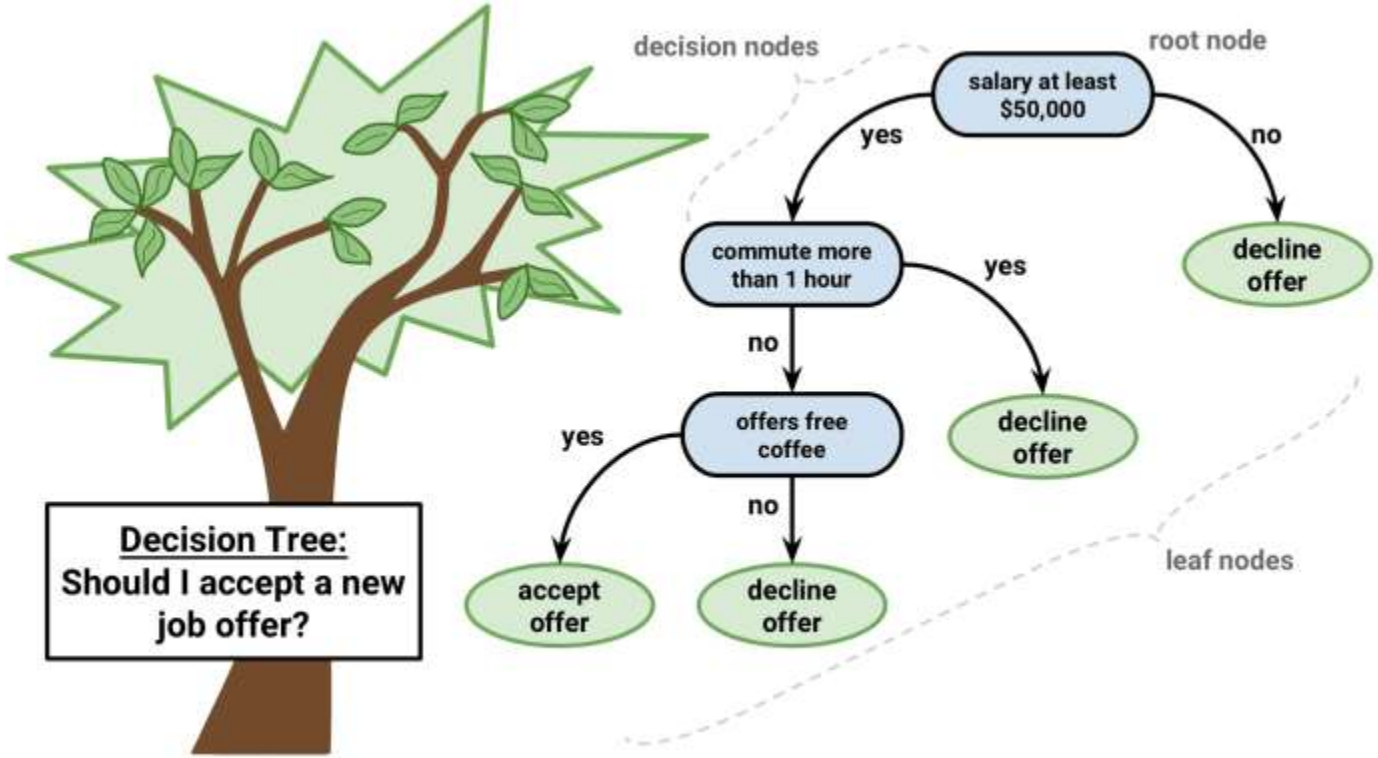
Umetna inteligenca

Odločitvena drevesa

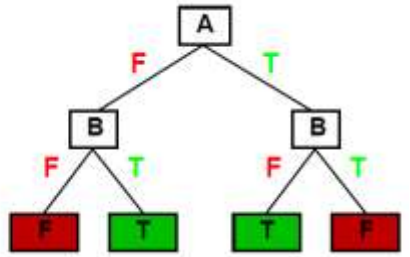
- Gradnja drevesa
- Ocenjevanje atributov
- Binarizacija atributov
- Kratkovidnost
- Učenje iz šumnih podatkov
- Rezanje drevesa



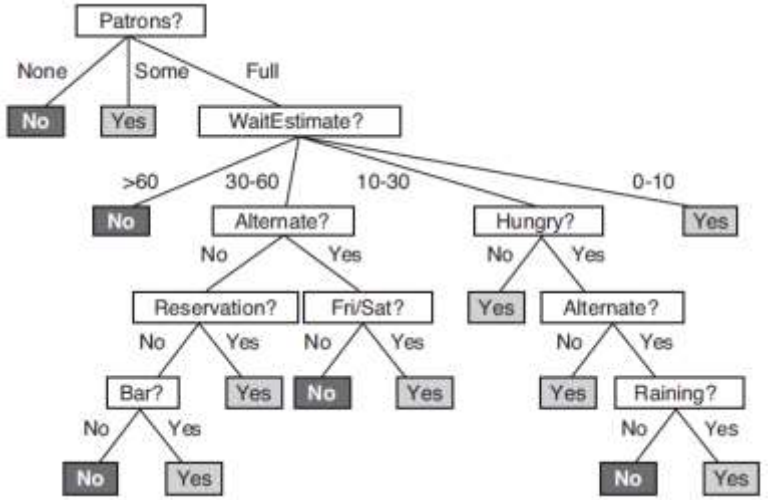
Odločitveno drevo

- ponazarja relacijo med vhodnimi vrednostmi (atributi) in odločitvijo (ciljna spremenljivka – razred)
  - notranja vozlišča: test glede na vrednost posameznega atributa
  - listi: odločitev (vrednost ciljne spremenljivke)
  - pot: konjunkcija pogojev v notranjih vozliščih na poti, ki vodi do lista
- poseben primer: binarna klasifikacija (razred ima dve možni vrednosti (npr. pozitivni/negativni, strupen/užiten itd.)

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

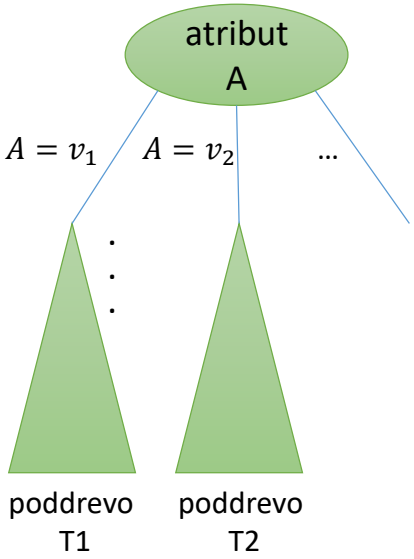


Gradnja odločitvenega drevesa

- cilj: zgradi **čim manjše** drevo, ki je **konsistentno** z učnimi podatki
- prostor iskanja: kombinatoričen, vsa možna drevesa (neučinkovito!)

- **hevristični požrešni algoritem**:
  - izberi najbolj pomemben atribut – tisti, ki najbolj odločilno vpliva na klasifikacijo primera – in razdeli vse učne primere v poddrevesa glede na njegove vrednosti,
  - rekurzivno ponovi za poddrevesa (na ustreznih podmnožicah primerov),
  - če vsi elementi v listu pripadajo istemu razredu ali vozlišča ni možno deliti naprej (ni razpoložljivih atributov), ustavi gradnjo.

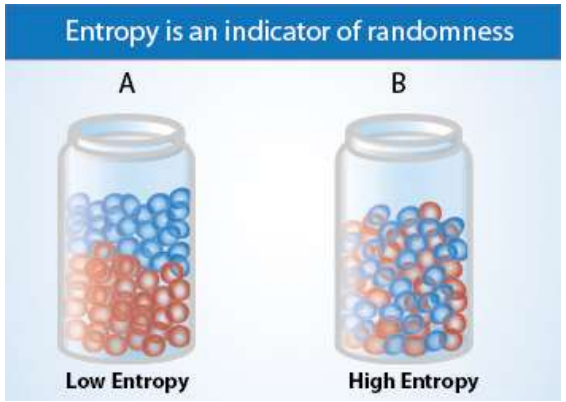
- imenovano tudi ***Top Down Induction of Decision Trees (TDIDT)***
- primeri implementacij: ID3, CART, Assistant, C4.5, C5, …



Izbor najbolj pomembnega atributa

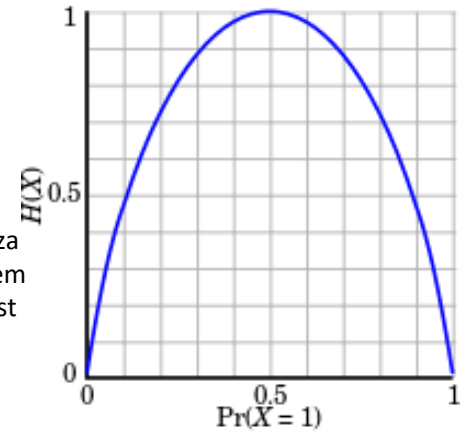
- najboljši atribut je tisti, ki razdeli učno množico v najbolj "čiste" podmnožice (glede na razred)
- uporabimo lahko **mero entropije:**

$$H = -\sum_{k} p_k \log_2 p_k$$



- mera nečistoče oz. mera nedoločenosti naključne spremenljivke (Shannon in Weaver, 1949)
- enota: količina informacije v bitih, ki jo pridobimo
- primeri:
  - met kovanca: 1 bit informacije
  - poskus s štirimi enako verjetnimi možnimi izidi: 2 bita informacije
  - poskus z dvema izidoma, od katerih je eden 99%: ~ 0 bitov informacije

primer: entropija za dvorazredni problem glede na verjetnost prvega razreda



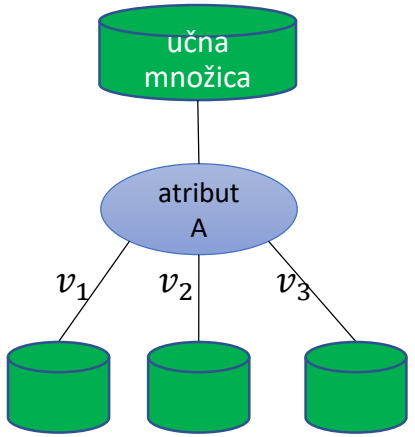
Informacijski prispevek

- dejansko nas zanima **znižanje entropije** (nedoločenosti) ob delitvi učne množice glede na vrednosti atributa A
- **informacijski prispevek**:

$$Gain(A) = I - I_{res}(A)$$

$$I_{res} = -\sum_{v_i \in A} p_{v_i} \sum_{c} p(c|v_i) \log_2 p(c|v_i)$$

- najbolj informativni atribut **maksimizira informacijski prispevek** (minimizira I_{res})



informacija (entropija)

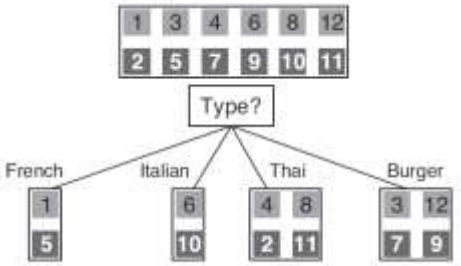
$$I = H(C)$$

rezidualna informacija (entropija)

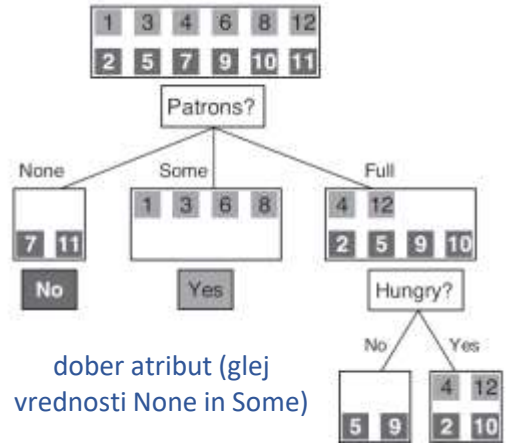
$$I_{res} = \sum_{i} p_{v_i} \cdot H(C|v_i)$$

Izbor najbolj pomembnega atributa

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T



slab atribut (slabo loči pozitivne in negativne primere)



dober atribut (glej vrednosti None in Some)

- znižanje entropije ob delitvi učne množice glede na vrednosti atributa A
- $Gain(A) = I - I_{res}(A)$

$$I = -p(T)\log_2 p(T) - p(F)\,log_2\,p(F) = -\frac{6}{12}\log_2\frac{6}{12} - \frac{6}{12}log_2\frac{6}{12} = -\log_2\frac{1}{2} = 1$$

$$I_{res}(Type) = -\frac{2}{12}\left[\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}log_2\frac{1}{2}\right] - \frac{2}{12}\left[\frac{1}{2}log_2\frac{1}{2} + \frac{1}{2}log_2\frac{1}{2}\right] - \frac{4}{12}\left[\frac{2}{4}log_2\frac{2}{4} + \frac{2}{4}log_2\frac{2}{4}\right] - \frac{4}{12}\left[\frac{2}{4}log_2\frac{2}{4} + \frac{2}{4}log_2\frac{2}{4}\right] = 1$$

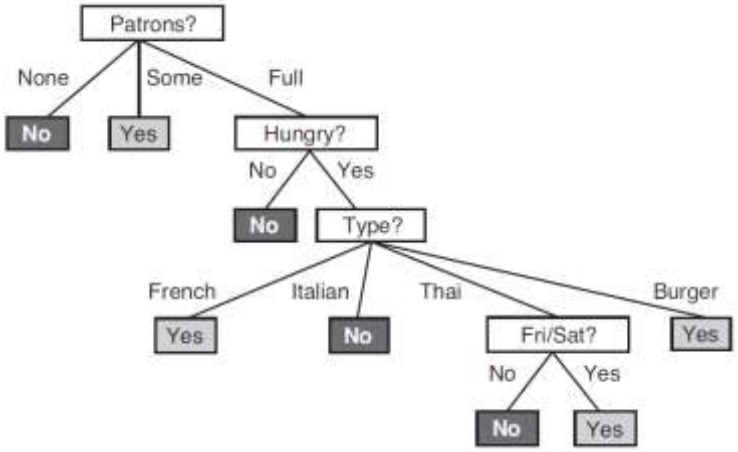
$$I_{res}(Patrons) = -\frac{2}{12}\cdot 0 - \frac{4}{12}\cdot 0 - \frac{6}{12}\left[\frac{2}{6}log_2\frac{2}{6} + \frac{4}{6}log_2\frac{4}{6}\right] \approx 0{,}46$$

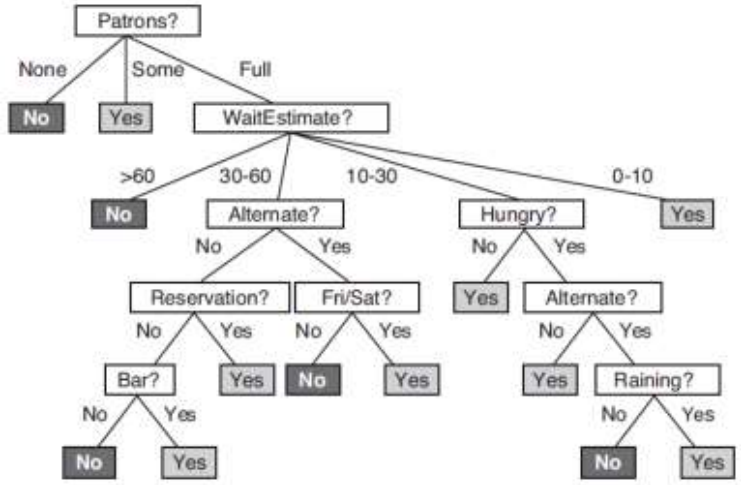
$$Gain(Type) = 1 - 1 = 0$$

$$Gain(Patrons) = 1 - 0{,}46 = 0{,}54$$

Primer

- naučeno odločitveno drevo (levo) je krajše od ročno zgrajenega drevesa (desno)





- obe drevesi sta konsistentni s primeri
- v zgrajenem drevesu ne nastopajo vsi atributi (npr. *Raining* in *Reservation*), zakaj?

Večvrednostni atributi

- težava z atributi, ki imajo več kot dve vrednosti: informacijski prispevek precenjuje njihovo kakovost (entropija je nižja na račun večjega števila vrednosti in ne na račun kakovosti atributa)

- rešitve:
  - normalizacija informacijskega prispevka (**relativni informacijski prispevek,** bolje: **mera razdalje**)
  - **binarizacija** atributov

Relativni informacijski prispevek

information gain ratio (sistem C4.5 Quinlan, 1986)

$$Gain(A) = I - I_{res}(A)$$

$$I(A) = -\sum_{v} p_v \log_2 p_v$$

$$GainRatio(A) = \frac{Gain(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

$\boldsymbol{v}$ $-$ *vrednost atributa,*
$\boldsymbol{c}$ $-$ *razred*

informacija, ki jo potrebujemo za določitev vrednosti atributa A (entropija atributa)

Problem: I(A) = 0 (oziroma zelo majhen)
Rešitev: upoštevaj samo atribute z nadpovprečnim Gain (A)

Binarizacija atributov

- alternativa za reševanje problematike z večvrednostnimi atributi
- zalogo vrednosti atributa lahko razbijemo v dve množici

- primer: atribut $barva \in \{rdeča, rumena, zelena, modra\}$
- strategije:
  - $\{\{rdeča\}, \{rumena, zelena, modra\}\}$ (one-vs-all)
  - $\{\{rdeča, rumena\}, \{zelena, modra\}\}$
  - vpeljava binarnih atributov za vsako barvo
  - itd.

- prednost: manjše vejanje drevesa (statistično bolj zanesljivo, možna višja klasifikacijska točnost)
  - različne binarne verzije atributa lahko nastopajo kot samostojni atributi, ki se v drevesu pojavijo večkrat

Kratkovidnost algoritma TDIDT

- TDIDT je požrešni algoritem, ki "lokalno" izbira najboljši atribut in ne upošteva, kako dobro drugi algoritmi dopolnjujejo izbrani atribut
- prednosti in slabosti zgornjega pristopa?

- kratkovidnost (angl. myopy) izbora atributa
- primer: problem XOR

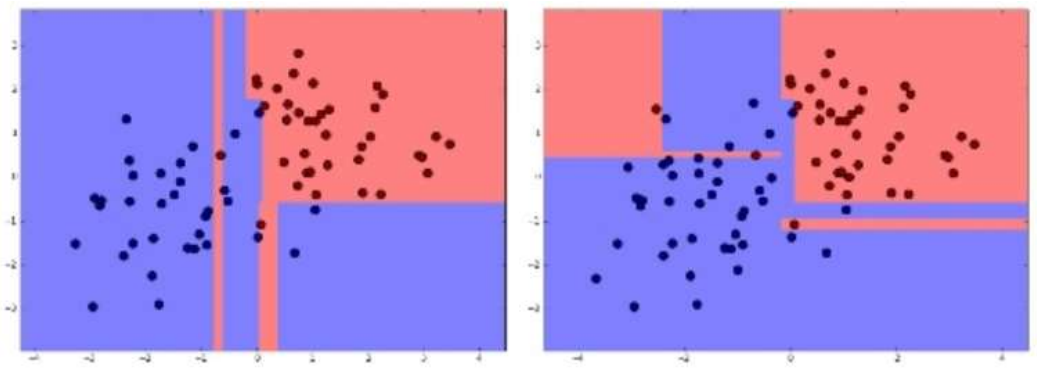
A_1	A_2	Razred
0	0	0
0	1	1
1	0	1
1	1	0

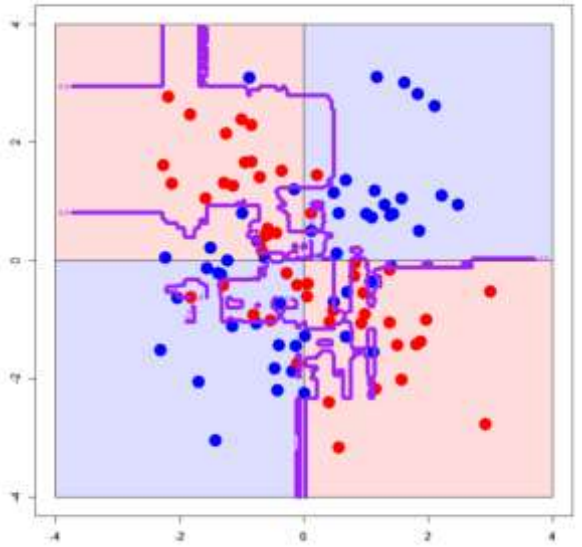
$Gain(A1) = ?$
$Gain(A_2) = ?$

Učenje dreves iz šumnih podatkov

- vir: nepopolni podatki, napake v učnih primerih
- težave:
  - učenje šuma in ne dejanske (skrite) funkcije, ki generira podatke
  - pretirano prilagajanje vodi v velika drevesa
  - slaba razumljivost dreves
  - posledica: nižja klasifikacijska točnost na novih podatkih

- rešitev: rezanje odločitvenega drevesa



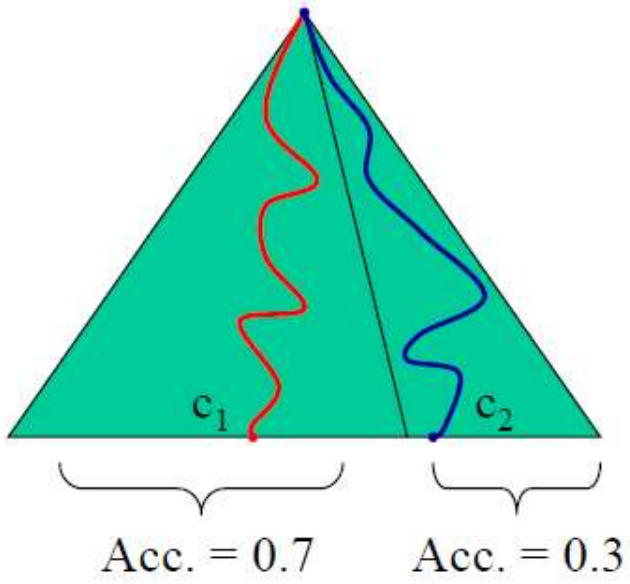


Rezanje odločitvenih dreves

- 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)

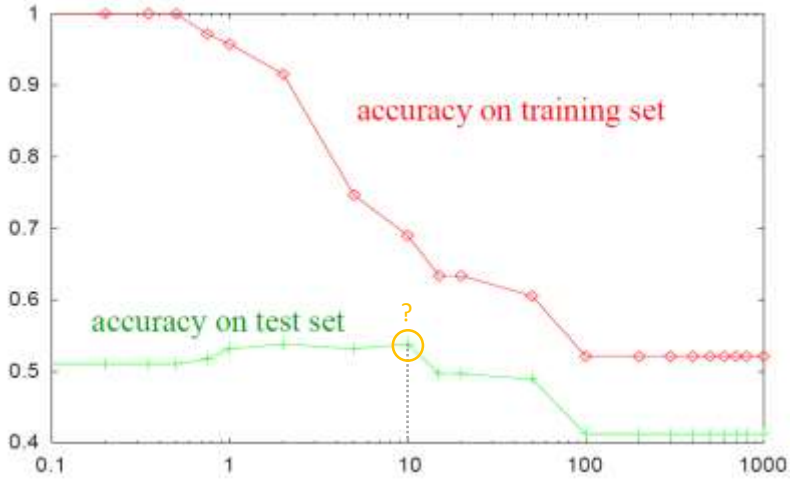
- primer nizke točnosti drevesa pri skrajnem primeru pretiranega prilagajanja:
  - dva razreda, c_1 in c_2, $p(c_1) = 0{,}7$, $p(c_2) = 0{,}3$
  - privzeta točnost (točnost večinskega razreda) = $0{,}7$
  - drevo, zgrajeno do konca (en primer v vsakem listu)
  - pričakovana točnost: $0{,}7 \times 0{,}7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



Rezanje odločitvenih dreves

- cilj: maksimiziraj pričakovano točnost (minimiziraj pričakovano napako) drevesa
- vprašanja:
  - kako to doseči,
  - kje rezati,
  - kombinatorično število možnih porezanih dreves
- primeri:

vpliv rezanja na točnost (B. Zupan, domena glass)

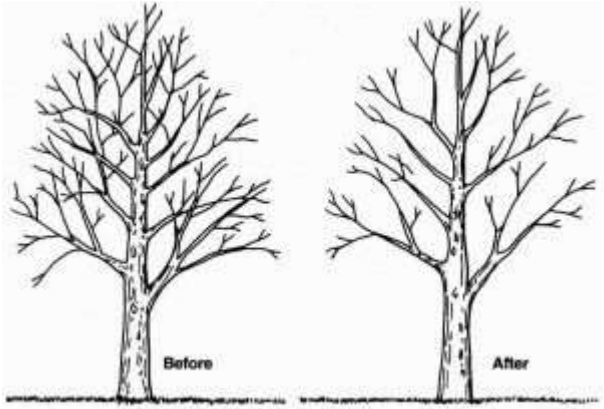


obseg rezanja drevesa

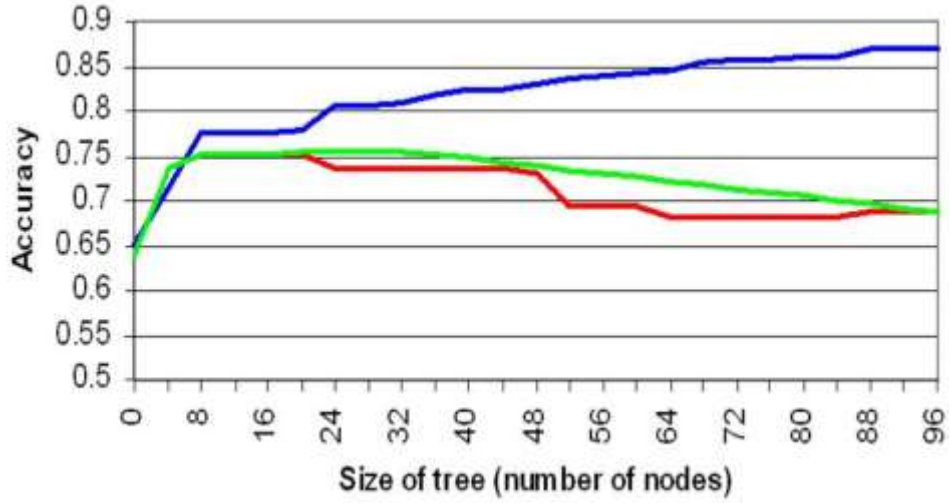
primer iz prakse: lociranje primarnega tumorja (domena *Primary tumor*

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

Rezalna množica



- ocenjevanje točnosti poddrevesa pri rezanju:
  - na učnih podatkih
  - na posebni množici testnih primerov (rezalna množica, validacijska množica) - če imamo dovolj podatkov (ostane manj podatkov za gradnjo)
- tipična delitev podatkov
  - učna množica (70%): od tega množica za gradnjo (growing set) 70% (torej 49%) in rezalna množica (pruning set) 30% (torej 21%)
  - testna množica (30%)



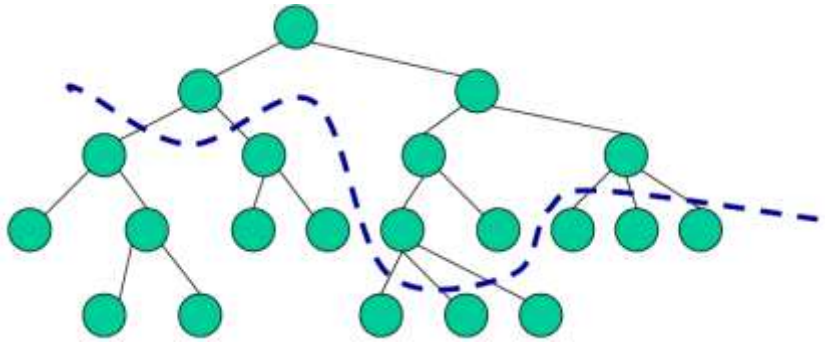
učna množica

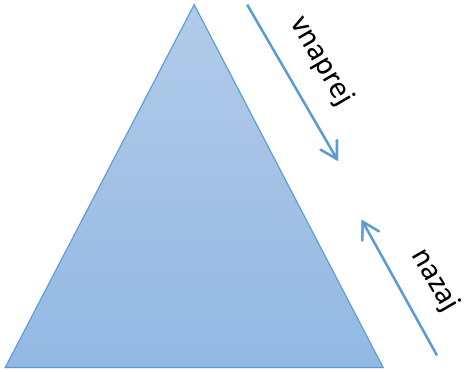
rezalna množica

testna množica

Strategije rezanja

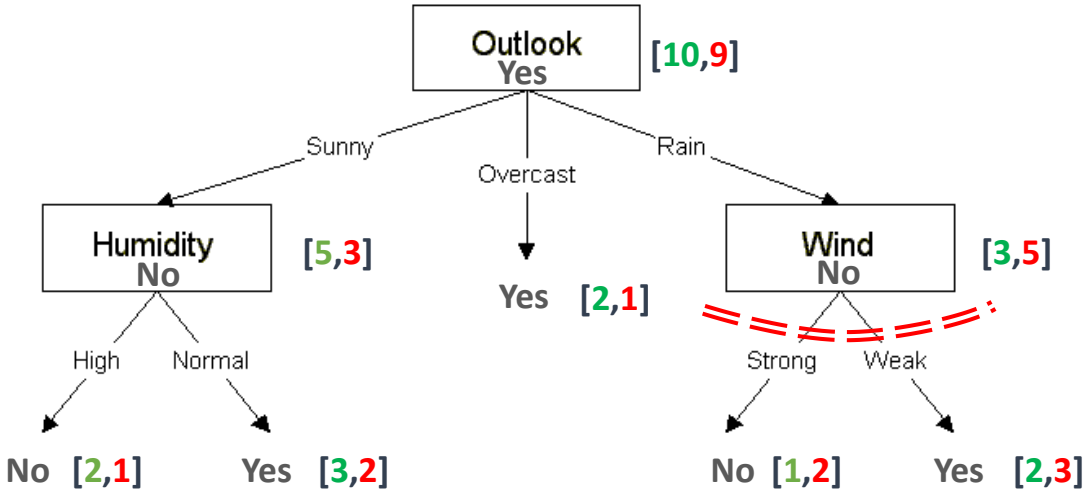
- **rezanje vnaprej** (angl. *forward pruning, pre-stopping, pre-prunning*): uporaba dodatnega kriterija za zaustavitev gradnje grevesa glede na obseg šuma (na podlagi: števila primerov, večinski razred, smiselnost delitve v poddrevesa glede na informacijski prispevek itd.)
  - hitrejše
  - kratkovidno,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

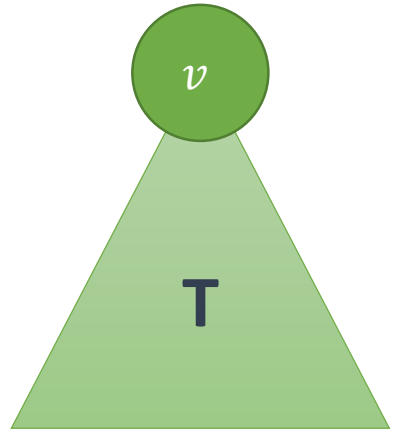




Rezanje z zmanjševanjem napake

- angl. *reduced error pruning* (REP)
- uporablja rezalno množico, potrebna primerna velikost za zanesljivost
- postopek:
  - potuj od listov navzgor (prični s starši listov)
  - za vsako notranje vozlišče izračunaj dobitek rezanja:
    št. napačnih klasifikacij v drevesu T − št. napačnih klasifikacij v vozlišču v
  - če je dobitek pozitiven, obreži in nadaljuj postopek s staršem
    sicer ustavi postopek
- Učna: Yes/No; Rezalna: [Yes,No]

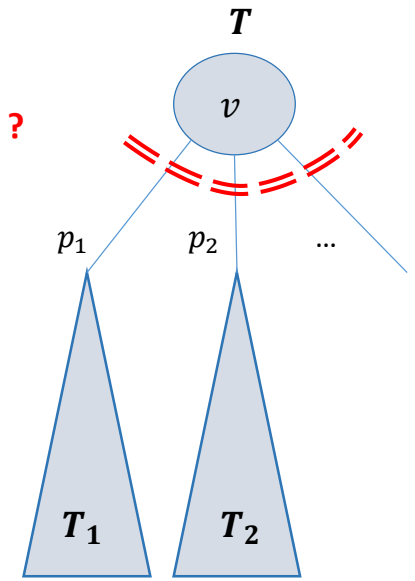




Rezanje z minimizacijo napake

- 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 točnost maksimalna (napaka minimalna)
- za vozlišče v izračunamo:
  - **statično napako** (verjetnost klasifikacije v napačni 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



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
  - smiselno je, da ocena verjetnosti upošteva tudi **apriorno verjetnost** (verjetnost, ki jo poznamo o problemu – npr. 50% za izid meta kovanca)

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**

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

delež upoštevanja apriorne verjetnosti

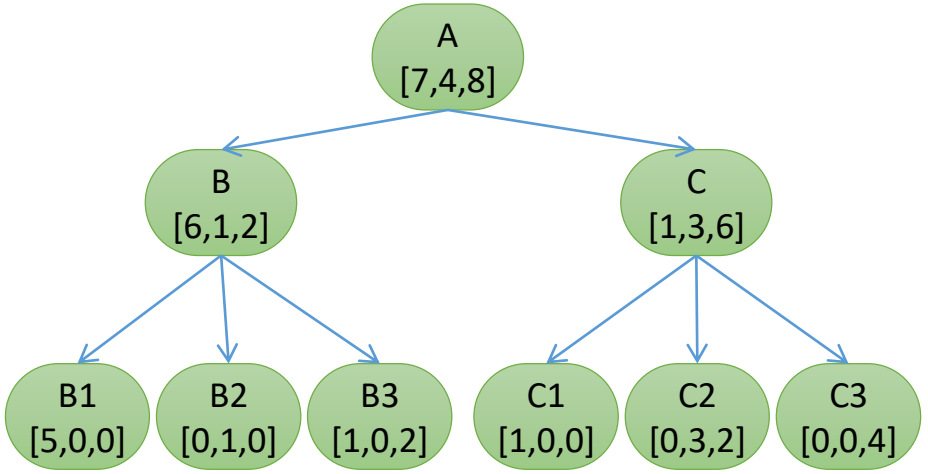
delež upoštevanja relativne frekvence

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

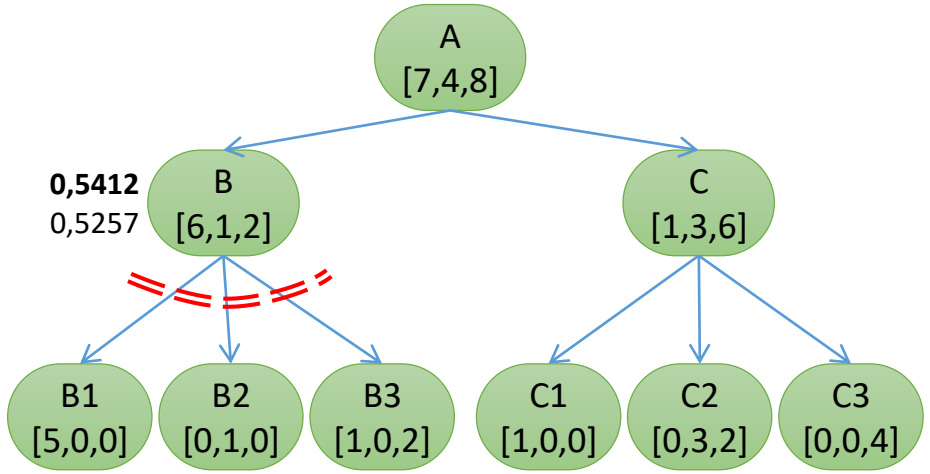
- primer: Bratko: Prolog Programming for AI
- Podano je odločitveno drevo za klasifikacijo v razrede x z naslednjimi apriornimi verjetnostmi razredov: $p_a(x) = 0{,}4$, $p_a(y) = 0{,}3$, $p_a(z) = 0{,}3$. Številke v oglatih oklepajih $[x, y, z]$ predstaljajo frekvence primerov v vozlišču, ki pripadajo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



Vaja

- klasifikacijske točnosti v listih B1, B2 in B3:
- $p(x|B1) = \frac{n+m\cdot p_a(x)}{N+m} = \frac{5+8\cdot 0{,}4}{5+8} = 0{,}6308$
- $p(y|B2) = \frac{1+8\cdot 0{,}3}{1+8} = 0{,}3778$
- $p(z|B3) = \frac{2+8\cdot 0{,}3}{3+8} = 0{,}4$
- 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:
  $p(x|B) = \frac{6+8\cdot 0{,}4}{9+8} = 0{,}5412$
- statična točnost je večja od vzvratne točnosti → porežemo

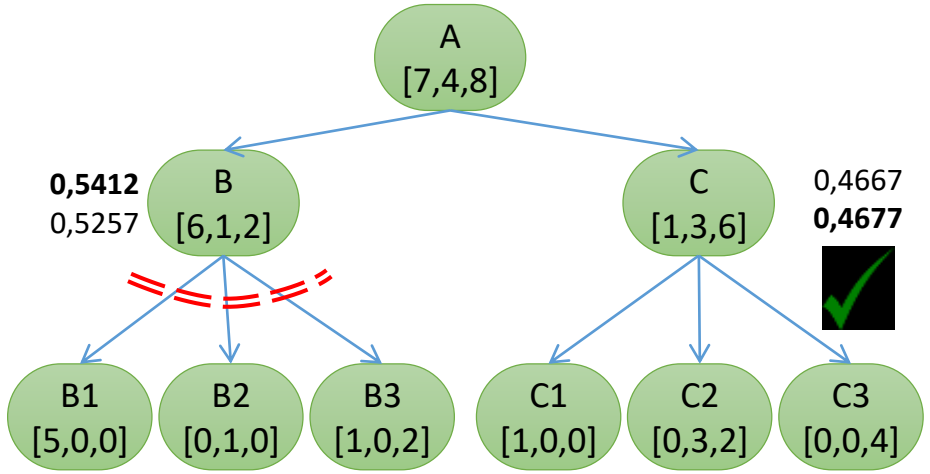
- nadaljujemo z vozliščema C in A ...



Vaja

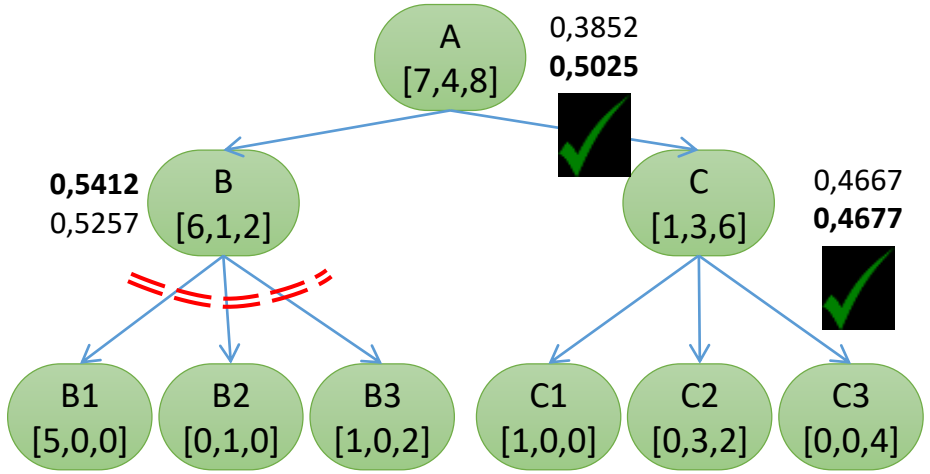
- klasifikacijske točnosti v listih C1, C2 in C3:
- $p(x|C1) = \frac{n+m\cdot p_a(x)}{N+m} = \frac{1+8\cdot 0{,}4}{1+8} = 0{,}4667$
- $p(y|C2) = \frac{3+8\cdot 0{,}3}{5+8} = 0{,}4154$
- $p(z|C3) = \frac{4+8\cdot 0{,}3}{4+8} = 0{,}5333$
- vzvratna točnost v vozlišču C: $\frac{1}{10}\cdot 0{,}4667 + \frac{5}{10}\cdot 0{,}4154 + \frac{4}{10}\cdot 0{,}5444 = 0{,}4677$
- statična točnost v vozlišču C:
  $p(z|C) = \frac{6+8\cdot 0{,}3}{10+8} = 0{,}4667$
- vzvratna točnost je večja od statične točnosti → ne porežemo

- nadaljujemo z vozliščem A ...



Vaja

- klasifikacijske točnosti v podrevesih s koreni v B in C:
- $E(B) = \min(e(B), \sum_i p_i E(B_i)) = 0{,}5412$
- $E(C) = \min(e(C), \sum_i p_i E(C_i)) = 0{,}4677$
- vzvratna točnost v vozlišču A: $\frac{9}{19} \cdot 0{,}5412 + \frac{10}{19} \cdot 0{,}4677 = 0{,}5025$
- statična točnost v vozlišču A:
  $p(z|A) = \frac{8+8 \cdot 0{,}3}{19+8} = 0{,}3852$
- vzvratna točnost je večja od statične točnosti → ne porežemo



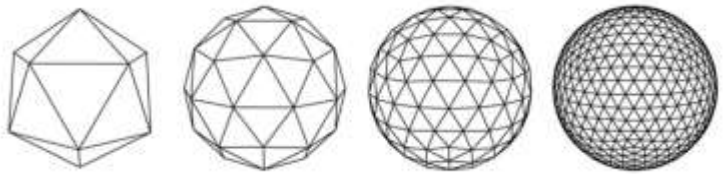
Obravnava atributov

potrebno je nasloviti še naslednja problema:

- **manjkajoči podatki** v atributih:
  - ignorirati cele primere z neznanimi vrednostmi?
  - uporabiti vrednost NA/UNKNOWN?
  - nadomestiti manjkajočo vrednost (povprečna, najbolj pogosta, naključna, napovedana)
  - primer obravnavamo verjetnostno glede na vse možne vrednosti atributa (s tako utežjo lahko sodeluje pri gradnji modela in klasifikaciji)

	Variables			
Respondent	A	B	C	D
1	1	2	3	4
2	1	2	3	4
3	4	3	2	1
4	4	3	2	1
5	1	2		1
6		2	2	1
7	1	2	2	
8	1		2	1

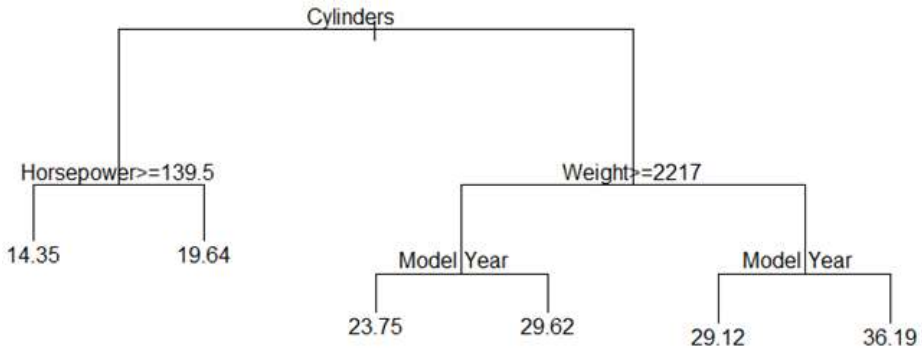
- **obravnava numeričnih atributov**: običajno izvedemo diskretizacijo v dva (binarizacija) ali več diskretnih intervalov
  - intervali z enako frekvenco primerov (equal-frequency)
  - intervali enake širine (equal-width)
  - intervali, ki maksimizirajo informacijski dobite



Regresijska drevesa

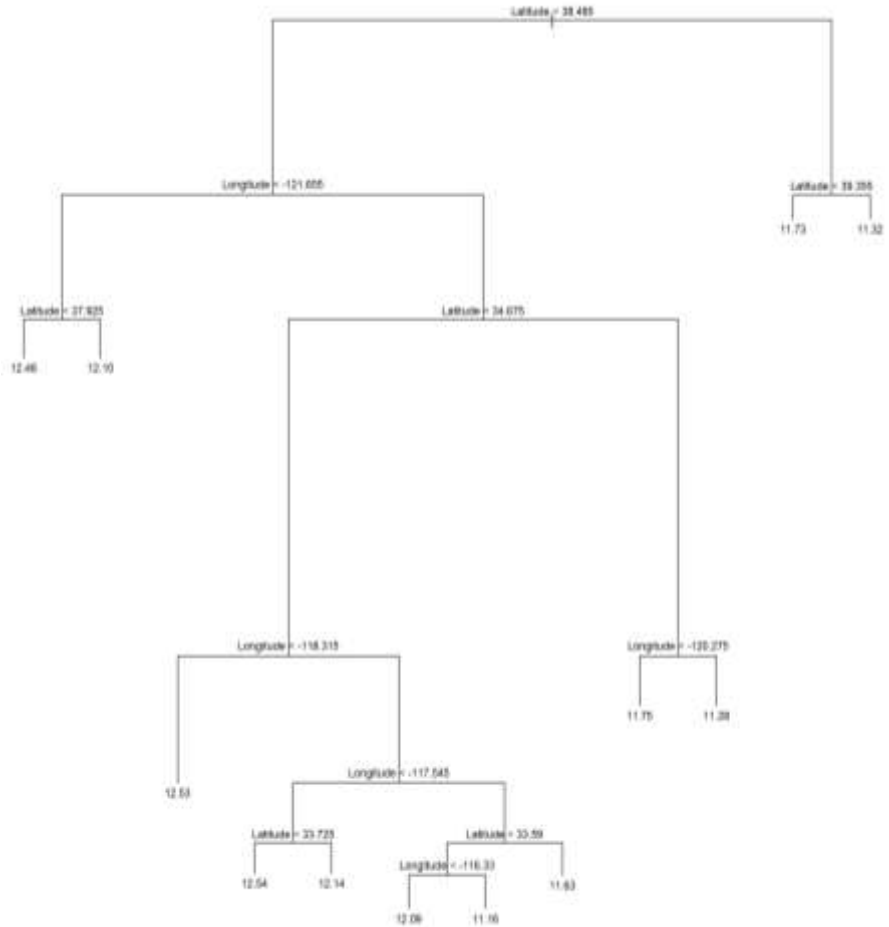
- zvezna ciljna spremenljivka – regresijski problem
- regresijska drevesa so podobna odločitvenim drevesom, le za regresijske probleme
- sistemi: CART (Breiman et al. 1984), RETIS (Karalič 1992), M5 (Quinlan 1993), WEKA (Witten and Frank, 2000)

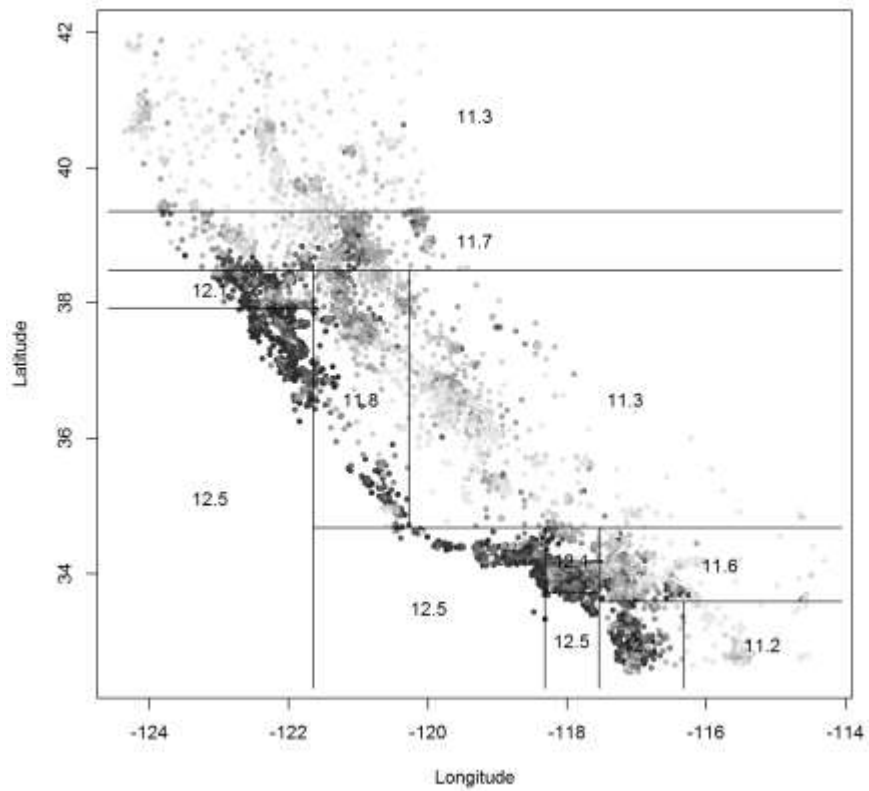
- listi v regresijskem drevesu predstavljajo:
  - povprečno vrednost označb ("razreda") primerov v listu
  - preprost napovedni model (npr. linearna regresija) za nove primere



Bora Beran, On Anything Data, domena MPG

Regresijska drevesa





João Neto, Classification & Regression Trees, http://www.di.fc.ul.pt/~jpn/r/tree/tree.html

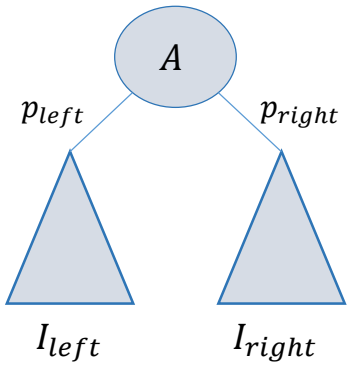
Gradnja regresijskih dreves

- 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

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



Rezanje regresijskega drevesa

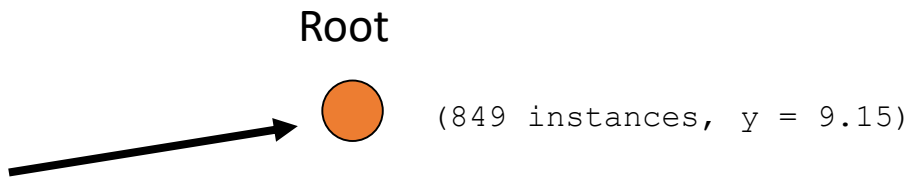
uporabimo prirejeno m-oceno:

$$e = \frac{n}{N+m} e_v + \frac{m}{N+m} e_k$$

n – število primerov v vozlišču,
N – število vseh primerov
e_v – povprečna napaka modela na teh primerih, če list
e_k – povprečna napaka tega istega modela na vseh učnih primerih.

Growing a Regression Tree

Leaves: 1, MSE = 41.46

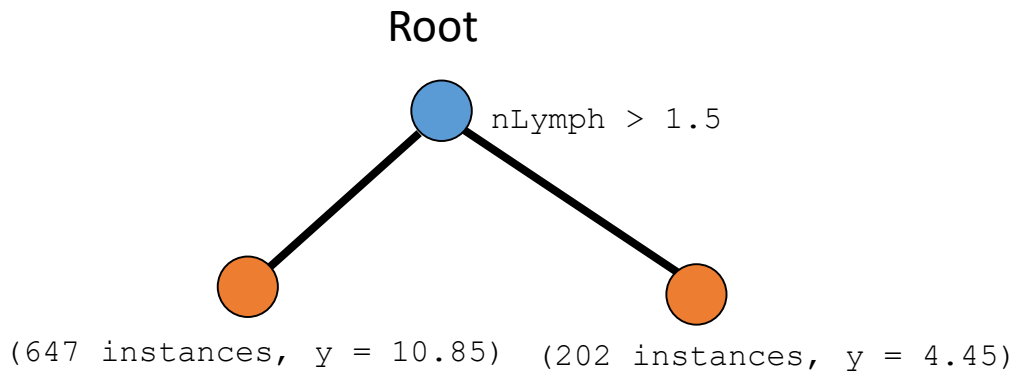


The most simple tree.

Growing a Regression Tree

Leaves: 1, MSE = 41.46

Leaves: 2, MSE = 36.32

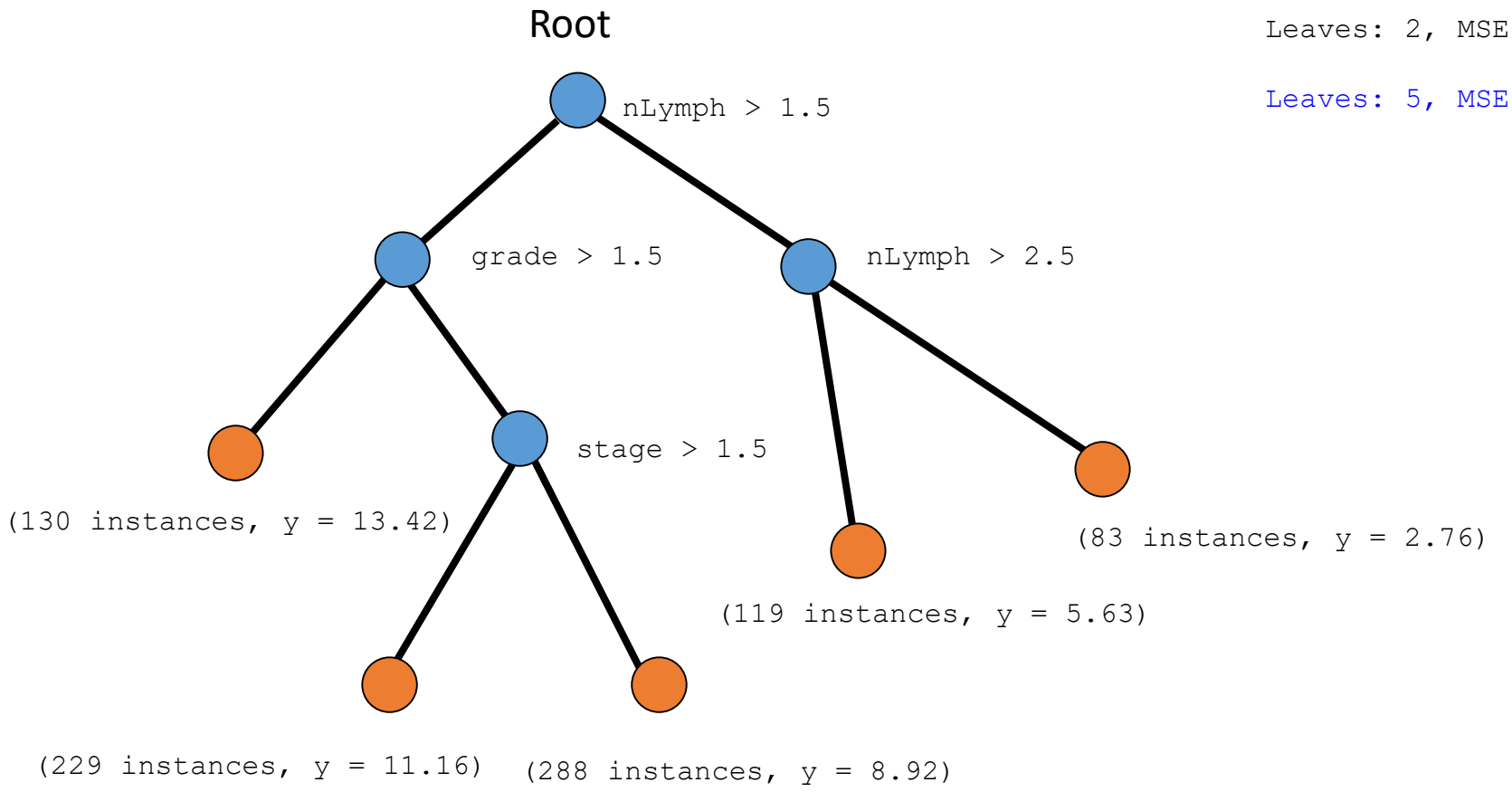


Growing a Regression Tree

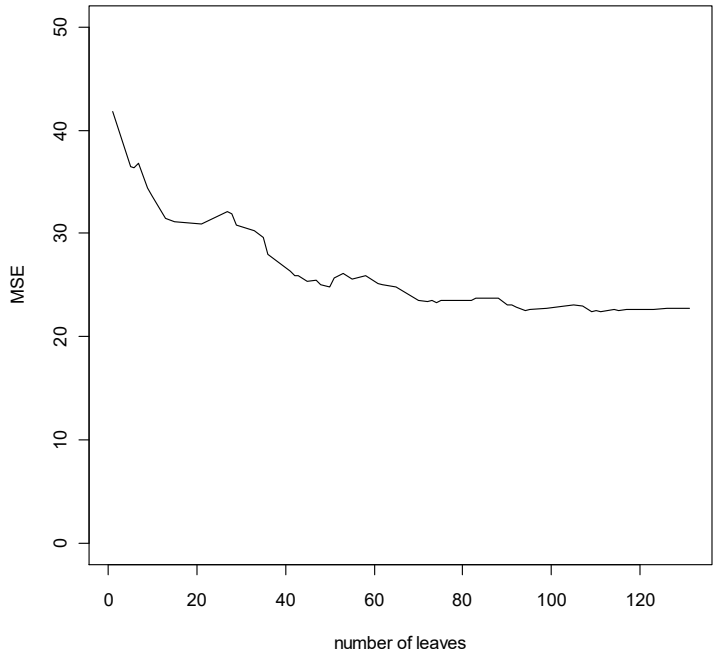
Leaves: 1, MSE = 41.46

Leaves: 2, MSE = 36.32

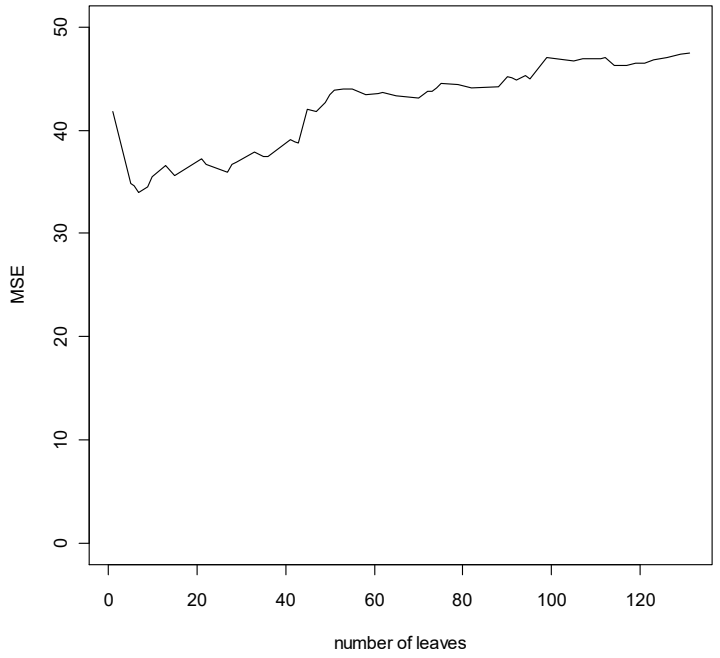
Leaves: 5, MSE = 34.81



Overfitting a Regression Tree



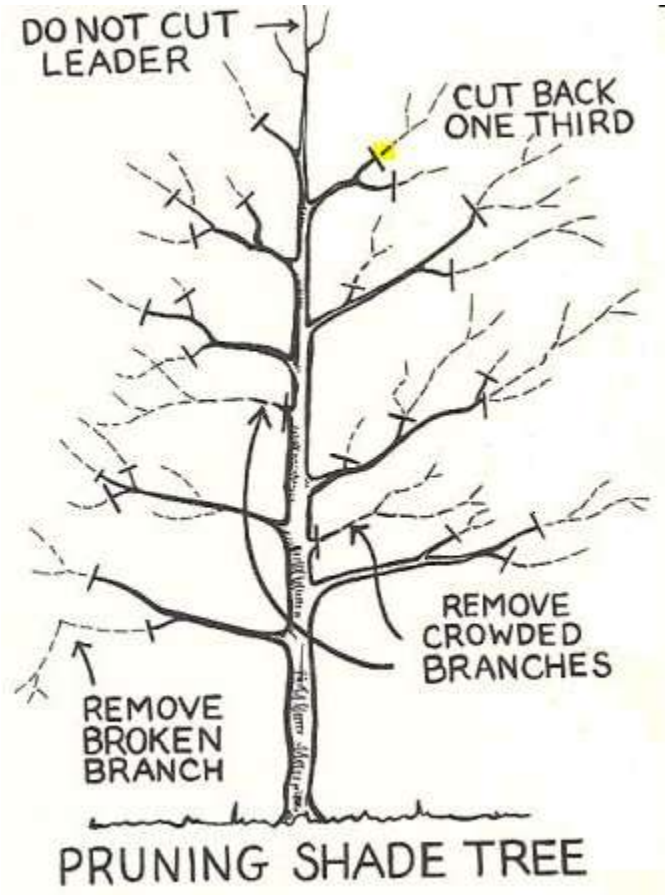
Results on the training data set



Results on the test data set

Further increasing the size of the tree may result in overfitting and a higher error.

Decision Tree Pruning



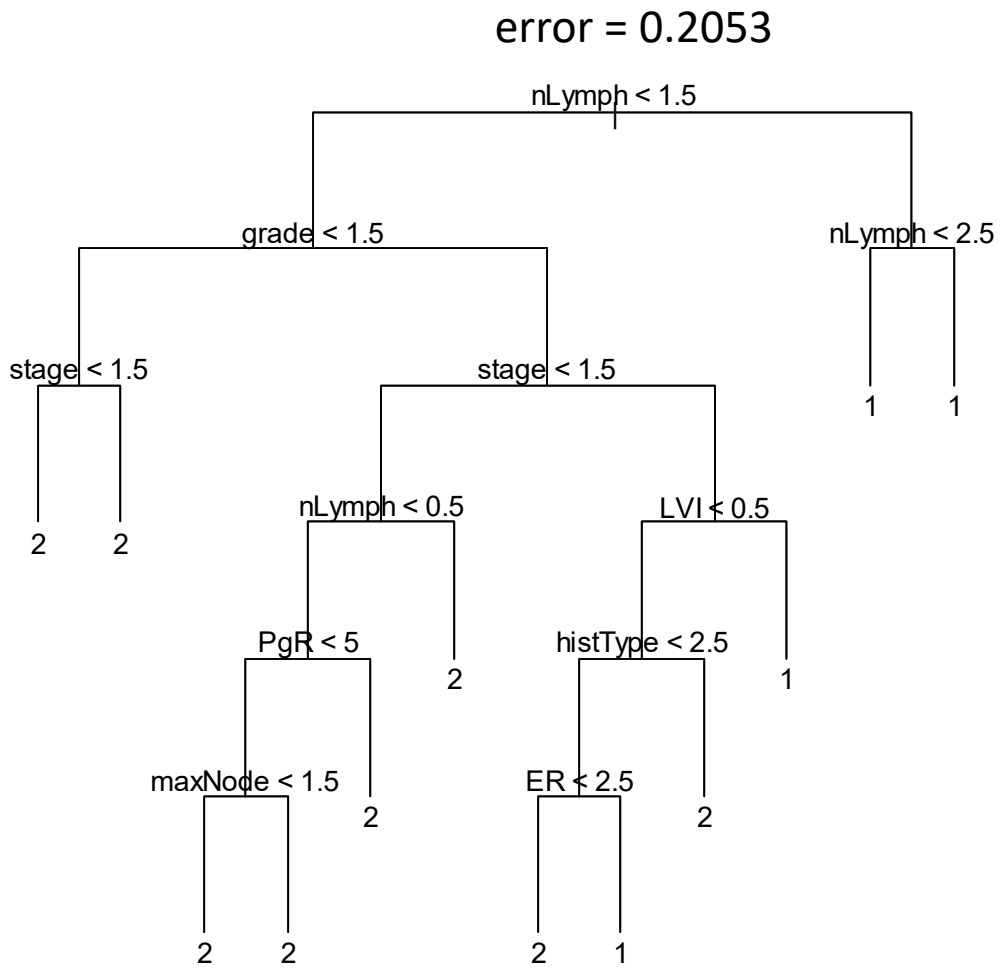
An Unpruned Classification Tree

Oncology Training Dataset

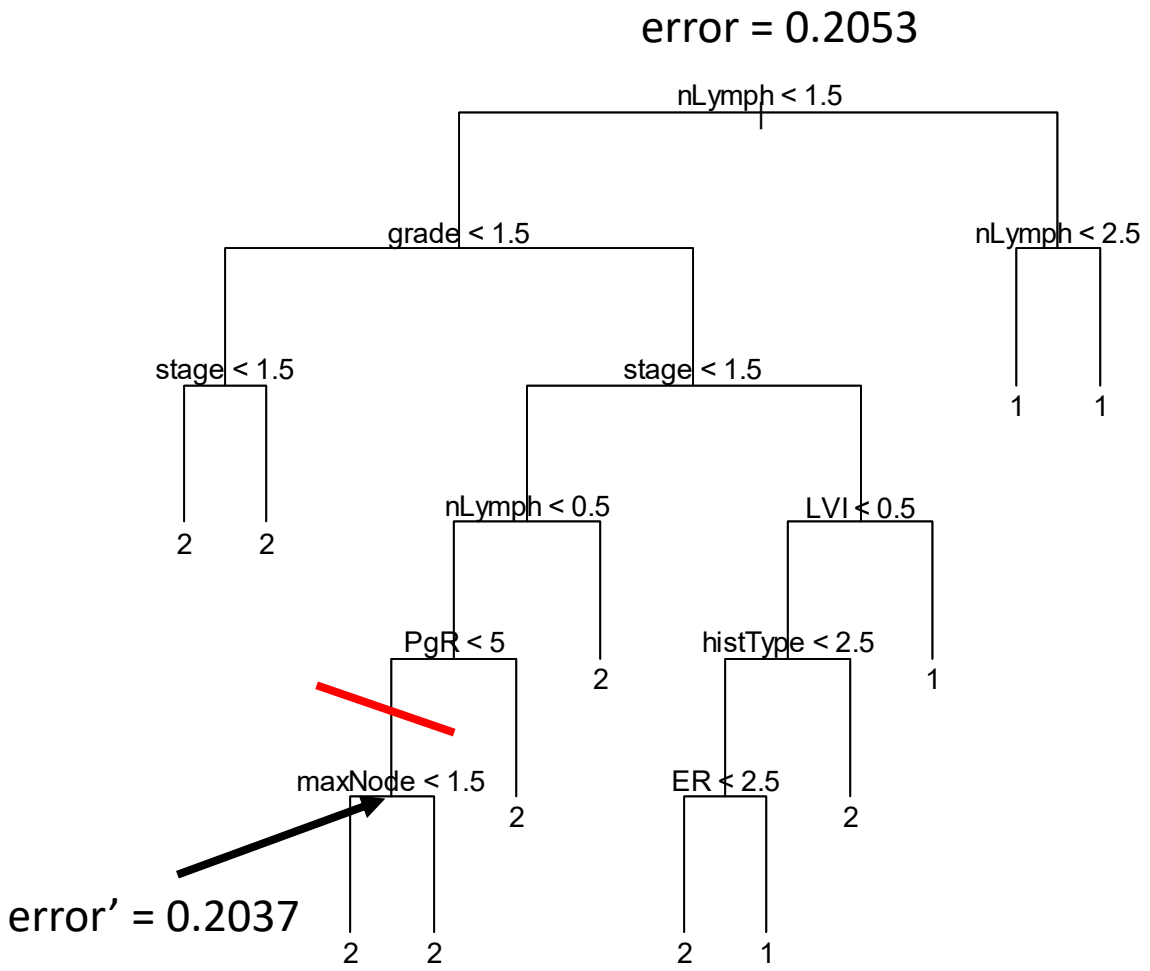
+

Independent Pruning Dataset

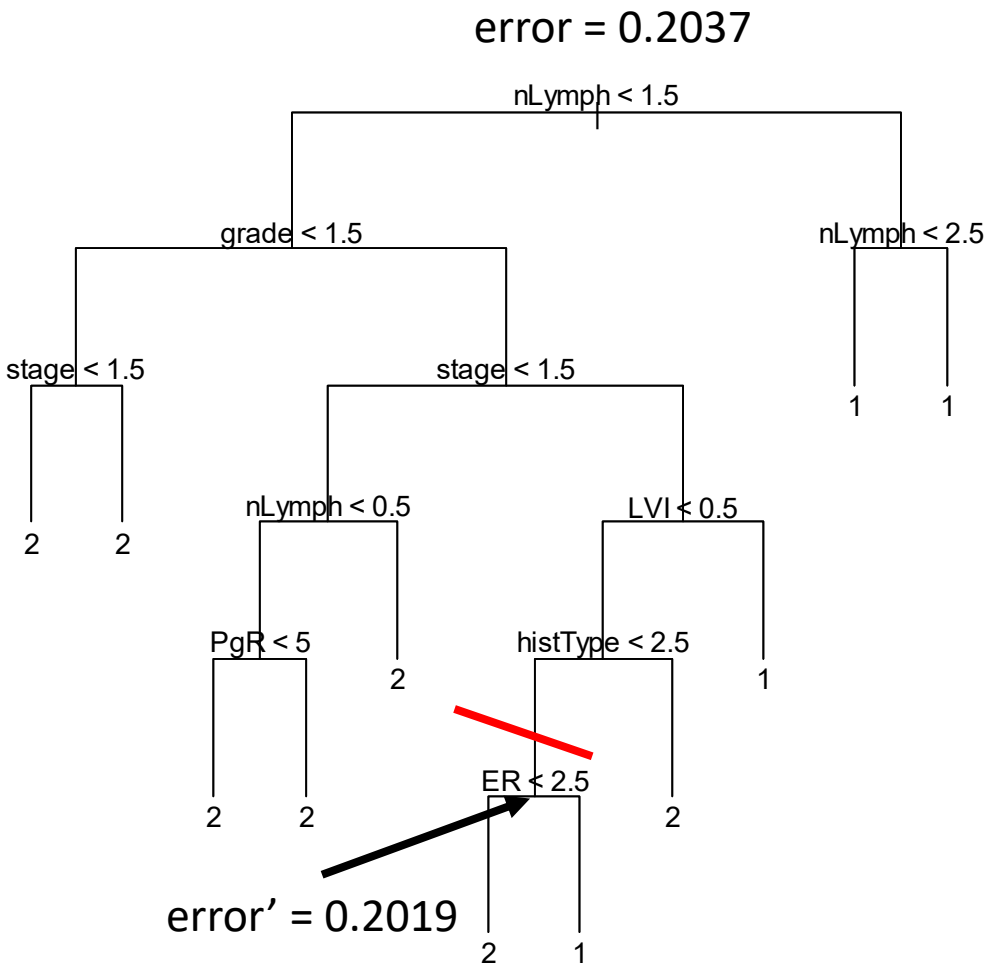
We use the Brier score to measure the error.



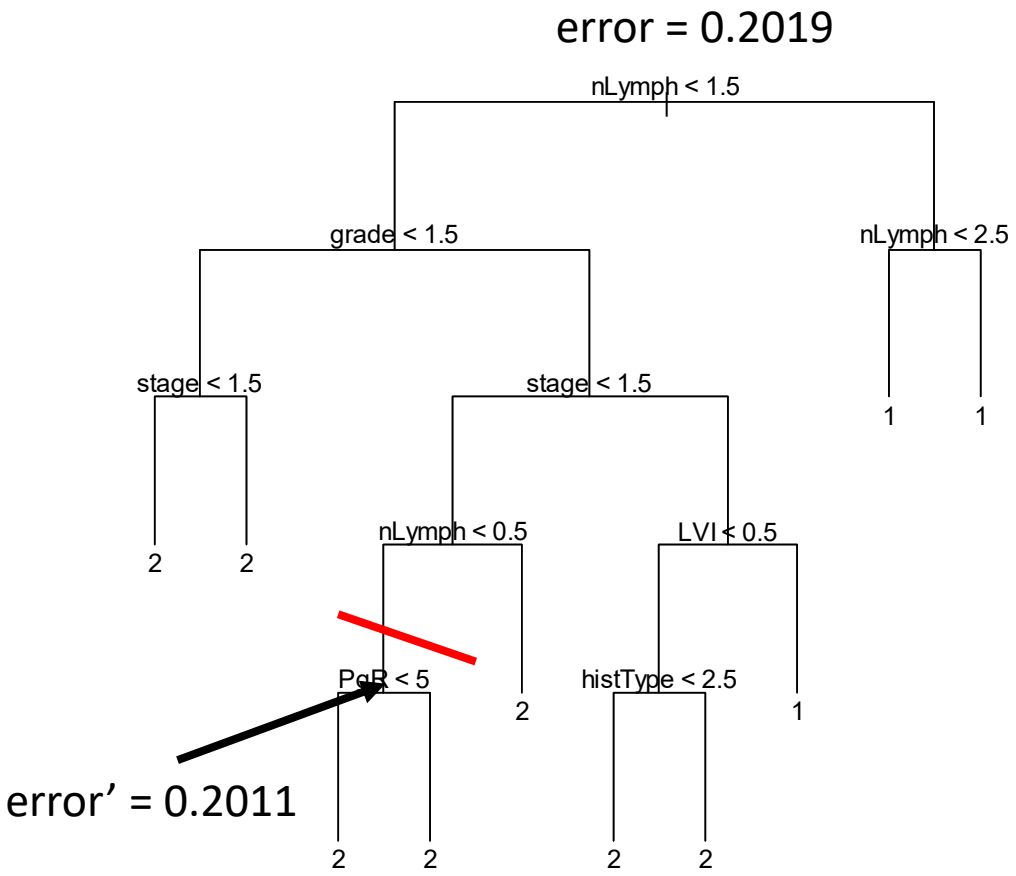
Snip,...


error = 0.2053
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
stage < 1.5
1
1
2
2
nLymph < 0.5
LVI < 0.5
PgR < 5
2
histType < 2.5
1
maxNode < 1.5
2
ER < 2.5
2
error’ = 0.2037
2
2
2
1

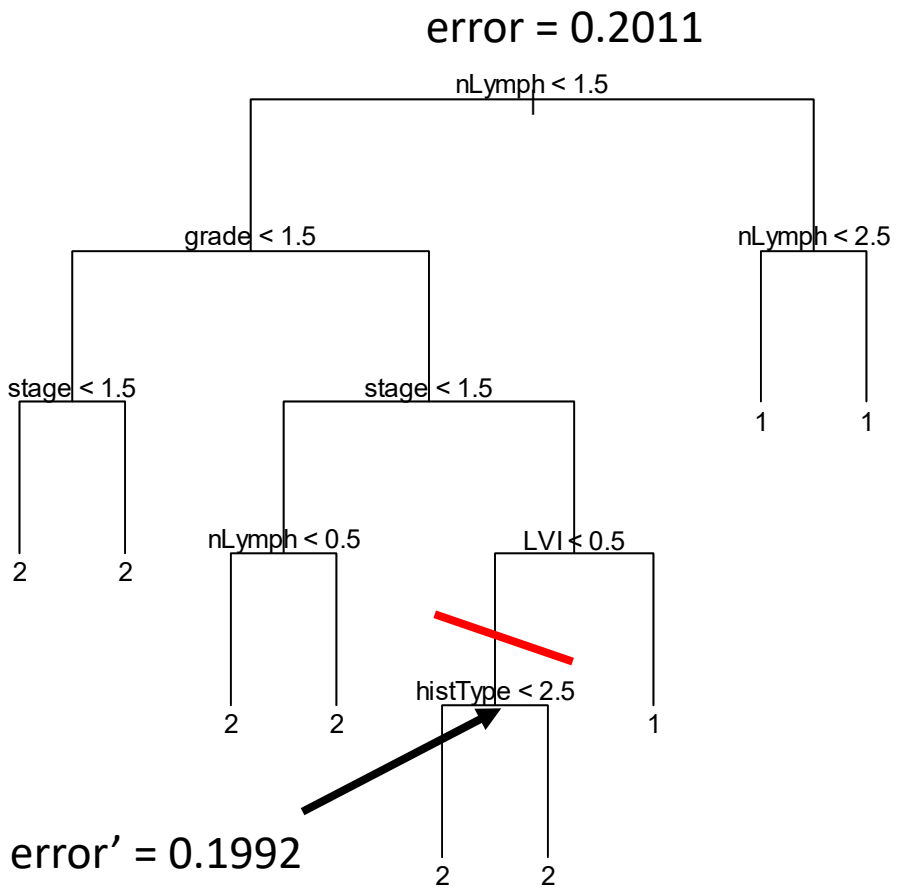
Snip,...


error = 0.2037
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
stage < 1.5
1
1
2
2
nLymph < 0.5
LVI < 0.5
PgR < 5
2
histType < 2.5
1
ER < 2.5
2
2
2
error’ = 0.2019
2
1

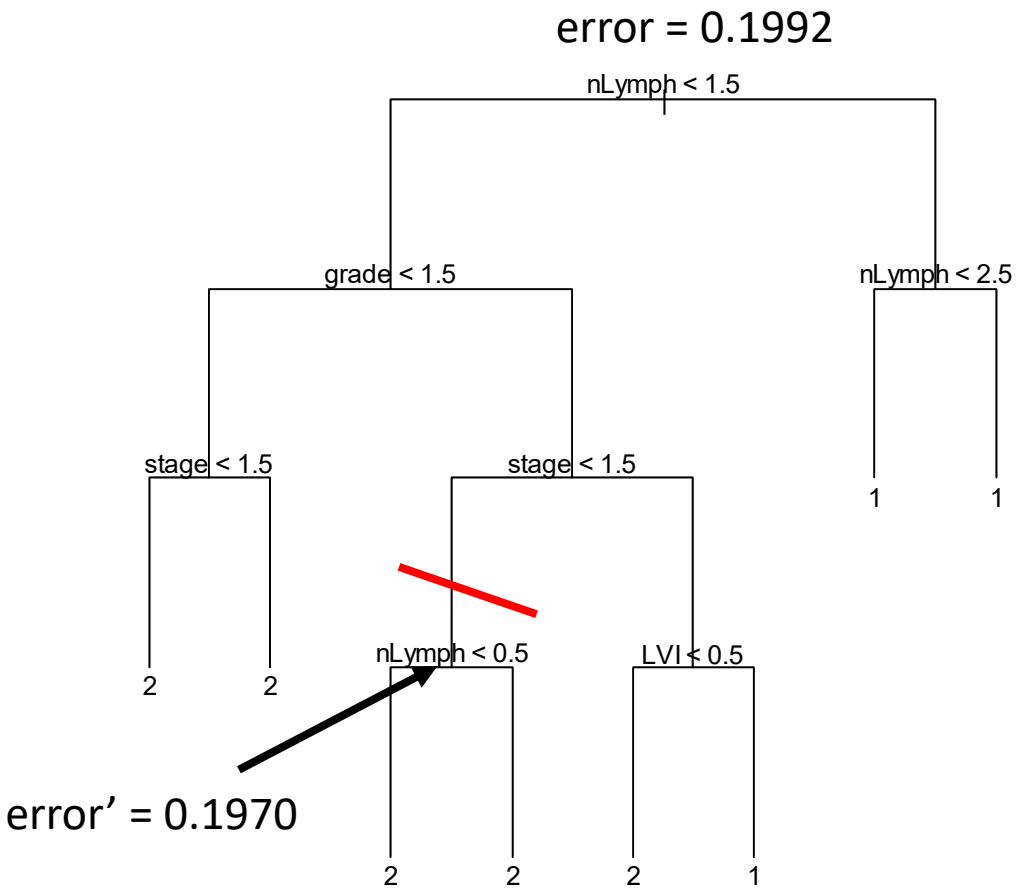
Snip,...


error = 0.2019
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
stage < 1.5
1
1
2
2
nLymph < 0.5
LVI < 0.5
PgR < 5
2
histType < 2.5
1
error’ = 0.2011
2
2
2
2

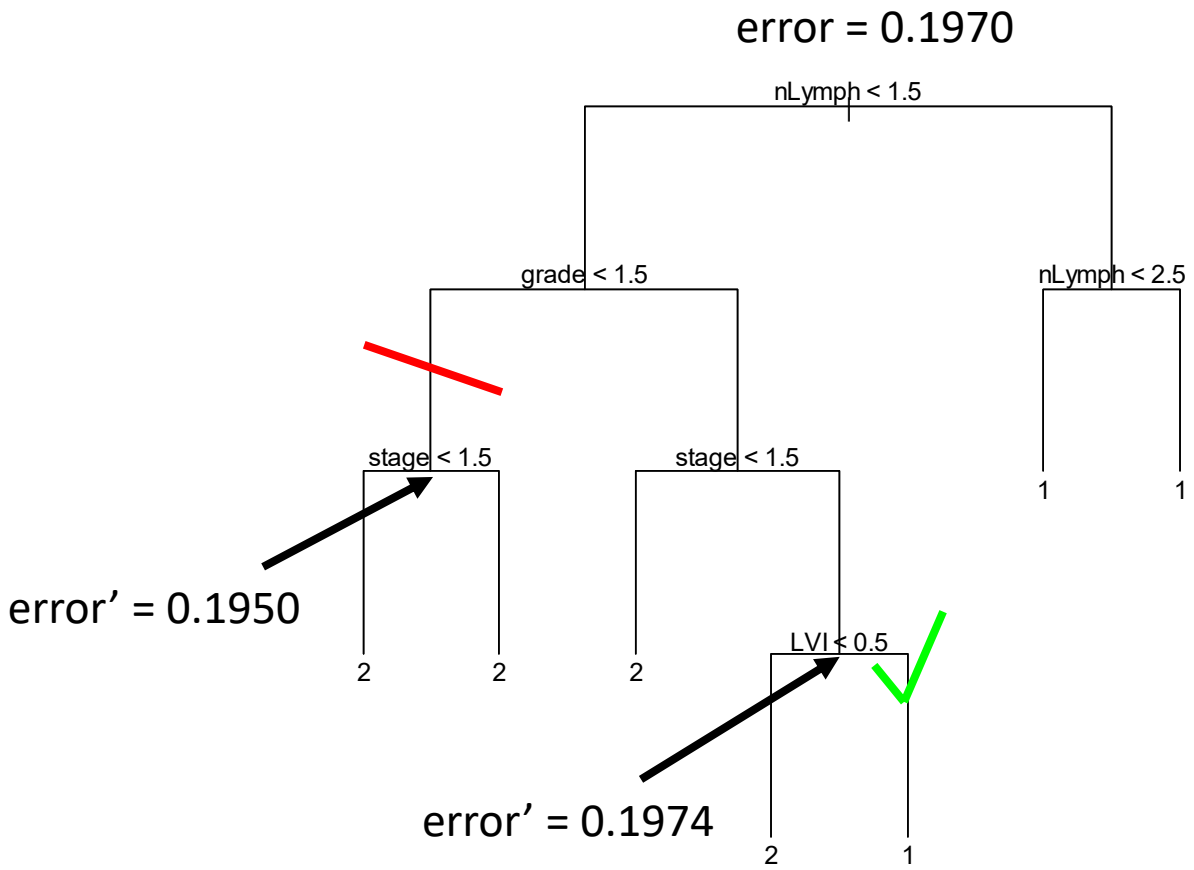
Snip,...


error = 0.2011
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
stage < 1.5
1
1
2
2
nLymph < 0.5
LVI < 0.5
2
2
histType < 2.5
1
error’ = 0.1992
2
2

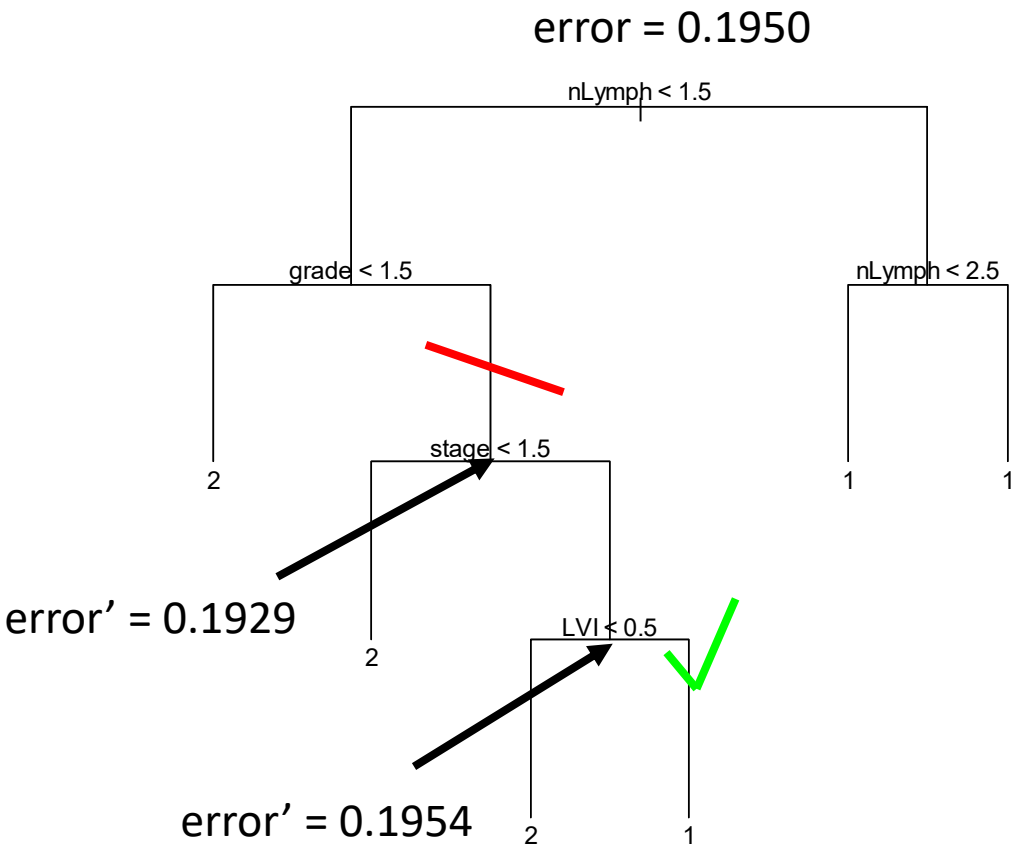
Snip,...


error = 0.1992
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
stage < 1.5
1
1
2
2
nLymph < 0.5
LVI < 0.5
error’ = 0.1970
2
2
2
1

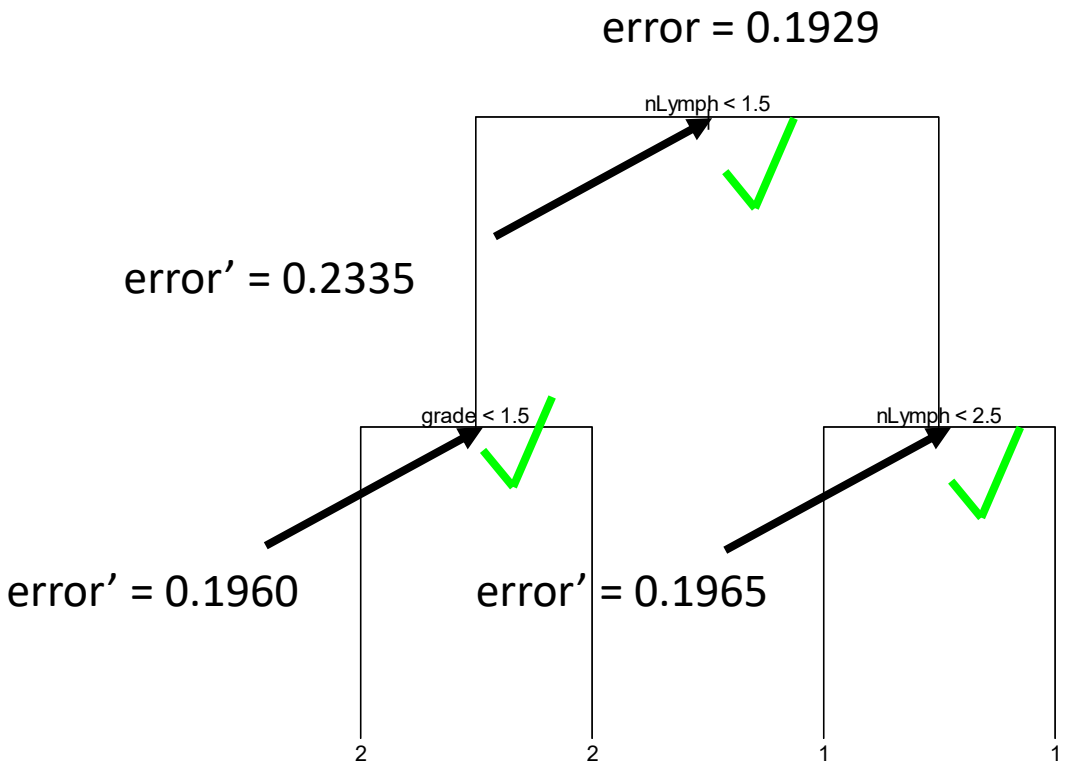
Snip,...


error = 0.1970
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
stage < 1.5
1
1
error’ = 0.1950
2
2
2
LVI < 0.5
error’ = 0.1974
2
1

Snip,...


error = 0.1950
nLymph < 1.5
grade < 1.5
nLymph < 2.5
stage < 1.5
2
1
1
2
error’ = 0.1929
LVI < 0.5
2
1
error’ = 0.1954

Pruned Tree



None of the remaining nodes justifies pruning.