

Szoftver mély neuronhálók alkalmazásához

3. előadás

Kovács Bálint, Varga Viktor
ELTE IK Mesterséges Intelligencia Tanszék

Jövő héttől: Canvas kvízek

- Minden előadás előtt (a 4. előadástól).
- Az előadás órarendi időpontját megelőző 24 órában tölthető ki.
- Elérhető hétfő 18:15-ig.
- Darabonként 1-1 pont szerezhető.

Előző órán - Felügyelt tanulás

Adott: A tanítóminta (training set), input-címke párok halmaza

$$\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$$

$$x \in X \subset \mathbb{R}^n, y \in Y \subset \mathbb{R}^k$$

Feladat: A címke (az elvárt output) minél jobb becslése az inputból.

Azaz, keresünk olyan h_θ függvényt (hipotézisfüggvényt), melyre:

$$h_\theta(x) = \hat{y} \approx y$$

Előző órán - Regresszió

Regresszió: Folytonos értékű címke becslése

$$|Y| = \infty$$

Példa: Autók számának, vagy életkor becslése

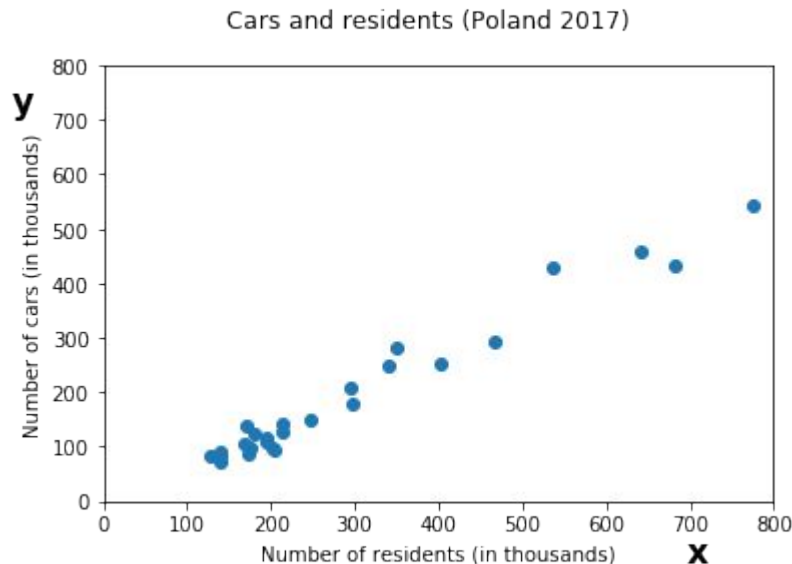
Legegyszerűbb módszer: lineáris regresszió

Előző órán - Regresszió

Példa: Becsüljük meg az autók számát egy adott városban, ha ismerjük a város lakosságának számát.

x: Egy adott város lakosságának száma

y: Egy adott városban megtalálható autók száma

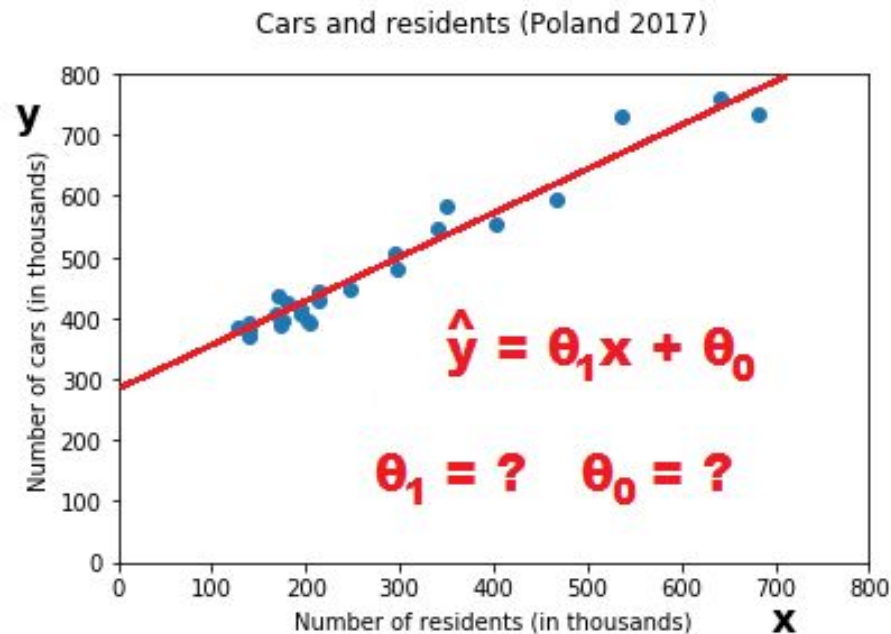


$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

Előző órán - Lineáris regresszió

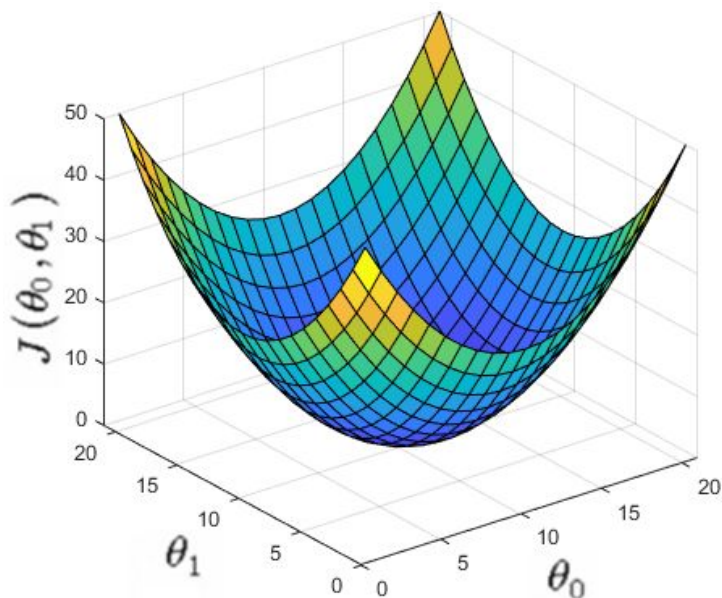
h hipotézisfüggvény: $y \approx \hat{y} = h(x) = \theta_1 x + \theta_0$

$$\theta_1, \theta_0 \in \mathbb{R}$$



Előző órán - Lineáris regresszió megoldás

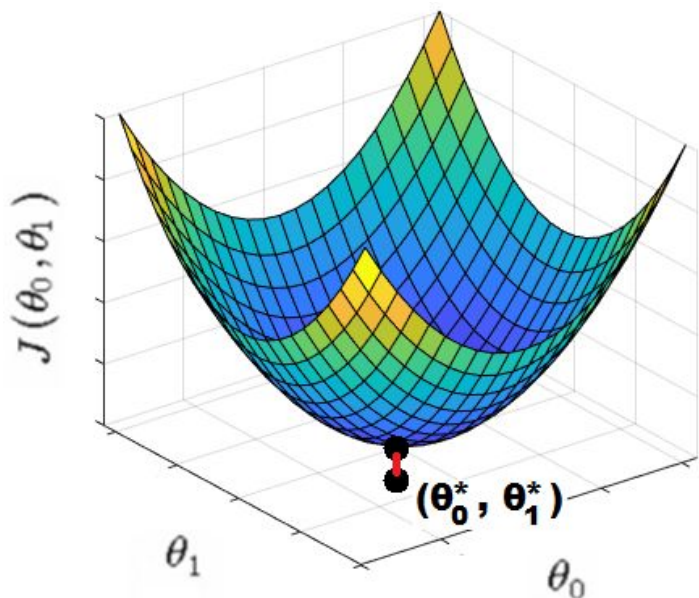
J költségfüggvény: kvadratikusan, kétváltozós (θ_0, θ_1): elliptikus paraboloid



$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^m \overbrace{(\theta_1 x^{(j)} + \theta_0 - y^{(j)})^2}^{h(x^{(j)}) = \hat{y}^{(j)}}$$

Előző órán - Lineáris regresszió megoldás

Feladat: Találjuk meg azokat a paramétereket a hipotézisfüggvényhez, melyekkel az minimális költségű!



$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^m (\theta_1 x^{(j)} + \theta_0 - y^{(j)})^2$$

$$\theta_0^*, \theta_1^* = \underset{\theta_0, \theta_1}{\operatorname{argmin}} J(\theta_0, \theta_1)$$

Előző órán - Lineáris regresszió megoldás

Az optimális paramétereket **gradiens módszerrel** keressük meg.

Gradiens módszer: iteratívan módosítsuk a paramétereket abba az irányba, amerre a költségfüggvény a leginkább lejt.

A legnagyobb meredekségű irányt a **gradiens vektor** adja meg.

$$\nabla J(\theta'_0, \theta'_1) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} J(\theta'_0, \theta'_1) \\ \frac{\partial}{\partial \theta_1} J(\theta'_0, \theta'_1) \end{bmatrix}$$

Előző órán - Lineáris regresszió megoldás

Az optimális paramétereket **gradiens módszerrel** keressük meg.

Gradiens módszer: iteratívan módosítsuk a paramétereket abba az irányba, amerre a költségfüggvény a leginkább lejt.

A legnagyobb meredekségű irányt a **gradiens vektor** adja meg.

$$\nabla J(\theta'_0, \theta'_1) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} J(\theta'_0, \theta'_1) \\ \frac{\partial}{\partial \theta_1} J(\theta'_0, \theta'_1) \end{bmatrix}$$

A J költségfv. parciális deriváltja **θ_0** szerint a paraméterek aktuális értékeinek pontjában

A J költségfv. parciális deriváltja **θ_1** szerint a paraméterek aktuális értékeinek pontjában

Előző órán - Lineáris regresszió megoldás

A költségfüggvény parciális deriváltjai:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^m (\theta_1 x^{(j)} + \theta_0 - y^{(j)})^2$$

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^m (\theta_1 x^{(j)} + \theta_0 - y^{(j)})$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^m (\theta_1 x^{(j)} + \theta_0 - y^{(j)}) \cdot x^{(j)}$$

A parciális deriváláskor a függvényt az egyik változó szerint deriváljuk. Ekkor a többi változót **konstansként kezeljük**.

érintő meredeksége **θ_0** irányban

érintő meredeksége **θ_1** irányban

Előző órán - Lineáris regresszió megoldás

Gradiensmódszer algoritmus két paraméter esetén:

repeat until convergence {

$$grad_0 := \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$grad_1 := \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \theta_0 - \alpha \cdot grad_0$$

$$\theta_1 := \theta_1 - \alpha \cdot grad_1$$

}

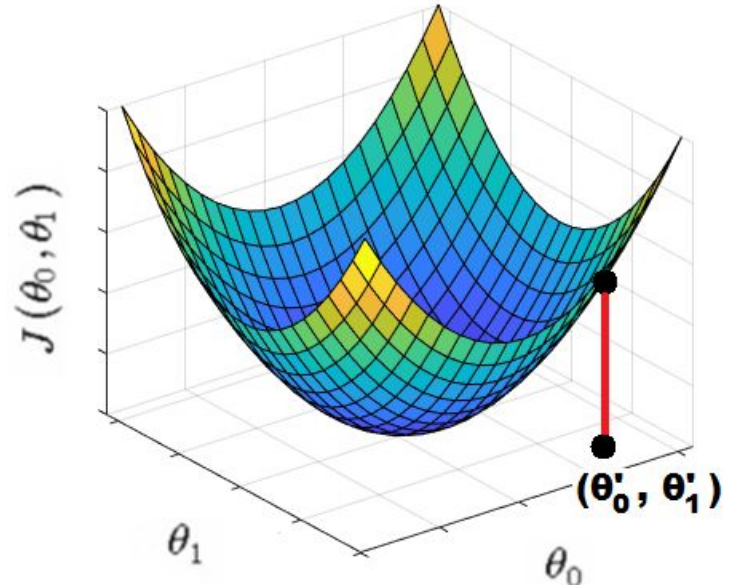
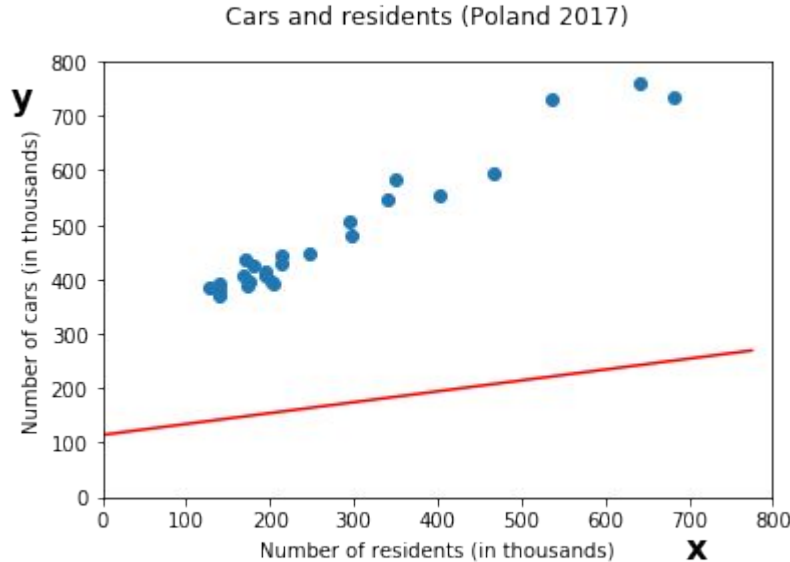
$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^m (\theta_1 x^{(j)} + \theta_0 - y^{(j)})$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^m (\theta_1 x^{(j)} + \theta_0 - y^{(j)}) \cdot x^{(j)}$$

alfa: tanulási ráta (learning rate); a lépések mérete skálázható vele

Előző órán - Lineáris regresszió megoldás

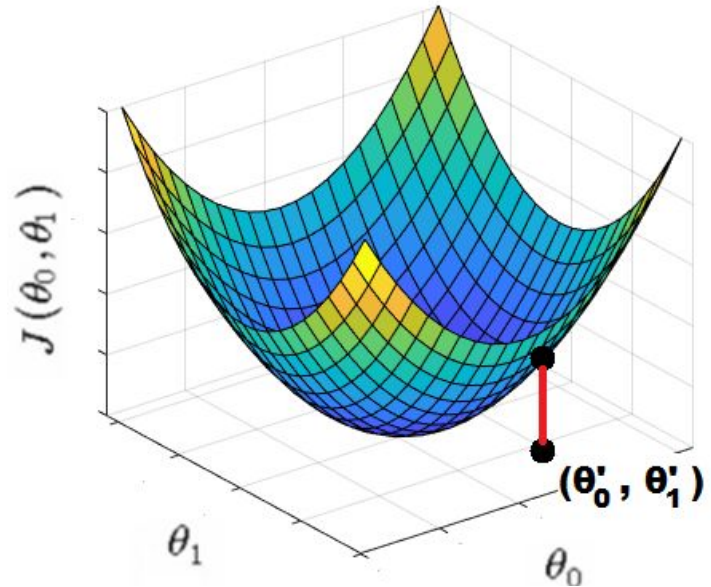
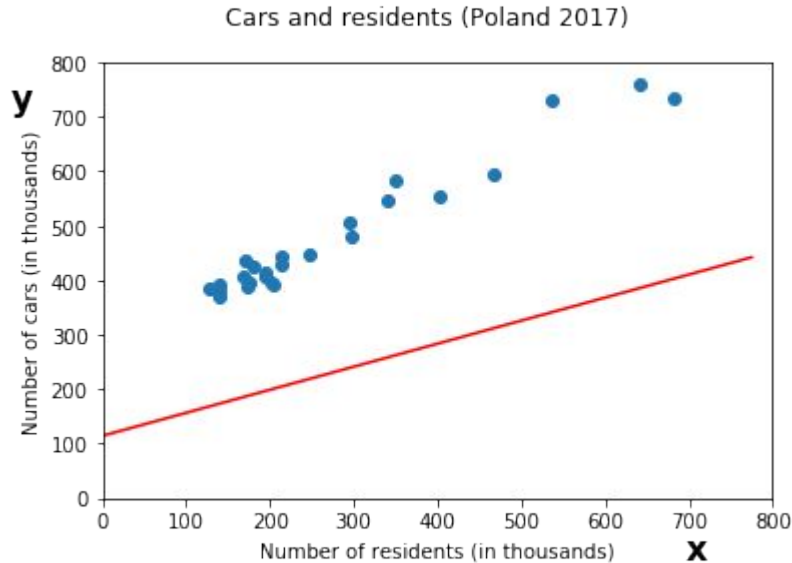
A gradiensmódszer alkalmazása, $T = 0$ (az első lépés előtt)



A kezdeti (θ_0, θ_1) -t véletlenszerűen választottuk.

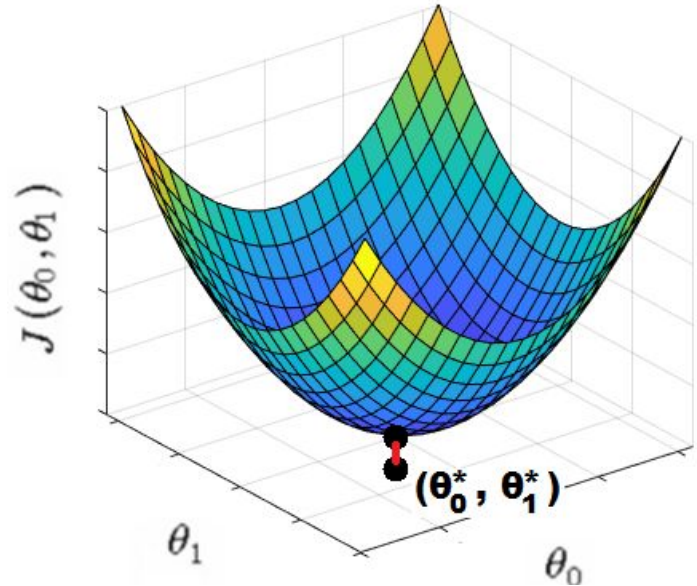
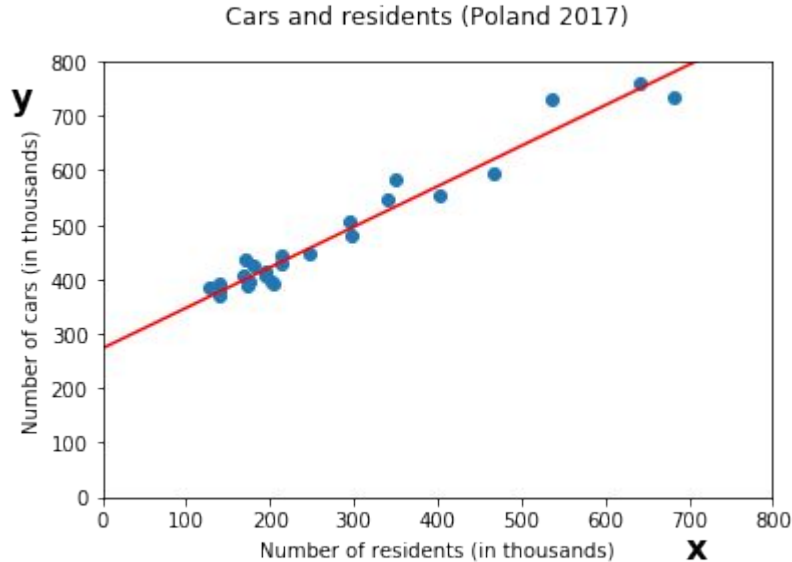
Előző órán - Lineáris regresszió megoldás

A gradiensmódszer alkalmazása, $T = 1$



Előző órán - Lineáris regresszió megoldás

A gradiensmódszer alkalmazása, $T = \langle \text{sok} \rangle$

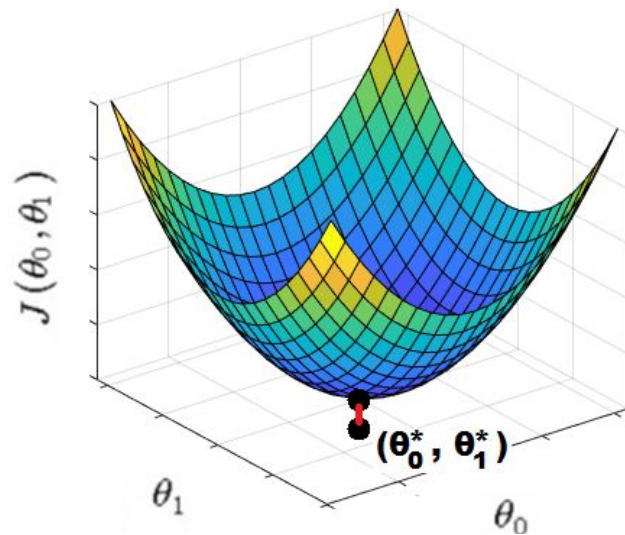


Előző órán - Lineáris regresszió megoldás

Megfelelő (alfa) lépésméret esetén **garantáltan megtaláljuk a lineáris regresszió költségfüggvényének globális minimumát** (azaz az optimális paramétereket) a gradiens módszerrel.

A lineáris regresszió speciális:
a költségfüggvény **konvex**.

Ha ez nem áll fenn, legfeljebb egy **lokális minimum** megtalálása garantált.



Előző órán - Lineáris regresszió megoldás

Mit értünk el?

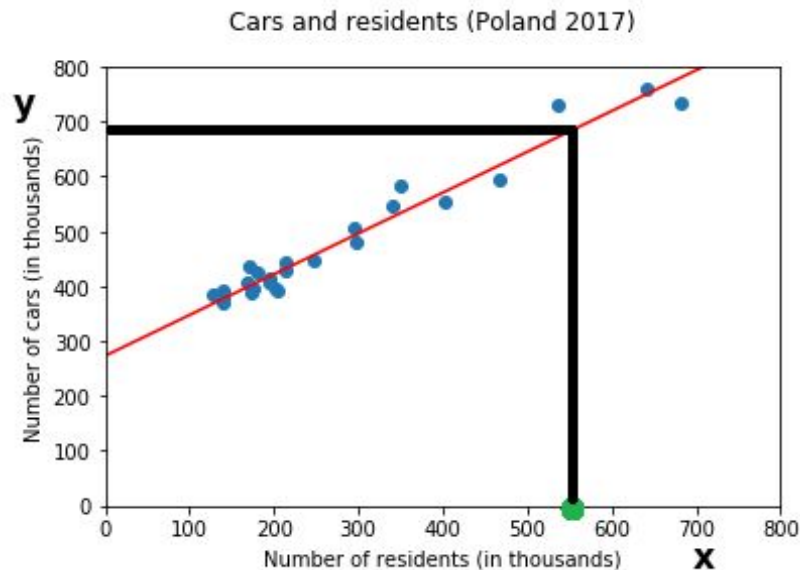
Betanítottunk egy egyszerű (lineáris) regressziós modellt.

Új, címkézetlen mintaelemekre fogunk tudni címkét becsülni.

Például a betanult modell paraméterei:

$$\theta_0 = 295.12, \quad \theta_1 = 0.6817$$

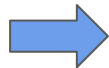
Ekkor, egy 550 ezer lakosú városra
 $295.12 + 550 * 0.6817 = 670.15$ ezer autót
becslünk



Egyváltozós lineáris regresszió

Eddigi példafeladatok:

x: Város lakosainak száma



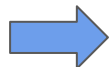
y: Autók száma

x: Lakás alapterülete



y: Lakás ára

x: Páciens tömege

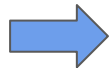


y: Páciens koleszterinszintje

Egyváltozós lineáris regresszió

Eddigi példafeladatok:

x: Város lakosainak száma



y: Autók száma

x: Lakás alapterülete



y: Lakás ára

x: Páciens tömege



y: Páciens koleszterinszintje

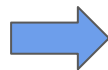
Túlzottan egyszerű modelleket tanulhatunk csak így...
Rengeteg egyéb tényező befolyásolhatja a címkét

Többváltozós lineáris regresszió

Új példafeladatok - több input változó (feature):

x_1 : Város lakosainak száma

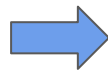
x_2 : Egy főre jutó GDP a városban



y : Autók száma

x_1 : Lakás alapterülete

x_2 : Távolság a városközponttól

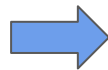


y : Lakás ára

x_1 : Páciens tömege

x_2 : Páciens életkora

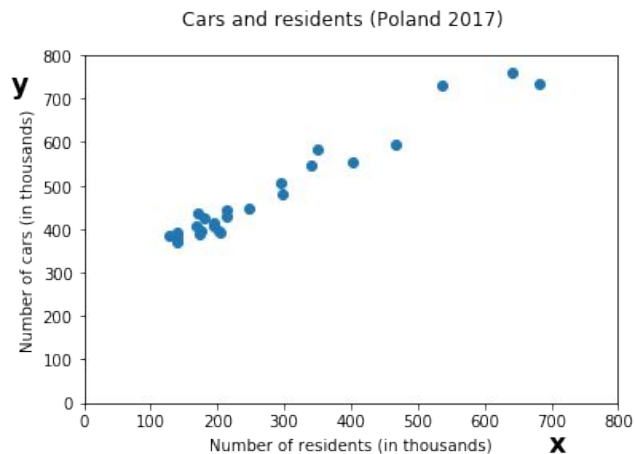
x_3 : Páciens neme



y : Páciens koleszterinszintje

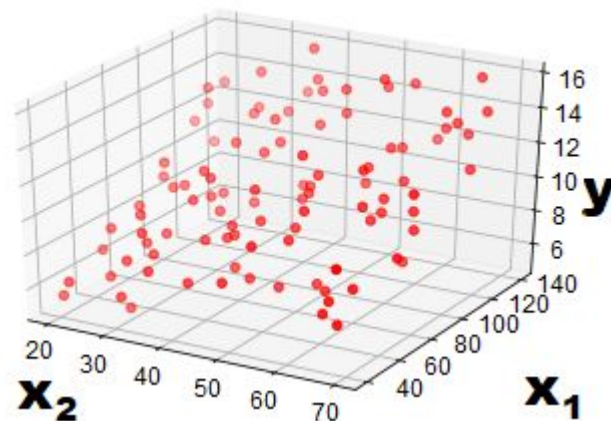
Többváltozós lineáris regresszió

Eddig: Egyváltozós minta



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

Ezután: Kétváltozós minta

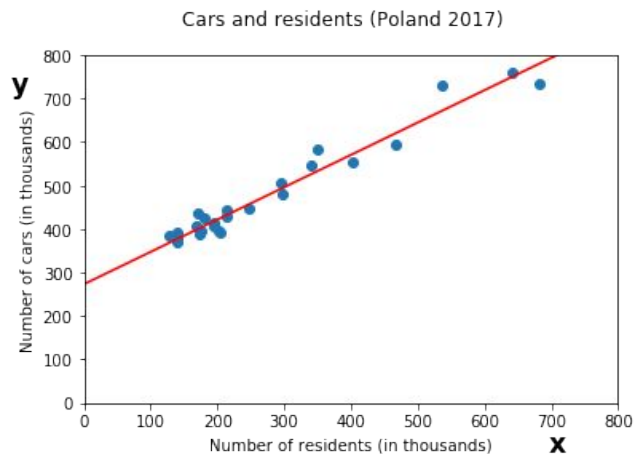


$$x^{(j)} \in \mathbb{R}^2, y^{(j)} \in \mathbb{R}$$

Többváltozós lineáris regresszió

Eddig: Egyváltozós minta

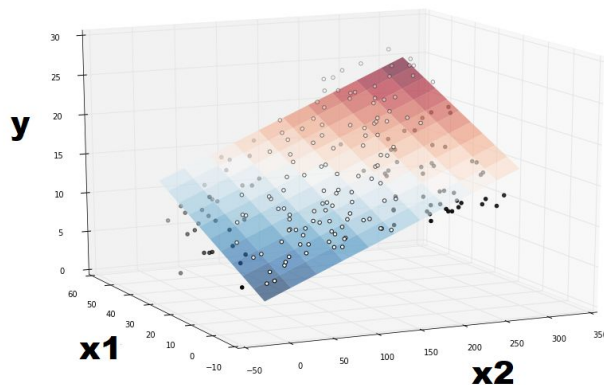
Hipotézis: Egyenes a síkon



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

Ezután: Kétváltozós minta

Hipotézis: Sík a térben

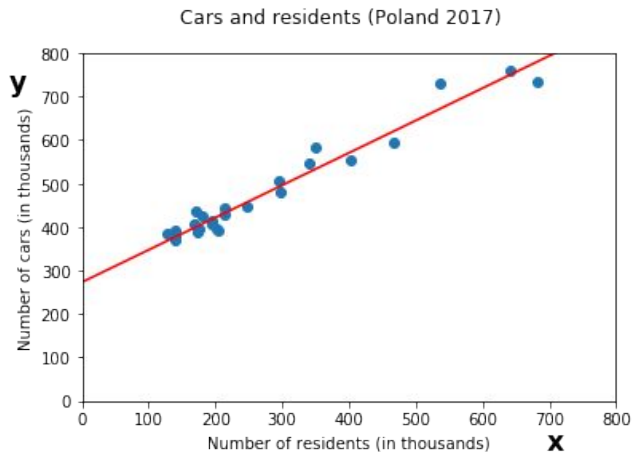


$$x^{(j)} \in \mathbb{R}^2, y^{(j)} \in \mathbb{R}$$

Többváltozós lineáris regresszió

Eddig: Egyváltozós minta

Hipotézis: Egyenes a síkon



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

Ezután: n változós minta

Hipotézis: n dimenziós hipersík
az $n+1$ dimenziós térben

$$x^{(j)} \in \mathbb{R}^n, y^{(j)} \in \mathbb{R}$$

Többváltozós lineáris regresszió

Hipotézisfüggvény - egy változó:

$$y \approx \hat{y} = h(x) = \theta_1 x + \theta_0$$

Hipotézisfüggvény - két változó:

$$y \approx \hat{y} = h(x) = \theta_2 x_2 + \theta_1 x_1 + \theta_0$$

Hipotézisfüggvény - n változó:

$$y \approx \hat{y} = h(x) = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0$$

Többváltozós lineáris regresszió

Hipotézisfüggvény - n változó:

$$y \approx \hat{y} = h(x) = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0$$

→ n változó, $n+1$ paraméter

Azaz, a költségfüggvény: $J : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$

$$h(x^{(j)}) = \hat{y}^{(j)}$$

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{j=1}^m \left(\overbrace{\theta_n x_n^{(j)} + \theta_{n-1} x_{n-1}^{(j)} + \cdots + \theta_1 x_1^{(j)} + \theta_0}^{h(x^{(j)})} - y^{(j)} \right)^2$$

Többváltozós lineáris regresszió

Hipotézisfüggvény - n változó:

$$y \approx \hat{y} = h(x) = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0$$

Hogyan lehetne egyszerűbben felírni?

Többváltozós lineáris regresszió

Hipotézisfüggvény - n változó:

$$y \approx \hat{y} = h(x) = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0$$

Hogyan lehetne egyszerűbben felírni?

Vektoros alakban: a becsült címke az input változók és paraméterek skalárszorzata!

$$y \approx \hat{y} = h(x^{(j)}) = \langle \theta, x^{(j)} \rangle = \sum_{i=0}^n \theta_i x_i$$

Többváltozós lineáris regresszió

Hipotézisfüggvény - n változó:

$$x_0 := 1$$

$$y \approx \hat{y} = h(x) = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0 x_0$$

Vektoros alakban: a becsült címke az input változók és paraméterek skalárszorzata!

$$y \approx \hat{y} = h(x^{(j)}) = \langle \theta, x^{(j)} \rangle = \sum_{i=0}^n \theta_i x_i$$

Egy extra “ál-változó”, x_0 bevezetése szükséges, hogy a skalárszorzat kijöjjön. **x_0 értéke mindig 1.**

Többváltozós lineáris regresszió

Hipotézisfüggvény - n változó:

$$x_0 := 1$$

$$y \approx \hat{y} = h(x) = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0 x_0$$

$$y \approx \hat{y} = h(x^{(j)}) = \langle \theta, x^{(j)} \rangle = \sum_{i=0}^n \theta_i x_i$$

Egyszerre az összes mintaelemre felírható:

$$y \approx \hat{y} = h(x) = X\theta$$

$$X \in \mathbb{R}^{m \times n}$$

Többváltozós lineáris regresszió

$$\langle x^{(j)}, \theta \rangle = \sum_{i=0}^n x_i^{(j)} \theta_i$$

skalárszorzat

$$\theta = \begin{bmatrix} \theta_0 \\ \dots \\ \theta_n \end{bmatrix} \in \mathbb{R}^n$$

$$x = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ \dots & \dots & \dots & \dots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \quad \langle x^{(1)}, \theta \rangle = \quad \hat{y}^{(1)} \approx \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^m$$

$$h(x) = X\theta = \hat{y} \approx y$$

Többváltozós lineáris regresszió

$$\langle x^{(j)}, \theta \rangle = \sum_{i=0}^n x_i^{(j)} \theta_i$$

skalárszorzat

$$\theta = \begin{bmatrix} \theta_0 \\ \dots \\ \theta_n \end{bmatrix} \in \mathbb{R}^n$$

$$x = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ \dots & \dots & \dots & \dots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \quad \langle x^{(1)}, \theta \rangle = \quad \hat{y}^{(1)} \approx \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^m$$

$$h(x) = X\theta = \hat{y} \approx y$$

`y_pred = np.dot(X, self.theta)`

Többváltozós lineáris regresszió

Gradiensmódszer általánosan:

repeat until convergence {

for $i \leftarrow 1 \dots n$ {

$$grad_i = \frac{\partial}{\partial \theta_i} J(\theta)$$


}

for $i \leftarrow 1 \dots n$ {

$$\theta_i = \theta_i - \alpha grad_i$$

}

}


$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum_{j=1}^m (\theta_n x_n^{(j)} + \dots + \theta_0 x_0^{(j)} - y^{(j)}) x_i^{(j)}$$

Érintő meredeksége θ_i
irányban

Többváltozós lineáris regresszió

Gradiensmódszer általánosan:

repeat until convergence {

for $i \leftarrow 1 \dots n$ {

$$grad_i = \frac{\partial}{\partial \theta_i} J(\theta)$$

}

for $i \leftarrow 1 \dots n$ {


$$\theta_i = \theta_i - \alpha grad_i$$

}

}

Kiszámoljuk a gradiensvektort:

ennek elemei a költségfüggvény parciális deriváltjai az egyes θ_i paraméterek szerint. (Megadja a költségfv.-nek az aktuális paraméterpontban a legnagyobb meredekség irányát.)


$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum_{j=1}^m (\theta_n x_n^{(j)} + \dots + \theta_0 x_0^{(j)} - y^{(j)}) x_i^{(j)}$$

Paraméter update: kivonjuk a gradiensvektort a tanulási rátával szorozva a régi paraméterekből (Lépünk egyet a költségfv. legnagyobb lejtése irányába...)

Többváltozós lineáris regresszió

Gradiensmódszer általánosan:

A költségfüggvény továbbra is kvadratikus, mindössze a változók száma lett nagyobb.

→ továbbra is konvex, ezért garantált, hogy megtaláljuk az optimális megoldást.

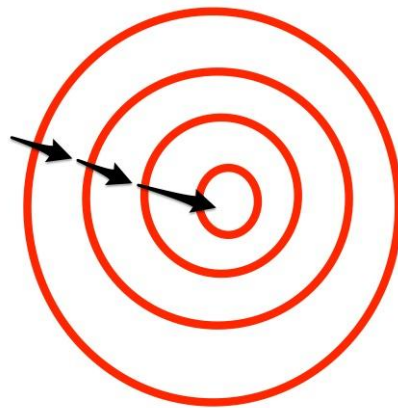
Többváltozós lineáris regresszió

Feature scaling: Az input változók azonos nagyságrendre hozása

Without feature scaling



With feature scaling



Többváltozós lineáris regresszió

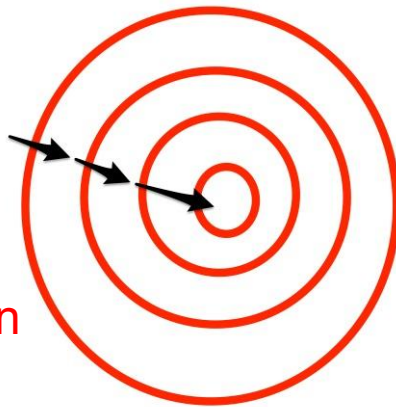
Költségfv. 2 paraméter
esetén - felülnézet
(szintvonalak)

Feature scaling: Az input változók azonos nagyságrendre hozása

Without feature scaling



With feature scaling



Ha az input változók egészen más nagyságrendben mozognak, a költségfüggvény extrém mértékben elnyújtott lesz. Ilyenkor a legnagyobb lejtés irányába lépkedés csak nagyon lassan konvergál.

Emiatt érdemes az input változókat azonos nagyságrendre skálázni.

Változónként
függetlenül!

Többváltozós lineáris regresszió

Feature scaling: Az input változók azonos nagyságrendre hozása

- $[0,1]$ intervallumra **min-max skálázás**:

$$\forall i, j : x_i^{(j)} = \frac{x_i^{(j)} - \min_j(\{x_i^{(j)}\})}{\max_j(\{x_i^{(j)}\}) - \min_j(\{x_i^{(j)}\})}$$

- **sztenderdizálás** (0 átlagra, 1 szórásra hozás):

$$\forall i, j : x_i^{(j)} = \frac{x_i^{(j)} - \mu_j(\{x_i^{(j)}\})}{\sigma_j(\{x_i^{(j)}\})}$$

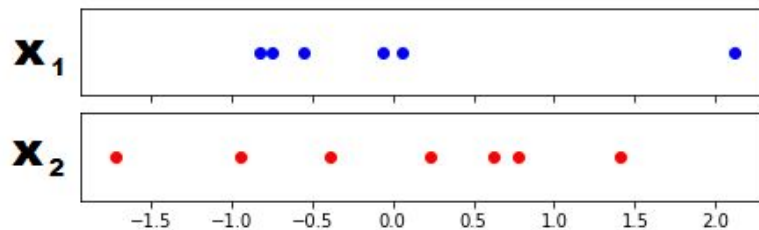
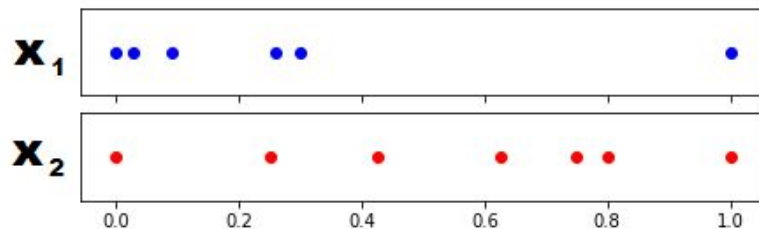
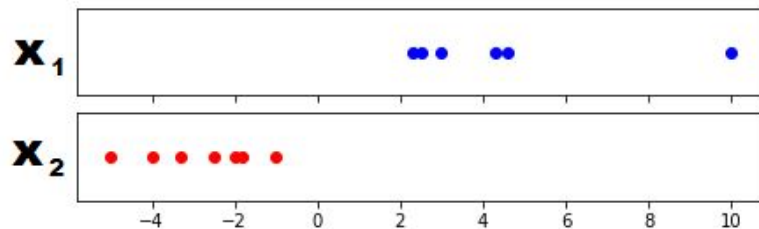
μ : átlag
 σ : szórás

Többváltozós lineáris regresszió

Feature scaling: Az input változók azonos nagyságrendre hozása

- $[0,1]$ intervallumra
min-max skálázás:
- **sztdenderdizálás**
(0 átlagra, 1 szórásra hozás):

Változónként
függetlenül!



Többváltozós lineáris regresszió

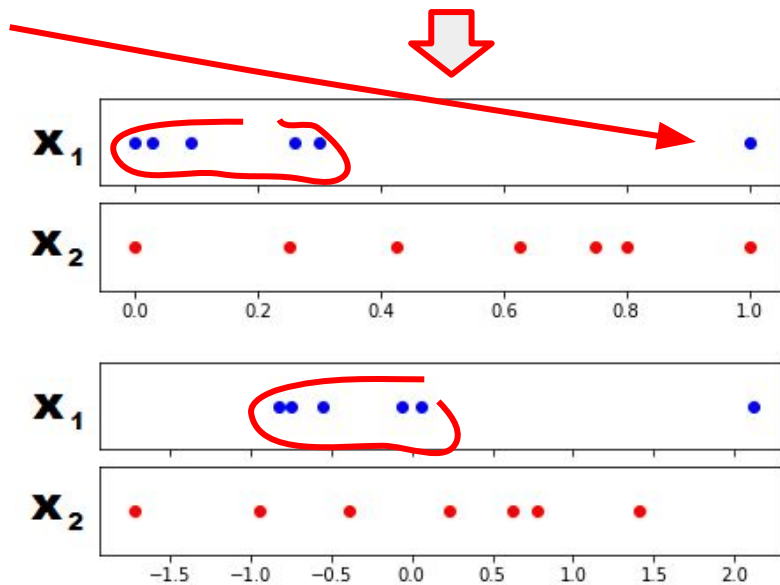
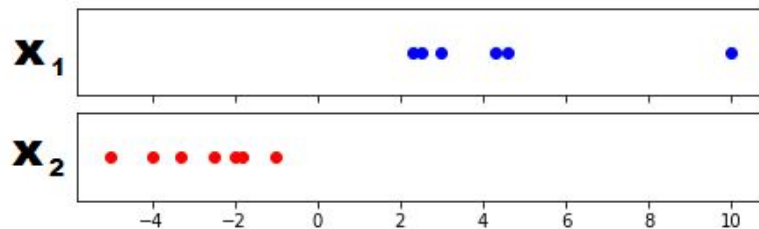
Feature scaling: Az input változók azonos nagyságrendre hozása

Min-max: Érzékenyebb a kiugróan eltérő (outlier) elemekre

- $[0,1]$ intervallumra
min-max skálázás:

- **sztdenderdizálás**
(0 átlagra, 1 szórásra hozás):

Változónként
függetlenül!



A felügyelt tanulás két fő feladata

Regresszió: Folytonos értékű címke becslése

$$|Y| = \infty$$

Példa: Autók számának, vagy életkor becslése

Klasszifikáció: Diszkrét értékű címke becslése

$$|Y| < \infty$$

Példa: Mintaelemek kategorizálása

- A lakosság számából eldönteni, hogy város-e, vagy falu egy adott település
- Mi a foglalkozása a képeken szereplő személyeknek?

Klasszifikáció

Példa: Kategorizáljuk a fényképeket a rajta szereplő állatok szerint!

Milyen címkéket tanuljunk? Egy lehetséges megközelítés:

Legyen pl. “kutya” = 1, “macska” = 2.

Használhatunk regressziót. Amelyikhez közelebb esik a becslés, azt a kategóriát becsüljük.

Klasszifikáció

Példa: Kategorizáljuk a fényképeket a rajta szereplő állatok szerint!

Mi a helyzet, ha több kategóriánk van?

Pl. “kutya” = 1, “macska” = 2, “papagáj” = 3.

Mi a probléma ezzel?

Klasszifikáció

Példa: Kategorizáljuk a fényképeket a rajta szereplő állatok szerint!

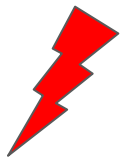
Mi a helyzet, ha több kategóriánk van?

Pl. “kutya” = 1, “macska” = 2, “papagáj” = 3.

Mi a probléma ezzel?

Regresszió esetén feltételezés:

$\frac{1}{2}$ kutya + $\frac{1}{2}$ papagáj = 1 macska



Más megközelítés kell...

Klasszifikáció

Példa: Kategorizáljuk a fényképeket a rajta szereplő állatok szerint!

Valószínűségeket fogunk becsülni!

Címke: Mennyi a valószínűsége, hogy egy mintaelem az adott kategóriába tartozik?

Bináris klasszifikáció

A legegyszerűbb eset: Bináris klasszifikáció (két kategória)

$$\hat{y} = h(x) = P(x \text{ egy macska})$$

$$\hat{y} = h(x) = 1 - P(x \text{ egy kutya})$$

A becslés folytonos érték. Legyen **P = 0.5** a határ.

Bináris klasszifikáció

A legegyszerűbb eset: Bináris klasszifikáció (két kategória)

$$\hat{y} = h(x) = P(x \text{ egy macska})$$

$$\hat{y} = h(x) = 1 - P(x \text{ egy kutya})$$

A becslés folytonos érték. Legyen **P = 0.5** a határ.

Ha a becslésünk (\hat{y}) nagyobb, mint **0.5**, azt mondjuk “*Ez egy macska*”.

Ha kisebb, azt mondjuk “*Ez egy kutya*”.

Bináris klasszifikáció

A legegyszerűbb eset: Bináris klasszifikáció (két kategória)

$$\hat{y} = h(x) = P(x \text{ egy macska})$$

$$\hat{y} = h(x) = 1 - P(x \text{ egy kutya})$$

A becslés folytonos érték. Legyen **P = 0.5** a határ.

Mit tehetünk, ha a h hipotézis függvényünk a [0,1] intervallumon kívüli értéket becsül?

Bináris klasszifikáció

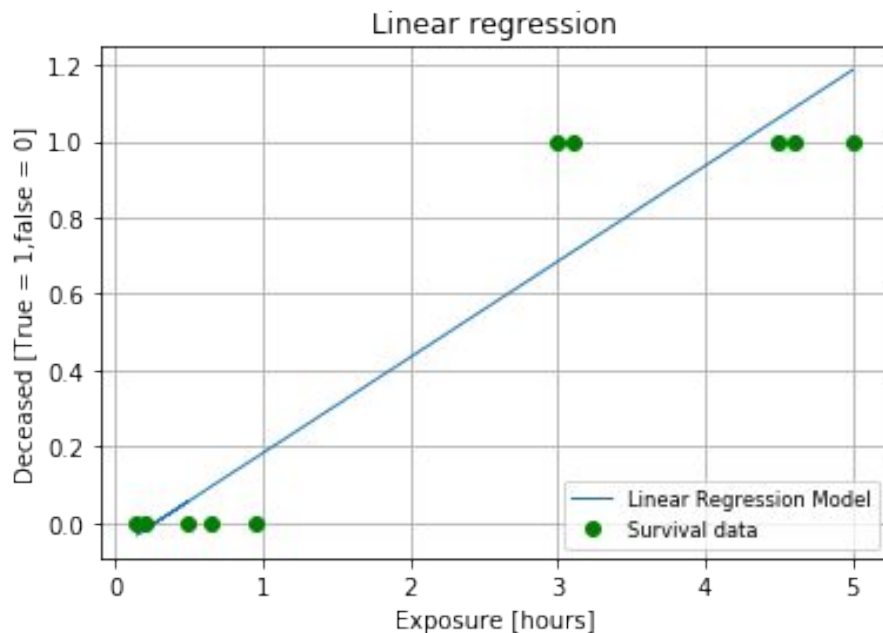
Mit tehetünk, ha a h hipotézis függvényünk a $[0,1]$ intervallumon kívüli értéket becsül?

A 0-nál kisebb becslést 0-nak, az 1-nél nagyobb becslést 1-nek tekinthetjük.

Mivel valószínűségeket becslünk, a címkék gyakorlatban folytonos értékek, így megpróbálkozhatunk a lineáris regresszióval!

Klasszifikáció lineáris regresszióval - ellenpélda

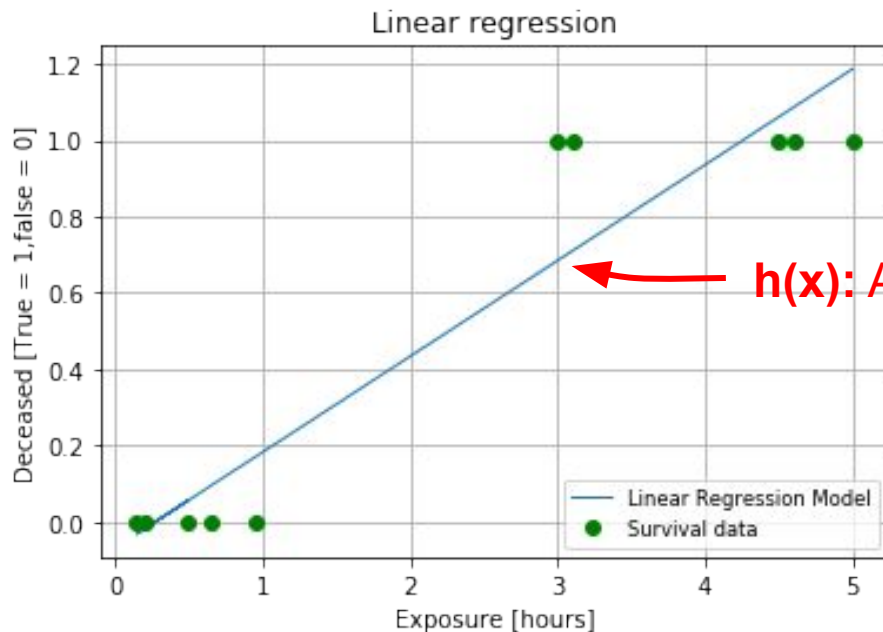
Példa: Radioaktív sugárzásnak tettünk ki embereket adott ideig. Túlélnek-e?



Klasszifikáció lineáris regresszióval - ellenpélda

Példa: Radioaktív sugárzásnak tettünk ki embereket adott ideig.
Túlélnek-e?

y: Meghalt-e?
(1: igen, 0: nem)

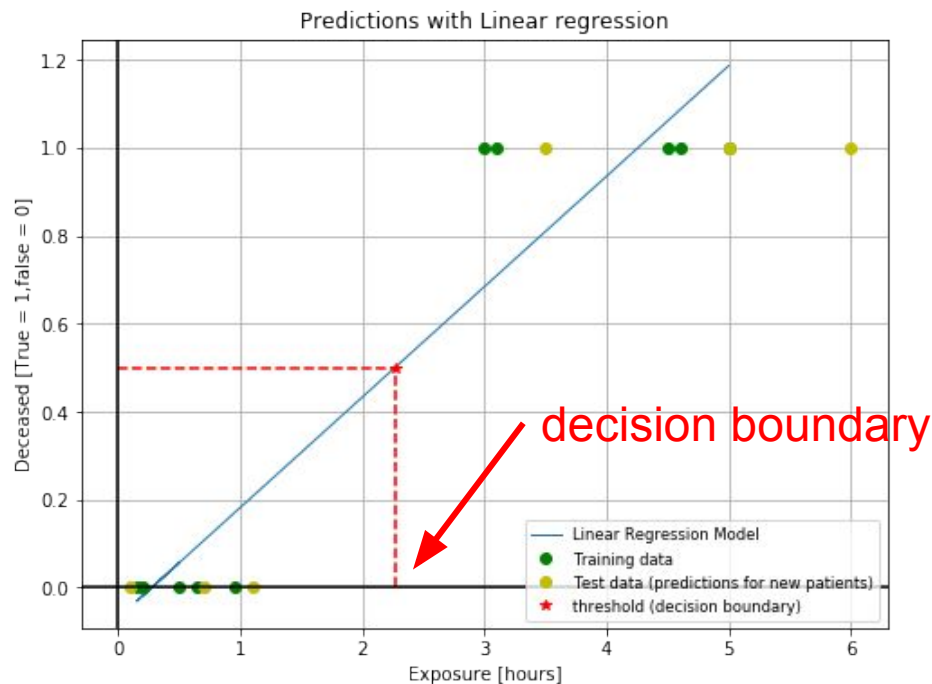


h(x): Az illesztett hipotézisfv.

x: Kitétség időtartama (óra)

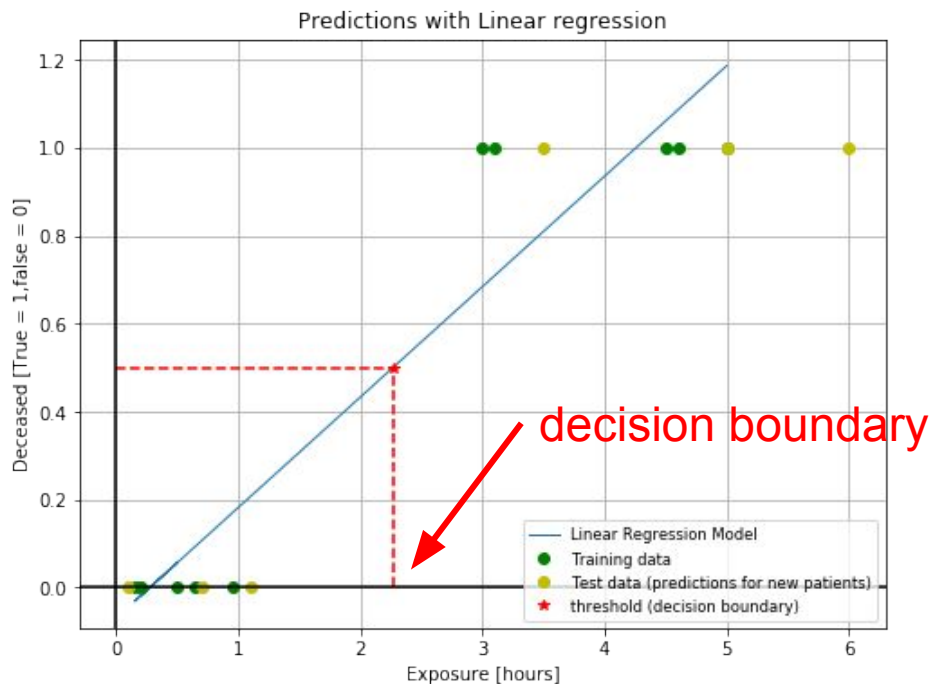
Klasszifikáció lineáris regresszióval - ellenpélda

Döntési felület (decision boundary): $\{x \mid \hat{y} = h(x) = 0.5\}$



Klasszifikáció lineáris regresszióval - ellenpélda

Döntési felület (decision boundary): $\{x \mid \hat{y} = h(x) = 0.5\}$



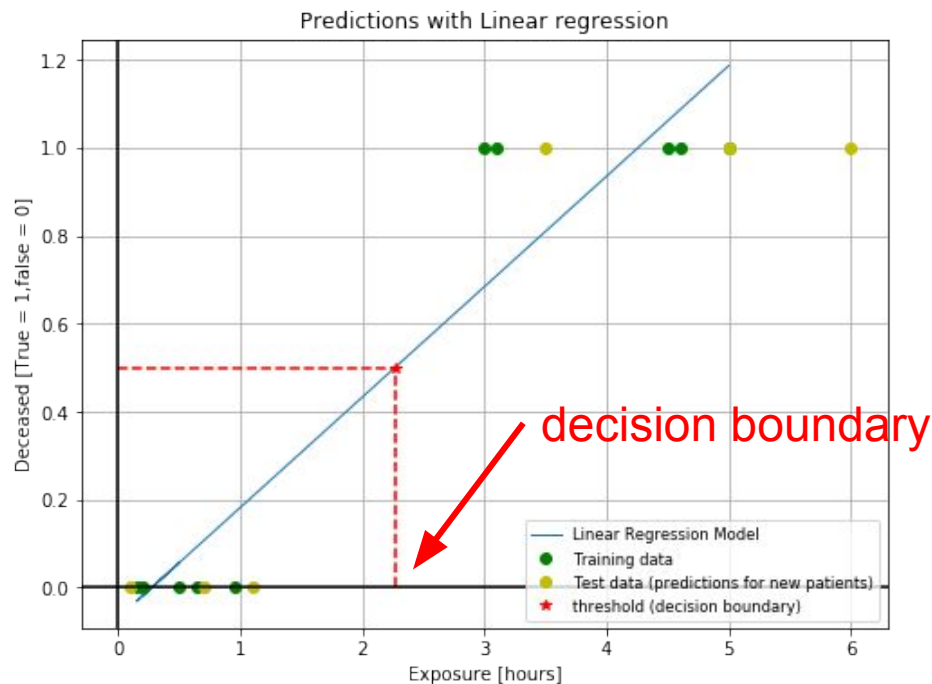
... azon **x**-ek halmaza, ahol a hipotézisfv. becslése pontosan **0.5** (feltéve, hogy h folytonos)
→ tipikusan egy (hiper)felület

- **egy változó**: pont
- **két változó**: egyenes vagy görbe a síkon

...

Klasszifikáció lineáris regresszióval - ellenpélda

Döntési felület (decision boundary): $\{x \mid \hat{y} = h(x) = 0.5\}$

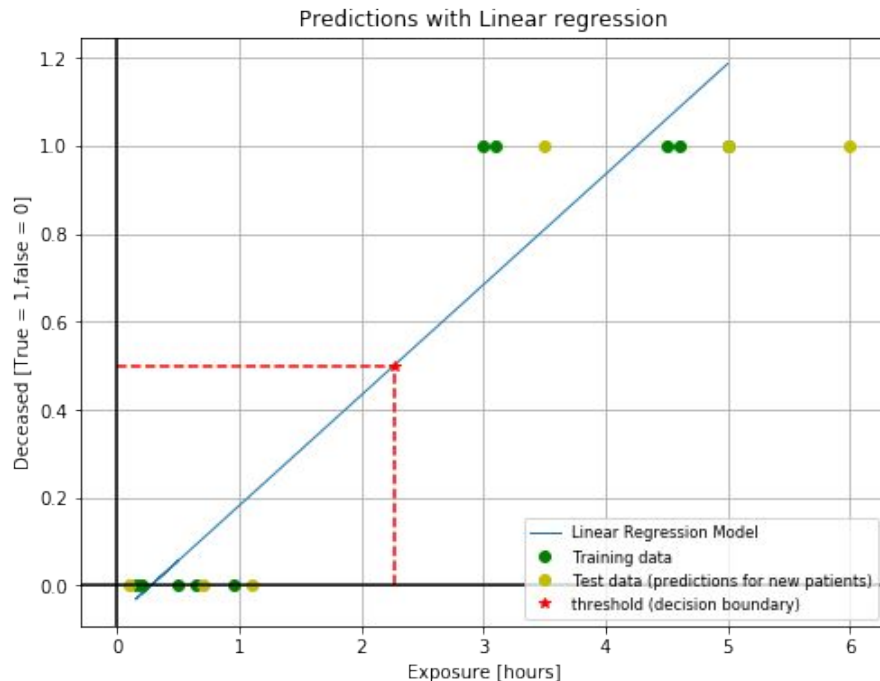


A lineáris regresszióval tanult
**döntési felület ebben a konkrét
példában: $x \approx 2.25$**

Azaz, a **becslésünk** az lesz,
hogy az illető **meghalt**, ha **több,
mint 2 és egynegyed** óráig volt
kitéve a sugárzásnak.

Klasszifikáció lineáris regresszióval - ellenpélda

Lehet-e probléma ezzel a megközelítéssel?



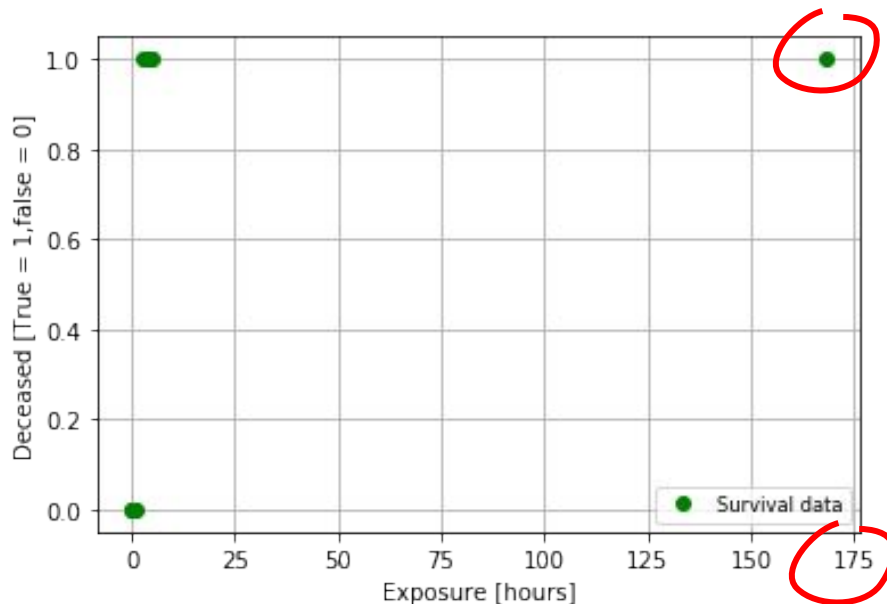
Klasszifikáció lineáris regresszióval - ellenpélda

Lehet-e probléma ezzel a megközelítéssel?

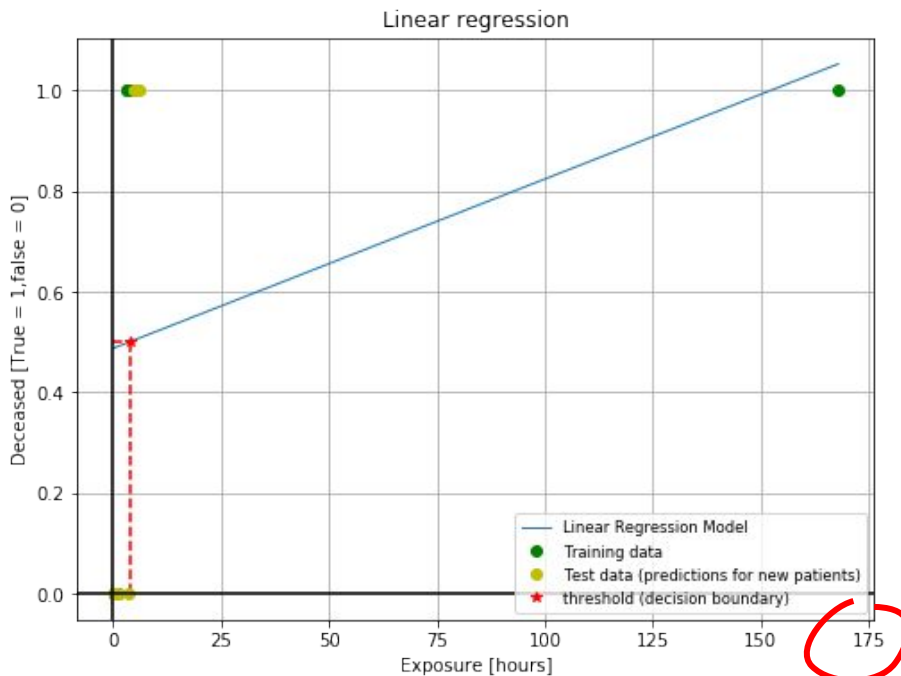
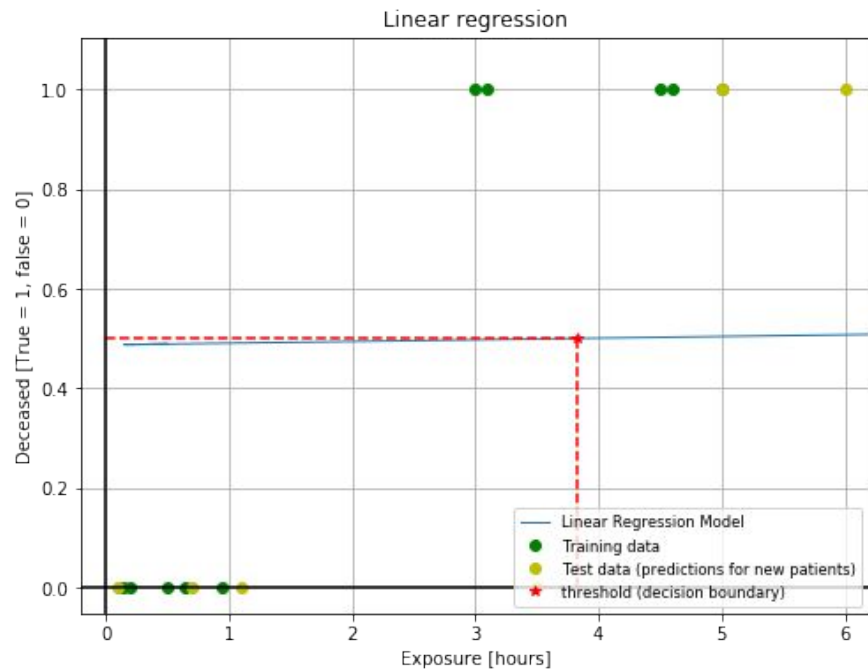
Adjunk hozzá egy új mintaelemet:

Találtunk egy halott páciens-t aki egy teljes hétig volt kitéve a sugárzásnak.

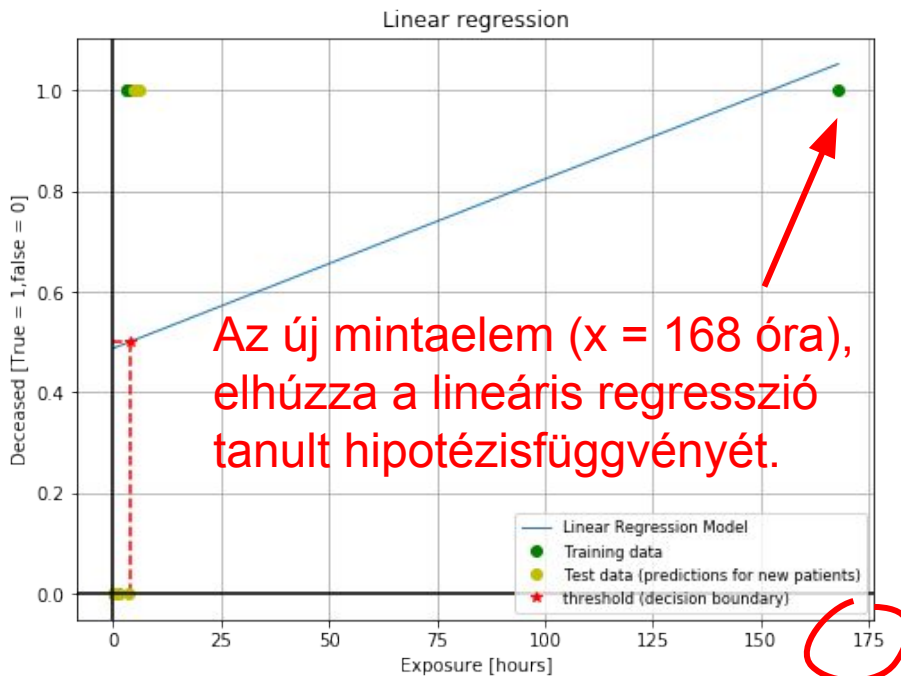
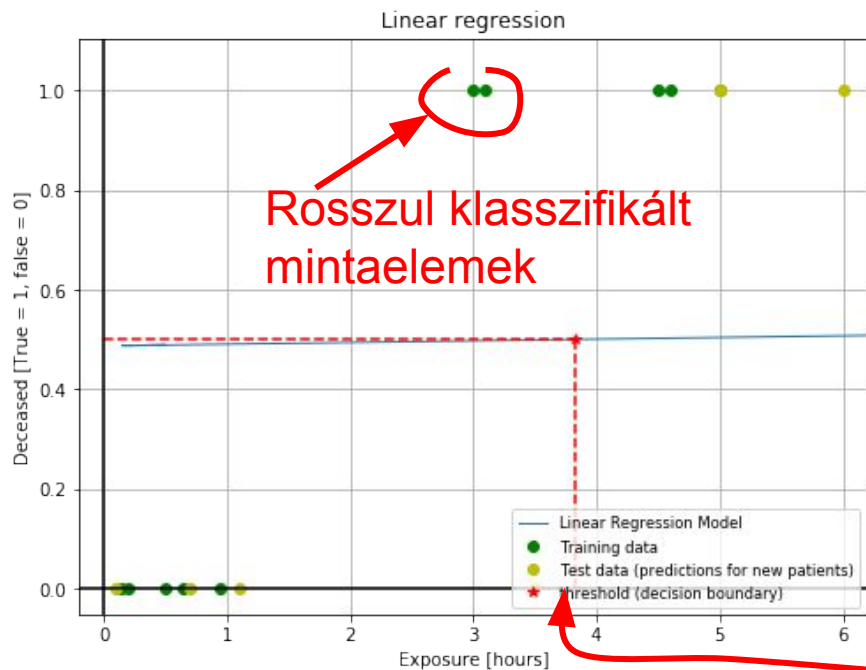
Mit tanul így a lineáris regresszió?



Klasszifikáció lineáris regresszióval - ellenpélda



Klasszifikáció lineáris regresszióval - ellenpélda



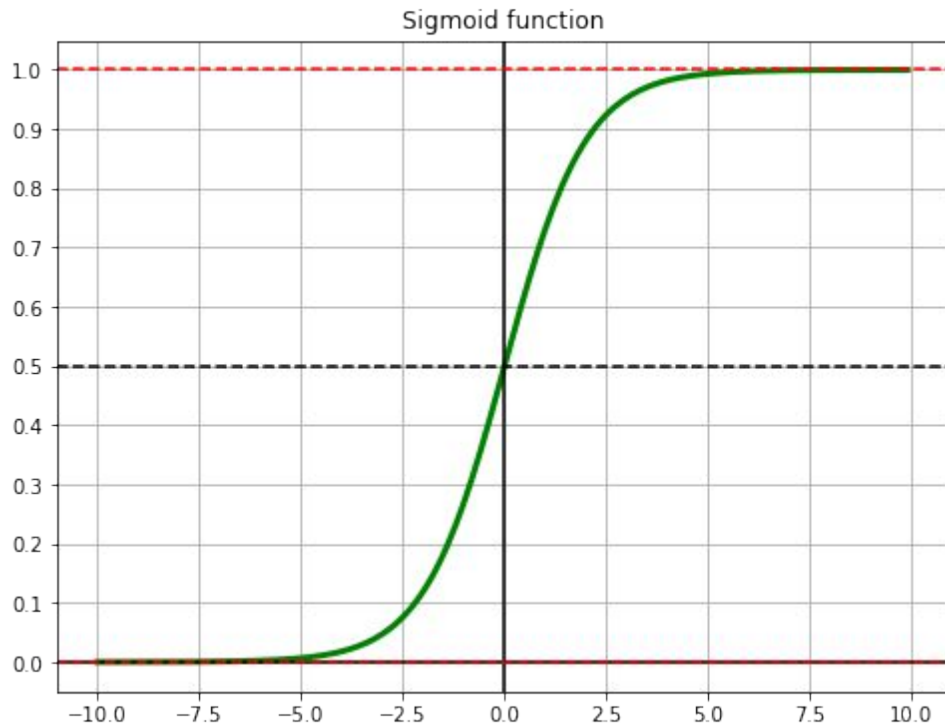
Az új mintaelem jelentősen befolyásolta a döntési felületet.
Így az már 3 óra 48 percre esik.

Klasszifikáció lineáris regresszióval - ellenpélda

A lineáris regresszió nem ideális klasszifikációs feladatok tanulására, mivel **nagyon érzékeny a kiugróan eltérő (outlier) mintaelemekre.**

Egyenes (hipersík) helyett mit illeszthetnénk az adatpontokra?

Klasszifikáció - szigmoid függvény



$$g(z) = \frac{1}{1+e^{-z}}$$

0 és 1 közé képez: **ideális
valószínűségek becslésére!**

Klasszifikáció - hipotézis függvény

Hogy lesz ebből hipotézis függvény?

- Lin. reg. hipotézis fv.:
$$h_{\theta}(x) = x\theta = \hat{y}$$

- Sigmoid fv...:
$$g(z) = \frac{1}{1+e^{-z}}$$

Logisztikus regresszió:
$$h_{\theta}(x) = g(x\theta) = \frac{1}{1+e^{-x\theta}} = \hat{y}$$

Klasszifikáció - hipotézis függvény

Logisztikus regresszió: $h_{\theta}(x) = g(x\theta) = \frac{1}{1+e^{-x\theta}} = \hat{y}$

Pl. egyváltozós, kifejtve:

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}} = \hat{y}$$

Klasszifikáció - hipotézis függvény

Egyváltozós logisztikus regresszió hipotézisfüggvénye:

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}} = \hat{y}$$

Hogyan befolyásolják a paraméterek a görbe alakját?

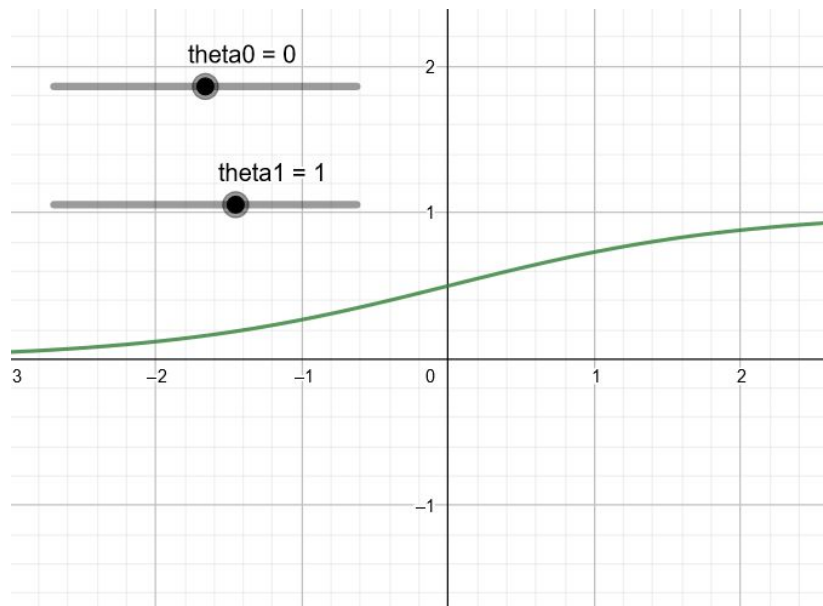
<https://www.geogebra.org/graphing/f5x9h7cr>

Klasszifikáció - hipotézis függvény

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}} = \hat{y}$$

$$\theta_0 = 0$$

$$\theta_1 = 1$$

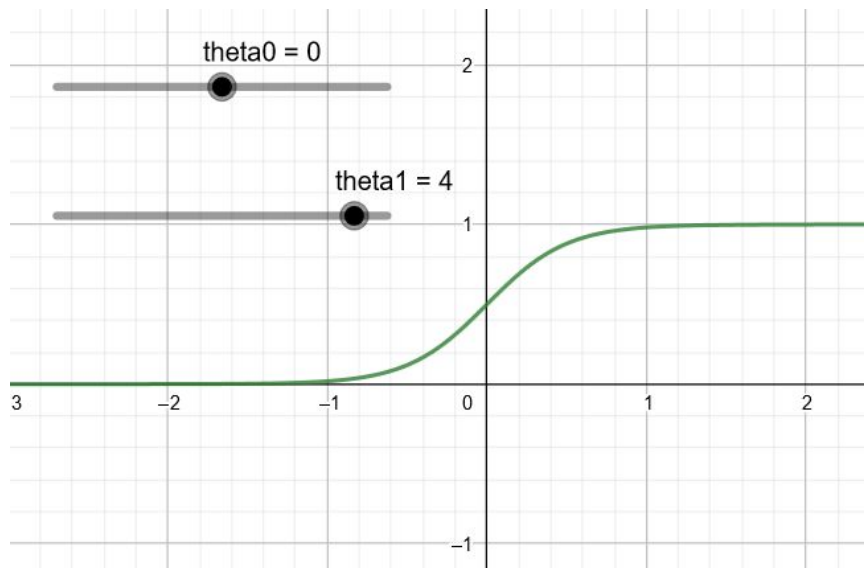


Klasszifikáció - hipotézis függvény

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}} = \hat{y}$$

$$\theta_0 = 0$$

$$\theta_1 = 4$$

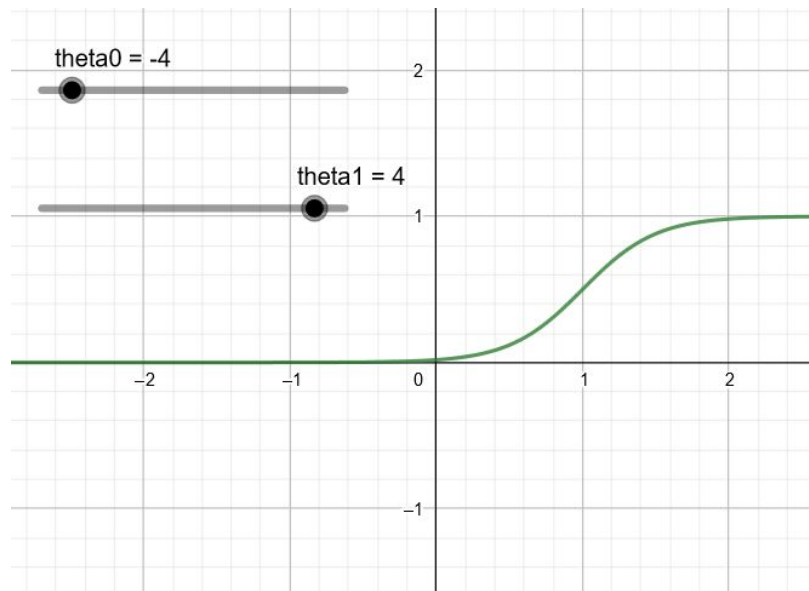


Klasszifikáció - hipotézis függvény

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}} = \hat{y}$$

$$\theta_0 = -4$$

$$\theta_1 = 4$$



Logisztikus regresszió

Döntési felület (decision boundary): $\{x \mid \hat{y} = h(x) = 0.5\}$

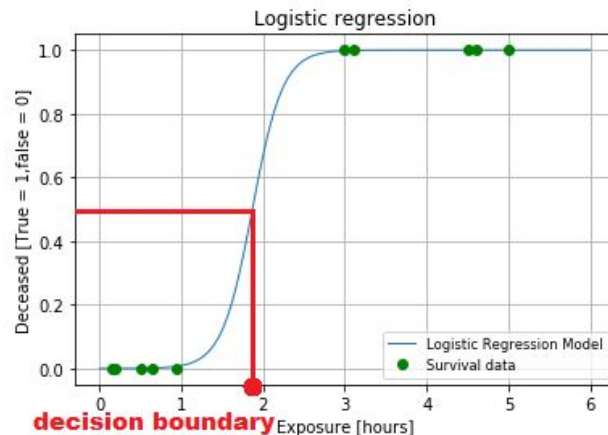
Illesztjük a h hipotézisfüggvényt az adathoz (betanítás).

A döntési felületünk azon x -ek halmaza, melyekre $h(x) = 0.5$

→ Egyváltozós esetben ez egy pont az x tengelyen, pl. $x = \text{“1.9 óra”}$

Új, címkézetlen mintaelemekre:

- ha $h(x) < 0.5$: 0-ás címkét becszlünk
- ha $h(x) > 0.5$: 1-es címkét becszlünk



Logisztikus regresszió

“Illesztjük a h hipotézisfüggvényt az adathoz...”

Hogyan? Milyen J költségfüggvényt használjunk?

Jó-e a lineáris regresszióhoz használt MSE költségfv.?

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)})^2$$

ahol,
$$h_{\theta}(x) = g(x\theta) = \frac{1}{1+e^{-x\theta}} = \hat{y}$$

Logisztikus regresszió

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)})^2$$

MSE loss

ahol, $h_{\theta}(x) = g(x\theta) = \frac{1}{1+e^{-x\theta}} = \hat{y}$

Probléma: A szigmoid nem konvex, a négyzete sem lesz az...

→ Több lokális szélsőérték lehet

→ A gradiens módszer nem feltétlenül találja meg az optimális megoldást

Logisztikus regresszió

Költséggüggvény logisztikus regresszióhoz:

$$J(\theta) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$

ahol
$$h_{\theta}(x) = g(x\theta) = \frac{1}{1+e^{-x\theta}} = \hat{y}$$

Logisztikus regresszió

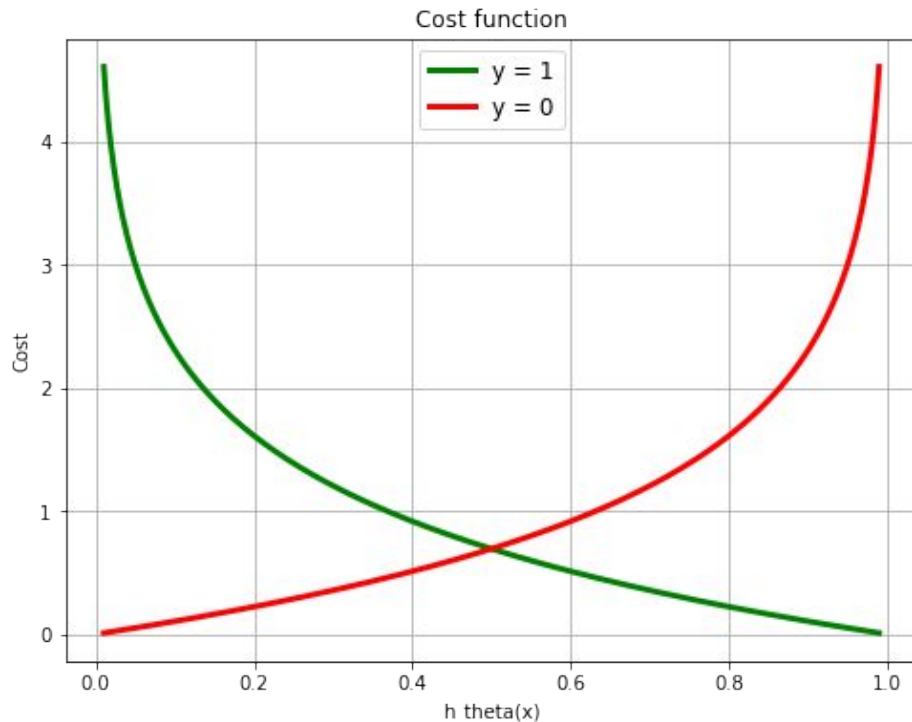
Költségfüggvény logisztikus regresszióhoz:

$$J(\theta) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$

Levezethető:

a költségfüggvény konvex

(a második deriváltja mindenhol pozitív)

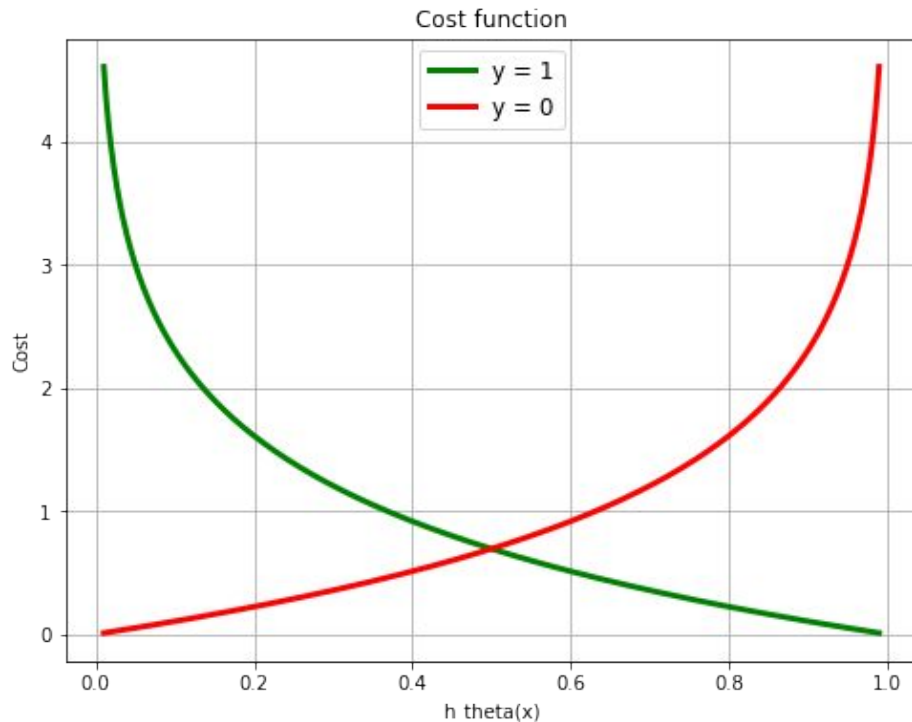


Logisztikus regresszió

Költségfüggvény logisztikus regresszióhoz:

$$J(\theta) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$

Esetek összevonása?



Logisztikus regresszió

Költségfüggvény, összevont alak (logistic loss, binary crossentropy):

$$J(\theta) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$



$$J(\theta) = \frac{1}{m} \sum_{j=1}^m \left[\underbrace{-y^{(j)} \log(h_{\theta}(x^{(j)}))}_{\text{= 0, ha az igazi } y \text{ címke 0}} - \underbrace{(1 - y^{(j)}) \log(1 - h_{\theta}(x^{(j)}))}_{\text{= 0, ha az igazi } y \text{ címke 1}} \right]$$

= 0, ha az igazi y címke 0

= 0, ha az igazi y címke 1

Logisztikus regresszió

Gradiens módszerhez derivált kell:

$$J(\theta) = \frac{1}{m} \sum_{j=1}^m [-y^{(j)} \log(h_{\theta}(x^{(j)})) - (1 - y^{(j)}) \log(1 - h_{\theta}(x^{(j)}))]$$

$$x_0 = 1$$

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\theta_0 x_0 + \theta_1 x_1 + \dots)}}$$

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)}) x_i^{(j)}$$

Logisztikus regresszió

Gradiens módszer általánosan (többváltozós):

repeat until convergence {

 for $i \leftarrow 1 \dots n$ {

$$\text{grad}_i = \frac{\partial}{\partial \theta_i} J(\theta)$$

 }

 for $i \leftarrow 1 \dots n$ {

$$\theta_i = \theta_i - \alpha \text{grad}_i$$

 }

}

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)}) x_i^{(j)}$$

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\theta_0 x_0 + \theta_1 x_1 + \dots)}}$$

Logisztikus regresszió

Gradiens módszer általánosan (többváltozós):

repeat until convergence {

for $i \leftarrow 1 \dots n$ {

$$grad_i = \frac{\partial}{\partial \theta_i} J(\theta)$$

}

for $i \leftarrow 1 \dots n$ {

$$\theta_i = \theta_i - \alpha grad_i$$

}

}

A derivált ránézésre ugyanaz, de ne feledjük: itt a hipotézisfüggvény más

$$\frac{\partial}{\partial \theta_i} J(\theta) = \frac{1}{m} \sum_{j=1}^m (h_{\theta}(x^{(j)}) - y^{(j)}) x_i^{(j)}$$

$$h_{\theta}(x) = \frac{1}{1 + e^{-(\theta_0 x_0 + \theta_1 x_1 + \dots)}}$$

Egyváltozós bináris klasszifikáció

Egyváltozós példafeladatok bináris klasszifikációra:

x: Sugárzásnak kitettség időtartama  **y:** Meghalt-e az illető?

x: Település lakosainak száma  **y:** Város-e a település?

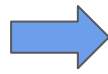
x: Állat tömege  **y:** Kutya-e, vagy macska?

Többváltozós bináris klasszifikáció

Többváltozós példafeladatok bináris klasszifikációra:

x_1: Sugárzásnak kitettség időtartama

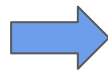
x_2: Távolság a sugárforrástól



y: Meghalt-e az illető?

x_1: Település lakosainak száma

x_2: Turisták éves száma

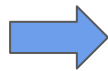


y: Város-e a település?

x_1: Állat tömege

x_2: Felköhögött szőrlabdák száma

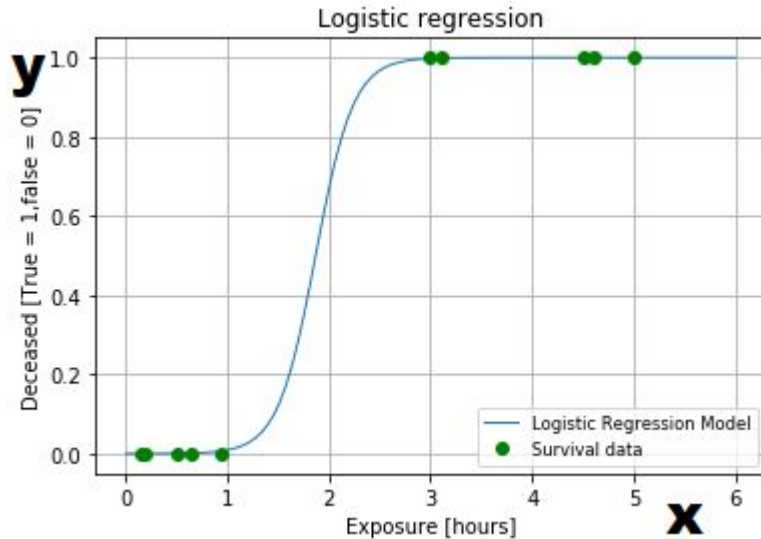
x_3: Nyáladzás mértéke



y: Kutya-e, vagy macska?

Többváltozós logisztikus regresszió

A h hipotézis függvény két változó esetén?

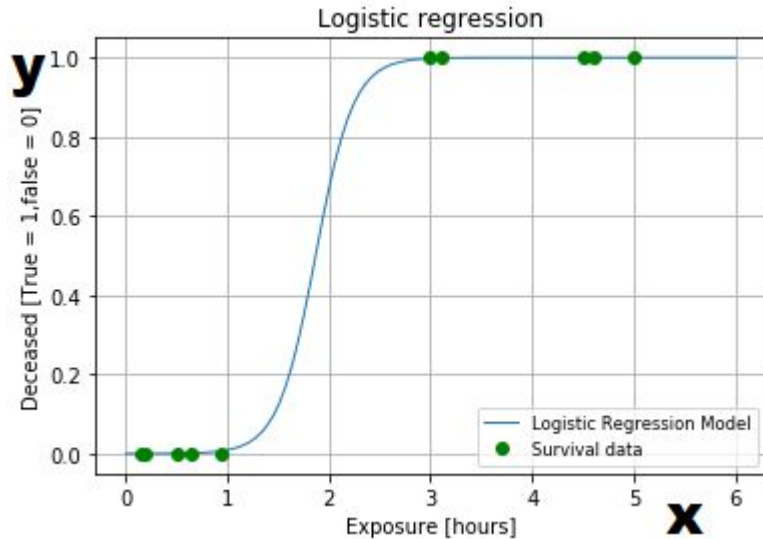


helyett

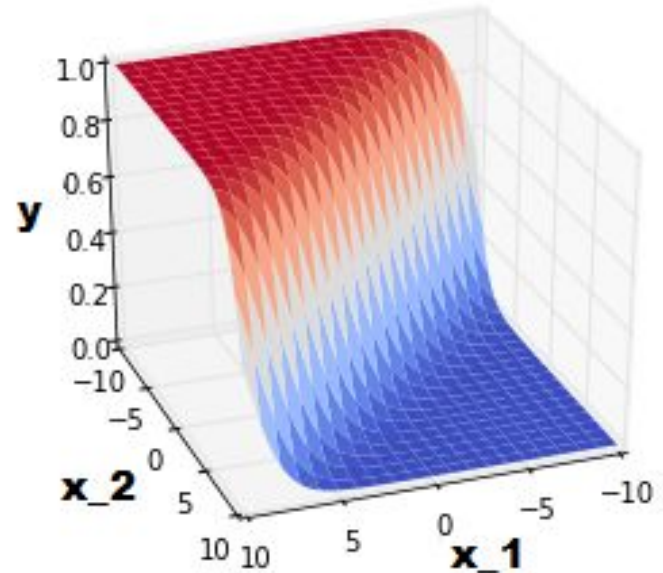
???

Többváltozós logisztikus regresszió

A h hipotézis függvény két változó esetén:

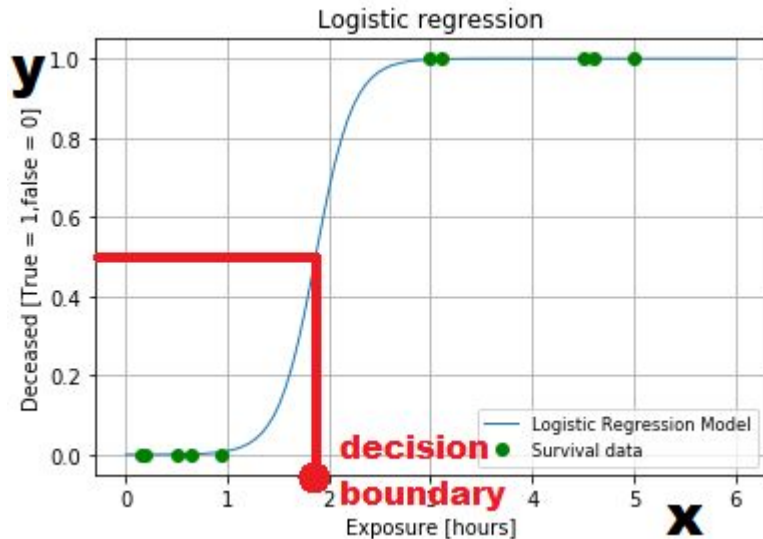


helyett

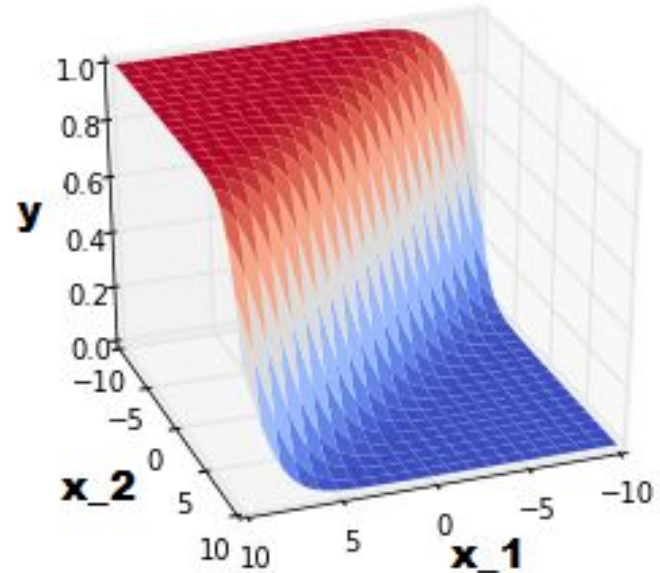


Többváltozós logisztikus regresszió

Döntési felület (decision boundary) két változó esetén?

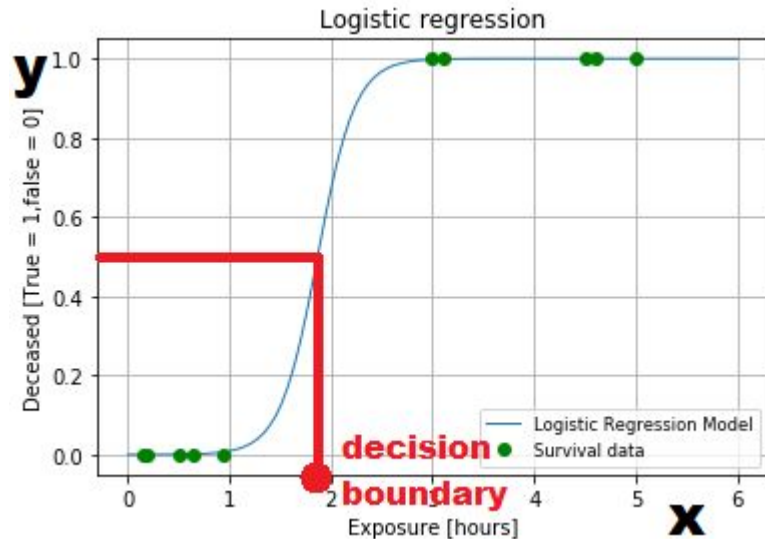


helyett

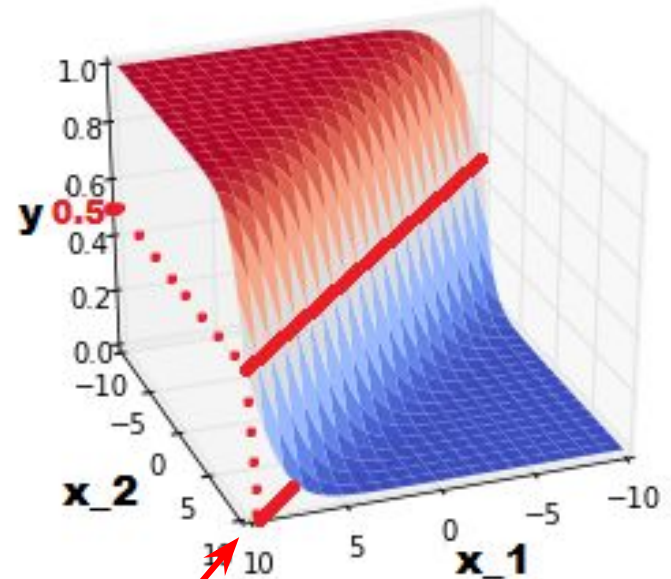


Többváltozós logisztikus regresszió

Döntési felület (decision boundary) két változó esetén:



helyett



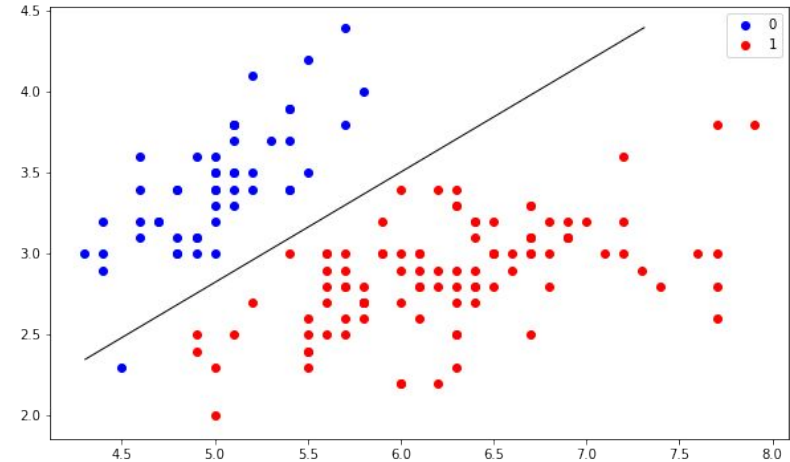
Döntési felület (egyenes)

Többváltozós logisztikus regresszió

Döntési felület (decision boundary) **két változó esetén:**



helyett



felülnézet

Többváltozós logisztikus regresszió

$$\langle x^{(j)}, \theta \rangle = \sum_{i=0}^n x_i^{(j)} \theta_i$$

skalárszorzat

$$\theta = \begin{bmatrix} \theta_0 \\ \dots \\ \theta_n \end{bmatrix} \in \mathbb{R}^n$$

$$x = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ \dots & \dots & \dots & \dots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \rightarrow \begin{matrix} \langle x^{(1)}, \theta \rangle \\ \dots \end{matrix} \quad g(\langle x^{(1)}, \theta \rangle) = \hat{y}^{(1)} \approx \begin{bmatrix} y^{(1)} \\ \dots \\ y^{(m)} \end{bmatrix}$$

$$h(x) = g(X\theta) = \hat{y} \approx y$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Többváltozós logisztikus regresszió - NumPy

```
def __sigmoid(self, z):  
    return 1 / (1 + np.exp(-z) + self.eps)  
  
h = self.__sigmoid(np.dot(X, self.theta))  
  
loss = np.mean(-y * np.log(h + self.eps) - \  
                (1 - y) * np.log(1 - h + self.eps))  
  
gradient = np.dot(X.T, (h - y)) / y.size  
self.theta -= self.lr * gradient
```

lásd log.reg. notebook...

Többváltozós logisztikus regresszió - NumPy

```
def __sigmoid(self, z):  
    return 1 / (1 + np.exp(-z) + self.eps)
```

szigmoid fv. (vektoros)

```
h = self.__sigmoid(np.dot(X, self.theta))
```

hipotézis

```
loss = np.mean(-y * np.log(h + self.eps) - \\  
               (1 - y) * np.log(1 - h + self.eps))
```

költség

```
gradient = np.dot(X.T, (h - y)) / y.size  
self.theta -= self.lr * gradient
```

a gradiens módszer egy lépése

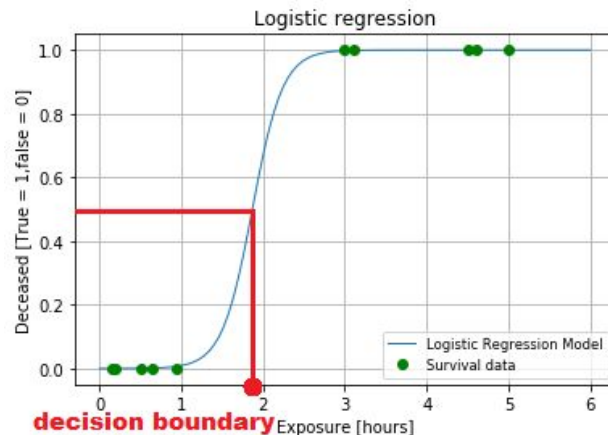
eps: epszilon a nullával osztást elkerülendő

Logisztikus regresszió

Mire jutottunk?

Logisztikus regresszióval **bináris klasszifikációt** tudunk végezni.

A gradiens módszerrel megtalálható a **globálisan optimális** megoldása a logisztikus regressziónak.



Logisztikus regresszió

Mire jutottunk?

Logisztikus regresszióval **bináris** klasszifikációt tudunk végezni.

A gradiens módszerrel megtalálható a **globálisan optimális** megoldása a logisztikus regressziónak.

Ne keverjük: A logisztikus regresszióval kategóriába tartozás valószínűségét becsüljük, ami folytonos érték. Azonban, a becslés 0 és 1 közé van szorítva, így nem használható tetszőleges regressziós feladat megoldására. Valószínűségek becslésére fogjuk használni, azaz (bináris) klasszifikációs feladatok megoldására.

