Classificazione e Predizione

Lezione di TDM – DM del 16 Aprile 2007

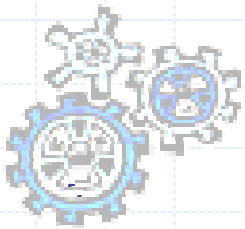
Francesco Bonchi,

KDD Lab Pisa,

ISTI-C.N.R.

Lezione odierna

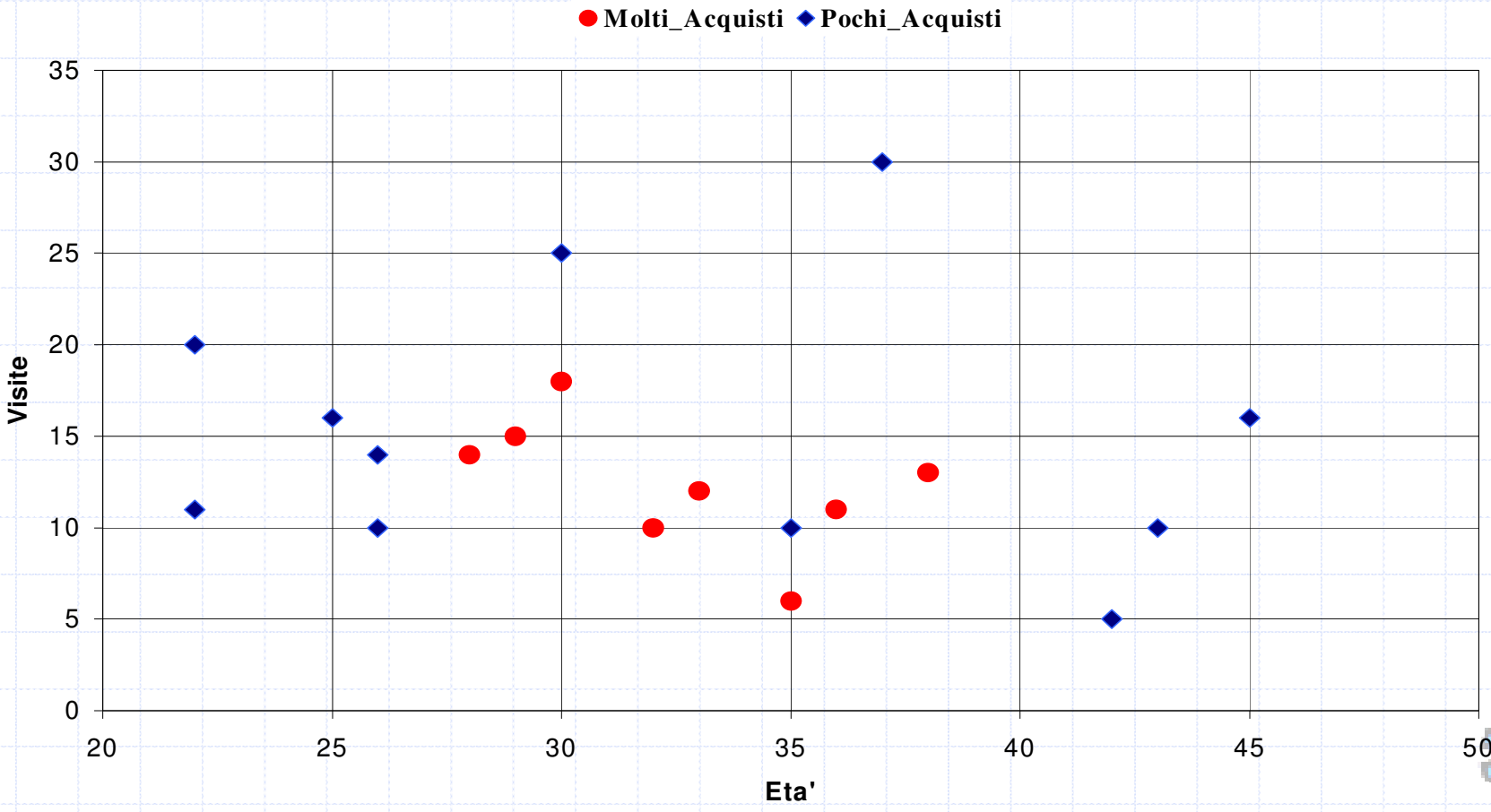
- Intuizioni sul concetto di classificazione
- Alberi di decisione
- Alberi di decisione con Weka
- Classificazione: aspetti pratici e concetti avanzati
- Classificazione con Weka e Clementine
- Esercitazione



Intuizioni sul concetto di classificazione

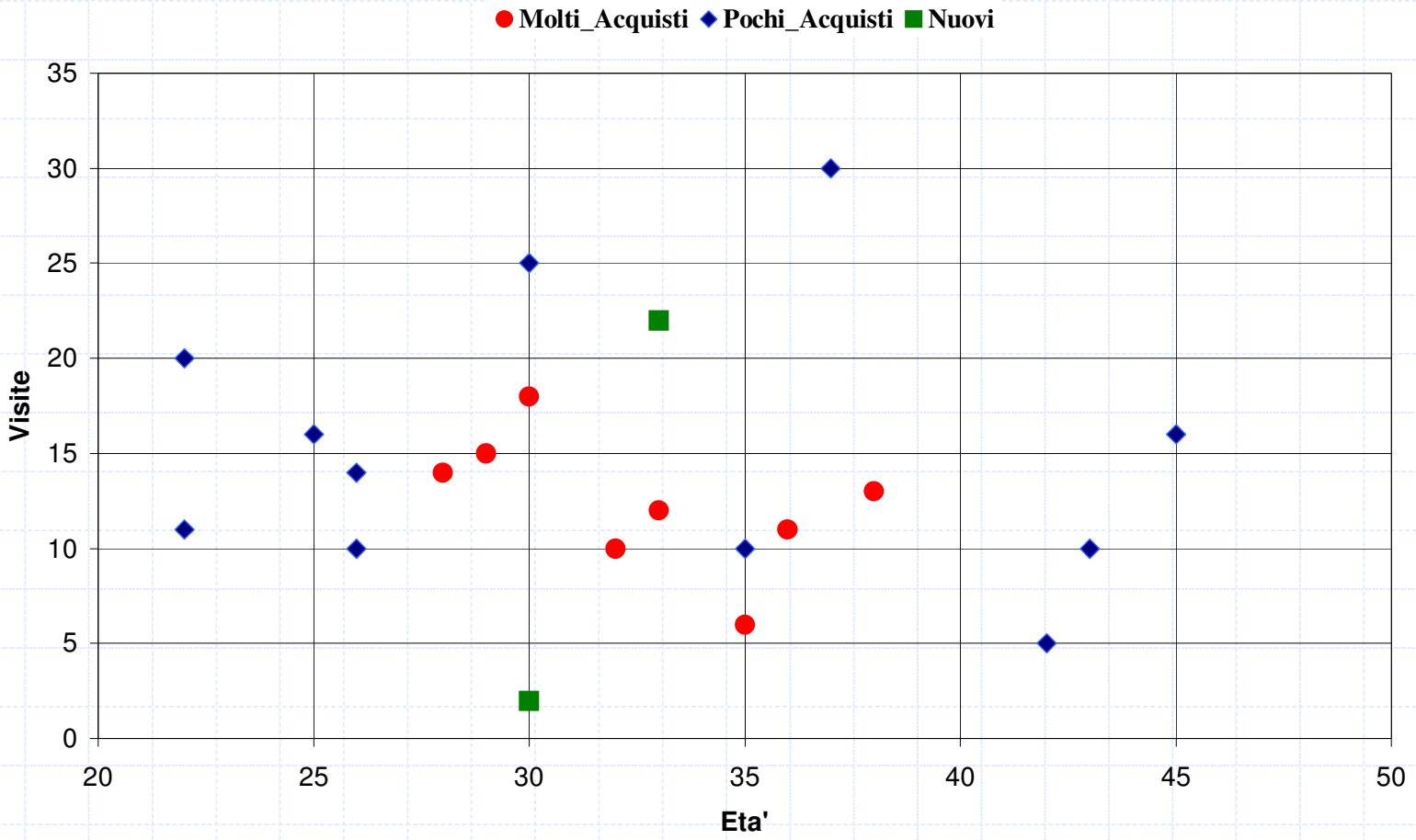
Descrivete i vostri clienti migliori

… in base ad età ed al numero di visite nell'ultimo periodo!



… ora provate a prevedere …

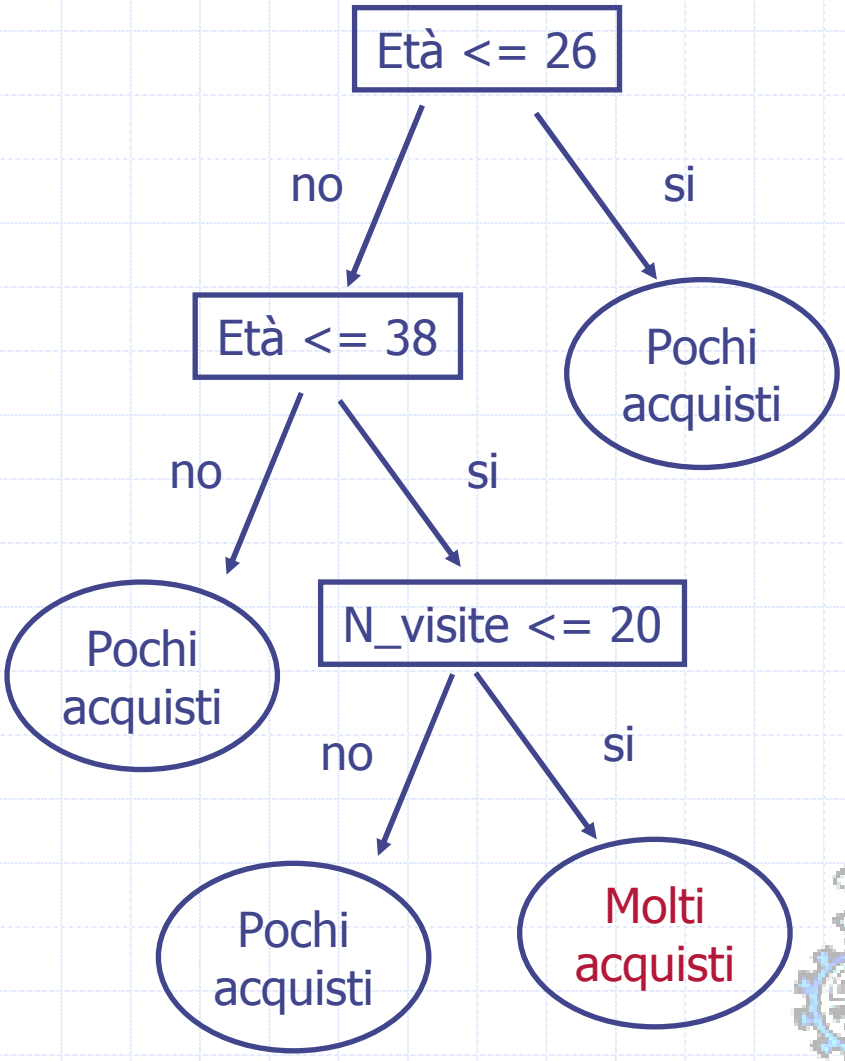
… la tipologia di due nuovi clienti, in base a età e numero visite.



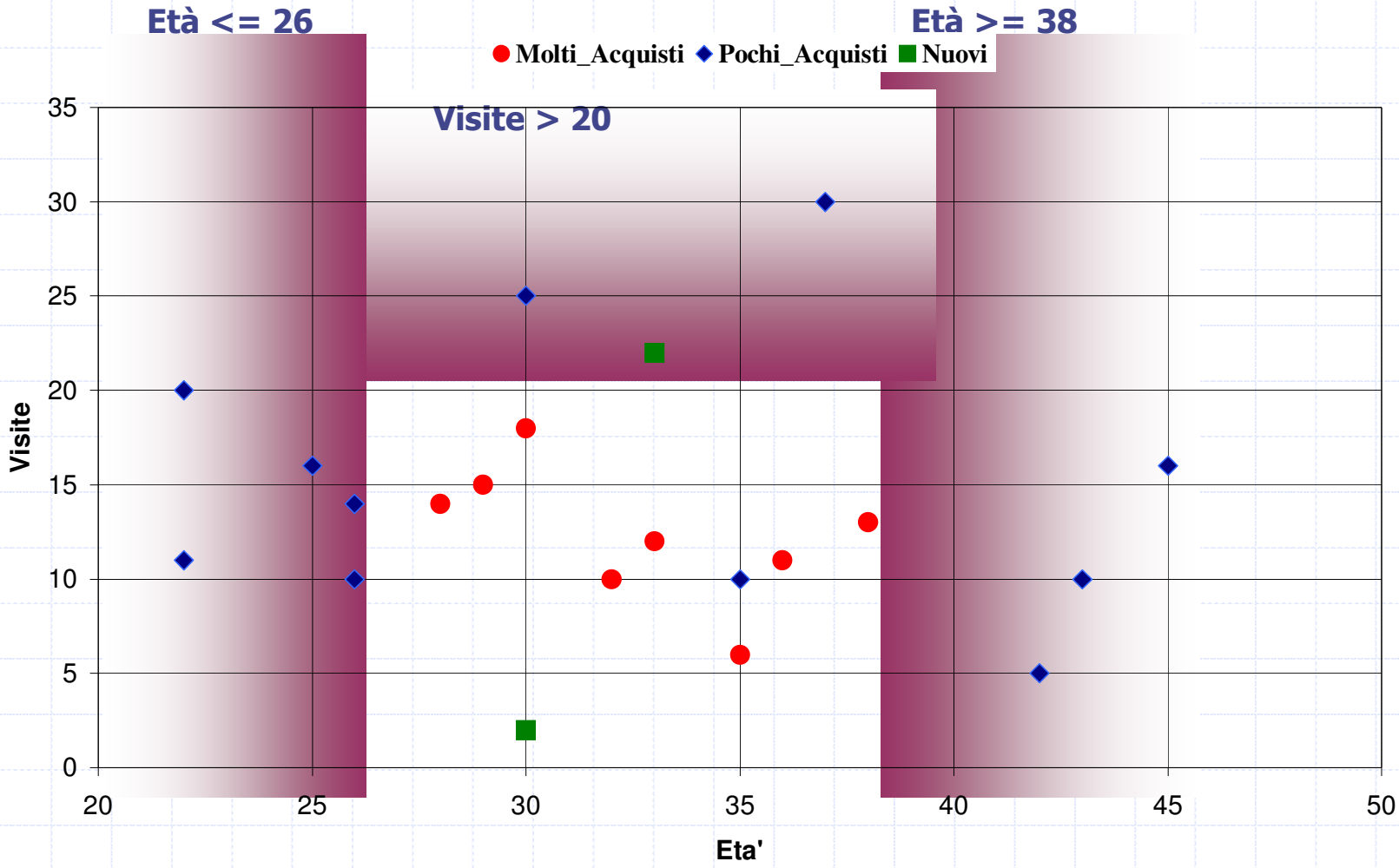
Una possibile soluzione ...

Results for es01

```
See5 INDUCTION SYSTEM [Release 1.08]    Sun Mar 3
-----------------------------------

Read 19 cases (2 attributes) from es01.data

Decision tree:

eta' <= 26: pochi_acquisti (5.0)
eta' > 26:
:...eta' > 38: pochi_acquisti (3.0)
    eta' <= 38:
    :...n visite <= 20: molti acquisti (9.0/1.0)
        n_visite > 20: pochi_acquisti (2.0)


Evaluation on training data (19 cases):

          Decision Tree
        ----------------
        Size      Errors

           4    1( 5.3%)    <<
```



… rappresentata graficamente

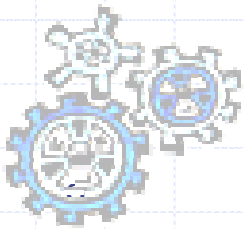


Il processo seguito:

- Abbiamo individuato una caratteristica "interessante" dei nostri clienti
  - Fare molti acquisti o pochi acquisti
- Abbiamo individuato altre caratteristiche "meno interessanti"
  - Età, Numero di visite
- Abbiamo raccolto dei dati storici
  - 19 clienti di cui si conoscono TUTTE le caratteristiche
- Abbiamo indotto dai dati storici una descrizione (**CLASSIFICAZIONE**) della caratteristica "interessante" sulla base di quelle "meno interessanti"
  - tale descrizione cerca di essere *concisa*, *intellegibile*, *informativa*
- (**PREDIZIONE**) Abbiamo utilizzato il modello di classificazione per derivare la caratteristica "interessante" di un nuovo cliente a partire da quelle "meno interessanti"

Applicazioni della classificazione:

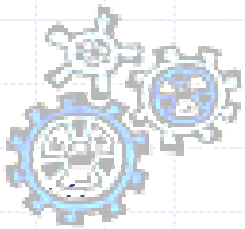
- Market analysis
  - customer profiling, individuare potenziali nuovi clienti, individuare i clienti piu' appropriati per una campagna promozionale, ecc.
- Risk analysis
  - individuare i clienti fedeli, controllo di qualita' dei processi, ecc.
- Fraud detection
  - individuare simulatori di incidenti stradali, movimenti bancari sospetti, acquisti con carta di credito sospetti, dichiarazioni dei redditi sospette, ecc.
- Web mining
  - categorizzare e-mail, personalizzazione di siti web, ecc.
- Altre applicazioni
  - medicina, astronomia, biologia, ecc.



Classificazione e predizione: input

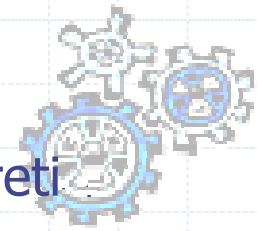
- Un insieme di esempi (dette **istanze** o casi) che descrivono un concetto (detto **classe**) sulla base di **attributi** (detti anche features)

outlook	temperature	humidity	windy	classe
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play



Classificazione e predizione: input

- La **classe** e gli **attributi** possono assume valori *continui, ordinali o discreti*
  - Sono valori **continui** i numeri reali
    - Due valori sono tra loro confrontabili ed esiste una nozione di distanza
    - Esempi: età, reddito, costo, temperatura, ecc.
  - Sono valori **ordinali** gli elementi di insiemi finiti ed ordinati
    - Due valori sono tra loro confrontabili, ma non esiste una nozione di distanza
    - Esempi: { freddo, tiepido, caldo }, { insufficiente, sufficiente, buono, ottimo }
  - Sono valori **discreti** gli elementi di insiemi finiti e non ordinati
    - Due valori non sono tra loro confrontabili
    - Esempi: { nuvoloso, soleggiato, ventoso }, { alimentari, elettrodom., abbigliamento }
- Discretizzazione
  - Funzione di trasformazione di valori continui in valori ordinali o discreti



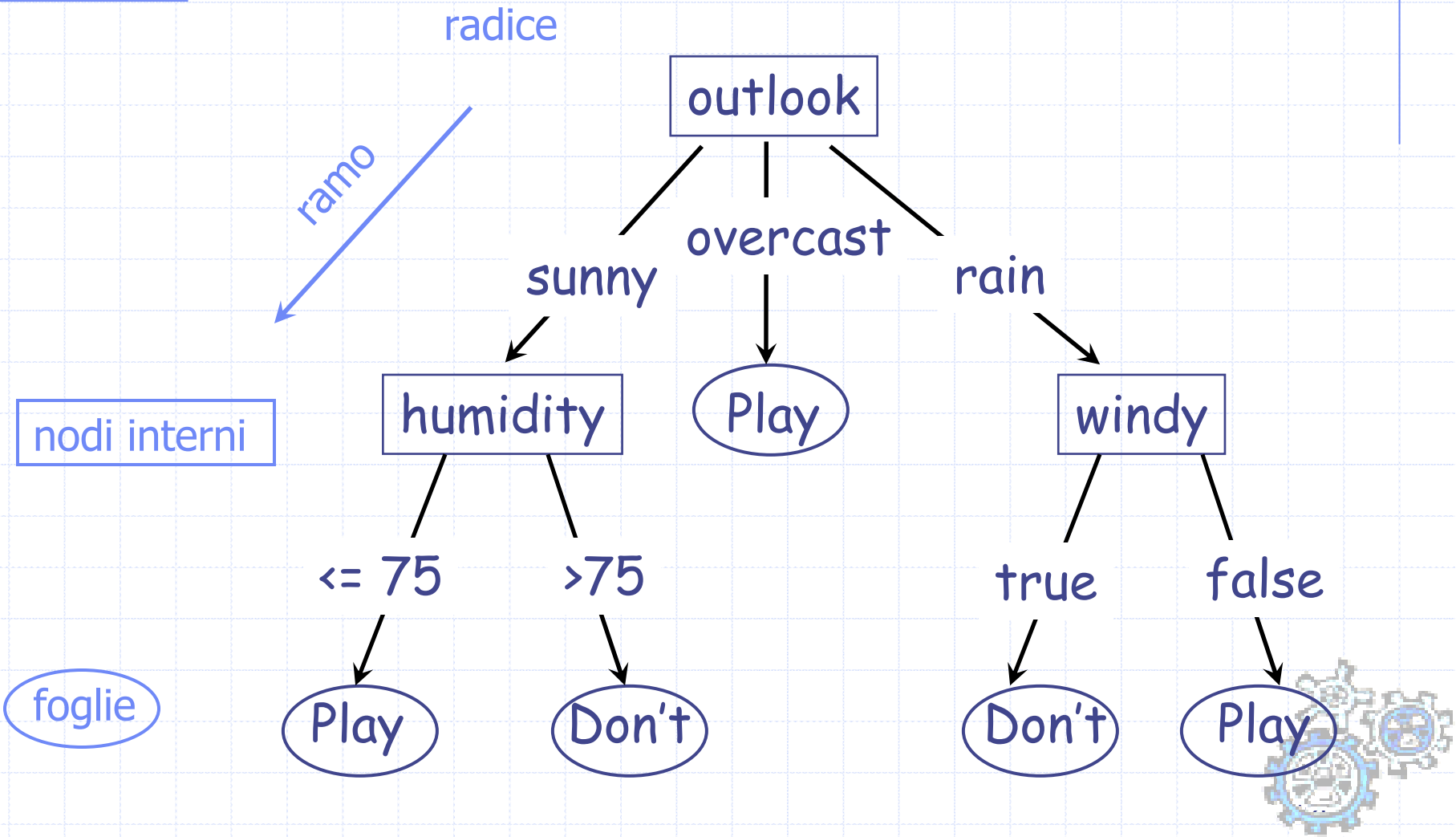
Classificazione e predizione: modelli

- A partire dagli esempi viene costruito un **modello** in grado di descrivere il valore della classe *in base a* quello degli attributi

- Quando la classe assume valori discreti, si parla di ***classificazione***
  - Esistono diverse possibili modelli di classificazione
    - Alberi di decisione, Regole di classificazione, Classificazione Bayesiana,K-nearest neighbors, Case-based reasoning, Genetic algorithms, Rough sets, Fuzzy logic, Association-based classification
- Quando la classe assume valori continui, si parla di ***regressione***
  - Esistono diversi possibili modelli di regressione
    - Alberi di regressione, Reti neurali, regressione lineare e multipla, regressione non-lineare, regressione logistica e di Poisson, regressione log-lineare

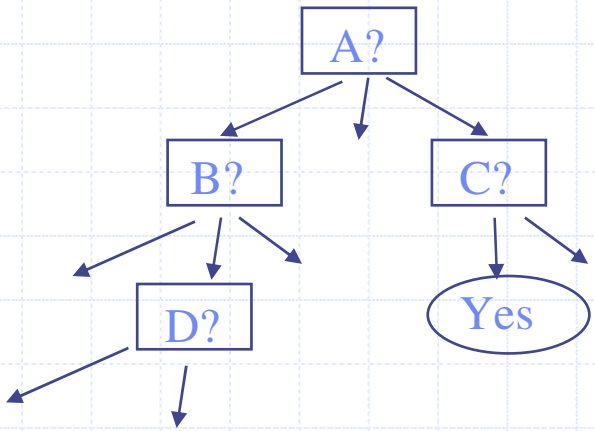
Alberi di decisione

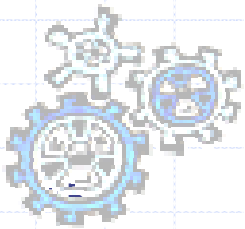
Modelli di classificazione: alberi di decisione



“Leggere” un albero di decisione

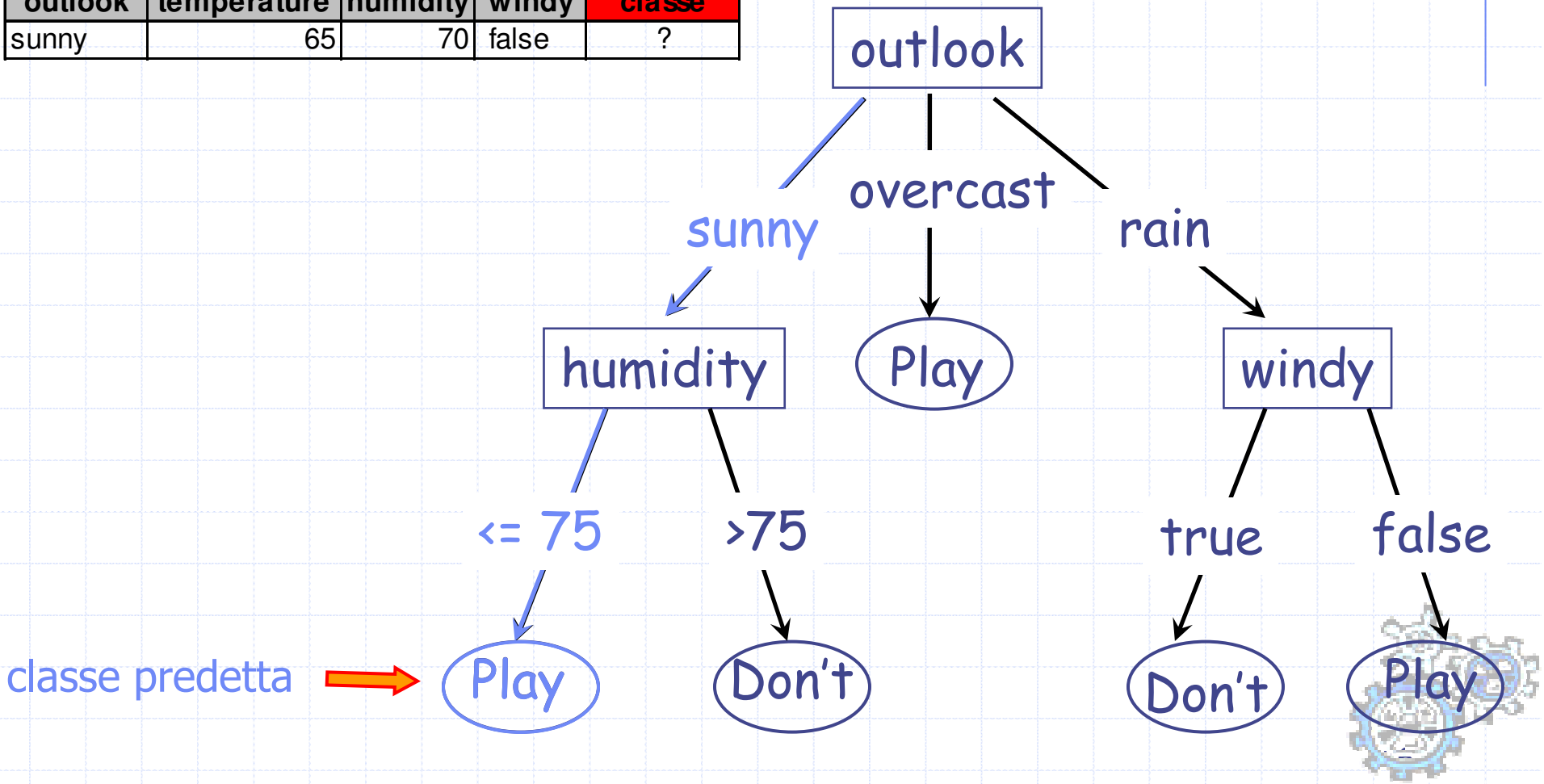
- nodo interno = test su un singolo attributo
- ramo = un possibile risultato del test
- foglia = valore della classe o distribuzione





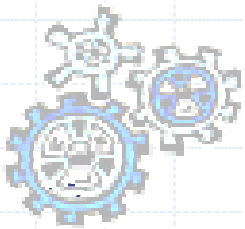
“Usare” un albero a scopo predittivo

outlook	temperature	humidity	windy	classe
sunny	65	70	false	?



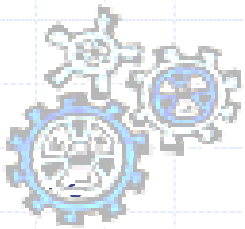
Accuratezza del modello (albero o altro)

- Cerchiamo una misura della bontà di un modello
  - **accuratezza**
    - percentuale di esempi la cui classe predetta coincide con la classe reale
  - **% di misclassificazione**
    - percentuale di esempi la cui classe predetta NON coincide con la classe reale
- … di quali esempi?
  - Di esempi di cui si conosce la classe reale!
  - L'accuratezza sugli esempi del training set è **ottimistica**



Accuratezza: metodo "holdout"

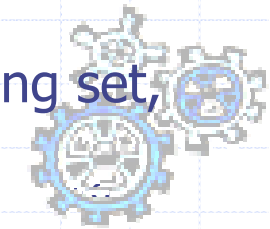
- **Metodo "holdout"**
  - valutare l'accuratezza del modello sul test set
- **Test set**
  - insieme di esempi indipendenti dal training set
  - di ciascun esempio si conoscono gli attributi e la classe reale
  - dato un insieme di esempi lo si divide (casualmente) in una percentuale per il training ed una per il test set (in genere 2/3 per il training e 1/3 per il test set)



Accuratezza: limiti inferiori

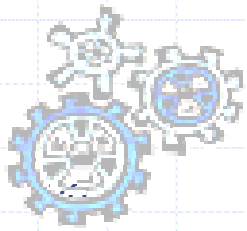
- **Accuratezza di un classificatore casuale**
  - classificatore che in modo casuale classifica in una delle k possibili classi
  - accuratezza = 1 / k * 100
  - esempio: per k = 2 l'accuratezza è del 50%
- **Qualsiasi classificatore deve fare almeno meglio del classificatore casuale!**
- **Accuratezza di un classificatore a maggioranza**
  - classificatore che classifica sempre come la classe di maggioranza nel training set
  - (supponendo la stessa percentuale nella popolazione)

  accuratezza >= 1 / k * 100
  - esempio: con due classi rappresentate al 98% e al 2% nel training set, l'accuratezza è del 98%



Uso di un modello e "score set"

- Un modello **sufficientemente accurato** può essere usato:
  - **a scopo descrittivo**
    - descrivere l'appartenenza ad una classe in base agli attributi
    - Esempi
      - descrivere le caratteristiche dei clienti che usufruiscono di un particolare servizio
      - descrivere le caratteristiche dei clienti raggruppati in un cluster
  - **a scopo predittivo**
    - predirre la classe di un nuovo esempio
    - **score set** = insieme di esempi di cui si conoscono gli attributi **ma non la classe**
    - Esempi
      - selezionare i clienti cui proporre una nuova campagna pubblicitaria
      - predirre la capacità di spesa di un cliente



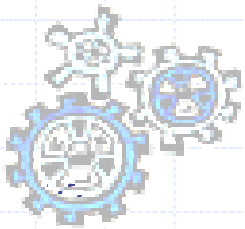
Costruzione di alberi di decisione

Algoritmo di base

- Costruzione top-down
  - Ciascun nodo ha associato un insieme di esempi (al nodo radice è associato tutto il training set)
  - In ogni nodo viene selezionato un singolo attributo
    - Per gli attributi discreti viene creato un figlio per ciascun possibile valore (variante con creazione di soli due figli)
    - Per gli attributi continui vengono creati due figli "<= threshold" e "> threshold" (variante con creazione di piu' di due figli)
  - Gli esempi del nodo vengono ripartiti tra i tra i figli del nodo
- Terminazione
  - (**Minsplit**) Numero di esempi nel nodo inferiore ad un valore fissato
  - (**Purity level**) percentuale di esempi appartenenti ad una classe maggiore di un valore fissato

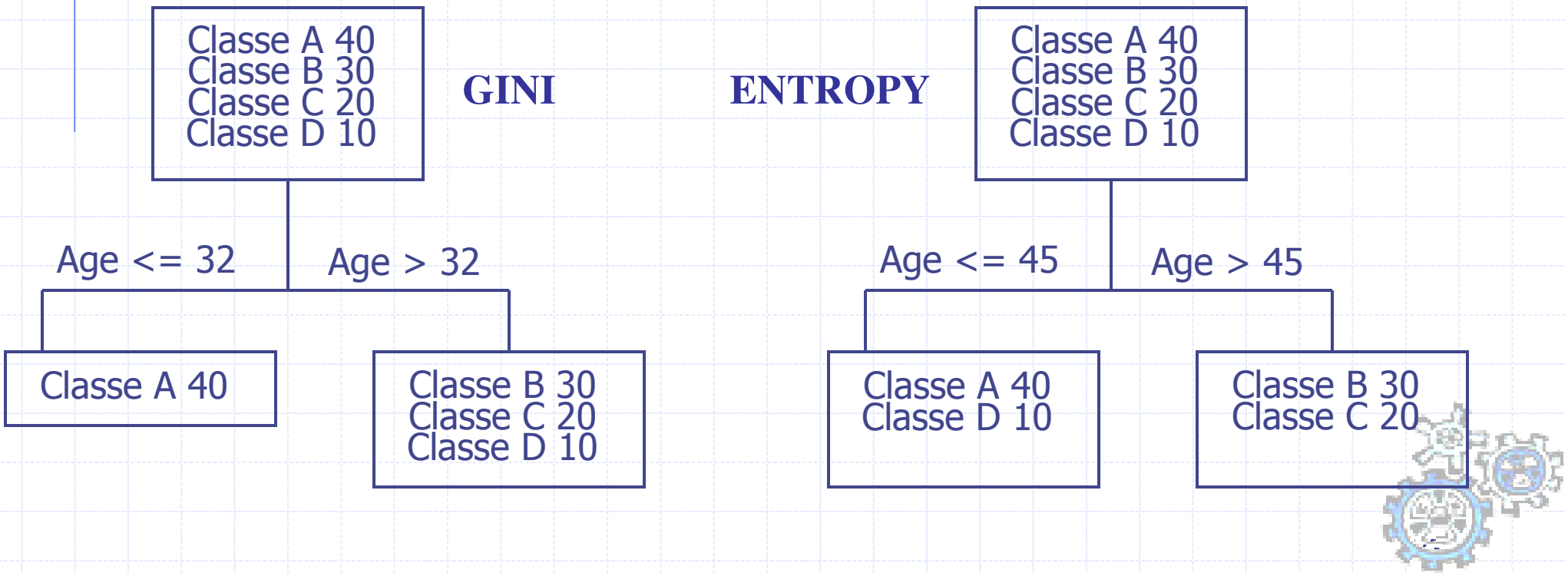
Costruzione di alberi di decisione

- Varianti dell'algoritmo di base:
  - dal **machine learning**: ID3, C4.5, C5.0 (Quinlan 86, 93)
  - dalla **statistica**: CART (Classification and Regression Trees) (Breiman et al 84)
  - da **pattern recognition**: CHAID (Chi-squared Automated Interaction Detection) (Magidson 94)
- Differenze principali
  - criterio di selezione dell'attributo ***(splitting criteria)***
    - **Information gain** (ID3, C4.5, C5.0)
    - **Gini index** -- noto anche come **Quadratic entropy** (CART)
    - **Twoing rule** (CART)
    - **χ2 contingency table statistic** (CHAID)
    - **Normalized distance measure**
    - **Training set error**
    - ... criteri proprietari ...



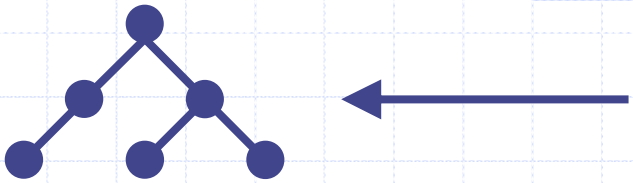
Splitting criteria

- **Obiettivo**: separare gli esempi con classi differenti in base al valore di un singolo attributo
  - Gini tende ad isolare la classe piu' ampia dalle altre classi
  - Entropy tende a isolare gruppi di classi che assommano al 50%



Alberi di decisione con Weka

ID3



- True Positive (TP) rate
  - (# esempi di classe c classificati come c) /(# esempi di classe c)
- False Positive (FP) rate
  - (# esempi di classe non c classificati come c) / (# esempi di classe non c)
- Precision
  - (# esempi di classe c classificati come c) / (# esempi classif. come c)
- Recall (= TP rate)
  - (# esempi di classe c classificati come c) / (# esempi di classe c)

```
outlook = sunny
|   humidity = high: no
|   humidity = normal: yes
outlook = overcast: yes
outlook = rainy
|   windy = TRUE: no
|   windy = FALSE: yes

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

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  1         0          1         1         1        yes
  1         0          1         1         1        no

=== Confusion Matrix ===

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

accuratezza

misclassificazione

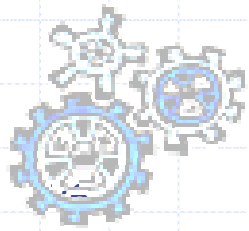
```
=== Confusion Matrix ===

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

Esempi di classe a

Esempi di classe b

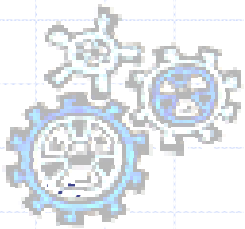
Esempi classificati in classe a

Esempi classificati in classe b



Pos	Neg	← classified as
TP	FN	Pos
FP	TN	Neg

classi reali



Total number of instances: 25 + 175 + 50 +50 = 300

Number of positive instances: FN + TP = 25 + 175 = 200

Number of negative instances: FP + TN = 50 + 50 = 100

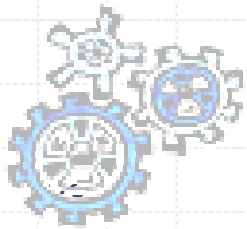
Incorretly classified instances: FN + FP = 50 + 25 = 75

Misclassification: 100 * (FN + FP) / 300 = 25%

Accuracy: 100 * (TN + TP) / 300 = 75%

Pos	Neg	← classified as
175	25	Pos
50	50	Neg

classi reali



TP Rate: 175 / 200 = 0.875

FP Rate: 50 / 100 = 0.5

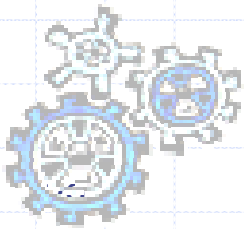
Precision: 175 / 225 = 0.77

Recall: 175 / 200 = 0.875

	Pos	Neg
Pos	175	25
Neg	50	50

← classified as

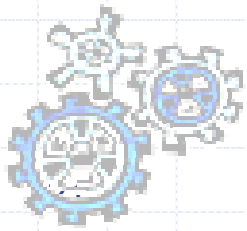
classi reali



Classificazione: concetti avanzati

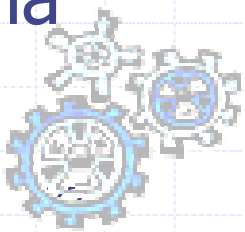
Pesi di Misclassificazione

- Possibilità di assegnare pesi differenti a ciascun errore di misclassificazione
  - Perdere la possibilità di attrarre un cliente che fà molte spese vale di più del rischio di attrarre anche un cliente che fà poche spese
  - Utile quando si voglia premiare la capacità di caratterizzare bene una classe, anche a scapito di un altra



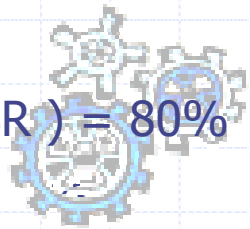
Pesi degli esempi

- Possibilità di assegnare pesi differenti a ciascun esempio del training set
  - Esempio: pesare gli esempi del training set di clienti rispetto al fatturato degli ordini
  - Un cliente pesa in proporzione ai suoi ordini
  - Utile quando si voglia premiare la capacità di pesare ciascun esempio in proporzione al suo "valore", oppure quando ogni esempio corrisponde ad una popolazione di soggetti (dati aggregati).



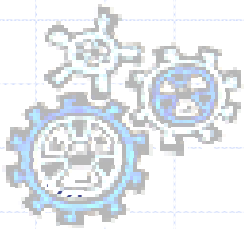
Classi sbilanciate

- Supponiamo due classi A e B
  - Training set: 98% degli esempi in A e 2% in B
    - Classificatore a maggioranza: accuratezza del 98%
  - Troppi pochi esempi di classe B per fare meglio del 98%
- **Oversampling**
  - Selezionare un training set con una distribuzione delle classi maggiormente equilibrata
    - 60-70% per la classe A e 30-40% per la classe B
    - Aumentando il numero di esempi di classe B …o diminuendo quelli di classe A
  - La costruzione del classificatore **ha maggiori esempi** per distinguere B da A
    - Da cui il nome "oversampling"
  - La confidenza nella classificazione di un esempio cambia!
    - Oversampling classe B da 2% al 50%, OSR = 50/2 = 25
    - Confidenza di una predizione B su oversampled CONF = 99%
    - Confidenza non oversampled R_CONF = CONF / (CONF + (1-CONF)*OSR) = 80%
- **Pesi di Misclassificazione**



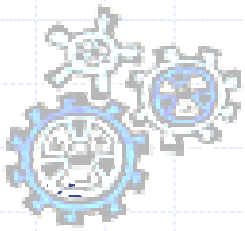
Valori non specificati

- Sia nel training che nei test e score set alcuni esempi possono avere degli **attributi** con valore non specificato
  - Sintassi: cambia da sistema a sistema
    - NULL quando i dati sono in una tabella di un DBMS
    - Stringa vuota oppure '?' quando i dati sono su file di testo
  - Semantica
    - La costruzione dell'albero può
      - Escludere tali esempi
      - Rimpiazzare i valori non specificati
      - **Tenerli in conto con qualche peso**
    - La determinazione della classe di un esempio può
      - Rimpiazzare i valori non specificati
      - **Tenerli in conto con qualche peso**



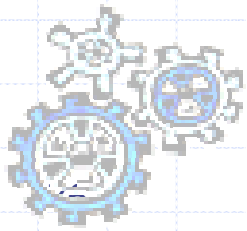
Overfitting: problema

- Obiettivo ideale della classificazione
  - trovare un modello che sia accurato sia sul training set che sugli insiemi di test e di score
- Problema dell'overfitting
  - Un albero di decisione, però può descrivere bene aspetti **troppo specifici** degli esempi del training set e non generalizzabili al test e score set
  - Sorgenti di overfitting:
    - selezione random del training set
    - rumore nei dati (noise)
    - dati al limite (outliers)
    - pochi esempi di training
    - limitazioni dell'algoritmo di costruzione dell'albero



Overfitting: euristiche

- Soluzioni?
  - Rasoio di Occam
    - A parità di accuratezza, meglio un classificatore "semplice" di uno complicato
  - Due euristiche per implementare il rasoio di Occam:
    - **Stop earlier**: fermare la crescita dell'albero durante la costruzione
    - **Post prune** (o semplificazione): costruire interamente l'albero e poi **semplificarlo** tagliando alcuni rami
    - Integrazione delle due
  - Il **post-pruning** produce risultati migliori
    - Richiede un maggior costo computazionale

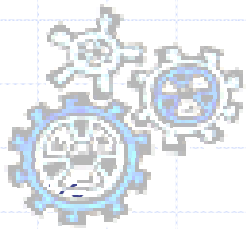


Accuratezza: "stratified holdout"

- **Metodo "stratified holdout"**
  - la divisione casuale di esempi in training e test set potrebbe generare un training set non rappresentativo della popolazione
    - proporzione delle classi diversa nel training e nel test set
    - esempio limite: tutti gli esempi con classe X nel training e quelli con classe Y nel test set
  - "stratified holdout" consiste in una divisione che mantenga nel training e test set la stessa distribuzione delle classi

Accuratezza: "stratified cross-validation"

- **Metodo "stratified n-fold cross-validation"**
  - Si dividono gli esempi in n insiemi S_1, ..., S_n delle stesse dimensioni e con stessa distribuzione delle classi
  - Per ciascuno insieme S_i
    - Si costruisce un classificatore sugli altri n-1 insiemi
    - Si usa l'insieme S_i come insieme di test del classificatore
  - L'accuratezza finale è data dalla **media** delle accuratezze degli n classificatori
  - Metodo standard: n = 10, stratified 10-fold cross validation

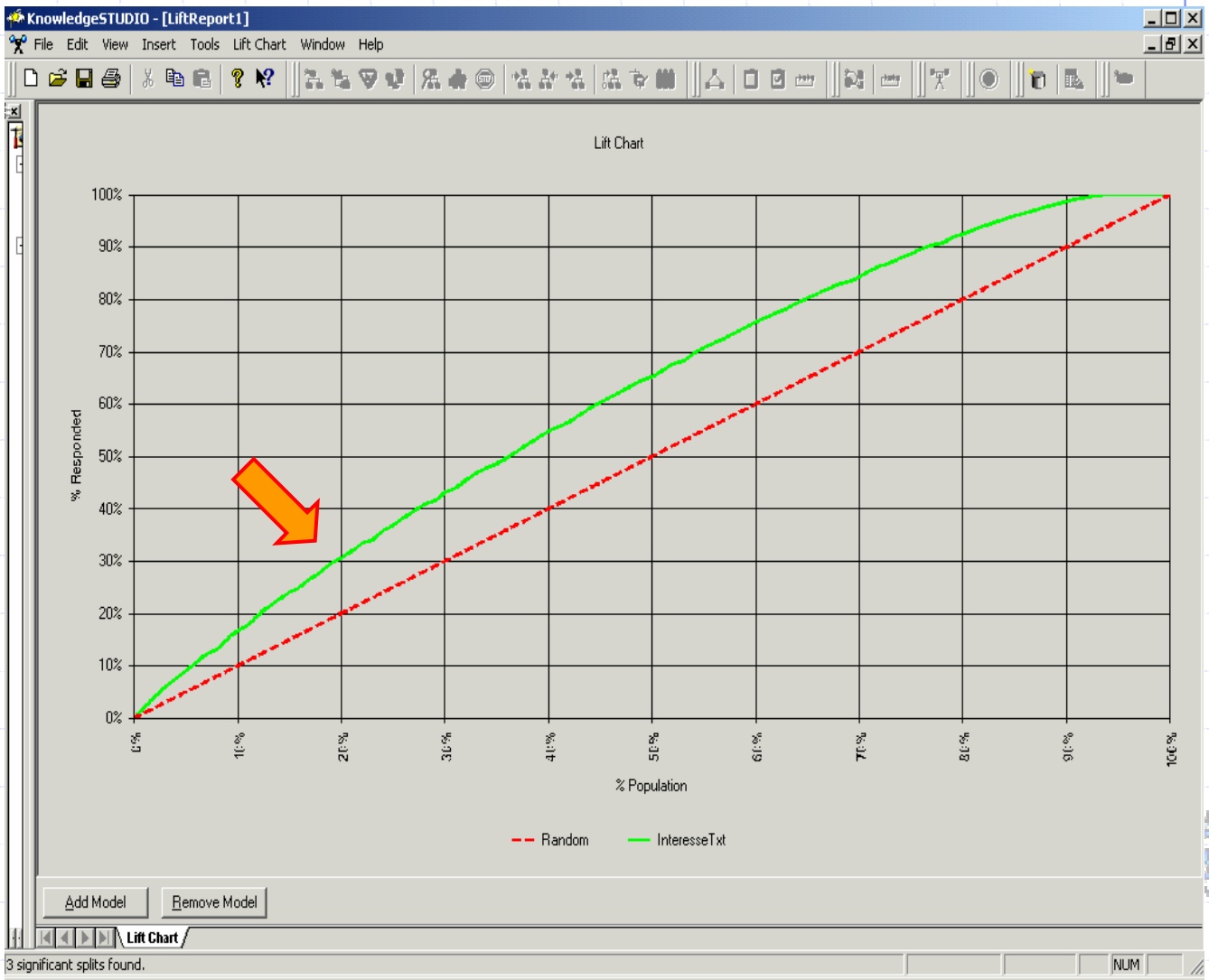


Lift report

- Quasi sempre un modello associa ad ogni classificazione una percentuale di confidenza
- Ordiniamo gli esempi del test set in base alla confidenza di essere in una determinata classe A
- Lift report
  - Sull'asse delle X consideriamo la % (rispetto al test set) di esempi secondo l'ordine precedente
  - Sull'asse delle Y consideriamo la % (rispetto al test set) degli esempi in X che hanno classe reale A
  - La bisettrice rappresenta la performance del classificatore random
- Uso
  - Stima del numero di esempi da selezionare nello score set per ottenere una determinata percentuale di esempi con classe A

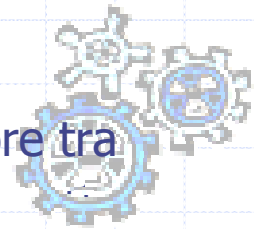
Lift report

- Con il 20% degli esempi si catturano il 30% di tutti coloro che hanno classe 'interesse alto'
- Lift di 1.5 rispetto al classificatore random



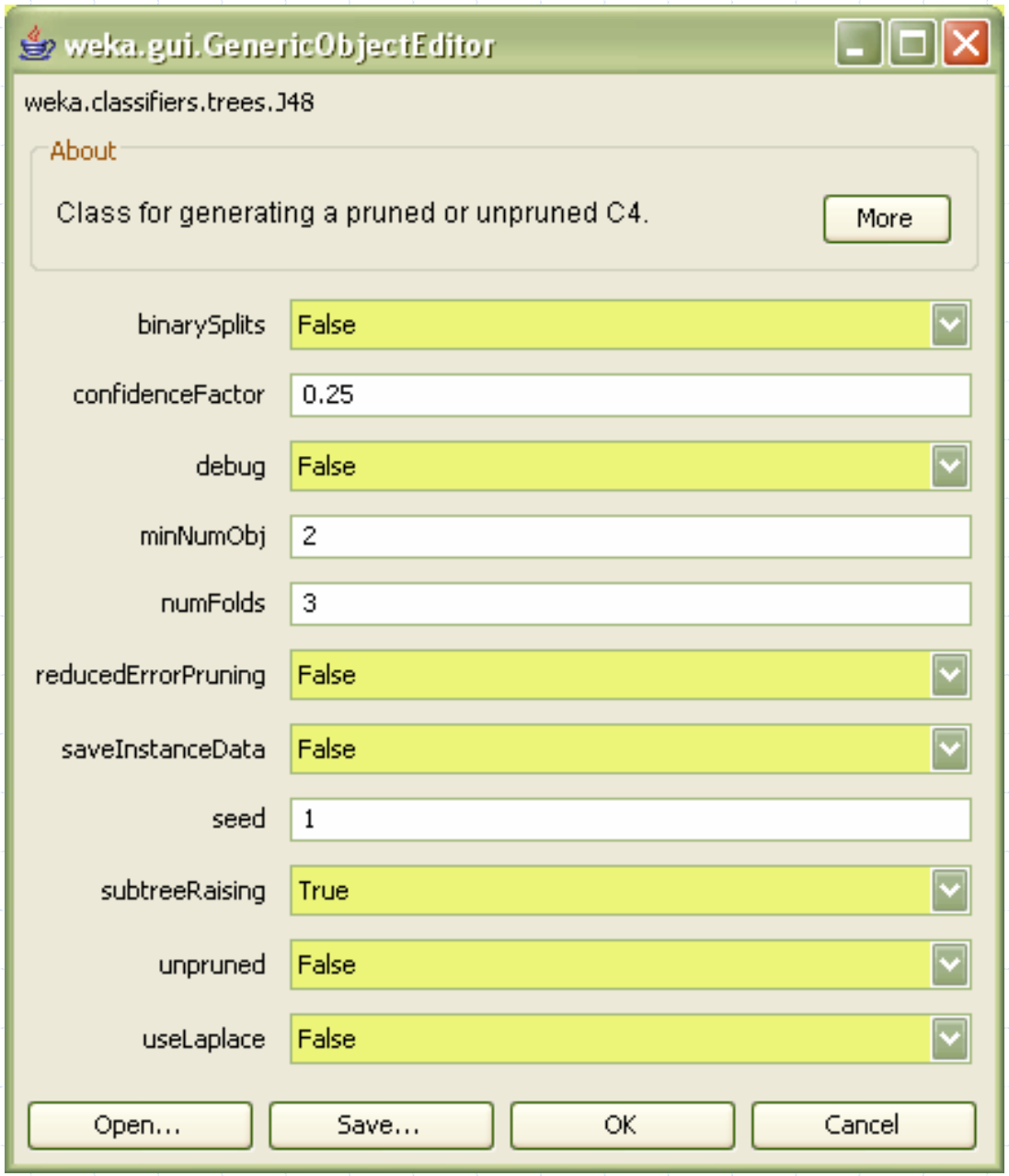
Incrementare l'accuratezza

- Costruendo due o più alberi
  - Su training set diversi
    - **Bagging**
      - costruisce alberi su training set della stessa dimensione ottenuti mediante selezione (con rimpiazzamento) da un insieme di esempi di partenza
      - classificazione mediante votazione a maggioranza
  - Sullo stesso training set, ma ogni volta con pesi diversi
    - **Boosting**
      - costruisce alberi rimodellando i pesi degli esempi misclassificati dagli alberi già costruiti
      - classificazione mediante votazione pesata rispetto all'accuratezza
  - Meta-learning
    - **Stacking**
      - costruisce un albero in grado di predirre quale predizione sia migliore tra quelle di due o più altri alberi



Classificazione con Weka

J48

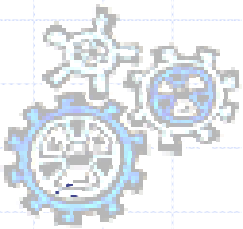


Split binario o no

Più basso → Più pruning

Numero minimo di istanze per foglia

Numero di fold = n → n-1 per training, 1 per pruning

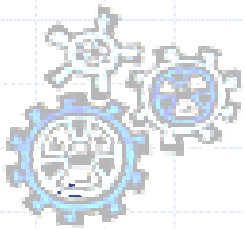


Esercitazione

Esercitazione

- Dataset: energiaelettrica.txt
- Dataset dei clienti di una azienda fornitrice di energia elettrica (in Germania dal 1999 il mercato è libero).
- Obiettivo: individuare tra i clienti quelli che probabilmente cancelleranno il loro contratto il prossimo anno.
- Questi clienti riceveranno un' offerta speciale sul prezzo dell'energia fornita al fine di trattenerli.

- Target: "Canceler" (yes, no)

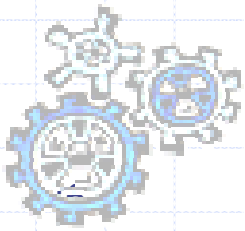


Esercitazione

- Attenzione!!!!

  Se il vostro classificatore sbaglia ridurrete il prezzo ad un cliente che non aveva intenzione di cambiare fornitore.

	Canceler	Cliente Fedele
Cliente riceve offerta	43,80 Euro	66,30 Euro
Cliente non riceve offerta	0 Euro	72,00 Euro

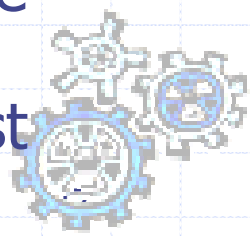


Esercitazione

1. Costruire un albero di decisione per individuare il profilo del cliente canceler, senza utilizzare i pesi di misclassificazione.
   Cosa succede? Perché?

2. Costruire un nuovo albero di classificazione utilizzando gli opportuni pesi di misclassificazione.
   Cosa succede?

3. Partizionare il dataset in training-set e test-set mantenendo la distribuzione iniziale dei dati, costruire un albero di decisione sul training-set utilizzando gli opportuni pesi di misclassificazione e validarlo sul test set.



Esercitazione

4. Costruire e testare più alberi cambiando:
   - La distribuzione delle classi (yes, no) nell'insieme di allenamento.
   - Il livello di pruning.
   - Il numero di casi minimi per nodo.
   - Applicando il boosting.

5. Qual è il vostro miglior modello?