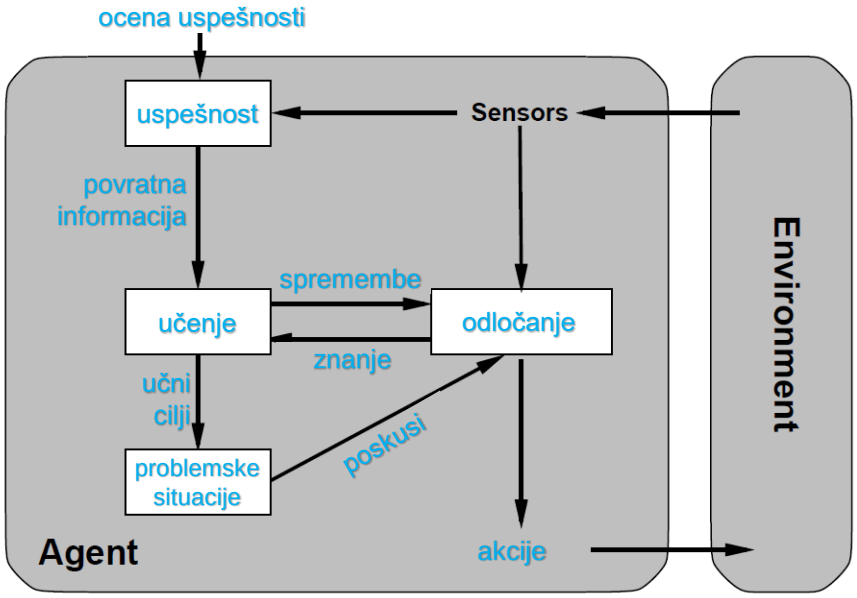
OSNOVE UMETNE INTELIGENCE

uvod v strojno učenje

učenje odločitvenih dreves

Strojno učenje

- angl. *machine learning*
- inteligentni agent se **uči**, če z opazovanjem okolja (z "izkušnjami") postaja **bolj učinkovit** pri prihodnjih nalogah
- zakaj narediti učečega se agenta in ne ga takoj naučiti vsega?
  - razvijalci programske opreme ne morejo predvideti vseh možnih situacij (različne problemske situacije),
  - razvijalci ne morejo predvideti sprememb okolja skozi čas (prilagodljivost),
  - razvijalci ne znajo sprogramirati agenta z znanjem (npr. razpoznava obrazov?)



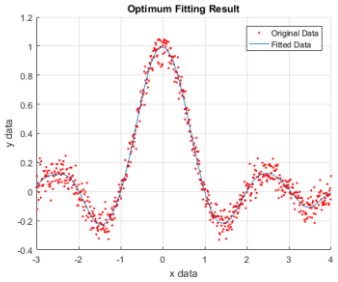
Vrste učenja

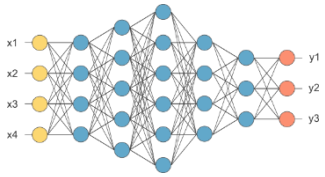
- induktivno učenje: učenje, pri katerem iščemo posplošeno *funkcijo*, ki opisuje množice vhodnih podatkov (učenje iz primerov)
  - učni primeri, atributi, ciljna spremenljivka
- učenje z odkrivanjem (*learning by discovery*): agent izvaja poskuse, zbira podatke, formulira problem, posplošuje podatke

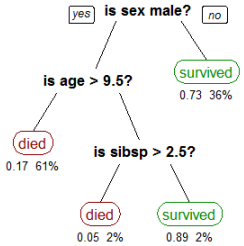
- **nadzorovano učenje** (angl. *supervised learning*):
  učni primeri so podani kot vrednosti vhodov in izhodov (učni primeri so označeni); učimo se funkcije, ki preslika vhode v izhode (npr. odločitveno drevo)
- **nenadzorovano učenje** (angl. *unsupervised learning*):
  učni primeri niso označeni (nimajo ciljne spremenljivke); učimo se vzorcev v podatkih (npr. gručenje)
- **spodbujevano učenje** (angl. *reinforcement learning*):
  inteligentni agent se uči iz zaporedja nagrad in kazni

Primeri

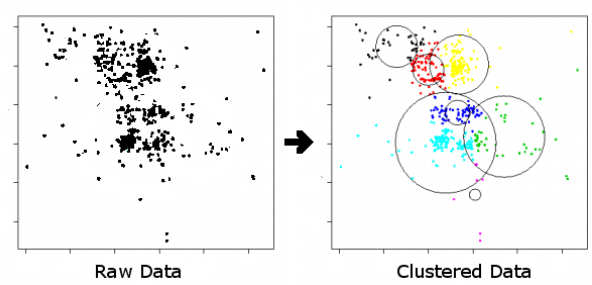
- nadzorovano učenje:

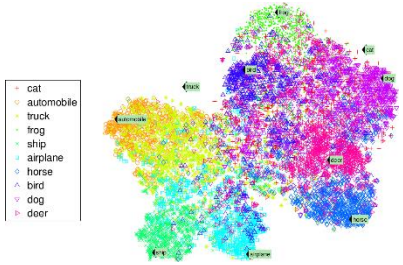




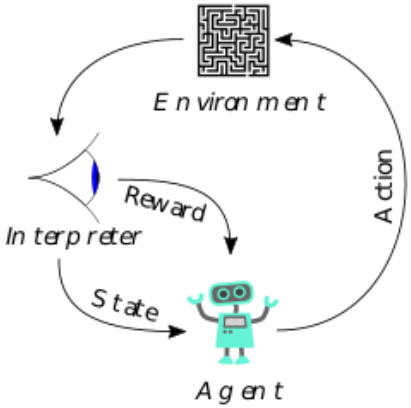


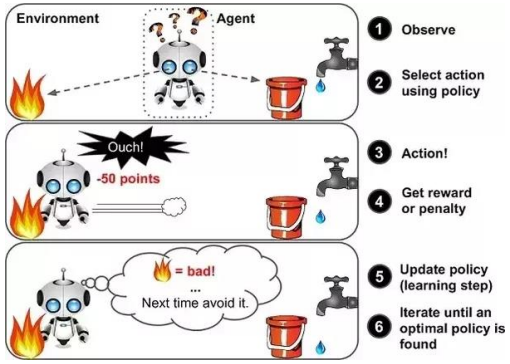
- nenadzorovano učenje:





- spodbujevano učenje:

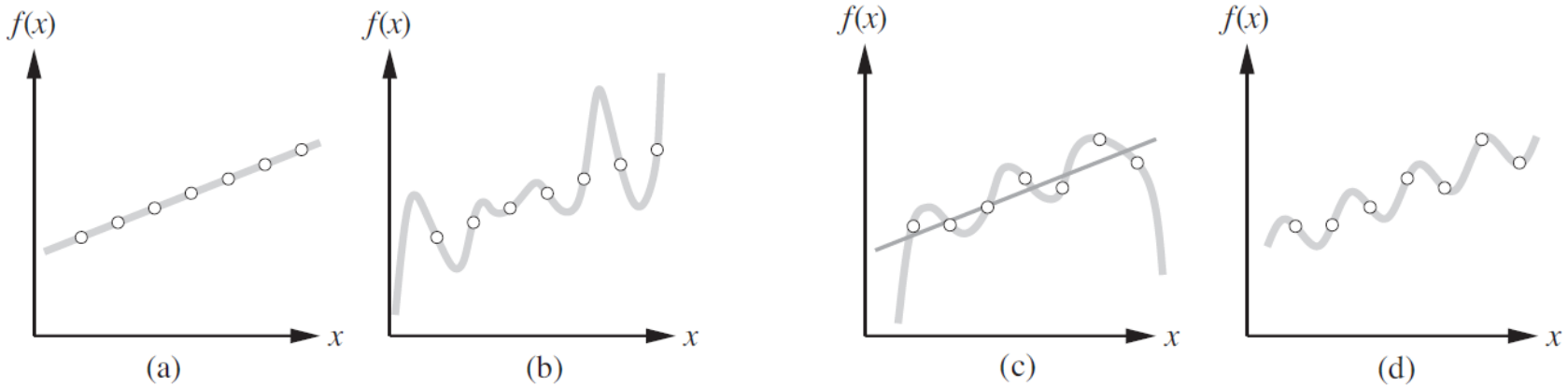




Nadzorovano učenje

- **podana**: množica **učnih primerov**
  $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$,
  kjer je vsak y_j vrednost neznane funkcije $y = f(x)$
- **naloga**: najdi funkcijo h, ki je najboljši približek funkciji f

- x_j so **atributi** (vrednost ali vektor)
- funkcijo h imenujemo **hipoteza**
- primeri hipotez skozi dve množici točk:

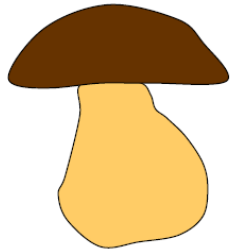


Atributna predstavitev podatkov

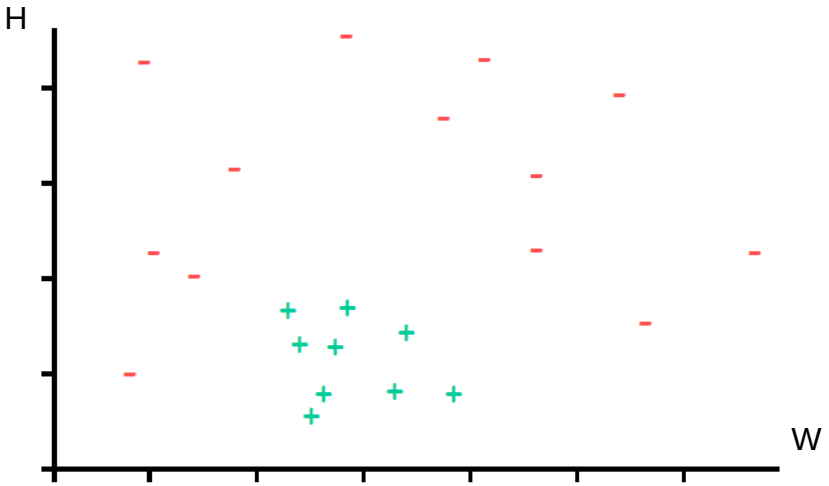
- učna množica: čakanje na prosto mesto v restavraciji
- ciljna spremenljivka: čakamo (T) ali ne čakamo (F)

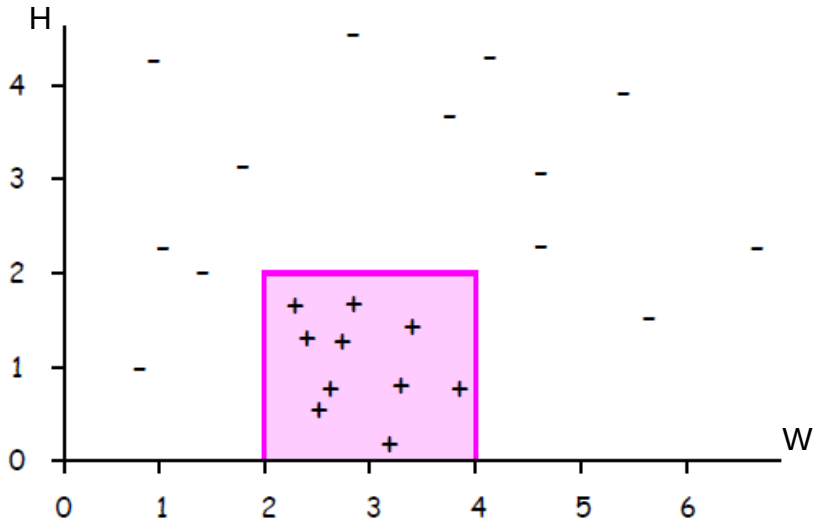
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*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*

Primer: gobe



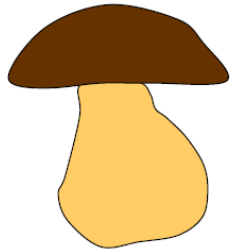
- razpoznavanje užitnih gob
- atributa (x): W (width) in H (height)
- razred (y): strupena (-), užitna (+)



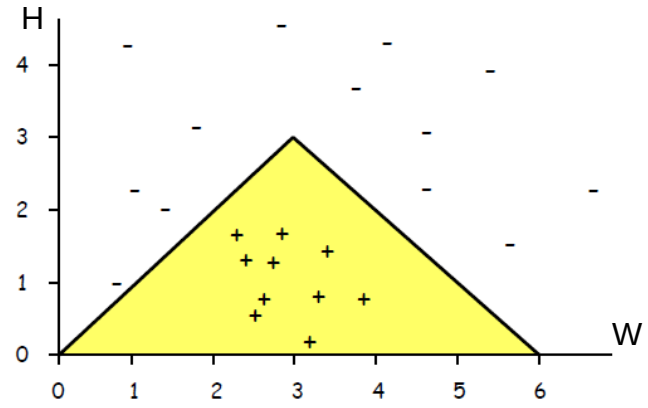



```
IF W>2 and W<4 and H<2
THEN "edible" ELSE "poisonous"
```

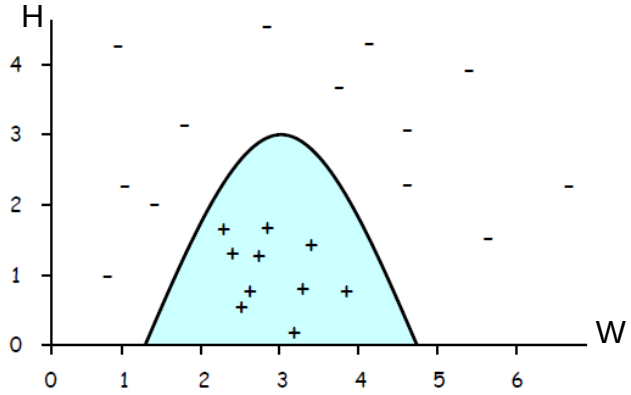
Primer: gobe



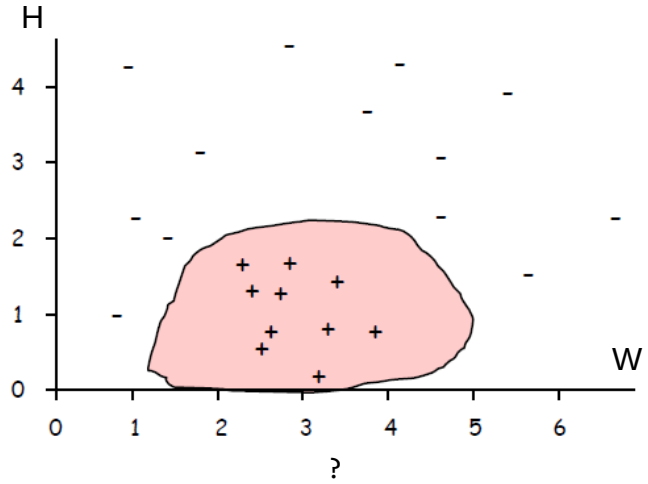
- ali pa …

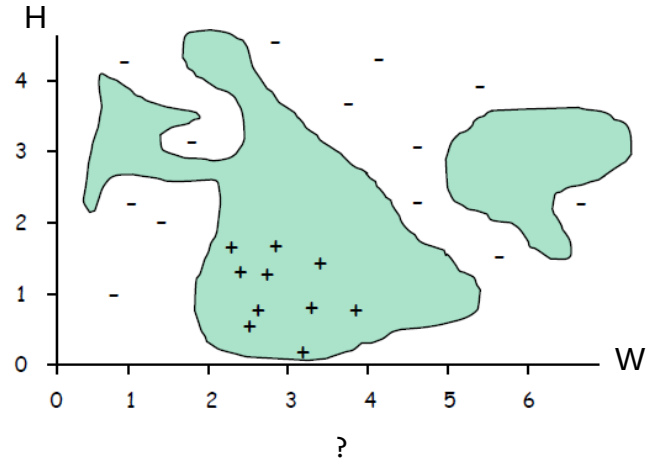



```
IF H > W THEN "poisonous"
ELSE IF H > 6 – W THEN "poisonous" ELSE "edible"
```



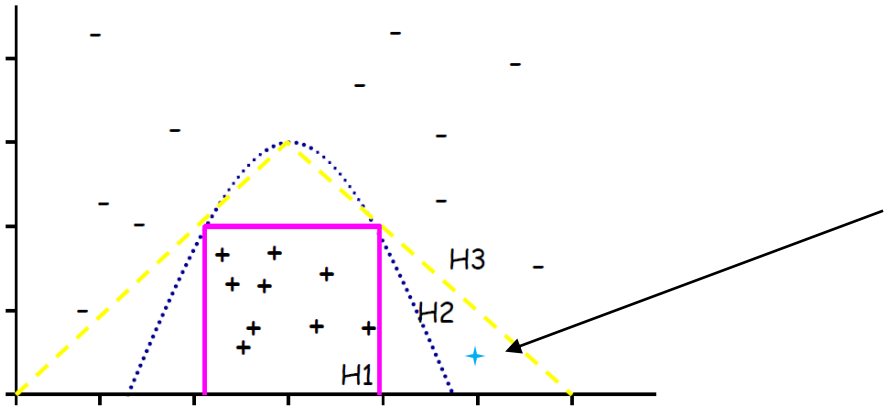

```
IF H < 3 – (W-3)² THEN "edible"
ELSE "poisonous"
```





Primer: gobe

- prostor hipotez vsebuje več hipotez
- vse prikazane hipoteze so **konsistentne** z učno množico
- dobra hipoteza je dovolj splošna (**general**), kar pomeni, da pravilno napoveduje vrednost y za nove (še nevidene) primere

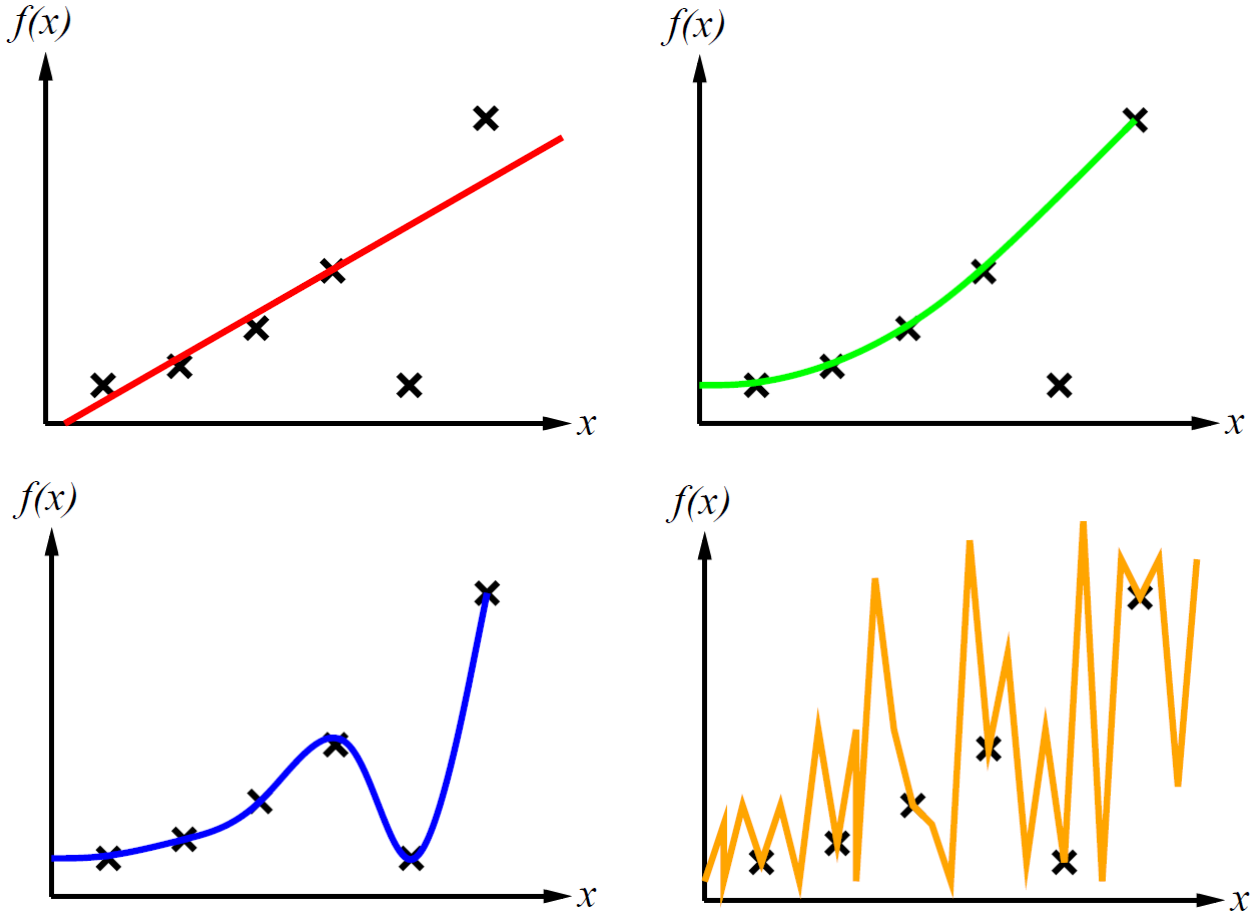


kam klasificirati ta primer?
(glede na H1 in H2 je -, glede na H3 je +)

- kako izbrati primerno hipotezo? Princip **Ockhamove britve** (*Ockham's razor*) (William o Ockham, 1320, angleški filozof):
  - prava hipoteza je najbolj preprosta hipoteza
  - *Entities should not be multiplied unnecessarily*
  - *Given two explanations of the data, all other things being equal, the simpler explanation is preferrable.*

Primer

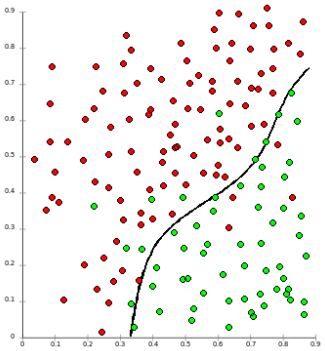
- podoben problem je tudi pri drugačnih primerih (iskanje funkcije, ki opisuje podane točke)

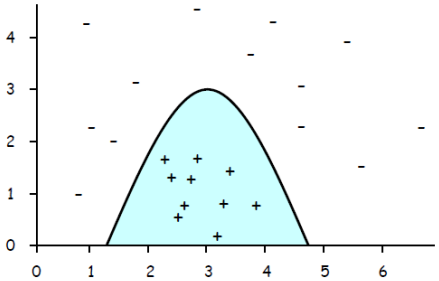


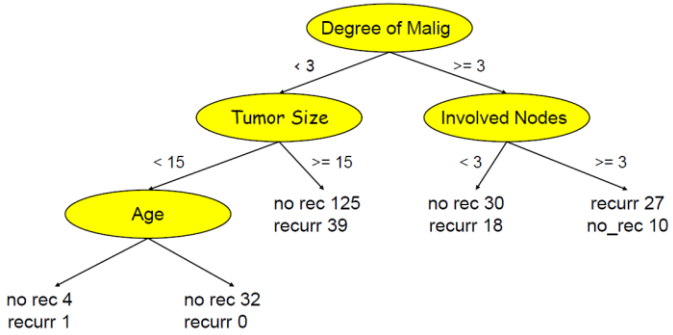
Vrste problemov

- **klasifikacija** in **regresija**

- **klasifikacija:**
  - y pripada **končnem naboru vrednosti** (je diskretna spremenljivka)
  - npr. $y \in \{užitna, strupena\}$, $y \in \{sonce, oblačno, dež\}$, $y \in \{zdrav, bolan\}$
  - y imenujemo **razred** *(angl. class)*
  - primeri:
    - napovedovanje vremena iz podatkov prejšnjih let
    - diagnosticiranje novih pacientov na osnovi znanih diagnoz za stare paciente
    - klasifikacija neželene elektronske pošte
    - napovedovanje vračila kredita



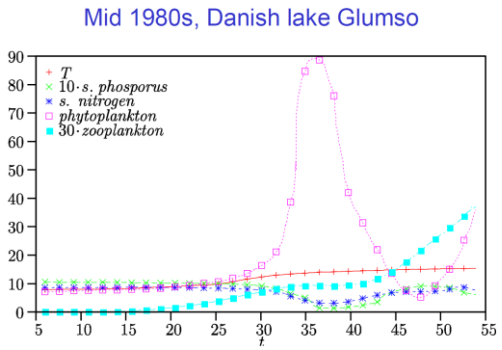




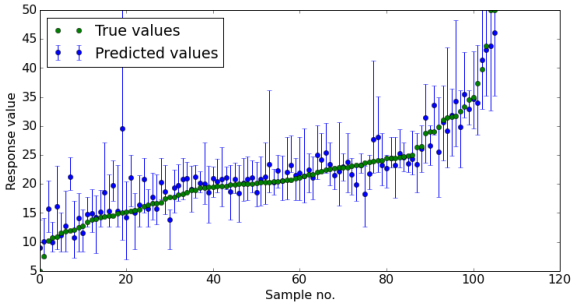
drevo, zgrajeno iz 139 učnih primerov; višja klasifikacijska točnost kot zdravniška

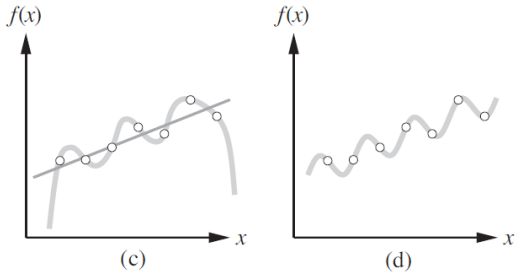
Vrste problemov

- **regresija:**
  - y je število (običajno $y \in \mathbb{R}$, je zvezna spremenljivka)
  - npr. y je temperatura,
  - y imenujemo **označba** (angl. *label*)
  - primeri:
    - napovedovanje razmnoževanja alg
    - medicinska diagnostika
    - napovedovanje vremena
    - napovedovanje koncentracije ozona
    - napovedovanje gibanja cen delnic

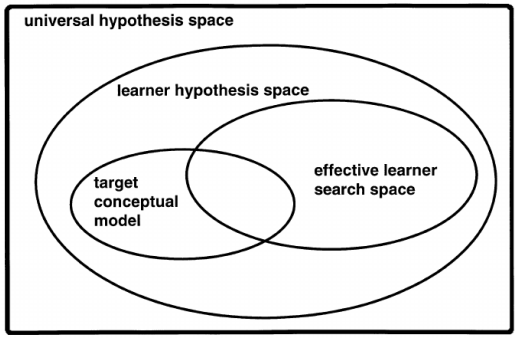


zakonitosti razmnoževanja alg





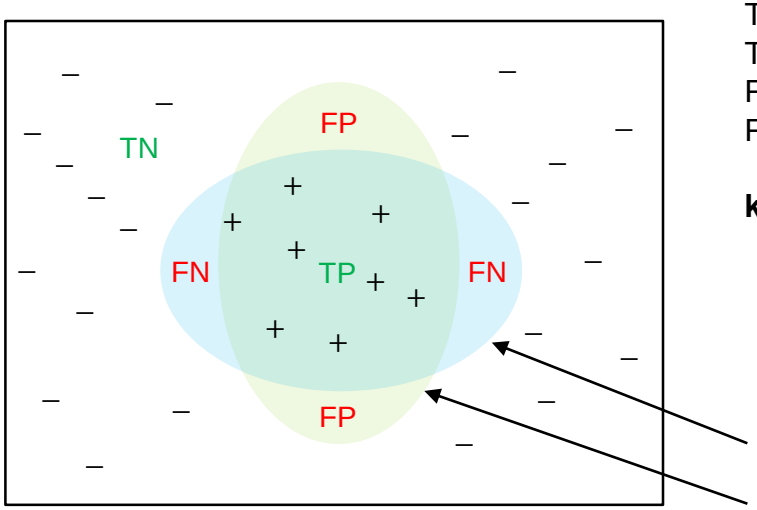
Prostor hipotez



- denimo, da imamo
  - binarno klasifikacijo
  - n binarnih atributov
- sledi:
  - → 2^n različnih učnih primerov
  - → 2^{2^n} hipotez (denimo, da lahko hipotezo opišemo s tabelo napovedi za vse primere)
- primer:
  - za 10 atributov izbiramo med 10^{308} možnimi hipotezami
  - za 20 atirbutov izbiramo med $10^{300.000}$ možnimi hipotezami
  - v resnici: hipotez je še več, izračunavajo lahko isto funkcijo
- potrebujemo:
  - zavedanje o pristranosti hipotez
  - algoritme za gradnjo "dobrih" hipotez
  - metode za ocenjevanje hipotez / ocenjevanje učenja

Evalviranje hipotez

- pomembni kriteriji:
  - **konsistentnost** hipotez s primeri
  - **razumljivost** (interpretability, comprehensibility) hipotez
  - **točnost** hipotez:
    - točnost na učnih podatkih? (pristranost hipotez?)
    - točnost na novih podatkih?
    - točnost na testnih podatkih?
- ocenjevanje uspešnosti pri klasifikaciji:



TP – pravilno pozitivno klasificirani primeri (angl. *true positive*)
TN – pravilno negativno klasificirani primeri (angl. *true negative*)
FP – napačno pozitivno klasificirani primeri (angl. *false positive*)
FN – napačno negativno klasificirani primeri (angl. *false negative*)

klasifikacijska točnost (angl. classification accuracy):

$$CA = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{N}$$

Izpitna naloga

- 1. izpitni rok, 30. 1. 2018

3. NALOGA:

Podan je primer učenja iz primerov z atributoma A in B ter razredom C. Atribut A in razred sta binarna, atribut B pa lahko zavzame tri vrednosti. Učno množico primerov, ki smo jih zajeli in v katerih ni šuma, prikazuje tabela na desni. Denimo, da vemo, da pravo odvisnost med atributi in razredom izraža funkcija $C = IF\ (AB^2) < (A + B)\ THEN\ 1\ ELSE\ 0$. Istočasno pa se z dvema različnima algoritmoma za učenje iz primerov naučimo naslednjih dveh hipotez:

H1: $C = A + 1{,}5 \cdot B - AB - 0{,}5 \cdot B^2$

H2: $C = \min(A + B, 1) - \min(A, B)$

A	B	C
0	0	**0**
0	1	**1**
0	2	**1**
1	0	**1**
1	2	**0**

Odgovori na naslednja vprašanja:

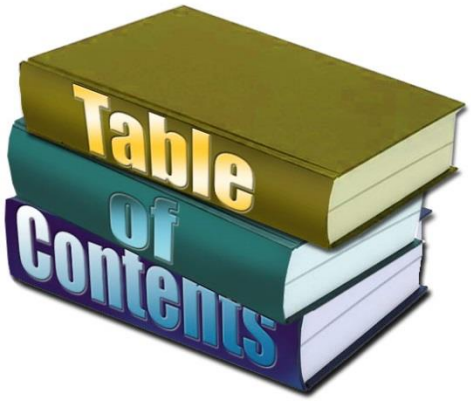
a) Katera od podanih hipotez je bolj splošna? Kaj to pomeni?

b) Katera od podanih hipotez ima višjo klasifikacijsko množico na učni množici?

c) Kakšna je razlika med nazorovanim in nenadzorovanim učenjem? S katerim imamo opravka pri zgornji nalogi? Podaj primer problema nadzorovanega in nenadzorovanega učenja iz prakse.

d) Kaj je to binarizacija atributa in zakaj je koristna? Podaj primer binarizacije atributa B.

Pregled

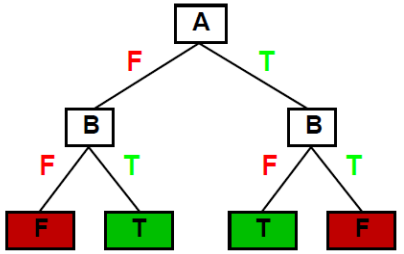


- strojno učenje
  - uvod v strojno učenje
  - učenje odločitvenih dreves

Odločitveno drevo

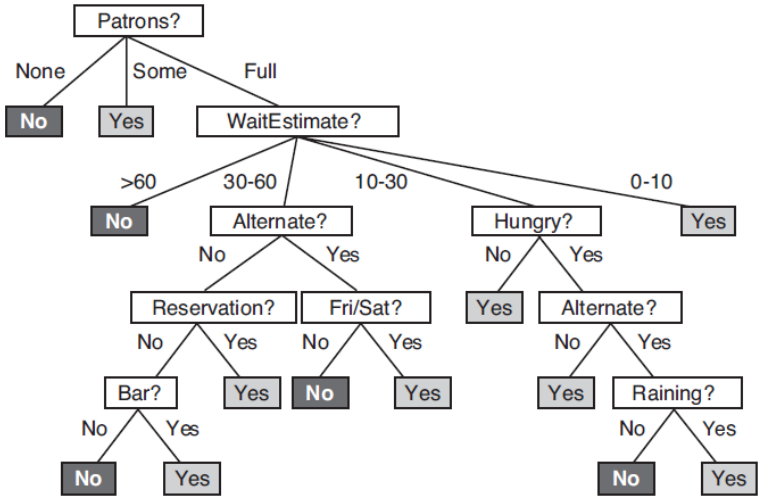
- ponazarja relacijo med vhodnimi vrednostmi (atributi) in odločitvijo (ciljna spremenljivka – razred ali označba)
  - notranja vozlišča: test glede na vrednost posameznega atributa
  - listi: odločitev (vrednost ciljne spremenljivke)
  - pot: konjunkcija pogojev v notranjih vozliščih na poti, ki vodi do lista

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



- poseben primer: binarna klasifikacija (razred ima dve možni vrednosti (npr. pozitivni/negativni, strupen/užiten itd.)

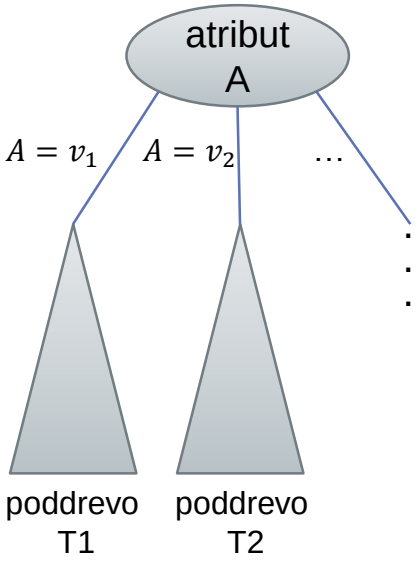
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	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** s strategijo **razveji in omeji**:
  - izberi najbolj pomemben atribut – tisti, ki najbolj odločilno vpliva na klasifikacijo primera – in razdeli primere v poddrevesa glede na njegove vrednosti,
  - rekurzivno ponovi za poddrevesa,
  - če vsi elementi v listu pripadajo istemu razredu ali vozlišča ni možno deliti naprej (ni razpoložljivih atributov), ustavi gradnjo.

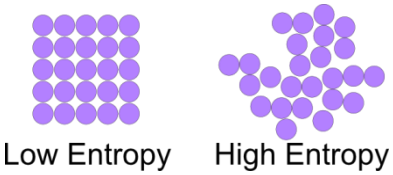


- imenovano tudi ***Top Down Induction of Decision Trees (TDIDT)***
- primeri implementacij: ID3, CART, Assistant, C4.5, C5, …

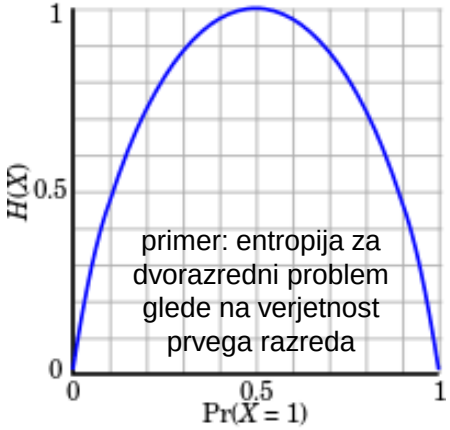
Izbor najbolj pomembnega atributa

- 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



Informacijski prispevek

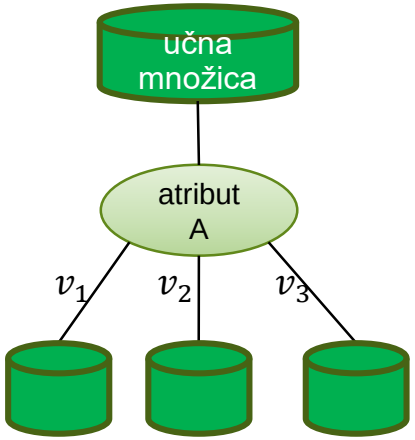
- dejansko nas zanima **znižanje entropije** (nedoločenosti) ob delitvi učne množice glede na vrednosti atributa A
- znižanje entropije ob delitvi učne množice glede na vrednosti atributa A

- **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)

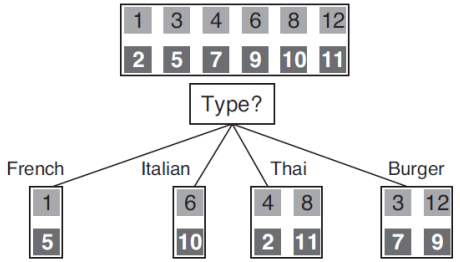
$$I = H(C)$$

rezidualna informacija (entropija)

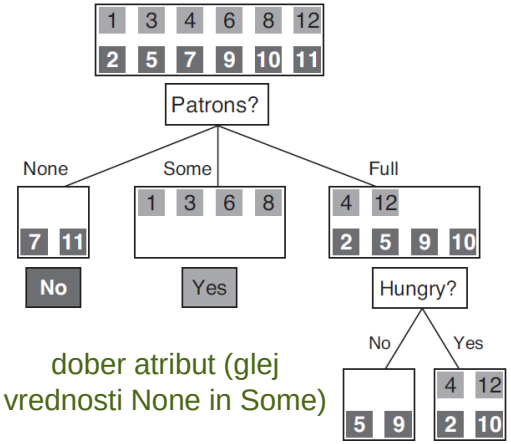
$$I_{res} = \sum_{i} p_{v_i} \cdot H(C|v_i)$$

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

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

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

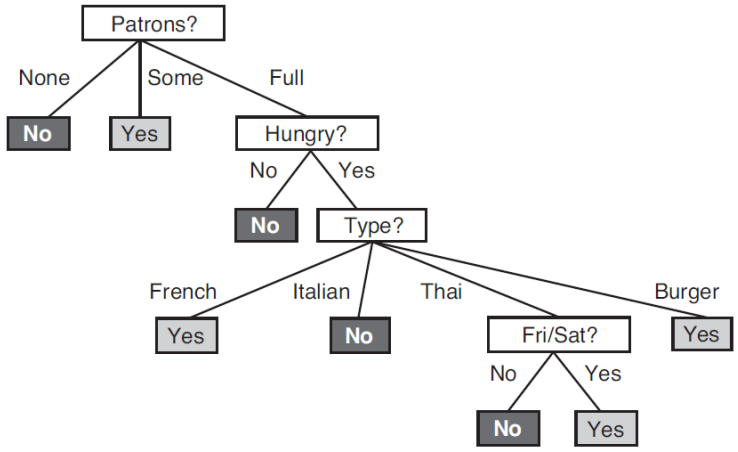
$$I_{res}(Patrons) = -\frac{2}{12}\cdot 0 - \frac{4}{12}\cdot 0 - \frac{6}{12}\left[\frac{2}{6}log_2\frac{2}{6} + \frac{4}{6}log_2\frac{4}{6}\right] \approx 0{,}46$$

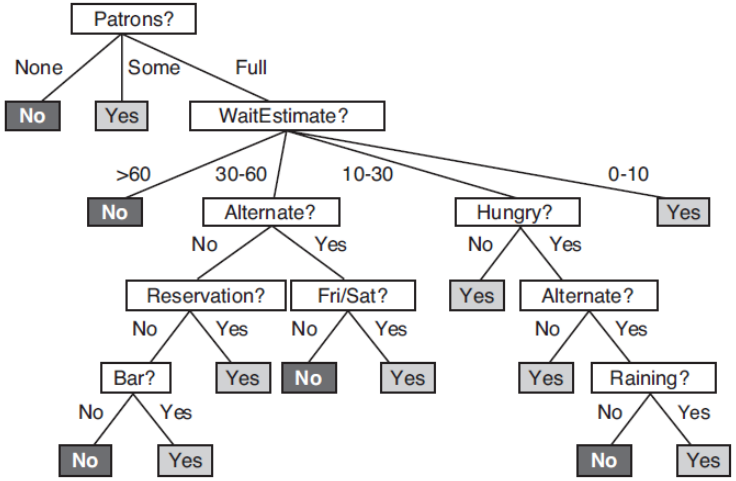
$$Gain(Type) = 1 - 1 = 0$$

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

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 višja na račun večjega števila vrednosti in ne na račun kakovosti atributa)

- rešitve:
  - normalizacija informacijskega prispevka (**relativni informacijski prispevek**)
  - uporaba **alternativnih mer** (informacijskih, ocene verjetnosti itd.)
  - **binarizacija** atributov

Relativni informacijski prispevek in Gini

- **information gain ratio** (sistem ID3, 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)

- **Gini index**
  - ocena pričakovane klasifikacijske napake (vsota produktov verjetnosti razredov)

$$Gini = \sum_{c_1 \neq c_2} p(c_1)p(c_2)$$

$$Gini(A) = \sum_{v} p(v) \sum_{c_1 \neq c_2} p(c_1|v)p(c_2|v)$$

Izpitna naloga

- 2. izpitni rok, 15. 2. 2018 (prilagojena naloga)

Podana je učna množica primerov, ki je prikazana v tabeli (*vreme* in *pritisk* sta atributa, *glavobol* pa je razred). Naloge:

a) Zgradi odločitveno drevo, pri čemer za ocenjevanje atributov uporabi informacijski prispevek. V primeru enakega števila primerov – predstavnikov obeh razredov – naj vozlišče klasificira v večinski razred iz učne množice.
b) Ali bi dobljeno drevo bilo drugačno, če bi uporabili razmerje informacijskega prispevka? Utemelji.
c) V kateri razred bi drevo klasificiralo učni primer z vrednostmi atributov *vreme=deževno, pritisk=srednji*?

vreme	pritisk	glavobol
sončno	nizek	ne
sončno	nizek	ne
sončno	srednji	da
sončno	visok	ne
sončno	nizek	ne
sončno	nizek	da
deževno	srednji	ne
deževno	srednji	da
deževno	visok	da

Binarizacija atributov

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

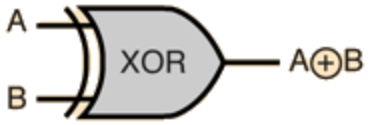
- primer: atribut $barva \in \{rdeča, rumena, zelena, modra\}$
- strategije:
  - $\{\{rdeča\}, \{rumena, zelena, modra\}\}$ (one-vs-all)
  - $\{\{rdeča, rumena\}, \{zelena, modra\}\}$
  - vpeljava binarnih atributov za vsako barvo
  - itd.

- prednost: manjše vejanje drevesa (statistično bolj zanesljivo, možna višja klasifikacijska točnost)
  - različne načine binarizacije 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, rezanje,
šumni podatki