OSNOVE UMETNE INTELIGENCE
2022/23

učenje dreves iz šumnih podatkov

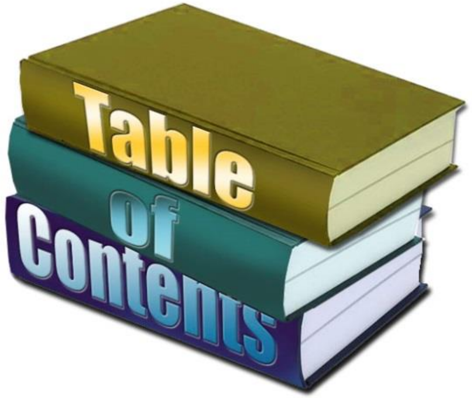
rezanje dreves: REP in MEP

ocenjevanje učenja

Pridobljeno znanje s prejšnjih predavanj

- **strojno učenje**
  - vrste učenja: nadzorovano, nenadzorovano, spodbujevano
  - atributna predstavitev podatkov: primeri, atributi, ciljna spremenljivka
  - klasifikacijski (diskretna ciljna spremenljivka – razred)
    in regresijski problemi (zvezna ciljna spremenljivka – označba)
  - hipoteze: konsistentnost, splošnost, razumljivost/kompleksnost
  - učenje odločitvenih dreves:
    - algoritem TDIDT
    - entropija, ocenjevanje kakovosti atributov z informacijskim prispevkom
  - težave z večvrednostnimi atributi (druge mere za kakovost atributov)
  - binarizacija atributov
  - kratkovidnost algoritma TDIDT

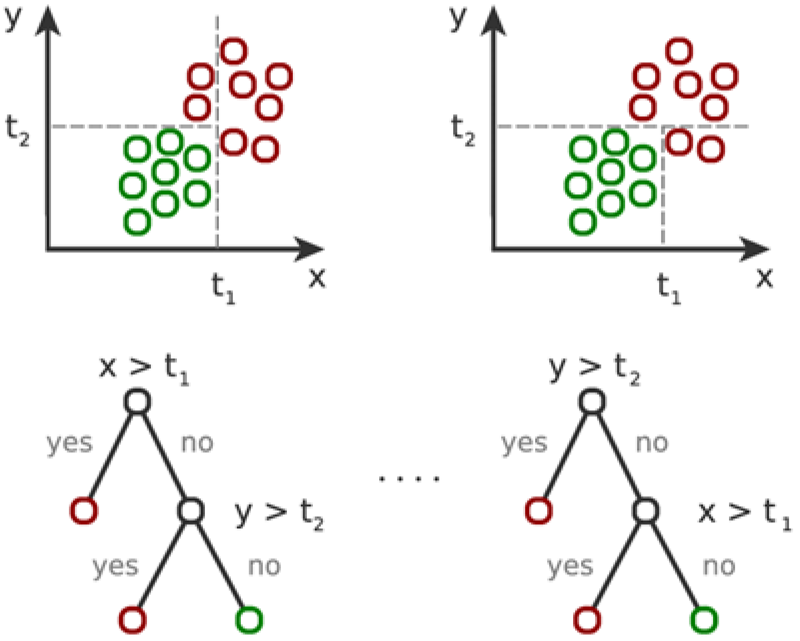
Pregled

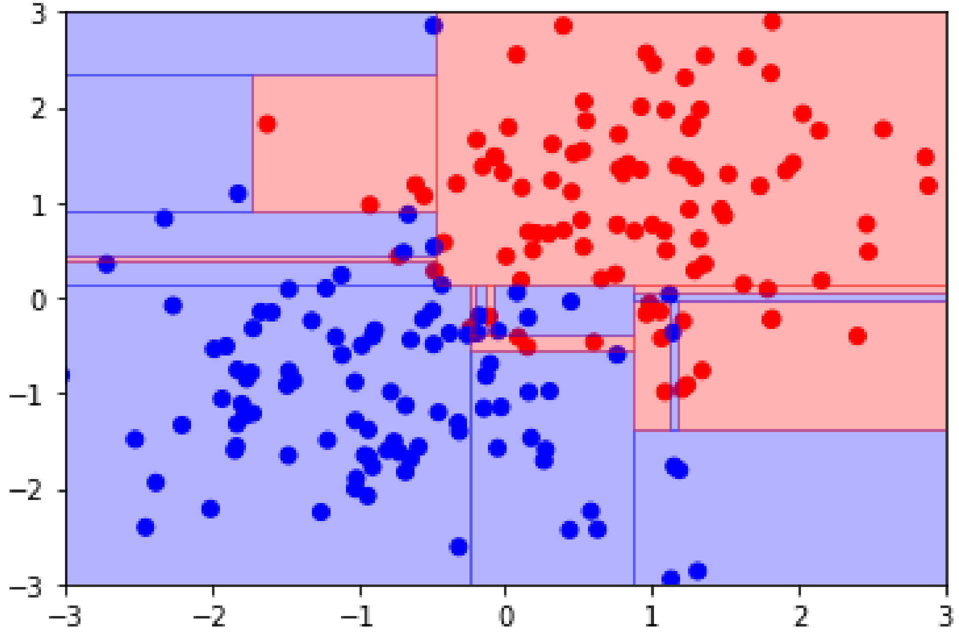


- strojno učenje
    - uvod v strojno učenje
    - učenje odločitvenih dreves
    - učenje dreves iz šumnih podatkov (rezanje dreves)
    - ocenjevanje učenja

Prostor hipotez odločitvenih dreves

- **zvezni atributi** (npr. višina, dolžina, IQ, koncentracija ozona, poraba el. energije, ipd.)
- v vozliščih običajno testiramo primerjavo zveznega atributa z izbrano mejo (večje/manjše)
- takšna odločitvena drevesa delijo prostor na particije (hiper-kvadre), katerih meje so vzporedne koordinatnim osem
- dva primera:

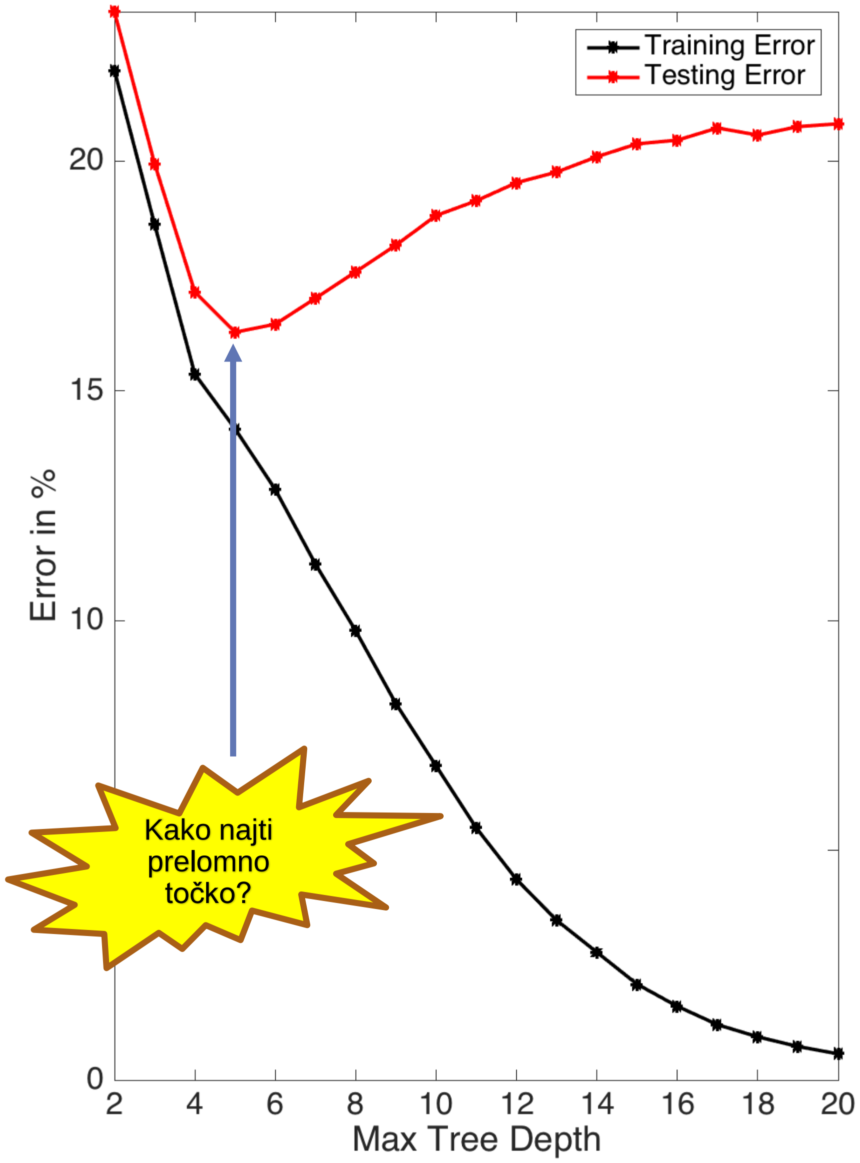




Pristranost na učni množici

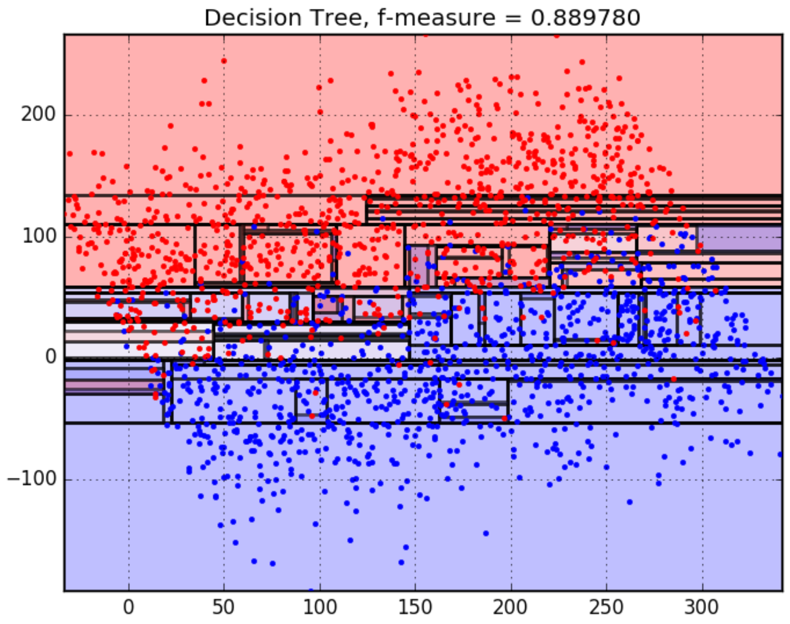
- cilj: maksimiziraj pričakovano točnost drevesa (vendar ne na učnih podatkih - **pretirano prilagajanje**?)

- alternativa – uporaba **nevidenih primerov**:
  - izvzamemo posebno množico **testnih primerov**, če imamo dovolj podatkov (ostane manj podatkov za gradnjo)
  - tipična delitev podatkov: **učna množica** (70%), **testna množica** (30%)



Učenje dreves iz šumnih podatkov

- večja drevesa → večje prilagajanje učnih podatkom
- kaj pa, če podatki **niso popolni** (premalo primerov/ atributov) ali so v učnih primerih **napake**? Pojavijo se lahko težave:
  - **učenje šuma** in ne dejanske (skrite) funkcije, ki generira podatke
  - pretirano prilagajanje vodi v **prevelika drevesa**
  - **slaba razumljivost** dreves
  - posledica: **nižja klasifikacijska točnost** na novih/nevidenih podatkih



- pojav **pretiranega prileganja** (angl. overfitting)
- rešitev: rezanje odločitvenega drevesa

- primer uspeha iz prakse → → → → →

primer iz prakse: lociranje primarnega tumorja (domena *Primary tumor*

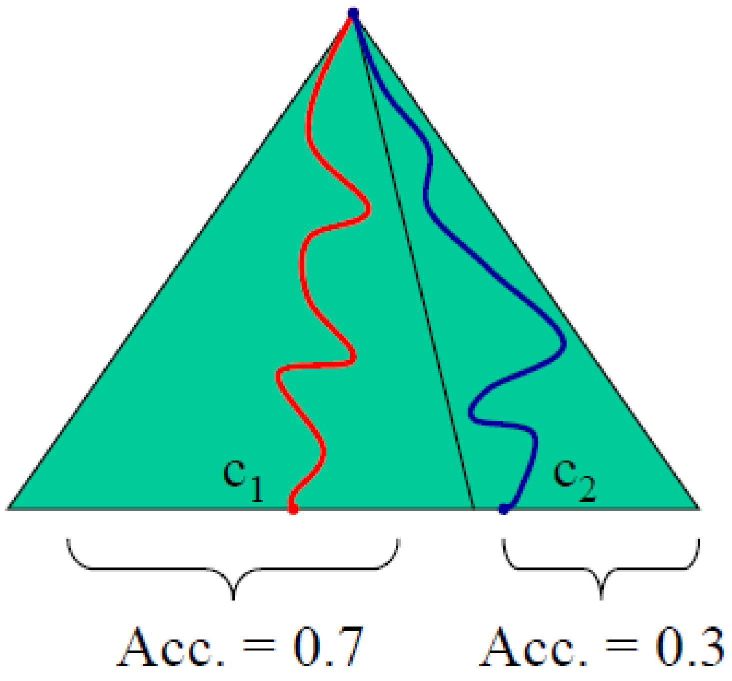
	Klas. točnost
Pretirano pril. drevo (150 vozlišč)	41%
Porezano drevo (15 vozlišč)	45%
Privzeta točnost	24,7%
Zdravniki	42%

Rezanje odločitvenih dreves – kako?

- **premislek**: nižji deli drevesa (bližji listom) predstavljajo večje lokalno prilagajanje učnim podatkom, ki so lahko posledica šuma

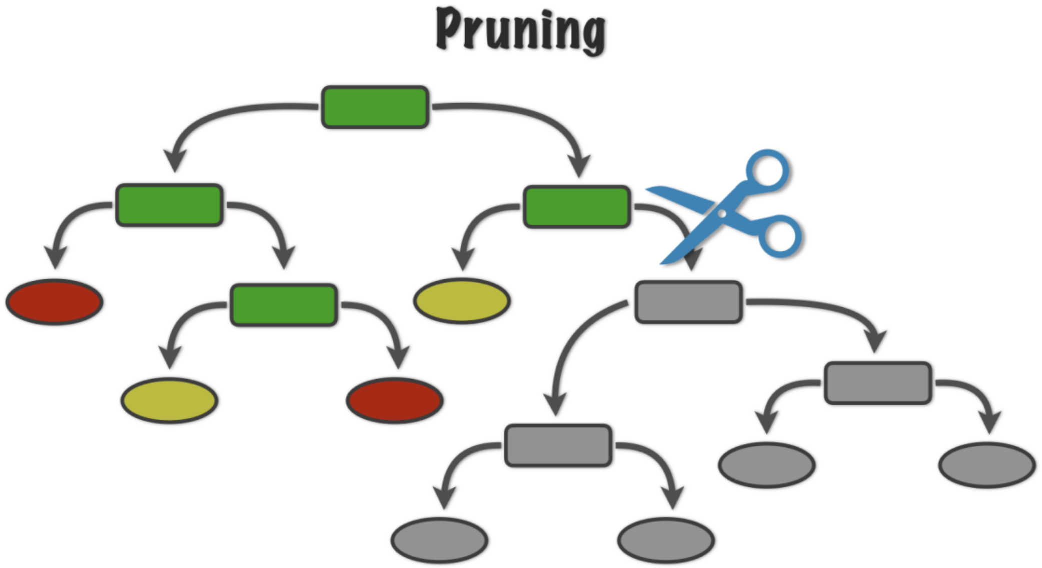
- **ideja**: odstranimo (režemo) spodnje dele drevesa, da dosežemo boljšo posplošitev naučenega drevesa (in klasifikacijsko točnost na nevidenih podatkih)

- primer nizke točnosti drevesa pri skrajnem primeru pretiranega prilagajanja:
  - dva razreda, c_1 in c_2, $p(c_1) = 0{,}7$, $p(c_2) = 0{,}3$
  - privzeta točnost (točnost večinskega razreda) = $0{,}7$
  - drevo, zgrajeno do konca (en primer v vsakem listu)
  - pričakovana točnost:
    $\mathrm{p}_{c1} \times CA_{c1} + \mathrm{p}_{c2} \times CA_{c2} =$
    $= 0{,}7 \times 0{,}7 + 0{,}3 \times 0{,}3 = 0{,}58$
    (manj kot privzeta točnost!)



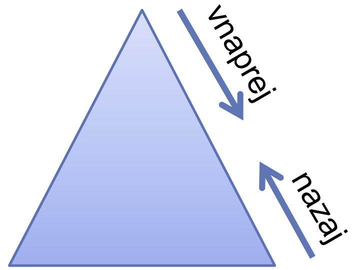
Rezanje odločitvenih dreves

- vprašanja:
  - kako to doseči,
  - kje rezati,
  - kombinatorično število možnih porezanih dreves

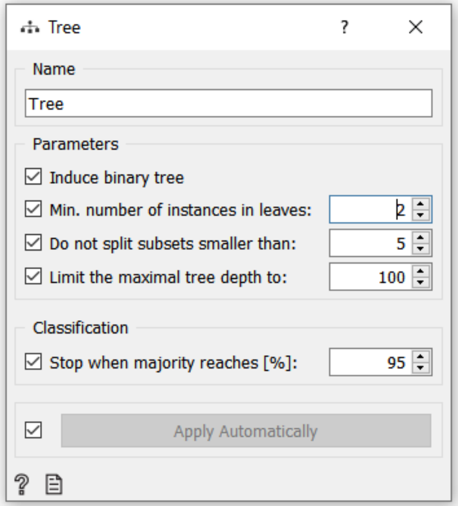


https://towardsdatascience.com/decision-trees-60707f06e836?gi=e5086ac9a15e

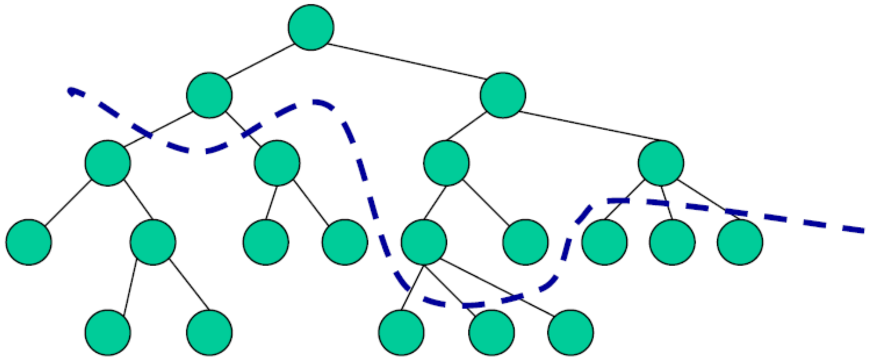
Strategije rezanja



- **rezanje vnaprej** (angl. *forward pruning, pre-stopping*): uporaba dodatnega kriterija za *zaustavitev gradnje drevesa* glede na obseg šuma (na podlagi: števila primerov, večinski razred, smiselnost delitve v poddrevesa glede na informacijski prispevek itd.)
  - **hitrejše**
  - **kratkovidno**, upošteva samo zgornji del drevesa



- **rezanje nazaj** (angl. *post-pruning*): rezanje, ki po gradnji celotnega drevesa, *odstrani manj zanesljive dele drevesa* (opisujejo šum, zgrajeni iz manj podatkov in z manj informativnimi atributi)
  - **počasneje**, oblika post-procesiranja
  - upošteva informacijo iz **celega drevesa**
  - pristopa:
    - **rezanje z zmanjševanjem napake (reduced error pruning, REP)**
    - **rezanje z minimizacijo napake (minimal error pruning, MEP)**

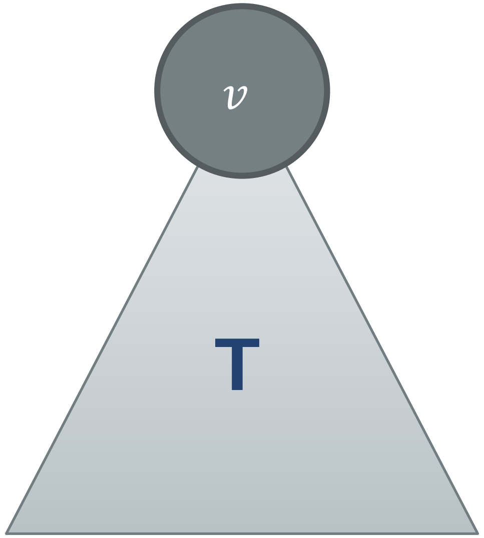


Rezanje z zmanjševanjem napake (REP)

- angl. *reduced error pruning* (REP)

- uporablja posebno rezalno (validacijsko) množico, potrebna primerna velikost za zanesljivost; tipične velikosti pri delitvi podatkov:
  - **učna množica** (70%), od tega:
    - množica za gradnjo (growing set) 70%
    - rezalna množica (pruning set) 30%
  - **testna množica** (30%)

- postopek:
  - potuj od listov navzgor (prični s starši listov)
  - za vsako vozlišče v izračunaj **dobitek rezanja**:
    št. napačnih klasifikacij v drevesu $T -$ št. napačnih klasifikacij v vozlišču v
  - če je dobitek ≥ 0, obreži in nadaljuj postopek s staršem, sicer ustavi postopek

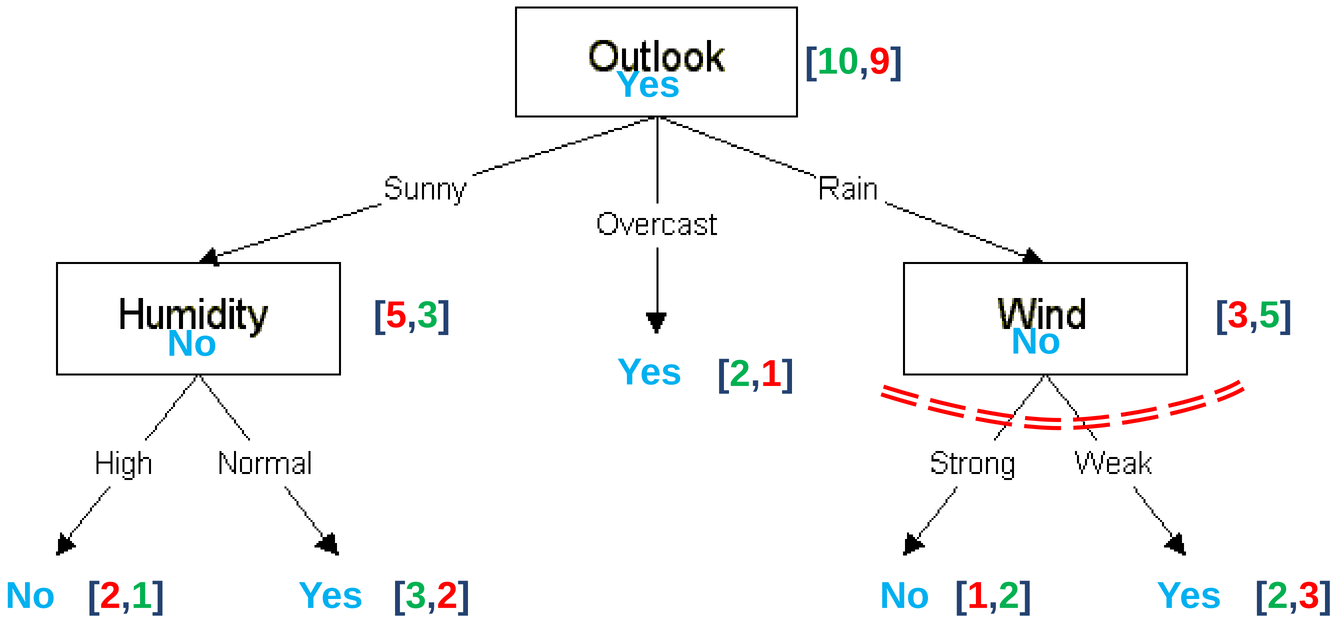


Rezanje z zmanjševanjem napake (REP)

- primer: podane so klasifikacije primerov iz rezalne množice v posameznih vozliščih, uporabi REP:

Legenda:

- **[#Yes,#No]**
- **razred vozlišča**
- **(zeleno) pravilna klasifikacija**
- **(rdeče) napačna klasifikacija**



Izpitna naloga

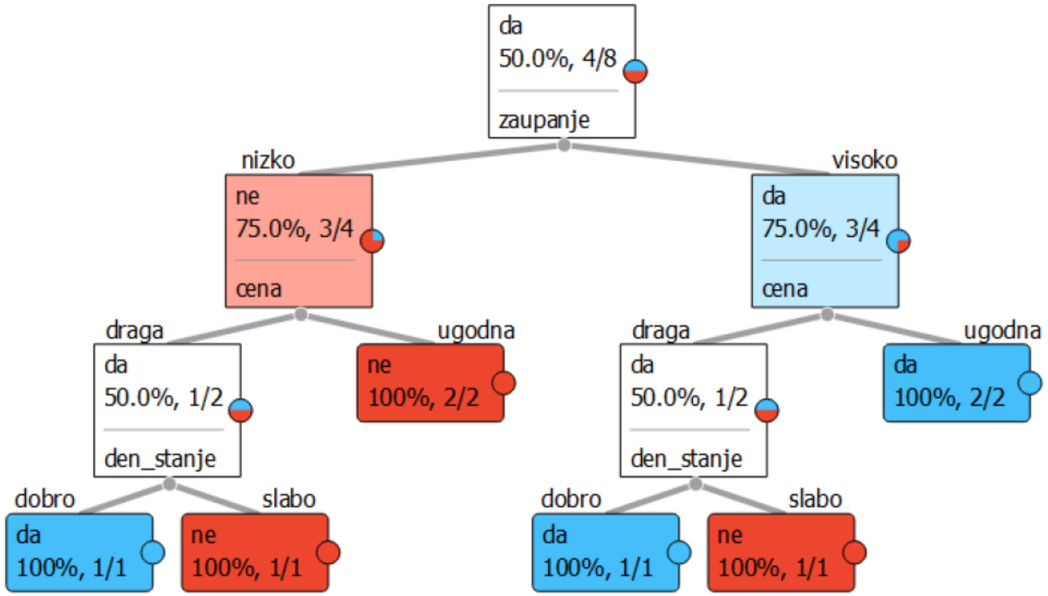
Legenda oznak v vozliščih:
- *večinski razred: da*
- *vseh primerov v vozlišču: 8*
- *točnost v vozlišču (delež primerov, ki pripadajo večinskemu razredu v vozlišču) je 4/8=50%*

- naloga, podobna izpitni nalogi (1. izpitni rok, 23. 1. 2019)

2. NALOGA (10t):

Podano je odločitveno drevo na sliki, ki ga uporabljamo za odločanje o nakupu valute Bitcoin. Drevo je zgrajeno iz učnih podatkov, ki imajo atribute: *zaupanje* (zaupanje v prodajalca – nizko ali visoko), *cena* (nakupna cena – ugodna ali draga) in *den_stanje* (lastno denarno stanje – slabo ali dobro). Razred je spremenljivka *nakup*, ki ima lahko vrednosti "da" (kupimo) ali "ne" (ne kupimo).

Legenda: Vozlišča v drevesu prikazujejo razred (da/ne), delež večinskega razreda in število primerov, ki pripadajo razredu da/ne.

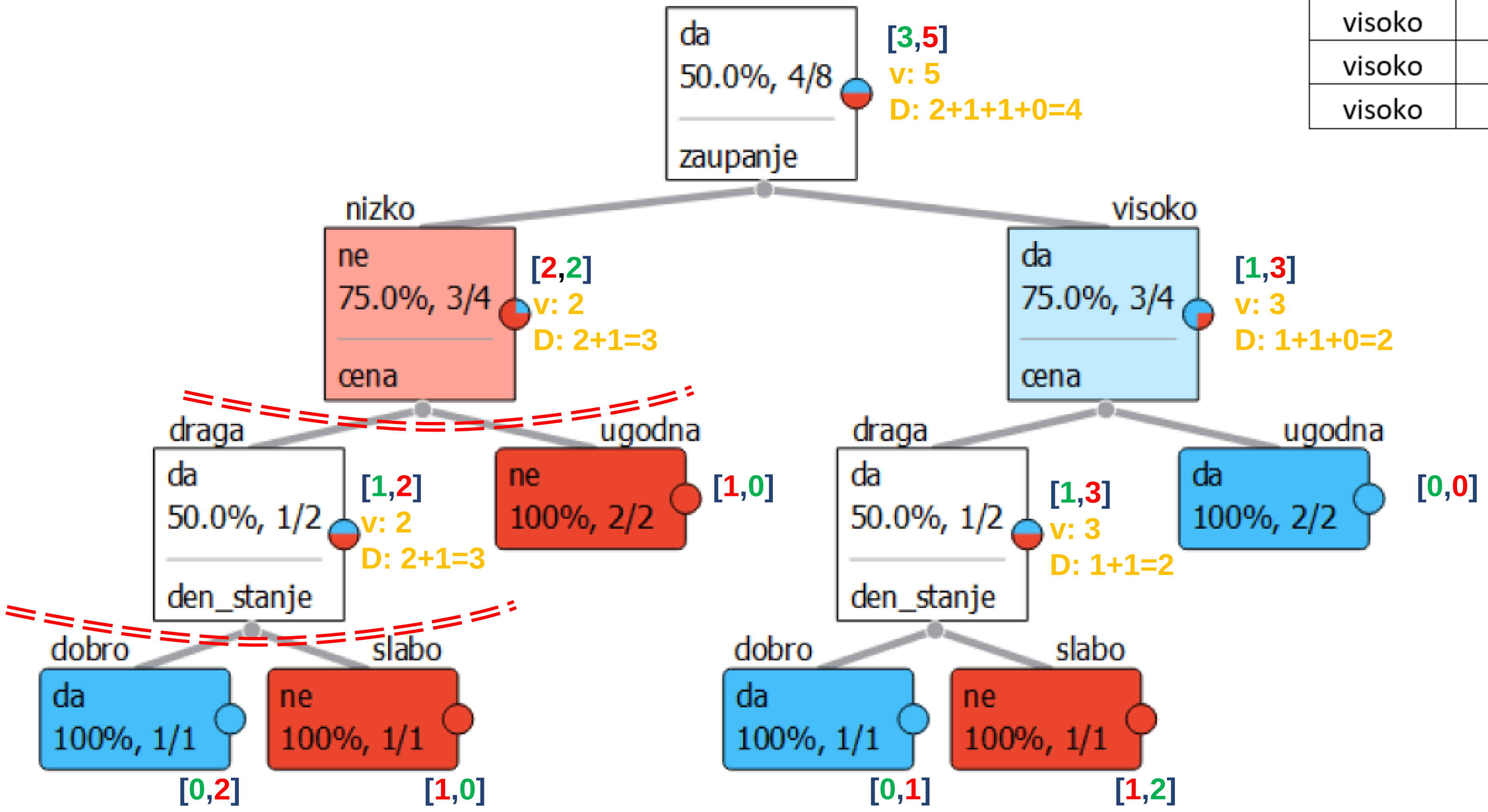


zaupanje	cena	den_stanje	nakup
nizko	draga	dobro	ne
nizko	draga	dobro	ne
nizko	draga	slabo	da
nizko	ugodna	dobro	da
visoko	draga	dobro	ne
visoko	draga	slabo	da
visoko	draga	slabo	ne
visoko	draga	slabo	ne

c) (3t) Z uporabo rezalne množice na desni strani poreži zgornje drevo s postopkom zmanjševanja napake (REP). Rezanje prikaži na zgornji skici drevesa.

Izpitna naloga

zaupanje	cena	den_stanje	nakup
nizko	draga	dobro	ne
nizko	draga	dobro	ne
nizko	draga	slabo	da
nizko	ugodna	dobro	da
visoko	draga	dobro	ne
visoko	draga	slabo	da
visoko	draga	slabo	ne
visoko	draga	slabo	ne

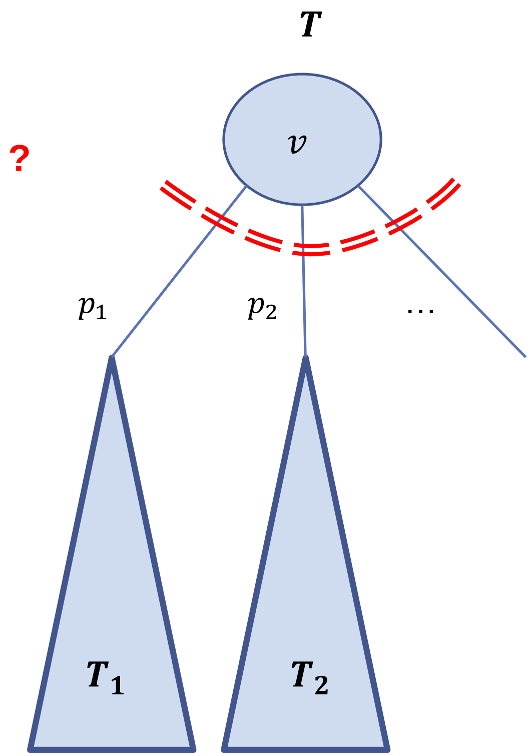


Rezanje z minimizacijo napake (MEP)

- angl. *minimal error pruning* (MEP) (Niblett in Bratko, 1986; Cestnik in Bratko, 1991)
- uporablja množico za gradnjo drevesa (in ne ločene rezalne množice)
- cilj: poreži drevo tako, da je ocenjena klasifikacijska napaka minimalna

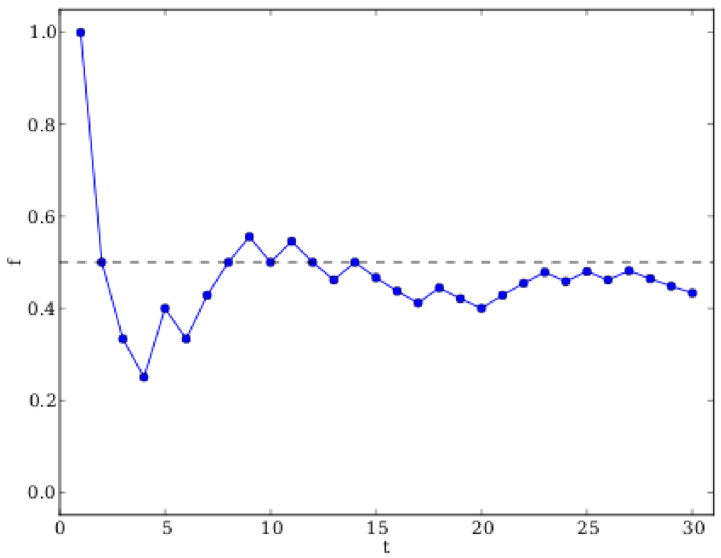
- za vozlišče v izračunamo:
  - **statično napako** (verjetnost klasifikacije v napačen razred)
    $e(v) = p(razred \neq C|v)$, C je večinski razred v v
  - **vzvratno napako** (angl. *backed-up error*)
    $\sum_i p_i E(T_i) = p_1 E(T_1) + p_2 E(T_2) + \cdots$
- režemo, če je **statična napaka manjša od vzvratne napake**
- **napaka optimalno obrezanega drevesa** je torej
  $E(T) = \min(e(v), \sum_i p_i E(T_i))$
  $E(T) = e(v)$, če je v list

- (namesto **minimizacije napake $\boldsymbol{E}$** (zgoraj) lahko problem obrnemo in **maksimiziramo točnost $\boldsymbol{CA}$** – primer, ki sledi)



Ocenjevanje verjetnosti

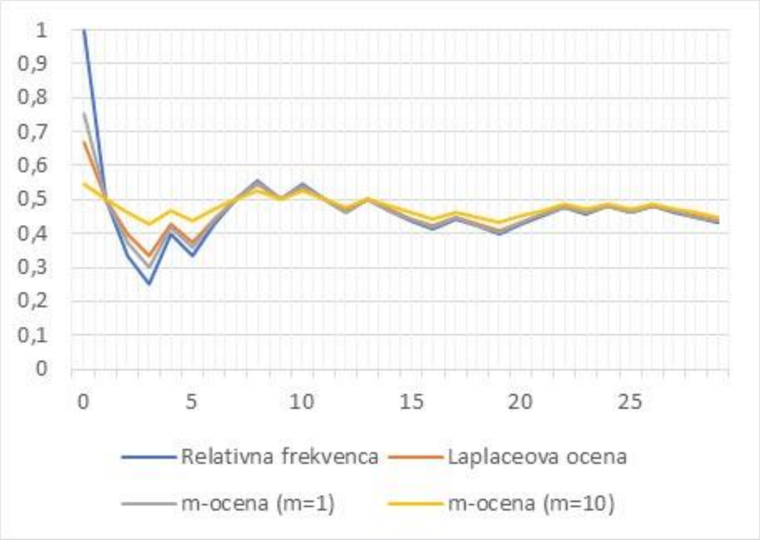
- kako oceniti statično napako v vozlišču v?
- primeri uporabe **relativne frekvence** (N – št. primerov v vozlišču, n – št. primerov, ki pripadajo večinskemu razredu C):
  - $N = 1, n = 1$ → točnost=100%
  - $N = 2, n = 1$ → točnost= 50% ? (samo z enim dodatnim primerom)



- težave:
  - potrebujemo *oceno verjetnosti*, ki je **stabilna** tudi pri manjšem številu primerov
  - ***ocena verjetnosti***: mera, ki izraža **približek prave verjetnosti dogodka** in ima **zaželene matematične lastnosti**, za njo pa ne veljajo nujno osnovni aksiomi s področja verjetnosti
  - smiselno je, da ocena verjetnosti upošteva tudi **apriorno verjetnost** (verjetnost, ki jo poznamo o problemu – npr. 50% za izid meta kovanca)

https://intl.siyavula.com/read/maths/grade-10/probability/14-probability-01

Ocenjevanje verjetnosti



boljši oceni verjetnosti:

- **Laplaceova ocena verjetnosti:**

$$p = \frac{n+1}{N+k}$$

n – št. primerov, ki pripadajo razredu C,
N – št. vseh primerov
k – št. vseh razredov

- k je problematičen parameter; ocena ne upošteva apriorne verjetnosti

- **m-ocena verjetnosti**

delež upoštevanja apriorne verjetnosti

delež upoštevanja relativne frekvence

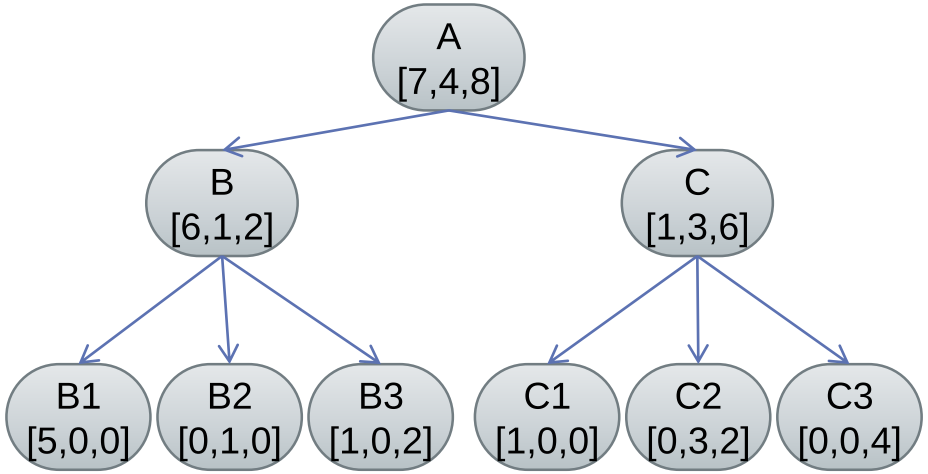
$$p = \frac{n + p_a m}{N+m} = p_a \cdot \frac{m}{N+m} + \frac{n}{N} \cdot \frac{N}{N+m}$$

p_a – apriorna verjetnost razreda C
m – parameter ocene (vpliva na delež upoštevanja apriorne verjetnosti)

- malo šuma – majhen m – malo rezanja / veliko šuma – velik m – veliko rezanja
- posplošitev Laplaceove ocene za $m = k$ in $p_a = 1/k$

Vaja

- primer: Bratko: Prolog Programming for AI
- Podano je odločitveno drevo za klasifikacijo v tri razrede (x, y in z) z naslednjimi apriornimi verjetnostmi razredov: $p_a(x) = 0{,}4$, $p_a(y) = 0{,}3$, $p_a(z) = 0{,}3$. Številke v oglatih oklepajih $[x, y, z]$ predstavljajo frekvence primerov v vozlišču, ki pripadajo ustreznim razredom. Obreži drevo s postopkom MEP in vrednostjo $m = 8$.



Vaja

$E(T) = \min(e(v), \sum_i p_i E(T_i))$, $E(T) = e(v)$, če je v list
oziroma
$\mathrm{CA(T)} = \max(ca(v), \sum_i p_i CA(T_i))$, $CA(T) = ca(v)$, če je v list

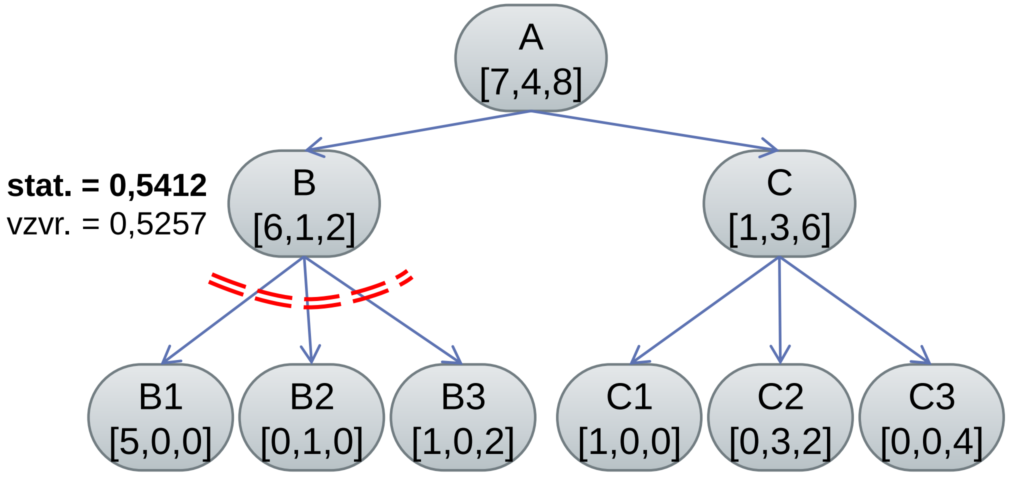
- klasifikacijske točnosti v listih B1, B2 in B3:

$$p(x|B1) = \frac{n+m\cdot p_a(x)}{N+m} = \frac{5+8\cdot 0{,}4}{5+8} = \mathbf{0{,}6308}$$
$$p(y|B1) = \frac{0+8\cdot 0{,}3}{5+8} = 0{,}1846$$
$$p(z|B1) = \frac{0+8\cdot 0{,}3}{5+8} = 0{,}1846$$

$$p(x|B2) = \frac{0+8\cdot 0{,}4}{1+8} = 0{,}3556$$
$$p(y|B2) = \frac{1+8\cdot 0{,}3}{1+8} = \mathbf{0{,}3778}$$
$$p(z|B2) = \frac{0+8\cdot 0{,}3}{1+8} = 0{,}2667$$

$$p(x|B3) = \frac{1+8\cdot 0{,}4}{3+8} = 0{,}3818$$
$$p(y|B3) = \frac{0+8\cdot 0{,}3}{3+8} = 0{,}2182$$
$$p(z|B3) = \frac{2+8\cdot 0{,}3}{3+8} = \mathbf{0{,}4}$$

- vzvratna točnost v vozlišču B: $\frac{5}{9}\cdot 0{,}6308 + \frac{1}{9}\cdot 0{,}3778 + \frac{3}{9}\cdot 0{,}4 = 0{,}5257$
- statična točnost v vozlišču B:
  $\boldsymbol{p(x|B)} = \frac{\mathbf{6+8\cdot 0{,}4}}{\mathbf{9+8}} = \mathbf{0{,}5412}$, $p(y|B) = 0{,}2$, $p(z|B) = 0{,}2588$
- statična točnost je večja od
  vzvratne točnosti → porežemo
- nadaljujemo z vozliščema
  C in A ...



Vaja

$E(T) = \min(e(v), \sum_i p_i E(T_i))$, $E(T) = e(v)$, če je v list
oziroma
$\mathrm{CA(T)} = \max(ca(v), \sum_i p_i CA(T_i))$, $CA(T) = ca(v)$, če je v list

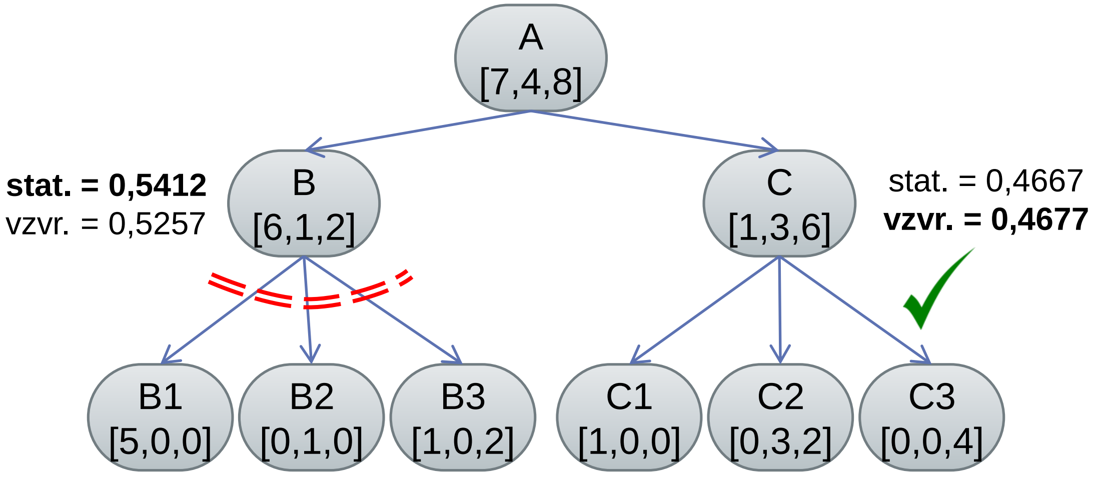
- klasifikacijske točnosti v listih C1, C2 in C3:

$$p(x|C1) = \frac{n+m\cdot p_a(x)}{N+m} = \frac{1+8\cdot 0{,}4}{1+8} = \mathbf{0{,}4667}$$
$$p(y|C1) = \frac{0+8\cdot 0{,}3}{1+8} = 0{,}2667$$
$$p(z|C1) = \frac{0+8\cdot 0{,}3}{1+8} = 0{,}2667$$

$$p(x|C2) = \frac{0+8\cdot 0{,}4}{5+8} = 0{,}2462$$
$$p(y|C2) = \frac{3+8\cdot 0{,}3}{5+8} = \mathbf{0{,}4154}$$
$$p(y|C2) = \frac{2+8\cdot 0{,}3}{5+8} = 0{,}3385$$

$$p(x|C3) = \frac{0+8\cdot 0{,}4}{4+8} = 0{,}2667$$
$$p(y|C3) = \frac{0+8\cdot 0{,}3}{4+8} = 0{,}2000$$
$$p(z|C3) = \frac{4+8\cdot 0{,}3}{4+8} = \mathbf{0{,}5333}$$

- vzvratna točnost v vozlišču C: $\frac{1}{10}\cdot 0{,}4667 + \frac{5}{10}\cdot 0{,}4154 + \frac{4}{10}\cdot 0{,}5333 = 0{,}4677$
- statična točnost v vozlišču C:
  $p(x|C) = 0{,}2333, p(y|C) = 0{,}3000, \boldsymbol{p(z|C)} = \mathbf{0{,}4667}$
- vzvratna točnost je večja od
  statične točnosti → ne porežemo
- nadaljujemo z vozliščem A ...



$E(T) = \min(e(v), \sum_i p_i E(T_i))$, $E(T) = e(v)$, če je v list
oziroma
$\mathrm{CA(T)} = \max(ca(v), \sum_i p_i CA(T_i))$, $CA(T) = ca(v)$, če je v list

Vaja

- klasifikacijske točnosti CA v podrevesih s koreni v B in C:

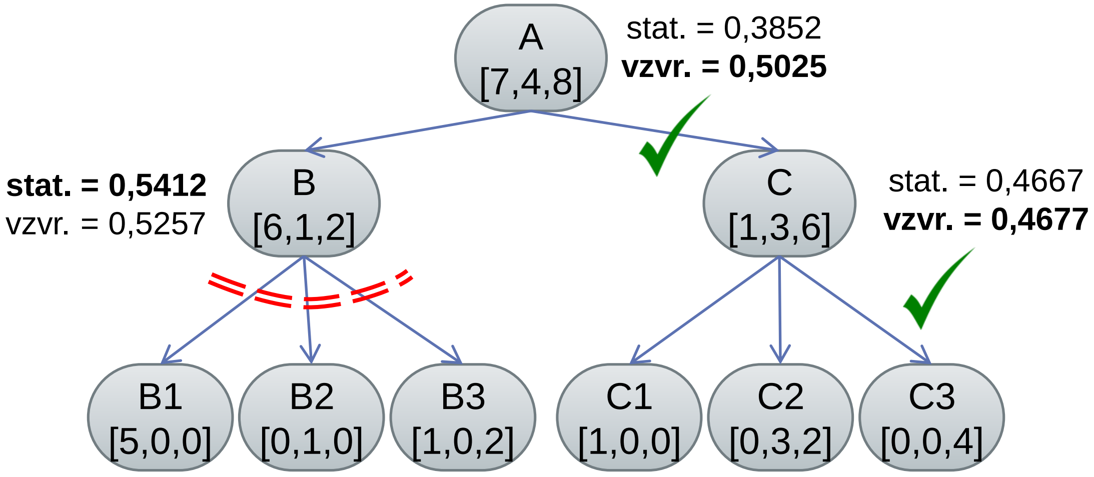
$$CA(B) = \max(ca(B), \sum_i p_i CA(B_i)) = 0{,}5412$$

$$CA(C) = \max(ca(C), \sum_i p_i CA(C_i)) = 0{,}4677$$

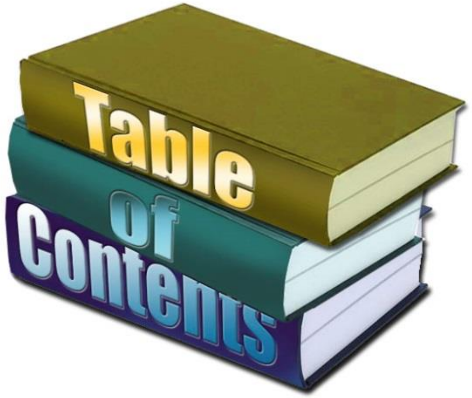
- vzvratna točnost v vozlišču A: $\frac{9}{19} \cdot 0{,}5412 + \frac{10}{19} \cdot 0{,}4677 = 0{,}5025$
- statična točnost v vozlišču A:

$$p(x|A) = 0{,}3778,\ p(y|A) = 0{,}2370,\ \boldsymbol{p(z|A)} = \frac{\mathbf{8+8 \cdot 0{,}3}}{\mathbf{19+8}} = \mathbf{0{,}3852}$$

- vzvratna točnost je večja od statične točnosti → ne porežemo

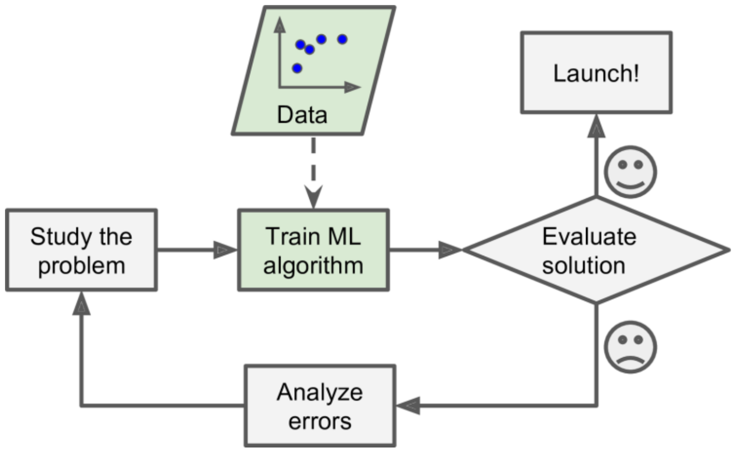


Pregled



- strojno učenje
  - uvod v strojno učenje
  - učenje odločitvenih dreves
  - učenje dreves iz šumnih podatkov (rezanje dreves)
  - ocenjevanje učenja

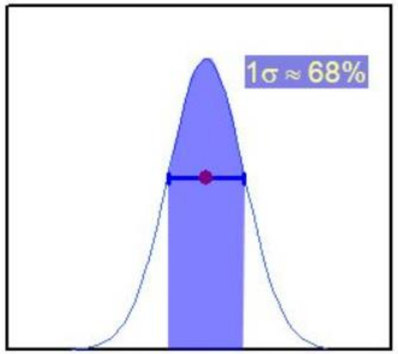
Ocenjevanje učenja

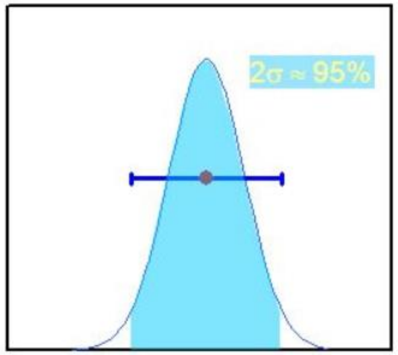


- kriteriji za ocenjevanje hipotez:
  - **točnost** (angl. *accuracy*) (**konsistentnost**, **splošnost**)
  - **razumljivost** (angl. *comprehensibility*) – subjektivni kriterij ali tudi **kompleksnost** (angl. *complexity*)

- ocenjevanje točnosti:
  - na **učnih** podatkih (angl. *training set, learning set)*
  - na **testnih** podatkih (angl. *testing set, test set*)
    - izločimo del učnih podatkov, s katerimi simuliramo ne-videne podatke
    - želimo si, da je testna množica reprezentativna za nove podatke
    - uporabimo lahko **intervale zaupanja** v oceno uspešnosti na testni množici, ki upoštevajo število testnih primerov
  - na **novih** (ne-videnih) podatkih (angl. *new data, unseen data*)
    - na njih bo naučeni sistem dejansko deloval

Ocenjevanje učenja





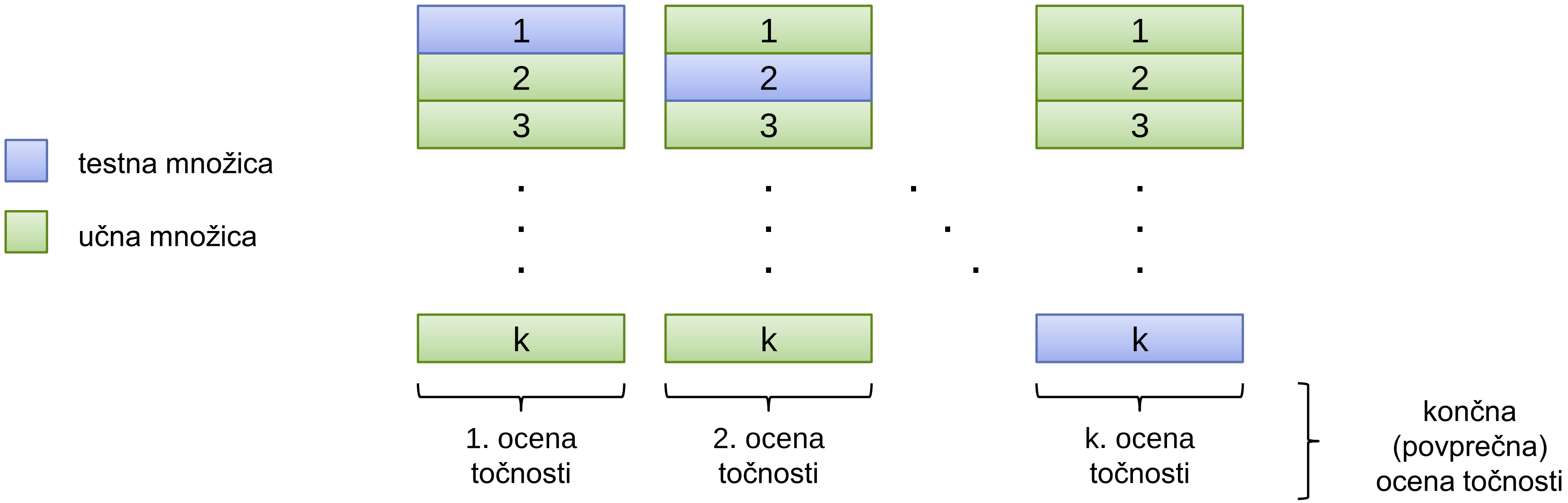
- nasprotujoča si cilja:
  - potrebujemo čim več podatkov za **uspešno učenje**
  - potrebujemo čim več podatkov za **zanesljivo ocenjevanje točnosti** (večje število testnih primerov nam daje ožji interval zaupanja v oceno točnosti)

- rešitev:
  - kadar je učnih podatkov dovolj, lahko izločimo **testno množico** (angl. *holdout test set*)
  - alternativa: **večkratne delitve** na učno in testno množico

- različni načini **vzorčenja testnih primerov**:
  - naključno, nenaključno (npr. prečno preverjanje)
  - poljubno ali stratificirano (zagotovimo enako porazdelitev razredov kot v učni množici)

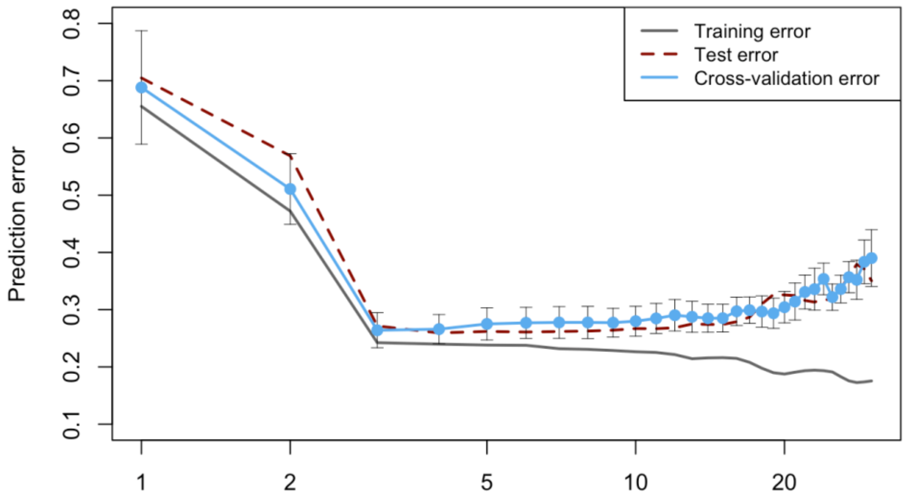
Prečno preveranje

- poseben primer **večkratnega učenja** in testiranja
- **k-kratno prečno preverjanje** (angl. *k-fold cross-validation*):
  - celo učno množico razbij na *k* disjunktnih podmnožic
  - za vsako od *k* podmnožic:
    - uporabi **množico** kot **testno množico**
    - uporabi **preostalih *k-1*** množic kot **učno množico**
  - povpreči dobljenih *k* ocen točnosti v končno oceno



Prečno preveranje

- v praksi **najpogosteje: k=10** (10-kratno prečno preverjanje)
- vplive izbranega razbitja podatkov na podmnožice lahko zmanjšamo tako, da tudi prečno preverjanje **večkrat (npr. 10x) ponovimo** (torej 10×10=100 izvajanj učnega algoritma) in rezultate povprečimo
- poseben primer prečnega preverjanja je metoda **izloči enega** (angl. leave-one-out, LOO)
  - *k* je enak številu primerov (vsaka testna množica ima samo en primer)
  - najbolj stabilna ocena glede učinkov razbitja na podmnožice
  - časovno zelo zamudno, primerno za manjše množice
- iz meritev na vseh podmnožicah je možno izračunati tudi varianco/ intervale zaupanja




Obravnava manjkajočih atributov,
naivni Bayesov klasifikator