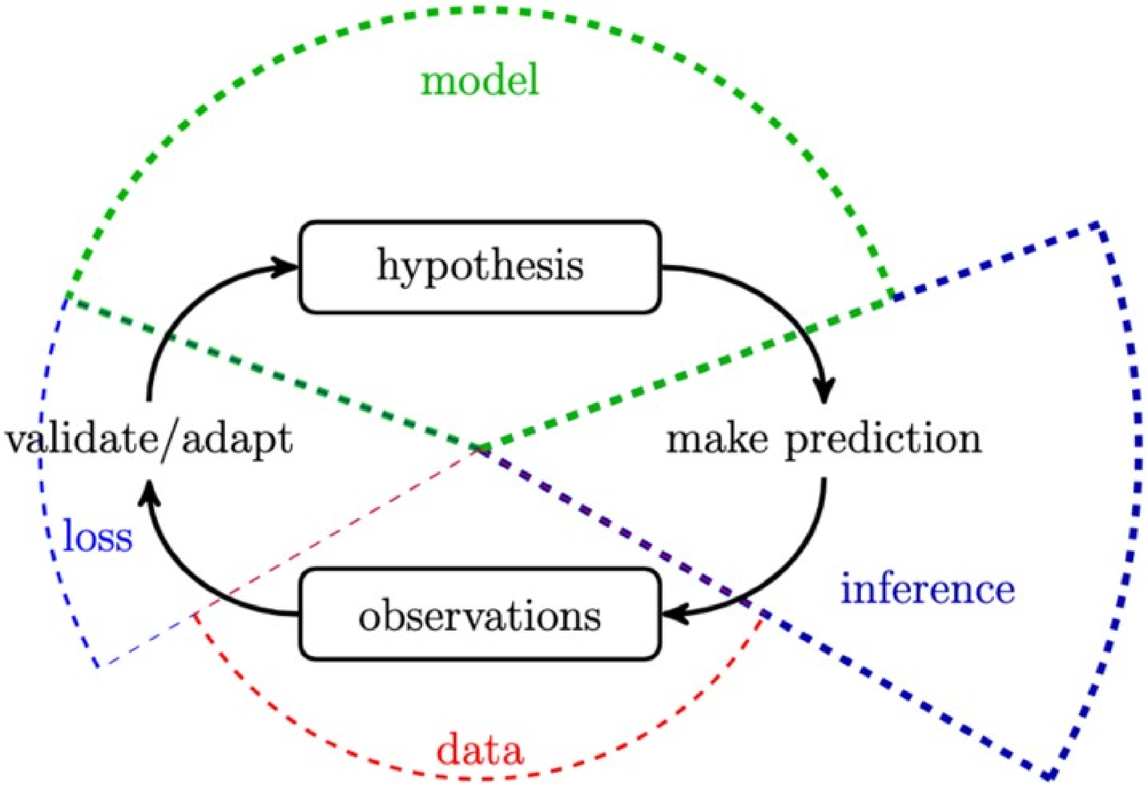
Model View



Loss View



ML process


model
hypothesis
validate/adapt
make prediction
loss
inference
observations
data

Any questions?

Self-assessment quiz

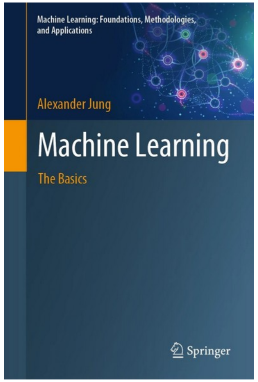
- Make a new example of data points, with their features and labels.
- Consider the following data points $x^{(1)}=(1,7,2.6,-2)$ and $x^{(2)}=(3,4,-10,0)$.

  Create a linear model by (randomly) choosing the weights **w**. $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

  Which weights did you choose and what is the output $h(x^{(1)})$ and $h(x^{(2)})$?
- What is the loss of the proposed model in $x^{(1)}$ e $x^{(2)}$ with respect to their true labels $y^{(1)}=1$ and $y^{(2)}=0$? Compute their squared error loss, their absolute error loss and 0/1 loss.

References: readings

- Chapter 2



Slide acknowledgments

- Alexander Jung – Aalto University