

# TEHNICI DE EXPLORARE A DATELOR -DATA MINING -

CURS 5

# Algoritmi fundamentali de data mining

- Constructia arborilor de decizie
- Constructia regulilor
- Explorarea regulilor de asociere
- Algoritmi pentru clustering

# Constructia arborilor de decizie

- Arborii de decizie sunt utilizați pentru clasificarea exemplilor necunoscute, prin testarea valorilor atributelor exemplilor prin arborele de decizie
- Procesul care creează arborele de decizie - *inducție*
  - cere un număr mic de treceri prin setul de antrenare.
- majoritatea metodelor de generare a arborilor de decizie trec prin două faze:
  - *faza de construcție (creștere) a arborelui*
  - *faza de tăiere (pruning)*

# Constructia arborilor de decizie

- **Faza de construcție a arborelui**
  - un proces iterativ care implică divizarea progresivă a datelor în subseturi mai mici.
  - prima iterație consideră că nodul rădăcină conține toate datele.
  - următoarele iterații lucrează pe noduri derivate care vor conține subseturi de date.
    - la fiecare divizare, variabilele sunt analizate și este aleasă cea mai bună divizare.
      - caracteristică importantă a divizării este aceea că nu se face o verificare dinainte în arbore să se vadă dacă o altă decizie ar produce un rezultat final mai bun.
- **Faza de tăiere**
  - identifică și înlătură ramurile care reflectă zgomote sau excepții



# Construcția arborilor de decizie

- Metodele de construcție a arborelui folosesc câteva reguli de oprire.
  - sunt în general bazate pe factori incluzând:
    - adâncimea maximă a arborelui,
    - numărul minim de elemente dintr-un nod care este considerat pentru divizare,
    - sau numărul minim de elemente care trebuie să fie într-un nou nod.
  - În cele mai multe implementări utilizatorul poate modifica parametri asociați cu aceste reguli.
  - Sunt algoritmi care constau în construcția arborilor la adâncimea lor maximă.
    - astfel de arbori pot preciza exact toate exemplele din setul de test (exceptând înregistrările conflictuale),
    - problema este aceea, că este foarte probabil să apară o supraestimare a datelor.
- Metoda de inducție a arborelui de decizie cuprinde o *procedură de bază și condițiile de oprire a partiționării*

Parameters

Decision Tree

criterion: information\_gain

minimal size for split: 4

minimal leaf size: 2

minimal gain: 0.0010

maximal depth: 20

confidence: 0.05

number of prepruning...: 2

☐ no pre pruning

☐ no pruning

Compatibility level: 5.0.10

# Constructia arborilor de decizie

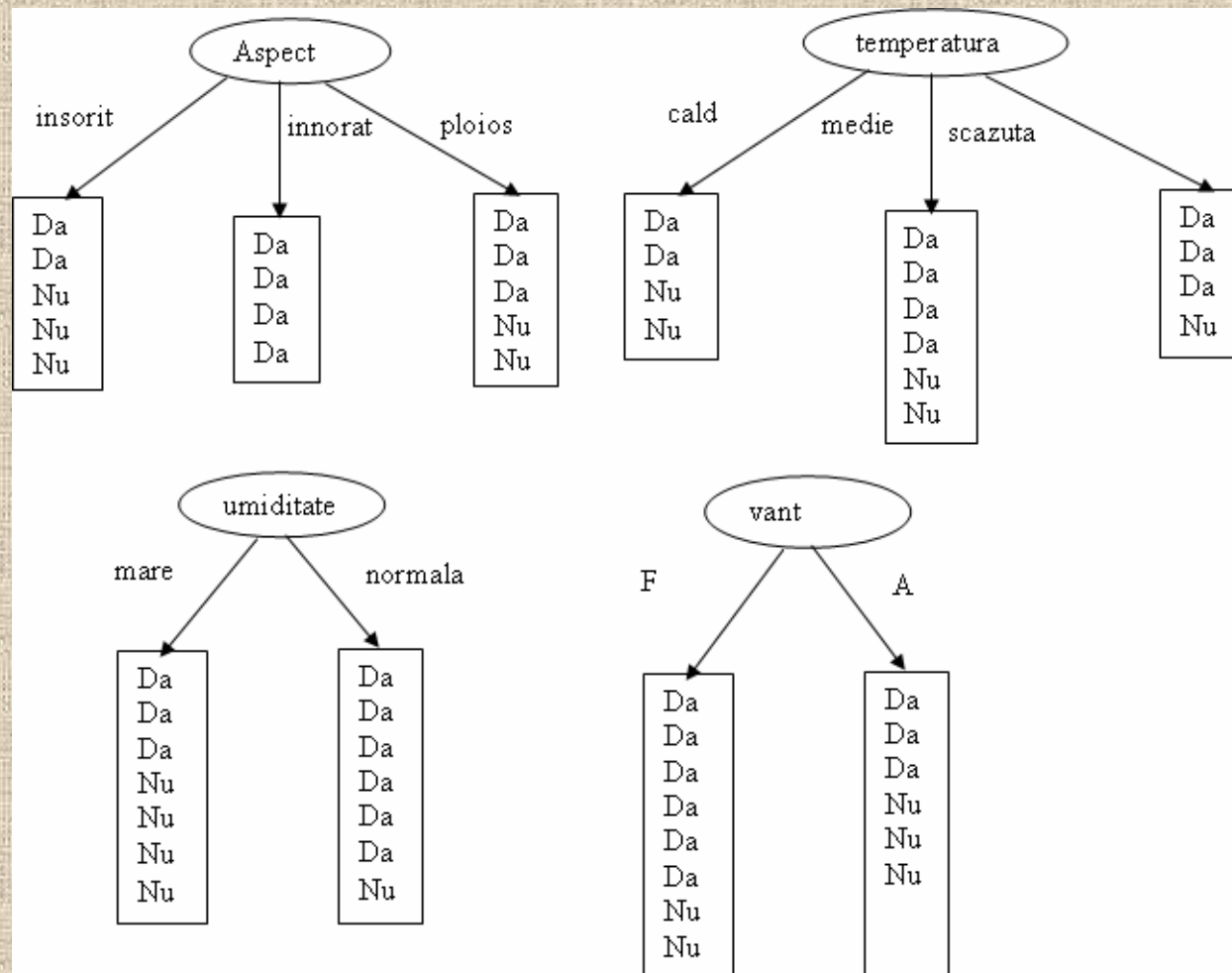
- Procedura de bază construiește arborele într-o manieră recursivă de sus în jos:
  - la început, toate exemplele de învățare sunt în nodul rădăcină;
  - attributele sunt de tip *enumerare (descriptive)*. Dacă valorile sunt continue, atunci mai întâi ele sunt *discretizate*.
  - exemplele sunt partiționate recursiv bazat pe attributele selectate;
  - attributele de test sunt selectate pe baza unei măsuri euristice sau statistice (de exemplu câștigul de informație).
- Oprirea partiționării se face în următoarele condiții:
  - toate exemplele pentru un anumit nod aparțin unei aceleași clase;
  - nu mai există nici un atribut pentru partiționări ulterioare;
  - nu rămâne nici un exemplu.
- După ce arborele este complet, se poate explora modelul pentru a găsi nodurile de ieșire sau subarborii care nu sunt utile, sau regulile care sunt apreciate ca fiind neadecvate.
- Tăierea este o tehnică obișnuită utilizată pentru a face un arbore mai general.
- Algoritmii care construiesc arborii la adâncimea maximă vor invoca automat tăierea.
- PROBLEMA – cum se decide care este atributul ale cărui valori se vor testa într-un nod pentru divizarea arborelui

# Exemplu

Aspect	Temperatura	Umiditate	Vant	Desfasurare joc
insonit	cald	mare	F	Nu
insonit	cald	mare	A	Nu
innorat	cald	mare	F	Da
ploios	medie	mare	F	Da
ploios	scazuta	normala	F	Da
ploios	scazuta	normala	A	Nu
innorat	scazuta	normala	A	Da
insonit	medie	mare	F	Nu
insonit	scazuta	normala	F	Da
ploios	medie	normala	F	Da
insonit	medie	normala	T	Da
innorat	medie	mare	T	Da
innorat	cald	normala	F	Da
ploios	medie	mare	T	Nu

Pornind de la datele de mai sus=> exista 4 posibilitati de alegere a atributului de divizare





- In frunze sunt indicate clasela (da/nu)
- Orice frunza care contine numai valori *Da* respectiv *Nu* nu va mai fi divizata ulterior
- Daca exista o masura a puritatii fiecarui nod, ar trebui alese pentru divizare in fiecare nod acel atribut care produce cele mai pure noduri copil

masura puritatii => informatia – masurata in biti



# Calculul informatiei

- Se presupune ca avem un nod al unui arbore in care exista un anumit numar de valori din fiecare clasa (presupunem initial ca sunt posibile doar doua clase, ca in exemplul precedent)
  - Proprietatile informatiei:
    - Cand numarul oricarei valori considerate este zero atunci informatia este zero
    - Cand numarul valorilor claselor este egal, informatia are nivelul maxim
- Aceste proprietati pot fi extinse si in cazul in care avem de-a face cu mai mult de doua clase

# Calculul informatiei

- Se porneste de la ideea există două clase,  $P$  și  $N$ , și un set de exemple  $S$  conținând  $p$  elemente din clasa  $P$  și  $n$  elemente din clasa  $N$ ;
  - Cantitatea de informație necesară pentru a decide dacă un exemplu arbitrar din  $S$  aparține clasei  $P$  sau clasei  $N$  este definită astfel

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

- Se presupune ca utilizând atributul  $A$  un set  $S$  va fi partiționat în seturi  $\{S_1, S_2, \dots, S_v\}$ 
  - Dacă  $S_i$  conține  $p_i$  exemple din  $P$  și  $n_i$  exemple din  $N$ , entropia sau informația necesară clasificării tuturor obiectelor din toți subarborii  $S_i$  este:

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

- Informația care ar fi câștigată pe ramura  $A$  este data de:

$$Gain(A) = I(p, n) - E(A)$$

# Exemplu de calcul

- In exemplul de mai inainte se alege calculul castigului de informatie daca se considera drept atribut de test *umiditatea*
  - Atributul tinta are 2 valori => exista doua clase:
    - *Da* cu 9/14 cazuri
    - *Nu* cu 5/14 cazuri
    - Informatia pentru acest nod se va calcula astfel:

$$I(9,5) = -9/14 * \log_2(9/14) - 5/14 * \log_2(5/14) = 0,940$$

- Deoarece atributul are doua valori distincte, setul initial de date se va imparti in doua subseturi
  - Entropia pentru atributul *umiditate* este:
  - $E(\text{umiditate}) = (3+4)/14 * I(3,4) + (6+1)/14 * I(6,1) = 0.7885$



# Exemplu de calcul

- In exemplul de mai inainte se alege calculul castigului de informatie daca se considera drept atribut de test *aspectul*
  - Atributul tinta are 2 valori => exista doua clase:
    - *Da* cu 9/14 cazuri
    - *Nu* cu 5/14 cazuri
      - Informatia pentru acest nod se va calcula astfel:
$$I(9,5) = -9/14 * \log_2(9/14) - 5/14 * \log_2(5/14) = 0,940 \text{ biti}$$
  - Deoarece atributul are trei valori distincte, setul initial de date se va imparti in trei subseturi
    - Entropia pentru atributul *aspect* este:
    - $E(\text{aspect}) = (2+3)/14 * I(2,3) + (4)/14 * I(4,0) + (3+2)/14 * I(3,2) = 0.693 \text{ biti}$



# Exemplu de calcul

- Similar se calculeaza entropia pentru temperatura care are tot trei valori distincte, deci setul initial de date se va imparti in trei subseturi
  - Entropia pentru atributul *temperatura* este:
  - $E(\text{temperatura}) = (2+2)/14 * I(2,2) + (4+2)/14 * I(4,2) + (3+1)/14 * I(3,1) = 0.912$  biti
- In urmatoarea etapa se calculeaza castigul de informatie pentru fiecare atribut in parte, astfel:

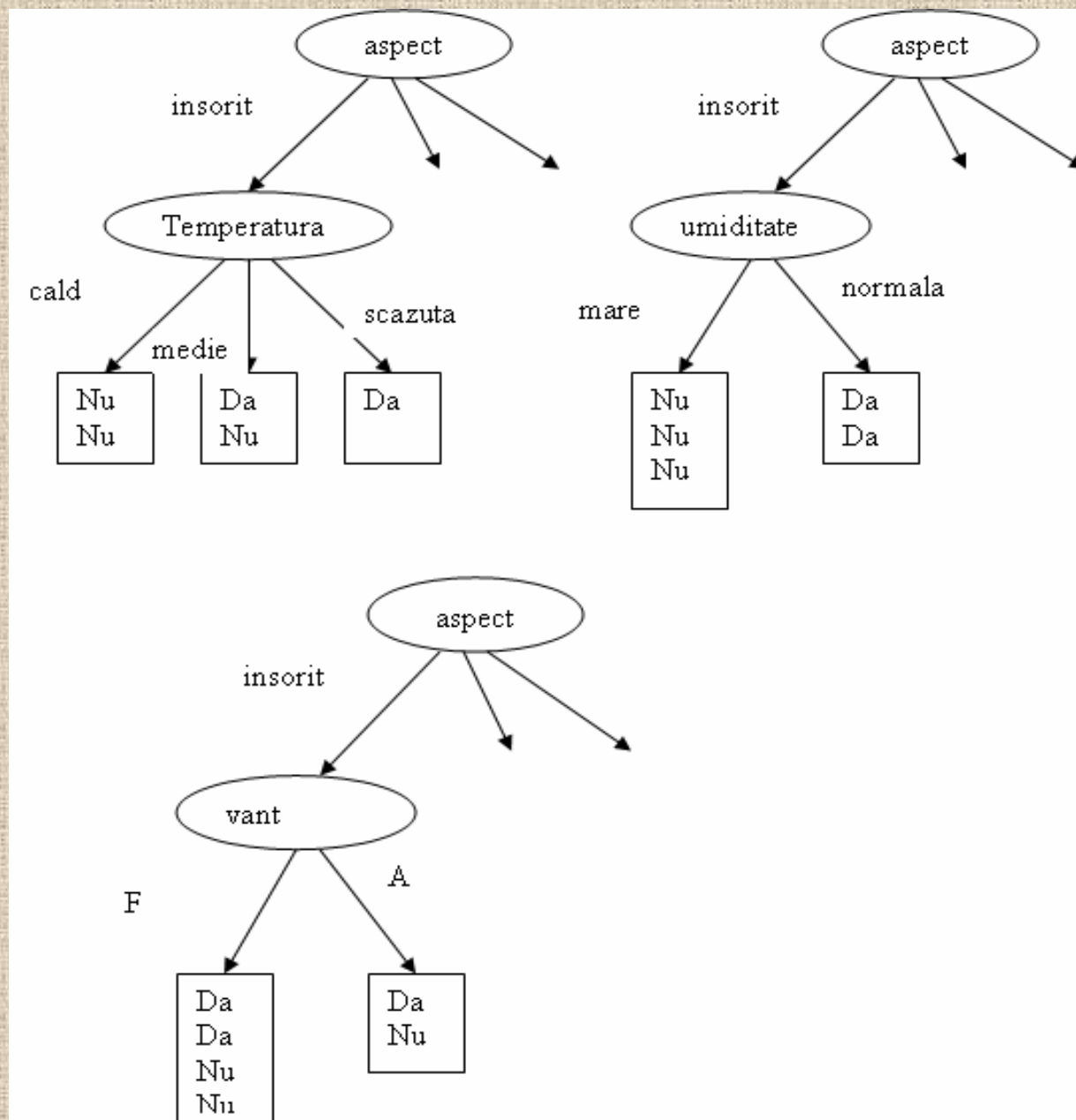
$$\text{Gain}(\text{umiditate}) = I(9,5) - E(\text{umiditate}) = 0,940 - 0,788 = 0,152 \text{ biti}$$

$$\text{Gain}(\text{aspect}) = I(9,5) - E(\text{aspect}) = 0,940 - 0,693 = 0,247 \text{ biti}$$

$$\text{Gain}(\text{temperatura}) = I(9,5) - E(\text{temperatura}) = 0,940 - 0,912 = 0,029 \text{ biti}$$

$$\text{Gain}(\text{vant}) = I(9,5) - E(\text{vant}) = 0,940 - 0,832 = 0,108 \text{ biti}$$

- Primul atribut dupa care se face diviziunea in nod este “*aspect*” deoarece produce cel mai mare castig de informatie
  - Un alt motiv care indica acest atribut ca un candidat la pozitia de radacina a arborelui este acela ca produce un nod frunza pur ( contine numai valori *Da*) si acesta nu mai trebuie divizat in continuare
- Se considera celelalte doua noduri pentru care procesul se reia recursiv
  - Deoarece atributul *aspect* nu mai poate produce informatii noi se vor lua in calcul celelalte trei attribute





- Se calculeaza informatia in nod si entropiile aferente fiecarui atribut posibil de folosit pentru divizare
  - In nod sunt 5 exemple impartite in doua clase:
    - Doua exemple clasificate cu *Da*
    - Trei exemple clasificate cu *Nu*
- Informatia in nod este:
 
$$I(2,3) = -2/5 * \log_2 2/5 - 3/5 * \log_2 3/5 = 0,971 \text{ biti}$$

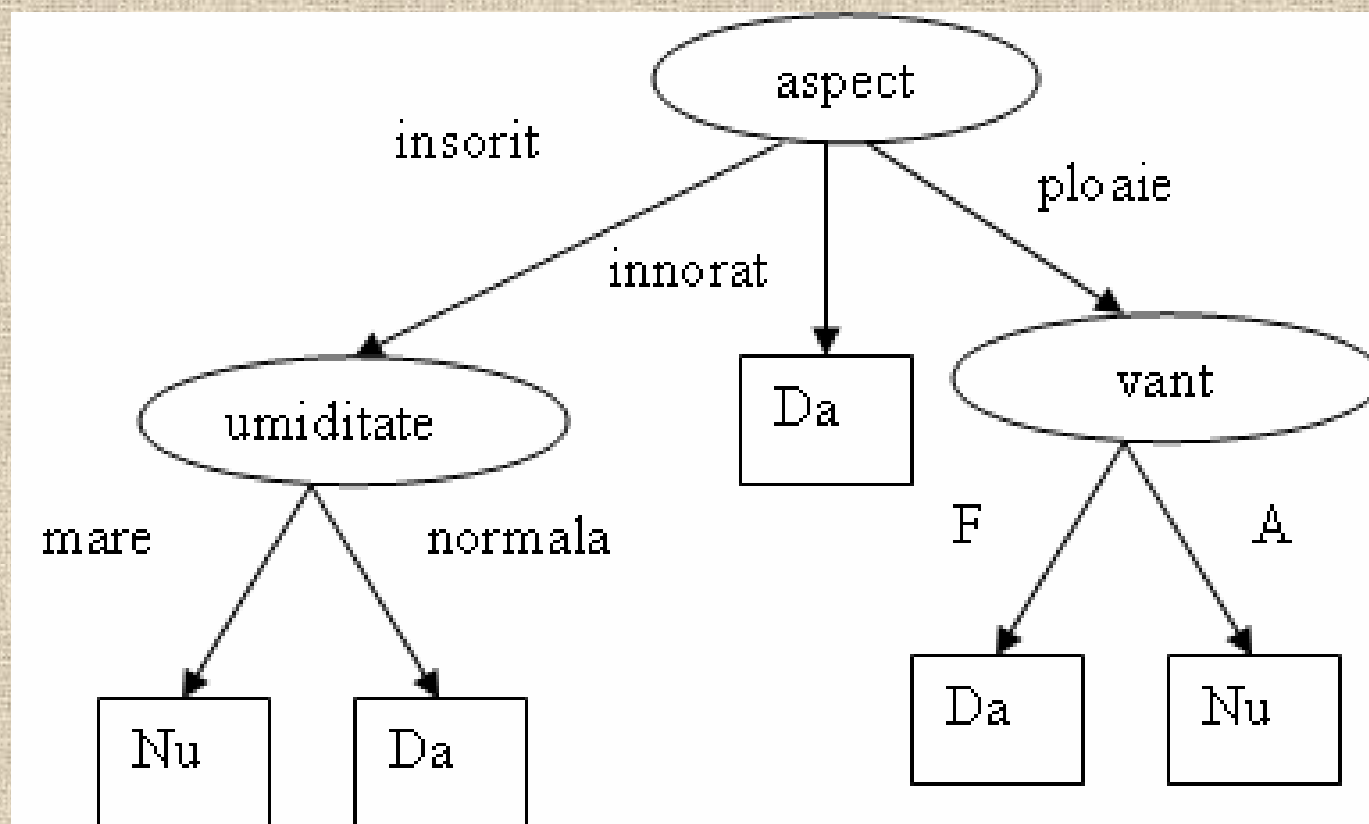
$$E(\text{umiditate}) = 3/5 * I(3,0) + 2/5 * I(2,0) = 0$$

$$E(\text{temperatura}) = 2/5 * I(2,0) + 2/5 * I(1,1) + 1/5 * I(1,0) = 0,4 \text{ biti}$$
  

$$\text{Gain}(\text{umiditate}) = I(2,3) - E(\text{umiditate}) = 0,971 - 0 = 0,971 \text{ biti}$$

$$\text{Gain}(\text{temperatura}) = I(2,3) - E(\text{temperatura}) = 0,971 - 0,4 = 0,571 \text{ biti}$$
  - => se va considera drept atribut de test – *umiditatea*
- pe ramura *ploaie* se va urma acelasi rationament
  - => atributul de test va fi considerat atributul *vant*





# Algoritmi care folosesc câștigul de informație

- Algoritmul ID3 are la bază calculul câștigului de informație pentru a alege attributele de test în punctele de divizare
  - se selectează attributele care furnizează cel mai mare câștig de informație
- C4.5 conține o serie de facilități suplimentare față de ID3.
  - în construcția arborelui de decizie se poate lucra cu seturi de formare care conțin înregistrări cu valori ale atributelor necunoscute.
    - În acest caz, câștigul de informație sau rata câștigului pentru un atribut se evaluează considerând numai înregistrările ale căror attribute au valori definite
- **restricții**
  - arborii de decizie permit numai o singură variabilă dependentă.
    - Pentru a prezice mai mult de o variabilă dependentă, trebuie create alte modele.
  - cei mai mulți algoritmi de inducție a arborilor de decizie cer ca datele continue să fie grupate sau convertite la date de tip enumerare.

# Atribute cu un numar mare de valori distincte

- Pot genera un numar mare de noduri copil  
=> apar probleme cu calculul castigului de informatie
- Exemplu – cazul extrem cand un atribut are cate o valoare diferita pentru fiecare instanta din multimea de date