3. Stabla odluke

3.1. Uvod

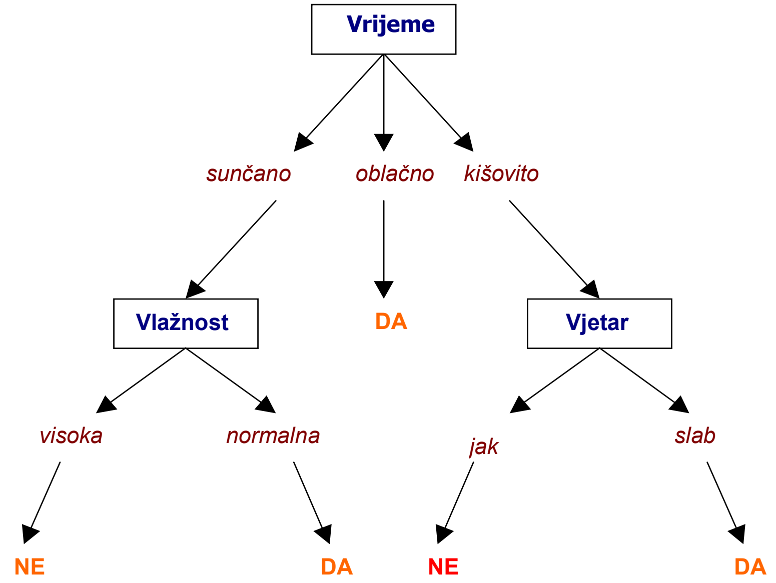
- Najčešće korištena metoda induktivnog zaključivanja (medicina, financije..)
- Metoda aproksimiranja funkcije diskretnih vrijednosti, robusna na šum, koja može učiti disjunktivne koncepte.
- Familija algoritama : ID3, ASSISTANT, C4.5
- Pretražuju potpun prostor hipoteza
- Induktivna pristranost: preferiraju se mala stabla u odnosu na velika
- Stabla odluke → (reinterpretacija) → skup ako-onda pravila

3.2. 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** - Da li je subotnje jutro pogodno za tenis?



Primjer:

(**Vrijeme** = *sunčano*, **Temperatura** = *vruće*, **Vlažnost** = *visoka*, **Vjetar** = *jak*)
→ (Klasifikacija, **Igranje_tenisa** = **NE**)

Općenito, stabla odluke predstavljaju disjunkciju konjunkcije uvjeta na vrijednosti atributa:

(**Vrijeme** = *sunčano* ∧ **Vlažnost** = *normalna*)
∨ (**Vrijeme** = *oblačno*)
∨ (**Vrijeme** = kišovito ∧ **Vjetar**= *jak*).

3.3. Problemi pogodni za oblikovanje stablima odluke

Problemi klasifikacije

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

3.4. 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.

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

sličan izgovor kao broj 3 – engl. ***three***

«Koji atribut odabrati za testiranje?»
→ testira se svaki atribut da se ocjeni kako dobro klasificira primjere
→ najbolji se odabire kao čvor, a njegove vrijednosti su silazne grane
→ primjeri za učenje se sortiraju prema odgovarajućem silaznom čvoru (niz onu granu koja odgovara vrijednosti tog atributa)
→ cijeli postupak se ponavlja koristeći primjere koji su dodijeljeni silaznom čvoru

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

3.4.1. 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.

3.4.1.1 Entropija mjeri homogenost primjera

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

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

Gdje je:
$\mathbf{p_+}$ proporcija pozitivnih primjera u S,
$\mathbf{p_-}$ proporcija negativnih primjera u S.
Po definiciji: $0\log_2 0 \equiv 0$.

Primjer:
S se sastoji od 9 pozitivnih i 5 negativnih primjera.
Usvojena notacija [9+, 5-].

Entropija([9+, 5-]) = $-(9/14) \log_2 (9/14) - (5/14) \log_2 (5/14) = \mathbf{0.940}$

1. Ako svi primjeri pripadaju istoj klasi, kolika je entropija?

2. Kolika je entropija za skup S koji sadrži isti broj pozitivnih i negativnih primjera?

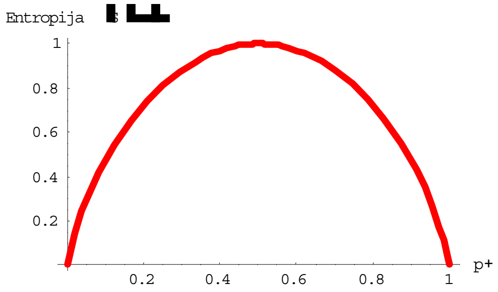
```
Entropy [illegible]
Entropy [illegible]

0.940286

Entropy [illegible]

   1.
```

Interpretacija entropije: *minimalni broj bitova potreban za kodiranje klasifikacije proizvoljnih članova skupa S.*



U slučaju c klasa:

$$\boxed{\textbf{Entropija(S)} = \sum_{i=1}^{c} -p_i \log_2 p_i}$$

3.4.1.2 Informacijska dobit mjeri očekivanu redukciju u entropiji

Entropija mjeri stupanj «neurednosti» podataka.

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 S jest:

$$\text{Informacijska\_dobit}(S,A) \equiv \text{Entropija}(S) - \sum_{v \in \text{Vrijednost}(A)} \frac{|S_v|}{|S|}\text{Entropija}(S_v)$$

Vrijednost(A) - skup svih mogućih vrijednosti atributa A

S_v - podskup od S za koji atribut A ima vrijednost v, tj. $S_v = \{s \in S | A(s) = v\}$

Entropija izvornog skupa S

Očekivana vrijednost entropije nakon podjele S na temelju atributa A

G(S, A) je informacija o vrijednosti ciljne funkcije, ako je dana vrijednost atributa A.
Vrijednost G(S, A) je ušteđen broj bitova sačuvan kod kodiranja ciljne funkcije proizvoljnog člana iz skupa primjera S, ako je poznata vrijednost atributa A.

Primjer:
Neka je S skup primjera opisan atributom ***Vjetar*** *={jak, slab}* i neka S ima 14 primjera , 9+ i 5-.

Od tih 14 primjera, ukupno 8 primjera (6 pozitivnih i 2 negativna) imaju vrijednost ***Vjetar*** *= slab*, a ostatak (6 primjera, od toga 3 pozitivna i 3 negativna) ima vrijednost ***Vjetar*** *= jak.*

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

A = ***Vjetar***
Vrijednost (***Vjetar***) = *slab, jak*
S = [9+, 5-]
$S_{slab} \leftarrow$ [6+,2-]…. ukupno 8 primjera
$S_{jak} \leftarrow$ [3+,3-] …. ukupno 6 primjera

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

$$\text{Informacijska\_dobit}(S,A) \equiv \text{Entropija}(S) - \sum_{v \in \text{Vrijednost}(A)} \frac{|S_v|}{|S|}\text{Entropija}(S_v)$$

Najprije računamo entropije skupova S, S_{slab}, S_{jak}.

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

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

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

Informacijska_dobit(**S**, **Vjetar**) ≡

≡ *Entropija*(**S**) – (8/14)*Entropija*($\mathbf{S_{slab}}$) – (6/14)*Entropija*($\mathbf{S_{jak}}$) =

≡ 0.940 – (8/14)0.811 – (6/14)1.00

≡ **0.048**

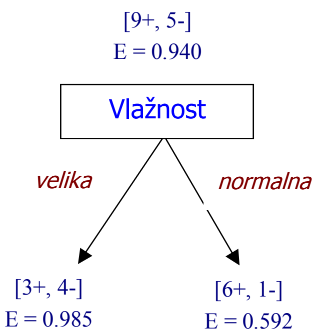
3.4.2. Primjer

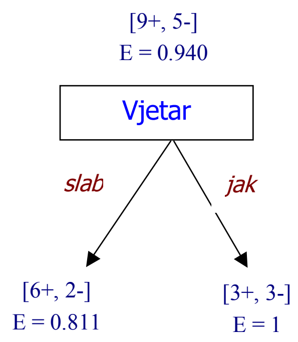
Da bi ilustrirali ID3 algoritam promotrimo slijedeći primjer.

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

	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 [3+, 3-]	[9+, 5-]

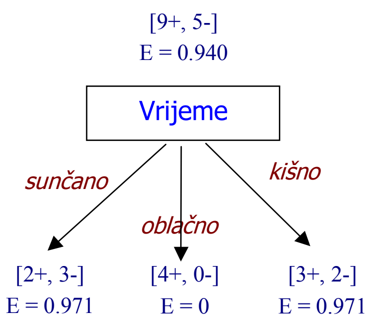
Računamo informacijsku dobit sva četiri atributa da bi odredili atribut s najvećom informacijskom dobiti koji će postati korijen stabla.

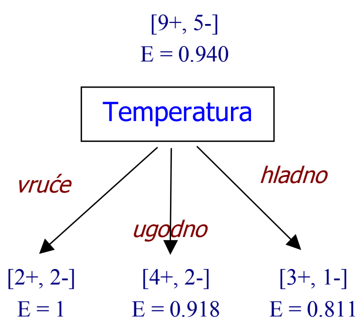




Informacijska_dobit(S, Vlažnost)
= 0.940 - (7/14) 0.985 - (7/14) 0.592
= **0.151 bita**

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





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

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

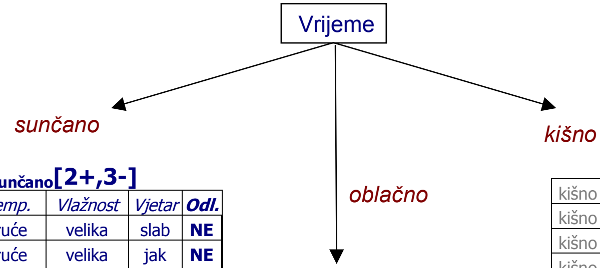
Najve•a informacijska dobit od sva •etiri mogu•a atributa pa •e atribut Vrijeme biti korijen stabla!

ID3 - Korijen stabla je ***Vrijeme***, listovi su vrijednosti tog atributa. Elementi skupa za učenje **S** podjele se u tri grupe ($\mathbf{S}_{\mathbf{sunčano}}$, $\mathbf{S}_{\mathbf{oblačno}}$ i $\mathbf{S}_{\mathbf{kišno}}$) prema vrijednostima atributa ***Vrijeme*** (sunčano, oblačno kišno).

Za svaki takav podskup $\mathbf{S}_{\mathbf{sunčano}}$, $\mathbf{S}_{\mathbf{oblačno}}$ i $\mathbf{S}_{\mathbf{kišno}}$ ponavlja se isti postupak.

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

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



$\mathbf{S}_{\mathbf{sunčano}}$**[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**

$\mathbf{S}_{\mathbf{kišno}}$

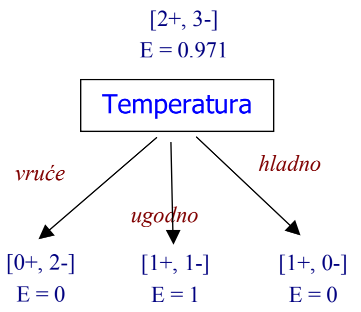
kišno	ugodno	velika	slab	DA
kišno	hladno	normalna	slab	DA
kišno	hladno	normalna	jak	NE
kišno	ugodno	normalna	slab	DA
kišno	ugodno	velika	jak	NE

$\mathbf{S}_{\mathbf{oblačno}}$

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

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

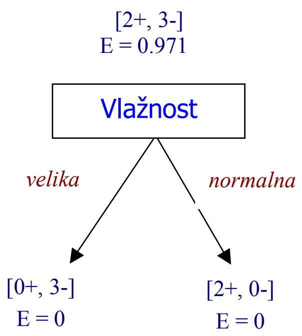
Unutar grane *sunčano* računamo informacijske dobiti za tri atributa, Temperatura, Vlažnost i Vjetar:

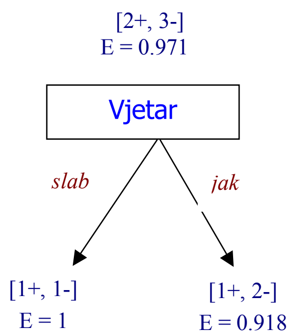


$S_{sunčano}$[2+,3-] označimo kao S

	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(S, **Temperatura**)
= 0.971 - (2/5) 0 - (2/5) 1 -(1/5) 0
= **0.4**

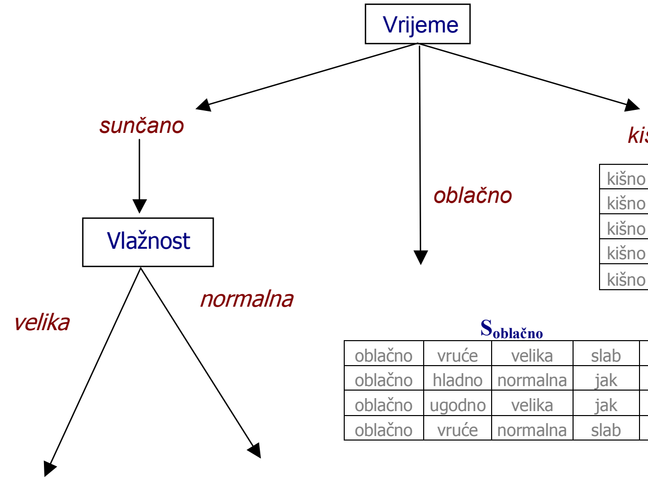




Informacijska_dobit(S, **Vlažnost**)
= 0.971 - (3/5) 0 - (2/5) 0
= **0.971**

Informacijska_dobit(S, **Vjetar**)
= 0.971 - (2/5) 1 - (3/5) 0.918
= **0.02**

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 $\mathbf{S}_{\mathbf{sunčano}}$ niz grane *normalna* (skup $\mathbf{S}_{\mathbf{sunčano,\ normalna}}$) i *velika* (skup $\mathbf{S}_{\mathbf{sunčano,\ velika}}$).



$S_{kišno}$

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

$S_{oblačno}$

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

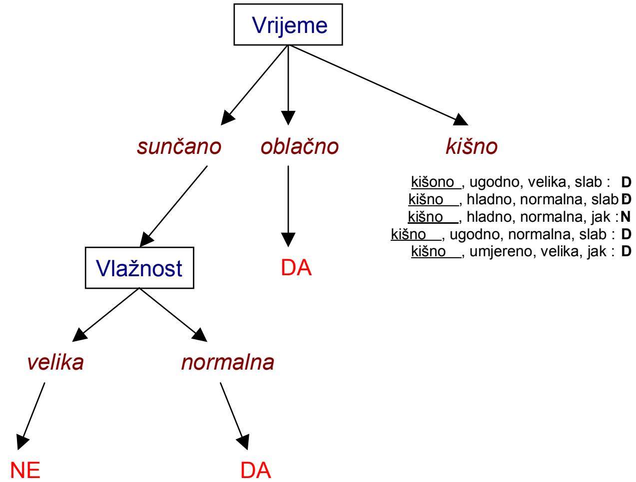
Vlažnost	*Vjetar*	***Igra***
velika	slab	**NE**
velika	jak	**NE**
velika	slab	**NE**

Zaustavljamo se – svi su primjeri iz iste klase – "NE"

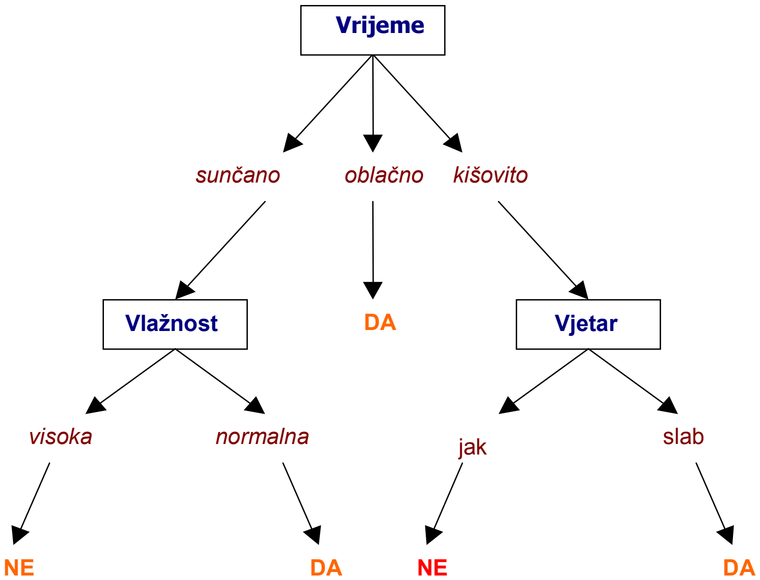
Vlažnost	*Vjetar*	***Igra***
normalna	slab	**DA**
normalna	jak	**DA**

Zaustavljamo se – svi su primjeri iz iste klase – "DA"

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



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



ID3(*Primjeri, Ciljni_atribut, Atributi*)

Primjeri su uzorci za učenje. 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 korektno klasificira dane primjere.

- Kreiraj 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 *Primjeri*
- Inače započni
  - A $\leftarrow$ atribut iz skupa *Atributa* koji najbolje klasificira *Primjere* (tj. ima najveću informacijsku dobit)
  - Atribut za odluku u korijenu je A tj. ROOT $\leftarrow$ A
  - Za svaku moguću vrijednost v_i od A,
    - Dodaj novu granu stabla ispod korijena ROOT, koja odgovara testu A = v_i
    - Neka *Primjeri*$_{v_i}$ označava podskup skupa *Primjeri* koji imaju vrijednost v_i za atribut A
    - Ako je skup *Primjeri*$_{v_i}$ prazan
      - Ispod nove grane dodaj završni čvor (list) čija je oznaka = najčešće pojavljivanoj vrijednosti atributa *Ciljni_atribut* u skupu *Primjeri*
      - Inače ispod nove grane dodaj stablo ID3(*Primjeri*$_{v_i}$, *Ciljni_atribut, Atributi – {A}*)
- Kraj
- Vrati ROOT

Generalizirani algoritam

Općenit slučaj je kada imamo N primjera razdijeljenih u skupove koji pripadaju razredima c_i , i=1, 2, 3, ..., C. Broj primjera u razredu c_i je N_i. Svaki primjer ima K atributa, a svaki atribut J_K vrijednosti. (Radi jednostavnosti, pretpostavit ćemo da svi atributi imaju J vrijednosti.) ID3 postupak za sintezu efektivnog stabla odluke je slijedeći:

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

$$\text{Entropija}(S) = \sum_{i=1}^{C} -\left(\frac{N_i}{N}\right)\log_2\left(\frac{N_i}{N}\right) = \sum_{i=1}^{C} -p_i \log_2 p_i \,.$$

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

a. Za svaki atribut A_k, k=1, 2, 3, ..., K, razdijeli originalni skup primjera na prvorazinske skupove prema vrijednostima a_{kj} od mogućih J vrijednosti atributa A_k. Postoji n_{kj} primjera u a_{kj} grani, ali ti uzorci ne moraju nužno biti iz jednog razreda.

b. Za svaki podskup grane n_{kj}, broj primjera koji pripadaju razredu c_i je $n_{kj}(i)$. Izračunati entropiju te grane koristeći relaciju

$$\text{Entropija}(S, A_k, j) = \sum_{i=1}^{C} -\left(\frac{n_{kj}(i)}{n_{kj}}\right)\log_2\left(\frac{n_{kj}(i)}{n_{kj}}\right)$$

Entropija sustava nakon testiranja atributa A_k je

$$\text{Entropija}(S, A_k) = \sum_{j=1}^{J}\sum_{i=1}^{C}\left(\frac{n_{kj}}{\sum_j n_{kj}}\right)\cdot\left[-\left(\frac{n_{kj}(i)}{n_{kj}}\right)\log_2\left(\frac{n_{kj}(i)}{n_{kj}}\right)\right]$$

c. Pad entropije (tj. informacijska dobit) kao rezultat testiranja atributa A_k je

$$\text{Inforcijska\_dobit}(k) = \text{Entropija}(S) - \text{Entropija}(S, A_k)$$

d. Izabrati atribut A_{ko} koji rezultira najvećom informacijskom dobiti, tj. za koju je $\text{Informacijska\_dobit}(k_0) > \text{Informacijska\_dobit}(k)$ za svaki k=1, 2, 3, ..., K, $k \neq k_0$.

e. Atribut A_{ko} postaje korijen stabla odluke.

Korak 3. *Izgraditi sljedeću razinu stabla odluke.* Izabrati atribut $A_{k'}$, koji će služiti kao prvorazinski čvor, takav da nakon testiranja $A_{k'}$ 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.

3.5. Pretraživanje prostora hipoteza u učenju stablu odluke.

Induktivne metode učenja:
Pretraživanje prostora hipoteza za onom koja najbolje odgovara primjerima za učenje.

Kakav prostor hipoteza pretražuje ID3?

Pretražuje se prostor svih mogućih stabala odluke, počevši 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.

- **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 ID3 izbjegava zamku pretraživanja nepotpunog prostora hipoteza koji ne sadrži ciljni koncept (npr. u slučaju kada su hipoteze u obliku konjunkcije atributa)

- **ID3 pronalazi samo jednu hipotezu**

  CA algoritam nalazi sve hipoteze konzistentne s primjerima. Ne znamo koliko je još stabala odluke konzistentno s primjerima za učenje, niti može učenik postaviti upit o primjeru koji će onda razriješiti između mogućih hipoteza.

- **ID3 u izvornom obliku se ne vraća unatrag u postupku pretraživanja**

  To svojstvo ima isti nedostatak kao i pretraga na vrh – mogućnost da se zaglavi u lokalnom optimumu.
  Lokalni optimum može biti manje dobar od nekog drugog rješenja koje bi možda mogli naći pretraživanjem niz neku drugu granu.

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

  Prednost uporabe statističkog svojstva svih primjera za učenje (tj. informacijske dobiti) je manja osjetljivost na pogreške u skupu primjera za učenje. CA i FS algoritmi donose odluke u koracima (inkrementalno) na temelju jednog predočenog primjera.

3.5. Induktivna pristranost ID3 algoritma

Na temelju čega ID3 može generalizirati i klasificirati još neviđene primjere?

(Induktivna prostranost je skup pretpostavki tako da skupa sa primjerima za učenje deduktivno potvrđuju klasifikaciju koju određuje učenik na novom primjeru)

(ID3 - metoda »uspona na vrh» - prihvaćanje prve odgovarajuće hipoteze)

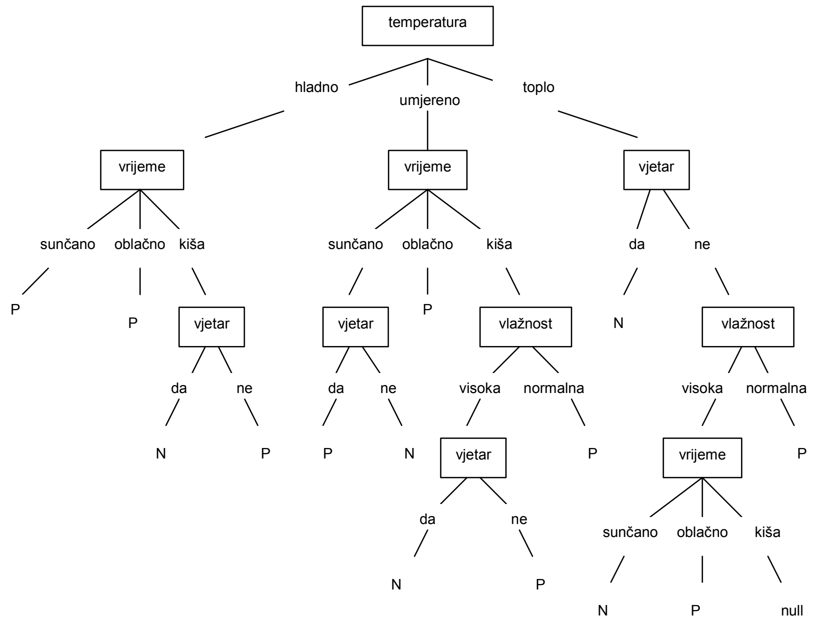
Induktivna pristranost ID3: Na temelju čega ID3 preferira jednu konzistentnu hipotezu u odnosu na drugu?

(a) ID3 izabire kraće stablo prije nego dulje stablo
(b) Izabire stablo koje stavlja atribute s većom informacijskom dobiti bliže korijenu

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

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

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



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 dizajniran upravo za drugi dio spektra, za koji je karakteristično mnogo atributa i gdje se skup za učenje sastoji od mnogo 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.

3.6.1. Pristranosti restrikcijom i pristranosti preferencijom

Različiti tipovi induktivne pristranosti

	ID3	**Eliminacija-kandidata (CA)**
Prostor hipoteza *koji se pretražuje*	Potpun	Nepotpun (onaj koji može izraziti)
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	<u>Preferencija</u> nekih hipoteza nad drugima	<u>Restrikcija</u> skupa razmatranih hipoteza
	Pristranost preferencijom ili pristranost pretraživanja *(engl. preference bias, search bias)*	**Pristranost restrikcijom ili pristranost jezika** *(engl. preference bias, search bias)*

Koja je pristranost općenito poželjnija?

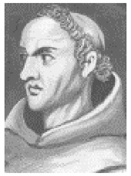
Neki sustavi strojnog učenja kombiniraju ove dvije vrste pristranosti.

Primjer sustav koji uči igrati igru DAME:

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

3.6.2 Zašto preferirati kraće hipoteze ?

Filozofsko pitanje



1320.g. William of Occam

"Pluralitas non est ponenda sine neccesitate"
(Množina ne treba biti postavljena bez potrebe)

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.

Occamova britva:
Preferirati jednostavnije hipoteze koje odgovaraju podacima

Zašto?

Obično ima manje jednostavnijih hipoteza od složenijih.

Primjer: stabla odluke, stablo s 5 čvorova se preferira u odnosu na stablo s 500 čvorova - manja vjerojatnost da ćemo naći manje stablo koje odgovara nego veće.

Problem kod takvog objašnjenja: možemo definirati neke druge manje skupove hipoteza i njih preferirati! (primjer: preferiramo stabla koja atribut A1 u korijenu, a zatim testiraju A3 i imaju 17 čvorova i 11 završnih listova.

Drugi problem. Veličina hipoteza je određena internom reprezentacijom koju učenik koristi.
Dva učenika s različitim reprezentacijama hipoteza i istim skupom za učenje mogu doći do različitih hipoteza uz Occamovu britvu!

3.7 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)

3.7.1. Izbjegavanje prekomjerne naučenosti podataka

Algoritam ID3 – rast stabla dok se svi podaci pavilno ne klasificiraju
To je problem ako su:

- podaci sa šumom
- skup za učenje je premalen.

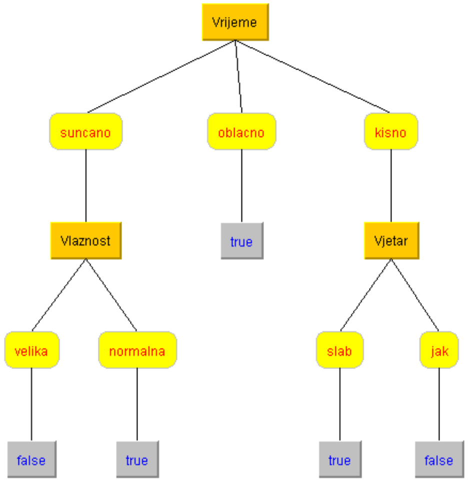
Tada može doći do prenaučenosti (*engl. overfit*) stabla odluke.

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.

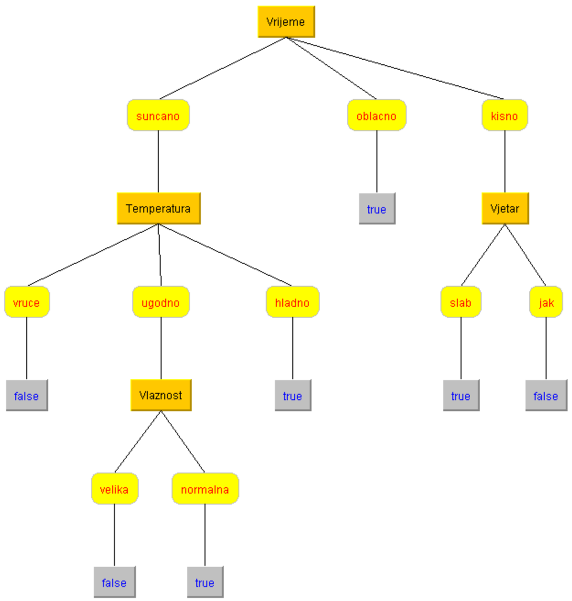
Primjer prenaučenosti
Pretpostavimo da je dodan 15. primjer u skup primjera koji je pogrešno klasificiran kao Igra = NE umjesto Igra = DA.

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



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



Postojeće stablo krivo klasificira 15.-ti primjer i zbog toga ID3 traži novi razradu stabla odluke (h) koje je složenije od izvornog (h'). Novo stablo odluke ima dodatni čvor koji je posljedica pogreške u podacima.
Hipoteza h će savršeno odgovarati svim primjerima za učenje (njih 15), dok h' neće.

Prenaučenost je moguća i kada nema pogreške, ali kada je broj primjera pridružen nekom listu premali. (Kada neki atribut dobro razdjeljuje primjere iako nije značajan za ciljnu dunkciju)

Stopa smanjenja točnosti ID3 zbog prenaučenasti 10-25%

Izbjegavanje prenaučenosti – dva pristupa:

- zaustavljanje rasta stabla prije savršene klasifikacije primjera za učenje
- naknadno podrezivanje prenaučenog stabla

uspješniji pristup u primjeni

Kako odrediti razumnu veličinu stabla?

- uvođenjem posebnog skupa podataka za vrednovanje najčešće u primjeni,
  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.

- uporaba statističkih testova (testiranja da 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 mjre kompleksnosti kodiranja primjera za učenje i stabla odluke i zaustavljanja kada je ta mjera minimalna. Primjer: Princip minimuma opisa (*engl. minimum description principle*)

3.7.1.1 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 ne ponaša lošije na skupu za vrednovanje.

Na taj se način uklanjaju čvorovi dodani zbog slučajnih neregularnosti u skupu za učenje kojih nema u skupu za vrednovanje.

Uklanjanje je iterativni 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.

3.7.1.2 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 ekvivalentni 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.

Primjer: Naj lijevija grana stabla odluke (Quinlan-ov 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*) $\neq$ 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.

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 – uklanjaju se prisutnost tog uvjeta (čvora) u svim pravilima (u kojima se pojavljuje na lijevoj strani), istodobno.

2. Uklanja se razlika između testiranja atributa koji se nalaze na dnu stabla (blizu listu) ili pri vrhu (korijenu).

3. Pravila povećavaju čitljivost, razumljivost.

3.7.2 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
$c \in$ domene(A)
Algoritam definira novi boolov atribut $\mathbf{A_c}$ takav da je
$\mathbf{A_c}$ **istinit** ako vrijednost(A) < c inače, $\mathbf{A_c}$ **lažan.**

Kako odabrati najbolju vrijednost za c?

Primjer

Pretpostavimo da primjeri za učenje pridruženi nekom čvoru imaju slijedeće 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.

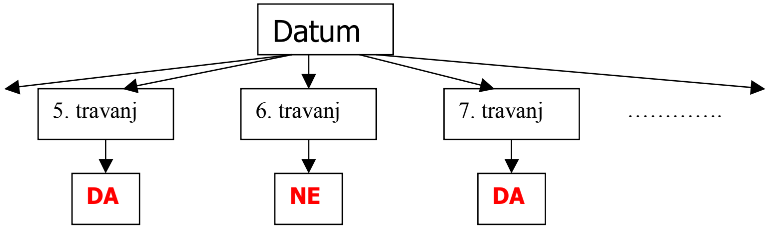
1.	Vrijednosti atributa A slože se u rastućem redoslijedu.	*već učinjeno u tablici*
2.	Odrede se one susjedne vrijednosti atributa koje se razlikuju u klasifikaciji ciljnog atributa.	(48, 68) i (80, 90)
3.	Nađe se srednja vrijednost takvih vrijednosti atributa. Te srednje vrijednosti čine kandidate za graničnu vrijednost c	C=(48+68)/2=54 C=(80+90)/2=85
4.	Računa se informacijska dobit za svaki takav kandidat za graničnu vrijednost c	$I(Temperatura_{>54})$ $I(Temperatura_{>85})$
5.	Odabire se c s najvećom vrijednošću $I(Temperatura_{>c})$	$I(Temperatura_{>54})$

3.7.3 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.

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.

$$Informacijska\_podijeljenost(S,A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

S_1 ,…, S_c su podskupovi skupa primjera S koji nastaju particijom S s obzirom na vrijednost atributa A. (*Entropija u odnosu na vrijednosti atributa*)

Zadatak

Kolika je informacijska podijeljenost atributa:

(a) koji uniformno distribuira vrijednosti poput atributa *Datum* ?

(b) Boolovog atributa koji dijeli n primjera točno na pola?

Odgovor:

(a) $\log_2 n$

(b) 1

Omjer dobitka se definira

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

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

Što ako je nazivnik blizu 0?

Za $|S_i| \approx |S|$, 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 *Datuma*)

3.7.4. Nedostajuće vrijednosti

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

Primjer:

A=vjetar

	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-]

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 sa c(x) u tom čvoru.

2. slab, 2. slab

Drugi pristup:

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

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*

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

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

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

izračunato na temelju 13 primjera

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

$$\text{Informacijska\_dobit}(S,A) \equiv \text{Entropija}(S) - \sum_{v \in \text{Vrijednost}(A)} \frac{|S_v|}{|S|}\text{Entropija}(S_v)$$

Najprije računamo entropije skupova S, S_{slab} , S_{jak}.

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

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

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

Informacijska_dobit(**S**, **Vjetar**) ≡

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

≡ 0.940 – (8/13)0.811 – (5/13)0.970 =

≡ **0.06784**