BAYESIAN CLASSIFIER

È un classificatore di tipo statistico, basato sul teorema di Bayes.

Supponiamo che X sia una tupla del nostro dataset la cui classe etichetta è sconosciuta.

Supponiamo che H sia l'ipotesi che X appartenga alla classe etichetta C.

Definiamo un po' di aspetti:

PROBABILITÀ A POSTERIORI: $P(H|X)$

È la probabilità che l'ipotesi H sia vera data la tupla osservata X.

PROBABILITÀ A PRIORI DI H: $P(H)$

Probabilità generica che l'ipotesi si verifichi, ad esempio può essere la probabilità che un cliente compri un PC, qualsiasi sia il cliente.

PROBABILITÀ A PRIORI DI X: $P(X)$

Probabilità che la tupla X venga osservata.

PROBABILITÀ A POSTERIORI X CONDIZIONATO H: $P(X|H)$

È la probabilità che, data l'ipotesi già verificata, essa si verifichi per la tupla X.

Dato che un cliente abbia acquistato un PC, con che probabilità quel cliente sia proprio X?

$P(H)$, $P(x)$ e $P(X|H)$ possono essere stimati dal dataset. $P(H|X)$ dal teorema di Bayes:

$$P(H|x) = \frac{P(X|H)P(H)}{P(x)}$$

Possiamo predire la classe di X con C_i sse la probabilità $P(C_i|X)$ è la maggiore rispetto alle probabilità di tutte le altre classi $P(C_k|X)$.
Quindi dobbiamo scegliere la classe che massimizza

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}$$

Considerando che $P(X)$ è l'elemento costante, ciò che fa variare la probabilità è il numeratore, il quale va massimizzato.

Sotto l'assunzione che gli attributi siano condizionalmente indipendenti possiamo calcolare

$$P(X|C_i) = \prod_{K=1}^{n} P(x_k|C_i)$$

Questo riduce il costo computazionale, perché conta solo la distribuzione delle classi fra le tuple del DB.
Se prendiamo un valore A_k di un attributo discreto, $P(x_k|C_i)$ è il numero di tuple che in X_k hanno valore A_k e classe C_i, diviso la cardinalità $|C_i, D|$.

$$P(x_k|C_i) = \frac{|x_k = A_k, C_i, D|}{|C_i, D|}$$

Se invece prendiamo A_k continuo, $P(x_k|C_i)$ è tipicamente stimato usando la distribuzione normale (Gaussiana)

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

e quindi $P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$.

Esempio:

RID	age	income	student	credit_rating	buy_computer
1	youth	high	no	fair	no
2	youth	high	no	exelent	no
3	middle	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	exelent	no
7	middle	low	yes	exelent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	exelent	yes
7	middle	low	yes	exelent	yes
7	middle	low	yes	exelent	yes
14	senior	medium	no	exelent	no

Consideriamo la seguente osservazione X:

X = AGE = youth, INCOME = medium, STUDENT = yes, CREDIT_R = fair

Iniziamo calcolando $P(C_i) \quad \forall i \in (1,2)$

$P(C_1 = yes) = 9/14 = 0,643$

$P(C_2 = no) = 5/14 = 0,357$

Ora dobbiamo calcolare $P(X|C_i) \quad \forall A \in X$ e $\forall C_i$ usando la formula:

$$P(x_k|C_i) = \frac{|x_k = A_k, C_i, D|}{|C_i, D|}$$

$$P(\text{AGE} = \text{"youth"} \mid \text{"yes"}) = 2/9 = 0{,}222$$

RID	age	income	student	credit_rating	buy_computer
9	youth	low	yes	fair	yes
11	youth	medium	yes	exelent	yes
1	youth	high	no	fair	no
2	youth	high	no	exelent	no
8	youth	medium	no	fair	no
5	senior	low	yes	fair	yes
6	senior	low	yes	exelent	no
4	senior	medium	no	fair	yes
10	senior	medium	yes	fair	yes
14	senior	medium	no	exelent	no
3	middle	high	no	fair	yes
13	middle	high	yes	fair	yes
7	middle	low	yes	exelent	yes
12	middle	medium	no	exelent	yes

Calcolando le alta probabilità sulla base dell'osservazione X:

P(age = "youth" | buys_computer = "yes") = 2/9 = 0.222
P(age = "youth" | buys_computer = "no") = 3/5 = 0.6
P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

Calcoliamo ora $P(X \mid C_i) \; \forall C_i \in (1,2)$

$$P(X \mid \text{"yes"}) = \prod P(x_i \mid \text{"yes"}) = 0{,}222 \cdot 0{,}444 \cdot 0{,}667 \cdot 0{,}667 = 0{,}044$$

$$P(X \mid \text{"no"}) = 0{,}6 \cdot 0{,}4 \cdot 0{,}2 \cdot 0{,}4 = 0{,}019.$$

Possiamo finalmente usare il teorema di Bayes:

$$P(X \mid C_i) \cdot P(C_i) = 0{,}044 \cdot 0{,}643 = 0{,}028 \;\} \text{"yes"}$$
$$P(X \mid C_i) \cdot P(C_i) = 0{,}019 \cdot 0{,}357 = 0{,}07 \;\} \text{"no"}$$

Quindi la tupla X viene classificata come "yes"

PROBLEMA

Questo tipo di classificatore, necessita che le varie probabilità interpellate non siano zero, altrimenti si annulla tutto.

Per ovviare a questo inconveniente si può aggiungere un caso a ogni conto, se per esempio abbiamo per un attributo di genere A_1 con classe C_1 zero esempi nel DB, possiamo aggiungere 1 a tutte gli esempi dell'attributo generale A.

Esempio

1000 tuple con un attributo INCOME:

$$\begin{cases} income = \text{"medium"} \end{cases} : 990 \text{ tuple}$$
$$\begin{cases} income = \text{"low"} \end{cases} : 0 \text{ tuple}$$
$$\begin{cases} income = \text{"high"} \end{cases} : 10 \text{ tuple}$$

Dato che la P("low") sarebbe 0/1000, per evitarlo aggiungiamo un esempio a tutti i casi

$$\begin{cases} income = \text{"medium"} \end{cases} : 991 \text{ tuple}$$
$$\begin{cases} income = \text{"low"} \end{cases} : 1 \text{ tuple}$$
$$\begin{cases} income = \text{"high"} \end{cases} : 11 \text{ tuple}$$

Così da non cambiare troppo la probabilità e non avere $P(A_k) = 0$.

VANTAGGI

1) Facile
2) Buoni risultati.

SVANTAGGI

1) Assumere che gli attributi sono indipendenti.
   Nella maggior parte dei casi esistono dipendenze.