

# Inteligență Artificială

Bogdan Alexe

[bogdan.alexe@fmi.unibuc.ro](mailto:bogdan.alexe@fmi.unibuc.ro)

Secția Tehnologia Informației, anul III, 2022-2023

Cursul 7

# Sistem de notare proiect Kaggle

- proiectul valorează 1.5 puncte din nota finală
- partea de concurs Kaggle = 1 punct
  - locul 1 = 1 punct
  - locul 2 = 0.99 puncte
  - ...
  - locul 50 = 0.51 puncte
  - locul 51+ = 0.5 puncte (cât timp aveți o performanță > baseline)
- partea de documentație + prezentare = 0.5 puncte
- important: pentru fiecare submisie să știți ce ați făcut (cod Python, parametri, etc). La final vă veți alege 2 submisii care credeți voi că sunt cele mai bune (pot fi același model cu parametri diferiți). Vă bifați cele 2 submisii în platforma Kaggle înainte de deadline.

# Documentație proiect - pdf

- descrieți în detaliu (1-2 pagini) **2 modele diferite** folosite (kNN, Naïve Bayes, SVM, perceptron, rețea neuronală):
  - ce caracteristici folosiți;
  - care sunt parametri, hiperparametri modelului;
  - cum antrenați parametri/hiperparametri;
  - cât durează antrenarea;
  - ce performanță ați obținut pe cele 40% de date din setul de date de test public pe Kaggle;
- **pentru un singur model prezentați rezultatele în urma antrenării în maniera 5 fold cross-validation + matricea de confuzie asociată**

# Predare proiect

- predarea proiectului înseamnă trimiterea documentației și a codului Python pentru fiecare submisie
- trimiteți la adresa de email: [ub.fmi.cti.ia@gmail.com](mailto:ub.fmi.cti.ia@gmail.com) un email până luni, 21 noiembrie, ora 23:59 cu următoarele fișiere:
  - un fișier pdf cu documentația voastră
  - două fișiere python cu codul pentru submisiile voastre
  - respectați formatul de mai jos



361\_Alexe\_Bogdan\_documentatie.pdf



361\_Alexe\_Bogdan\_submisie1\_cod.py



361\_Alexe\_Bogdan\_submisie2\_cod.py

# Prezentare proiect

- este individuală, are loc în săptămâna 8 (21-25 noiembrie)
- constă într-o discuție cu Alexandra/Sergiu/Bogdan de maxim 10 minute
  - prezentarea voastră 3-5 minute (ce modele ați folosit la cele 2 submisii)
  - 3-5 minute întrebări din partea noastră
- vom face o programare pe care o vom afișa luni, după concurs, în TEAMS:
  - grupa 361: marti 12-14-16
  - grupa 362: miercuri 8-10-12
  - grupa 363: miercuri 10-12-14
  - grupa 364: marti 10-12-14
- dacă nu puteți veni la grupa voastră sau doriți să prezentați într-un anumit interval orar vă rog să îmi scrieți mie pe email până duminică seara

# Recapitulare – cursul trecut

1. Alte reguli de învățare pentru perceptron
2. Rețele feedforward multistrat de perceptroni

# Cuprinsul cursului de azi

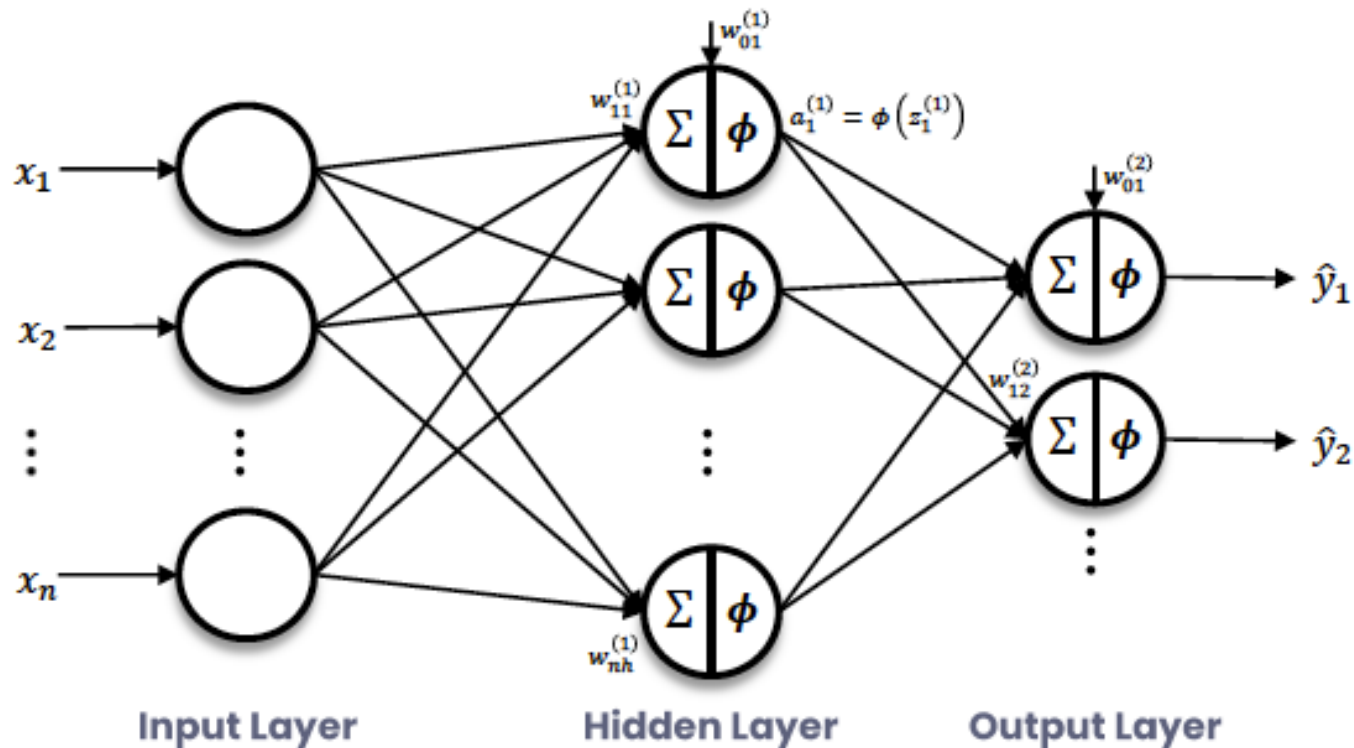
1. Rețele feedforward multistrat de perceptroni
2. Regresia liniară simplă și multiplă

Rețele feedforward multistrat de  
perceptroni  
(Multilayer perceptrons)



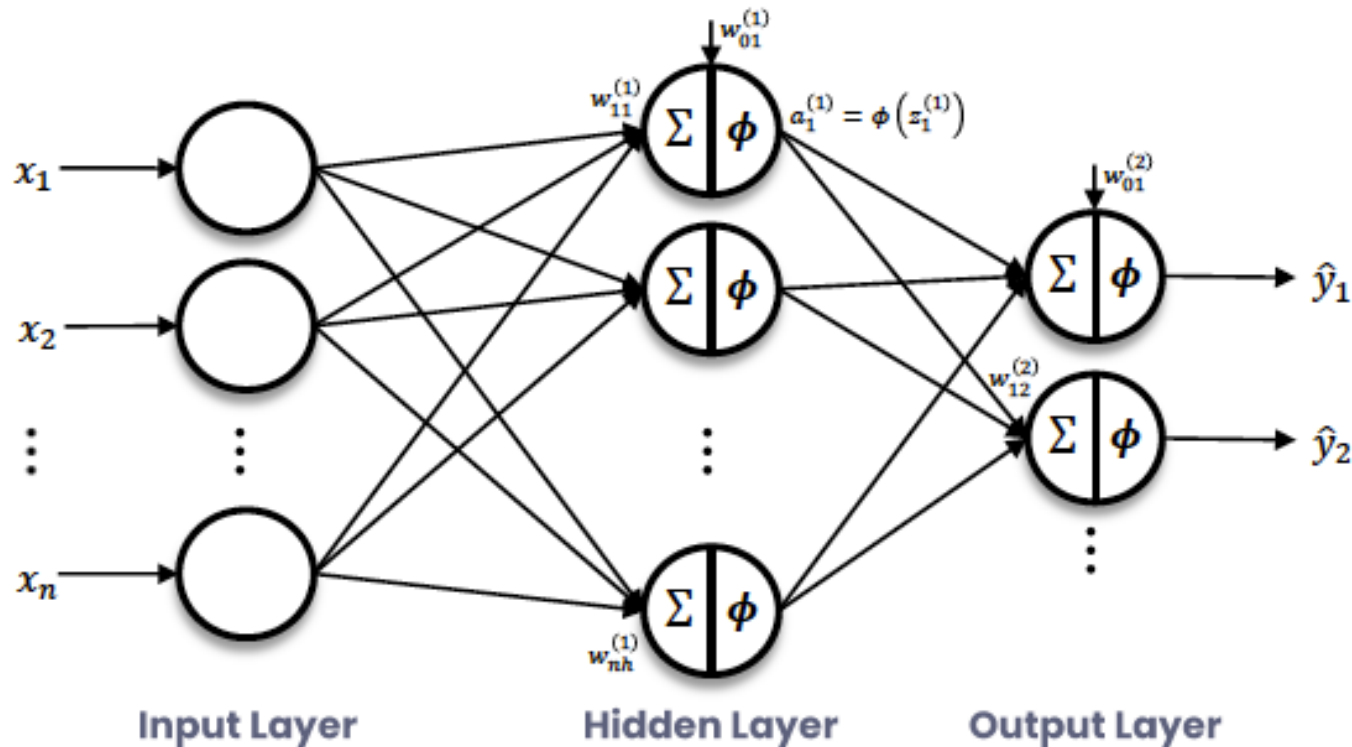
# Rețele feedforward multistrat de neuroni

- O rețea feedforward multistrat de neuroni (perceptroni) este o rețea de neuroni grupați pe straturi (layere), în care propagarea informației se realizează numai dinspre intrare spre ieșire (de la stânga la dreapta). Rețeaua are un strat de intrare (input layer), unul sau mai multe straturi ascunse (hidden layers) și un strat de ieșire (output layer).



# Rețele feedforward multistrat de neuroni

- Toți neuronii din rețea, cu excepția celor din stratul de intrare aplică o funcție de activare sumei ponderate ale intrărilor.
- Fiecare pereche de neuroni din două straturi consecutive are o pondere asociată



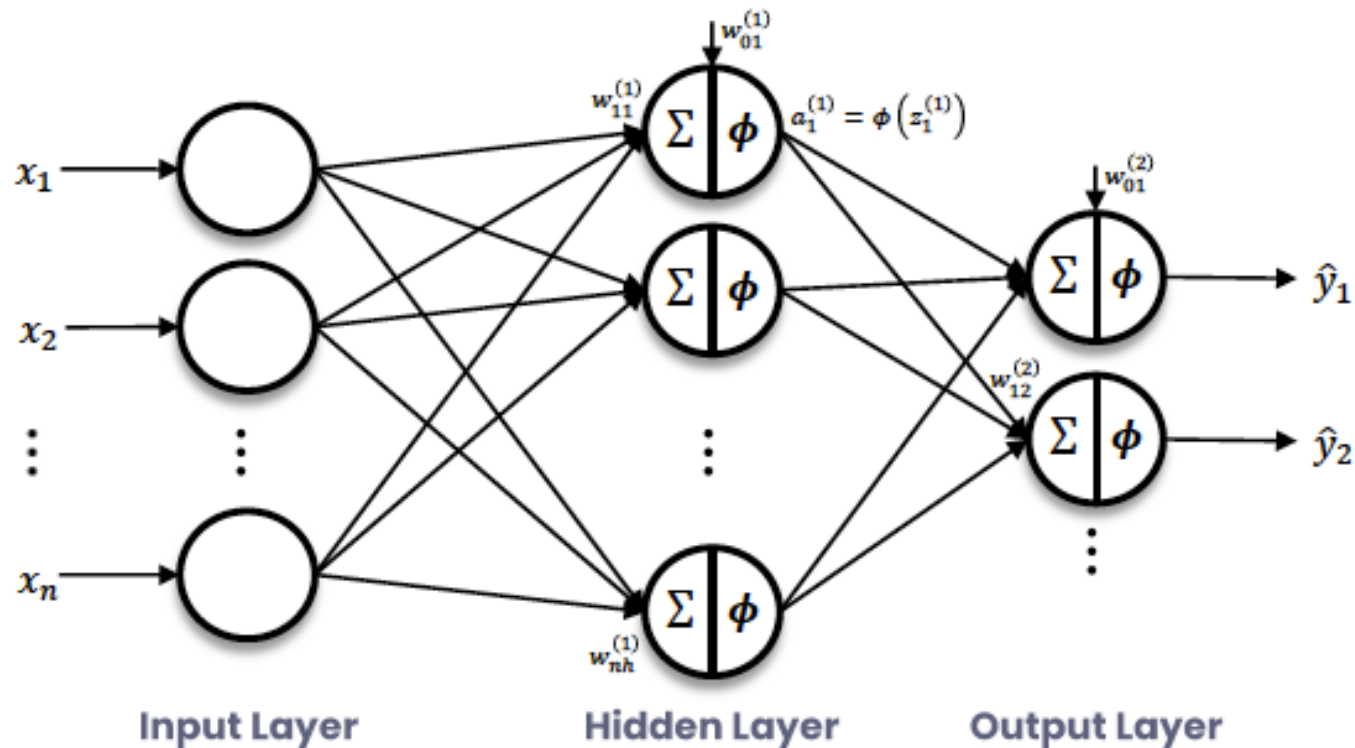
# Rețele feedforward multistrat de neuroni

$w_{ij}^{(l)}$  este ponderea neuronului  $i$  de pe stratul  $l - 1$  către neuronul  $j$  pe stratul  $l$ .

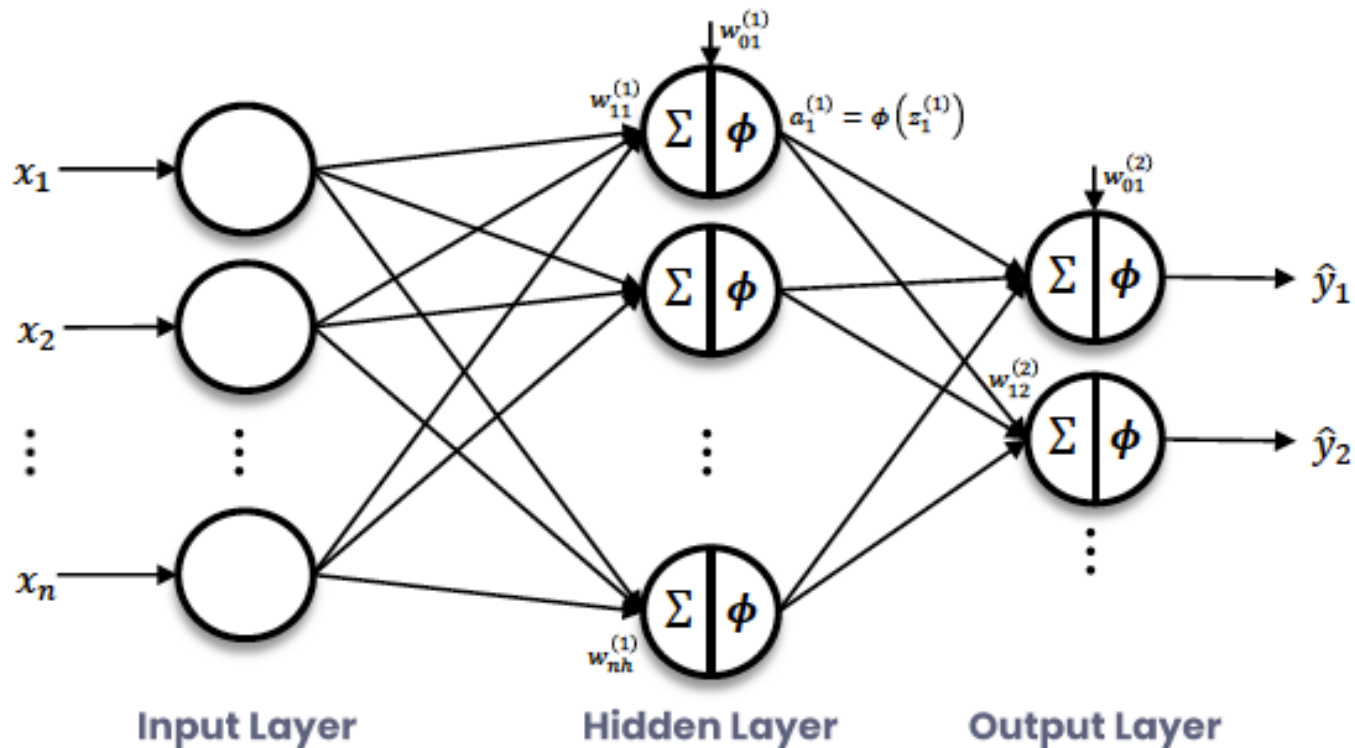
$w_{0j}^{(l)}$  este bias-ul (deplasarea) neuronului  $j$  de pe stratul  $l$ .

$z_j^{(l)}$  este ieșirea neuronului  $j$  de pe stratul  $l$  după însumarea intrărilor ponderate de la toți ceilalți neuroni.

$a_j^{(l)}$  este ieșirea neuronului  $j$  de pe stratul  $l$  după aplicarea funcției de activare  $\phi$  ieșirii  $z_j^{(l)}$ .



# Rețele feedforward multistrat de neuroni reprezentate în format matriceal



# Rețele feedforward multistrat de neuroni reprezentate în format matriceal

$$\vec{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad W^{(1)} = \begin{bmatrix} w_{01}^{(1)} & w_{11}^{(1)} & \cdots & w_{n1}^{(1)} \\ w_{02}^{(1)} & w_{12}^{(1)} & \cdots & w_{n2}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{0h}^{(1)} & w_{1h}^{(1)} & \cdots & w_{nh}^{(1)} \end{bmatrix}_{h \times n+1}$$

$$W^{(1)}\vec{x} = \begin{bmatrix} \sum_{i=0}^n x_i w_{i1}^{(1)} \\ \sum_{i=0}^n x_i w_{i2}^{(1)} \\ \vdots \\ \sum_{i=0}^n x_i w_{ih}^{(1)} \end{bmatrix} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ \vdots \\ z_h^{(1)} \end{bmatrix} = \vec{z}^{(1)}$$

$$\phi(\vec{z}^{(1)}) = \begin{bmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_h^{(1)} \end{bmatrix} \quad \vec{a}^{(1)} = \begin{bmatrix} 1 \\ a_1^{(1)} \\ \vdots \\ a_h^{(1)} \end{bmatrix} \quad \hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_o \end{bmatrix} = \phi(W^{(2)}\vec{a}^{(1)})$$



# Antrenarea unei rețele feedforward multistrat de neuroni

- O rețea se antrenează folosind algoritmul de backpropagation = propagarea erorii înapoi
- Varianta cea mai folosită este utilizarea unui algoritm stohastic de coborâre pe gradient (stochastic gradient descent)
  - “stochastic” întrucât gradientul se calculează în funcție de un exemplu sau o mulțime redusă de exemple (batch) și nu în raport cu întreaga mulțime
- Trebuie să calculăm gradientul funcției de eroare  $E$  în raport cu fiecare pondere din rețea și să facem actualizările corespunzătoare:

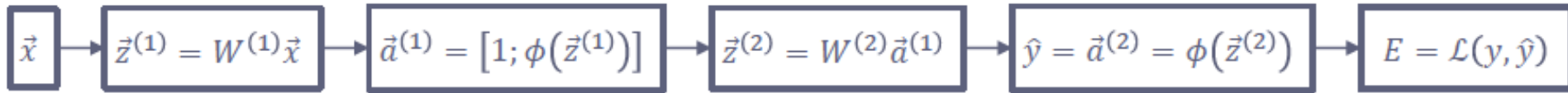
$$E(\vec{x}) = \mathcal{L}(y, \hat{y})$$

$$\Delta w_{ij}^{(l)} = -\eta \frac{\partial E}{\partial w_{ij}^{(l)}}$$

Funcția de eroare  $E$  ia valori mici când cele două etichete iau valori apropiate și ia valori mari altfel

- În format matriceal putem scrie:  $\Delta W^{(l)} = -\eta \frac{\partial E}{\partial W^{(l)}}$

# Regula de înlănțuire a derivatelor



- Trebuie să calculăm gradientul funcției de eroare  $E$  în raport cu fiecare pondere din rețea și să facem actualizările corespunzătoare
- Funcția eroare  $E$  depinde de toate ponderile din rețea:

$$E(\vec{x}) = \mathcal{L}\left(y, \phi\left(W^{(2)}\phi(W^{(1)}\vec{x})\right)\right)$$

- Putem alege orice pondere  $w_{ij}^{(l)}$  și calcula prin regula de înlănțuire derivata ei corepunzătoare  $\frac{\partial E}{\partial w_{ij}^{(l)}}$ .

- În format matriceal putem scrie:


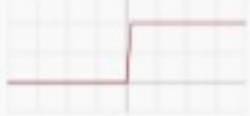


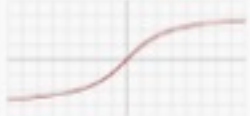


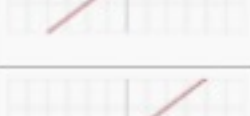

$$\frac{\partial E}{\partial W^{(1)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \vec{z}^{(2)}} \frac{\partial \vec{z}^{(2)}}{\partial \vec{a}^{(1)}} \frac{\partial \vec{a}^{(1)}}{\partial \vec{z}^{(1)}} \frac{\partial \vec{z}^{(1)}}{\partial W^{(1)}}$$

$$\frac{\partial E}{\partial W^{(2)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \vec{z}^{(2)}} \frac{\partial \vec{z}^{(2)}}{\partial W^{(2)}}$$

# Alegerea funcției de activare

- Funcția hardlim (funcția de activare a perceptronului) nu este derivabilă în 0 iar în toate celelalte puncte derivata sa este nulă, deci orice actualizare este nulă.
- Funcții uzuale de activare:
  - funcția identitate  $f(x) = x$ ;
  - funcția logistică  $f(x) = 1/(1+e^{-x})$ ;
  - funcția tangentă hiperbolică  $f(x) = \tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ ;
  - funcția relu  $f(x) = \max(0, x)$ .



Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

# Alegerea funcției de eroare E

- Funcția pătratică de eroare (folosită de Adaline)

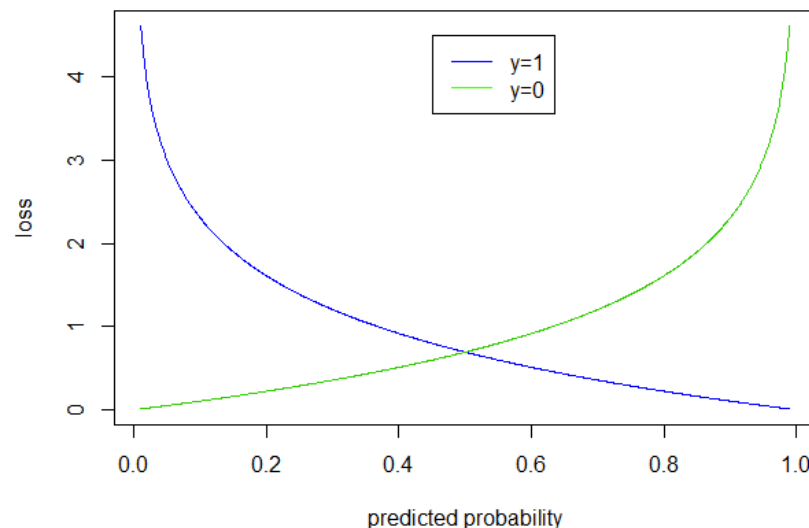
$$\mathcal{L}(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

$$\mathcal{L}'(y, \hat{y}) = -(y - \hat{y})$$

- În practică, se folosește funcția de eroare bazată pe cross-entropie:

$$\mathcal{L}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

$$\mathcal{L}'(y, \hat{y}) = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}}$$



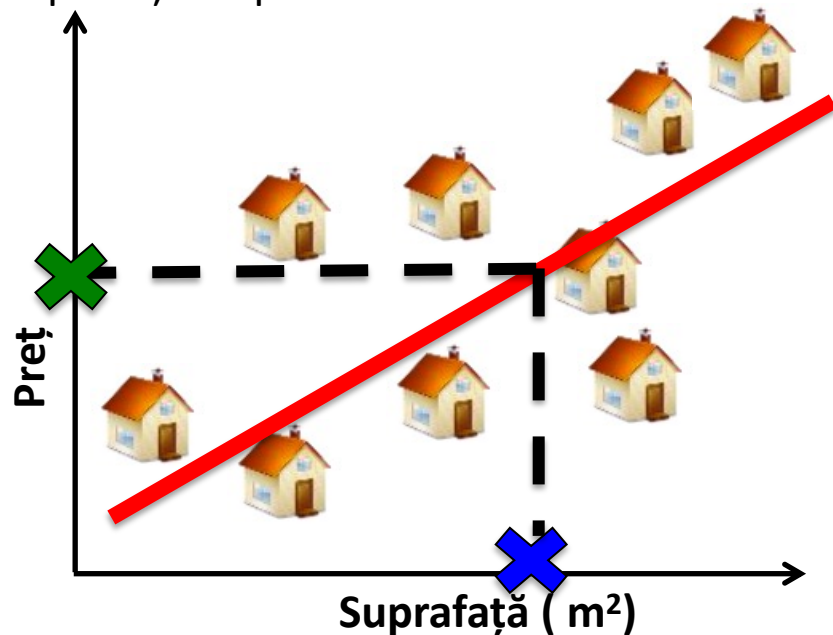
# Regresia liniară simplă și multiplă

# Cursul 2: Învățarea supervizată

- vrem să prezicem o etichetă
  - avem nevoie de date etichetate

**Regresie:** eticheta este o valoare continuă

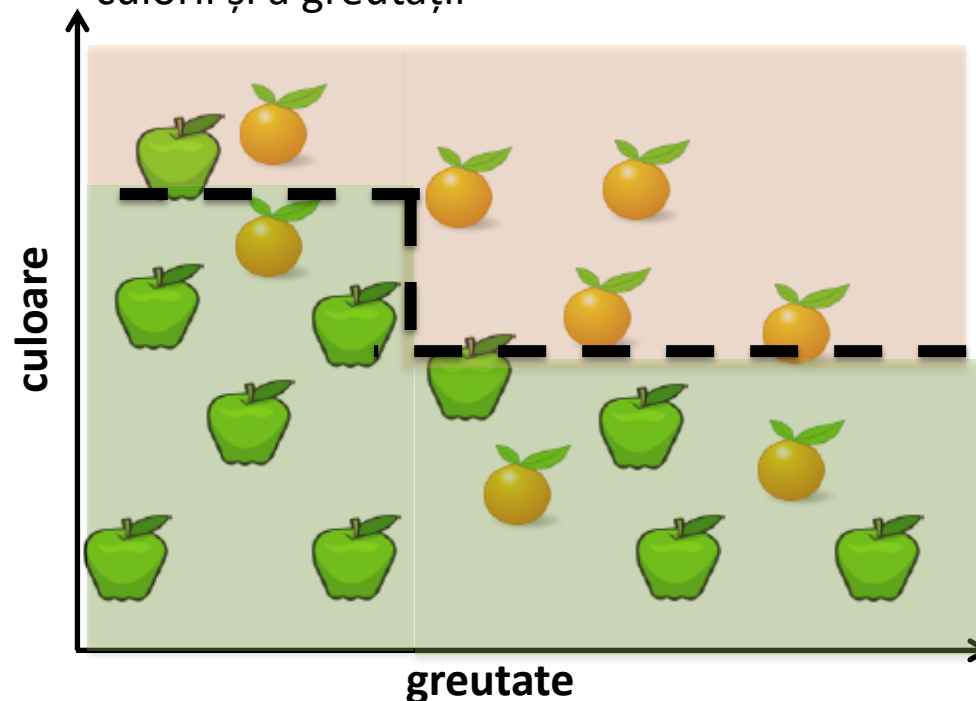
Exemplu: prezicerea prețului unei case pe baza suprafeței exprimată în  $m^2$



Regresie liniară  
Regresie pe baza kNN

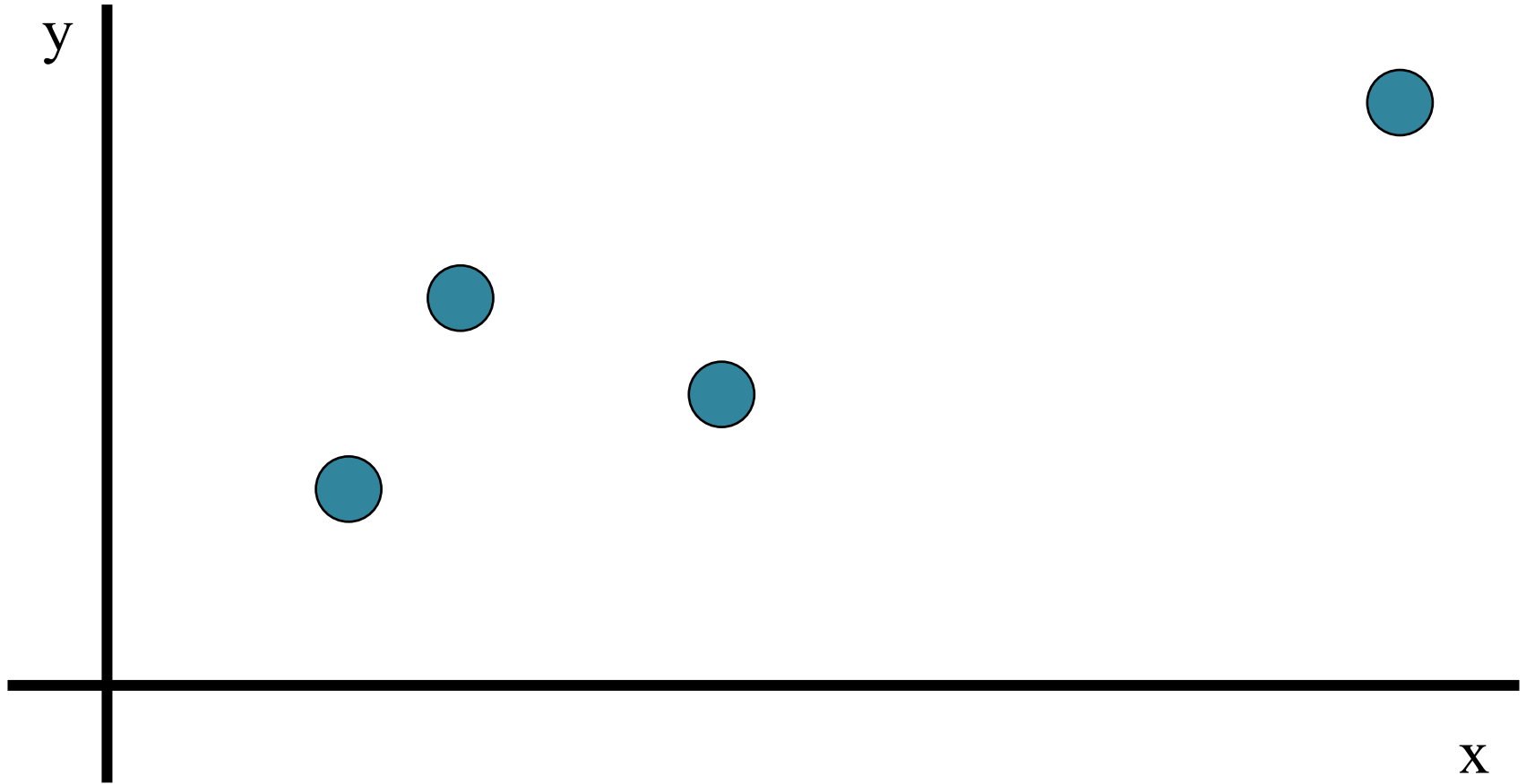
**Clasificare:** eticheta este o valoare discretă

Exemplu: prezicerea tipului de fruct pe baza culorii și a greutății

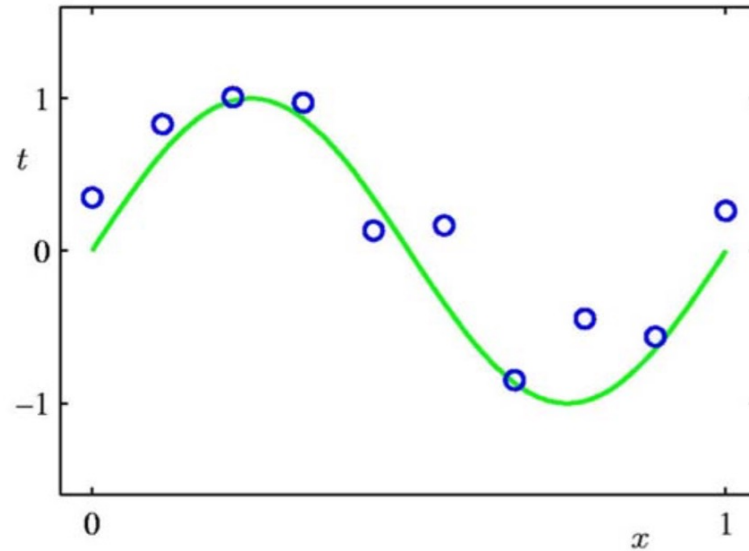


Cei mai apropiați k vecini (kNN),  
SVMs, Rețele neuronale

# Cursul 3: Modelul celor mai apropiați k-vecini pentru probleme de regresie



# Regresie din exemple etichetate



- Presupunem că avem un set de  $N$  exemple de antrenare:

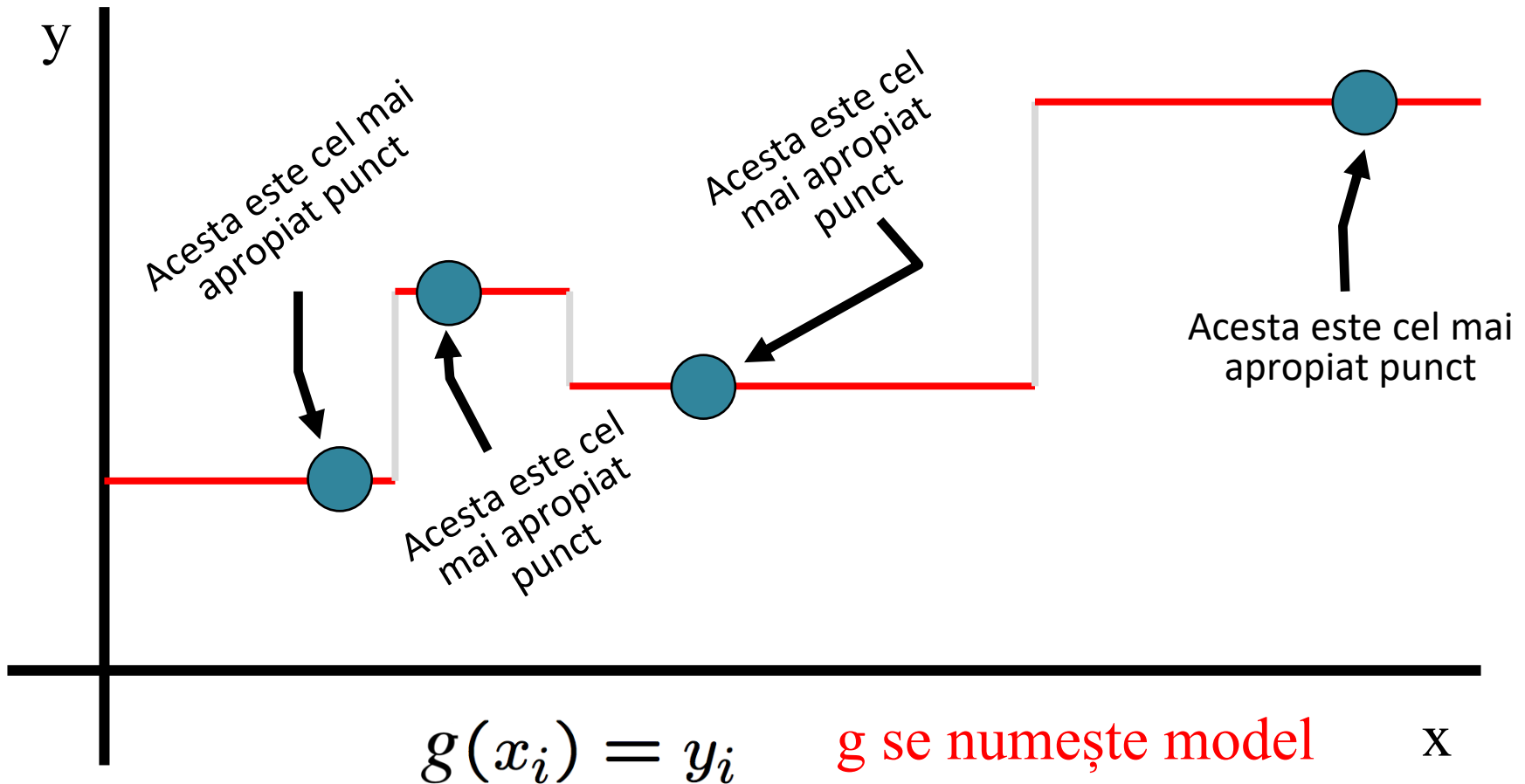
$$(x_1, \dots, x_N) \quad \text{și} \quad (y_1, \dots, y_N), \quad x_i, y_i \in \mathbb{R}$$

- Problema regresiei constă în estimarea funcției  $g(x)$  a.î.:

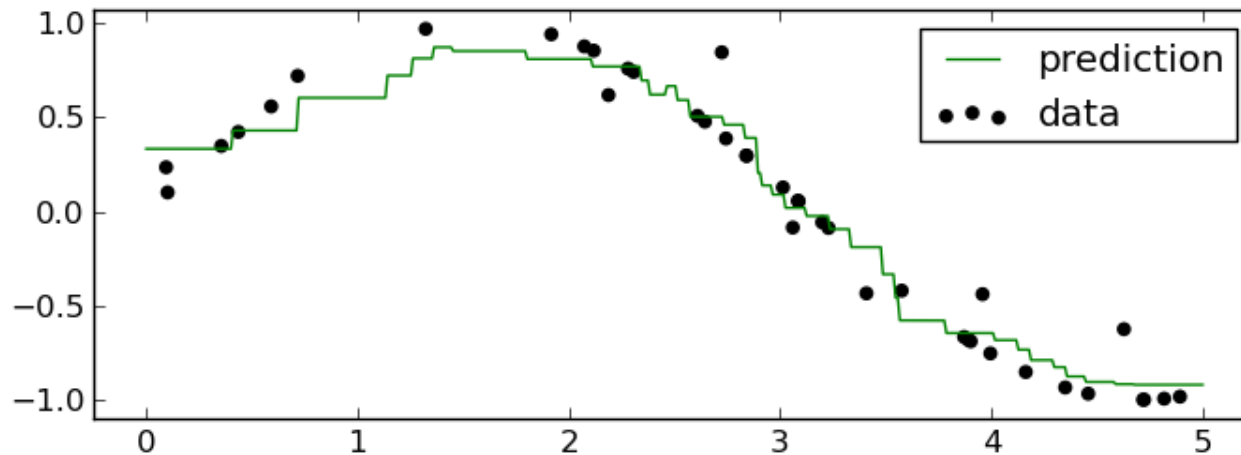
$$g(x_i) = y_i \quad \text{g se numește model}$$

# Modelul celor mai apropiați k-vecini pentru probleme de regresie

**K = 1**



# Modelul celor mai apropiați k-vecini pentru probleme de regresie



Algoritmul de regresie bazat pe cei mai apropiați k-vecini:

- 1) Pentru fiecare exemplu de test  $x$ , găsim cei mai apropiați  $k$  vecini și etichetele lor
- 2) Predicția este media etichetelor celor  $k$  vecini

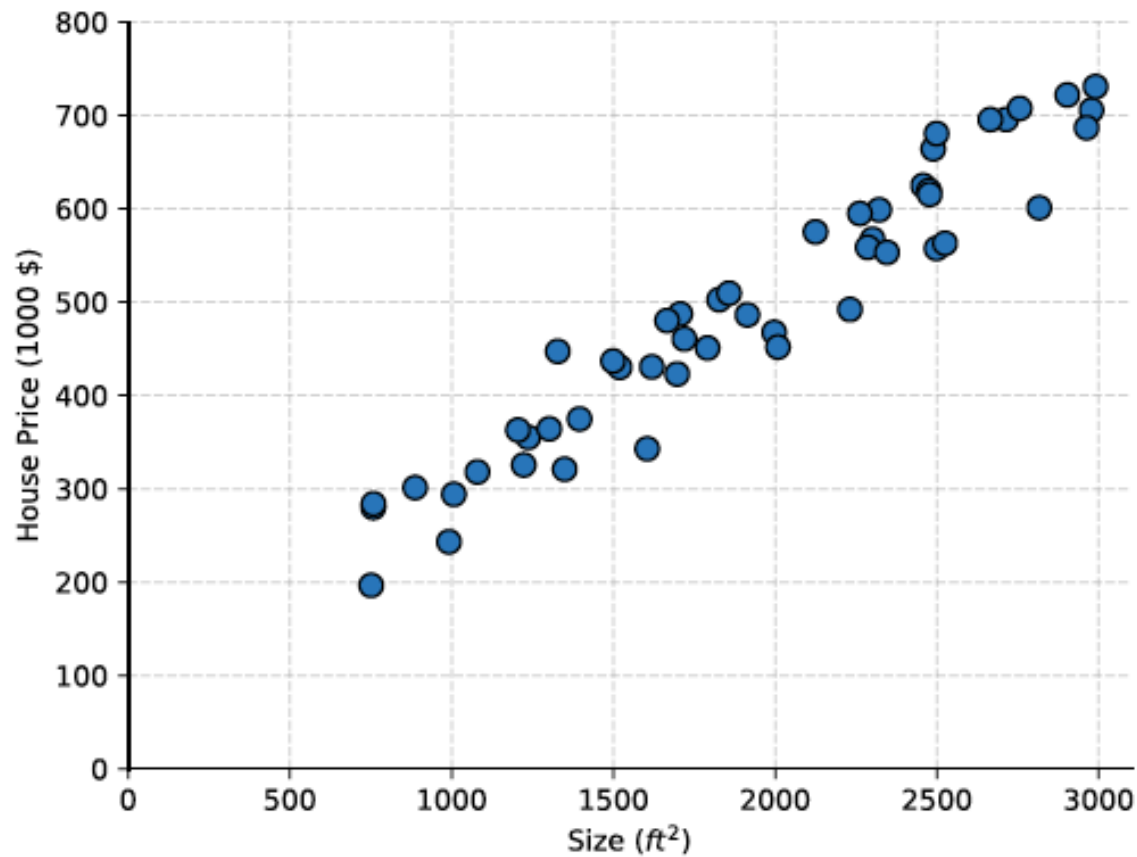
$$f(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^K y_i$$



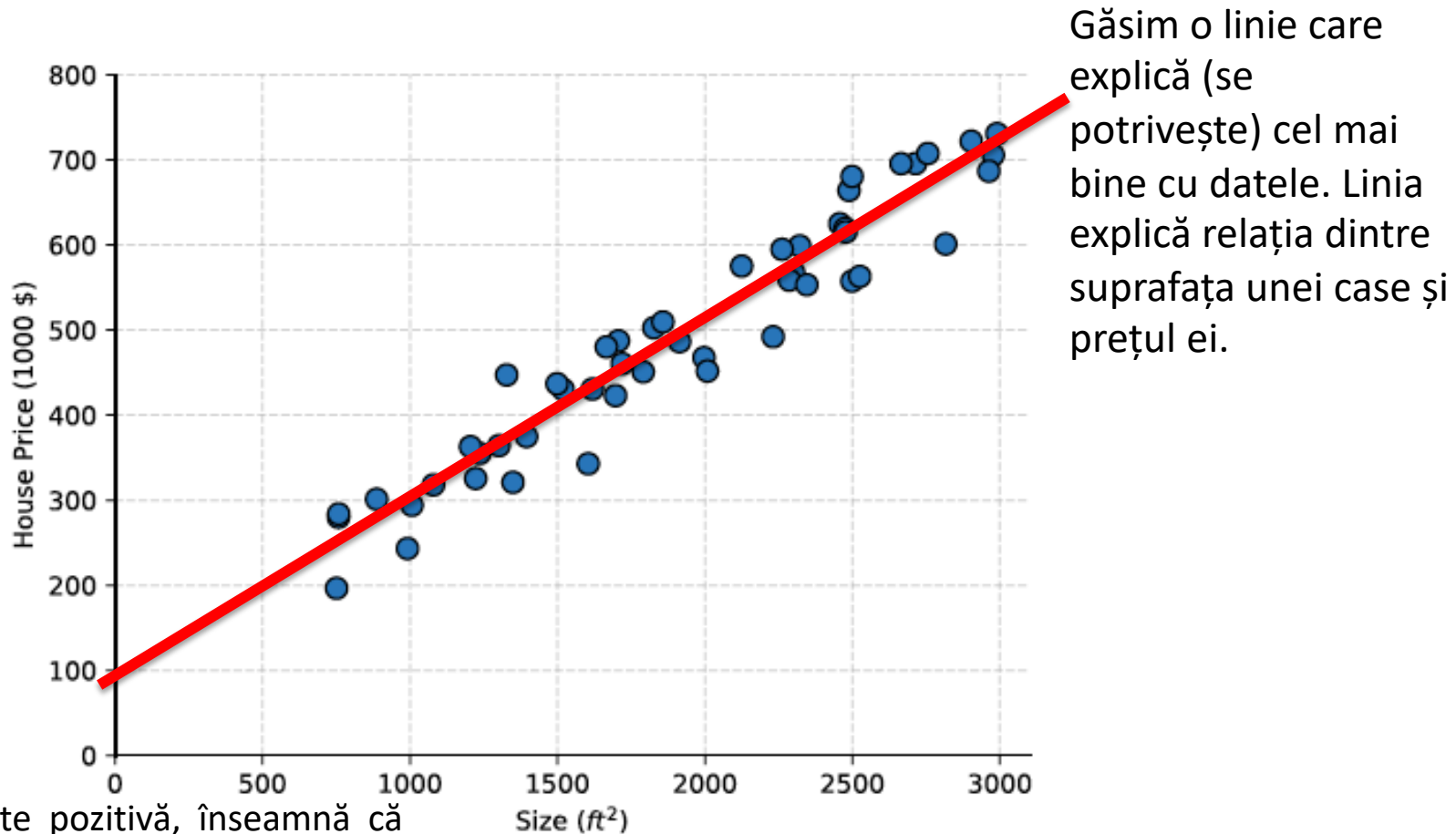
# Regresia liniară - obiective

- Stabilirea unei relații (liniare) între o variabilă dependentă și una sau mai multe variabile independente
  - variabila dependentă se mai numește etichetă, răspuns, output
  - variabilele independente se mai numesc caracteristici, trăsături, attribute, predictor
  - dacă avem o singură variabilă independentă atunci vorbim despre *regresie liniară simplă*
  - dacă avem mai multe variabile independente atunci vorbim despre *regresie liniară multiplă*
- Realizarea de predicții
  - folosește relația liniară obținută mai înainte pentru realizarea de predicții pe date noi

# Regresia liniară simplă

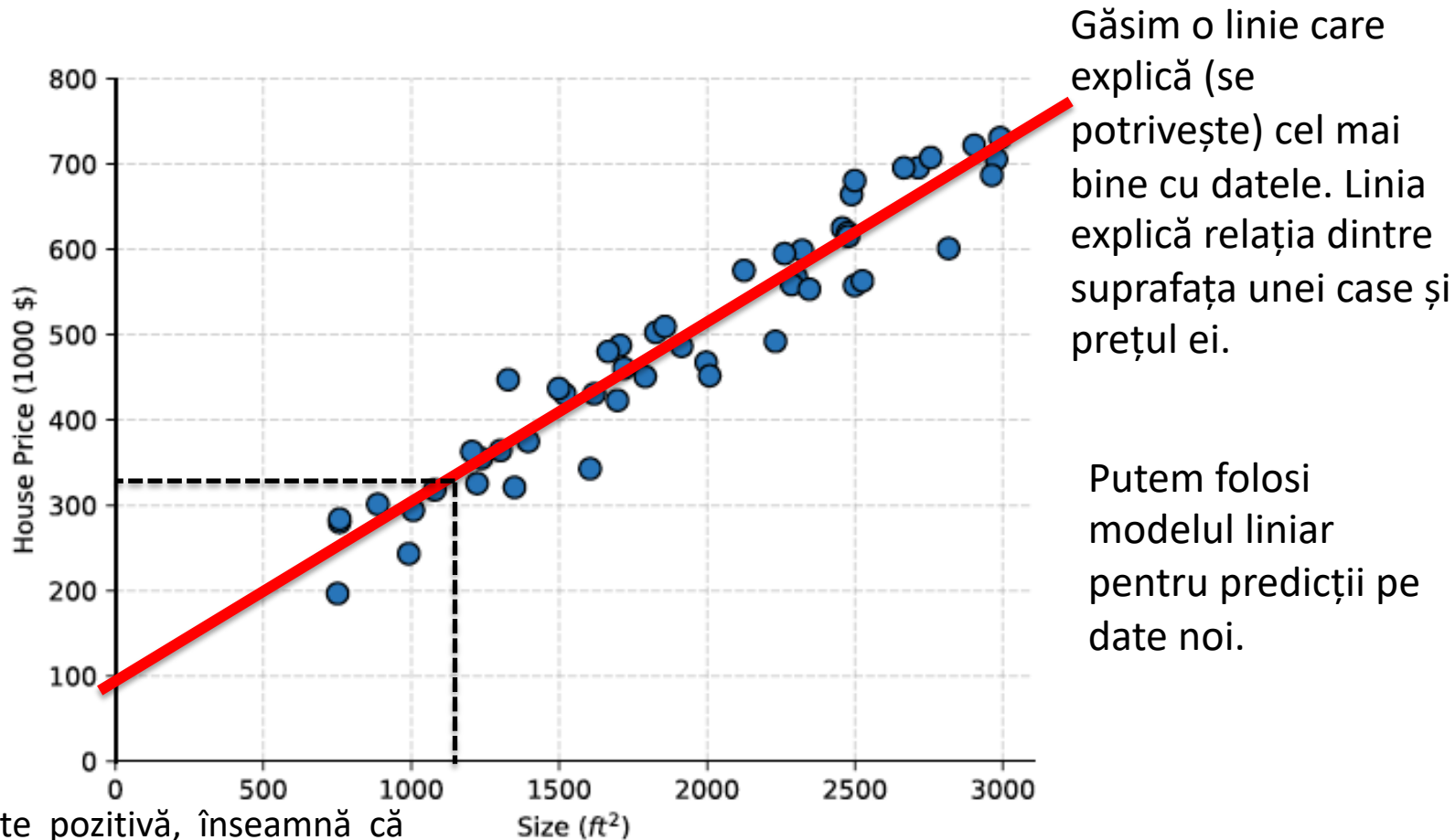


# Regresia liniară simplă



Panta dreptei este pozitivă, înseamnă că avem o corelație pozitivă între suprafață și preț (prețul crește pe măsura ce suprafața crește).

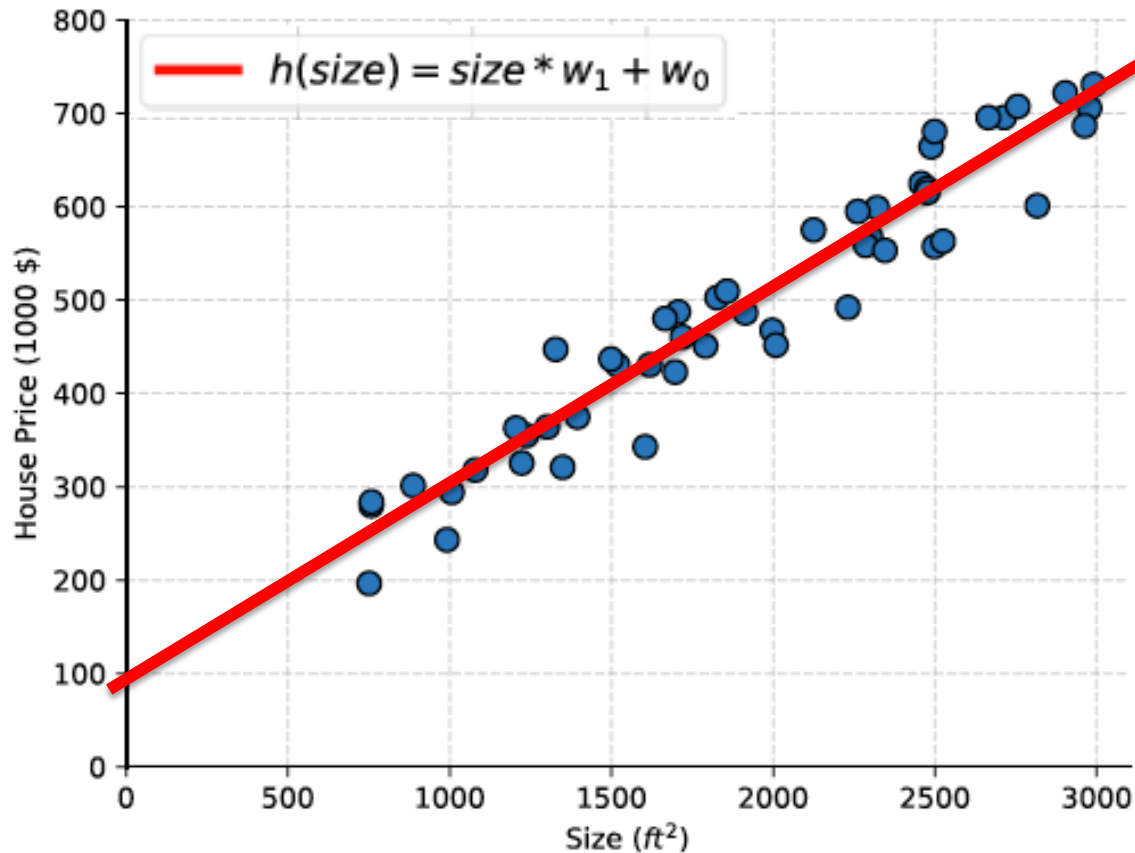
# Regresia liniară simplă



Panta dreptei este pozitivă, înseamnă că avem o corelație pozitivă între suprafață și preț (prețul crește pe măsura ce suprafața crește).

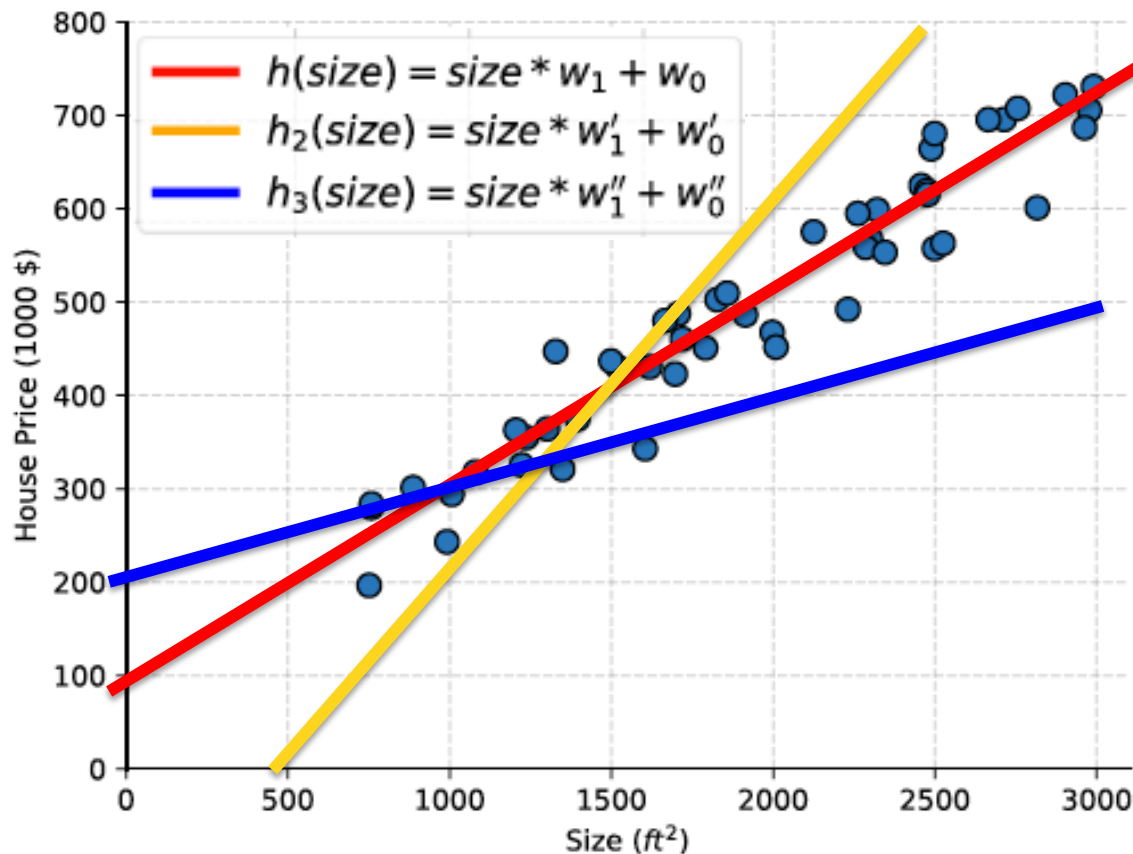
# Regresia liniară simplă

- Relația este modelată de funcție liniară



# Regresia liniară simplă

- Relația este modelată de funcție liniară



Există multe linii  
posibile candidat.

Avem nevoie de o  
metodă care găsește  
cea mai bună linie,  
adică cele mai bune  
valori  $w_0$  și  $w_1$ .

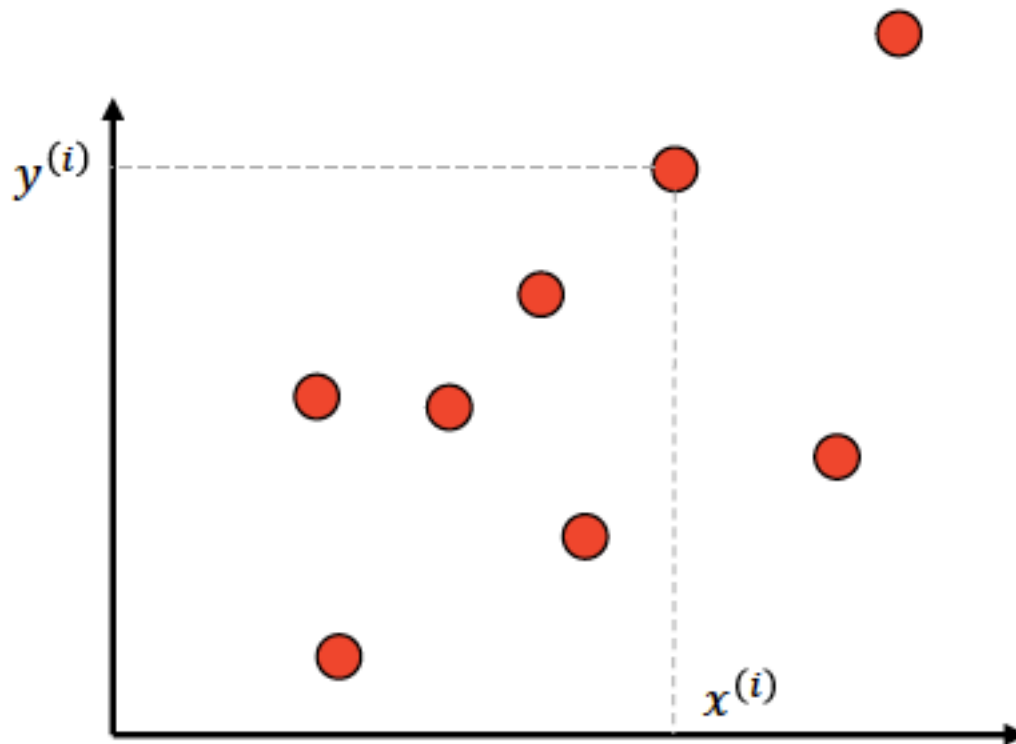
# Notății

- $x \in \mathbb{R}$  este variabila independentă (adică suprafața)
- $y \in \mathbb{R}$  este variabila dependentă (adică prețul)
- $h : \mathbb{R} \rightarrow \mathbb{R}$  este ipoteza (dreapta) pe care o căutăm, are parametri  $w_0$  și  $w_1$
- $\hat{y} = h(x)$  este valoarea prezisă pentru inputul  $x$
- Regresia liniară simplă:  $\hat{y} = h(x) = w_0 + w_1 x$
- Vrem să găsim parametri  $w_0$  și  $w_1$  astfel încât  $\hat{y}$  este cât mai aproape de  $y$

# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

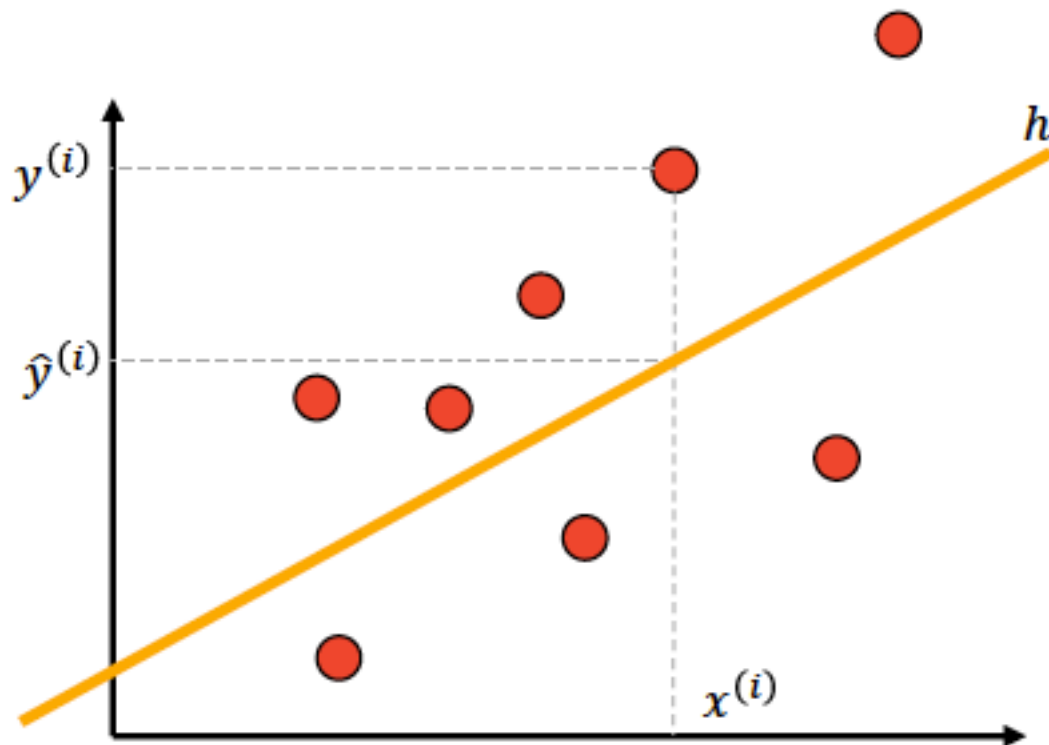




# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$



# Metoda celor mai mici pătrate

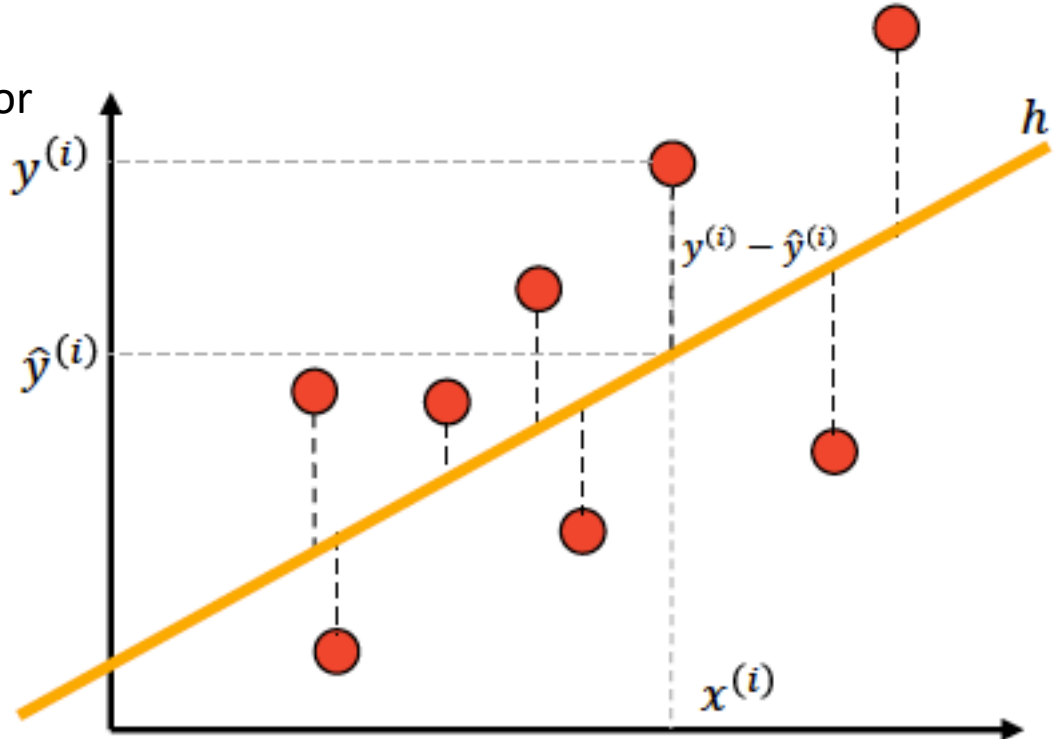
- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

Funcția cost = suma pătratelor erorilor



# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

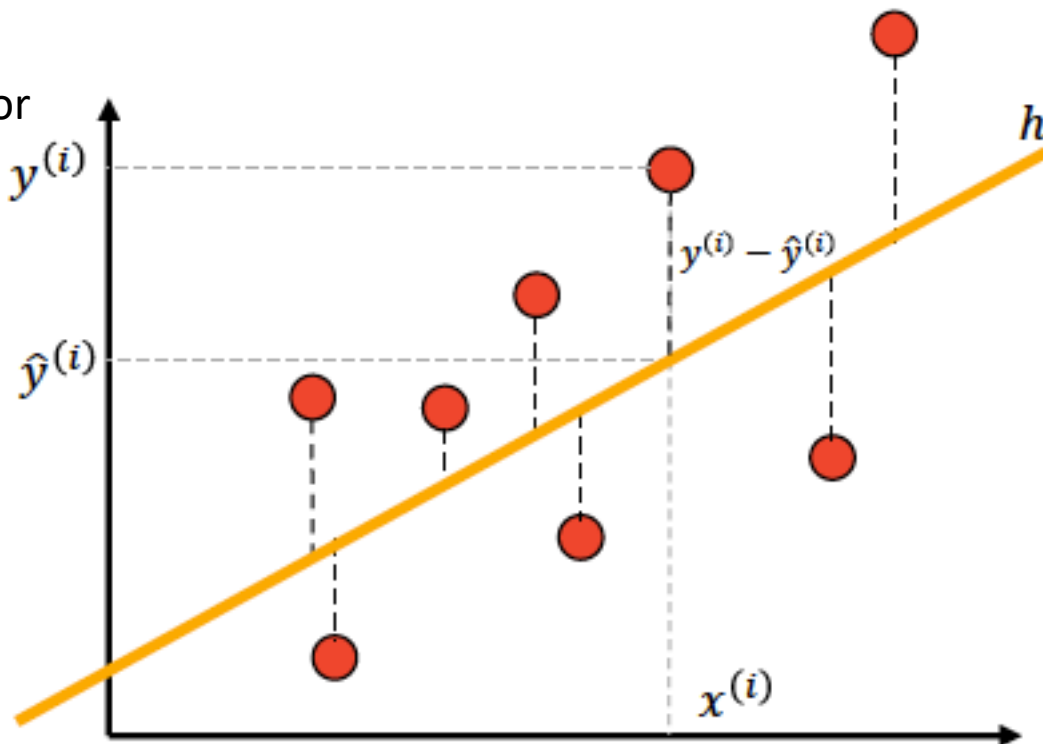
$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

Funcția cost = suma pătratelor erorilor

- Minimizăm  $\mathcal{L}_E$  în raport cu  $w_0$  și  $w_1$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$



# Calculule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

# Calculule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} =$$

# Calculule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) =$$

# Calculule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) = 2 \left( mw_0 - \sum y + w_1 \sum x \right) = 0$$

# Calculule - Metoda celor mai mici pătrate

- Notăm  $x^{(i)} = x$ ,  $y^{(i)} = y$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = \sum (2w_0 - 2y + 2w_1 x) = 2 \left( mw_0 - \sum y + w_1 \sum x \right) = 0$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$



# Calculule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) =$$

# Calculule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

# Calculule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left( \sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

# Calculule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left( \sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2} =$$

# Calculule - Metoda celor mai mici pătrate

$$\mathcal{L}_E = \sum_{i=1}^m (y - w_1 x - w_0)^2 = \sum (y^2 + w_1^2 x^2 + w_0^2 - 2w_1 xy - 2w_0 y + 2w_0 w_1 x)$$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\frac{\partial \mathcal{L}_E}{\partial w_1} = \sum (2w_1 x^2 - 2xy + 2w_0 x) = 2 \left( w_1 \sum x^2 - \sum xy + \frac{1}{m} \left( \sum y - w_1 \sum x \right) \sum x \right) = 0$$

$$w_1 \sum x^2 - w_1 \frac{1}{m} \left( \sum x \right)^2 - \sum xy + \frac{1}{m} \sum x \sum y = 0$$

$$\Rightarrow w_1 = \frac{\sum xy - \frac{1}{m} \sum x \sum y}{\sum x^2 - \frac{1}{m} (\sum x)^2} = \frac{m \sum xy - \sum x \sum y}{m \sum x^2 - (\sum x)^2} = \dots = \frac{\sum [(x - \bar{x})(y - \bar{y})]}{\sum (x - \bar{x})^2}$$

unde

$$\bar{x} = \frac{1}{m} \sum x^{(i)}$$

# Metoda celor mai mici pătrate

- Avem mulțimea de date de antrenare:

$$E = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}, \text{ cu } x^{(i)}, y^{(i)} \in \mathbb{R}$$

$$\mathcal{L}_E = \sum_i (y^{(i)} - \hat{y}^{(i)})^2 = \sum_i (y^{(i)} - w_1 x^{(i)} - w_0)^2$$

Squared loss

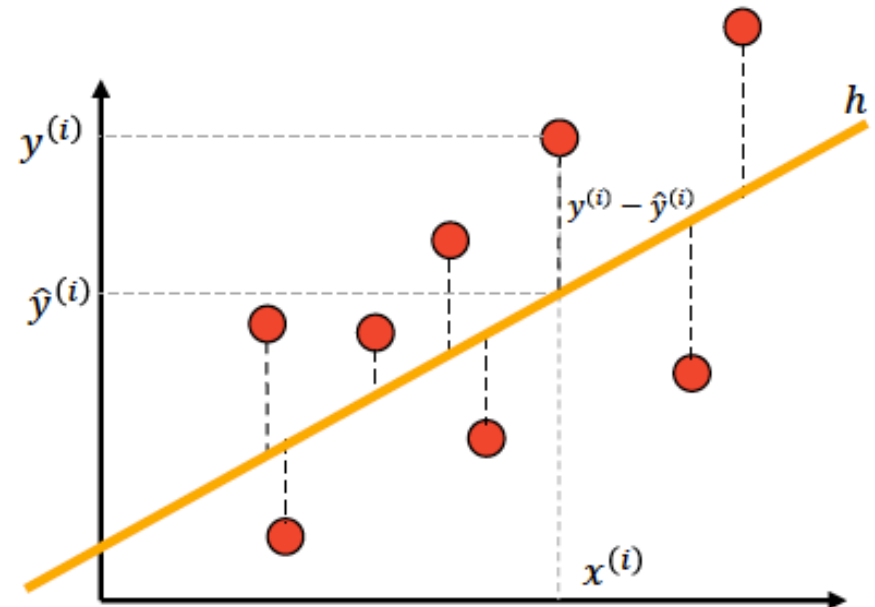
Funcția cost = suma pătratelor erorilor

- Minimizăm  $\mathcal{L}_E$  în raport cu  $w_0$  și  $w_1$

$$\frac{\partial \mathcal{L}_E}{\partial w_0} = 0, \quad \frac{\partial \mathcal{L}_E}{\partial w_1} = 0$$

$$w_1 = \frac{\sum_i [(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})]}{\sum_i (x^{(i)} - \bar{x})^2}$$

$$w_0 = \frac{1}{m} \left( \sum_i y^{(i)} - w_1 \sum_i x^{(i)} \right)$$



# Metoda celor mai mici pătrate

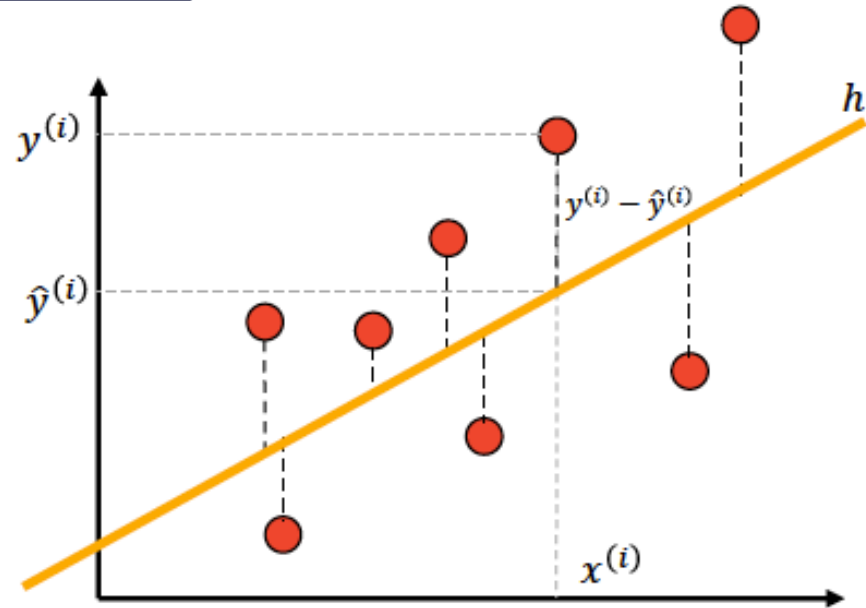
- Observație: dreapta găsită pe baza celor mai mici pătrate trece prin punctul de coordonate  $(\bar{x}, \bar{y})$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right)$$

$$\bar{x} = \frac{1}{m} \sum x^{(i)}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$\bar{y} = w_0 + w_1 \bar{x}$$



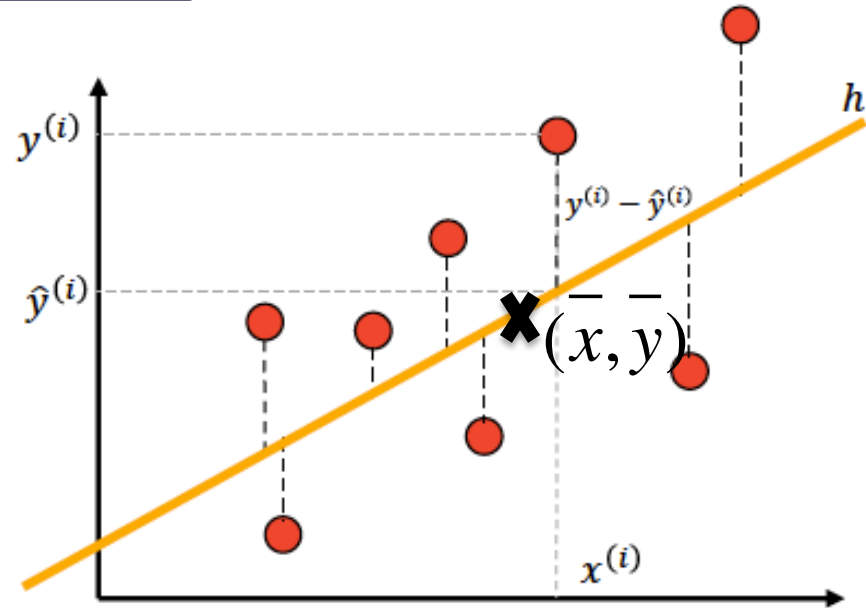
# Metoda celor mai mici pătrate

- Observație: dreapta găsită pe baza celor mai mici pătrate trece prin punctul de coordonate  $(\bar{x}, \bar{y})$

$$w_0 = \frac{1}{m} \left( \sum y - w_1 \sum x \right) \quad \bar{x} = \frac{1}{m} \sum x^{(i)}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

$$\bar{y} = w_0 + w_1 \bar{x}$$





# Regresia liniară multiplă

- $\vec{x} \in \mathbb{R}^n = [x_1 \ x_2 \ \dots \ x_n],$
- $\vec{w} \in \mathbb{R}^n = [w_1 \ w_2 \ \dots \ w_n], w_0 \in \mathbb{R}$

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n = w_0 + \langle \vec{w}, \vec{x} \rangle$$

# Regresia liniară multiplă

- $\vec{x} \in \mathbb{R}^n = [x_1 \ x_2 \ \dots \ x_n],$
- $\vec{w} \in \mathbb{R}^n = [w_1 \ w_2 \ \dots \ w_n], w_0 \in \mathbb{R}$

$$\hat{y} = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n = w_0 + \langle \vec{w}, \vec{x} \rangle$$

- Înlocuim termenul liber  $w_0$  (bias-ul) făcând notațiile:

$$\vec{x} \in \mathbb{R}^{n+1} = [1 \ x_1 \ x_2 \ \dots \ x_n]$$

$$\vec{w} \in \mathbb{R}^{n+1} = [w_0 \ w_1 \ w_2 \ \dots \ w_n]$$

$\Rightarrow$

$$\hat{y} = \langle \vec{w}, \vec{x} \rangle$$

# Regresia liniară multiplă

- Avem mulțimea de date de date de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- Folosim înmulțirea matricelor pentru a calcula predicțiile:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$$

# Regresia liniară multiplă

- Avem mulțimea de date de date de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- Folosim înmulțirea matricelor pentru a calcula predicțiile:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$$

# Regresia liniară multiplă

- Avem mulțimea de date de antrenare:

$$E = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(m)}, y^{(m)})\}, \vec{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \mathbb{R}$$

$$\hat{y}^{(i)} = \langle \vec{w}, \vec{x}^{(i)} \rangle = w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

- Folosim înmulțirea matricelor pentru a calcula predicțiile:

$$\begin{pmatrix} \hat{y}^{(1)} \\ \hat{y}^{(2)} \\ \vdots \\ \hat{y}^{(m)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} \Rightarrow \hat{\mathbf{Y}} = \mathbf{X}\mathbf{w}$$

$$\mathcal{L}_E = \sum_{i=1}^m (y^{(i)} - \hat{y}^{(i)})^2 = (\mathbf{Y} - \hat{\mathbf{Y}})^T (\mathbf{Y} - \hat{\mathbf{Y}}) \stackrel{\text{not}}{=} (\mathbf{Y} - \hat{\mathbf{Y}})^2$$

# Regresia liniară multiplă

$$\mathcal{L}_E = (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T (Y - Xw)$$

# Regresia liniară multiplă

$$\begin{aligned}\mathcal{L}_E &= (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T (Y - Xw) \\ &= (Y^T - w^T X^T)(Y - Xw)\end{aligned}$$

# Regresia liniară multiplă

$$\begin{aligned}\mathcal{L}_E &= (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T(Y - Xw) \\ &= (Y^T - w^T X^T)(Y - Xw) \\ &= Y^T Y - Y^T Xw - w^T X^T Y + w^T X^T Xw = Y^T Y - 2w^T X^T Y + w^T X^T Xw\end{aligned}$$



# Regresia liniară multiplă

$$\begin{aligned}\mathcal{L}_E &= (Y - \hat{Y})^2 = (Y - Xw)^2 = (Y - Xw)^T(Y - Xw) \\ &= (Y^T - w^T X^T)(Y - Xw) \\ &= Y^T Y - Y^T Xw - w^T X^T Y + w^T X^T Xw = Y^T Y - 2w^T X^T Y + w^T X^T Xw\end{aligned}$$

- $Y^T Xw = [\cdot]_{1 \times m} [\cdot]_{m \times n} [\cdot]_{n \times 1} = [\cdot]_{1 \times 1}$  (scalar)
- $Y^T Xw = (w^T X^T Y)^T$

Un scalar (un număr) este o matrice 1 x 1

Transpusa unui scalar este același scalar

# Regresia liniară multiplă

$$\mathcal{L}_E = Y^T Y - 2w^T X^T Y + w^T X^T X w$$

- Minimizăm  $\mathcal{L}_E$  în raport cu  $w = (w_0, w_1, \dots, w_n)$ , toate derivatele parțiale sunt 0:

$$\frac{\partial \mathcal{L}_E}{\partial w} = 0 \Rightarrow -2X^T Y + 2X^T X w = 0 \Rightarrow X^T X w = X^T Y \Rightarrow$$

$$w = (X^T X)^{-1} X^T Y$$

- Matricea  $X^T X$  trebuie să fie inversabilă