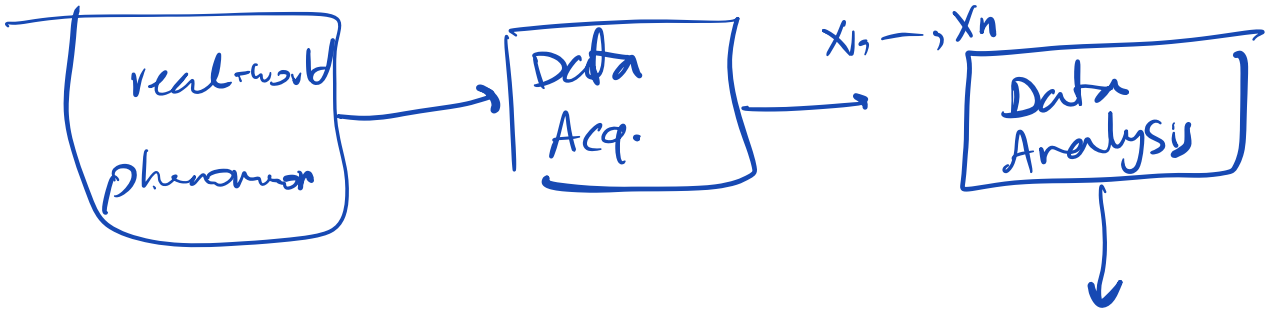
Lecture 13:



Modules 1, 2 :

- learned to estimate properties/parameters connected to the underlying phenomenon
- hypothesis testing: verify different hypotheses connected with the underlying phenomenon.

Module 3: Supervised learning: "learn" some "predictive relation" connected to the phenomenon.

Supervised learning:

We assume that each data point x_i is associated with a "label" (which is a part of the data). I.e. each data point is of the form (x_i, y_i), where x_i is the "feature vector" and "y_i" is the label associated with x_i.

Examples:

- x_i: could be an image
- y_i: could an object inside the image

- x_i: could be (the text of) an email
- y_i: could be whether/not the email is spam.

x_i: a vector representing a patients profile, $X_i = (\text{age}, \text{height}, \text{weight}, \text{MRI}, \ldots)$

y_i: cancer / not cancer

Data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Goal: $y_i \approx f(x_i)$

- y_i: spam/ham (0 / 1)
- f: predictor
- x_i: email text

→ Main goal: Find the best predictor f.

Setting: Data: $\{(x_i, y_i)\}_{i=1}^{n}$

$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p$ ←

- image: pixel representation
- profile of the i-th patient

y_i = label

Supervised learning
- Regression
  - label $y_i \in \mathbb{R}$
  - e.g. probability of having cancer
- classification
  - $y_i \in A$

where A is a discrete set

e.g. $A = \{0, 1\}$

e.g. Spam / Ham

Regression:

- Let's start with the simplest setting (for the regression problem):

- 1-d regression

Data: $\{(x_i, y_i)\}_{i=1}^{n}$

$x_i \in \mathbb{R}$

$y_i \in \mathbb{R}$ ($p = 1$)

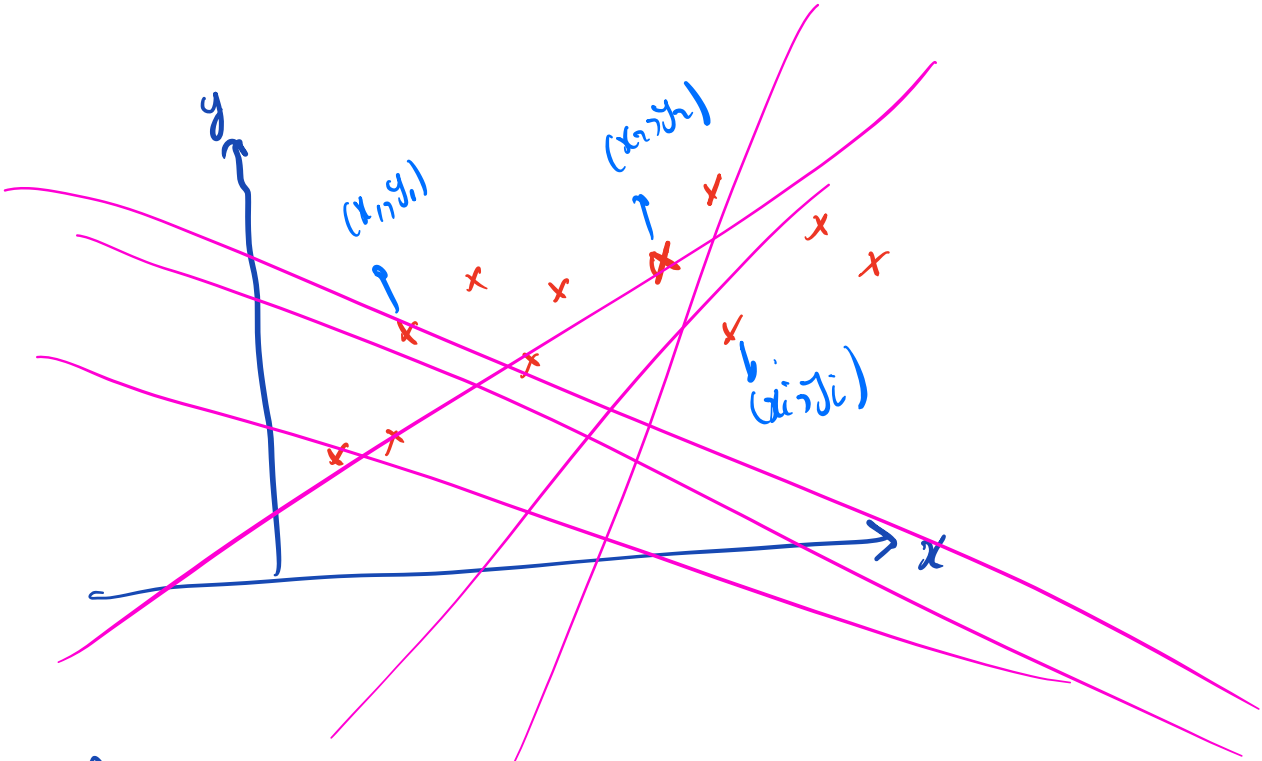
Goal: learn a "predictive" relation:

$$\hat{y} = \hat{f}(x)$$

example: x = blood sugar at the age of 20

y = probability of having Diabetes at the age of 40.

Data: $\{(x_i, y_i)\}_{i=1}^{\hat{}}$



$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \leftarrow$$

$$\underbrace{\hat{\beta}_0 + \hat{\beta}_1 x}_{f(x)}$$

Approach:

Goal: learn $y = f(x)$

- Search for the best "candidate" for f inside a prescribed family of functions F.
- find the best f inside F
- typically F is a parametric family $\rightarrow f(x; \beta)$.

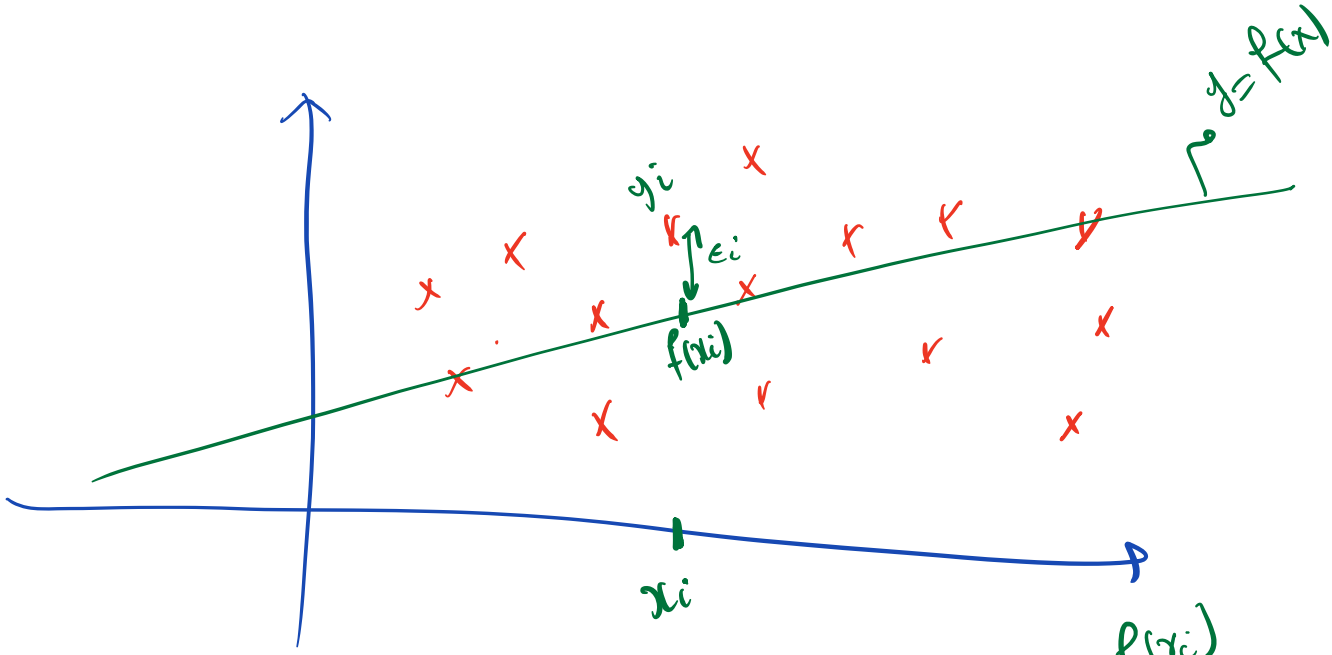
  e.g. linear functions

- Let's consider the family of linear functions as our first step. If $x \in \mathbb{R}$ then this family is described by two parameters $\hat{\beta}_0, \hat{\beta}_1$

$$f(x; \hat{\beta}_0, \hat{\beta}_1) = \hat{\beta}_0 + \hat{\beta}_1 x$$

β

Goal: find the best choice of $\hat{\beta}_0, \hat{\beta}_1$ from data.



$$y_i = f(x_i) + \varepsilon_i$$

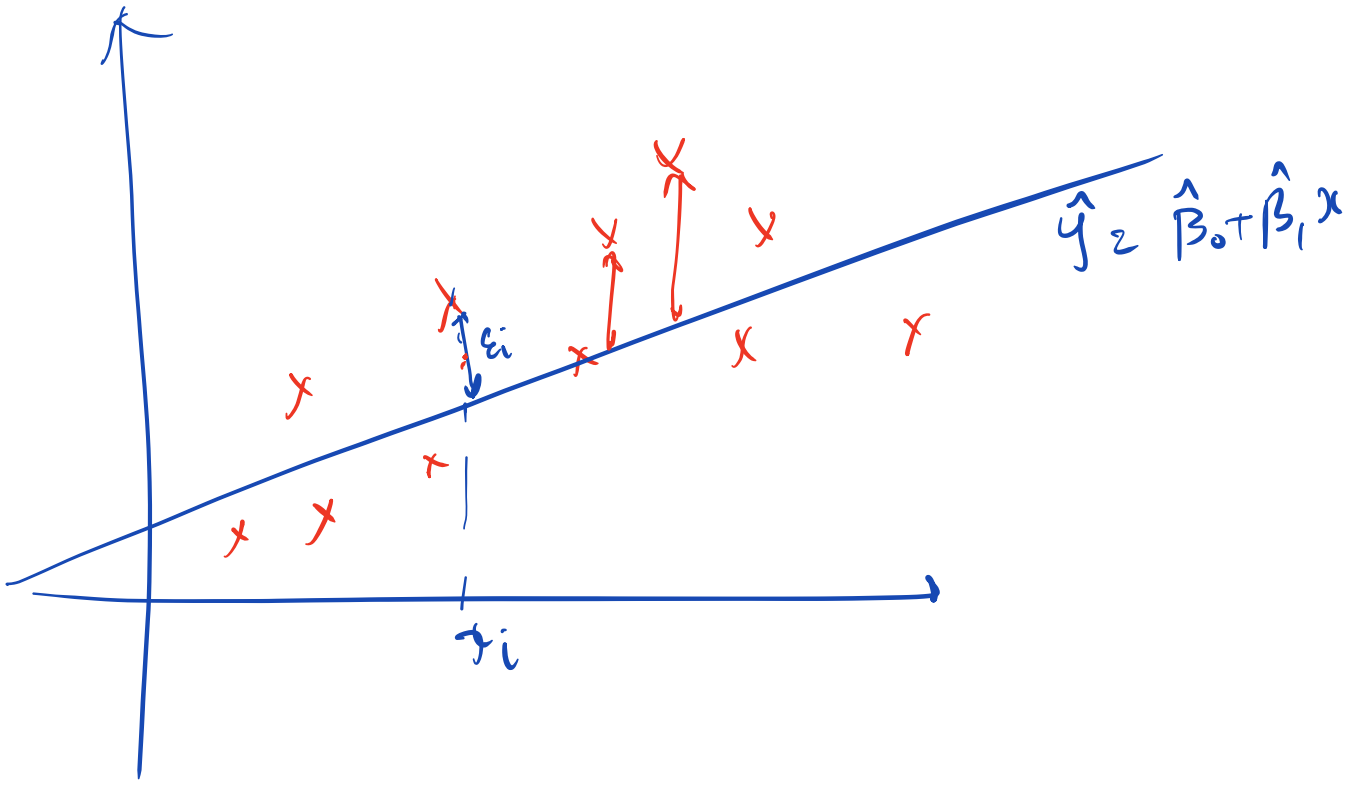
ε_i → error $= \varepsilon_i := y_i - f(x_i)$, with $f(x_i) = \hat{\beta}_0 + \hat{\beta}_1 x_i$

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \varepsilon_i$$

→ the error term models what we miss when we use this simple linear model. The true relation between the input and output is probably not linear as many other factors can be involved

(there may also be noise in the label)

Question: What is the fundamental procedure principle behind learning/choosing the parameters $\hat{\beta}_0, \hat{\beta}_1$?



Idea 1: find $\hat{\beta}_0, \hat{\beta}_1$ s.t.

$$\varepsilon_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

Sum of the errors

$$\sum_{i=1}^{n} |\varepsilon_i|$$

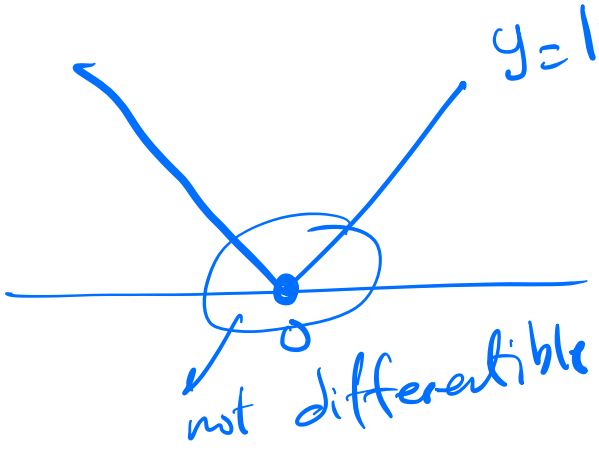
is minimized

$$\min_{\hat{\beta}_0, \hat{\beta}_1} \sum_{i=1}^{n} |y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i|$$

↓

- In order to solve the above optimization problem, we'll have to take the derivative wrt. $\hat{\beta}_0, \hat{\beta}_1$ and set it to zero.

- The problem with the above objective is that it's not differentiable.



- Idea 2: To make things differentiable let's consider the "square" of the errors:

$$\underset{\hat{\beta}_0, \hat{\beta}_1}{\text{minimize}} \sum_{i=1}^{n} \underbrace{\left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)\right)^2}_{\varepsilon_i^2}$$