Curs 3:

Tehnici de clasificare a datelor

= introducere =

Structura

- Motivare
- Concepte de bază în clasificare
- Măsuri de performanţă
- Clasificatori simpli
  - Bazați pe clasa majoritară (ZeroR)
  - Reguli simple de clasificare (OneR)

Motivare

Reminder: exemple de probleme de clasificare

- Clasificarea celulelor tumorale in benigne sau maligne (diagnoză medicală)
- Clasificarea tranzacţiilor efectuate cu cărți de credit ca fiind legitime sau frauduloase (detecție activități frauduloase)
- Clasificarea știrilor pe domenii: finanțe, meteo, divertisment, sport, etc (clasificare documente)
- Clasificarea e-mail-urilor ca spam sau utile (spam filtering)
- alte exemple ...

Motivare

- Diagnoza medicală = predicția prezenței/absenței unei boli pe baza informațiilor disponibile într-o înregistrare medicală

Exemplu de set de date (breast-cancer-wisconsin - arff format – Lab 1)

```
@relation wisconsin-breast-cancer
@attribute Clump_Thickness integer [1,10]
@attribute Cell_Size_Uniformity integer [1,10]
@attribute Cell_Shape_Uniformity integer [1,10]
@attribute Marginal_Adhesion integer [1,10]
@attribute Single_Epi_Cell_Size integer [1,10]
@attribute Bare_Nuclei integer [1,10]
@attribute Bland_Chromatin integer [1,10]
@attribute Normal_Nucleoli integer [1,10]
@attribute Mitoses integer [1,10]
@attribute Class { benign, malignant}
@data
5,1,1,1,2,1,3,1,1,benign
5,4,4,5,7,10,3,2,1,benign
3,1,1,1,2,2,3,1,1,benign
8,10,10,8,7,10,9,7,1,malignant
1,1,1,1,2,10,3,1,1,benign
```

Concepte de bază

Ce se cunoaște?

- O colecție de înregistrări (instanțe) pentru care se cunoaște clasa căreia îi aparțin (set de date etichetate)
- Fiecare înregistrare conține un set de atribute, iar unul dintre aceste atribute este eticheta clasei

Ce se caută?

- un model care „captează” relația dintre atributul asociat clasei și celelalte atribute (modelul este construit folosind un set de antrenare printr-un proces numit antrenare/învățare supervizată)

Care este scopul final?

- Să se poată utiliza modelul construit prin antrenare pentru a determina clasa căreia îi aparține o nouă dată

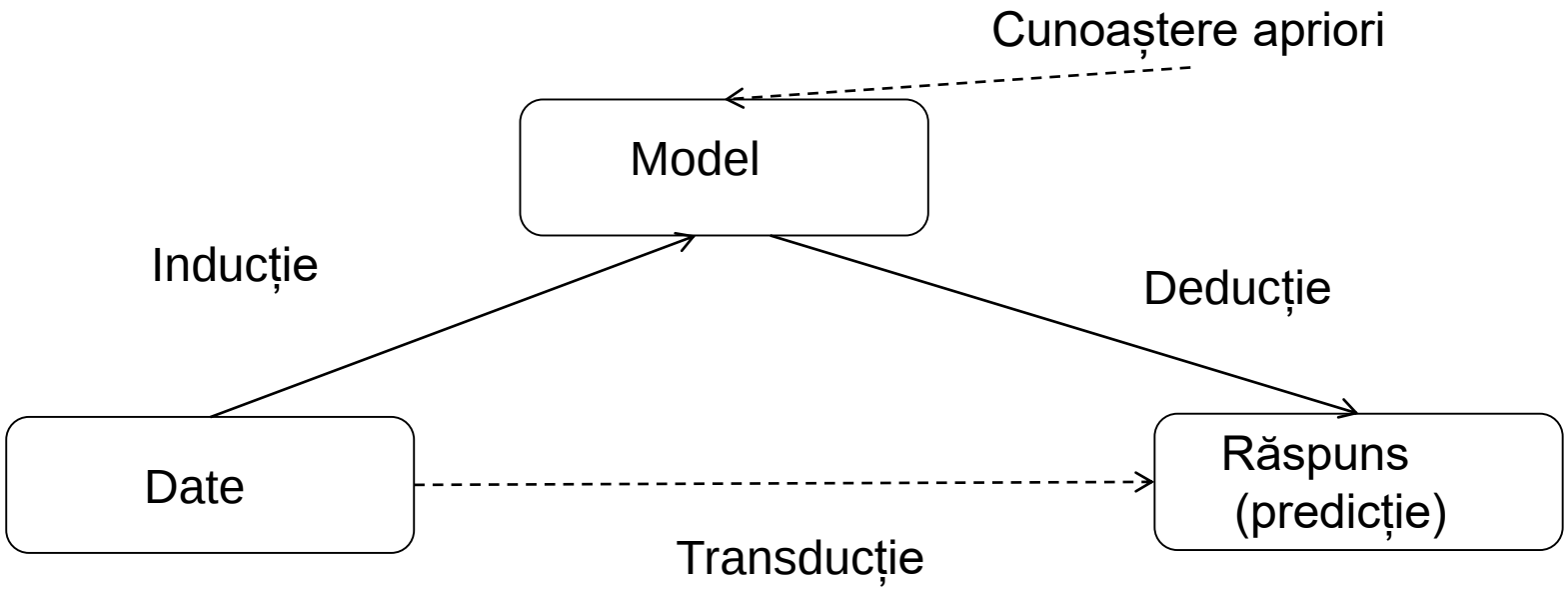
Obs:

- Pentru a fi util un model trebuie să aibă o bună acuratețe; acuratețea se măsoară analizând comportamentul modelului pentru date care nu au fost folosite în etapa de antrenare (set de testare)

Concepte de bază

Invățare/ inducție/ inferență = construirea unui model pornind de la date (și eventual de la cunoștințe apriori privind domeniul)

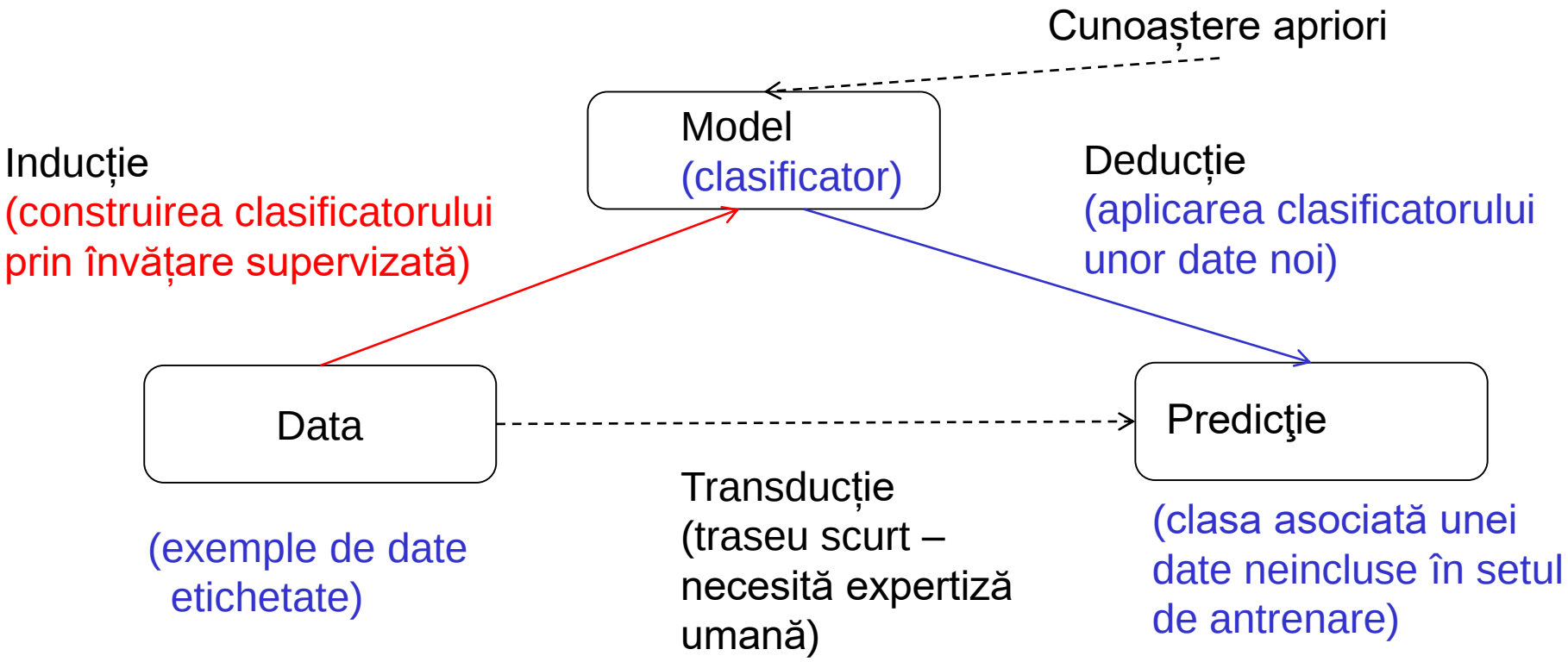
Între date, model, cunoștințe si răspunsuri există diferite legături:
inducție vs deducţie vs transducţie



Concepte de bază

Invățare/ inducție/ inferență = construirea unui model pornind de la date (si eventual de la cunoștințe apriori privind domeniul)

Intre date, model, cunoștințe si răspunsuri există diferite legături:
inducție vs deducție vs transducție



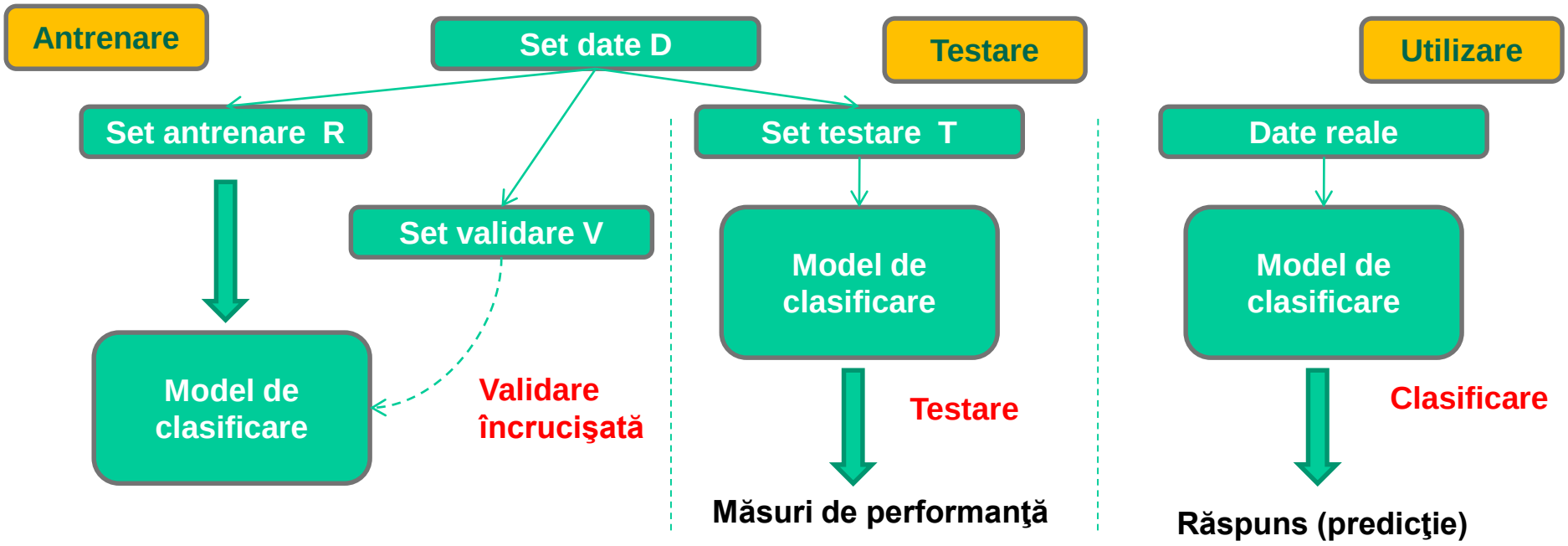
Concepte de bază

Informaţie disponibilă:

- Set de date etichetate:
  - $D=\{(x_1,c_1),(x_2,c_2),\ldots,(x_N,c_N)\}$
  - fiecare x_i are n atribute
  - Eticheta clasei c_i aparţine mulţimii $\{1,2,\ldots,K\}$

Scop:

- Construirea unui clasificator C folosind setul de date D a.î.
  - C poate prezice cărei clase îi aparţine o nouă dată x



Modele de clasificare

Un model de clasificare este o "mapare" între valori ale atributelor şi etichete ale claselor

Exemple de modele de clasificare:

- Arbori de decizie
- Reguli de clasificare
- Modele bazate pe prototipuri
- Modele probabiliste
- Modele bazate pe funcții (reţele neuronale, clasificatori cu vectori suport etc).

Un model de clasificare trebuie să fie:

- Acurat:
  - Identifică clasa corectă
- Compact / comprehensibil
  - Uşor de interpretat de către utilizator (preferabil să nu fie de tip "cutie neagră")
- Eficient în
  - Etapa de antrenare
  - Etapa de clasificare

Modele de clasificare

Exemplu

@relation wisconsin-breast-cancer
@attribute Clump_Thickness integer [1,10]
@attribute Cell_Size_Uniformity integer [1,10]
@attribute Cell_Shape_Uniformity integer [1,10]
@attribute Marginal_Adhesion integer [1,10]
@attribute Single_Epi_Cell_Size integer [1,10]
@attribute Bare_Nuclei integer [1,10]
@attribute Bland_Chromatin integer [1,10]
@attribute Normal_Nucleoli integer [1,10]
@attribute Mitoses integer [1,10]
@attribute Class { benign, malignant}
@data
5,1,1,1,2,1,3,1,1,benign
5,4,4,5,7,10,3,2,1,benign
3,1,1,1,2,2,3,1,1,benign
8,10,10,8,7,10,9,7,1,malignant
1,1,1,1,2,10,3,1,1,benign
….

Regulă simplă de clasificare:
IF (Cell_Size_Uniformity< 3.5)
THEN benign
ELSE malignant

Modele de clasificare

Exemplu

@relation wisconsin-breast-cancer
@attribute Clump_Thickness integer [1,10]
@attribute Cell_Size_Uniformity integer [1,10]
@attribute Cell_Shape_Uniformity integer [1,10]
@attribute Marginal_Adhesion integer [1,10]
@attribute Single_Epi_Cell_Size integer [1,10]
@attribute Bare_Nuclei integer [1,10]
@attribute Bland_Chromatin integer [1,10]
@attribute Normal_Nucleoli integer [1,10]
@attribute Mitoses integer [1,10]
@attribute Class { benign, malignant}
@data
5,1,1,1,2,1,3,1,1,benign
5,4,4,5,7,10,3,2,1,benign
3,1,1,1,2,2,3,1,1,benign
8,10,10,8,7,10,9,7,1,malignant
1,1,1,1,2,10,3,1,1,benign
....

Regulă simplă de clasificare:

IF (Cell_Size_Uniformity< 3.5)
  THEN benign
  ELSE malignant

Întrebare: Cât de bună este această regulă?

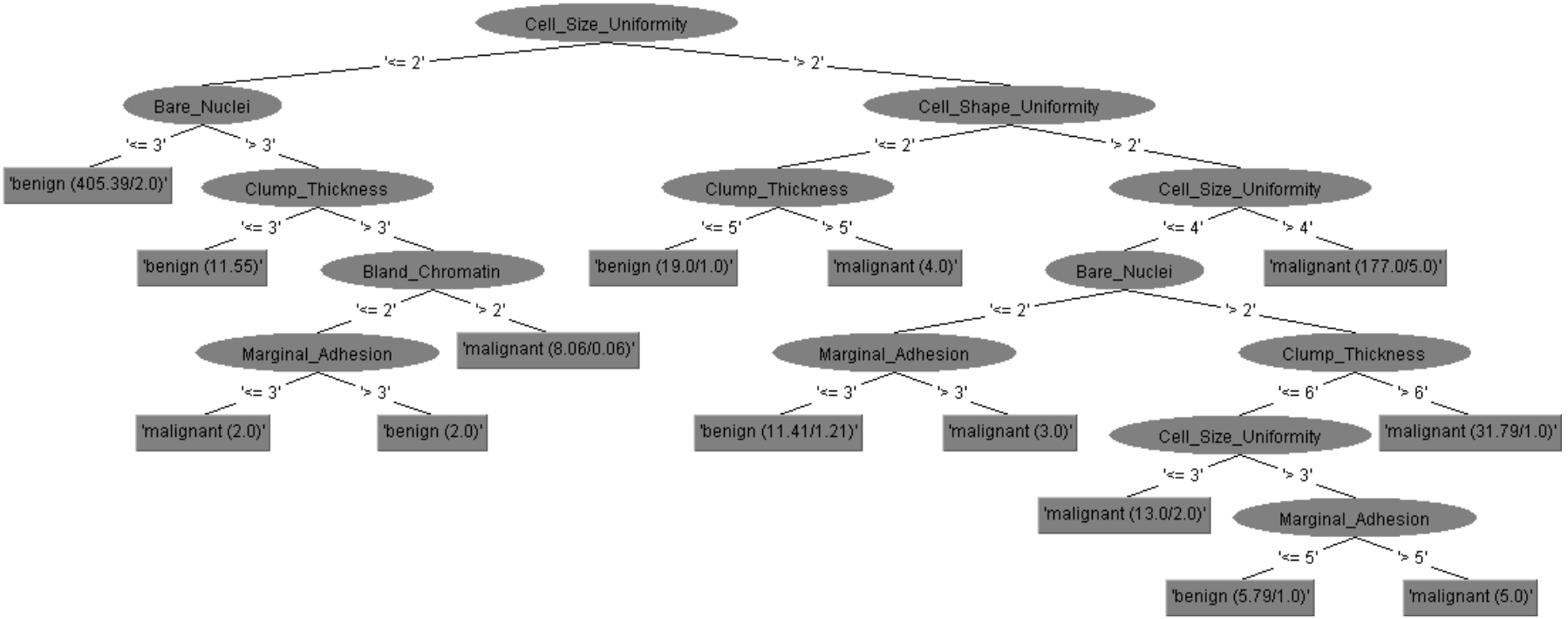
Pentru 92.7% dintre exemplele din setul de antrenare indică clasa corectă

Cum a fost calculată această valoare?

Cum ar trebui interpretată?

Modele de clasificare

Exemplu: un model mai complex (arbore de decizie)



Performanţa: în 94.56% din cazuri clasificatorul indică clasa corectă

Ce se poate spune despre lizibilitatea clasificatorului?

Măsuri de performanţă

Context: considerăm o problemă de clasificare în 2 clase

- Clasa P – pozitivă (ex: malign)
- Clasa N – negativă (ex: benign)

Cel mai simplu mod de a măsura performanţa este de a analiza în câte cazuri clasificatorul indică răspunsul corect – această informaţie poate fi furnizată prin intermediul matricii de confuzie

Matrice de confuzie:

	CP	CN
CP	TP	FN
CN	FP	TN

← răspunsul clasificatorului

↑ Clasa adevărată

TP = True Positive = nr de cazuri din CP care sunt clasificate (corect) în CP

TN = True Negative = nr de cazuri din CN care sunt clasificate (corect) în CN

FP = False Positive = nr de cazuri din CN dar care sunt clasificate (incorect) în CP

FN = False Negative = nr de cazuri din CP dar care sunt clasificate (incorect) în CN

Măsuri de performanţă

Cazul a K clase:

- Se poate construi câte o matrice de confuzie 2x2 pt fiecare dintre clase (clasa curentă este considerată clasa pozitivă şi toate celelalte clase sunt agregate în clasa negativă)
- Se extinde matricea la cazul a K clase: K linii şi K coloane

Matrice de confuzie:

	C_1	C_2	...	C_j	… C_K
C_1	T_1	F_{12}	…	F_{1j}	… F_{1K}
C_2	F_{21}	T_2	…	F_{2j}	… F_{2K}
…		…	…	…	
C_i	F_{i1}	F_{i2}		F_{ij}	F_{iK}
…		…	…	…	
C_K	F_{K1}	F_{K2}		F_{1Kj}	T_K

↑ Clasa adevărată

← răspunsul clasificatorului

F_{ij} = nr de cazuri care ar trebui clasificate în C_i dar sunt clasificate în C_j

T_i = nr de cazuri din C_i care sunt corect clasificate în C_i

Măsuri de performanţă

TP = True Positive = nr de cazuri din CP care sunt clasificate (corect) în CP
TN = True Negative = nr de cazuri din CN care sunt clasificate (corect) în CN
FP = False Positive = nr de cazuri din CN dar care sunt clasificate (incorect) în CP
FN = False Negative = nr de cazuri din CP dar care sunt clasificate (incorect) în CN

Acurateţe = (TP+TN)/(TP+TN+FP+FN) = nr date clasificate corect/ nr total de date

Sensitivitate = TP/(TP+FN) (TP rate sau recall = rata de regăsire)

Specificitate = TN/ (TN+FP) (TN rate), 1-specificitate=FP/(TN+FP) = FP rate

Precizie = TP/(TP+FP) (nr cazuri real pozitive/ nr cazuri clasificate ca fiind pozitive)

Obs:

- Toate valorile sunt în [0,1]; valori mai mari sugerează performanţă mai bună
- Sensitivitatea şi specificitatea sunt utilizare frecvent în analiza datelor medicale
- Precizia şi rata de regăsire se folosesc în domeniul regăsirii informaţiei (information retrieval)

Măsuri de performanţă

TP = True Positive = nr de cazuri din CP care sunt clasificate (corect) în CP
TN = True Negative = nr de cazuri din CN care sunt clasificate (corect) în CN
FP = False Positive = nr de cazuri din CN dar care sunt clasificate (incorect) în CP
FN = False Negative = nr de cazuri din CP dar care sunt clasificate (incorect) în CN

In contextul regăsirii informaţiei:

Precision = TP/(TP+FP) = card(relevant şi regăsit)/ card(regăsit)

Recall = TP/(TP+FN) = card(relevant şi regăsit)/ card(relevant)

Obs: Precision = 1 ➔ FP=0 Recal = 1 ➔ FN=0

O variantă agregată frecvent utilizată este media armonică a acestora:

F-measure=2*precision*recall/(precision+recall)

Măsuri de performanţă

Acurateţe ponderată de costuri (Cost sensitive accuracy)

- In anumite cazuri (ex: diagnoză medicală) clasificarea incorectă într-o clasă poate avea un impact mai mare decât clasificarea incorectă în altă clasă (e.g. nedetectarea unui caz de cancer poate fi mai periculoasă decât nedetectarea unui caz normal) - FN ar trebui să fie cât mai mic (senzitivitate mare)
- In alte cazuri (detecţie malware) ar trebui ca FP să fie cât mai mic (specificitate mare)

In astfel de situaţii se pot utiliza ponderi (interpretate ca fiind costuri ale erorii în clasificare) diferite pentru cele două tipuri de erori

- $CostAccuracy=(cost_1*n_1*sensitivity+cost_2*n_2*specificity)/(cost_1*n_1+cost_2*n_2)$
  - $cost_i$ = costul clasificării incorecte a datelor din clasa C_i
  - n_i = numărul de date din C_i

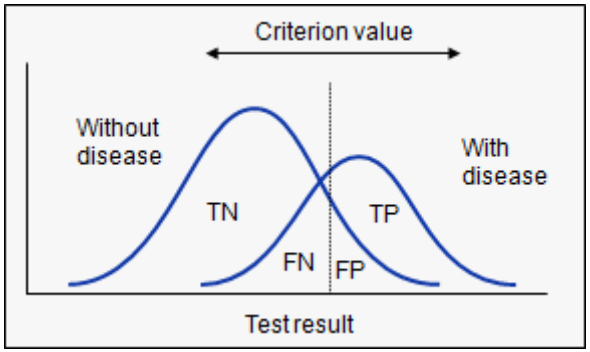
C_1 = CP = clasa pozitivă, C_2 = CN = clasa negativă

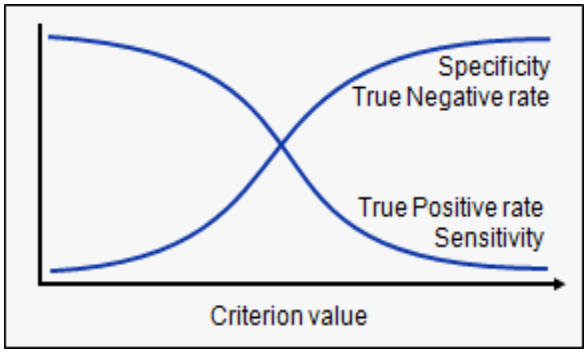
Măsuri de performanţă

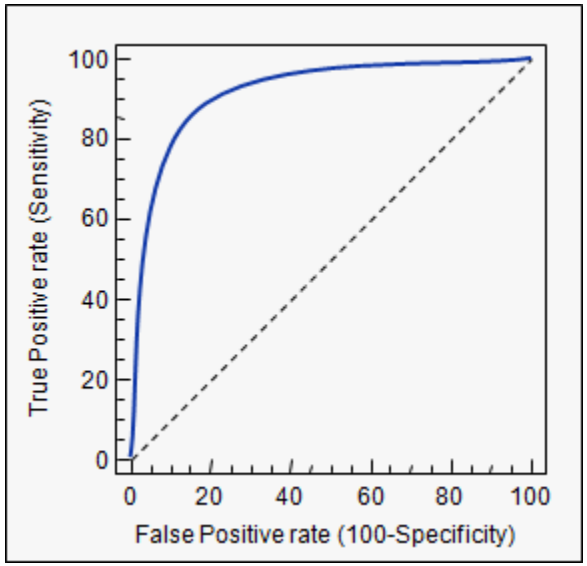
Curba ROC (Receiver Operator Characteristics) si Aria de sub curbă (AUC)

- util pentru evaluarea performanţei unui clasificator bazat pe valoare prag
  (ex: IF (Cell_Size_Uniformity< 3.5) THEN benign ELSE malignant)
- se reprezintă grafic punctele cu coordonatele
  (FP rate, TP rate) = (1-specificitate, senzitivitate)

pentru diferite valori ale pragului (sau pentru diferite subseturi de date în cazul în care se foloseşte validare încrucişată)







Sursa: https://www.medcalc.org/manual/roc-curves.php

Cel mai simplu clasificator

Exemplu:

- Considerăm setul de date “sick” de la UCI Machine Learning
- Conţine 3772 înregistrări (aferente unor pacienţi), dintre care:
  - 231 sunt bolnavi (clasa C1 – pozitivă) – 6% dintre pacienţi
  - 3541 sunt sănătoşi (clasa C2 – negativă) – 94% dintre pacienţi
- Ne interesează să construim un clasificator a cărui acurateţe pt acest set de date să fie cel puţin egală cu 0.9 (90%)
- Care este cel mai simplu clasificator care satisface această cerinţă?

Cel mai simplu clasificator

Exemplu:

- Considerăm setul de date “sick” de la UCI Machine Learning
- Conţine 3772 înregistrări (aferente unor pacienţi), dintre care:
  - 231 sunt bolnavi (clasa C1 – pozitivă) – 6% dintre pacienţi
  - 3541 sunt sănătoşi (clasa C2 – negativă) – 94% dintre pacienţi
- Ne interesează să construim un clasificator a cărui acurateţe pt acest set de date să fie cel puţin egală cu 0.9 (90%)
- Care este cel mai simplu clasificator care satisface această cerinţă?
- Considerând regula: “indiferent de valorile atributelor clasa este C2 (negativă)” obţinem acurateţea=3541/3772=0.94>0.9
- Este un astfel de clasificator adecvat? Are vreo utilitate?

Cel mai simplu clasificator

- Este un astfel de clasificator adecvat? Are vreo utilitate?

- Acest clasificator, denumit ZeroR (întrucât se bazează pe o regulă de clasificare care nu conţine nici un atribut în membrul stâng) utilizează doar distribuţia datelor în cele două clase şi va returna întotdeauna eticheta celei mai populare clase (se bazează pe un mecanism simplu de votare)

- Nu este adecvat întrucât produce răspuns incorect pt toate datele din clasa cu mai puţine elemente

…din nou la evaluarea performanţei

- Utilizarea întregului set de date disponibile pentru construirea clasificatorului nu este o abordare prea înţeleaptă întrucât poate conduce la supra-antrenare:
  - Clasificatorul se comportă bine pentru datele din setul de antrenare...
  - ... dar are performanţe slabe pentru alte date
- O abordare mai bună este să se dividă setul de date în:
  - Subset de antrenare (utilizat pt construirea clasificatorului)
  - Subset de testare (utilizat pt estimarea performanţei)

- Există diferite strategii de divizare a setului de date în subseturi (antrenare şi testare)

Obs: Pe lângă subsetul de testare se poate folosi şi un subset de validare (utilizat pentru ajustarea parametrilor clasificatorului)

…din nou la evaluarea performanţei

Strategii de divizare:

- Holdout
  - Se reţin 2/3 din set pt antrenare şi 1/3 pt testare (procentul datelor utilizate pentru antrenare poate fi altul, însă de regulă este mai mare de 50%)

- Holdout repetat
  - Se repetă partiţionarea (performanţa este calculată ca medie a valorilor determinate la fiecare repetare a divizării)

- Validare încrucişată
  - Se divide aleator setul de date în k subseturi disjuncte
    - Obs: presupune aplicarea unei permutări aleatoare asupta setului inițial de date
  - k-fold: se folosesc k-1 subseturi pt antrenare, iar al k-lea se foloseşte pt testare (evaluarea performanţei)
  - Leave-one-out: k=n (caz particular – se utilizează, la fiecare etapă, un singur exemplu pentru evaluarea performanței)

…din nou la evaluarea performanţei

Strategii de divizare:

- Eşantionare repetată (util în cazul seturilor nebalansate)
  - oversampling vs undersampling
- Bootstrap
  - selecţie cu revenire (se selectează un exemplu din set după care se pune la loc pentru a putea fi reselectat)

Dincolo de ZeroR

Set de date: sick.arff, 29 atribute, 3772 instanţe (231 în clasa C1, 3541 în clasa C2), 2 clase

ZeroR (clasa e întotdeauna C2): acurateţe=0.94

OneR: permite construirea de reguli de clasificare care conţin un singur atribut în membrul stâng

Exemple de reguli (obţinute folosind Weka OneR):

If T3< 0.25 then C2 (negative)

If T3 in [0.25, 0.35) then C1 (sick)

If T3 in [0.35, 0.55) then C2 (negative)

If T3 in [0.55, 1.15) then C1 (sick)

If T3 >= 1.15 then C2 (negative)

If T3 value is missing then C2 (negative) Acurateţe: 0.96

OneR

Ideea principală: identifică atributul cu cea mai mare putere de discriminare şi îl utilizează pentru a defini regulile de clasificare

Obs: este adecvat pentru atributele care au valori discrete

Algoritm:

FOR each attribute A_i do

FOR each value v_{ij} of A_i construct

R_{ij} : if $A_i = v_{ij}$ then class $C_{k(i,j)}$

(clasa majoritară pt instanţele care au $A_i = v_{ij}$)

se combină regulile într-un set R_i corespunzător lui A_i şi se calculează

Err_i (nr date clasificate incorect)

ENDFOR

ENDFOR

Selectează setul de reguli cu eroarea cea mai mică

OneR

Exemplu: weather/play dataset

Outlook: err=4

sunny: 2 yes/ 3 no (→ no)

overcast: 4 yes/ 0 no (→ yes)

rainy: 3 yes/2 no (→yes)

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

OneR

Example: weather/play dataset

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Outlook: err=4

sunny: 2 yes/ 3 no (→ no)

overcast: 4 yes/ 0 no (→ yes)

rainy: 3 yes/2 no (→yes)

Temperature: err=5

hot: 2 yes/2 no (→ yes)

mild: 4 yes/2 no (→ yes)

cool: 3 yes/ 1no (→ yes)

OneR

Example: weather/play dataset

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Outlook: err=4

sunny: 2 yes / 3 no (→ no)

overcast: 4 yes / 0 no (→ yes)

rainy: 3 yes / 2 no (→yes)

Temperature: err=5

hot: 2 yes / 2 no (→ yes ?)

mild: 4 yes / 2 no (→ yes)

cool: 3 yes / 1no (→ yes)

Humidity: err=5

high: 4 yes / 4 no (→ yes ?)

normal: 6 yes / 1 no (→ yes)

OneR

Exemplu: weather/play dataset

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Outlook: err=4

sunny: 2 yes / 3 no (→ no)

overcast: 4 yes / 0 no (→ yes)

rainy: 3 yes /2 no (→yes)

Temperature: err=5

hot: 2 yes /2 no (→ yes)

mild: 4 yes /2 no (→ yes)

cool: 3 yes / 1no (→ yes)

Humidity: err=5

high: 4 yes / 4 no (→ yes)

normal: 6 yes / 1 no (→ yes)

Windy: err=5

true: 3 yes / 3 no (→ yes)

false: 6 yes / 2 no (→ yes)

Reguli: weather/play dataset

If outlook=sunny then "no"

If outlook=overcast then "yes"

If outlook=rainy then "yes"

Acurateţe (set antrenare): 0.71

Acurateţe (validare încrucişată): 0.43 (!!)

OneR

Exemplu: weather/play dataset

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Reguli: weather/play dataset

If outlook=sunny then “no”

If outlook=overcast then “yes”

If outlook=rainy then “yes”

Etapa de clasificare:

- Altă zi: (outlook=rainy, temperature=cool, humidity=high, windy=false)
- Răspuns: Yes

OneR

Sumar implementare OneR

- Construirea setului de reguli (etapa de antrenare)
  - Input: set de antrenare (instanţe etichetate)
  - Output: set de reguli simple (toate regulile implică un singur atribut – acelaşi atribut în toate)
  - Algoritm: se analizează toate atributele şi valorile corespunzătoare acestora şi se selectează atributul pentru care eroarea de clasificare este minimă
- Utilizarea regulilor (etapa de clasificare)
  - Input: set de reguli, dată (instanţă) nouă
  - Output: eticheta clasei
  - Algoritm:
    - Identifică regula care se potriveşte cu data
    - Returnează clasa corespunzătoare regulii identificate

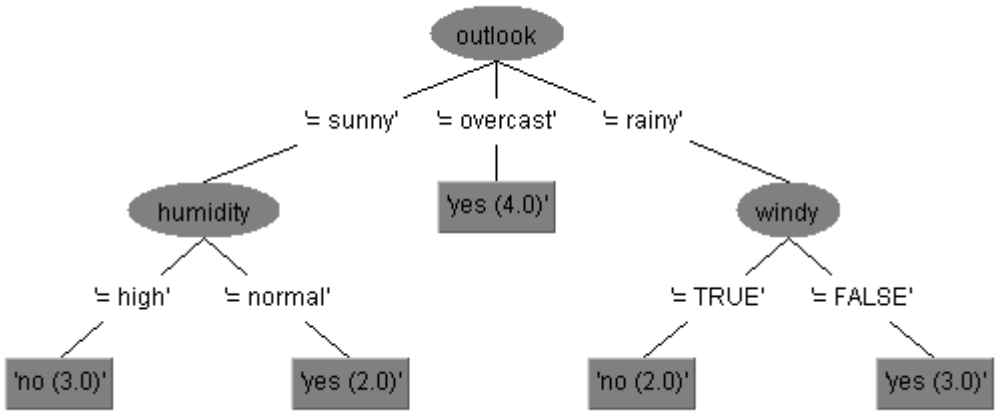
Etapa următoare: arbori de decizie

Set de date: weather/play

Relation: weather.symbolic

No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	**play** Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Arbore de decizie (construit folosind Weka)



Cum poate fi construit un arbore de decizie? Cum poate fi utilizat un arbore de decizie?

Ce clasă corespunde unei noi instanţe?
(outlook=sunny, temperature=mild, humidity=normal, windy=False)?

Sumar curs 3

- Set de date etichetate → (set de antrenare, set de validare, set de testare)
- Set de antrenare (+ set de validare) → model de clasificare
- Set de testare + măsuri de performanță → evaluare model
- Măsuri de performanță: acurațe, sensitivitate, specificitate, F1, AuC ...
- Alte caracteristici: interpretabilitate (explicabilitate)
- Cele mai simple modele: ZeroR, OneR (ușor de interpretat, acuratețe scăzută pe date reale)
- Pasul următor: modele mai complexe

Curs următor

- Arbori de decizie
  - Criterii de ramificare
  - Algoritmi de construire
- Reguli de clasificare
  - Seturi de reguli și proprietăți
  - Extragerea regulilor folosind algoritmi de acoperire