Data Mining e Bioimmagini

ALBERO DECISIONALE

Prof. Pierangelo Veltri – 16/10/2023- Autori: Gulizia, Cassalia - Revisionatori: Gulizia, Cassalia

DESCRIZIONE DI UN ALBERO DECISIONALE

Un **albero decisionale** è un tipo di diagramma che aiuta a comprendere meglio le scelte e i potenziali risultati. I dati sono rappresentati con delle *tuple* o dei *record*, cioè con delle caratteristiche, e poi è presente un'ulteriore informazione che prende il nome di *elemento di classe (esempi: sano-malato, potenzialmente a rischio) o elemento discriminante*.

Il concetto di albero decisionale come strumento di classificazione è quello di definire, a partire dagli attributi, l'elemento di classificazione rappresentato da una colonna della tabella che si verrà a creare; le altre colonne della tabella sono rappresentate dagli attributi. Una volta creata questa tabella, si devono selezionare le colonne, metterle in ordine e scegliere quella dalla quale partire per costruire l'albero.
La costruzione di un albero segue usualmente un processo in due fasi: **crescita** (*growth*) e **potatura** (*pruning*). Nella fase finale di **testing**, la valenza di un albero si misura attraverso delle *valutazioni di performance,* ovvero si utilizzano dei metodi in cui si rappresenta in maniera numerica e compatta i risultati del test-set. Si confronta il numero di risposte che si sono date correttamente rispetto al valore vero; si valutano inoltre, le classificazioni negative rispetto a ciò che si è tirato fuori; poi viceversa: *ad esempio se la classe è vera si ha sbagliato; se la classe è falsa si ha sbagliato*.

Le performance si rappresentano con delle **matrici.** In questo caso la matrice rappresentata è una matrice 2x2 dove i valori di classificazione possono essere di due tipi (vero-falso, sano-malato).

1 = vero *0 = falso*		**Predicted Class**	
		Class = 1	*Class = 0*
Actual Class	*Class = 1*	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

La matrice 2x2 prende il nome di **matrice di confusione**, nel caso specifico 2x2.

Esempio:
Se si dovessero avere tre elementi di interesse nel rischio cardiovascolare, la matrice di confusione sarebbe formata da tre righe e tre colonne.

C1 = basso			
C2 = medio			
C3 = alto			

La valutazione della performance avviene anche grazie all'**accuratezza**, la somma degli elementi che si hanno, siano essi veri e falsi, diviso il numero degli elementi totali che si sono predetti; e all'**error rate**, è una parte complementare, ovvero il numero di errori che si sono fatti.
Il concetto di indice si usa per misurare la robustezza di un test che si è costruito.

TECNICHE DI CLASSIFICAZIONE

Con **tecnica di classificazione** si intende uno dei possibili approcci al problema di classificazione, costituita da una certa famiglia di modelli e di algoritmi di apprendimento.
Tecniche di base:

- **Albero decisionale**
  - ***Explainability*** → *è il concetto che un modello di apprendimento automatico e il suo output possono essere spiegati in un modo che "ha senso" per un essere umano a un livello accettabile. Alcune classi di algoritmi, compresi gli algoritmi di apprendimento automatico più tradizionali, tendono ad essere più facilmente spiegabili, pur essendo potenzialmente meno performanti.*
- Regole di decisione
- Tecniche nearest-neighbor
- Reti neurali
- Reti bayesiane
- Support vector machine

Tecniche ensemble (di insieme):

- Boosting
- Bagging
- Random forest

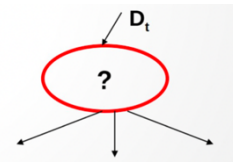
COSTRUZIONE: HUNT ALGO

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

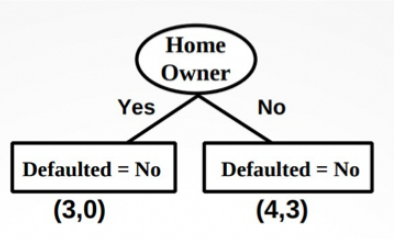
Per poter costruire un albero decisionale si ha bisogno di sapere quali siano gli attributi su cui potersi basare.
È necessario seguire una procedura generale che riguarda:

- Se D_t contiene tutti i record con la stessa classe y_t, allora t è un nodo foglia con classe y_t
- Se D_t contiene record che appartengono a più classi, determinare una condizione di test per scindere i dati in sottoinsiemi più piccoli. Applicare la procedura ricorsivamente a ogni sottoinsieme.

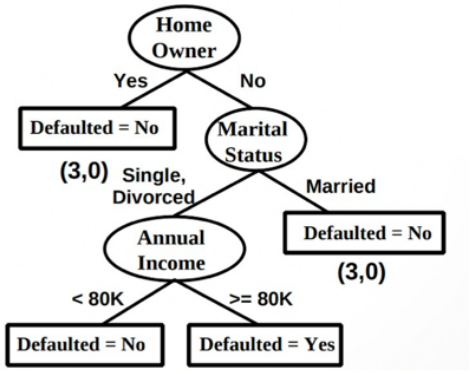
Successivamente scelgo un nodo pivot assegnato a un attributo. *Se **defaulted** è pari a no, allora è affidabile.*



Considerando il dataset, laddove si hanno, ad esempio, tre valori su un attributo, i quali sono associati allo stesso valore di classe, si possono rappresentare nel modo seguente poiché si va a ricercare un elemento che riduce lo spazio di ricerca:

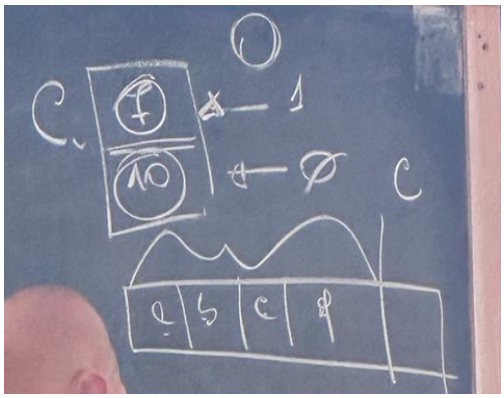


Le informazioni non appartengono sempre alle stesse classi, infatti in alcuni casi bisogna definire ulteriori elementi di dettaglio.



Questa tecnica di costruzione considera lo spazio di ricerca e lo divide (split), non necessariamente a metà. Risulta essere rilevante la costruzione dei dati poiché si possono anche non avere attributi adeguati, un dataset con informazioni con impossibilità di riorganizzazione dove l'*accuratezza* non riesce ad andare oltre l'80%.

La metodologia è basata su varie tecniche, tra cui *l'algoritmo ricorsivo*, che si costruisce a partire dal sottoalbero destro, utilizzando in particolare gli elementi di classificazione, e riesce a suddividere lo spazio di riferimento. Un'ulteriore suddivisione va fatta in base a informazioni relative alle capacità del singolo attributo a dividere le classi. Se si ha una classe binaria (affidabile-non affidabile) bisogna sapere, una volta che si possiede l'attributo, quanti elementi appartengono a una classe e quanti ad un'altra classe.



Se si ha una classe C_i binaria, indichiamo con 7 record quella associata ad una classe 1 e 10 record associata alla classe 0. Cosa vuol dire? In tal modo si ha la possibilità di definire se la domanda è fortemente discriminante; se si riesce ad essere totalmente selettivi vuol dire che la scelta è molto utile nella selezione del dataset. Dato un nodo i-esimo, la capacità, con il suo numero di query, di distinguere l'appartenenza del dataset ad una classe oppure ad un'altra, è un *indicatore di qualità*. Se si deve costruire un albero, più un nodo (sull'insieme di dati di partenza) è in grado di essere discriminante più è interessante utilizzarlo.

ESEMPIO
Se si ha una classe di cento studenti e bisogna costruire due bagni, l'elemento gender nella costruzione del bagno è fondamentale perché l'elemento di discriminazione riesce a fare una distinzione seria.

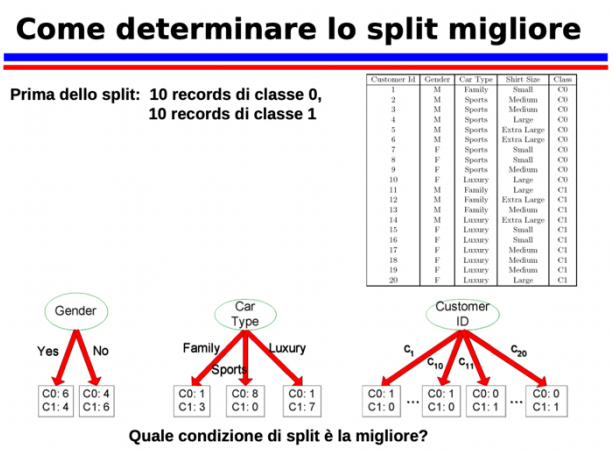
Pseudo-codice

```
// Let E be the training set and F the attributes

Result = PostPrune(TreeGrowth(E,F));

TreeGrowth(E,F)
        if StoppingCond(E,F)= TRUE then
                leaf=CreateNode();
                leaf.label = Classify(E);
                return leaf;
        else
                root = CreateNode();
                root.test_cond = FindBestSplit(E,F);
                let V = {v | v is a possible outcome of
root.test_cond}
                for each v ∈V do
                        E_v = {e | root.test_cond(e)=v and e ∈E}
                        child = TreeGrowth(E_v,F);
                        add child as descendants of root and
                        label edge (root→child) as v
        return root;
```

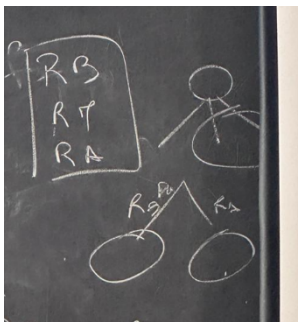
Se si deve costruire un albero di decisione, utilizzando i record di riferimento, bisogna identificare i nodi dell'albero, ovvero i nodi che consentono di fare splitting. La scelta va guidata, utilizzando delle unità di misura che devono evidenziare lo split migliore.



Per poter identificare il numero di split, bisogna essere in grado di misurare a partire da un attributo, quanto l'elemento di split riesce a fare da discriminante.

La definizione di un'unità di misura viene effettuata con dei criteri. La definizione di necessità di misurazione è legata al **come** rappresentare l'albero.
L'algoritmo deve essere utilizzato per la costruzione di un albero mediante la capacità ricorsiva *(si considerano tutti gli attributi, si selezionano, si associano ad un indice e di volta in volta se ne scegli qualcuno. Successivamente si va a verificare la discriminazione del nodo).*

La discriminazione dei dati la si fa su due classi e, ovviamente, la discrezionalità per ogni nodo cambia. Se si fa il rapporto tra le due percentuali (che vediamo nella slide), esce un indice.



ESEMPIO
Indichiamo con RB (rischio basso), RM (rischio medio), RA (rischio alto). Una prima cosa da fare su un nodo è andarlo a definirlo in base ai valori (minore, compreso e maggiore), ma un primo step richiede l'accoppiamento dei rischi (basso+medio ←→ alto). Nel dataset vengono messi insieme dei valori di riferimento e questo è utile al fine di trarre informazioni sul tipo di rischio che bisogna valutare.

La tipologia dei dati influenza la scelta dei nodi durante la costruzione. Prendendo delle tabelle si possono costruire alberi di decisione; è importante però capire come muoversi sui dati: se si hanno dati discreti e già classificati, allora il problema non si pone. L'inverso avviene se si hanno dei dati continum (*ad esempio il salario*; in questo caso si possono avere dati di diverso tipo descritti in tabella: quindi o si stabilizzano dei valori soglia, oppure delle sottoclassi e queste ultime non necessariamente devono essere due o tre, ma possono essere definite su intervalli differenti). I calcoli di affidabilità nella scelta risultano essere diversi.

L'esperto di dominio è quello che dice a che cosa si è interessati, giocano un'importante funzione durante la fase di **visualizzazione dei dati**, ovvero quando i sdati vengono visionati ed interpretati. (*esempio: se faccio una classificazione per l'ufficio del fisco, identifichiamo delle transazioni anomale a quelli che hanno uno stipendio dichiarato inferiore ai 10.000 euro, transazioni di 30.000 euro ad esempio)*.

Con **over-fitting** si indica un quantitativo di informazioni over che in realtà non aiuta nella fase di utilizzo (sembra essere molto dettagliato nella fase di learning ma non aiuta nella fase di discriminazione). A partire dalla tabella con le colonne bisogna fare un pre-processing, nel caso in cui ci siano informazioni ridondanti bisogna valutare quali sono quelle che servono e quelle che non servono in quanto bisogna garantire un livello di affidabilità elevato. Nel caso contrario si parla di **under-fitting,** ovvero non si ha un quantitativo di informazioni adeguate.