

3. Regresia liniară

Reprezentarea modelului de regresie liniară
Funcția de cost pătratică
Algoritmul de minimizare al gradientului

În această lucrare de laborator vom crea un algoritm de predicție pentru prețul unei case, dându-se ca și caracteristică de intrare prețul terenului. Posibile exemple de perechi suprafață teren - preț casă, ce compun baza de date de antrenare (training set), sunt redată în tabelul 3.1.

Suprafață teren [mp]	Preț [1000 EUR]
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Tab. 3.1 Exemple de perechi de antrenare suprafață teren - preț casă.

Distribuția prețurilor caselor, în funcție de suprafața terenului, este redată în figura 3.1.

3.1 Reprezentarea modelului

În cele ce urmează, vom utiliza următoarele notații:

- $x^{(i)}$: vectorul caracteristicilor de intrare (features); în cazul exemplului curent caracteristica de intrare este suprafața terenului;
- $y^{(i)}$: valoarea de ieșire - prețul casei;
- $(x^{(i)}, y^{(i)})$: exemplu de antrenare;
- $(x^{(i)}, y^{(i)}); i = 1, \dots, m$: baza de date de antrenare (training set), compusă din m exemple;
- i : index către elementele din training set;
- \mathbb{X} : spațiul caracteristicilor de intrare;
- \mathbb{Y} : spațiul caracteristicilor de ieșire;

În exemplul curent, atât spațiul caracteristicilor de intrare, cât și spațiul caracteristicilor de ieșire aparțin mulțimii numerelor reale: $\mathbb{X} = \mathbb{Y} = \mathbb{R}$.

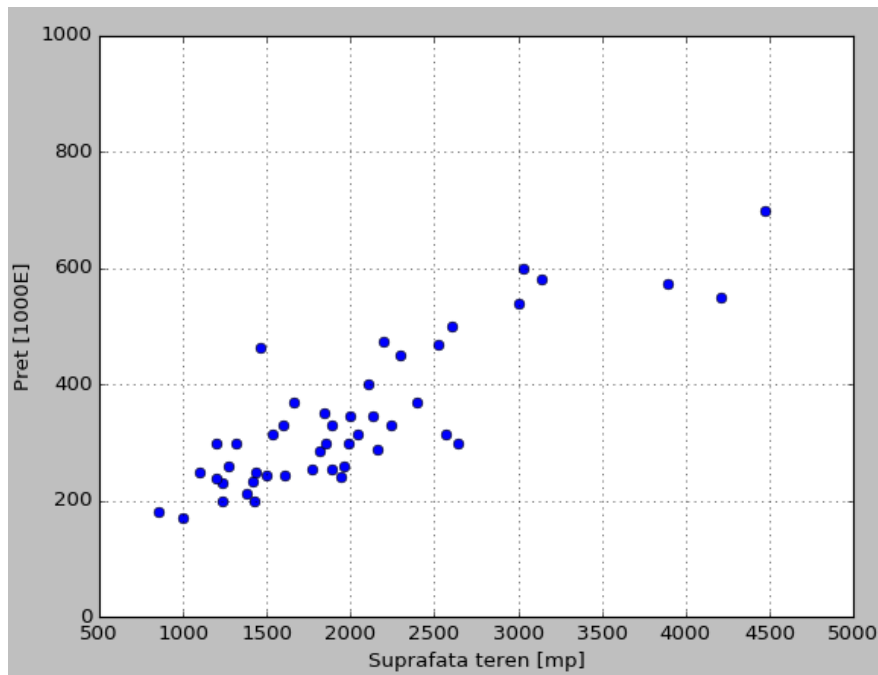


Fig. 3.1 Distribuția perechilor suprafață teren - preț casă din baza de date de antrenare.

Obiectivul algoritmului de învățare este de a învăța funcția $h_{\theta}(x) : \mathbb{X} \rightarrow \mathbb{Y}$, în așa fel încât ipoteza $h_{\theta}(x)$ să reprezinte un predictor "optim" pentru valoarea corespondentă a lui y [1]. Coeficienții θ , denumiți *parametrii modelului*, sunt valorile ce trebuie învățate prin algoritmul de învățare.

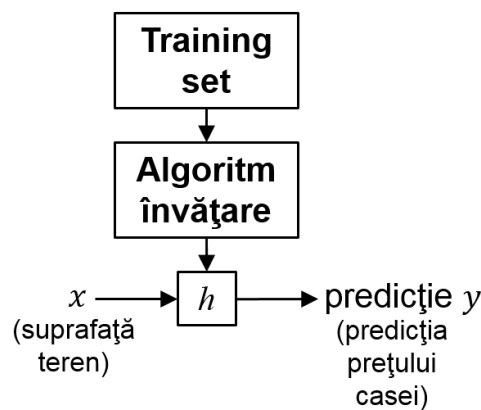


Fig. 3.2 Schema bloc al unui algoritm de învățare.

Atunci când variabila pe care dorim să o prezicem este continuă ($y \in \mathbb{R}$), cum este cazul exemplului de față, problema de învățare este denumită *regresie*. În cazul în care y poate lua doar câteva valori discrete ($y \in \{0, 1, \dots, k\}$) (de exemplu, atunci când dorim să prezicem dacă proprietatea este o casă $y = 0$ sau un apartament $y = 1$), problema de învățare se numește *clasificare*.

Pentru exemplul curent vom folosi o funcție de aproximare liniară:

$$h_{\theta}(x) = \theta_0 + \theta_1 x, \quad (3.1)$$

unde x reprezintă suprafața terenului.

În cazul în care, pe lângă suprafața terenului, dispunem de caracteristici (features) adiționale ce pot ajuta la predicția prețului (de exemplu, numărul de dormitoare), funcția de aproximare poate fi rescrisă astfel:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2, \quad (3.2)$$

unde $x = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}$ este în acest caz un vector de caracteristici compus din x_1 (este suprafața terenului) și x_2 (numărul de dormitoare), iar $x_0 = 1$.

Ecuția 3.2 a modelului poate fi generalizată sub forma:

$$h_{\theta}(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x, \quad (3.3)$$

unde n este numărul de caracteristici (features).

3.2 Funcția de cost

Acuratețea funcției de aproximare $h_{\theta}(x)$ se măsoară cu ajutorul unei funcții de cost $J(\theta)$. Aceasta ia în considerare diferența medie dintre perechile caracteristici de intrare x - valoarea y a ieșirii. În cazul modelului 3.1 utilizat în această lucrare de laborator, funcția de cost are forma:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2, \quad (3.4)$$

unde m este numărul de exemple de antrenare, iar $h_{\theta}(x_i) - y_i$ reprezintă diferența dintre valoarea prezisă $h_{\theta}(x_i)$ și valoarea reală y_i . Datorită exponentului, funcția poartă denumirea de *funcție de eroare pătratică*, sau *funcție de eroare medie*. Cu alte cuvinte, $J(\theta)$ exprimă cât de bine sunt approximate exemplele de antrenare de către $h_{\theta}(x)$, așa cum este ilustrat în figura 3.3.

Obiectivul algoritmului de învățare este de a determina parametrii θ_0 și θ_1 , în așa fel încât $h_{\theta}(x)$ să fie cât mai aproape de y pentru exemplele de antrenare (x, y) . Acest lucru se întâmplă atunci când funcția de cost are valoarea minimă:

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1) = \min_{\theta_0, \theta_1} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2. \quad (3.5)$$

3.3 Algoritm de minimizare al gradientului

În această fază avem definit modelul ipotezei și o modalitate de măsurare a gradului de potrivire a modelului pe datele de antrenare. În următorul pas vom estima parametrii modelului utilizând algoritmul de minimizare al gradientului.

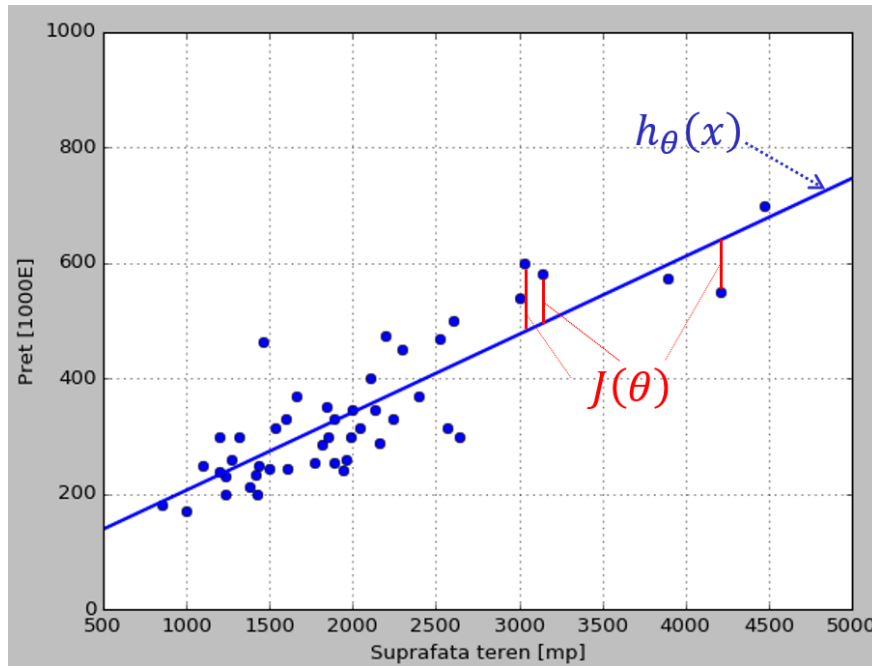


Fig. 3.3 Diferențele dintre ipoteza $h_{\theta}(x)$ și valoarea reală a lui y , exprimate prin intermediul funcției de cost $J(\theta)$.

În figura 3.4 este redată o posibilă formă a funcției de cost $J(\theta)$, în funcție de parametrii θ_0 și θ_1 . Coeficienții θ_0 și θ_1 se află pe axele x și y , iar valoarea lui $J(\theta)$ pe axa z . Fiecare poziție 3D din forma funcției reprezintă valoarea lui $J(\theta)$ pentru o combinație de valori θ_0 și θ_1 . Două perechi de parametri optimi se găsesc la cele două minime ale funcției, indicate de săgețile roșii.

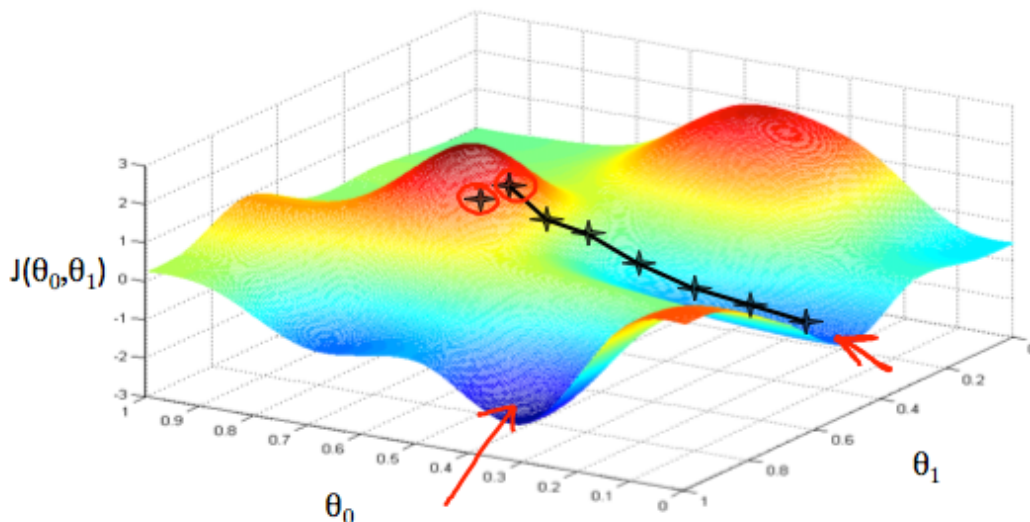


Fig. 3.4 Posibilă formă a funcției de cost $J(\theta)$.

Modalitatea de determinare a minimului lui $J(\theta)$ se bazează pe calculul derivatei funcției de cost (linia tangentă la valoarea funcției). Panta tangentei este derivata la punctul respectiv, ce ne va indica direcția de deplasare către minimul funcției. Algoritmul va efectua pași

către valori inferioare ale funcției de cost prin coborâri cât mai abrupte. Mărimea fiecărui pas este determinată de parametrul α , denumit *rată de învățare* (eng. learning rate).

Spre exemplu, distanța dintre fiecare "stea" din graficul de mai sus reprezintă un pas determinat de parametrul α . Un α de valoare mică va produce pași de dimensiune mai mică. Analog, o valoare mare pentru α va produce pași de dimensiuni mai mari. Direcția în care va fi efectuat un pas este dată de derivata parțială a lui $J(\theta_0, \theta_1)$. În funcție de poziția din graf de unde se pornește căutarea minimumului, putem ajunge la diferite puncte de minim. În figura 3.4 sunt redate două asemenea puncte de minim.

Pseudocodul algoritmului de minimizare al gradientului are următoarea formă:

$$\begin{aligned} &\text{Repetă până la convergență: } \{ \\ &\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\ &\quad \} \end{aligned}$$

unde $j = 0, 1$ reprezintă indexul caracteristicilor. După fiecare iterație a algoritmului, parametrii θ trebuie actualizați simultan.

Următorul pas în estimarea parametrilor θ este calculul derivatei lui $J(\theta_0, \theta_1)$, dată în relația următoare pentru un singur exemplu de antrenare:

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2 \quad (3.6)$$

$$= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \quad (3.7)$$

$$= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^n \theta_i x_i - y \right) \quad (3.8)$$

$$= (h_\theta(x) - y) x_j. \quad (3.9)$$

Astfel, ecuația 3.6 devine:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=0}^m ((h_\theta(x) - y) x_j). \quad (3.10)$$

3.4 Cerințe

Să se scrie un script Python ce determină vârsta unei persoane pe baza înălțimii sale.

3.4.1 Baza de data de antrenare

Pentru a implementa algoritmul, descărcați datele din arhiva "lab3Data.zip".

Fișierele conțin câteva exemple de antrenare cu înălțimi pentru diferite persoane cu vârste cuprinse între doi și opt ani. Valorile y sunt înălțimile măsurate în metrii, iar valorile x sunt vârstele persoanelor pentru înălțimile respective.

Fiecare pereche înălțime-vârstă constituie un exemplu de antrenare $(x^{(i)}, y^{(i)})$ din baza de date compusă din $m = 50$ exemple de antrenare).

Bibliografie

- [1] A. Ng, "Stanford cs229 - machine learning," 2008.