Učenje na primjerima

Poglavlje 18.2–3

Prema slajdovima Stuarta Russela, Krunoslava Puljića, Tomislava Šmuca, Hantao Zhanga (hvala)!

Strojno učenje pod nadzorom, klasifikacija

- Učenje – iz prošlih iskustava, opažanja, primjera.
- Računalo nema "prošlih iskustava".
- Računalo uči iz podataka, koji predstavljaju neko "prošlo iskustvo".
- Cilj: napraviti funkciju cilja (hipotezu h) koja će se moći upotrijebiti za predviđanje vrijednosti diskretnog atributa — tzv. klase,
  (generiranog nepoznatom funkcijom f s diskretnom kodomenom)
  na primjer: odobrava / ne odobrava, visoki rizik / niski rizik,
- Taj zadatak uobičajeno se naziva:
  učenje pod nadzorom ili induktivno učenje,
  a pripada kategoriji klasifikacije (diskretna kodomena).

Podaci i cilj

- Podaci "za učenje":
  Skup podataka (tzv. primjeri, instance ili slučajevi) opisani
  - s k atributa: $A_1, A_2, \ldots, A_k$ (domena)
  - klasa: svaki primjer označen je unaprijed definiranom klasom (vrijednost funkcije).

- Cilj: naučiti model klasifikacije iz danih podataka, tako da se može koristiti za predviđanje klasa novih (budućih ili test) slučajeva.

Primjer podataka – traženje kredita

Identifikacija	Dob	Ima_posao	Ima_kuću	Kreditni_status	Klasa - kredit odobren ili ne
1	mlad	ne	ne	lošs	ne
2	mlad	ne	ne	dobar	ne
3	mlad	da	ne	dobar	da
4	mlad	da	da	lošs	da
5	mlad	ne	ne	lošs	ne
6	srednje g.	ne	ne	lošs	ne
7	srednje g.	ne	ne	dobar	ne
8	srednje g.	da	da	dobar	da
9	srednje g.	ne	da	izvrstan	da
10	srednje g.	ne	da	izvrstan	da
11	stariji	ne	da	izvrstan	da
12	stariji	ne	da	dobar	da
13	stariji	da	ne	dobar	da
14	stariji	da	ne	izvrstan	da
15	stariji	ne	ne	lošs	ne

Napomena: identifikacija nije atribut, služi samo za lakše snalaženje.

Primjer učenja

- nauči model klasifikacije iz podataka
- iskoristi model za klasifikaciju budućih zahtjeva za kreditom u
  – da (odobren kredit) i
  – ne (kredit nije odobren).
- Koja je klasa za sljedeći slučaj / instancu?

Dob	Ima_posao	Ima_kuću	Kreditni_status	Klasa - kredit odobren ili ne
mlad	ne	ne	dobar	?

Učenje s nadzorom vs. učenje bez nadzora

- Učenje s nadzorom:
  klasifikacija se tretira kao učenje pod nadzorom iz danih primjera.
  – Nadzor: Podaci (opažanja, mjerenja, itd.) označavaju se predefiniranim klasama — kao da ih učitelj zadaje.
  – Test podaci se, također, klasificiraju u te klase.

- Učenje bez nadzora (klasteriranje):
  – Oznake klasa nisu unaprijed poznate.
  – Za dani skup podataka, zadatak je pokazati da postoje klase ili klasteri među podacima — i treba ih pronaći.

Učenje s nadzorom – dvofazni proces

- Učenje ili treniranje: učenje modela iz podataka za treniranje
- Testiranje: testiranje modela korištenjem dotad nepoznatog skupa podataka za provjeru točnosti modela

$$\text{Točnost} = \frac{\text{broj korektnih klasifikacija}}{\text{broj testiranih slučajeva}}$$

Što je strojno učenje?

Definicija strojnog učenja (Tom M. Mitchell):

Kažemo da računalo ili program uči
iz iskustva (zadanih podataka) E (experience),
za obavljanje nekog zadatka T (task),
uz mjeru uspješnosti P (performance measure),
ako se
uspješnost u obavljanju T, prema mjeri P, povećava s iskustvom E.

Dakle, ključ = poboljšanje performanse kroz iskustvo.

Primjer – odobravanje kredita

- Podaci: tablica molitelja kredita.
- Zadatak: predvidjeti hoće li kredit biti odobren ili ne.
- Mjera uspješnosti: točnost.
- Bez učenja:
  klasificiraj sve buduće zahtjeve (test podatke) u većinsku klasu,
  tj. u klasu "da":

$$\text{točnost} = \frac{9}{15} = 60\%.$$

- S učenjem se mogu postići bolji rezultati no što je to 60%.

Fundamentalna pretpostavka učenja

Pretpostavka:

Distribucija primjera za treniranje identična je distribuciji test primjera, uključivo i budućih, još nepoznatih podataka.

U praksi, ta je pretpostavka, često, do neke mjere narušena.

Ozbiljna narušenost — rezultira slabom točnošću klasifikacije.

Drugim riječima:

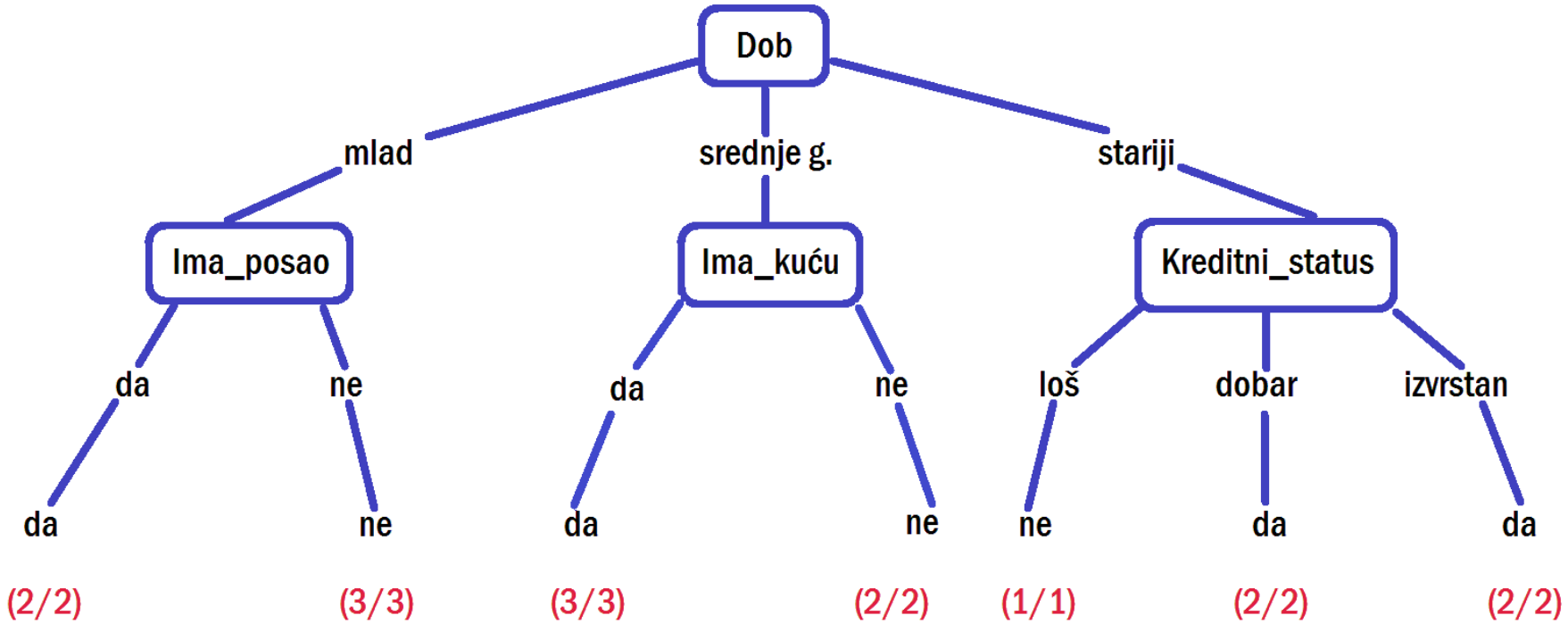
Da bi se postigla dobra točnost klasifikacije na test podacima, primjeri za treniranje moraju biti dovoljno reprezentativni za test podatke.

Učenje korištenjem stabala odlučivanja — uvod

- Učenje korištenjem stabala odlučivanja je jedna od najčešćih tehnika klasifikaciju.
  - Točnost klasifikacije usporediva je s ostalim metodama.
  - Metoda je vrlo efikasna.
  - Model klasifikacije je stablo, zvano stablo odlučivanja — koje se sastoji samo od točaka/čvorova za odluku, nema slučajnosti.
- Primjer: reprodukcija podataka iz tablice zahtjeva za kreditima.

Stablo odlučivanja za zahtjeve za kreditima

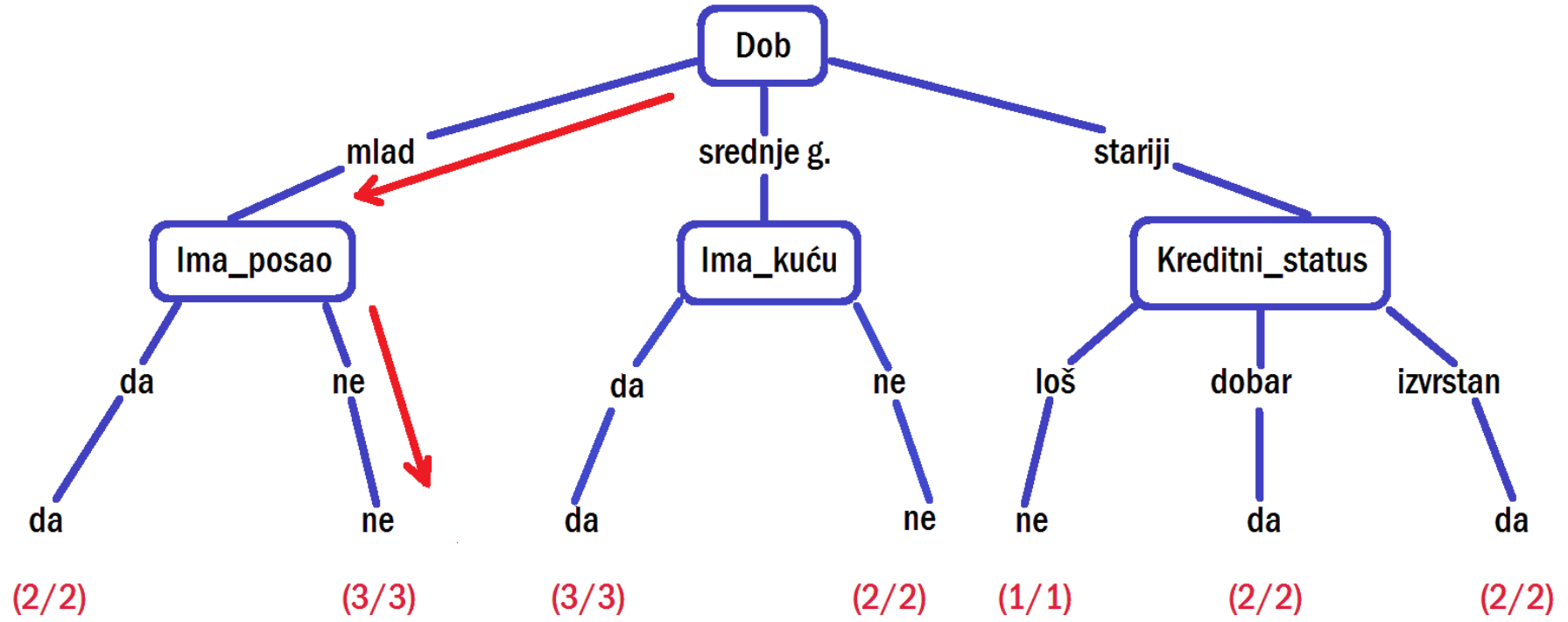
Čvorovi odlučivanja i listovi (klase)



Brojevi na dnu = ispravno klasificirani / svi u toj kategoriji (grani).

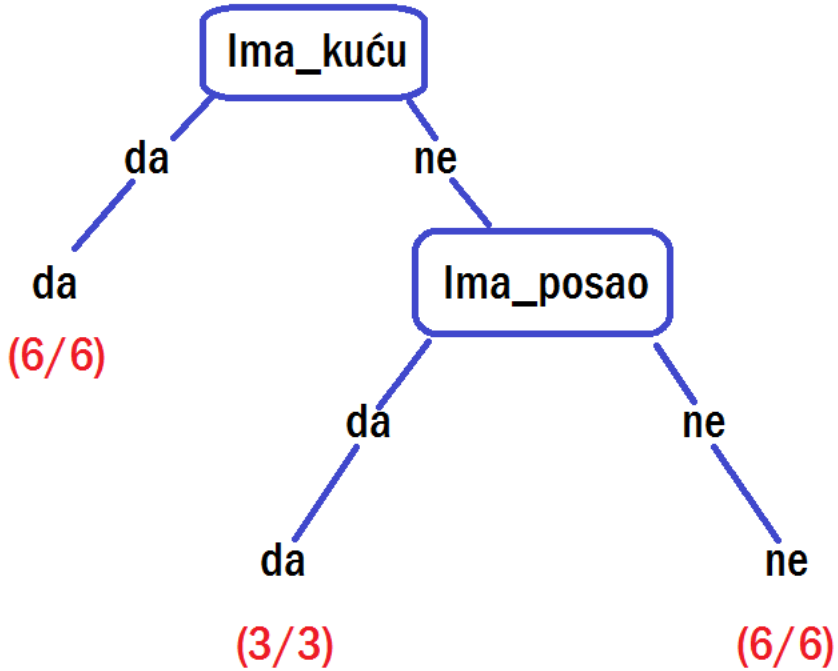
Stablo odlučivanja — primjena na test–primjer

Dob	Ima_posao	Ima_kuću	Kreditni_status	Klasa - kredit odobren ili ne
mlad	ne	ne	dobar	NE



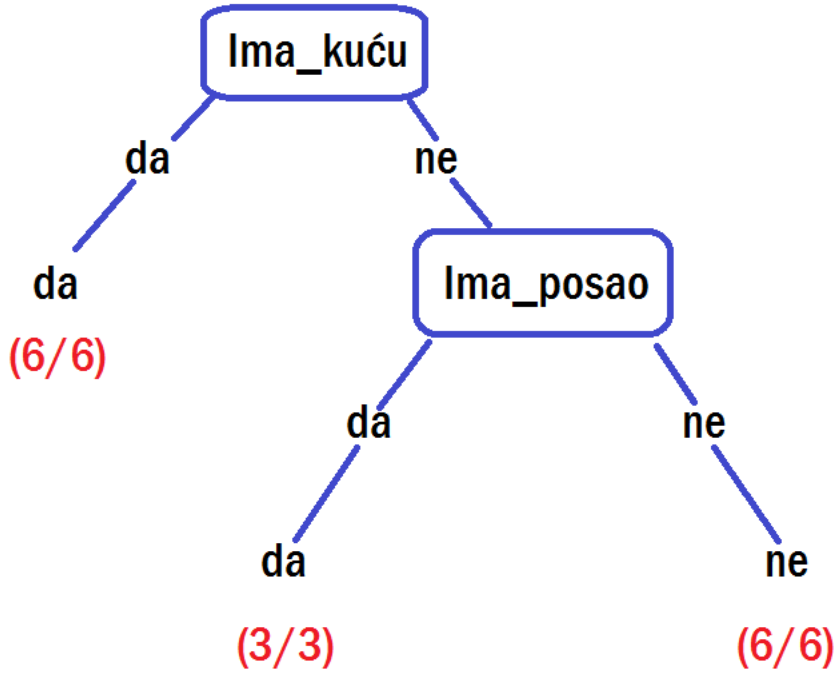
Je li stablo odlučivanja jedinstveno?

- Ne. Postoji i jednostavnije stablo (v. sliku).
- Želimo manje i točnije stablo.
  – Lako za razumijevanje i ima bolje performanse.
- Nalaženje najboljeg stabla je NP-teško
- Svi današnji algoritmi konstrukcije stabala su heuristički.



Od stabla odlučivanja do skupa pravila

- Stablo odlučivanja može se prevesti u skup pravila.
- Svaki put od korijena do lista je pravilo.



$$\text{Ima\_kuću} = \text{da} \rightarrow \text{Klasa} = \text{da} \quad 6/15 \text{ sluč., potvr. } 6/6$$
$$\text{Ima\_kuću} = \text{ne}, \text{Ima\_posao} = \text{da} \rightarrow \text{Klasa} = \text{da} \quad 3/15 \text{ sluč., potvr. } 3/3$$
$$\text{Ima\_kuću} = \text{ne}, \text{Ima\_posao} = \text{ne} \rightarrow \text{Klasa} = \text{ne} \quad 6/15 \text{ sluč., potvr. } 6/6$$

Algoritam učenja stabala odlučivanja

Osnovni algoritam = pohlepni algoritam podijeli-pa-vladaj:

- Pretpostavi da su svi atributi diskretni (može se raditi i s neprekidnim atributima).
- Stablo se konstruira od vrha prema dnu u rekurzivnom postupku.
- Na početku, svi primjeri za treniranje su u korijenu.
- Primjeri se particioniraju rekurzivno, bazirano na odabranim atributima
- Atributi se selektiraju na osnovu funkcije "nečistoće" (na primjer, dobitka na informaciji)

Uvjeti za kraj particioniranja:

- Svi primjeri za dani čvor pripadaju istoj klasi
- Nema više atributa za daljnje particioniranje – većinska klasa je list
- Nema više preostalih primjera

Algoritam učenja stabala odlučivanja

```
Algorithm decisionTree(D, A, T)
1   if D contains only training examples of the same class c_j ∈ C then
2      make T a leaf node labeled with class c_j;
3   elseif A = ∅ then
4      make T a leaf node labeled with c_j, which is the most frequent class in D
5   else  // D contains examples belonging to a mixture of classes. We select a single
6         // attribute to partition D into subsets so that each subset is purer
7      p_0 = impurityEval-1(D);
8      for each attribute A_i ∈ {A_1, A_2, ..., A_k} do
9         p_i = impurityEval-2(A_i, D)
10     end
11     Select A_g ∈ {A_1, A_2, ..., A_k} that gives the biggest impurity reduction,
          computed using p_0 − p_i;
12     if p_0 − p_g < threshold then    // A_g does not significantly reduce impurity p_0
13        make T a leaf node labeled with c_j, the most frequent class in D.
14     else                             // A_g is able to reduce impurity p_0
15        Make T a decision node on A_g;
16        Let the possible values of A_g be v_1, v_2, ..., v_m. Partition D into m
             disjoint subsets D_1, D_2, ..., D_m based on the m values of A_g.
17        for each D_j in {D_1, D_2, ..., D_m} do
18           if D_j ≠ ∅ then
19              create a branch (edge) node T_j for v_j as a child node of T;
20              decisionTree(D_j, A-{A_g}, T_j)// A_g is removed
21           end
22        end
23     end
24  end
```

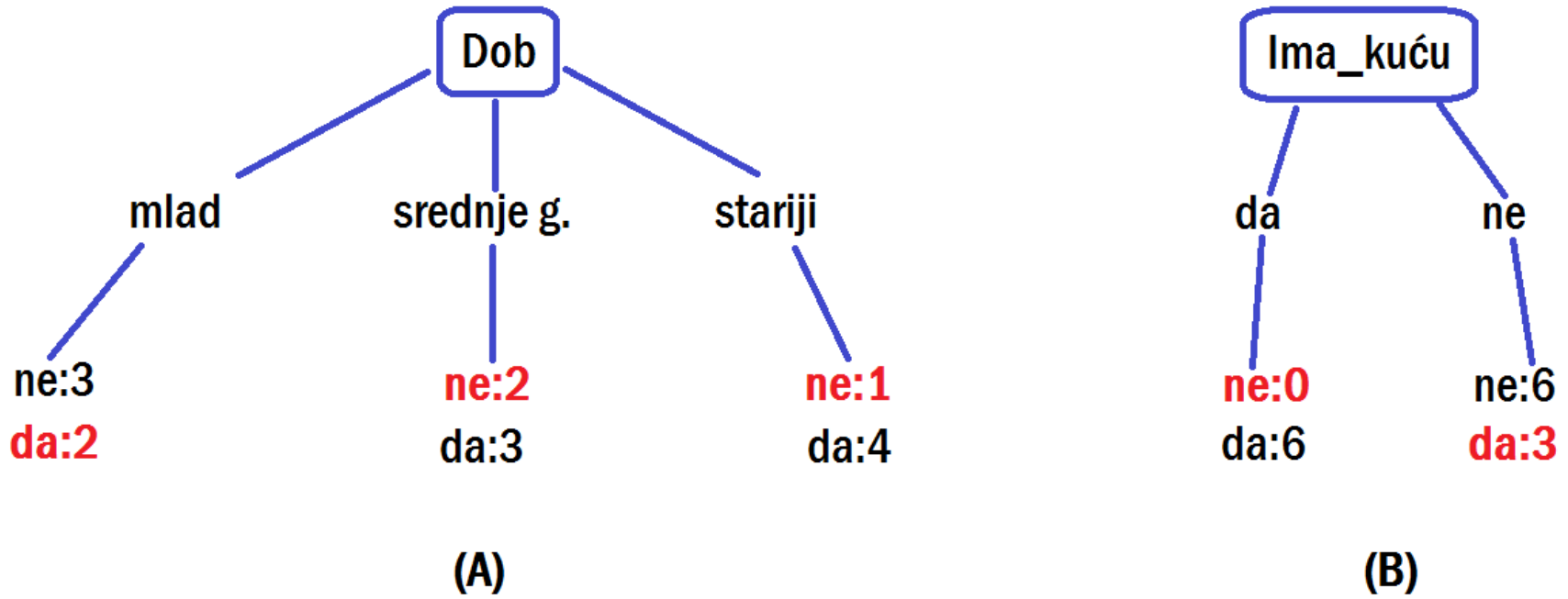
Izbor atributa za particioniranje podataka

- Ključ konstrukcije stabla odlučivanja — koji atribut treba izabrati za grananje.
- Cilj je reducirati, što je više moguće, nečistoću ili nesigurnost u podacima.
- Podskup podataka je čist ako sve instance pripadaju istoj klasi.
- Heuristika u algoritmu je kako izabrati atribut koji daje maksimalni dobitak na informacijama (engl. Information Gain ili Gain Ratio – termin se koristi u teoriji informacija).

Reprodukcija odluka iz primjera traženja kredita

Identifikacija	Dob	Ima_posao	Ima_kuću	Kreditni_status	Klasa - kredit odobren ili ne
1	mlad	ne	ne	loš	ne
2	mlad	ne	ne	dobar	ne
3	mlad	da	ne	dobar	da
4	mlad	da	da	loš	da
5	mlad	ne	ne	loš	ne
6	srednje g.	ne	ne	loš	ne
7	srednje g.	ne	ne	dobar	ne
8	srednje g.	da	da	dobar	da
9	srednje g.	ne	da	izvrstan	da
10	srednje g.	ne	da	izvrstan	da
11	stariji	ne	da	izvrstan	da
12	stariji	ne	da	dobar	da
13	stariji	da	ne	dobar	da
14	stariji	da	ne	izvrstan	da
15	stariji	ne	ne	loš	ne

Dva moguća korijena – koji je bolji?



Crveno je broj "manjinskih" primjera (pogrešna klasifikacija).

- Čini se da je (B) bolji.

Teorija informacija

- Teorija informacija daje matematičku osnovu za mjerenje sadržaja informacija.
- Da biste razumjeli pojam informacije, razmislite kako pružiti odgovor na pitanje, na primjer, hoće li pasti glava na novčiću.
  - Ako netko već dobro nagađa odgovor, onda je stvarni odgovor manje informativan.
  - Ako netko već zna da je novčić "obrađen" tako da će pasti glava s vjerojatnošću 0.99,
  - tada poruka o stvarnom ishodu bacanja vrijedi manje nego što bi vrijedila za simetrični novčić $(50 : 50)$.

Teorija informacija (nastavak)

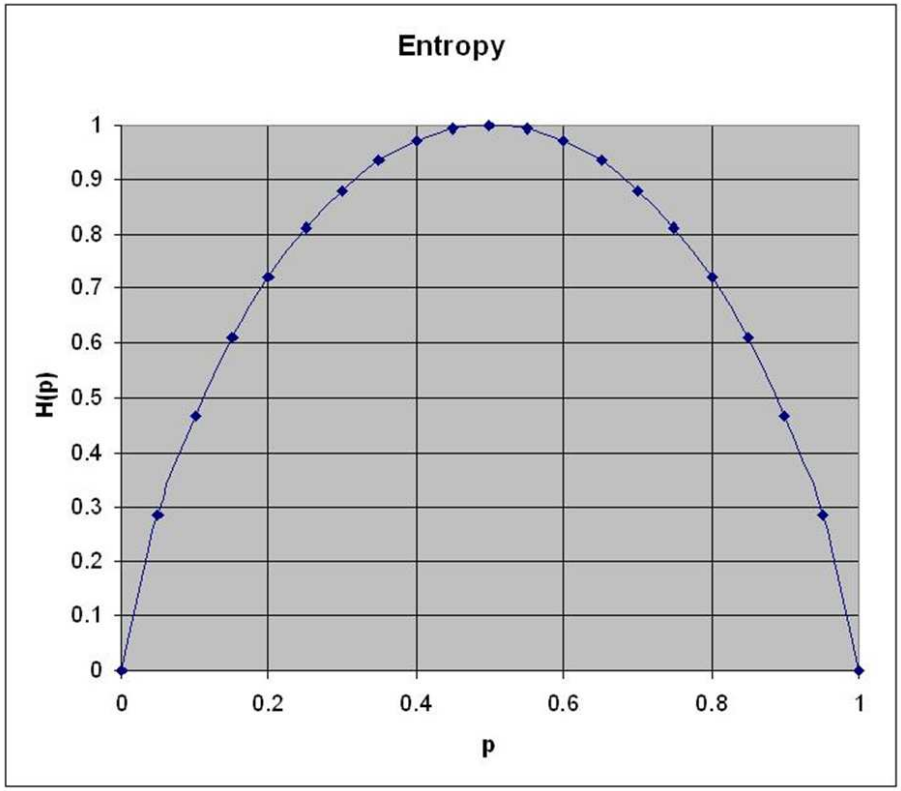
- Za simetrični (pošteni) novčić, nemate dovoljno informacije, pa ste spremni platiti više (recimo u $) za dodatnu informaciju – što manje znate, to je informacija vrednija.

- Teorija informacije koristi istu intuiciju, ali
  umjesto mjerenja vrijednosti informacije u dolarima,
  mjeri vrijednost sadržaja informacija u bitovima.

- Jedan bit informacije dovoljan je za odgovor na pitanje da / ne na pitanje o kojem osoba nema nikakvu ideju,
  – recimo, na koju će stranu pasti simetrični novčić.

Shannonova entropija

Booleova varijabla, vjerojatnost za $true$ je p.

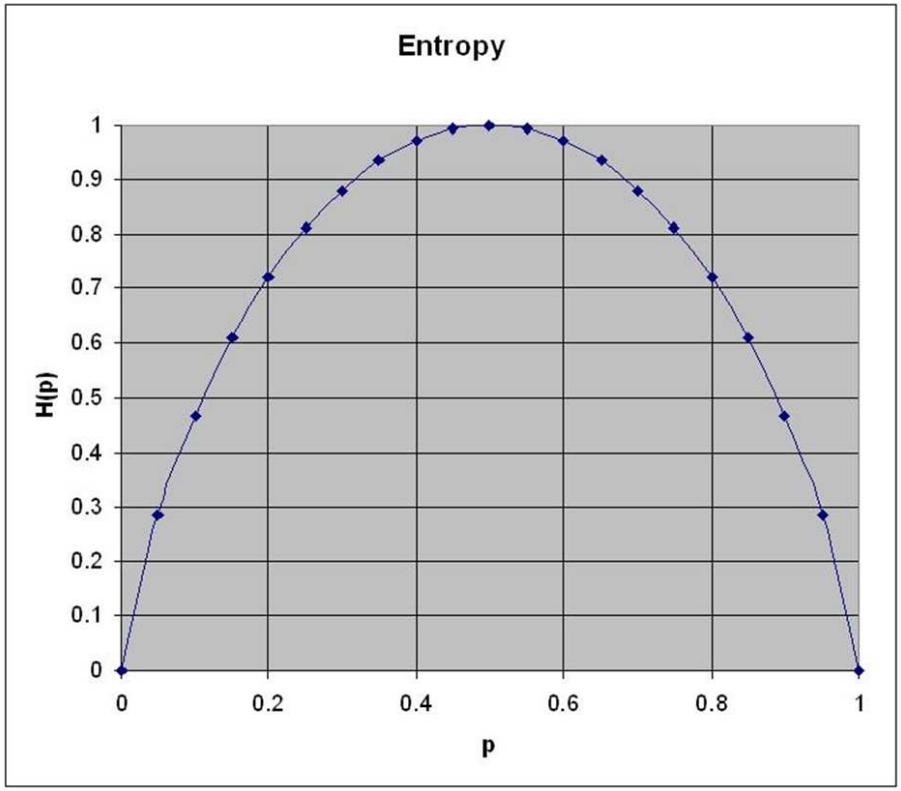


- $H(0.5) = 1$
- $H(0) = 0$
- $H(1) = 0$
- $H(0.4) = 0.97$
- $H(0.33) = 0.9$
- $H(0.2) = 0.71$
- $H(p) = ?$

Uočiti simetriju oko $1/2$, analogno za $true$ i $false$.

Shannonova entropija

Booleova varijabla, vjerojatnost za $true$ je p.



- $H(0.5) = 1$
- $H(0) = 0$
- $H(1) = 0$
- $H(0.4) = 0.97$
- $H(0.33) = 0.9$
- $H(0.2) = 0.71$

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p).$$

Mjera za entropiju

Formula za entropiju

$$\text{entropija}(D) = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j), \quad \sum_{j=1}^{|C|} \Pr(c_j) = 1,$$

gdje je $\Pr(c_j)$ je vjerojatnost klase c_j u skupu podataka D.

Sasvim analogno, za slučajnu varijablu V, s vrijednostima v_k

$$\text{entropija} \quad H(V) = -\sum_k P(v_k) \log_2 P(v_k), \quad \sum_k P(v_k) = 1,$$

gdje je $P(v_k)$ vjerojatnost ishoda v_k.

Entropiju koristimo kao mjeru nečistoće ili nereda u skupu podataka D (ili kao mjeru informacije u stablu).

Uobičajeno, količina informacija je $1 -$ entropija.

Mjera entropije — primjeri

1. Skup podataka D ima 50% pozitivnih ($\Pr(\text{poz}) = 0.5$) i 50% negativnih primjera ($\Pr(\text{neg}) = 0.5$).

$$\text{entropija}(D) = -0.5 \cdot \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

2. Skup podataka D ima 20% pozitivnih ($\Pr(\text{poz}) = 0.2$) i 80% negativnih primjera ($\Pr(\text{neg}) = 0.8$).

$$\text{entropija}(D) = -0.2 \cdot \log_2 0.2 - 0.8 \log_2 0.8 = 0.722$$

3. Skup podataka D ima 100% pozitivnih ($\Pr(\text{poz}) = 1$) i 0% negativnih primjera ($\Pr(\text{neg}) = 0$).

$$\text{entropija}(D) = -1 \cdot \log_2 1 - 0 \log_2 0 = 0$$

- Što podaci postaju čišći, to vrijednost entropije postaje sve manja i manja, što je vrlo korisno.

Dobitak informacija

- Dan je skup primjera D. Prvo izračunamo njegovu entropiju:

$$\text{entropija}(D) = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j).$$

- Ako atribut A_i, s v vrijednosti, uzmemo kao korijen trenutnog stabla, to particionira skup D u v podskupova $D_1, D_2, \ldots D_v$, prema vrijednosti atributa A_i.

Očekivana entropija, ako je A_i uzet kao trenutni korijen, je:

$$\text{entropija}_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{entropija}(D_j).$$

To je prosječna entropija preko svih skupova D_j.

Dobitak informacija (nastavak)

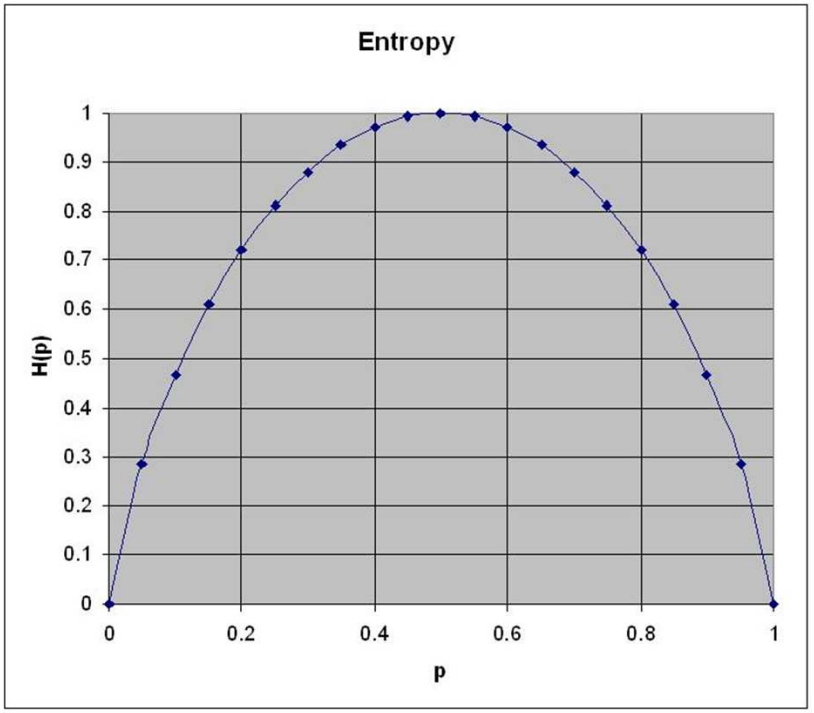
Dobitak informacije (engl. information gain) dobiven odabirom baš tog atributa A_i za grananje ili particiju podataka je smanjenje entropije

$$\text{dobitak}(D, A_i) = \text{entropija}(D) - \text{entropija}_{A_i}(D)$$

- Za grananje/diobu trenutnog stabla, među preostalim atributima, biramo onaj atribut koji tog trena daje najveći dobitak informacije. (To je onaj s najmanjom entropijom = najviše informacija.)

Napomena: ponekad se kao heuristika za izbor koristi i pojam omjer dobitka (engl. gain ratio), no značenje je isto — uzmi A_i s najmanjom entropijom.

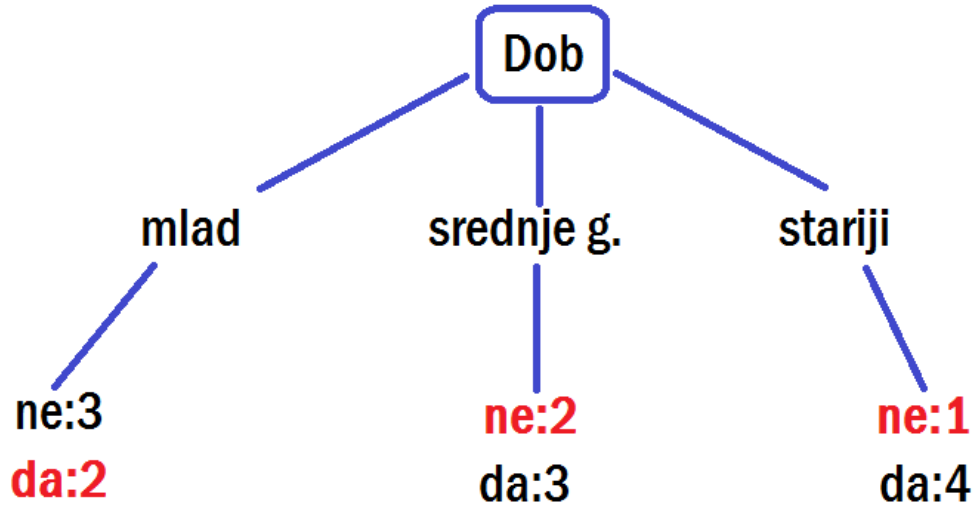
Primjer — traženje kredita



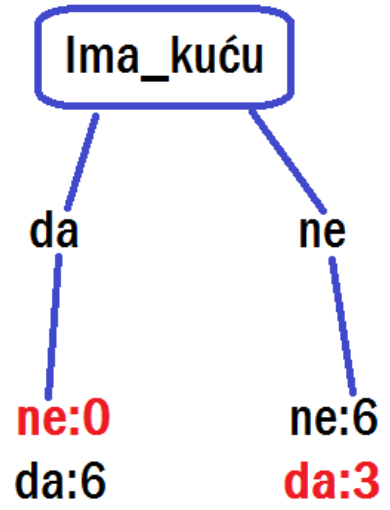
Entropija skupa D — imamo 6 primjera u klasi Ne i 9 u klasi Da:

$$\text{entropija}(D) = -\frac{6}{15} \cdot \log_2 \frac{6}{15} - \frac{9}{15} \cdot \log_2 \frac{9}{15} = 0.971$$

Primjer — traženje kredita



(A)



(B)

$$\text{entropija}_{\text{Ima\_kuću}}(D) = -\frac{6}{15} \cdot \text{entropija}(D_1) - \frac{9}{15} \cdot \text{entropija}(D_2)$$

$$= \frac{6}{15} \cdot 0 + \frac{9}{15} \cdot 0.918 = 0.551$$

Primjer — traženje kredita

Dob	Da	Ne	Entropija (Di)
mlad	2	3	0.971
srednje g.	3	2	0.971
stariji	4	1	0.722

Imamo po 5 primjera svake vrijednosti.

$$\text{entropija}_{\text{Dob}}(D) = -\frac{5}{15} \cdot \text{entropija}(D_1) - \frac{5}{15} \cdot \text{entropija}(D_2) - \frac{5}{15} \cdot \text{entropija}(D_3)$$

$$= \frac{5}{15} \cdot 0.971 + \frac{5}{15} \cdot 0.971 + \frac{5}{15} \cdot 0.722 = 0.888$$

Primjer — traženje kredita

Kad izračunamo dobitak informacije za svaki atribut, izlazi

$$\begin{aligned}
\text{dobitak}(D, \text{Dob}) &= 0.971 - 0.888 = 0.083 \\
\text{dobitak}(D, \text{Ima_kuću}) &= 0.971 - 0.551 = 0.420 \\
\text{dobitak}(D, \text{Ima_posao}) &= 0.971 - 0.647 = 0.324 \\
\text{dobitak}(D, \text{Kreditni_status}) &= 0.971 - 0.608 = 0.363
\end{aligned}$$

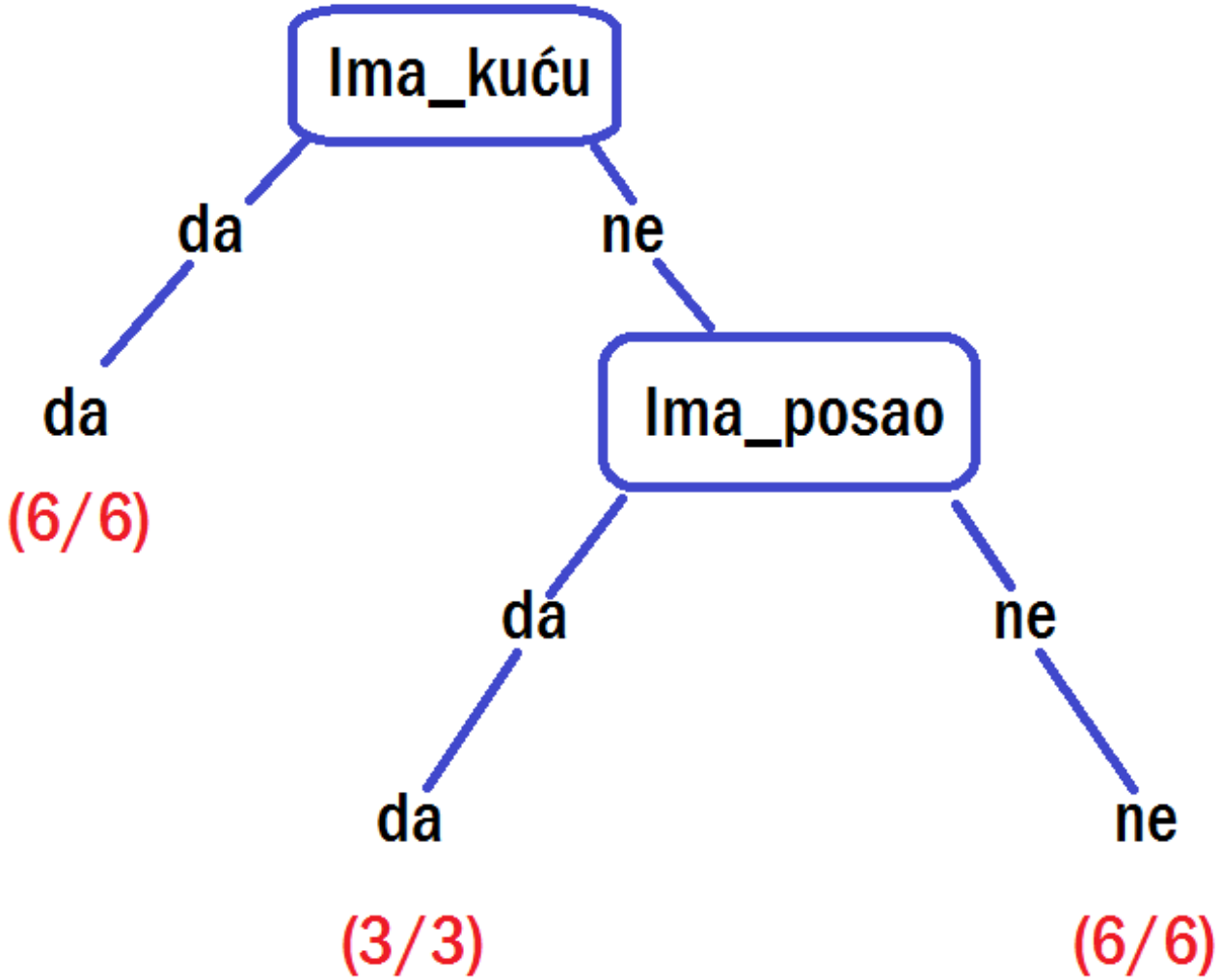
Zaključak: Najbolji izbor korijena je = Ima_kuću.

Vrijednost "da" potpuno klasificira pripadnih 6 primjera u klasu Da. Dobivamo list stabla!

Vrijednost "ne" ima 9 primjera — 6 u klasi Ne i 3 u klasi Da.

Nastavljamo konstrukciju stabla u tom čvoru.

Konačno stablo odlučivanja



Kako se posluje s neprekidnim atributima

Rješenje = diskretizacija.

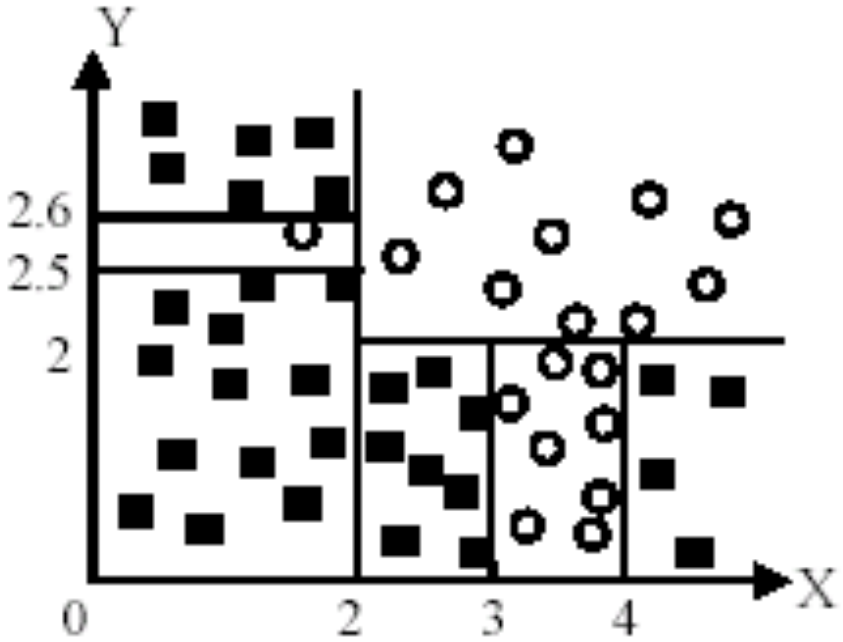
Neprekidni atribut se u svakom čvoru rastavi na dva intervala (može i više).

Kako se nalazi najbolji prag za ovu podjelu?

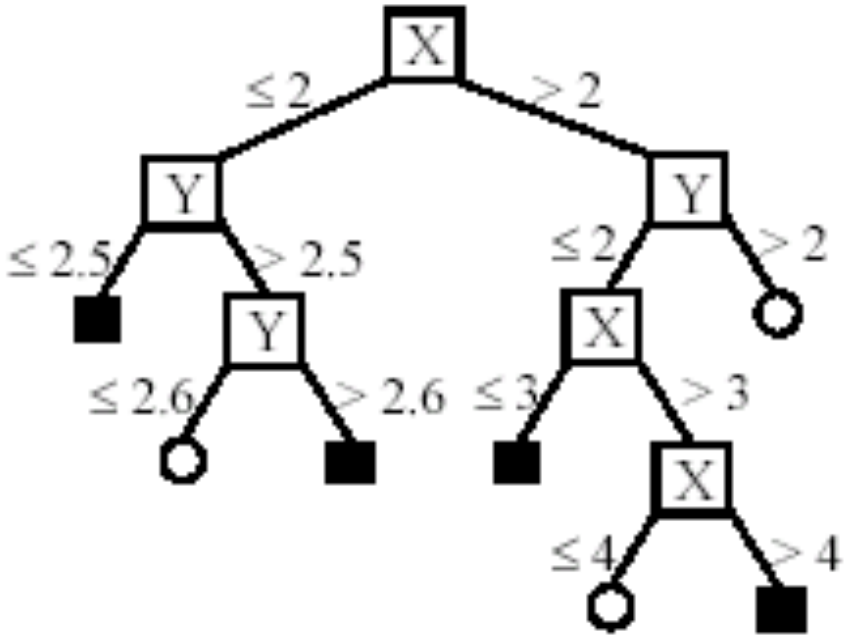
- Ponovno se koristi dobitak ili omjer dobitaka.
- Sortiraju se sve vrijednosti tog atributa u rastućem poretku $\{v_1, v_2, \ldots, v_r\}$
- Jedan od mogućih pragova za dvije susjedne vrijednosti v_i i v_{i+1}: isprobaj sve moguće pragove i nađi onaj koji maksimizira dobitak ili omjer dobitaka (ovo može biti skupo).

Alternativa: podjela ide samo između vrijednosti v_i i v_{i+1} kod kojih dolazi do promjene klase (ako takvih promjena nema previše).

Primjer za neprekidni prostor



(A) A partition of the data space



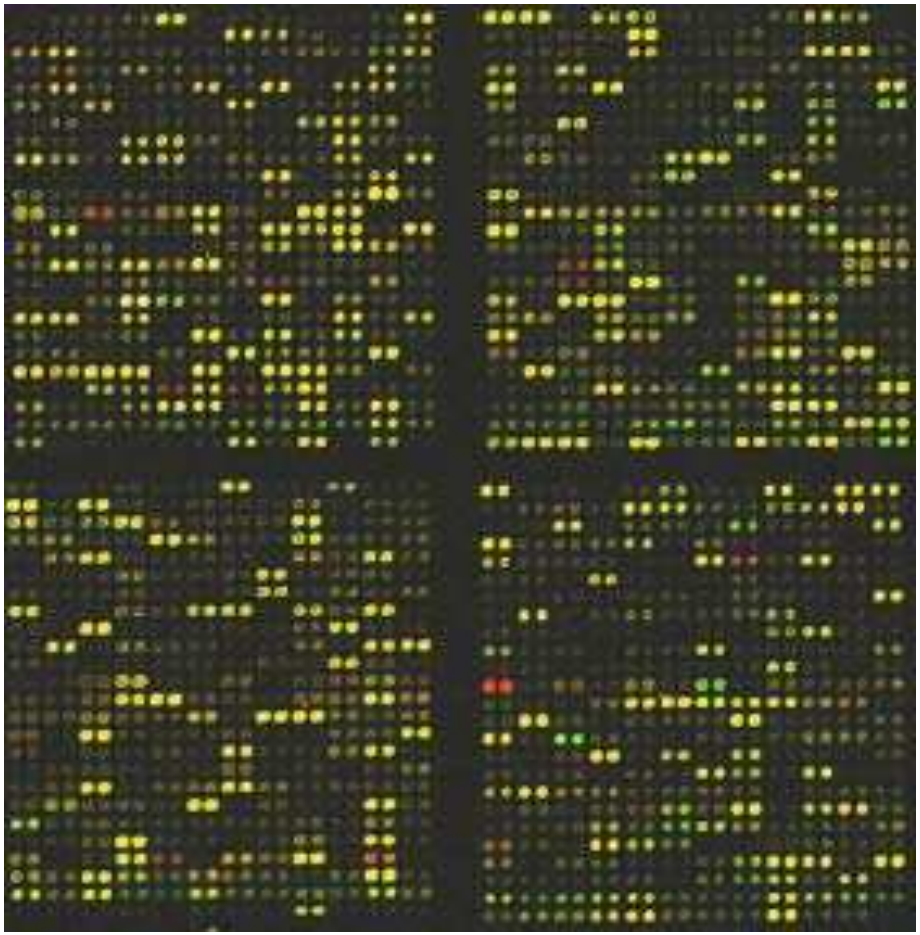
(B). The decision tree

Primjeri iz stvarnog života

- Biologija sustava – genska ekspresija mikropolja podataka
- Kategorizacija teksta: detekcija tzv. spama
- Otkrivanje kupaca
- Prepoznavanje lica
- Prepoznavanje potpisa
- Medicina: Predviđanje bolesti premalog protoka krvi kroz srce spektralnom analizom EKG-a.

Mikropolja podataka

Separacija malignog tkiva od zdravog tkiva na bazi mRNA ekspresije profila tkiva.



Otkrivanje kupaca

Predvidi hoće li kupac vjerojatno kupiti odgovarajući proizvod prema bazi podataka o profilima kupaca i njihovoj povijesti kupovanja.

Prepoznavanje lica

Razlikovanje ljudskih lica od ne-lica.

Prepoznavanje potpisa

Prepoznavanje potpisa po strukturnim sličnostima koje je teško kvantificirati.

Pripada li potpis nekoj osobi ili ne, na primjer Tonyju Blairu.

Prepoznavanje znakova (u više kategorija)

Identifikacija rukom pisanih znakova: klasificiraj svaku sliku znaka u jednu od 10 kategorija '0', '1', '2', ...

Raspoznavanje rukom pisanog teksta — znamenke

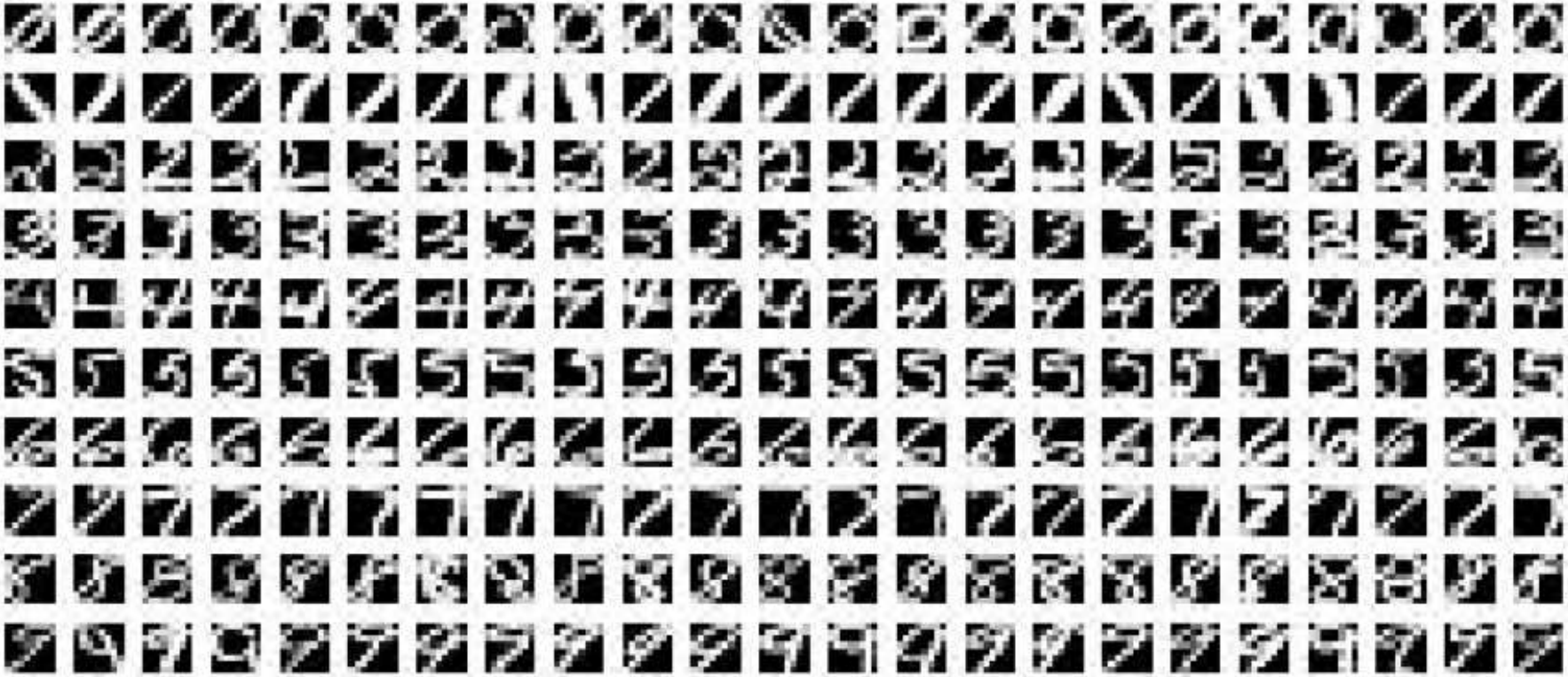
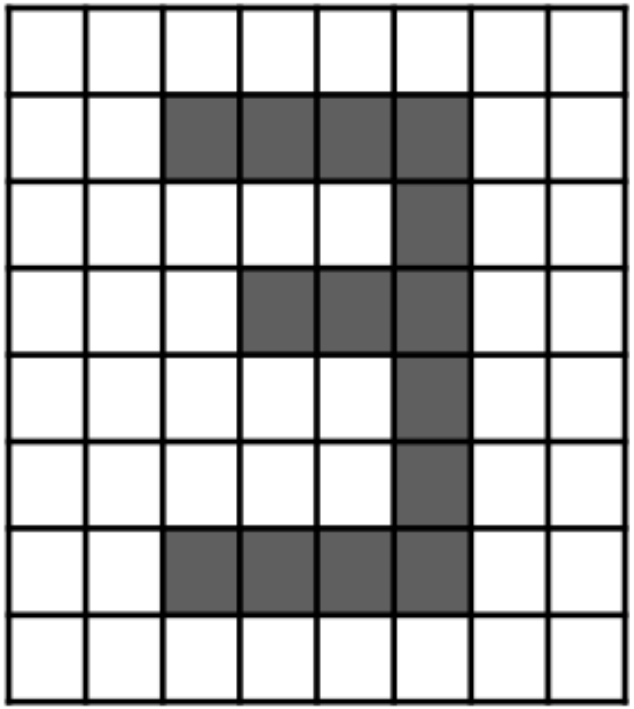


Fig. 3. Images of handwritten digits, normalized for horizontal and vertical scale and translation and sampled on an 8×8 pixel grid. Different writing angles introduce different levels of shearing in each image.

Raspoznavanje rukom pisanih znakova

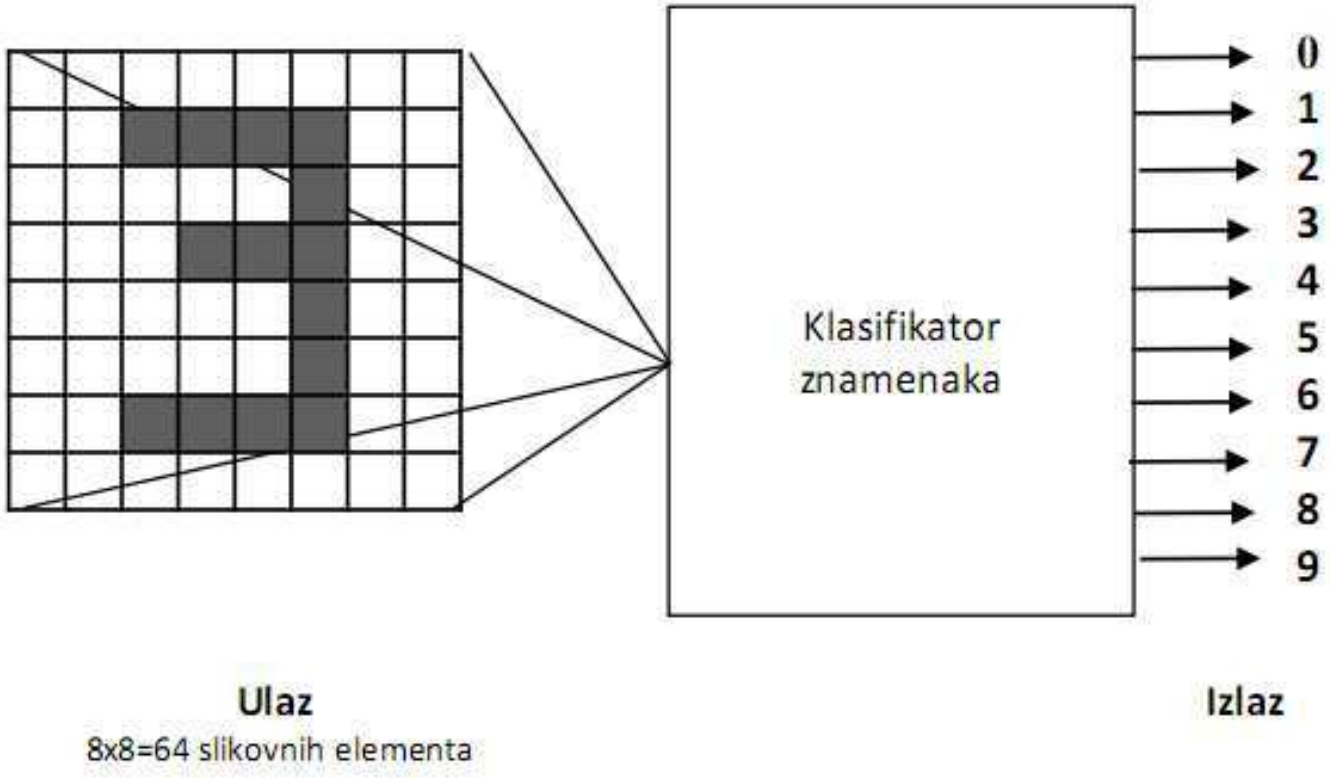
Primjer: klasifikator znamenaka

Znak "3" kako je tiskan u knjizi i njegova digitalizirana slika
u razlučivosti 8×8 slikovnih elemenata (piksela):



Klasifikator znamenaka

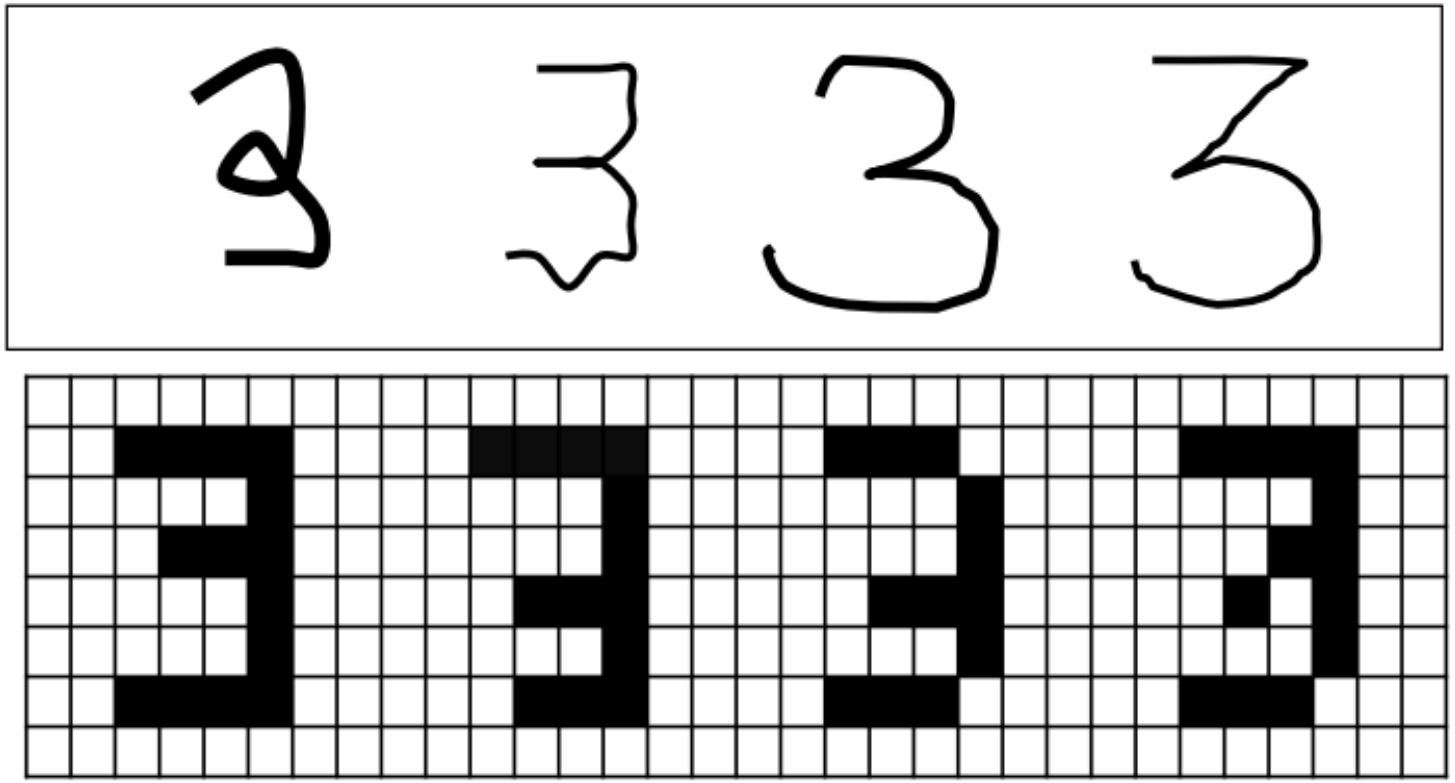
Treba odlučivati kojoj od unaprijed definiranih klasa pripada predočeni ulazni uzorak = klasifikacija uzoraka.



Klasifikator znamenaka

Zamislimo sada da želimo načiniti sustav koji će
raspoznavati rukom pisane znakove.

Problem: postoji nebrojeno varijacija između uzoraka za isti znak:



Klasifikator znamenaka

- Modeliranje sustava
  - pohrana mogućih uzoraka u memoriji
    - 64 ulaza za jedan uzorak
    - pohranjivanje $2^{64} \approx 1.8 \cdot 10^{19}$ različitih uzoraka u memoriji
    - uspoređivanje ulaznog uzorka s onim u memoriji
    - kad bi uspoređivanje uzorka na ulazu s jednim uzorkom u memoriji trajalo $0.1\ \mu\mathrm{s} = 10^{-7}\ \mathrm{s}$, za uspoređivanje jednog uzorka sa svima u memoriji bilo bi potrebno oko

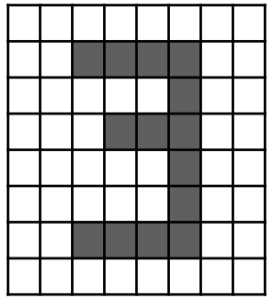
$$1.8 \cdot 10^{12}\ \mathrm{s} = 5.7 \cdot 10^{4}\ \mathrm{godina}.$$

  - Rješenje konceptualno jednostavno, nije prikladno za primjenu, jer nije učinkovito.

Rješenje: učenje sustava onako kako to radi čovjek!

Klasifikator znamenaka

"Jednostavniji" pristup:



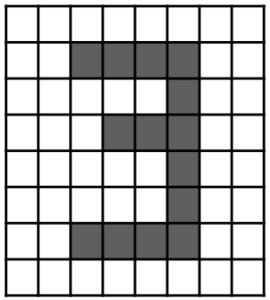
Stanjimo sliku na debljinu jednog piksela (ovo je netrivijalan algoritam iz računalne grafike).

Za svaki piksel brojimo koliko se piksela nalazi u njegovoj 8-okolini.

Zatim promatramo broj i poziciju "karakterističnih" točka:

- krajnjih točaka = točke u čijoj je okolini samo jedan piksel (1),
- točaka oštrog ruba = točke u čijoj su okolini tri piksela $(2, 5, 7)$,
- točaka presjeka = točke u čijoj su okolini četiri piksela $(4, 8)$.

Klasifikator znamenaka



1 – dvije krajnje točke jedna iznad druge

4 – četiri krajnje točke, jedna točka presjeka
i dvije oštre točke,

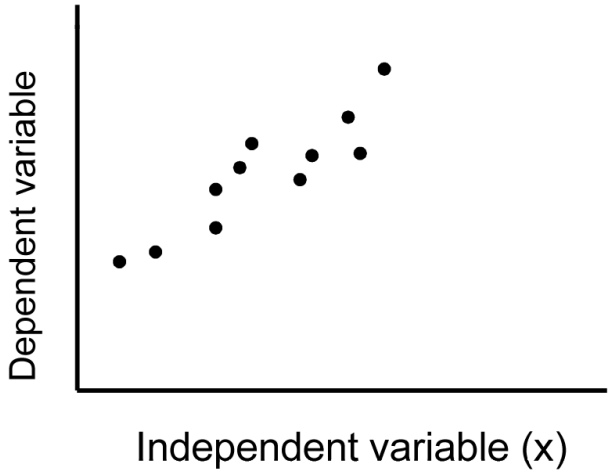
0 – bez krajnjih točaka i bez točaka presjeka,

8 – bez krajnjih točaka i s jednom točkom presjeka,

9 – jedna krajnja točka u donjoj polovini slike, i jedna točka presjeka,

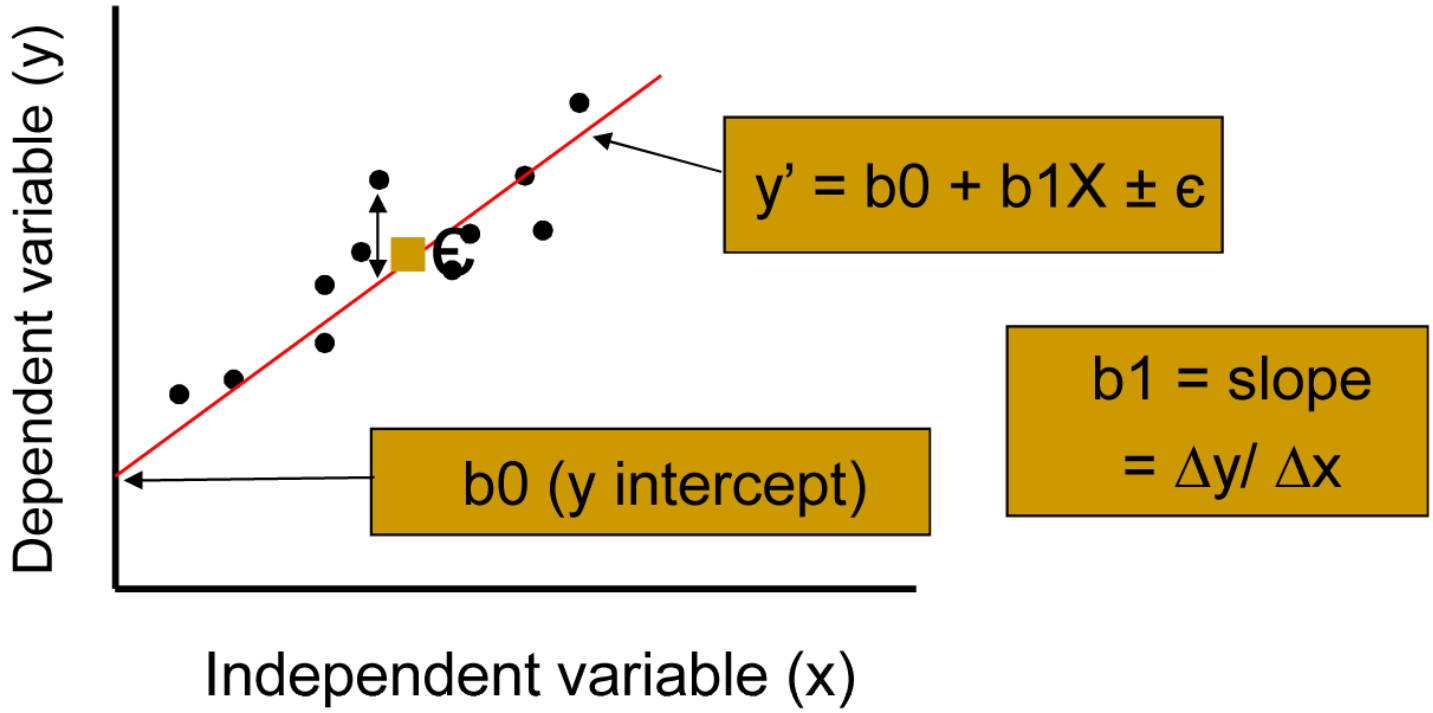
itd.

Regresija



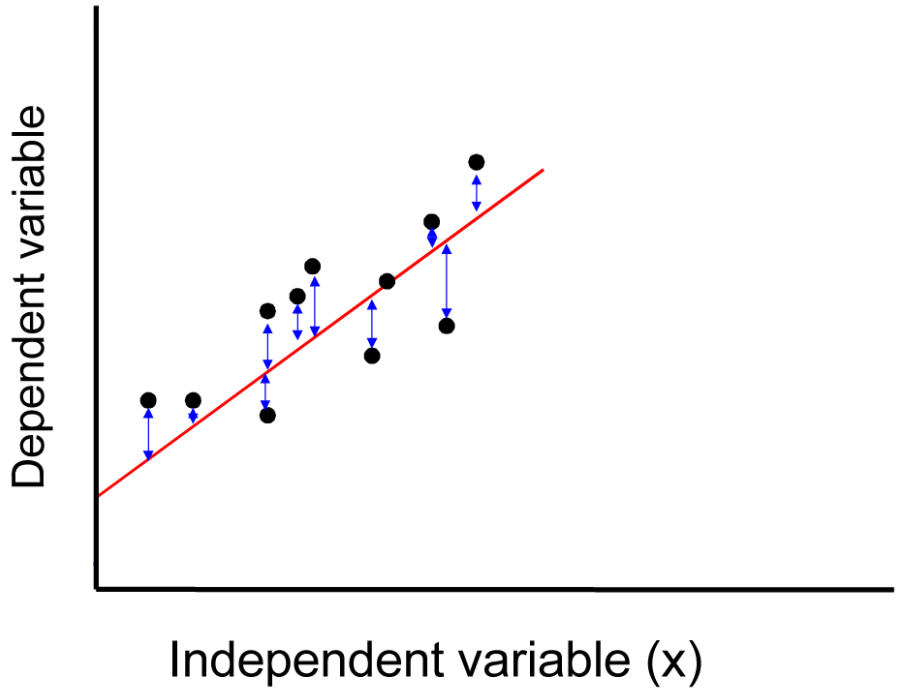
- Regresijska analiza je statistički proces predviđanja veza među varijablama.
- Regresija može objasniti promjenu zavisne varijable korištenjem promjene nezavisne varijable i na taj način objasniti uzročnosti.
- Ako su nezavisna varijabla (ili varijable) dovoljno dobro objašnjavaju varijaciju zavisne varijable, model se može iskoristiti za predviđanje.

Linearna regresija



- Izlaz regresije je funkcija koja predviđa vrijednosti zavisne varijable bazirano na vrijednostima nezavisne varijable.
- Jednostavna regresija aproksimira podatke pravcem.

Regresija metodom najmanjih kvadrata

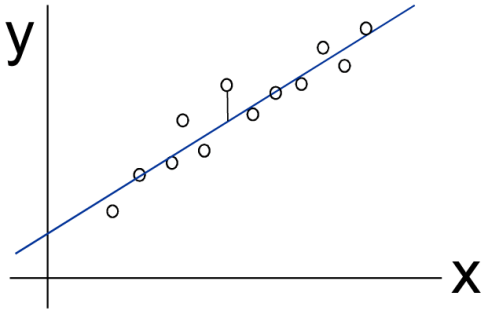


- Regresija metodom najmanjih kvadrata nalazi pravac s najmanjim zbrojem kvadrata predviđenih grešaka.
- Taj broj zove se suma kvadrata grešaka ili S^2.

Regresija metodom najmanjih kvadrata

- Najmanji kvadrati:
  - zadano je n točaka u ravnini: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
  - Nađi pravac $y = ax + b$ koji minimizira sumu kvadrata grešaka:

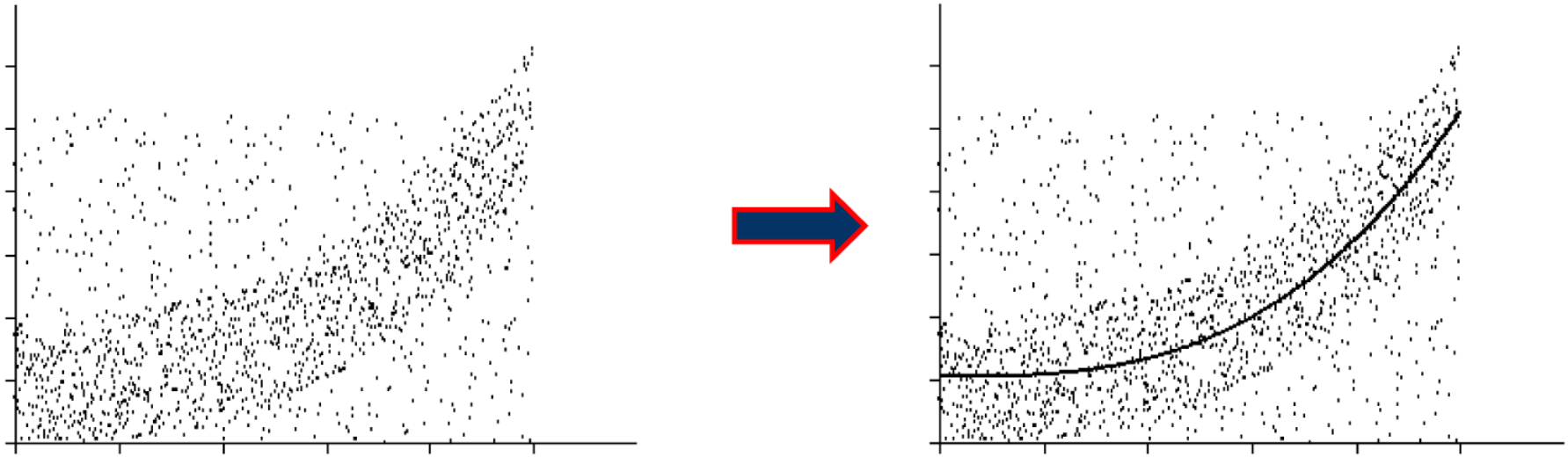
$$S^2 = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$



- Rješenje – nužni uvjeti za postojanje lokalnog ekstrema (sve parcijalne derivacije jednake 0):

$$a = \frac{n \sum_i x_i y_i - \left(\sum_i x_i\right)\left(\sum_i y_i\right)}{n \sum_i x_i^2 - \left(\sum_i x_i\right)^2}, \qquad b = \frac{\sum_i y_i - a \sum_i x_i}{n}.$$

Nelinearna regresija



- I nelinearne funkcije se mogu koristiti u regresiji.
- Uobičajeni izbori su:
  potencije, logaritamske, eksponencijalne i logističke funkcije.
- Općenito, mogu se koristiti bilo koje neprekidne funkcije opisane s nekoliko parametara.

Treniranje — Fitting

- Uobičajeni problem – "pripasavanje" (engl. fitting) modela na neke zadane podatke:
  - regresijom
  - treniranjem neuronske mreže
  - aproksimiranjem
  - rudarenjem podataka
  - učenjem.

Mjere za treniranost — pristranost, varijanca

- Pristranost (eng. Bias, inductive bias)
  - mjeri kako dobro naši modeli (struktura hipoteza) mogu učiti odnose između varijabli i ciljnog koncepta (hipoteze).

- Varijanca
  - otkriva kakve su fluktuacije u kvaliteti aproksimacije ciljnog koncepta/modela kojeg treniramo na različitim (pod)skupovima za učenje.

Mali bias = dobra aproksimacija na podacima za učenje, ali zato i visoka varijanca predviđanja na novim podacima = slaba generalizacija.

Visoki bias = relativno slaba aproksimacija na podacima za učenje, ali mala varijanca predviđanja na novim podacima = dobra generalizacija.

Pretreniravanje

- Pretreniravanje (engl. overfitting)
  – aproksimacija podataka presloženom funkcijom, u želji da se pokupe sve informacije iz zadanih podataka.

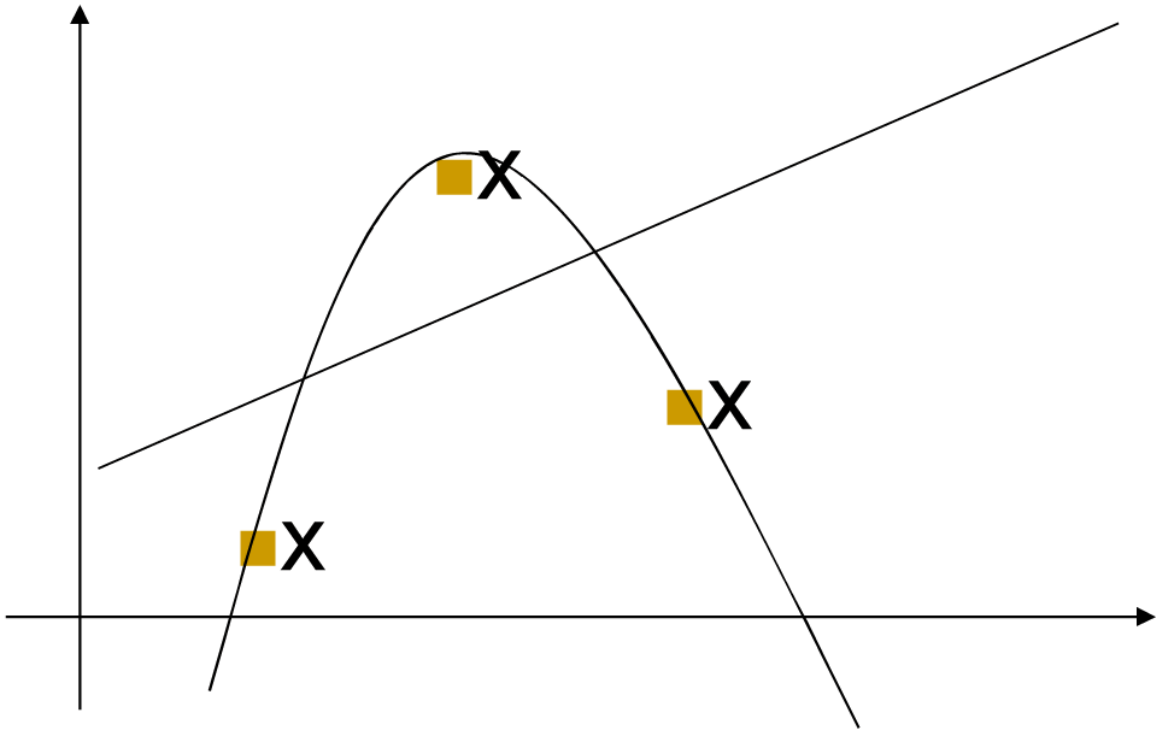
- Tijekom učenja naš model / hipoteza se "pretjerano prilagođava" podacima za učenje (mali bias), i zato predstavlja loš prediktor za neke od novih primjera.

- Grubi pokazatelji pretreniravanja: modeli kod kojih je
  – $broj\ svojstava \gg broj\ primjera\ za\ učenje$ (neodređenost), ili
  – $broj\ svojstava \rightarrow broj\ primjera\ za\ učenje$ (blisko tome).

- Uspješnost modela na skupu za učenje, u principu, reducira dobra generalizacijska svojstva na novim primjerima.

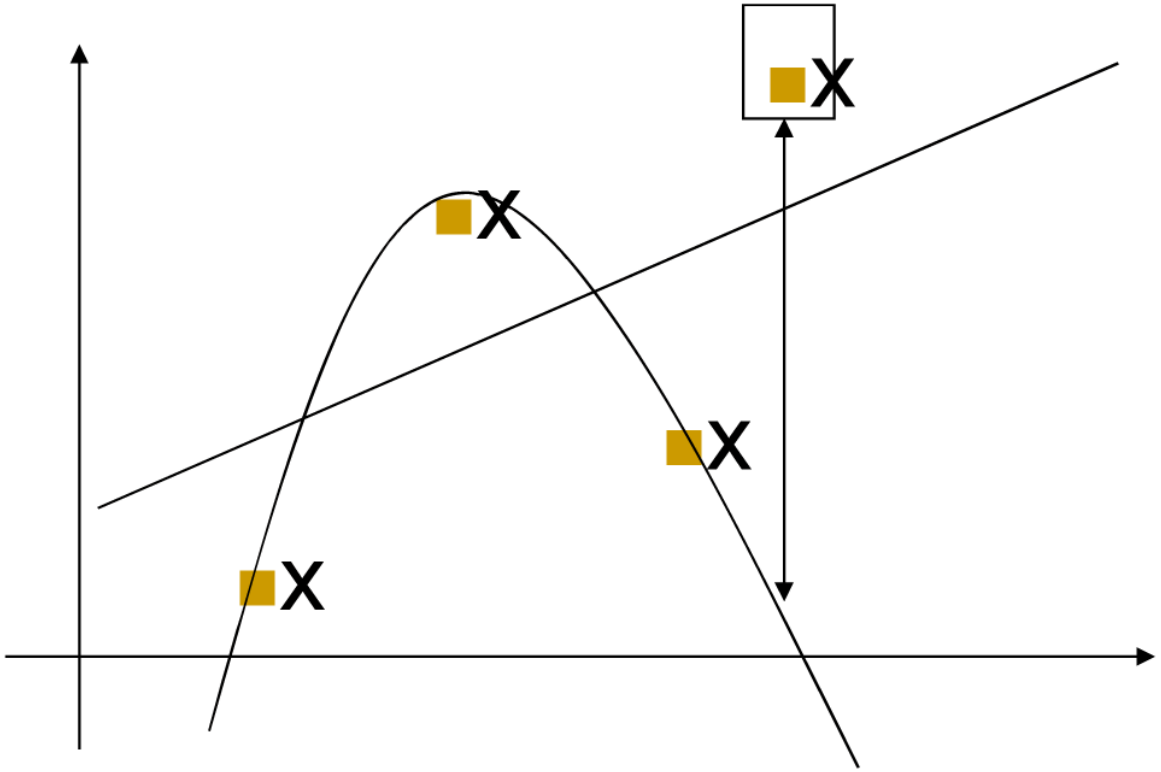
Jednostavni primjer

- Overfitting: model koji se "pripasava" je presložen



Jednostavni primjer (nastavak)

- Overfitting: loša prognostička snaga



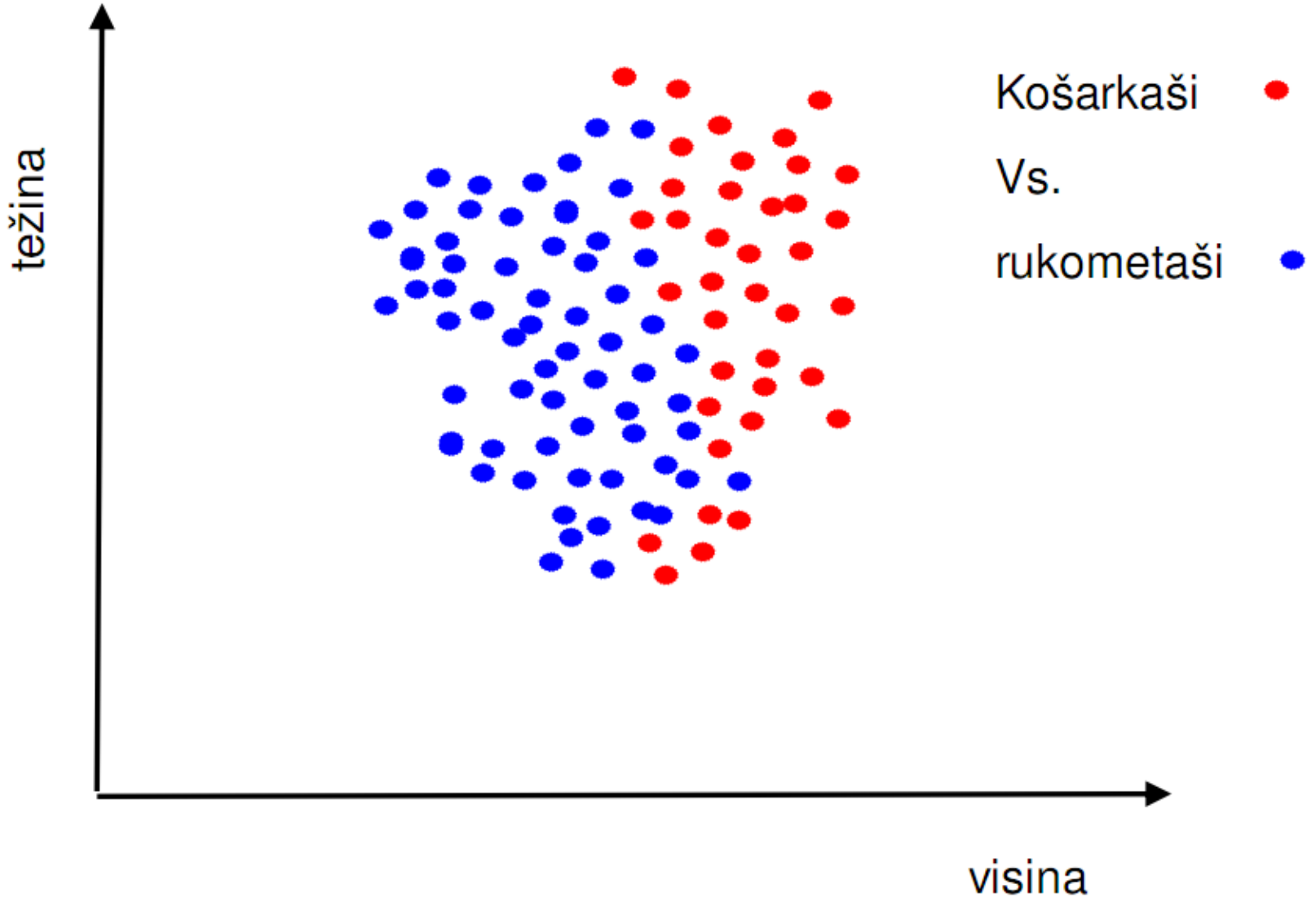
Uzroci pretreniravanja

Obično je uvjetovana
- izborom svojstava/varijabli kojima su opisani primjeri,
- njihovim ponašanjem na nekim primjerima.

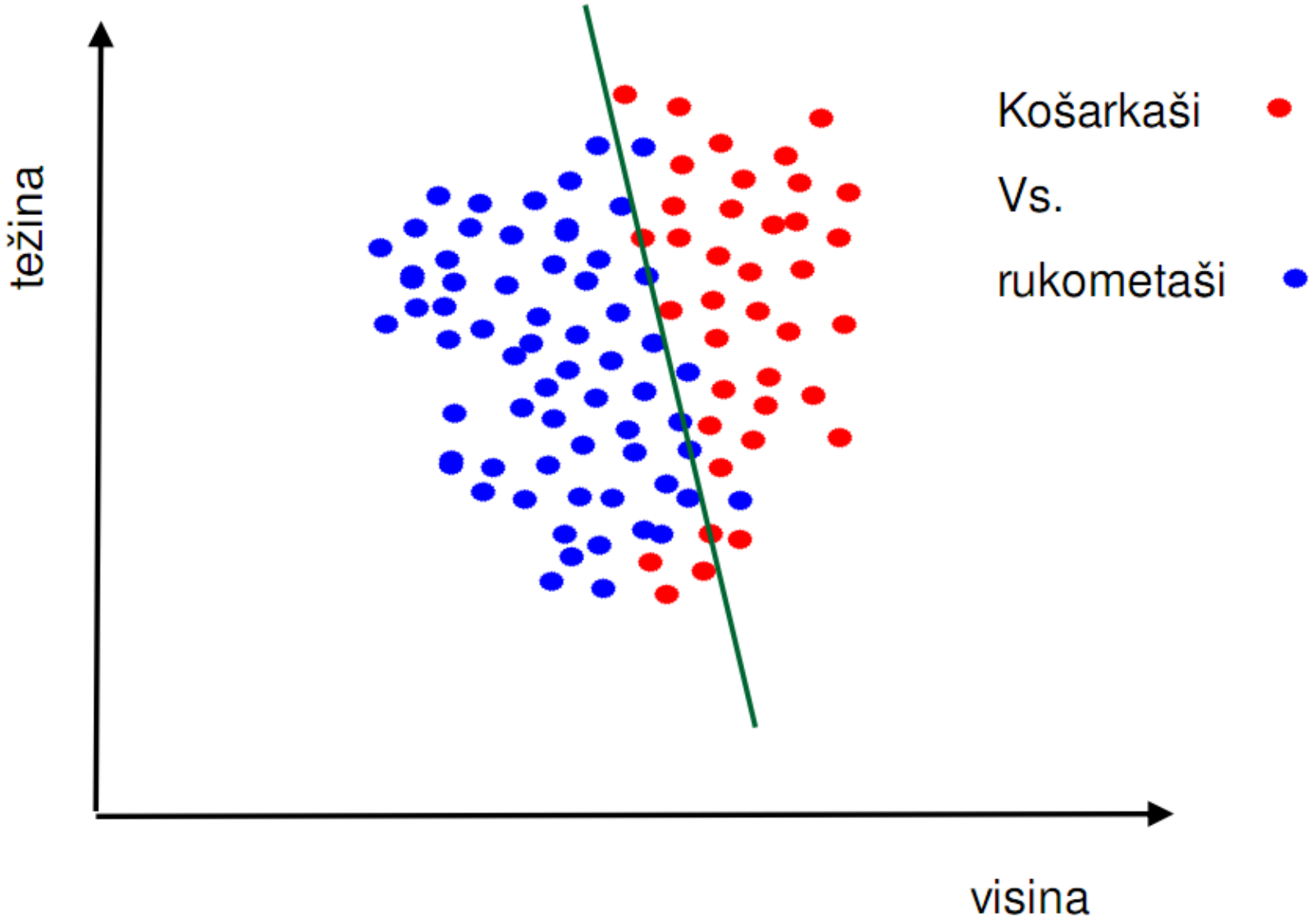
- Šum u sustavu:
  - velika varijabilnost podataka.

- Složenost modela $\Longrightarrow$ mnogo parametara $\Longrightarrow$ više stupnjeva slobode $\Longrightarrow$ veća varijabilnost (= manja nezavisnost)

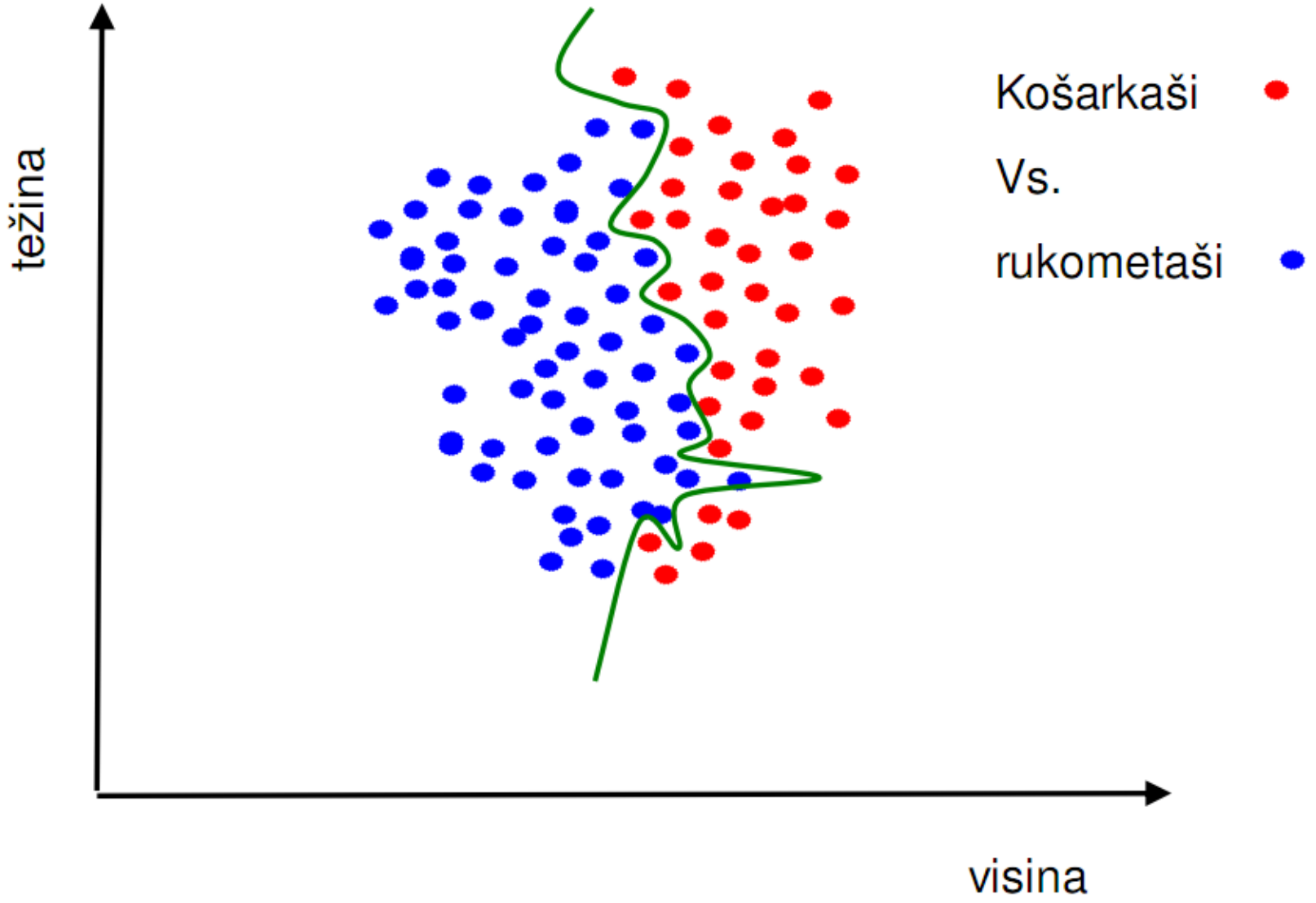
Pretreniravanje — primjer



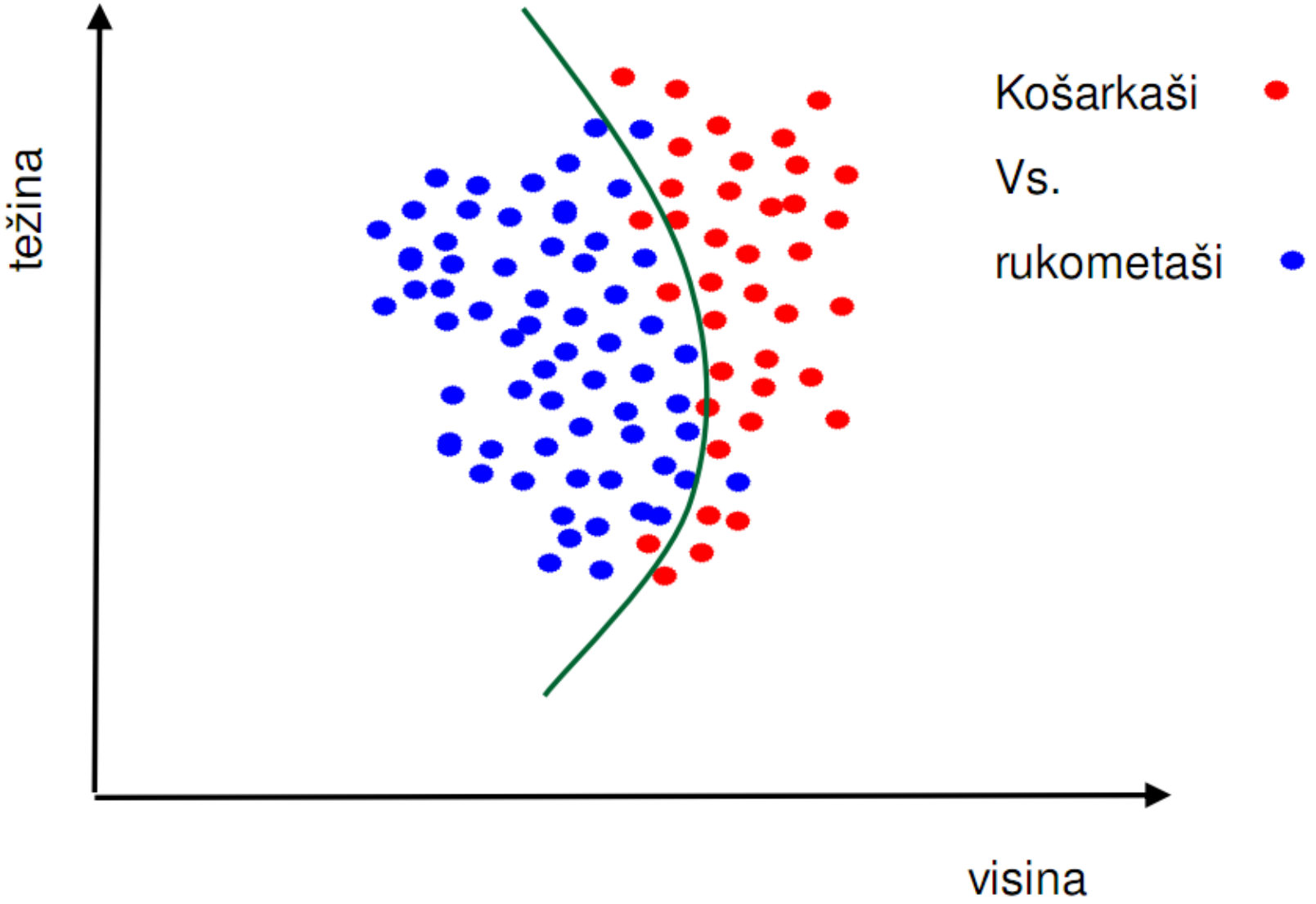
Pretreniravanje — primjer



Pretreniravanje — primjer



"Najbolji" model?

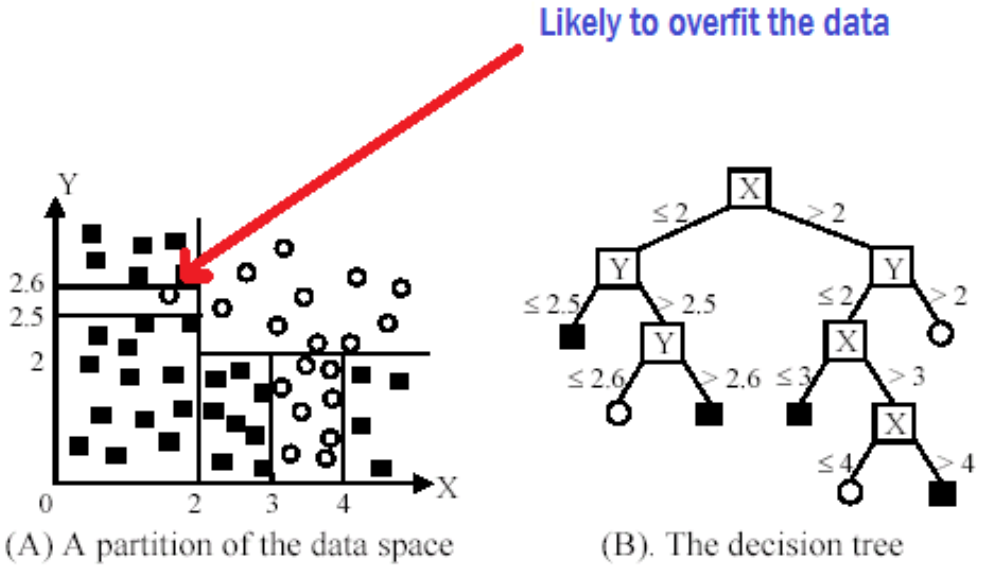


Izbjegavanje pretreniravanja u klasifikaciji

- Overfitting: stablo može napraviti overfit na podacima za treniranje
  - dobra točnost na treniranim podacima, ali loša na test-podacima
  - simptomi: stablo je preduboko i ima previše grana, od kojih neke mogu reflektirati anomalije obzirom na šum ili izdvojene podatke (engl. outliers)

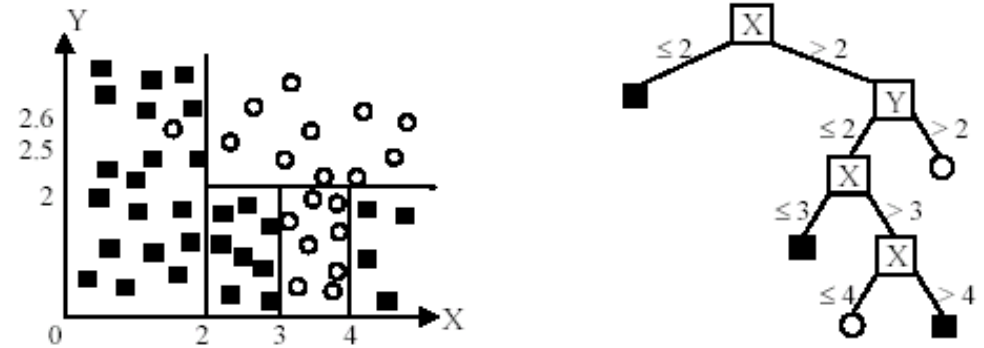
- Dva pristupa za izbjegavanje — oba su rezanje (engl. pruning)
  - Pred-obrezivanje: rano zaustavljanje konstrukcije stabla
    - Teška odluka, jer se ne zna što se može dogoditi ako ostavimo da stablo raste.
  - Post-obrezivanje: uklanjanje grana ili podstabala u "razraslom" stablu.
    - Ova se metoda često koristi. Postoji i statistička metoda za procjenu greške u svakom čvoru za obrezivanje.
    - Validacijski skup se, također, može koristiti za obrezivanje.

Primjer



(A) A partition of the data space

(B). The decision tree



Ostalo o učenju korištenjem stabala odlučivanja

- Od stabla do pravila, "obrezivanje" pravila
- Obrada promašenih vrijednosti
- Obrada iskrivljenih distribucija
- Obrada atributa i klasa različitih cijena
- Konstrukcija atributa
- ...