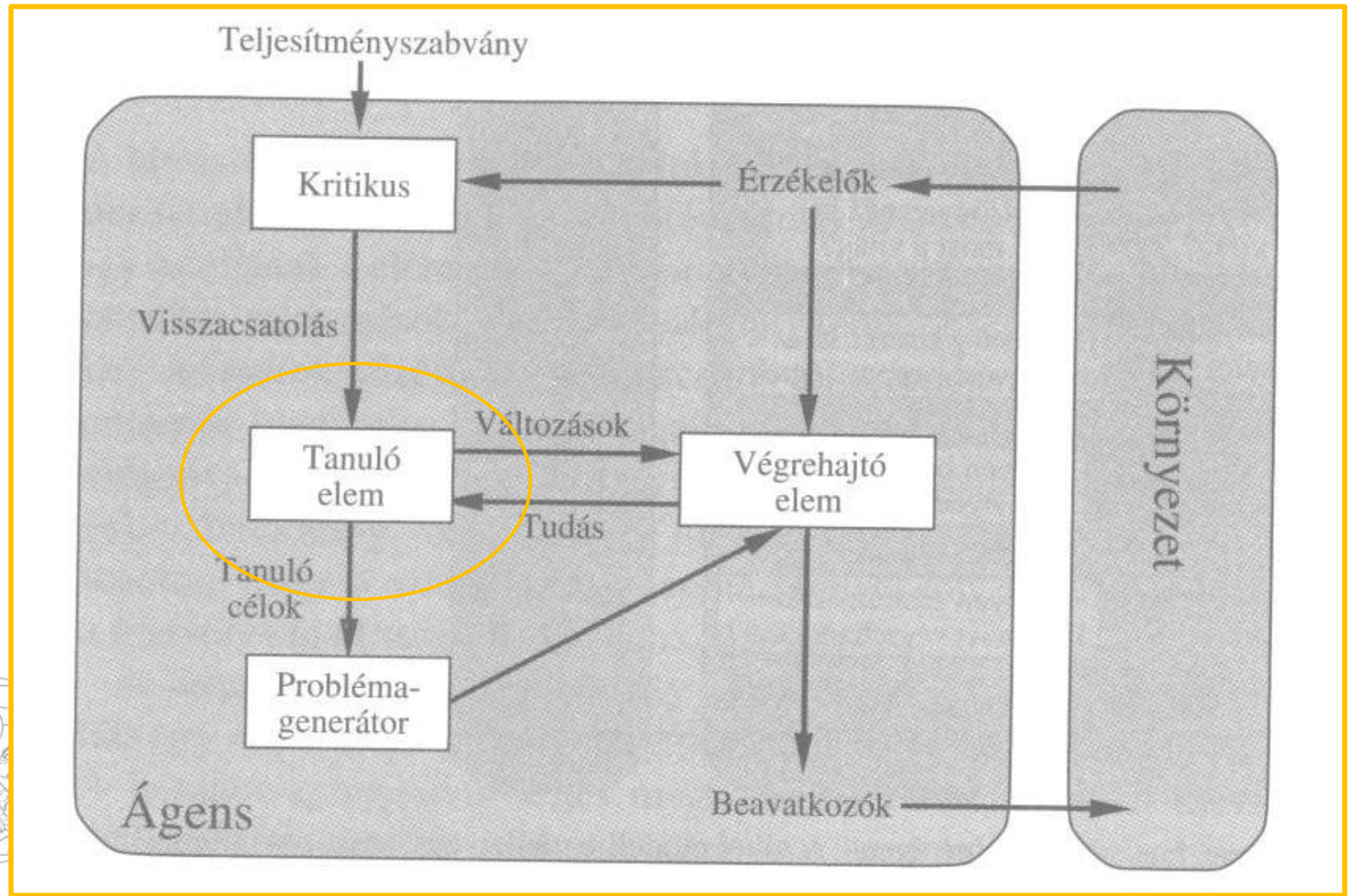




Tanulás

Tanuló ágensek





Bevezetés

- **Tanulás: tapasztalati tények felhasználása arra, hogy egy racionális ágens teljesítményét növeljük**
- **Visszacsatolás: ez alapján tanul az ágens**
- **Reprezentáció: hogyan definiáljuk a problémát?**

Felügyelt tanulás

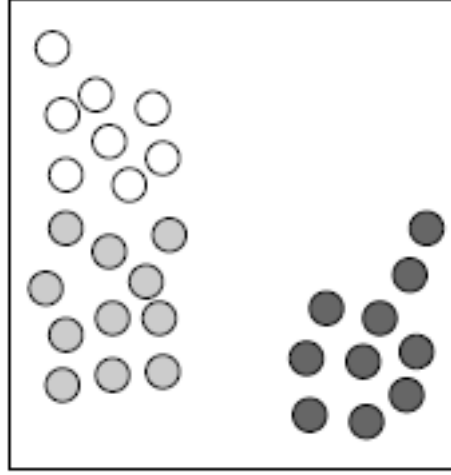
- $f : A \rightarrow B$ függvényt kell megtalálni
- példák alapján, amelyek
 $(a_1; f(a_1)); \dots; (a_n; f(a_n))$ alakban
adottak
- pl. A: emailek halmaza, $B = \{\text{spam}, \neg\text{spam}\}$
- a példák kézzel osztályozott emailek



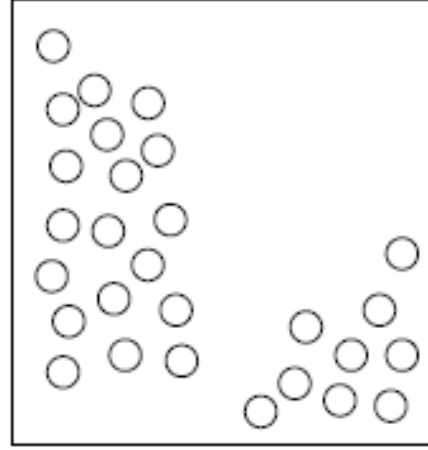
Felügyelet nélküli tanulás

- a példák csak $a_1; \dots; a_n$ alakban adóttak
- nem függvényt kell keresni, hanem mintázatokot (klasztereket vagy eloszlást)





felügyelt tanulás



felügyelet nélküli tanulás



Megerősítéses tanulás

- tanuló példák nem adottak direkt módon
- a feladat: egy állapottérben az ágens megtanulja az optimális stratégiát úgy, hogy csak a célállapotokban értelmezett jutalomfüggvény maximalizálását érje el
- Pl: taxisofőr a borravaló nagyságából tanul

Reprezentáció

- Mik legyenek az A és B halmazok, ill. az $A \rightarrow B$ függvény alakja?
- Lehet folytonos vagy konceptuális, valószínűségi vagy **logikai** reprezentáció...
- Felügyelt esetben a függvényt, vagy felügyelet nélküli esetben a mintát mindig adott alakban keressük, pl. adott fokszámú polinom, Boole-függvények adott osztálya stb.
- **polinom** (vagy *többtagú algebrai kifejezés*) egy olyan kifejezés, melyben csak számok és változók nemnegatív egész kitevőjű hatványainak szorzatai, illetve ilyenek összegei szerepelnek





Felügyelt (induktív) tanulás

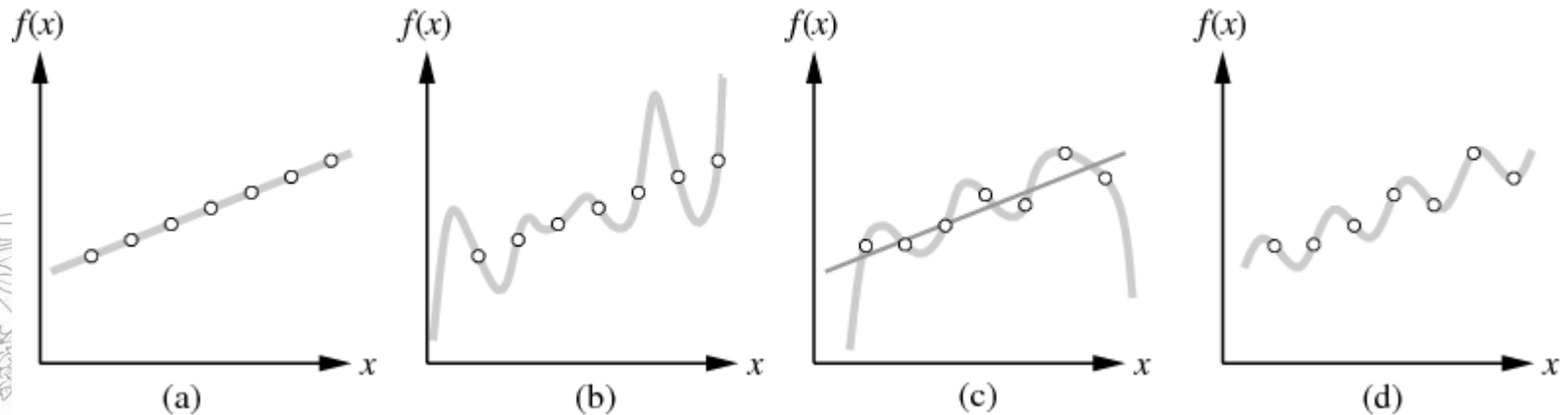
- Adott egy $f : A \rightarrow B$ függvény és $x_1; f(x_1); \dots; x_n; f(x_n)$ tanuló példák
- Keressünk egy $h : A \rightarrow B$ függvényt, amely f -et közelíti
- A h függvény konzisztens az adatokkal, ha $h(x_i) = f(x_i)$ minden példára.
- A h függvény mindig egy H hipotézistér eleme
- A tanulás realizálható, ha létezik $h \in H$, amelyre h konzisztens
- A gyakorlatban elég, ha h „közel” van a példákhoz, nem kell pontosan illeszkednie feltétlenül

Indukció

- f jó közelítése olyan amely a példákon kívül is jól közelíti f -et, azaz jól általánosít
- Magolás: az a h , amelyre $h(x) = f(x)$ minden példára, egyébként $h(x) = 0$, tökéletesen megtanulja a példákat, de rosszul általánosít
- Ockham borotvája: ha más alapján nem tudunk választani, akkor a lehető legtömörebb leírást kell venni



- (a): elsőfokú polinom (konzisztens)
- (b): nagyfokú polinom (konzisztens)
- (c) ábra: hatodfokú polinom (konzisztens) – egyenes (nem konzisztens, de általánosabb)
- (d): szinuszos illesztés





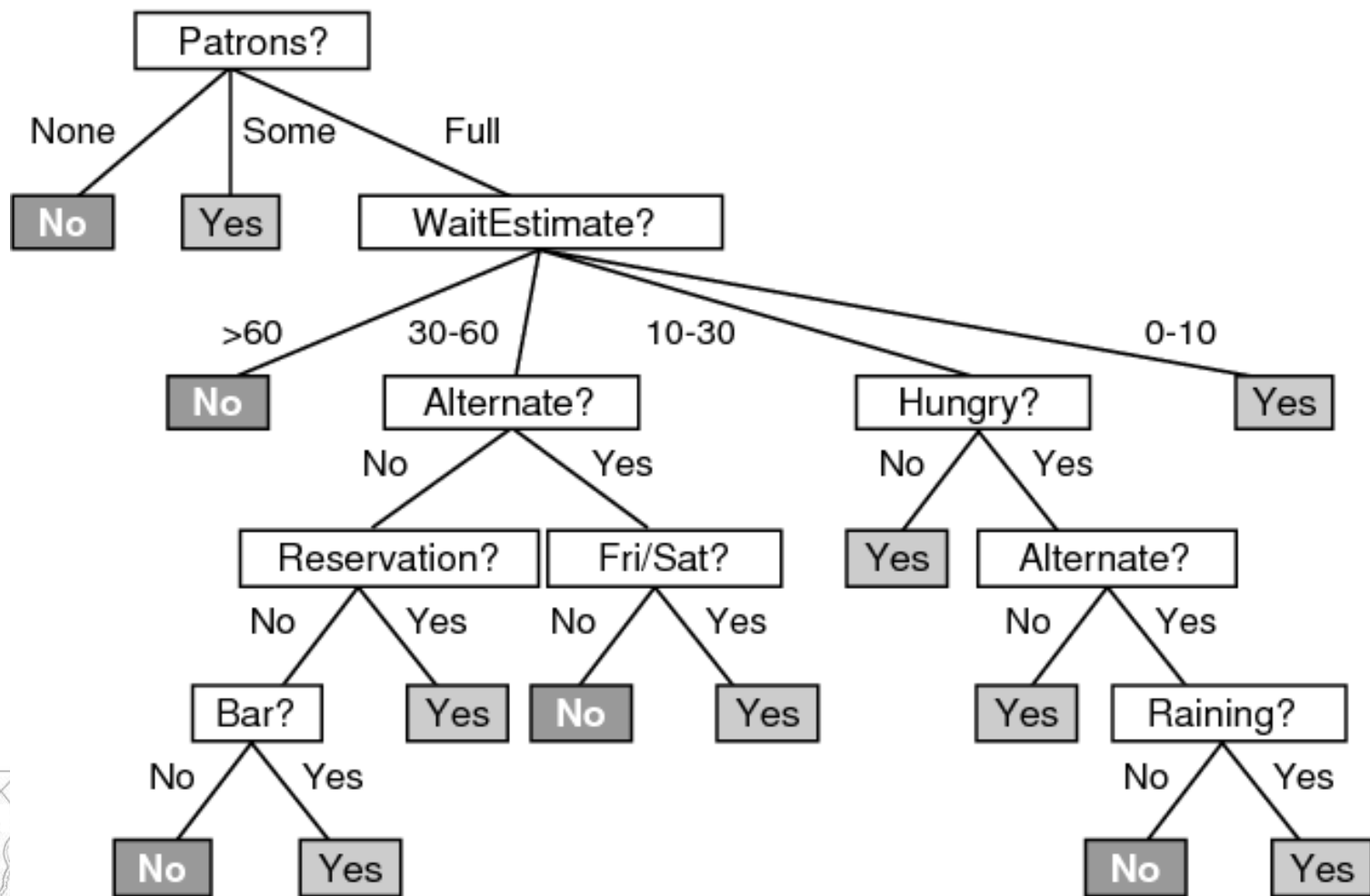
Döntési fák

- Induktív tanulás egy példája
- **Bemenet:** jellemzőkkel leírt objektumok
- **Kimenet:** döntés (jóslt érték a bemenetre)
- **Osztályozás:** diszkrét értékkészletű függvény
- **Regresszió:** folytonos értékkészletű függvény
- **Bináris osztályozás:** igen/nem (pozitív/negatív) döntések

Példa

- **Étterembe akarunk menni**
- **Várjunk-e asztalra?**
- **Milyen jellemzőkkel dolgozunk?**
(diszkrét változók, eldöntendő kérdések)
- **Ember számára is jól értelmezhető döntések (neurális hálókra nem igaz)**





Kifejező erő

- A fából logikai formula gyártható és viszont
- Ítéletkalkulus
- $Fa \equiv (\text{Vendégek}=\text{néhány})$
 $\vee (\text{Vendégek}=\text{tele} \wedge \text{Várakozás}=0-10)$
 $\vee \dots$
- Igen nagy fák keletkeznek...





Döntési fa építése

- **Tanító halmaz: pozitív és negatív példák sorozata, adott jellemzőkkel**
- **Általánosítani kell a tanító halmaz példái alapján**
- **Ötlet: válasszuk először azt a jellemzőt, ami a leghatékonyabban választja szét a példákat**
- **Ezt az eljárást ismételjük rekurzívan a még fel nem használt jellemzőkre**



- ha csak pozitív vagy negatív példa van, akkor levélhez értünk, megcímkezzük megfelelő módon
- ha üres halmazról van szó (ez akkor lehet, ha többértékű változó a szülő): alapértelmezett érték
- ha pozitív és negatív példa is van, de nincs több változó (zajos példák): ekkor a többségi szavazattal címkézhetjük a levelet



Információelméleti megközelítés

- Adott változó ismerete megnöveli az információt a példa kategóriájáról
- Információtartalom: p valószínűségű esemény információtartalma – $\log p$ ($\log_2$ esetén mértékegysége a bit)

$$I(P(v_1), \dots, P(v_n)) = \sum_{i=1}^n -P(v_i) \log_2 P(v_i)$$

A pénzfeldobás esetre ellenőrizve az összefüggést, a következőt kapjuk:

$$I\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = 1 \text{ bit}$$

- Egy döntési fa egy pontjában az összes konzisztens példában p pozitív és n negatív példa

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

- Mennyi infó kell még ezután?
- Információnyereség: eredeti (adott jellemző felhasználása előtti) és a teszt utáni információszükséglet különbsége
- A maximális nyereségű jellemzőt válasszuk ki

$$\text{MaradékInfoSzükséglet}(A) = \sum_{i=1}^v \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

$$\text{InfoNyereség}(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \text{MaradékInfoSzükséglet}(A)$$



Túlillesztés

- Túl jól írjuk le az adatokat
- Túl általános a modell (mindenre passzol)
- Irreleváns jellemzőket használunk (milyen színű kockával dobunk...), de találunk rá szabályosságot
- Megnézzük, hogy az információnyereség statisztikailag szignifikáns-e





A tanuló algoritmusok kiértékelése

- **Tanuló halmaz:** ezen hozzuk létre a modellt (pl. döntési fa)
- **Teszthalmaz:** ezen értékeljük ki
- **Kiértékelési metrika:** sokféle lehet
- **Döntési fáknál** gyakran a jól osztályozott példák százalékos aránya

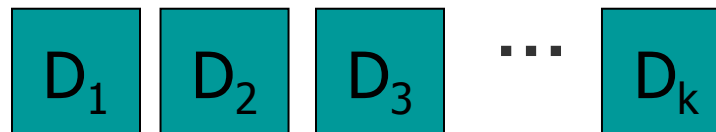


Kiértékelési módszerek

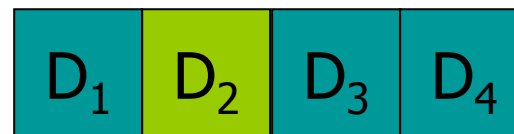
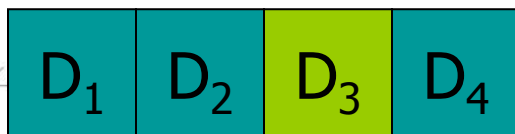
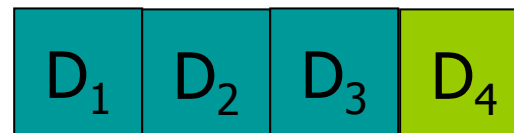
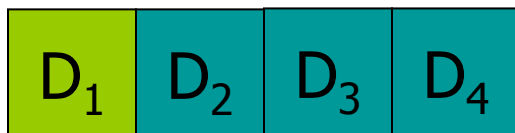
- **Tanító halmaz vs. teszthalmaz**
- **Keresztvalidáció**
- **„egy kimarad” („leave one out”)**
- **Baseline: alapmegoldás - pl. adjuk a többségi címkét minden egyednek**

Keresztvalidáció

- A k -szoros keresztvalidálásnál a D halmazt k kölcsönösen diszjunkttra bontjuk: $D_1, D_2, \dots, D_k$



A tanuló algoritmust k alkalommal tanítjuk és teszteljük, tanítjuk $D \setminus D_i$ -n, teszteljük D_i -n



Leave one out: D_i mindig egy egység (1 szituáció, 1 objektum...)



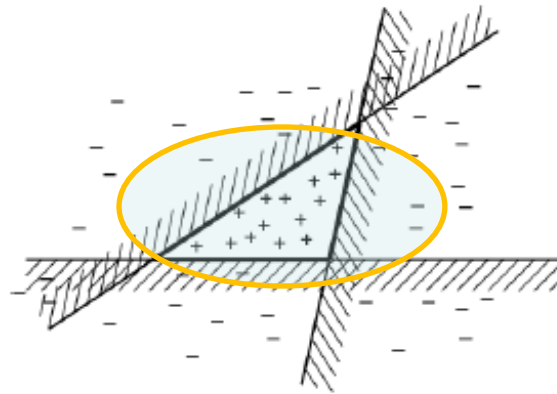


Turbózás (boosting)

- Érdemes lehet több modellt tanítani
- A különböző modellek hibái egymástól függetlenek
- Kevésbé valószínű, hogy több modell is ugyanott hibázik
- Egyszerű modellek kombinációja megfelelhet komplexebb modelleknek

Súlyozás

- Minden példa és az egyes modellek is kapnak súlyt
- A nehéz példák és a jó hipotézisek nagyobb súlyt kapnak
- A súlyok összegződhetnek



Gyakorlati példa

- **Van-e a betegnek enyhe kognitív zavara?**
- **Bemenet: a beteg beszédátirata, annak morfológiailag és szintaktikailag elemzett változata**
- **Kimenet: igen/nem válasz (beteg/egészséges döntés)**
- **Jellemzők: morfológiai, lexikai, szemantikai...**
(fejlesztés alatt...)



Jellemzőreprésentáció

187,4,5,1,3,3,2.673797,0.534759,1.604278,1.604278,24,0,0,5,0,0,0,0,positive

107,2,5,4,0,4,4.672897,3.738318,0,3.738318,11,4,10,2,0,0,0,0,positive

191,12,12,7,1,8,6.282723,3.664921,0.52356,4.188482,30,9,8,1,0,0,0,0,negative

76,8,3,0,0,1,3.947368,0,0,1.315789,13,0,0,0,0,0,0,0,negative

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,OTHER,OTHER,ATT,none,none,other,R,V,none,none,,el,temet,none,none,"","","--",negative

0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,ATT,none,none,ATT,MODE,A,none,none,N,other,egyes,none,none,forrás,szerint,gy,"frr,-gy-frr,weasel

0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,ATT,ATT,none,MODE,ATT,N,A,none,other,P,forrás,egyes,none,szerint,maga,frr,gy,szr,gy-frr-szr,negative

0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,MODE,ATT,ATT,ATT,OTHER,other,N,A,P,N,szerint,forrás,egyes,maga,attila,szr,frr,mg,frr-szr-mg,doxastic

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,ATT,MODE,ATT,OTHER,SUBJ,P,other,N,N,N,maga,szerint,forrás,attila,vezér,mg,szr,tt,szr-mg-tt,negative

hesitRate <= 1.223242: negative (9.0)

hesitRate > 1.223242

| sentenceNum <= 4: positive (10.0)

| sentenceNum > 4

| | hesitRate <= 14.814815

| | | uncertainNum <= 10

| | | | topicWord1 <= 4

| | | | | topicWord1 <= 3

| | | | | hesitNum <= 21

| | | | | | hesitNum <= 13: negative (53.0/24.0)

| | | | | | hesitNum > 13: positive (12.0/2.0)

| | | | | | hesitNum > 21: negative (4.0)

| | | | | topicWord1 > 3: positive (6.0)

| | | | | topicWord1 > 4

| | | | | memoryNum <= 0

| | | | | negationWord <= 1

| | | | | | uncertainRate <= 2.409639: negative (3.0)

| | | | | | uncertainRate > 2.409639: positive (3.0)

| | | | | | negationWord > 1: negative (2.0)

| | | | | | memoryNum > 0: negative (5.0)

| | | | | uncertainNum > 10

| | | | | memoryRate <= 0.588235: positive (12.0)

| | | | | memoryRate > 0.588235

| | | | | memoryRate <= 1.169591: negative (3.0)

| | | | | memoryRate > 1.169591: positive (4.0)

| | hesitRate > 14.814815: positive (12.0)



Eredmények

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 96
69.5652 %

Incorrectly Classified Instances 42
30.4348 %

=== Confusion Matrix ===

a b <-- classified as
61 20 | a = positive
22 35 | b = negative

