

Decision trees

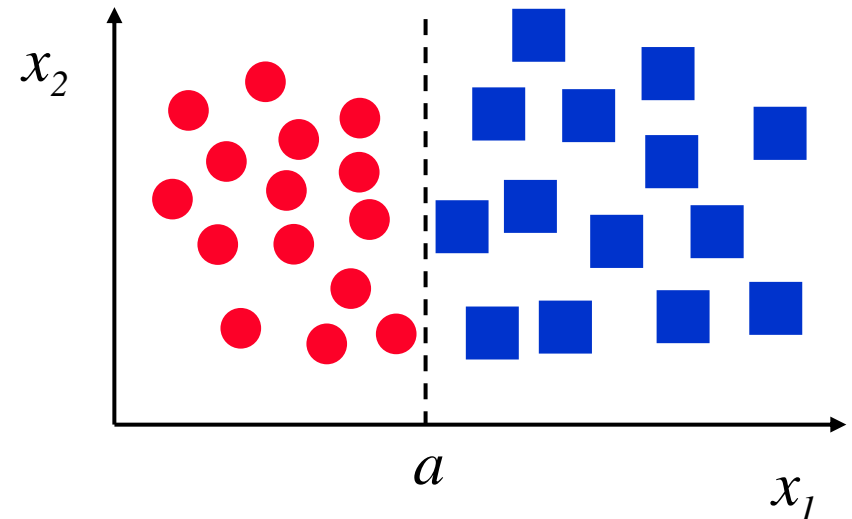


Separazione nello spazio delle istanze

☞ Ci sono situazioni facili da separare

■ per $x_1 > a$

● per $x_1 < a$



☞ Altre lo sono di meno e richiedono partizioni gerarchizzate

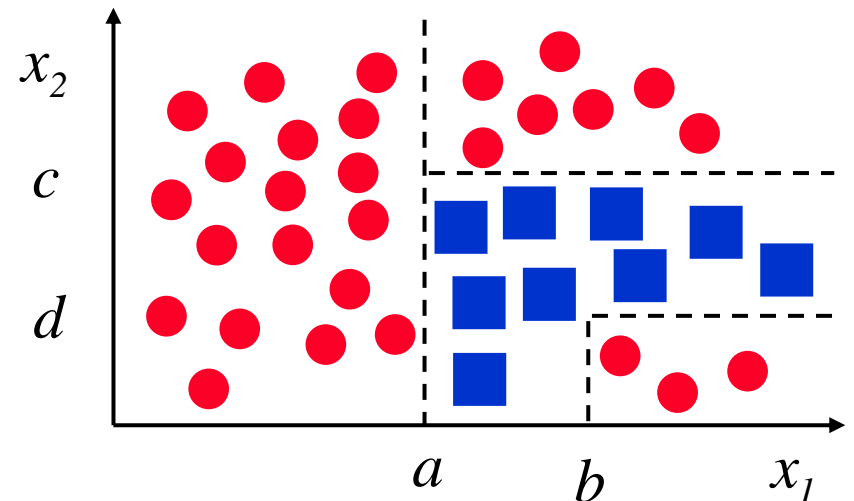
● $x_1 < a$

$x_1 > a$

$x_2 < c$

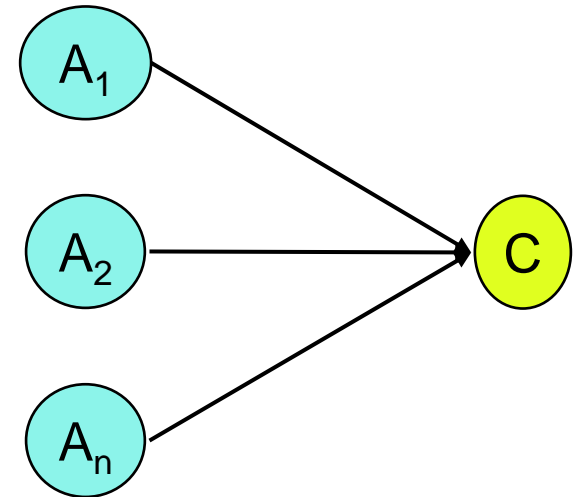
$x_1 < b$

■ $x_2 > d$



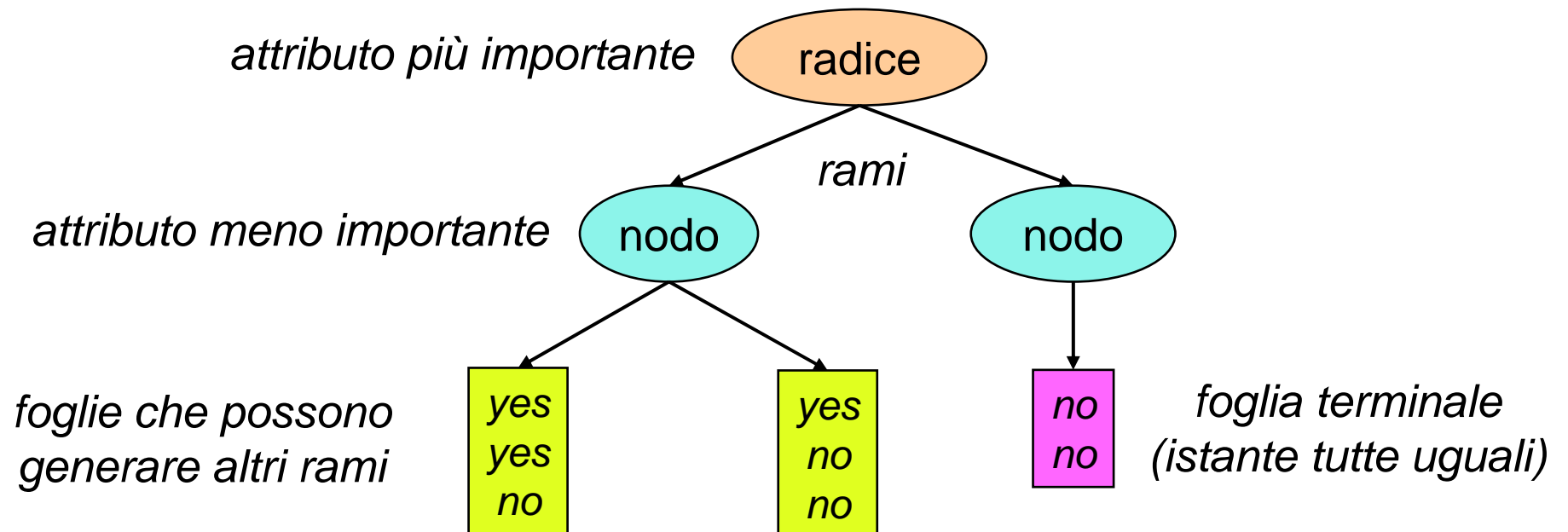
Alberi delle decisioni

- ☞ Come per lo stimatore bayesiano, si vuole stimare lo stato di un evento C sulla base di alcune condizioni antecedenti $A_1 \dots A_n$
- ☞ A differenza dello stimatore bayesiano però gli antecedenti vengono *gerarchizzati*
- ☞ Da un antecedente scelto come *radice* si elencano tutte le sue possibili *istanze*, che costituiscono i *rami* dell'albero
- ☞ Il processo si ripete ricorsivamente, usando solo le istanze che raggiungono quel ramo
- ☞ La costruzione finisce quando ogni ramo termina in una foglia con un'unica classificazione.



Parti dell'albero

- ☞ **Radice:** attributo iniziale dell'albero, da cui partono le relative istanze
- ☞ **Ramo:** implicazione verso altre istanze
- ☞ **Nodo:** Attributo da cui parto istanze diverse
- ☞ **Foglia:** Punto terminale dell'albero, dove tutte le istanze hanno una medesima classificazione

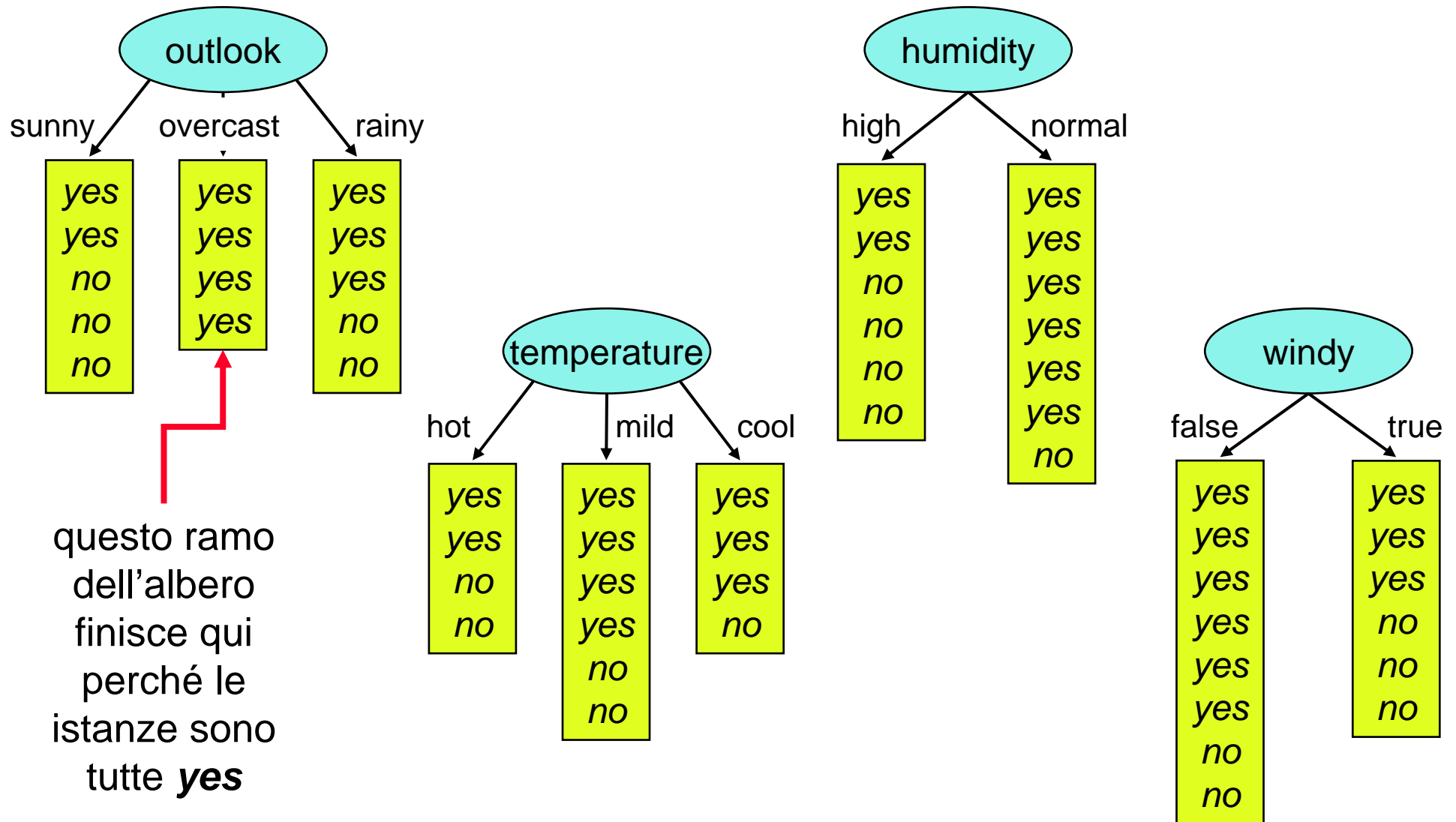


Esempio: the weather data

Proviamo a costruire l'albero partendo da ciascuno degli attributi

| instances | outlook | temp. | humidity | windy | play |
|-----------|----------|-------|----------|-------|------|
| 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 |

Risultato delle quattro radici



Come scegliere la radice migliore?

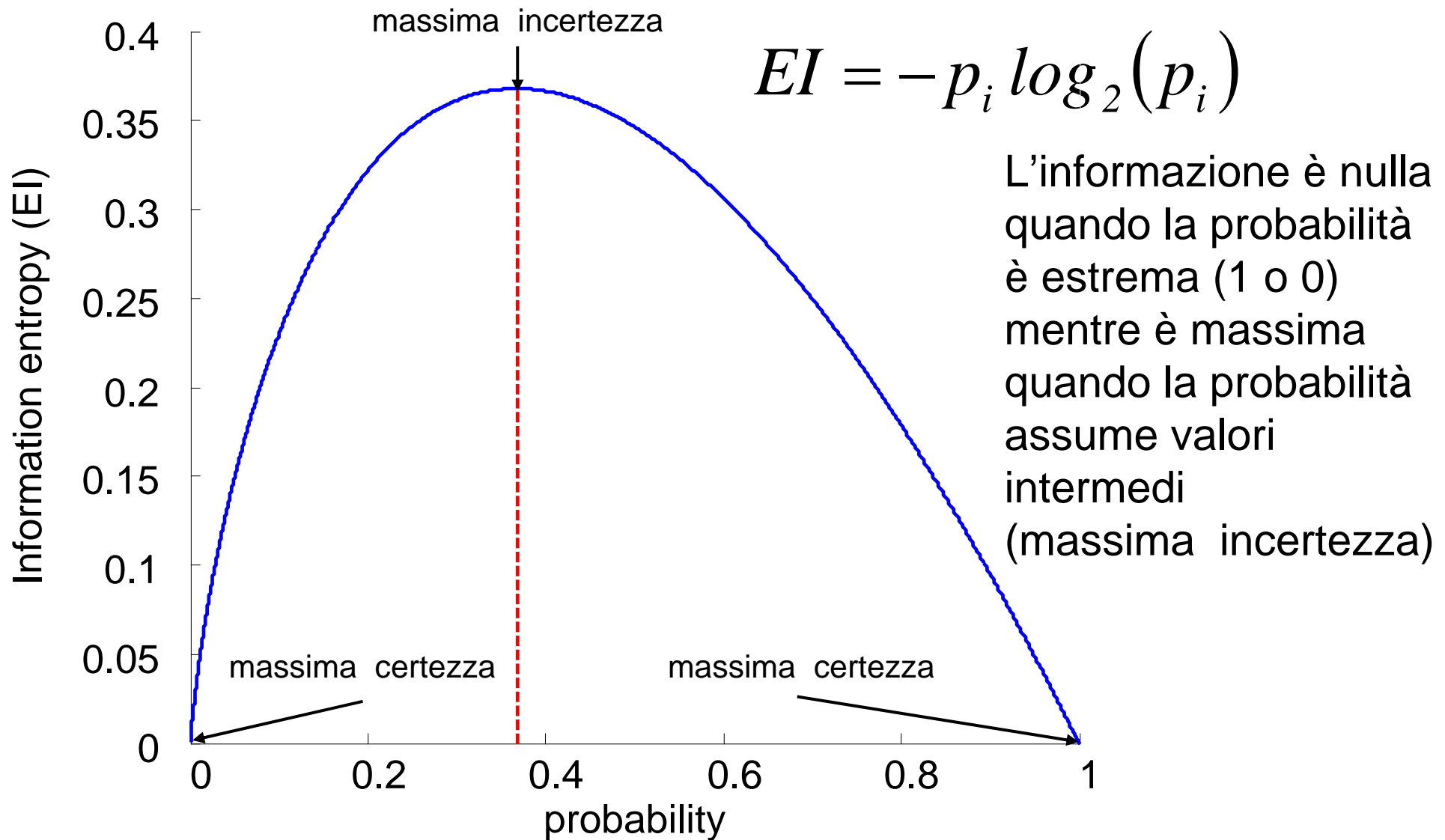
- ☞ Ricordiamo che l'albero finisce quando tutte le foglie sono terminali (istante tutte uguali)
- ☞ Dovremo scegliere l'attributo le cui foglie hanno la massima divisione
- ☞ L'*efficacia di divisione* si misura con l'*entropia di informazione (EI)*
- ☞ Sceglieremo l'attributo che massimizza l'entropia di informazione

$$EI = \sum_i -p_i \log_2(p_i)$$

dove le p_i rappresentano la probabilità (frequenza relativa) con cui ogni istanza in uscita dal nodo viene classificata in un dato attributo

- ☞ Nota: il segno $-$ è necessario per ottenere una EI positiva dato che ogni $p_i < 0$

Andamento dell'entropia di informazione

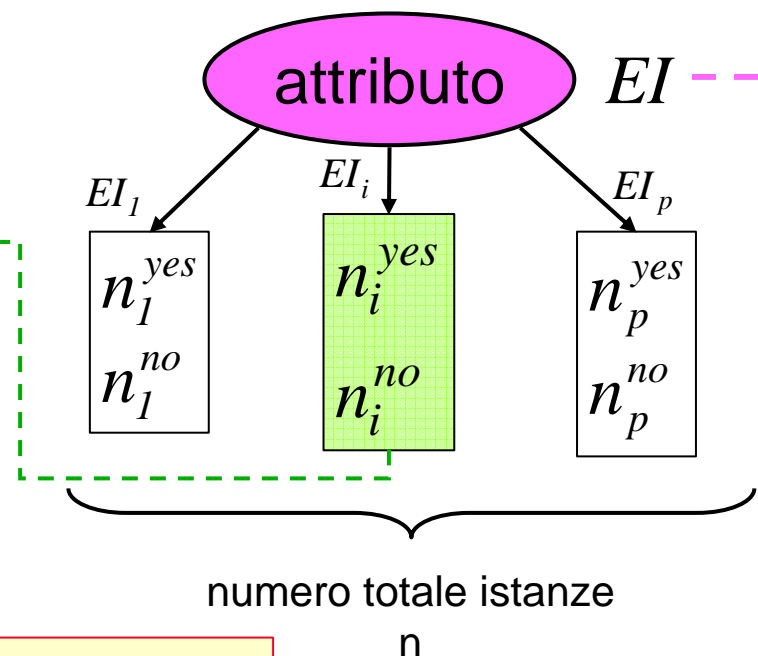


Calcolo dell'entropia di informazione

$$EI = \sum_i -p_i \log_2(p_i)$$

L'entropia del generico ramo i-esimo si calcola a partire dalle frequenze delle varie uscite, in questo caso yes e no.

$$EI_i = -\frac{n_i^{yes}}{n_i^{yes} + n_i^{no}} \log_2 \left(\frac{n_i^{yes}}{n_i^{yes} + n_i^{no}} \right) - \frac{n_i^{no}}{n_i^{yes} + n_i^{no}} \log_2 \left(\frac{n_i^{no}}{n_i^{yes} + n_i^{no}} \right) = -p_i^{yes} \log_2(p_i^{yes}) - p_i^{no} \log_2(p_i^{no})$$



L'entropia dell'attributo, da cui escono p rami, è la somma delle entropie dei vari rami pesata dal numero istanze di ciascun ramo

$$EI = \sum_{i=1}^p \frac{n_i^{yes} + n_i^{no}}{n} EI_i$$

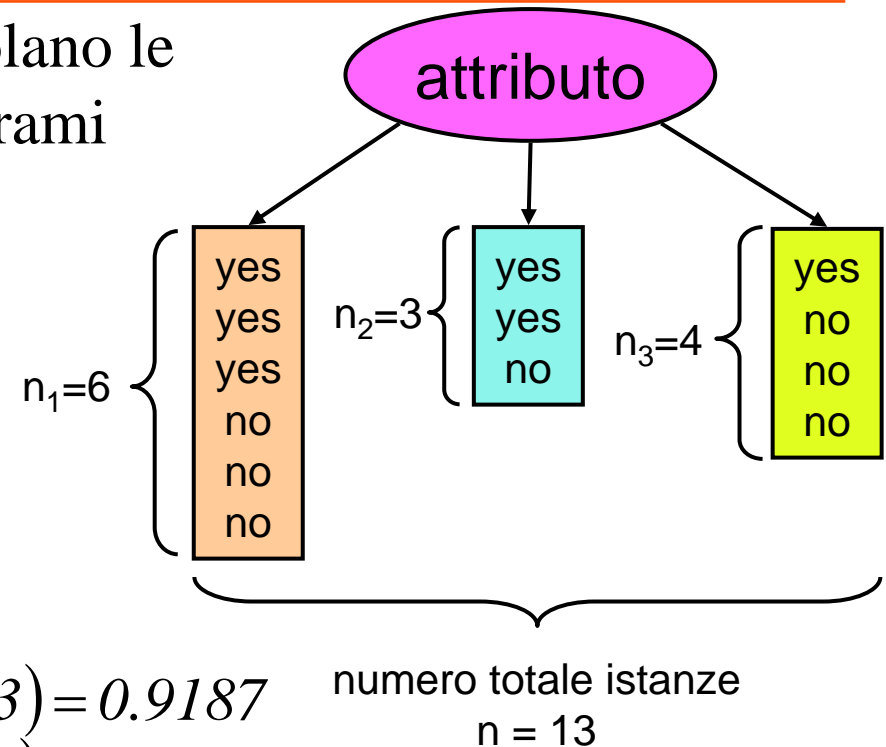
Calcolo di EI in uscita dalla radice

☞ Data una generica partizione si calcolano le frazioni di ciascun attributo nei vari rami

$$\begin{aligned}
 p_1^{yes} &= 3/6 = 0.5 & p_1^{no} &= 3/6 = 0.5 \\
 p_2^{yes} &= 2/3 = 0.6667 & p_2^{no} &= 1/3 = 0.3333 \\
 p_3^{yes} &= 1/4 = 0.25 & p_3^{no} &= 3/4 = 0.75
 \end{aligned}$$

☞ Da cui l'entropia di ciascun ramo

$$\begin{aligned}
 EI_1 &= -0.5 \times \log_2(0.5) - 0.5 \times \log_2(0.5) = 1 \\
 EI_2 &= -0.67 \times \log_2(0.67) - 0.33 \times \log_2(0.33) = 0.9187 \\
 EI_3 &= -0.25 \times \log_2(0.25) - 0.75 \times \log_2(0.75) = 0.8113
 \end{aligned}$$



☞ La EI in uscita è la somma pesata delle entropie dei vari rami

$$EI_1 = \frac{6}{13} EI_1 + \frac{3}{13} EI_2 + \frac{4}{13} EI_3 = \frac{6}{13} \times 1 + \frac{3}{13} \times 0.9187 + \frac{4}{13} \times 0.8113 = 0.9232$$

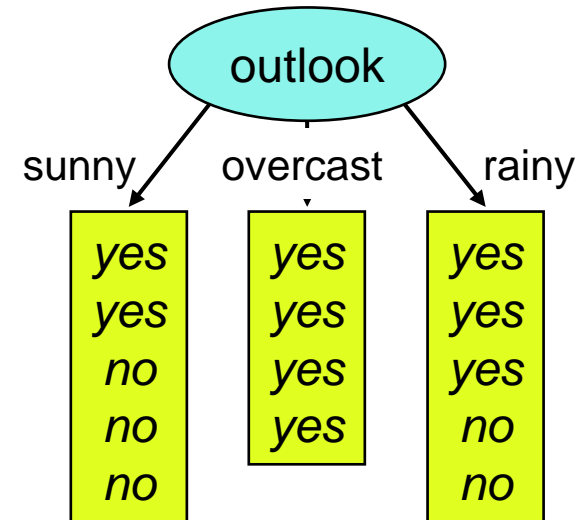
Calcolo di EI per gli *weather data*

☞ Consideriamo l'attributo *outlook* e calcoliamo la proporzione fra *yes* e *no* in ciascun ramo e le relative probabilità come frequenze relative, ottenendo:

sunny [2,3] = [2/5, 3/5] = [0.4, 0.6]

overcast [4,0] = [4/4, 0/4] = [1, 0]

rainy [3,2] = [3/5, 2/5] = [0.6, 0.4]



☞ Da queste si calcola la *EI* per ciascun ramo

sunny [0.4, 0.6] $\Rightarrow -0.4 \times \log_2(0.4) - 0.6 \times \log_2(0.6) = 0.9710$

overcast [1, 0] $\Rightarrow -1 \times \log_2(1) - 0 \times \log_2(0) = 0$

rainy [0.6, 0.4] $\Rightarrow -0.6 \times \log_2(0.6) - 0.4 \times \log_2(0.4) = 0.9710$

☞ La media delle *EI* in uscita dalla radice “outlook” è

$$EI = \frac{5}{14} \times 0.9710 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.9710 = 0.6936$$

Guadagno di EI

- ☞ Per prima cosa si calcola l'entropia di informazione generale, considerando la quantità totale di *yes* e *no* **non strutturati** (prima di qualsiasi partizione)

$$EI_{tot} \left(n_{tot}^{yes}, n_{tot}^{no} \right) = -p_{tot}^{yes} \log_2 \left(p_{tot}^{yes} \right) - p_{tot}^{no} \log_2 \left(p_{tot}^{no} \right)$$

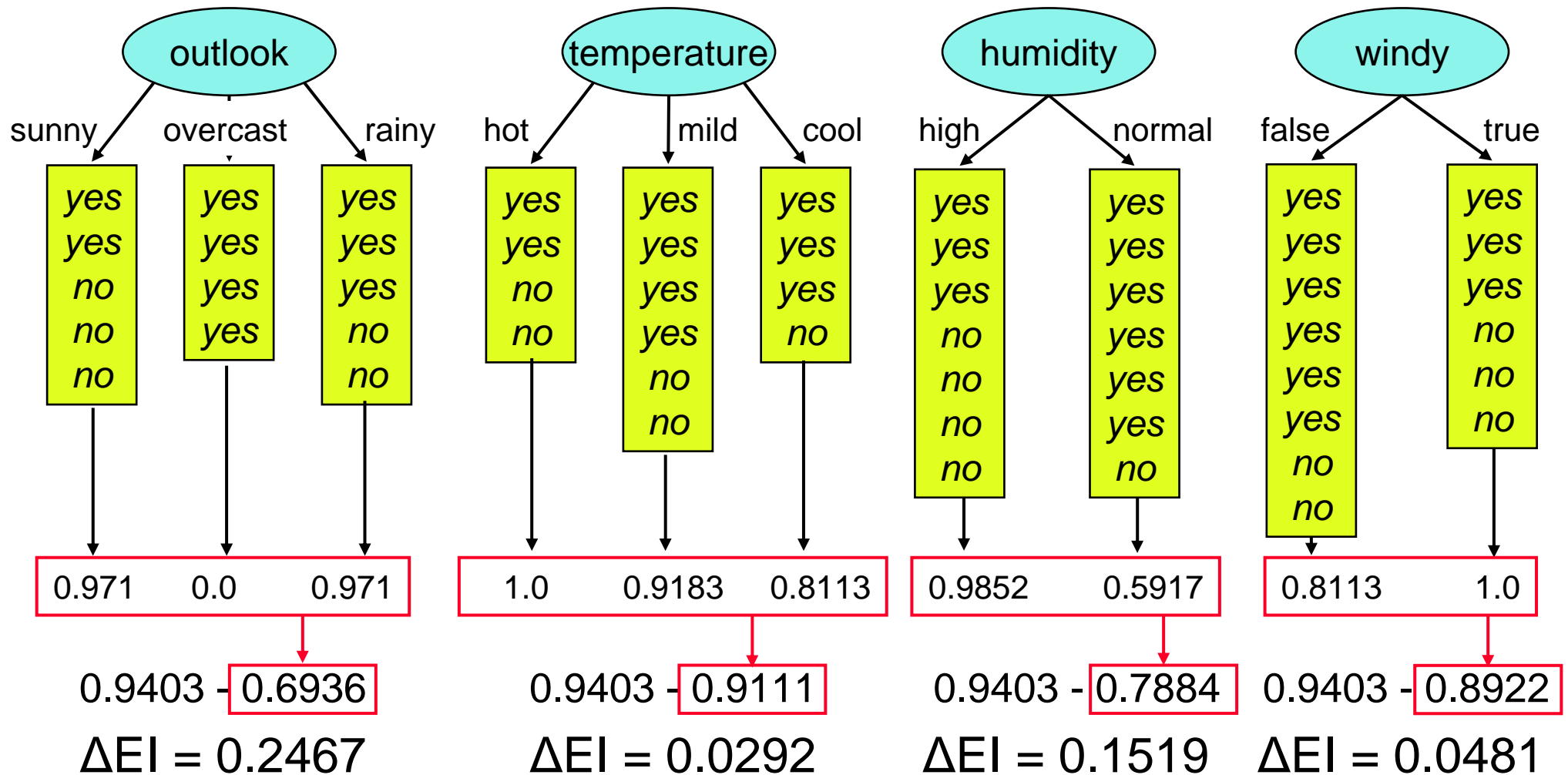
- ☞ Il guadagno si calcola come differenza fra EI_{tot} e l' EI generata da ogni nodo tenendo conto di tutti i possibili attributi in uscita

- ☞ Si sceglierà l'attributo che massimizza $\Delta EI(\text{attrib})$

$$\Delta EI(\text{attrib}) = EI_{tot} - EI_{nodo}$$

ΔEI dei 4 attributi assunti come radice

$$EI_{tot}(9,5) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.9403$$



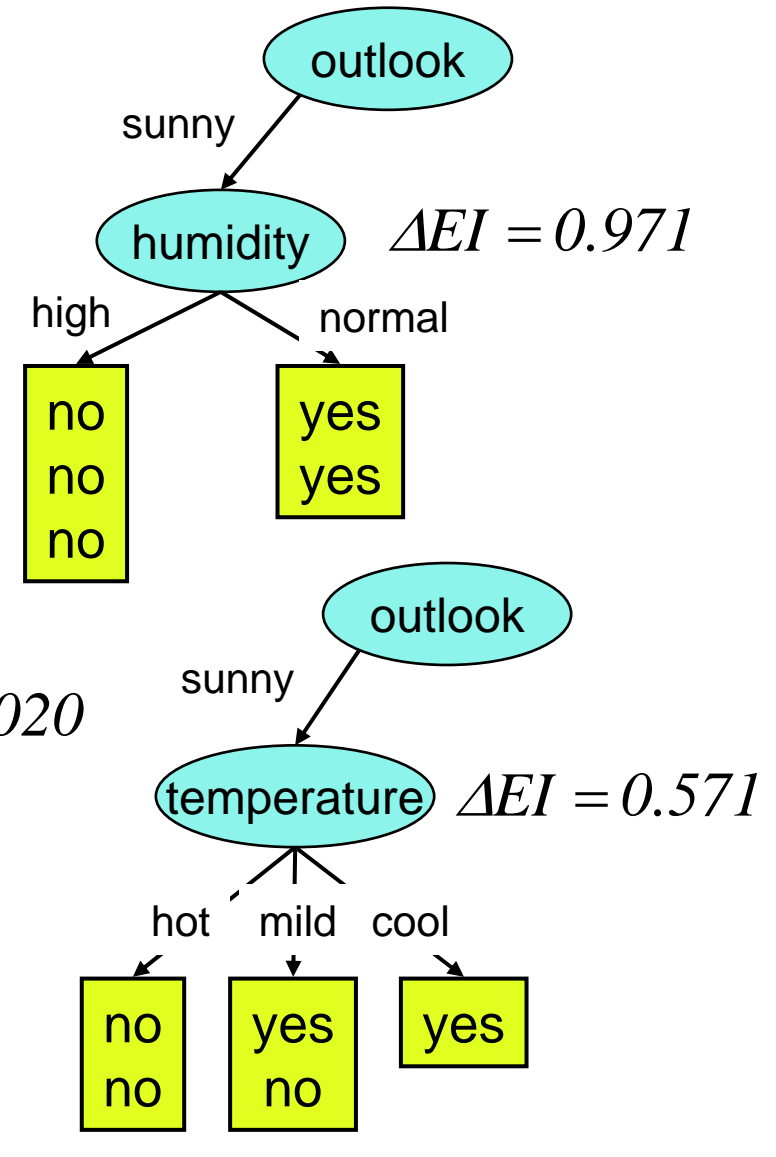
Prima divisione

Ordiniamo i 4 guadagni di partizione:

Outlook: $\Delta EI = 0.2467$
 Humidity: $\Delta EI = 0.1519$
 Windy: $\Delta EI = 0.0481$
 Temperature: $\Delta EI = 0.0292$

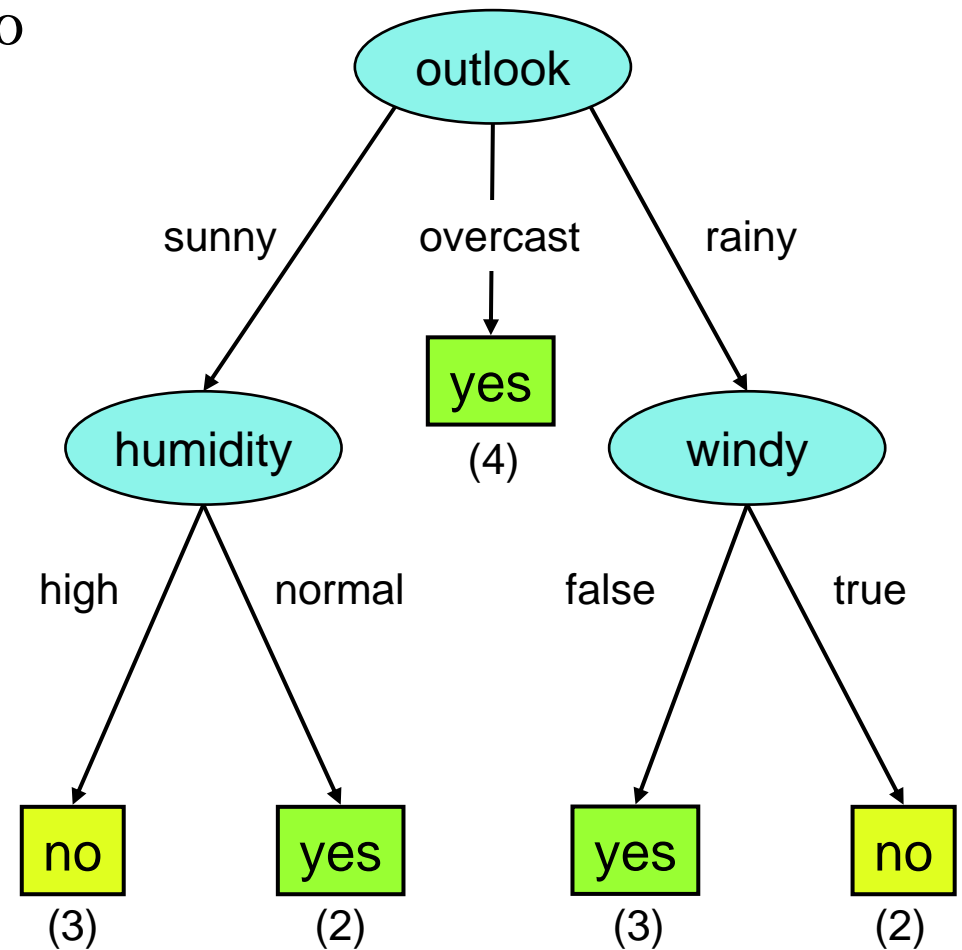
Continuando ricorsivamente, esaminiamo ora i guadagni che si ottengono dalle ramificazioni del ramo *sunny*

La seconda migliore partizione è *humidity*



Albero finale

- ☞ Continuando ricorsivamente si opera una divisione sull'attributo che fornisce il più alto guadagno
- ☞ Si procede ricorsivamente con il calcolo dei guadagni di partizione degli attributi rimanenti fino ad avere foglie terminali
- ☞ I numeri sotto ciascuna foglia terminale indica quante istanze di quel tipo sono presenti nel training set.



L'ambiente Data Mining WEKA

- ☞ **WEKA** è l'acronimo di "**W**aikato **E**nvironment for **K**nowledge **A**nalysis".
- ☞ È un software sviluppato nell'università di Waikato in Nuova Zelanda, è open source e viene rilasciato con licenza GNU.
- ☞ WEKA è anche il nome di un uccello privo di volo, simile al Kiwi, presente solo nelle isole della Nuova Zelanda, definito *inquisitive* (*curioso*) dagli autori.
- ☞ E' un ambiente di Data Mining, in grado di estrarre informazioni da un insieme di dati più o meno strutturati, senza alcuna informazione a priori.



Struttura di WEKA

☞ WEKA è un ambiente software interamente scritto in Java. Applica dei metodi di apprendimento automatici (*learning methods*) ad un set di dati (*dataset*) estraendo l'eventuale informazione in essi contenuta. È possibile attraverso questi metodi, avere quindi una previsione di comportamenti futuri.

☞ L'interfaccia di Weka è composta da:

Explorer: ambiente che consente di esplorare i dati

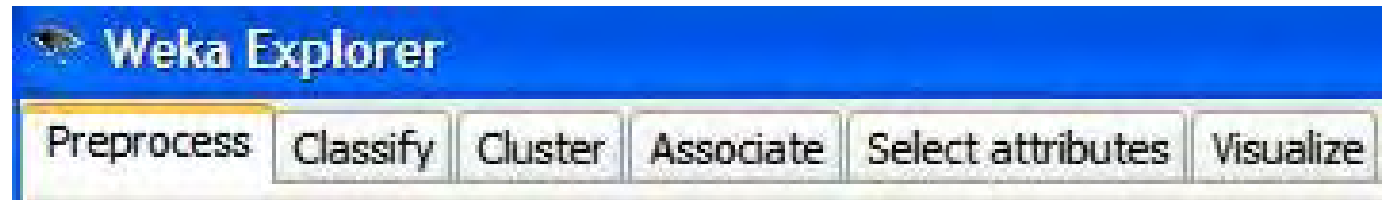
Experimenter: compie test statistici fra i diversi algoritmi di data mining


Knowledge Flow: rappresenta graficamente il flusso di informazione nei dati

Simple CLI: l'interfaccia dalla linea di comando.



L'ambiente Explorer

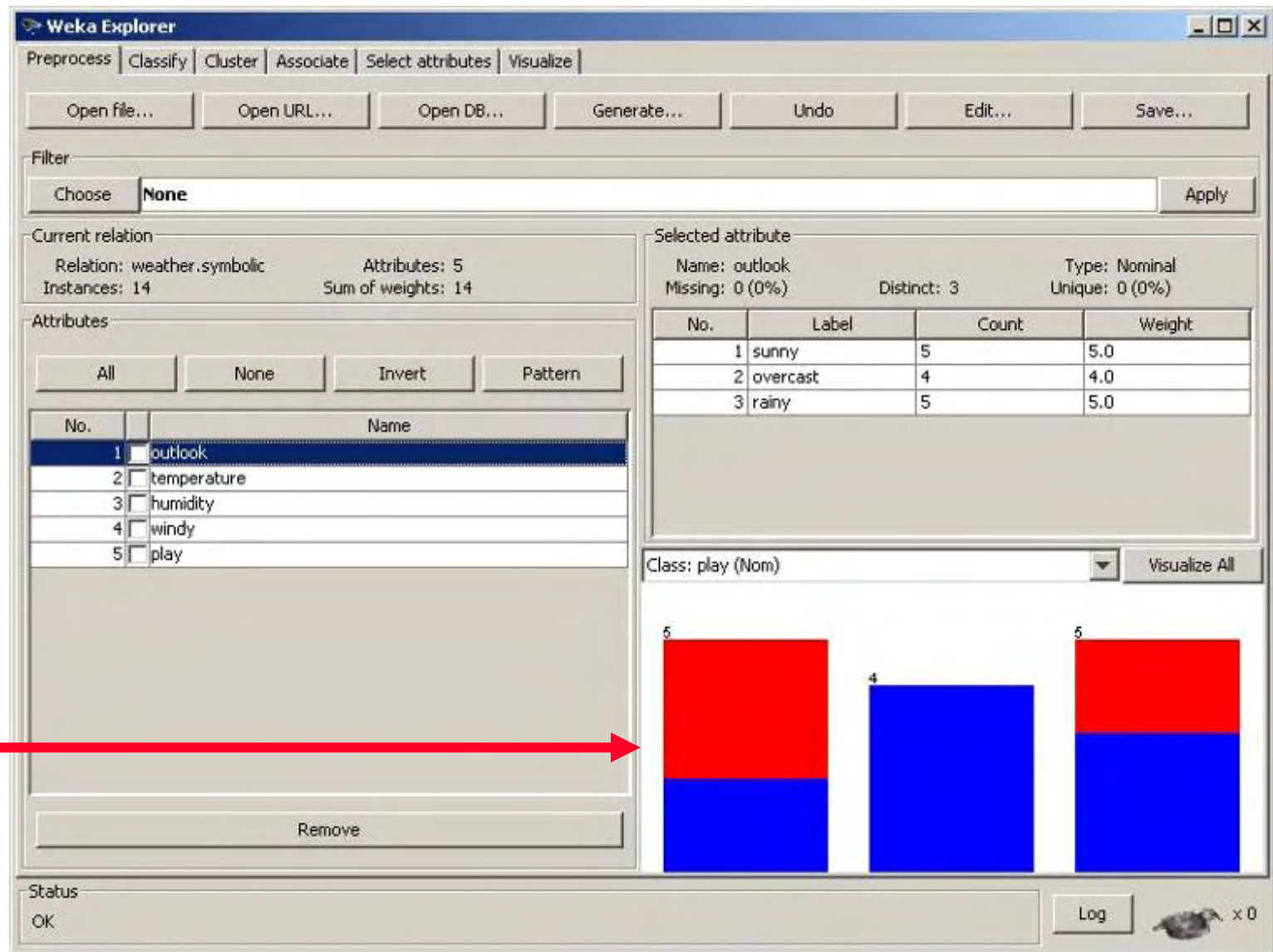


 Questo ambiente consente di esaminare i dati a disposizione e di applicare molti algoritmi di data mining

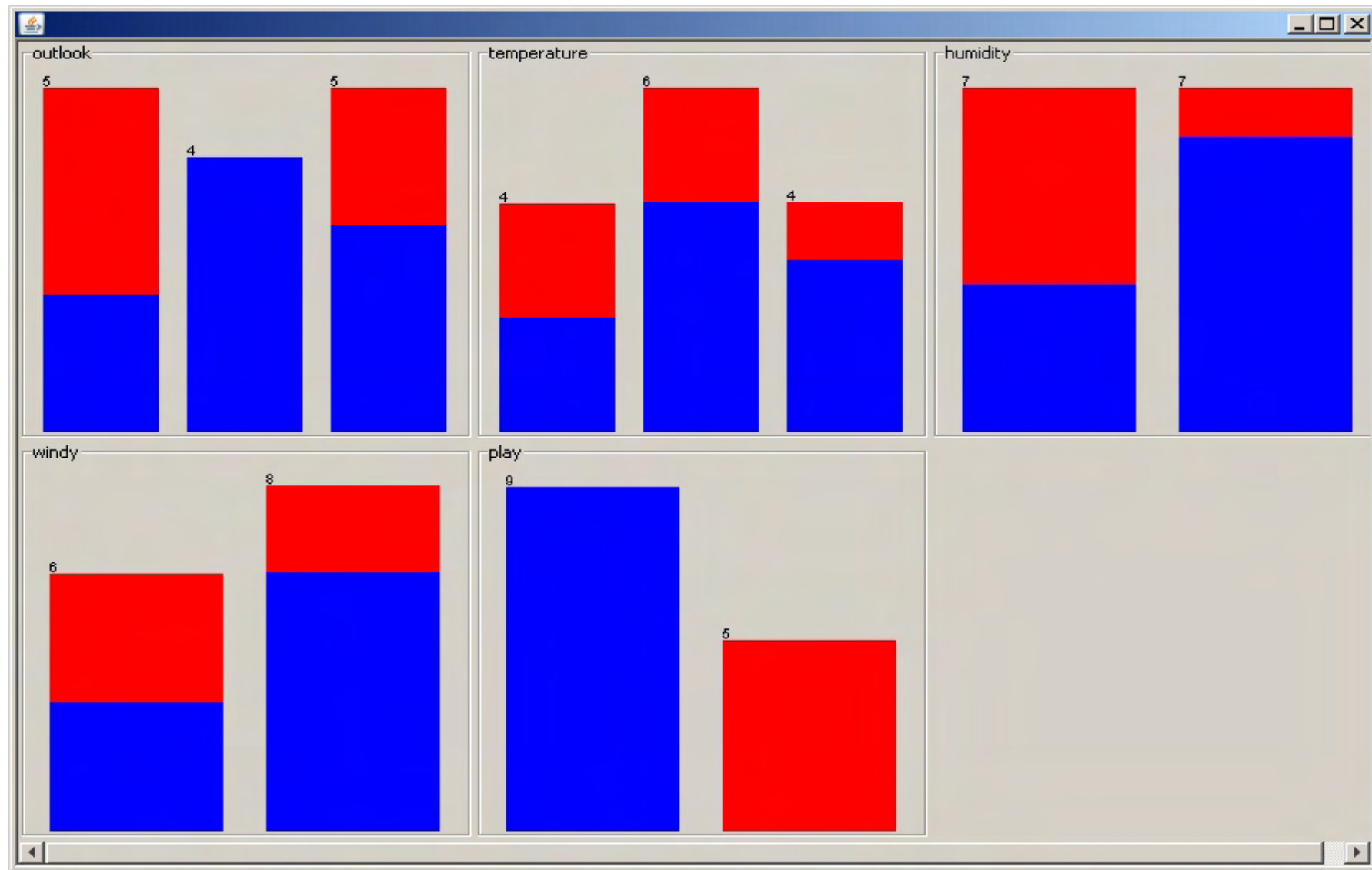
- **Preprocess:** applica ai dati vari tipi di filtraggio
- **Classify:** applica algoritmi di classificazione, fra cui i Decision Trees
- **Cluster:** applica algoritmi di raggruppamento dei dati in insiemi distinti
- **Associate:** applica algoritmi associativi per stabilire relazioni fra i dati
- **Select attributes:** cerca fra tutte le possibili combinazioni di attributi dei dati quelli più efficaci per la predizione
- **Visualize:** mostra graficamente i risultati di ciascuna delle operazioni precedenti

Preview dei dati (weather data)

L'istogramma a barre mostra per l'attributo *outlook* quante istanze *yes* e *no* sono presenti in ciascun ramo



Preview di tutti i dati



Risultato dell'albero delle decisioni di Weka

The screenshot displays the Weka Explorer interface. The 'Classifier' tab is active, showing the 'Choose' button and the selected classifier 'J48 -C 0.25 -M 2'. The 'Test options' section includes radio buttons for 'Use training set', 'Supplied test set', 'Cross-validation', and 'Percentage split'. The 'Classifier output' pane shows the following information:

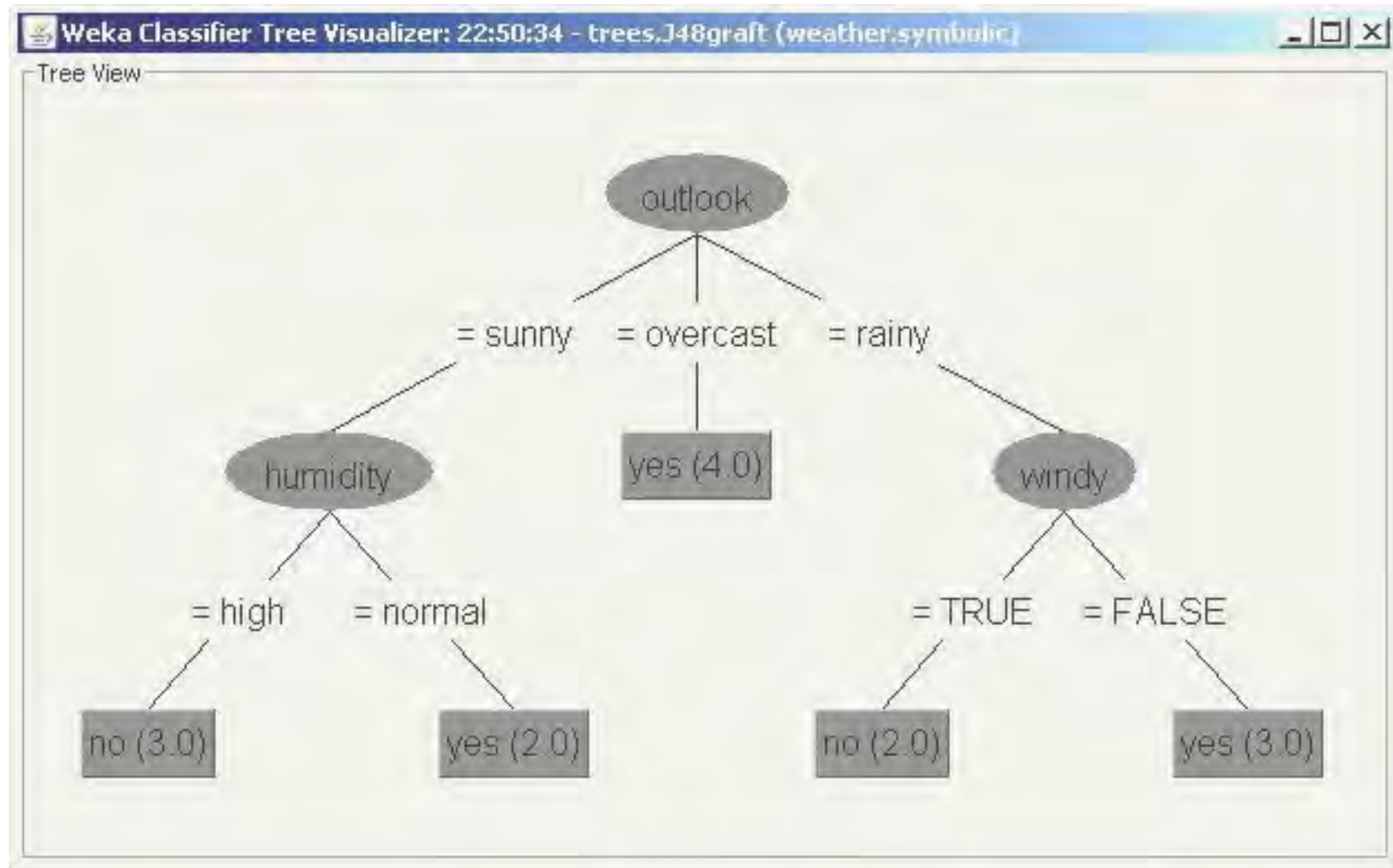
```
=== Run information ===  
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2  
Relation: weather.symbolic  
Instances: 14  
Attributes: 5  
outlook  
temperature  
humidity  
windy  
play  
Test mode: evaluate on training data  
=== Classifier model (full training set) ===  
J48 pruned tree  
-----  
outlook = sunny  
| humidity = high: no (3.0)  
| humidity = normal: yes (2.0)  
outlook = overcast: yes (4.0)  
outlook = rainy  
| windy = TRUE: no (2.0)  
| windy = FALSE: yes (3.0)  
Number of Leaves : 5  
Size of the tree : 8
```

Red boxes and arrows highlight the classifier name 'J48 -C 0.25 -M 2' and the decision tree structure. The status bar at the bottom shows 'OK' and a 'Log' button.

Risultato dell'albero delle decisioni di Weka

- outlook = sunny
 - humidity = high : no (3)
 - humidity = normal : yes (2)
- outlook = overcast : (4)
- outlook = rainy
 - windy = true : no (2)
 - windy = false : yes (3)

Visualizzazione dell'albero



Valutazione dell'albero

Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

| | | | |
|----------------------------------|----|-----|---|
| Correctly Classified Instances | 14 | 100 | % |
| Incorrectly Classified Instances | 0 | 0 | % |
| Kappa statistic | 1 | | |
| Mean absolute error | 0 | | |
| Root mean squared error | 0 | | |
| Relative absolute error | 0 | % | |
| Root relative squared error | 0 | % | |
| Total Number of Instances | 14 | | |

percentuali di classificazione

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure |
|---------------|---------|---------|-----------|--------|-----------|
| | 1 | 0 | 1 | 1 | 1 |
| | 1 | 0 | 1 | 1 | 1 |
| Weighted Avg. | 1 | 0 | 1 | 1 | 1 |

=== Confusion Matrix ===

a b <-- classified as
9 0 | a = yes
0 5 | b = no

matrice di confusione

La matrice di confusione ha sulla diagonale le istanze correttamente classificate e fuori diagonale quelle classificate erroneamente.

Nel caso perfetto essa è diagonale, altrimenti ha fuori diagonale il numero di istanze yes classificate come no e viceversa.

Ovviamente non c'è motivo perché sia simmetrica.

Opzioni di Output

Time taken to build model: 0 seconds

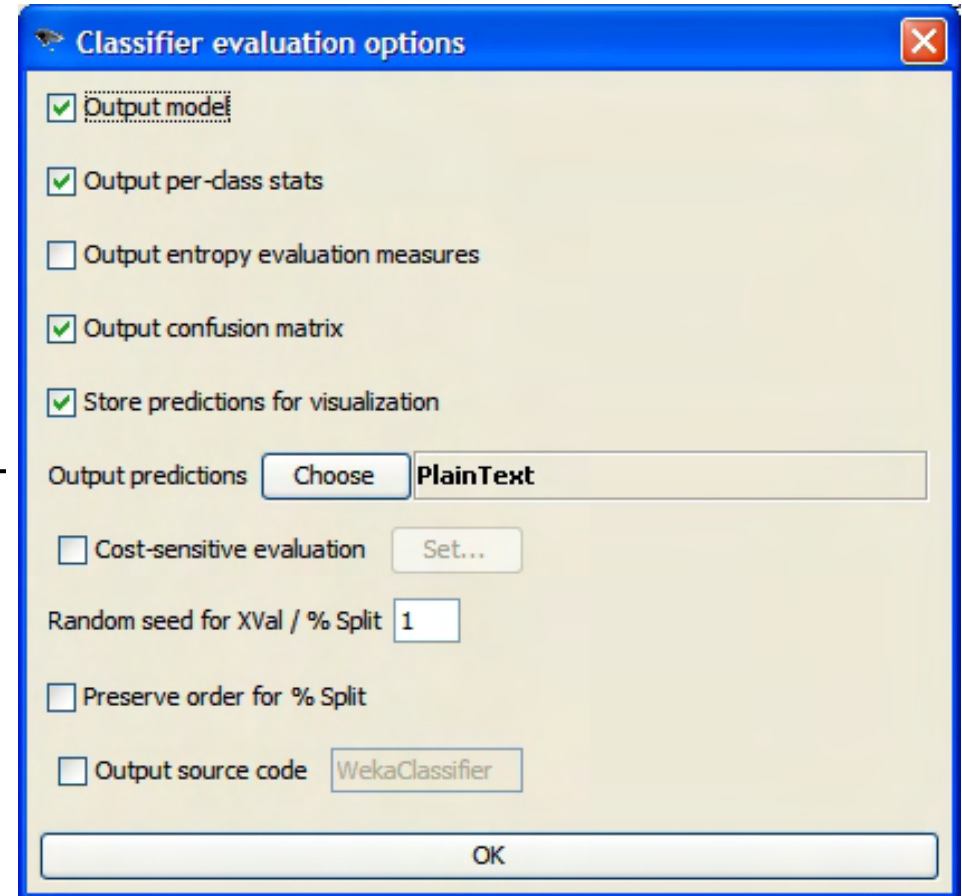
=== Predictions on training set ===

| inst# | actual | predicted | error | prediction |
|-------|--------|-----------|-------|------------|
| 1 | 2:no | 2:no | 1 | |
| 2 | 2:no | 2:no | 1 | |
| 3 | 1:yes | 1:yes | 1 | |
| 4 | 1:yes | 1:yes | 1 | |
| 5 | 1:yes | 1:yes | 1 | |
| 6 | 2:no | 2:no | 1 | |
| 7 | 1:yes | 1:yes | 1 | |
| 8 | 2:no | 2:no | 1 | |
| 9 | 1:yes | 1:yes | 1 | |
| 10 | 1:yes | 1:yes | 1 | |
| 11 | 1:yes | 1:yes | 1 | |
| 12 | 1:yes | 1:yes | 1 | |
| 13 | 1:yes | 1:yes | 1 | |
| 14 | 2:no | 2:no | 1 | |

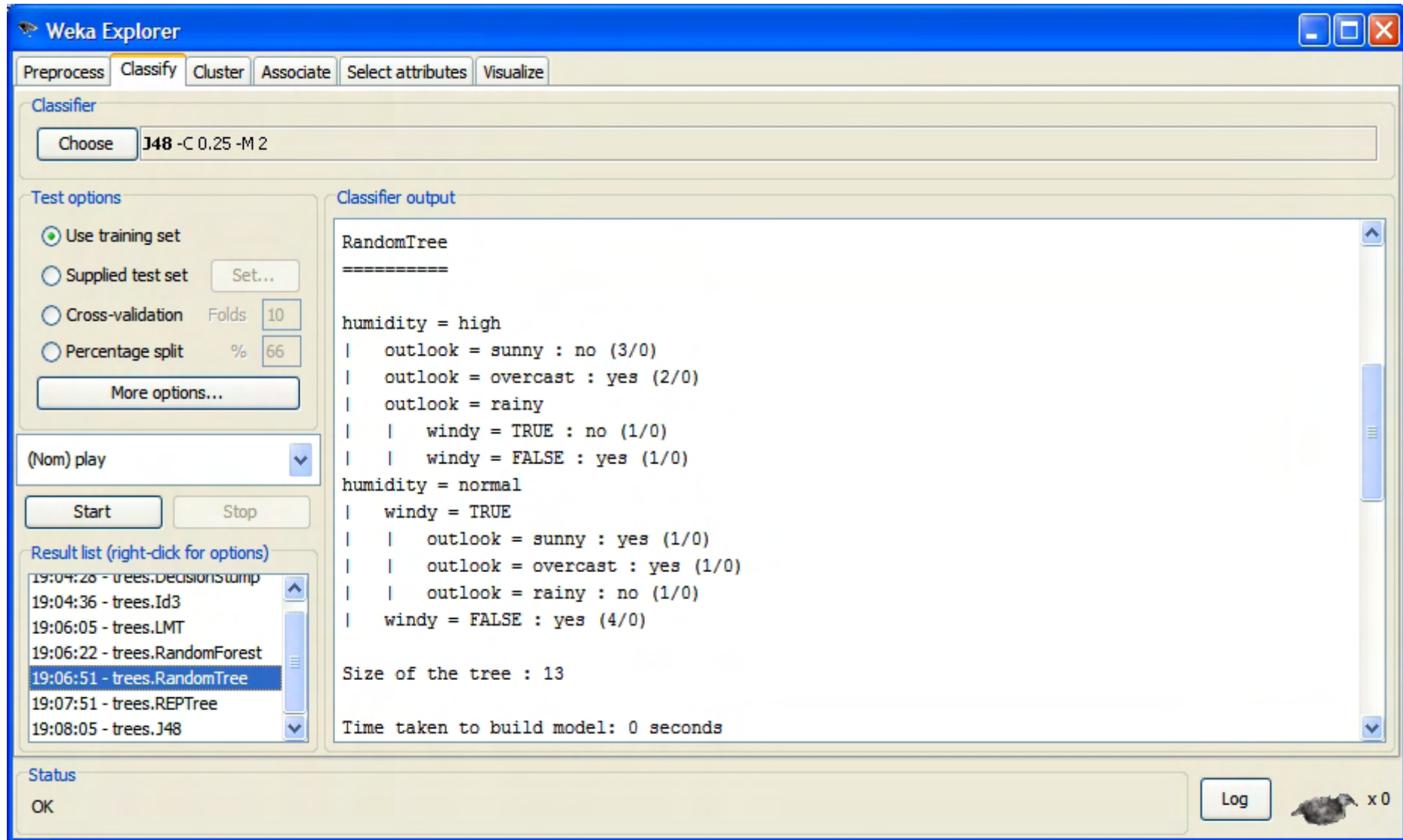
=== Evaluation on training set ===

=== Summary ===

| | | | |
|----------------------------------|----|-----|---|
| Correctly Classified Instances | 14 | 100 | % |
| Incorrectly Classified Instances | 0 | 0 | % |



Algoritmo “Random Tree”



The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 -C 0.25 -M 2'. The 'Test options' section is set to 'Use training set'. The 'Classifier output' window displays the following text:

```
RandomTree
=====

humidity = high
| outlook = sunny : no (3/0)
| outlook = overcast : yes (2/0)
| outlook = rainy
| | windy = TRUE : no (1/0)
| | windy = FALSE : yes (1/0)
humidity = normal
| windy = TRUE
| | outlook = sunny : yes (1/0)
| | outlook = overcast : yes (1/0)
| | outlook = rainy : no (1/0)
| windy = FALSE : yes (4/0)

Size of the tree : 13

Time taken to build model: 0 seconds
```

The 'Result list' shows several models, with '19:06:51 - trees.RandomTree' selected. The 'Status' bar at the bottom indicates 'OK'.

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

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

Test options

Use training set

Supplied test set

Cross-validation Folds

Percentage split %

(Nom) play

Result list (right-click for options)

- 19:04:28 - trees.DecisionStump
- 19:04:36 - trees.Id3
- 19:06:05 - trees.LMT
- 19:06:22 - trees.RandomForest
- 19:06:51 - trees.RandomTree**
- 19:07:51 - trees.REPTree
- 19:08:05 - trees.J48

Classifier output

```


=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances      14      100 %
Incorrectly Classified Instances    0        0 %
Kappa statistic                    1
Mean absolute error                 0
Root mean squared error             0
Relative absolute error             0 %
Root relative squared error         0 %
Coverage of cases (0.95 level)     100 %
Mean rel. region size (0.95 level)  50 %
Total Number of Instances          14

=== Detailed Accuracy By Class ===
                TP Rate  FP Rate  Precision  Recall  F-Measure  ROC Area  Class
                1      0      1      1      1      1      yes
                1      0      1      1      1      1      no
Weighted Avg.   1      0      1      1      1      1

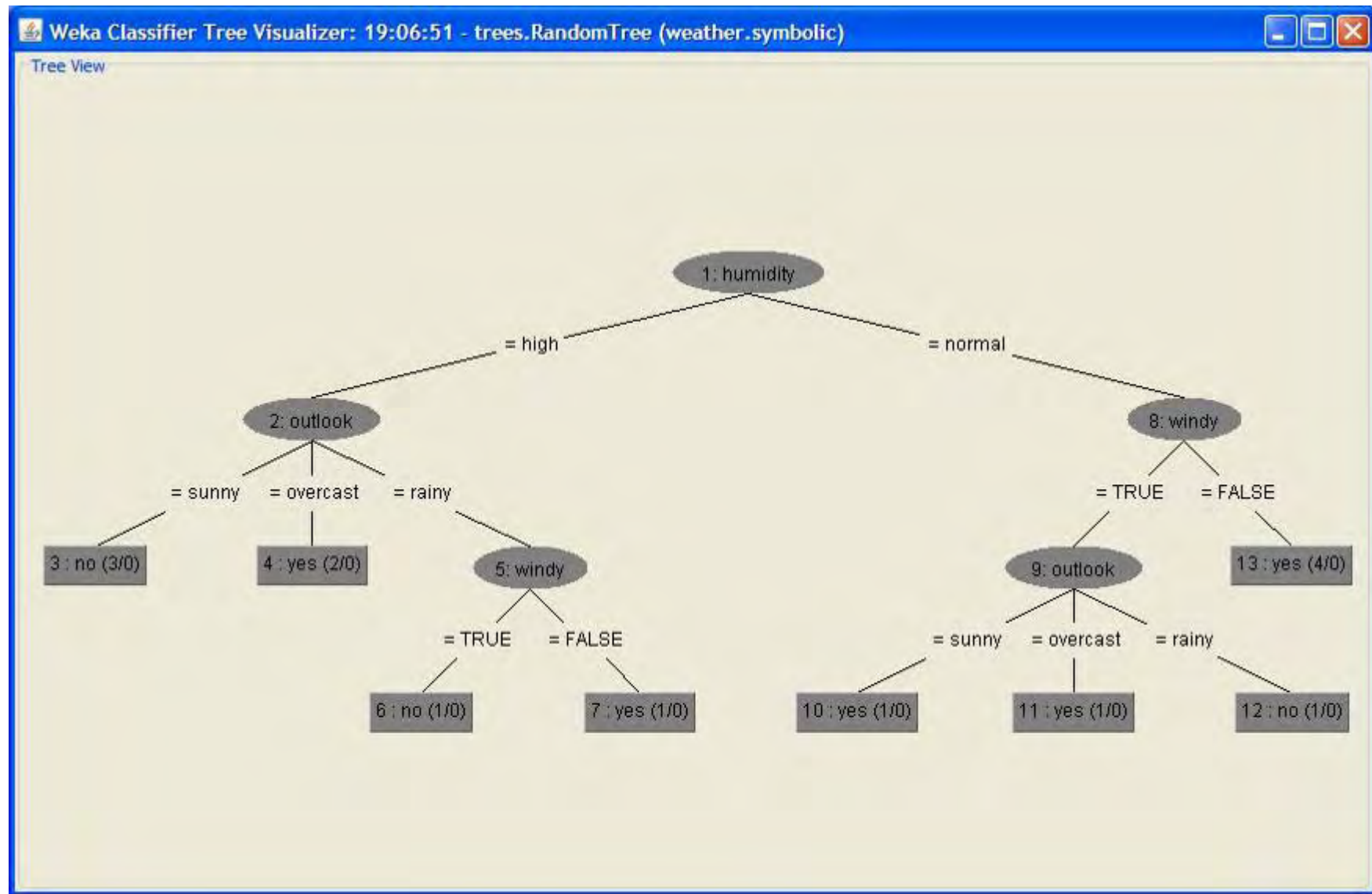
=== Confusion Matrix ===
 a b  <-- classified as
 9 0 | a = yes
 0 5 | b = no

```



Status

OK  x 0

Albero generato da “Random Tree”



Test options

-  Una volta scelto l'algoritmo di classificazione, il test può essere fatto con le seguenti 4 opzioni:
1. ***Use training set.*** Il classificatore è valutato sulla bontà di predizione delle istanze usate per l'allenamento.
 2. ***Supplied test set.*** Il classificatore è valutato sulla bontà di predizione di istanze caricate da un file (ovviamente dovrà essere stato allenato in precedenza su altre istanze).
 3. ***Cross-validation.*** Il classificatore è valutato mediante validazione incrociata, prendendo un dato ogni N dal training set. N è specificabile come "Folds".
 4. ***Percentage split.*** Il classificatore è valutato sulla bontà di predizione di una certa percentuale (specificabile dall'utente) di istanze che erano state accantonate per il test.
-  ***Nota:*** Indipendentemente dal metodo di valutazione, il modello presentato è sempre costruito sull'intero training set.



Controllo dell'aerazione nell'impianto di depurazione di San Colombano

dalla tesi di Laurea
di
Stefano Banchelli

Applicazione dell'albero delle decisioni

Tipo di controllore

- ⇒ Albero delle decisioni per ottenere un rapporto ottimale fra volume aerato ed anossico

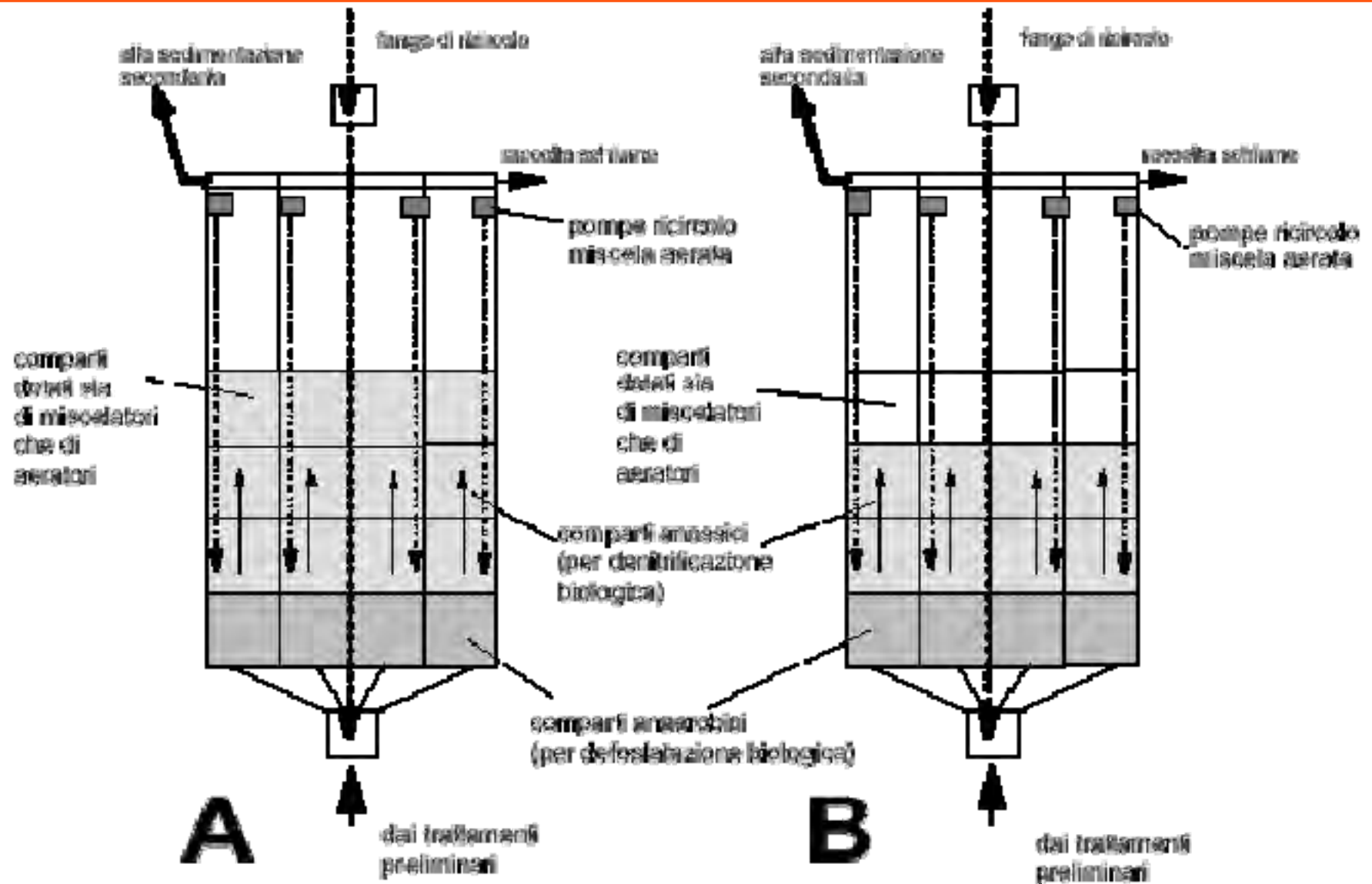
Motivazioni per il controllo

- ⇒ L'alto rapporto COD/N ostacola lo sviluppo della biomassa
- ⇒ E' necessario limitare i superamenti di N_{tot} rispetto ai limiti di legge
- ⇒ Riduzione del costo di aerazione

Struttura del controllore

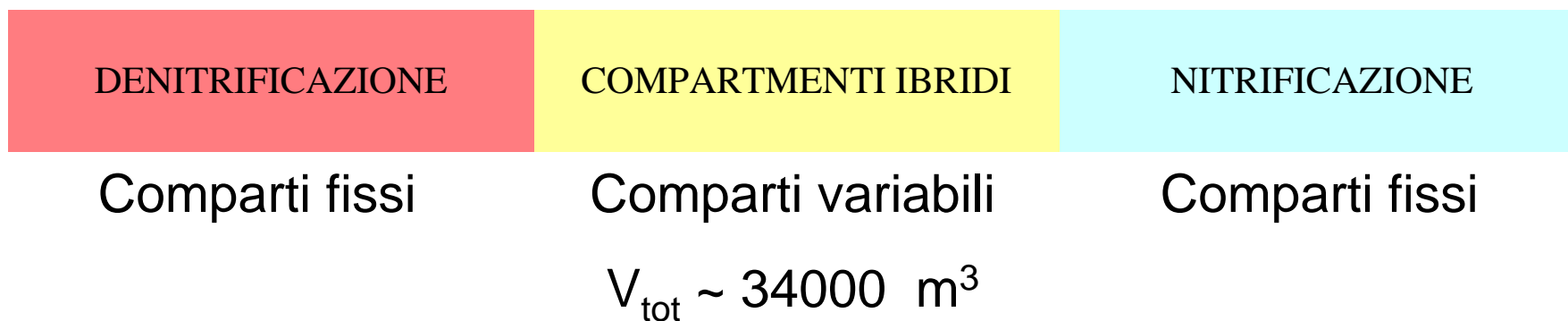
- ⇒ Un controllo tradizionale è irrealizzabile per mancanza di strumentazione
- ⇒ Si sfrutta la struttura a comparti dell'ossidazione variandone la configurazione (partizione variabile aerobica/anossica)
- ⇒ L'aggiornamento della configurazione avviene ogni 3 ore.

Vasche di ossidazione a configurazione variabile



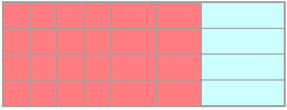
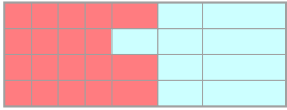
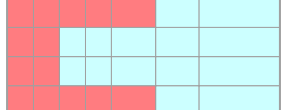
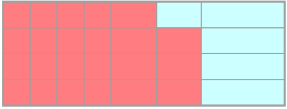
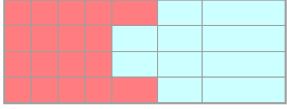
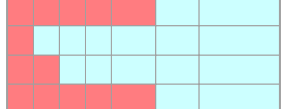
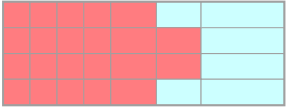
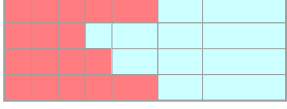
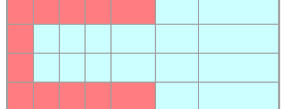
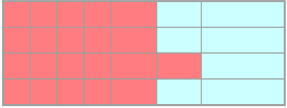
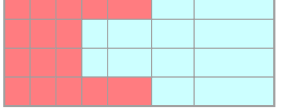
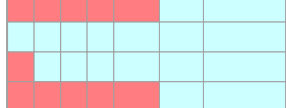
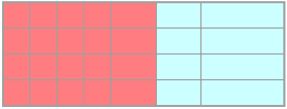
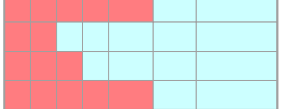
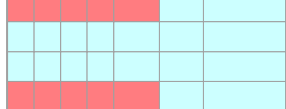
Possibilità di ripartizione del volume aerato

| | | | | | | |
|--------------------|--------------------|--------------------|--------------------|--------------------------|--------------------------|---------------------------|
| 652 m ³ | 652 m ³ | 652 m ³ | 652 m ³ | 979 m³ | 979 m³ | 3932 m³ |
| 652 m ³ | 652 m ³ | 652 m ³ | 652 m ³ | 979 m³ | 979 m³ | 3932 m³ |
| 652 m ³ | 652 m ³ | 652 m ³ | 652 m ³ | 979 m³ | 979 m³ | 3932 m³ |
| 652 m ³ | 652 m ³ | 652 m ³ | 652 m ³ | 979 m³ | 979 m³ | 3932 m³ |



Simulazione della strategia di controllo (1)

☞ La qualità dell'effluente viene simulata ad intervalli di 3 ore per le 15 possibili configurazioni (V_A = Vol. Aerato, V_D = Vol. Denitro)

| | | | | | |
|---|----------------------------|--|----------------------------|---|---------------------------|
|  | $V_A=15730$ $V_D=18270$ |  | $V_A=20624$ $V_D=13376$ |  | $V_A=24212$ $V_D=9788$ |
|  | $V_A=16709$ $V_D=17291$ |  | $V_A=21602$ $V_D=12397$ |  | $V_A=24865$ $V_D=9135$ |
|  | $V_A=17688$ $V_D=16312$ |  | $V_A=22254$ $V_D=11746$ |  | $V_A=25517$ $V_D=8483$ |
|  | $V_A=18667$ $V_D=15333$ |  | $V_A=22907$ $V_D=11093$ |  | $V_A=26170$ $V_D=7830$ |
|  | $V_A=19645$ $V_D=14355$ |  | $V_A=23560$ $V_D=10440$ |  | $V_A=26822$ $V_D=7178$ |

Posizione del problema di controllo ottimale

☞ Definita la qualità dell'effluente come

$$EQ = \frac{1}{T \cdot 1000} \int_t^T [2 \cdot TSS_e(t) + COD_e(t) + 2 \cdot BOD_e(t) + 20 \cdot TKN_e(t) + 20 \cdot S_{NOX_e}(t)] \cdot Q_e(t) \cdot dt$$

☞ Fra le 15 configurazioni possibili, trovare quella che minimizza EQ

$$\{V_A, V_D\}_{opt} = \arg \min_{\{V_A, V_D\}} EQ$$

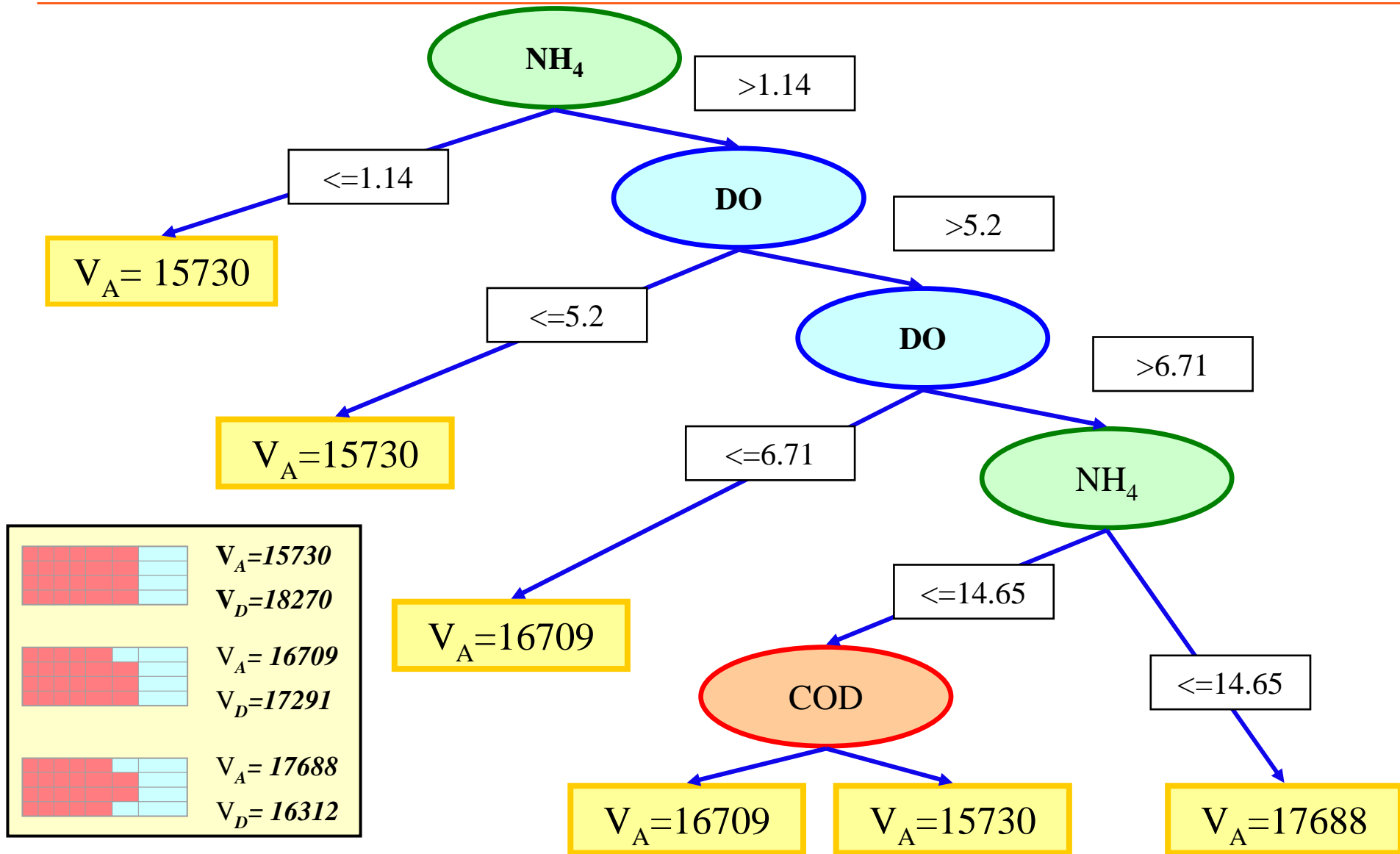
☞ I valori di EQ nelle varie configurazioni sono quelli ottenuti simulando il modello per un orizzonte di 3 ore.

Struttura del controllore

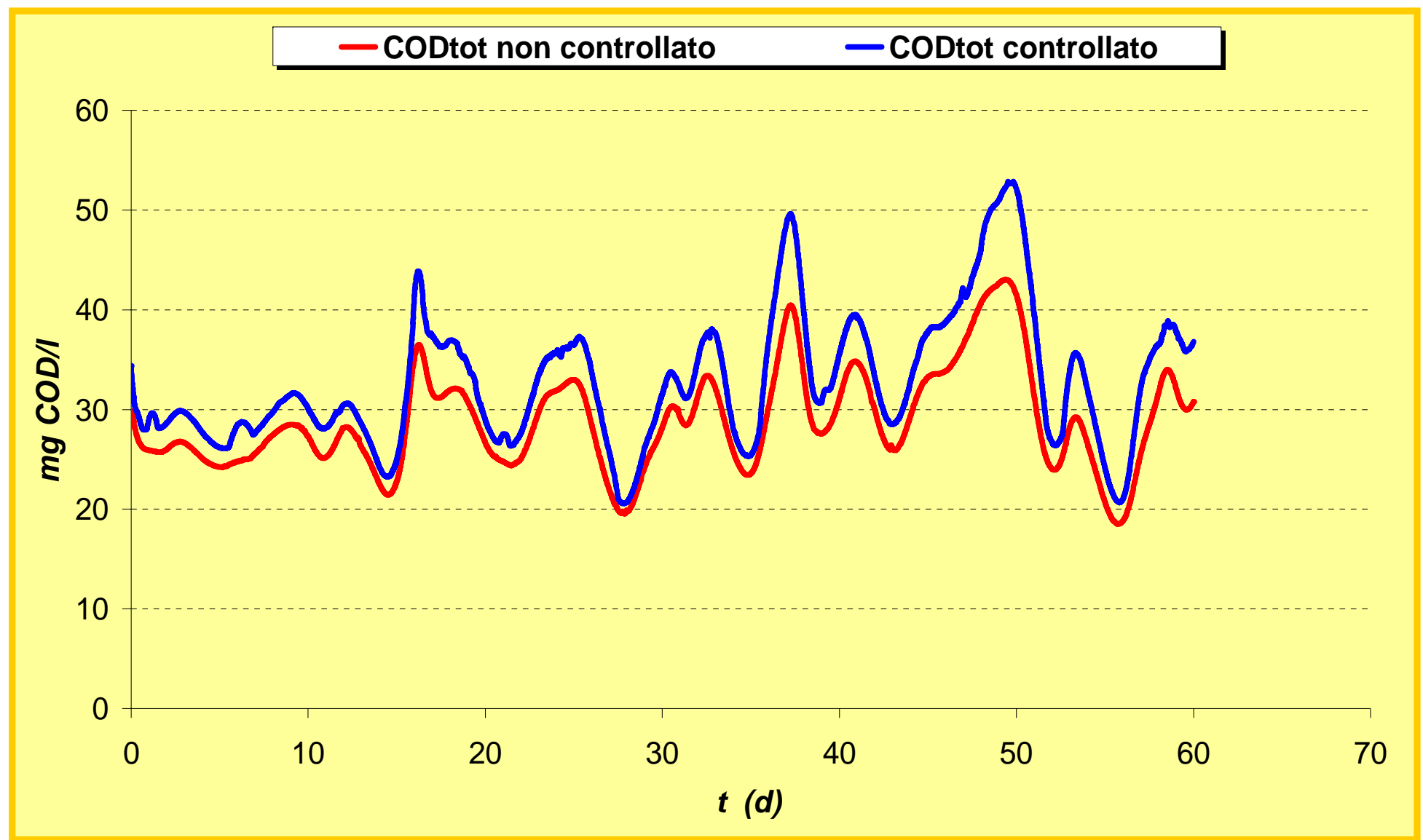
- 👉 Sistema campionato a intervalli di 3 ore
- 👉 Il controllore consiste in una serie di regole per determinare la migliore partizione fra volume aerato (V_A) e volume per la denitro (V_D)
- 👉 Il relativo albero delle decisioni è prodotto tramite WEKA
- 👉 I dati di allenamento (e di decisione) sono i campioni a 3 ore di:
 - ➡ COD
 - ➡ NH_4
 - ➡ DO
- 👉 Il parametro di uscita (attributo di classe) è il volume aerato V_A
- 👉 Deve comunque essere soddisfatto il vincolo sul volume totale

$$V_{tot} = V_A + V_D = 34000 \text{ m}^3$$

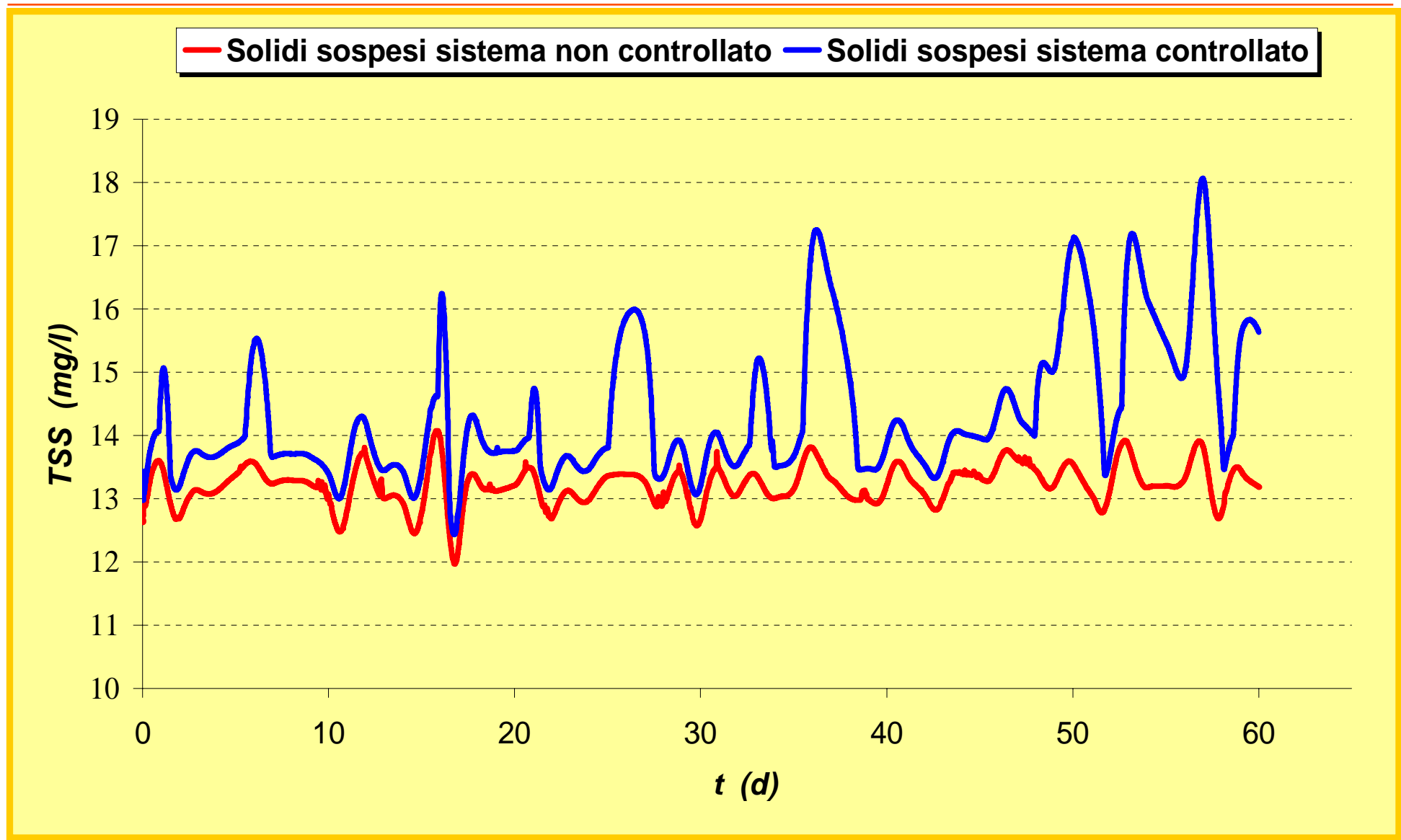
Struttura dell'albero prodotto da WEKA



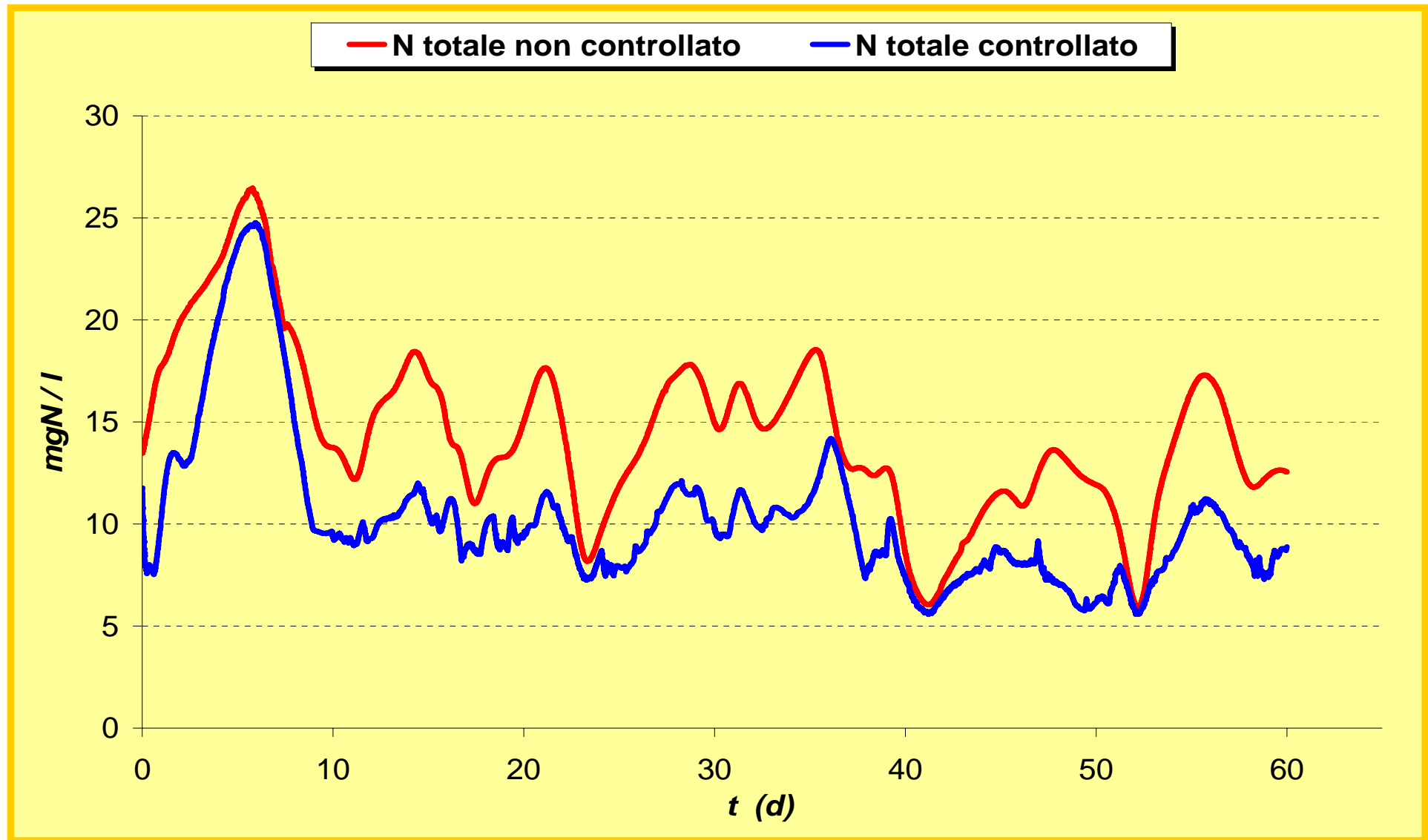
Prestazione del sistema controllato: COD



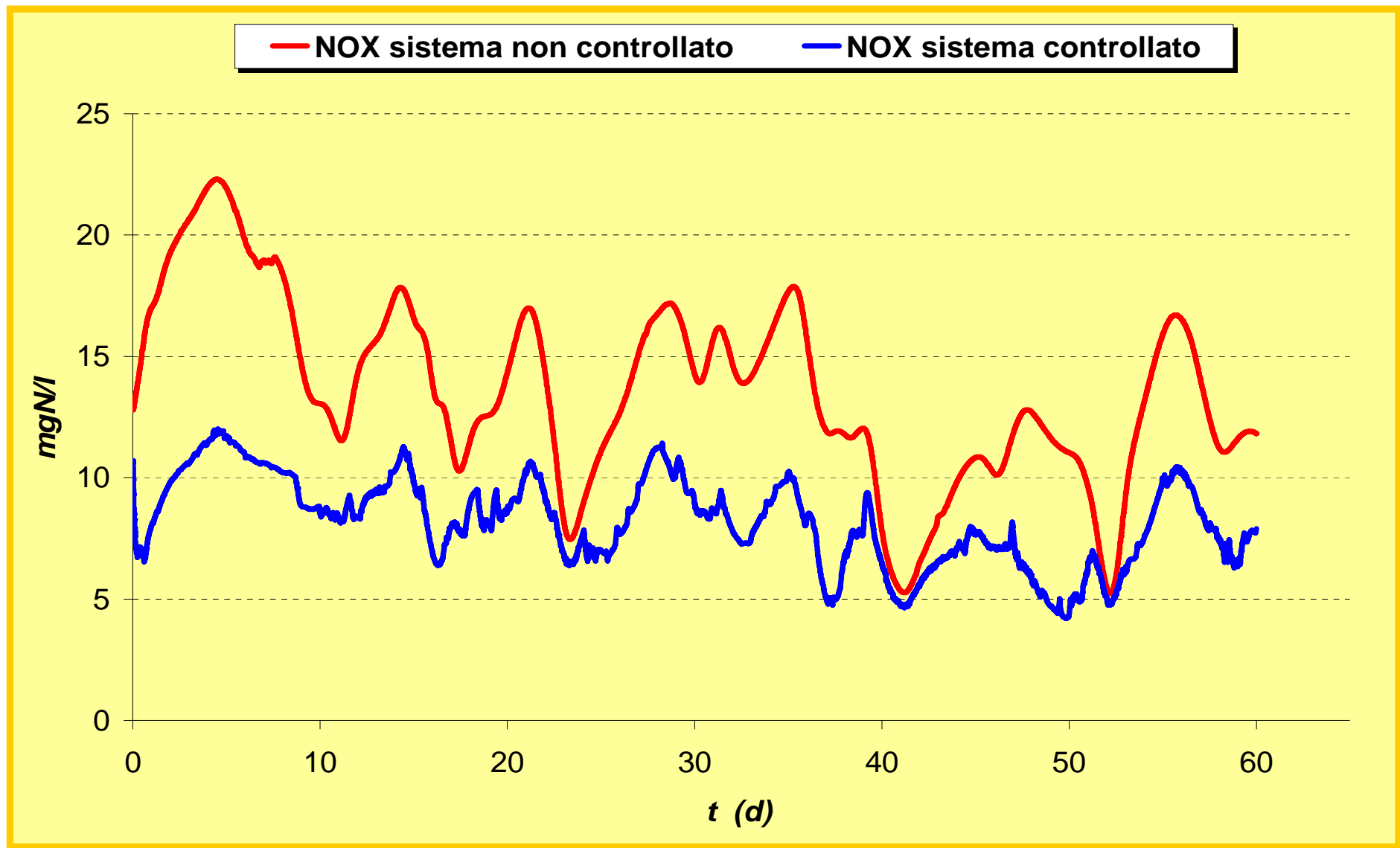
Prestazione del sistema controllato: SS_{tot}



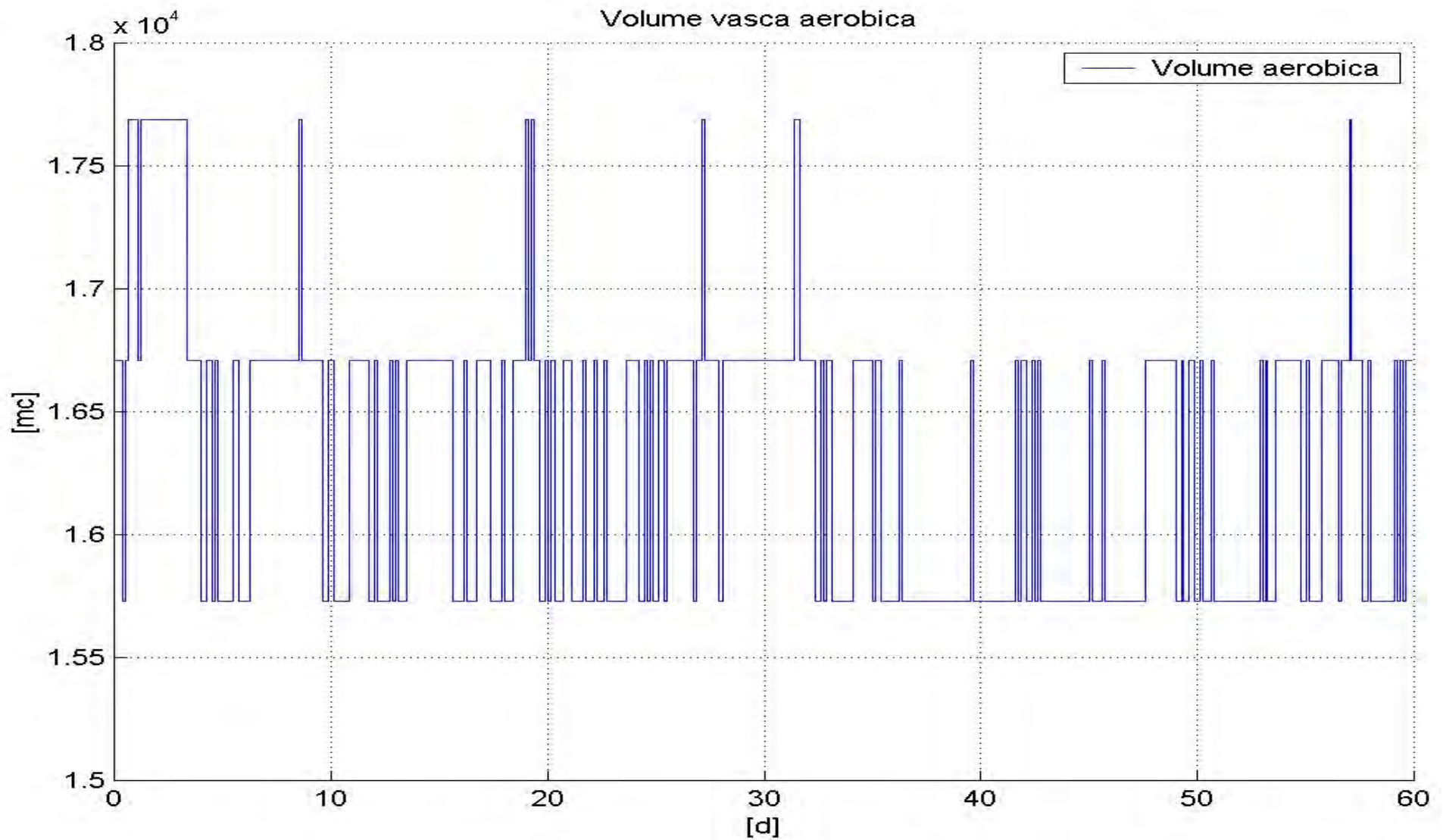
Prestazione del sistema controllato: N_{tot}



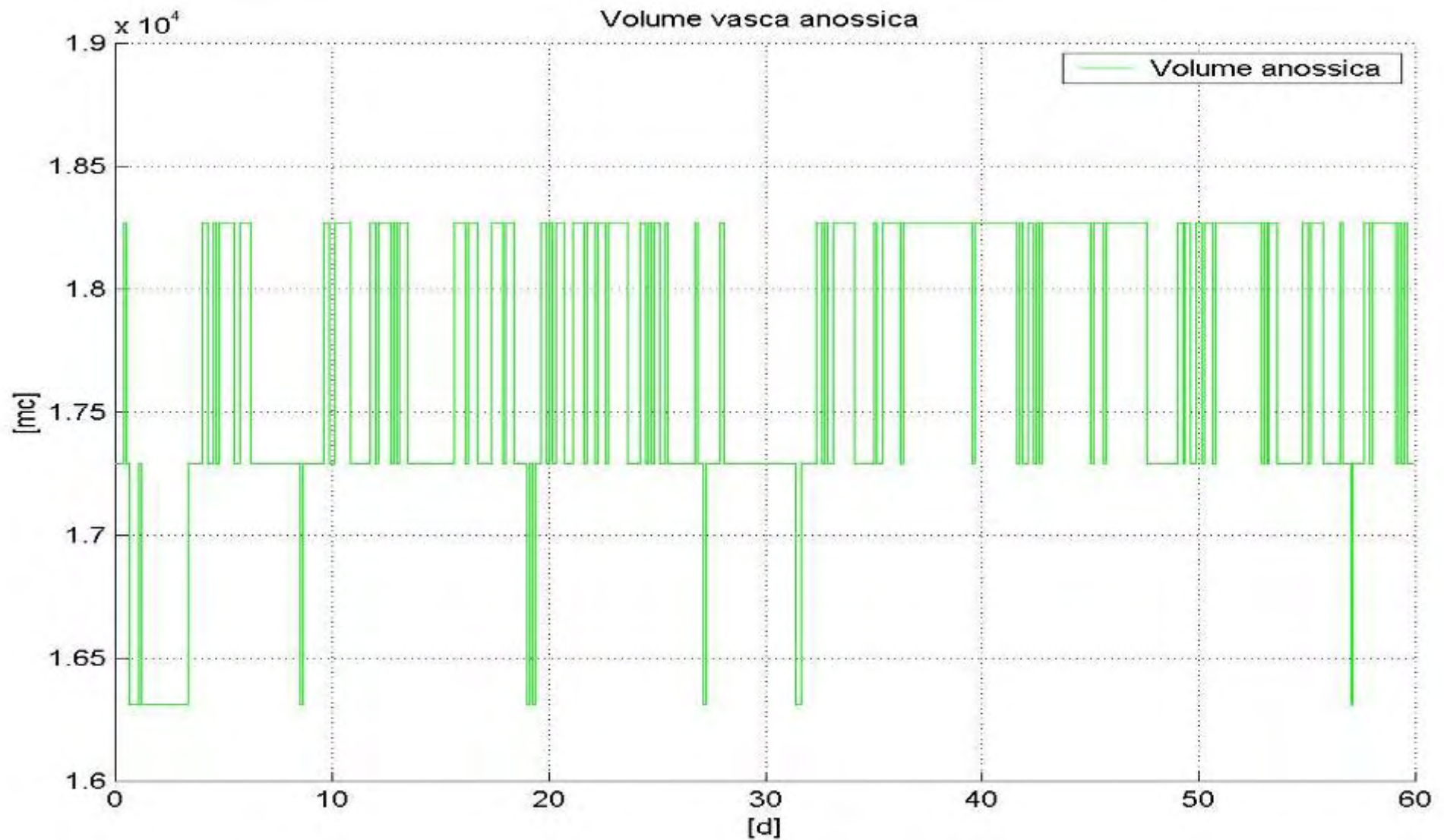
Prestazione del sistema controllato: NO_x



Variazione controllata del volume aerato



Variazione controllata del volume anossico



Risultati di controllo

- 👉 Miglioramento della qualità dell'effluente EQ
 - ➡ 21083 → 13109
- 👉 Riduzione dell'azoto totale in uscita (obiettivo primario), con rientro nei limiti allo scarico
- 👉 Lieve aumento della produzione di fanghi (t/d)
 - ➡ 173.42 → 182.15
- 👉 Minor concentrazione di solidi in uscita (mg/L)
 - ➡ 8.73 → 8.19