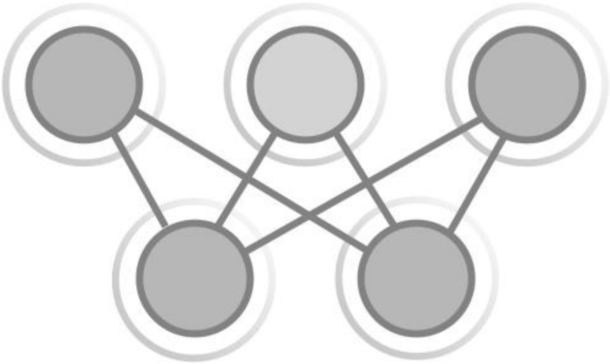
Prof.dr.sc. Bojana Dalbelo Bašić

Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

www.zemris.fer.hr/~bojana
bojana.dalbelo@fer.hr

Stabla odluke



FER

studeni 2010.

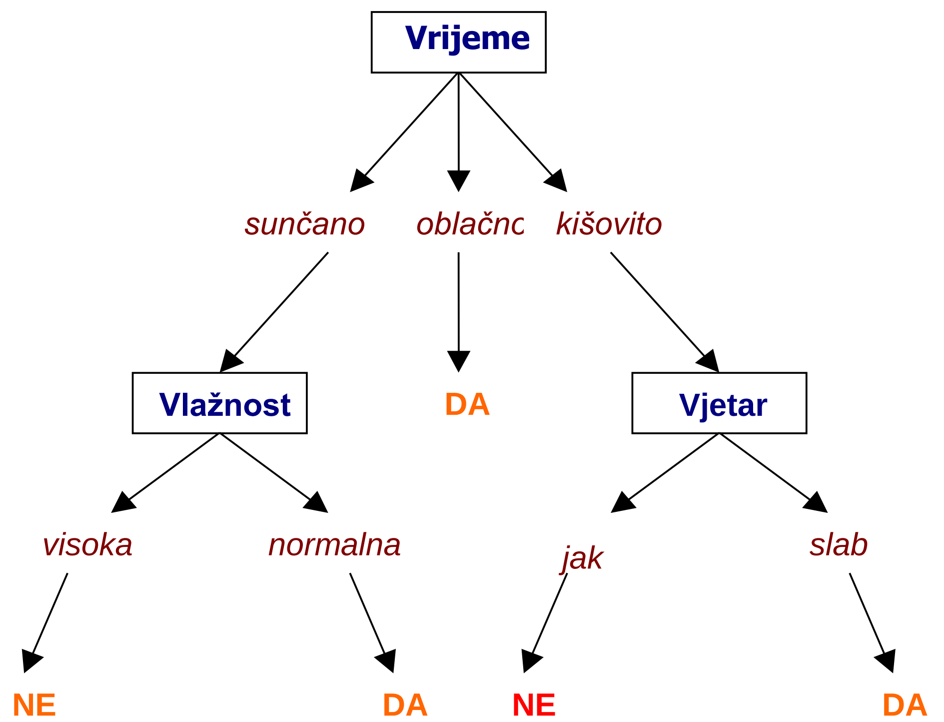
STABLA ODLUKE

- Najčešće korištena metoda induktivnog zaključivanja (medicina, financije..)
- **Neparametarska metoda**
- **Diskriminativna metoda**
- Metoda aproksimiranja funkcije diskretnih (i kontinuiranih vrijednosti ->CART) robusna na šum, koja <u>može učiti disjunktivne koncepte</u>.
- Familija algoritama: **ID3**, ASSISTANT, C4.5,
- Pretražuju potpun prostor hipoteza (pristranost?)
- Stabla odluke $\rightarrow$ (reinterpretacija) $\rightarrow$ skup ako-onda pravila

PREDSTAVLJANJE STABLA ODLUKE

- Klasifikacija primjera odozgo, od korijena prema listovima
  - Čvor (*engl. node*) – test atributa
  - Grana (*engl. branch*) – odgovara vrijednosti atributa

Primjer: **Klasifikacija DA/NE** - Je li subotnje jutro pogodno za tenis?



PREDSTAVLJANJE STABLA ODLUKE

Primjer:

- (**Vrijeme** = *sunčano*, **Temperatura** = *vruće*, **Vlažnost** = *visoka*, **Vjetar** = *jak*) $\rightarrow$ (**Igranje_tenisa** = **NE**)

- Općenito, stabla odluke predstavljaju <u>disjunkciju konjunkcije uvjeta na vrijednosti atributa</u>:

  (**Vrijeme** = *sunčano* $\wedge$ **Vlažnost** = *normalna*)

  $\vee$ (**Vrijeme** = *oblačno*)

  $\vee$ (**Vrijeme** = kišovito $\wedge$ **Vjetar**= *jak*)

(DNF – disjunktivna normalna forma)

PROBLEMI POGODNI ZA OBLIKOVANJE STABLIMA ODLUKE

Prikladni za probleme kod kojih…

- Primjeri su predstavljeni parovima atribut – vrijednost (posebno: mali broj mogućih vrijednosti atributa)
- Ciljna funkcija poprima diskretne vrijednosti (u gornjem primjeru Booleova klasifikacija: DA i NE). Algoritam se može proširiti i na učenje funkcije s više vrijednosti ili s realnim vrijednostima
- Stabla odluke prirodno predstavljaju disjunktivni izraz
- Podaci za učenje mogu sadržavati pogreške
- Tolerantnost na nedostajuće vrijednosti

OSNOVNI ALGORITAM UČENJA STABLA ODLUKE

- Quinlan, J.R. (1986) Induction of Decision Trees. *Machine Learning*, 1(1), 81-106

- Temeljni algoritam **Quinlan** je nazvao ID3, a proširenje C4.5 (Quinlan, 1993).

- **ID3** (engl. ***I****nduction of* ***D****ecision* ***T****rees*)

OSNOVNI ALGORITAM UČENJA STABLA ODLUKE

Koji atribut odabrati za testiranje?

- testira se svaki atribut da se ocijeni kako dobro klasificira primjere
- najbolji se odabire kao čvor, a njegove vrijednosti su silazne grane
- primjeri za učenje sortiraju se prema odgovarajućem silaznom čvoru (niz onu granu koja odgovara vrijednosti tog atributa)
- cijeli postupak se ponavlja koristeći primjere koji su dodijeljeni silaznome čvoru

- ID3 spada u **pohlepne algoritme** (*engl. greedy*) zato jer se nikad ne vraća zbog ponovnog razmatranja prethodnih čvorova

KOJI ATRIBUT JE NAJBOLJI KLASIFIKATOR?

Najvažniji izbor:

- **Odabir atributa koji će se testirati u pojedinom čvoru stabla**
- Koja je dobra kvantitativna mjera vrijednosti nekog atributa?
- **Informacijska dobit** (*engl. information gain*) – mjera kako dobro pojedini atribut odjeljuje primjere za učenje u skladu s ciljnom klasifikacijom

ENTROPIJA MJERI HOMOGENOST PRIMJERA

- Neka skup D sadrži pozitivne i negativne primjere nekog ciljnog koncepta. **Entropija** u odnosu na skup D jest:

$$\text{Entropija}(D) \equiv - p_{+} \log_2 p_{+} - p_{-} \log_2 p_{-}$$

gdje je:

- **p+** proporcija pozitivnih primjera u D,
- **p-** proporcija negativnih primjera u D.
- Po definiciji: $0\log_2 0 \equiv 0$

ENTROPIJA MJERI HOMOGENOST PRIMJERA

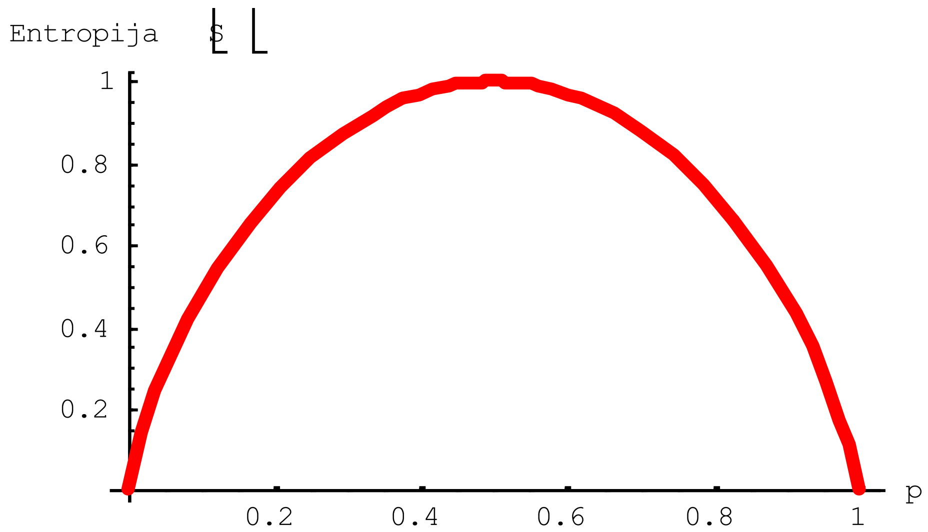
Primjer:

- D se sastoji od 14 primjera: 9 pozitivnih i 5 negativnih primjera.
- Usvojena notacija **[9+, 5-]**
- **Entropija**([9+, 5-]) = $-(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$ = **0.940**

1. Ako svi primjeri pripadaju istoj klasi, kolika je entropija?
2. Kolika je entropija za skup D koji sadrži isti broj pozitivnih i negativnih primjera?

ENTROPIJA MJERI HOMOGENOST PRIMJERA

- Interpretacija entropije: *minimalan broj bitova potreban za kodiranje klasifikacije proizvoljnih članova skupa D*



- U slučaju *K* klasa:

$$\text{Entropija(D)} = \sum_{j=1}^{K} - P(C_j) \log_2 P(C_j)$$

Entropija mjeri stupanj «neurednosti» podataka

INFORMACIJSKA DOBIT MJERI OČEKIVANU REDUKCIJU U ENTROPIJI

Informacijska dobit je očekivana redukcija entropije uzrokovana podjelom primjera za učenje u skladu s tim atributom

- **Informacijska dobit** (*engl. gain*) atributa A u odnosu na skup primjera D jest:

$$Informacijska\_dobit(D,A) \equiv Entropija(D) - \sum_{v \in Vrijednost(A)} \frac{|Dv|}{|D|} Entropija(Dv)$$

Entropija izvornog skupa D

Očekivana vrijednost entropije nakon podjele D na temelju atributa A

- Vrijednost(A) - skup svih mogućih vrijednosti atributa A
- Dv - podskup od D za koji atribut A ima vrijednost v, tj.

  Dv = {x $\in$ D| A(x)= v}

INFORMACIJSKA DOBIT MJERI OČEKIVANU REDUKCIJU U ENTROPIJI

- Informacijska dobit IG(D, A) je informacija o vrijednosti ciljne funkcije, ako je dana vrijednost atributa A
- Vrijednost IG(D, A) je ušteđen broj bitova sačuvan kod kodiranja ciljne funkcije proizvoljnog člana iz skupa primjera D, ako je poznata vrijednost atributa A

Primjer:

- Neka je D skup primjera opisan atributom ***Vjetar*** *={jak, slab}* i neka D ima 14 primjera , 9+ i 5-.
- Od tih 14 primjera,
  - ukupno 8 primjera (6 pozitivnih i 2 negativna) imaju vrijednost ***Vjetar*** *= slab*
  - ostatak 6 primjera, (3 pozitivna i 3 negativna) ima vrijednost ***Vjetar*** *= jak*

INFORMACIJSKA DOBIT MJERI OČEKIVANU REDUKCIJU U ENTROPIJI

- Informacijska dobit od klasificiranja izvornih 14 primjera po atributu *vjetar* se računa na slijedeći način:

A = ***Vjetar***

Vrijednost (***Vjetar***) = *slab, jak*

D = [9+, 5-]

$D_{slab} \leftarrow$ [6+,2-]…. ukupno 8 primjera

$D_{jak} \leftarrow$ [3+,3-] …. ukupno 6 primjera

Informacijska dobit (***Gain***) zbog odjeljivanja primjera skupa D na temelju vrijednosti atributa **Vjetar** jest:

$$Informacijska\_dobit(D, A) \equiv Entropija(D) - \sum_{v \in Vrijednost(A)} \frac{|Dv|}{|D|} Entropija(Dv)$$

INFORMACIJSKA DOBIT MJERI OČEKIVANU REDUKCIJU U ENTROPIJI

- Najprije računamo entropije skupova D, D_{slab}, D_{jak}

  Entropija(**D**) = 0.940 *(vidi prethodni primjer!)*

  Entropija(**D_{slab}**) = *Entropija*([6+, 2-]) = - (6/8)$\log_2$(6/8) - (2/8)$\log_2$(2/8) = 0.811

  Entropija(**D_{jak}**) = *Entropija*([3+, 3-]) = -(3/6)$\log_2$(3/6) - (3/6)$\log_2$(3/6) = 1

- ***Informacijska_dobit***(**D**, ***Vjetar***) ≡

  ≡ *Entropija*(**D**) – (8/14)***Entropija*(D$_{slab}$)** – (6/14)***Entropija*(D$_{jak}$)**

  ≡ 0.940 – (8/14)0.811 – (6/14)1.00 ≡ **0.048**

PRIMJER

- Da bismo ilustrirali algoritam ID3 promotrimo sljedeći primjer

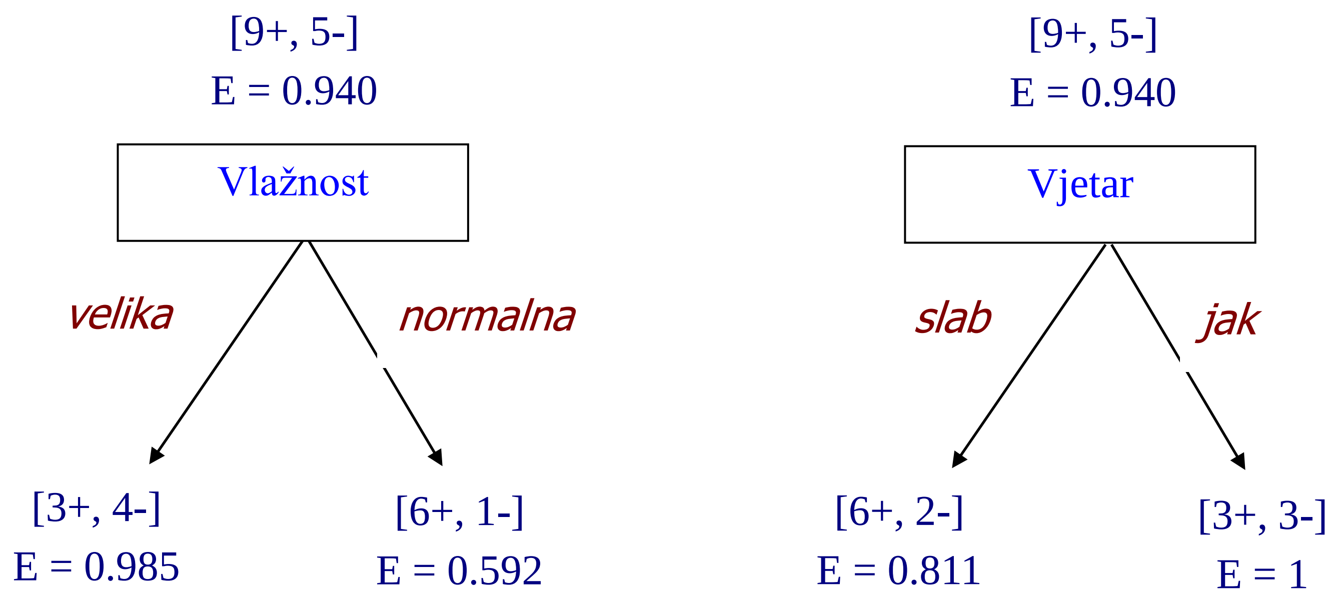
- **Vrijeme** {sunčano, oblačno, kišno}
- **Temperatura** {hladno, ugodno, vruće}
- **Vlažnost**{velika, normalna}
- **Vjetar** {jak, slab}

- Računamo informacijsku dobit sva četiri atributa da bismo odredili atribut s najvećom informacijskom dobiti koji će postati korijen stabla

PRIMJER

	Vrijeme	Temperatura	Vlažnost	Vjetar	Igra
1.	sunčano	vruće	velika	slab	NE
2.	sunčano	vruće	velika	jak	NE
3.	oblačno	vruće	velika	slab	DA
4.	kišno	ugodno	velika	slab	DA
5.	kišno	hladno	normalna	slab	DA
6.	kišno	hladno	normalna	jak	NE
7.	oblačno	hladno	normalna	jak	DA
8.	sunčano	ugodno	velika	slab	NE
9.	sunčano	hladno	normalna	slab	DA
10.	kišno	ugodno	normalna	slab	DA
11.	sunčano	ugodno	normalna	jak	DA
12.	oblačno	ugodno	velika	jak	DA
13.	oblačno	vruće	normalna	slab	DA
14.	kišno	ugodno	Velika	jak	NE
	sunčano[2+,3-] oblačno[4+,0-] kišno[3+,2-]	hladno[3+,1-] ugodno[4+,2-] vruće[2+,2-]	velika [3+, 4-] normalna [6+, 1-]	slab [6+, 2-] jak [3+, 3-]	[9+, 5-]

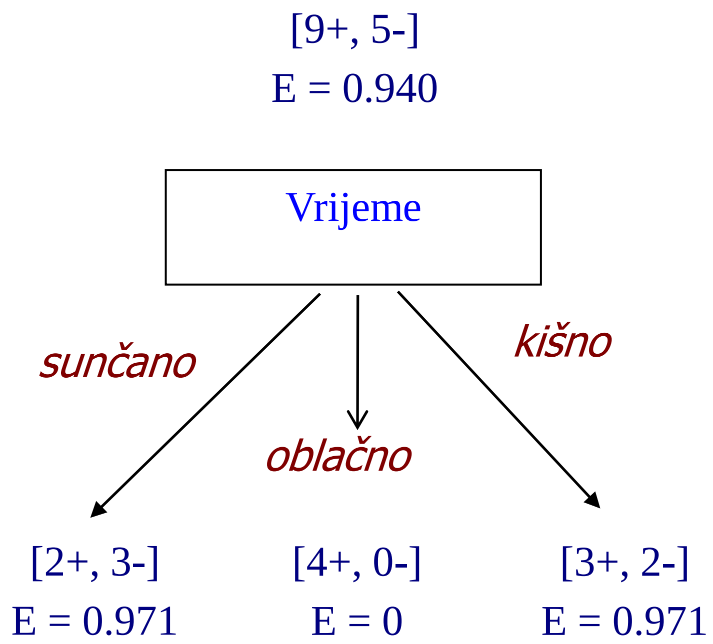
PRIMJER

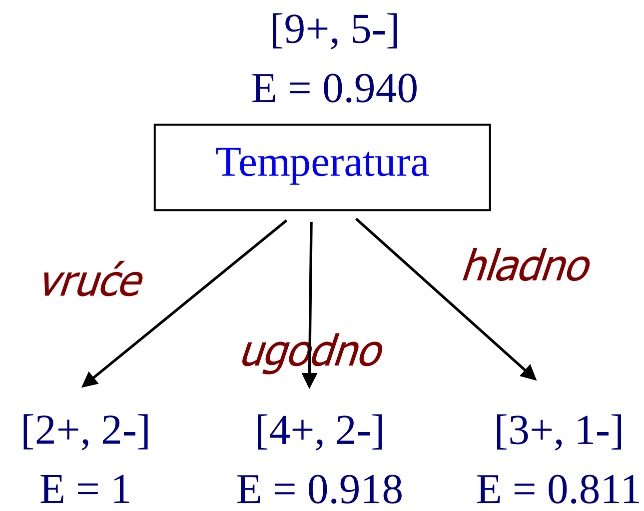


Informacijska_dobit(D, Vlažnost)
= 0.940 - (7/14) 0.985 - (7/14) 0.592
= **0.151 bita**

Informacijska_dobit(D, Vjetar)
= 0.940 - (8/14) 0.811 - (6/14) 1
= **0.048 bita**

PRIMJER





Informacijska_dobit(D, **Vrijeme**)
= 0.940 - (5/14) 0.971 - (4/14) 0
= (5/14) 0.971= **0.246 bita**

Informacijska_dobit(D, **Temperatura**)
= 0.940 - (4/14) 1 - (6/14) 0.918 -(4/14) 0.811= **0.029 bita**

Najveća informacijska dobit od sva četiri moguća atributa, pa će atribut Vrijeme biti korijen stabla!

PRIMJER

- **ID3** - Korijen stabla je ***Vrijeme***, listovi su vrijednosti tog atributa

- Elementi skupa za učenje **D** podjele se u tri grupe ($\mathbf{D_{sunčano}}$, $\mathbf{D_{oblačno}}$ i $\mathbf{D_{kišno}}$) prema vrijednostima atributa ***Vrijeme*** (sunčano, oblačno, kišno)

- **Za svaki takav podskup $\mathbf{D_{sunčano}}$, $\mathbf{D_{oblačno}}$ i $\mathbf{D_{kišno}}$ ponavlja se isti postupak**

- Entropija unutar grane *sunčano* tj. skupa $\mathbf{D}_{\mathbf{sunčano}}$

Vrijeme	Temperatura	Vlažnost	Vjetar	Igra
sunčano	vruće	velika	slab	NE
sunčano	vruće	velika	jak	NE
oblačno	vruće	velika	slab	DA
kišno	ugodno	velika	slab	DA
kišno	hladno	normalna	slab	DA
kišno	hladno	normalna	jak	NE
oblačno	hladno	normalna	jak	DA
sunčano	ugodno	velika	slab	NE
sunčano	hladno	normalna	slab	DA
kišno	ugodno	normalna	slab	DA
sunčano	ugodno	normalna	jak	DA
oblačno	ugodno	velika	jak	DA
oblačno	vruće	normalna	slab	DA
kišno	ugodno	velika	jak	NE

PRIMJER

Vrijeme

sunčano *oblačno* *kišno*

$D_{sunčano}[2+,3-]$

	Temp.	*Vlažnost*	*Vjetar*	*Odl.*
Sunčano	vruće	velika	slab	**NE**
Sunčano	vruće	velika	jak	**NE**
Sunčano	ugodno	velika	slab	**NE**
Sunčano	hladno	normalna	slab	**DA**
Sunčano	ugodno	normalna	jak	**DA**

$D_{kišno}[3+,2-]$

	ugodno	velika	slab	DA
	hladno	normalna	slab	DA
	hladno	normalna	jak	NE
	ugodno	normalna	slab	DA
	ugodno	velika	jak	NE

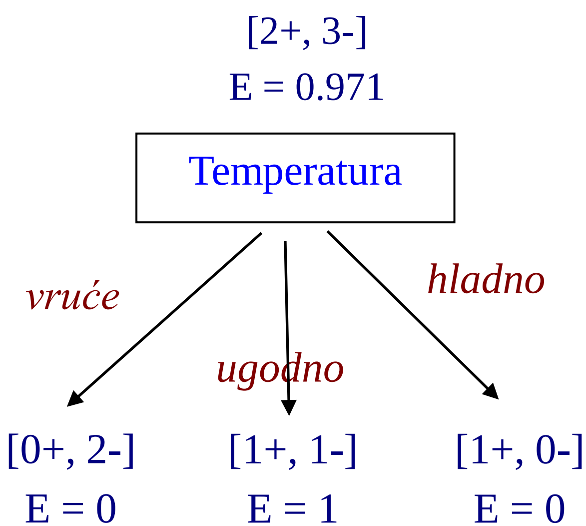
$D_{oblačno}[4+,0-]$

	vruće	velika	slab	**DA**
	hladno	normalna	jak	**DA**
	ugodno	velika	jak	**DA**
	vruće	normalna	slab	**DA**

$$\text{Entropija}(\mathbf{D}_{\mathbf{sunčano}}) = \text{Entropija}([2+,3-]) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

PRIMJER

- Unutar grane *sunčano* računamo informacijske dobiti za tri atributa, **Temperatura, Vlažnost** i **Vjetar**:



$D_{\text{sunčano}}[2+,3-]$ označimo kao D

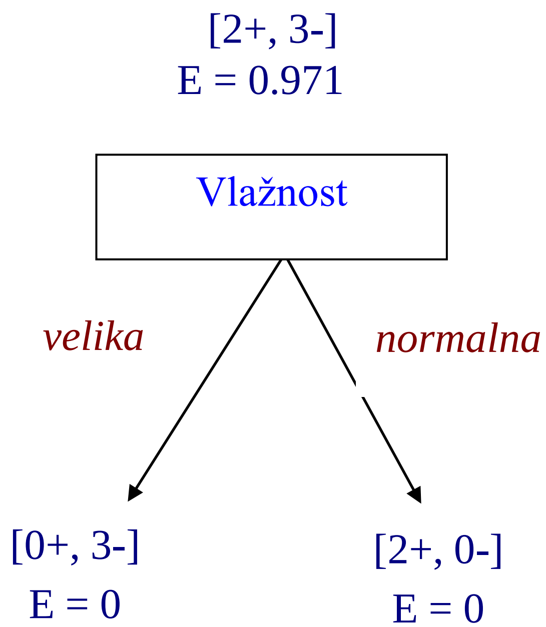
	Temper.	*Vlažnost*	*Vjetar*	
sunčano	vruće	velika	slab	**NE**
sunčano	vruće	velika	jak	**NE**
sunčano	ugodno	velika	slab	**NE**
sunčano	hladno	normalna	slab	**DA**
sunčano	ugodno	normalna	jak	**DA**

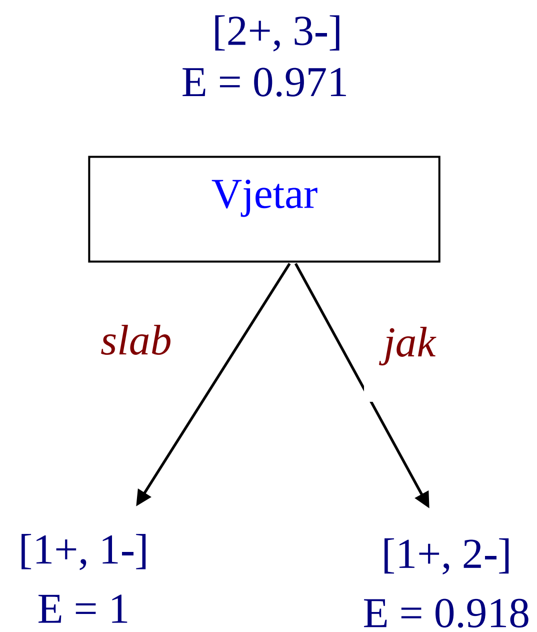
Informacijska_dobit($D_{\text{sunčano}}$, **Temperatura**)

= 0.971 - (2/5) 0 - (2/5) 1 -(1/5) 0

= **0.571**

PRIMJER





Informacijska_dobit(D $_{\text{sunčano}}$,, **Vlažnost**)
= 0.971 - (3/5) 0 - (2/5) 0
= **0.971**

Informacijska_dobit(D $_{\text{sunčano}}$,, **Vjetar**)
= 0.971 - (2/5) 1 - (3/5) 0.918
= **0.02**

PRIMJER

- Unutar grane *sunčano* najveću informacijsku dobit ima atribut **Vlažnost**, stoga je atribut **Vlažnost** čvor u drugoj razini stabla odluke niz granu *sunčano*

- Gore opisani postupak primjenjuje se na čvor **Vlažnost**. Razdjeljuje se skup primjera $\mathbf{D}_{\mathbf{sunčano}}$ niz grane *normalna* (skup $\mathbf{D}_{\mathbf{sunčano}}$**, normalna**) i *velika* (skup $\mathbf{D}_{\mathbf{sunčano}}$**, velika**)

PRIMJER

Vrijeme

sunčano

oblačno

kišno

$D_{kišno}$

kišno	ugodno	velika	slab	DA
kišno	hladno	normalna	slab	DA
kišno	hladno	normalna	jak	NE
kišno	ugodno	normalna	slab	DA
kišno	ugodno	velika	jak	NE

oblačno	vruće	velika	slab	**DA**
oblačno	hladno	normalna	jak	**DA**
oblačno	ugodno	velika	jak	**DA**
oblačno	vruće	normalna	slab	**DA**

Vlažnost

velika

Vlažnost	*Vjetar*	***Igra***
velika	slab	**NE**
velika	jak	**NE**
velika	slab	**NE**

Zaustavljamo se – svi su primjeri iz iste klase – “NE”

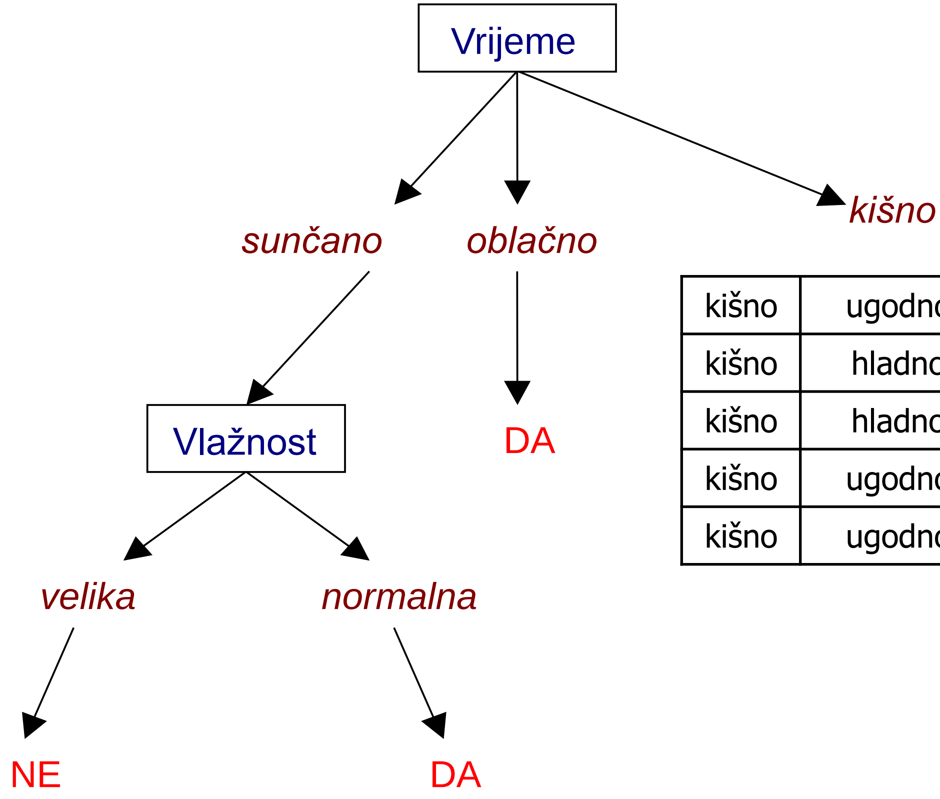
normalna

Vlažnost	*Vjetar*	***Igra***
normalna	slab	**DA**
normalna	jak	**DA**

Zaustavljamo se – svi su primjeri iz iste klase – “DA”

PRIMJER

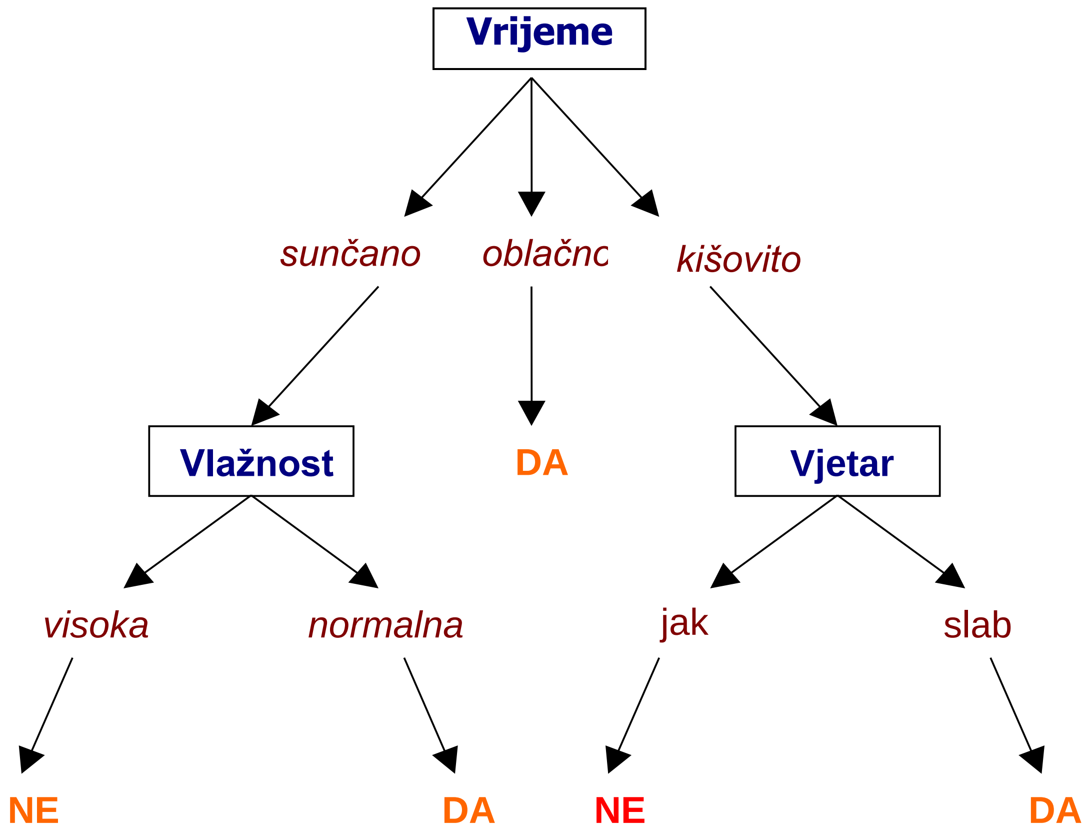
- Da svi primjeri nisu iz iste klase, trebalo bi još dodati čvor za vrijednost atributa **Temperatura** ili **Vjetar**



kišno	ugodno	velika	slab	DA
kišno	hladno	normalna	slab	DA
kišno	hladno	normalna	jak	NE
kišno	ugodno	normalna	slab	DA
kišno	ugodno	velika	jak	NE

PRIMJER

- Nakon analize **D_{kišno}**, tj. procjene informacijske dobiti za atribute **Temperatura, Vlažnost i Vjetar** konačno stablo odluke je oblika:



ALGORITAM ID3

- **ID3(*Primjeri, Ciljni_atribut, Atributi*)**

  Ciljni atribut je atribut čije vrijednosti trebaju biti određene stablom odluke. Atributi su lista drugih atributa koji mogu biti ispitani u postupku učenja stabla odluke. Algoritam vraća stablo odluke koje ispravno klasificira dane primjere.

Stvori korijen stabla ROOT

Ako su svi primjeri pozitivni, vrati stablo s jednim čvorom čija je oznaka = +

Ako su svi primjeri negativni, vrati stablo s jednim čvorom čija je oznaka = -

Ako je atribut prazan, vrati stablo s jednim čvorom ROOT, s oznakom = najčešća vrijednost *Ciljnog _atributa* u skupu *(Q: kada se ovo događa?)*

Inače

A ← atribut iz skupa *Atributa* koji najbolje klasificira *primjere* (tj. ima najveću informacijsku dobit)

ID3 ALGORITAM

Atribut za odluku u korijenu je A tj. ROOT $\leftarrow$ A

Za svaku moguću vrijednost ν_i od A

Dodaj novu granu stabla ispod korijena ROOT, koja odgovara testu A = ν_i

Neka $D_{\nu i}$ označava podskup skupa *Primjeri* koji imaju vrijednost ν_i za atribut A

Ako je skup $D_{\nu i}$ prazan

Ispod nove grane dodaj završni čvor (list) čija je oznaka jednaka najčešće pojavljivanoj vrijednosti atributa *Ciljni_atribut* u skupu *D*

(Q: kada se ovo događa?)

Inače ispod nove grane dodaj stablo

ID3($D_{\nu i}$, *Ciljni_atribut, Atributi* \ {*A*})

GENERALIZIRANI ALGORITAM (K>2 klasa)

- Općenit slučaj je kada imamo N primjera razdijeljenih u skupove koji pripadaju razredima C_j , j = 1, 2, 3,…, K
- Broj primjera u razredu C_j neka je N_j. Svaki primjer ima atribute A_a , a=1,..,n. Svaki atribut ima vrijednosti V_{av} , $v=1,..M_a$. (Radi jednostavnosti, pretpostavit ćemo da svi atributi imaju M vrijednosti.)
- Postupak ID3 za izgradnju stabla odluke je sljedeći:

Korak 1. *Izračunati početnu vrijednost entropije.* U skupu za učenje, pripadnost razredu je poznata za sve primjere. Zbog toga je početna entropija skupa D koji se sastoji od N primjera

$$Entropija(D) = \sum_{j=1}^{K} -\left(N_j / N \right) \log_2 \left(N_j / N \right) = \sum_{j=1}^{K} -P(C_j) \log_2 P(C_j)$$

GENERALIZIRANI ALGORITAM

- Korak 2. *Odabrati atribut koja će biti korijen stabla odluke.*

a) Za svaki atribut A_a, a=1, 2, 3, …, n, razdijeli originalni skup primjera na prvorazinske skupove prema vrijednostima V_{av} od mogućih M vrijednosti atributa A_a. Postoji N_{av} primjera u V_{av} grani, ali ti uzorci ne moraju nužno biti iz jednog razreda.

b) Za svaki podskup grane V_{av}, broj primjera koji pripadaju razredu C_j je $N_{av}(j)$. Izračunati entropiju te grane koristeći relaciju

$$Entropija(D, A_a, v) = \sum_{j=1}^{K} -\left(N_{av}(j) / N_{av} \right) \log_2 \left(N_{av}(j) / N_{av} \right)$$

GENERALIZIRANI ALGORITAM

Entropija sustava nakon testiranja atributa A_a je

$$Entropija(D, A_a) = \sum_{v=1}^{M} \sum_{j=1}^{K} \left(N_{av} \Big/ \sum_{v} N_{av} \right) \cdot \left[-\left(\frac{N_{av}(j)}{N_{av}} \right) \log_2 \left(\frac{N_{av}(j)}{N_{av}} \right) \right]$$

c) Pad entropije (tj. informacijska dobit) kao rezultat ispitivanja atributa A_a je

$$Informacijska\_dobit(a) = Entropija(D) - Entropija(D, A_a)$$

d) Izabrati atribut A_{a*} koji rezultira najvećom informacijskom dobiti, tj. za koju je *informacijska_dobit(a*)>informacijska_dobit(a)* za svaki a=1, 2, 3, …, *n*, *a* ≠ *a**.

e) Atribut A_{a*} postaje korijen stabla odluke

GENERALIZIRANI ALGORITAM

- Korak 3. *Izgraditi sljedeću razinu stabla odluke*. Izabrati atribut $A_{a'}$, koji će služiti kao prvorazinski čvor, takav da nakon testiranja $A_{a'}$ za sve grane dobijemo maksimalnu dobit informacijskog sadržaja ili maksimalni pad entropije

- Korak 4. *Ponavljati korake 1 do 3*. Nastavljati dok svi podskupovi ne budu iz jednog razreda tj. entropija sustava postane jednaka nuli

PRETRAŽIVANJE PROSTORA HIPOTEZA U UČENJU STABLA ODLUKE

Induktivne metode učenja:

- **Pretraživanje prostora hipoteza za onom koja najbolje odgovara primjerima za učenje**

Kakav prostor hipoteza pretražuje ID3?

- Svih mogućih stabala odluke, od praznog stabla prema složenijima koje ispravno klasificira primjere za učenje
- ID3 možemo promatrati kao pretraživanje prostora hipoteza metodom «uspona na vrh» (*engl. hill-climbing*) u kojem je heuristička funkcija (koja vodi pretraživanje) informacijska dobit
- Pohlepna metoda

ID3 pretražuje potpun prostor hipoteza

- Prostor hipoteza ID3 je prostor svih mogućih funkcija s konačno diskretnih vrijednosti (u odnosu na broj atributa). Svaka takva funkcija se može predočiti stablom odluke pa <u>ID3 izbjegava zamku pretraživanja nepotpunog prostora</u> hipoteza koji ne sadrži ciljni koncept (npr. u slučaju kada su hipoteze u obliku konjunkcije atributa)

ID3 pronalazi samo jednu hipotezu

- CE - nalazi sve hipoteze konzistentne s primjerima. Ne znamo <u>koliko je još stabala odluke konzistentno</u> s primjerima za učenje,
- Učenik <u>ne može postaviti upit</u> o primjeru koji će onda razriješiti između mogućih hipoteza

PRETRAŽIVANJE PROSTORA HIPOTEZA U UČENJU STABLA ODLUKE

ID3 u izvornom obliku se ne vraća unatrag u postupku pretraživanja

- To svojstvo ima isti nedostatak kao i pretraga metodom uspona na vrh – mogućnost da se zaglavi u lokalnom optimumu.

ID3 u svakom koraku koristi sve primjere za učenje da bi statistički rafinirao tekuću hipotezu

- Prednost uporabe statističkog svojstva svih primjera za učenje (tj. informacijske dobiti) jest manja osjetljivost na pogreške u skupu primjera za učenje.
  Algoritmi CE i Find-S donose odluke u koracima (inkrementalno) na temelju jednog predočenog primjera

INDUKTIVNA PRISTRANOST ALGORITMA ID3

- Induktivna pristranost je skup pretpostavki tako da skupa s primjerima za učenje deduktivno potvrđuju klasifikaciju koju određuje učenik na novom primjeru
- *ID3 - metoda »uspona na vrh» - prihvaćanje prve odgovarajuće hipoteze*
- Na temelju čega ID3 može generalizirati i klasificirati još neviđene primjere?
- Induktivna pristranost ID3: Na temelju čega ID3 preferira jednu konzistentnu hipotezu u odnosu na drugu?
  a) ID3 izabire kraće stablo prije nego dulje stablo
  b) Izabire stablo koje stavlja atribute s većom informacijskom dobiti bliže korijenu

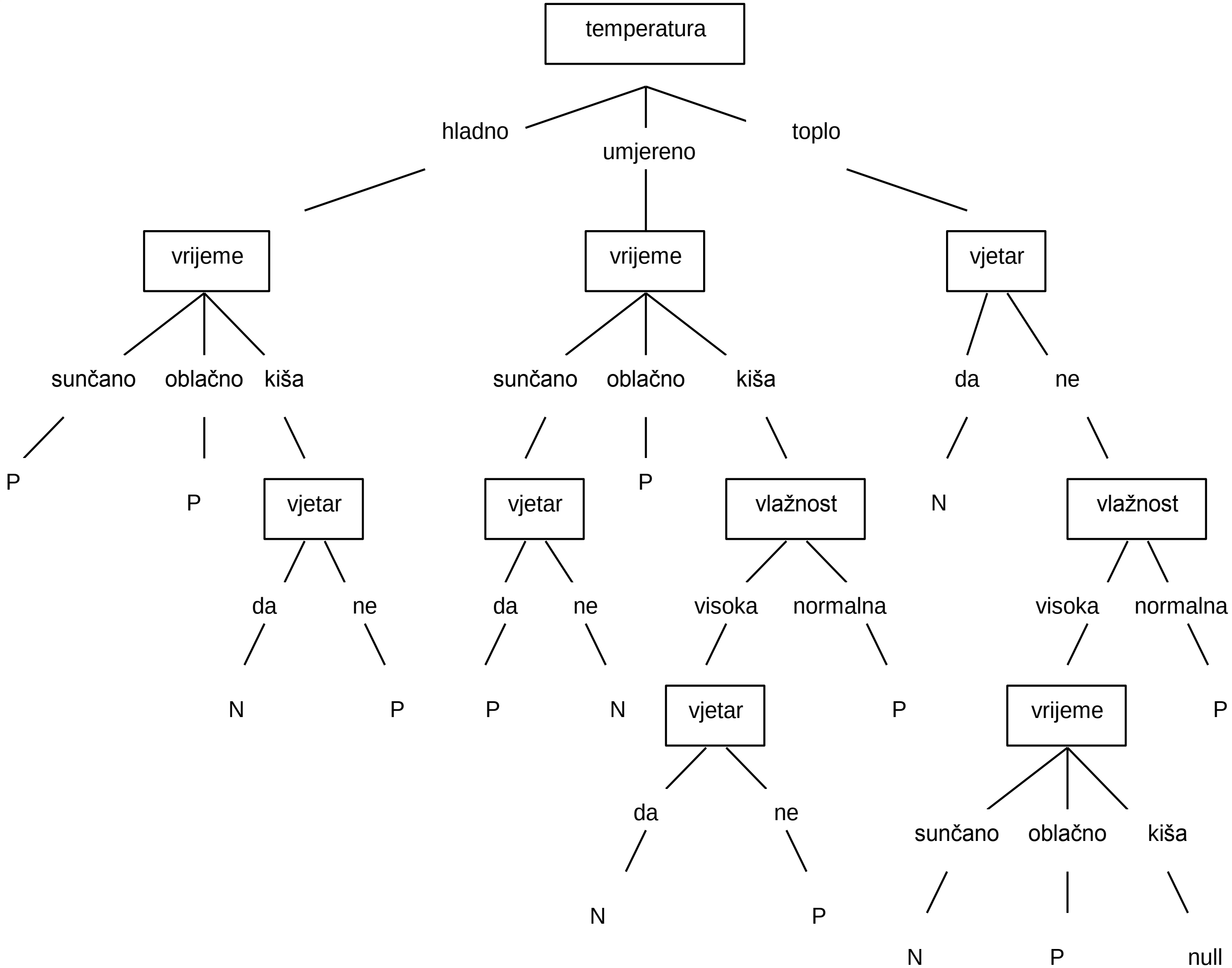
INDUKTIVNA PRISTRANOST ALGORITMA ID3

Približna induktivna pristranost ID3:
Preferiraju se kraća stabla odluke nad većima.

- Usporedba BFS *(engl. breadth first search)* i algoritma ID3

Bolja približna induktivna pristranost ID3:
Preferiraju se kraća stabla odluke nad većima.
Preferiraju se stabla koje stavljaju atribute s većom informacijskom dobiti bliže korijenu

INDUKTIVNA PRISTRANOST ALGORITMA ID3



INDUKTIVNA PRISTRANOST ALGORITMA ID3

- Jedan pristup zadatku zaključivanja bio bi generiranje svih mogućih stabala odluke koja ispravno klasificiraju uzorke iz skupa za učenje, te izabiranje najjednostavnijeg stabla. Broj takvih stabala je konačan, ali vrlo velik, pa je ovakav pristup primjenjiv jedino za manje zahtjevne zadatke
- ID3 je pogodan za zadatke za koje je karakteristično puno atributa i gdje se skup za učenje sastoji od puno uzoraka, ali ipak je moguće ostvariti prilično dobro stablo bez previše računanja
- Općenito, ID3 gradi jednostavna stabla odluke, ali pristup koji koristi ne garantira da se bolje stablo ne može pronaći

PRISTRANOSTI RESTRIKCIJOM I PRISTRANOSTI PREFERENCIJOM

Različiti tipovi induktivne pristranosti

	ID3	**Eliminacija kandidata (CE)**
Prostor hipoteza *koji se pretražuje*	Potpun	Nepotpun (onaj koji se može izraziti hipotezom)
Način pretraživanja *tog prostora*		
Induktivna pristranost isključivo povezana s:		
Induktivna pristranost		

PRISTRANOSTI RESTRIKCIJOM I PRISTRANOSTI PREFERENCIJOM

Različiti tipovi induktivne pristranosti

	ID3	**Eliminacija kandidata (CE)**
Prostor hipoteza *koji se pretražuje*	Potpun	Nepotpun (onaj koji se može izraziti hipotezom)
Način pretraživanja *tog prostora*	Nepotpuno pretraživanje (od jednostavnijih do složenijih) dok ne nađe hipotezu konzistentnu s podacima	Temeljito (potpuno) pretraživanje
Induktivna pristranost isključivo povezana s:		
Induktivna pristranost		

PRISTRANOSTI RESTRIKCIJOM I PRISTRANOSTI PREFERENCIJOM

Različiti tipovi induktivne pristranosti

	ID3	Eliminacija kandidata (CE)
Prostor hipoteza *koji se pretražuje*	Potpun	Nepotpun (onaj koji se može izraziti hipotezom)
Način pretraživanja *tog prostora*	Nepotpuno pretraživanje (od jednostavnijih do složenijih) dok ne nađe hipotezu konzistentnu s podacima	Temeljito (potpuno) pretraživanje
Induktivna pristranost isključivo povezana s:	uređajnom strategijom pretraživanja hipoteza	ekspresivnom moći predstavljanja hipoteza
Induktivna pristranost		

PRISTRANOSTI RESTRIKCIJOM I PRISTRANOSTI PREFERENCIJOM

Različiti tipovi induktivne pristranosti

	ID3	**Eliminacija kandidata (CE)**
Prostor hipoteza *koji se pretražuje*	Potpun	Nepotpun (onaj koji se može izraziti hipotezom)
Način pretraživanja *tog prostora*	Nepotpuno pretraživanje (od jednostavnijih do složenijih) dok ne nađe hipotezu konzistentnu s podacima	Temeljito (potpuno) pretraživanje
Induktivna pristranost isključivo povezana s:	uređajnom strategijom pretraživanja hipoteza	ekspresivnom moći predstavljanja hipoteza
Induktivna pristranost	Preferencija nekih hipoteza nad drugima	Restrikcija skupa razmatranih hipoteza

PRISTRANOSTI RESTRIKCIJOM I PRISTRANOSTI PREFERENCIJOM

Različiti tipovi induktivne pristranosti

	ID3	**Eliminacija kandidata (CE)**
Prostor hipoteza *koji se pretražuje*	Potpun	Nepotpun (onaj koji se može izraziti hipotezom)
Način pretraživanja *tog prostora*	Nepotpuno pretraživanje (od jednostavnijih do složenijih) dok ne nađe hipotezu konzistentnu s podacima	Temeljito (potpuno) pretraživanje
Induktivna pristranost isključivo povezana s:	uređajnom strategijom pretraživanja hipoteza	ekspresivnom moći predstavljanja hipoteza
Induktivna pristranost	Preferencija nekih hipoteza nad drugima	Restrikcija skupa razmatranih hipoteza
	Pristranost preferencijom ili pristranost pretraživanja *(engl. preference bias, search bias)*	**Pristranost restrikcijom ili pristranost jezika** *(engl. preference bias, search bias)*

PRISTRANOSTI RESTRIKCIJOM I PRISTRANOSTI PREFERENCIJOM

- Koja je pristranost općenito poželjnija?
- Neki sustavi strojnog učenja kombiniraju ove dvije vrste pristranosti

Primjer

Sustav koji uči igrati igru DAME:

Pristranost jezika	**Pristranost pretraživanja**
Izbor linearne evaluacijske funkcije značajki igre	Izbor algoritma LMS u odnosu na druge moguće algoritme za podešavanja parametara

ZAŠTO PREFERIRATI KRAĆE HIPOTEZE?

- Filozofsko pitanje

1320.g. William of Occam

"Pluralitas non est ponenda sine neccesitate"

Primjer: Za neki skup podataka može biti nebrojeno teorija koje ih objašnjavaju. Četiri točke na pravcu – postoji bezbroj krivulja koje se mogu povući kroz te točke, no pravac je najjednostavnija

PRAKTIČNI PROBLEMI VEZANI ZA UČENJE STABLA ODLUKE

- određivanje dubine rasta stabla
- atributi s kontinuiranim vrijednostima
- mjera za izbor atributa
- nedostajuće vrijednosti atributa
- efikasnost računanja

Proširenje ID3 – algoritam C4.5 (Quinlan, 1993)

IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA

- Algoritam ID3 – rast stabla dok se svi podaci pravilno ne klasificiraju
- To je problem ako su:
  - podaci sa šumom
  - skup za učenje je premalen.
- Tada može doći do prenaučenosti (*engl. overfit*) stabla odluke

IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA

- Algoritam ID3 – rast stabla dok se svi podaci pravilno ne klasificiraju
- To je problem ako su:
  - podaci sa šumom
  - skup za učenje je premalen.
- Tada može doći do prenaučenosti (*engl. overfit*) stabla odluke

Definicija

- Neka je dan prostor hipoteza H. **Hipoteza h $\in$ H je prenaučena** ako postoji hipoteza h' $\in$ H takva da h ima manju pogrešku nego h' na na primjerima za učenje, ali h' ima manju pogrešku nego h na cijelom prostoru primjera

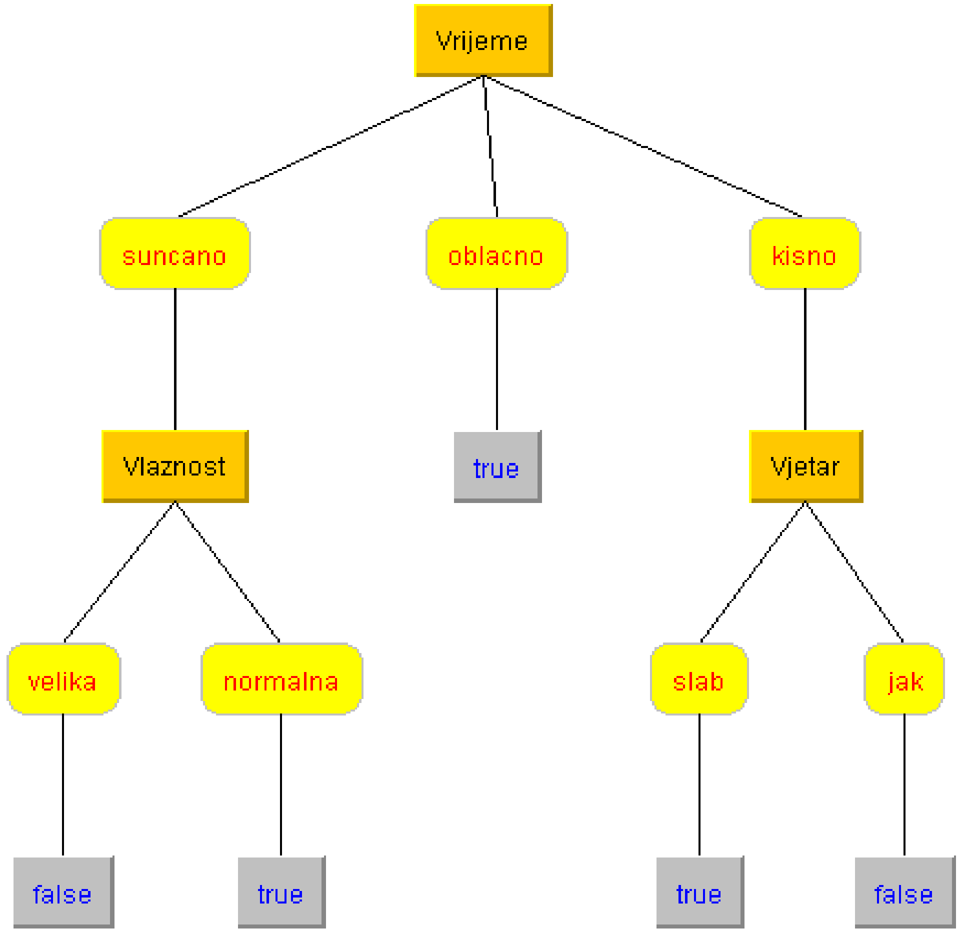
IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA

Primjer prenaučenosti

- Pretpostavimo da je dodan 15. primjer u skup primjera koji je pogrešno klasificiran kao Igra = NE umjesto Igra = DA.
- Primjetimo da bi postojeće stablo ispravno klasificiralo ispravan primjer (igra = DA) u istu granu kao i 9. i 11. primjer

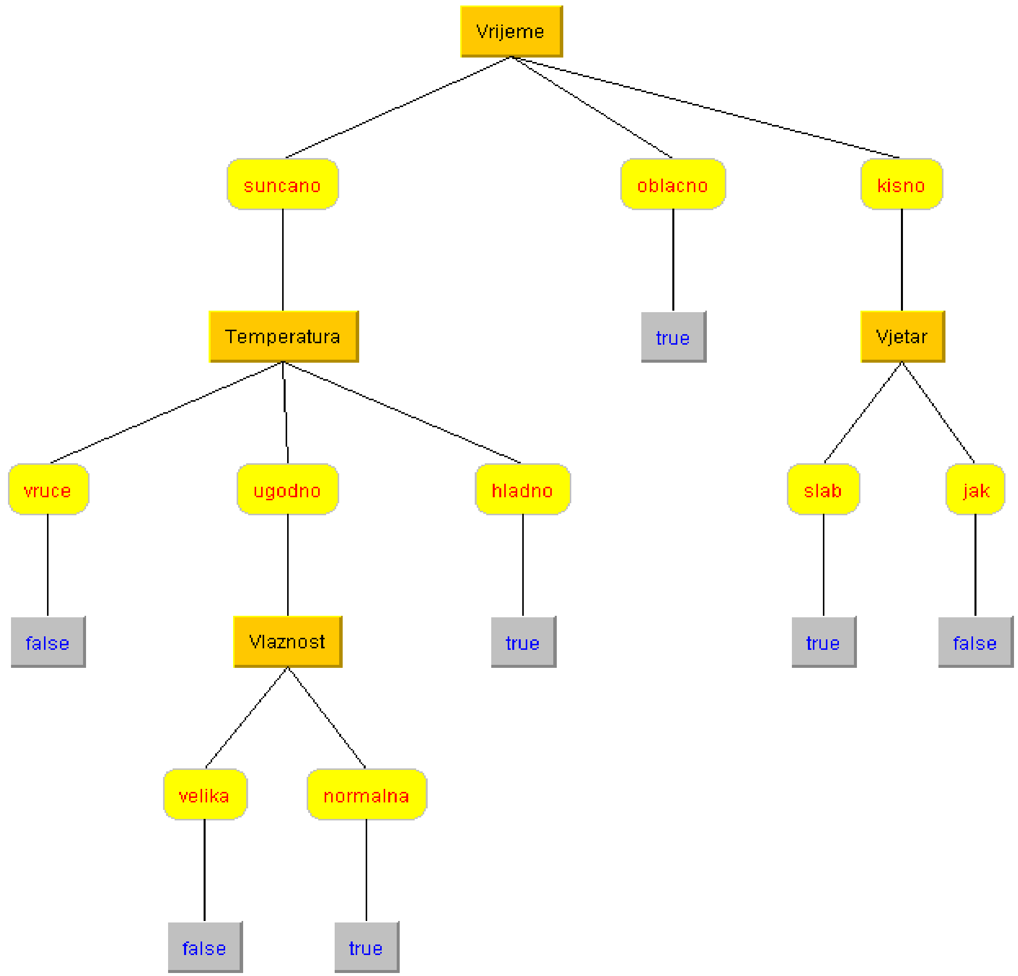
	Vrijeme	Temperatura	Vlažnost	Vjetar	Igra
15.	sunčano	vruće	normalna	jak	NE

IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA



IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA

	Vrijeme	Temperatura	Vlažnost	Vjetar	Igra
15.	sunčano	vruće	normalna	jak	NE



IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA

Izbjegavanje prenaučenosti – dva pristupa:

- zaustavljanje rasta stabla prije savršene klasifikacije primjera za učenje
- naknadno podrezivanje prenaučenog stabla → uspješniji pristup u praksi

Kako odrediti razumnu veličinu stabla?

- uvođenjem posebnog skupa podataka za vrednovanje (najčešće u praksi)

  skup primjera:
- skup za učenje (*engl. training set*)
- skup za vrednovanje (*engl. validation set*) → osigurava da ne dođe do prenaučenosti

ideja: mala je vjerojatnost da skup za vrednovanje ima ista slučajna odstupanja kao i skup za učenje

IZBJEGAVANJE PREKOMJERNE NAUČENOSTI PODATAKA

- uporaba statističkih testova - testiranja je li uvođenje ili uklanjanje čvora donosi poboljšanje u odnosu na cjelokupnu distribuciji (a ne samo na primjerima za učenje, primjer: Quinlan, 1986., χ^2 test)

- uvođenje eksplicitne mjere kompleksnosti kodiranja primjera za učenje i stabla odluke i zaustavljanja kada je ta mjera minimalna.

  Primjer: Načelo najmanje duljine opisa (*engl. minimum description length principle*)

SMANJIVANJE POGREŠKE PODREZIVANJEM

Podrezivanje stabla znači uklanjanje čvora i pripadnog podstabla koje ima korijen u tom čvoru, zamjenjeujući ga s listom tako da se listu pridruži najčešća vrijednost ciljnog aributa u tom podčvoru.

- Svaki je čvor kandidat za podrezivanje
- Čvorovi se uklanjaju samo ako se dobiveno podrezano stablo <u>ne ponaša lošije na skupu za vrednovanje</u>
- Na taj se način uklanjaju čvorovi dodani zbog slučajnih nepravilnosti u skupu za učenje kojih nema u skupu za vrednovanje

SMANJIVANJE POGREŠKE PODREZIVANJEM

- Uklanjanje je iterativan postupak – traje sve dok se ne počne smanjivati točnost na skupu za vrednovanje

Tri skupa:

  - skup za učenje
  - skup za vrednovanje – (ovaj skup vodi postupak podrezivanja)
  - skup za testiranje

- Ovakav pristup podrazumijeva veliki skup ulaznih podataka

NAKNADNO PODREZIVANJE PRAVILA

- Ideja:

stablo odluke → pravila → podrezivanje → podrezana pravila

- Ovu metodu koristi C4.5.

1. Nauči stablo odluke iz skupa za učenje sve dok svi podaci ne pristaju dobro, dozvoljavajući prenaučenost
2. Pretvori stablo u ekvivalentan skup pravila stvarajući jedno pravilo za svaku stazu od korijena do lista
3. Podrezuj (poopći) pravila uklanjajući bilo koji preduvjet koji rezultira u poboljšanju procijenjene točnosti
4. Složi podrezana pravila po procijenjenoj točnosti i razmatraj ih u tom nizu kod klasificiranja primjera

NAKNADNO PODREZIVANJE PRAVILA

- Primjer: Najljevija grana stabla odluke (Quinlanov primjer)

AKO (**Vrijeme** = *sunčano*) $\wedge$ (**Vlažnost** = *visoka*)
ONDA (**Igranje_tenisa** = **NE**)

- Pravilo se podrezuje tako da se uklanjaju uvjeti iz lijevog dijela pravila ((**Vrijeme** = *sunčano*) i (**Vlažnost** = *visoka*)) čije uklanjanje ne pogoršava procijenjenu točnost

Kako procijeniti točnost pravila?

1. Uporaba skupa za vrednovanje (*engl. validation set*) ≠ od skupa za učenje

2. Računanje točnosti pravila na skupu za učenje i računanju donje granice intervala pouzdanosti pretpostavljajući binomnu distribuciju. Ta se donja granica smatra mjerom preformanse pravila. Procjena donje granice intervala pouzdanosti ovisi o veličini skupa za testiranje

NAKNADNO PODREZIVANJE PRAVILA

Zašto konvertirati stablo odluke u pravila?

1. Pravila omogućuju razlikovanje konteksta u kojem je čvor korišten. Čvor se razmatra zasebno u svakom pravilu i kojem sudjeluje (zato što grana koja daje pravilo prolazi kroz taj čvor). Ako se čvor uklanja u stablu – istodobno se uklanja prisutnost tog uvjeta (čvora) u svim pravilima (u kojima se taj uvjet pojavljuje na lijevoj strani)
2. Uklanja se razlika između testiranja atributa koji se nalaze na dnu stabla (blizu listu) ili pri vrhu (korijenu)
3. Pravila povećavaju čitljivost i razumljivost

ATRIBUTI S KONTINUIRANIM VRIJEDNOSTIMA

- Atributi koji se testiraju morali su imati konačan skup diskretnih vrijednosti. Ovo ograničenje može se ukloniti dinamičkim definiranjem novih diskretnih vrijednosti atributa u obliku skupa diskretnih intervala
- A - atribut s kontinuiranim vrijednostima
- $c \in$ domena(A)
- Algoritam definira novi Booleov atribut $\mathbf{A_c}$ takav da je
  $\mathbf{A_c}$ **istinit** ako vrijednost(A) < c
  inače $\mathbf{A_c}$ **lažan**

ATRIBUTI S KONTINUIRANIM VRIJEDNOSTIMA

Kako odabrati najbolju vrijednost za c?

Primjer

- Pretpostavimo da primjeri za učenje pridruženi nekom čvoru imaju sljedeće kontinuirane vrijednosti za atribut **Temperatura** i za ciljni koncept **Igranje_tenisa**

Temperatura	40	48	60	72	80	90
Igranje_tenisa	NE	NE	DA	DA	DA	NE

- Želimo izabrati *c* tako da imamo najveću informacijsku dobit

ATRIBUTI S KONTINUIRANIM VRIJEDNOSTIMA

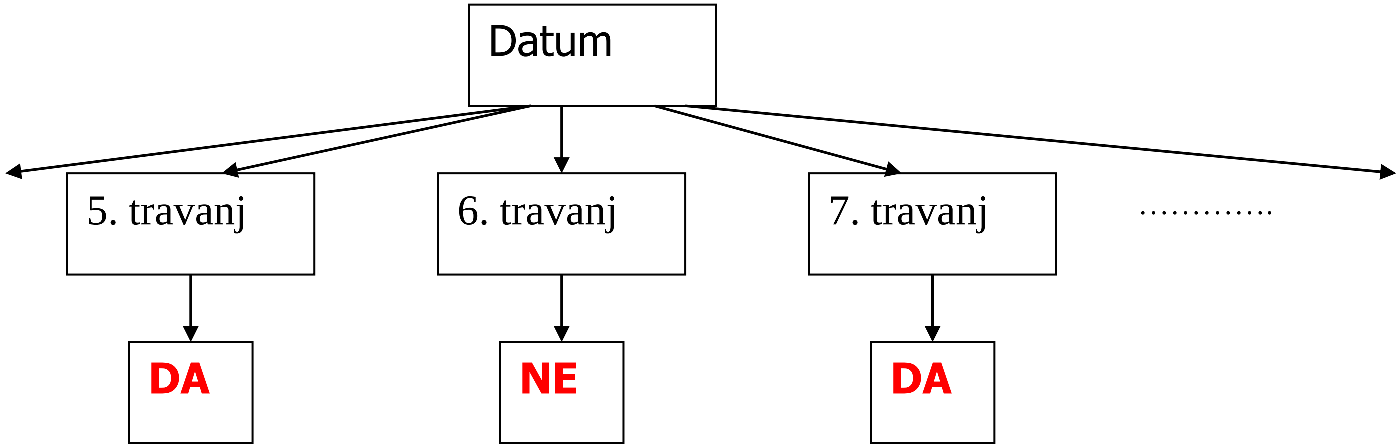
1	Vrijednosti atributa A slože se u rastućem redoslijedu.	*Već učinjeno u tablici*
2	Odrede se one susjedne vrijednosti atributa koje se razlikuju u klasifikaciji ciljnog atributa.	(48, 60) i (80, 90)
3	Nađe se srednja vrijednost takvih vrijednosti atributa. Te srednje vrijednosti čine kandidate za graničnu vrijednost *c*	c=(48+60)/2=54 c=(80+90)/2=85
4	Računa se informacijska dobit za svaki takav kandidat za graničnu vrijednost *c*	$IG(Temperatura_{>54})$ $IG(Temperatura_{>85})$
5	Odabire se *c* s najvećom vrijednošću $I(Temperatura_{>c})$	$IG(Temperatura_{>54})$

ALTERNATIVNE MJERE ZA IZBOR ATRIBUTA

- Informacijska dobit sadrži pristranost koja preferira atribute s više vrijednosti

Primjer

- Kada bi dodali atribut *Datum* u tablicu tada bi datum imao najveću informacijsku dobit zato što bi savršeno predviđao vrijednost ciljnog atributa



- Ovakvo bi se stablo ponašalo loše na novim podacima

ALTERNATIVNE MJERE ZA IZBOR ATRIBUTA

- Alternativna mjera **Omjer dobitka** (*engl. gain ratio*) (Quinlan, 1986) koji kažnjava atribute poput *Datum* zbog člana informacijska podijeljenost (*engl. split information*) koji je osjetljiv na to koliko široko i uniformno atribut dijeli podatke

$$Informacijska\_podijeljenost(D, A) = -\sum_{v=1}^{M} \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|}$$

- $D_1, \ldots, D_v, \ldots, D_M$ su podskupovi skupa primjera D koji nastaju particijom D s obzirom na vrijednost *v* atributa A

(*Entropija u odnosu na vrijednosti atributa*)

Zadatak

Kolika je informacijska podijeljenost atributa:

1. koji uniformno distribuira vrijednosti poput atributa *Datum*?
2. Booleovog atributa koji dijeli *n* primjera točno na pola?

ALTERNATIVNE MJERE ZA IZBOR ATRIBUTA

Odgovor:

1. $\log_2 M$
2. 1

- Omjer dobitka se definira

$$Omjer\_dobitka(D,A) = \frac{Informacijska\_dobit(D,A)}{Informacijska\_podijeljenost(D,A)}$$

- Ako dva atributa imaju istu informacijsku dobit preferirati će se onaj koji ima manju informacijsku podijeljenost

ALTERNATIVNE MJERE ZA IZBOR ATRIBUTA

Što ako je nazivnik blizu 0?

- Za $|D_v| \approx |D|$, nazivnik je blizu 0 što čini omjer dobitka vrlo velik ili nedefiniran za atribute koji imaju skoro svuda istu vrijednost

- *Izbjegavanje takve situacije:* Za sve atribute se računa informacijska dobit, a omjer dobitka se računa samo za one atribute koji imaju informacijsku dobit iznad prosječne vrijednosti (a to su upravo problematični atributi poput atributa *Datum*)

NEDOSTAJUĆE VRIJEDNOSTI

- Pretpostavimo da u nekom čvoru stabla trebamo računati informacijsku dobit atributa A te da postoji primjer *(x, y)* za koje je vrijednost atributa nepoznata

Primjer:

A=Vjetar

NEDOSTAJUĆE VRIJEDNOSTI

	Vrijeme	Temperatura	Vlažnost	Vjetar	Igra
1	sunčano	vruće	velika	slab	NE
2	sunčano	vruće	velika	jak	NE
3	oblačno	vruće	velika	slab	DA
4	kišno	ugodno	velika	slab	DA
5	kišno	hladno	normalna	slab	DA
6	kišno	hladno	normalna	jak	NE
7	oblačno	hladno	normalna	jak	DA
8	sunčano	ugodno	velika	slab	NE
9	sunčano	hladno	normalna	slab	DA
10	kišno	ugodno	normalna	slab	DA
11	sunčano	ugodno	normalna	jak	DA
12	oblačno	ugodno	velika	jak	DA
13	oblačno	vruće	normalna	slab	DA
14	kišno	ugodno	velika	jak	NE
	sunčano[2+,3-] oblačno[4+,0-] kišno[3+,2-]	hladno[3+,1-] ugodno[4+,2-] vruće[2+,2-]	vel. [3+, 4-] norm. [6+,1-]	**slab [6+, 2-]** **jak [2+, 3-]**	[9+, 5-]

NEDOSTAJUĆE VRIJEDNOSTI

Prvi pristup:

1. Pridjeliti najćešću vrijednost tog atributa na temelju primjera u tom čvoru ili
2. Pridjeliti najčešću vrijednost tog atributa koja se pojavljuje među primjerima klasificiranim s *y* u tom čvoru

Drugi pristup:

- Pridjeljivanje vjerojatnosti svakoj mogućoj vrijednosti atributa u tom čvoru. Vjerojatnost se temelji na relativnim frekvencijama poznatih primjera

NEDOSTAJUĆE VRIJEDNOSTI

Primjer:

Vjerojatnost P(*Vjetar*=**jak**) = 5/13

Vjerojatnost P(*Vjetar*=**slab)** = 8/13

Sada se ti omjeri koriste za računanje informacijske dobiti

A = ***Vjetar***

Vrijednost (***Vjetar***) = *slab, jak*

D = [9+, 5-] - izračunato na temelju 14 primjera

$D_{slab} \leftarrow$ [6+,2-] …. ukupno 8 primjera

$D_{jak} \leftarrow$ [2+,3-] …. ukupno 5 primjera

izračunato na temelju 13 primjera

NEDOSTAJUĆE VRIJEDNOSTI

- **Informacijska dobit** (***Gain***) zbog odjeljivanja primjera skupa D na temelju vrijednosti atributa **Vjetar** jest

$$Informacijska\_dobit(D, A) \equiv Entropija(D) - \sum_{v \in Vrijednosti(A)} \frac{|Dv|}{|D|} Entropija(Dv)$$

- Najprije računamo entropije skupova D, D_{slab}, D_{jak}

Entropija(**D**) = 0.940 *(vidi raniji primjer!)*

Entropija($\mathbf{D_{slab}}$) = *Entropija*([6+, 2-]) = - (6/8)$\log_2$(6/8) - (2/8)$\log_2$(2/8) = 0.811

Entropija($\mathbf{D_{jak}}$) = *Entropija*([2+, 3-]) = -(3/5)$\log_2$(3/5) - (2/5)$\log_2$(2/5) = 0.970

NEDOSTAJUĆE VRIJEDNOSTI

***Informacijska_dobit*(D, Vjetar)** ≡

≡ *Entropija*(**D**) – (8/13)*Entropija*($\mathbf{D_{slab}}$) – (5/13)*Entropija*($\mathbf{D_{jak}}$) =

≡ 0.940 – (8/13)0.811 – (5/13)0.970 =

≡ **0.06784**