Adatbányászat: Osztályozás Alapfogalmak, döntési fák, kiértékelés

4. fejezet

Tan, Steinbach, Kumar

Bevezetés az adatbányászatba

előadás-fóliák

fordította

Ispány Márton

Az osztályozás definíciója

- Adott rekordok egy halmaza (*tanító adatállomány*)
  - Minden rekord *attributumok* értékeinek egy halmazából áll, az attributumok egyike az ún. *osztályozó vagy cél* változó.
- Találjunk olyan *modellt* az osztályozó attributum-ra, amely más attributumok függvényeként állítja elő.
- Cél: korábban nem ismert rekordokat kell olyan pontosan osztályozni ahogyan csak lehetséges.
  - A *teszt adatállomány* a modell pontosságának meghatározására szolgál. Általában az adatállományt két részre bontjuk, a tanítón illesztjük a modellt, a tesztelőn pedig validáljuk.

Adócsalók elfogása

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Adó visszatérítések a 2011-es évre

Egy új adó visszaigénylés 2012-ben
Csalás ez vagy sem?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Egy példa osztályozási feladatra: találjunk egy olyan módszert amely különbséget tud tenni a rekordok különböző osztályai között (csalás vs nem-csalás)

Mi az osztályozás?

- **Osztályozás** egy olyan **f** célfüggvény megtanulása, amely az **x** attribútumhalmazt képezi le **y** cimkék egy előredefiniált halmazára

kategorikus kategorikus folytonos osztály

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Az egyik attribútum az osztályozó attribútum
Esetünkben: Csalás (Cheat)

Két osztály címke (vagy osztály):
Yes (1), No (0)

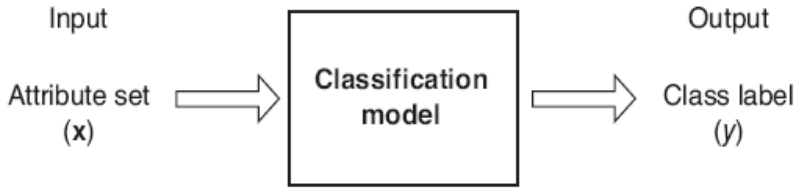


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

Miért osztályozás?

- Az f célfüggvényt osztályozási modellnek nevezzük

- Leíró modellezés: Magyarázó eszköz arra, hogy különbséget tegyünk objektumok különböző osztályai között (pl., megértsük, hogy miért csalják el egyes emberek az adójukat)

- Prediktív modellezés: Jelezzük előre korábban nem látott rekordok osztályát

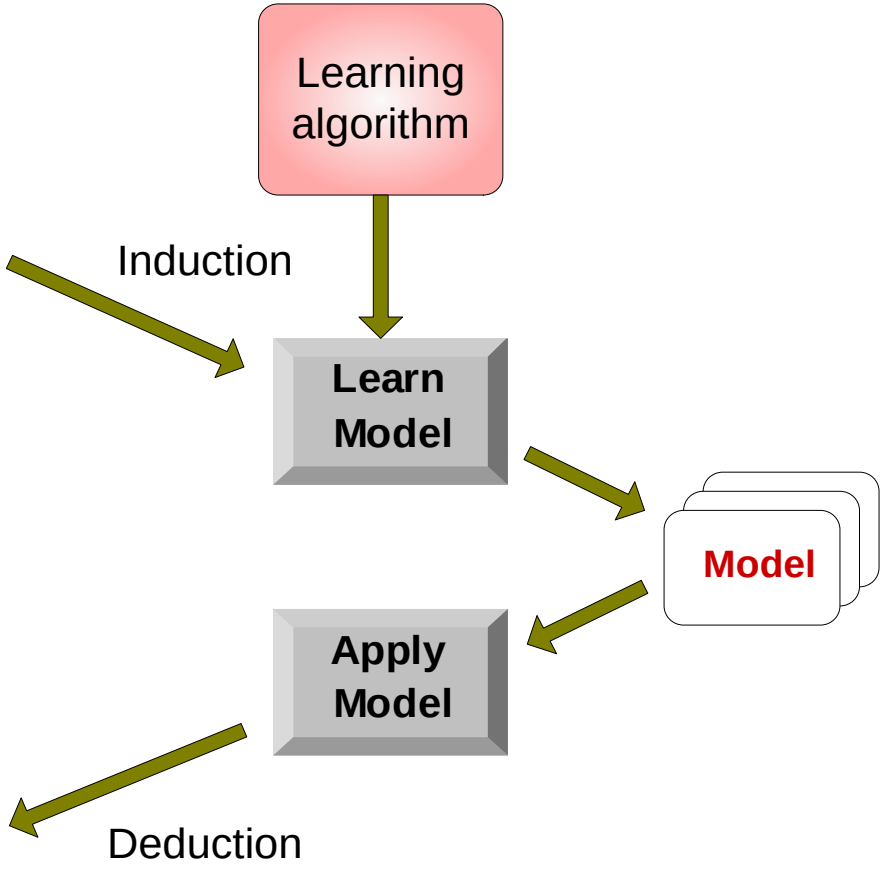
Az osztályozás általános megközelítése

- Tanuló adatállomány: olyan rekordokból áll, amelyeknél ismerjük az osztály címkét
- A tanuló adatállományt arra használjuk, hogy egy osztályozó modellt építsünk
- Korábban nem használt rekordok címkézett teszt adatállományát használjuk a modell jóságának kiértékelésére
- Az osztályozási modellt olyan új rekordokra alkalmazzuk, amelyeknél ismeretlen az osztály címke

Az osztályozási feladat

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



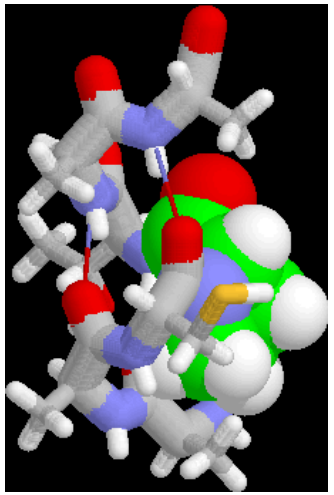
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Példák osztályozási feladatra

- A daganatos sejtek előrejelzése: jó vagy rossz indulatú.
- Hitelkártya tranzakciók osztályozása: legális vagy csalás.

- A fehérjék másodlagos szerkezetének osztályozása: alpha-helix, beta-híd, véletlen spirál.



- Újsághírek kategórizálása: üzlet, időjárás, szórakozás, sport stb.

Osztályozási módszerek

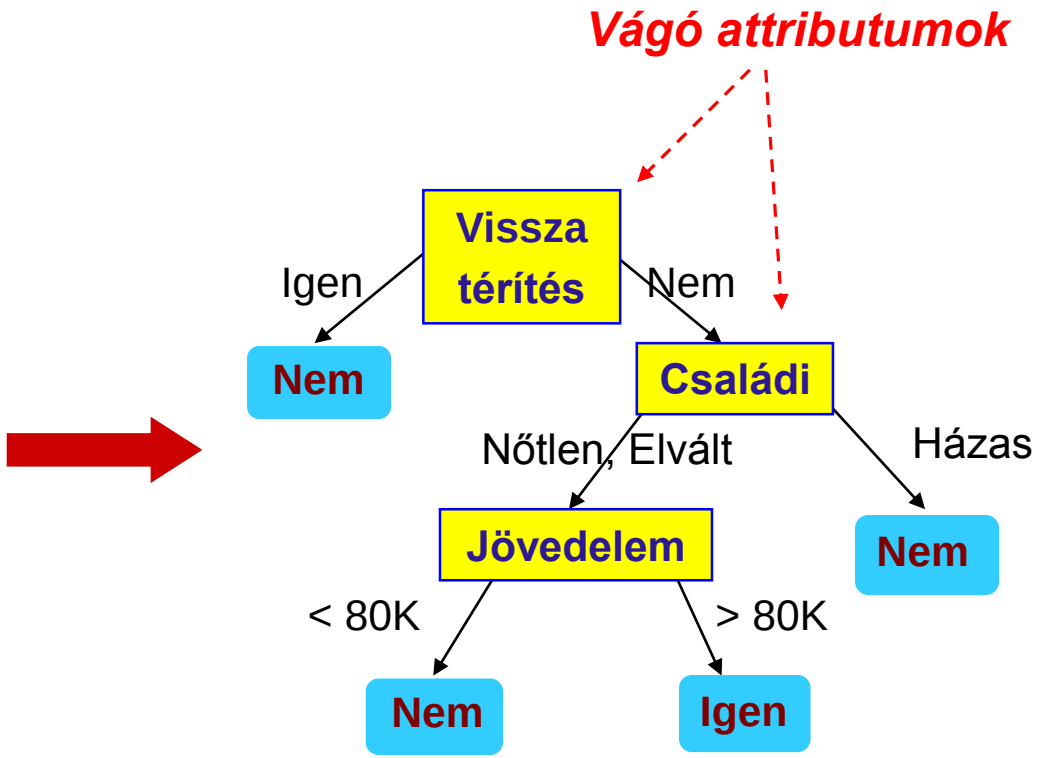
- Döntési fák
- Szabály alapú módszerek
- Memória alapú módszerek (legközelebbi *k-társ*)
- Logisztikus regresszió
- Neurális hálók
- Naív Bayes módszer és Bayes hálók
- Vektorgépek: SVM

Példa döntési fára

kategórikus kategórikus folytonos osztályozó

Tid	Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
1	Igen	Nőtlen	125K	Nem
2	Nem	Házas	100K	Nem
3	Nem	Nőtlen	70K	Nem
4	Igen	Házas	120K	Nem
5	Nem	Elvált	95K	Igen
6	Nem	Házas	60K	Nem
7	Igen	Elvált	220K	Nem
8	Nem	Nőtlen	85K	Igen
9	Nem	Házas	75K	Nem
10	Nem	Nőtlen	90K	Igen

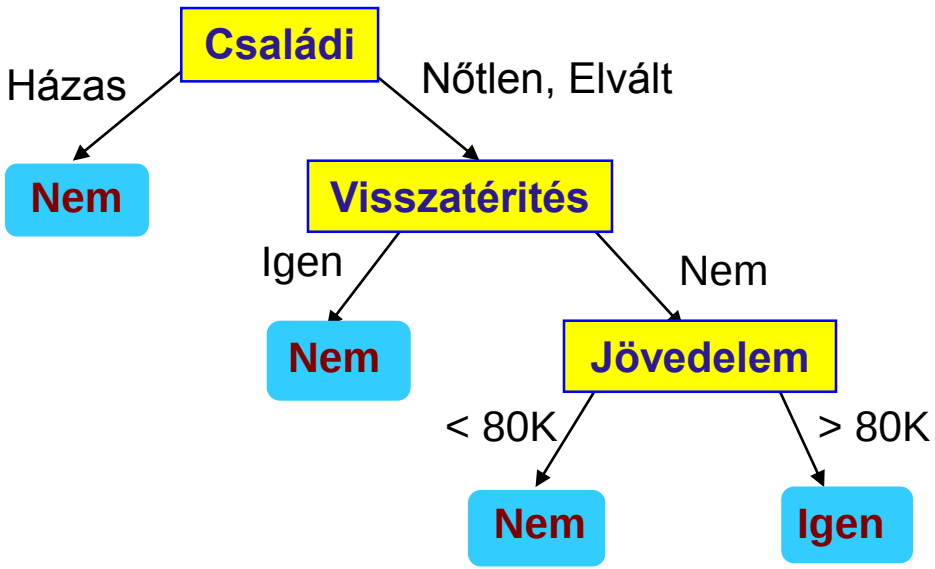
Tanító adatállomány



Model: Döntési fa

Másik példa döntési fára

	kategórikus	kategórikus	folytonos	osztályozó
Tid	Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
1	Igen	Nőtlen	125K	Nem
2	Nem	Házas	100K	Nem
3	Nem	Nőtlen	70K	Nem
4	Igen	Házas	120K	Nem
5	Nem	Elvált	95K	Igen
6	Nem	Házas	60K	Nem
7	Igen	Elvált	220K	Nem
8	Nem	Nőtlen	85K	Igen
9	Nem	Házas	75K	Nem
10	Nem	Nőtlen	90K	Igen



Több fa is illeszkedhet ugyanazokra az adatokra!

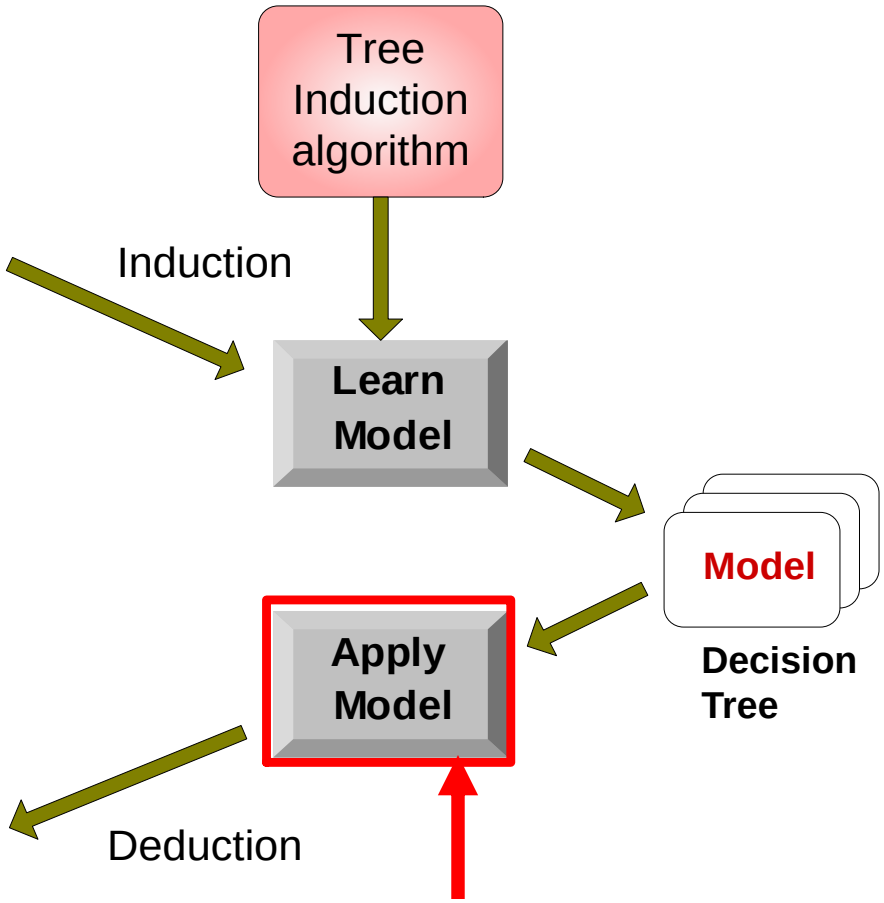
Osztályozás döntési fával

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

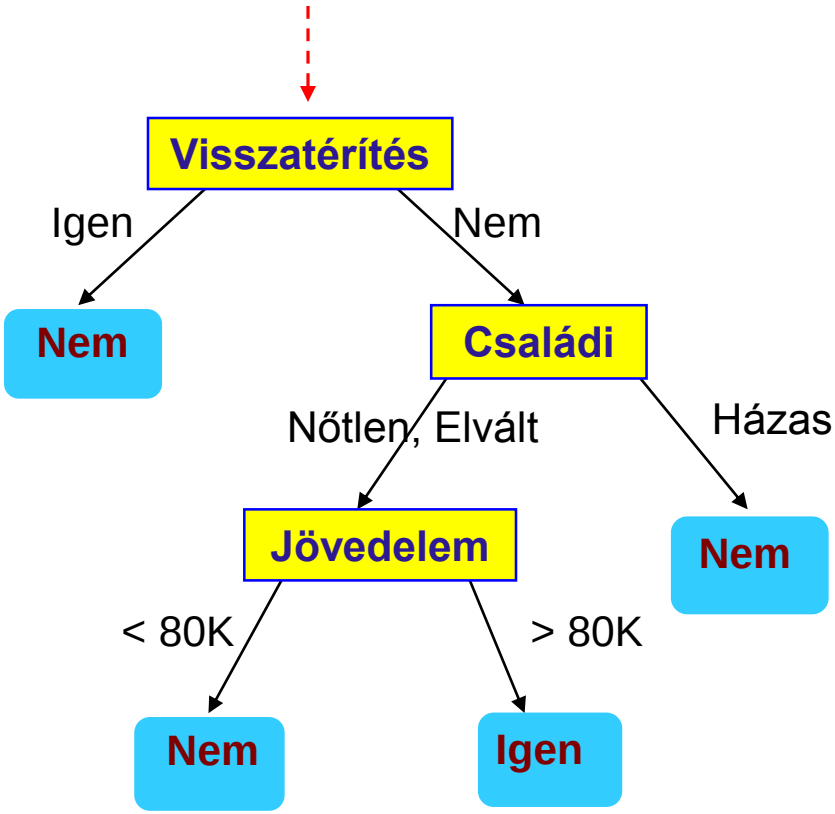
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Alkalmazzuk a modellt a teszt adatokra

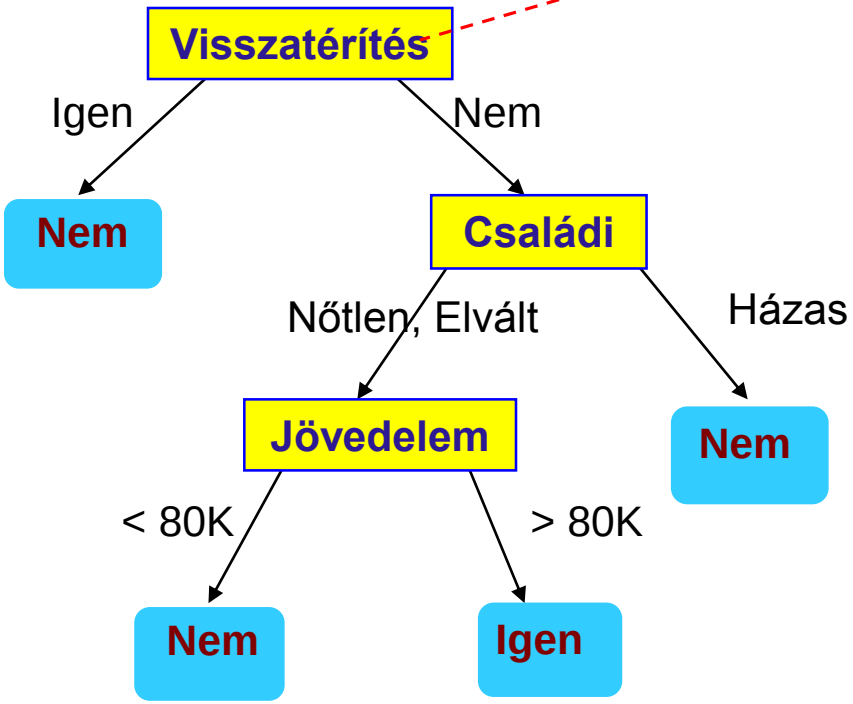
Induljunk a fa gyökerétől.



Teszt adatok

Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
Nem	Házas	80K	?

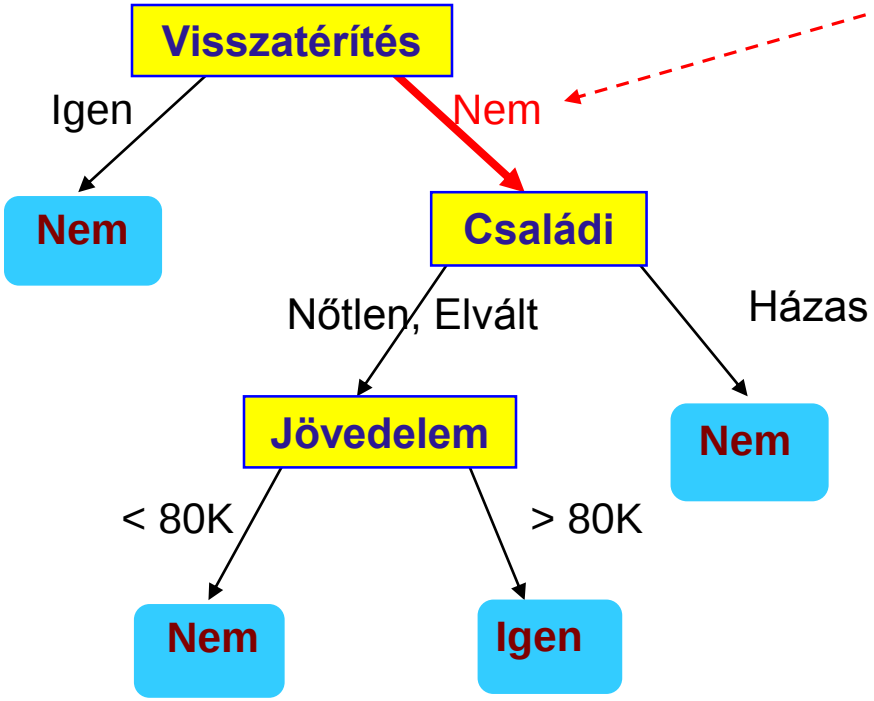
Alkalmazzuk a modellt a teszt adatokra



Teszt adatok

Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
Nem	Házas	80K	?

Alkalmazzuk a modellt a teszt adatokra



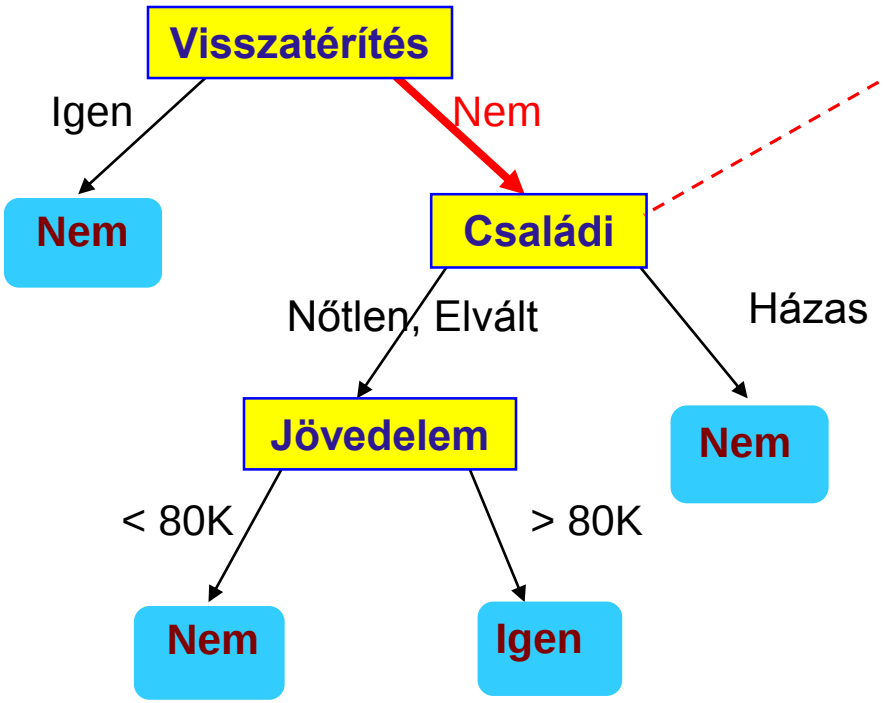
Teszt adatok

Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
Nem	Házas	80K	?

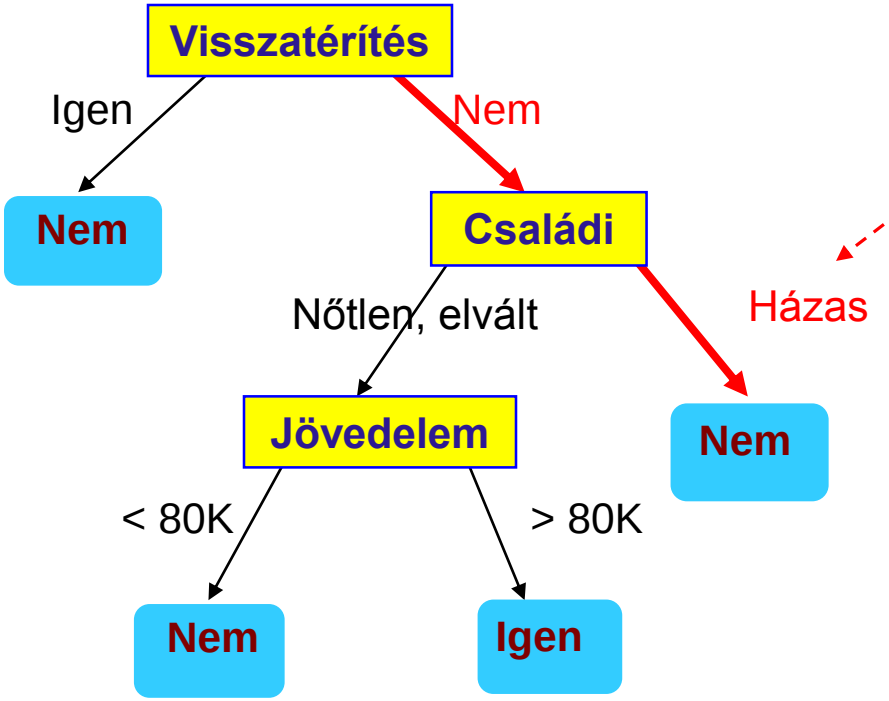
Alkalmazzuk a modellt a teszt adatokra

Teszt adatok

Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
Nem	Házas	80K	?



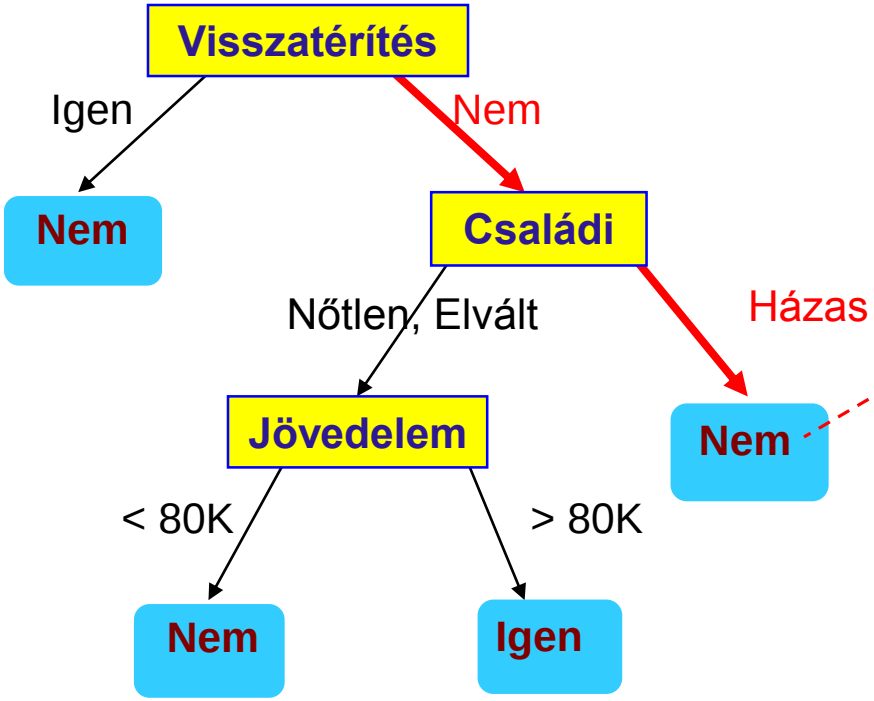
Alkalmazzuk a modellt a teszt adatokra



Teszt adatok

Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
Nem	Házas	80K	?

Alkalmazzuk a modellt a teszt adatokra



Teszt adatok

Vissza-térítés	Családi állapot	Adóköteles jövedelem	Csalás
Nem	Házas	80K	?

A Csalás attribútumhoz rendeljünk ,,Nem”-et

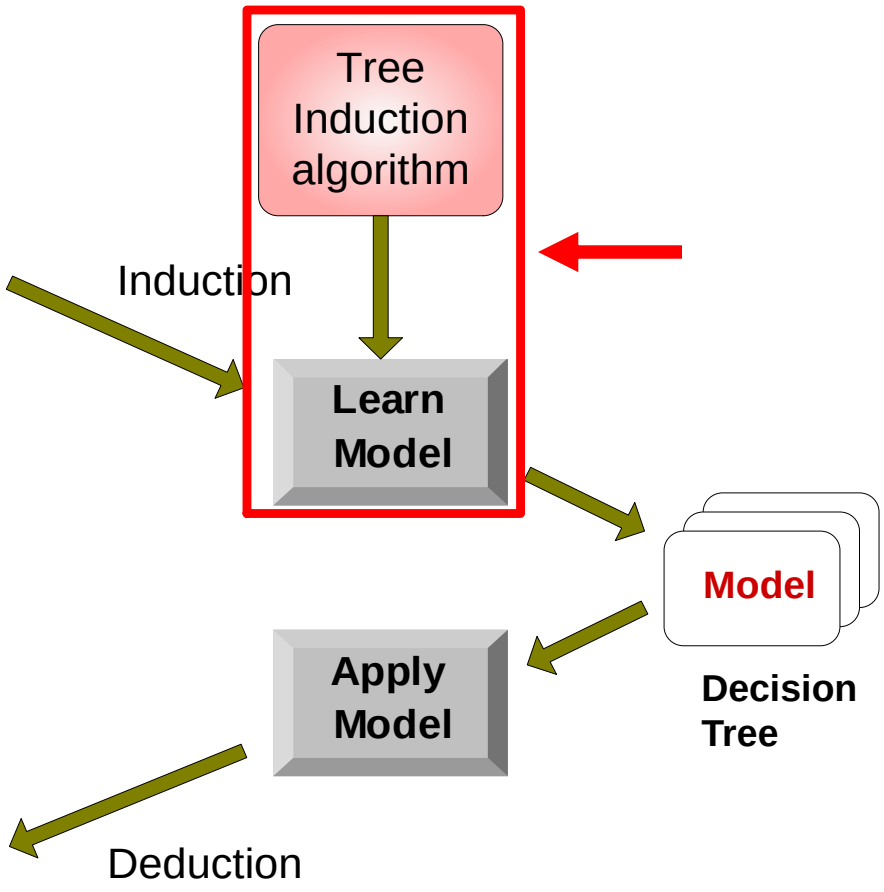
Osztályozás döntési fával

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



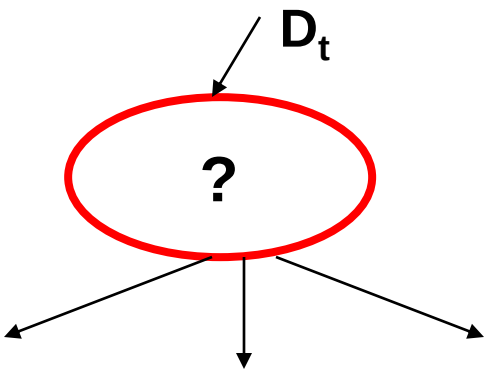
Döntési fa alapú következtetés

- Sok algoritmus:
  - Hunt algoritmusa (az egyik legkorábbi)
  - CART (Classification & Regression Trees -- osztályozási és regressziós fák)
  - ID3 (Interaction Detection), C4.5
  - AID, CHAID (Automatic Interaction Detection, Chi-Square based AID)
  - SLIQ, SPRINT

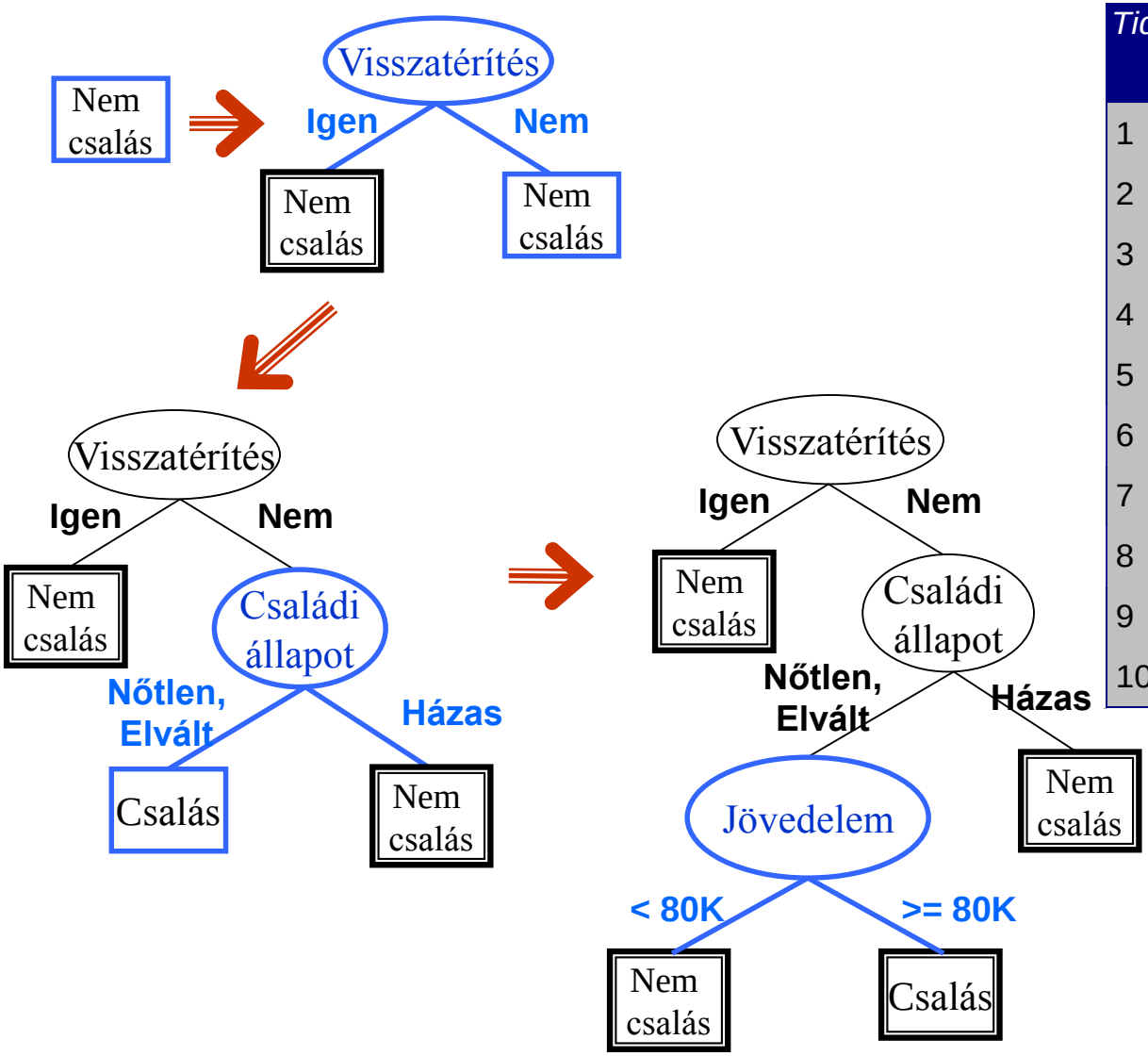
A Hunt algoritmus általános szerkezete

- Legyen D_t a tanító rekordok halmaza a t csúcspontban.
- Általános eljárás:
  - Ha D_t csak olyan rekordokat tartalmaz, amelyek ugyanahhoz az y_t osztályhoz tartoznak, akkor a t csúcspont címkéje legyen y_t .
  - Ha D_t üres halmaz, akkor a t levél kapja az y_d default címkét.
  - Ha D_t egynél több osztályból tartalmaz rekordokat, akkor egy attributum szerinti teszt alapján osszuk fel a rekordok halmazát kisebb részhalmazokra. Majd rekurzívan alkalmazzuk az eljárást minden kapott részhalmazra.

Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
1	Igen	Nőtlen	125K	Nem
2	Nem	Házas	100K	Nem
3	Nem	Nőtlen	70K	Nem
4	Igen	Házas	120K	Nem
5	Nem	Elvált	95K	Igen
6	Nem	Házas	60K	Nem
7	Igen	Elvált	220K	Nem
8	Nem	Nőtlen	85K	Igen
9	Nem	Házas	75K	Nem
10	Nem	Nőtlen	90K	Igen



Hunt algoritmusa



Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
1	Igen	Nőtlen	125K	Nem
2	Nem	Házas	100K	Nem
3	Nem	Nőtlen	70K	Nem
4	Igen	Házas	120K	Nem
5	Nem	Elvált	95K	Igen
6	Nem	Házas	60K	Nem
7	Igen	Elvált	220K	Nem
8	Nem	Nőtlen	85K	Igen
9	Nem	Házas	75K	Nem
10	Nem	Nőtlen	90K	Igen

Tiszta csomópont

Nem tiszta csomópont

Fa alapú következtetés

- Mohó stratégia.
  - Vágjuk részekre a rekordok halmazát egy attributum szerinti teszt alapján egy alkalmas kritériumot optimalizálva.

- Szempontok
  - Hogyan vágjuk részekre a rekordokat?
    - Hogyan határozzuk meg az attributumok szerinti teszt feltételeket?
    - Hogyan határozzuk meg a legjobb vágást?
  - Mikor álljunk le a vágással?

Fa alapú következtetés

- Mohó stratégia.
  - Vágjuk részekre a rekordok halmazát egy attributum szerinti teszt alapján egy alkalmas kritériumot optimalizálva.

- Szempontok
  - Hogyan vágjuk részekre a rekordokat?
    - Hogyan határozzuk meg az attributumok szerinti teszt feltételeket?
    - Hogyan határozzuk meg a legjobb vágást?
  - Mikor álljunk le a vágással?

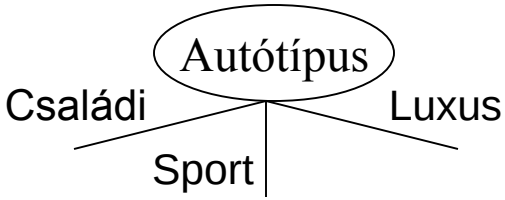
Hogyan határozzuk meg a tesztfeltételt?

- Függ az attributumok típusától:
  - névleges,
  - sorrendi,
  - folytonos (különbségi, hányados).

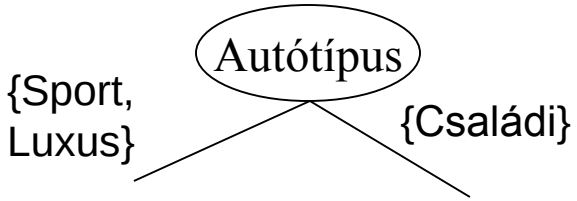
- Függ attól, hogy hány részre vágunk:
  - két részre, ágra (bináris) vágás,
  - több részre, ágra vágás.

Vágás névleges attributum alapján

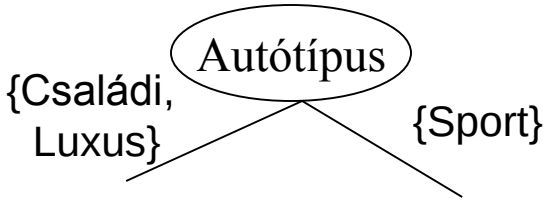
- Több részre vágás: Annyi részt használjunk amennyi különböző érték van.



- Bináris vágás: Osszuk az értékeket két részre. Az optimális partíciót találjuk meg.

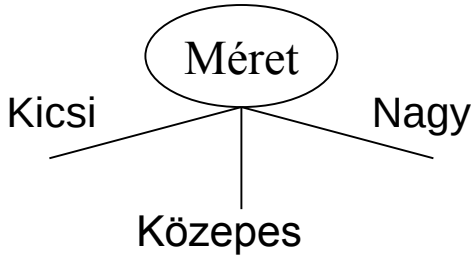


vagy

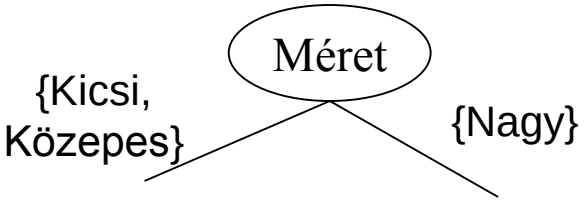


Vágás sorrendi attribútum alapján

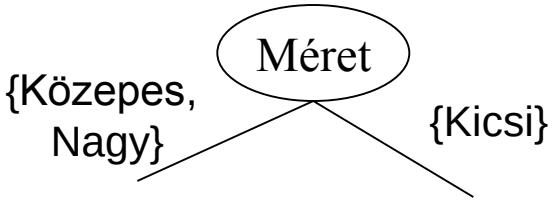
- Több részre vágás : Annyi részt használjunk amennyi különböző érték van.



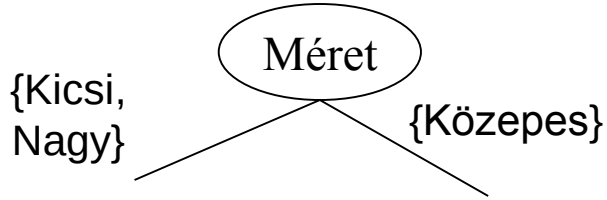
- Bináris vágás: Osszuk az értékeket két részre. Az optimális partíciót találjuk meg.



vagy



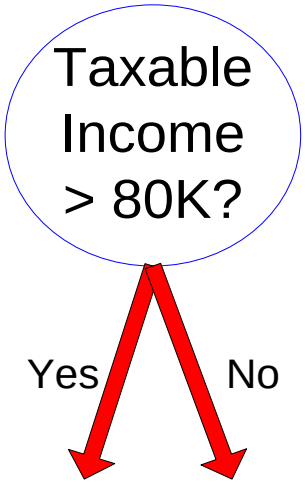
- Mi a helyzet ezzel a vágással?



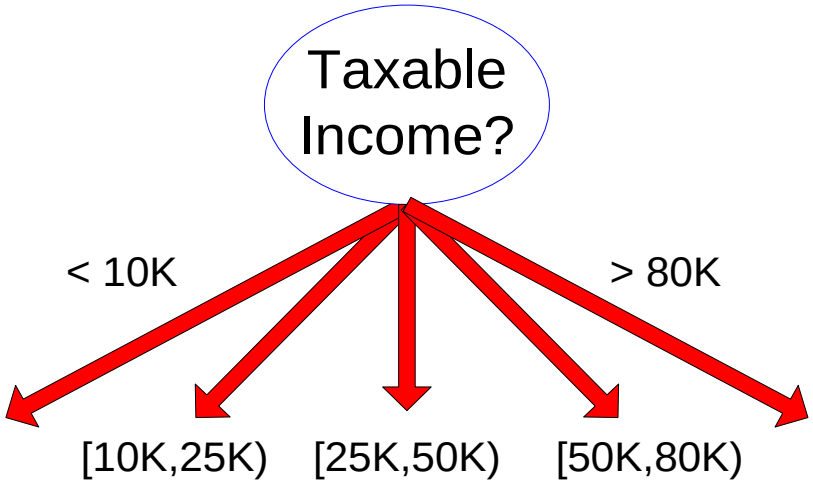
Vágás folytonos attribútum alapján

- Többféle módon kezelhető:
  - Diszkretizáció, hogy sorrendi kategórikus attribútumot állítsunk elő
    - statikus – egyszer, kezdéskor diszkretizálunk,
    - dinamikus – a tartományokat kaphatjuk egyenlő hosszú v. egyenlő gyakoriságú intervallumokra való beosztással illetve klaszterosítással.
  - Bináris döntés: $(A < v)$ vagy $(A \geq v)$
    - Tekintsük az összes lehetséges vágást és találjuk meg a legjobbat.
    - Számításigényes lehet.

Vágás folytonos attribútum alapján



(i) Binary split



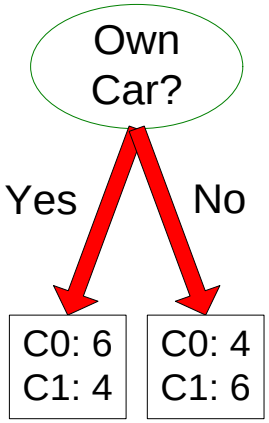
(ii) Multi-way split

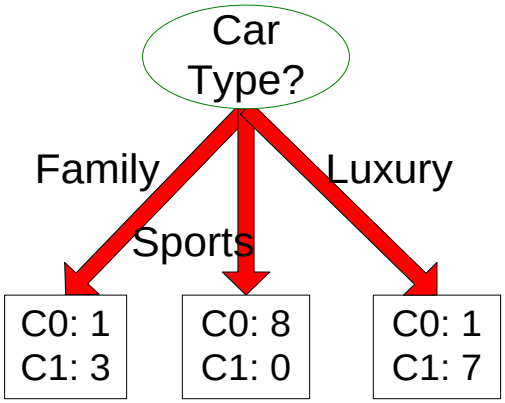
Fa alapú következtetés

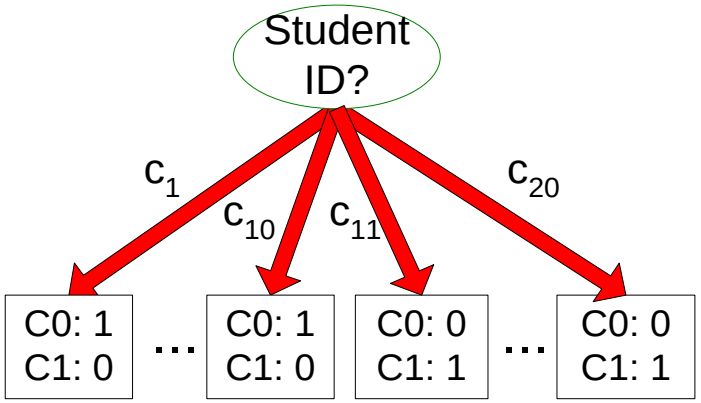
- Mohó stratégia.
  - Vágjuk részekre a rekordok halmazát egy attributum szerinti teszt alapján egy alkalmas kritériumot optimalizálva.
- Szempontok
  - Hogyan vágjuk részekre a rekordokat?
    - Hogyan határozzuk meg az attributumok szerinti teszt feltételeket?
    - Hogyan határozzuk meg a legjobb vágást?
  - Mikor álljunk le a vágással?

Mi lesz a legjobb vágás?

Vágás előtt: 10 rekord a 0 osztályból,
10 rekord az 1 osztályból







Melyik tesztfeltétel a legjobb?

Mi lesz a legjobb vágás?

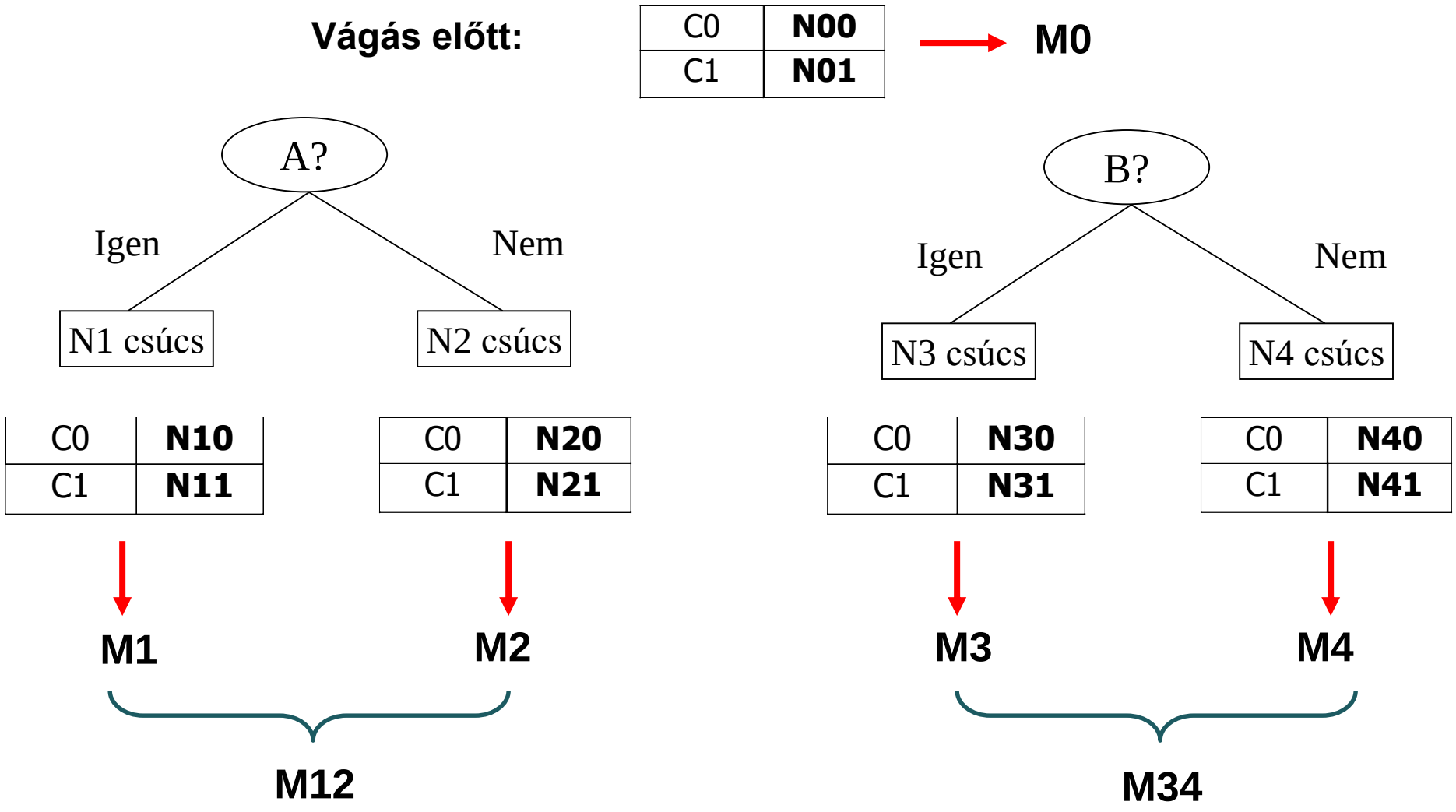
- Mohó megközelítés:
  - A homogén osztály-eloszlást eredményező csúcspontokat preferáljuk.
- Szennyezettségi mérőszámra van szükségünk:

C0: 5 C1: 5	C0: 9 C1: 1
Nem homogén, **nagyon szennyezett**	**Homogén,** **kicsit szennyezett**

Szennyezettség mérése

- Gini index
- Entrópia
- Téves osztályozási hiba

Mi lesz a legjobb vágás?



Szennyezettség mérése: GINI index

- Gini index egy *t* csúcspontban:

$$G(t) = 1 - \sum_j [p(j \mid t)]^2$$

($p(j \mid t)$ a j osztály relatív gyakorisága a t csúcspontban).

  - A maximum $(1 - 1/n_c)$ amikor a rekordok egyenlően oszlanak meg az osztályok között, ahol n_c az osztályok száma (legkevésbé hasznos információ).
  - A minimum *0.0* amikor minden rekord ugyanahhoz az osztályhoz tartozik (leghasznosabb információ).

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

A Gini index számolása

$$G(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)² – P(C2)² = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)² – (5/6)² = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)² – (4/6)² = 0.444

Vágás a Gini index alapján

- A CART, SLIQ, SPRINT algoritmusok használják.
- Ha a *t* csúcspontot (szülő) *k* részre (gyerekek) osztjuk fel, akkor a vágás jóságát az alábbi képlettel számoljuk:

$$G_{vágás} = \sum_{i=1}^{k} \frac{n_i}{n} G(i)$$

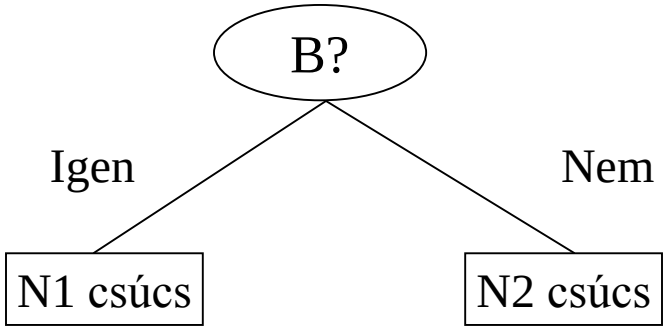
ahol n_i = rekordok száma az *i*–edik gyereknél,

n = rekordok száma a *t* csomópontban,

$G(i)$ = az *i*-edik gyerek Gini indexe.

Gini index bináris attributumokra

- Két ágra vágás
- Az ágak súlyozásának hatása:
  - minél nagyobb és tisztább ágakat keresünk.



	Szülő
C1	**6**
C2	**6**
Gini = 0.500	

G(N1)
= 1 – (5/7)² – (2/7)²
= 0.408

G(N2)
= 1 – (1/5)² – (4/5)²
= 0.32

	N1	N2
C1	**5**	**1**
C2	**2**	**4**
Gini=0.333		

G(gyerek)
= 7/12 * 0.408 +
5/12 * 0.32
= 0.371

Gini index diszkrét attributumokra

- Minden különböző vágó értékre határozzuk meg az egyes osztályok előfordulási gyakoriságát az egyes ágakra.
- Használjuk a gyakorisági mátrixot a döntésnél.

Több ágra vágás

	Autótípus		
	Családi	Sport	Luxus
C1	1	2	1
C2	4	1	1
Gini	0.393		

Bináris vágás
(találjuk meg a legjobb partíciót)

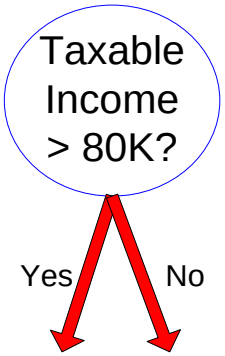
	Autótípus	
	{Sport, Luxus}	{Családi}
C1	3	1
C2	2	4
Gini	0.400	

	Autótípus	
	{Sport}	{Családi, Luxus}
C1	2	2
C2	1	5
Gini	0.419	

Gini index folytonos attributumokra

- Használjunk egy értéken alapuló bináris döntéseket.
- Számos lehetséges vágó érték:
  - Lehetséges vágások száma = Különböző értékek száma
- Mindegyik vágó értékhez tartozik egy gyakorisági mátrix.
  - Az ágak mindegyikében számoljuk össze az $A < v$ és $A \geq v$ osztályok gyakoriságait.
- Heurisztika a legjobb v megtalálására:
  - Minden v-re fésüljük át az adatbázist a gyakorisági mátrix meghatározására és számoljuk ki a Gini indexet.
  - Numerikusan nem hatékony! (Sok ismétlés)

Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
1	Igen	Nőtlen	125K	**Nem**
2	Nem	Házas	100K	**Nem**
3	Nem	Nőtlen	70K	**Nem**
4	Igen	Házas	120K	**Nem**
5	Nem	Elvált	95K	**Igen**
6	Nem	Házas	60K	**Nem**
7	Igen	Elvált	220K	**Nem**
8	Nem	Nőtlen	85K	**Igen**
9	Nem	Házas	75K	**Nem**
10	Nem	Nőtlen	90K	**Igen**



Gini index folytonos attributumokra

- Hatékony számolási algoritmus: mindegyik attributumra
  - Rendezzük az attributumot értékei mentén.
  - Lineárisan végigfésülve ezeket az értékeket mindig frissítsük a gyakorisági mátrixot és számoljuk ki a Gini indexet.
  - Válasszuk azt a vágó értéket, amelynek legkisebb a Gini indexe.

Csalás	Nem	Nem	Nem	Igen	Igen	Igen	Nem	Nem	Nem	Nem
Adóköteles jövedelem										
Rendezett értékek →	60	70	75	85	90	95	100	120	125	220
Vágó értékek →	55	65	72	80	87	92	97	110	122	172	230
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Igen	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
Nem	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

Entrópia alapú vágási kritérium

- Entrópia a *t* csúcsban:

$$E(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

(ahol $p(j \mid t)$ a *j*-edik osztály relatív gyakorisága a *t* csúcsban).

  - Egy csúcs homogenitását méri.
    - Maximuma $log_2 n_c$, amikor a rekordok egyenlően oszlanak meg az osztályok között, ahol n_c az osztályok száma (legrosszabb eset).
    - Minimuma *0.0*, amikor minden rekord egy osztályba tartozik (legjobb eset).
  - Az entrópia számolása hasonló a Gini index számolásához.

Az entrópia számolása

$$E(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entrópia = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entrópia = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entrópia = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Entrópia alapú vágás

- Információ nyereség (INY):

$$INY_{vágás} = E(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} E(i) \right)$$

A *t* szülő csúcsot *k* ágra bontjuk:

n_i a rekordok száma az *i*-edik ágban

  - Az entrópia csökken a vágás miatt. Válasszuk azt a vágást, amelynél a csökkenés a legnagyobb (maximalizáljuk a nyereséget).
  - Az ID3 és C4.5 algoritmusok használják.
  - Hátránya: olyan vágásokat részesít előnyben, amelyek sok kicsi de tiszta ágat hoznak létre.

Entrópia alapú vágás

- Nyereség hányados (NYH):

$$NYH_{vágás} = \frac{I_{vágás}}{VE}$$

$$VE = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

A *p* szülő csúcsot *k* ágra bontjuk:

n_i a rekordok száma az *i*-edik ágban

  - Az információ nyereséget módosítja a vágás entrópiájával (VE). A nagy entrópiájú felbontásokat (sok kis partíció) bünteti!
  - A C4.5 algoritmus használja.
  - Az információ nyereség hátrányainak kiküszöbölésére tervezték.

Téves osztályozási hiba alapú vágás

- Osztályozási hiba a *t* csúcsban :

$$H(t) = 1 - \max_i P(i \mid t)$$

- Egy csúcspontbeli téves osztályozás hibáját méri.
  - Maximuma *1 - 1/n_c,* amikor a rekordok egyenlően oszlanak meg az osztályok között, ahol n_c az osztályok száma (legrosszabb eset).
  - Minimuma *0.0*, amikor minden rekord egy osztályba tartozik (legjobb eset).

Példa a hiba számolására

$$H(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Hiba = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Hiba = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

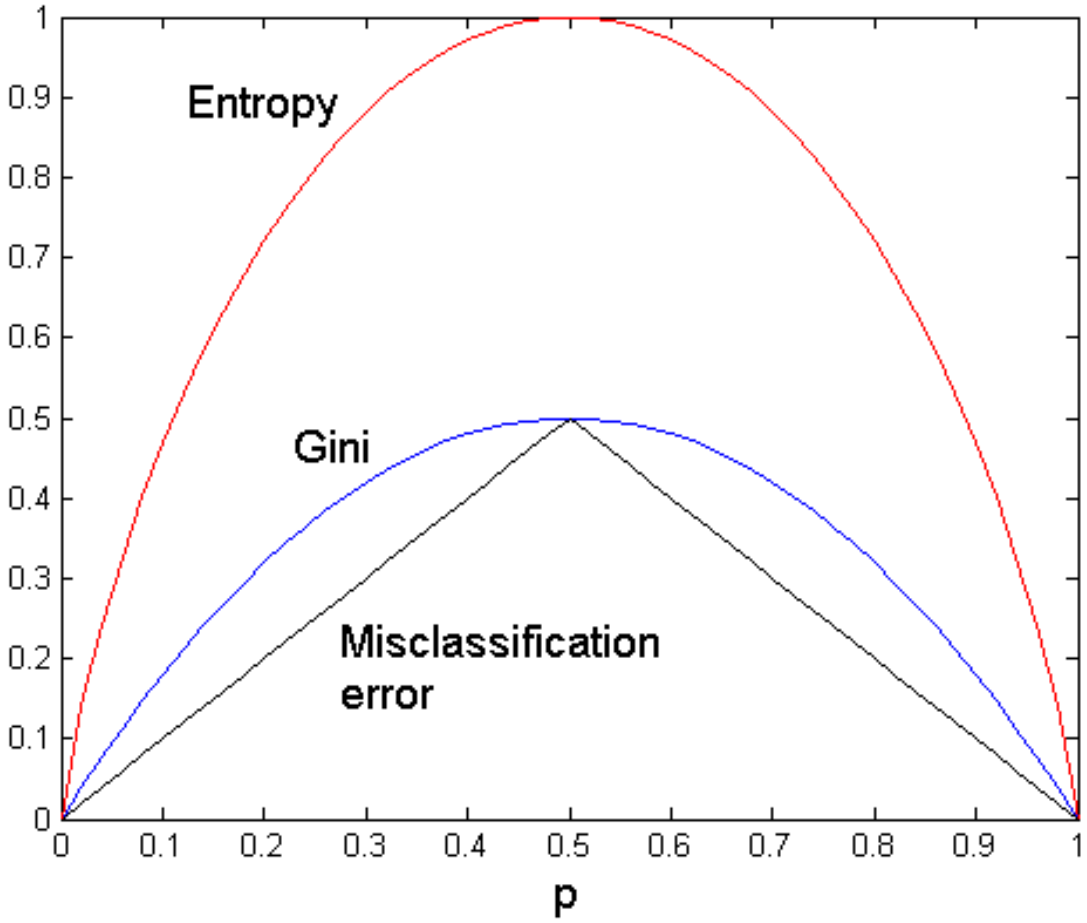
C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

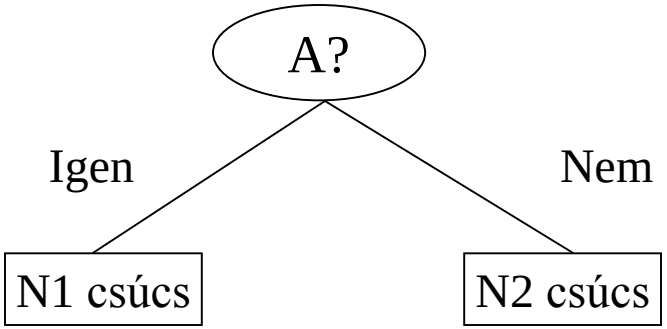
Hiba = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Vágási kritériumok összehasonlítása

Bináris osztályozási feladat:



Téves osztályozás vagy Gini



	Szülő
C1	7
C2	3
Gini = 0.42	

G(N1)
= 1 – (3/3)² – (0/3)²
= 0

G(N2)
= 1 – (4/7)² – (3/7)²
= 0.489

	N1	N2
C1	3	4
C2	0	3
Gini=0.361		

G(gyerek)
= 3/10 * 0
+ 7/10 * 0.489
= 0.342

A Gini javít, a másik nem !!

Fa alapú következtetés

- Mohó stratégia.
  - Vágjuk részekre a rekordok halmazát egy attributum szerinti teszt alapján egy alkalmas kritériumot optimalizálva.
- Szempontok
  - Hogyan vágjuk részekre a rekordokat?
    - Hogyan határozzuk meg az attributumok szerinti teszt feltételeket?
    - Hogyan határozzuk meg a legjobb vágást?
  - Mikor álljunk le a vágással?

Megállási szabály döntési fáknál

- Ne osszunk tovább egy csúcsot ha minden rekord ugyanahhoz az osztályhoz tartozik.
- Ne osszunk tovább egy csúcsot ha minden rekordnak hasonló attributum értékei vannak.
- Korai megállás (később tárgyaljuk).

Döntési fa alapú osztályozás

- Előnyök:
  - Kis költséggel állíthatóak elő.
  - Kimagaslóan gyors új rekordok osztályozásánál.
  - A kis méretű fák könnyen interpretálhatóak.
  - Sok egyszerű adatállományra a pontosságuk összemérhető más osztályozási módszerekével.

Példa: C4.5

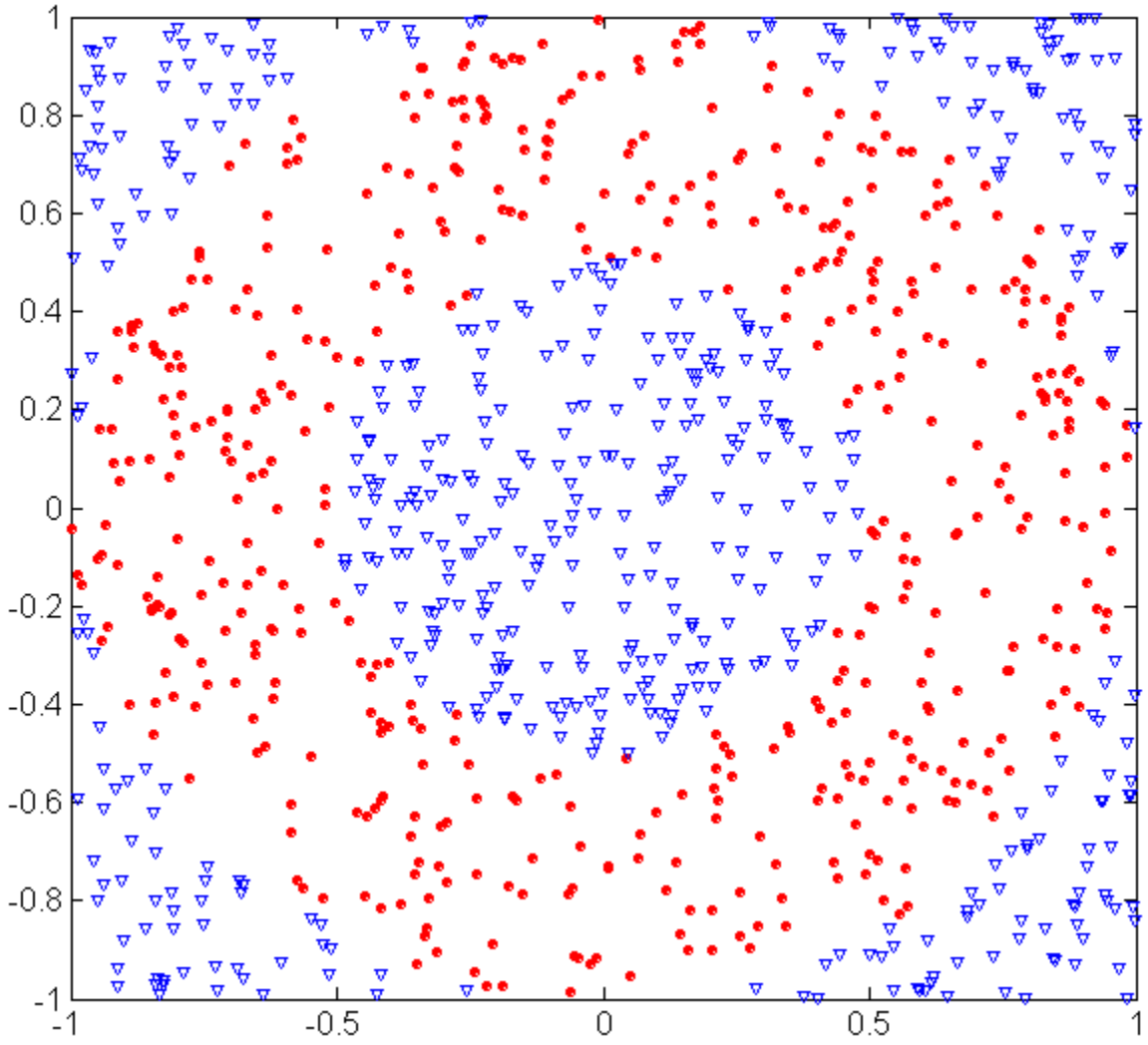
- Egyszerű, egy mélységű keresés.
- Információ nyereséget használ.
- Minden csúcsnál rendezi a folytonos attributumokat.
- Az összes adatot a memóriában kezeli.
- Alkalmatlan nagy adatállományok kezelésére.
  - Memórián kívüli rendezést igényel (lassú).

- Szoftver letölthető az alábbi címről: http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz

Az osztályozás gyakorlati szempontjai

- Alul- és túlillesztés
- Hiányzó értékek
- Az osztályozás költsége

Példa alul- és túlillesztésre



500 piros kör és 500 kék háromszög

Piros körök:

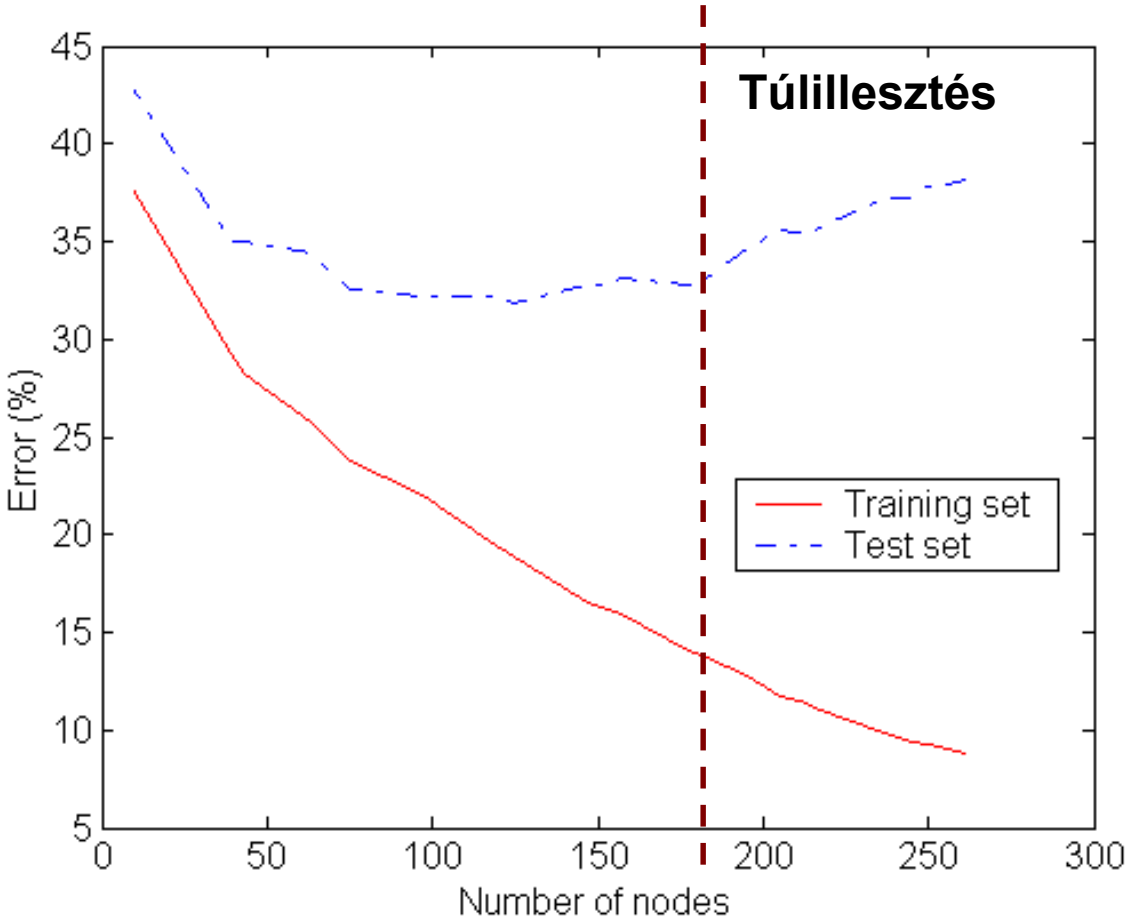
$0.5 \leq \text{sqrt}(x_1^2+x_2^2) \leq 1$

Kék háromszögek:

$\text{sqrt}(x_1^2+x_2^2) > 0.5$ **or**

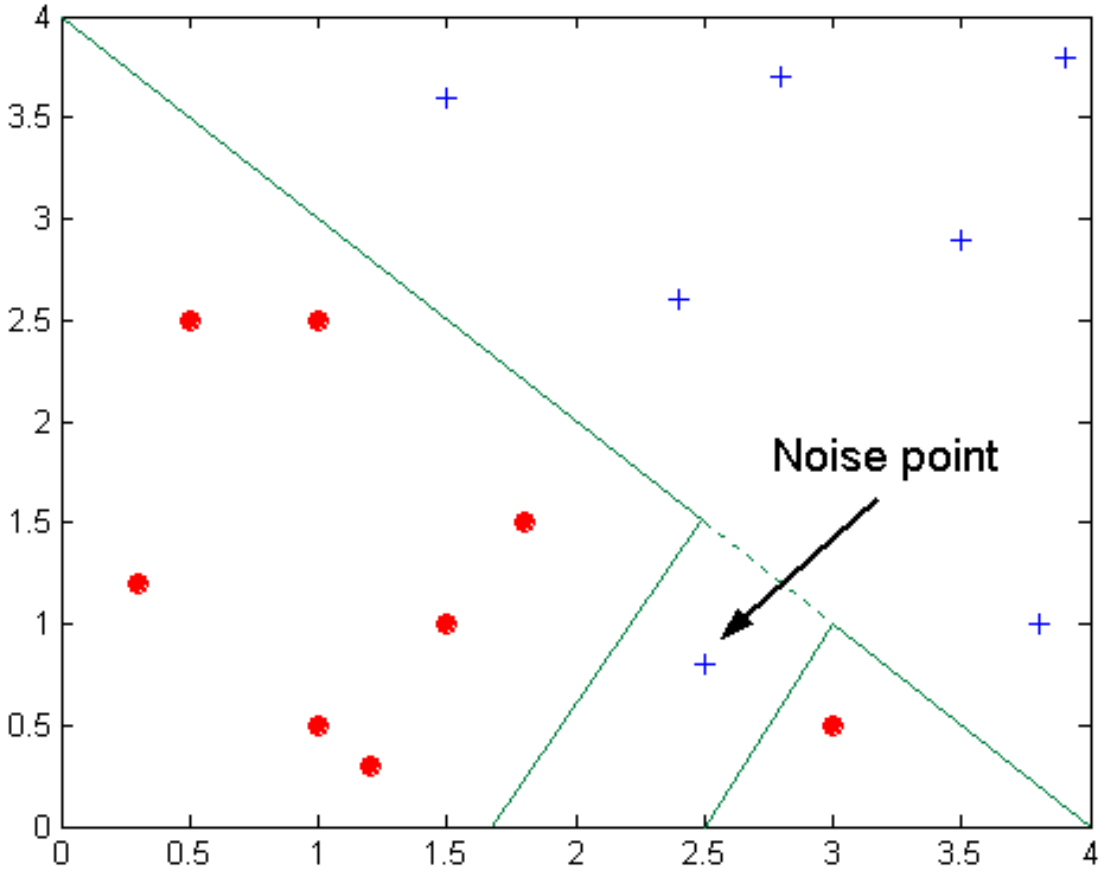
$\text{sqrt}(x_1^2+x_2^2) < 1$

Alul- és túlillesztés



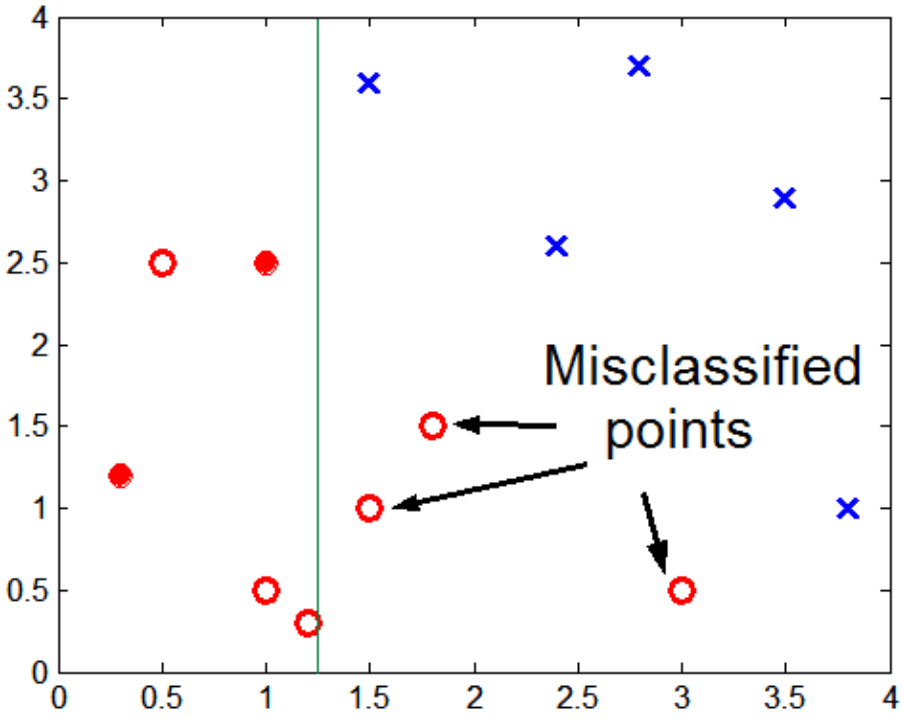
Alulillesztés: amikor a modell túl egyszerű a tanító és a teszt hiba egyaránt nagy

Túlillesztés hiba miatt



A döntési határ torzul a zaj miatt

Túlillesztés elégtelen minta miatt



Nehéz helyesen előrejelezni az ábra alsó felében lévő pontokat mivel azon a területen nincsenek adatok.

- Elégtelen számú tanító rekord egy területen azt okozhatja, hogy a döntési fa olyan tanító rekordok alapján prediktál a teszt példákra, amelyek az osztályozási feladat számára irrelevánsak.

Túlillesztés: megjegyzések

- A túlillesztés döntési fák esetén azt eredményezheti, hogy a fa szükségtelenül nagy (összetett) lesz.
- A tanítás hibája nem ad jó becslést arra hogyan fog működni a fa új rekordokra.
- A hiba becslésére új módszerek kellenek.

Az általánosítási hiba becslése

- Behelyettesítési hiba: hiba a tanító állományon (Σ e(t))
- Általánosítási hiba: hiba a teszt állományon (Σ e'(t))
- Módszerek az általánosítási hiba becslésére:
  - Optimista megközelítés: e'(t) = e(t)
  - Pesszimista megközelítés:
    - Minden levélre: e'(t) = (e(t)+0.5)
    - Teljes hiba: e'(T) = e(T) + N $\times$ 0.5 (N: levelek száma)
    - Egy 30 levelű fára 10 tanítási hiba mellett (1000 rekord):

      Tanítási hiba = 10/1000 = 1%

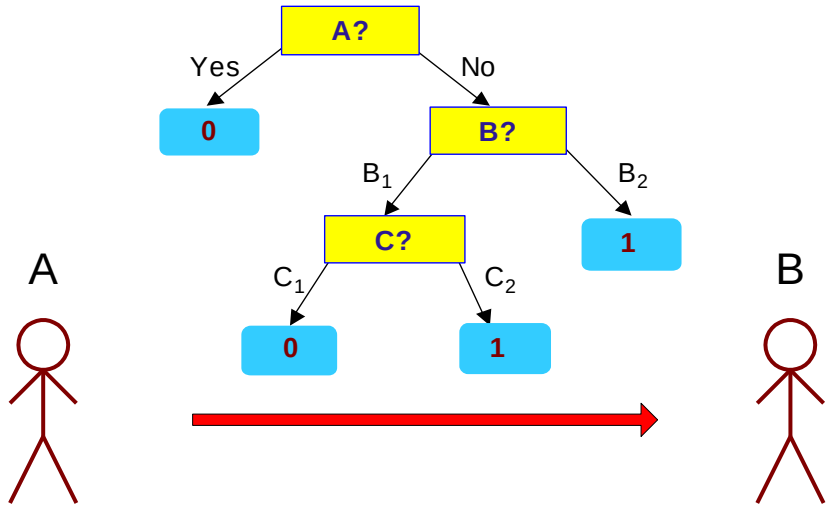
      Általánosítási hiba = (10 + 30$\times$0.5)/1000 = 2.5%
  - Hiba csökkentés tisztítással (REP – reduced error pruning):
    - használjunk egy ellenőrző adatállományt az általánosítási hiba becslésére.

Occam borotvája

- Két hasonló általánosítási hibájú modell esetén az egyszerűbbet részesítjük előnyben a bonyolultabbal szemben.
- Bonyolult modelleknél nagyobb az esélye annak, hogy az csak véletlenül illeszkedik az adatokban lévő hiba miatt.
- Ezért figyelembe kell venni a modell komplexitását amikor kiértékeljük.

Minimális leíró hossz (MDL)

X	y
X_1	1
X_2	0
X_3	0
X_4	1
…	…
X_n	1



X	y
X_1	?
X_2	?
X_3	?
X_4	?
…	…
X_n	?

- Költség(Modell,Adat)=Költség(Adat|Modell)+Költség(Modell)
  - Költség: a kódoláshoz szükséges bitek száma.
  - A legkisebb költségű modellt keressük.
- Költség(Adat|Modell) a téves osztályozás hibáját kódolja.
- Költség(Modell) a fát, csúcsokat és leveleket (azok számát) és a vágási feltételeket kódolja.

Hogyan kezeljük a túlillesztést

- Előtisztítás (korai megállási szabály)
  - Állítsuk le az algoritmust mielőtt a fa teljes nem lesz.
  - Jellegzetes megállási szabályok egy csúcsban:
    - Álljunk meg, ha minden rekord ugyanahhoz az osztályhoz tartozik.
    - Álljunk meg, ha az összes attributum értéke egyenként azonos.
  - További megszorító feltételek:
    - Álljunk meg, ha a rekordok száma kisebb egy a felhasználó által meghatározott értéknél.
    - Álljunk meg, ha az osztályok eloszlása a rekordokon független a célváltozótól (használjunk pl. χ^2 próbát).
    - Álljunk meg, ha az aktuális csúcspont vágása nem javítja a szennyezettség mértékét (pl. a Gini indexet vagy az információ nyereséget).

Hogyan kezeljük a túlillesztést

- Utótisztítás
  - Építsük fel a teljes döntési fát.
  - Metszük a fát alulról felfelé bizonyos csúcspontokban vágva.
  - Ha javul az általánosítási hiba a metszés után, akkor helyettesítsük a levágott részfát egy levéllel.
  - Ennek a levélnek az osztály címkéjét a levágott részfabeli rekordok osztályai alapján többségi elvet alkalmazva kapjuk.
  - Az MDL elvet is használhatjuk utótisztításra.

Példa utótisztításra

Osztály = Igen	20
Osztály = Nem	10
Hiba = 10/30	

Tanítási hiba (vágás előtt) = 10/30

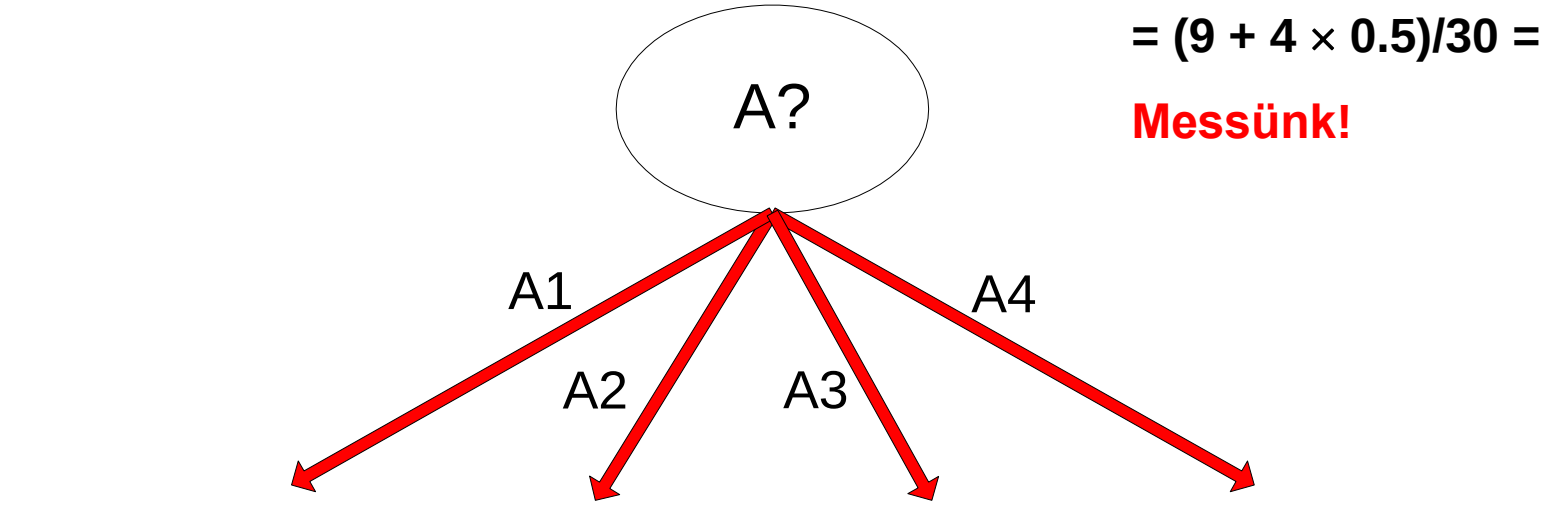
Pesszimista hiba (vágás előtt) = (10 + 0.5)/30 = 10.5/30

Tanítási hiba (vágás után) = 9/30

Pesszimista hiba (vágás után)

= (9 + 4 × 0.5)/30 = 11/30

Messünk!



Osztály = Igen	8
Osztály = Nem	4

Osztály = Igen	3
Osztály = Nem	4

Osztály = Igen	4
Osztály = Nem	1

Osztály = Igen	5
Osztály = Nem	1

Példa utótisztításra

- Optimista hiba?

  Egyik esetben se messünk

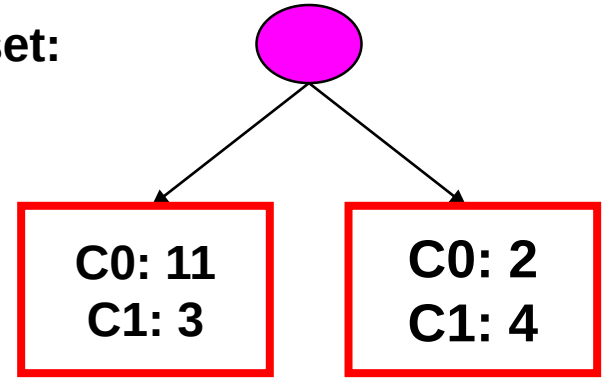
- Pesszimista hiba?

  Ne messünk az 1. esetben, messünk a 2.-ban

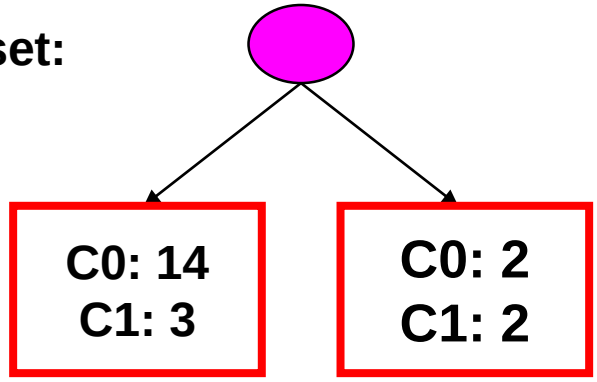
- Hiba csökkentés tisztítással?

  Függ az ellenőrző állománytól

1. eset:



2. eset:



Hiányzó attributum értékek kezelése

- A hiányzó értékek három különböző módon befolyásolják a döntési fa konstrukcióját:
  - Hogyan számoljuk a szennyezettségi mutatókat?
  - Hogyan oszlanak el a hiányzó értékeket tartalmazó rekordok a gyerek csúcsok között?
  - Hogyan osztályozzuk a hiányzó értékeket tartalmazó teszt rekordokat?

Szennyezettségi mutató számolása

Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
1	Igen	Nőtlen	125K	Nem
2	Nem	Házas	100K	Nem
3	Nem	Nőtlen	70K	Nem
4	Igen	Házas	120K	Nem
5	Nem	Elvált	95K	Igen
6	Nem	Házas	60K	Nem
7	Igen	Elvált	220K	Nem
8	Nem	Nőtlen	85K	Igen
9	Nem	Házas	75K	Nem
10	?	Nőtlen	90K	Igen

Hiányzó érték

Vágás előtt:

Entrópia(Szülő)
= -0.3 log(0.3)-(0.7)log(0.7) = 0.8813

	Csalás = Igen	Csalás = Nem
Visszatérítés=Igen	**0**	**3**
Visszatérítés=Nem	**2**	**4**
Visszatérítés=?	**1**	**0**

Vágás Visszatérítés mentén:

Entrópia(Visszatérítés=Igen) = 0

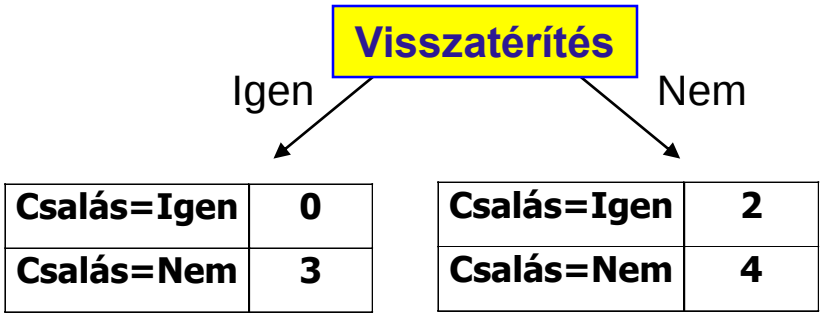
Entrópia(Visszatérítés=Nem)
= -(2/6)log(2/6) – (4/6)log(4/6) = 0.9183

Entrópia(gyerek)
= 0.3 (0) + 0.6 (0.9183) = 0.551

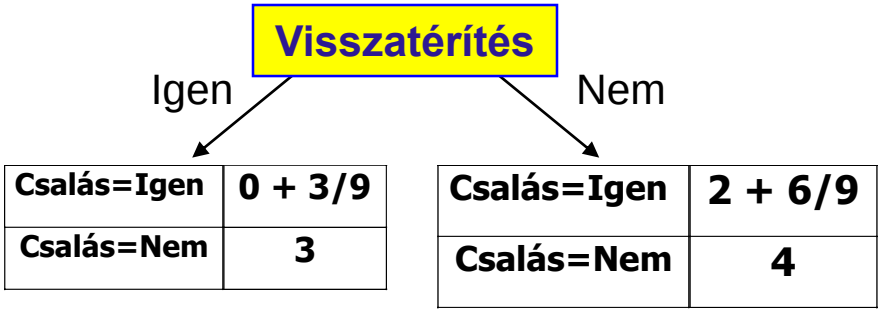
Nyereség = 0.9 × (0.8813 – 0.551) = 0.3303

Rekordok eloszlása

Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
1	Igen	Nőtlen	125K	Nem
2	Nem	Házas	100K	Nem
3	Nem	Nőtlen	70K	Nem
4	Igen	Házas	120K	Nem
5	Nem	Elvált	95K	Igen
6	Nem	Házas	60K	Nem
7	Igen	Elvált	220K	Nem
8	Nem	Nőtlen	85K	Igen
9	Nem	Házas	75K	Nem



Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
10	?	Nőtlen	90K	Igen



A Visszatérítés=Igen valószínűsége 3/9

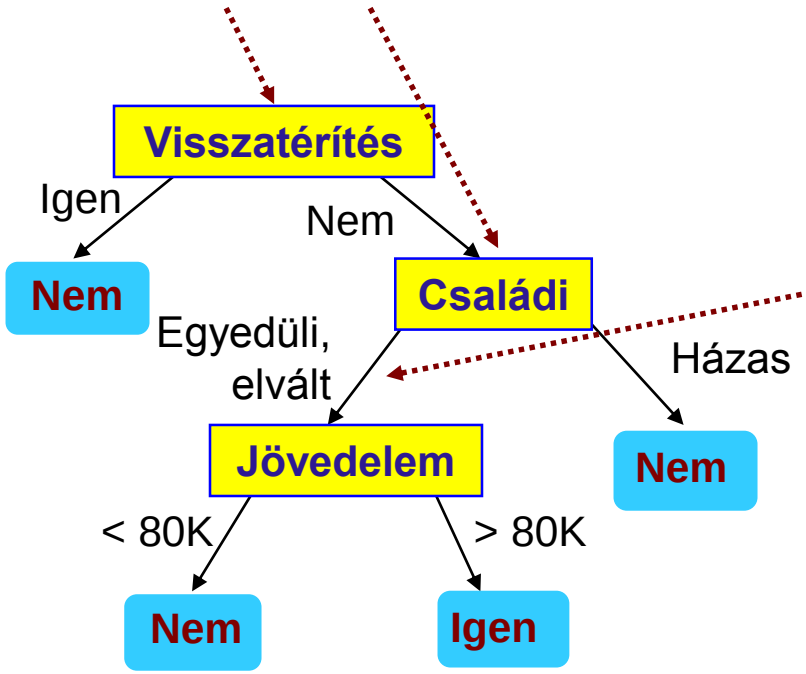
A Visszatérítés=Nem valószínűsége 6/9

Rendeljük a rekordot a bal csúcshoz 3/9 súllyal és 6/9 súllyal a jobb csúcshoz.

Rekordok osztályozása

Új rekord:

Tid	Vissza-térítés	Családi állapot	Jöve-delem	Csalás
11	Nem	?	85K	?



	Házas	Nőtlen	Elvált	Összes
Csalás=Nem	3	1	0	4
Csalás=Igen	6/9	1	1	2.67
Összes	3.67	2	1	6.67

A Családi állapot = Házas valószínűsége 3.67/6.67

A Családi állapot={Nőtlen, Elvált} valószínűsége 3/6.67

További szempontok

- Adat-töredezettség
- Keresési stratégiák
- Kifejezőképesség
- Fa ismétlődés

Adat-töredezettség

- A rekordok száma egyre kevesebb lesz ahogy lefelé haladunk a fában.
- A levelekbe eső rekordok száma túl kevés lehet ahhoz, hogy statisztikailag szignifikáns döntést hozzunk.

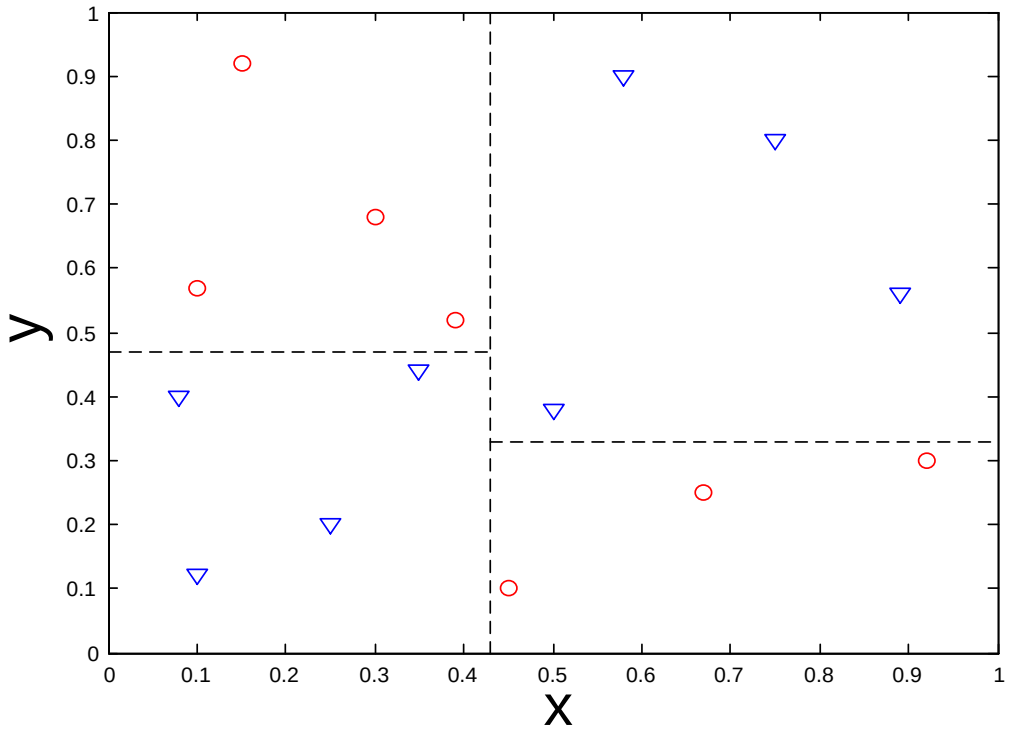
Keresési stratégiák

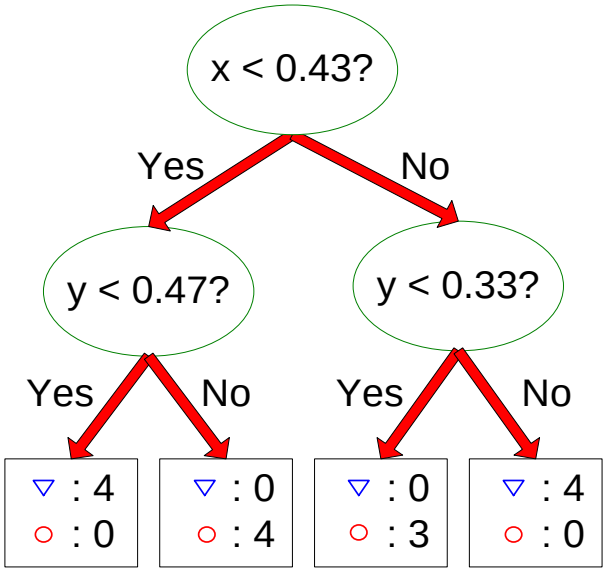
- Az (egy) optimális döntési fa megtalálása NP-nehéz feladat.
- Az eddig bemutatott algoritmusok mohó, fentről lefelé haladó rekurzív partícionáló stratégiák, melyek elfogadható megoldást eredményeznek.
- Más stratégiák?
  - Lentről felfelé
  - Kétirányú
  - Sztochasztikus

Kifejezőképesség

- A döntési fa kifejező reprezentációt ad diszkrét értékű függvények tanításánál.
  - Azonban nem általánosítható jól bizonyos logikai (Boole) függvények esetén.
    - Példa: paritás függvény
      - Osztály = 1 ha páros számú olyan attributum van, amely igaz
      - Osztály = 0 ha páratlan számú olyan attributum van, amely hamis
    - Pontos modellhez egy teljes fára van szükségünk.
- Nem elég kifejező folytonos változók modellezésénél.
  - Különösen ha a teszt feltétel egyszerre csak egy attributumot tartalmaz.

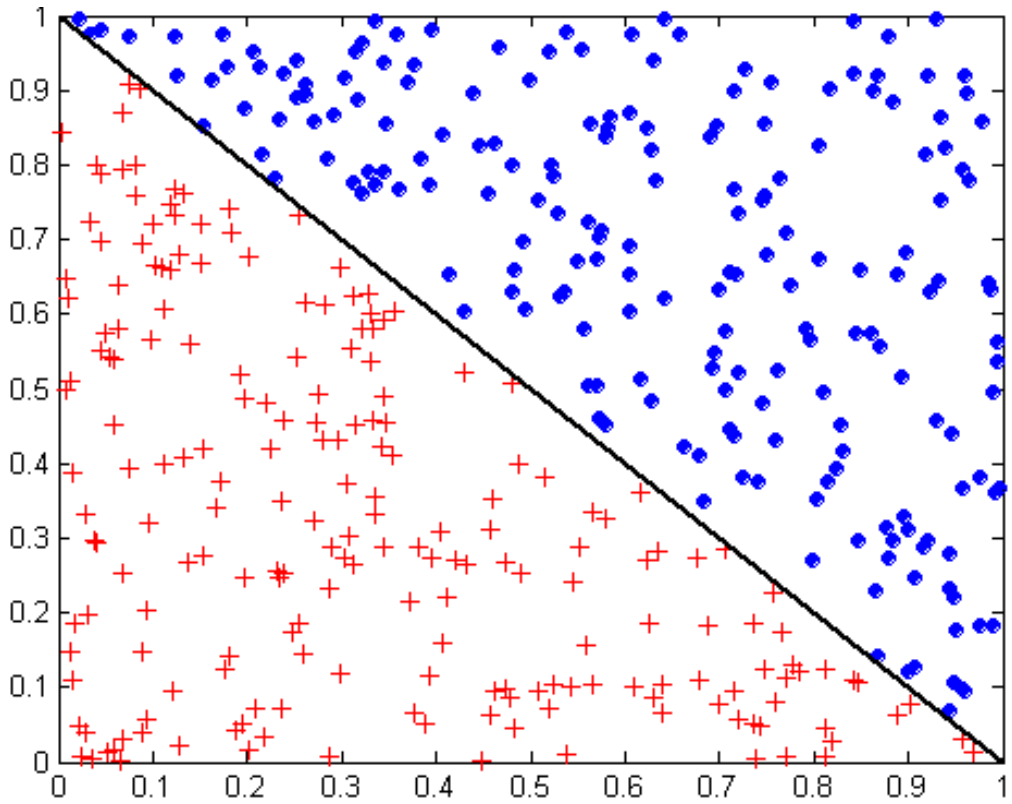
Döntési határ

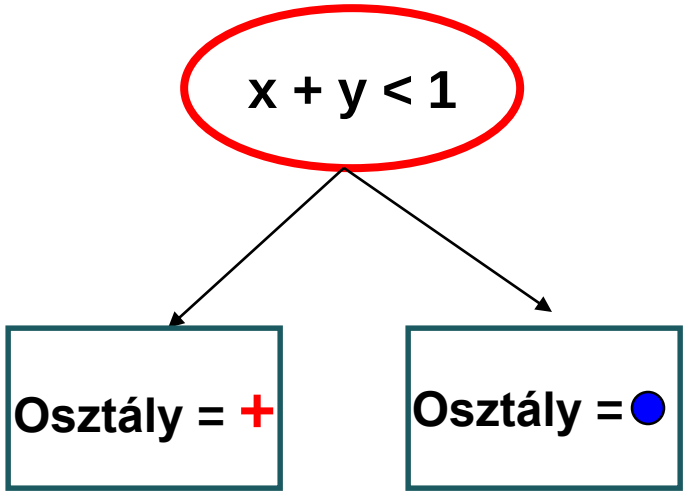




- **Két különböző osztályhoz tartozó szomszédos tartomány közötti határvonalat döntési határnak nevezzük.**
- **A döntési határ párhozamos a tengelyekkel mivel a teszt feltétel egy időben csak egy attributumot tartalmaz.**

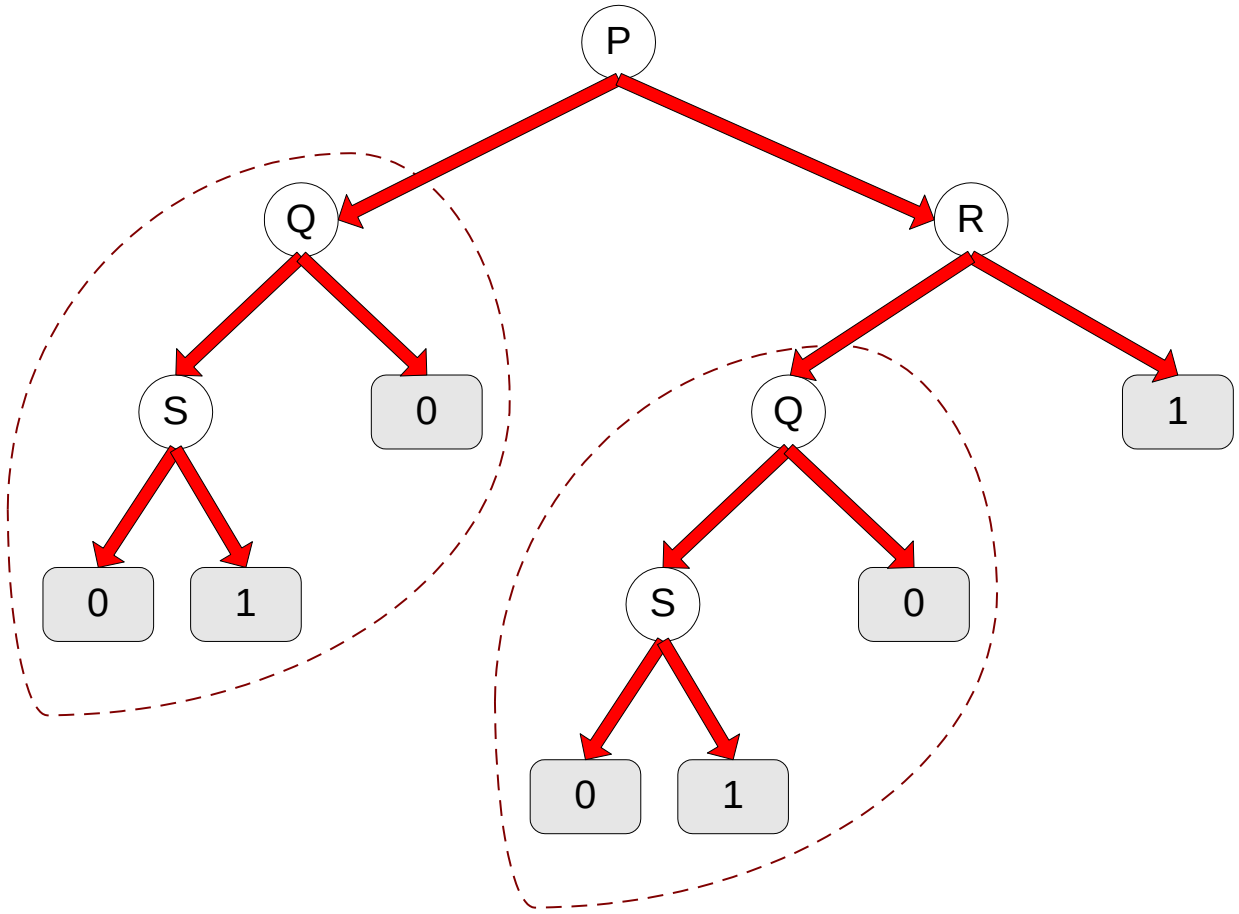
Ferde döntési fa





- **A teszt feltétel több attributumot is tartalmazhat.**
- **Kifejezőbb reprezentáció**
- **Az optimális teszt feltétel megtalálása számítás igényes.**

Fa ismétlődés



- **Ugyanaz a részfa fordul elő több ágban.**

Modellek kiértékelése

- Metrikák hatékonyság kiértékelésre
  - Hogyan mérhetjük egy modell hatékonyságát?
- Módszerek a hatékonyság kiértékelésére
  - Hogyan kaphatunk megbízható becsléseket?
- Módszerek modellek összehasonlítására
  - Hogyan hasonlíthatjuk össze a versenyző modellek relatív hatékonyságát?

Modellek kiértékelése

- Metrikák hatékonyság kiértékelésre
  - Hogyan mérhetjük egy modell hatékonyságát?
- Módszerek a hatékonyság kiértékelésére
  - Hogyan kaphatunk megbízható becsléseket?
- Módszerek modellek összehasonlítására
  - Hogyan hasonlíthatjuk össze a versenyző modellek relatív hatékonyságát?

Metrikák hatékonyság kiértékelésre

- A hangsúly a modellek prediktív képességén van
  - szemben azzal, hogy milyen gyorsan osztá-lyoz vagy épül a modell, skálázható-e stb.
- Egyetértési mátrix:

	Előrejelzett osztály		
Aktuális osztály		Osztály= Igen	Osztály= Nem
	Osztály= Igen	a	b
	Osztály= Nem	c	d

a: TP (igaz pozitív)

b: FN (hamis negatív)

c: FP (hamis pozitív)

d: TN (igaz negatív)

Metrikák hatékonyság kiértékelésre

		Előrejelzett osztály	
Aktuális osztály		Osztály= Igen	Osztály= Nem
	Osztály= Igen	a (TP)	b (FN)
	Osztály= Nem	c (FP)	d (TN)

- Leggyakrabban használt metrika:

$$\text{Pontosság} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

A pontosság határai

- Tekintsünk egy bináris osztályozási feladatot:
  - a 0 osztályba tartozó rekordok száma = 9990,
  - az 1 osztályba tartozó rekordok száma = 10.

- Ha a modell minden rekordot a 0 osztályba sorol, akkor a pontosság 9990/10000 = 99.9 %.
  - A pontosság félrevezető mivel a modell az 1 osztályból egyetlen rekordot sem vesz figyelembe.

Költségmátrix

		Előrejelzett osztály	
Aktuális osztály	C(i|j)	**Osztály = Igen**	**Osztály = Nem**
	Osztály = Igen	C(Igen|Igen)	C(Nem|Igen)
	Osztály = Nem	C(Igen|Nem)	C(Nem|Nem)

C(i|j): a téves osztályozás költsége, a *j* osztályba eső rekordot az *i* osztályba soroljuk

Osztályozás költségének kiszámolása

Költség mátrix	Előrejelzett osztály		
	C(i|j)	+	-
Aktuális osztály	+	-1	100
	-	1	0

M_1 modell	Előrejelzett osztály		
		+	-
Aktuális osztály	+	150	40
	-	60	250

Pontosság = 80%

Költség = 3910

Model M_2	Előrejelzett osztály		
		+	-
Aktuális osztály	+	250	45
	-	5	200

Pontosság = 90%

Költség = 4255

Költség vagy pontosság

Darab	Előrejelzett osztály		
Aktuális osztály		Osztály = Igen	Osztály = Nem
	Osztály = Igen	a	b
	Osztály = Nem	c	d

Költség	Előrejelzett osztály		
Aktuális osztály		Osztály = Igen	Osztály = Nem
	Osztály = Igen	p	q
	Osztály = Nem	q	p

A pontosság arányos a költséggel ha

1. C(Igen|Nem)=C(Nem|Igen) = q
2. C(Igen|Igen)=C(Nem|Nem) = p

$N = a + b + c + d$

$\text{Pontosság} = (a + d)/N$

$\text{Költség} = p\,(a + d) + q\,(b + c)$

$= p\,(a + d) + q\,(N - a - d)$

$= q\,N - (q - p)(a + d)$

$= N\,[q - (q-p) \times \text{Pontosság}]$

Költség-érzékeny mutatók

$$\text{Pozitív pontosság} \quad p = \frac{a}{a+c}$$

$$\text{Pozitív emlékezet} \quad r = \frac{a}{a+b}$$

$$F \text{ mutató} \quad F = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- A pozitív pontosság torzított a C(Igen|Igen) és C(Igen|Nem) felé
- A pozitív emlékezet torzított a C(Igen|Igen) és C(Nem|Igen) felé
- Az *F* mutató torzított a C(Nem|Nem) kivételével az összes felé

$$\text{Súlyozott pontosság} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Modellek kiértékelése

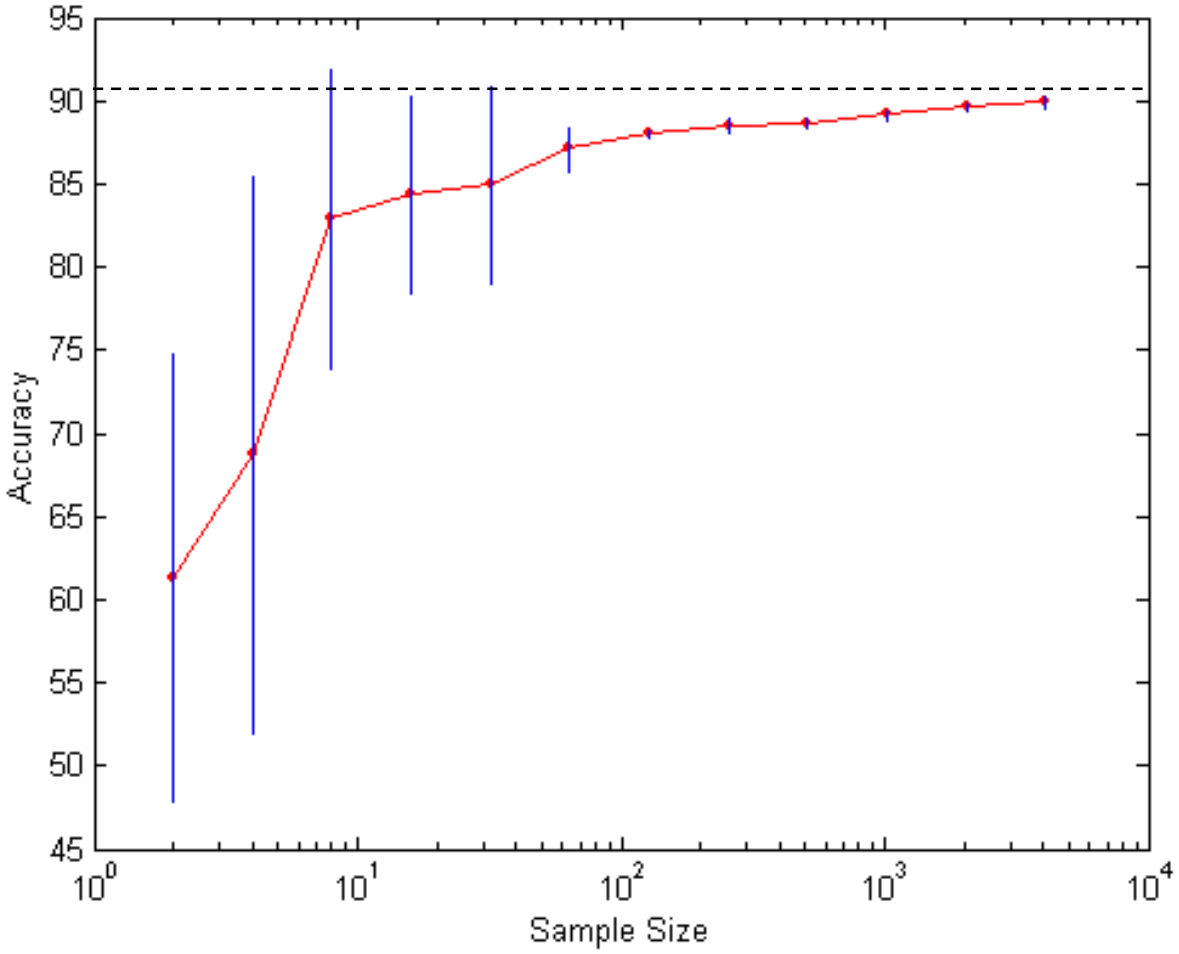
- Metrikák hatékonyság kiértékelésre
  - Hogyan mérhetjük egy modell hatékonyságát?
- Módszerek a hatékonyság kiértékelésére
  - Hogyan kaphatunk megbízható becsléseket?
- Módszerek modellek összehasonlítására
  - Hogyan hasonlíthatjuk össze a versenyző modellek relatív hatékonyságát?

Módszerek hatékonyság kiértékelésére

- Hogyan kaphatunk megbízható becslést a hatékonyságra?

- Egy modell hatékonysága a tanító algoritmus mellett más faktoroktól is függhet:
  - osztályok eloszlása,
  - a téves osztályozás költsége,
  - a tanító és tesz adatállományok mérete.

Tanulási görbe



- A tanulási görbe mutatja hogyan változik a pontosság a mintanagyság függvényében
- Mintavételi ütemterv szükséges a tanulási görbe elkészítéséhez:
  - Aritmetikai mintavétel (Langley & tsai)
  - Geometriai mintavétel (Provost & tsai)
  - A kis minta hatása:
    - Torzítás
    - Variancia

Becslési módszerek

- Felosztás
  - Tartsuk fenn a 2/3 részt tanításra, az 1/3 részt tesztelésre.
- Véletlen részminták
  - Ismételt felosztás
- Keresztellenőrzés
  - Osszuk fel az adatállományt *k* diszjunkt részhalmazra.
  - Tanítsunk *k-1* partíción, teszteljünk a fennmaradón.
  - Hagyjunk ki egyet: *k=n* (diszkriminancia analízis).
- Rétegzett mintavétel
  - Felül- vagy alulmintavételezés
- Bootstrap
  - Visszatevéses mintavétel

Modellek kiértékelése

- Metrikák hatékonyság kiértékelésre
  - Hogyan mérhetjükegy modell hatékonyságát?
- Módszerek a hatékonyság kiértékelésére
  - Hogyan kaphatunk megbízható becsléseket?
- Módszerek modellek összehasonlítására
  - Hogyan hasonlíthatjuk össze a versenyző modellek relatív hatékonyságát?

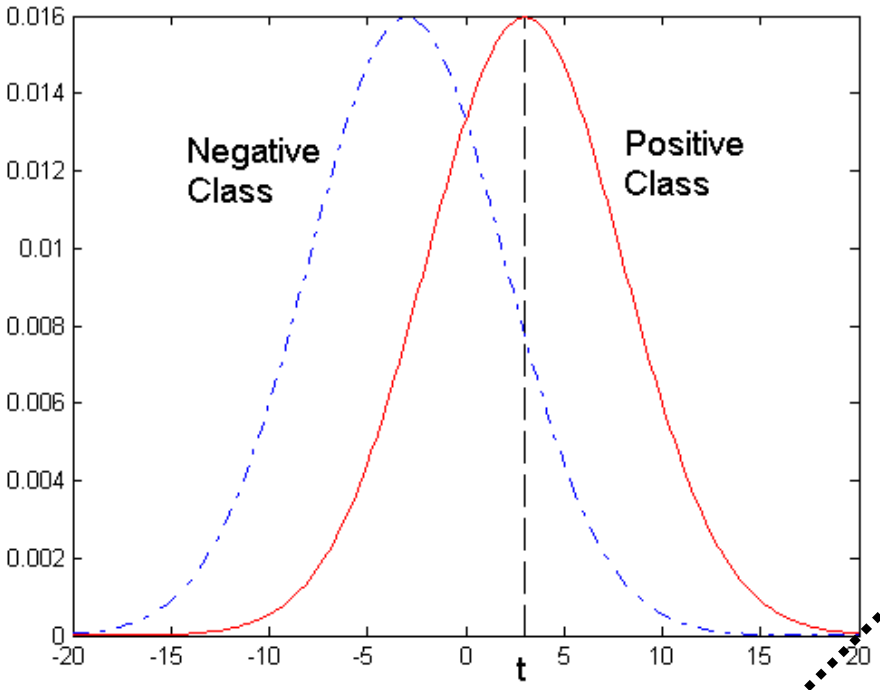
ROC (Receiver Operating Characteristic)

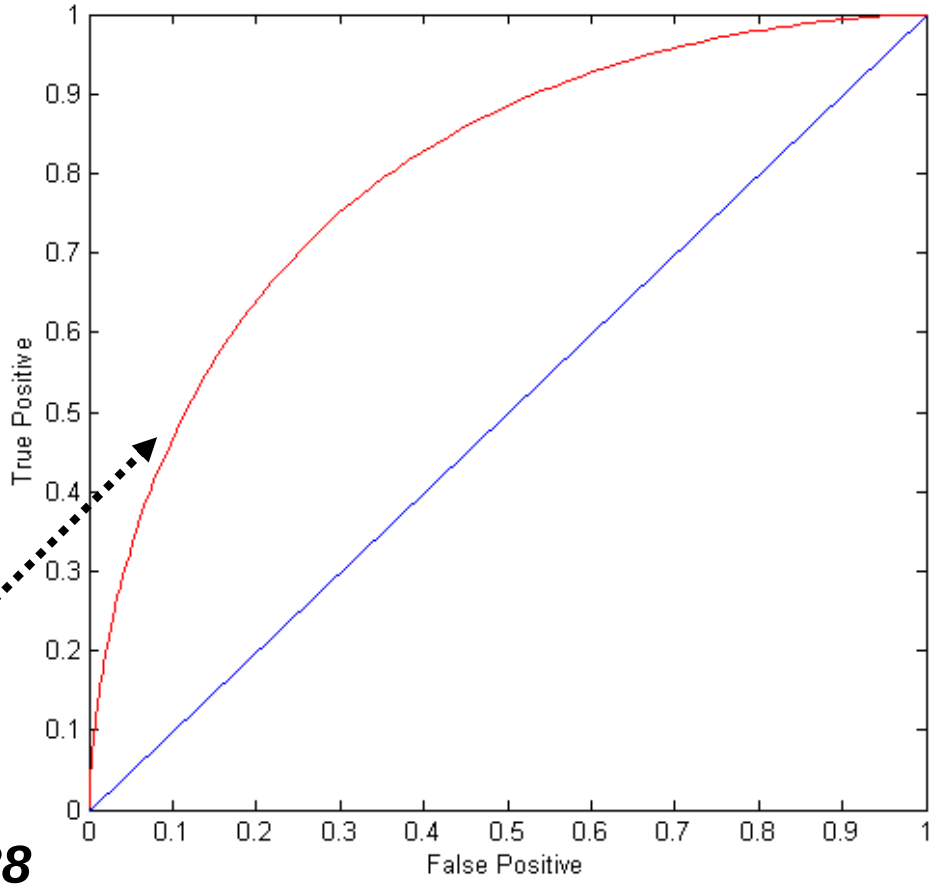
- Vevő oldali működési jellemző
- Az 50-es években fejlesztették ki a jelfeldolgozás számára zajos jelek vizsgálatára.
  - A pozitív találatok és a hamis riasztások közötti kompromisszumot írja le.
- A ROC görbe a IP (y tengely) eseteket ábrázolja a HP (x tengely) függvényében.
- Minden osztályozó hatékonysága reprezentálható egy ponttal a ROC görbén.
  - Az algoritmusbeli küszöbértéket megváltoztatva a mintavételi eloszlás vagy a költségmátrix módosítja a pont helyét.

ROC görbe

Egy dimenziós adatállomány, amely két osztályt tartalmaz (pozitív és negatív).

Minden x > t pontot pozitívnak osztályozunk, a többi negatív lesz.





A *t* küszöb értéknél:

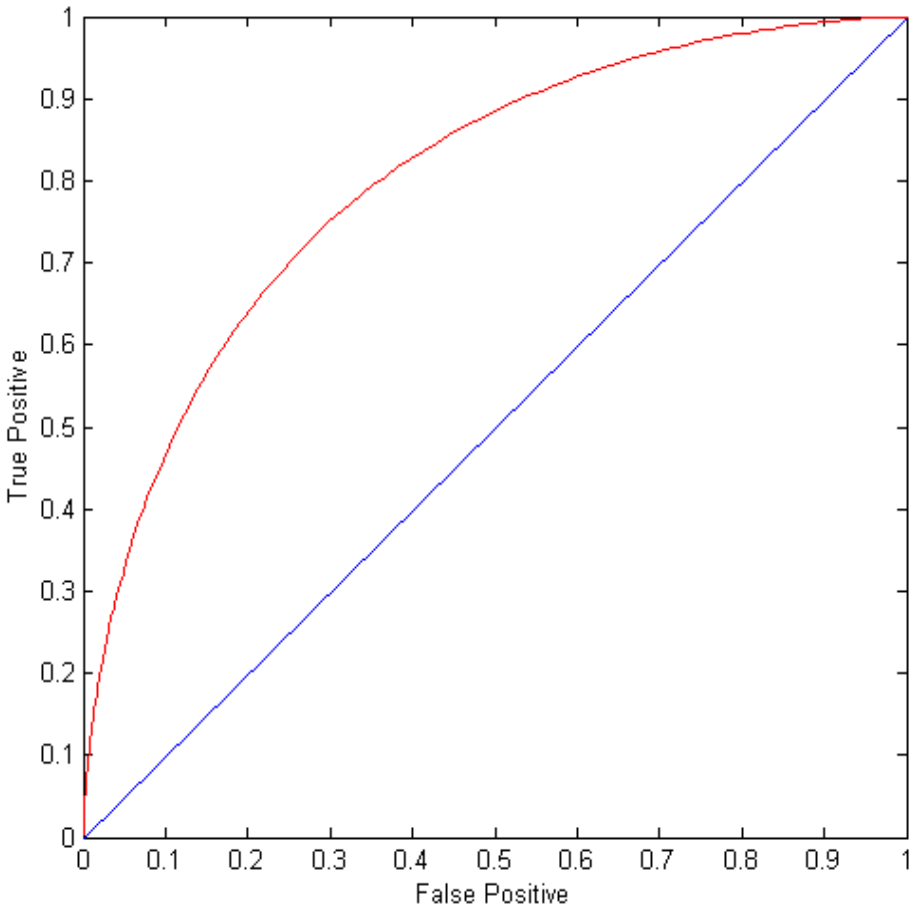
TP=0.5, FN=0.5, FP=0.12, FN=0.88

ROC görbe

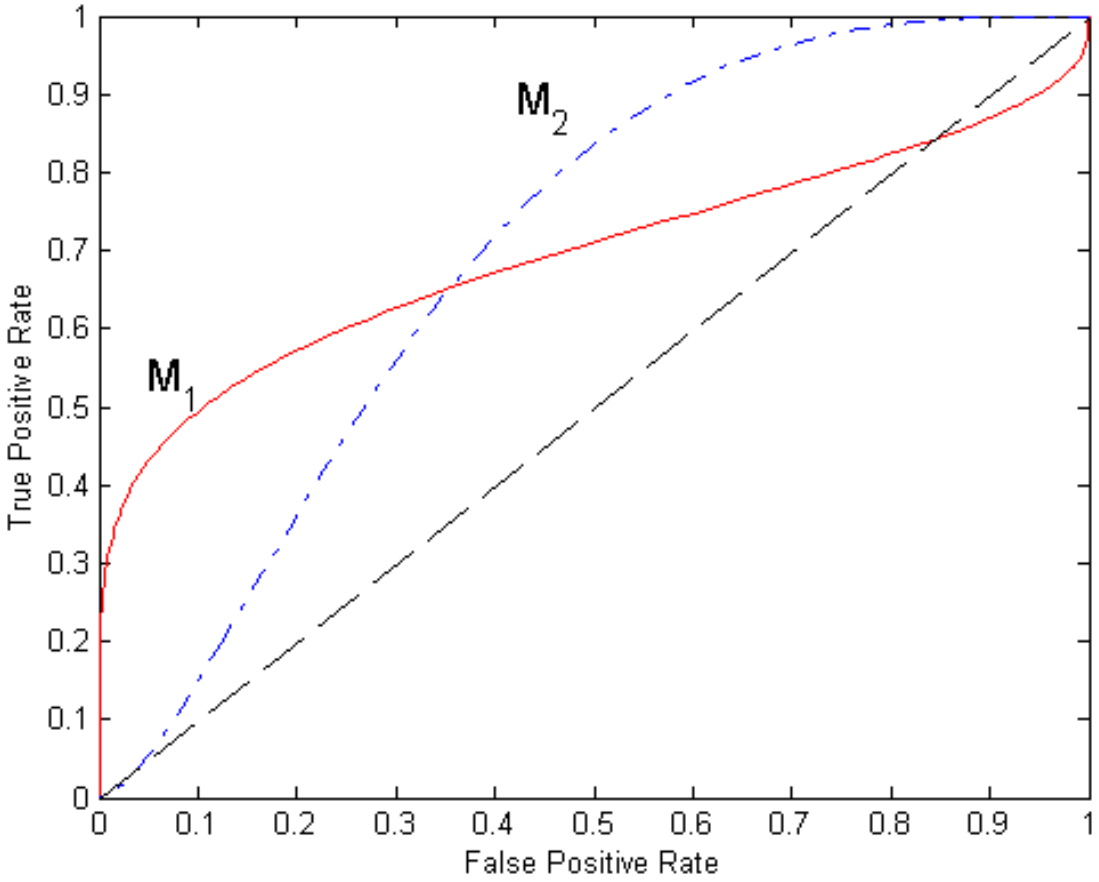
(IP,HP):

- (0,0): mindenki a negatív osztályba kerül
- (1,1): mindenki a pozitív osztályba kerül
- (1,0): ideális

- Diagonális vonal:
  - Véletlen találgatás
  - A diagonális vonal alatt:
    - az előrejelzés a valódi osztály ellentéte



Modellek összehasonlítása ROC görbével



- Általában nincs olyan modell, amely következetesen jobb a többinél:
  - M_1 jobb kis HPR esetén,
  - M_2 jobb nagy HPR esetén.
- A ROC görbe alatti terület:
  - Ideális:
    - Terület = 1
  - Véletlen találgatás:
    - Terület = 0.5

Hogyan szerkesszünk ROC görbét

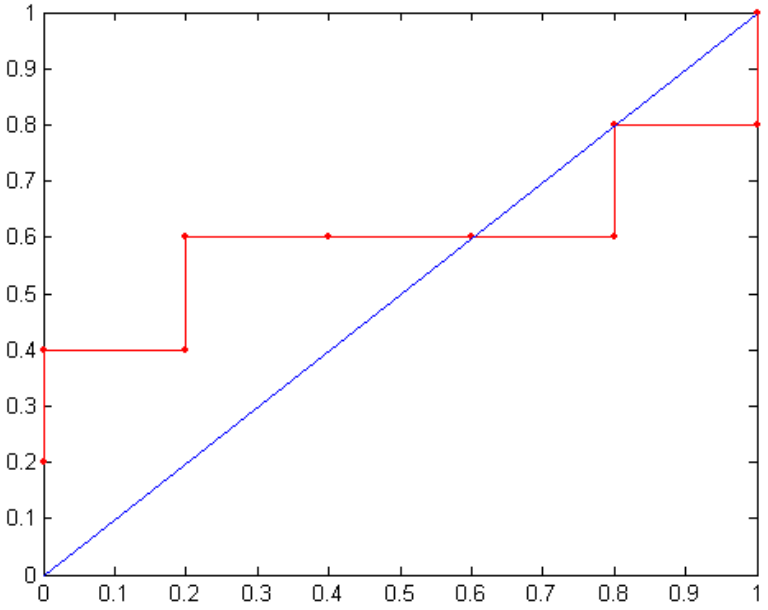
Rekord	P(+|A)	Igaz osztály
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Alkalmazzunk egy olyan osztályozót, amely minden rekordra meghatározza a P(+|A) poszterior valószínűséget.
- Rendezzük a rekordokat P(+|A) szerint csökkenően.
- Válasszuk küszöbnek minden egyes különböző P(+|A) értéket.
- Minden küszöb értéknél számoljuk össze: IP, HP, IN, HN.
- IP ráta, IPR = IP/(IP+HN)
- HP ráta, HPR = HP/(HP + IN)

Hogyan szerkesszünk ROC görbét

	Osztály	+	-	+	-	-	-	+	-	+	+	
Küszöb >=		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
→	IPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→	HPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC görbe:



Szignifikancia vizsgálat

- Adott két modell:
  - M1 modell: pontosság = 85% 30 rekordon tesztelve
  - M2 modell: pontosság = 75% 5000 rekordon tesztelve

- Mondhatjuk azt, hogy M1 jobb mint M2?
  - Mekkora megbízhatóságot tulajdoníthatunk az M1 és M2 modellek pontosságának?
  - A hatékonysági mérőszámokbeli különbség a teszt állományokbeli véletlen ingadozásnak köszönhető vagy szisztematikus az eltérés?

Konfidencia intervallum a pontosságra

- Az előrejelzés Bernoulli kísérletnek tekinthető.
  - A Bernoulli kísérletnek 2 lehetséges kimenetele van.
  - Az előrejelzés lehetséges eredménye: helyes vagy hibás.
  - Független Bernoulli kísérletek összege binomiális eloszlású:
    - $x \sim Bin(N, p)$ x: a helyes előrejelzések száma
    - Pl.: Egy szabályos érmét 50-szer feldobva mennyi fejet kapunk?
      A fejek várt száma = $N \times p = 50 \times 0.5 = 25$

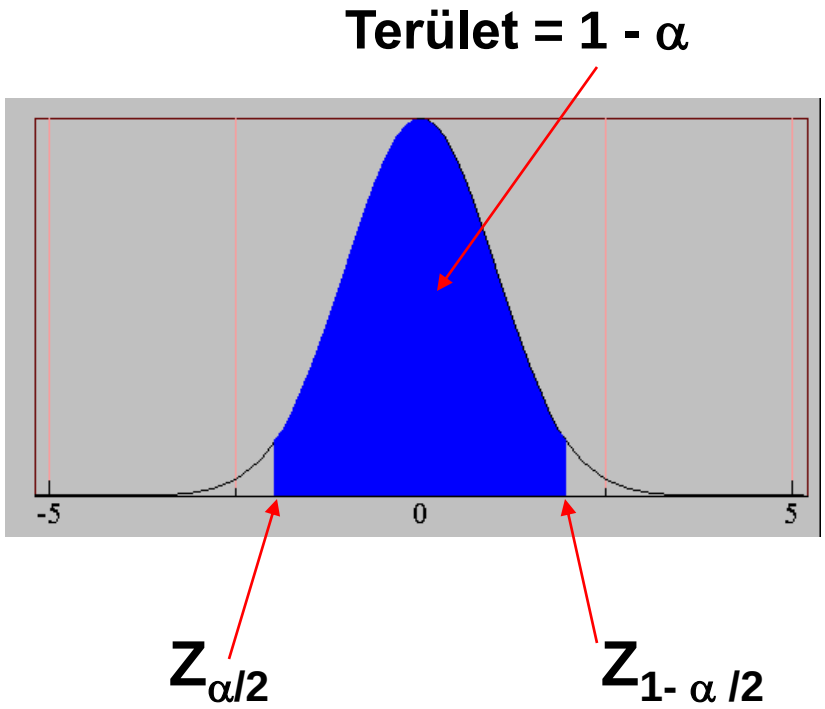
- Adott x (a helyes előrejelzések száma) vagy azok x/N aránya és N (teszt rekordok száma) mellett:

  Tudjuk-e előrejelezni p-t (a modell pontosságát)?

Konfidencia intervallum a pontosságra

- Nagy mintákra (*N > 30*),
  - A helyesek aránya normális eloszlású *p* várható értékkel és *p(1-p)/N* varianciával.

$$P(Z_{\alpha/2} < \frac{x/N - p}{\sqrt{p(1-p)/N}} < Z_{1-\alpha/2}) = 1-\alpha$$



- Konfidencia intervallum *p*-re:

$$p = \frac{2\times x + Z_{\alpha/2}^2 \pm \sqrt{Z_{\alpha/2}^2 + 4\times x - 4\times x^2/N}}{2(N + Z_{\alpha/2}^2)}$$

Konfidencia intervallum a pontosságra

- Tekintsünk egy modellt, amely pontossága 80% volt, amikor 100 teszt rekordon értékeltük ki:
  - N=100, x/N = 0.8
  - legyen 1-α = 0.95 (95% konfidencia)
  - a normális táblázatból $Z_{\alpha/2}$=1.96

1-α	Z
0.99	2.58
0.98	2.33
0.95	1.96
0.90	1.65

N	50	100	500	1000	5000
p(alsó)	0.670	0.711	0.763	0.774	0.789
p(felső)	0.888	0.866	0.833	0.824	0.811

Két modell összehasonlítása

- Két modell, M1 és M2, közül melyik a jobb?
  - M1-t a D1-en (n1 rekord) teszteljük, hiba ráta = e_1
  - M2-t a D2-ön teszteljük (n2 rekord), hiba ráta = e_2
  - Tegyük fel, hogy D1 és D2 függetlenek
  - Ha n1 és n2 elegendően nagy, akkor

$$e_1 \sim N(\mu_1, \sigma_1)$$

$$e_2 \sim N(\mu_2, \sigma_2)$$

  - Közelítőleg: $\hat{\sigma}_i = \frac{e_i(1-e_i)}{n_i}$

Két modell összehasonlítása

- Vizsgáljuk meg, hogy a hiba ráták különbsége szignifikáns-e: d = e1 – e2
  - d ~ $N(d_t,\sigma_t)$ ahol d_t az igazi különbség
  - Mivel D1 és D2 függetlenek a varianciáik összeadódnak:

$$\sigma_t^2 = \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$

$$= \frac{e1(1-e1)}{n1} + \frac{e2(1-e2)}{n2}$$

  - Konfidencia intervallum (1-α) szinten: $d_t = d \pm Z_{\alpha/2}\hat{\sigma}_t$

Szemléltető példa

- Adott: M1: n1 = 30, e1 = 0.15
  M2: n2 = 5000, e2 = 0.25
- d = |e2 – e1| = 0.1 (2-oldali próba)

$$\hat{\sigma}_d = \frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000} = 0.0043$$

- 95% szinten $Z_{\alpha/2}$=1.96

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

=> az intervallum tartalmazza 0 => a különbség nem szignifikáns

2 algoritmus összehasonlítása

- Minden tanuló algoritmus *k* modellt hoz létre:
  - *L1* a *M11, M12, …, M1k* modelleket
  - *L2* a *M21, M22, …, M2k* modelleket
- A modelleket ugyanazon a teszthalmazokon *D1,D2, …, Dk* vizsgáljuk (pl. keresztellenőrzés)
  - Mindegyik halmazra: számoljuk ki $d_j = e_{1j} – e_{2j}$
  - d_j várható értéke d_t varianciája σ_t
  - Becsüljük:

$$\hat{\sigma}_t^2 = \frac{\sum_{j=1}^{k}(d_j - \bar{d})^2}{k(k-1)}$$

$$d_t = d \pm t_{1-\alpha,k-1}\hat{\sigma}_t$$