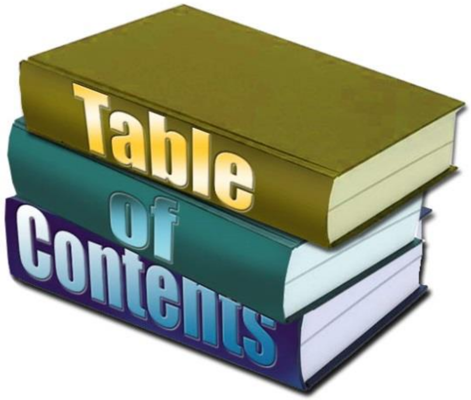
OSNOVE UMETNE INTELIGENCE

2018/19

regresija

linearne in lokalne metode

Pregled

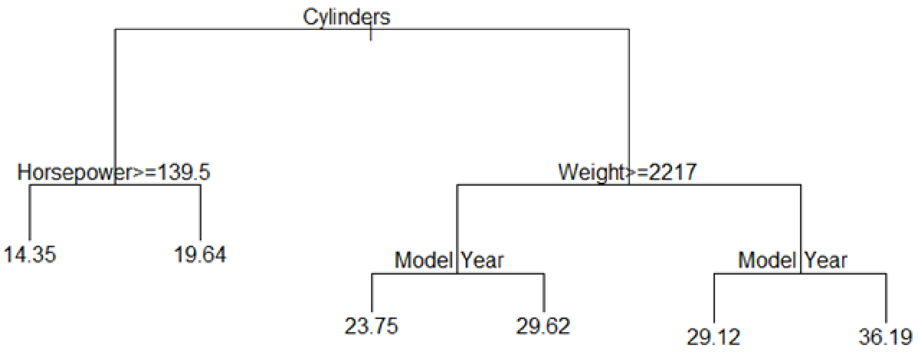


- strojno učenje
  - uvod v strojno učenje
  - učenje odločitvenih dreves
  - učenje dreves iz šumnih podatkov (rezanje dreves)
  - regresijska drevesa
  - linearni modeli
  - metoda k najbližjih sosedov

Regresijska drevesa

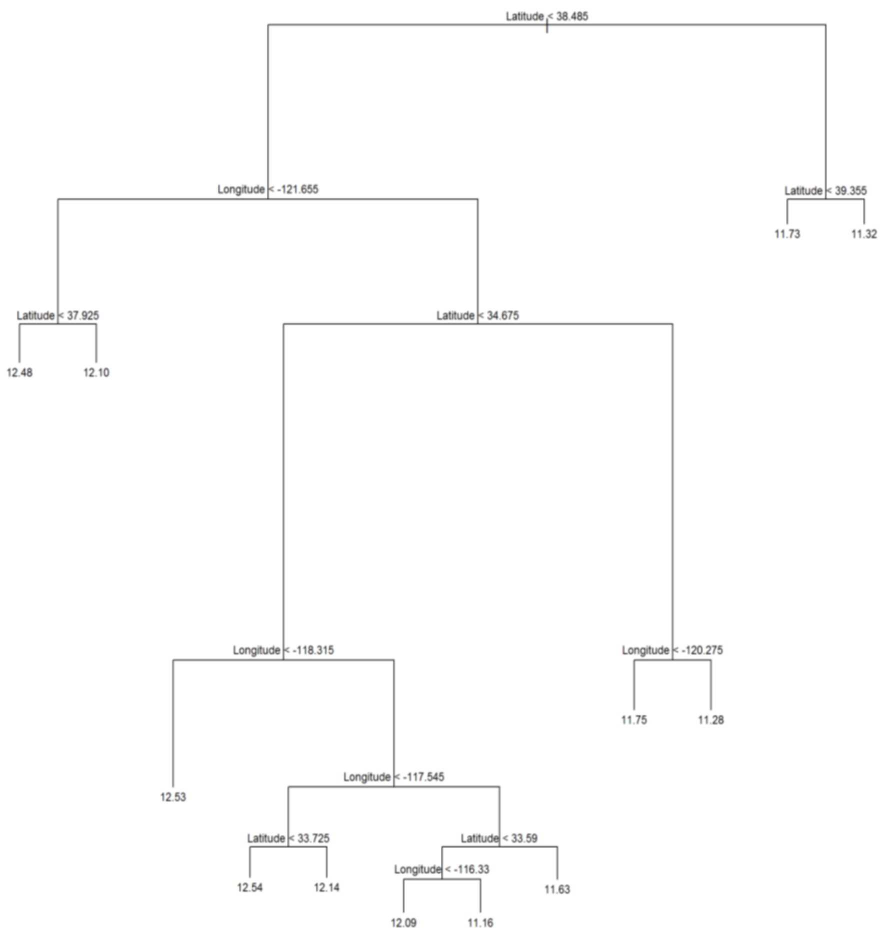
- zvezna ciljna spremenljivka – regresijski problem
- regresijska drevesa so podobna odločitvenim drevesom, le za regresijske probleme
- sistemi: CART (Breiman et al. 1984), RETIS (Karalič 1992), M5 (Quinlan 1993), WEKA (Witten and Frank, 2000)

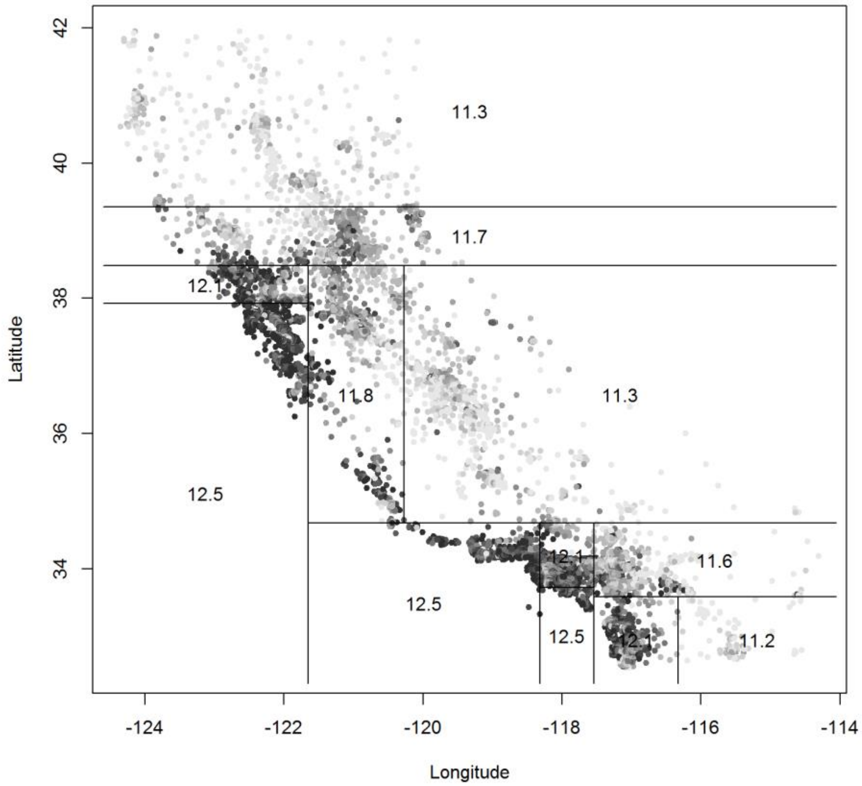
- listi v regresijskem drevesu predstavljajo:
  - predstavljajo povprečno vrednost označb ("razreda") primerov v listu
  - preprost napovedni model (npr. linearna regresija) za nove primere



Bora Beran, On Anything Data, domena MPG

Regresijska drevesa





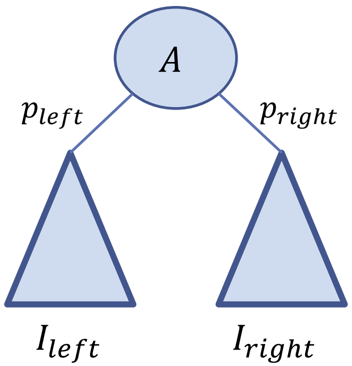
Gradnja regresijskih dreves

- atribut delimo glede na izbrano mejno vrednost
- drugačna mera za merjenje nedoločenosti/nečistoče: srednja kvadratna napaka v vozlišču v:

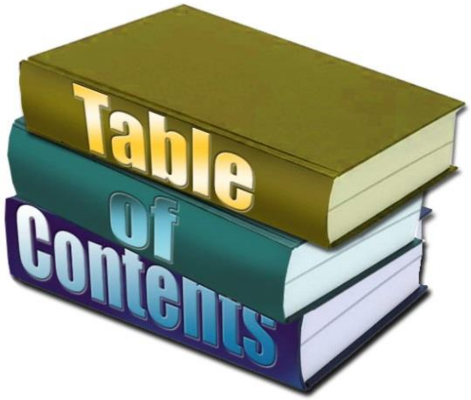
$$MSE(v) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

- cilj: minimiziramo rezidualno nedoločenost po delitvi primerov glede na vrednosti atributa A
- pričakovana rezidualna nečistost

$$I_{res}(A) = p_{left} \cdot I_{left} + p_{right} \cdot I_{right}$$



Pregled



- strojno učenje
  - uvod v strojno učenje
  - učenje odločitvenih dreves
  - učenje dreves iz šumnih podatkov (rezanje dreves)
  - regresijska drevesa
  - linearni modeli
  - metoda k najbližjih sosedov

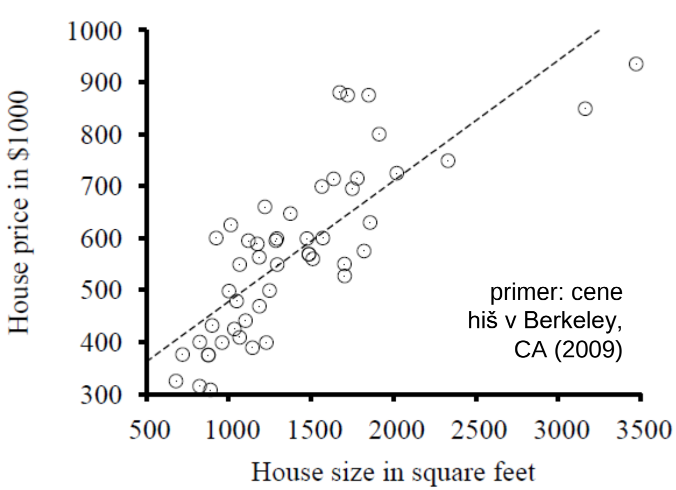
Linearni modeli

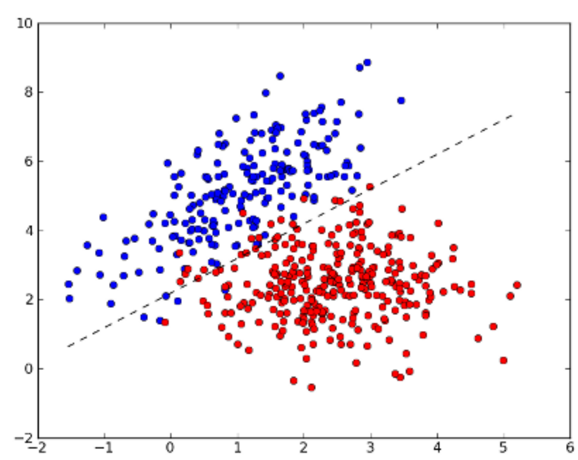
- uporaba pri **klasifikaciji** (kot separator razredov) in **regresiji** (kot prileganje skozi podane točke)
- linearni model z **eno odvisno** spremenljivko (angl. *univariate linear model*):

$$h(x) = w_1 x + w_0$$

  w_0 in w_1 sta **uteži** (angl. *weights*) spremenljivk (koeficienta)

- **linearna regresija**: postopek iskanja funkcije $h(x)$ (oziroma uteži w_0 in w_1), ki se najbolje prilega učnim podatkom





Linearna regresija

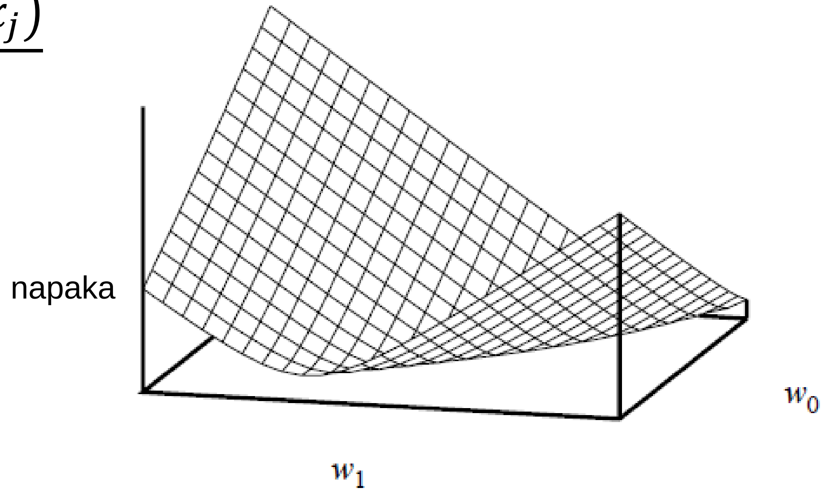
- optimizacijo izvedemo z minimizacijo srednje kvadratne napake:

$$napaka(h) = \sum_{j=1}^{N} \left(y_j - (w_1 x_j + w_0)\right)^2$$

- prostor koeficientov je konveksen, lokalni minimumi ne obstajajo (samo globalni)
- obstaja analitična rešitev:

$$w_1 = \frac{N\left(\sum x_j y_j\right) - \left(\sum x_j\right)\left(\sum y_j\right)}{N\left(\sum x_j^2\right) - \left(\sum x_j\right)^2}$$

$$w_0 = \frac{\sum y_j - w_1\left(\sum x_j\right)}{N}$$



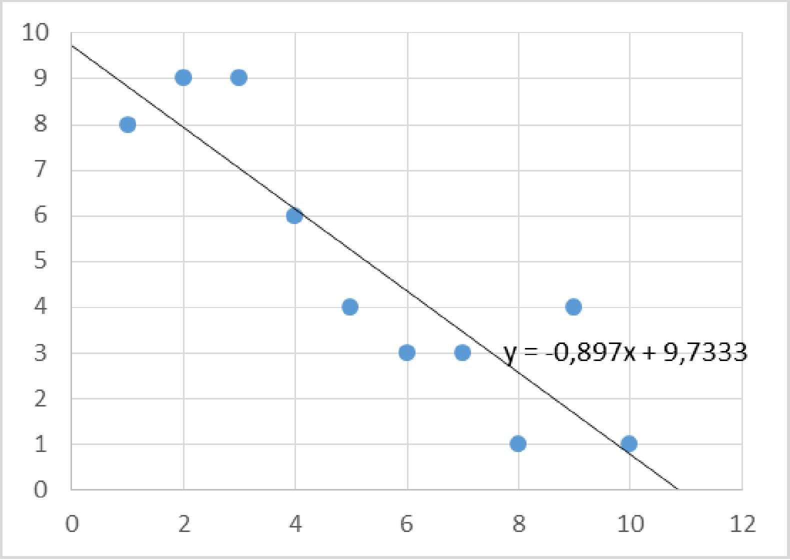
Linearna regresija

- primer linearne regresije

x_j	y_j	$x_j y_j$	x_j^2
1	8	8	1
2	9	18	4
3	9	27	9
4	6	24	16
5	4	20	25
6	3	18	36
7	3	21	49
8	1	8	64
9	4	36	81
10	1	10	100
$\sum x_j = 55$	$\sum y_j = 48$	$\sum x_j y_j = 190$	$\sum x_j^2 = 385$

$$w_1 = \frac{N\left(\sum x_j y_j\right) - \left(\sum x_j\right)\left(\sum y_j\right)}{N\left(\sum x_j^2\right) - \left(\sum x_j\right)^2} = \frac{10 \cdot 190 - 55 * 48}{10 \cdot 385 - 55^2} = = -0{,}897$$

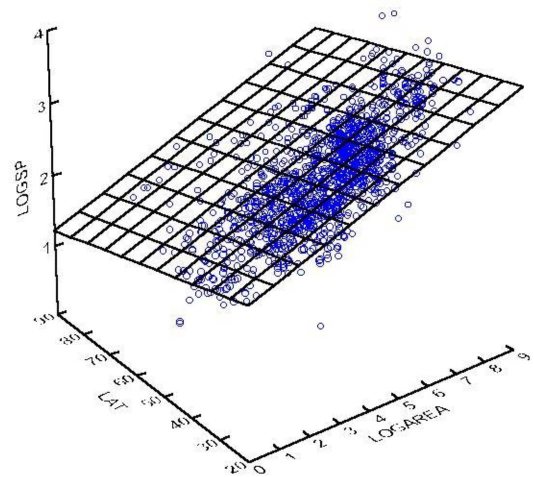
$$w_0 = \frac{\sum y_j - w_1\left(\sum x_j\right)}{N} = \frac{48 - (-0{,}897) \cdot 55}{10} = 9{,}733$$



Posplošitev v več dimenzij

- možna je posplošitev v **višje število dimenzij** – več neodvisnih spremenljivk (atributov) (angl. *multivariate linear regression*)
  $h(x) = w_0 + \sum_i w_i x_{j,i}$
  kjer so w_i uteži (koeficienti), $x_{j,i}$ pa i-ta spremenljivka (atribut) primera x_j
- uteži lahko določimo **analitično**: $\boldsymbol{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$
  kjer je $\boldsymbol{X}$ matrika s podatki (vrstice – učni primeri, stolpci – atributi), $\boldsymbol{y}$ pa vektor z vrednostmi odvisnih spremenljivk primerov
- v praksi se odločamo za iskanje koeficientov
  z **gradientnim spustom**

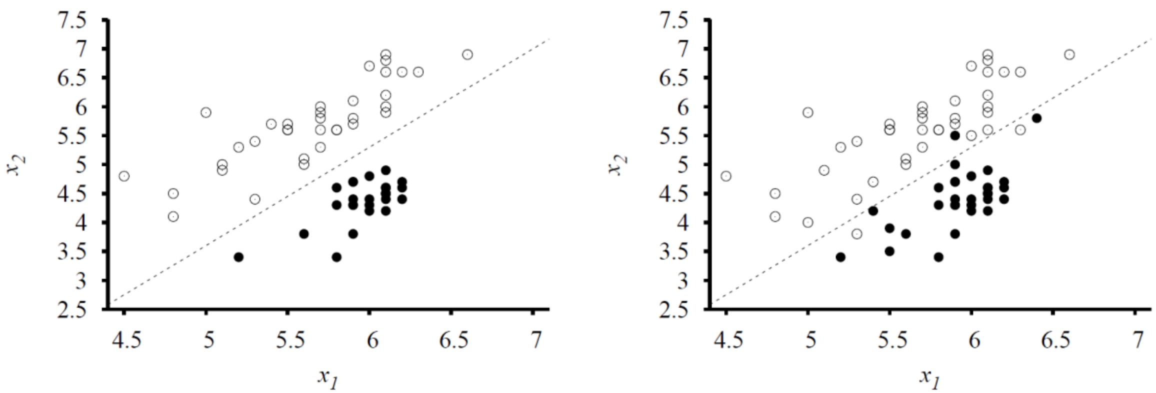
  $\boldsymbol{w} \leftarrow$ naključna začetna rešitev
  ponavljaj do konvergence
      za vsak w_i v $\boldsymbol{w}$:
      $w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} napaka(\boldsymbol{w})$



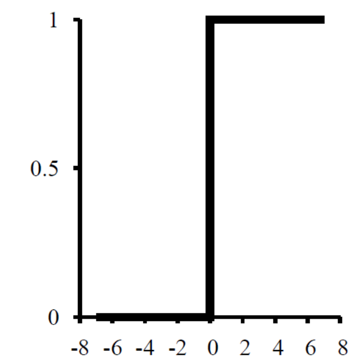
- problem s pretiranim prilagajanjem, regularizacija

Linearni modeli pri klasifikaciji

- linearni model se uporablja za ločevanje primerov, ki pripadajo različnim razredom
- iščemo **odločitveno mejo** (angl. *decision boundary*) oz. **linearni separator** (obstaja samo pri linearno ločljivih problemih)
- za spodnji primer je linearno separator lahko funkcija $-4.9 + 1.7x_1 - x_2 = 0$
- hipoteza je torej: $h(\boldsymbol{x}) = prag(\boldsymbol{w} \cdot \boldsymbol{x})$, kjer $prag(z) = \begin{cases} 1 & z \geq 0 \\ 0 & sicer \end{cases}$



primer linearno ločljivega in neločljivega problema (domena o potresih), x_1 - jakost v tleh, x_2 - jakost na površju



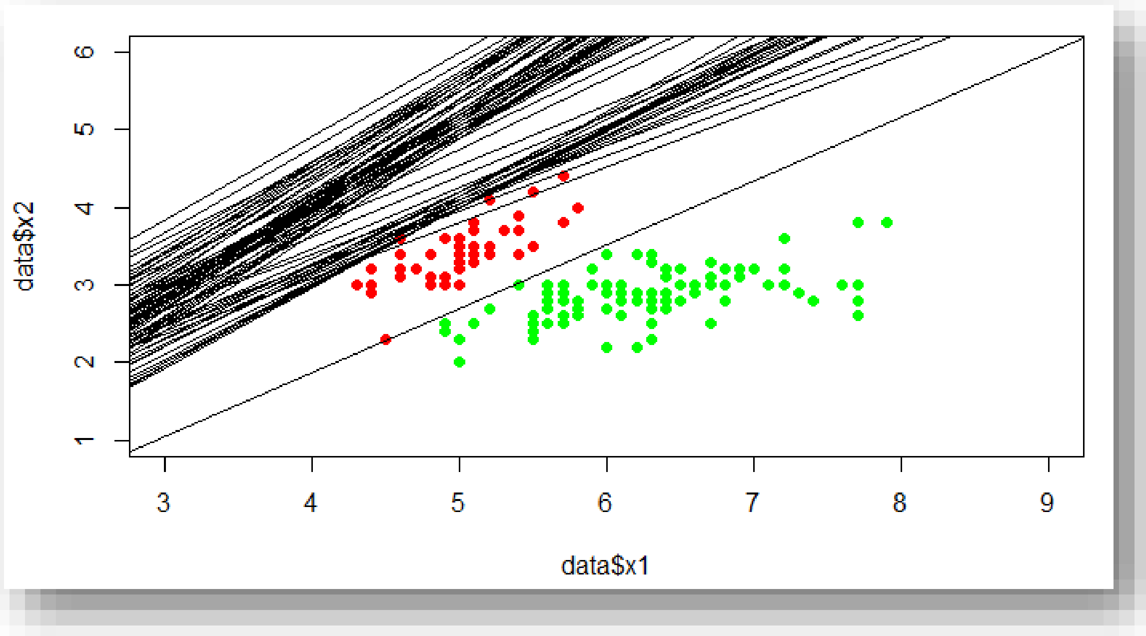
stopničasta pragovna funkcija

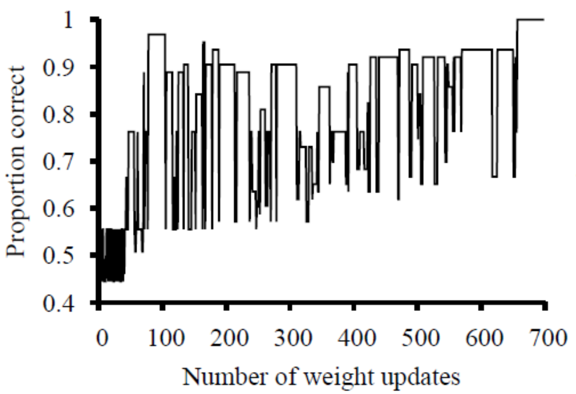
Linearni modeli pri klasifikaciji

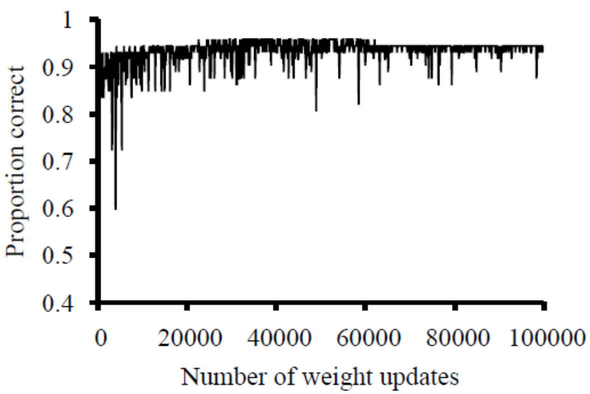
- možnih ustreznih premic je več
- preprosto iskanje rešitve – **stohastični gradientni spust** s posodabljanjem uteži
- za vsak učni primer $(\boldsymbol{x}, y)$ izvedi posodobitev uteži:
  $$w_i \leftarrow w_i + \alpha\big(y - h(\boldsymbol{x})\big) \times x_i$$
  kjer so w_i uteži (koeficienti), α pa vpliva na hitrost spremembe (korak)
- intuicija:
  - če $y = h(\boldsymbol{x})$, potem se w_i ne spremeni
  - če $y = 1$ in $h(\boldsymbol{x}) = 0$ (**prenizka** vrednost hipoteze), potem se za pozitiven x_i utež **poveča** in za negativen x_i utež **zmanjša**
  - če $y = 0$ in $h(\boldsymbol{x}) = 1$ (**previsoka** vrednost hipoteze), potem se za pozitiven x_i utež **zmanjša** in za negativen x_i utež **poveča**
- algoritem lahko pri ustreznem α najde optimalno rešitev tudi za linearno neločljive podatke
- smiselna izboljšava: logistična pragovna funkcija

Linearni modeli pri klasifikaciji

- demo

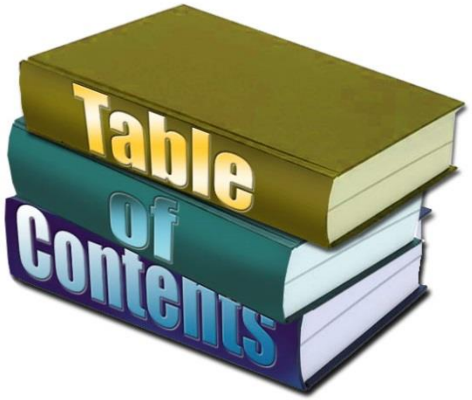






konvergenca algoritma pri linearno ločljivih podatkih (levo) in linearno neločljivih podatkiih (desno)

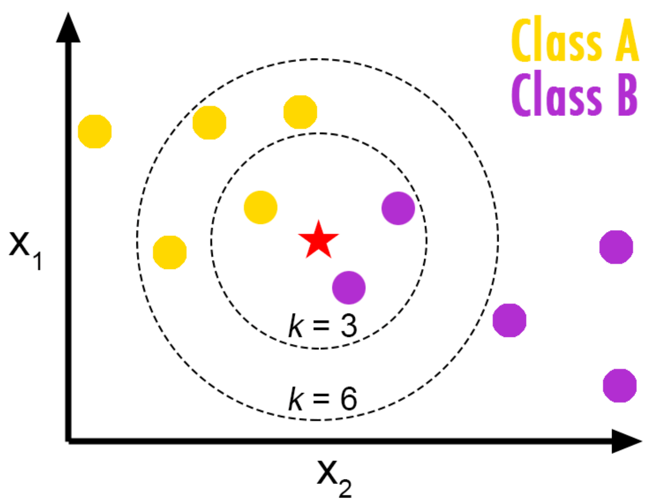
Pregled



- strojno učenje
  - uvod v strojno učenje
  - učenje odločitvenih dreves
  - učenje dreves iz šumnih podatkov (rezanje dreves)
  - regresijska drevesa
  - linearni modeli
  - metoda k najbližjih sosedov

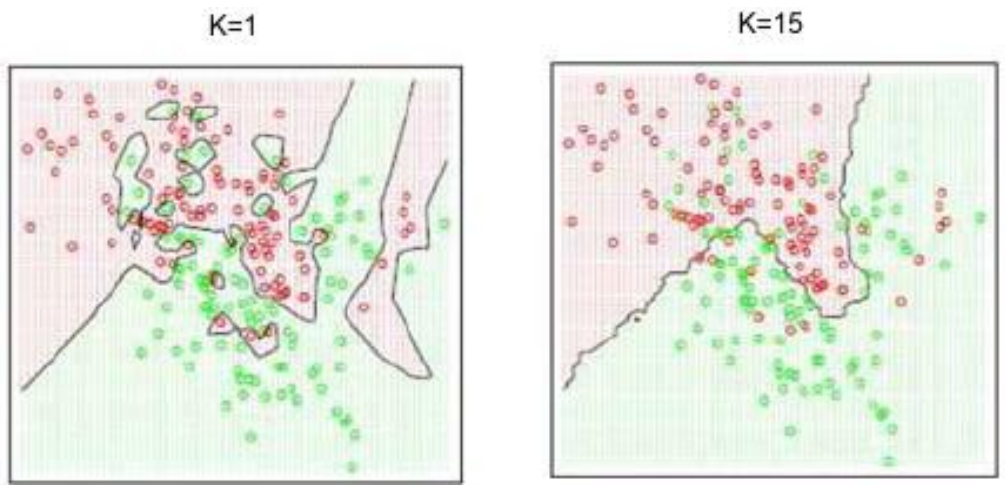
Metoda k najbližjih sosedov

- angl. k *nearest neighbors*
- lastnosti:
  - **neparametrična** metoda (ne ocenjuje parametrov izbranega modela)
  - učenje na podlagi **posameznih primerov** (angl. *instance-based learning*)
  - **leno učenje** (angl. *lazy learning*): z učenjem odlaša vse do povpraševanja o novem primeru
- ideja: ob vprašanju po vrednosti odvisne spremenljivke za novi primer:
  - poišči **k primerov**, ki so **najbližji** glede na podano **mero razdalje**
  - napovej
    - pri klasifikaciji: npr. večinski razred med sosedi
    - pri regresiji: npr. povprečno vrednost/mediano označb sosedov
- v izogib neodločenemu glasovanju za večinski razred pri klasifikaciji običajno izberemo, da je *k* liho število



Metoda k najbližjih sosedov

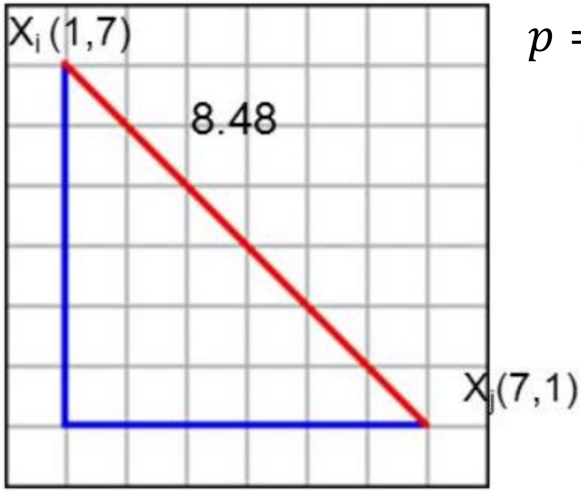
- pomembna je izbira ustreznega k:
  - premajhen k: pretirano prilagajanje
  - prevelik k: prešibko posploševanje (pri $k = N$: napoved večinskega razreda)
  - v praksi običajno: $k = 5$



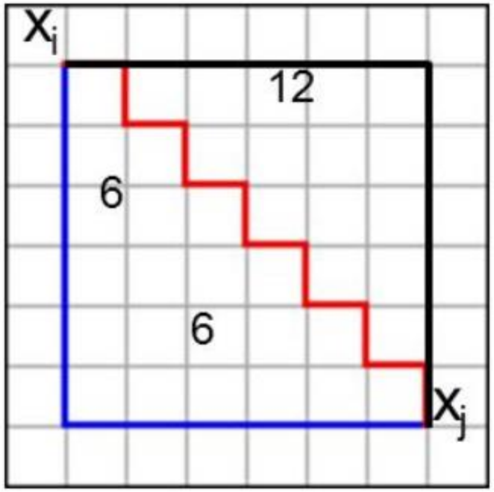
Hastie, Tibshirani, Friedman: Elements of Statistical Learning, 2009

Metoda k najbližjih sosedov

- razdaljo običajno merimo z razdaljo Minkowskega: $L^p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_k |x_{i,k} - x_{j,k}|^p\right)^{\frac{1}{p}}$
  - za $p = 2$ je to evklidska razdalja: $L^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\sum_k (x_{i,k} - x_{j,k})^2}$
  - za $p = 1$ je to manhattanska razdalja: $L^1(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_k |x_{i,k} - x_{j,k}|$
- za zvezne atribute: razlika med vrednostima atributov (normalizacija?)
- za diskretne atribute: Hammingova razdalja (število diskretnih diskretnih atributov z ujemajočimi vrednostmi pri obeh primerih)



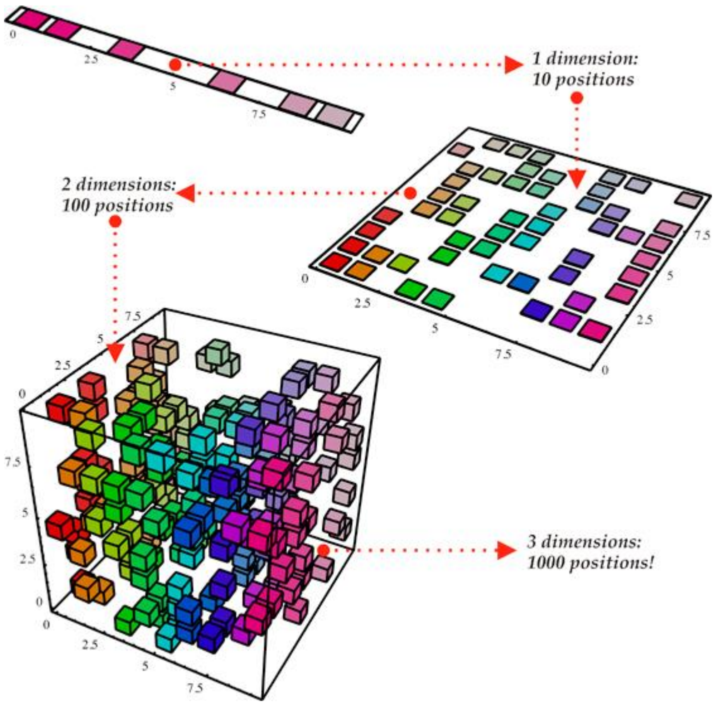
$p = 2$

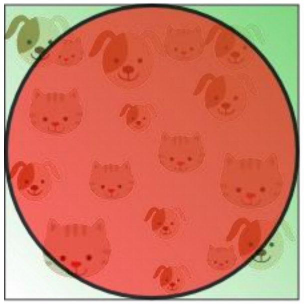


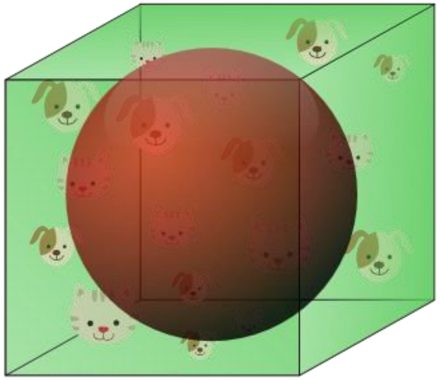
$p = 1$

Opombe

- vpliv intervala vrednosti na izračunano razdaljo vpliva na najdene najbližje sosede → potrebna **normalizacija**
- pri velikem številu dimenzij lahko postanejo primeri zelo oddaljeni – **prekletstvo dimenzionalnosti** (angl. *the curse of dimensionality*)
- implementacije iskanja najbližjih sosedov: $O(N)$, $O(logN)$, $O(1)$







Naivni Bayesov klasifikator