# Introduzione al Data Mining

## e applicazioni al dominio del Contact Management

*Parte* II*: principali modelli per il Supervised Learning*

Andrea Brunello

Università degli Studi di Udine



*In collaborazione con dott. Enrico Marzano, CIO Gap srl*
*progetto Active Contact System*

# Contenuti

Parte II:

- principali modelli per il *Supervised Learning*:
  - alberi di decisione;
  - regole di classificazione;
  - Instance-based learning;
  - regressione (lineare);
  - Naive Bayes.

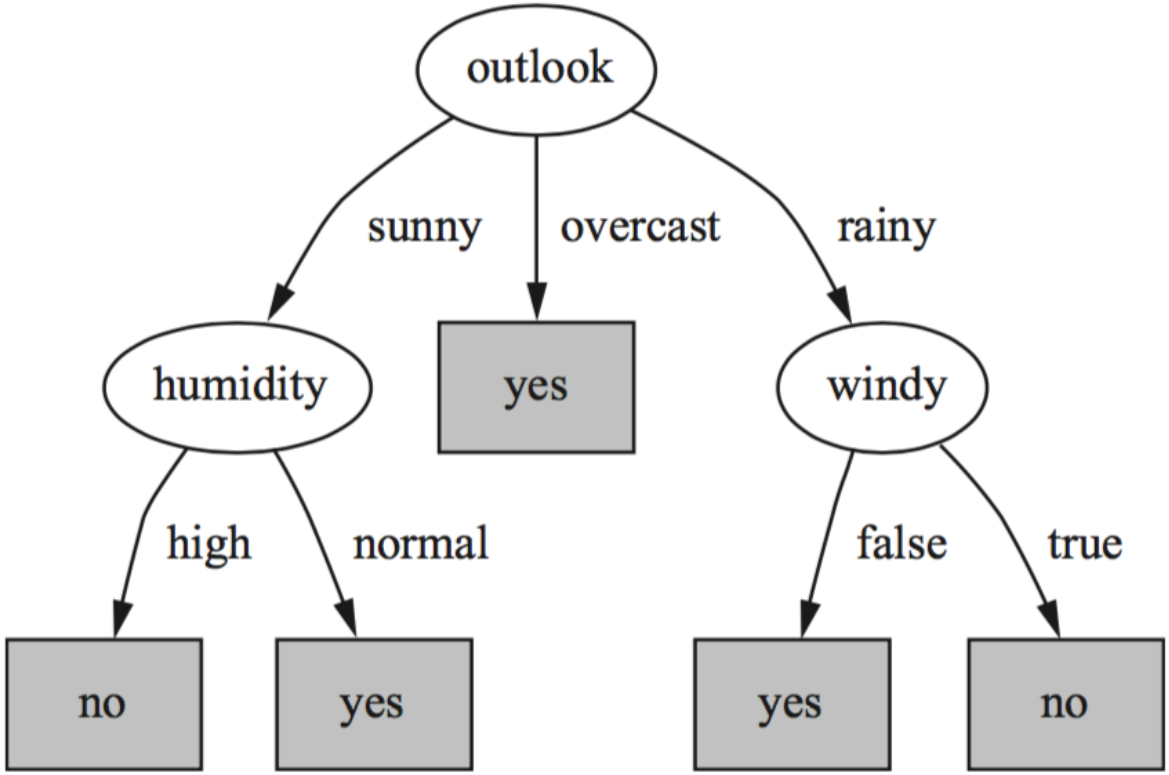
# Alberi di decisione

Metodo semplice e compatto per rappresentare l'output di un processo di **classificazione**.

- ciascun nodo interno corrisponde alla valutazione di determinato attributo;
- la classificazione di un'istanza avviene partendo dalla radice, e scendendo via via sino a giungere ad una foglia etichettata con una determinata classe (o insieme di classi, distribuzione di probabilità);
- variante per problemi di regressione: *alberi di regressione*.

# Alberi di decisione (2)

L'albero di decisione classifica correttamente tutte le istanze del *Weather Problem*.




Osserviamo che non viene mai presa in considerazione la *temperatura*.

# Costruzione di un albero di decisione

Presentiamo una metodologia generale per la creazione di un albero di decisione, simile a quella adottata da *ID.3*:

- costruzione dell'albero ricorsiva, partendo dalla radice;
- ad ogni passo si sceglie l'attributo su cui effettuare lo split, intuitivamente quello che porta al maggior *information gain* (alberi più bassi, metodo *greedy*);
- sino alla generazione di una foglia, in corrispondenza di un caso base:
  - gli elementi nel nodo hanno stessa classe (attenzione all'overfitting, *i.e.* modello costruito ad-hoc sui dati, che non generalizza), oppure
  - si è raggiunto un sufficiente grado di purezza del nodo .

# Costruzione di un albero di decisione (2)

Come misurare il guadagno di informazione dato dallo split su un attributo?

- insieme $T$ di istanze partizionate nelle classi $C = \{C_1, \ldots, C_k\}$ dall'attributo obiettivo;
- distribuzione di probabilità associata a $T$:

$$P = (|C_1|/|T|, |C_2|/|T|, ..., |C_k|/|T|)$$

- definiamo l'informazione necessaria ad identificare la classe di un elemento di $T$ come:

$$Info(T) = H(P) = -\sum_{i=1}^{k} p_i * log(p_i)$$

- intuitivamente, se quasi tutti gli elementi fanno parte di una stessa classe, $Info(T)$ è bassa, ed aumenta all'aumentare della "confusione" (entropia).

# Costruzione di un albero di decisione (3)

Cosa succede partizionando l'insieme di istanze sulla base di un attributo?

- supponiamo di suddividere $T$ in sottoinsiemi $T_1, \ldots, T_n$ sulla base del valore di una delle *feature*, diciamo $X$;
- l'informazione necessaria ad identificare la classe di un elemento di $T$ è la media pesata dell'informazione necessaria ad identificare la classe dell'elemento all'interno di ciascun sottoinsieme:

$$Info(X, T) = \sum_{i=1}^{n} (|T_i|/|T|) * Info(T_i)$$

# Costruzione di un albero di decisione (4)

Diamo infine la definizione di *information gain*:

$$Gain(X, T) = Info(T) - Info(X, T)$$

- rappresenta la differenza fra l'informazione necessaria ad identificare la classe di un elemento di $T$ e l'informazione necessaria dopo la suddivisione di $T$ in sottoinsiemi attraverso l'attributo $X$;
- in altre parole, il guadagno d'informazione dovuto all'attributo $X$ (alto è meglio).

# Costruzione di un albero di decisione (5)

Considerazioni finali:

- utilizzare l'information gain può portare a preferire la scelta di attributi con un gran numero di valori (*highly branching attributes*);
- il che porta spesso a sua volta a fenomeni di overfitting;
- l'*information gain ratio* considera anche il numero e la dimensione dei vari sottoinsiemi che verrebbero generati.

Oltre all'entropia, esistono altre misure di impurità dei nodi.

- indice di impurità di Gini;
- se l'attributo da predire è numerico, si può utilizzare *RMSE* rispetto alla media del nodo;
- ...

# Alberi di decisione: algoritmi

Diversi algoritmi per la costruzione degli alberi:

- *ID.3, Iterative Dichotomizer 3* (attributi nominali);
- *C4.5* (implementazione Weka *J48*);
- *C5.0*;
- *CART, Classification And Regression Trees*;
- ...

# Regole di classificazione

Alternativa rispetto agli alberi di decisione.

$$\begin{aligned} \textit{Condizioni} = \textit{soleggiato} \wedge \textit{Umidita} = \textit{alta} &\rightarrow \textit{Si\_gioca} = \textit{no} \\ \textit{Condizioni} = \textit{pioggia} \wedge \textit{Vento} = \textit{vero} &\rightarrow \textit{Si\_gioca} = \textit{no} \\ \textit{Condizioni} = \textit{nuvoloso} &\rightarrow \textit{Si\_gioca} = \textit{si} \\ \textit{Umidita} = \textit{normale} &\rightarrow \textit{Si\_gioca} = \textit{si} \\ \textit{Else Si\_gioca} &= \textit{si} \end{aligned}$$

- parte a sinistra: precondizioni
- parte a destra: classe dell'attributo su cui si sta facendo predizione
- tipicamente intese per essere considerate in ordine
- derivabili da un albero di decisione, o costruzione ad-hoc es. tramite il *metodo delle coperture*

# Instance-based learning

A differenza che nei metodi precedenti non vi è una fase iniziale di “training”, l’idea è la seguente:

- si ha a disposizione un insieme di istanze delle quali è nota la classe;
- data una nuova istanza, essa viene ricondotta ad uno o più dei casi noti:
  - *nearest neighbour*
  - *k-nearest neighbour*
- diverse nozioni di “distanza”;
- l’output può essere qualitativo o quantitativo.

# Modelli di regressione lineare

Output numerico (*variabile dipendente*), sulla base di attributi in input numerici (*variabili indipendenti*).

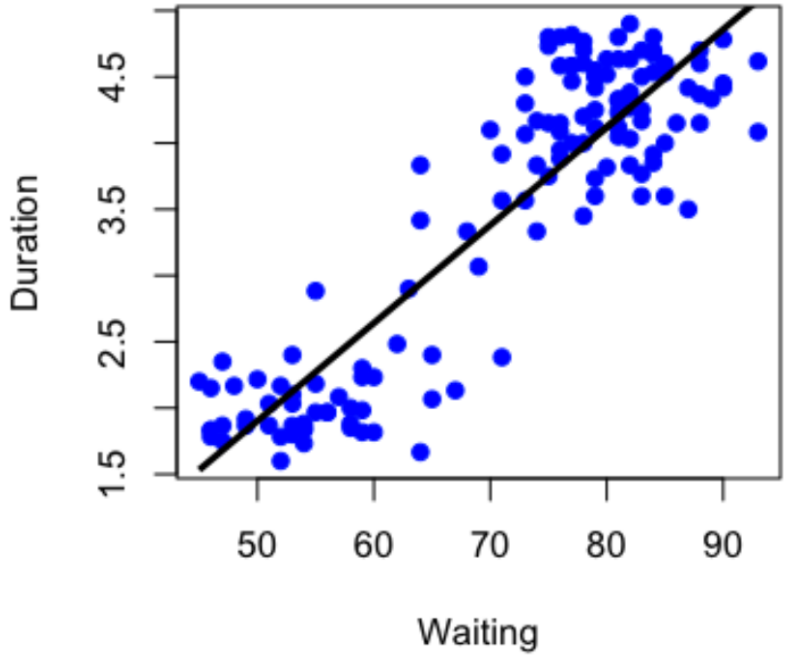
- si vuole esprimere la variabile dipendente come combinazione di una o più variabili indipendenti:

$$x = w_0 + w_1 a_1 + w_2 a_2 + ... + w_k a_k$$

- metodo semplice e di intuitiva interpretazione, tuttavia ha delle limitazioni:
  - la relazione fra variabile dipendente ed indipendenti deve essere lineare;
  - le variabili indipendenti non devono essere "correlate" fra loro (no *multicollinearità*);
  - altre condizioni di applicabilità: `https://en.wikipedia.org/wiki/Linear_regression`.

# Modelli di regressione lineare (2)

| Durata eruzione | Tempo attesa |
|---|---|
| 2.883 | 55 |
| 1.883 | 54 |
| 2.167 | 52 |
| 1.600 | 52 |
| 1.750 | 47 |
| 1.967 | 55 |



# Modelli di regressione lineare (3)

Esistono varianti del modello di regressione (ed ancora più algoritmi che le implementano):

- *regressione polinomiale*: per relazioni non lineari
- *regressione multivariata*: per predire il valore di più di una variabile
- *regressione logistica*: per classificazione binaria
- *regressione logistica multinomiale* e *analisi discriminante lineare*: per classificazione multiclasse

# Naive Bayes

I Naive Bayes sono una famiglia di classificatori probabilistici, basati sul teorema di Bayes della probabilità condizionata:

- l'idea alla base è che ciascuna feature nel dataset contribuisca in modo indipendente, e con lo stesso peso, alla determinazione della classe dell'istanza;
- l'assunzione è molto forte;
- tuttavia, il metodo Naive Bayes funziona sorprendentemente bene nei casi reali.

# Naive Bayes (2)



Tabella con dati riassuntivi sul *Weather Problem*:

**Table 4.2** Weather Data with Counts and Probabilities

| **Outlook** | | | **Temperature** | | | **Humidity** | | | **Windy** | | | **Play** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ***yes*** | ***no*** | | ***yes*** | ***no*** | | ***yes*** | ***no*** | | ***yes*** | ***no*** | ***yes*** | ***no*** |
| sunny | 2 | 3 | hot | 2 | 2 | high | 3 | 4 | false | 6 | 2 | 9 | 5 |
| overcast | 4 | 0 | mild | 4 | 2 | normal | 6 | 1 | true | 3 | 3 | | |
| rainy | 3 | 2 | cool | 3 | 1 | | | | | | | | |
| sunny | 2/9 | 3/5 | hot | 2/9 | 2/5 | high | 3/9 | 4/5 | false | 6/9 | 2/5 | 9/14 | 5/14 |
| overcast | 4/9 | 0/5 | mild | 4/9 | 2/5 | normal | 6/9 | 1/5 | true | 3/9 | 3/5 | | |
| rainy | 3/9 | 2/5 | cool | 3/9 | 1/5 | | | | | | | | |

Supponiamo ora di voler classificare una nuova istanza:

| **Condizioni** | **Temp.** | **Umidità** | **Vento** | **Si_gioca** |
|---|---|---|---|---|
| soleggiato | fredda | alta | vero | *?* |

# Naive Bayes (3)

Come classificare la nuova istanza?

- otteniamo la “tendenza” a giocare moltiplicando le frazioni corrispondenti:

$$(2/9) * (3/9) * (3/9) * (3/9) * (9/14) = 0.0053$$

- allo stesso modo otteniamo la “tendenza” a non giocare:

$$(3/5) * (1/5) * (4/5) * (3/5) * (5/14) = 0.0206$$

- dunque, la tendenza a non giocare è circa 4 volte superiore a quella di farlo;
- possiamo trasformare i numeri in probabilità:
  - a favore: $0.0053/(0.0053 + 0.0206) = 20.5\%$
  - contro: $0.0206/(0.0053 + 0.0206) = 79.5\%$

# Naive Bayes (4)

Metodo basato sul Teorema di Bayes:

$$Pr[H|E] = (Pr[E|H] * Pr[H])/Pr[E]$$

Nel nostro caso:

- $H$: corrisponde al fatto di giocare o meno;
- $E$: corrisponde alla combinazione dei valori delle 4 feature $E_1, E_2, E_3, E_4$

Sostituendo nella formula otteniamo, facendo uso dell'ipotesi di indipendenza delle variabili:

$$\begin{aligned} & Pr[si|E] \\ =& (Pr[E_1|si] * Pr[E_2|si] * Pr[E_3|si] * Pr[E_4|si] * Pr[si])/Pr[E] \\ =& [(2/9) * (3/9) * (3/9) * (3/9) * (9/14)]/Pr[E] \end{aligned}$$

Il denominatore scompare effettuando la normalizzazione dei valori, ottenendo lo stesso risultato di prima.

# Riferimenti


I. H. Witten, E. Frank, M. A. Hall: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 3rd edition, 2011.

G. James, D. Witten, T. Hastie, R. Tibshirani: *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.