Umetna inteligenca

Osnovne metode strojnega učenja

klasifikacija:

- Odločitvena drevesa
- Klasifikator z najbližjimi sosedi
- naivni Bayesov klasifikator
- Diskriminantne funkcije
- metoda podpornih vektorjev (SVM)
- Naključni gozdovi
- Umetne nevronske mreže
- Globoke nevronske mreže

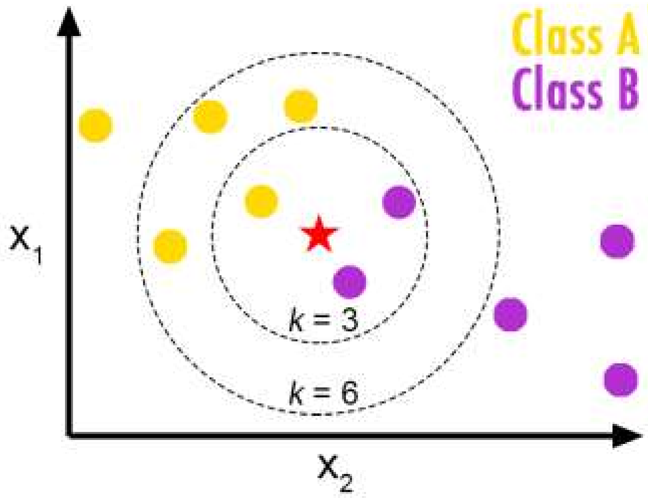
regresija:

- Regresijska drevesa
- Lokalno utežena regresija
- Linearna regresija
- Regresijske funkcije
- Metoda podpornih vektorjev
- Naključni gozdovi
- Umetne nevronske mreže
- Globoke nevronske mreže

Metoda k najbližjih sosedov

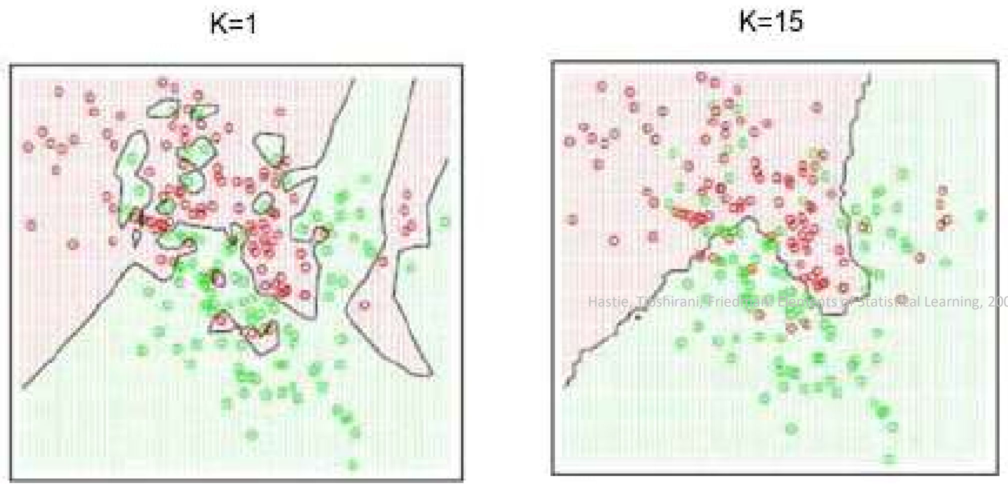
Metoda k najbližjih sosedov

- angl. k *nearest neighbors*
- lastnosti:
  - 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

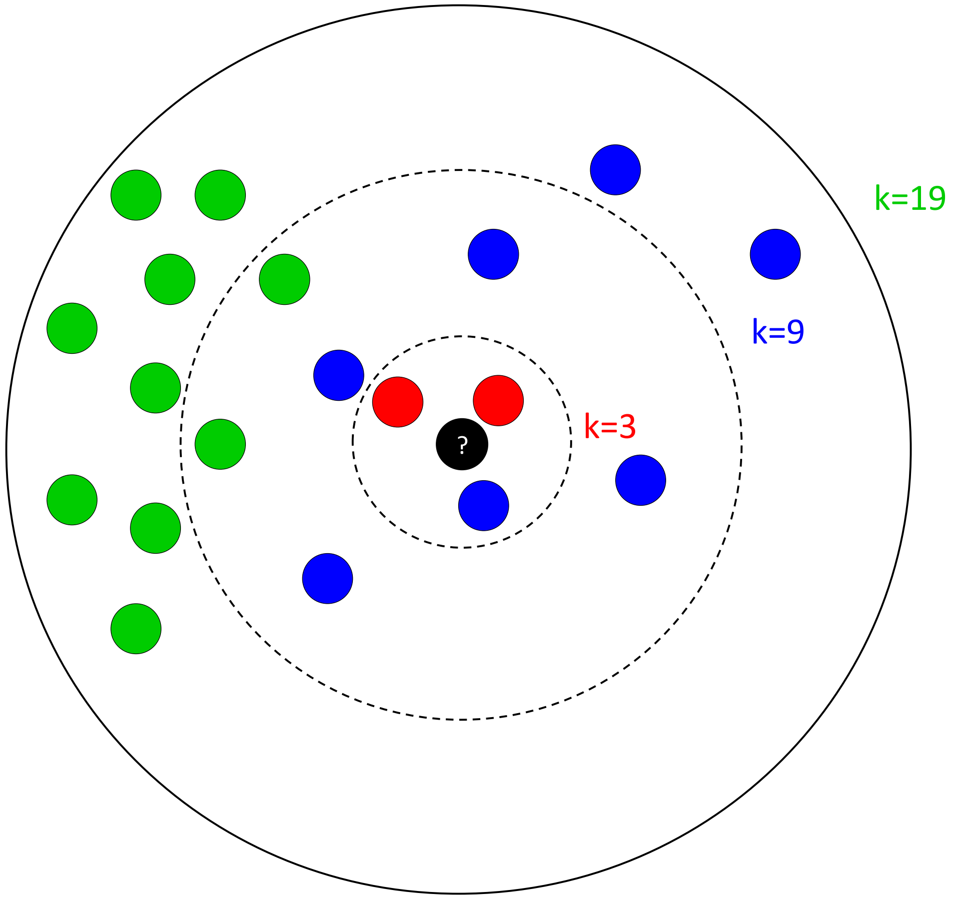
- k ne določa velikosti okolice novega primera, ampak se okolica dinamično spreminja
- pomembna je izbira ustreznega k:
  - premajhen k: pretirano prileganje
  - prevelik k: prešibko posploševanje (pri $k = N$: napoved večinskega razreda)
  - v praksi običajno: $k = 5,\ 9,\ 15$



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

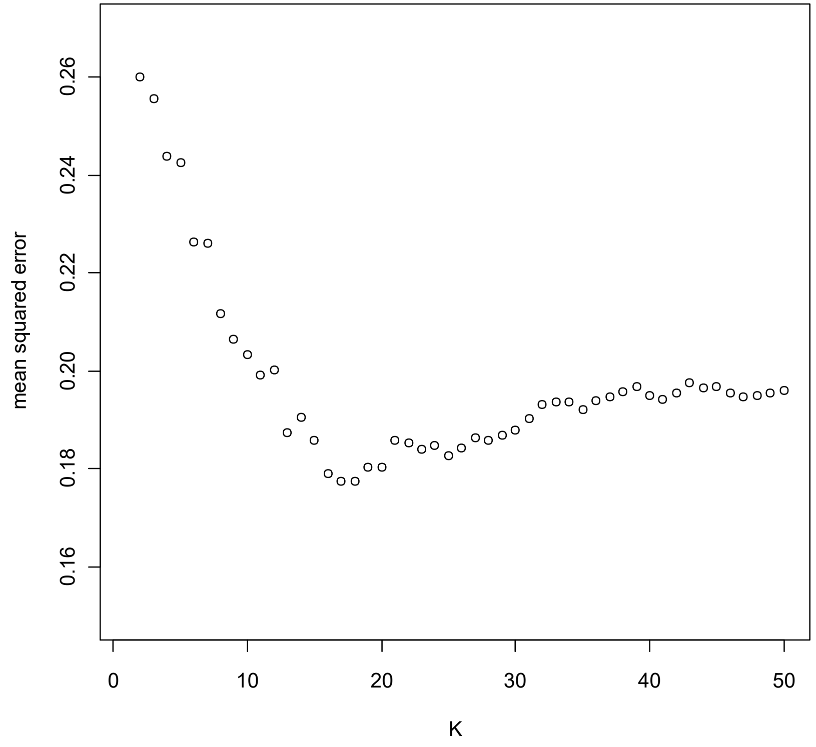
k-Nearest Neighbors

Different values of *k* can produce very different results:



k-Nearest Neighbors

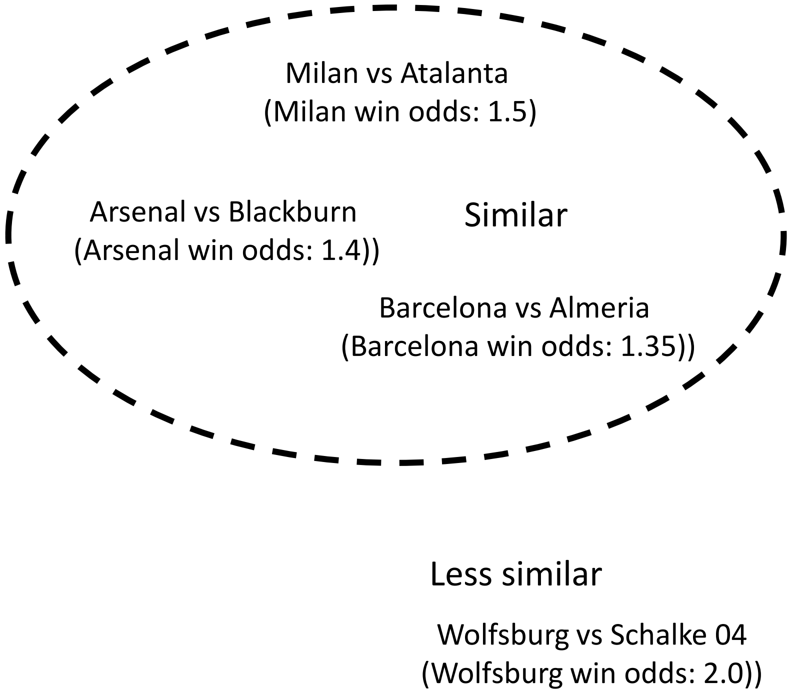
Example:
- Soccer data, 100 test instances, 200 training instances
- Predicting home win from nearest neighboring past matches
- Similarity measure: absolute difference in home win probabilities (from bookmaker)



lower *k* => more variance in predictions (noise)

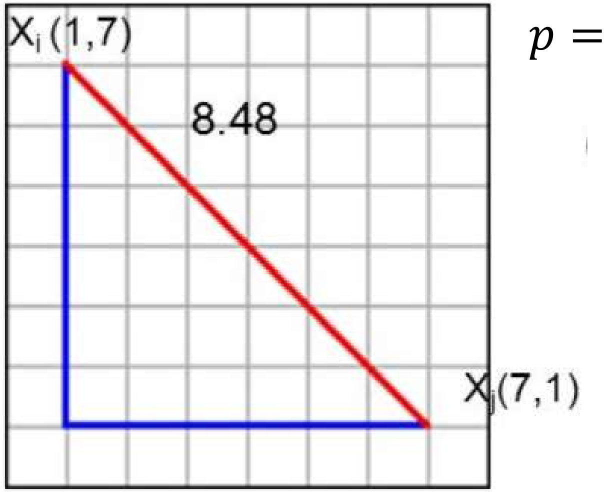
Tradeoff

higher *k* => a larger bias (less distinct boundaries)

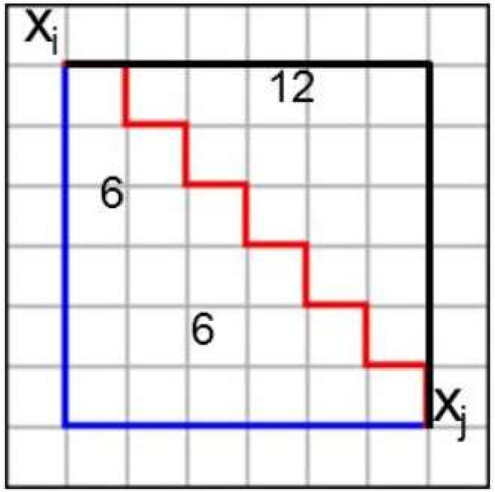


Metoda k najbližjih sosedov

- 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: 0 (enaki vrednosti) in 1 (različni vrednosti)
- Samo diskretni atributi: Hammingova razdalja (število diskretnih atributov z različnimi vrednostmi pri obeh primerih)



$p = 2$



$p = 1$

S kvaliteto atributov utežena razdalja

Evklidska razdalja: $D(u_l, u_j) = \sqrt{\Sigma_{i=1}^{a} d(v^{(i,l)}, v^{(i,j)})^2}$

kjer za zvezni atribut A_i velja:

$$d(v^{(i,l)}, v^{(i,j)}) = |v^{(i,l)} - v^{(i,j)}|$$

in za diskretnega:

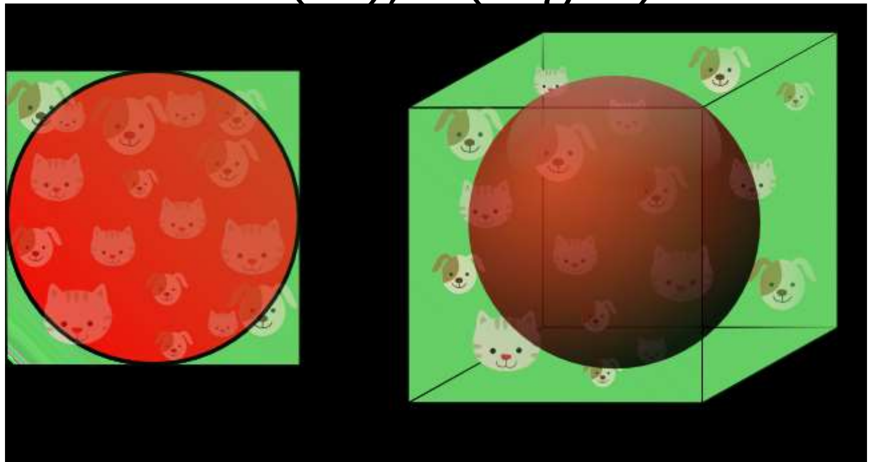
$$d(v^{(i,l)}, v^{(i,j)}) = \begin{cases} 0, & v^{(i,l)} = v^{(i,j)} \\ 1, & v^{(i,l)} \neq v^{(i,j)} \end{cases}$$

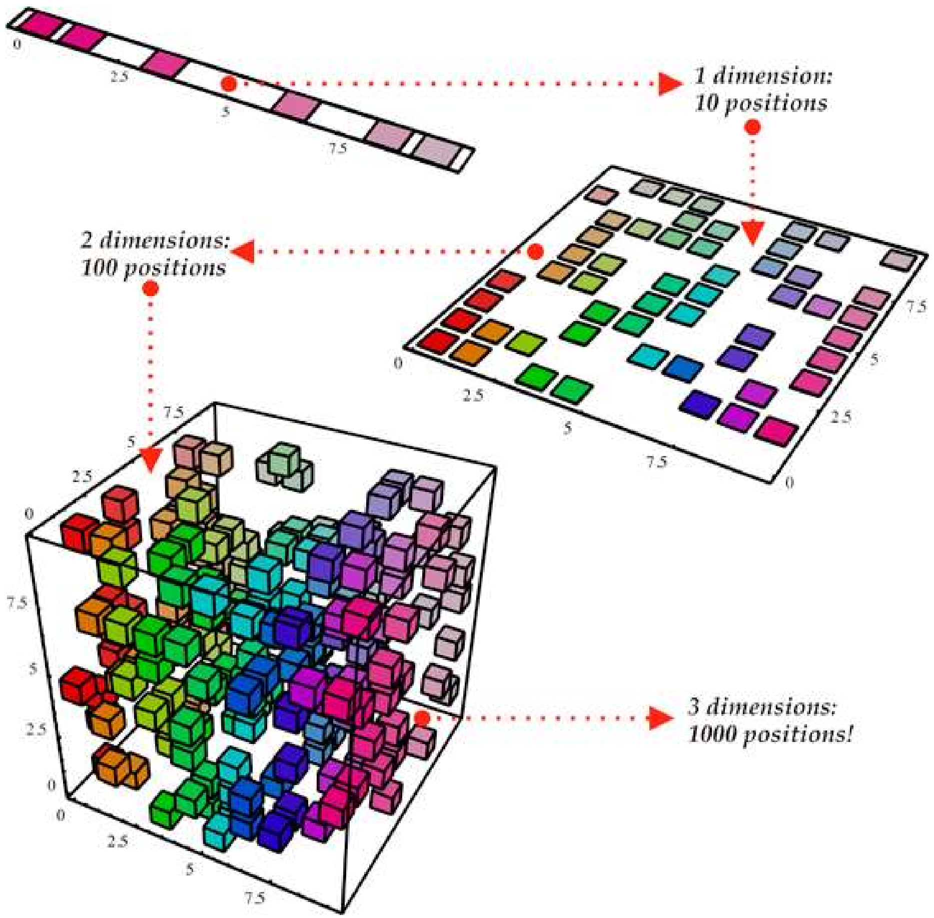
Uteževanje vpliva atributov na celotno razdaljo:

$$D(u_l, u_j) = \sqrt{\sum_{i=1}^{a} q(A_i) d(v^{(i,l)}, v^{(i,j)})^2}$$

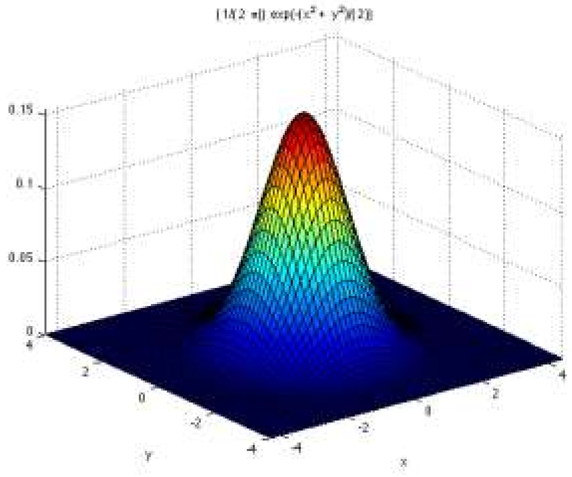
Pomembno:

- 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: $O(N)$, $O(logN)$





Z razdaljo uteženih *k*-najbližjih sosedov

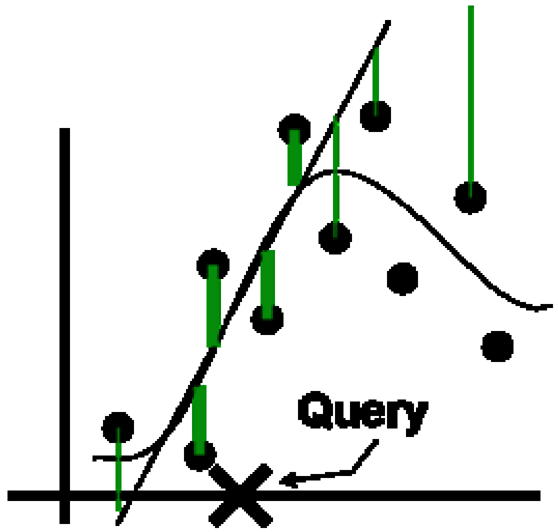


Namesto, da vsi sosedje vplivajo enako na napoved:

- Bližji imajo večjo težo
- Bolj oddaljeni imajo manjšo težo
- *K* je lahko večji
- Če vpliv z razdaljo eksponentno pada, je *K* lahko poljubno velik (ga ne potrebujemo), saj vpliv z razdaljo hitro postane zanemarljiv

Lokalno utežena regresija

- K-NN povpreči vrednost ciljne spremenljivke *k* najbližjih sosedov
- Namesto povprečenja lahko uporabimo poljubno regresijsko funkcijo skozi *k* najbližjih sosedov.
- Najpogosteje linearna lokalno utežena regresija, ki je linearna regresija, uporabljena na *k* najbližjih sosedih
- nevarnost prevelikega prileganja
- časovna zahtevnost!



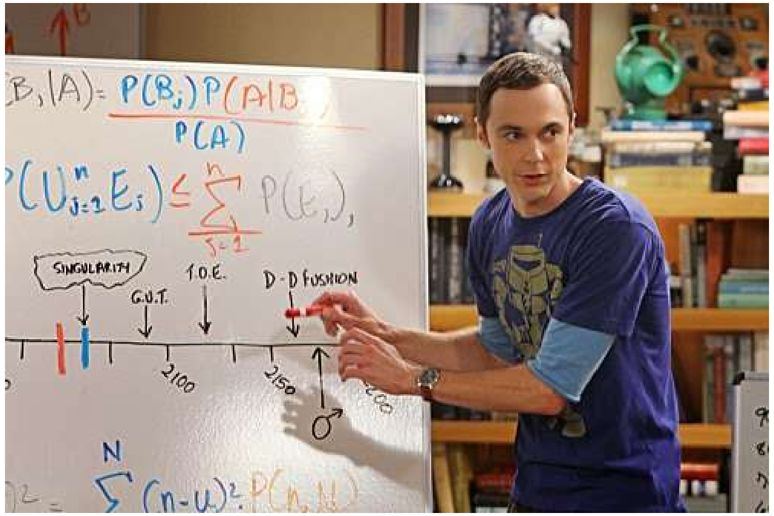
Naivni Bayesov klasifikator

Naivni Bayesov klasifikator

- Thomas Bayes, 1702 – 1761
- opomnik iz teorije o verjetnosti:

$$P(AB) = P(A|B) \cdot P(B)$$
$$P(AB) = P(B|A) \cdot P(A)$$
$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$
$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

Bayesovo pravilo

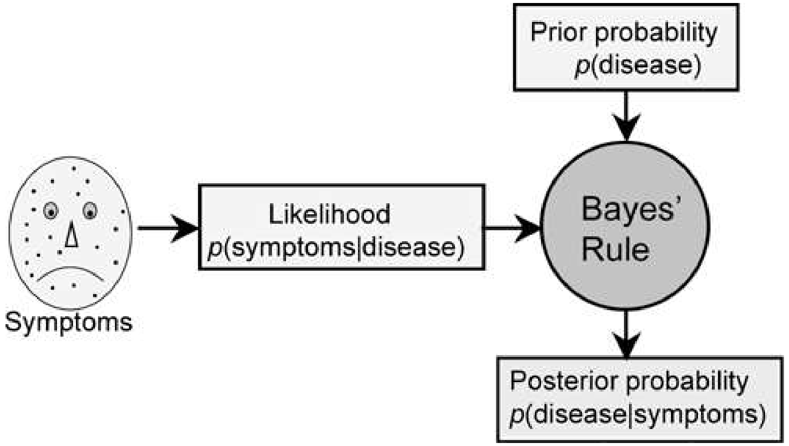


Naivni Bayesov klasifikator

- aplikacija v medicini:

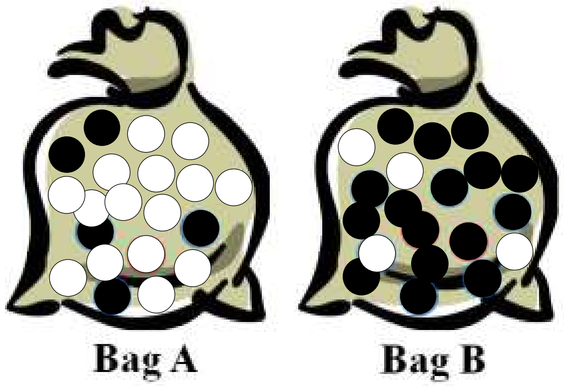
$$P(hipoteza|opažanje) = \frac{P(opažanje|hipoteza) \cdot P(hipoteza)}{P(opažanje)}$$

- zdravniki razpolagajo z vzročno in statistično informacijo:
  - verjetnost izraženih simptomov pri neki bolezni - $P(opažanje|hipoteza)$
  - verjetnost določene bolezni - $P(hipoteza)$
  - verjetnost določenega simptoma - $P(opažanje)$
- Bayesovo pravilo nam izraža **diagnostično pogojno verjetnost** $P(hipoteza|opažanje)$ na podlagi **vzročne pogojne verjetnosti** $P(opažanje|hipoteza)$



Vaja

- dve vrsti vrečk s frnikulami:
  - 4 vrečke tipa A (vsaka 5 črnih, 15 belih frnikul
  - 1 vrečka tipa B (16 črnih, 4 bele frnikule)



- možna vprašanja:
  - Kakšna je verjetnost, da je to vrečka tipa B?
    $P(B) = ?$
  - Kakšna je verjetnost, da naključno izberemo črno frnikulo, če izbiramo iz vrečke tipa B?
    $P\left(Č \middle| B\right) = ?$
  - Naključno izberemo eno izmed vrečk in iz nje naključno izberemo frnikulo. Kakšna je verjetnost, da smo izbrali črno frnikulo iz vrečke tipa B?
    $P\left(BČ\right) = P(B) \cdot P\left(Č \middle| B\right) = ?$
  - Naključno izberemo eno izmed vrečk in iz nje naključno izberemo frnikulo. Kakšna je verjetnost, da smo izbrali črno frnikulo?
    $P\left(Č\right) = P(B) \cdot P\left(Č \middle| B\right) + P(A) \cdot P(Č|A)$

Vaja

- Ena vrečka ima poškodovan ovoj tako, da se skozi njega vidi črna frnikula. Kakšna je verjetnost, da je to vrečka tipa B?
  $P(B|\text{Č}) =?$
- B = hipoteza, Č = evidenca, opažanje
- verjetnost $P(B|\text{Č})$ lahko določimo iz drugih bolj očitnih verjetnosti z Bayesovo formulo:

$$P(B|\text{Č}) = \frac{P(B) \cdot P(\text{Č}|B)}{P(\text{Č})}$$

- $P(B) = \frac{1}{5} = 0{,}2$
  $P(\text{Č}|B) = \frac{16}{20} = 0{,}8$
  $P(\text{Č}) = \frac{4 \cdot 5 + 1 \cdot 16}{5 \cdot 20} = 0{,}444$
- $P(B|\text{Č}) = \frac{0{,}2 \cdot 0{,}8}{0{,}444} = 0{,}360$

dve vrsti vrečk s frnikulami:

- 4 vrečke tipa A (vsaka 5 črnih, 15 belih frnikul)
- 1 vrečka tipa B (16 črnih, 4 bele frnikule)

Naivni Bayes v strojnem učenju

- evidenca → atributi
  hipoteza → razred
- zanima nas, kakšna je verjetnost razreda C pri podanih vrednostih atributov $A_1 = X_1, A_2 = X_2, \dots, A_n = X_n$:

$$P(C|X_1 X_2 \dots X_n) = \frac{P(C) \cdot P(X_1 X_2 \dots X_n|C)}{P(X_1 X_2 \dots X_n)}$$

Naivni Bayes v strojnem učenju

$$P(C|X_1X_2\dots X_n) = \frac{P(C)\cdot P(X_1X_2\dots X_n|C)}{P(X_1X_2\dots X_n)}$$

- $P(X_1X_2\dots X_n|C) = P(X_1|C)\cdot P(X_2\dots X_n|X_1C) =$
  $= P(X_1|C)\cdot P(X_2|X_1C)\cdot P(X_3\dots X_n|X_1X_2C) =$
  $= P(X_1|C)\cdot P(X_2|X_1C)\cdot P(X_3|X_1X_2C)\cdot \dots \cdot P(X_n|X_1X_2\dots X_{n-1}C)$
- $P(X_1X_2\dots X_n) = P(X_1|X_2\dots X_n)\cdot P(X_2|X_3\dots X_n)\cdot \dots \cdot P(X_{n-1}|X_n)\cdot P(X_n)$

- potrebujemo veliko število pogojnih verjetnosti, katerih poznavanje je v praksi težavno
- število kombinacij pogojnih verjetnosti je glede na zaloge vrednosti atributov $X_1X_2\dots X_n$ eksponentno
- praktična rešitev: naivni Bayesov klasifikator

Naivni Bayes v strojnem učenju

- predpostavimo, da so atributi med seboj **neodvisni pri danem razredu** in poenostavimo:
  $P(X_1 X_2 \dots X_n | C) = P(X_1|C) \cdot P(X_2|X_1 C) \cdot \dots \cdot P(X_n | X_1 X_2 \dots X_{n-1} C)$
  $P(X_1 X_2 \dots X_n) = P(X_1|X_2 \dots X_n) \cdot P(X_2|X_3 \dots X_n) \cdot \dots \cdot P(X_{n-1}|X_n) \cdot P(X_n)$

$$P(X_1 X_2 \dots X_n | C) \approx P(X_1|C) \cdot P(X_2|C) \cdot \dots \cdot P(X_n|C)$$
$$P(X_1 X_2 \dots X_n) \approx P(X_1) \cdot P(X_2) \cdot \dots \cdot P(X_{n-1}) \cdot P(X_n)$$

- približki so dobri, če so atributi med seboj dovolj neodvisni
- velja torej:

$$P(C|X_1 X_2 \dots X_n) = \frac{P(C) \cdot P(X_1 X_2 \dots X_n | C)}{P(X_1 X_2 \dots X_n)} = \frac{P(C) \cdot \prod_i P(X_i|C)}{\prod_i P(X_i)}$$

$$P(C|X_1 X_2 \dots X_n) = \frac{P(C) \cdot \prod_i P(C|X_i)}{\prod_i P(C)}$$

Naivni Bayes v strojnem učenju

- Bayesov klasifikator: primer **klasificiramo v razred, ki je najbolj verjeten**:

- **učenje**: ocenimo verjetnosti $P(C_k)$ in $P(C_k|X_i)$ za vse razrede C_k in vrednosti atributov X_i
- **napovedovanje**: Izračunamo verjetnost za vse razrede pri danih vrednostih atributov

- *opomba: s poenostavitvijo formule lahko izgubimo verjetnostno interpretacijo (verjetnosti razredov se ne seštejejo več v 1). Problem rešujemo npr. z normalizacijo rezultatov.*

Ocenjevanje verjetnosti pri NB

pogojna neodvisnost atributov pri danem razredu:

$$P(r_k|V) = P(r_k) \prod_{i=1}^{a} \frac{P(r_k|v_i)}{P(r_k)}$$

apriorne verjetnosti: Laplaceov zakon zaporednosti:

$$P(r_k) = \frac{N_k + 1}{N + n_0}$$

pogojne verjetnosti: m-ocena:

$$P(r_k|v_i) = \frac{N_{k,i} + mP(r_k)}{N_i + m}$$

Lastnosti Naivnega Bayesa

- Naivni Bayesov klasifikator vedno uporabi vse znane atribute
- neznane vrednosti: primere izpustimo
- zvezni atribut najprej (*mehko*) diskretiziramo
- pogojna neodvisnost pogosto sprejemljiva (Simptomi pri pacientih so odvisni od bolezni, Prikazovalnik cifer LCD: okvare žarnic neodvisne)
- Ocenjevanje verjetnosti relativno zanesljivo: ni prevelikega prileganja učni množici.
- Če neodvisnost ni popolna: „rezerva“ ne pokvari vrstnega reda verjetnosti razredov.
- Močne odvisnosti med atributi: naivni Bayes odpove
- Razlaga odločitev naivnega Bayesa:

$$-\log_2 P(r_k|V) = -\log_2 P(r_k) - \sum_{i=1}^{a} \left(\log_2 P(r_k|v_i) - \log_2 P(r_k)\right)$$

apriorna količina informacije, da primer razvrstimo v razred, minus vsota informacijskih prispevkov posameznih atributov.

- Inkrementalno učenje

Primer

- Zajeli smo podatke za 1000 sadežev, ki so lahko bodisi: *banana*, *pomaranča* ali *drugi sadež* (= vrednosti **razreda**). Za vsakega izmed sadežov smo izmerili, ali je *podolgovat*, *sladek* in *rumen* (= **atributi**). Meritve smo zapisali v tabelo:

sadež	podolgovat		sladek		rumen		skupaj
	da	ne	da	ne	da	ne	
banana	400	100	350	150	450	50	500
pomaranča	0	300	150	150	300	0	300
drugo	100	100	150	50	50	150	200
	500	500	650	350	800	200	1000

- iz tabele lahko razberemo različne verjetnosti, npr.:
  - verjetnosti razredov: $P(banana) = \frac{500}{1000} = 0{,}5$, $P(pomaranča) = 0{,}3$, $P(drugo) = 0{,}2$
  - pogojne verjetnosti: $P(banana|dolg) = \frac{4}{5} = 0{,}8$

$$P(pomaranča|dolg) = \frac{0}{5} = 0{,}0$$

A simple explanation of Naive Bayes Classification, https://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification

Primer

sadež	podolgovat		sladek		rumen		skupaj
	da	ne	da	ne	da	ne	
banana	400	100	350	150	450	50	500
pomaranča	0	300	150	150	300	0	300
drugo	100	100	150	50	50	150	200
	500	500	650	350	800	200	1000

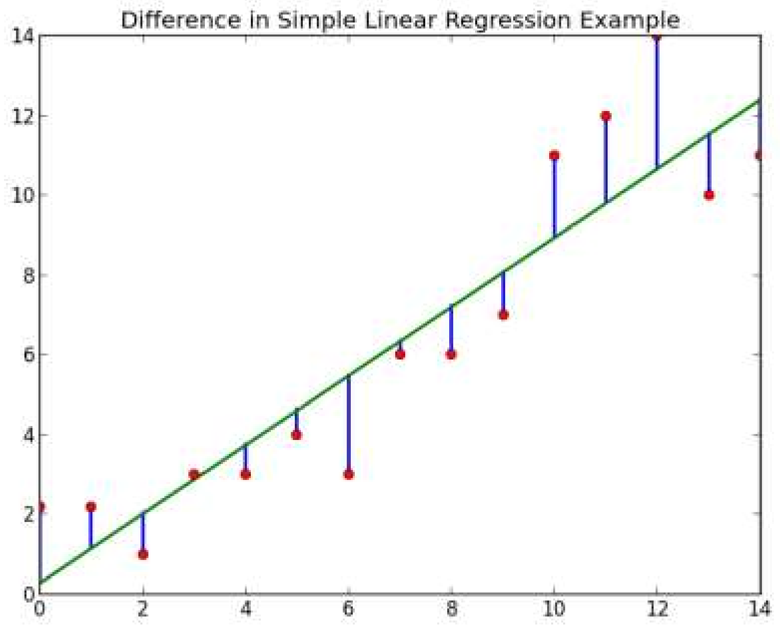
- Imamo sadež, ki ni podolgovat, ni sladek, je pa rumen. Kateri sadež je to?

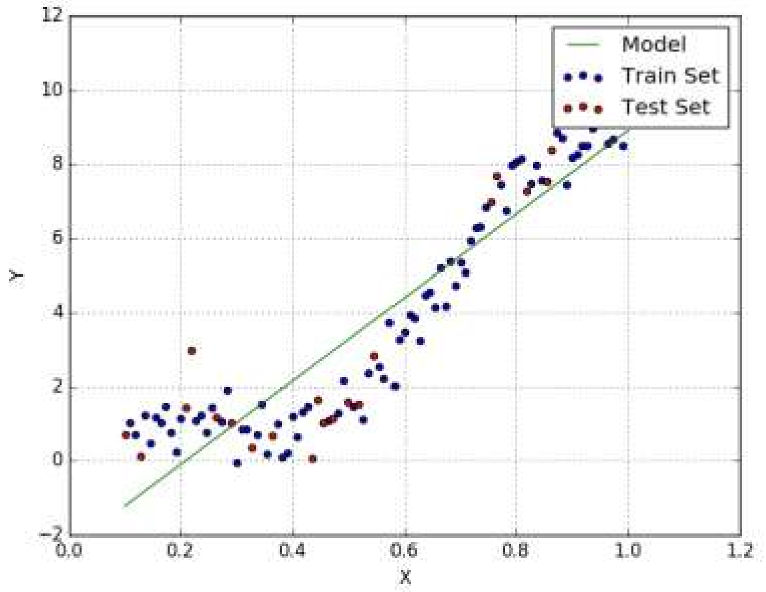
  - $P(banana|neP, neS, daR)$
    $= P(banana) \cdot P(banana|neP)/P(banana) \cdot P(banana|neS)/P(banana)$
    $\cdot P(banana|daR)/P(banana) =$
    $= \frac{500}{1000} \cdot \frac{100}{500} \cdot \frac{1000}{500} \cdot \frac{150}{350} \cdot \frac{1000}{500} \cdot \frac{450}{800} \cdot \frac{1000}{500} = 0{,}5 \cdot 0{,}2 \cdot 2 \cdot 0{,}429 \cdot 2 \cdot 0{,}563 \cdot 2 = 0{,}193$
  - $P(pomaranča|neP, neS, daR) = 0{,}3 \cdot 0{,}6 \cdot 3{,}33 \cdot 0{,}429 \cdot 3{,}33 \cdot 0{,}375 \cdot 3{,}33 = 1{,}069$ $\longleftarrow$ ta sadež je najverjetneje pomaranča
  - $P(drugo|neP, neS, daR) = 0{,}2 \cdot 0{,}2 \cdot 5 \cdot 0{,}143 \cdot 5 \cdot 0{,}063 \cdot 5 = 0{,}045$

Linearna regresija

- **Linearna funkcija:**
  atributi → ciljna regresijska spremenljivka
- želimo minimizirati napako na učni množici
- Najpogosteje: minimiziramo **kvadratno napako**
- Postopek:
  - Analitična rešitev

  ALI
  - Iterativni postopek minimizacije napake
- Lastnosti:
  1. **Preprosta**
  2. **Hitra** za izračun
  3. **Ni prevelikega prileganja**
  4. Lahko reši samo linearne probleme





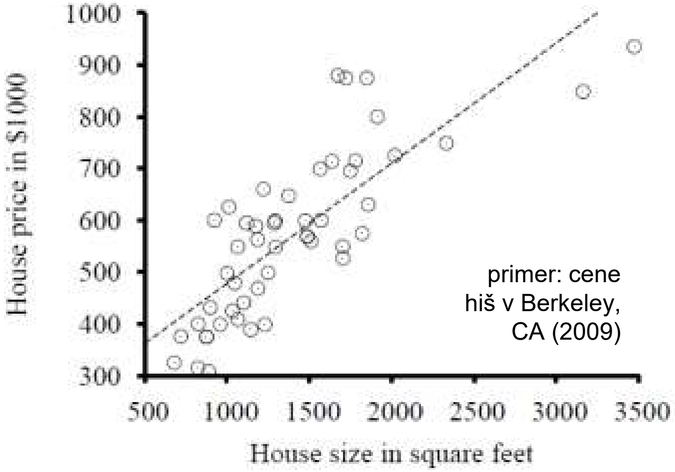
Linearni modeli

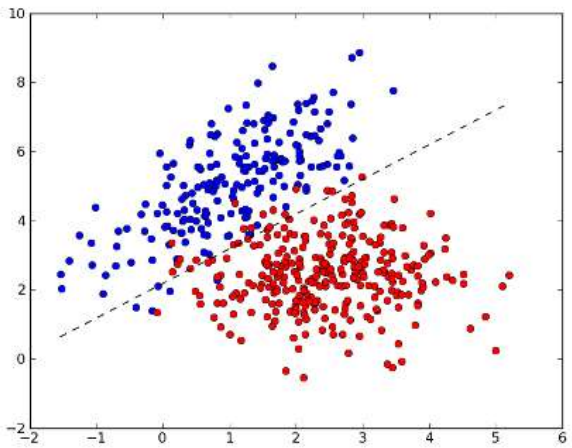
- uporaba pri klasifikaciji (kot separator razredov) in regresiji (kot prileganje skozi podane točke)
- linearni model z eno odvisno spremenljivko (angl. *univariate linear model*):

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

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

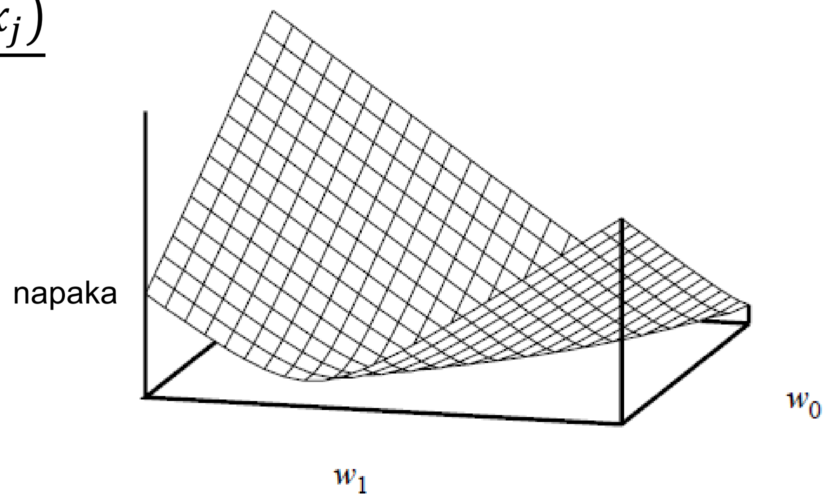
- optimizacijo izvedemo z minimizacijo srednje kvadratne napake:

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

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

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

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



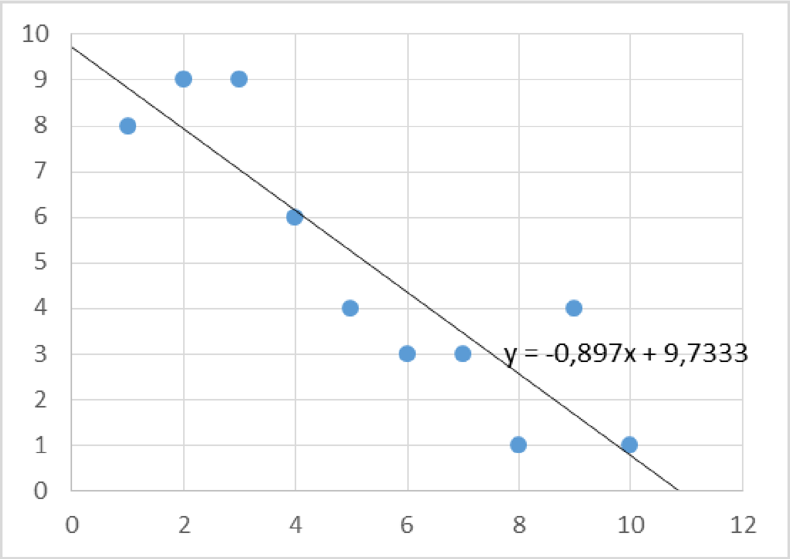
Linearna regresija

- 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
$\sum x_j = 55$	$\sum y_j = 48$	$\sum x_j y_j = 190$	$\sum x_j^2 = 385$

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

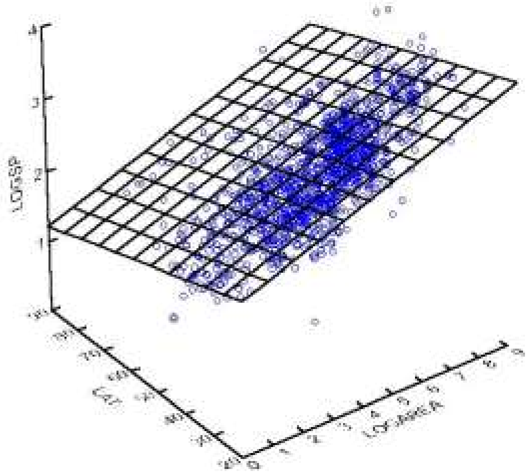
$$w_0 = \frac{\sum y_j - w_1(\sum x_j)}{N} = \frac{48 - (-0{,}897) \cdot 55}{10} = 9{,}733$$



Posplošitev v več dimenzij

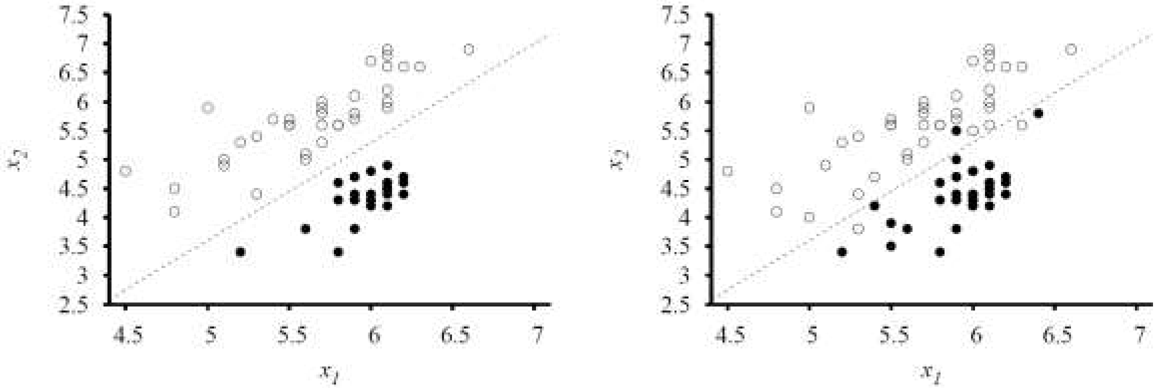
- možna je posplošitev v višje število dimenzij – več neodvisnih spremenljivk (atributov) (angl. *multivariate linear regression*)
  $h(x) = w_0 + \sum_i w_i x_{j,i}$
  kjer so w_i uteži (koeficienti), $x_{j,i}$ pa i-ta spremenljivka (atribut) primera x_j
- uteži lahko določimo analitično: $\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$
  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 $\boldsymbol{w}$:
    $w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} napaka(\boldsymbol{w})$

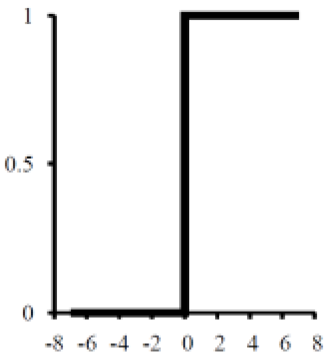


Linearni modeli pri klasifikaciji

- 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

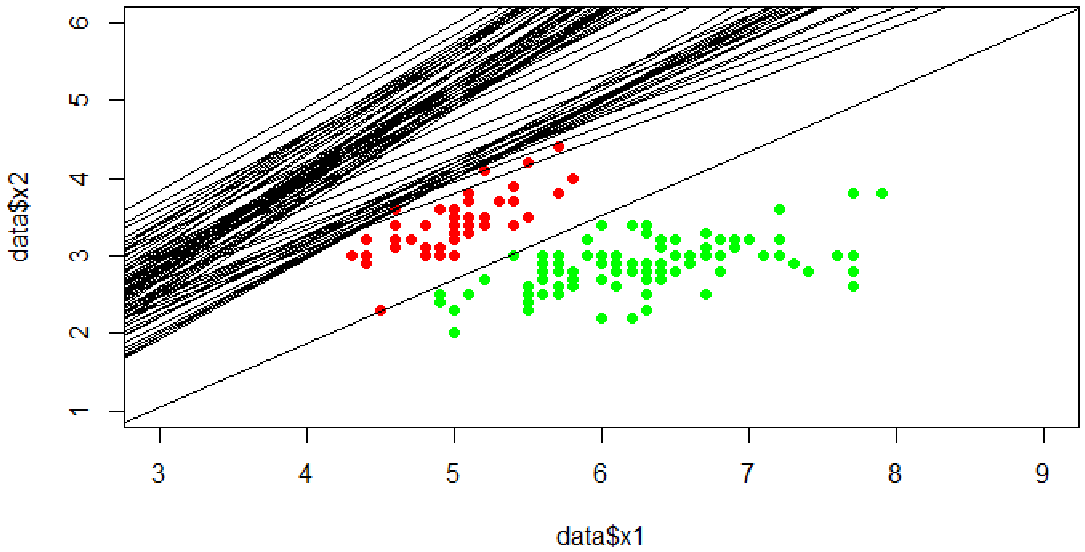


stopničasta pragovna funkcija

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



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