Tecniche di segmentazione classificazione: IL CART

Nel 1984, Breiman ed altri studiosi introdussero in letteratura una tecnica di segmentazione innovativa rispetto a quelle usate fino a quel momento. Tale tecnica prende il nome di *Classification And Regression Trees* o CART. Esso si presenta come una metodologia di partizione ricorsiva e binaria, dove con il termine “binaria” si intende che ciascun nodo padre di un albero decisionale è sempre diviso in esattamente due nodi figli, mentre col termine “ricorsiva” ci si riferisce al fatto che il processo può essere applicato più e più volte così che ogni nodo padre può dare origine a due nodi figli e, a turno, ciascuno di questi può esso stesso essere suddiviso dando origine ad altri due nodi figli. Infine, col termine “partizione” si intende che il dateset delle unità statistiche originario è suddiviso in sezioni o parti.

- la variabile dipendente può essere sia qualitativa che quantitativa e, infatti, nel primo caso si parla di “alberi di classificazione”, mentre nel secondo di “alberi di regressione”;
- i predittori possono essere, all’interno della medesima analisi, sia qualitativi che quantitativi;
- gli split possono essere eseguiti usando come predittori delle combinazioni lineari di variabili quantitative;
- il criterio di split viene scelto definendo il concetto di “impurità” di un nodo e selezionando la variabile che produce la massima riduzione dell’impurità;
- i dati mancanti vengono trattati sulla base dell’originale concetto di *surrogate split*;
- per trovare la dimensione ottimale degli alberi di grossa dimensione si usa una procedura di potatura o *pruning*;
- essendo un processo non parametrico, non richiede ipotesi sulla distribuzione dei valori della variabile predittore e, quindi, CART può trattare variabili numeriche che hanno andamento asimmetrico, distorto o multi-modale, cosi come variabili categoriche sia ordinali che non;
- poiché si avvale dell’aiuto di algoritmi molto efficienti, è in grado di testare tutte le possibili variabili che fungeranno da split anche in problemi con centinaia e centinaia di possibili predittori.

- Dal punto di vista formale, il CART utilizza un processo che è scomponibile in quattro passi base.
- 1)Costruzione dell'albero (*Tree Building*)
- 2)Definizione della regola di arresto (*Stopping Tree Building)*
- 3)Potatura dell'albero (*Tree Pruning)*
- 4)Selezione dell'albero ottimale (*Optimal Tree Selection*)

- Si inizia dal nodo radice, che include tutte le unità del *learning dataset*.
- Il CART cerca la miglior variabile, tra tutte quelle a disposizione, per dividere il nodo in questione in due nodi figli. Per effettuare la ricerca, il software controlla tutte le possibili variabili che potrebbero fungere da split (chiamate *splitter*) così come tutti i possibili valori della variabile da usare per la divisione.
  - **Caso di una variabile dipendente categorica (alberi di classificazione),**
  - **Caso di una variabile dipendente quantitativa (alberi di regressione),**

Alberi di classificazione (variabile dipendente categorica)

- il numero di possibili split cresce velocemente in base al numero delle sue modalità e, quindi, è utile comunicare al programma il numero massimo di livelli per ciascuna variabile categorica.
- In termini operativi, partendo dal nodo radice t si cerca la variabile che produce la miglior suddivisione degli n casi contenuti in t nei due nodi figli t_1 e t_2, di numerosità n_1 e n_2 , in modo che questi ultimi siano più omogenei o "puri" rispetto al loro padre. Il concetto di "impurità" si riferisce, quindi, all'eterogeneità delle unità statistiche all'interno di ogni gruppo creato.
- Esistono varie misure di purezza, chiamate *splitting criteria* o *splitting function*. I più usat sono l'indice di impurità di Gini, e l'entropia di Shannon seguiti dall'indice *Twoing* dall' *Ordered Twoing*, che è una variazione del precedente e da altri indici.

In generale, l'indice di eterogeneità di Gini è così definito:

$$G = 1 - \sum_{j=1}^{k} f_j^2$$

dove f_j è la frequenza relativa dell'j-esima modalità di una variabile qualitativa con k modalità. G assume tutti i valori compresi tra zero (massima omogeneità) e (k-1)/k (massima eterogeneità).

Indice relativo

$G' = G/G_{max}$

In generale, l'entropia di Shannon è così definita

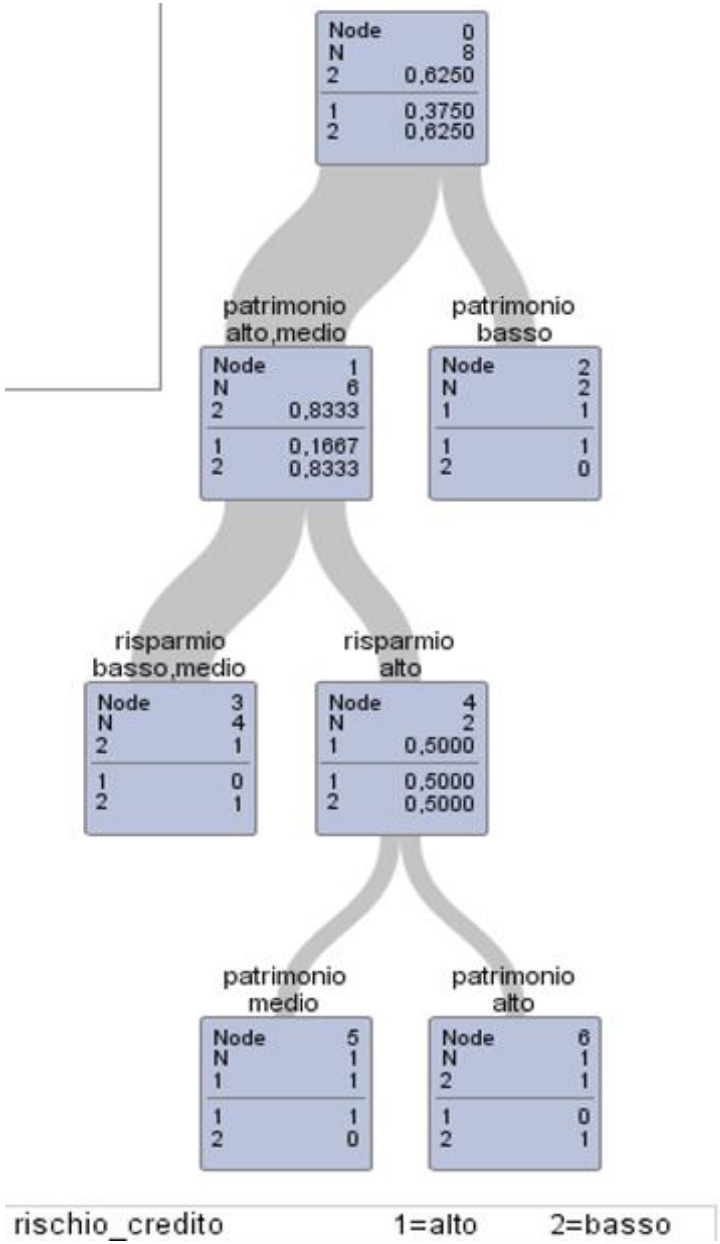
$$H = \sum_{j=1}^{k} f_j \log 1/f_j = -\sum_{j=1}^{k} f_j \log f_j$$

dove f_j è la frequenza relativa dell'j-esima modalità di una variabile qualitativa con k modalità. H assume tutti i valori compresi tra zero (massima omogeneità) e log k (massima eterogeneità).

Indice relativo

$H' = H/H_{max}$

Esempio


Node 0
N 8
2 0,6250
1 0,3750
2 0,6250
patrimonio alto,medio
patrimonio basso
Node 1
N 6
2 0,8333
1 0,1667
2 0,8333
Node 2
N 2
1 1
1 1
2 0
risparmio basso,medio
risparmio alto
Node 3
N 4
2 1
1 0
2 1
Node 4
N 2
1 0,5000
1 0,5000
2 0,5000
patrimonio medio
patrimonio alto
Node 5
N 1
1 1
1 1
2 0
Node 6
N 1
2 1
1 0
2 1
rischio_credito 1=alto 2=basso

Esempio

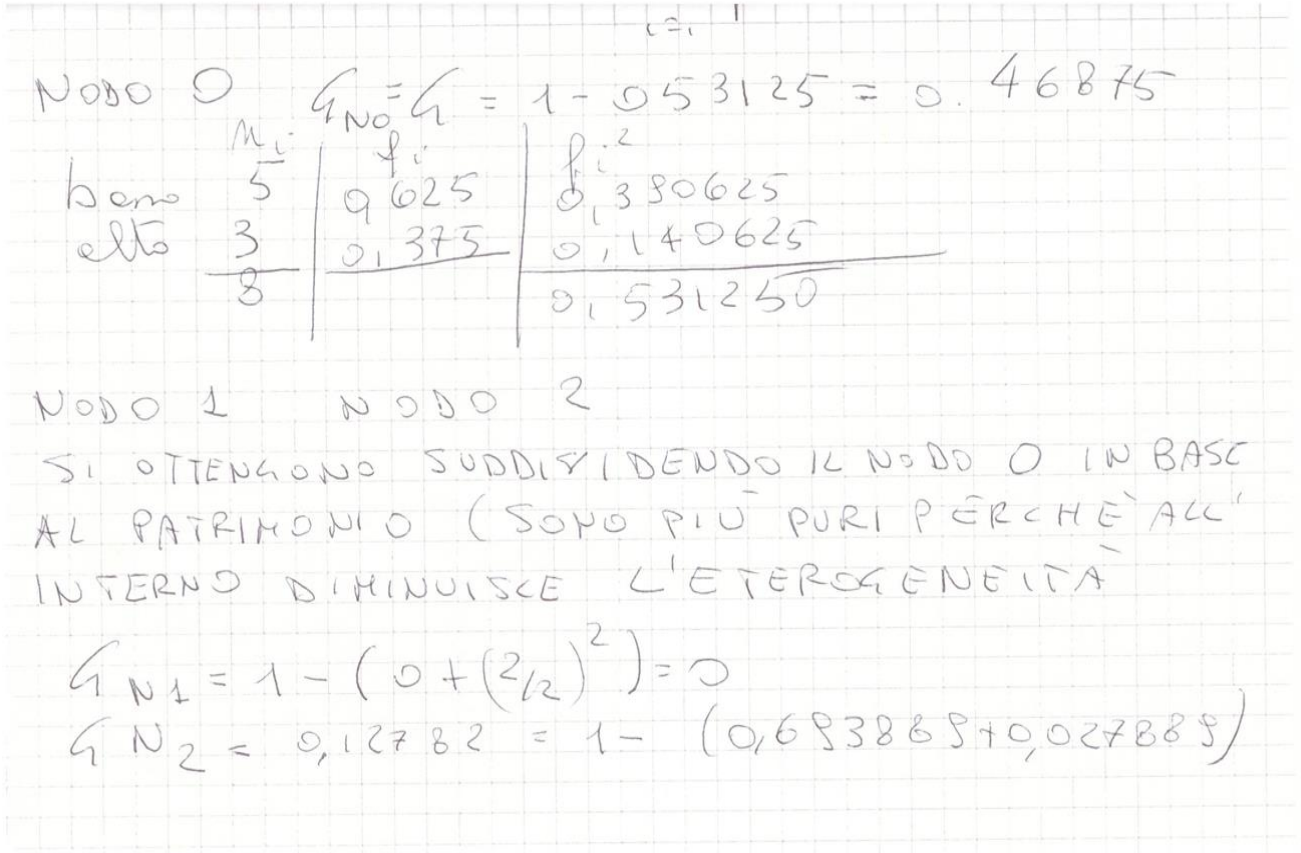
- In generale, l'indice di eterogeneità di Gini è così definito:

$$G = 1 - \sum_{i=1}^{r} f_i^2$$

Ad es. considerando esempio

(Zani Cerioli 2007)

- Nodo 0 G=0,468765
- nodo 2 G=0 nodo 1 G=0,2782.



- In questo ambito l'indice di Gini assume una diversa notazione: indicando con P(j/t) la proporzione dei casi appartenenti alla classe j che sono presenti nel nodo t, ove j=1,2,…,J e P(1/t)+…+P(J/t)=1, si definisce "misura di impurità" associata al nodo t la funzione

$$imp(t) = \Phi[P(1/t),\ldots,P(j/t),\ldots,P(J/t)]$$

con Φ funzione non negativa.

- Così lo stesso indice di Gini è definito, ora, come:

$$imp(t) = \sum_{j \neq j'} P(j/t)P(j'/t) = 1 - \sum_{j} P^2(j/t)$$

Pertanto, l'impurità di un nodo è massima quando tutte le classi della variabile dipendente vi compaiono nella stessa proporzione, mentre è minima quando il nodo contiene casi appartenenti ad un'unica classe.

- una volta fissata la misura di impurità si definisce la “misura del decremento di impurità”, Δimp(s,t), di un nodo t associata ad un determinato split s :

$$\Delta imp(\tilde{s}, t) = imp(t) - P_l imp(t_l) - P_r imp(t_r)$$

Dove P_l e P_r sono la porzione di casi che cadono nel nodo di sinistra (left) e nel nodo di destra (right).

- Dopo aver creato tutte le possibili dicotomizzazioni delle variabili esplicative, gli alberi decisionali vengono costruiti scegliendo, per un fissato nodo t, lo split $\tilde{s}$ che produce la massima riduzione di impurità dell’albero, cioè:

$$\Delta imp(\tilde{s}, t) = \max_{s \in \Theta} \Delta imp(s, t)$$

- dove Θ è l’insieme di tutte le possibili suddivisioni che è possibile fare in relazione al nodo t. La scelta di $\tilde{s}$ viene effettuata per ogni nodo e ad ogni livello dell’albero.

- Sia P_t la proporzione di unità statistiche nel nodo *t* e *IMP(t)= P_t imp(t)*, l'impurità totale di un albero T è: $$IMP(T) = \sum_{t \in \tilde{T}} IMP(t)$$

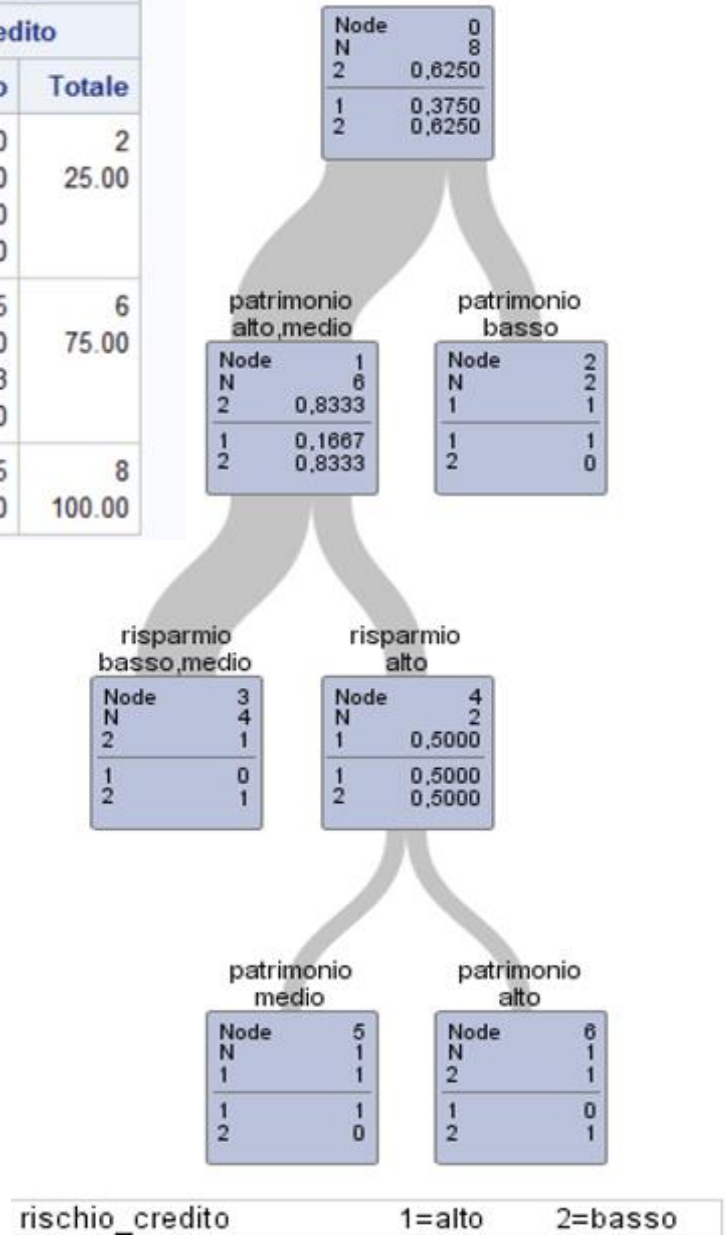
- Se la variabile scelta come miglior *splitter*, in base al procedimento appena descritto, presenta dei valori mancanti per una singola osservazione, non viene certo scartata ma sostituita, per quel valore, da una variabile surrogato (*surrogate splitting variable*) molto simile a quella primaria.
- La fase successiva all'individuazione dei nodi consiste nell'assegnare ad ogni nodo, compresa la radice, una classe predefinita. E' necessario fare ciò, perché, altrimenti, non ci sarebbe modo di sapere, durante la costruzione dell'albero, quali nodi diverranno nodi terminali dopo la "potatura".

- Il processo di costruzione si va avanti finché è possibile continuare.
- Questo vuol dire che si ferma solo quando ogni nodo figlio contiene una sola osservazione oppure quando tutte le osservazioni all'interno del nodo figlio hanno la medesima distribuzione delle variabili predittive, rendendo perciò impossibile ulteriori split. Oppure il processo si arresta se viene fissato dall'esterno, a priori, un numero limite di livelli nella dimensione dell'albero massimo che verrà costruito.
- In genere, l'albero massimo risulta molto "pesante", cioè eccessivamente frazionato. Si capisce, quindi, che, ad un certo punto, la ripartizione effettuata potrebbe risultare inutile, per quanto riguarda la spiegazione del modello, rispetto alla precedente; inoltre, può succedere che un ramo dell'albero necessiti della divisione in due o tre livelli mentre un altro ramo richieda una ripartizione molto più profonda e numerosa.

Esempio

$$imp(t) = \sum_{j \neq j'} P(j/t)P(j'/t) = 1 - \sum_{j} P^2(j/t)$$

Tabella di patrimonio1 per rischio_credito			
	rischio_credito		
patrimonio1	alto	basso	Totale
basso	2 25.00 100.00 66.67	0 0.00 0.00 0.00	2 25.00
medal	1 12.50 16.67 33.33	5 62.50 83.33 100.00	6 75.00
Totale	3 37.50	5 62.50	8 100.00



Nell'esempio usando l'indice di Gini:
Nodo 0 G=0,468765
imp (0)= 1- $(0,625)^2$ - $(0,375)^2$ = 0,468765
Sia s_1 lo split della variabile patrimonio nelle classi (basso) (medio, alto) che dà origine alla tabella di contingenza 1
nodo 2 G=0
imp (2)= 1- $(0,0)^2$ - $(1,0)^2$ = 0
nodo 2 G=0,2782
imp (2)= 1- $(0,833)^2$ + $(0,167)^2$ = 0,2782

$$imp(t) = \sum_{j \neq j'} P(j/t)P(j'/t) = 1 - \sum_{j} P^2(j/t)$$

Esempio

$$\Delta imp(\tilde{s}, t) = imp(t) - P_l imp(t_l) - P_r imp(t_r)$$

imp (0)= 1- $(0,625)^2$ - $(0,375)^2$ = 0,468765

Sia s_1 lo split della variabile patrimonio nelle classi (basso) e (medio, alto)

imp (2)= 1- $(0,0)^2$ - $(1,0)^2$ = 0

imp (1)= 1- $(0,833)^2$ + $(0,167)^2$ = 0,2782

$\Delta imp(s_1,0)$= 0,468765 – 0,75x0,2782- 0,25x0 = 0,26

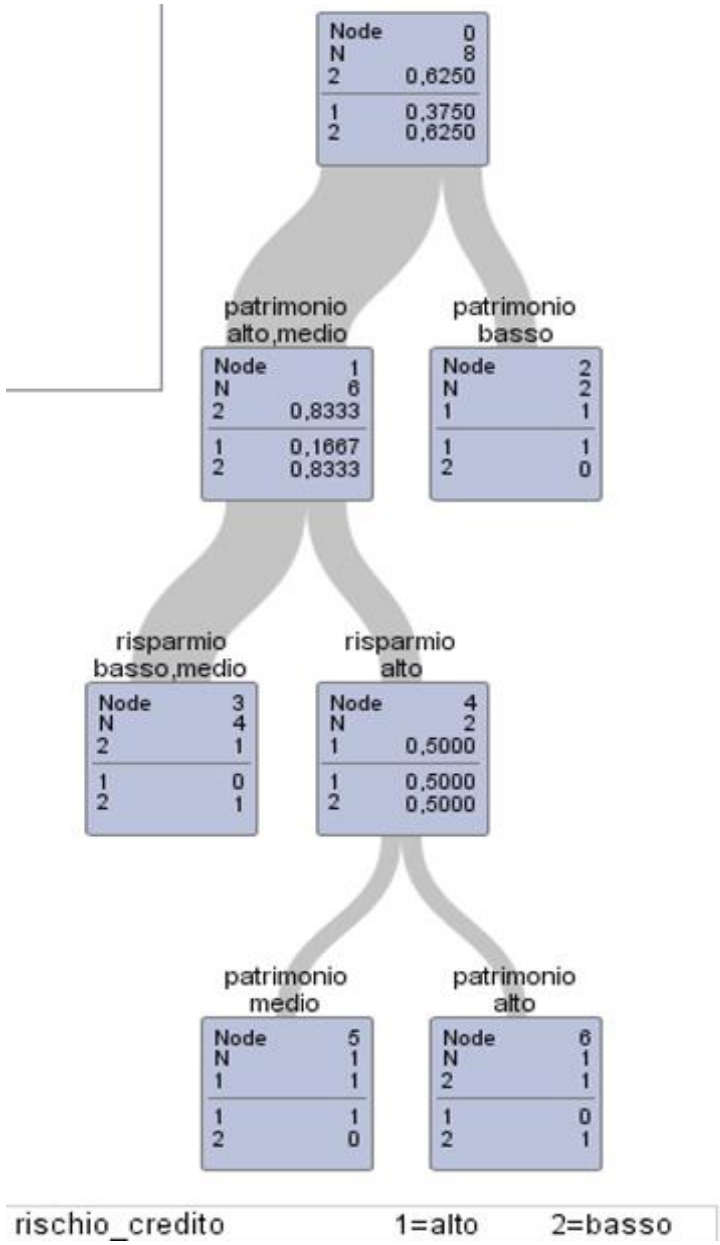
Sia s_2 lo split della variabile patrimonio nelle classi (basso, medio) e (alto)

imp (2)= 1- $(0,0)^2$ - $(1,0)^2$ = 0

imp (1)= 1- $(0,833)^2$ + $(0,167)^2$ = 0,2782

$\Delta imp(s_2,0)$= 0,468765 – 0,25x0,5 – 0,75x0,0 = 0,0925

lo split s_1 è preferibile allo split s_2



- La maggior scoperta del metodo CART consiste proprio nel mettere in luce il fatto che non c'è modo di sapere quando fermarsi durante la creazione dell'albero. Infatti, non si risolve il problema nemmeno usando un ragionevole criterio di stop, che consiste nel fissare una soglia minima β per il decremento di impurità al di sotto della quale il processo si arresta, cioè:

$$\max_{s \in S} \Delta imp(s,t) < \beta$$

- Con questo metodo appare subito chiaro che, se β è troppo piccolo, è probabile che si ottenga un albero finale troppo profondo (cioè con molte foglie) con conseguenti problemi interpretativi, mentre, se è troppo grande, un nodo *t* può essere dichiarato terminale quando in realtà i suoi nodi discendenti, se creati, fornirebbero una conoscenza più approfondita del problema.
- Proprio per ovviare a questi problemi è nata la tecnica di *pruning*.

Potatura dell'albero (*Tree Pruning)*

Questo processo rappresenta una delle caratteristiche principali del CART. Le fasi del processo sono le seguenti:

- si crea l'albero massimo T_{max}, cioè quello con una soglia β pari a zero, per cui le foglie sono costituite da casi appartenenti alla stessa classe o al limite da un solo caso;
- si selezionano i sottoalberi che si possono ottenere tagliando l'albero massimo in determinati punti e si stimano i diversi tassi di errata classificazione $R(T)$. L'albero viene perciò "sfrondato" eliminando alcuni rami secondari e non influenti per lo studio del caso;
- si sceglie il sottoalbero migliore, cioè quello che fornisce la miglior stima di $R(T)$ con T generico albero di classificazione.

- Il numero di possibili sottoalberi può essere molto elevato anche quando l'albero massimo ha un numero limitato di foglie e, per questo motivo, si utilizza una procedura di *pruning* selettivo.
- Tale metodo consiste nel fissare un parametro di complessità α (numero reale non negativo) che aumenta durante il processo di "potatura". Esso misura quanto uno split sia valido per la creazione dell'albero mettendo a confronto l'accuratezza addizionale nella spiegazione del modello fornita da tale split con l'aumento nella complessità generale dell'albero che esso provoca.
- Più α cresce e più nodi di importanza crescente vengono eliminati, andando a creare degli alberi via via sempre più semplici.

- Per ottenere la sequenza di alberi di dimensione decrescente ottimale si definisce per ogni albero $T \leq T_{max}$ una funzione costo-complessità:

$$R_\alpha(T) = \hat{R}(T) + \alpha\left|\tilde{T}\right|$$

$\hat{R}(T)$ è la stima per risostituzione del tasso di errata classificazione,

$\left|\tilde{T}\right|$ è il numero di foglie dell'albero .

- Si ricerca quel sotto albero $T(\alpha) < T_{max}$ tale che:

$$R_\alpha(T(\alpha)) = \min_{T < Tmax} R_\alpha(T)$$

- I sotto alberi appartenenti alla sequenza ottimale si confrontano mediante una stima del tasso di errata classificazione (Si veda Zani, Cerioli 2007):

  - per risostituzione,
  - mediante un campione test,
  - mediante cross validation.

Osservazione

Fra tutti i possibili sottoalberi selezionati si sceglie quello migliore.

Bisogna sottolineare, però, che molto spesso le performance dell'albero sul *training dataset* originario sono sopravalutate. Perciò, generalmente, per selezionare l'albero ottimale è richiesto l'uso di un altro gruppo di dati, *independent set*, da usare come test per correggere eventuali distorsioni del modello principale.

La tecnica della *Cross-Validation* si occupa proprio di questo. Essa è un metodo per convalidare una procedura di costruzione di modelli e richiede, appunto, l'uso di un *independent dataset.* Il dataset di partenza viene casualmente diviso in N sezioni in base alla variabile presa come target (il numero N viene definito dall'utente) e, tra queste, una viene scelta per fungere da *independent dataset* su cui fare il test, mentre le altre N-1 vengono usate per costruire il modello.

- Tutto questo procedimento viene ripetuto cambiando di volta in volta il campione test e, quindi, alla fine si ottengono N differenti modelli ciascuno dei quali testato con un diverso *independent set.*
- La *Cross Validation* si basa sul fatto che la performance media di questi N modelli fornisce un'eccellente stima della performance del modello originale, prodotto usando l'intero *training dateset*.
- Quando questo metodo viene applicato nel CART, viene prodotta una sequenza di N alberi che saranno confrontati sulla base del numero di nodi terminali; così viene determinato l'albero migliore in funzione del numero di nodi terminali o complessità minore.

Alberi di regressione (variabile dipendente quantitativa)

- Data una variabile dipendente Y che assume valori in R e p variabili esplicative (quantitative o qualitative) $X_1,...,X_p$, rilevate su n unità statistiche .
- Si indichi con $\mathbf{x}_i=[x_{i1},...,x_{is},...,x_{ip}]$ il vettore che contiene le informazioni sulle variabili esplicative relative all'*i*-esima unità statistica (valori numerici per le variabili quantitative o codici per quelle qualitative).
- Sia infine X lo spazio dei valori che possono assumere le p variabili esplicative, tale insieme è detto spazio degli attributi (feature space).
- **Obiettivo** di un albero di regressione è costruire una funzione d(x) detta regola di previsione o previsore che associa ad ogni elemento del sottospazio X un numero reale. Per costruire d(x) lo spazio degli attributi viene suddiviso utilizzando split binari fino a raggiungere un insieme di nodi terminali. In ogni nodo terminale *t* il valore previsto per la variabile dipendente *y(t)* è costante.

La costruzione di una regola di previsione gerarchica avviene attraverso:

- Scelta di un criterio per la selezione di uno split ad ogni nodo intermedio
- Fissazione di una regola di stop per l'individuazione di nodi terminali
- Costruzione di una procedura per l'assegnazione di un valore *y(t)* ad ogni nodo terminale.

- Per definire tali fasi si deve fissare un criterio di accuratezza della regola di previsione, in genere si utilizza l'errore quadratico medio *R(d)* del previsore *d*, che si può stimare con diversi criteri:
- *Stima per risostituzione*

$$\hat{R}(d) = \frac{1}{n} \sum_{i=1,\ldots,n} [y_i - d(\mathbf{x}_i)]^2$$

- *Stima basata sul campione test*

Si suddivide casualmente il campione S in due sottocampioni S_1 e S_2 rispettivamente di numerosità n_1 e n_2, tali che $S_1 \cup S_2 = S$ e $S_1 \cap S_2 = \emptyset$.

S_1 si chiama "campione di apprendimento" (*learning sample*) ed S_2 si chiama "campione test" (*testing sample*). La regola di stima che ne deriva è:

$$\hat{R}(d) = \frac{1}{n_2} \sum_{i \in S_2} [y_i - d(\mathbf{x}_i)]^2$$

è la stima sul campione test, dato che la regola *d* viene costruita tramite il campione S_1 e viene testata stimando $R(d)$ su S_2.

- *Stima basata sulla cross-validation (V-fold Cross-Validation Estimate)*

Il campione S viene suddiviso casualmente in V>2 sottocampioni $S_1,\ldots,S_v,\ldots,S_V$ di dimensione in più possibile simile fra loro. Si costruisce una regola di classificazione $d^{(v)}(x)$ sul campione S - S_v (v=1,...,V) e si fa una stima di $R(d^{(v)})$ basata sul campione test

Se V è sufficientemente elevato, ogni classificatore $d^{(v)}(x)$ viene costruito usando un campione di apprendimento di dimensione $n(1-1/V)$ ossia prossima alla dimensione di S. La base portante del metodo basato sulla *cross-validation* è quindi la stabilità, perché ogni classificatore $d^{(v)}(x)$ ha un tasso di errata classificazione $R(d^{(v)})$ molto prossimo a $R(d)$.

- La stima del tasso di errata classificazione calcolato con questo metodo ha la seguente formula:

$$\hat{R}_{cv}(d) = \frac{1}{V}\sum_{v=1}^{V} \sum_{i \in S_v} [y_i - d\ (\mathbf{x}_i)]^2 \ .$$

- OSSERVAZIONE: negli alberi di regressione l’errore quadratico medio è influenzato dalla scala si può passare a quello relativo dividendo per la varianza.

Si può sostituire la notazione *R(d)* con *R(T)* con *T* generico albero di regressione.

- una volta fissata la stima del tasso di errata classificazione si definisce la stima della misura del decremento Δ *R(s,t),* di un nodo *t* associata ad un determinato split *s*

$$\Delta\hat{R}(s,t) = \hat{R}(t) - \hat{R}(t_l) - \hat{R}(t_r)$$

- Per ogni split di t in t_l nodo di sinistra (left) e nel nodo di destra (right) t_r.

- Dopo aver creato tutte le possibili dicotomizzazioni delle variabili esplicative, gli alberi decisionali vengono costruiti scegliendo, per un fissato nodo t, lo split $\tilde{s}$ tale che:

$$\Delta\hat{R}(\tilde{s},t) = \max_{s\in\Theta} \Delta\hat{R}(s,t)$$

- Dove Θ è l'insieme di tutte le possibili suddivisioni che è possibile fare in relazione al nodo t. La scelta dello split viene effettuata per ogni nodo e ad ogni livello dell'albero.

- Anche per gli alberi di regressione il numero di possibili sottoalberi può essere molto elevato anche quando l'albero massimo ha un numero limitato di foglie e, per questo motivo, si utilizza una procedura di *pruning* selettivo.
- Tale metodo consiste nel fissare un parametro di complessità α (numero reale non negativo) che aumenta durante il processo di "potatura". Esso misura quanto uno split sia valido per la creazione dell'albero mettendo a confronto l'accuratezza addizionale nella spiegazione del modello fornita da tale split con l'aumento nella complessità generale dell'albero che esso provoca.
- Più α cresce e più nodi di importanza crescente vengono eliminati, andando a creare degli alberi via via sempre più semplici.

- Per ottenere la sequenza di alberi di dimensione decrescente ottimale si definisce per ogni albero $T \leq T_{max}$ una funzione costo-complessità:

$$R_\alpha(T) = \hat{R}(T) + \alpha\left|\tilde{T}\right|$$

$\hat{R}(T)$ è la stima per risostituzione del tasso di errata classificazione,

$\left|\tilde{T}\right|$ è il numero di foglie dell'albero .

- Si ricerca quel sotto albero $T(\alpha) < T_{max}$ tale che:

$$R_\alpha(T(\alpha)) = \min_{T < Tmax} R_\alpha(T)$$

- I sotto alberi appartenenti alla sequenza ottimale si confrontano mediante una stima del tasso di errata classificazione (Si veda Zani, Cerioli 2007):

  - per risostituzione,
  - mediante un campione test,
  - mediante cross validation.