

LAPI – Laboratorul de
Analiza și Prelucrarea
Imaginilor



Universitatea
POLITEHNICA din
București



Facultatea de Electronică,
Telecomunicații și
Tehnologia Informației

TACAI - Tehnici de Analiză și Clasificare Automată a Informației

Note de laborator

Dr.ing. Ionuț Mironică

Conf.dr.ing. Bogdan Ionescu

Laborator 1

Cuprins

- Introducere în algoritmi de clasificare
- Introducere Weka
- Evaluarea performanței de clasificare
- Exerciții

I. Introducere

Ce este Machine learning?

- **Exemplu aplicație:** Este foarte greu de scris un algoritm care recunoaște un set de gesturi statice / dinamice sau care să rezolve problema de recunoaștere a feței:
 - Nu știm cum funcționează creierul uman pentru a clasifica gesturile;
 - Chiar dacă am ști nu am avea idee cum să programăm deoarece ar fi foarte complicat;
 - Ar trebui să scriem o funcție diferită pentru fiecare gest.
- În loc să scriem programe foarte multe, putem colecta exemple care specifică fiecare gest;
- Un algoritm de învățare va prelua aceste exemple și va “creea” un program care va face această clasificare în mod automat;

I. Introducere

Ce este Machine learning?

- Există mii de algoritmi de învățare / sute dintre ei apar anual;
- Există mai multe tipuri de învățare:

Învățare supervizată

- datele de antrenare conțin și ieșirea dorită;

Învățare nesupervizată

- datele de antrenare nu conțin ieșirea dorită (clusterizare);
- ideea de bază este de a se găsi șabloane și pattern-uri în date care să fie evidențiate în mod automat.

Învățare semi-supervizată

- doar o parte din datele de antrenare conțin ieșirea dorită;

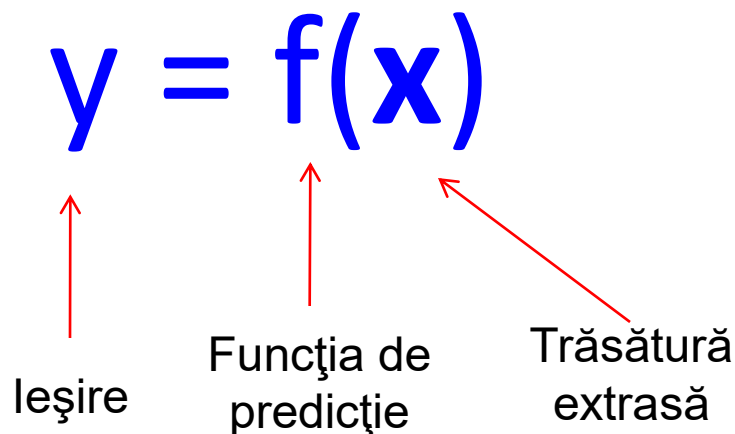
Reinforcement Learning

- se învață în funcție de feedback-ul primit după ce o decizie este luată.

I. Introducere

Învățare supervizată

Model de bază



Antrenare: fiind dată o mulțime de antrenare împreună cu răspunsul dorit $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, se estimează predicția funcției f prin minimizarea erorii de predicție pe mulțimea de antrenare;

Testare: se aplică funcția f pe un exemplu de test $\mathbf{x}$ (care nu a fost folosit în procesul de antrenare) și prezintă ieșirea funcției $y = f(\mathbf{x})$.

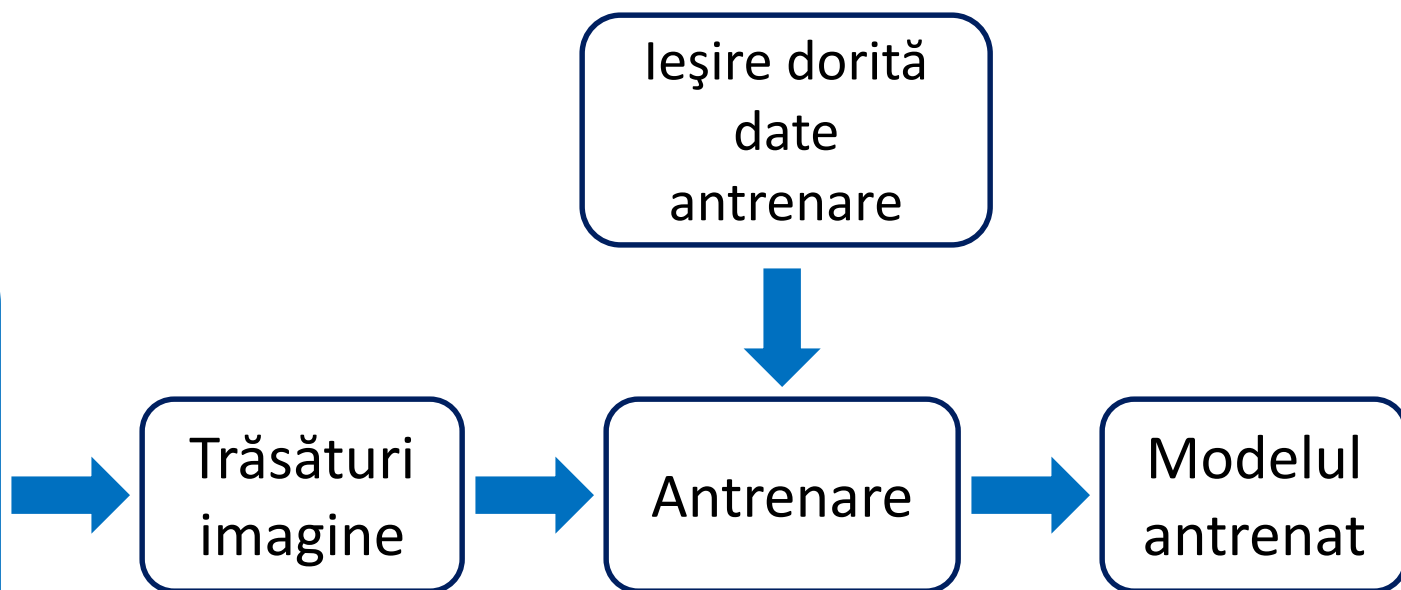
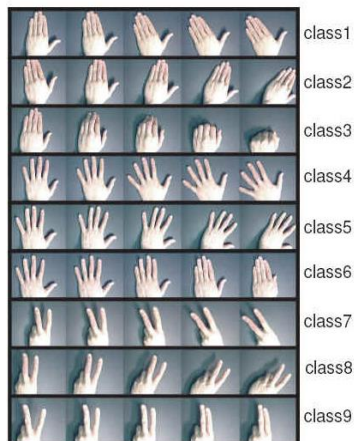
I. Introducere

Învățare supervizată

Model de bază

Antrenare

Imaginile de
antrenare

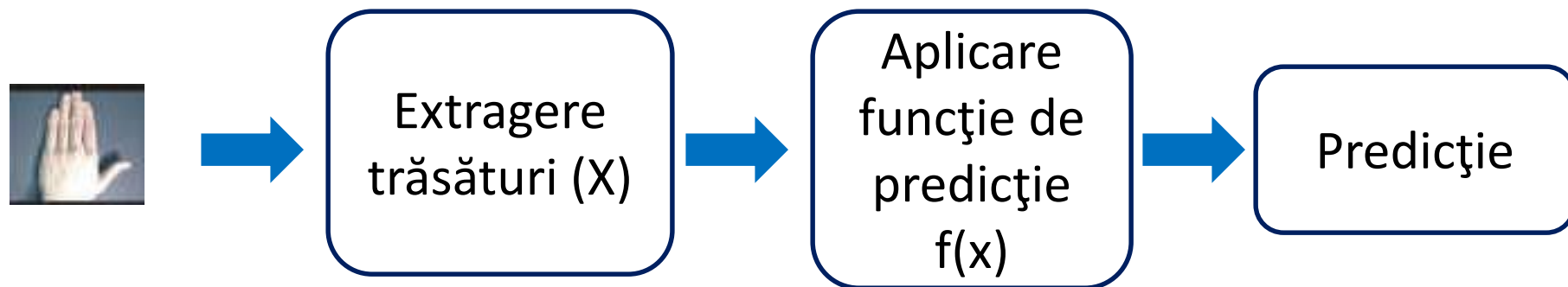


I. Introducere

Învățare supervizată

Model de bază

Testare



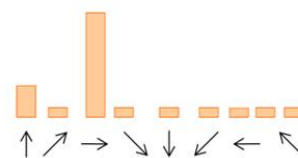
I. Introducere

Învățare supervizată

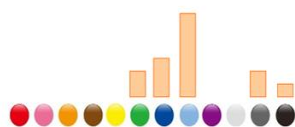
Ce sunt trăsăturile?



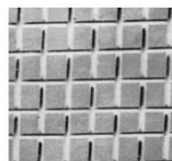
pixeli



muchii



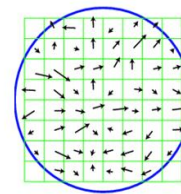
culoare



textură



formă

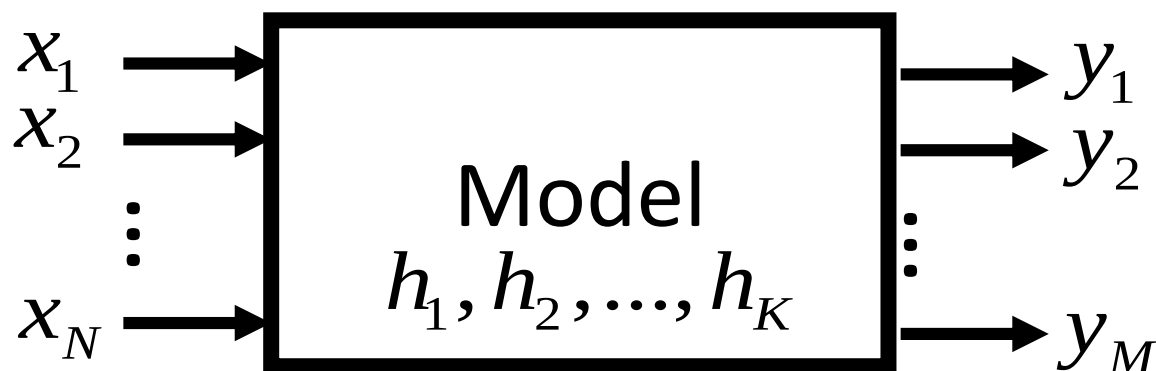


puncte de
interes

I. Introducere

Învățare supervizată

Învățare supervizată – schemă de bază



Variabile de intrare: $\mathbf{x} = (x_1, x_2, \dots, x_N)$

Variabile ascunse: $\mathbf{h} = (h_1, h_2, \dots, h_K)$

Variabile de ieșire: $\mathbf{y} = (y_1, y_2, \dots, y_K)$

I. Introducere

Învățare supervizată

Algoritmi existenți

- Support vector machines (SVM),
- Deep learning,
- Rețele recurente,
- Rețele bayesiene,
- Random forest,
- K-nearest neighbor (k-NN),

Etc.

Care este cel mai bun algoritm?

I. Introducere

Învățare supervizată

Teorema “No free lunch”



[<https://cynicalbabblings.wordpress.com>]

II. Weka

Informații generale

- Reprezintă o colecție de algoritmi de Machine Learning pentru diferite probleme de data-mining;
- Dezvoltat de către Universitatea din Waikato, Noua Zeelandă;
- Open-Source dezvoltat in JAVA (GNU General Public License);
- Trăsături principale:
 - instrumente de preprocesare,
 - algoritmi de învățare,
 - algoritmi de clusterizare,
 - reguli de asociere,
 - metode de evaluare,
 - interfață grafică,
 - instrumente pentru comparația clasificatorilor.

II. Weka

Documentație

- **Site-ul Weka:**

<http://www.cs.waikato.ac.nz/~ml/weka/>

- **Documentație Weka:**

<http://transact.dl.sourceforge.net/sourceforge/weka/WekaManual-3.6.0.pdf>

<http://www.cs.waikato.ac.nz/ml/weka/documentation.html>

- **Tutoriale video:**

<https://weka.waikato.ac.nz/explorer>

II. Weka

Interfață

Exporer:

- reprezintă componenta principală vizuală din Weka. Conține mai multe componente:

- *Preprocess*: opțiuni de încărcare a bazelor de date (format arff / csv) / conectare Sql / procesări atașate datelor.
- *Clasiffy*: algoritmi de clasificare;
- *Associate*: algoritmi de asociere;
- *Cluster*: algoritmi de învățare nesupervizată
- *Select attributes*: algoritmi de detecție a atributelor
- *Visualize*: modalități de vizualizare a datelor.

Experimenter:

- permite configurarea unui sistem ce asigură comparația sistematică a performanțelor algorimilor de clasificare pe diferite colecții de baze de date.

KnowledgeFlow:

- permite configurarea de fluxuri automate.

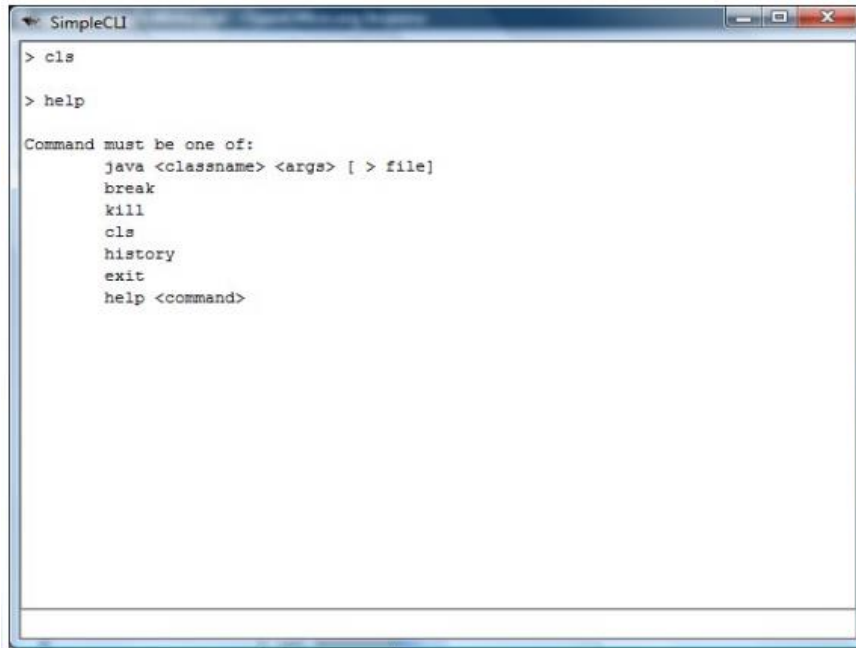
SimpleCLI:

- consolă de apelare a funcțiilor de clasificare.



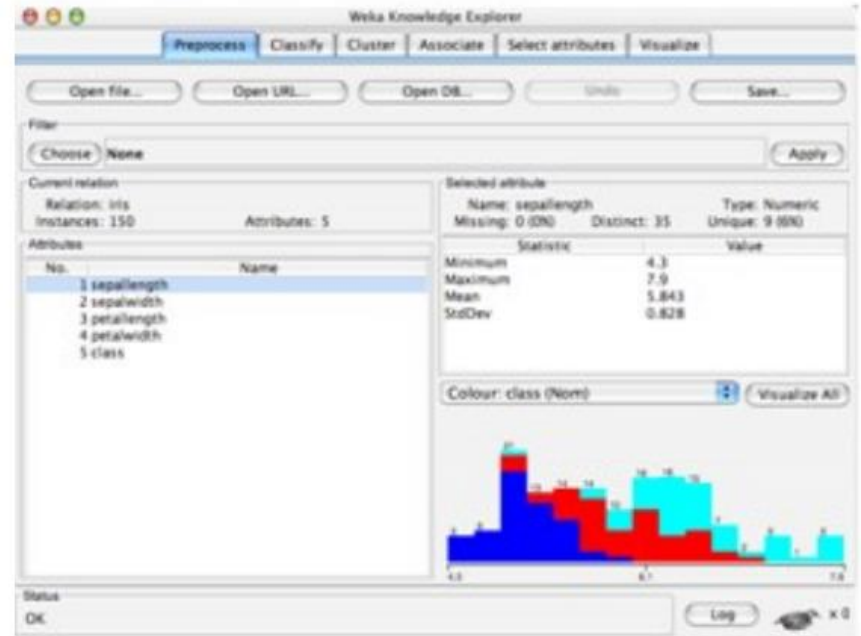
II. Weka

CLI vs GUI



```
> cls
> help

Command must be one of:
  java <classname> <args> [ > file]
  break
  kill
  cls
  history
  exit
  help <command>
```



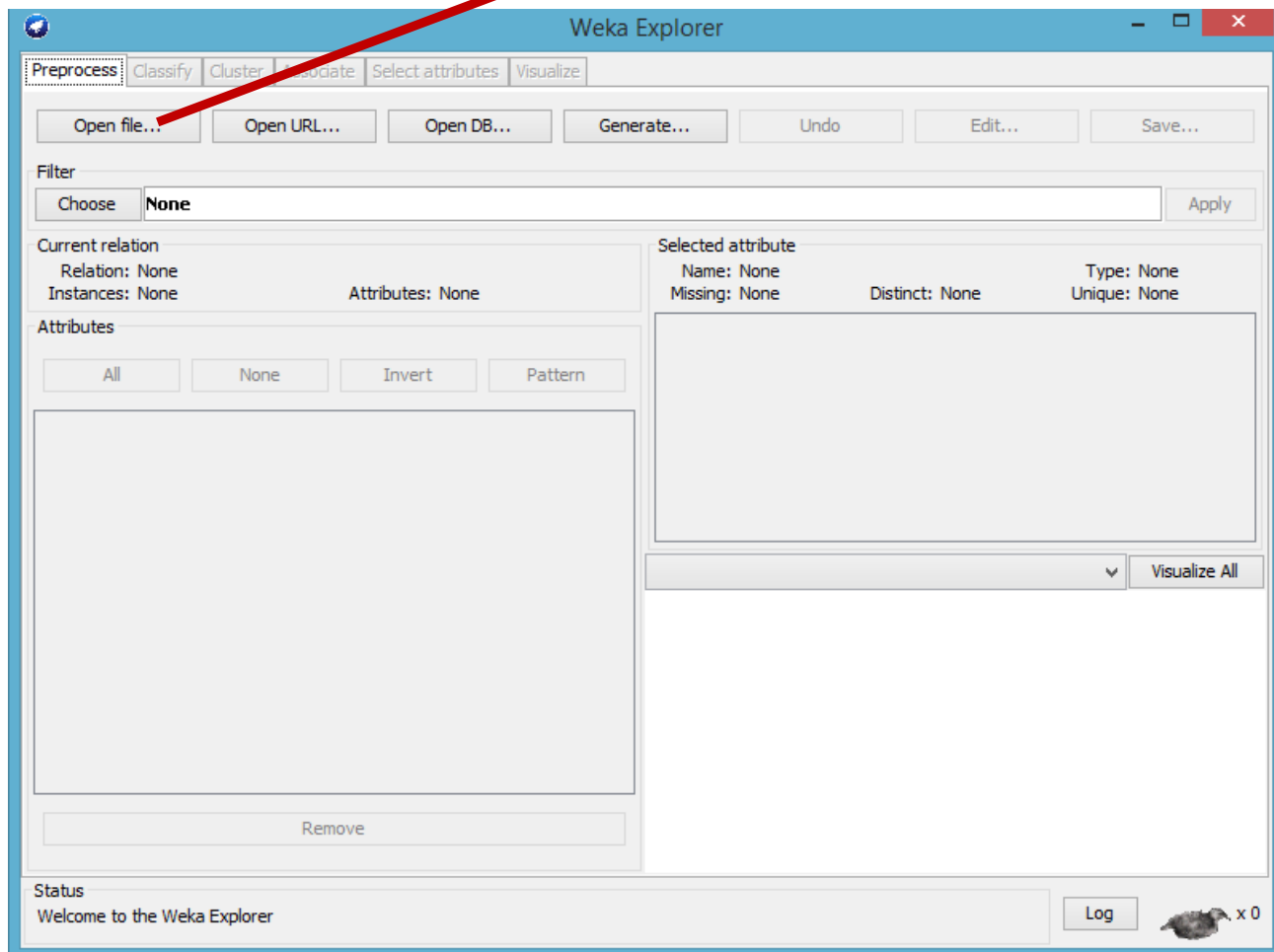
- Recomandat pentru utilizarea în profunzime a algoritmilor;
- Oferă funcționalități indisponibile în interfața grafică.
- Mai ușor de utilizat;
- Oferă funcționalități intuitive: Explorer, Experimenter și KnowledgeFlow.

II. Weka

Explorer: Preprocess

Încărcare fișiere

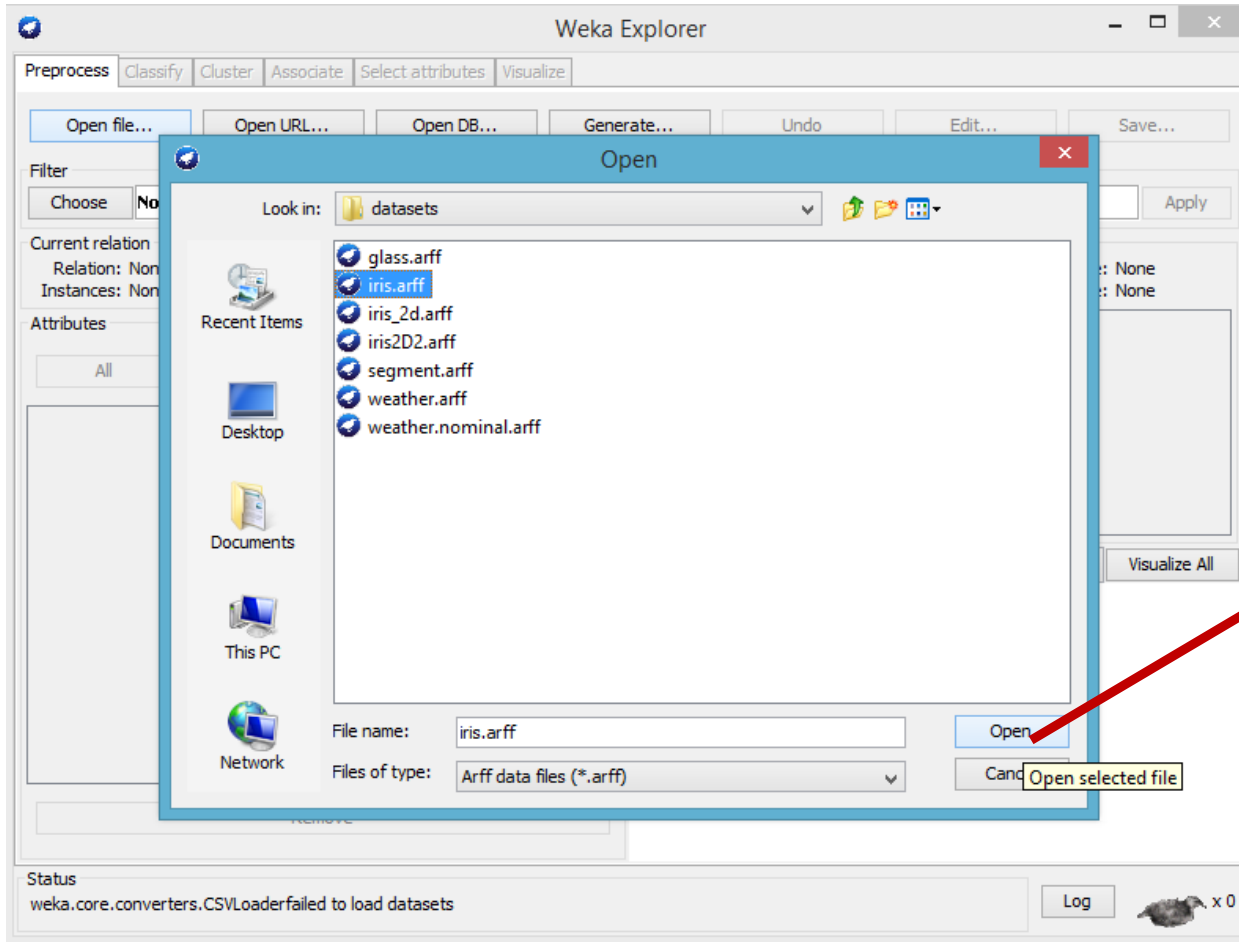
Open File



II. Weka

Explorer: Preprocess

Încărcare fișiere



Open:
Selecție fișier
(arff / csv)

II. Weka

Explorer: Preprocess

Încărcare fișiere

The screenshot shows the Weka Explorer interface in the Preprocess tab. The 'Open file...' button is highlighted. The 'Current relation' is 'iris' with 150 instances and 5 attributes. The 'Attributes' list shows 'sepalength' selected. The 'Selected attribute' panel displays statistics for 'sepalength': Name: sepalength, Missing: 0 (0%), Distinct: 35, Type: Numeric, Unique: 9 (6%). The 'Class' is set to 'class (Nom)'. A histogram at the bottom shows the distribution of 'sepalength' values across three classes: blue (class 0), red (class 1), and cyan (class 2). The histogram bars are labeled with their respective counts: 16 for class 0, 30 for class 1, 34 for class 2, 28 for class 1, 25 for class 2, 10 for class 1, and 7 for class 2.

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Class	Count
0 (blue)	16
1 (red)	30
2 (cyan)	34

II. Weka

Explorer: Preprocess

Structură fișier arff

```
% This is a toy example, the UCI weather dataset.  
% Any relation to real weather is purely coincidental  
  
@relation weather  
  
@attribute outlook {sunny, overcast, rainy}  
@attribute temperature real  
@attribute humidity real  
@attribute windy {TRUE, FALSE}  
@attribute play {yes, no}  
  
@data  
sunny,85,85,FALSE,no  
sunny,80,90,TRUE,no  
overcast,83,86,FALSE,yes  
rainy,70,96,FALSE,yes  
rainy,68,80,FALSE,yes  
rainy,65,70,TRUE,no  
overcast,64,65,TRUE,yes  
sunny,72,95,FALSE,no  
sunny,69,70,FALSE,yes  
rainy,75,80,FALSE,yes  
sunny,75,70,TRUE,yes  
overcast,72,90,TRUE,yes  
overcast,81,75,FALSE,yes  
rainy,71,91,TRUE,no
```

Comment (points to the first two lines of comments)

Dataset name (points to `@relation weather`)

Attributes (points to the list of attributes: `@attribute outlook {sunny, overcast, rainy}`, `@attribute temperature real`, `@attribute humidity real`, `@attribute windy {TRUE, FALSE}`, `@attribute play {yes, no}`)

Target / Class variable (points to the last attribute in the list: `@attribute play {yes, no}`)

Data Values (points to the data rows under the `@data` section)

II. Weka

Explorer: Preprocess

Attribute fișier arff

- **Attribute nominale:** valorile sunt selectate dintr-o listă predefinită;
- **Attribute numerice:** valorile sunt întregi sau reale;
- **Șiruri de caractere (string):** sunt încadrate între ghilimele;
- **Relaționale:** pentru date care aparțin în mai multe tipuri de instanțe.

II. Weka

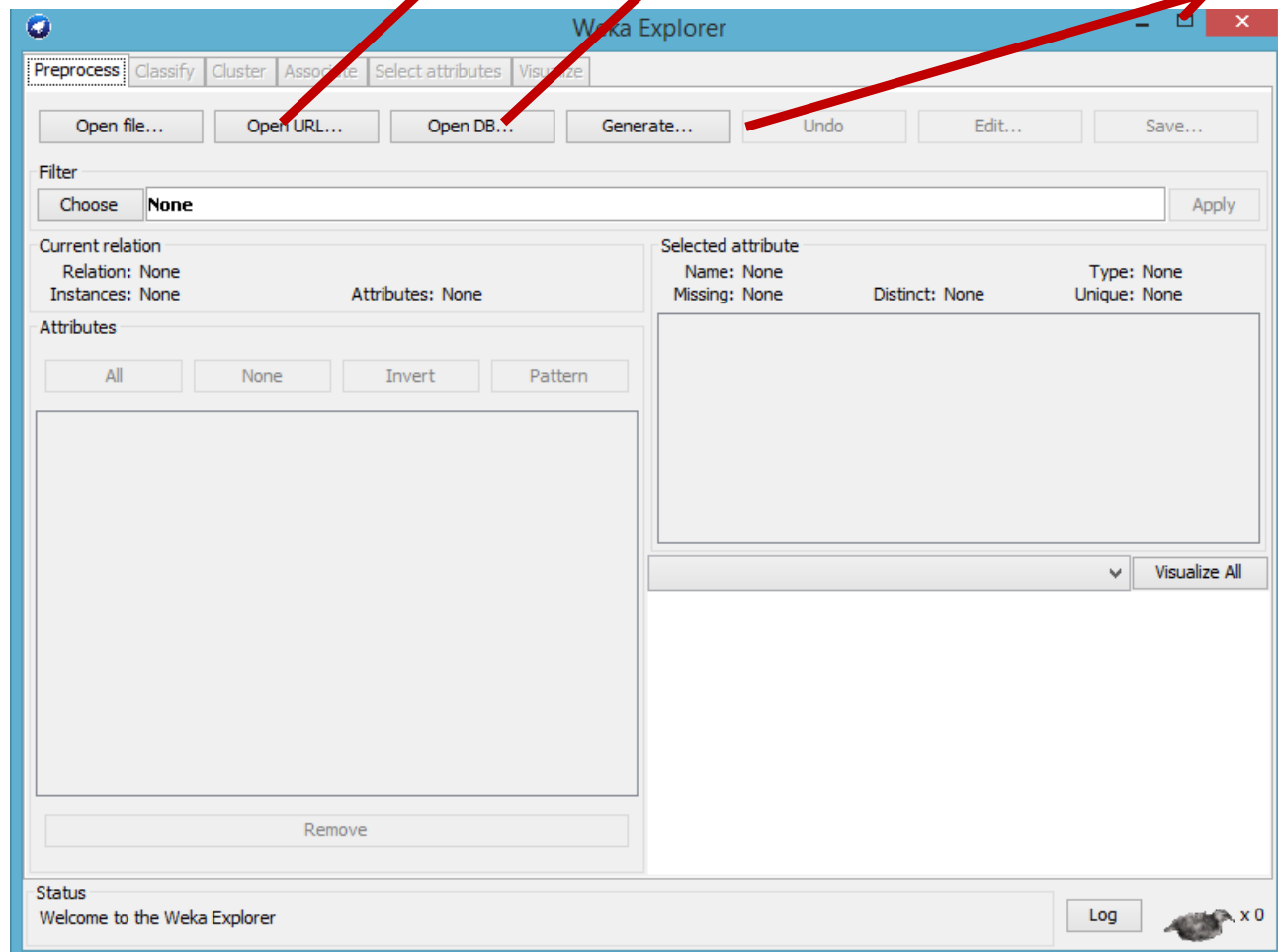
Explorer: Preprocess

Încărcare fișiere

Open URL: încărcare fișiere aflate la un link;

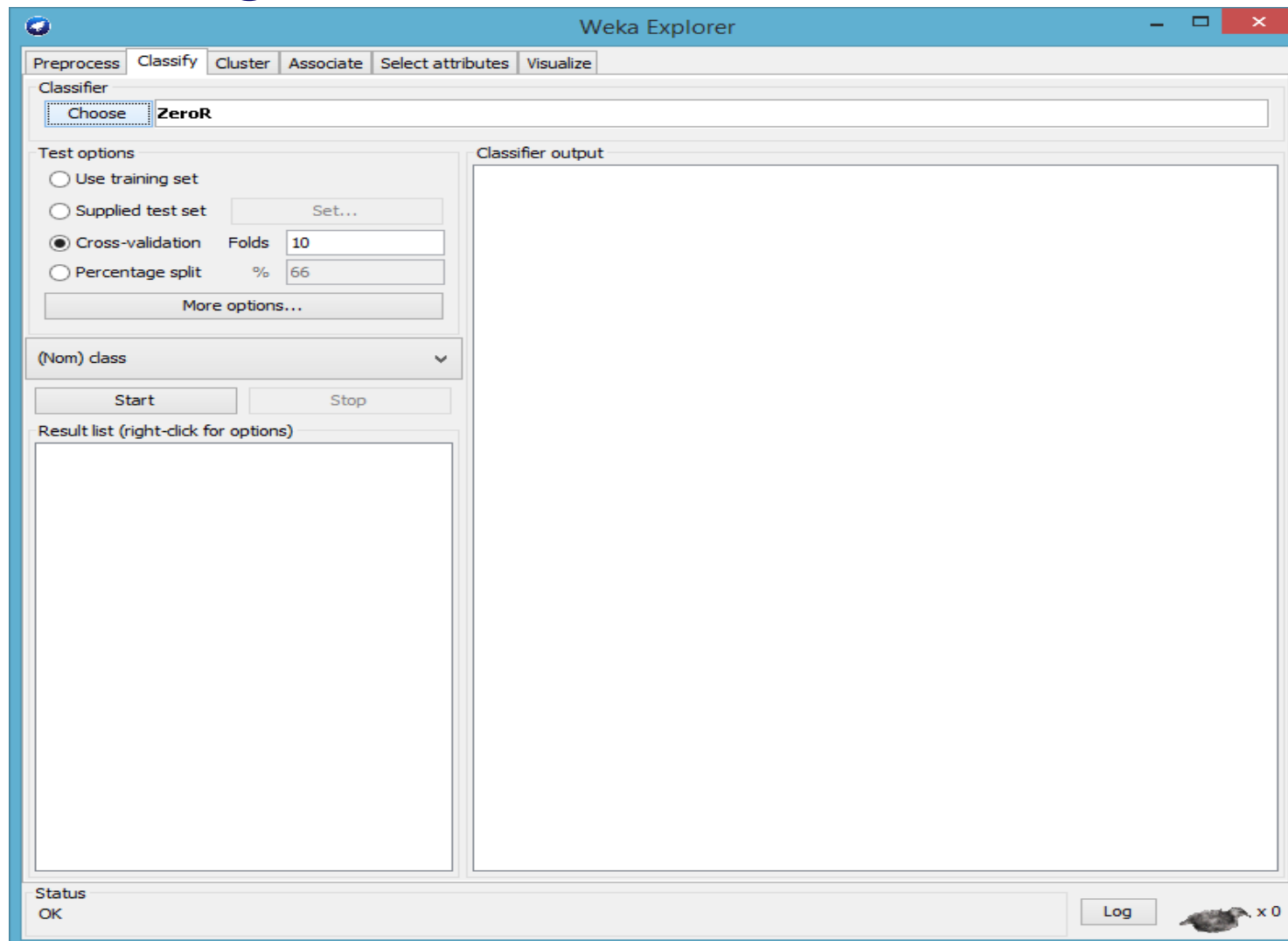
Open DB: conectare la o baza de date relațională (SQL) prin driver JDBC;

Generate: generare baze de date.



II. Weka

Utilizare algoritmi de clasificare



II. Weka

Explorer: Preprocess

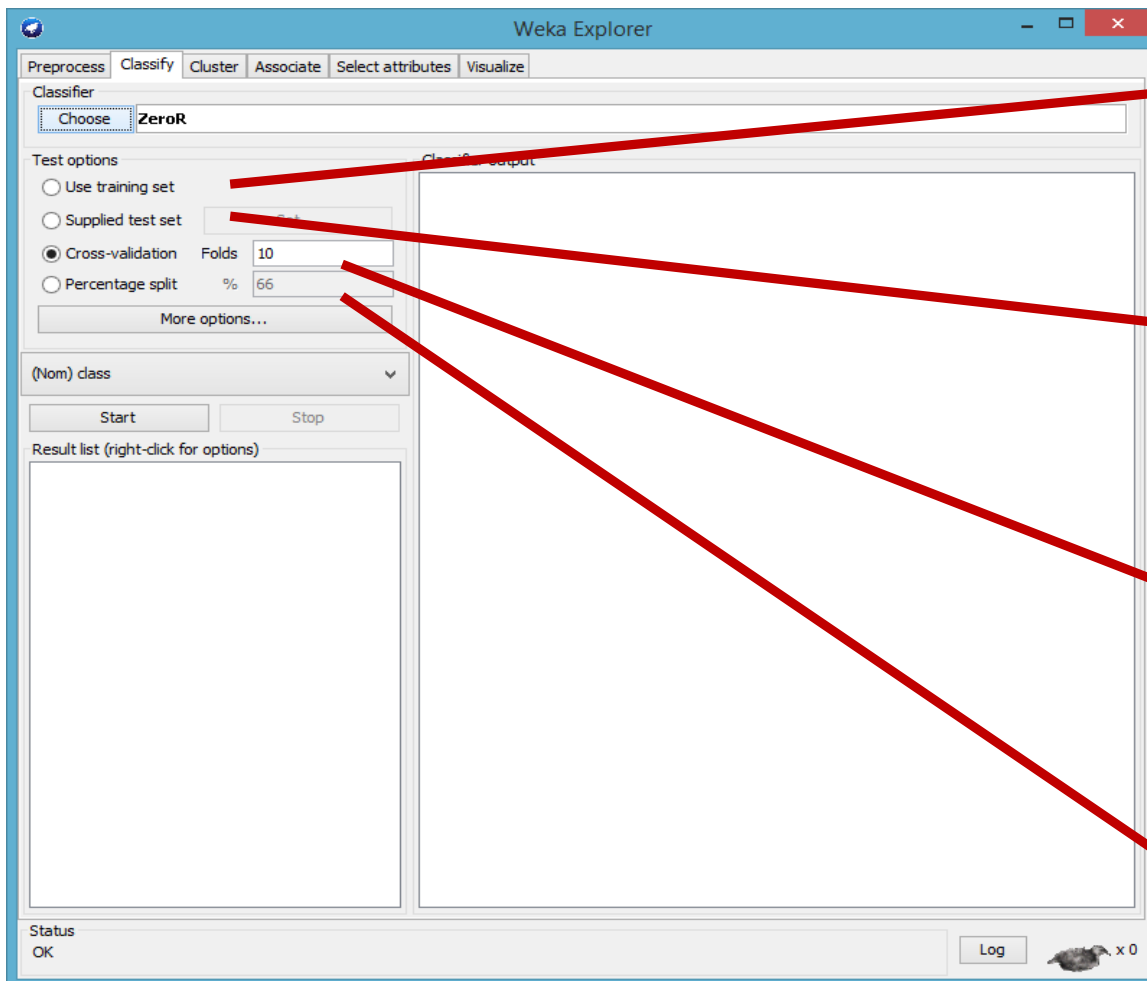
Încărcare fișiere

1. Deschide weka GUI
2. Click 'Explorer'
3. 'Open file...'
4. Selectează tabul 'Classify' Alege un clasificator
5. Selectează parametrii de clasificare
7. Click 'Start'
8. Wait...
9. Vizualizează rezultate
 - a. 'Salvează rezultat'
 - b. 'Salvează clasificator'

II. Weka

Utilizare algoritmi de clasificare

Opțiuni de împărțire a bazei de date (Test Options)



Use training set: utilizarea setului de antrenare și pentru testare;

Supplied testset: baza de date de test încărcată separat;

Cross-validation: împărțirea setului de antrenare în mai multe seturi succesive de antrenare / testare;

Percentage split: împărțirea setului de antrenare într-un set de antrenare și testare.

III. Evaluarea performanței de clasificare

Parametrii de evaluare

Acuratețe de clasificare

Procent clasificări incorecte

Precizie

Reamintire

F-measure

Matricea de confuzie

Weka Explorer

Classifier: ZeroR

Test options:
☐ Use training set
☐ Supplied test set
☒ Cross-validation Folds: 10
☐ Percentage split %: 66

Classifier output:

ZeroR predicts class value: Iris-setosa

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

Correctly Classified Instances	50	33.3333 %
Incorrectly Classified Instances	100	66.6667 %
Kappa statistic	0	
Mean absolute error	0.4444	
Root mean squared error	0.4714	
Relative absolute error	100 %	
Root relative squared error	100 %	
Total Number of Instances	150	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	1	0.333	1	0.5	0.5	Iris-setosa
	0	0	0	0	0	0.5	Iris-versicolor
	0	0	0	0	0	0.5	Iris-virginica
Weighted Avg.	0.333	0.333	0.111	0.333	0.167	0.5	

=== Confusion Matrix ===

a	b	c	-- classified as
50	0	0	a = Iris-setosa
50	0	0	b = Iris-versicolor
50	0	0	c = Iris-virginica

III. Evaluarea performanței de clasificare

Acuratețea de clasificare

Acuratețea se calculează ca raportul dintre numărul de date etichetate corect împărțit la numărul total de exemple.

$$\text{Acuratețea} = \frac{TP+TN}{TP+TN+FN+FP}$$

$$\begin{aligned}\text{Procentul de clasificări incorecte} &= \frac{FN+FP}{TP+TN+FN+FP} \\ &= 1 - \text{acuratețe}\end{aligned}$$

	Relevant	Nerelevant
Regăsit	True positive (TP)	False pozitiv (FP)
Neregăsit	False negative (FN)	True negative (TN)

III. Evaluarea performanței de clasificare

Acuratețea de clasificare

Dezavantaje in cazurile:

- Clase neechilibrate ca numar de date pentru fiecare (clase nebalansate);
- Costuri diferite de clasificare greșită pentru fiecare clasa în parte.

III. Evaluarea performanței de clasificare

Precizia și reamintirea

- **Precizia (precision):** procentul de documente returnate care sunt și relevante.
- **Reamintirea (recall):** procentul de documente relevante care sunt și regăsite.

	Relevant	Nerelevant
Regăsit	True positive (TP)	False positive (FP)
Neregăsit	False negative (FN)	True negative (TN)

- $\text{Precizia} = \text{tp} / (\text{tp} + \text{fp})$
- $\text{Reamintirea} = \text{tp} / (\text{tp} + \text{fn})$

III. Evaluarea performanței de clasificare

Precizia și reamintirea

- Precizia = $tp / (tp + fp)$
- Reamintirea = $tp / (tp + fn)$

	Relevant	Nerelevant
Regăsit	True positive (TP)	False pozitive (FP)
Neregăsit	False negative (FN)	True negative (TN)

Relevanță							
							
Rezultate							

- Calculați precizia și reamintirea pentru această situație.

III. Evaluarea performanței de clasificare

Precizia și reamintirea

■ Întrebări:

- 1) Ce este mai important? Să avem o precizie ridicată sau o reamintire ridicată?
- 2) Se poate să avem reamintire mare și precizie foarte mică?

III. Evaluarea performanței de clasificare

F-Measure

- F-measure combină precizia cu reamintirea (medie armonică ponderată):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- De obicei se selectează F_1 measure
 - $\beta = 1$ sau $\alpha = \frac{1}{2}$
 - Putem alege și alte variante?

III. Evaluarea performanței de clasificare

Matricea de confuzie

- Pentru k clase, este o matrice de $k \times k$
- Elementul de la poziția (i, j) arată numărul de date care fac parte din clasa i iar clasificatorul le etichetează cu clasa j .
- Tabelul următor ilustrează cazul particular pentru trei clase

		Clasa prezisă		
		Clasa 1	Clasa 2	Clasa 3
Clasa adevărată	Clasa 1	3	0	0
	Clasa 2	0	2	1
	Clasa 3	1	0	3

III. Evaluarea performanței de clasificare

Matricea de confuzie

		Clasa prezisă		
		Clasa 1	Clasa 2	Clasa 3
Clasa adevărată	Clasa 1	3	0	0
	Clasa 2	0	2	1
	Clasa 3	1	0	3

$$Acurate\text{ț}ea = \frac{8}{10} = 80\%$$

IV. Preprocesarea trăsăturilor

Studiu de caz

Detecția persoanelor care nu își plătesc creditul

Starea contului curent	Istoric credit	Țară	Vârstă	Salariu	Telefon
0 <= ... < 200 EUR	Fără istoric credite	Germania	22	1000	nu
>= 2000 EUR	Toate creditele plătite integral	Franța	38	500	nu
0 <= ... < 200 EUR	Întârzieri de plată		28		da
200 <= ... < 2000 EUR		Tailanda	55	100000	da
>= 2000 EUR	Fără istoric credite		55	1000	

IV. Preprocesarea trăsăturilor

Prelucrare date lipsă

Starea contului curent	Istoric credit	Țară	Age	Salariu	Telefon
$0 \leq \dots < 200$ EUR	Fără istoric credite	Germania	22	1000	nu
≥ 2000 EUR	Toate creditele plătite integral	Franța	38	500	nu
$0 \leq \dots < 200$ EUR	Întârzieri de plată	?	28	?	da
$200 \leq \dots < 2000$ EUR	?	Tailanda	55	100000	da
≥ 2000 EUR	Fără istoric credite	?	55	1000	?

IV. Preprocesarea trăsăturilor

Prelucrare date lipsă

Varianta 1: eliminare linii

Alternativa 2:

Utilizarea unor date prelucrate:

- Medie, valoare mediană, valoare minimă și maximă
- Cea mai utilizată valoare
- Adăugarea unei valori noi pentru date lipsă

Alternativa 3: Realizarea unui model care să estimeze valoarea lipsă

IV. Preprocesarea trăsăturilor

Valori nominale

Varianta 1: binarizarea valorilor



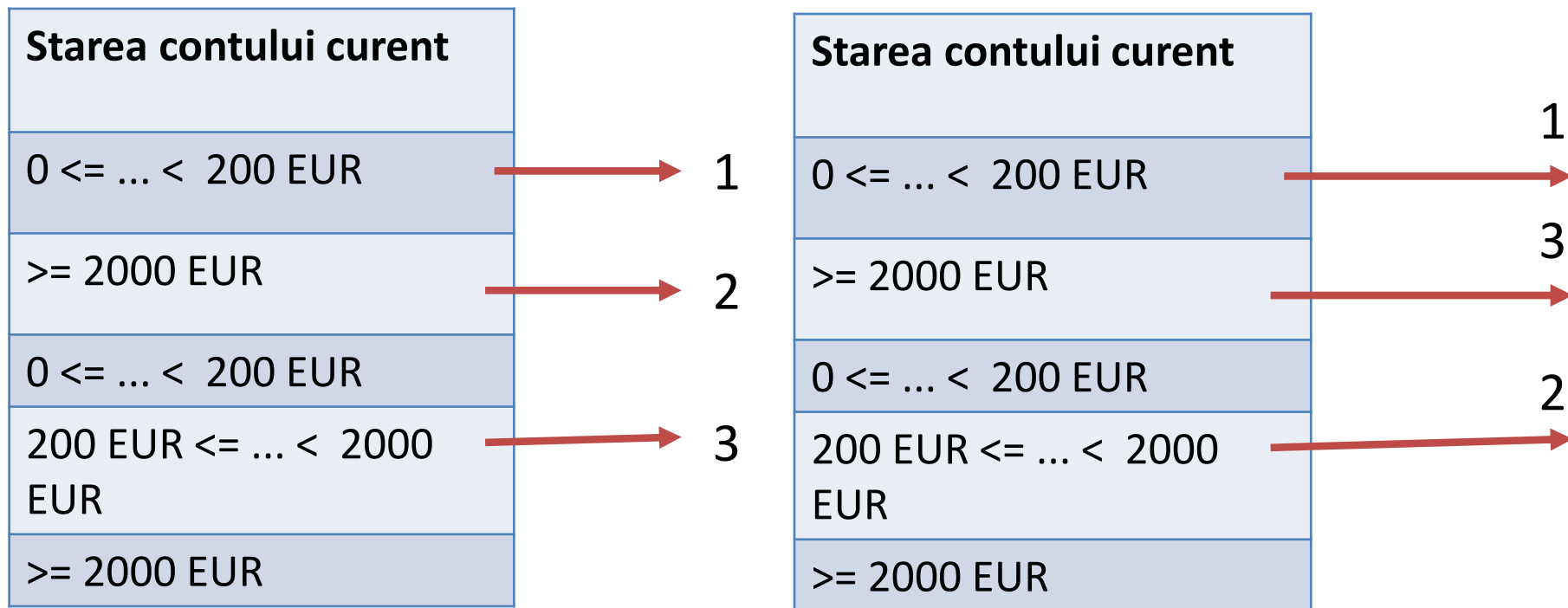
Telefon	Telefon
nu	0
nu	0
da	1
da	1
nu	0

IV. Preprocesarea trăsăturilor

Valori nominale

Varianta 2

Label Encoder: aplică valori între **0** și **nr_clase-1**.



IV. Preprocesarea trăsăturilor

Valori nominale

Varianta 3

Dummy Encoder: realizează **n** coloane pentru fiecare categorie.

În Weka această procesare se poate aplica utilizând opțiunea:

Filter -> filters -> unsupervised -> attribute -> NominalToBinary

Istoric credit	Fără istoric credite	Toate creditele plătite integral	Întârzieri de plată
Fără istoric credite	1	0	0
Toate creditele plătite integral	0	1	0
Întârzieri de plată	0	0	1
Toate creditele plătite integral	0	1	0
Fără istoric credite	1	0	0

IV. Preprocesarea trăsăturilor

Valori nominale

Dummy Encoder: În cazul în care sunt prea multe coloane, aceasta variantă nu este fezabilă deoarece numărul de trăsături va fi foarte ridicat

Țară	Germania	Franța	...	SUA
Germania	1	0		0
Franța	0	1		0
Africa de Sud	0	0		0
Tailanda	0	0		0
SUA	0	0		1

250 coloane

IV. Preprocesarea trăsăturilor

Valori nominale

Posibilă soluție: clusterizarea categoriilor

Țară	Regiune	Europa	Africa	Asia	America
Germania	Europa	1	0	0	0
Franța	Europa	1	0	0	0
Africa de Sud	America	0	1	0	0
Tailanda	Asia	0	0	1	0
SUA	America	0	0	0	1

IV. Preprocesarea trăsăturilor

Game de valori diferite

Vârstă	Salariu
22	1000
38	500
28	1000000
55	100000
55	1000



Vârstă	Salariu
0.22	0.001
0.38	0.0005
0.28	1
0.55	0.1
0.55	0.001

[18 - 80] [0 - 1000000]

Normalizare

$$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$

În Weka această procesare se poate aplica utilizând opțiunea:
Filter -> filters -> unsupervised -> attribute -> Normalize

V. Modele de Machine Learning

ZeroR

Cel mai simplu clasificator din Weka:

- Determină care este cea mai des întâlnită clasă
- În caz de regresie, va utiliza valoarea mediană
- Poate fi utilizată ca și limită inferioară a performanței pe care o poate primi un algoritm

V. Modele de Machine Learning

ZeroR

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



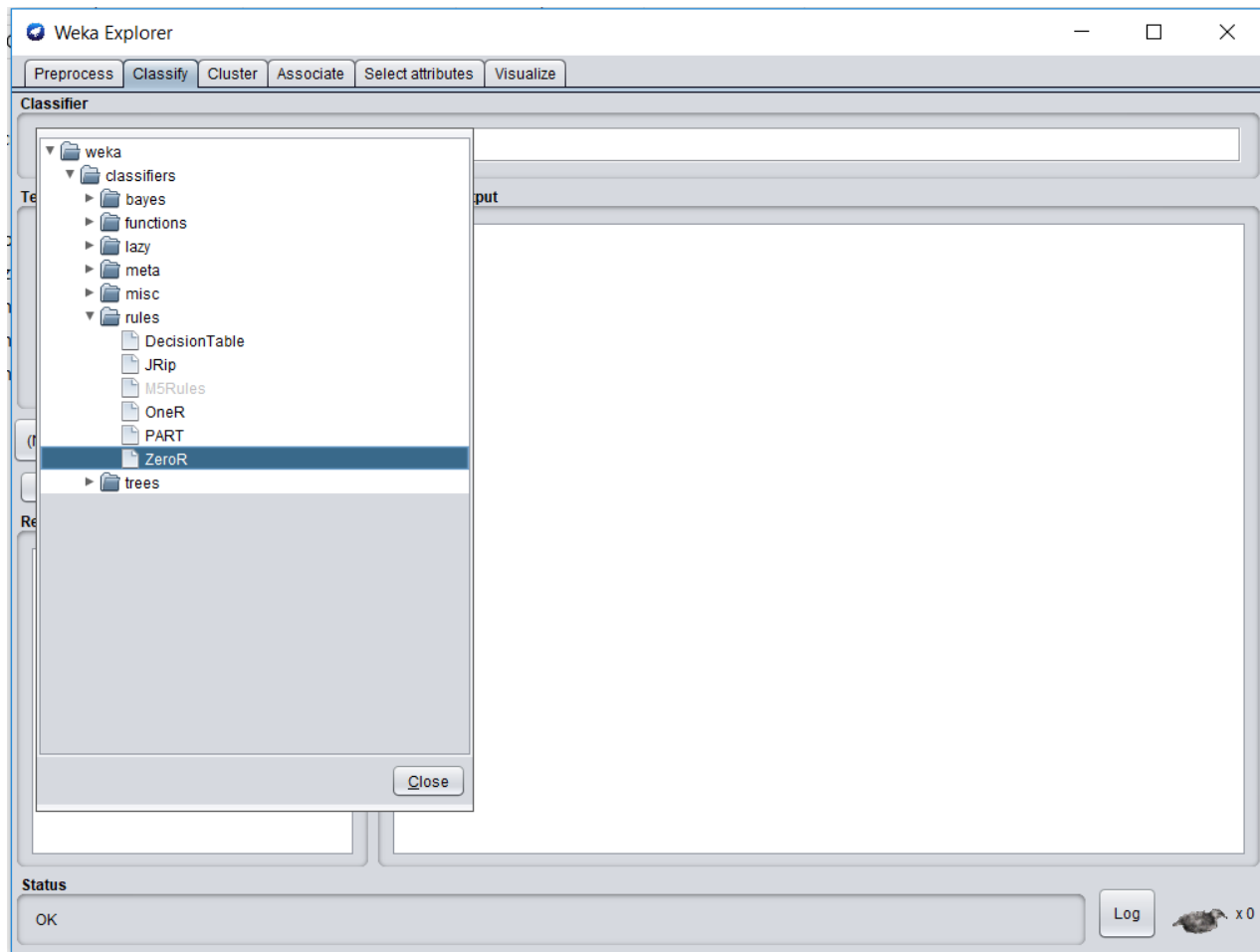
Play Golf	
Yes	No
9	5

Toate instanțele vor fi clasificate în clasa „Yes”

V. Modele de Machine Learning

ZeroR

În imagine se poate vizualiza opțiunea de aplicare a algoritmului ZeroR în Weka.



V. Modele de Machine Learning

OneR

OneR („One Rule”) reprezintă o metodă naivă de clasificare care generează o singură regulă utilizând o singură trăsătură.

Algoritmul găsește trăsătura care maximizează probabilitatea de clasificare.

V. Modele de Machine Learning

OneR

Which one is the best predictor ?

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Frequency Tables

★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

În final vom avea următoarea regulă generată:

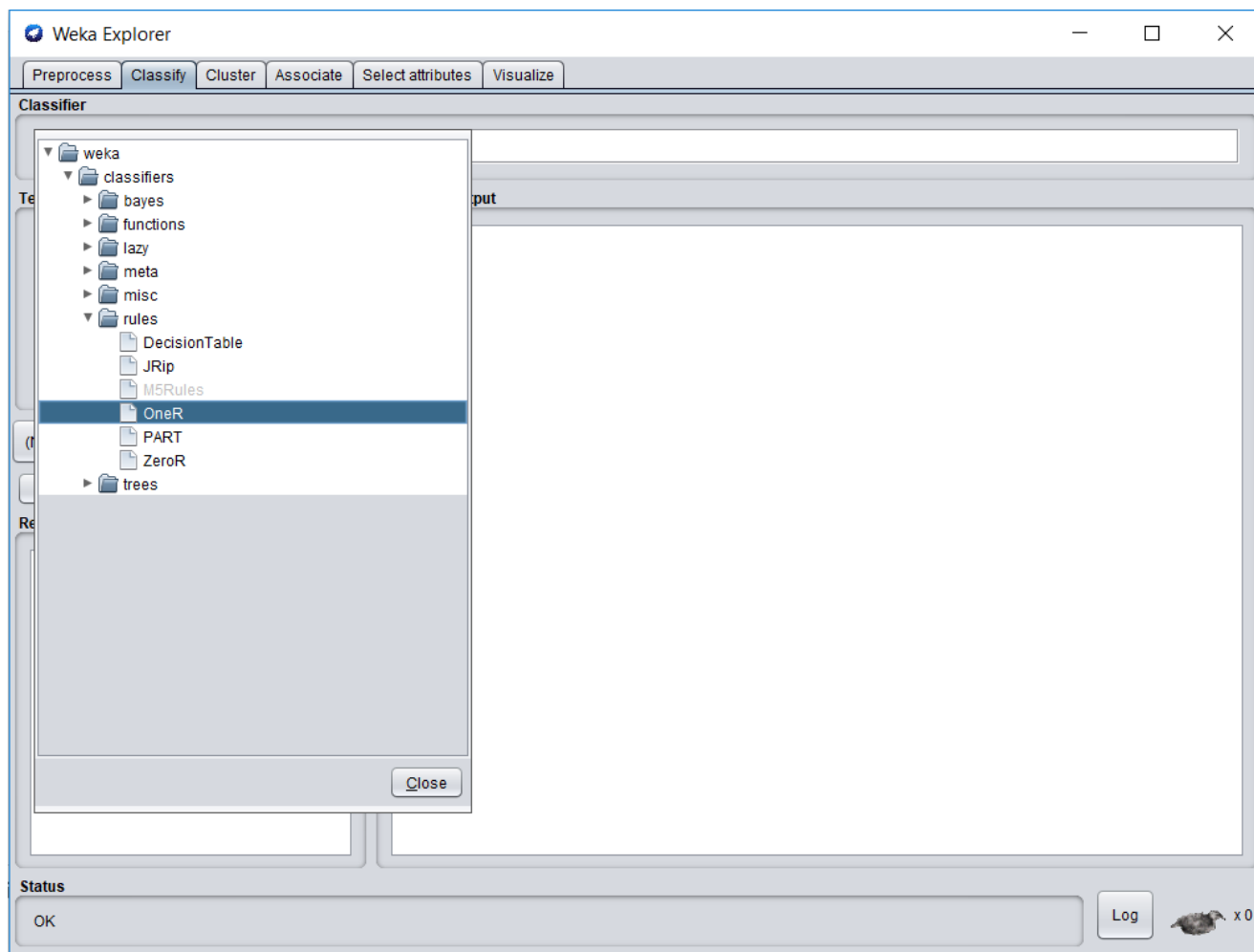
★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

IF Outlook = Sunny THEN PlayGolf = Yes
 IF Outlook = Overcast THEN PlayGolf = Yes
 IF Outlook = Rainy THEN PlayGolf = No

V. Modele de Machine Learning

OneR

În imagine se poate vizualiza opțiunea de aplicare a algoritmului OneR în Weka.

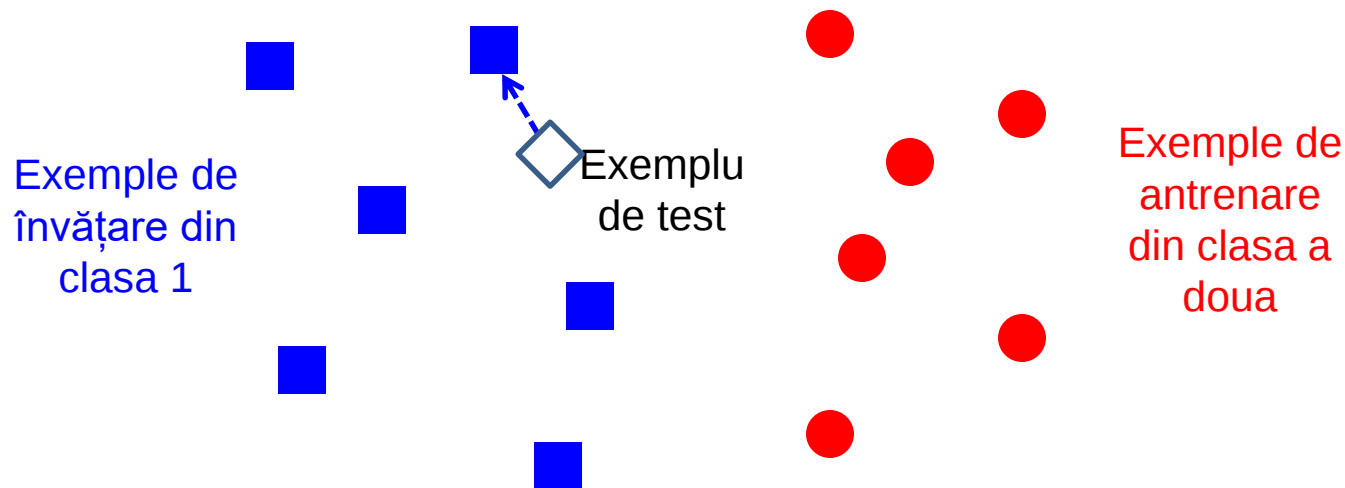


V. Modele de Machine Learning

K-Nearest Neighbor

(Cei mai apropiați vecini)

K=1



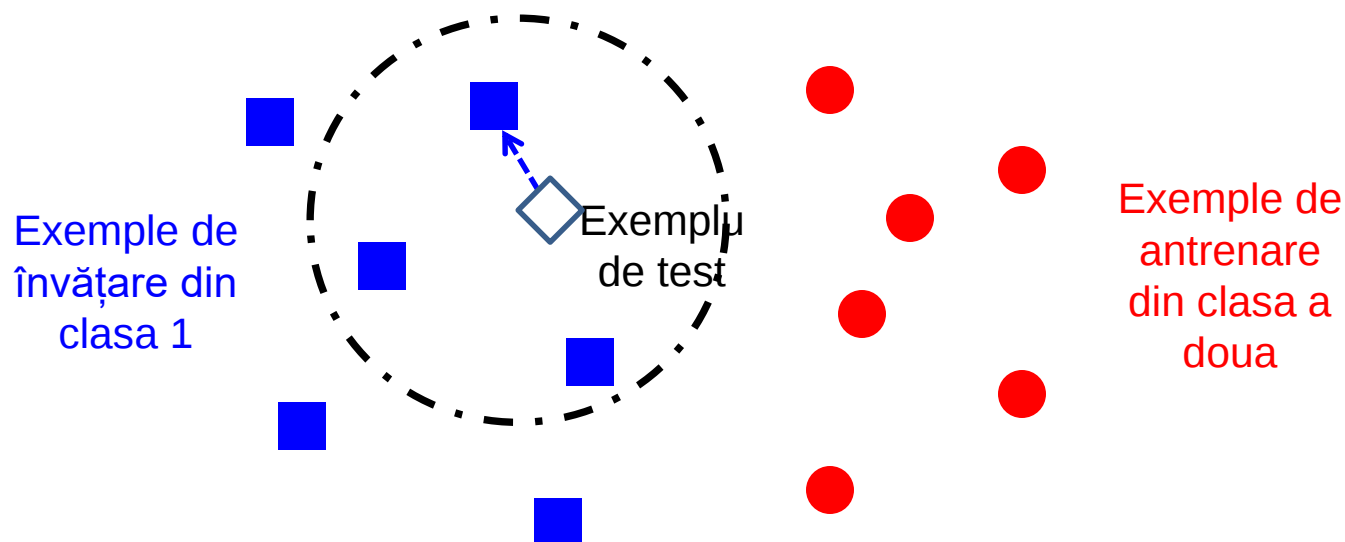
$f(\mathbf{x})$ = va lua clasa celui mai apropiat vecin a lui $\mathbf{x}$

V. Modele de Machine Learning

K-Nearest Neighbor

(Cei mai apropiați vecini)

$K=3$



V. Modele de Machine Learning

K-Nearest Neighbor

(Cei mai apropiați vecini)

Avantaje

- Simplu (ușor de implementat pentru o primă încercare);
- Nu are nevoie de multe date de învățare

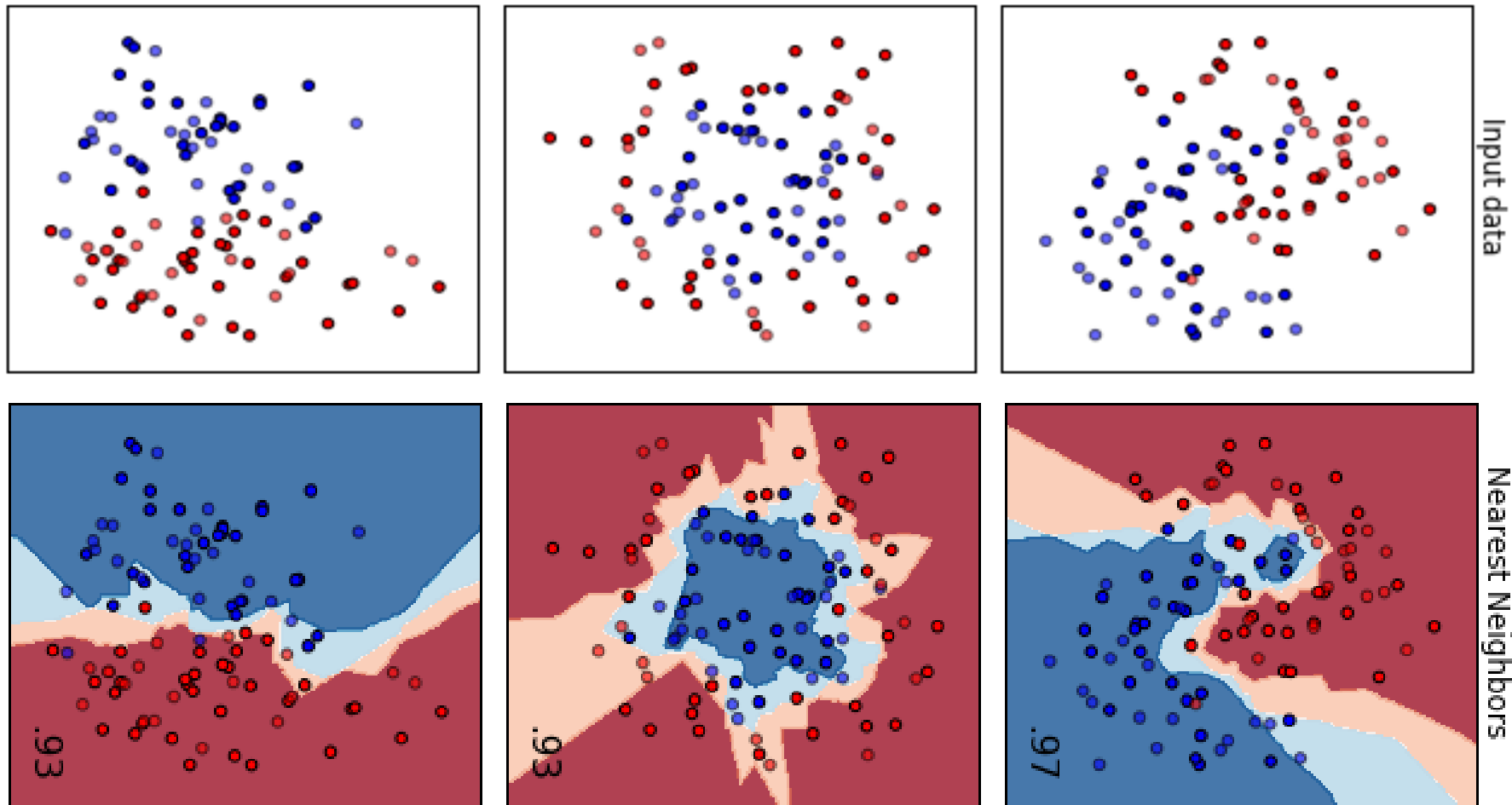
Dezavantaje

- Ocupă foarte multă memorie (se reține în memorie toată baza de antrenare);
- Lent – este nevoie să se compare elementul de clasificat cu toate trăsăturile din baza de date;
- Erori mari de clasificare (nu are rezultate bune pentru probleme complexe);
- Dependent de tipul de metrică utilizat.

V. Modele de Machine Learning

K-Nearest Neighbor

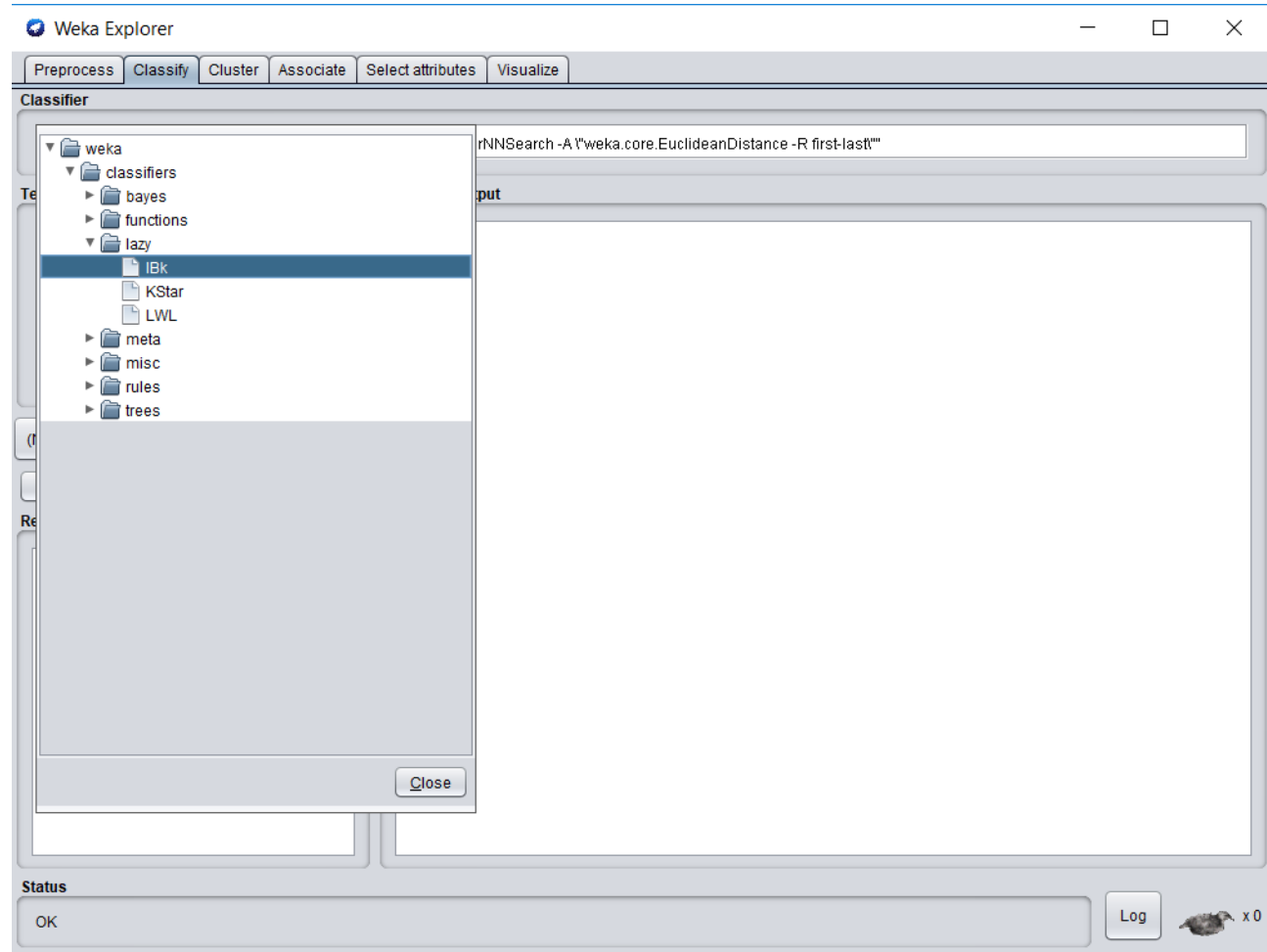
(Cei mai apropiați vecini)



V. Modele de Machine Learning

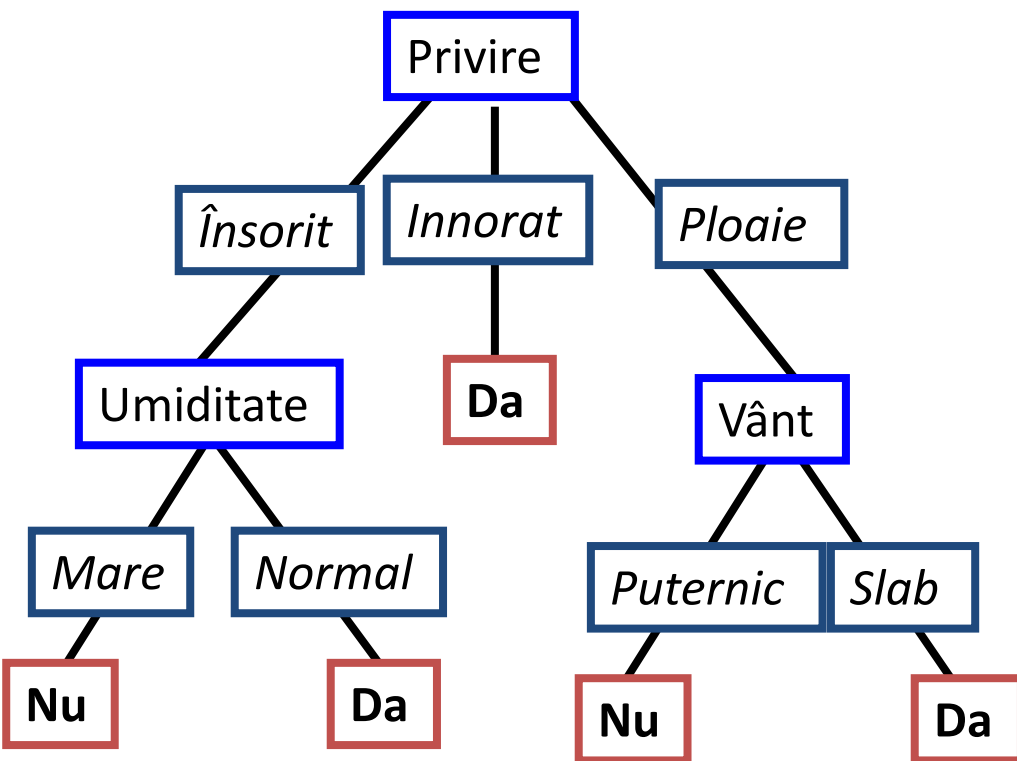
K-Nearest Neighbor

În imagine se poate vizualiza opțiunea de aplicare a algoritmului K-Nearest Neighbor în Weka.

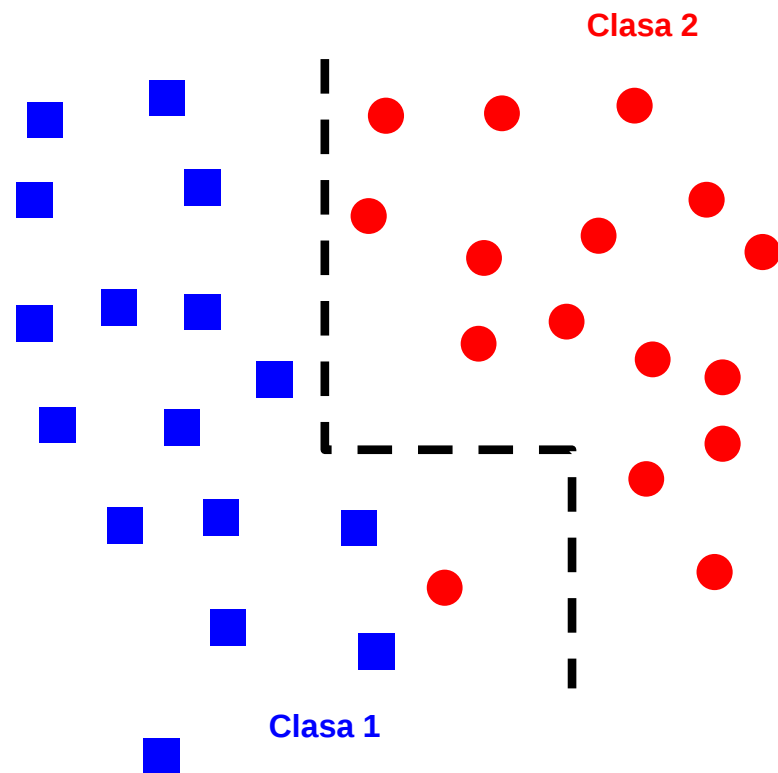


V. Modele de Machine Learning

Arbori de decizie

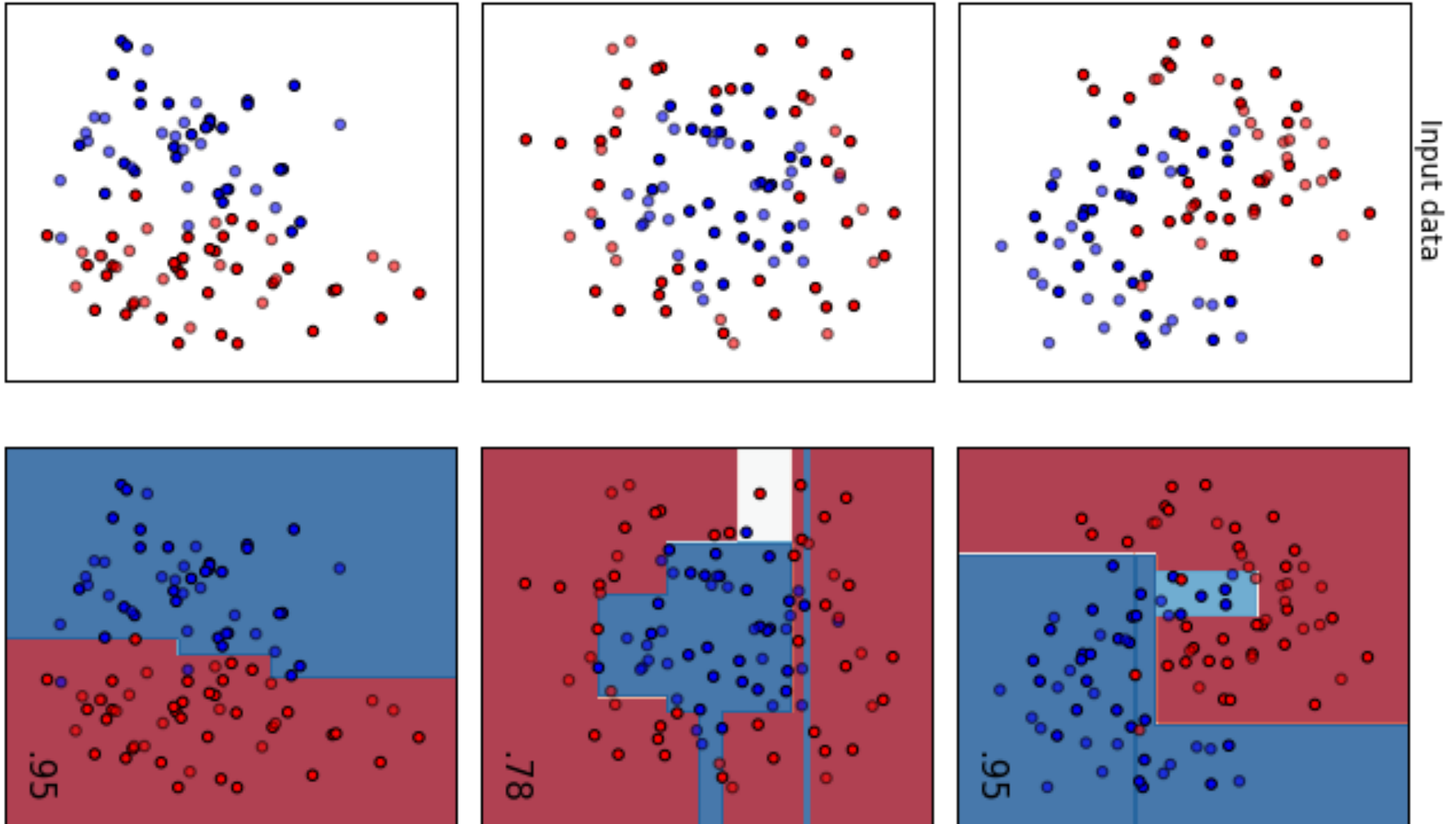


Decizia de a juca tenis



V. Modele de Machine Learning

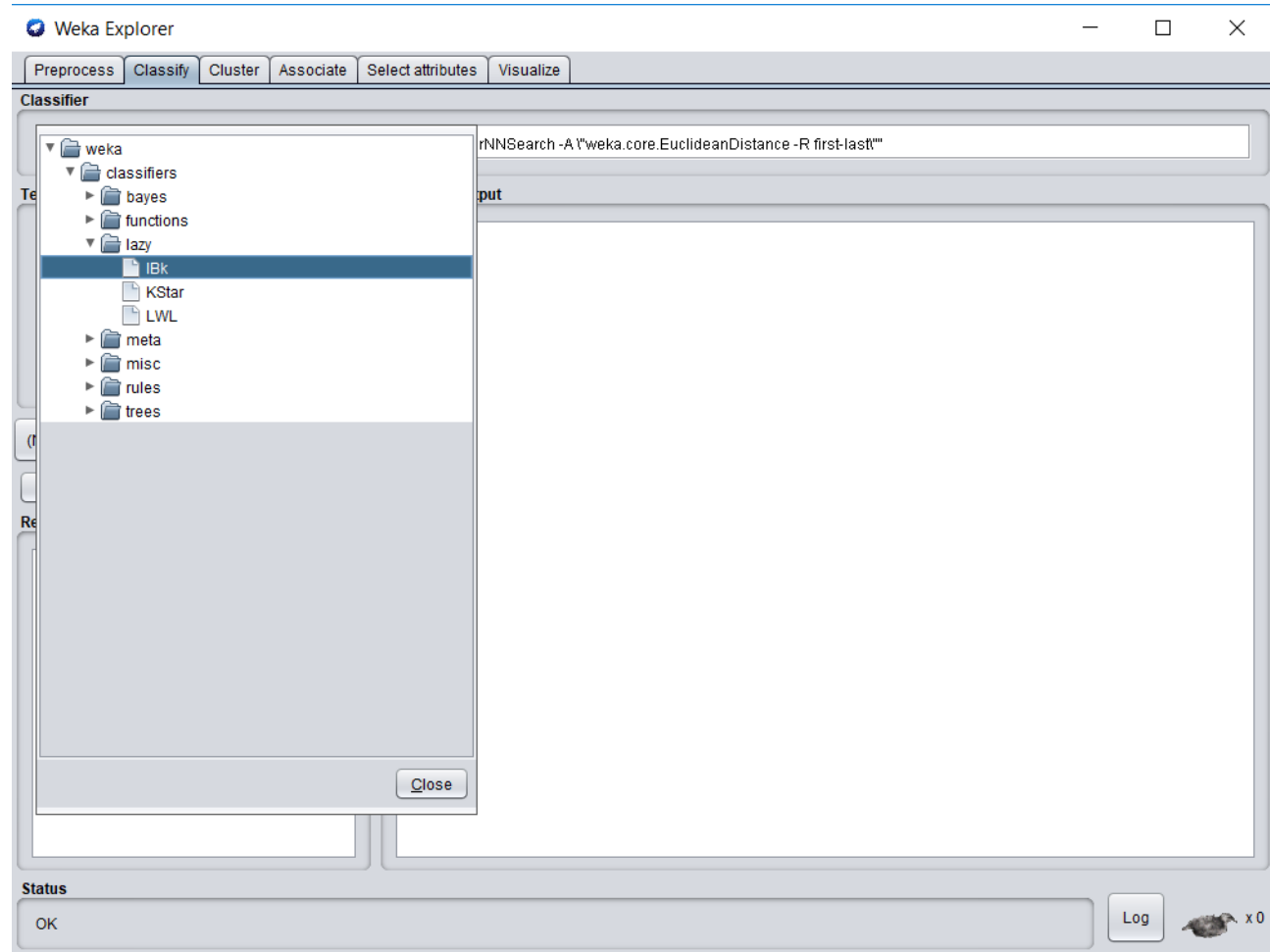
Arbori de decizie



V. Modele de Machine Learning

Arbori de decizie

În imagine se poate vizualiza opțiunea de aplicare a algoritmului K-Nearest Neighbor în Weka.



V. Modele de Machine Learning

Vizualizare arbore de decizie

Classifier

Choose **J48 -C 0.25 -M 2**

Test options

☐ Use training set

☐ Supplied test set **Set...**

☒ Cross-validation Folds **10**

☐ Percentage split % **66**

More options...

(Nom) class

Start **Stop**

Result list (right-click for options)

- 21:57:21 - bayes.NaiveBayes
- 21:57:30 - trees.J48

Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      144           96 %
Incorrectly Classified Instances     6            4 %
Kappa statistic                    0.94
Mean absolute error                  0.035
Root mean squared error              0.1586
Relative absolute error              7.8705 %
Root relative squared error          33.6353 %
Total Number of Instances           150

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Iris-setosa	0.980	0.000	1.000	0.980	0.990	0.985	0.990	0.987	Iris-setosa
Iris-versicolor	0.940	0.030	0.940	0.940	0.940	0.910	0.952	0.880	Iris-versicolor
Iris-virginica	0.960	0.030	0.941	0.960	0.950	0.925	0.961	0.905	Iris-virginica

Confusion Matrix ===

```
<-- classified as
a = Iris-setosa
b = Iris-versicolor
c = Iris-virginica
```

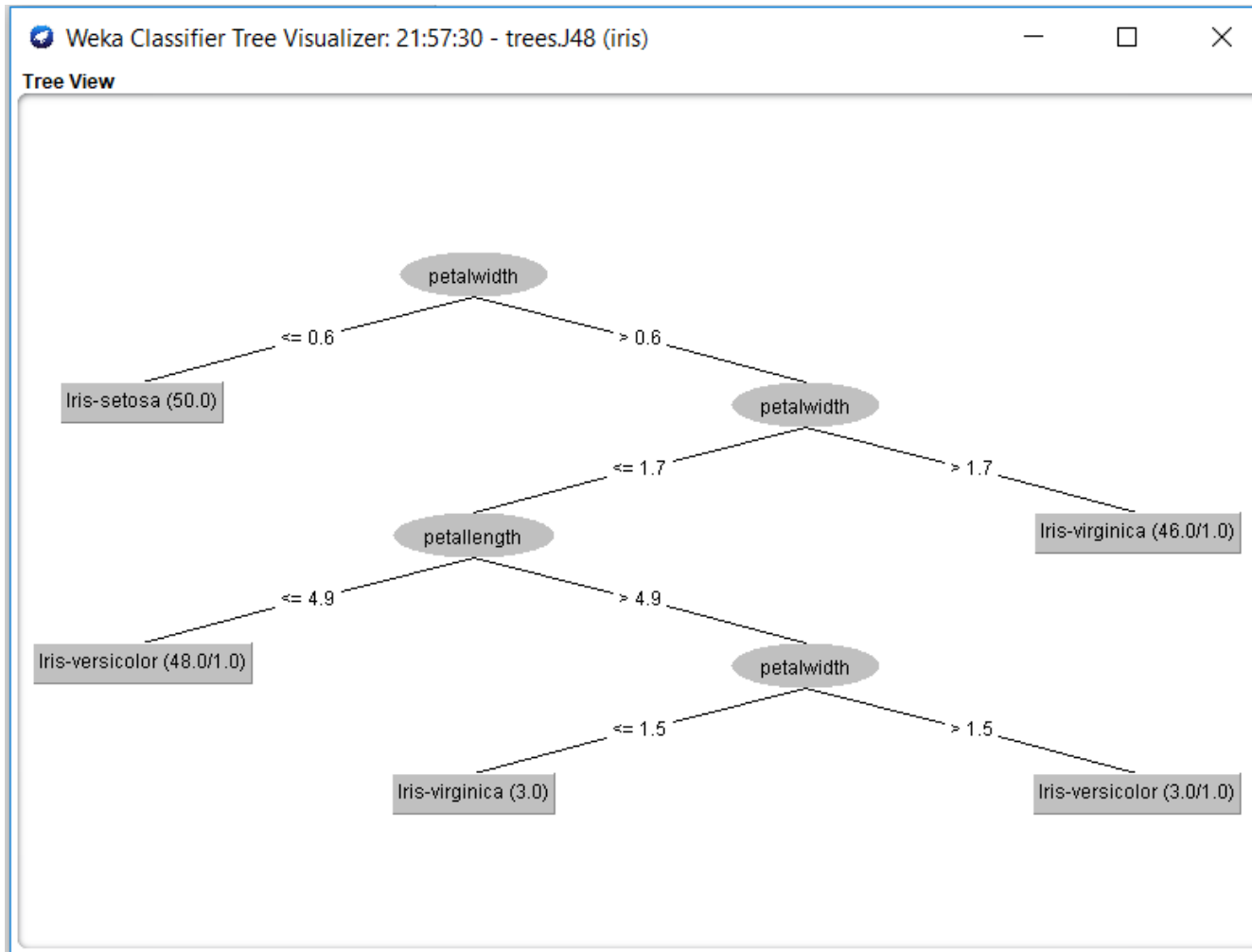
Visualize tree

Status

OK **Log** x 0

V. Modele de Machine Learning

Vizualizare arbore de decizie



V. Modele de Machine Learning

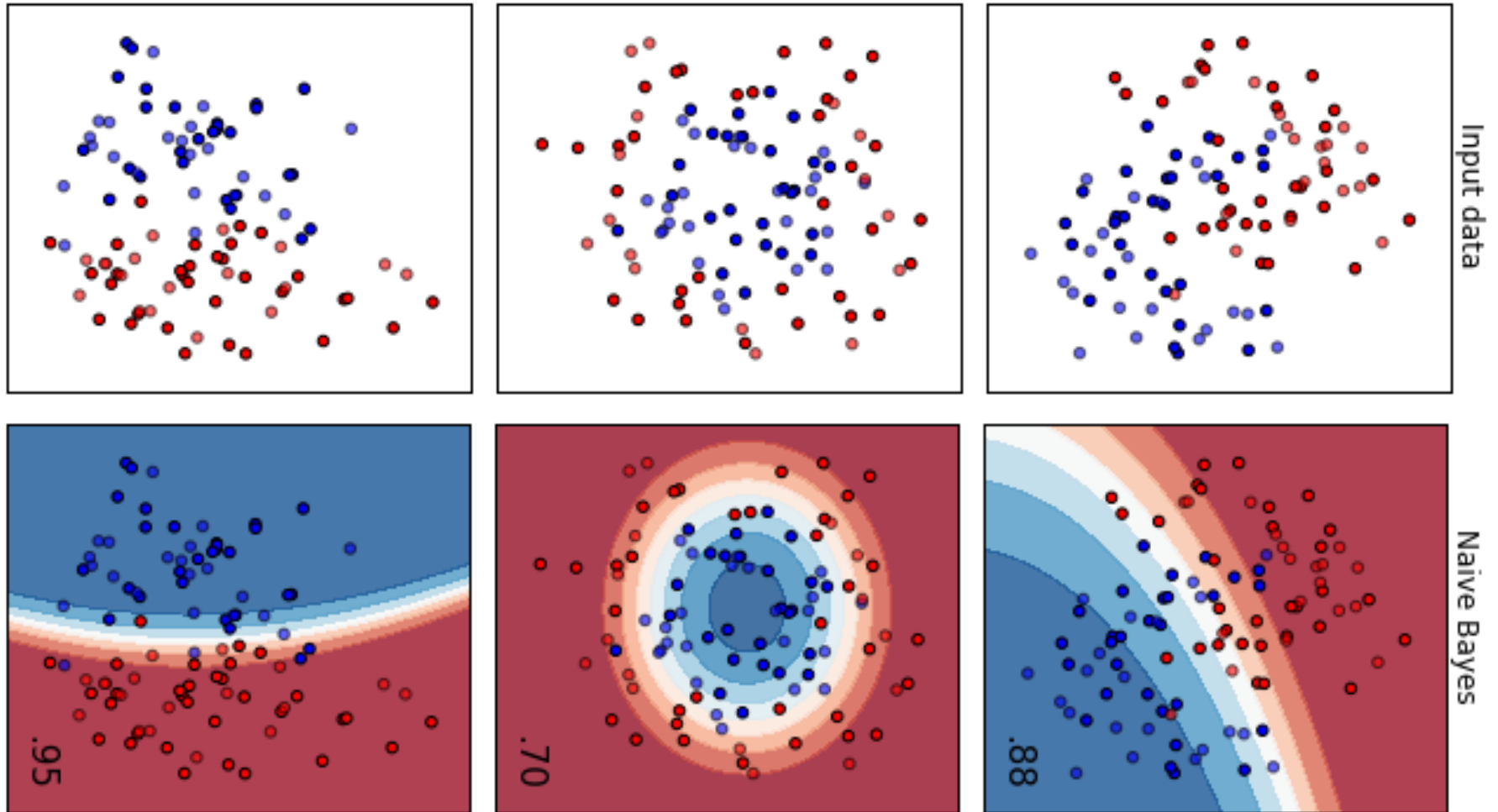
Naive Bayes

Clasificatorul Naive Bayes reprezintă un algoritm probabilistic bazat pe teorema lui Bayes, care pleacă de la premisa de independență între trăsături:

$$P(Y|X_1, \dots, X_n) = \frac{P(X_1, \dots, X_n|Y)P(Y)}{P(X_1, \dots, X_n)}$$

V. Modele de Machine Learning

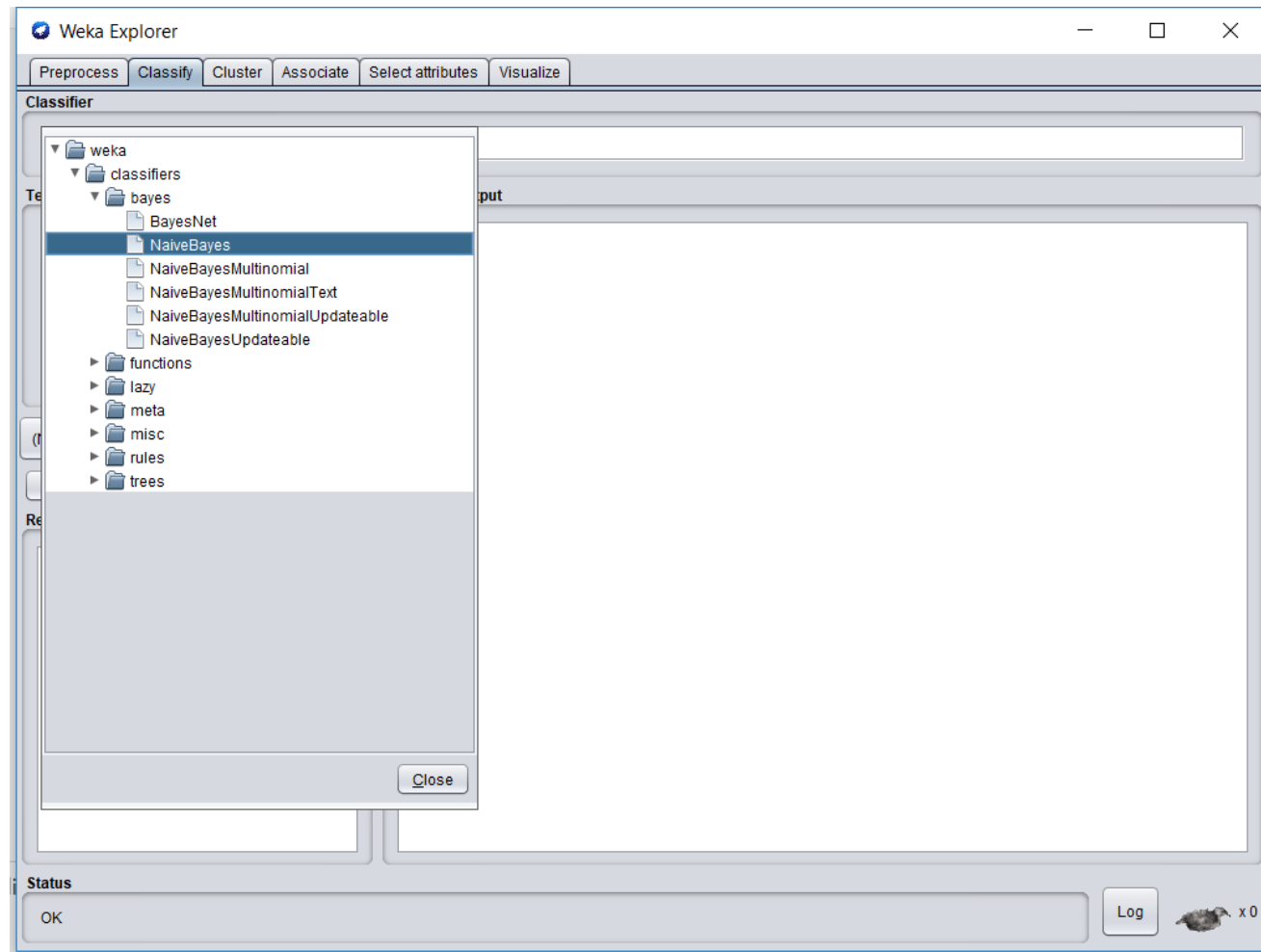
Naive Bayes



V. Modele de Machine Learning

Naive Bayes

În imagine se poate vizualiza opțiunea de aplicare a algoritmului Naive Bayes în Weka.



VI. Exerciții

- Încărcați una din bazele de date:
Weather.arff, *Iris.arff* și *Glass.arff*;
- Utilizați ca și clasificatori: ZeroR, OneR, arbori de decizie (J48), Nearest Neighbor (IBK) și Naive Bayes. Verificați care este cel mai bun clasificator:
 - Rețineți acuratețea de clasificare a fiecărui clasificator în parte;
 - Precizați care este cel mai bun clasificator pentru fiecare problemă în parte.
- Eliminați rând pe rând câte o coloană din bazele de date și comparați acuratețea cu cea obținută anterior;

VI. Exerciții

- Încărcați baza: *dataset_31_credit-g.arff*
- Aplicați posibili algoritmi de preprocesare (prelucrare valori nominale, normalizare)
- Utilizați ca și clasificatori: ZeroR, OneR, arbori de decizie (J48), Nearest Neighbor (IBK) și Naive Bayes. Verificați care este cel mai bun clasificator:
 - Rețineți acuratețea de clasificare a fiecărui clasificator în parte;
 - Precizați care este cel mai bun clasificator pentru fiecare clasă în parte.

Spor!