

DECISION TREE

È uno dei più diffusi classificatori, grazie alla sua facile interpretazione (Anche umana).

In un DT :

1) **Nodi Interni**: devono il test da fare a un specifico attributo di una tupla in ingresso, per far sì che la classificazione delle tuple stesse passi per quel branch.

2) **BRANCH**: è l'outcome del test svolto su un input. Come arriva alla labeled-class.

3) **Nodi Foglia**: Contengono le classi etichetta, target per un determinato branch.

ALGORITMO GREEDY PER LA COSTRUZIONE DI UN DT.

La prima idea è quella di prendere la decisione migliore ogni volta bisogna scegliere su quale feature bisogna fare lo split del DB.

1) All'inizio **tutte** le tuple (o esempi) sono nella **root**.

2) È necessario che gli attributi siano **categorici**.

3) Gli esempi vengono divisi a seconda di determinati valori degli attributi.

4) Per prendere la decisione su quale feature splittare il DB, si utilizzano delle metriche (ATTRIBUTES SELECTION FEATURES)

5) **Stopping Condition**:

- Tutti gli split hanno un'unica classe.

- Sono finiti gli attributi oppure le tuple.

ATTRIBUTES SELECTION MEASURES

Sono delle heuristiche utili a capire dove conviene fare lo split per ~~classi~~ individuare i ~~dataset~~ D.

GOAL:

Dividere D in sottoinsiemi di tuple aventi la stessa labeled-class.

Bisogna trovare un criterio per capire come dividere il dataset.

INFORMATION GAIN

Serve a selezionare il miglior attributo su cui fare la divisione di D.

Viene scelto l'attributo che ha il valore maggiore di $GAIN(A_i)$.

Partiamo nell'analizzare di quanto informazione abbiamo bisogno per classificare una tupla in D.

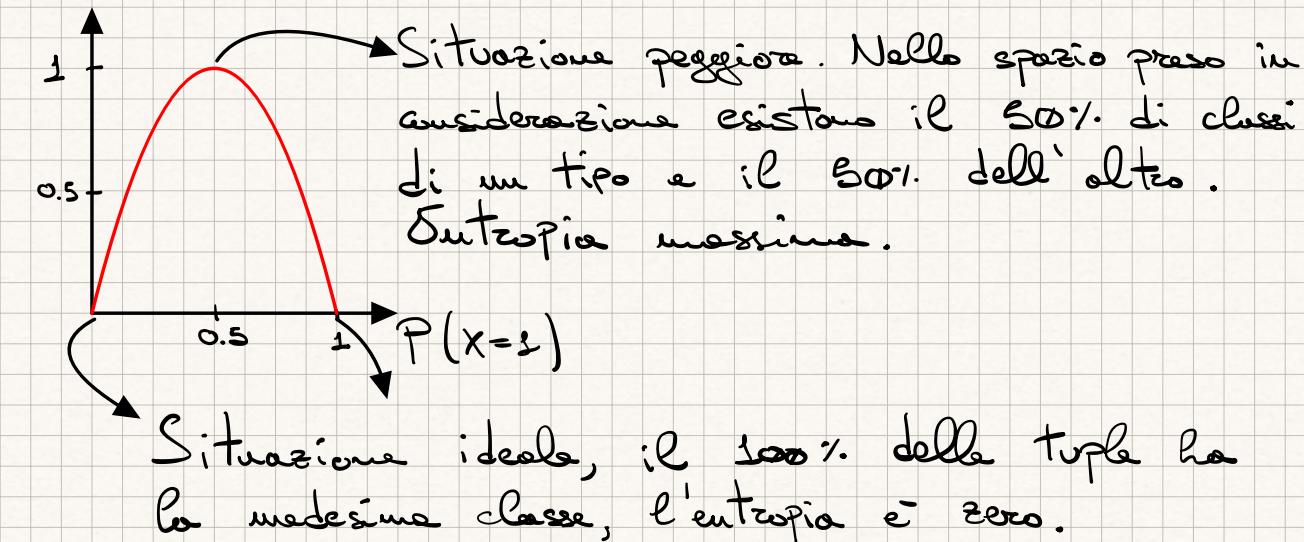
$$Info(D) = - \sum_{i=1}^n p_i \log(p_i)$$

Dove p_i è la probabilità che una tupla arbitraria in D abbia la classe C_i

$$p_i = \frac{|C_{i,D}|}{|D|}$$

Questa formula è anche la formula dell'entropia, e ci dà informazione su quanti tipi di classi etichetta sono presenti nel DB o nella porzione di DB che stiamo analizzando. Meno sono i tipi, più è facile classificare. O ancora, più c'è una classe di maggioranza più è facile classificare, ma questo è da intendersi nel singolo split.

Grafico entropia: per una classe binaria



Difatti $\text{Info}(D)$ ci dice quanta informazione è necessaria per identificare una classe etichetta di una tupla di D .

Ora, supponiamo di dividere D secondo i valori di un attributo A avente v diversi valori $\{a_1, \dots, a_v\}$. Il risultato sarà:

$$\{D_1, D_2, \dots, D_v\}$$

Idealmente vorremmo che queste partizioni siano pure, ovvero che abbiano all'interno la stessa class - label. Questo però non è quasi mai vero.

Quanta informazione necessitiamo per avere una classificazione esatta? La risposta è data dalla seguente metriza:

$$\text{Info}_A(D) = \sum_{j=1}^{\text{v}} \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$

Più è piccolo $\text{Info}_A(D)$ più forte lo split su A porta ad avere partizioni pure.

Il valore $\frac{|D_j|}{|D|}$ fa da peso per il j-esimo set (D_j)

In altre parole $\text{Info}_A(D)$ ci dà indicazione sulla quantità di informazione richiesta per classificare una tupla di D partizionando per A.

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

Iniziamo calcolando $\text{Info}(D)$:

$$\text{Info}(D) = - \left(\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14} \right) = 0.94 \text{ bits}$$

Continuiamo utilizzando ancora $\text{Info}(D)$ per i due sottoinsiemi trovati, questa volta usando AGE tenendo in considerazione sempre la classe labeled.

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

Dividiamo il dataset per età, all'interno di ogni età vediamo chi ha o non ha comprato un PC.

$$\text{Info}_{\text{AGE}}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2)$$

Dove:

$\frac{5}{14} \Rightarrow$ numero di youth e così per le altre età

$I(2,3) =$ Nella classe "youth" 2 hanno comprato PC
e 3 no

$$\text{Info}_{\text{AGE}}(\Delta) = \emptyset, 694 \quad 1$$

$$\text{Info}_{\text{Incomes}}(\Delta) = \emptyset, 911 \quad 4$$

$$\text{Info}_{\text{student}}(\Delta) = \emptyset, 789 \quad 2$$

$$\text{Info}_{\text{Gender}}(\Delta) = \emptyset, 892 \quad 3$$

GAIN

L'information gain è tradotto:

$$\text{GAIN}(A) = \text{Info}(\Delta) - \text{Info}_A(\Delta)$$

Ora c'è il guadagno nel dividere Δ per A .

Si sceglie l'attributo con $\text{GAIN}(A)$ maggiore per poter poi fare lo split per A .

Calcoliamo il Gain(age) dell'esempio:

$$\text{Gain}(\text{age}) = \text{Info}(\Delta) - \text{Info}_{\text{AGE}}(\Delta) = \emptyset, 246$$

che è il maggior guadagno rispetto agli altri attributi quindi usiamo AGE per splittere.

Dopo la prima iterazione avremo il seguente albero:

NULL					
1 youth	high	no	fair	no	
2 youth	high	no	exelent	no	
8 youth	medium	no	fair	no	
9 youth	low	yes	fair	yes	
11 youth	medium	yes	exelent	yes	
3 middle	high	no	fair	yes	
7 middle	low	yes	exelent	yes	
7 middle	low	yes	exelent	yes	
7 middle	low	yes	exelent	yes	
4 senior	medium	no	fair	yes	
5 senior	low	yes	fair	yes	
6 senior	low	yes	exelent	no	
10 senior	medium	yes	fair	yes	
14 senior	medium	no	exelent	no	

GAIN RATIO

Il $GAIN(A)$ è molto legato al numero di valori che un attributo contiene. Tende a favorire attributi con molti valori. Se per esempio calcoliamo il $GAIN$ per un attributo identificatore (D), esso prenderà immancabilmente lo stesso valore per ogni riga e quindi un solo valore della classe target. Dato che ogni partizione è pura, l'informazione extra necessaria a classificare è zero, da cui deriva un $GAIN$ massimale, ma chiaramente questo split è inutile ai fini della classificazione.

Per superare questo bias, si fa una sorta di normalizzazione usando $SplitInfo$:

$$SplitInfo_A(D) = - \sum_{j=1}^V \frac{|D_j|}{|D|} \cdot \log_2 \left(\frac{|D_j|}{|D|} \right)$$

Questa metrica rappresenta l'informazione potenzialmente generata splttando D per i V valori di A .

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

Si sceglie l'attributo con il maggior $GainRatio(A)$

GINI INDEX

È la misura dell'impurità di D :

$$\text{gini}(D) = 1 - \sum_{i=1}^M p_i^2$$

Dove $p_i = \frac{|C_i, D|}{|D|}$ probabilità che una tupla $\in C_i$.

Lo Gini index considera split binari per ogni attributo. Se prendiamo in esame $A = \{a_1, \dots, a_v\}$, per determinare il miglior split binario bisogna valutarli tutti. Per v valori di A ci sono $2^v - 2$ possibili subset da analizzare.

Considerando split binari, valutiamo l'impurità di ogni partizione:

$$Gini_A(D) = \frac{|D_s|}{|D|} \text{gini}(D_s) + \frac{|D_e|}{|D|} \text{gini}(D_e)$$

Bisogna valutare $Gini_A(D)$ per ogni possibile split binario.

Viene selezionato il subset che ha il minimo Gini index. Per attributi continui bisogna valutare ogni possibile split-point. Similmente a come visto nel GAIN(A). La riduzione dell'impurità data dallo split-binario è rappresentata da:

$$\Delta \text{Gini}(A) = \text{Gini}(D) - Gini_A(D)$$

L'attributo A con $Gini(A)$ minore è selezionato come attributo su cui fare lo split. Questo è anche detto **PARTITION CRITERIUM**.

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

Come classe etichetta abbiamo 5 "no" e 9 "yes"

$$Gini(D) = 1 - \left(\frac{5}{14}\right)^2 - \left(\frac{9}{14}\right)^2 = 0,489$$

Audiamo a considerare il solo attributo:

$$\text{"income"} = \{\text{low}, \text{medium}, \text{high}\}$$

Con il sottoinsieme $\{\text{low}, \text{medium}\}$ che ha dimensione 10. (D_1)

$$Gini_{\text{income} \in \{\text{low}, \text{medium}\}}(D) = D_1$$

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
13	middle	high	yes	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	exelent	no
7	middle	low	yes	exelent	yes
9	youth	low	yes	fair	yes
4	senior	medium	no	fair	yes
8	youth	medium	no	fair	no
10	senior	medium	yes	fair	yes
11	youth	medium	yes	exelent	yes
12	middle	medium	no	exelent	yes
14	senior	medium	no	exelent	no

} 10

$$\frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) =$$

$$= \frac{10}{14} \left(1 - \left(\frac{2}{10}\right)^2 - \left(\frac{3}{10}\right)^2 \right) + \frac{4}{14} \left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 \right) = 0,44$$

"yes"

"no" ←

Calcolando il $Gini(\cdot)$ per gli altri sottoinsiemi binari dell'attributo "income":

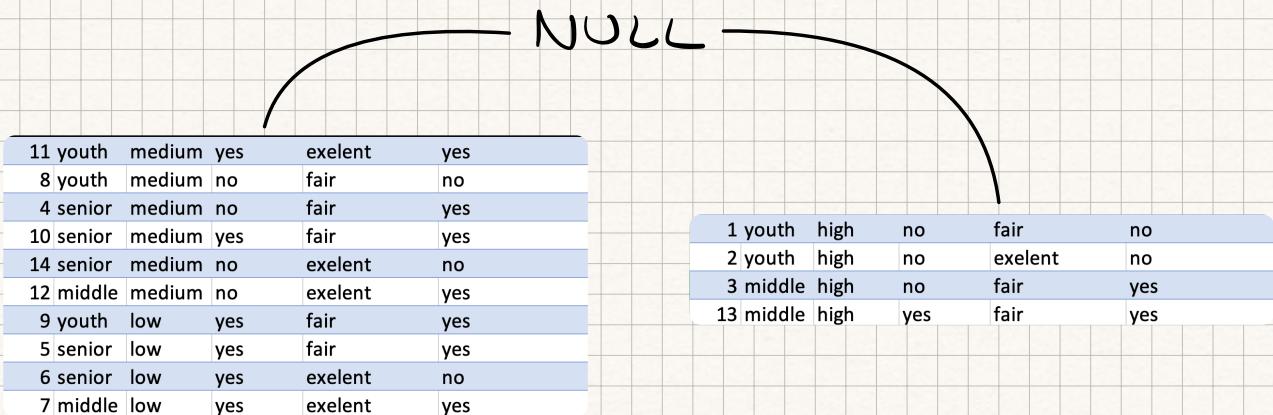
$$\text{gini}\{\text{low}, \text{high}\}(\Delta) = \text{gini}\{\text{medium}\}(\Delta) = 0,458$$

$$\text{gini}\{\text{medium}, \text{high}\}(\Delta) = \text{gini}\{\text{low}\}(\Delta) = 0,450$$

Da cui deriviamo che il minore $\text{gini}(\Delta) - \text{gini}_A(\Delta)$ è
per $\{\text{low}, \text{medium}\} \sqcup \{\text{high}\}$ split: 0,016

$$\text{gini}(\Delta) - \text{gini}_A(\Delta) = 0,016$$

Dopo la prima iterazione otteniamo il seguente albero:



PROBLEMI DI VARI INDICI

Information Gain: soffre gli attributi non binari

Gain Ratio: Tende a preferire gli attributi che portano ad avere split bilanciati in dimensioni (Uno split con molte meno tuple degli altri)

Gini Index:

- Anche qui basato dagli attributi multivalue
- Va in difficoltà quando il numero di classi è grande
- Tende a favorire situazioni in cui le due partitioni sono più pure possibile e di uguali dimensione.

