

# Klasszifikátorok

Adott:  $N$  elemű tanítóhalmaz (training set)  $S$  a tanítóhalmaz minden eleméhez egy  $d$  elemű attribútumhalmaz (feature vector,  $X$ ), illetve egy osztályváltozó (class attribute,  $Y$ )

Legyen  $y = f(x)$  egy a tanítóhalmaz attribútumhalmazán értelmezett klasszifikátor, melynek értékkészlete megegyezik az osztályváltozó értékkészletével)

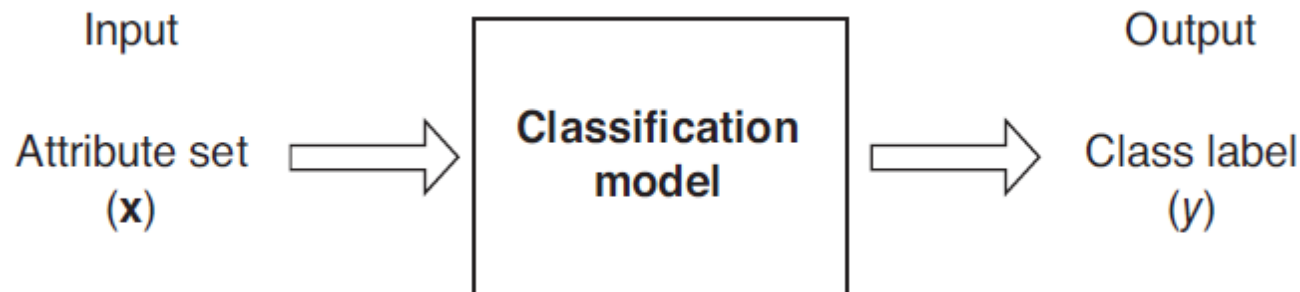
Általában  $x \in \mathbb{R}^d$  és  $y \in \mathbb{Z}$

Keressünk azon  $f(X)$ -et melyre  $E(L(Y, f(X)))$  minimális.  $L(Y, f(X))$  a veszteségfüggvény vagy várható osztályozási hiba. (pl.  $L(y, f(x)) = 0$  ha  $f(x) = y$ , s  $L(y, f(x)) = 1$  ha nem)

# Klasszifikátorok

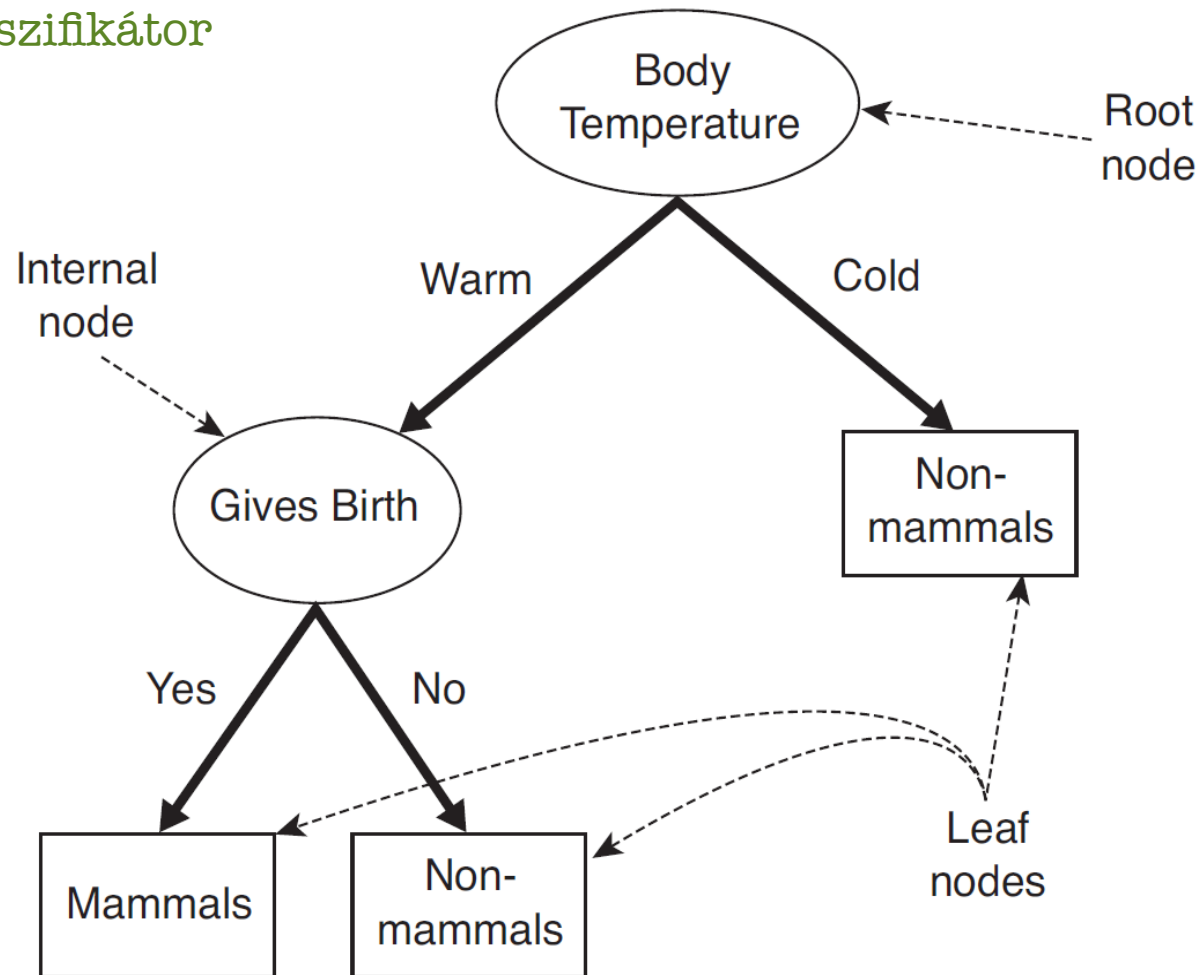
Amennyiben egyosztályos (bináris) klasszifikációról beszélünk, a veszteség leírható egy hibafüggvénnyel. Többosztályos klasszifikáció során az  $L(Y, f(X))$  egy mátrix, melynek  $L(i, j)$  eleme meghatározza a hiba mértékét, ha  $i$  osztály helyett  $j$  osztályt jósoltuk.

A tanulás fontos része a visszajelzés. Ez történhet leíró illetve prediktív jelleggel. A leíró módszerek a tanulóhalmaz struktúráját határozzák meg, míg a jósló módszerek egy teszhalmaz segítségével döntenek a megfelelő modell kiválasztásáról



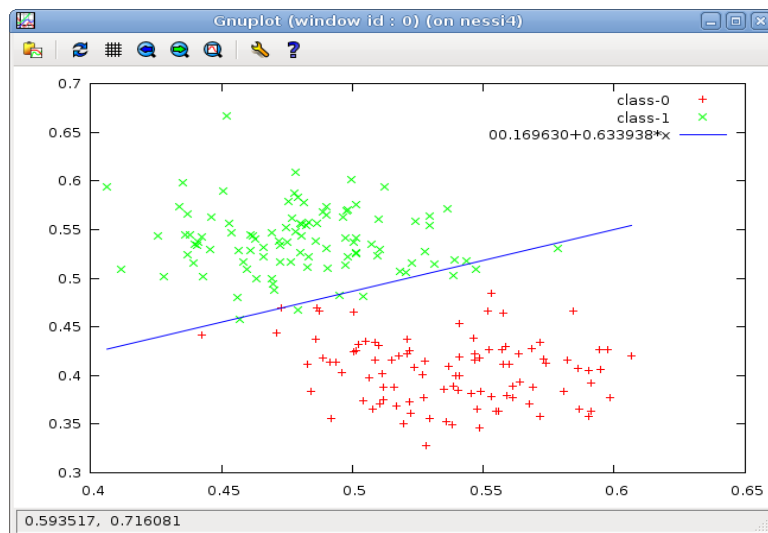
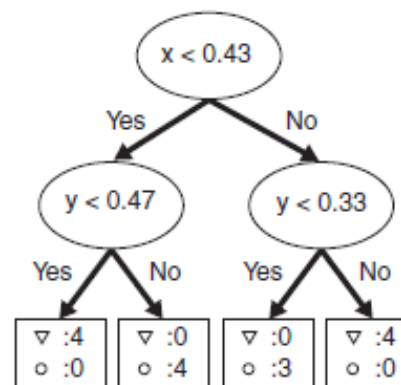
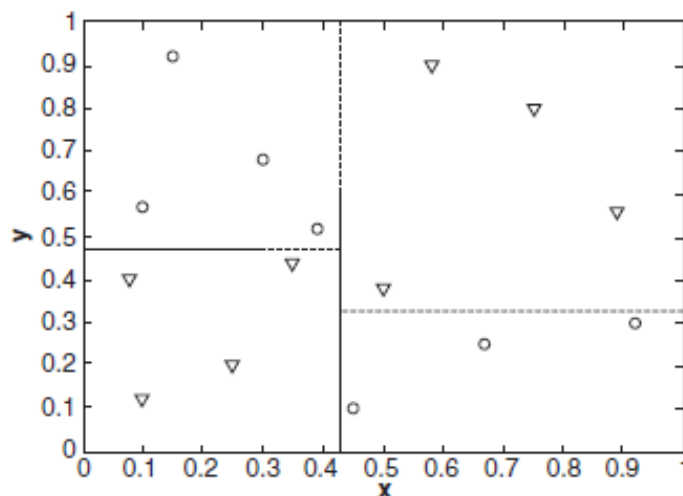
# Döntési fák (Hunt algoritmus)

Pl. emlős klasszifikátor



# Döntési fák (Hunt algoritmus)

Bináris és nominális problémákon kívül csak szögletes módon képes a teret felbontani:



Megfelelő döntési fa?

# Döntési fák

## (Hunt algoritmus)

Hunt algoritmus:

Iteratív algoritmus, csomópontok és levelek kialakítása attribútumok kiválasztásával

Ha egy csomóponton csak egy osztály elemei találhatóak

- > van egy levelünk

A meglévő csomópontokat bontsuk fel egy választott attribútum szerint:

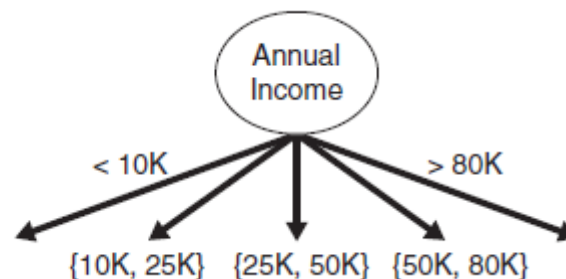
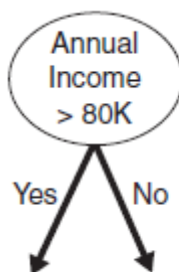
- bináris
- nominális
- ordinális
- intervallum szerint

Meddig van értelme bontogatni?

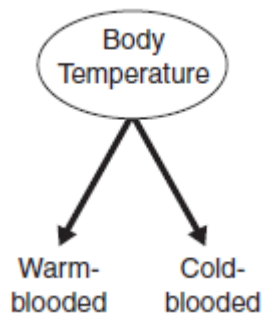
# Döntési fák

## (Hunt algoritmus)

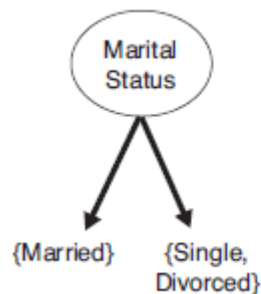
Vágás attribútumok  
típusa szerint



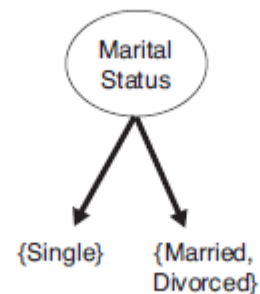
skála szerint



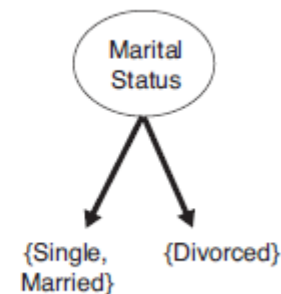
bináris



OR

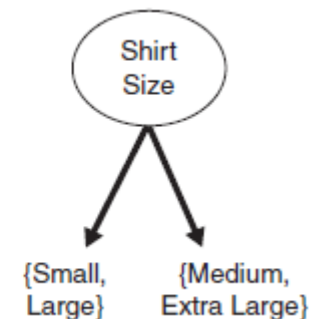
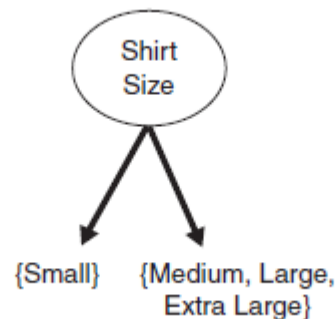
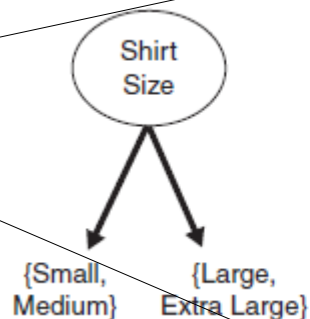


OR



nominális

Különbség?



ordinális

# Döntési fák

## (Hunt algoritmus)

**Procedure** Faépítés (data)

**If** (ha az adat nincs tökéletesen klasszifikálva)

**Find** legjobb szétválasztási attribútumot

**For each** minden A elemre

**Create** gyerek csomópontra

Data\_a = data ahol A = a

Faépítés (Data\_a)

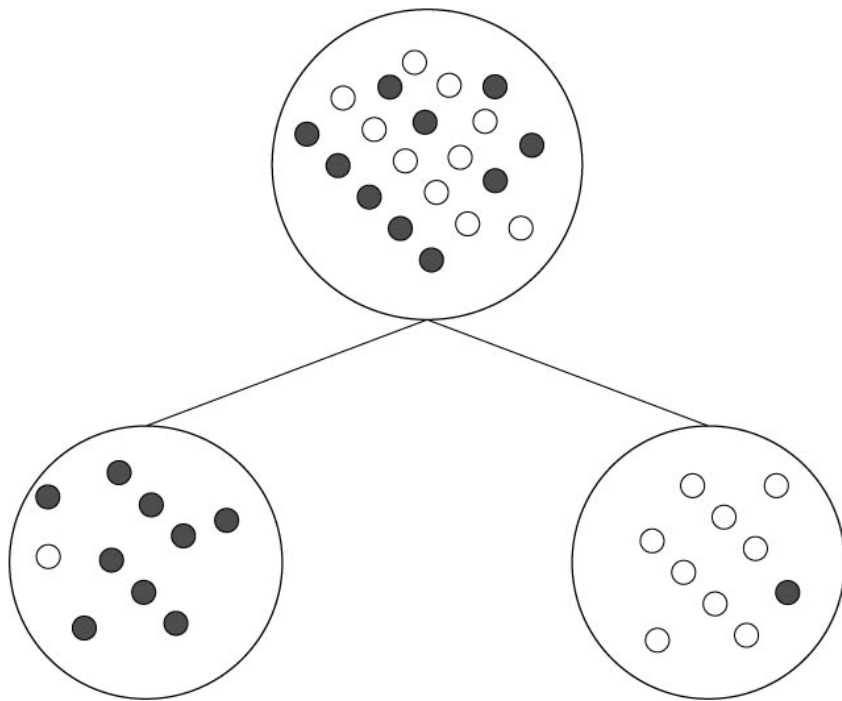
**Endfor**

**Endif**

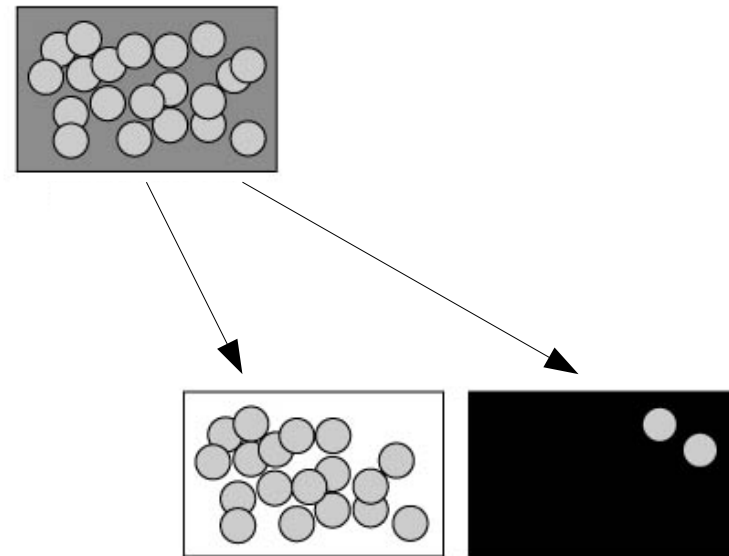
**EndProcedure**

# Döntési fák

(Hunt algoritmus)



Jó vágás



Rossz vágás



# Döntési fák

## (Hunt algoritmus)

Jó vágási attribútum:

- legjobban szeparálja az osztályokat külön ágakra  
→ növeli a tisztaságot (purity)
- kiegyensúlyozott

Purity mértékek lehetnek:

- klasszifikációs hiba
- entrópia
- gini
- vagy ami épp szükséges az adathoz

# Döntési fák

## (Hunt algoritmus)

Klasszifikációs hiba:

$p(i|t)$  : egy adott  $t$  csomópontban az  $i$  osztályba tartozó elemek aránya

Classification error:  $1 - \max(p(i,t))$

Növekmény(gain): 
$$\Delta = I(\text{parent}) - \sum_{j=1}^k \frac{N(v_j)}{N} I(v_j)$$

Itt  $I(\text{parent})$  az adott csomópont tisztasága,  $k$  a vágás után keletkező ágak száma,  $N(v_j)$  az adott ágon található elemek száma,  $N$  a csomópontban található elemek száma,  $I(v_j)$  pedig a  $j$ -dik ág tisztasága

# Döntési fák

## (Hunt algoritmus)

Példa:

Bináris klasszifikációt építünk, A vagy B attribútum szerint érdemes vágni, ha a tisztasági mérték a klasszifikációs hiba?

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
T	F	-

# Döntési fák

## (Hunt algoritmus)

Példa:

Bináris klasszifikációt építünk, A vagy B attribútum szerint érdemes vágni, ha a tisztasági mérték a klasszifikációs hiba?

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
T	F	-

Vágás A alapján:

$$\text{MCE} = 0 + 3/7$$

Vágás B alapján:

$$\text{MCE} = 1/4 + 1/6$$

Vágjunk B szerint!

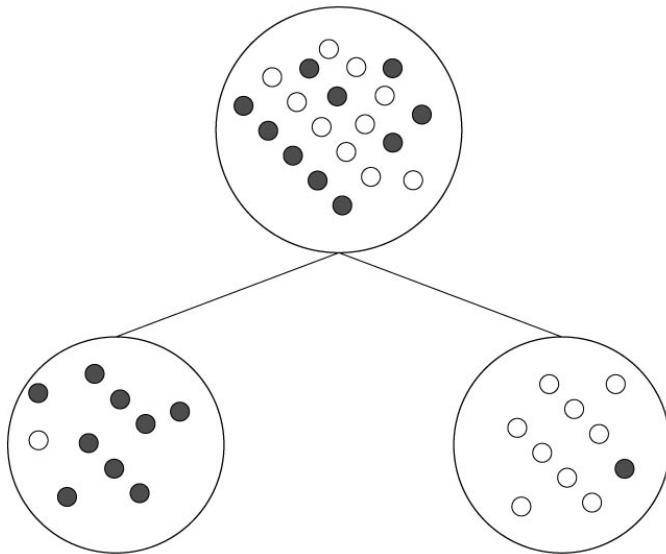
# Döntési fák

## (Hunt algoritmus)

Gini (population diversity)

$p(i|t)$  : egy adott  $t$  csomópontban az  $i$  osztályba tartozó elemek aránya

Gini: 
$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$



Gini a gyökérben?  
Gini a leveleknél?

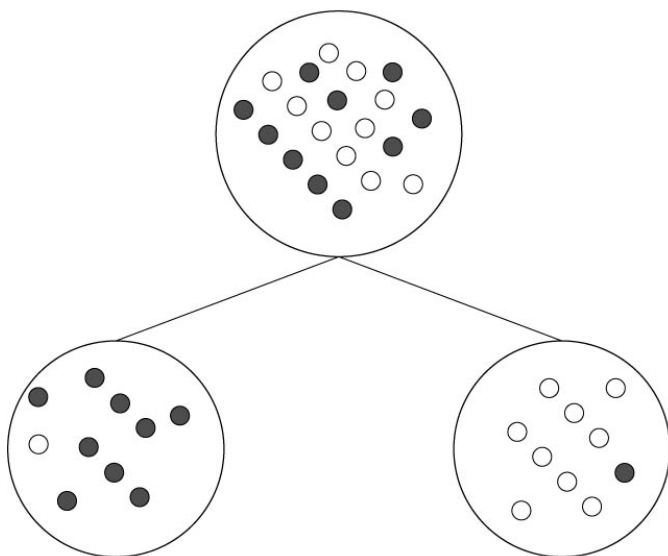
# Döntési fák

## (Hunt algoritmus)

Gini (population diversity)

$p(i|t)$  : egy adott  $t$  csomópontban az  $i$  osztályba tartozó elemek aránya

Gini: 
$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$



$$\text{Gini}(\text{gyökér}) = 0.5 = 0.5^2 + 0.5^2$$

$$\text{Gini}(\text{level}) = 0.82 = 0.1^2 + 0.9^2$$

# Döntési fák

## (Hunt algoritmus)

### Entrópia (információ)

$p(i|t)$  : egy adott  $t$  csomópontban az  $i$  osztályba tartozó elemek aránya

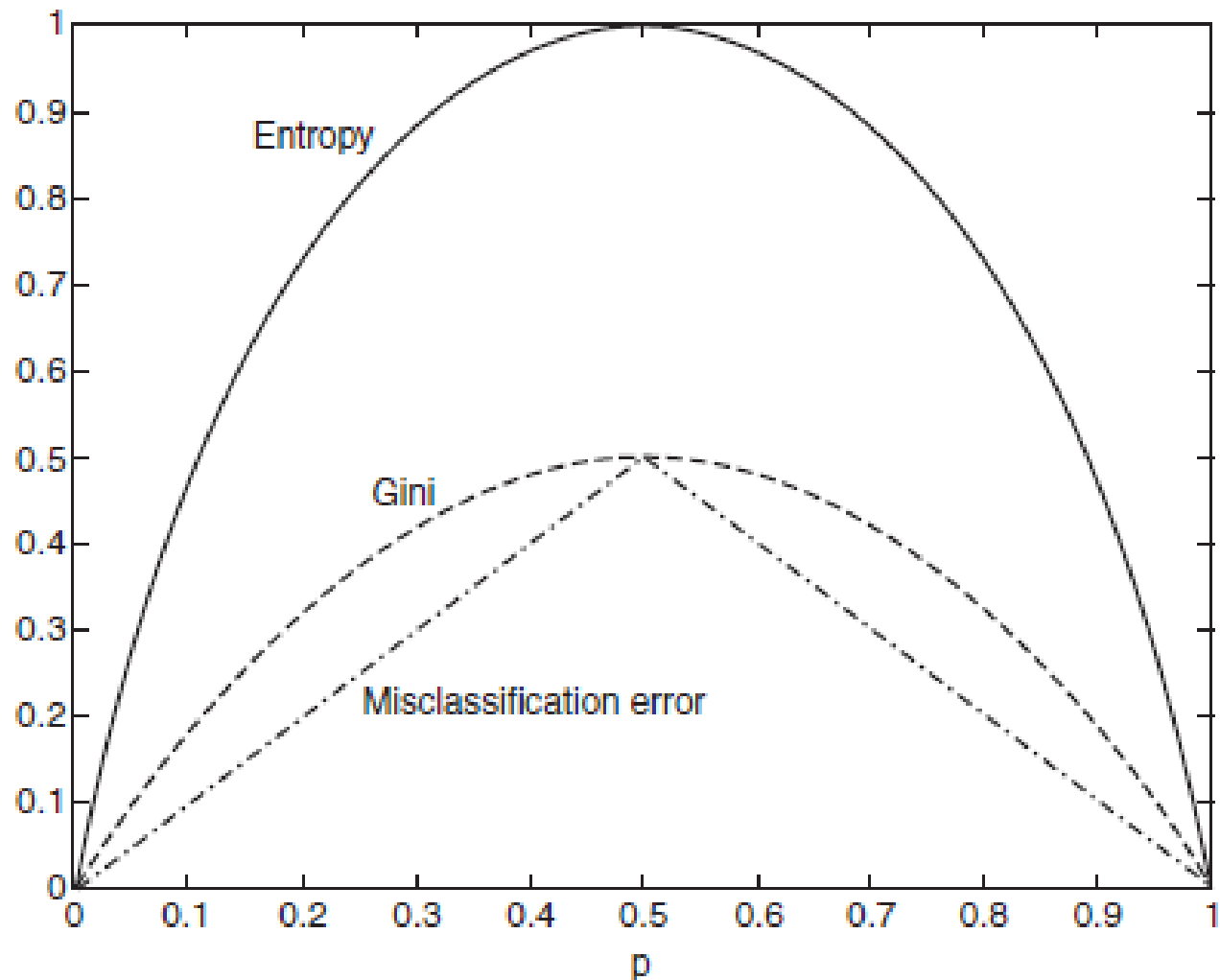
Entrópia: 
$$I(v_j) = - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t)$$

Mindhárom mérték közös tulajdonsága, hogy maximumukat 0.5-nél érik el. Mindhárom inkább a több részre szabdalást preferálja. (nem bináris attribútumoknál)

$$\text{Gain ratio} = \frac{\Delta_{\text{info}}}{-\sum_{i=1}^k P(v_i) \log_2 P(v_i)}$$

# Döntési fák

(Hunt algoritmus)





# Döntési fák

$$\text{Entropy}(t) = - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i [p(i|t)],$$

Node $N_1$	Count
Class=0	0
Class=1	6

Node $N_2$	Count
Class=0	1
Class=1	5

Node $N_3$	Count
Class=0	3
Class=1	3

Mennyi az entrópia,  
klasszifikációs hiba és  
a gini?

# Döntési fák

$$\text{Entropy}(t) = - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t),$$

$$\text{Gini}(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2,$$

$$\text{Classification error}(t) = 1 - \max_i [p(i|t)],$$

Node $N_1$	Count
Class=0	0
Class=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$

$$\text{Entropy} = -(0/6) \log_2(0/6) - (6/6) \log_2(6/6) = 0$$

$$\text{Error} = 1 - \max[0/6, 6/6] = 0$$

Node $N_2$	Count
Class=0	1
Class=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

$$\text{Entropy} = -(1/6) \log_2(1/6) - (5/6) \log_2(5/6) = 0.650$$

$$\text{Error} = 1 - \max[1/6, 5/6] = 0.167$$

Node $N_3$	Count
Class=0	3
Class=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

$$\text{Entropy} = -(3/6) \log_2(3/6) - (3/6) \log_2(3/6) = 1$$

$$\text{Error} = 1 - \max[3/6, 3/6] = 0.5$$

# Döntési fák

## (Hunt algoritmus)

### Döntési fák tulajdonságai:

1. bármilyen attribútumot kezelnek
2. szemléletesek
3. viszont sajnos NP nehéz probléma optimális fát építeni
4. zajra robusztusak
5. egyes alfák akár többször is előfordulnak (nincsenek keresztnyilak)
6. Túltanulási problémák

Akár minden tanulóhalmazbeli elemre építhetünk egy külön levelet!

### Két fajta probléma:

- túl nagy a fa, kevés a leveleknél a tanulópont
- nem megfelelő tanulópontok, az adott tulajdonságot nem tudja megtanulni, csak a tanulóhalmaz elemeire

Megoldás: **pruning!** (azaz le kell hagyni egyes ágakat, leveleket)

**Probléma jelzése:** - crossvalidation

- heldout halmaz
- stb.

# Döntési fák

Megoldások hibák kiküszöbölésére:

- ha két fa hasonlóan teljesít mindig válasszuk a kevésbé komplexet
- csak adott treshold feletti javulás esetén vágjunk (early pruning)
- utólag összevonjuk azon leveleket, melyek a legkevesebb hibát okozzák, s a leggyakoribb osztályra döntünk (post-pruning)
- MDL: azt a fát válasszuk, melynek a leírása kisebb

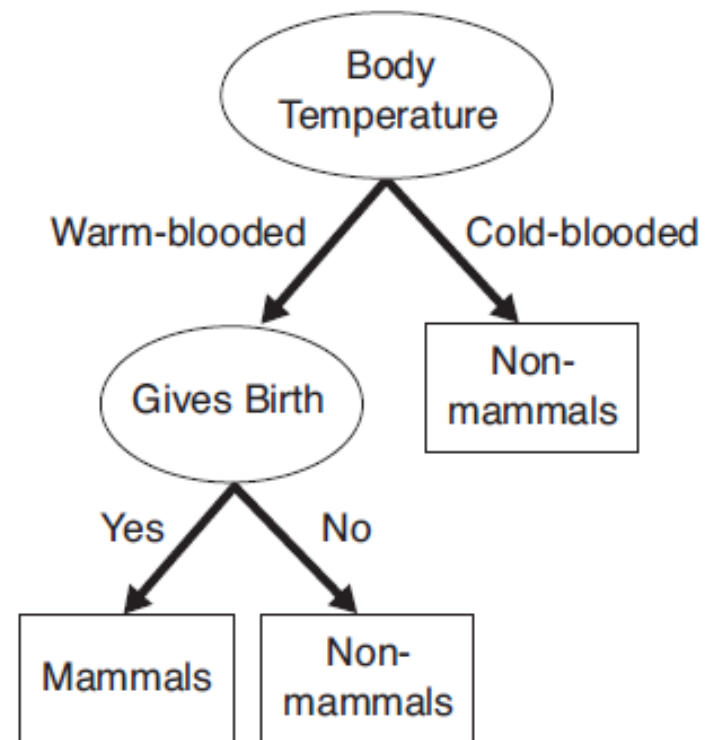
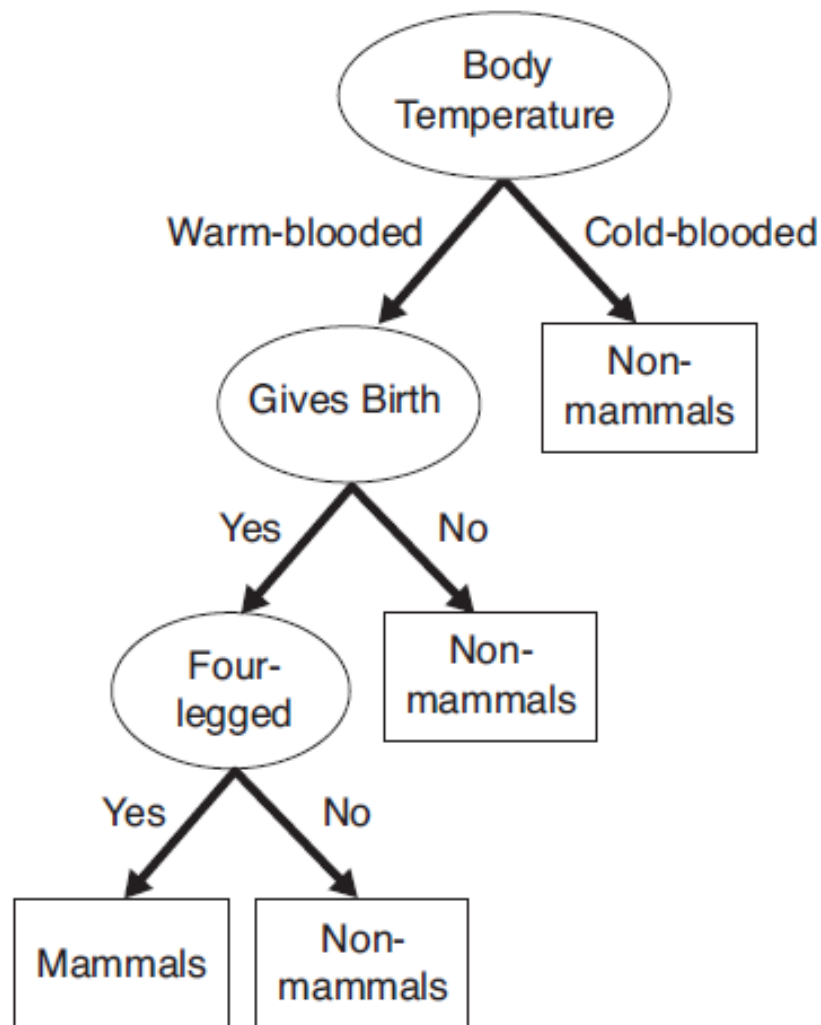
Példa hibára: mi lesz egy delfinnel? (tesztadat)

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no*
whale	warm-blooded	yes	no	no	no*
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

# Döntési fák

(Hunt algoritmus)

Zaj?



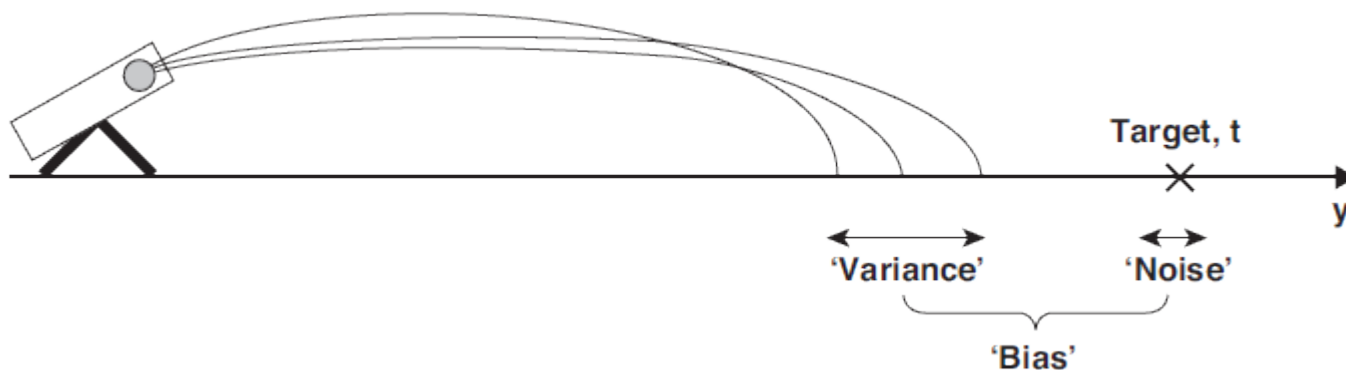
# Döntési fák

## (Hunt algoritmus)

Túl kevés adat?

Name	Body Temperature	Gives Birth	Four-legged	Hibernates	Class Label
salamander	cold-blooded	no	yes	yes	no
guppy	cold-blooded	yes	no	no	no
eagle	warm-blooded	no	no	no	no
poorwill	warm-blooded	no	no	yes	no
platypus	warm-blooded	no	yes	yes	yes

Mikor találjuk el?



# Döntési fák

## (Hunt algoritmus)

### Hiányzó adatok

?,?,?,?,no,auto  
xstab,?,?,?,yes,noauto  
stab,LX,?,?,?,yes,noauto  
stab,XL,?,?,?,yes,noauto  
stab,MM,nn,tail,?,yes,noauto  
?,?,?,?,OutOfRange,yes,noauto  
stab,SS,?,?,Low,yes,auto  
stab,SS,?,?,Medium,yes,auto  
stab,SS,?,?,Strong,yes,auto  
stab,MM,pp,head,Low,yes,auto  
stab,MM,pp,head,Medium,yes,auto  
stab,MM,pp,tail,Low,yes,auto  
stab,MM,pp,tail,Medium,yes,auto  
stab,MM,pp,head,Strong,yes,noauto  
stab,MM,pp,tail,Strong,yes,auto

### Feladat 1

DT weka-val:

- Shuttle-landing-control
- house-votes-84
- bank-data

### Mit kezdhethetünk velük?

- median/átlag?
- az összes lehetőség kibontása?

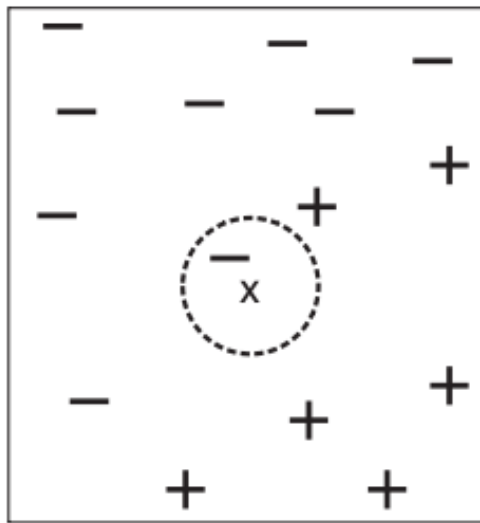
## K - nearest neighbor (K-NN)

Alapelv: Minden elem tulajdonságait a hozzá hasonlókat határozzák meg.

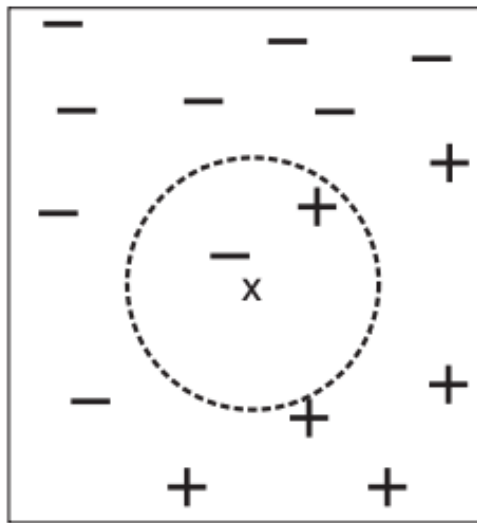
“Úgy megy mint egy kacska, úgy úszik mint egy kacska, úgy eszik mint egy kacska, tehát kacska!”

1. Minden tesztadathoz keressük meg a K legközelebbi elemet.
2. A legtöbbet reprezentált osztályt jósoljuk az adott tesztadatnak

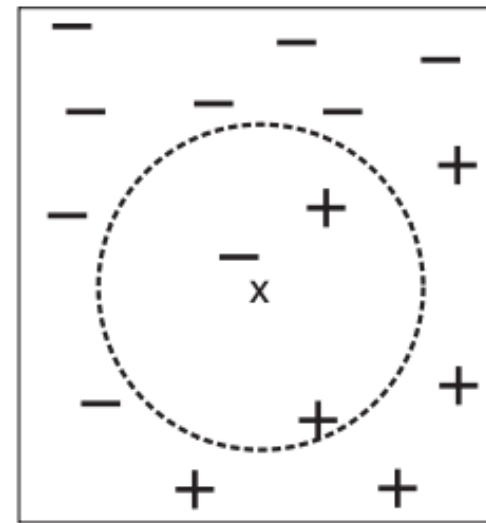
Példa:



(a) 1-nearest neighbor



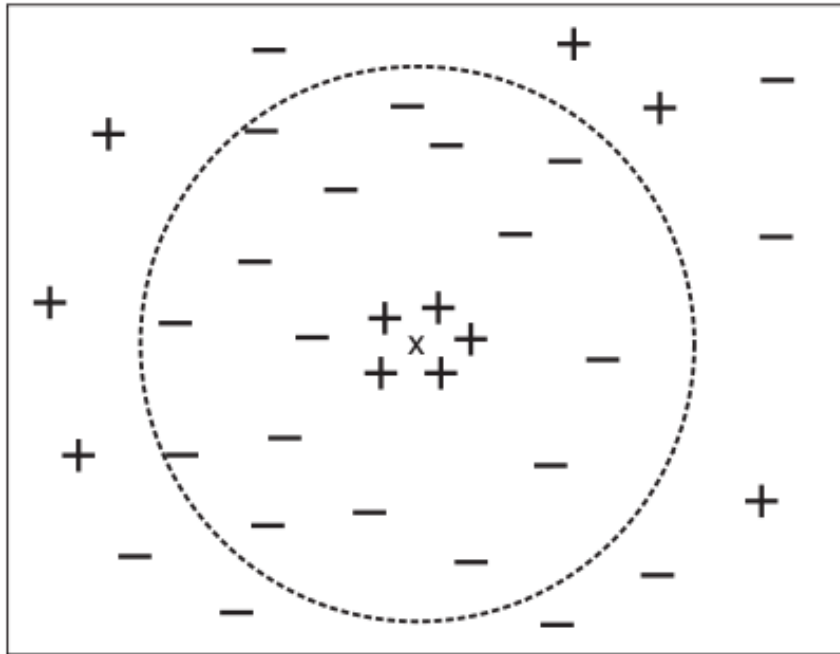
(b) 2-nearest neighbor



(c) 3-nearest neighbor



## K - nearest neighbor (K-NN)



Mi a gond?

- mohó (eager) algoritmus: kész modell készítése amint a tanuló adat létezik → többé nincs szükség a tanulóhalmazra
- lusta (lazy) algoritmus: kész jóslatot a tanulóhalmaz segítségével készít → szükséges a tanulóhalmaz a jósláshoz

A DT mohó míg a K-NN lusta algoritmus :

Méret?

Bonyolultság?

Skálázhatóság?

# Naïve-Bayes

Bayes-i alapok:

$$p(h | d) = \frac{P(d | h)P(h)}{P(d)}$$

Priori:  $P(h)$

Posteriori:  $P(h | d)$

Evidence: biztosan igaz fent:  $P(d)$

Total probability:  $P(h) = P(h, d) + P(h, d^c) = P(h | d)P(d) + P(h | d^c)P(d^c)$

**Feladat:** Ha az épületben megforduló emberek tizede tanár a többi diák és 10-ből 8 diáknak saját lappal rendelkezik míg minden 10 tanárból csak 5nek van laptopja, mennyi a valószínűsége, hogy ha egy ember bejön az ajtón:

- a) tanár
- b) diák
- c) van laptopja
- d) van laptopja és diák
- e) van laptopja és tanár

Melyik valószínűségeknek kell összegben 1-et adniuk?

# Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínűségeket:

$P(\text{class}=1 \mid A_1, A_2, \dots, A_n)$  és  $P(\text{class}=0 \mid A_1, A_2, \dots, A_n)$

Amelyik nagyobb arra döntünk.  $P(A_1, A_2, \dots, A_n \mid \text{class}=1) P(\text{class}=1) / P(A_1, A_2, \dots, A_n)$


$$P(A_1, A_2, \dots, A_n \mid \text{class}=1) = \prod P(A_i \mid \text{class}=1)$$



konstans!

Saját háza van	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

# Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínűségeket:

$P(\text{class}=1 \mid A_1, A_2, \dots, A_n)$  és  $P(\text{class}=0 \mid A_1, A_2, \dots, A_n)$

Amelyik nagyobb arra döntünk.  $P(A_1, A_2, \dots, A_n \mid \text{class}=1) P(\text{class}=1) / P(A_1, A_2, \dots, A_n)$

$P(A_1, A_2, \dots, A_n \mid \text{class}=1) = \prod P(A_i \mid \text{class}=1)$

konstans!

Saját háza van	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

$P(\text{saját ház}=nincs \mid \text{hitel}=nincs)=4/7$   
 $P(\text{saját ház}=van \mid \text{hitel}=nincs)=3/7$   
 $P(\text{saját ház}=nincs \mid \text{hitel}=van)=1$   
 $P(\text{saját ház}=van \mid \text{hitel}=van)=0$   
 $P(\text{családi\_áll}=egyedülálló \mid \text{hitel}=nincs)=2/7$   
 $P(\text{családi\_áll}=házas \mid \text{hitel}=nincs)=4/7$   
 $P(\text{családi\_áll}=elvált \mid \text{hitel}=nincs)=1/7$   
 $P(\text{családi\_áll}=egyedülálló \mid \text{hitel}=van)=2/3$   
 $P(\text{családi\_áll}=házas \mid \text{hitel}=van)=0$   
 $P(\text{családi\_áll}=elvált \mid \text{hitel}=van)=1/3$

$P(\text{kereset}=? \mid \text{hitel}=van)=?$   
 $P(\text{kereset}=? \mid \text{hitel}=nincs)=?$

# Naïve-Bayes

Folytonos változók esetében

- diszkrétizáljuk az adatot : 0-90e Ft-ig 91-125e stb
- modellezzük a problémát normál eloszlással!

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp -\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}$$

Saját háza van	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

Kereset:

hitel=nincs:

Átlag: 110e

Szórás négyzet: 2975

hitel=van:

Átlag: 90e

Szórás négyzet: 25

# Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínűségeket:

$P(\text{class}=1 \mid A_1, A_2, \dots, A_n)$  és  $P(\text{class}=0 \mid A_1, A_2, \dots, A_n)$

Amelyik nagyobb arra döntünk.  $P(A_1, A_2, \dots, A_n \mid \text{class}=1)P(\text{class}=1)/P(A_1, A_2, \dots, A_n)$


$$P(A_1, A_2, \dots, A_n \mid \text{class}=1) = \prod P(A_i \mid \text{class}=1)$$



konstans!

Saját ház	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

Van-e hitele a következő tulajdonságokkal rendelkező személynek:

- nincs saját háza
- házas
- 120e a keresete

$P(\text{hitel}=\text{van} \mid \text{saját ház}=\text{nincs}, \text{családi\_áll}=\text{házas}, \text{kereset}=120\text{e})=?$

$P(\text{hitel}=\text{van} \mid \text{saját ház}=\text{nincs}, \text{családi\_áll}=\text{házas}, \text{kereset}=120\text{e})=?$

# Naïve-Bayes

Szeretnénk attribútumok alapján meghatározni a posteriori valószínűségeket:

$P(\text{class}=1 \mid A_1, A_2, \dots, A_n)$  és  $P(\text{class}=0 \mid A_1, A_2, \dots, A_n)$

Amelyik nagyobb arra döntünk.  $P(A_1, A_2, \dots, A_n \mid \text{class}=1) P(\text{class}=1) / P(A_1, A_2, \dots, A_n)$


$$P(A_1, A_2, \dots, A_n \mid \text{class}=1) = \prod P(A_i \mid \text{class}=1)$$

Saját ház	Családi állapot	kereset	Van hitele?
van	egyedülálló	125e	nincs
nincs	házas	100e	nincs
nincs	egyedülálló	70e	nincs
van	házas	120e	nincs
nincs	elvált	95e	van
nincs	házas	60e	nincs
van	elvált	220e	nincs
nincs	egyedülálló	85e	van
nincs	házas	75e	nincs
nincs	egyedülálló	90e	van

Van-e hitele a következő tulajdonságokkal rendelkező személynek:

- nincs saját háza
- házas
- 120e a keresete

$P(\text{hitel}=\text{van} \mid \text{saját ház}=\text{nincs}, \text{családi\_áll}=\text{házas}, \text{kereset}=120\text{e}) = P(\text{saját ház}=\text{nincs} \mid \text{hitel}=\text{van}) * P(\text{családi\_áll}=\text{házas} \mid \text{hitel}=\text{van}) * P(\text{kereset}=120\text{e} \mid \text{hitel}=\text{van}) = 1 * 0 * 1.2 * 10^{-9} = 0$

$P(\text{hitel}=\text{nincs} \mid \text{saját ház}=\text{nincs}, \text{családi\_áll}=\text{házas}, \text{kereset}=120\text{e}) = P(\text{saját ház}=\text{nincs} \mid \text{hitel}=\text{nincs}) * P(\text{családi\_áll}=\text{házas} \mid \text{hitel}=\text{nincs}) * P(\text{kereset}=120\text{e} \mid \text{hitel}=\text{nincs}) = 3/7 * 4/7 * 0.0072 = 0.0024$

Jóslat: Nincs hitele!

# Naïve-Bayes

A 0-a valószínűségek több esetben zajként jelentkeznek (pl. nincs rá elem az eredeti adathalmazban ergo nem is lehetséges)

## M-estimate:

Legyen  $p$  egy előre meghatározott minimum valószínűség

Módosítsuk a feltételes valószínűségek kiszámítását:

$$P(x_i|y_j) = \frac{n_c + mp}{n + m}$$

Ahol  $m$  egy előre meghatározott konstans,  $n_c$  azon  $x_i$  tulajdonsággal rendelkező tanulóponatok száma melyek osztályváltozója  $y_i$ ,  $n$  pedig az összes ide tartozó tanulóponatok száma.

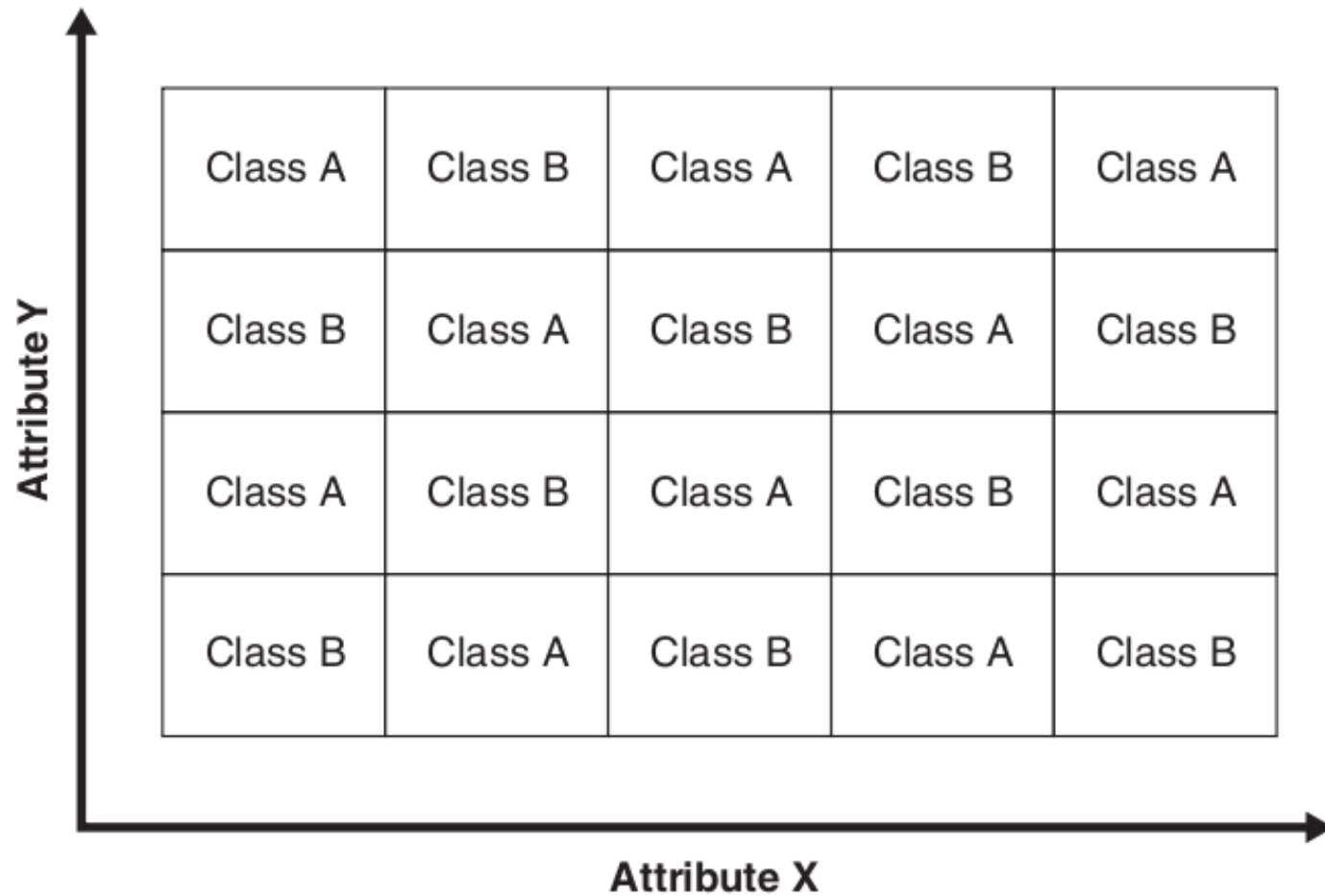
Olyan esetekben is  $p$  lesz a posteriori valószínűség, ha egyáltalán nincs olyan tanulóponatok melynek  $y_i$  az osztályváltozója.

M-estimate nagyban segíti a zajos, hiányos vagy egyszerűen speciális esetek korrekt kiértékelését anélkül, hogy azok szerepeltek volna az tanulómátrixban.



## Órai feladat 2:

Melyik klasszifikátor működik jól/rosszul s miért?  
(K-NN, Naïve-Bayes, DT)



# Kiértékelés

Confusion matrix:

Alapigazság/ predikció	p	n	Total
p	True Positive (TP)	False Negative (FN)	TP+FN
n	False Positive (FP)	True Negative (TN)	FP+TN
Total	TP+FN	FP+TN	

# Kiértékelés

**Accuracy:** a helyesen klasszifikálás valószínűsége

$$TP+TN/(TP+FP+TN+FN)$$

**Precision (p):** egy relevans dokumentum helyes klasszifikálásának valószínűsége

$$TP/(TP+FP)$$

**Recall (r):** annak a valószínűsége, hogy egy releváns dokumentumot helyesen klasszifikálunk

$$TP/(TP+FN)$$

**F-measure:** a precision és a recall harmónikus közepe  
( $2 * p * r / (p + r)$ )

# Kiértékelés

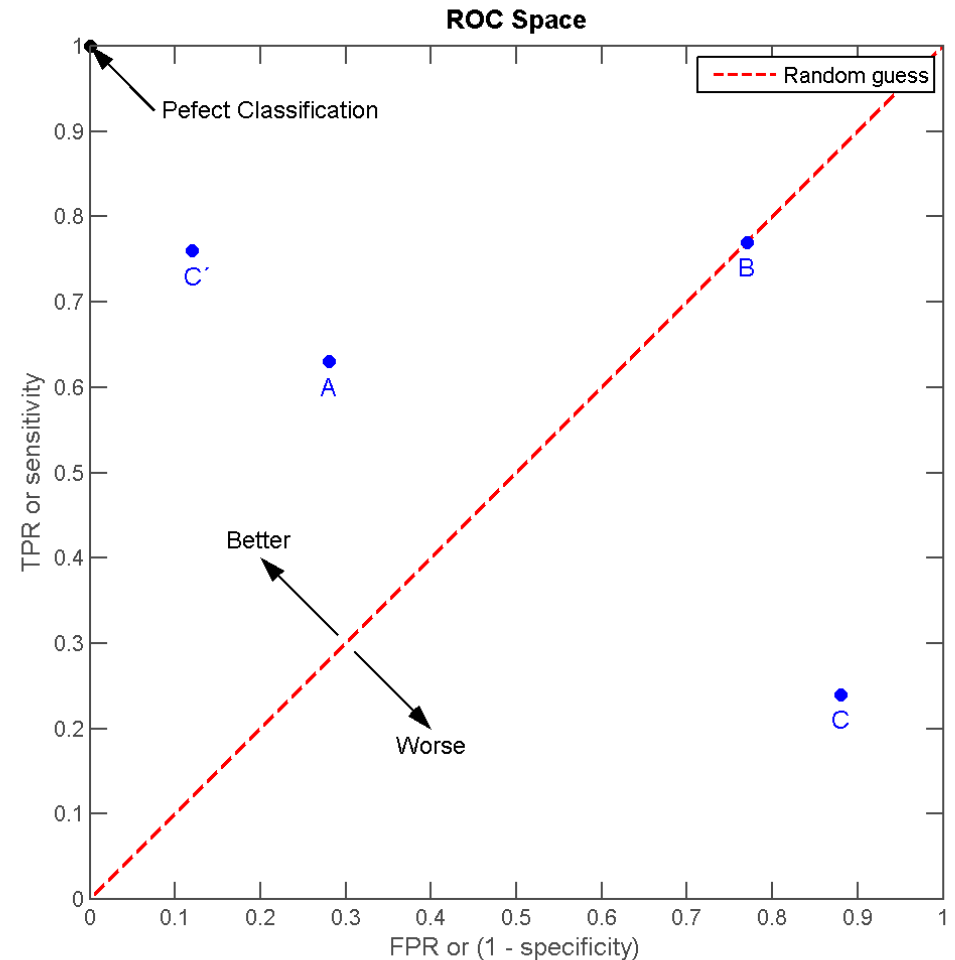
False-Positive Rate (FPR) =  
 $\text{FP}/(\text{FP}+\text{TN})$

True-Positive Rate (TPR) =  
 $\text{TP}/(\text{TP}+\text{FN})$

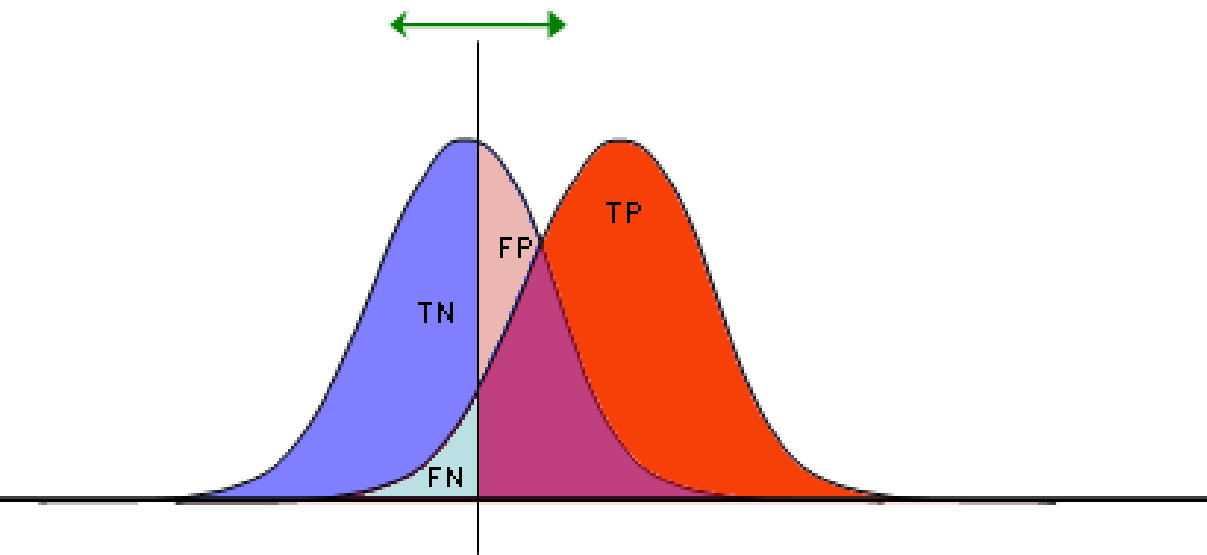
ROC: Receiver Operating  
Characteristic

MAP: Mean Average  
Precision

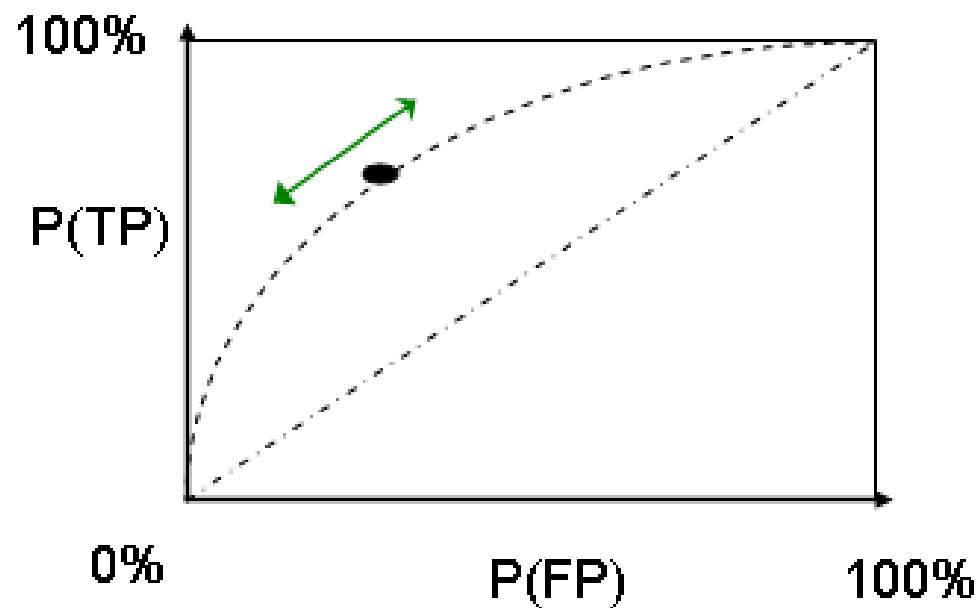
$$\text{AveP} = \frac{\sum_{r=1}^N (P(r) \times \text{rel}(r))}{\text{number of relevant documents}}$$



# Kiértékelés



TP	FP
FN	TN
1	1



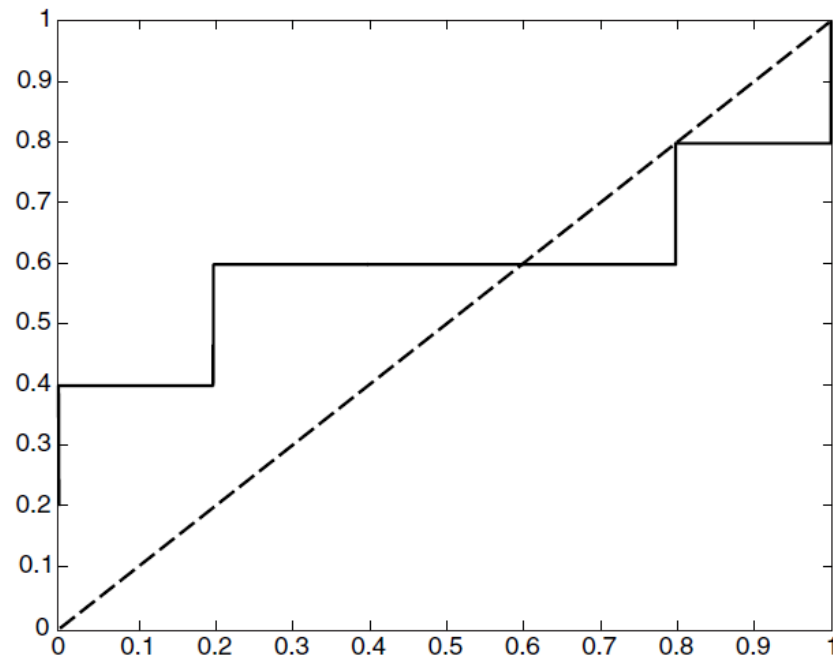
# ROC: Receiver Operating Characteristic

- csak **binaris** osztályozásnál használható (i.e. egy osztályra)
- **Area Under Curve:** AUC annak a valószínűsége, hogy véletlen pozitív elemeket előrébb sorol mint véletlen negatívakat
- mivel az jóslatok sorrendjéből számítjuk, a klasszifikálónak nem csak bináris jóslatokat kell visszaadnia
- előnye, hogy **nem függ a vágási ponttól**

Class	+	−	+	−	−	−	+	−	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

# ROC: Receiver Operating Characteristic

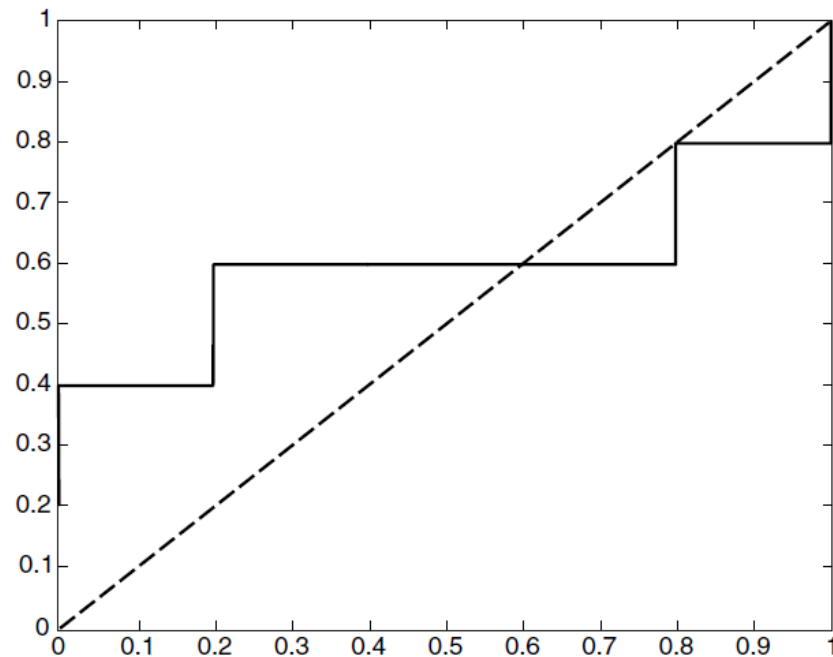
Class	+	-	+	-	-	-	+	-	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0



AUC=?

# ROC: Receiver Operating Characteristic

Class	+	-	+	-	-	-	+	-	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0



AUC=0.6!

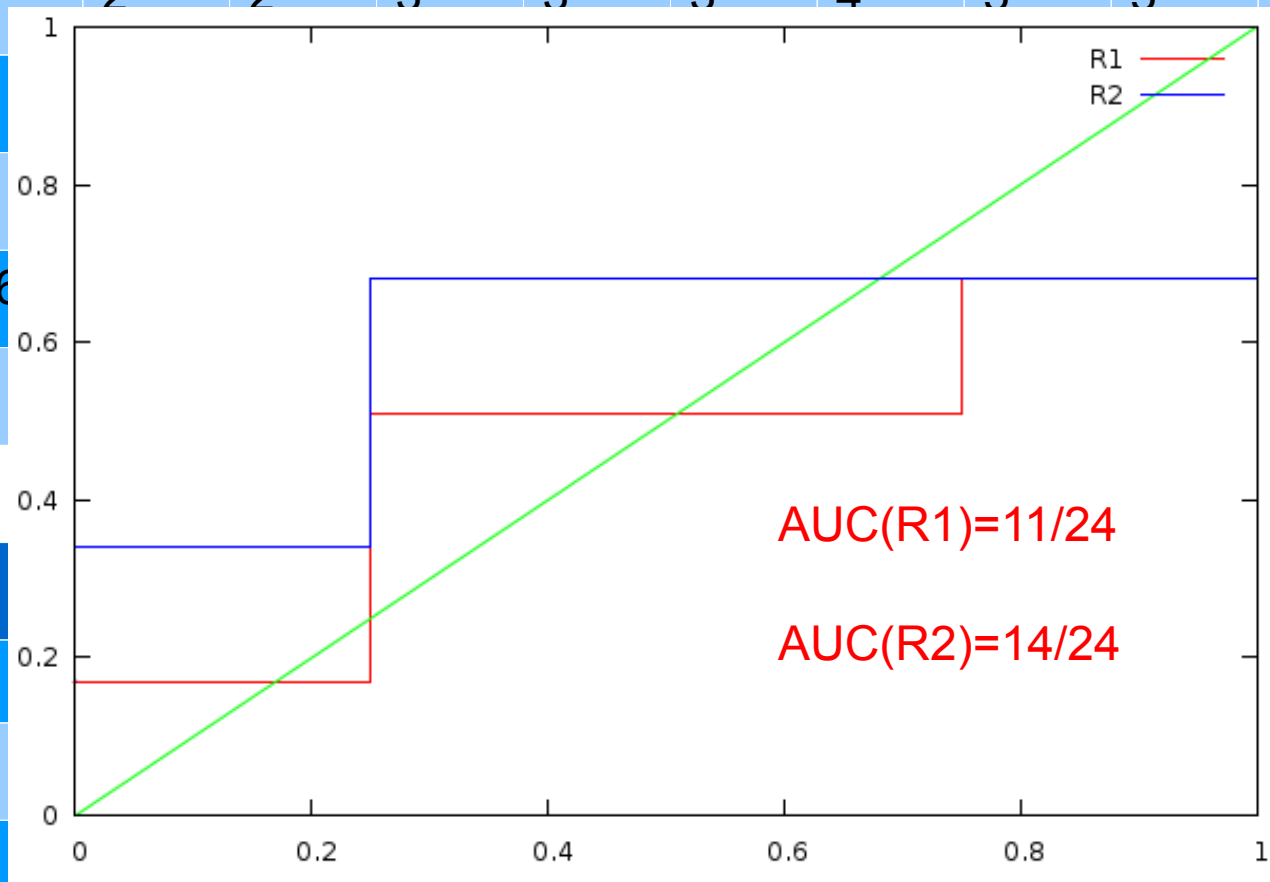




+	+	-	+	-	-	+	+	-	+	
6	5	4	4	3	3	3	2	1	1	TP
0	1	2	2	3	3	3	4	5	5	FN
0	0	0	1	1	2	3	3	3	4	TN
4	4	4	3	3	2	1	1	1	0	FP
1	5/6	4/6	4/6	3/6	3/6	3/6	2/6	1/6	1/6	TPR
1	1	1	3/4	3/4	2/4	1/4	1/4	1/4	0	FPR

+	+	-	-	-	+	+	-	+	+	
6	5	4	4	4	4	3	2	2	1	TP
0	1	2	2	2	2	3	4	4	5	FN
0	0	0	1	2	3	3	3	4	4	TN
4	4	4	3	2	1	1	1	0	0	FP
1	5/6	4/6	4/6	4/6	4/6	3/6	2/6	2/6	1/6	TPR
1	1	1	3/4	2/4	1/4	1/4	1/4	0	0	FPR

+	+	-	+	-	-	+	+	-	+	
6	5	4	4	3	3	3	2	1	1	TP
0	1	2	2	3	3	3	4	5	5	FN
0	0									N
4	4									P
1	5/6									PR
1	1									PR



N  
P  
PR  
PR

	TP
	FN
	TN

+										
6										
0										
0										
4	4	4	3	2	1	1	1	0	0	FP
1	5/6	4/6	4/6	4/6	4/6	3/6	2/6	2/6	1/6	TPR
1	1	1	3/4	2/4	1/4	1/4	1/4	0	0	FPR