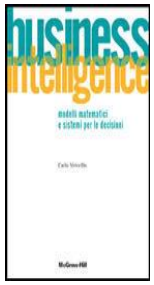
CLASSIFICAZIONE

INTRODUZIONE

Classificazione

Parte dei contenuti della presente lezione sono tratti dai testi elencati di seguito.



Carlo Vercellis (2006). *Business intelligence: modelli matematici e sistemi per le decisioni*, McGraw-Hill.

Introduzione

Il *Supervised Learning* è una tipologia di analisi caratterizzata da:

- *un insieme di attributi, anche noti con il termine di variabili esplicative*
- *una variabile target, classe di appartenenza o variabile di risposta*

ed è orientata a *predizione e interpretazione* del valore assunto dalla variabile target.

Il *Supervised Learning* utilizza le variabili esplicative; continue, categoriche ordinali o nominali, per risolvere problemi di

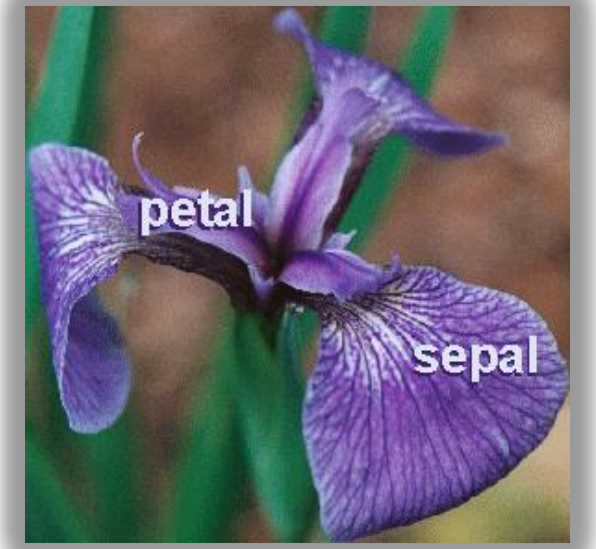
- **Classificazione:** la variabile target è categorica nominale o ordinale, comunque a supporto finito
- **Regressione:** la variabile target è continua

Introduzione: classificazione

SETOSA

VERSICOLOR

VIRGINICA



Introduzione: classificazione

Variabili esplicative

- **sepalo** *lunghezza*, *larghezza*
- **petalo** *lunghezza*, *larghezza*

Classe

- ***setosa***, ***versicolor***, ***virginica***

SETOSA

VERSICOLOR

VIRGINICA

Introduzione: classificazione

Dato un insieme di casi o osservazioni per le quali sono noti i valori assunti dalle *variabili esplicative* e dalla *variabile target* o *classe*, abbiamo l'obiettivo di sviluppare un modello in grado di fornire previsioni circa la variabile di classe, possibilmente per valori delle variabili esplicative non disponibili all'interno dell'insieme dei casi o osservazioni disponibili.

Consideriamo un dataset

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

contenente "m" osservazioni relative ad "n" attributi (variabili esplicative) e $\mathbf{X}^j$, j=1,…, n, ed una variabile target Y. Gli *attributi* possono essere *continui*, *ordinali* o *nominali*, mentre la variabile target Y viene anche riferita con il termine di *classe* o *etichetta*.

Le "m" *osservazioni* del dataset sono anche riferite con il termine di *esempi* o *istanze*.

La variabile target assume un numero finito di valori e di norma verranno affrontati problemi di *classificazione binaria* o *classificazione multi-classe* (multi-categorica).

Introduzione: classificazione

I modelli di classificazione hanno l'obiettivo di scoprire i legami tra la variabile target e le variabili esplicative. Tali legami vengono successivamente utilizzati per prevedere la variabile target a partire da un'istanza delle variabili esplicative.

Formalmente, in un problema di classificazione è disponibile un dataset

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

contenente "m" osservazioni $\underline{\mathbf{x}}_i$, i=1, …, m, relative ad "n" attributi (variabili esplicative) $\mathbf{X}^j$, j=1,…, n. La variabile target Y ha *supporto finito*

$$H = \{v_1, \ldots, v_H\}$$

In un problema di classificazione binaria H=2 e le due classi possono essere indicate tramite {0,1} o equivalentemente tramite {-1, +1}.

Introduzione: classificazione

Indichiamo con F un insieme di funzioni definite sullo spazio delle variabili esplicative verso il supporto della variabile target o variabile di classe:

$$f(\underline{x}): R^n \rightarrow H$$

Queste funzioni vengono assimilate ad *ipotesi* o per meglio dire a possibili relazioni esistenti tra le variabili esplicative e la variabile di classe.

Il *problema di classificazione* consiste nella:

- *definizione di uno spazio delle ipotesi*
- *progettazione o scelta di un algoritmo*

che consenta di selezionare una funzione $f^* \in F$ tale da descrivere in modo "**ottimale**" la relazione incognita esistente tra le variabili esplicative **X** e la variabile di classe Y.

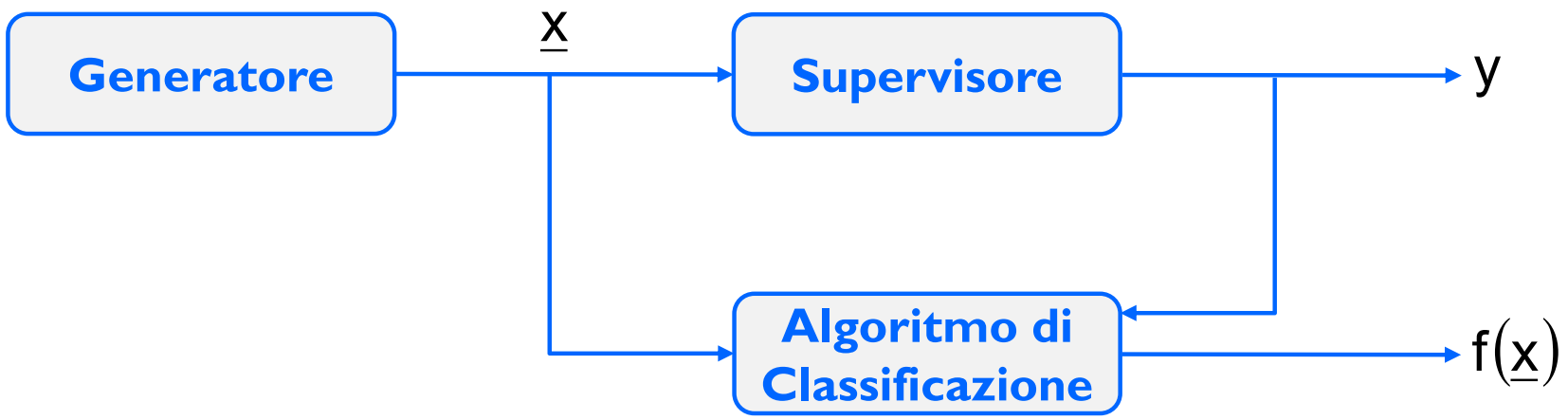
Introduzione: classificazione

Ipotizziamo che le coppie $(\underline{x}_i, y_i)$ provengano da una distribuzione di probabilità $P_{\underline{X},Y}(\underline{x},y)$ incognita.

$$f(\underline{x}): R^n \rightarrow H$$

estrae campioni $\underline{x}$ *dalla distribuzione* $P_X(\underline{x})$ *incognita*

restituisce per ogni vettore $\underline{x}$ *il valore della classe in accordo a* $P_{Y|\underline{X}}(y|\underline{x})$



classificatore che minimizza una determinata funzione di perdita

Introduzione: classificazione

Il dataset

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

viene in parte utilizzato per l'apprendimento del classificatore, la cosiddetta fase di *training* del modello, in parte utilizzato per scegliere quale modello sia il "**modello migliore**" e per stimare il valore delle *misure di prestazione* con esso ottenute.

Lo sviluppo del modello di classificazione prevede le seguenti fasi

- **Training**, effettuata utilizzando un sottoinsieme delle istanze dell'insieme *D*, si effettua la scelta della forma del modello e la scelta del valore ottimale dei suoi parametri,
- **Testing**, il modello appreso tramite la fase di Training viene interrogato su istanze appartenenti al dataset *D* che non sono state utilizzate per la fase di Training per ottenere una stima (non distorta) delle misure di prestazione,
- **Predizione**, effettivo utilizzo del classificatore per assegnare ad ogni istanza **x** il valore della classe Y.

Introduzione: classificazione

I *modelli di classificazione si suddividono* in

- **Modelli di regressione**, ipotizzano un'esplicita forma funzionale per la probabilità condizionata $P_{y|\underline{x}}(y|\underline{x})$. Ne fa parte la *regressione logistica*,
- **Modelli euristici**, utilizzano procedure di classificazione basate su schemi algoritmici elementari ed intuitivi (*nearest neighbor*, *decision trees*, *k-star*),
- **Modelli di separazione**, permettono di separare le osservazioni sulla base della classe di appartenenza, ogni regione è costituita da un insieme composito ottenuto mediante operatori insiemistici (unione, intersezione) applicati a regioni dalla forma elementare (semi-spazi o ipersfere), ne fanno parte l'*analisi discriminante*, le *Support Vector Machines*, le *Artificial Neural Networks*,
- **Modelli probabilistici**, formulano un'ipotesi circa la forma funzionale delle probabilità condizionate delle osservazioni data la classe target di appartenenza, indicate come probabilità condizionate alle classi. Si sfrutta una stima della probabilità a priori e il teorema di Bayes per ricavare la probabilità a posteriori della variabile di classe. (*Naïve Bayes*, *HNB*, *BN*, …).

Introduzione: classificazione

I modelli di classificazione ammettono di norma una *score function*

$$g(\underline{x}) : R^n \rightarrow R$$

che associa ad ogni osservazione **x** un numero reale che può essere ricondotto alla stima della probabilità che la classe predetta per l'osservazione **x** da parte del classificatore sia effettivamente corretta.

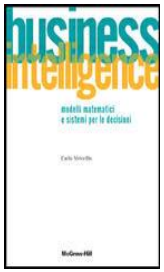
La *score function* consente di ricavare una regola di classificazione per prevedere la classe target associata all'osservazione **x**.

CLASSIFICAZIONE

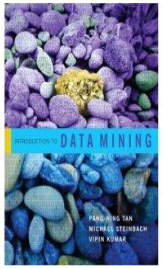
VALUTAZIONE MODELLI

Classificazione

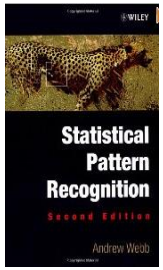
Parte dei contenuti della presente lezione sono tratti dai testi elencati di seguito.



Carlo Vercellis (2006). *Business intelligence: modelli matematici e sistemi per le decisioni*, McGraw-Hill.



Pang-Ning Tan, Michael Steinbach and Vipin Kumar (2006). *Introduction to Data Mining*, Pearson International.



Andrew R Webb (2002). *Statistical Pattern Recognition*, John Wiley and Sons.

Valutazione Modelli

Nel corso di un'analisi di classificazione è opportuno sviluppare diversi modelli sia in termini dell'algoritmo selezionato sia in termini di valori dei relativi parametri. Questo viene di norma finalizzato alla "*selezione*" di una particolare istanza di modello di classificazione, algoritmo e valore dei parametri, in grado di "*garantire*" la "*massima accuratezza predittiva*".

I modelli di classificazione vengono valutati in base a:

- ***Accuratezza***
- ***Velocità***
- ***Robustezza***
- ***Scalabilità***
- ***Interpretabilità***

Valutazione Modelli: accuratezza

Importante da valutare in quanto

- Rappresenta un indicatore della propensione del modello a fornire previsioni attendibili in corrispondenza di nuove osservazioni (osservazioni non disponibili al momento della costruzione del modello),
- Consente di "*selezionare*" il modello di classificazione che sarà presumibilmente in grado di ottenere la "*migliore prestazione predittiva*" su nuovi dati.

Adottiamo la seguente notazione

D_T training set, contiene "t" osservazioni

D_V test set, contiene "v" osservazioni

$$D = D_T \cup D_V, \quad D_T \cap D_V = \varnothing, \quad m = t + v$$

Valutazione Modelli: accuratezza

Intuitivamente, l'indicatore per sintetizzare l'accuratezza del classificatore è rappresentato dalla percentuale di osservazioni del test set D_V che esso classifica correttamente.

Se indichiamo con y_i la classe di appartenenza associata all'osservazione $\underline{x}_i \in D_V$ e se si indica con $f(\underline{x}_i)$ la classe di appartenenza della medesima osservazione così come prevista mediante la funzione $f \in F$ che implementa l'algoritmo di classificazione $A = A_F$, allora possiamo adottare la seguente funzione di perdita

$$L(y_i, f(\underline{x}_i)) = \begin{cases} 0 & \text{se } y_i = f(\underline{x}_i) \\ 1 & \text{se } y_i \neq f(\underline{x}_i) \end{cases}$$

e computare l'*accuratezza* del modello $A = A_F$ come segue

$$acc_A(D_V) = acc_{A_F}(D_V) = 1 - \frac{1}{v}\sum_{i=1}^{v} L(y_i, f(\underline{x}_i))$$

In alcuni casi si preferisce utilizzare la *percentuale di errori* commessi dal classificatore

$$err_A(D_V) = err_{A_F}(D_V) = 1 - acc_{A_F}(D_V) = \frac{1}{v}\sum_{i=1}^{v} L(y_i, f(\underline{x}_i))$$

Valutazione Modelli: velocità

Gli algoritmi di classificazione differiscono per due caratteristiche fondamentali di complessità:

- *tempo di apprendimento*
- *spazio di memoria richiesto*

La selezione di un modello di classificazione per affrontare un determinato problema di Data Mining risente molto delle caratteristiche sopra riportate.

Un classificatore che richieda tempo o spazio di memoria notevoli per essere addestrato può essere comunque istruito previa formazione di un campione dei dati disponibili che abbia una consistenza tale da rendere applicabile il modello in questione.

Un tale approccio rinuncia ad utilizzare informazione disponibile in favore dell'applicabilità di un determinato modello di classificazione che si ritiene possa essere in grado di offrire buone prestazioni.

Valutazione Modelli: robustezza e scalabilità

Un algoritmo di classificazione può essere *robusto* o meno rispetto a fattori quali

- *variazione dei dati di training e test*
- *presenza di missing data*
- *presenza di osservazioni outliers*

Di un classificatore si dice che scala (*è scalabile*) se è predisposto ad apprendere da grandi quantità di dati. Questa proprietà è intrinsecamente collegata alla velocità di apprendimento.

Nei casi in cui l'analisi di Data Mining sia orientata alla comprensione o *interpretabilità* del problema sotto indagine e non solamente a garantire un buon livello di predizione, è importante che le regole prodotte siano semplici e comprensibili per l'*esperto di dominio* del problema che si sta analizzando.

Valutazione Modelli: hold-out

L'insieme dei dati "D" viene suddiviso in due sottoinsiemi disgiunti

D_T training set, contiene "t" osservazioni

D_V test set, contiene "v" osservazioni

$$D = D_T \cup D_V, \quad D_T \cap D_V = \varnothing, \quad m = t + v$$

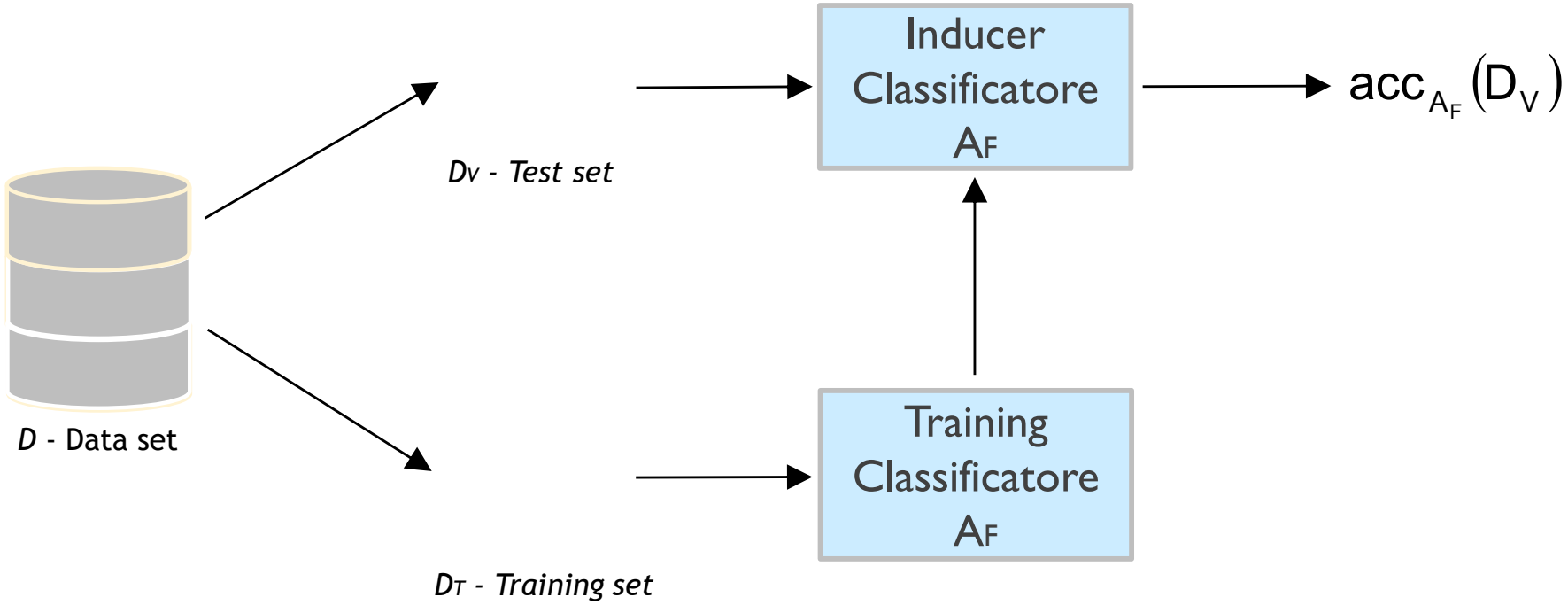
L'accuratezza di un modello di classificazione viene stimata computando la quantità

$$acc_A(D_V) = acc_{A_F}(D_V) = 1 - \frac{1}{v}\sum_{i=1}^{v} L(y_i, f(\underline{x}_i))$$

L'insieme "D" viene ripartito in due sottoinsiemi di norma tramite l'applicazione di una procedura di campionamento casuale (semplice). La "*best practice*" suggerisce una ripartizione "2/3" – "1/3" per i due sottoinsiemi di training e di test.

Valutazione Modelli: hold-out

Il metodo *Hold-Out* consiste nel non consentire al modello di classificazione di accedere a tutti i dati a disposizione ma solo ad alcuni di essi, riservando alcuni casi per stimare il grado di affidabilità del medesimo nell'effetuare il task per cui viene implementato.



La stima dell'accuratezza dipende dalla scelta del test set, pertanto è possibile sovrastimare o sottostimare il reale valore dell'accuratezza. Una stima maggiormente robusta può essere ottenuta implementando diverse strategie come *Iterated Hold-Out* e *Cross Validation* che descriveremo nel seguito.

Valutazione Modelli: iterated hold-out

Noto con il termine di metodo dei campionamenti casuali ripetuti o *Iterated Hold-Out*.

Consiste nel replicare "R" volte l'applicazione del metodo dell'Hold-Out.

Per ogni iterazione "r" si estrae un campione casuale indipendente, indicato con "D_{Tr}", di cardinalità pari a "t" osservazioni. Si ottengono pertanto

$$D_{V_r} = D - D_{T_r}$$

La procedura viene ripetuta per "R" volte e l'accuratezza del classificatore "A_F" viene stimata tramite la media campionaria seguente:

$$acc_A = acc_{A_F} = \frac{1}{R}\sum_{r=1}^{R} acc_{A_F}\left(D_{V_r}\right)$$

Il numero di iterazioni "R" può essere selezionato a priori sfruttando tecniche di dimensionamento dei campioni per l'inferenza statistica.

Valutazione Modelli: iterated hold-out

Il metodo in questione è decisamente preferibile rispetto al metodo dell'Hold-Out in quanto in grado di ottenere una stima maggiormente attendibile.

Tuttavia il metodo dell'Iterated Hold-Out non consente di controllare in alcun modo il numero di volte che ogni osservazione compare nel training set e nel test set.

Questo fatto potrebbe portare a distorsioni importanti nel caso in cui per esempio esista un'osservazione dominante, osservazione inusuale, anomala, insomma un'osservazione outlier.

È importante in questi casi ricorrere a schemi di stima più robusti che aiutino a mitigare l'effetto della presenza di outlier sul livello di distorsione dell'accuratezza del classificatore.

Valutazione Modelli: cross validation

Assicura che ogni osservazione del dataset "D" compaia un egual numero di volte negli insiemi di training ed esattamente una volta nell'insieme di test.

Il dataset "D" viene ripartito in "K" sottoinsiemi disgiunti, esaustivi (partizione di "D") e di cardinalità il più possibile uguale, che indicheremo come segue

$$D_1, D_2, \ldots, D_K$$

Vengono effettuate "K" iterazioni di apprendimento ed altrettante di test. All'iterazione "k-ma" si ha quanto segue

$$D_{T_k} = \{D_1, \ldots, D_{k-1}, D_{k+1}, \ldots, D_K\}$$

$$D_{V_k} = D_k$$

L'insieme "D_{T_k}" viene utilizzato come insieme di training mentre l'insieme "D_k" viene utilizzato come insieme di test per l'iterazione "k-ma".

Valutazione Modelli: cross validation

L'algoritmo di classificazione viene sottoposto a "K" fasi di apprendimento ed altrettante fasi di predizione sui "K" insiemi "D_k", con le "K" stime di accuratezza così ottenute viene computata la media aritmetica

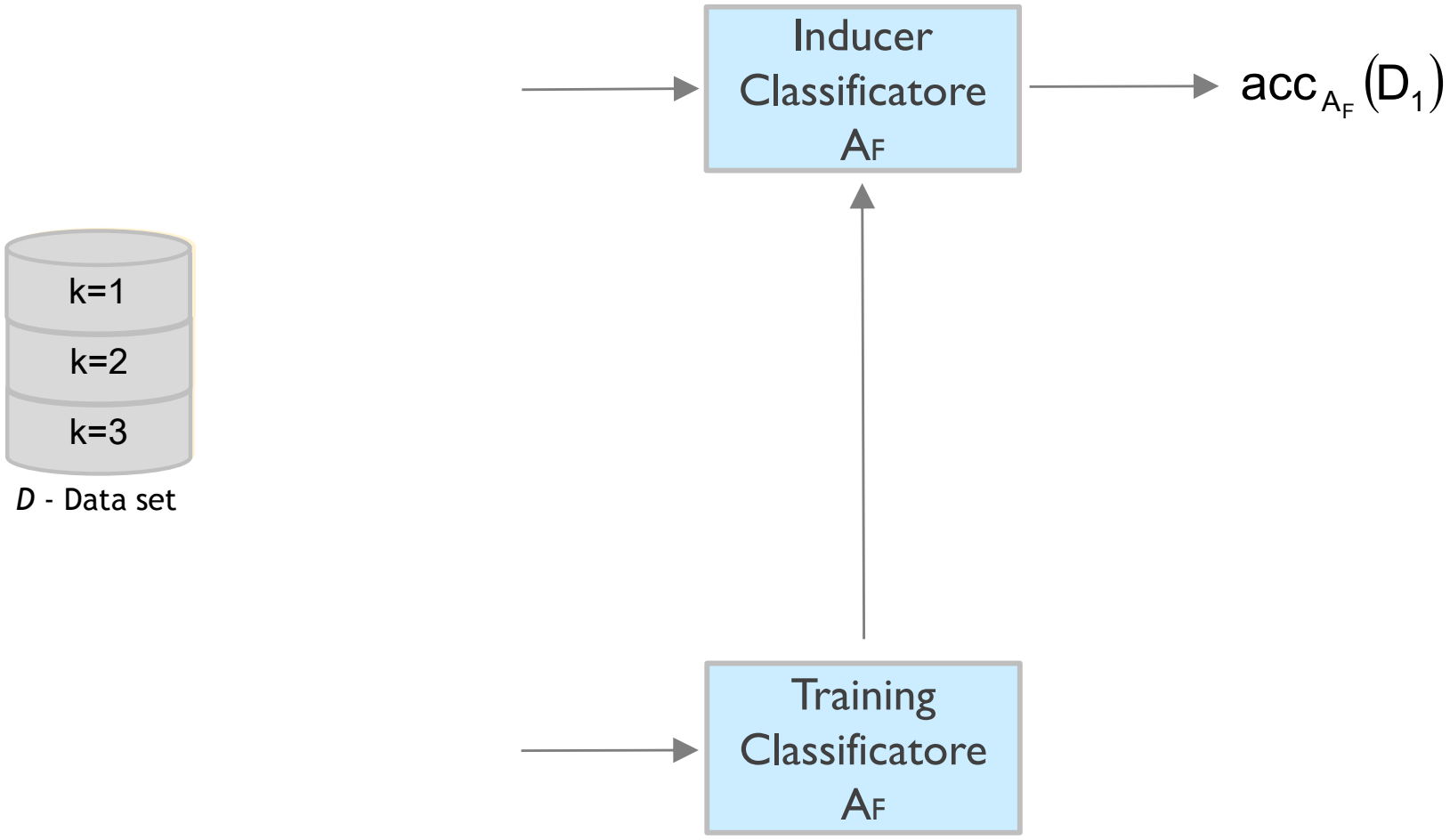
$$acc_A = acc_{A_F} = \frac{1}{K}\sum_{k=1}^{K} acc_{A_F}(D_k)$$

che offre uno stimatore più robusto dell'effettiva accuratezza del classificatore.

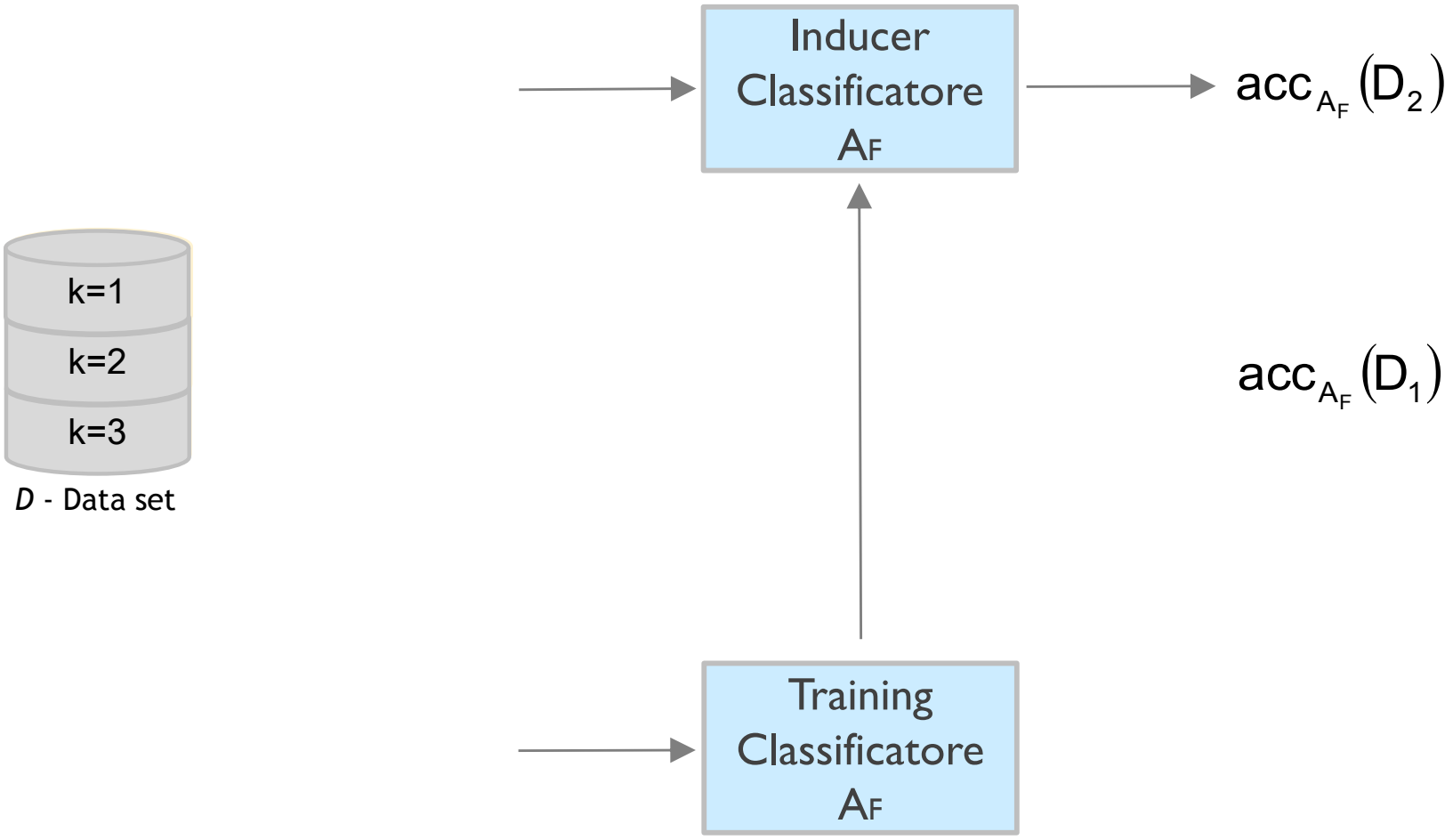
Esistono diverse possibilità per la scelta del valore del parametro "K" dello schema di cross validation. Valori usuali sono K=3, 5, 10, mentre una degenerazione spesso utilizzata nella letteratura specializzata, soprattutto nel caso in cui la numerosità dei dati disponibili sia bassa, è nota con il nome di *Leave One Out Cross Validation* (**LOOCV**).

LOOCV si ottiene assumendo che ogni singolo dato sia un sottoinsieme della nostra partizione per cui in questo caso avremo che il valore di "K" sarà pari al numero di osservazioni disponibili nel dataset "D".

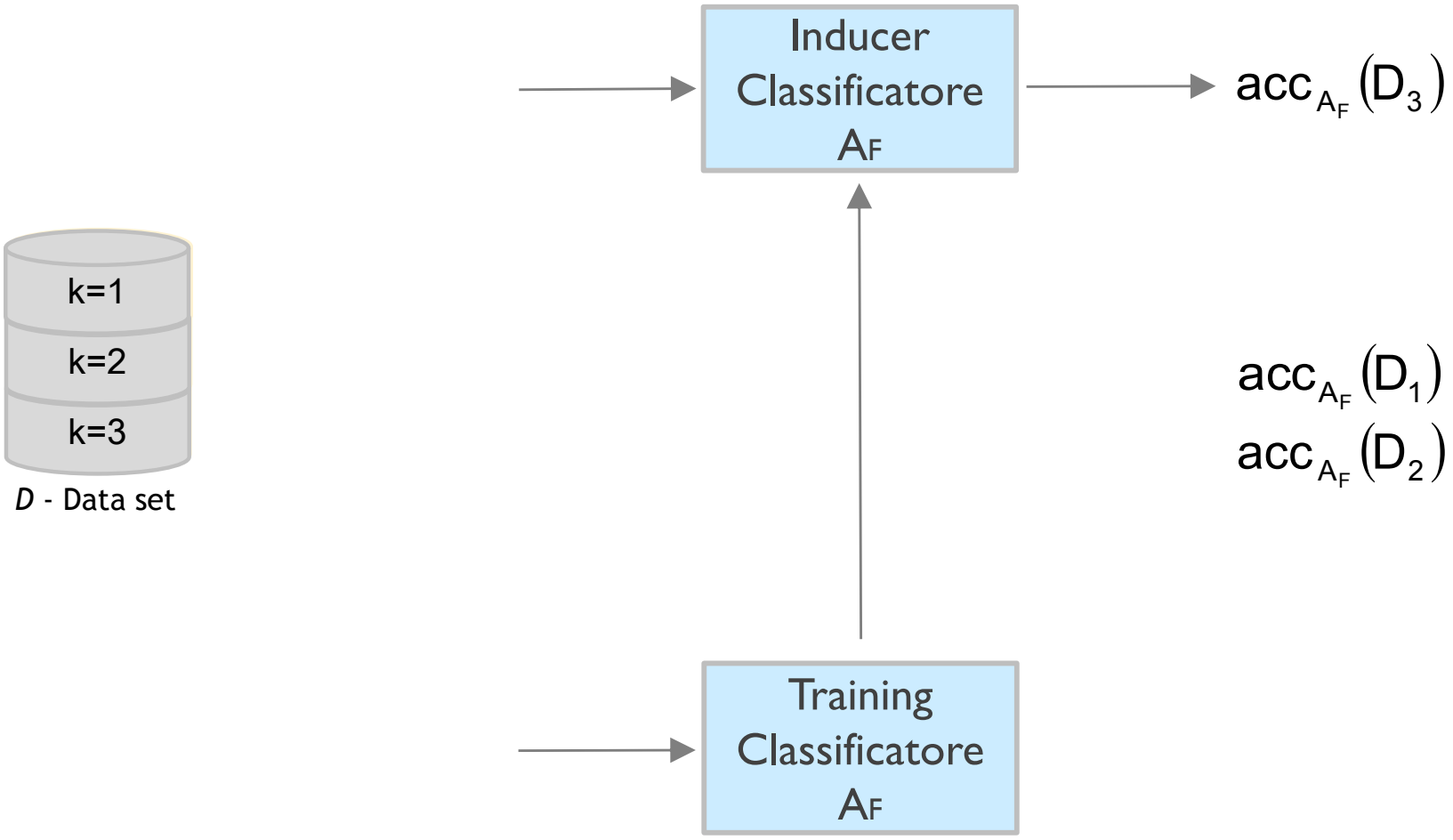
Valutazione Modelli: cross validation



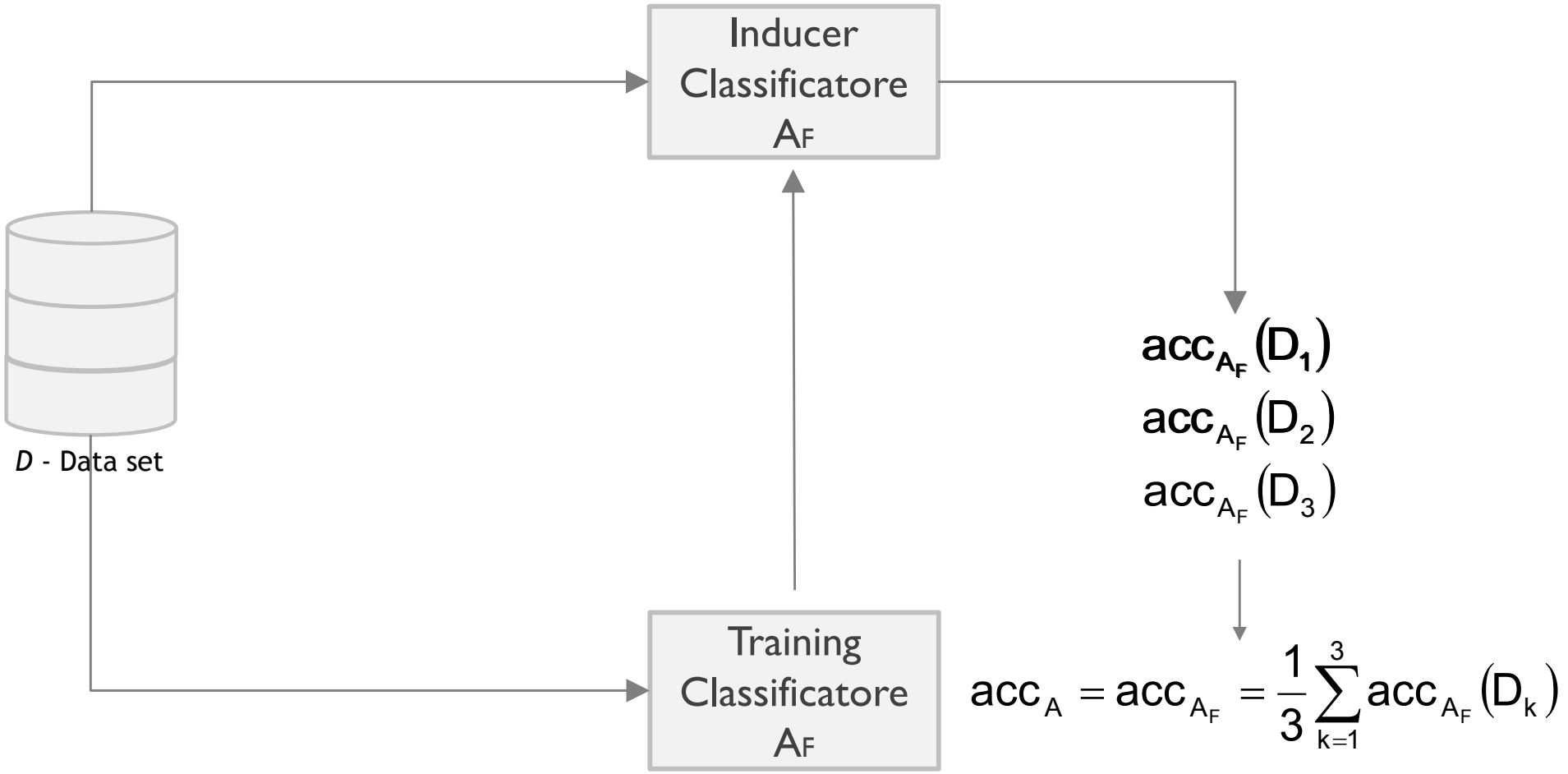
Valutazione Modelli: cross validation



Valutazione Modelli: cross validation



Valutazione Modelli: cross validation



Valutazione Modelli: cross validation

Un'ultima precisazione è relativa al fatto che di norma si richiede che ogni sottoinsieme della partizione sia tale da contenere approssimativamente la stessa percentuale di osservazioni per ognuno dei valori che può assumere la variabile di classe.

In alcuni casi particolari, quando le numerosità dei diversi valori della variabile di classe sono molto differenti tra loro, esistono valori della variabile classe che presentano un numero limitato di osservazioni, si ricorre ad un *campionamento stratificato*.

Il *campionamento stratificato* costruisce sottoinsiemi cercando di fare in modo che per ogni valore che può assumere la variabile di classe, la percentuale di casi presenti in ogni sottoinsieme sia circa uguale alla percentuale di casi, per il medesimo valore della variabile di classe, che costituisce l'intero dataset "D".

Valutazione Modelli: matrici di confusione

Accuratezza ed errore non sempre sono esaustive ed adeguate per valutare la qualità di un modello di classificazione, per effettuare una comparazione con conseguente scelta tra diversi modelli di classificazione.

Supponiamo di aver sviluppato un modello di classificazione per diagnosticare una malattia genetica a partire dalla misurazione dei livelli di espressione genica di un individuo. L'accuratezza del miglior modello di classificazione è stata stimata essere pari a **0.997**, per cui su una popolazione infinita di individui sottoposti a misurazione del livello di espressione genica ci attendiamo che la diagnosi proposta dal classificatore sia corretta nel **99.7%** dei casi. Questo valore è molto elevato e potremmo essere estremamente soddisfatti di questo risultato.

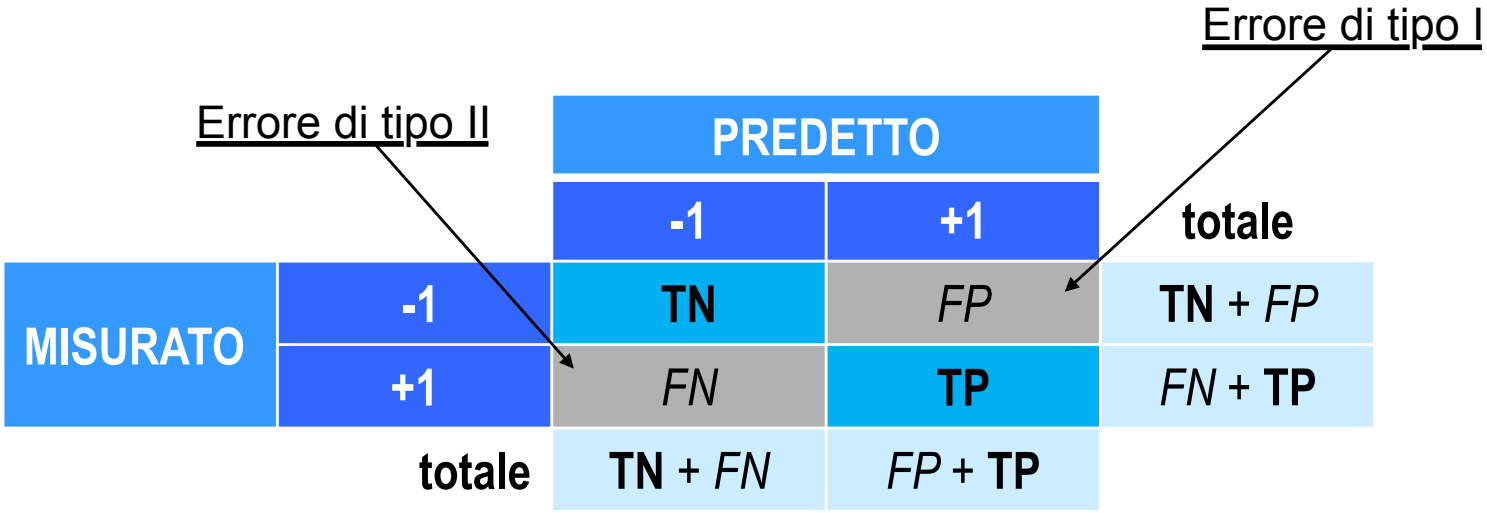
*Come cambierebbe il nostro punto di vista se apprendessimo che esattamente il **3 per mille** della popolazione soffre di tale difetto genetico?*

Valutazione Modelli: matrici di confusione

Non è pertanto sufficiente limitarsi a misurare (stimare) la percentuale di predizioni corrette ma è necessario valutare come il modello di classificazione commette i propri errori.

Le *matrici di confusione* offrono uno strumento molto utile per valutare in modo approfondito il comportamento di un modello di classificazione.

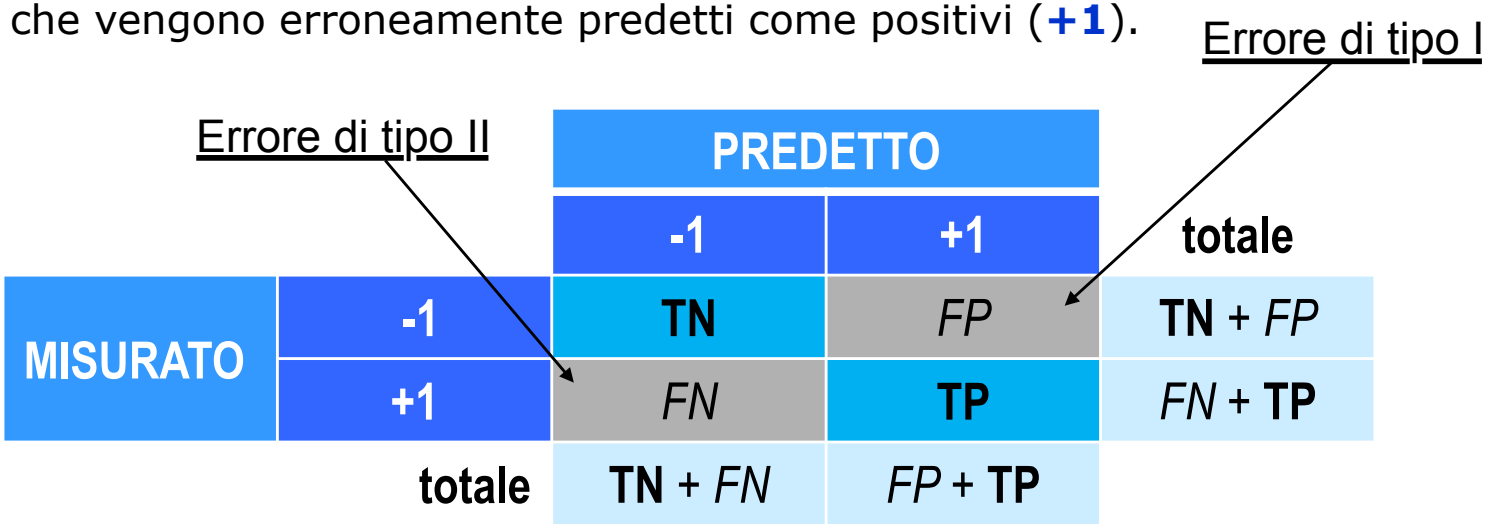
Consideriamo per comodità un problema di classificazione binaria con la variabile di classe che può assumere valore negativo (**-1**) o valore positivo (**+1**).



Valutazione Modelli: matrici di confusione

Gli *elementi riga-colonna della matrice di confusione* hanno il seguente significato:

- **TN**, veri negativi (**true negative**), numero di osservazioni con valore della classe negativo (**-1**) e che vengono correttamente predetti come negativi (**-1**),
- *FN*, falsi negativi (**false negative**), numero di osservazioni con valore della classe positivo (**+1**) e che vengono erroneamente predetti come negativi (**-1**),
- **TP**, veri positivi (**true positive**), numero di osservazioni con valore della classe positivo (**+1**) e che vengono correttamente predetti come positivi (**+1**),
- *FP*, falsi positivi (**false positive**), numero di osservazioni con valore della classe negativo (**-1**) e che vengono erroneamente predetti come positivi (**+1**).



Valutazione Modelli: matrici di confusione

Gli elementi della matrice di confusione consentono di definire i seguenti indicatori per la validazione di un modello di classificazione:

Accuratezza
accuracy

$$\text{acc} = \frac{TP + TN}{TP + FP + FN + TN} = \frac{TP + TN}{m}$$

% True Negative
specificity

$$\%TN = \frac{TN}{TN + FP}$$

% False Negative
miss rate

$$\%FN = \frac{FN}{FN + TP}$$

% True Positive
sensitivity

$$\%TP = \frac{TP}{FN + TP}$$

% False Positive
fall-out

$$\%FP = \frac{FP}{TN + FP}$$

		PREDETTO		
		-1	**+1**	**totale**
MISURATO	**-1**	**TN**	*FP*	**TN** + *FP*
	+1	*FN*	**TP**	*FN* + **TP**
	totale	**TN** + *FN*	*FP* + **TP**	

Valutazione Modelli: matrici di confusione

Inoltre, vengono definite le seguenti quantità per il valore **+1** della variabile di classe

Precision
positive predictive value

$$prc = \frac{TP}{FP + TP}$$ = Predittività (frequenza con cui vengono attribuiti TP)

Recall
sensitivity

$$rec = \frac{TP}{FN + TP}$$ = Sensibilità (frequenza con cui NON vengono attribuiti FN)
ad es. capacità di individuare soggetti malati

F-Measure

$$F_\beta = \frac{(1+\beta^2)\cdot prc \cdot rec}{\beta^2 \cdot prc + rec} \qquad \beta \in [0,+\infty)$$

regola l'importanza relativa di "**prc**" rispetto a "**rec**".

		PREDETTO		
		-1	**+1**	**totale**
MISURATO	**-1**	**TN**	*FP*	**TN** + *FP*
	+1	*FN*	**TP**	*FN* + **TP**
	totale	**TN** + *FN*	*FP* + **TP**	

Valutazione Modelli: matrici di confusione

Inoltre, vengono definite le seguenti quantità per il valore **+1** della variabile di classe

Precision $$prc = \frac{TP}{FP + TP}$$

Recall $$rec = \frac{TP}{FN + TP}$$

F-Measure $$F_1 = \frac{2 \cdot rec \cdot prc}{prc + rec} \qquad \beta = 1$$

		PREDETTO		
		-1	**+1**	**totale**
MISURATO	**-1**	**TN**	*FP*	**TN** + *FP*
	+1	*FN*	**TP**	*FN* + **TP**
	totale	**TN** + *FN*	*FP* + **TP**	

Valutazione Modelli: matrici di costo

È possibile, per determinati modelli di classificazione, assegnare una matrice detta

matrice dei costi di mis-classificazione

Il costo associato alle osservazioni classificate correttamente (**TP** e **TN**) viene usualmente posto a zero mentre viene utilizzato un costo positivo per la classificazione errata

FP e *FN*

in modo tale che il ***decision maker*** riesca ad implementare i propri obiettivi di analisi.

Ritornando al caso della diagnosi tramite la misurazione dei livelli di espressione genica il ***decision maker*** agirà in modo tale che il costo di una classificazione errata nel caso di false negative (*FN*) sia molto più elevato del costo di una classificazione errata nel caso di false positive (*FP*) al fine di individuare il massimo numero di individui affetti dalla malattia genetica e metterli al corrente dei rischi nei quali incorrono nel caso di procreazione.

Valutazione Modelli: matrici di costo

Una *matrice di costo* è una matrice quadrata che associa ad ogni elemento, riga-colonna, un numero reale che traduce una *valutazione economica o simbolica* specificata dal ***decision maker***.

		PREDETTO	
		+1	**-1**
MISURATO	**+1**	$C_{(+1,+1)}$	$C_{(-1,+1)}$
	-1	$C_{(+1,-1)}$	$C_{(-1,-1)}$

$C_{(i,j)}$ = costo della classificazione di posizione (i,j)

Valutazione Modelli: matrici di costo

Matrice di costo		PREDETTO	
		+1	-1
MISURATO	+1	-1	100
	-1	1	0

Modello A		PREDETTO	
		+1	-1
MISURATO	+1	150	40
	-1	60	250

Accuratezza = 0.8

Costo = 3,910

Modello B		PREDETTO	
		+1	-1
MISURATO	+1	250	45
	-1	5	200

Accuratezza = 0.9

Costo = 4,255

Valutazione Modelli: matrici di costo

Conteggio		PREDETTO	
		+1	**-1**
MISURATO	**+1**	**TP**	*FN*
	-1	*FP*	**TN**

Costo		PREDETTO	
		+1	**-1**
MISURATO	**+1**	p	q
	-1	q	p

Accuratezza proporzionale a ***Costo*** se

1. $C(+1,-1) = C(-1,+1) = q$
2. $C(+1,+1) = C(-1,-1) = p$

$$m = \mathbf{TP} + \mathit{FN} + \mathit{FP} + \mathbf{TN}$$

Accuratezza = (**TP**+**TN**)/m

Costo = $p \times (\mathbf{TP}+\mathbf{TN}) + q \times (\mathit{FN}+\mathit{FP})$

= $p \times (\mathbf{TP}+\mathbf{TN}) + q \times (m-\mathbf{TP}-\mathbf{TN})$

= $q \times m - (q-p) \times (\mathbf{TP}+\mathbf{TN})$

= $m \times [q - (q-p) \times$ ***Accuratezza***$]$

Valutazione Modelli: matrici di costo

Misure di performance sensibili al costo.

Precision $$\text{prc} = \frac{TP}{FP + TP}$$

Recall $$\text{rec} = \frac{TP}{FN + TP}$$

F-Measure $$F = \frac{\left(1+\beta^2\right)\cdot \text{prc} \cdot \text{rec}}{\beta^2 \cdot \text{prc} + \text{rec}} \qquad \beta \in [0,+\infty)$$

La ***Precision*** è distorta verso i termini $C_{(+1,+1)}$ e $C_{(+1,-1)}$.

La ***Recall*** è distorta verso i termini $C_{(+1,+1)}$ e $C_{(-1,+1)}$.

La ***F-Measure*** è distorta verso tutti i termini tranne il termine $C_{(-1,-1)}$.

Valutazione Modelli: grafico ROC

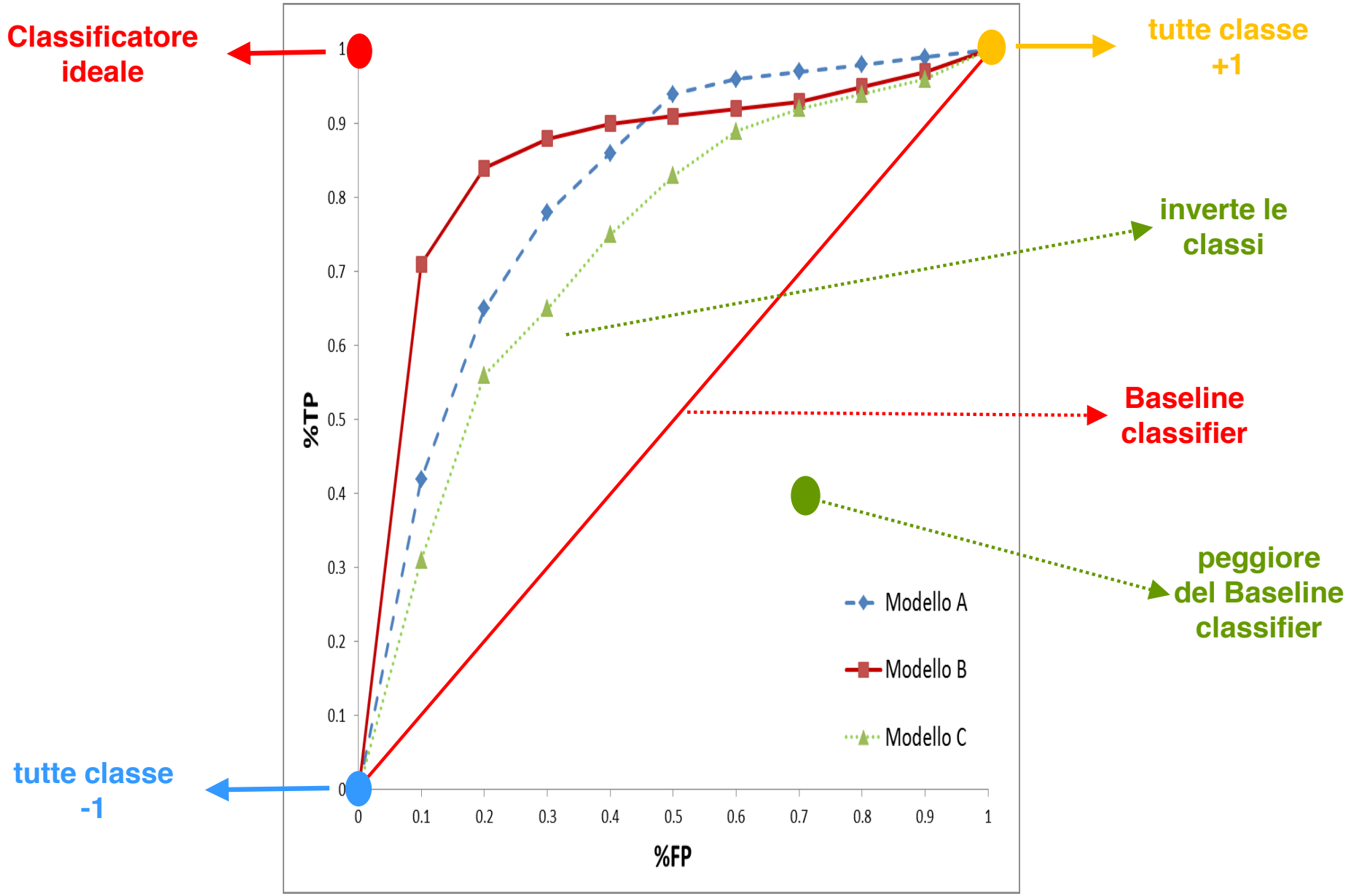
I grafici *Receiving Operating Characteristic* curve (**ROC**) consentono di valutare visivamente la qualità di un classificatore ed al contempo consentono di comparare tra loro diversi modelli di classificazione.

Forniscono la *sintesi delle informazioni ricavabili tramite una sequenza di matrici di confusione* e consentono di determinare il trade-off ottimale tra il numero di osservazioni positive classificate correttamente (**TP**) ed il numero di osservazioni negative classificate in modo errato (*FP*).

Offrono un'alternativa all'assegnazione dei costi di mis-classificazione. Il ROC è un grafico che riporta:

- sull'asse delle *ascisse* la percentuale di false positive (**%FP**) = 1 - sensitivity
- sull'asse delle *ordinate* la percentuale di true positive (**%TP**) = specificity

Valutazione Modelli: grafico ROC



Valutazione Modelli: grafico ROC

La maggior parte dei classificatori permette di effettuare tuning di alcuni parametri in modo da aumentare il numero di true positive (**TP**) a discapito di un conseguente quanto inevitabile aumento del numero di false positive (*FP*).

La curva ROC, per ogni modello di classificazione, viene ottenuta tramite le coppie di valori

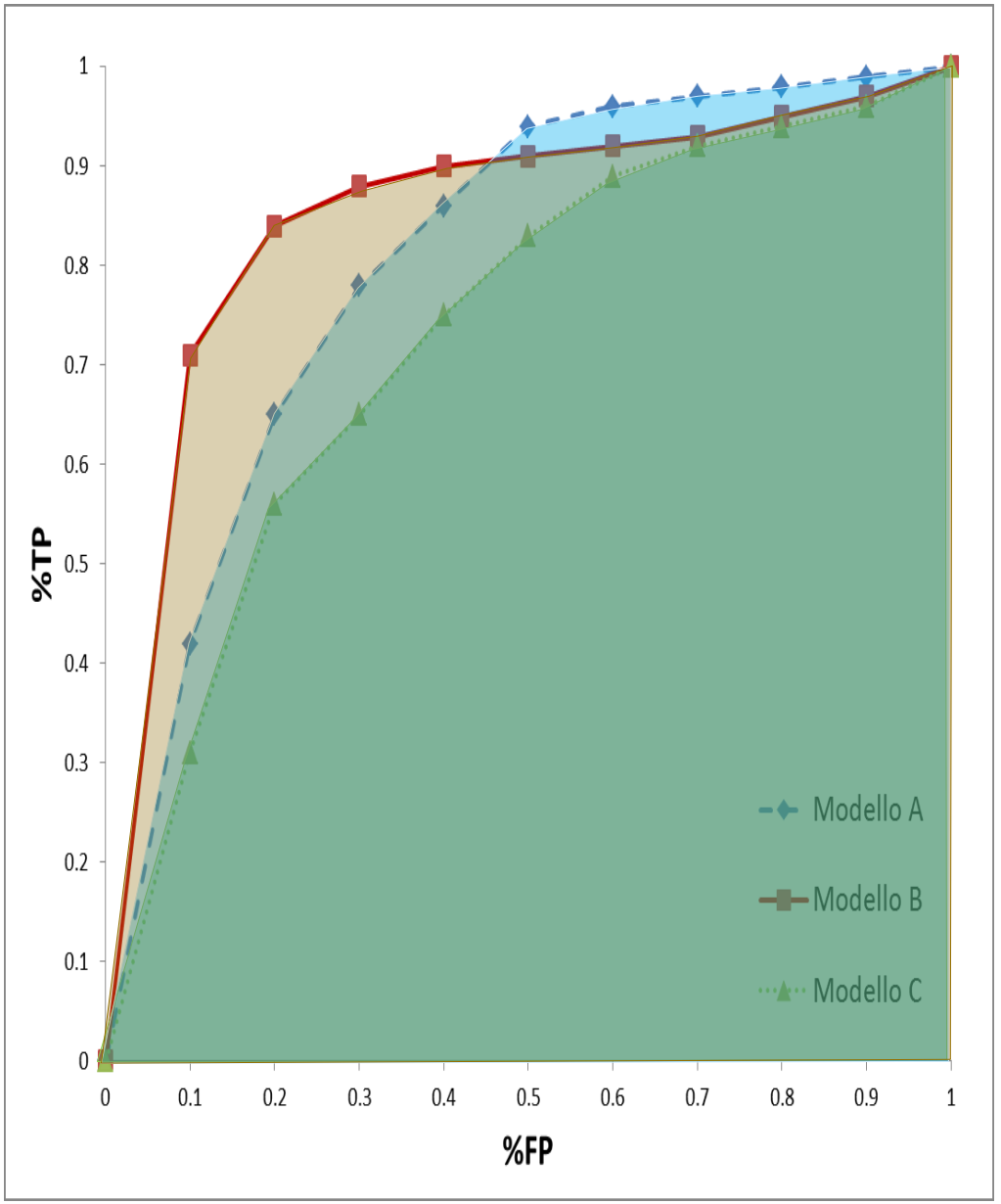
(**TP**, *FP*)

ottenute empiricamente in corrispondenza di diverse regolazioni dei parametri del modello di classificazione in analisi.

Se un classificatore non ammette parametri allora è univocamente associato ad un singolo punto nel piano del grafico ROC.

L'area sottesa dalla curva ROC rappresenta una misura sintetica che consente di comparare la qualità di diversi modelli di classificazione: è preferibile un classificatore cui competa un valore dell'area sottesa dalla curva ROC (*Area Under Curve*, **AUC**) maggiore.

Valutazione Modelli: grafico ROC

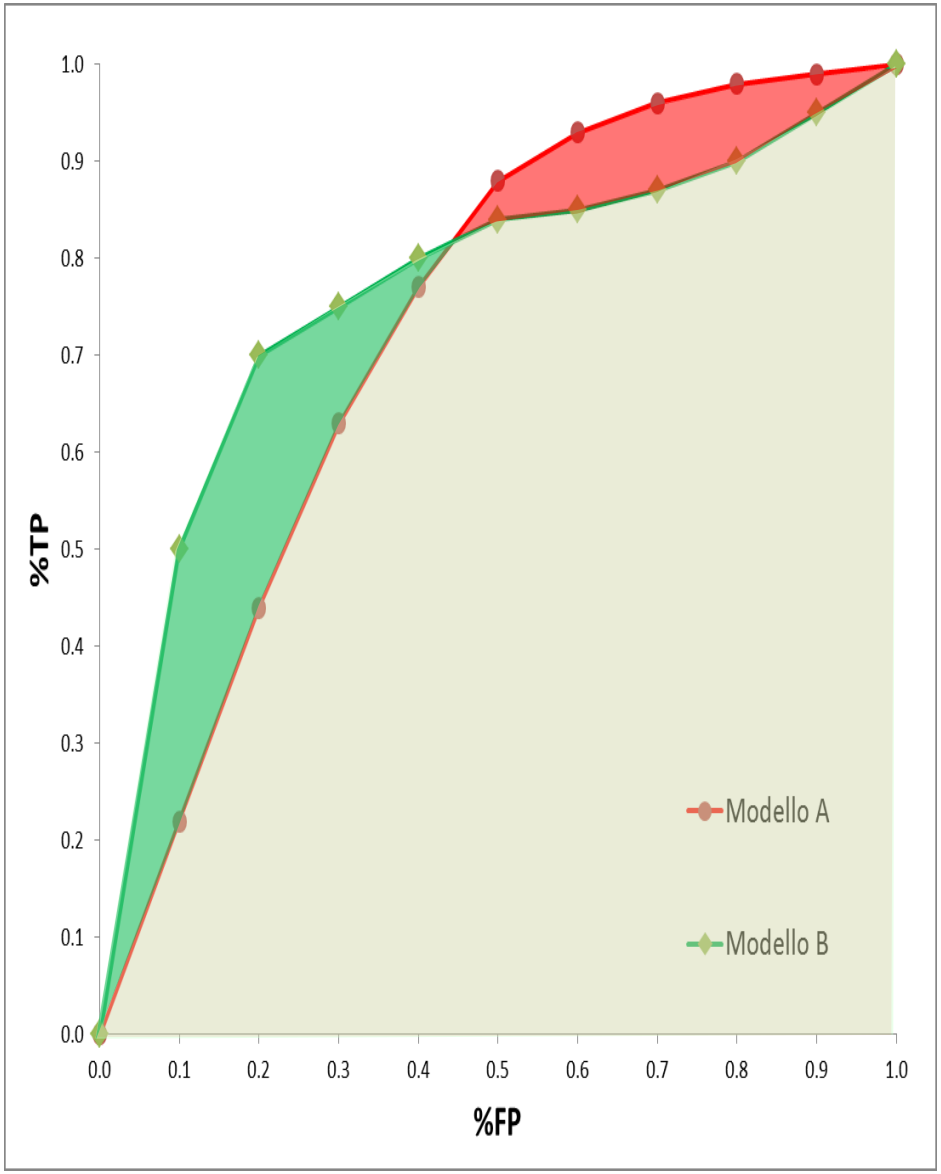


Valutazione Modelli: grafico ROC

Nessun modello è consistentemente migliore dell'altro.

Modello A migliore per grandi valori di %TP

Modello B migliore per piccoli valori di %TP



Valutazione Modelli: lift e guadagno cumulato

Il **lift** consente di valutare l'accuratezza di un modello tramite la densità delle osservazioni veramente positive appartenenti ad un campione di casi selezionati (classificati) come positivi dal modello medesimo. Indichiamo con *S* un sottoinsieme di osservazioni con cardinalità "s", osservazioni selezionate tramite il modello come appartenenti alla classe (**+1**).

		PREDETTO	
		+1	-1
MISURATO	+1	TP	*FN*
	-1	*FP*	TN

Il **lift** è definito come segue

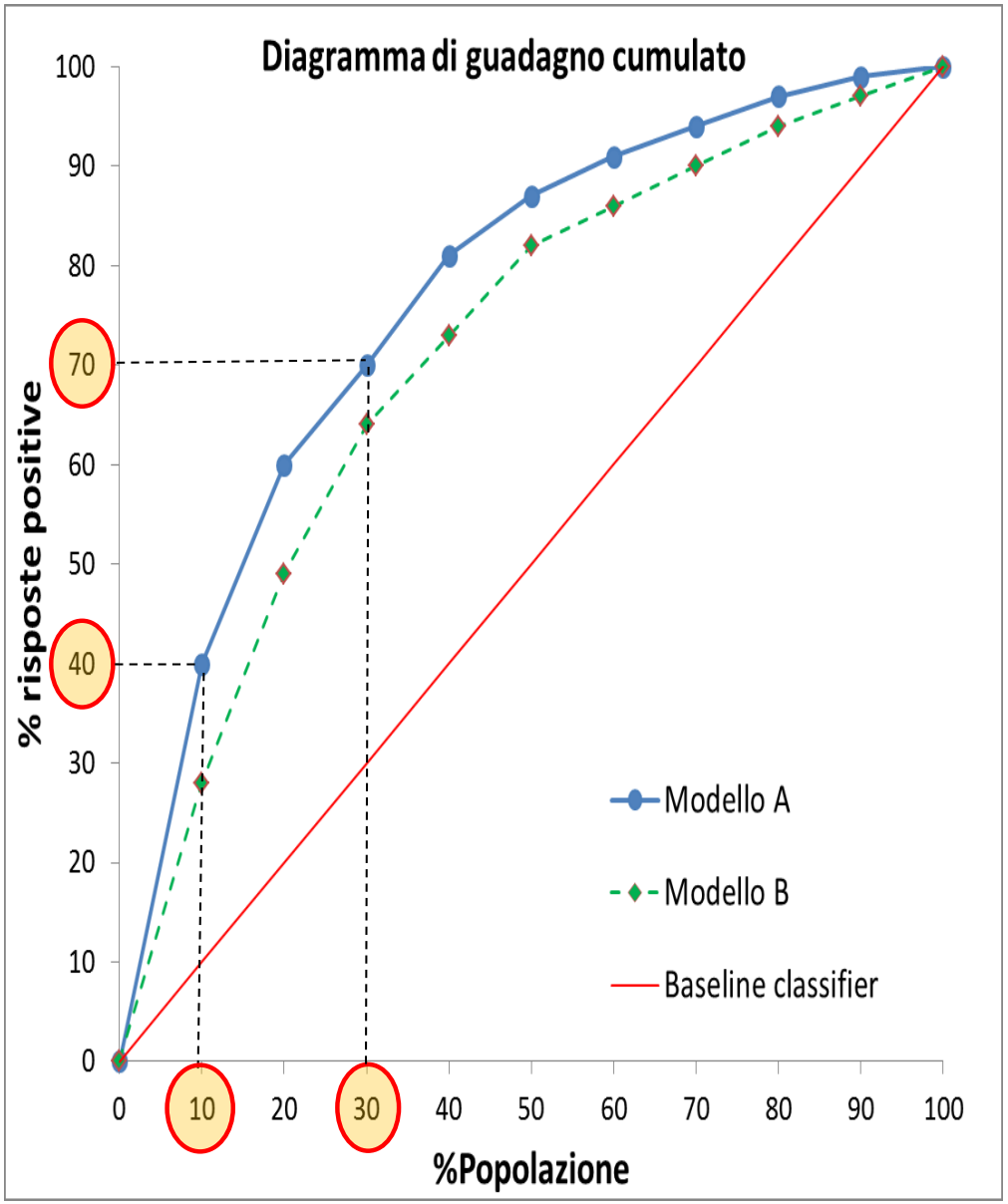
$$\text{lift}(+1) = \frac{a/s}{b/m} = \frac{\%\text{Pos}(S)}{\%\text{Pos}(D)} = \frac{\frac{TP(S)}{TP(S)+FP(S)}}{\frac{TP(D)}{TP(D)+FP(D)}} = \frac{\text{precision}(S)}{\text{precision}(D)}$$

Valutazione Modelli: lift e guadagno cumulato

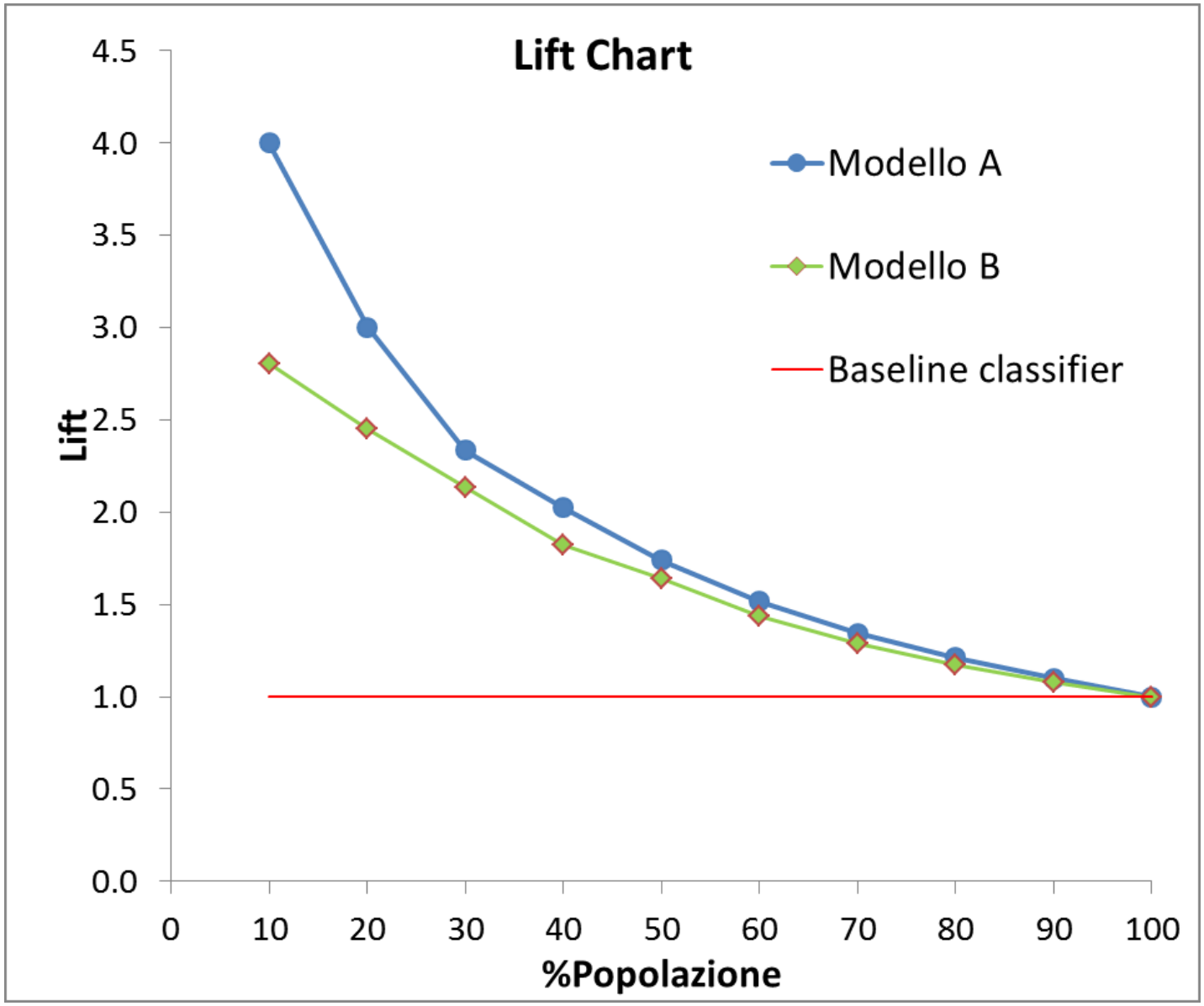
Un'azienda ha *m=1,000,000* clienti, e stima che la percentuale di clienti che risponderebbe ad una campagna promozionale è pari al *2%*.

Da una campagna rivolta a "*s*" clienti ci attendiamo un numero di risposte pari a *0.02s*.

La procedura di selezione casuale dei destinatari della campagna è rappresentata dal **Baseline classifier**, un classificatore per essere efficace deve offrire un guadagno rispetto a quanto è possibile ottenere tramite l'uso del **Baseline classifier**.

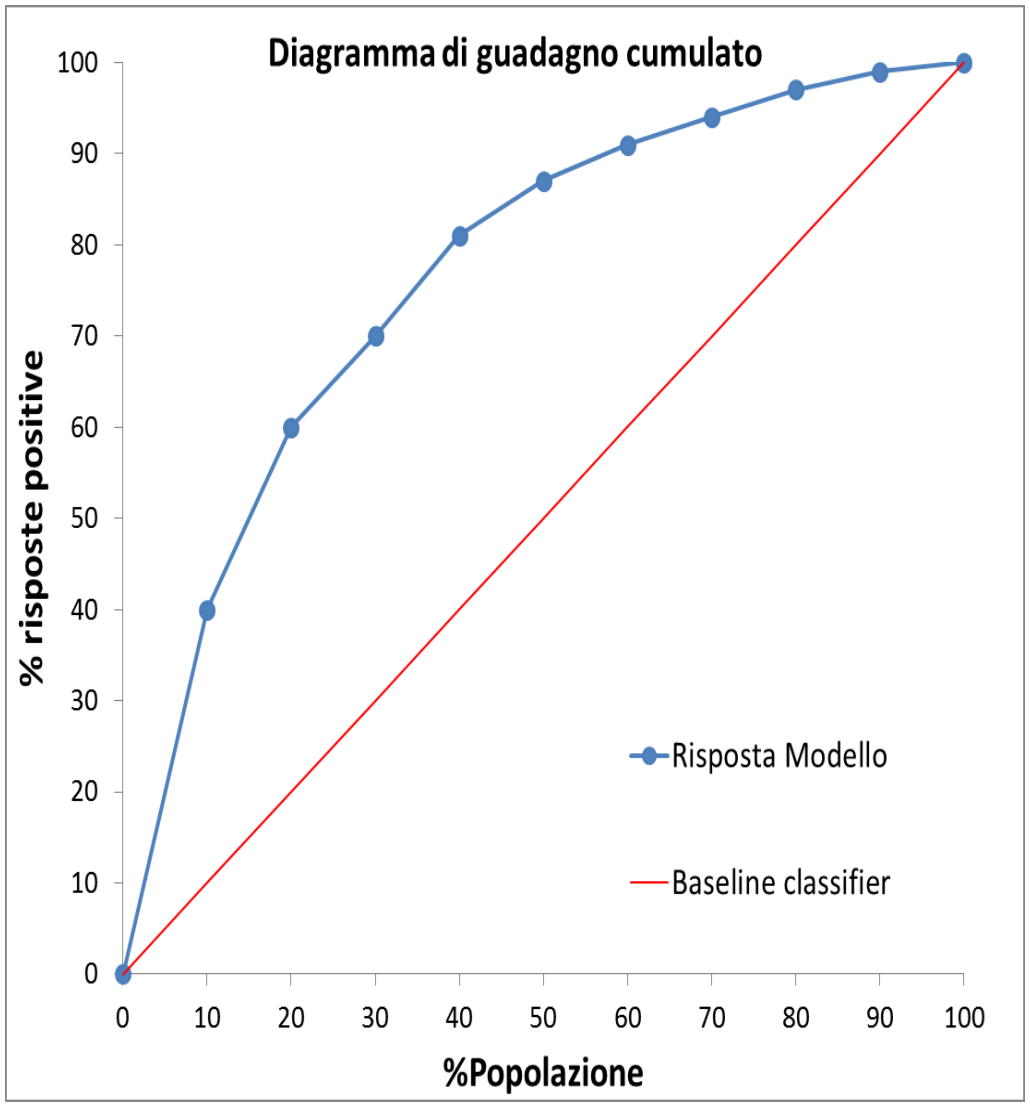


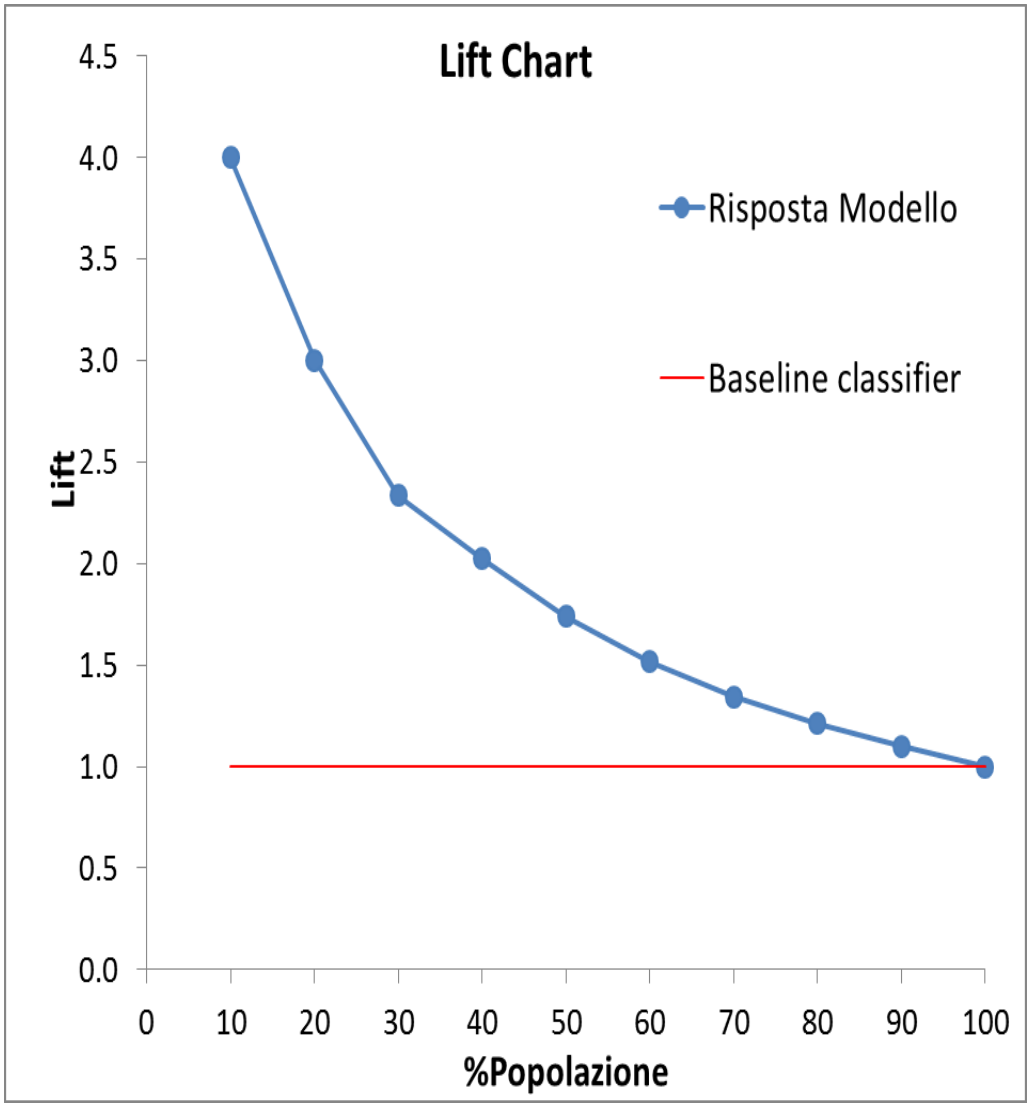
Valutazione Modelli: lift e guadagno cumulato



Valutazione Modelli: lift e guadagno cumulato

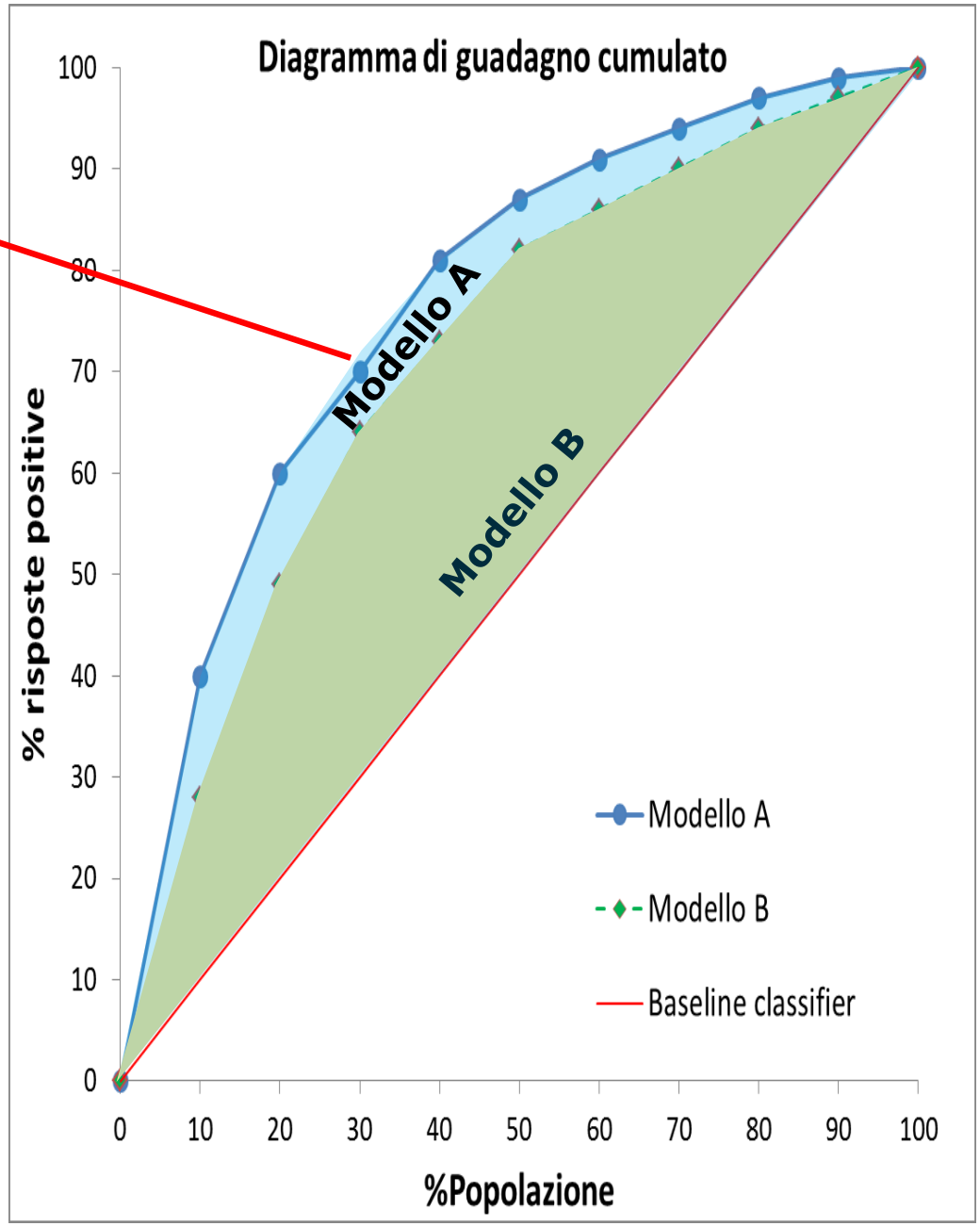
Il **Lift Chart** viene ricavato dalla curva di guadagno cumulato dividendo la percentuale di risposte positive ottenute col modello per la percentuale di risposte positive della selezione casuale.





Valutazione Modelli: lift e guadagno cumulato

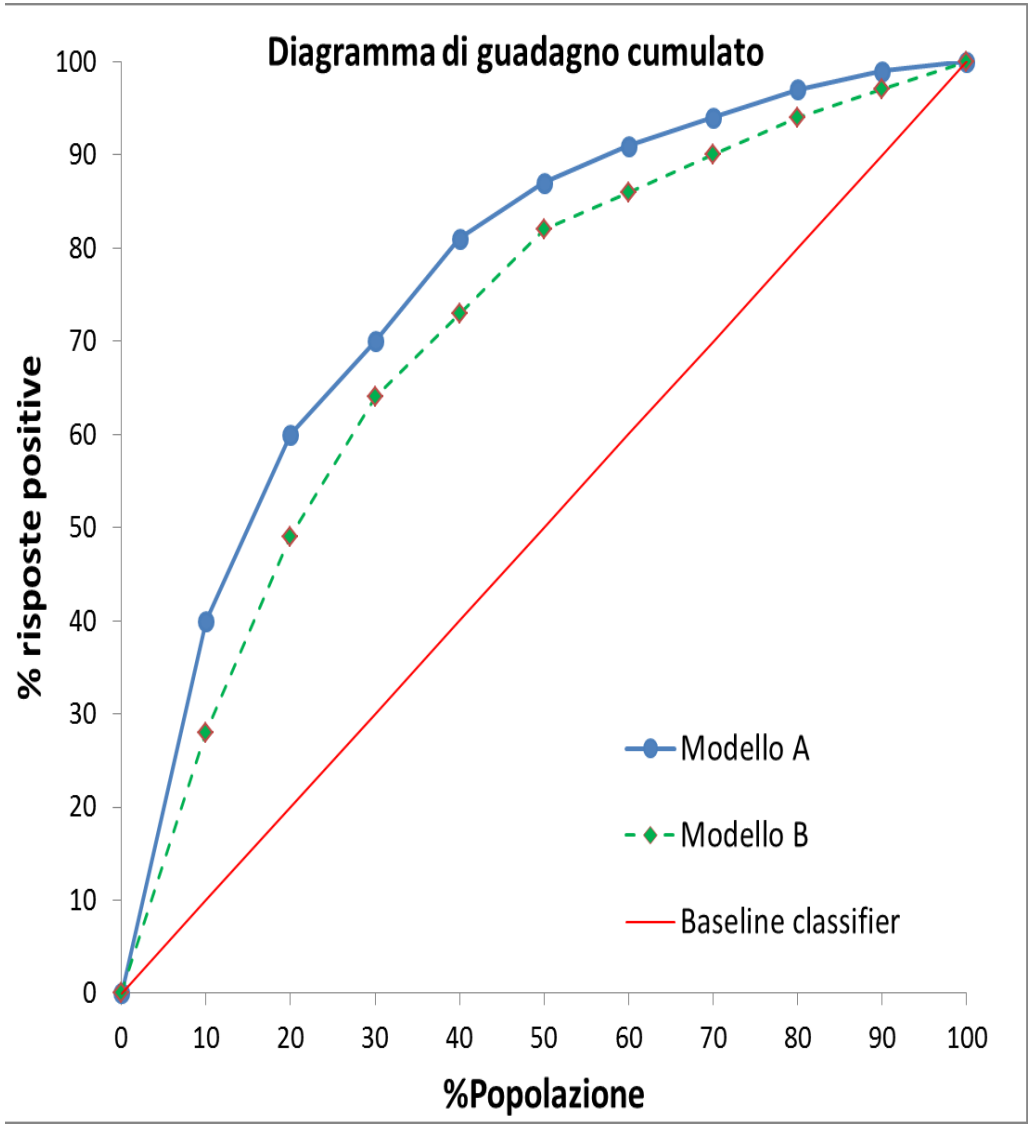
Modello ottimale

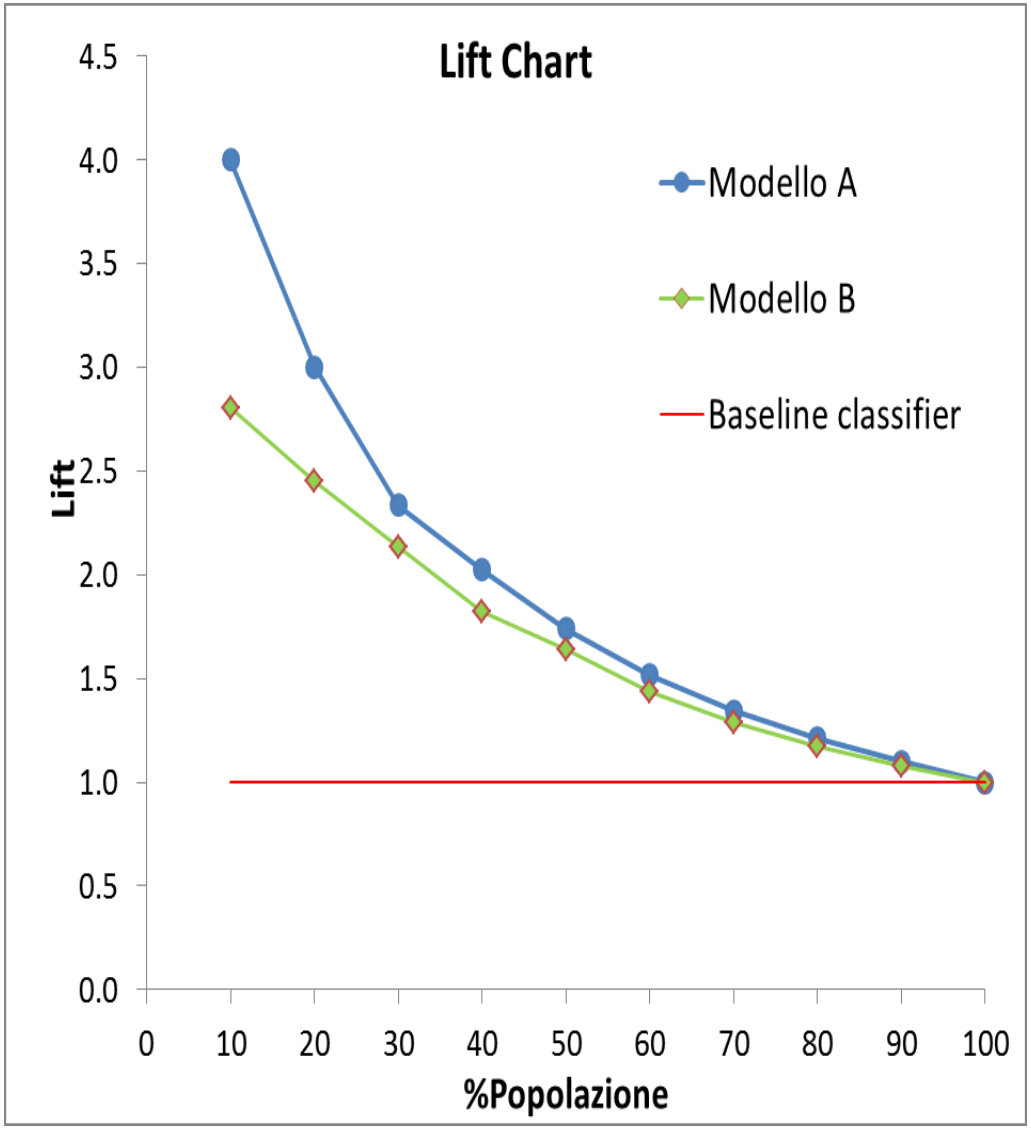


Il **diagramma di guadagno cumulato** consente di comparare due diversi modelli di classificazione, si considera l'area compresa tra la retta a 45% (**Baseline classifier**) e la curva associata alla risposta fornita dal modello (**Modello A**, **Modello B**).

Valutazione Modelli: lift e guadagno cumulato

Utilizzando il **Lift Chart** viene selezionato il modello che a parità del valore dell'asse delle ascisse è in grado di garantire il valore più elevato del lift.





Valutazione Modelli: intervallo fiduciario

Disponendo di due modelli di classificazione:

Modello A: accuratezza = 0.87, computata tramite 30 istanze

Modello B: accuratezza = 0.75, computata tramite 5,000 istanze

- *È possibile affermare che il **Modello A** è migliore del **Modello B**?*
- *Quale confidenza associamo alla stima di accuratezza dei due modelli?*
- *È possibile che la differenza in termini dell'accuratezza stimata sia imputabile a fluttuazioni casuali dovute alla dimensione del campione (numero di istanze) utilizzato per ottenere la stima?*

Valutazione Modelli: intervallo fiduciario

I dati di testing vengono utilizzati per selezionare uno tra diversi modelli di classificazione che sono stati sviluppati utilizzando lo stesso training set.

Ogni modello, dopo la fase di apprendimento, viene interrogato con le osservazioni che appartengono ad un test set.

Una misura generale di prestazione di un modello di classificazione è la *percentuale di errori*

$$\mathrm{err}(D_V) \approx \frac{\#\,\mathrm{errori}}{v}$$

Quale confidenza abbiamo in base a tale stima sul vero valore della percentuale di errori?

Possiamo rispondere a tale quesito ricorrendo a test di ipotesi ed alla computazione dell'intervallo di confidenza per la *percentuale di errori* *che*, nel caso in cui la dimensione del test set "v" risulti maggiore di 30, è una variabile aleatoria con distribuzione di probabilità ben approssimata da una distribuzione normale.

Valutazione Modelli: intervallo fiduciario

L'esito della predizione del modello di classificazione può essere modellato tramite una *variabile aleatoria di Bernoulli*, con due possibili esiti, *successo* (p) o *fallimento* (1-p).

La variabile aleatoria X ottenuta come somma di un insieme di variabili aleatorie distribuite secondo Bernoulli con probabilità di successo pari a "p" e tra loro indipendenti è distribuita secondo una *distribuzione Binomiale*, possiamo scrivere quanto segue

$$X \cong \mathrm{Bin}(v,p)$$

dove *X conta il numero di successi,* osservazioni *previste correttamente*, sui "v" casi che sono stati previsti dal modello di classificazione.

Lancio una moneta bilanciata per v=50 volte, quante volte dovrei ottenere esito "testa" ?

$$E[X] = v \cdot p = 50 \cdot 0.5 = 25$$

Disponendo di x (# di predizioni corrette) o equivalentemente, noto il valore dell'accuratezza

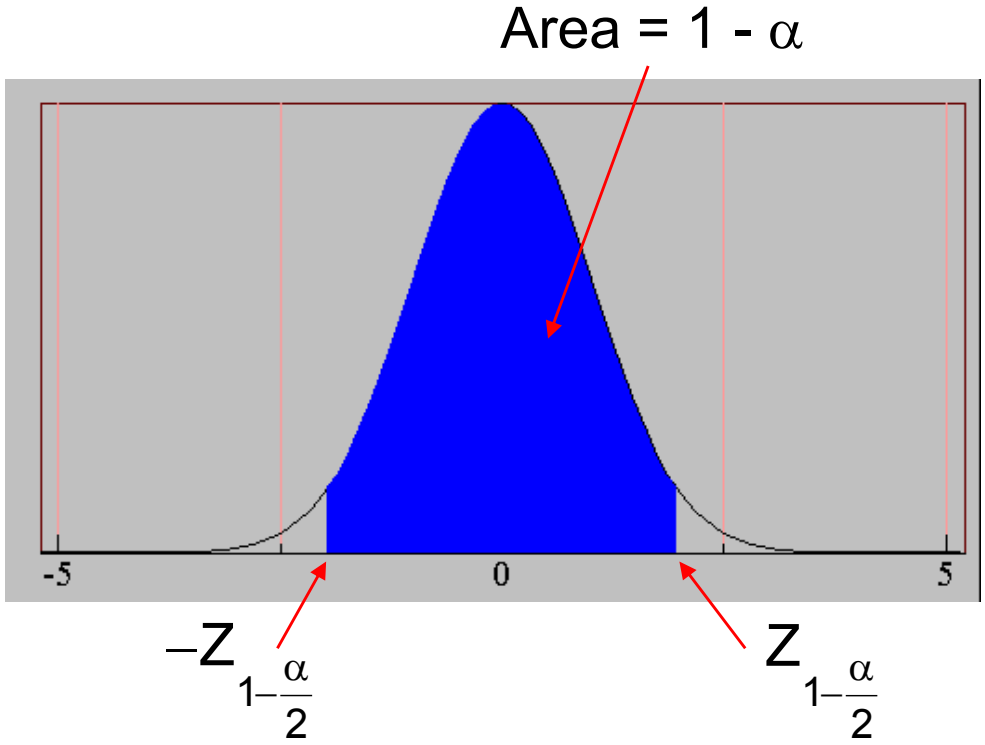
$$acc = \frac{x}{v}$$

ci chiediamo se sia possibile fornire valutazioni circa il valore vero "p" dell'accuratezza.

Valutazione Modelli: test di significatività

Sappiamo che per valori di "v>30" il teorema limite centrale ci consente di approssimare la distribuzione della variabile aleatoria "acc" tramite una distribuzione normale con media "p" e varianza pari a "p(1-p)/v"

$$P\left(-Z_{1-\frac{\alpha}{2}} < \frac{acc-p}{\sqrt{p(1-p)/v}} < Z_{1-\frac{\alpha}{2}}\right) = 1-\alpha$$



L'intervallo fiduciario per il parametro "p" è

$$\frac{2\times v\times acc + Z^2_{1-\frac{\alpha}{2}} \pm Z_{1-\frac{\alpha}{2}} \times \sqrt{Z^2_{1-\frac{\alpha}{2}} + 4\times v\times acc - 4\times v\times acc^2}}{2\times (v + Z^2_{1-\frac{\alpha}{2}})}$$

Valutazione Modelli: test di significatività

Pertanto, possiamo affermare che il vero valore dell'accuratezza del modello di classificazione "p" verifica la seguente relazione:

$$P\left(\frac{2\times v\times acc+Z^2_{1-\frac{\alpha}{2}}-Z_{1-\frac{\alpha}{2}}\times\sqrt{Z^2_{1-\frac{\alpha}{2}}+4\times v\times acc-4\times v\times acc^2}}{2\times(v+Z^2_{1-\frac{\alpha}{2}})}\le p\le\frac{2\times v\times acc+Z^2_{1-\frac{\alpha}{2}}+Z_{1-\frac{\alpha}{2}}\times\sqrt{Z^2_{1-\frac{\alpha}{2}}+4\times v\times acc-4\times v\times acc^2}}{2\times(v+Z^2_{1-\frac{\alpha}{2}})}\right)=1-\alpha$$

Consideriamo un modello di classificazione che raggiunga un'accuratezza pari a 0.8 stimata tramite un campione di 100 osservazioni:

v=100, acc = 0.8

Fissiamo un livello di confidenza del 95%, α=0.05

v	50	100	500	1,000	5,000
lower limit	0.670	0.711	0.763	0.774	0.789
upper limit	0.888	0.866	0.833	0.824	0.811

1-α	Z
0.99	2.58
0.98	2.33
0.95	1.96
0.90	1.65

Valutazione Modelli: confronto modelli

Per confrontare le prestazioni di due modelli di classificazione addestrati sullo stesso training dataset, ovvero per rispondere alla seguente domanda

*Esiste una differenza significativa tra l'errore percentuale ottenuto dal **Modello 1** e l'errore percentuale ottenuto dal **Modello 2** addestrati utilizzando lo stesso training dataset?*

è possibile procedere in tre modi differenti:

1) *l'accuratezza dei modelli è comparata utilizzando due test set indipendenti selezionati a partire da un insieme di dati*
2) *viene utilizzato lo stesso test dataset, il confronto è basato su una comparazione caso per caso*
3) *lo stesso test dataset viene utilizzato per comparare la correttezza di classificazione complessiva dei due modelli.*

Ipotizziamo che n_1 ed n_2 (dimensioni di due dataset di test differenti) siano sufficientemente grandi, i tassi di errore e_1 ed e_2 son approssimabili tramite distribuzione normale.

Se la differenza dei tassi di errore è

$$d = e_1 - e_2$$

Sarà anche essa distribuita secondo una normale con

$$\text{media} = d_t \qquad \text{varianza} = \sigma_d^2$$

La varianza di **d** è stimabile come segue

$$\sigma_d^2 \cong \hat{\sigma}_d^2 = \frac{e_1 \cdot (1 - e_1)}{n_1} + \frac{e_2 \cdot (1 - e_2)}{n_2}$$

L'intervallo di confidenza per la vera differenza dei tassi di errore d_t è

$$\left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

$$\left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

Se l'intervallo di confidenza contiene lo 0, concludiamo che la differenza osservata d non è statisticamente significativa al livello α.

$$0 \in \left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

$$\left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

I modelli M_1 ed M_2 non sono significativamente differenti in termini di tasso di errore che ottengono al livello α.

Se l'intervallo di confidenza contiene lo 0, concludiamo che la differenza osservata d non è statisticamente significativa al livello α.

$$0 \in \left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

$$\left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

Se il valore 0 è più piccolo dell'estremo inferiore dell'intervallo di confidenza allora il classificatore M_1 è migliore del classificatore M_2 con livello α.

$$0 \in \left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

$$0 < d - z_{1-\alpha/2} \cdot \hat{\sigma}_d$$

$$\left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

Se il valore 0 è più grande dell'estremo superiore dell'intervallo di confidenza allora il classificatore M_2 è migliore del classificatore M_1 con livello α.

$$0 \in \left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

$$0 < d - z_{1-\alpha/2} \cdot \hat{\sigma}_d$$

$$0 > d + z_{1-\alpha/2} \cdot \hat{\sigma}_d$$

$$\left(d - z_{1-\alpha/2} \cdot \hat{\sigma}_d, d + z_{1-\alpha/2} \cdot \hat{\sigma}_d\right)$$

CLASSIFICAZIONE

MODELLI DI REGRESSIONE

Classificazione

Parte dei contenuti della presente lezione sono tratti dai testi elencati di seguito.

Carlo Vercellis (2006). *Business intelligence: modelli matematici e sistemi per le decisioni*, McGraw-Hill.

Modelli di Regressione

Regressione Logistica Binomiale

Riconduce i problemi di classificazione binaria alla regressione lineare sfruttando un'opportuna trasformazione dei dati.

Ipotizziamo che la classe Y assuma valori in {0,1}, allora il modello di regressione logistica rappresenta la probabilità a posteriori della classe Y dato il vettore delle variabili esplicative **X**

$$P(Y \mid \underline{X})$$

tramite una funzione logistica

$$P(Y=0 \mid \underline{X}=\underline{x}) = \frac{1}{1+\exp(\underline{w}\cdot\underline{x})} \qquad P(Y=1 \mid \underline{X}=\underline{x}) = \frac{\exp(\underline{w}\cdot\underline{x})}{1+\exp(\underline{w}\cdot\underline{x})}$$

dove il vettore **w** è il vettore dei parametri del modello di regressione logistica ed ha la stessa dimensione del vettore **X** delle variabili esplicative.

Modelli di Regressione

Le due relazioni precedenti possono essere opportunamente combinate per ottenere la seguente relazione

$$\log \frac{P(Y=1 \mid \underline{X}=\underline{x})}{P(Y=0 \mid \underline{X}=\underline{x})} = \underline{w} \cdot \underline{x}$$

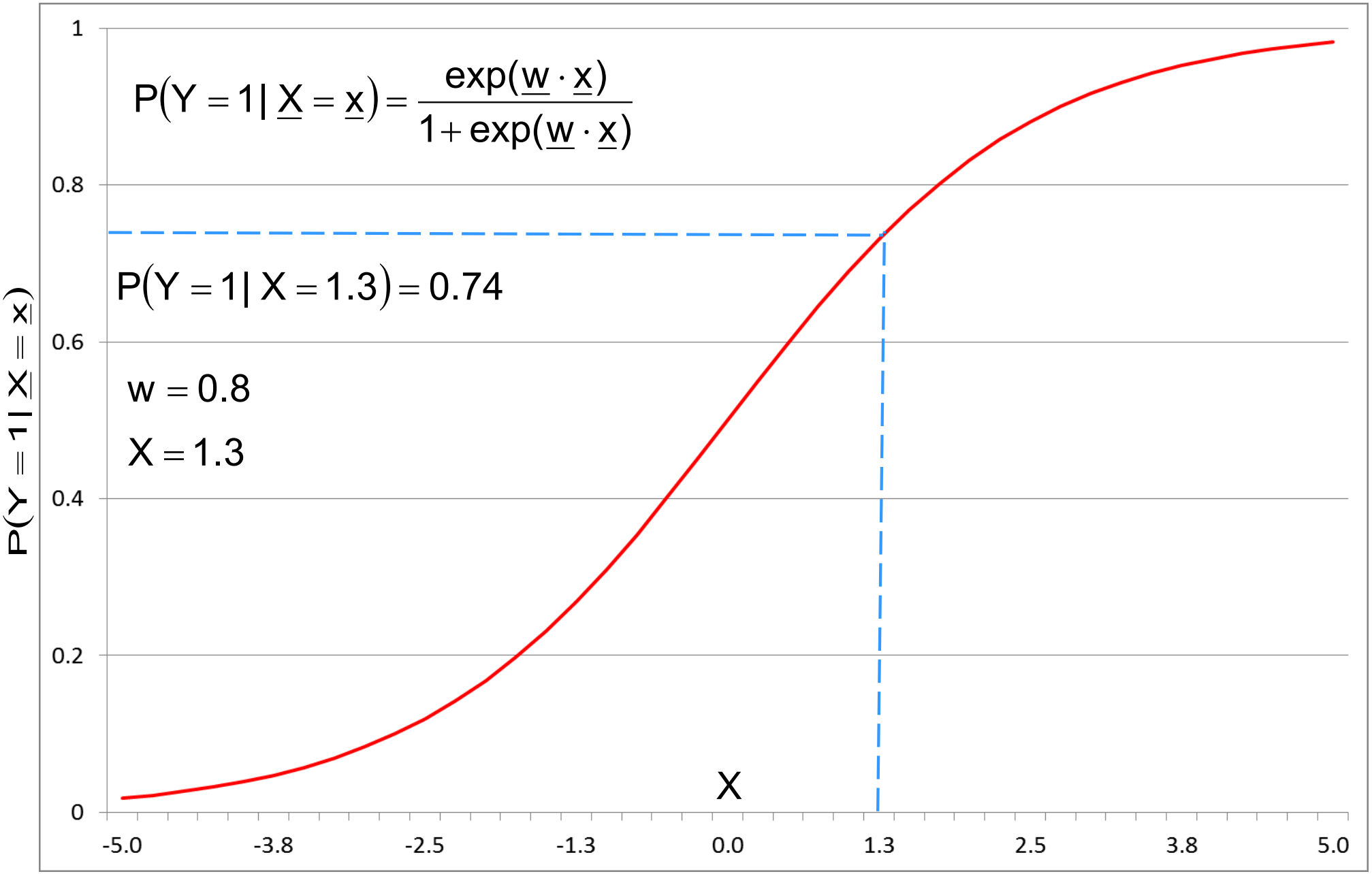
Pertanto, se poniamo

$$Z = \log \frac{P(Y=1 \mid \underline{X}=\underline{x})}{P(Y=0 \mid \underline{X}=\underline{x})}$$

Il problema di classificazione binaria viene ricondotto a quello della identificazione di un modello di regressione lineare tra la variabile dipendente Z e le variabili esplicative $\underline{X}$.

Dopo aver determinato i coefficienti del modello di regressione, aver verificato la significatività del medesimo si applica l'antitrasformazione della variabile Z per utilizzare successivamente il modello a fini previsivi su una qualsiasi nuova istanza $\underline{x}$ del vettore delle variabili esplicative.

Modelli di Regressione



Modelli di Regressione

I modelli di regressione logistica soffrono delle stesse problematiche delle quali soffrono i modelli di regressione lineare.

Il fenomeno della multicollinearità, che pregiudica la significatività dei coefficienti di regressione, richiede di affrontare una fase di selezione delle variabili esplicative (feature selection).

Alcune caratteristiche note del modello di regressione logistica sono

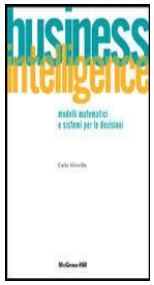
- *accuratezza usualmente inferiore a quella di altri modelli di classificazione*
- *maggiore laboriosità nella fase di costruzione rispetto ad altri modelli di classificazione*
- *estremamente complesso trattare dataset estesi, numero di variabili esplicative, numerosità delle istanze.*

CLASSIFICAZIONE

MODELLI DI SEPARAZIONE

Classificazione

Parte dei contenuti della presente lezione sono tratti dalle seguenti fonti



Carlo Vercellis (2006). *Business intelligence: modelli matematici e sistemi per le decisioni*, McGraw-Hill.

Modelli di Separazione

I modelli di separazione che presenteremo sono i seguenti:

- *Artificial Neural Networks*
- *Support Vector Machines*

Nello specifico per quanto riguarda le *Artificial Neural Networks*, data la ricchezza di tale classe di modelli connessionisti, presenteremo nel dettaglio solamente i seguenti modelli di classificazione:

- *Feedforward Neural Networks*
- *Radial Basis Function Networks*

Per quanto riguarda le *Support Vector Machines*, presenteremo i seguenti modelli:

- *Linear hard margin*
- *Linear soft margin*
- *Non-linear*

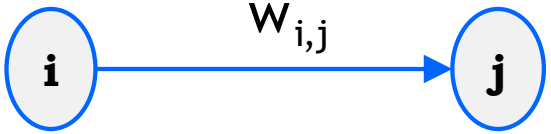
Feedforward Neural Networks

Ogni *neurone ha* tipicamente un *insieme* di

- *neuroni di input*
- *neuroni di output*

Neurone "i" è *input* del neurone "j". Neurone "j" è *output* per il neurone "i".

Neuroni collegati in modo *orientato* dalla sinapsi che è *associata* ad un *valore reale* (peso).



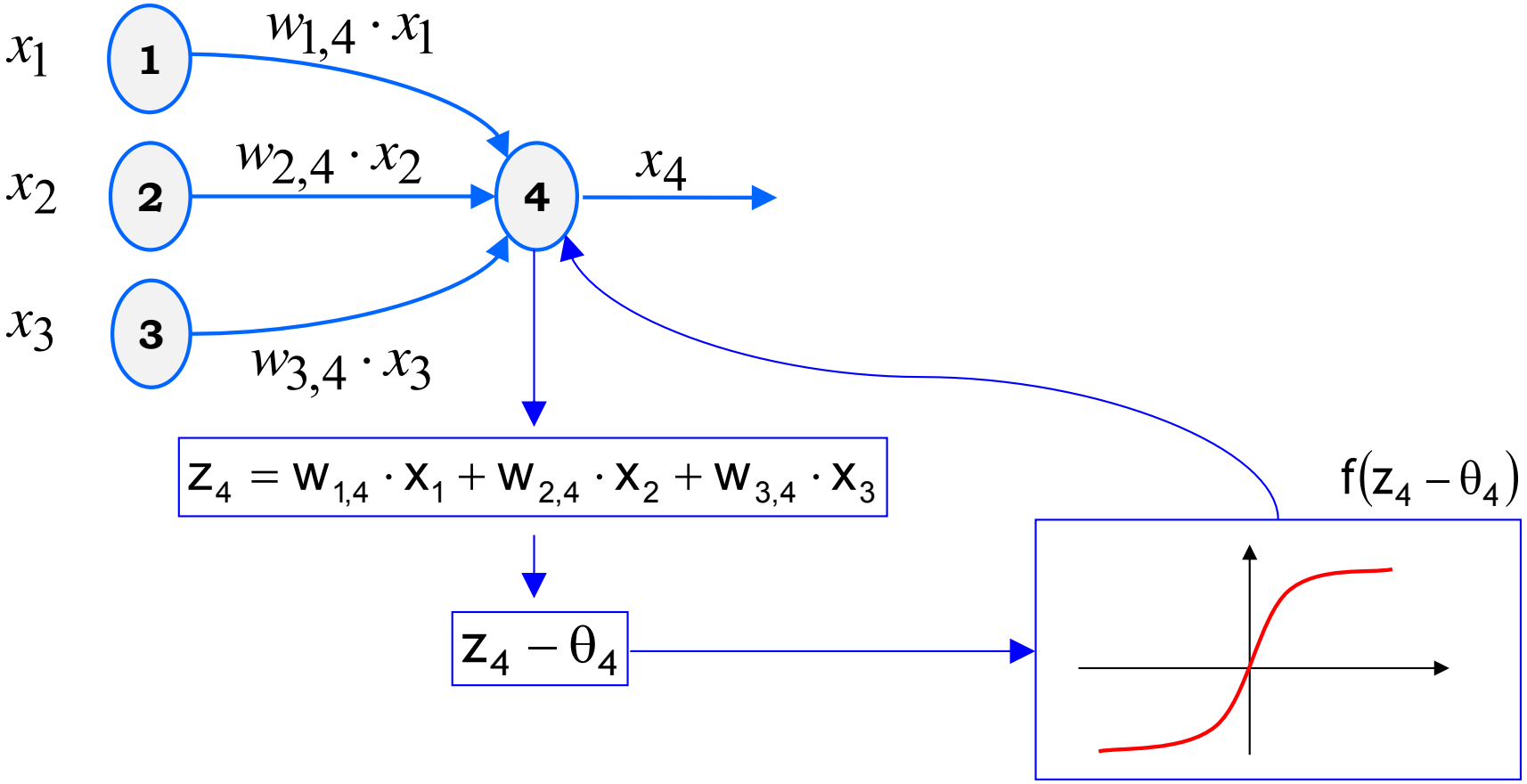
Ogni *neurone* è *caratterizzato* da due elementi:

- *soglia, bias o threshold*
- *funzione di attivazione o di trasferimento*

Modelli di Separazione: reti neurali

Ogni neurone:

- riceve segnali da altri neuroni (neuroni di input)
- invia segnali ad altri neuroni (neuroni di output)



Formalmente il *neurone* "j" *calcola* la seguente funzione:

$$y_j = f\left(\sum_{i=1}^{n} w_{i,j} \cdot x_i - \theta_j\right)$$

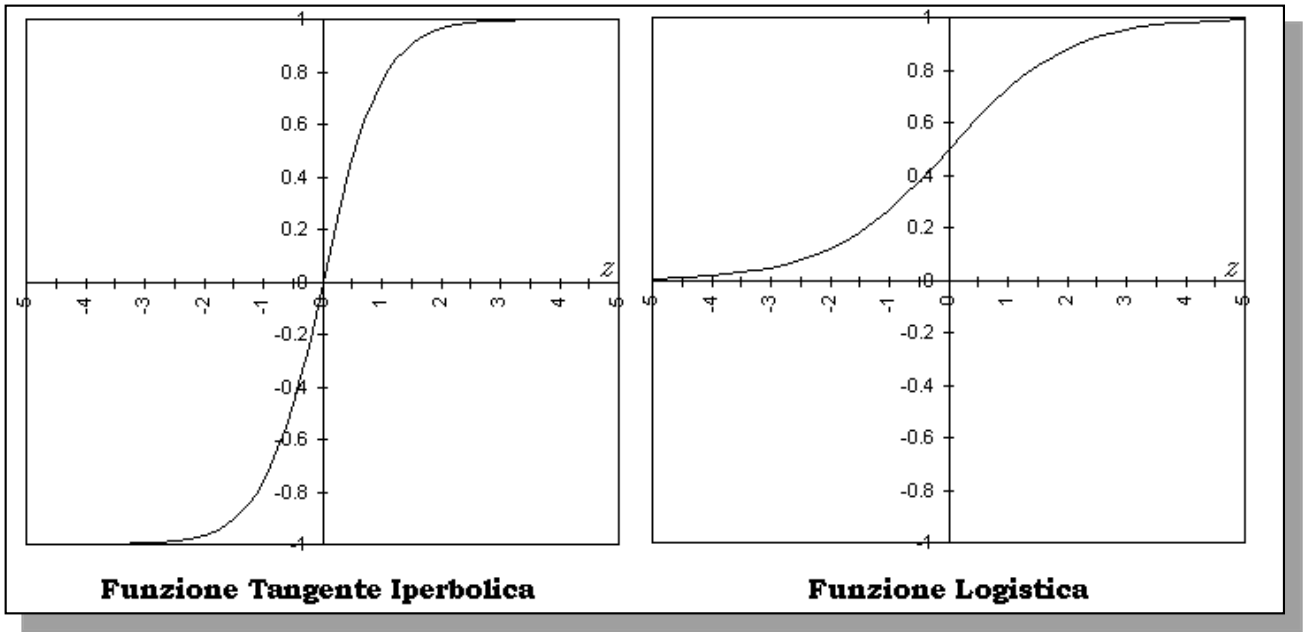
Quali sono le *principali funzioni di attivazione* ?

Tangente Iperbolica

$$f(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}$$

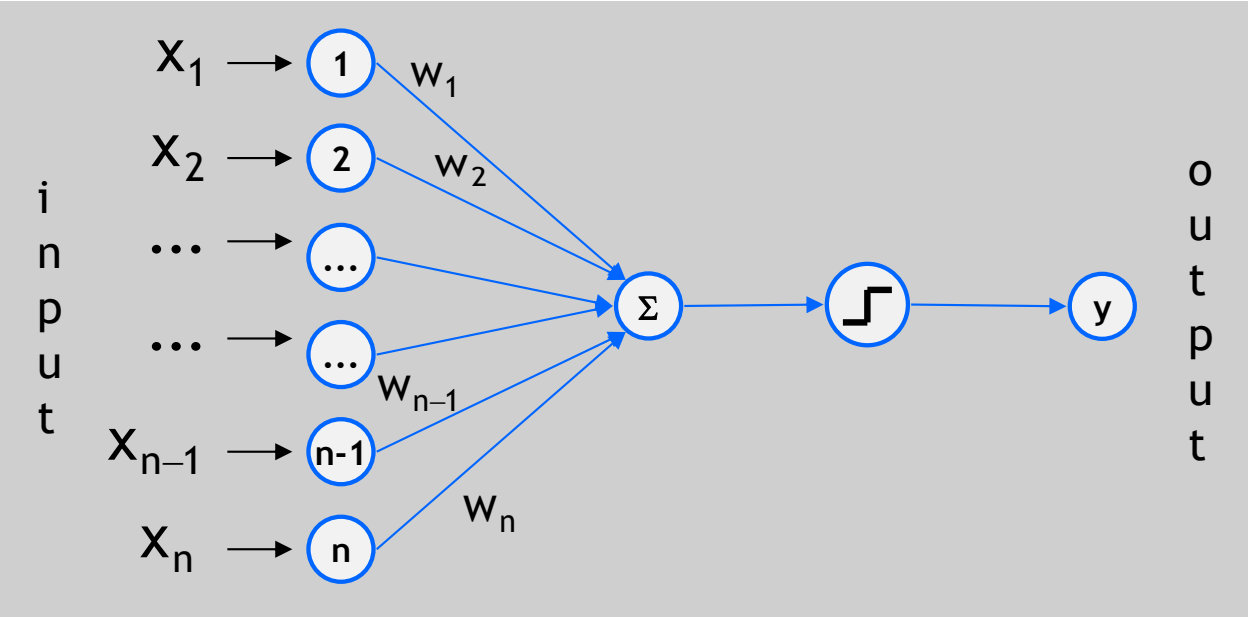
Logistica

$$f(z) = \frac{1}{1 + \exp(-z)}$$



Percettrone Lineare

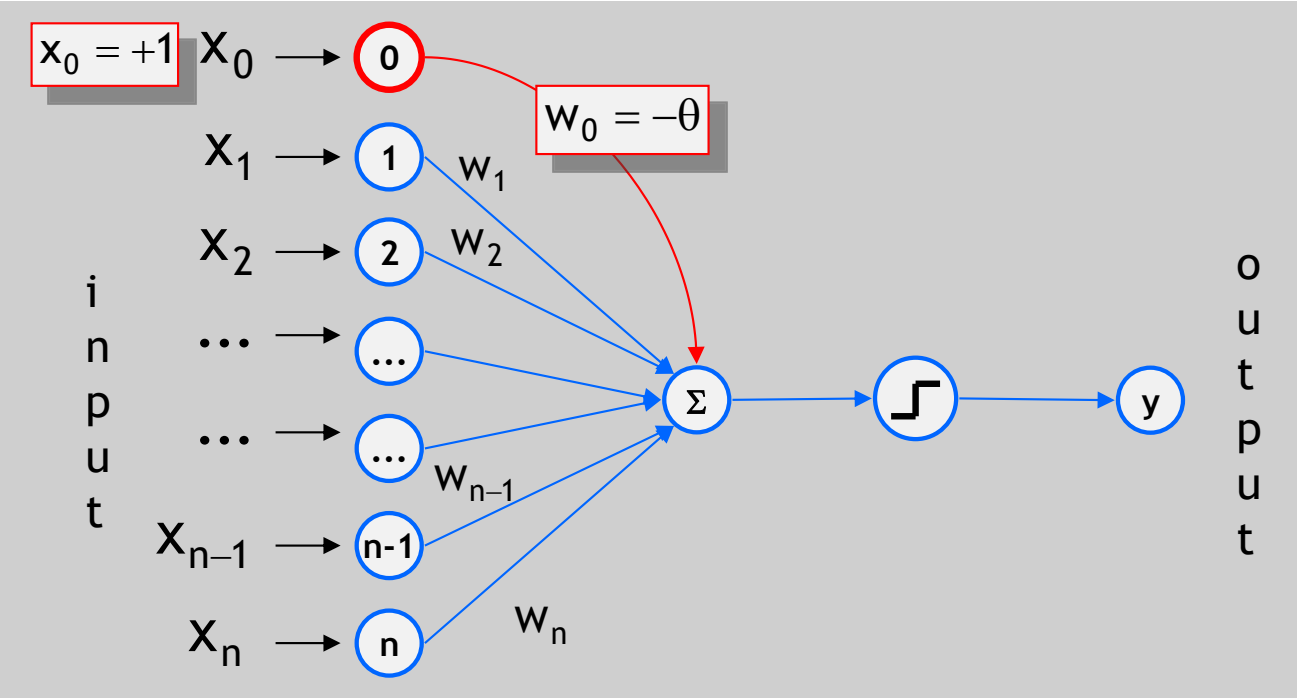
Il *modello più elementare* è il *Percettrone Lineare*.



$$a = \sum_{i=1}^{n} w_i x_i = w_1 x_1 + w_2 x_2 + \ldots + \ldots + w_n x_n$$

$$y = \begin{cases} 1 & \text{se } a \geq \theta \\ 0 & \text{altrimenti} \end{cases}$$

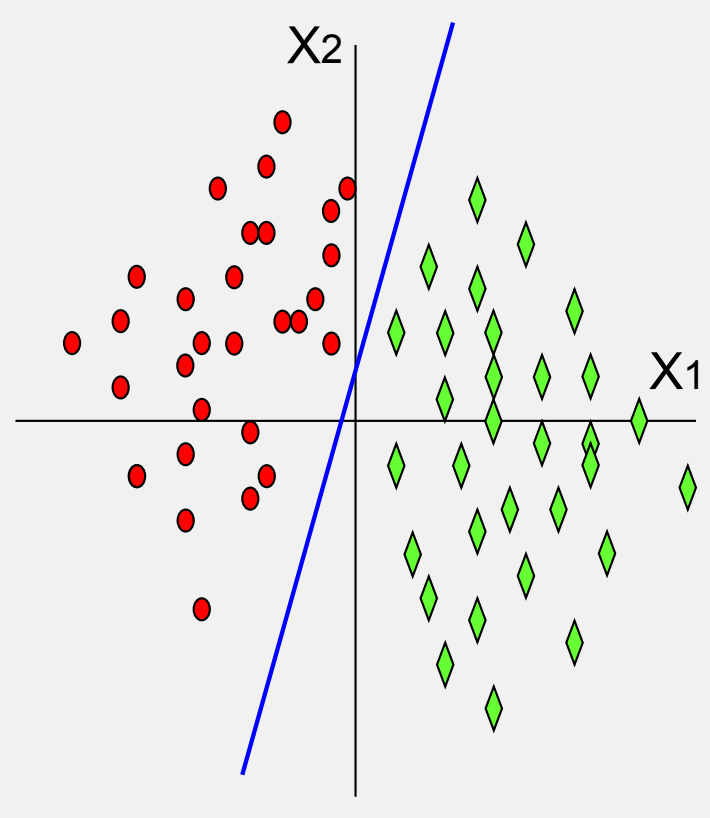
Rappresentabile in modo generale.

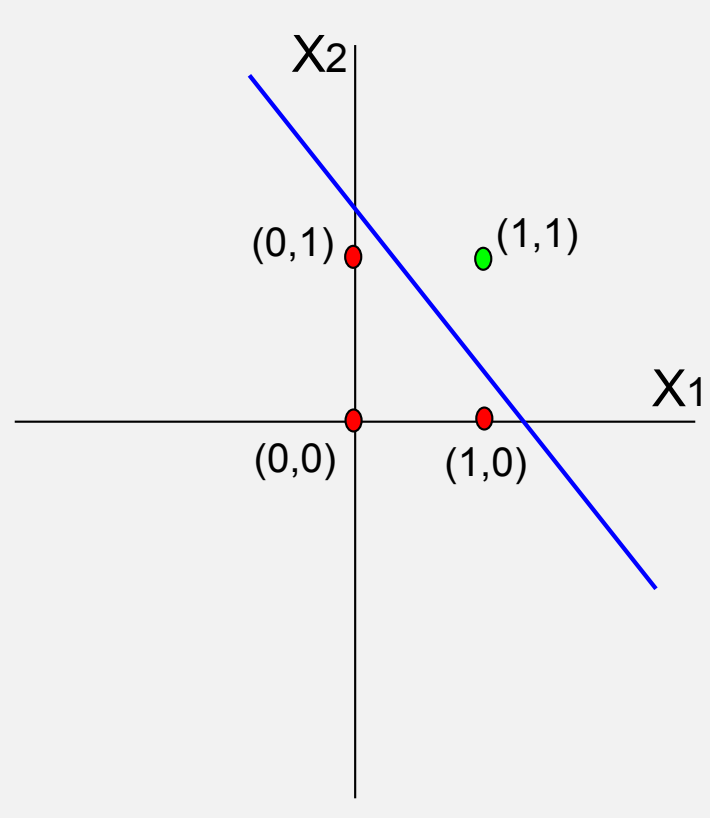


$$a = \sum_{i=0}^{n} w_i x_i = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots + \ldots + w_n x_n$$

$$y = \begin{cases} 1 & se\ a \geq 0 \\ 0 & altrimenti \end{cases}$$

Il percettrone monostrato *implementa* un *iperpiano* nello *spazio* "*n-dimensionale*".

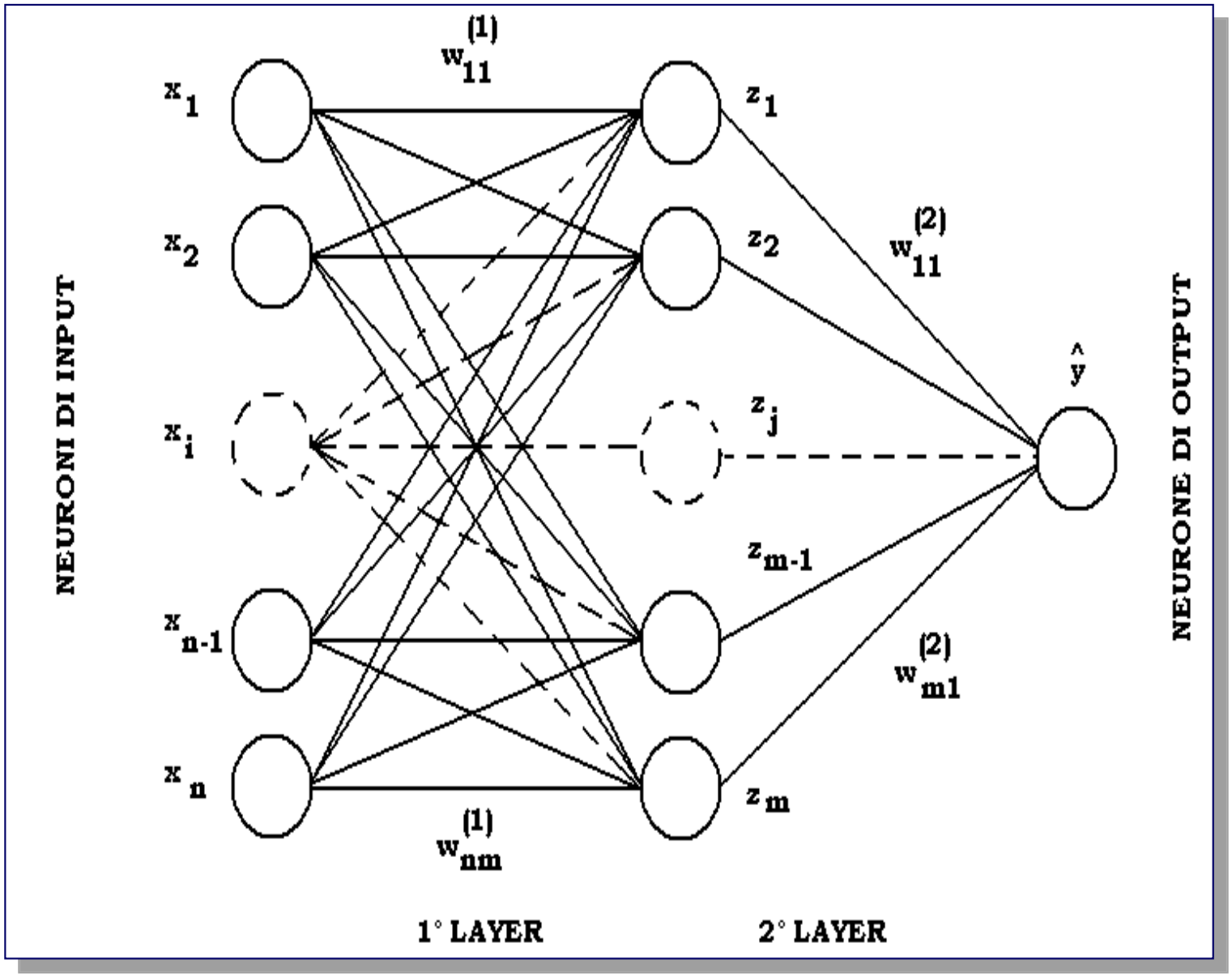




funzione AND

Il *Percettrone* è in grado di *apprendere qualsiasi funzione* ?

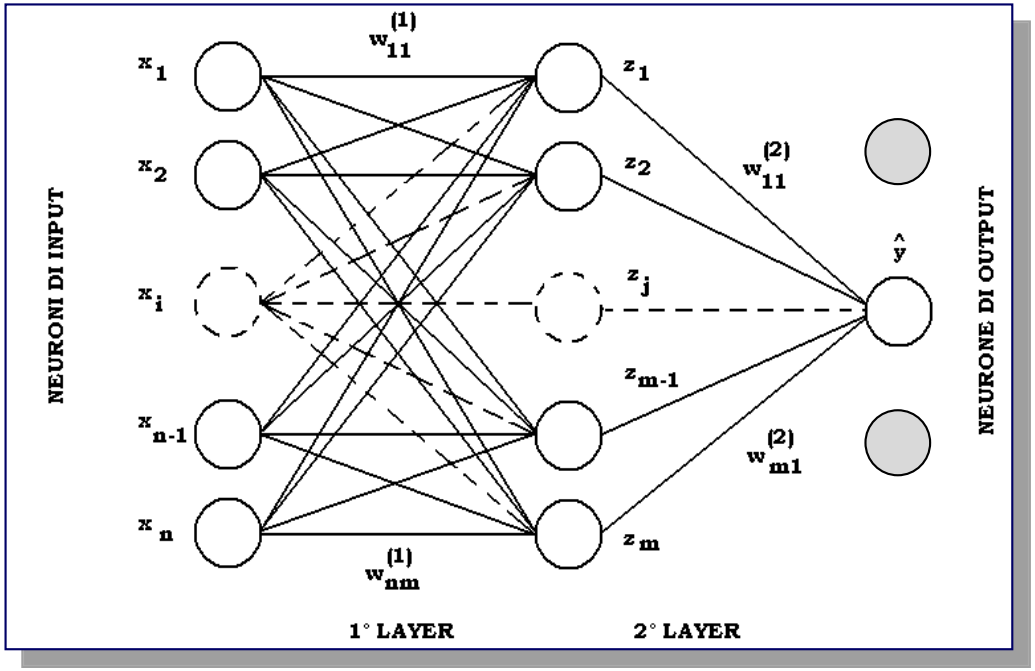
Percettrone Multi-strato o Feedforward Neural Network



Una FNN come quella riportata in precedenza calcola la seguente funzione dell'input **x**

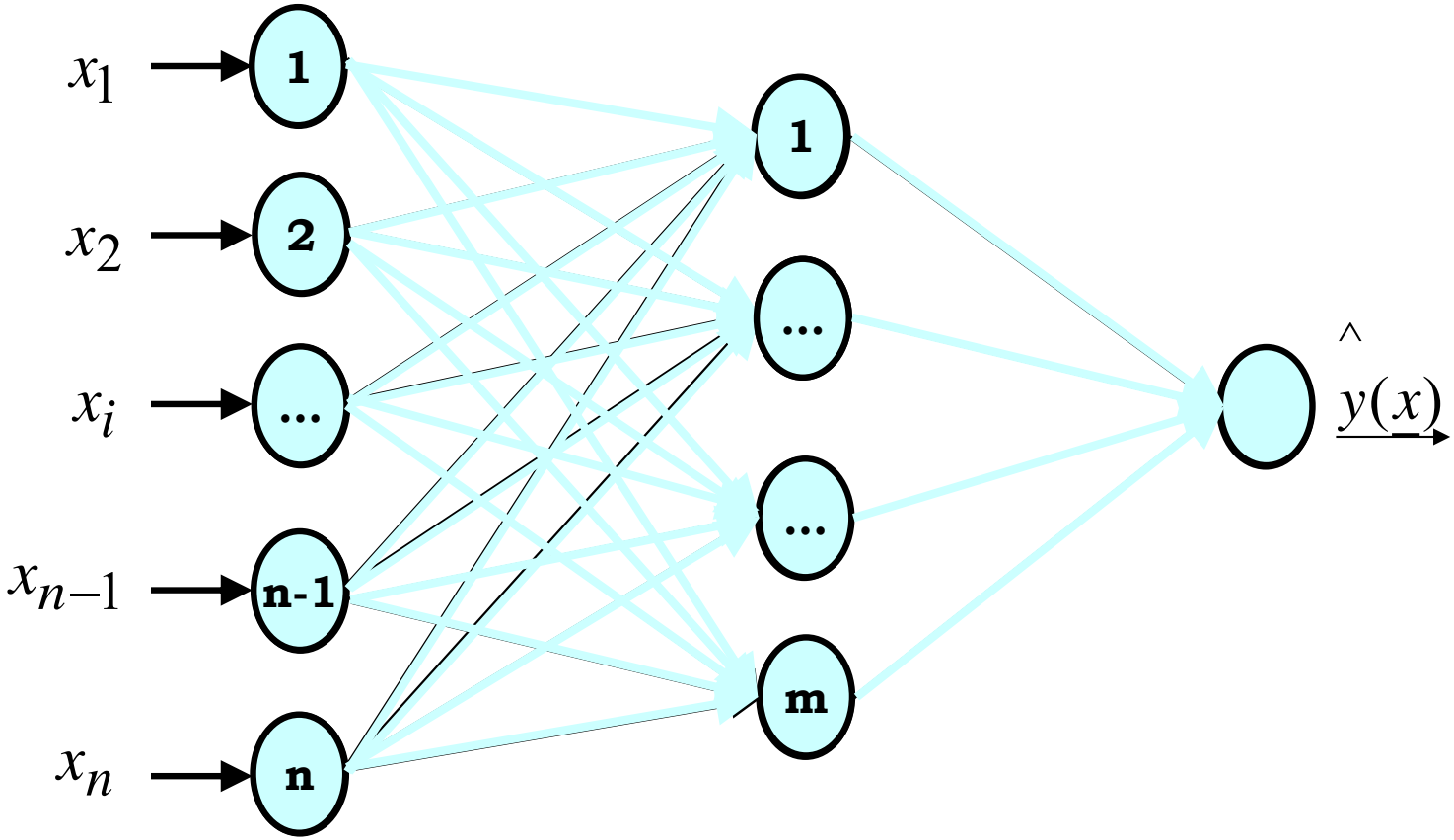
$$\hat{y}(\underline{x}) = f^{(2)}\left(\sum_{j=1}^{m} w_{j1}^{(2)} f_j^{(1)}\left(\sum_{i=1}^{n} w_{ij}^{(1)} x_i - w_{0j}^{(1)}\right) - w_0^{(2)}\right)$$

Generalizzazione della FNN, caratterizzata da più neuroni di output.



Calcolo dell'output di una FNN a fronte di un dato input

$$\hat{y}(\underline{x}) = f^{(2)}\left(\sum_{j=1}^{m} w_{j1}^{(2)} f_j^{(1)}\left(\sum_{i=1}^{n} w_{ij}^{(1)} x_i - w_{0j}^{(1)}\right) - w_0^{(2)}\right)$$



L'apprendimento consiste nella selezione dei valori ottimali dei parametri liberi della rete.

I parametri liberi della rete sono i pesi delle connessioni tra neuroni e le relative soglie.

Se l'architettura della rete è assunta fissata e nota ovvero se il numero di strati nascosti è fissato così come il numero di neuroni intermedi per ogni strato, allora l'apprendimento può essere formalizzato tramite il seguente problema dei minimi quadrati non-lineare

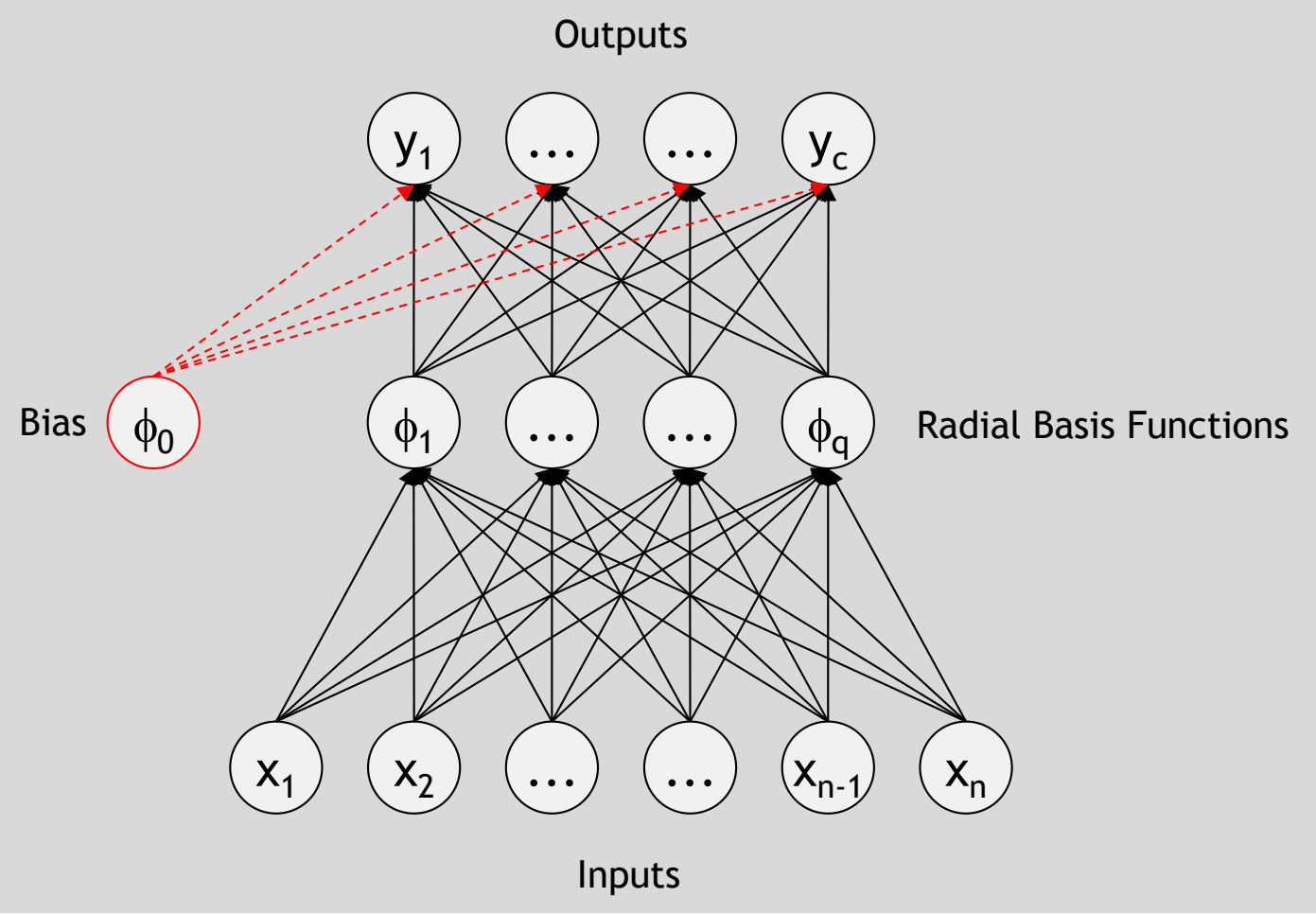
$$\min_{\underline{w}} E(\underline{w}) = \min_{\underline{w}} \sum_{s=1}^{N} \left(y^{(s)} - \hat{y}(\underline{x}^{(s)}) \right)^2$$

Esistono diverse alternative per l'apprendimento, tra le quali ricordiamo

- *Apprendimento Bayesiano*
- *Weight decay*
- *Regolarizzazione*

Radial Basis Function Network

Modello locale caratterizzato da *due parametri* (media, dispersione).



Modelli di Separazione: reti neurali

Le RBFN utilizzano *funzioni radiali*

$$\phi\left(\left\|\underline{x}-\underline{\mu}\right\|\right)$$

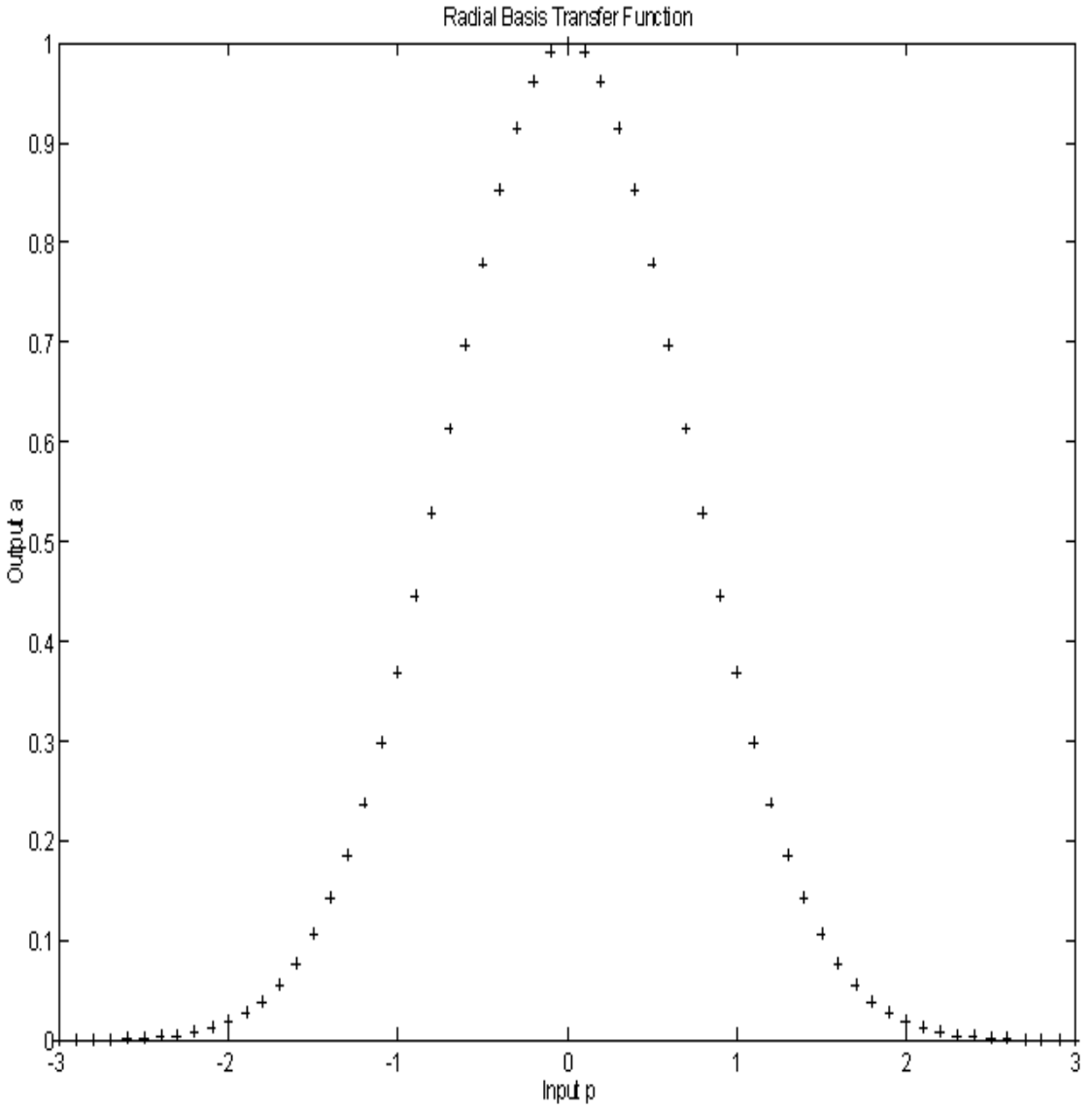
dove

$\left\|\underline{x}-\underline{\mu}\right\|$ distanza tra vettori

$\phi(\cdot)$ Funzione non lineare

Come è fatta una RBF ?

$$\phi\left(\left\|\underline{x}-\underline{\mu}\right\|\right)=exp\left(-\frac{\left\|\underline{x}-\underline{\mu}\right\|^2}{2\sigma^2}\right)$$



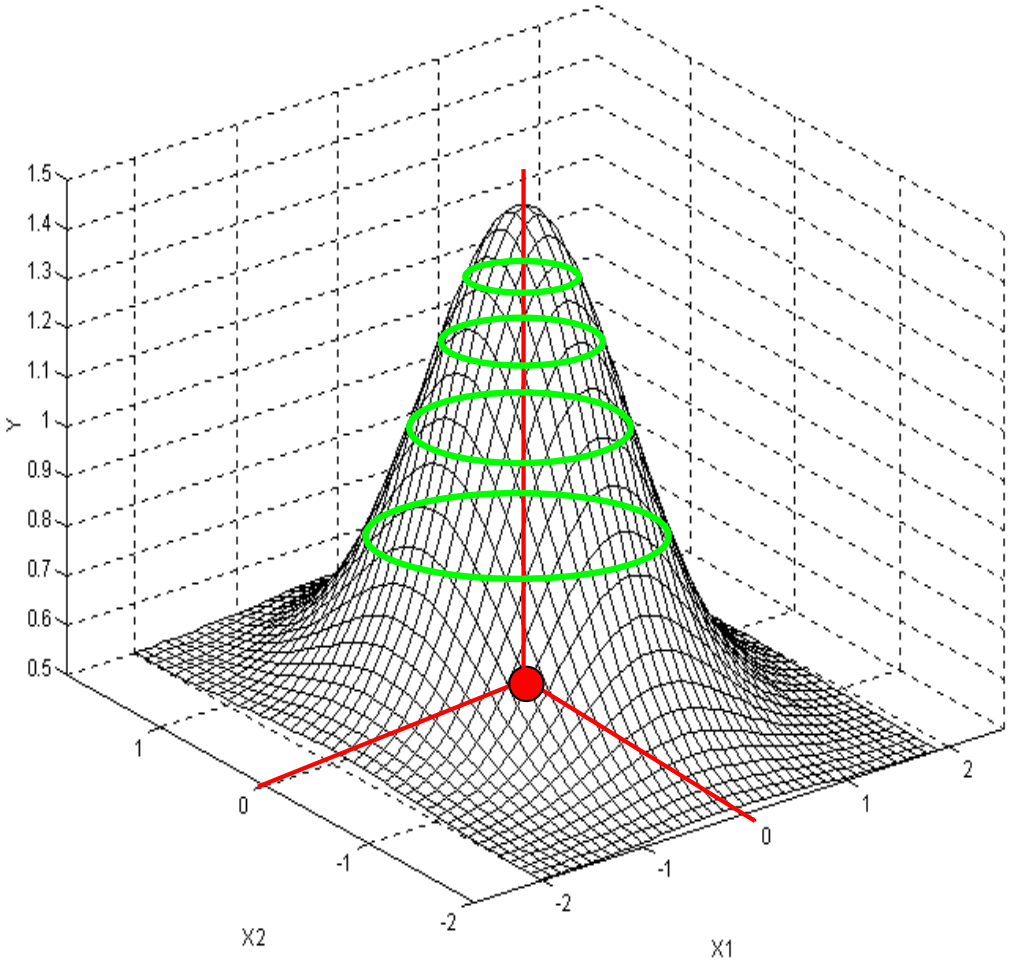
Computa la seguente funzione

$$\hat{y}_k(\underline{x}) = \sum_{j=1}^{q} w_{kj}\phi_j(\underline{x}) + w_{k0}$$

In particolare nel caso in cui si adottino

GAUSSIAN BASIS FUNCTION avremo

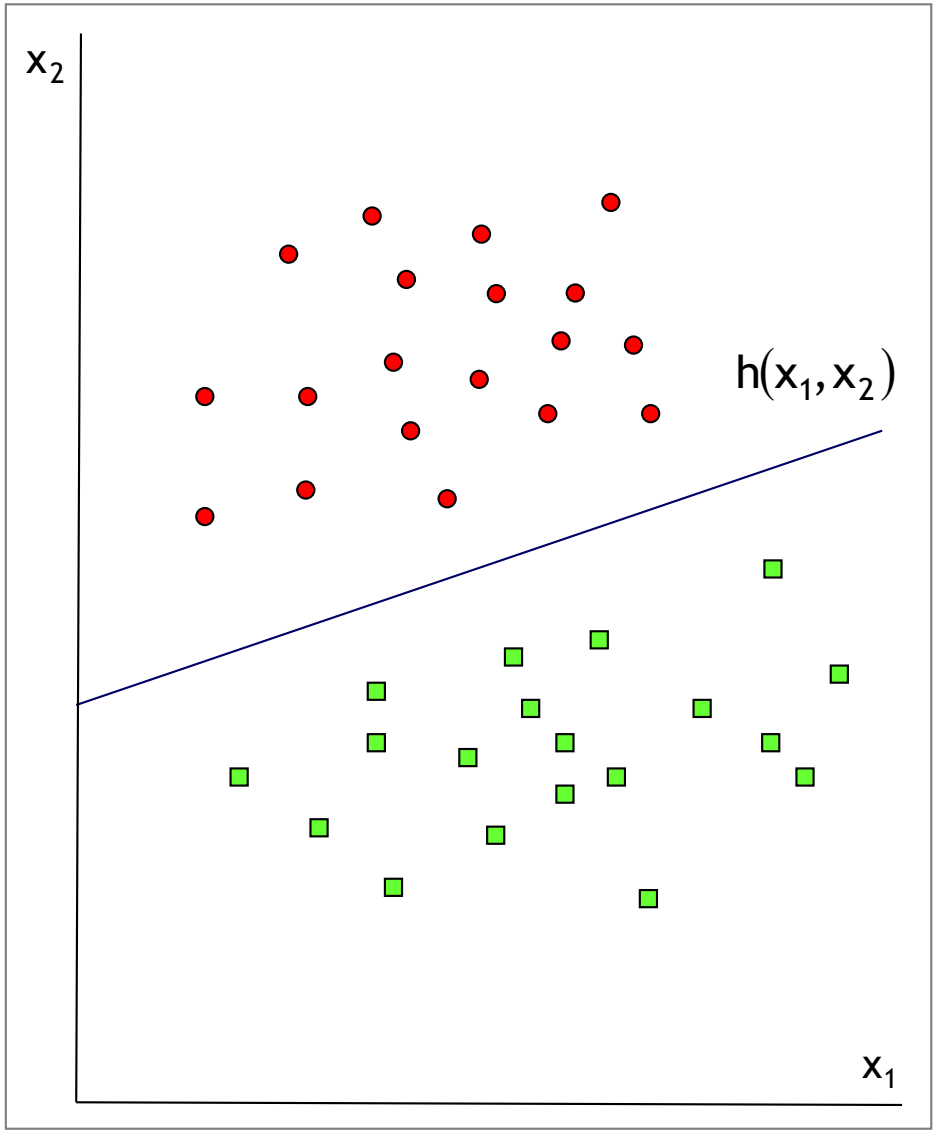
$$\phi_j(\underline{x}) = \exp\left(-\frac{\|\underline{x} - \underline{\mu}_j\|^2}{2\sigma_j^2}\right)$$



dove

- $\underline{\mu}_j$, centro della *RBF j-ma*
- σ^2_j, smoothing factor della *RBF j-ma*

Modelli di Separazione: SVM



Apprendono funzioni lineari con soglia

$$h(\underline{x}) = \text{sign}\{\underline{w} \cdot \underline{x} + b\} = \begin{cases} +1 & \text{se } \underline{w} \cdot \underline{x} + b \geq 0 \\ -1 & \text{atrimenti} \end{cases}$$

La funzione h(•) ha come argomento un *iperpiano* nello spazio delle variabili esplicative. Ogni istanza viene classificata in base alla parte dell'iperpiano nella quale si trova.

Modelli di Separazione: SVM

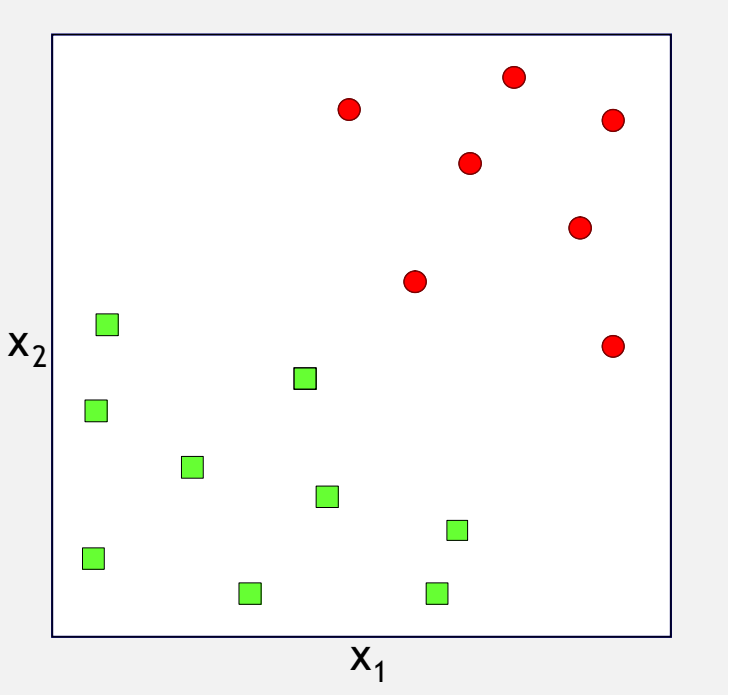
Dato un insieme di "m" dati di training in R^2

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

dove appunto si abbia

$$\underline{x}_i \in R^2$$

$$y_i \in \{+1, -1\}$$



È possibile separare i cerchi rossi (**y=+1**) dai quadrati verdi (**y=-1**) ?

Modelli di Separazione: SVM

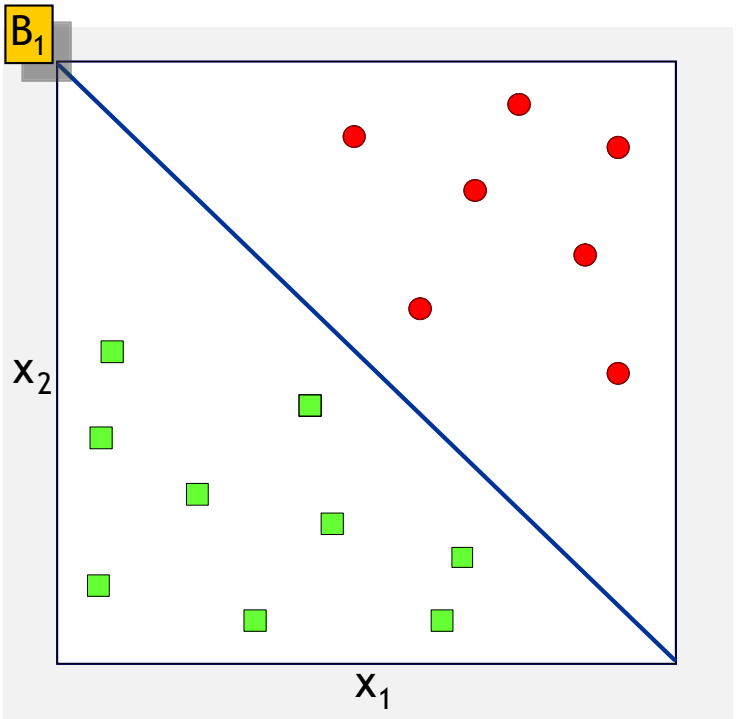
Dato un insieme di "m" dati di training in R^2

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

dove appunto si abbia

$$\underline{x}_i \in R^2$$

$$y_i \in \{+1, -1\}$$



Una risposta è offerta dalla retta riportata in figura, avente equazione

$$\underline{w} \cdot \underline{x} + b = w_1 x_1 + w_2 x_2 + b = 0$$

La retta è completamente determinata tramite i parametri

$$\underline{w} = [w_1, w_2] \qquad b$$

È possibile separare i cerchi rossi (**y=+1**) dai quadrati verdi (**y=-1**) ?

Modelli di Separazione: SVM

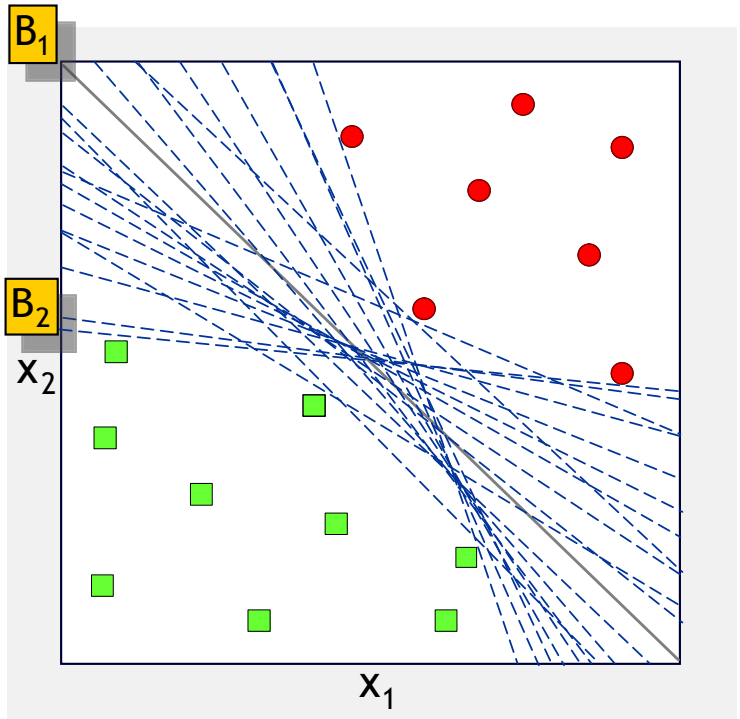
Dato un insieme di "m" dati di training in R^2

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

dove appunto si abbia

$$\underline{x}_i \in R^2$$

$$y_i \in \{+1, -1\}$$



Esistono molteplici soluzioni

- Tutte le rette separatrici sono egualmente nobili ?
- Quale retta separatrice scelgo per classificare ?
- Quale tra le rette separatrici è la "migliore" ?

È possibile separare i cerchi rossi (**y=+1**) dai quadrati verdi (**y=-1**) ?

Modelli di Separazione: SVM

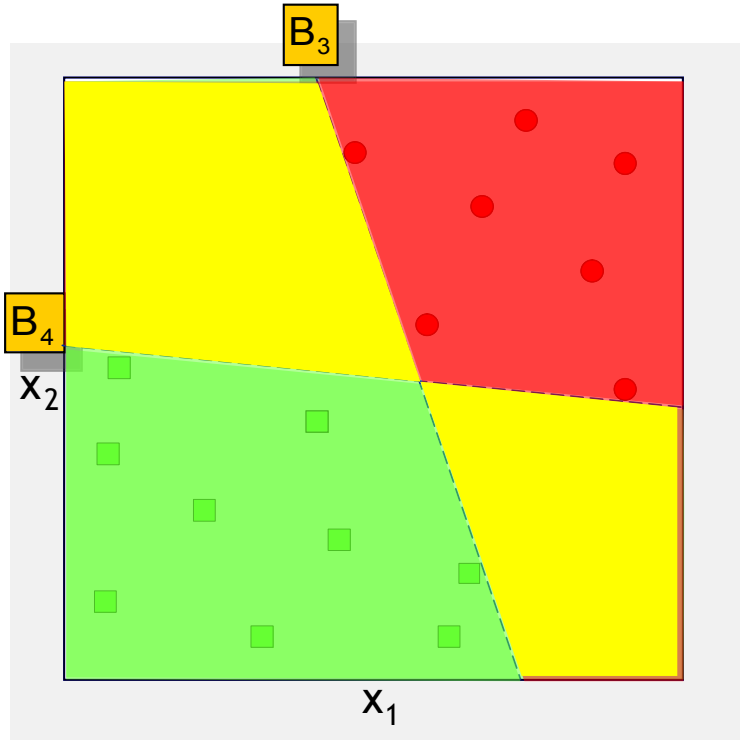
Dato un insieme di "m" dati di training in R^2

$$D = \{(\underline{x}_1, y_1), \ldots, (\underline{x}_m, y_m)\}$$

dove appunto si abbia

$$\underline{x}_i \in R^2$$

$$y_i \in \{+1, -1\}$$

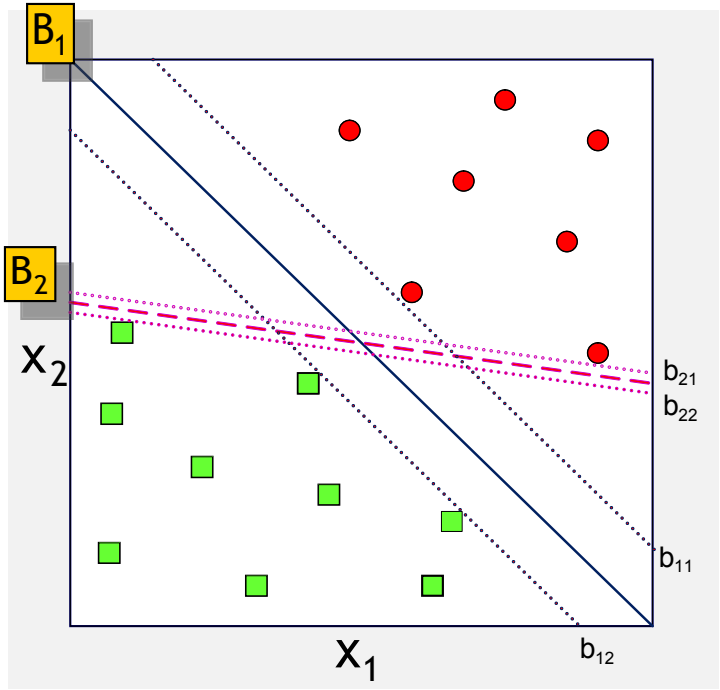


Esistono molteplici soluzioni

- Tutte le rette separatrici sono egualmente nobili ?
- Quale retta separatrice scelgo per classificare ?
- Quale tra le rette separatrici è la "migliore" ?

È possibile separare i cerchi rossi (**y=+1**) dai quadrati verdi (**y=-1**) ?

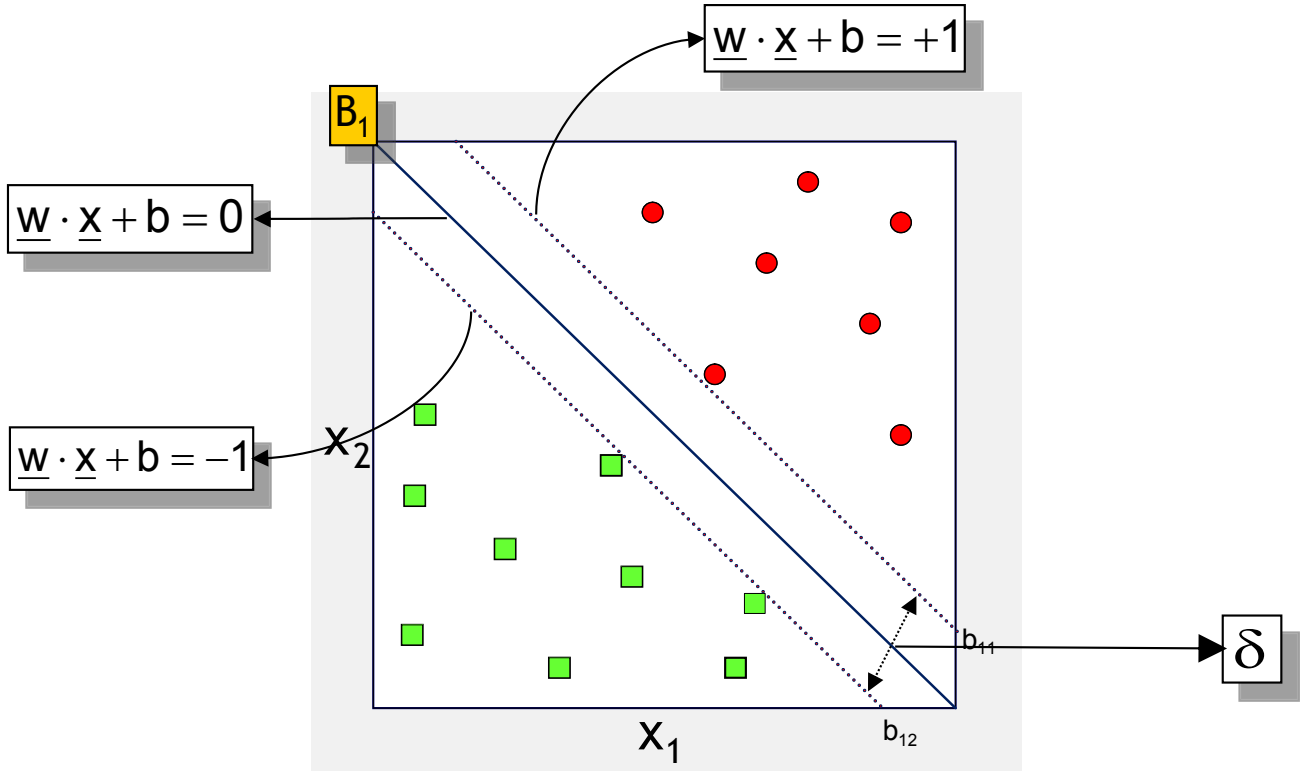
Come è possibile scegliere in termini univoci la retta separatrice ottimale ?



La *retta separatrice ottimale massimizza* il *margine*!!!

B_1 è preferibile a B_2

Caratterizziamo la retta separatrice ottimale.



La funzione calcolata dalla SVM è la seguente

$$h(\underline{x}) = \begin{cases} +1 & \text{se } \underline{w} \cdot \underline{x} + b \geq 0 \\ -1 & \text{altrimenti} \end{cases}$$

Il *margine* è

$$\delta$$

In definitiva, quello che desideriamo è *determinare* la *retta separatrice* che *rende massimo* il *margine soggetto* al *vincolo* che

- i cerchi rossi (y=+1, casi positivi) siano da una parte della retta
- i quadrati verdi (y=-1, casi negativi) siano dall'altra parte della retta.

Linear Hard-Margin SVM

SVM che determina l'iperpiano separatore ottimale, massimizzando il margine

$$\delta$$

e rispettando vincoli del tipo

$$\underline{w} \cdot \underline{x}_i + b \geq +1 \quad se\ y_i = +1$$

$$\underline{w} \cdot \underline{x}_i + b \leq -1 \quad se\ y_i = -1$$

ognuno dei quali garantisce che il rispettivo caso "i" sia classificato correttamente.

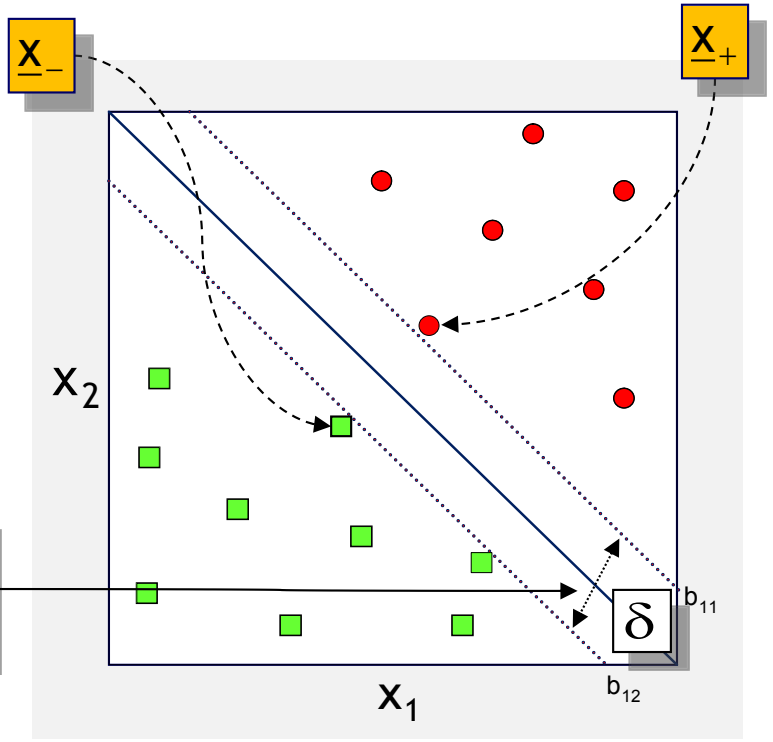
Imponiamo che la *funzione*, implementata dall'iperpiano, *restituisca*, *in corrispondenza* dei *punti ad esso più vicini*, i valori "**+1**" e "**-1**" per le *due regioni* individuate dall'*iperpiano*.

Questa condizione consente di rimuovere il grado di libertà associato al *fattore di scala* per

$$\underline{w} \cdot \underline{x} + b$$

Indichiamo ora tramite $\underline{x}_+$ e $\underline{x}_-$ rispettivamente il *caso positivo* e *negativo più prossimi all'iperpiano* considerato, ovvero avremo

$$\underline{w} \cdot \underline{x}_+ + b = +1 \qquad \underline{w} \cdot \underline{x}_- + b = -1$$



Il ***margine*** viene allora ricavato come segue

$$\left(\left(\frac{\underline{w}}{\|\underline{w}\|} \cdot \underline{x}_+ + \frac{b}{\|\underline{w}\|}\right) - \left(\frac{\underline{w}}{\|\underline{w}\|} \cdot \underline{x}_- - \frac{b}{\|\underline{w}\|}\right)\right) = \frac{1}{\|\underline{w}\|}\left((\underline{w} \cdot \underline{x}_+ + b) - (\underline{w} \cdot \underline{x}_- - b)\right) = \frac{2}{\|\underline{w}\|}$$

I *vincoli* che garantiscono che tutti i casi vengano classificati correttamente sono

$$\begin{cases} \underline{w} \cdot \underline{x}_i + b \geq +1 & se\ y_i = +1 \\ \underline{w} \cdot \underline{x}_i + b \leq -1 & se\ y_i = -1 \end{cases} \quad \Leftrightarrow \quad y_i(\underline{w} \cdot \underline{x}_i + b) - 1 \geq 0 \quad \forall i = 1, \ldots, m$$

In definitiva, il *training consiste* nel risolvere il seguente *problema di programmazione matematica*

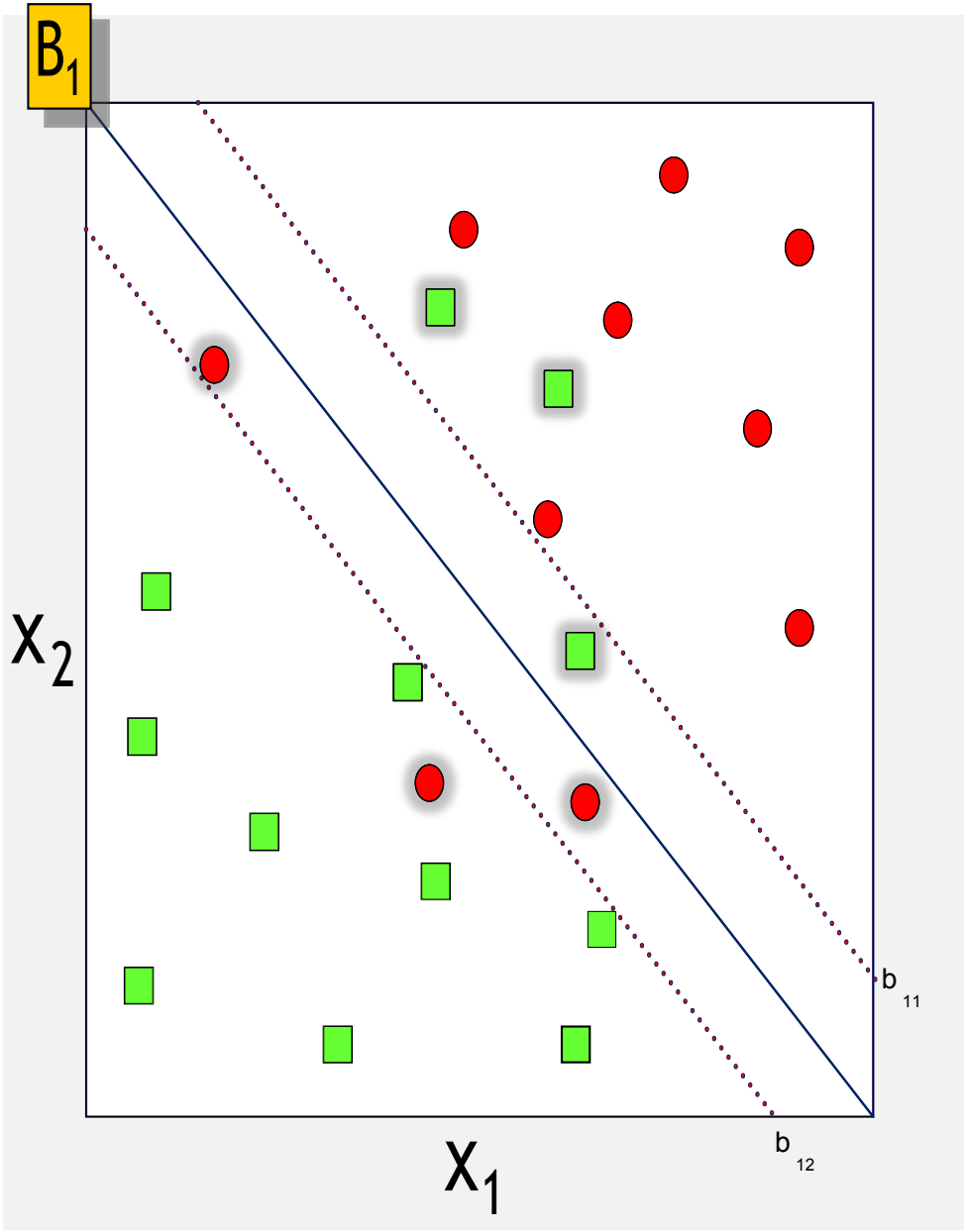
$$\min_{\underline{w},b} \frac{1}{2} \underline{w} \cdot \underline{w}^T$$

soggetto ai vincoli

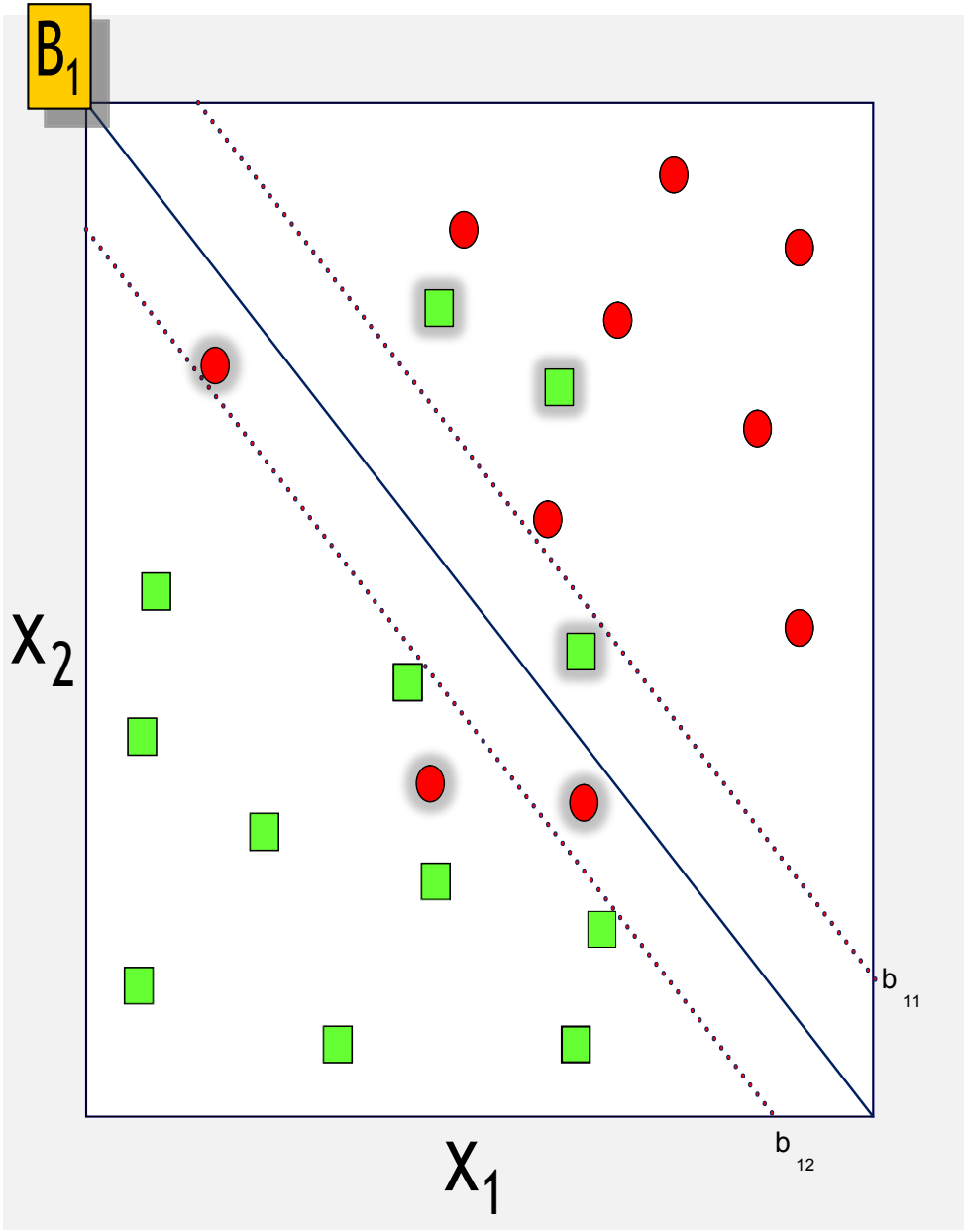
$$y_i(\underline{w} \cdot \underline{x}_i + b) \geq 1 \qquad \forall i = 1, \ldots, m$$

Il problema in oggetto è un problema di *programmazione quadratica* con *vincoli lineari* che può essere risolto tramite opportune procedure numeriche.

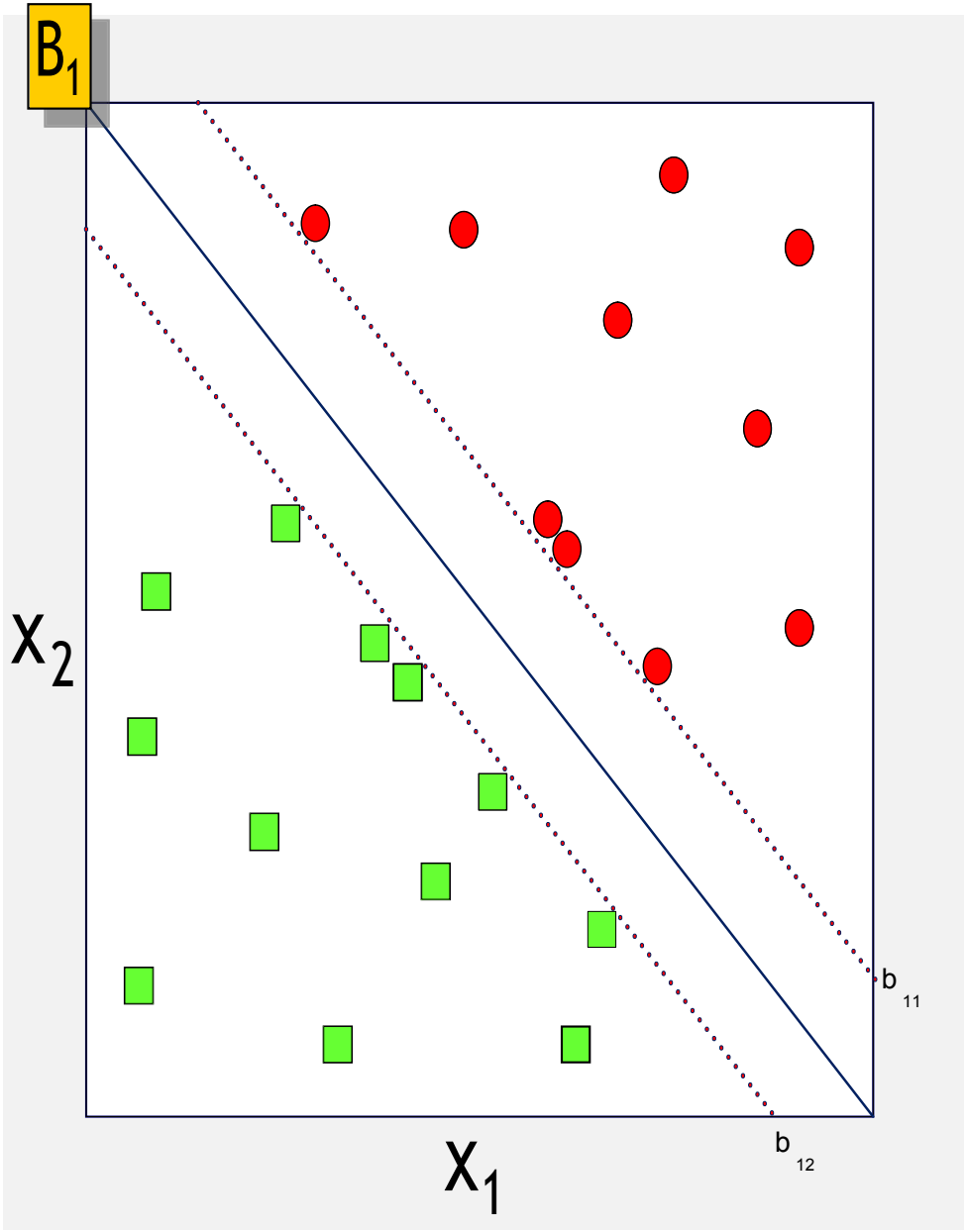
Modelli di Separazione: SVM soft margin



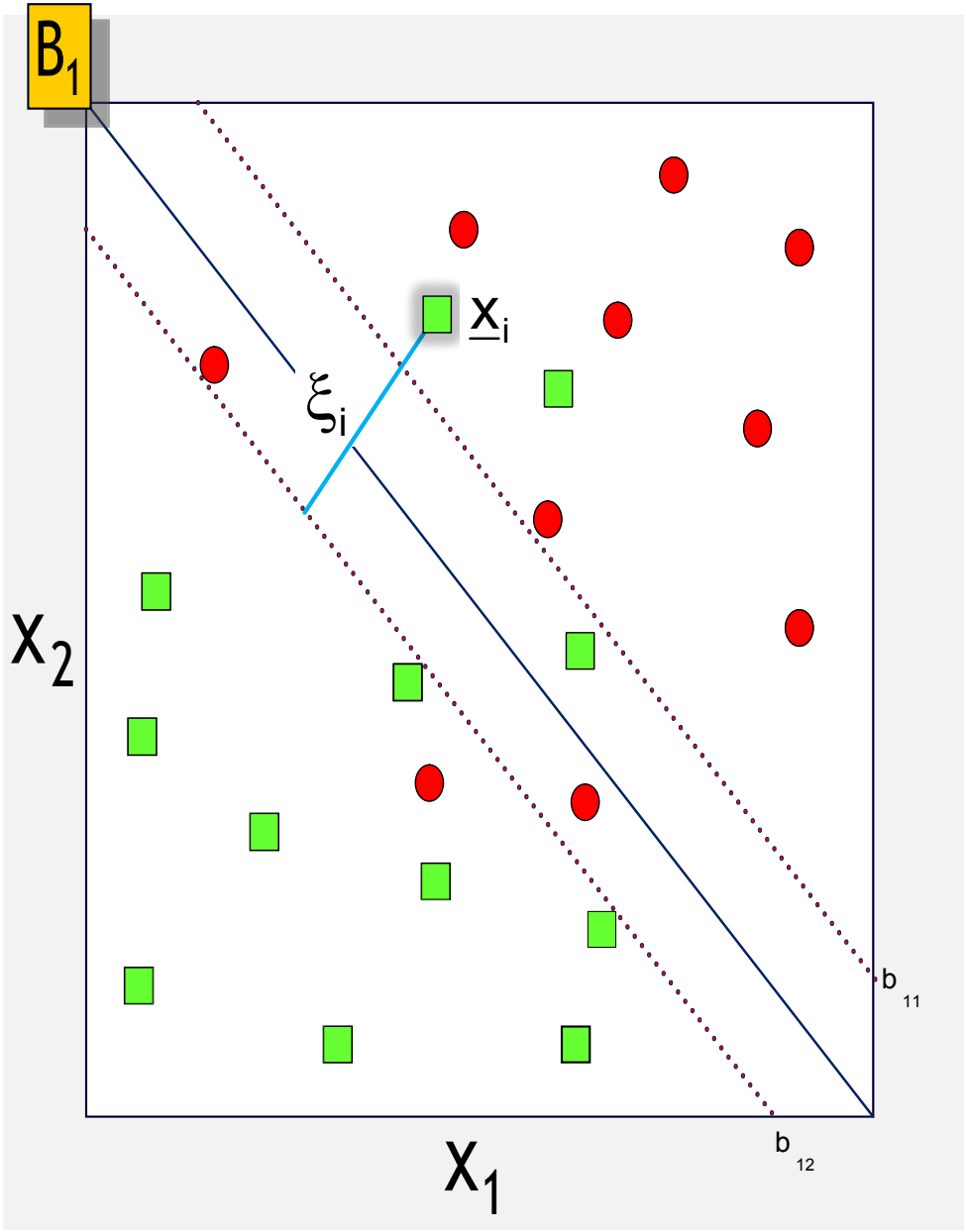
Modelli di Separazione: SVM soft margin



Modelli di Separazione: SVM soft margin



Modelli di Separazione: SVM soft margin



Linear Soft Margin SVM

Se l'insieme dei dati di *training non* è *linearmente separabile* allora il *problema di ottimizzazione* introdotto in precedenza non ammette soluzione pertanto è necessario formulare un *nuovo problema di ottimizzazione* basato sulle *Linear Soft Margin SVM*.

In particolare, viene introdotto il concetto di *limite superiore* al *numero di errori* commessi dalla SVM. Successivamente viene *minimizzato* il *limite superiore* congiuntamente alla *lunghezza* del *vettore dei pesi* della SVM.

$$\min_{\underline{w},b} \frac{1}{2}\underline{w}\cdot\underline{w}^T$$

soggetto ai vincoli

$$y_i\left(\underline{w}\cdot\underline{x}_i + b\right) \geq 1 \qquad \forall i = 1,\ldots,m$$

$$\min_{\underline{w},b,\underline{\xi}} \frac{1}{2}\underline{w}\cdot\underline{w}^T + \Delta\sum_{i=1}^{m}\xi_i$$

soggetto ai vincoli

$$\forall_{i=1}^{m} : y_i\left(\underline{w}\cdot\underline{x}_i + b\right) \geq 1 - \xi_i$$

$$\forall_{i=1}^{m} : \xi_i \geq 0$$

Non Linear SVM

La complessità del problema di classificazione da risolvere dipende dalla sua rappresentazione. Nel Data Mining, un approccio comune consiste nel cambiare la rappresentazione dei dati:

$$(x_1, x_2, \ldots, x_n) \longrightarrow \phi(\underline{x}) = (\phi_1(\underline{x}), \phi_2(\underline{x}), \ldots, \phi_N(\underline{x}))$$

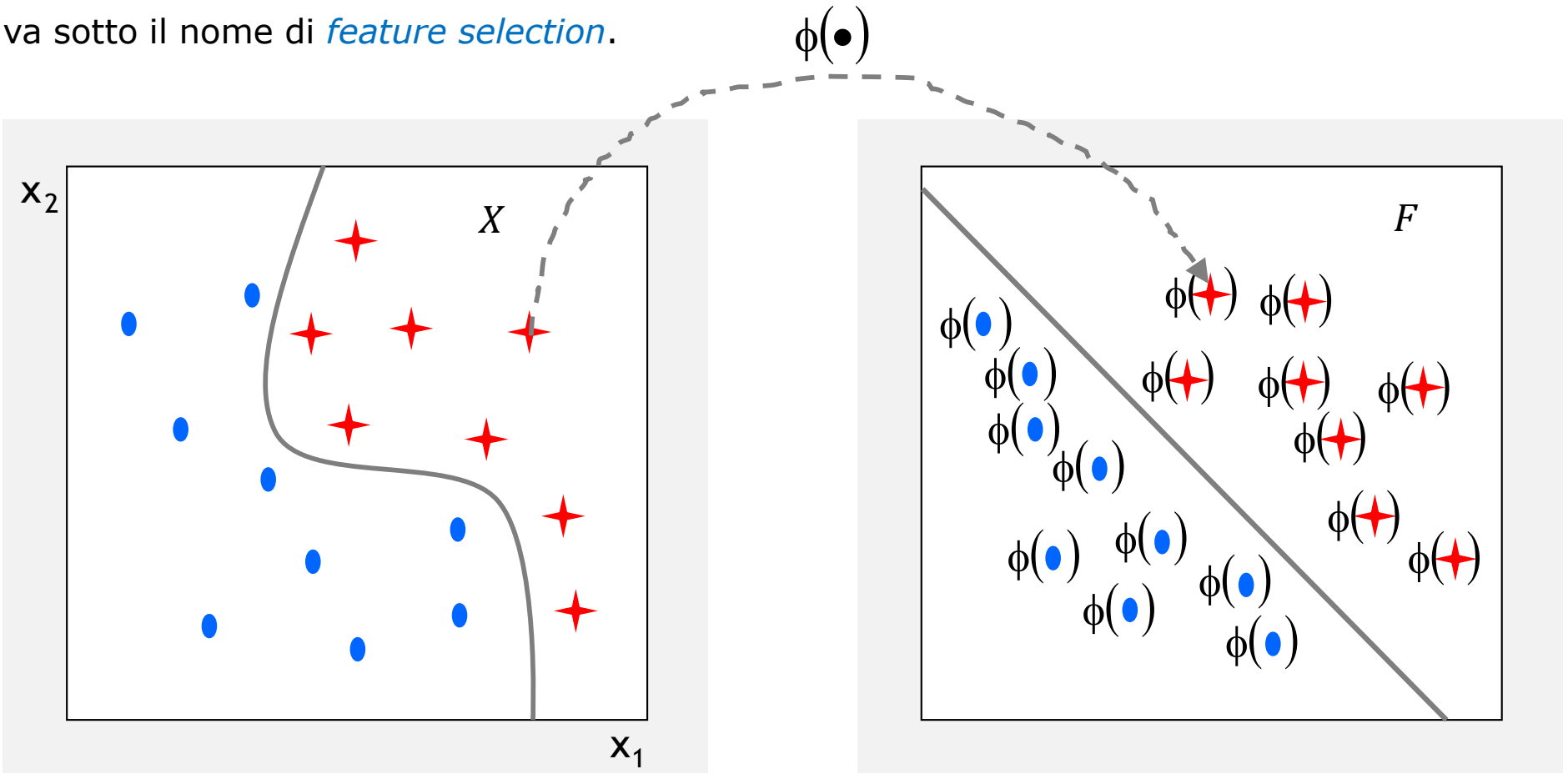
Questo passo equivale a mappare lo spazio di input *X* in un nuovo spazio detto *spazio delle features*

$$F = \{\phi(\underline{x}) \mid \underline{x} \in X\}$$

Il fatto che questa operazione possa semplificare il problema di classificazione è noto da tempo sia nell'ambito del Machine Learning che del Data Mining. Molte tecniche sono state proposte e sviluppate per identificare la migliore rappresentazione dei dati.

Le quantità nello spazio trasformato vengono usualmente riferite con il termine di *features* mentre le quantità nello spazio originale sono riferite con il termine di *attributi*.

Il problema della scelta della miglior rappresentazione dei dati, lo abbiamo già presentato, va sotto il nome di *feature selection*.



Sono disponibili diversi approcci per realizzare la feature selection.

Spesso si cerca di identificare il minimo numero di features in grado di garantire l'informazione essenziale contenuta negli attributi originali.

Questo approccio prende il nome di *riduzione della dimensionalità*

$$(x_1, x_2, \ldots, x_n) \longrightarrow \phi(\underline{x}) = (\phi_1(\underline{x}), \phi_2(\underline{x}), \ldots, \phi_d(\underline{x})) \qquad d < n$$

Porta i seguenti vantaggi:

- *riduzione del costo computazionale*
- *riduzione del degrado di performance dovuto al numero elevato di attributi (curse of dimensionality)*

Un altro task di feature selection è quello finalizzato a identificare e rimuovere gli attributi irrilevanti.

Esistono casi in cui uno spazio delle features con dimensione maggiore di quella dello spazio originale porta benefici.

$$(x_1, x_2) \longrightarrow \phi(x_1, x_2) = (x_1^2, x_2^2, x_1 x_2)$$

I modelli *Non-Linear SVM* affrontano il *problema di classificazione* in *due fasi*:

- *Proiezione*; induce un mapping tra lo spazio originale *X* ed un nuovo spazio delle features *F* tipicamente di dimensionalità differente.
- *Classificazione*; agisce sul nuovo spazio delle features *F* e applica regole di classificazione lineare.

Pertanto la complessità del modello SVM utilizzato è ancora paragonabile alla complessità delle SVM lineari (Hard e Soft) ma la capacità espressiva viene di molto aumentata.

Apprendere relazioni non lineari tramite una SVM lineare richiede di selezionare un insieme di features e di esprimere i dati nello spazio delle features.

Questa operazione equivale all'applicazione di un mapping non lineare ai dati nello spazio originale per ottenere dati nello spazio delle features che verranno trattati con SVM lineari.

Le funzioni che considereremo saranno del tipo:

$$f(\underline{x}) = \sum_{i=1}^{N} w_i \phi_i(\underline{x}) + b$$

dove

$$\phi : X \rightarrow F$$

è un mapping non lineare dallo spazio degli attributi originali allo spazio delle features.

Significa che costruiamo macchine non lineari in due fasi, come già segnalato in precedenza.

Abbiamo accennato come il problema di apprendimento in SVM lineari sia risolto nello spazio duale. Questo significa che le funzioni che trattiamo sono esprimibili come combinazioni lineari dei dati di training, in modo tale che la classificazione sia basata sulla valutazione del prodotto interno tra i dati di training ed il dato di test

$$f(\underline{x}) = \sum_{i=1}^{l} \alpha_i y_i \langle \phi(\underline{x}_i), \phi(\underline{x}) \rangle + b$$

l = numero vettori di supporto.

Formalmente "i" assume valori da 1 ad "m", numero di osservazioni, ma alpha è nullo per le osservazioni che non siano vettori di supporto.

Se disponessimo di un modo per computare il prodotto interno

$$\langle \phi(\underline{x}_i), \phi(\underline{x}) \rangle$$

direttamente nello spazio delle features, come funzione dei dati di training nello spazio originale, diverrebbe possibile fondere le due fasi realizzando una macchina di apprendimento non lineare.

Un tale metodo di computazione diretta lo chiameremo funzione kernel.

Una *funzione kernel K* è una funzione tale che per ogni coppia di punti **x**, **z** $\in$ *X* valga la seguente uguaglianza

$$K(\underline{x},\underline{z}) = \langle \phi(\underline{x}) \cdot \phi(\underline{z}) \rangle$$

dove, ϕ è un mapping dallo spazio degli attributi originale *X* verso lo spazio delle features *F*.

La disponibilità di una funzione kernel consente di riscrivere la funzione che implementa la classificazione come segue

$$f(\underline{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\underline{x}_i, \underline{x}) + b$$

Non siamo obbligati a conoscere il mapping ϕ per apprendere nello spazio delle features *F*.

Kernels

Esiste uno studio molto intenso sulle "*funzioni kernel*", un esempio di "*funzione kernel*" particolarmente importante ed utilizzata viene mostrato di seguito

$$K(\underline{x}_i, \underline{x}_j) = \exp\left\{-\frac{\|\underline{x}_i - \underline{x}_j\|^2}{2\sigma^2}\right\}$$

Nuovi kernels possono essere resi "valid kernels" tramite alcune operazioni (addizione, moltiplicazione e rescaling) applicate a kernels di partenza, portando ad ottenere "proper kernels" a condizione che la corrispondente "Gram matrix" sia "positiva definita".

$$K(x_1, x_2) = K_1(x_1, x_2) + K_2(x_1, x_2)$$
$$K(x_1, x_2) = \lambda K_1(x_1, x_2)$$
$$K(x_1, x_2) = K_1(x_1, x_2) K_2(x_1, x_2)$$

Data una funzione a valori reali f(**x**) con input **x**, il seguente è un "*valid kernel*"

$$K(\underline{x}_1, \underline{x}_2) = f(\underline{x}_1) f(\underline{x}_2)$$

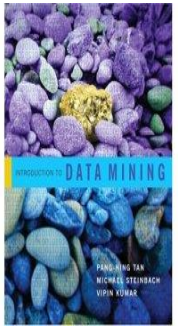
CLASSIFICAZIONE

MODELLI EURISTICI

Classificazione

Parte dei contenuti della presente lezione sono tratti dai testi elencati di seguito.

Carlo Vercellis (2006). *Business intelligence: modelli matematici e sistemi per le decisioni*, McGraw-Hill.



Pang-Ning Tan, Michael Steinbach and Vipin Kumar (2006). *Introduction to Data Mining*, Pearson International.

Modelli Euristici

Utilizzano procedure di classificazione basate su schemi algoritmici elementari ed intuitivi.

A questa categoria appartengono:

- ***Nearest Neighbor***, basati sulla nozione di distanza tra le osservazioni
- ***Alberi di classificazione***, adottano schemi divide and conquer per indurre raggruppamenti di osservazioni quanto più possibile omogenee rispetto alla specifica classe target presa in considerazione
- ***Random Forest***, sfruttano lo schema degli alberi di classificazione per sviluppare modelli efficienti ed efficaci combinando tra loro diverse previsioni

Modelli Euristici: alberi di classificazione

Metodo di apprendimento molto noto ed utilizzato per applicazioni di Data Mining. Gli alberi di classificazione devono la loro popolarità a:

- *semplicità concettuale*
- *facilità di impiego*
- *velocità di elaborazione*
- *robustezza rispetto a dati mancanti ed outlier*
- *interpretabilità delle regole generate.*

Separano osservazioni appartenenti a classi differenti, ricavando rappresentazioni ad albero che offrono regole facilmente interpretabili, mostrano in termini esplicativi le relazioni esistenti tra gli attributi ed il target.

L'albero di classificazione viene costruito tramite la fase di apprendimento, procedura euristica ricorsiva basata sullo schema *divide-and-conquer* (Top Down Induction of Decision Trees (TDIDT)).

Procedura TDIDT

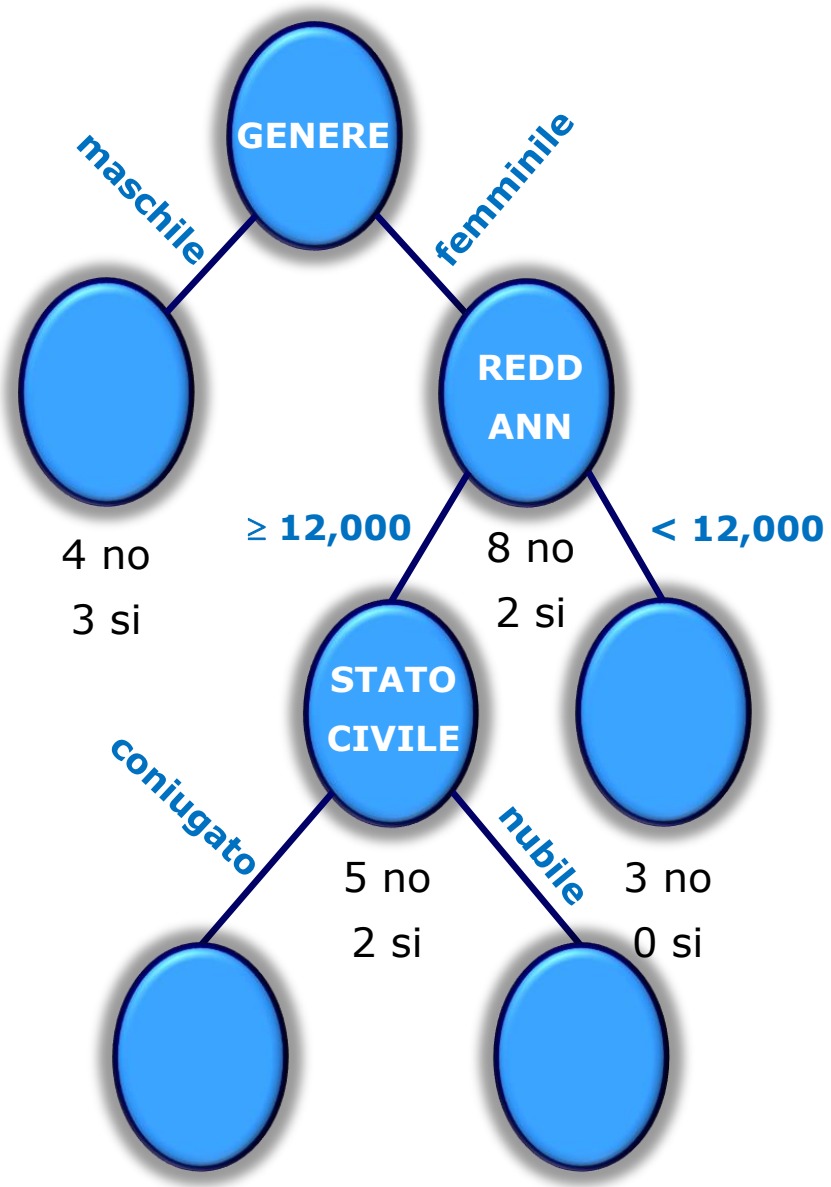
1. Ogni osservazione viene inclusa nel *nodo radice* dell'albero. La radice viene inserita nella lista "*L*" dei *nodi attivi*.

2. Se la lista "*L*" è vuota la procedura si arresta. Altrimenti si seleziona un nodo "*J*" appartenente alla lista "*L*", lo si rimuove dalla lista medesima e lo si utilizza come *nodo di analisi*.

3. Determinare la *regola ottimale di separazione* delle osservazioni presenti in "*J*", sulla base di un opportuno criterio prestabilito. Applicare la regola di separazione generata e costruire i *nodi discendenti* del nodo "*J*" suddividendo le osservazioni presenti in "*J*". Per ogni nodo discendente, si verificano le condizioni per arrestare la suddivisione. Se sono soddisfatte, il nodo "*J*" costituisce una *foglia* cui viene assegnata la *classe target* determinata dalla maggioranza delle osservazioni presenti in "*J*". Altrimenti i nodi discendenti vengono aggiunti alla lista "*L*". Si ripete il **Passo 2**.

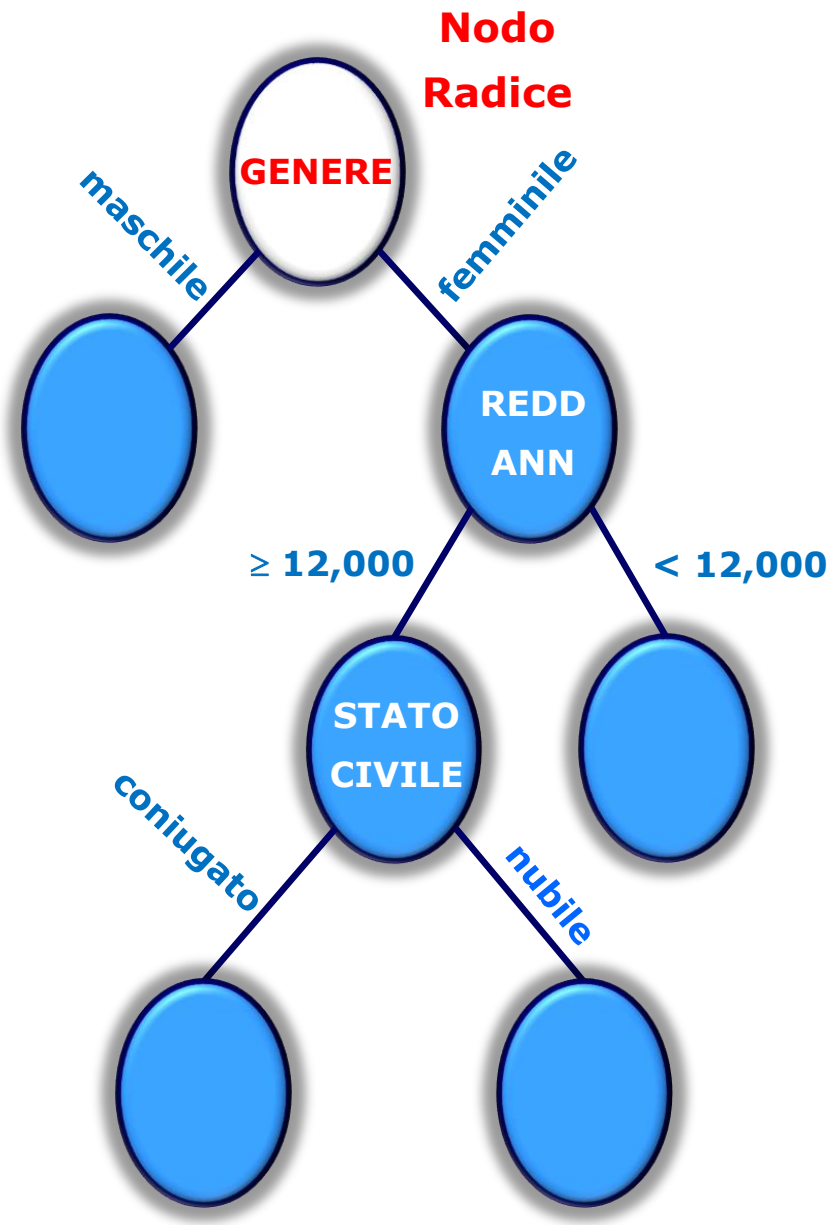
Modelli Euristici: alberi di classificazione

GENERE	ETA	PROVINCIA	REGIONE	REDD ANN	...	...	STATO CIVILE	EVASORE
maschile	32	VA	Lombardia	20,000 €			celibe	no
femminile	45	AQ	Abruzzo	13,500 €			coniugato	si
femminile	21	RM	Lazio	11,600 €			nubile	no
femminile	62	RM	Lazio	15,350 €			nubile	no
maschile	68	RC	Calabria	10,945 €			divorziato	no
maschile	19	AN	Marche	10,233 €			celibe	no
maschile	24	LT	Lazio	10,450 €			coniugato	si
femminile	22	VI	Veneto	11,567 €			coniugato	no
femminile	29	NO	Piemonte	16,350 €			nubile	no
maschile	52	FI	Toscana	11,245 €			nubile	si
femminile	34	MI	Sicilia	13,450 €			coniugato	no
femminile	33	MI	Basilicata	7,500 €			coniugato	no
femminile	55	TN	Trentino	13,450 €			coniugato	si
maschile	39	FR	Lazio	11,590 €			celibe	si
femminile	55	MI	Lombardia	23,500 €			coniugato	no
maschile	27	MI	Lombardia	35,800 €			celibe	no
femminile	31	AQ	Abruzzo	14,750 €			coniugato	no



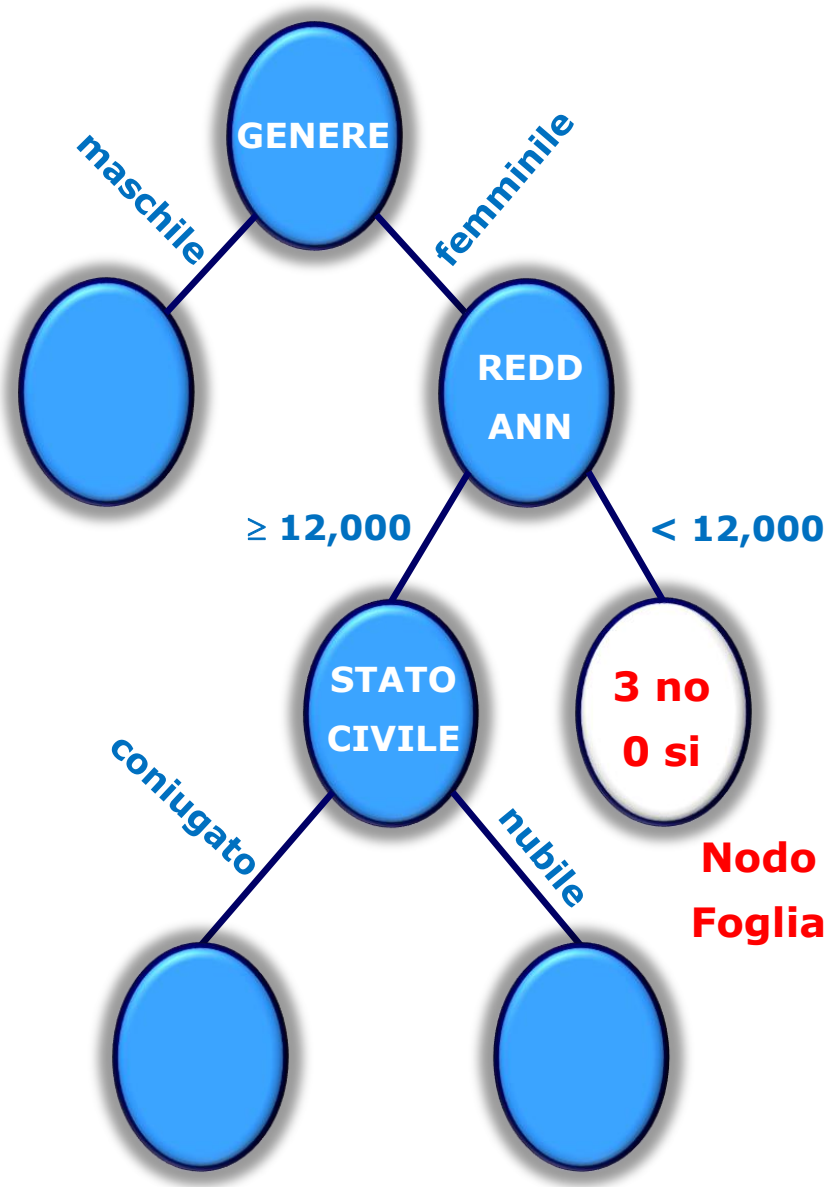
Modelli Euristici: alberi di classificazione

GENERE	ETA	PROVINCIA	REGIONE	REDD ANN	...	...	STATO CIVILE	EVASORE
maschile	32	VA	Lombardia	20,000 €			celibe	no
femminile	45	AQ	Abruzzo	13,500 €			coniugato	si
femminile	21	RM	Lazio	11,600 €			nubile	no
femminile	62	RM	Lazio	15,350 €			nubile	no
maschile	68	RC	Calabria	10,945 €			divorziato	no
maschile	19	AN	Marche	10,233 €			celibe	no
maschile	24	LT	Lazio	10,450 €			coniugato	si
femminile	22	VI	Veneto	11,567 €			coniugato	no
femminile	29	NO	Piemonte	16,350 €			nubile	no
maschile	52	FI	Toscana	11,245 €			nubile	si
femminile	34	MI	Sicilia	13,450 €			coniugato	no
femminile	33	MI	Basilicata	7,500 €			coniugato	no
femminile	55	TN	Trentino	13,450 €			coniugato	si
maschile	39	FR	Lazio	11,590 €			celibe	si
femminile	55	MI	Lombardia	23,500 €			coniugato	no
maschile	27	MI	Lombardia	35,800 €			celibe	no
femminile	31	AQ	Abruzzo	14,750 €			coniugato	no



Modelli Euristici: alberi di classificazione

GENERE	ETA	PROVINCIA	REGIONE	REDD ANN	...	...	STATO CIVILE	EVASORE
maschile	32	VA	Lombardia	20,000 €			celibe	no
femminile	45	AQ	Abruzzo	13,500 €			coniugato	si
femminile	21	RM	Lazio	11,600 €			nubile	no
femminile	62	RM	Lazio	15,350 €			nubile	no
maschile	68	RC	Calabria	10,945 €			divorziato	no
maschile	19	AN	Marche	10,233 €			celibe	no
maschile	24	LT	Lazio	10,450 €			coniugato	si
femminile	22	VI	Veneto	11,567 €			coniugato	no
femminile	29	NO	Piemonte	16,350 €			nubile	no
maschile	52	FI	Toscana	11,245 €			nubile	si
femminile	34	MI	Sicilia	13,450 €			coniugato	no
femminile	33	MI	Basilicata	7,500 €			coniugato	no
femminile	55	TN	Trentino	13,450 €			coniugato	si
maschile	39	FR	Lazio	11,590 €			celibe	si
femminile	55	MI	Lombardia	23,500 €			coniugato	no
maschile	27	MI	Lombardia	35,800 €			celibe	no
femminile	31	AQ	Abruzzo	14.750 €			coniugato	no

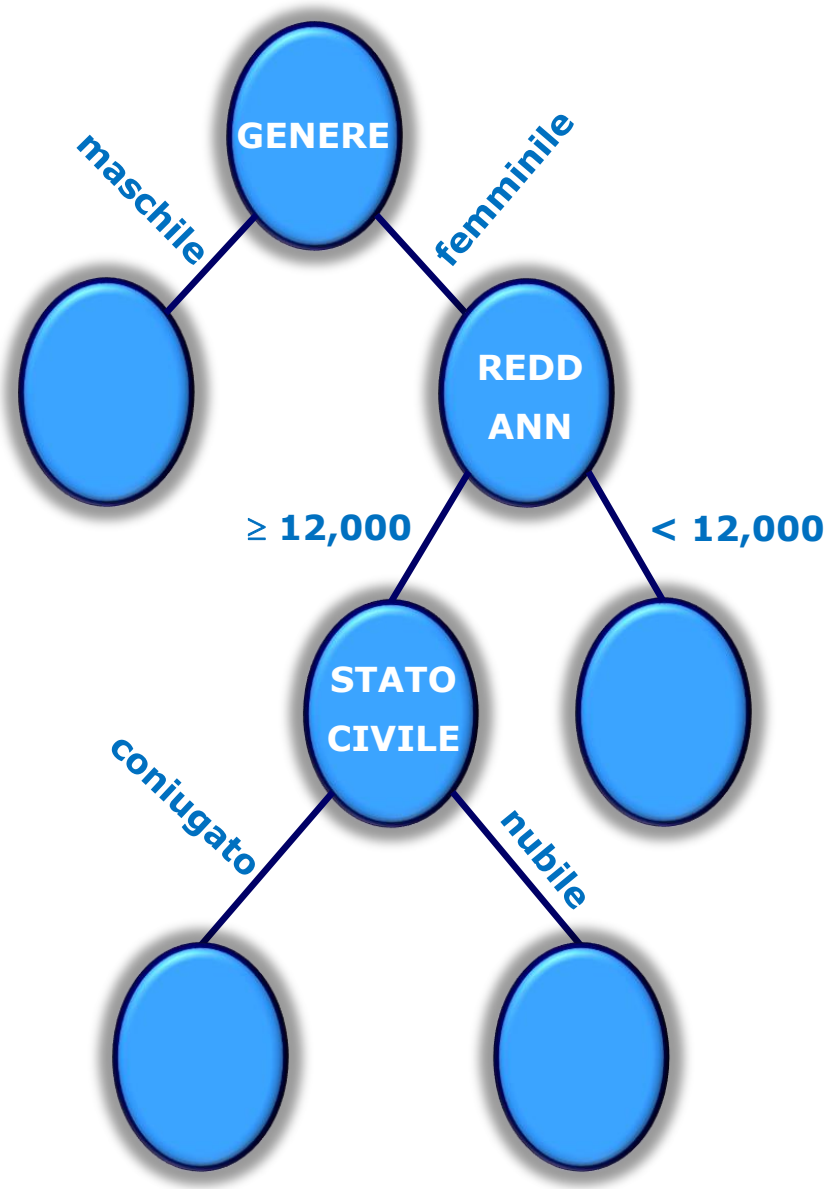


Strategia Greedy (miope)

Il nodo di analisi sottopone le osservazioni che vi appartengono ad uno "*split*" basato su un test che ottimizza un criterio.

Questioni

- *Come effettuare lo "split"?*
  - ✓ *come specificare la condizione di test?*
  - ✓ *come determinare lo "split ottimale"?*
- *Determinare una regola di arresto*



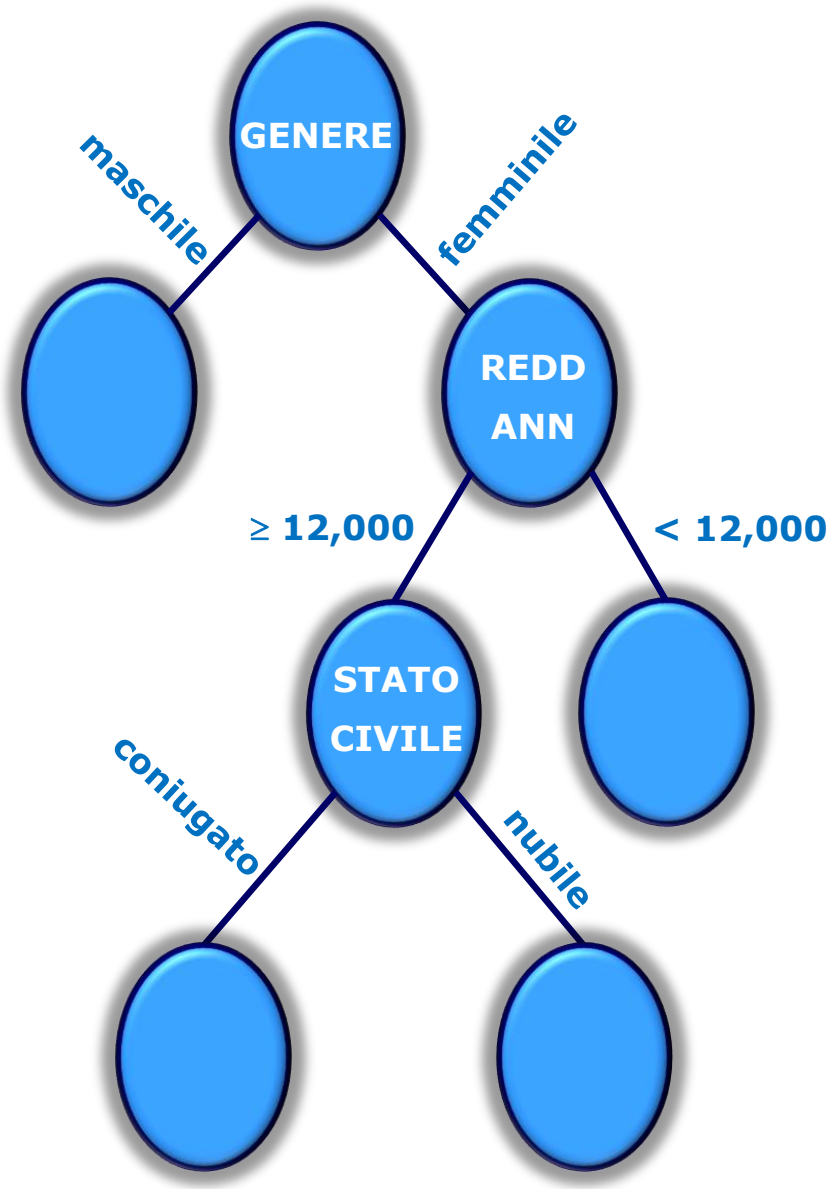
Come specificare la condizione di test?

Dipende dal tipo dell'attributo che deve essere sottoposto allo "*split*"

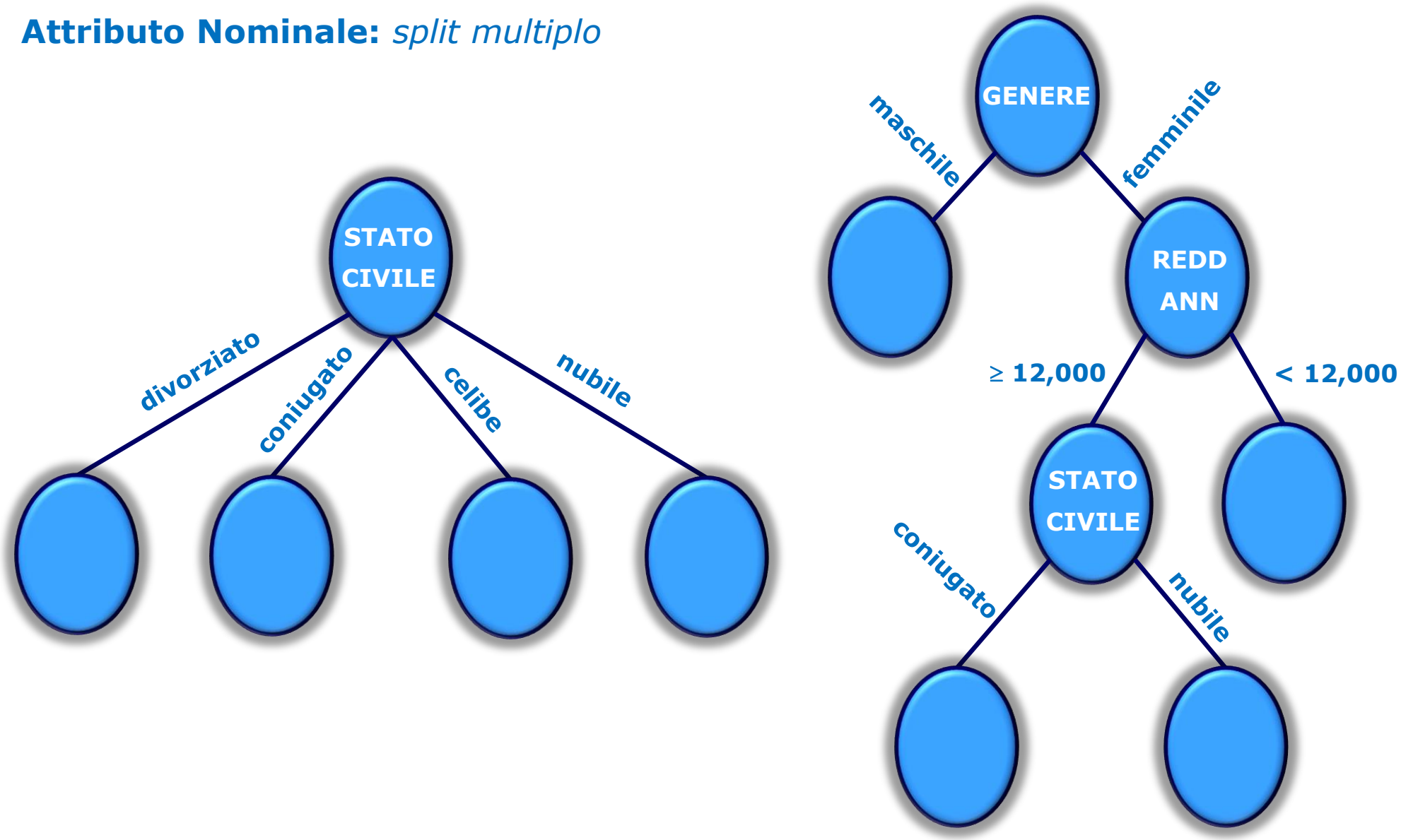
- *nominale*
- *ordinale*
- *continuo*

Dipende dalle possibili modalità dello "*split*"

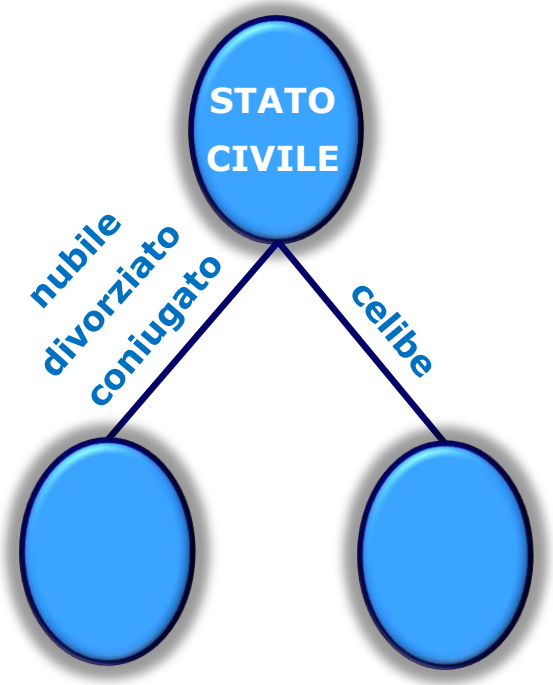
- *binario*
- *multiplo*

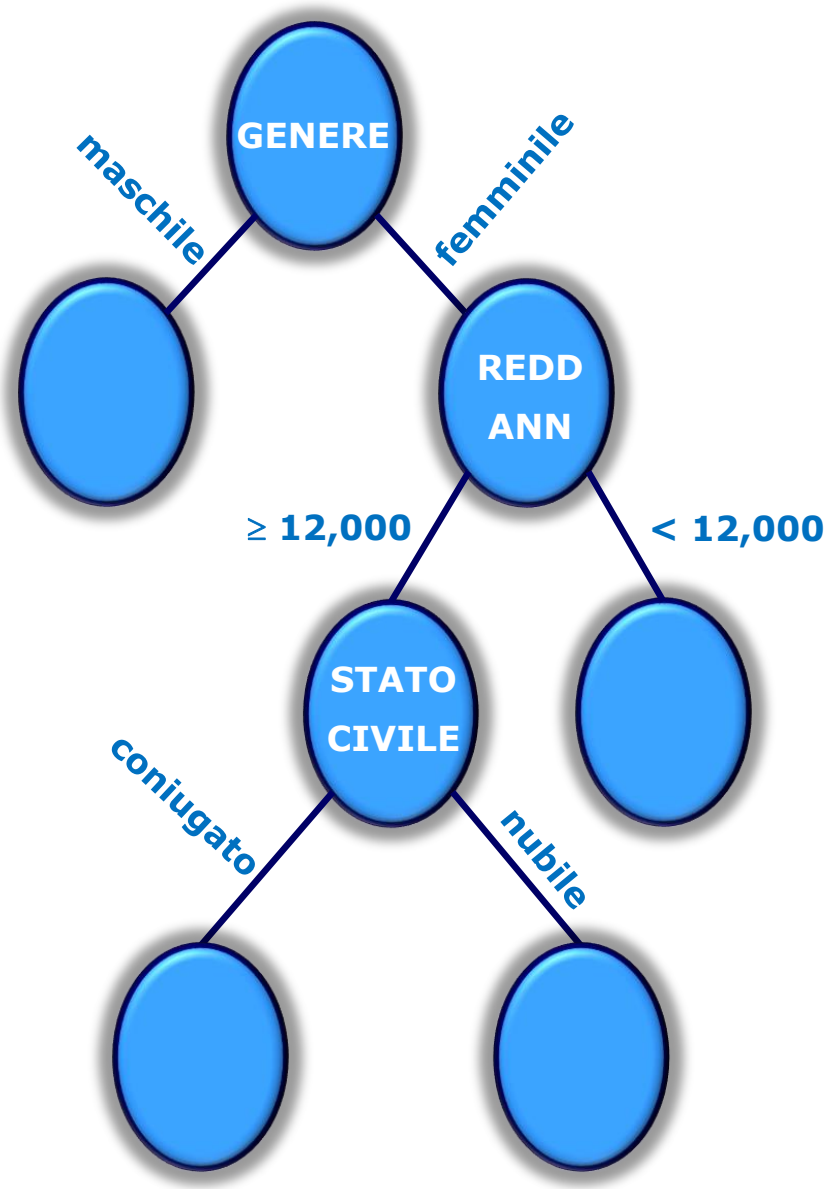


Attributo Nominale: *split multiplo*

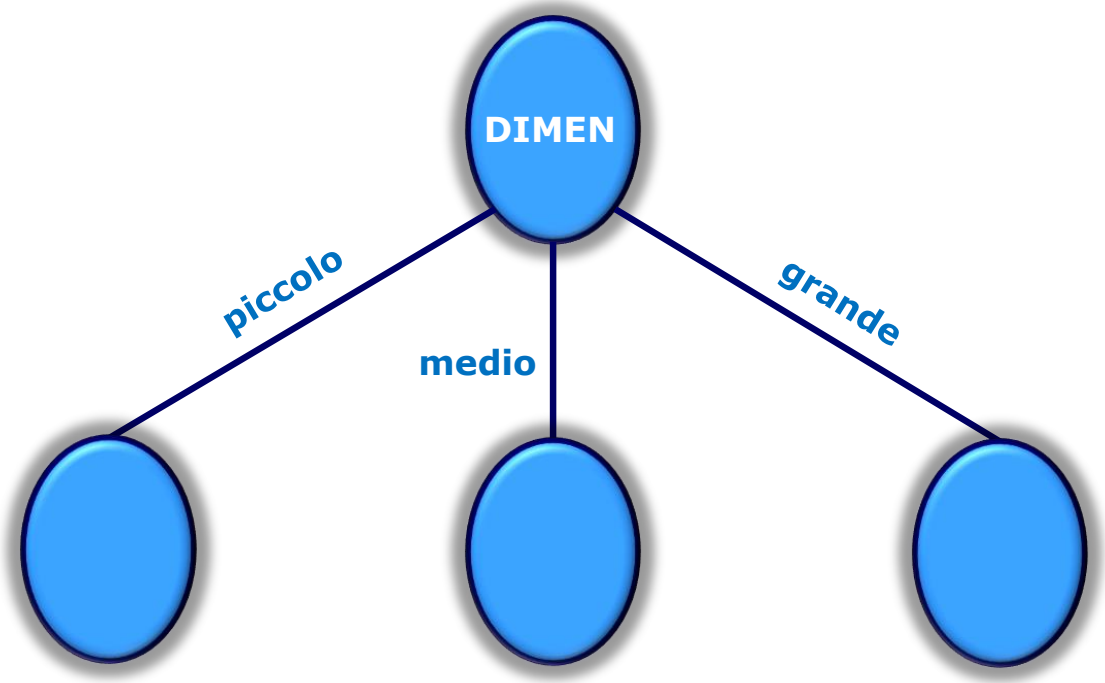


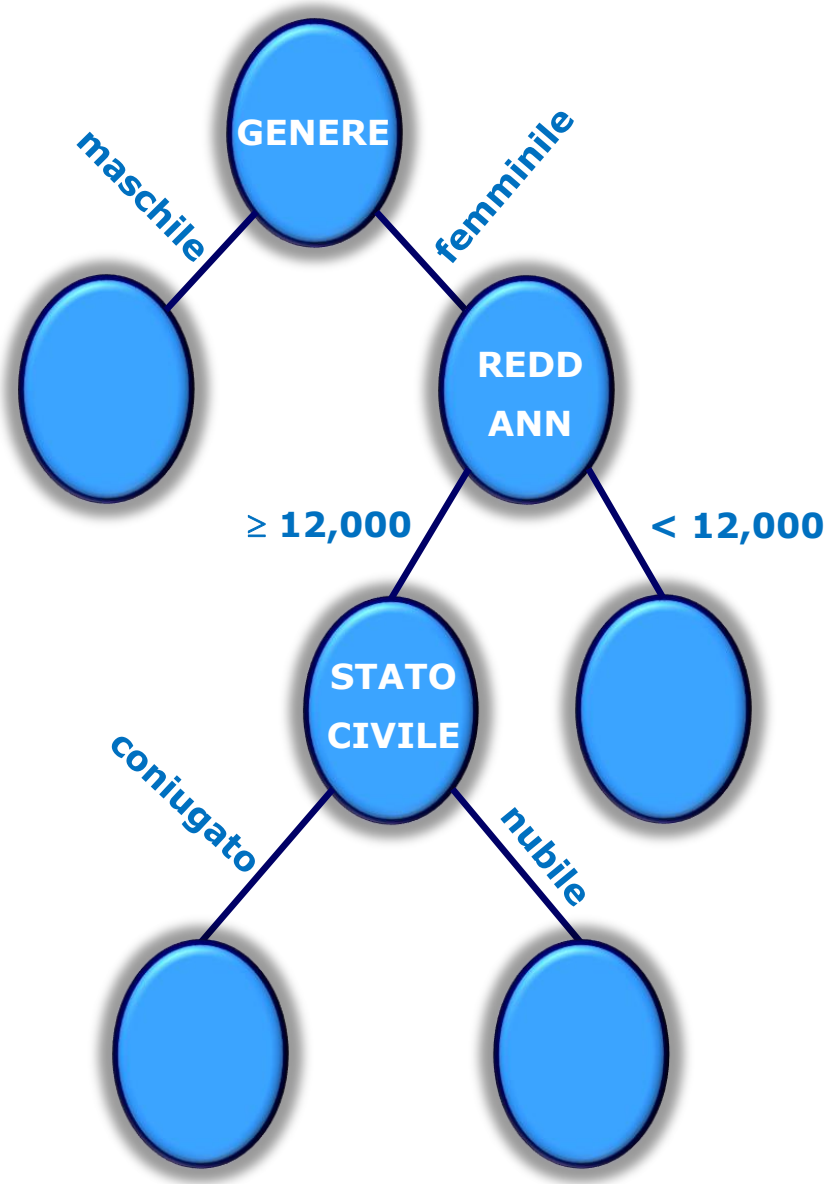
Attributo Nominale: *split binario*



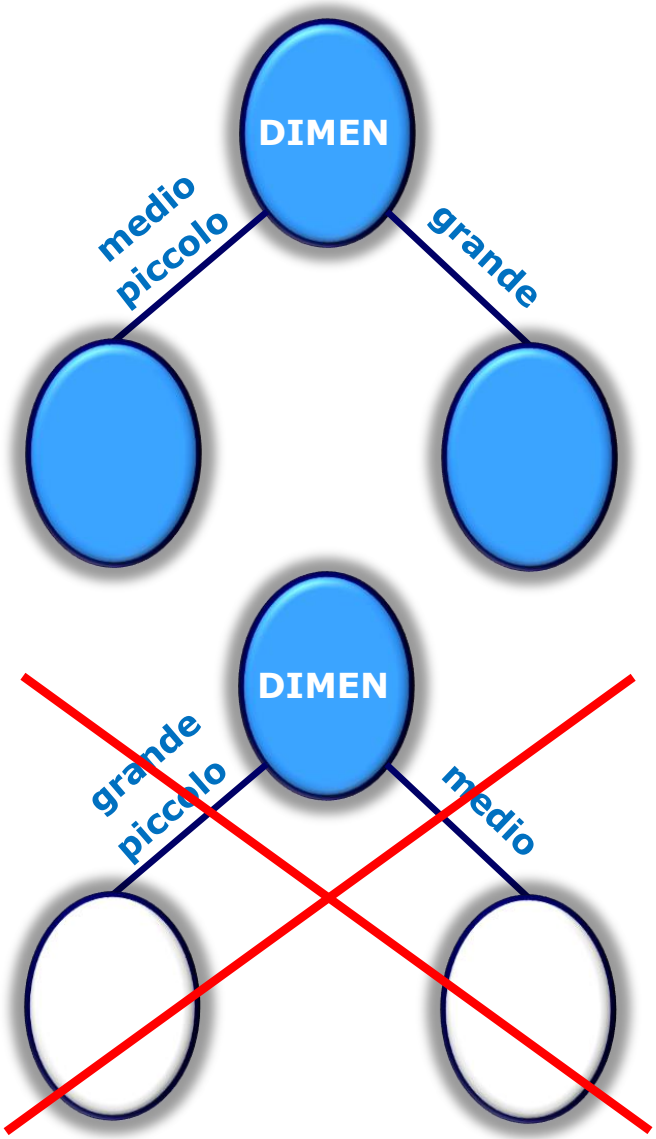


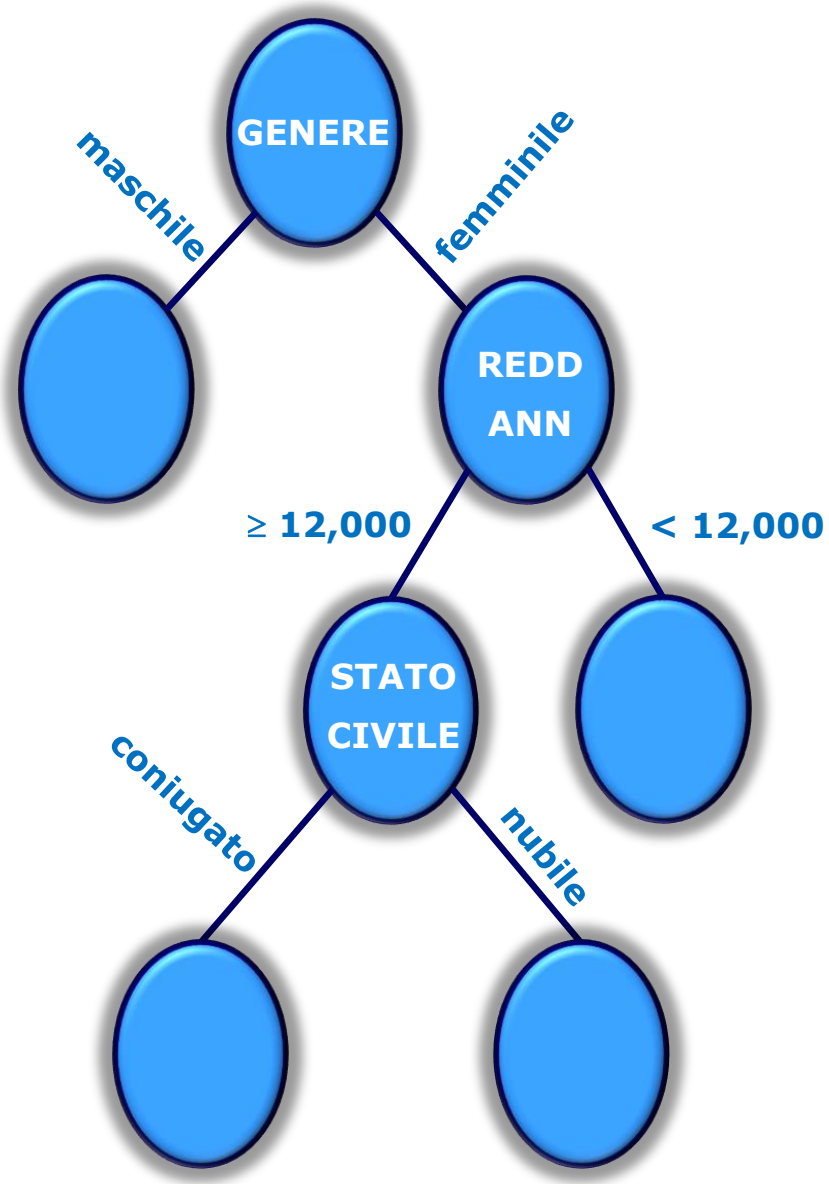
Attributo Ordinale: *split multiplo*





Attributo Ordinale: *split binario*





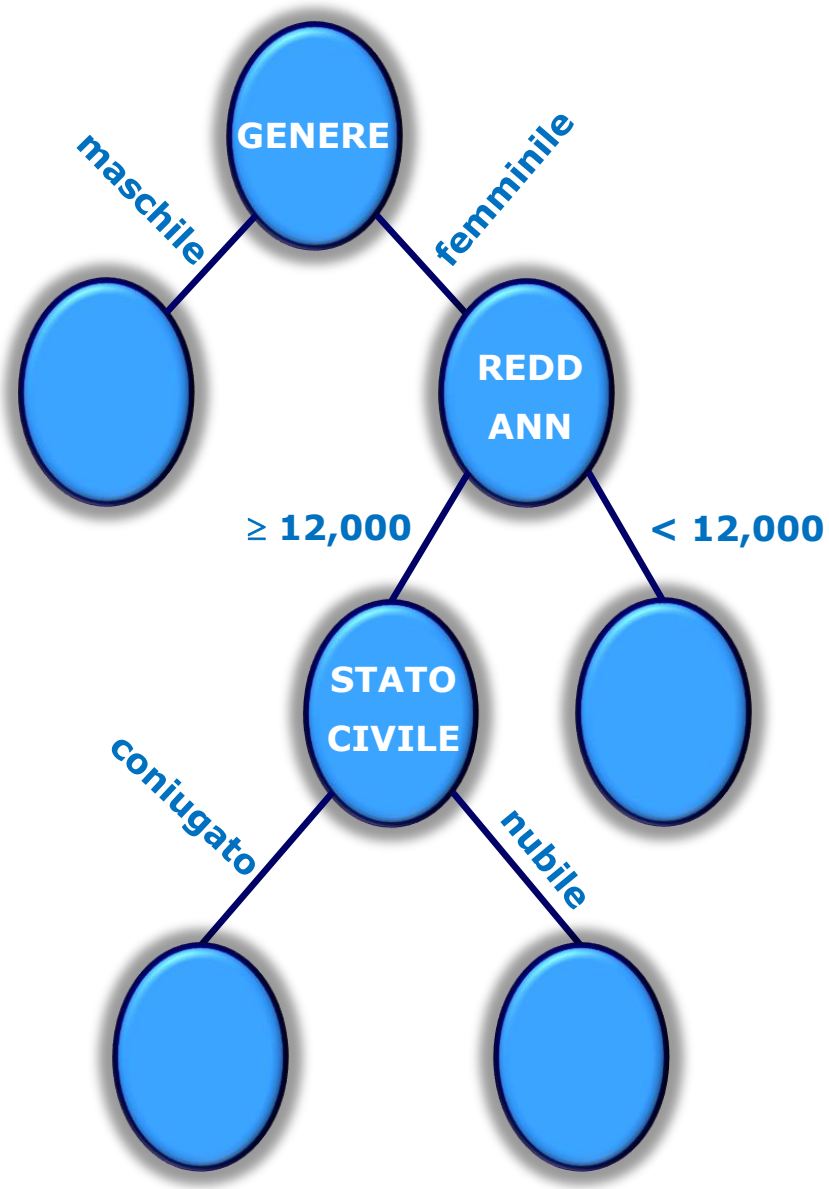
Attributo Continuo

Discretizzazione

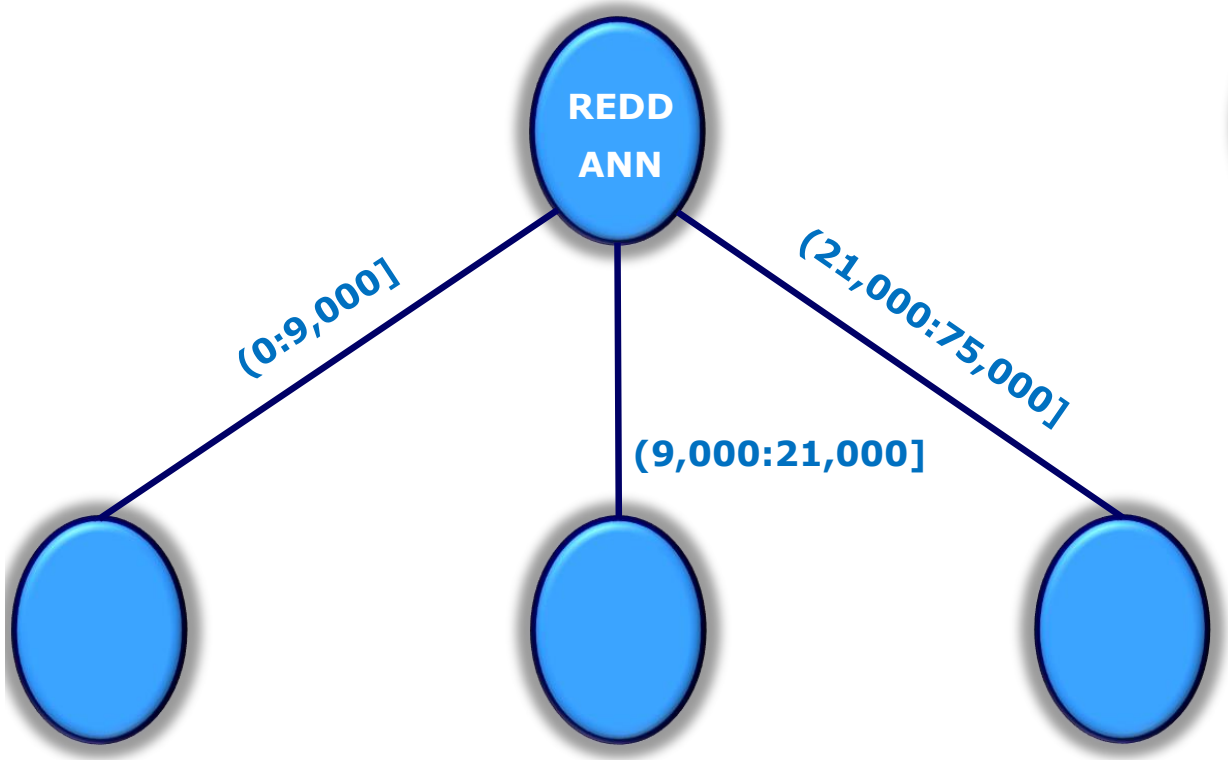
- *statica*, all'inizio dell'apprendimento
- *dinamica*, in corso d'opera, bucketing, clustering, percentili, …

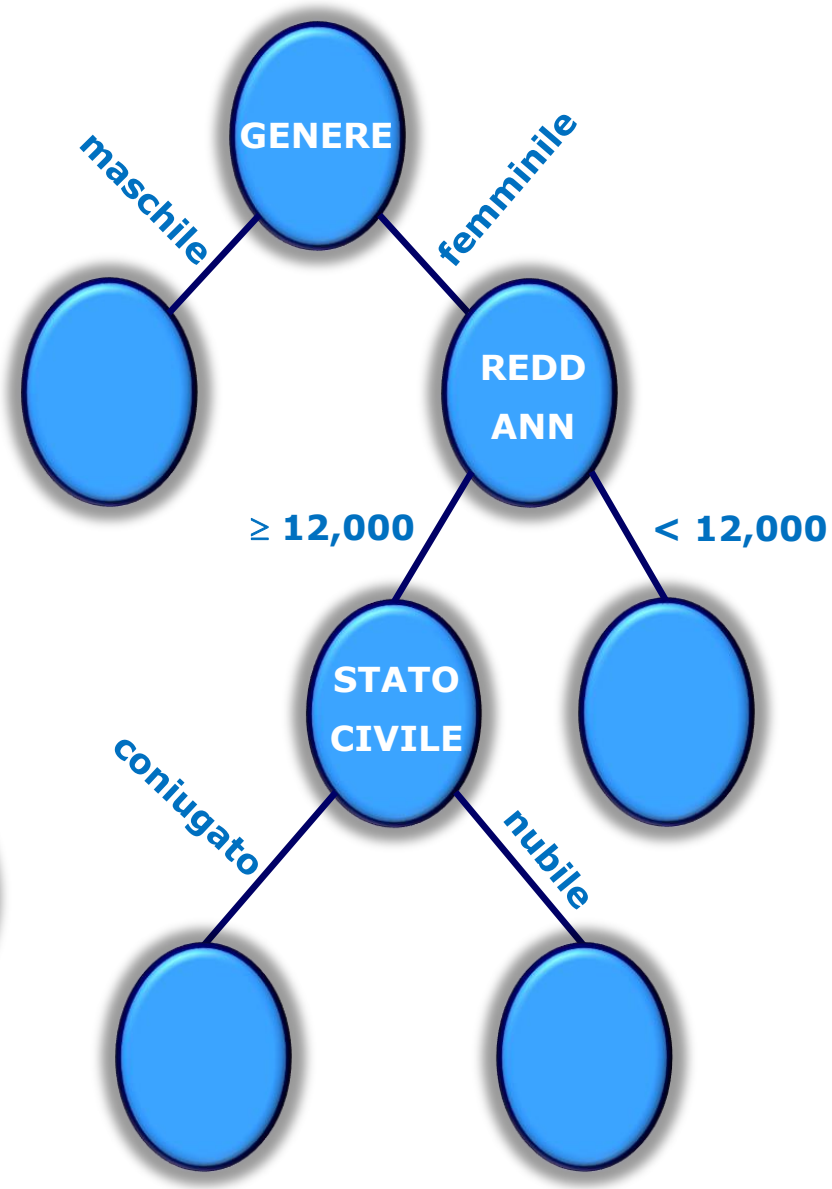
Separazione binaria (A < v) or (A ≥ v)

- *considerare tutti i possibili valori di split "v" e utilizzare il migliore*
- *adottare tecnica più costosa dal punto di vista computazionale*



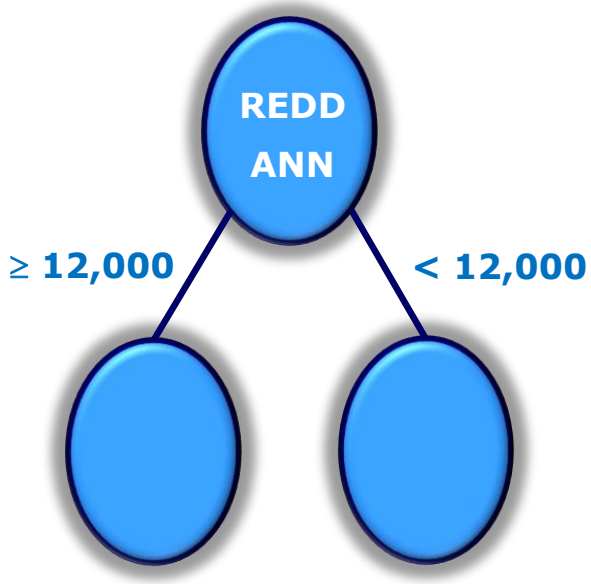
Attributo Continuo: *split multiplo*

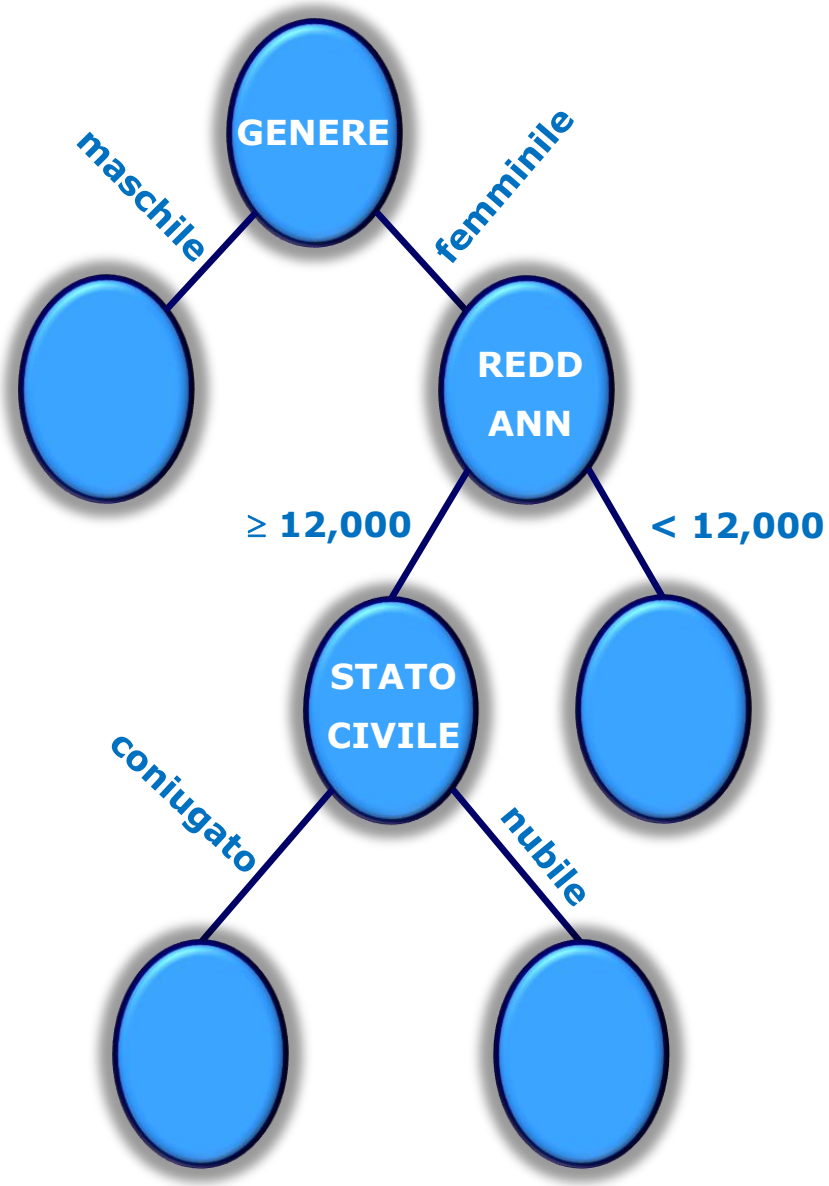




Modelli Euristici: alberi di classificazione

Attributo Continuo: *split binario*





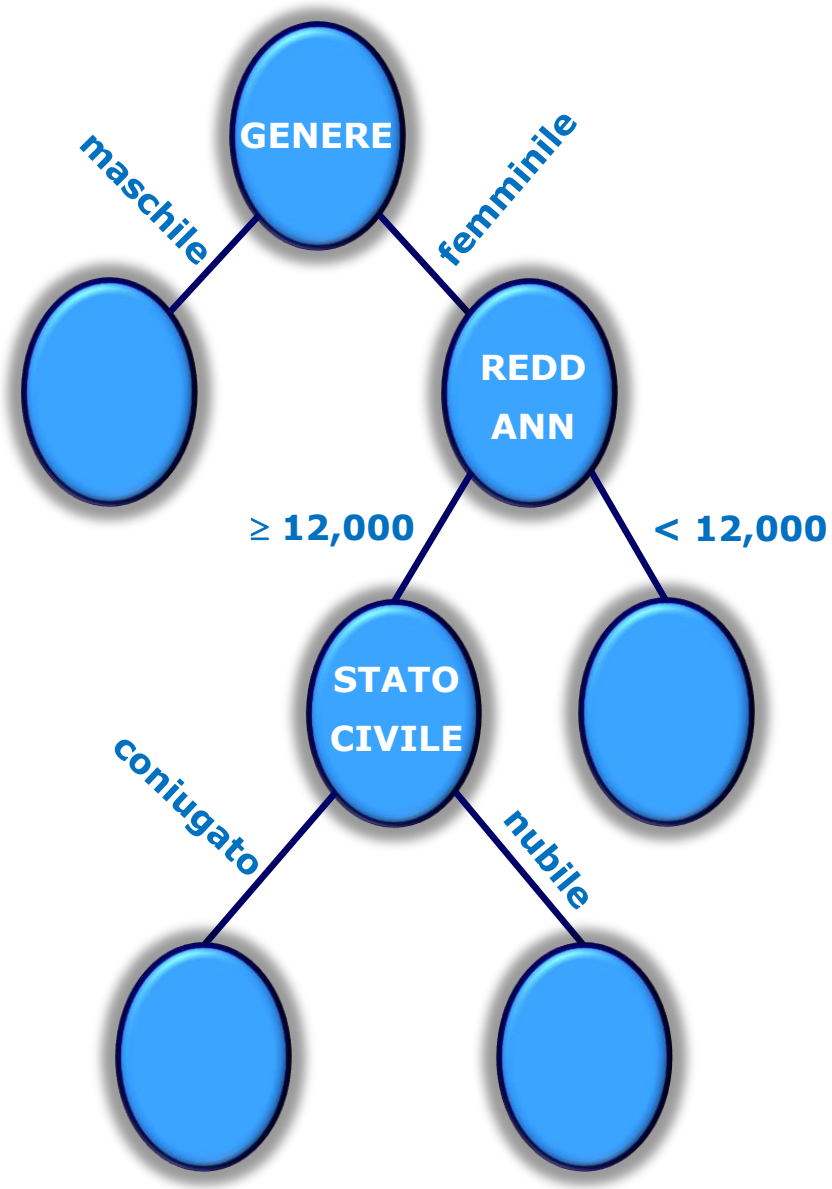
Come determinare lo "split ottimale"?

Dipende dal tipo dell'attributo che deve essere sottoposto allo "*split*"

- *nominale*
- *ordinale*
- *continuo*

Dipende dalle possibili modalità dello "*split*"

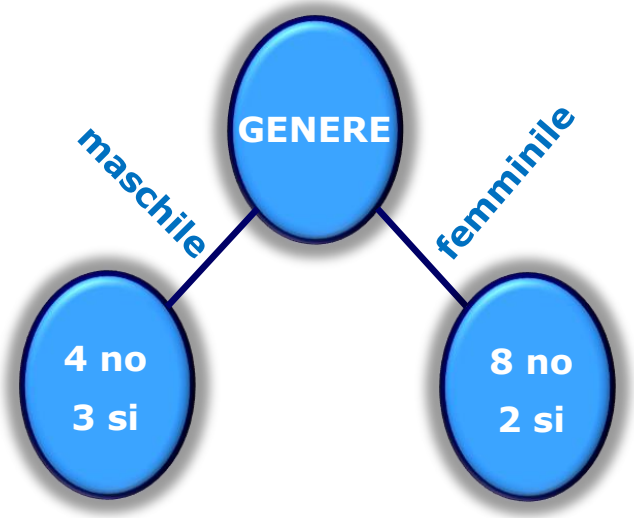
- *binario*
- *multiplo*

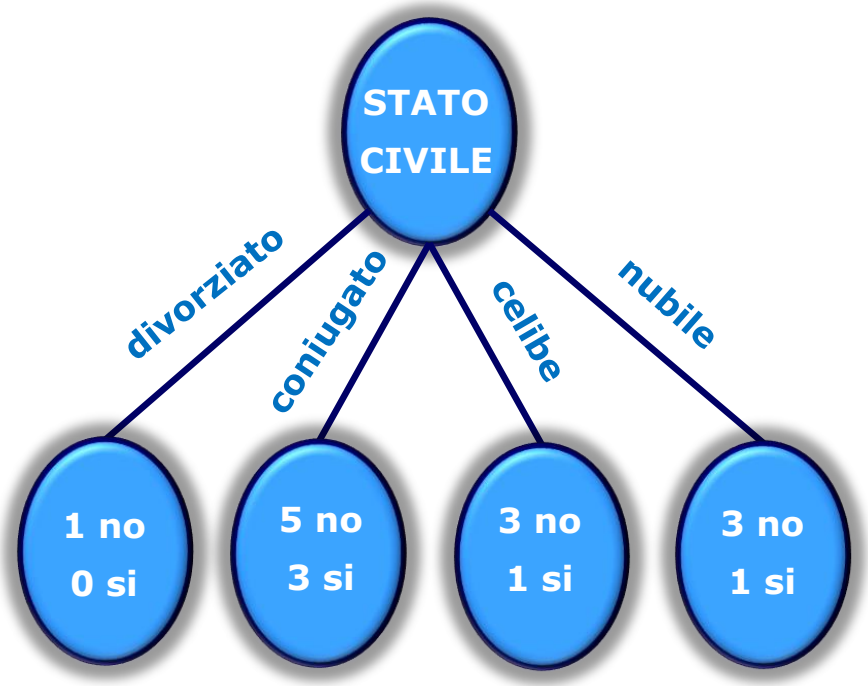


Modelli Euristici: alberi di classificazione

Distribuzione iniziale: **12 no - 5 si**

GENERE	ETA	PROVINCIA	REGIONE	REDD ANN	...	...	STATO CIVILE	EVASORE
maschile	32	VA	Lombardia	20,000 €			celibe	no
femminile	45	AQ	Abruzzo	13,500 €			coniugato	si
femminile	21	RM	Lazio	11,600 €			nubile	no
femminile	62	RM	Lazio	15,350 €			nubile	no
maschile	68	RC	Calabria	10,945 €			divorziato	no
maschile	19	AN	Marche	10,233 €			celibe	no
maschile	24	LT	Lazio	10,450 €			coniugato	si
femminile	22	VI	Veneto	11,567 €			coniugato	no
femminile	29	NO	Piemonte	16,350 €			nubile	no
maschile	52	FI	Toscana	11,245 €			nubile	si
femminile	34	MI	Sicilia	13,450 €			coniugato	no
femminile	33	MI	Basilicata	7,500 €			coniugato	no
femminile	55	TN	Trentino	13,450 €			coniugato	si
maschile	39	FR	Lazio	11,590 €			celibe	si
femminile	55	MI	Lombardia	23,500 €			coniugato	no
maschile	27	MI	Lombardia	35,800 €			celibe	no
femminile	31	AQ	Abruzzo	14,750 €			coniugato	no





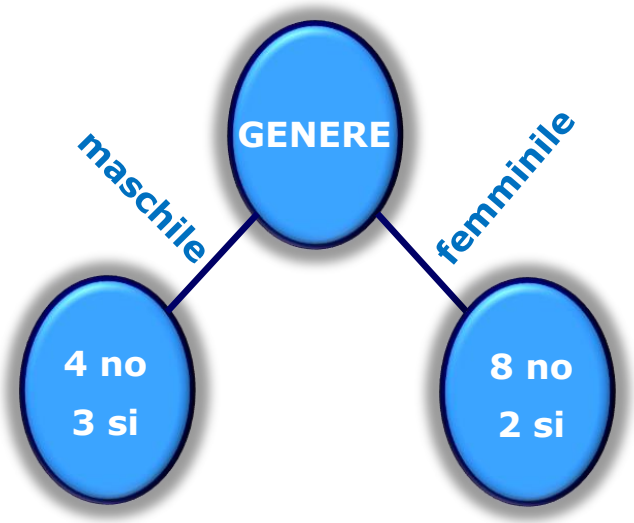
Quale split è preferibile ?

Modelli Euristici: alberi di classificazione

Distribuzione iniziale: **12 no - 5 si**

GENERE	ETA	PROVINCIA	REGIONE	REDD ANN	...	...	STATO CIVILE	EVASORE
maschile	32	VA	Lombardia	20,000 €			celibe	no
femminile	45	AQ	Abruzzo	13,500 €			coniugato	si
femminile	21	RM	Lazio	11,600 €			nubile	no
femminile	62	RM	Lazio	15,350 €			nubile	no
maschile	68	RC	Calabria	10,945 €			divorziato	no
maschile	19	AN	Marche	10,233 €			celibe	no
maschile	24	LT	Lazio	10,450 €			coniugato	si
femminile	22	VI	Veneto	11,567 €			coniugato	no
femminile	29	NO	Piemonte	16,350 €			nubile	no
maschile	52	FI	Toscana	11,245 €			nubile	si
femminile	34	MI	Sicilia	13,450 €			coniugato	no
femminile	33	MI	Basilicata	7,500 €			coniugato	no
femminile	55	TN	Trentino	13,450 €			coniugato	si
maschile	39	FR	Lazio	11,590 €			celibe	si
femminile	55	MI	Lombardia	23,500 €			coniugato	no
maschile	27	MI	Lombardia	35,800 €			celibe	no
femminile	31	AQ	Abruzzo	14,750 €			coniugato	no

Quale split è preferibile ?



non omogeneo
elevata impurità

omogeneo
bassa impurità

Approccio Greedy

I nodi con distribuzione omogenea delle classi sono preferiti.

Necessità di una misura di purezza/impurità del nodo.

Misure di purezza dei nodi

- *Indice di Gini*
- *Indice di Entropia*
- *Indice di mis-classificazione*

Indichiamo con "p_h" la percentuale di osservazioni di classe target "v_h", contenute in un generico nodo "q" e indichiamo con "N_q" il numero di osservazioni complessive associate a tale nodo, per definizione deve valere la seguente relazione

$$\sum_{h=1}^{H} p_h = 1$$

L'indice di eterogeneità "$I(q)$" è funzione delle frequenze relative "p_h", degli "H" valori della classe target per le osservazioni presenti nel nodo "q". Deve soddisfare tre requisiti:

- *assumere valore massimo se le osservazioni sono ripartite in modo equo sulle classi*
- *assumere valore minimo quando le osservazioni appartengono alla stessa classe*
- *rappresentare una funzione simmetrica rispetto alle frequenze relative "p_h".*

Indice di Gini

Definito come

$$G(q) = 1 - \sum_{h=1}^{H} p_h^2$$

Assume *valore nullo* nel caso di *minima eterogeneità*, ovvero quando una sola classe assume valore con frequenza pari a 1 e tutte le altre classi hanno frequenza pari a zero.

Se tutte le classi assumono egual valore della frequenza empirica relativa ($1/H$), l'indice di Gini assume il suo *valore massimo* pari a

$$G(q) = 1 - \sum_{h=1}^{H} p_h^2 = 1 - \sum_{h=1}^{H} \left(\frac{1}{H}\right)^2 = 1 - \frac{H}{H^2} = \frac{H - 1}{H}$$

È possibile normalizzare l'indice in modo tale che assuma valori nell'intervallo [0,1]:

$$G_{rel}(q) = \frac{G(q)}{\left(\frac{H - 1}{H}\right)}$$

Indice di Entropia

Definito come

$$E(q) = -\sum_{h=1}^{H} p_h log_2 p_h$$

Assume *valore nullo* nel caso di *minima eterogeneità*, ovvero quando una sola classe assume valore con frequenza pari a 1 e tutte le altre classi hanno frequenza pari a zero.

Se tutte le classi assumono egual valore della frequenza empirica relativa (1/*H*), l'indice di Entropia assume il suo *valore massimo* pari a

$$E(q) = -\sum_{h=1}^{H} p_h log_2 p_h = -\sum_{h=1}^{H} \frac{1}{H} log_2 \frac{1}{H} = -\frac{1}{H} \sum_{h=1}^{H} log_2 \frac{1}{H} = log_2 H$$

È possibile normalizzare l'indice in modo tale che assuma valori nell'intervallo [0,1]:

$$E_{rel}(q) = \frac{E(q)}{log_2 H}$$

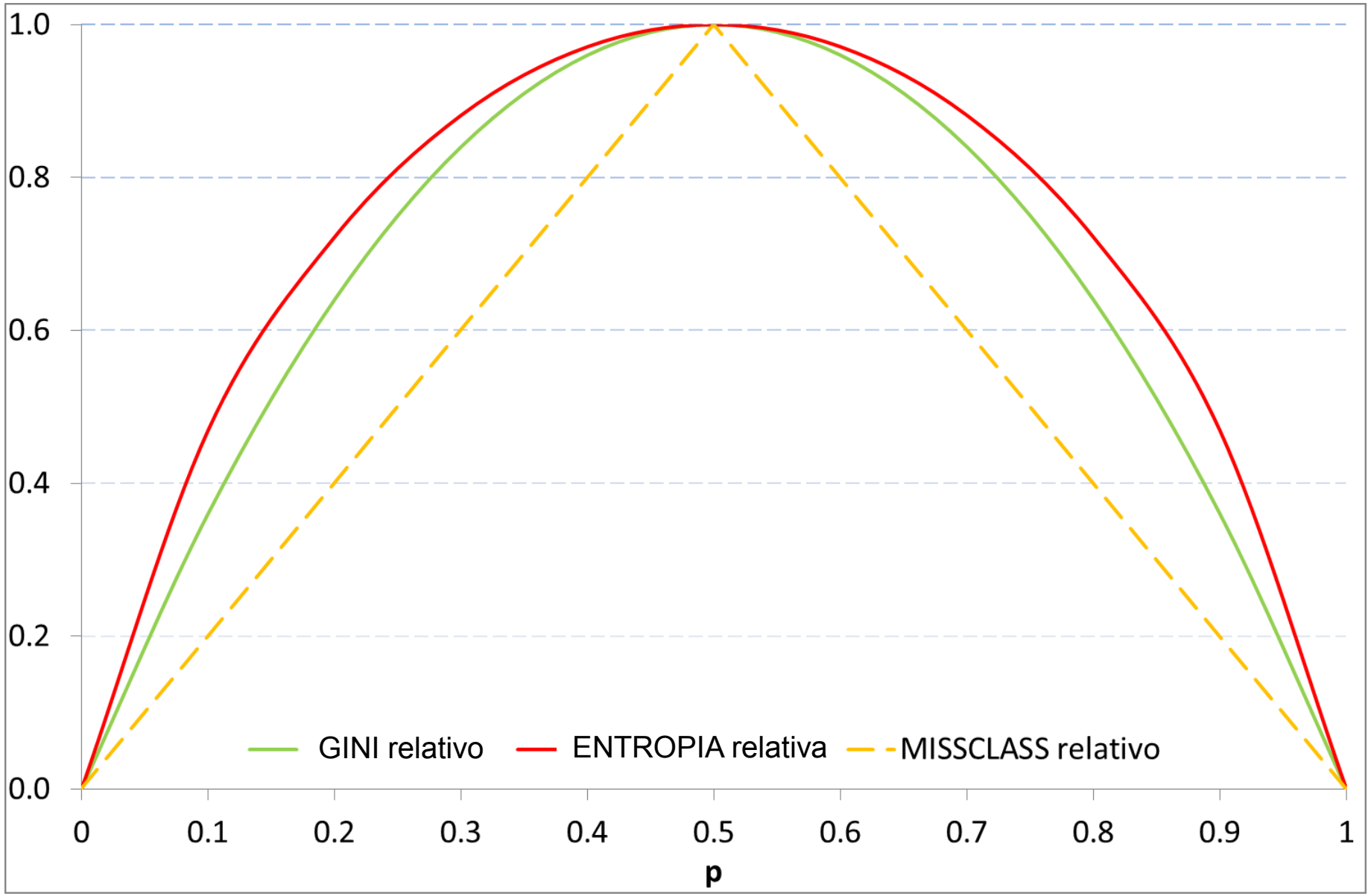
Indice di Mis-classificazione

Definito come

$$Misc(q) = 1 - \max_h p_h$$

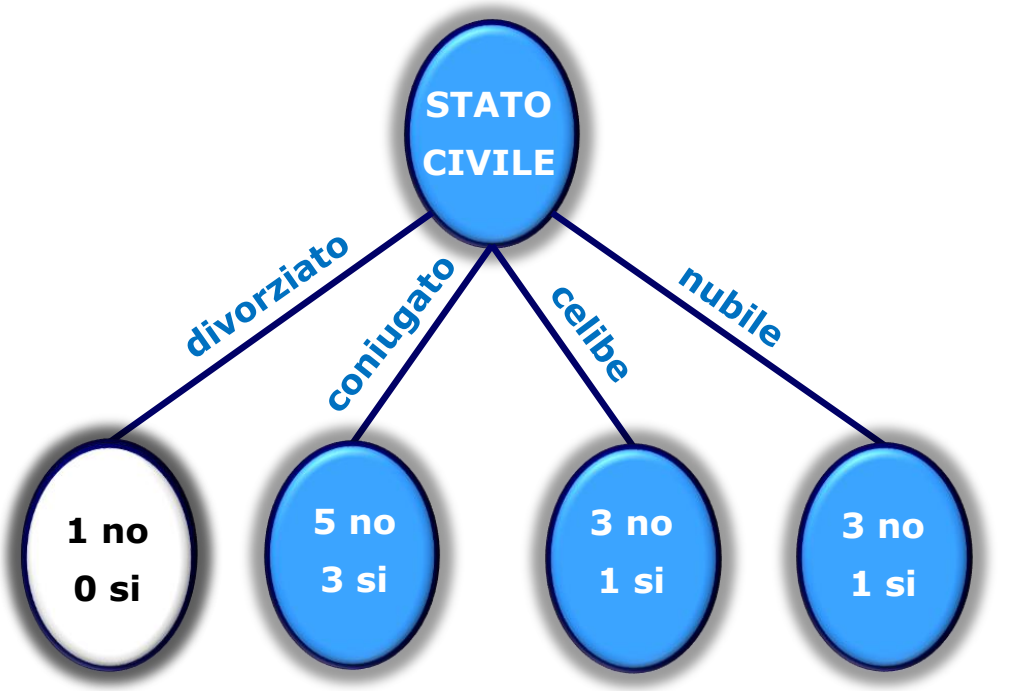
Misura la percentuale di osservazioni appartenenti al nodo "*q*" che vengono classificate in modo errato sotto l'ipotesi di assegnare a tutte le osservazioni il valore della classe cui compete il massimo numero di osservazioni presenti nel nodo "*q*" (criterio di *majority voting*).

Misure di Purezza – Comparazione per classificazione binaria



Misure di Purezza – Indice di Gini

$$G(q) = 1 - \sum_{h=1}^{H} p_h^2$$



$$G(divorziato) = 1 - \sum_{h=1}^{2} p_h^2 =$$

no	1
si	0
0.000	

no	5
si	3
0.468	

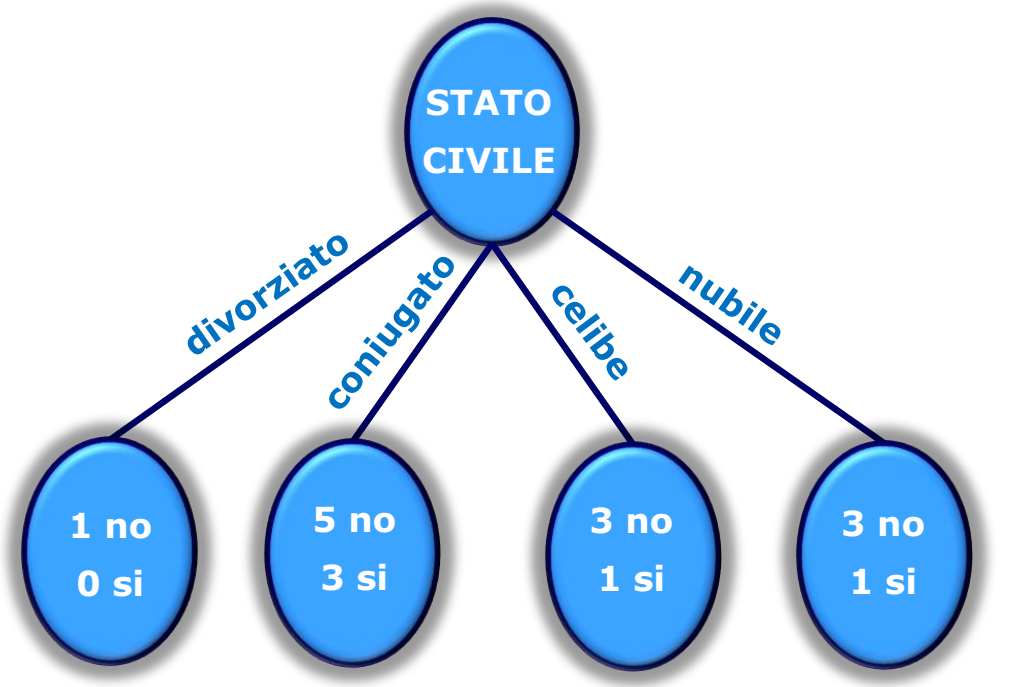
no	3
si	1
0.375	

no	3
si	1
0.375	

$$p_{no} = \frac{1}{1} = 1 \quad p_{si} = \frac{0}{1} = 0$$

Indice di Gini – qualità dello split

$$G(q) = 1 - \sum_{h=1}^{H} p_h^2$$



no	1
si	0
0.000	

no	5
si	3
0.468	

no	3
si	1
0.375	

no	3
si	1
0.375	

$$G_{split}(q) = \sum_{i=1}^{k} \frac{n_i}{N_q} G(q_i) = \frac{1}{17} \times 0 + \frac{8}{17} \times 0.468 + \frac{4}{17} \times 0.375 + \frac{4}{17} \times 0.375 = 0.397$$

Indice di Gini – attributo categorico

Split Multiplo

	Tipo vettura		
	jeep	sport	lusso
si	1	2	1
no	4	1	1
	0.393		

Split Binario

	Tipo Vettura	
	{sport,lusso}	{jeep}
si	3	1
no	2	4
	0.400	

	Tipo Vettura	
	{sport}	{jeep,lusso}
si	2	2
no	1	5
	0.419	

	Tipo Vettura	
	{sport,jeep}	{lusso}
si	3	1
no	5	1
	0.475	

Lo split multiplo ottiene un valore dell'Indice di Gini migliore (più piccolo).

Indice di Gini – attributo continuo

- Utilizzare splitting binario che sfrutta la scelta di un valore "**v**" dell'attributo continuo
- Diverse alternative per la scelta del valore "**v**" di splitting
  - Numero di valori di splitting candidati pari al numero di valori distinti dell'attributo
- Ogni valore di splitting è associato ad una matrice di conteggi
  - Si registra il conteggio per la classe in corrispondenza di ogni partizione

$$A < v \quad \text{o} \quad A \geq v$$

- Scelta banale, scegliere il miglior valore di splitting "**v**"
  - per ogni valore "**v**", scandire il dataset e costuire la corrispondente matrice di conteggi, calcolare il corrispondente valore dell'Indice di Gini
  - computazionalmente inefficiente, si effettua la stessa operazione più e più volte.

Indice di Gini – attributo continuo

Computazione efficiente

- ordinare in modo crescente i valori assunti dall'attributo
- scandire sequenzialmente l'ordinamento precedente, aggiornando la matrice dei conteggi per la classe e computando il relativo valore dell'Indice di Gini
- splitting ottimale, quello cui compete il minimo valore dell'Indice di Gini

Evasore	no	no	no	si	si	si	no	no	no	no
	Reddito									
Valori ordinati →	60	70	75	85	90	95	100	120	125	220
Posizione di split →	55	65	72	80	87	92	97	110	122	172	230

	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
si	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
no	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
	0.42		0.40		0.37		0.34		0.42		0.40		***0.30***		0.34		0.37		0.40		0.42	

Indice di Entropia – Splitting basato sull'Information Gain

Dato il nodo "*q*", suddiviso in "*k*" partizioni, indichiamo con "n_i" il numero di osservazioni appartenenti alla partizione "*i*". Si calcola l'*Information Gain* come segue:

$$GAIN_{split}(q) = E(q) - \left(\sum_{i=1}^{k} \frac{n_i}{N_q} E(q_i) \right)$$

- misura la riduzione dell'entropia ottenuta tramite lo split. Si sceglie lo split che garantisce la massima riduzione (massimizza l'Information Gain)
- approccio utilizzato da istanze specifiche (**ID3** e **C4.5**)
- **svantaggio:** tende a preferire split che portano a un numero elevato di partizioni, ogni partizione è piccola ma pura.

Indice di Entropia – Splitting basato sull'Information Gain

Correzione per mitigare i problemi dell'*Information Gain*

$$SplitINFO(q) = -\sum_{i=1}^{k} \frac{n_i}{N_q} \times log\left(\frac{n_i}{N_q}\right)$$

$$GAINratio_{split}(q) = \frac{GAIN_{split}(q)}{SplitINFO(q)}$$

- Partizionamenti con maggiori valori di entropia (numero elevato di piccole partizioni) venono penalizzate
- Adottato da **C4.5**

Splitting basato sull'Indice di Mis-classificazione

Splitting basato sull'*Indice di mis-classificazione*

$$Misc(q) = 1 - \max_h p_h$$

- Raggiunge il massimo quando le osservazioni sono egualmente distribuite tra tutte le classi rappresentate nel nodo, indica che nel nodo è racchiusa poca informazione.
- Raggiunge valore minimo se tutte le osservazioni appartengono alla stessa classe, indica che il nodo contiene il massimo di informazione possibile.

Condizione di arresto del processo di splitting

Il processo di splitting può essere arrestato nei seguenti modi

- se un nodo contiene osservazioni tutte appartenenti alla stessa classe
- se un nodo contiene osservazioni con valori identici degli attributi
- terminazione precoce (*Pre-Pruning*), ci si arresta se
  - *il numero di osservazioni è minore di una certa soglia specificata dall'utente*
  - *le distribuzioni delle classi sono indipendenti dagli attributi disponibili (impiego di un test Chi-quadro, …)*
  - *espandendo il nodo corrente non viene migliorato il valore della misura di purezza (Indice di Gini, Information Gain, …)*

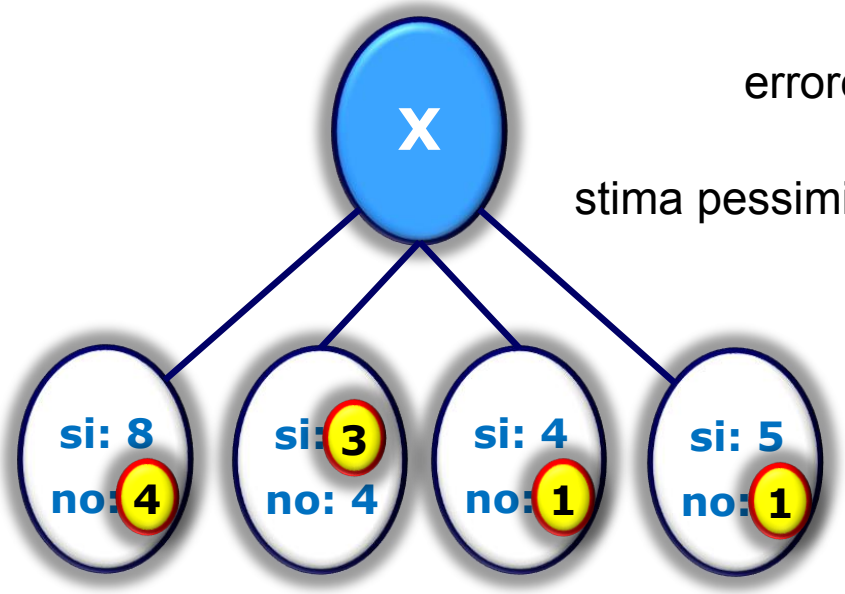
Post-Pruning

Prima di effettuare lo splitting del nodo abbiamo la seguente situazione

classe = si	20
classe = no	10

errore di training (pre-split) = 10/30

stima pessimistica errore = (10+0.5)/30 = 10.5/30

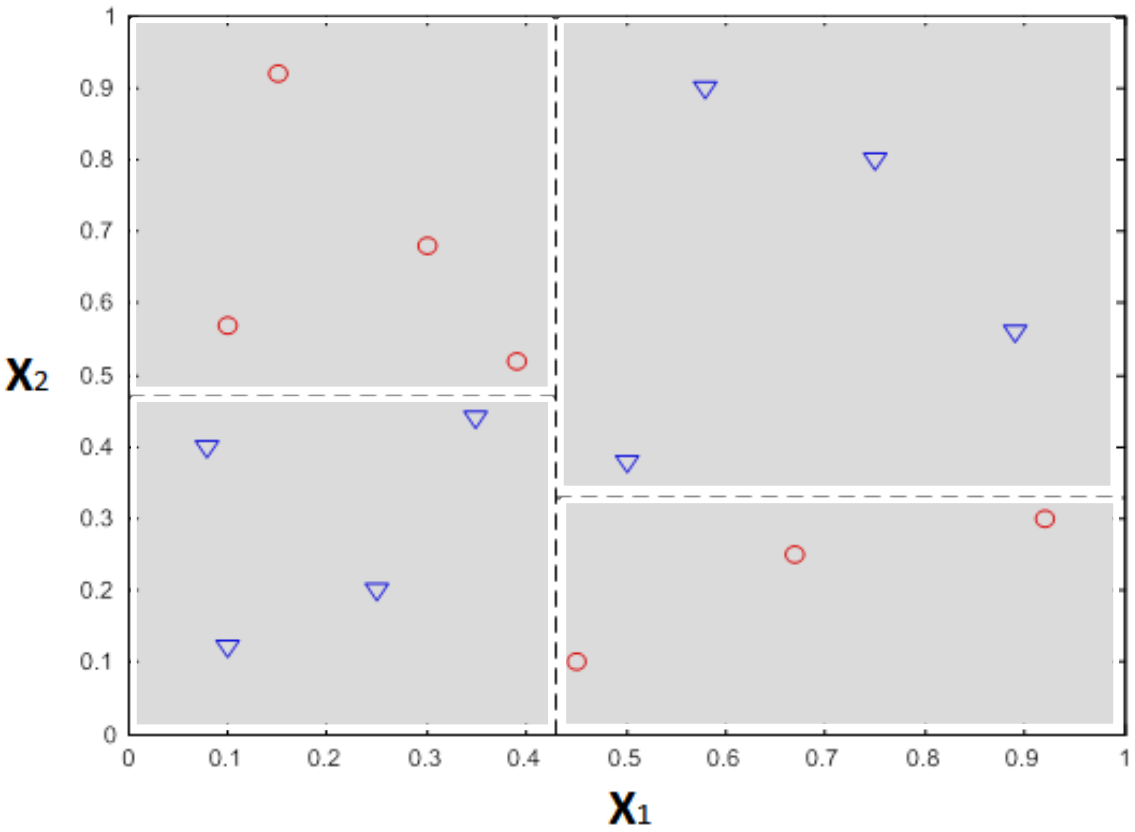


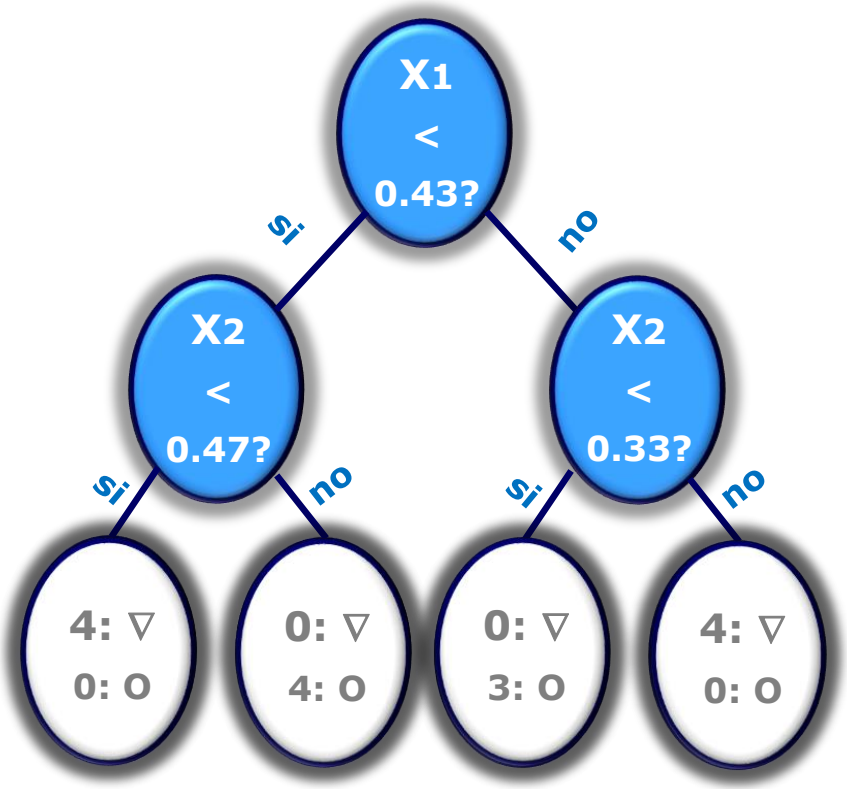
errore di training (post-split) = (4+3+1+1)/30 = 9/30

stima pessimistica errore (post-split) = (9+4x0.5)/30 = 11/30

DECISIONE → PRUNE!

Modelli Euristici: alberi di classificazione

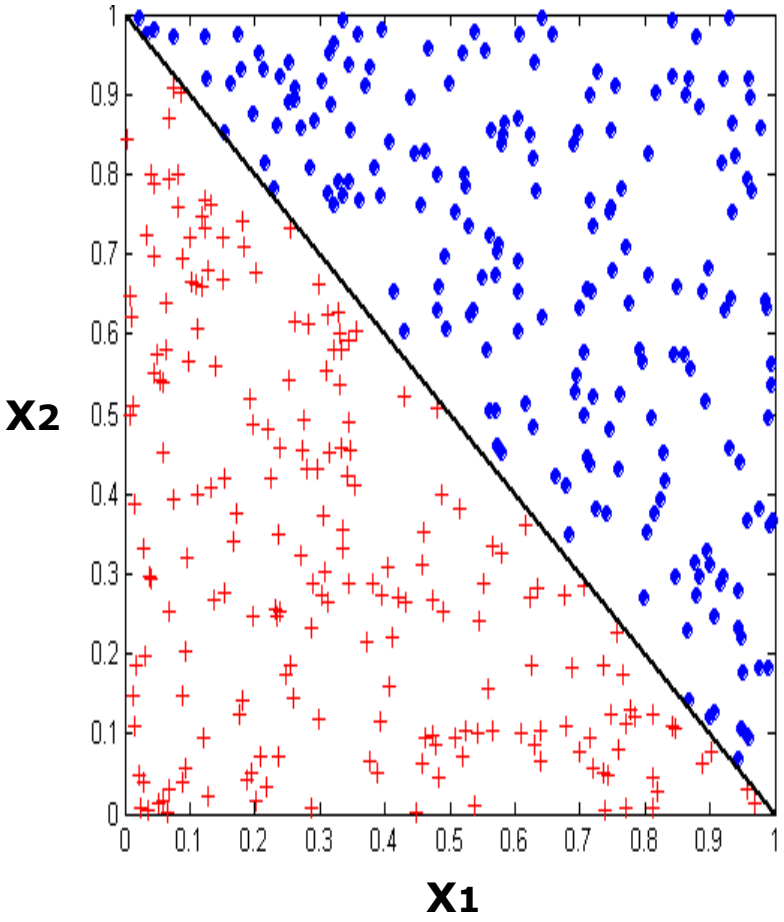


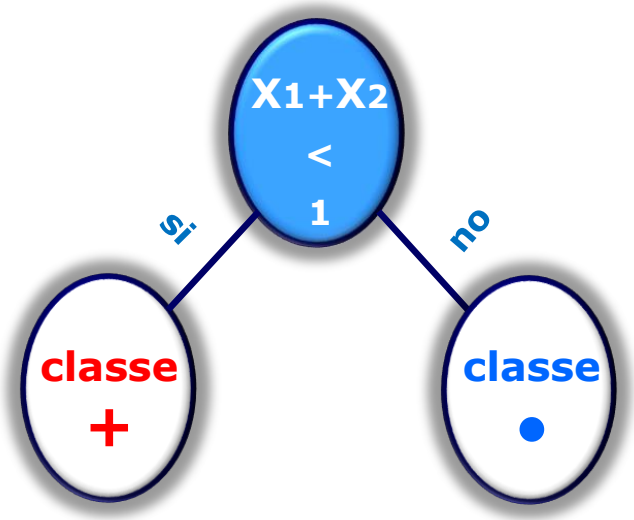


Decision boundary, definiscono un cambio di classificazione.

Parellele agli assi associati agli attributi in quanto l'albero di decisione che abbiamo presentato conduce test univariati.

Modelli Euristici: alberi di classificazione



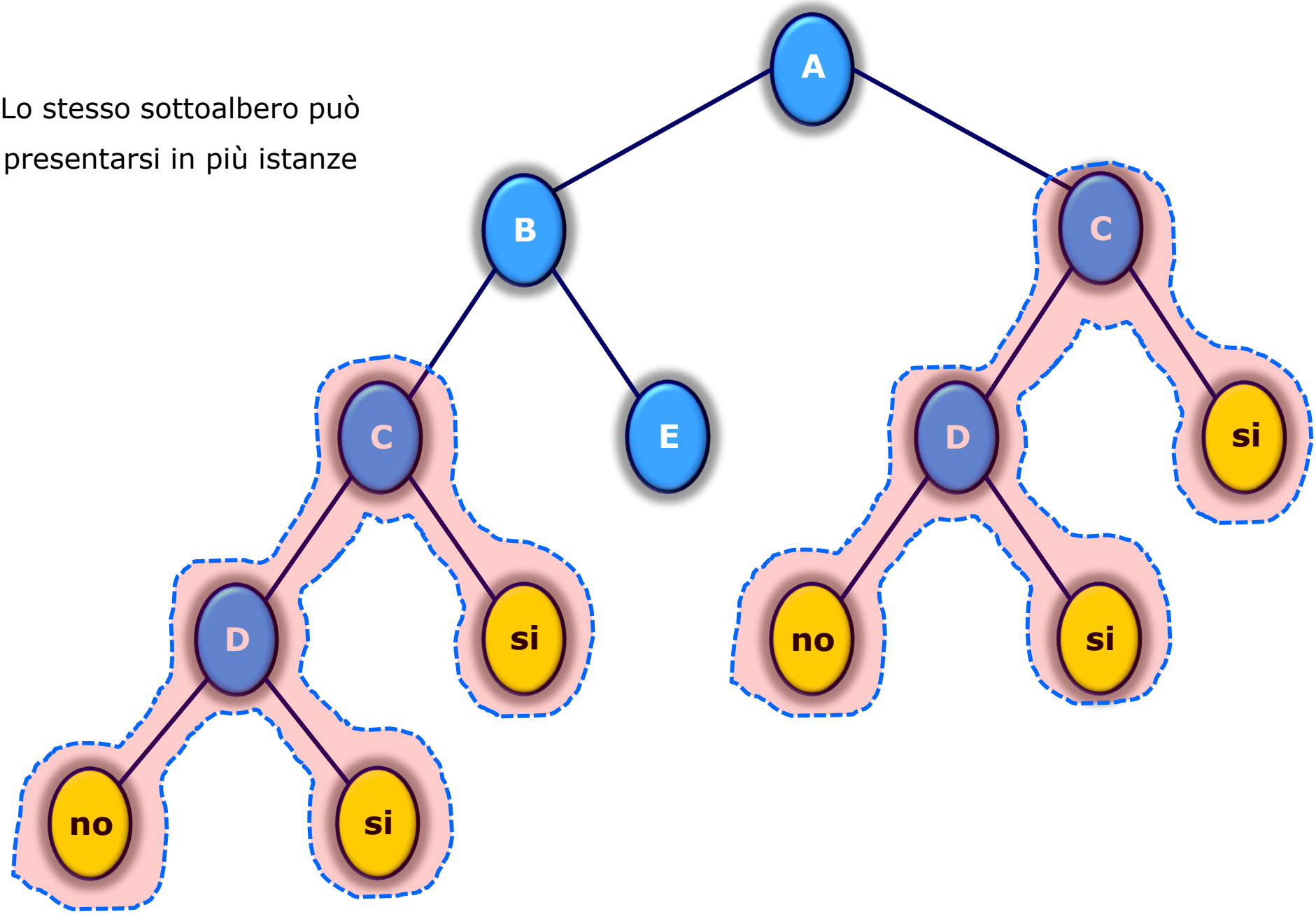


La condizione di test può coinvolgere più attributi contemporaneamente

Maggiore capacità espressiva

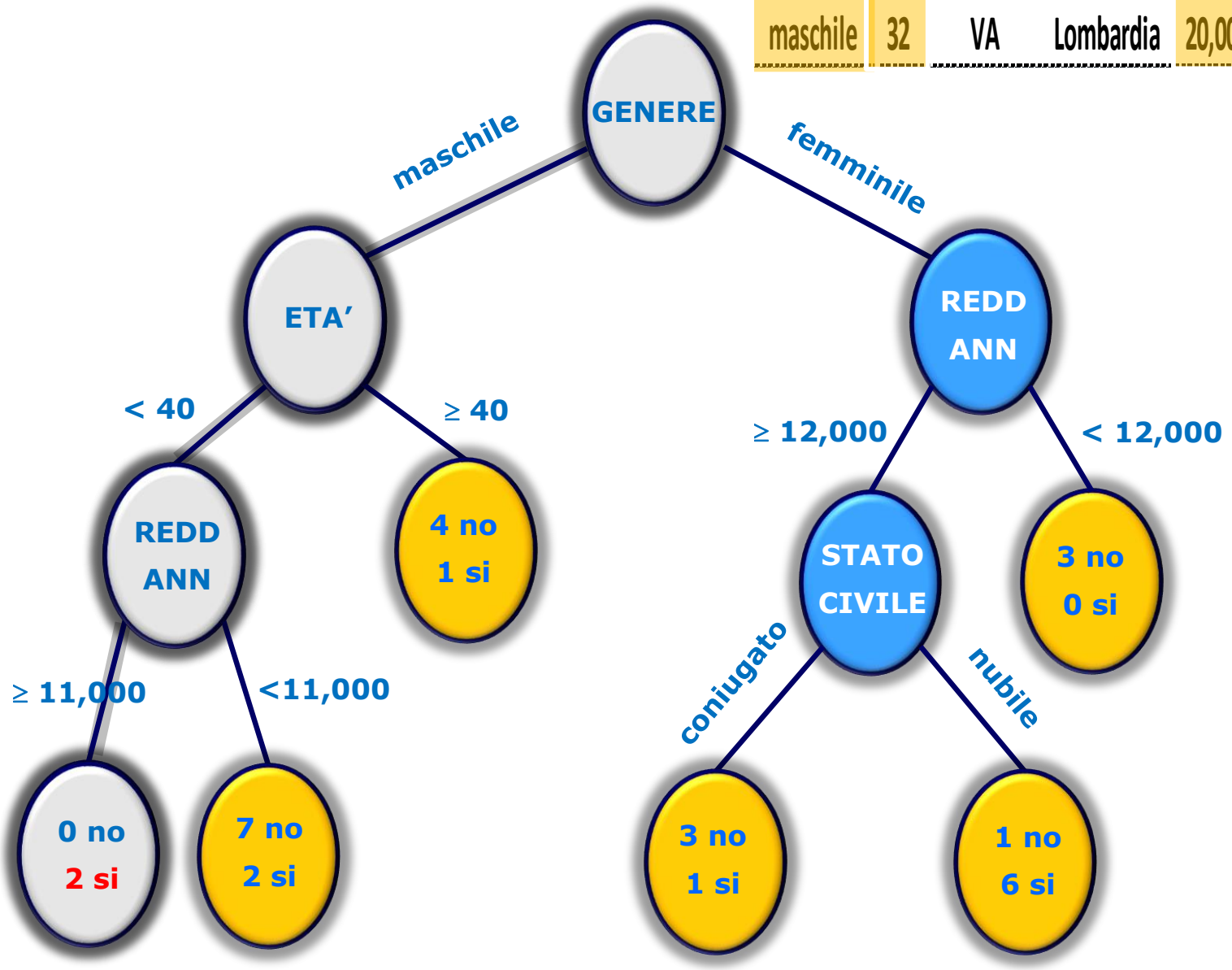
Trovare il test ottimale da effettuare è computazionalmente molto costoso.

Lo stesso sottoalbero può presentarsi in più istanze



Modelli Euristici: alberi di classificazione

GENERE	ETA	PROVINCIA	REGIONE	REDD ANN	...	...	STATO CIVILE	EVASORE
maschile	32	VA	Lombardia	20,000 €			celibe	si



Modelli Euristici: nearest neighbor

Classificatori che appartengono alla categoria degli *Instance Based Classifier*, sono noti anche come metodi pigri di apprendimento o *lazy learner*.

Richiedono siano disponobili tre ingredienti:

- *insieme dei dati di apprendimento*
- *misura di distanza*
- *valore del parametro "k", numero di vicini che verranno interrogati*

Un'osservazione (nuova) viene classificata come segue:

- *calcolare la sua distanza dalle osservazioni del training dataset*
- *identificare le "k" osservazioni più prossime all'osservazione da classificare*
- *associare la classe che è maggiormente rappresentata tra i "k" vicini dell'osservazione da classificare (majority voting).*

Modelli Euristici: nearest neighbor

fissiamo k=4

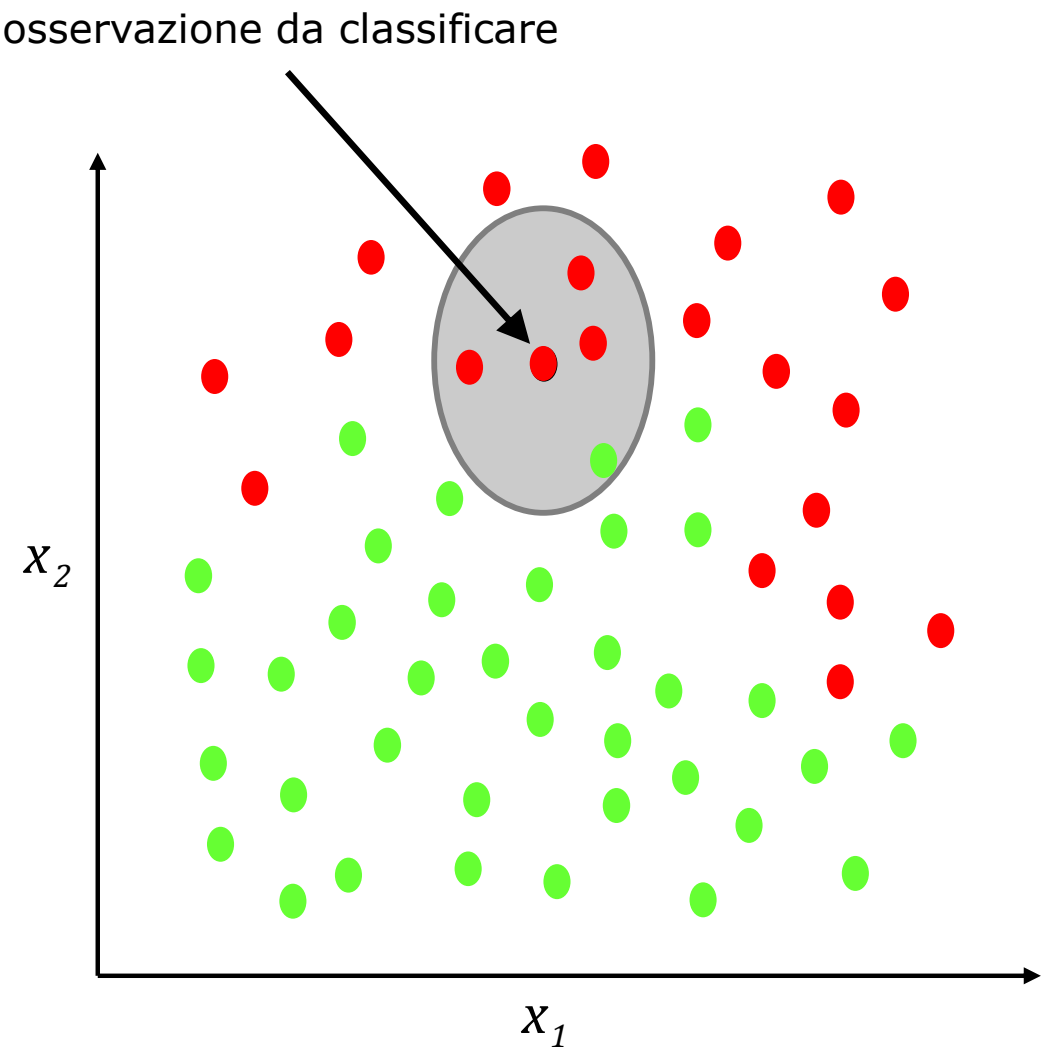
distanza Euclidea

due classi

3 rosse

1 verde

rosso

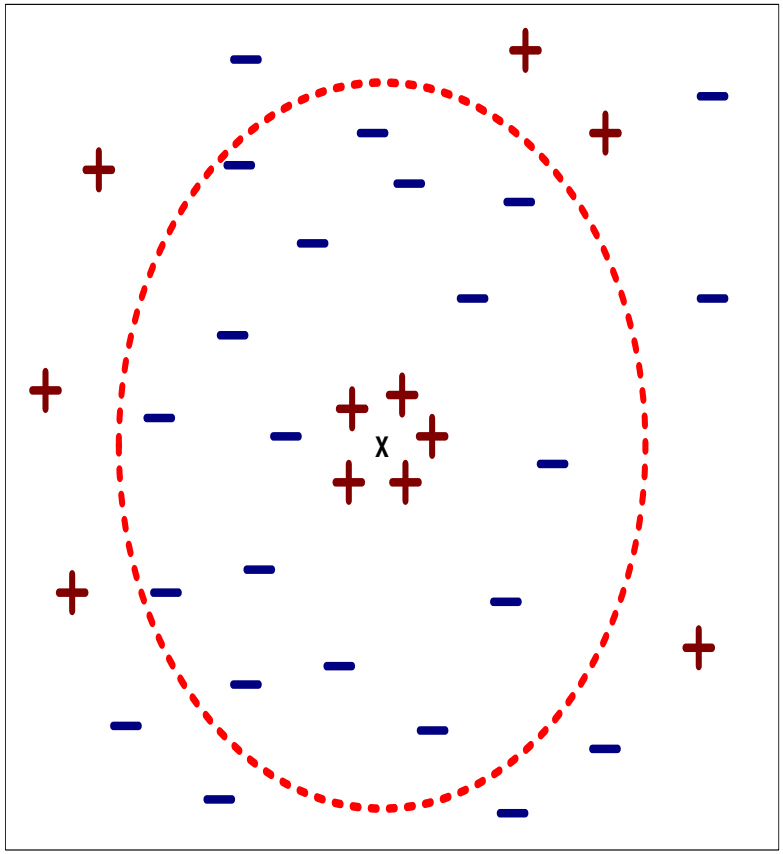


È possibile pesare il voto di ogni osservazione in modo inversamente proporzionale alla distanza.

Scelta del valore di "*k*"

Se il valore "*k*" è troppo piccolo, elevata sensibilità al rumore

Se il valore di "*k*" è troppo grande, il vicinato (*neighborhood*) potrebbe includere osservazioni provenienti da altre classi.



Scaling dei dati

Alcuni attributi potrebbero dominare la misura di distanza, per rimuovere questo rischio è possibile applicare scaling e standardizzazioni ai valori degli attributi.

Problema della distanza Euclidea

Problema computazionale se si ha a che fare con dataset altamente dimensionali.

Possibile ottenere risultati contro intuitivi.

1 1 1 1 1 1 1 1 1 1 1 0		1 0 0 0 0 0 0 0 0 0 0 0	
	vs		
0 1 1 1 1 1 1 1 1 1 1 1		0 0 0 0 0 0 0 0 0 0 0 1	
d = 1.4142		d = 1.4142	

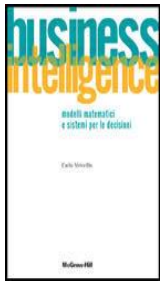
Soluzione: normalizzare i vettori ad avere lunghezza unitaria.

CLASSIFICAZIONE

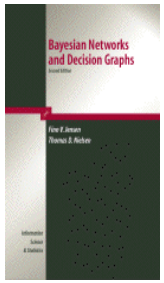
MODELLI PROBABILISTICI

Classificazione

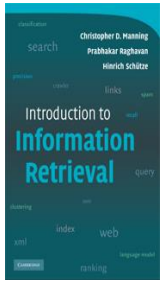
Parte dei contenuti della presente lezione sono tratti dai testi elencati di seguito.



Carlo Vercellis (2006). *Business intelligence: modelli matematici e sistemi per le decisioni*, McGraw-Hill.



Finn V. Jensen and Thomas D. Nielsen (2007). *Bayesian networks and decision graphs*, Springer.



Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze (2008). *Introduction to information retrieval*, Cambridge University Press.

Modelli Probabilistici

I modelli probabilistici risolvono il problema della classificazione supervisionata tramite l'utilizzo della seguente probabilità condizionata

$$P(Y \mid \underline{X})$$

dove per il momento assumiamo che Y sia una variabile binaria mentre con $\underline{\mathbf{X}}$ indichiamo un vettore binario *n*-dimensionale. Inoltre, indicheremo con X_i la componente *i*-ma del vettore $\underline{\mathbf{X}}$.

In accordo alla *formula di Bayes* possiamo scrivere

$$P(Y = y_i \mid \underline{X} = x_k) = \frac{P(\underline{X} = x_k \mid Y = y_i) \cdot P(Y = y_i)}{\sum_j P(\underline{X} = x_k \mid Y = y_j) \cdot P(Y = y_j)}$$

dove y_i indica l'i-mo elemento del supporto di Y, mentre x_k indica la *k*-ma possibile assegnazione del vettore $\underline{\mathbf{X}}$.

Modelli Probabilistici

Il modello *Naive Bayes* sfrutta la nozione di indipendenza condizionale per rendere trattabile il processo di stima dei parametri della probabilità condizionata $P(\underline{X}|Y)$. Tale modello è in grado di garantire una riduzione eccezionale del numero di parametri da stimare

$$\theta_{ki} = P(\underline{X} = x_k \, / \, Y = y_i) \quad k \in 2^n, \quad y_i \in \{-1,+1\}$$

consente di passare *da $2(2^n-1)$ a solo $2n$!!!*

Indipendenza Condizionale

Date tre variabili aleatorie X, Y e Z, diremo che X è *condizionalmente indipendente* da Y dato (noto) Z, se e solo se la probabilità che governa X è indipendente dal valore assunto dalla variabile Y una volta che sia noto il valore assunto dalla variabile Z; formalmente

$$\forall i,j,k \quad P(X = x_i \mid Y = y_j, Z = z_k) = P(X = x_i \mid Z = z_k)$$

Modelli Probabilistici: naïve Bayes

L'ipotesi di *indipendenza condizionale degli attributi* X_1, X_2, …, X_n *data la conoscenza della variabile di classe* Y, fatta dal modello Naive Bayes, riduce drasticamente la complessità del problema di stima delle probabilità condizionate $P(\underline{X}|Y) = P(X_1, ..., X_n|Y)$, consentendo di scrivere la seguente relazione

$$P(X_1,...,X_n \mid Y) = \prod_{i=1}^{n} P(X_i \mid Y)$$

Passiamo a descrivere nel dettaglio il modello, assumendo che Y sia una variabile discreta che assuma un pluralità di valori e che gli attributi X_1, …, X_n siano variabili discrete o reali.

Il nostro obiettivo consiste nell'apprendere un classificatore che restituisca un valore di probabilità per i valori assunti dalla variabile di classe Y, in corrispondenza di ogni nuova istanza **$\underline{X}$** che gli venga sottoposta.

L'espressione della probabilità che Y assuma il *k*-mo valore, in accordo alla regola di Bayes, è

$$P(Y = y_k \mid X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n \mid Y = y_k)\cdot P(Y = y_k)}{\sum_j P(X_1,\ldots,X_n \mid Y = y_j)\cdot P(Y = y_j)}$$

Sfruttando l'ipotesi di indipendenza condizionale degli attributi data la classe otterremo:

$$P(Y = y_k \mid X_1,\ldots,X_n) = \frac{P(Y = y_k)\cdot \prod_{i=1}^{n} P(X_i \mid Y = y_k)}{\sum_j P(Y = y_j)\cdot \prod_{i=1}^{n} P(X_i \mid Y = y_j)}$$

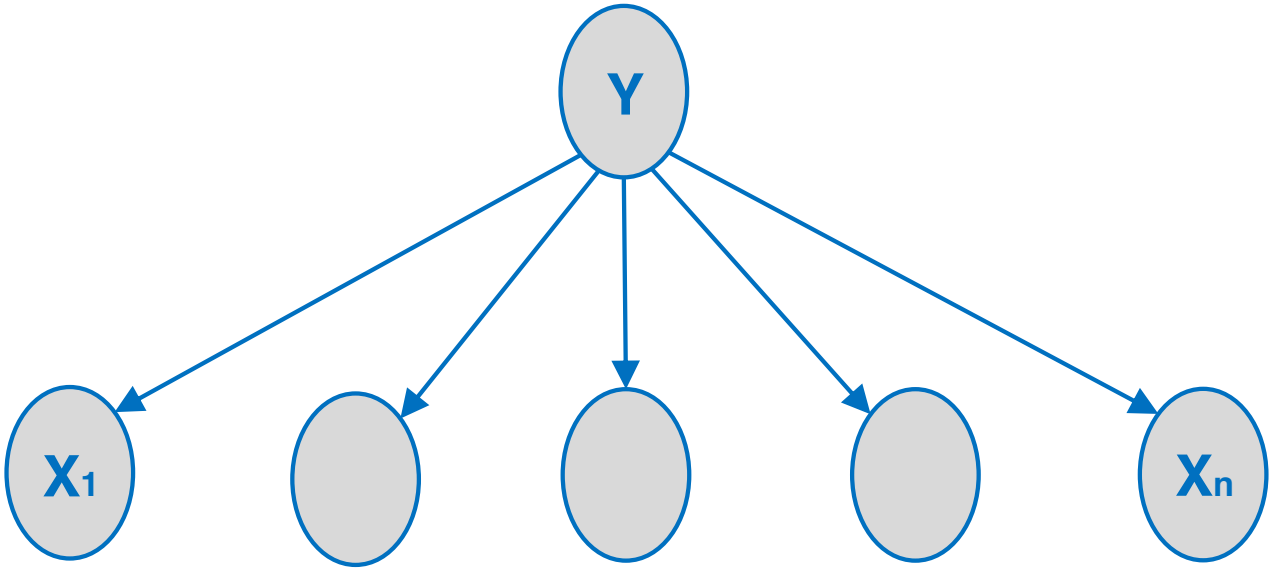
Il training set serve per stimare $P(Y)$ e $P(X_i|Y)$ per $i=1, \ldots, n$.

Nel caso si sia interessati solo alla classe maggiormente probabile basta computare

$$\arg\max_{y_k} P(Y = y_k \mid X_1,\ldots,X_n)$$

$$\arg\max_{y_k} P(Y = y_k)\cdot \prod_{i=1}^{n} P(X_i \mid Y = y_k)$$

Naïve Bayes



Naïve Bayes per attributi discreti

Gli *n* attributi sono discreti, ognuno di loro può assumere *J* diversi valori, mentre la variabile di classe Y può assumere *K* valori. Dobbiamo stimare due insiemi di parametri

$$\theta_{ijk} \equiv P(X_i = x_{ij} \,/\, Y = y_k) \qquad i = 1,\ldots,n;\ j = 1,\ldots,J;\ k = 1,\ldots,K$$

per ogni X_i, per ogni possibile valore che esso può assumere e per ogni possibile valore y_k della variabile Y. Avremo un totale di *nJK parametri* di tale tipologia da stimare.

Il secondo insieme di parametri da stimare è relativo alla probabilità a priori sulla classe Y:

$$\pi_k \equiv P(Y = y_k) \quad k = 1,\ldots,K$$

In questo caso si tratterà di stimare *K parametri*.

La stima può essere effettuata tramite schema *maximum likelihood* oppure tramite un approccio *Bayesian MAP* (impiego di priori sui parametri).

La stima *maximum likelihood per il parametro* θ_{ijk}, avendo a disposizione un training set D, è:

$$\hat{\theta}_{ijk} \equiv \hat{P}(X_i = x_{ij} \mid Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\}}{\#D\{Y = y_k\}}$$

dove #D{**x**} è un operatore che restituisce il numero di elementi (training samples) appartenenti all'insieme D che soddisfano la condizione **x**.

Può accadere che si ottengano conteggi nulli per qualche parametro. Può accadere che alcune configurazioni del numeratore non si presentino nell'insieme dei dati di training a nostra disposizione. Per evitare tale inconveniente è pratica comune adottare uno *schema che ammorbidisca la stima*

$$\hat{\theta}_{ijk} \equiv \hat{P}(X_i = x_{ij} \mid Y = y_k) = \frac{\#D\{X_i = x_{ij} \wedge Y = y_k\} + \lambda}{\#D\{Y = y_k\} + \lambda \cdot J}$$

dove λ determina il grado di morbidezza applicato.

Questa espressione corrisponde ad una stima di tipo MAP per il parametro θ_{ijk} nel caso in cui si ipotizzi una *distribuzione a priori sui parametri di tipo Dirichlet*, con eguali valori dei parametri.

Nel caso in cui λ sia posto pari ad 1 si ottiene lo *smoothing secondo Laplace*.

La stima di *massima verosimiglianza per* π_k è

$$\hat{\pi}_k \equiv \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\}}{|D|}$$

In alternativa possiamo utilizzare una *stima MAP* basata sull'impiego di una *distribuzione a priori di Dirichlet associata a* π_k utilizzando la seguente espressione

$$\hat{\pi}_k \equiv \hat{P}(Y = y_k) = \frac{\#D\{Y = y_k\} + \lambda}{|D| + \lambda \cdot K}$$

Il valore associato al parametro λ determina il grado di morbidezza applicato.

Naïve Bayes per attributi continui

Nel caso in cui gli *attributi* siano *continui* dobbiamo scegliere un modo alternativo di rappresentare la probabilità $P(X_i|Y)$. Un approccio usualmente utilizzato consiste nell'assumere che per ogni valore discreto y_k che può assumere la variabile di classe Y, la distribuzione dei valori dell'attributo X_i sia *gaussiana*; con media e deviazione standard specifiche per ogni attributo X_i e per ogni valore discreto y_k.

Il processo di stima richiede la *stima della media e della deviazione standard di ognuna di tali distribuzioni gaussiane*

$$\mu_{ik} = E[X_i / Y = y_k]$$

$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 / Y = y_k]$$

Si devono *stimare 2nK parametri* di questo tipo.

La stima della priori sulla variabile di classe Y deve ovviamente sempre essere effettuata.

La *stima di massima verosimiglianza per* μ_{ik} è

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta\left(Y^j = y_k\right)} \sum_j X_i^j \cdot \delta\left(Y^j = y_k\right)$$

dove con *j* indichiamo il *j-mo* esempio di apprendimento, e $\delta(Y^i=y_k)$ è una funzione indicatrice che assume valore 1 nel caso in cui $Y=y_k$ mentre assume valore nullo in caso contrario. Lo scopo della funzione indicatrice δ è di selezionare solo quegli esempi del training set per i quali $Y=y_k$. La *stima di massima verosimiglianza per* σ^2_{ik} è

$$\hat{\sigma}^2_{ik} = \frac{1}{\sum_j \delta\left(Y^j = y_k\right)} \sum_j \left(X_i^j - \hat{\mu}_{ik}\right)^2 \cdot \delta\left(Y^j = y_k\right)$$

Stimatore distorto, per tale ragione viene di norma preferito il seguente stimatore:

$$\hat{\sigma}^2_{ik} = \frac{1}{\sum_j \delta\left(Y^j = y_k\right) - 1} \sum_j \left(X_i^j - \hat{\mu}_{ik}\right)^2 \cdot \delta\left(Y^j = y_k\right)$$

Hidden Naïve Bayes, Tree Augmented Naïve Bayes e AODE

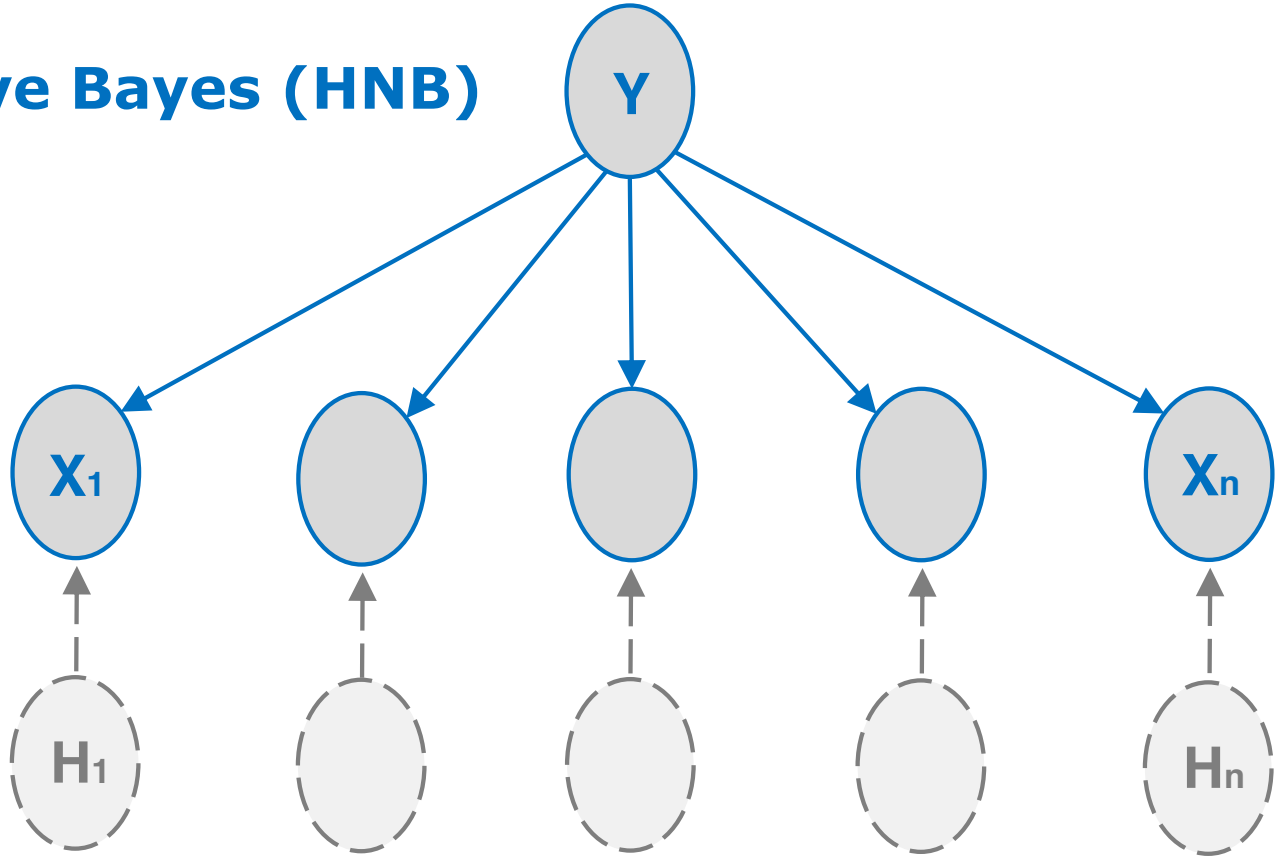
Il Naive Bayes (**NB**) si basa sull'ipotesi molto forte dell'indipendenza condizionale degli attributi (variabili esplicative) data la classe. Nonostante questa ipotesi sia molto restrittiva e di rado soddisfatta il NB garantisce di norma prestazioni competitive con diversi modelli di classificazione.

Esistono però situazioni estreme nelle quali modelli che si basano su ipotesi meno restrittive possono garantire prestazioni migliori come ad esempio:

- *Hidden Naive Bayes*
- *Tree Augmented Naive Bayes*
- *Averaged one-dependence estimators*

Questi modelli rilassano in parte l'ipotesi restrittiva sulla quale si basa il NB e cercano di implementare un trade-off "*ottimale*" tra complessità del modello e prestazioni assicurate dal classificatore quando utilizzato per processare istanze non appartenenti al dataset disponibile per condurre la fase di apprendimento.

Hidden Naïve Bayes (HNB)



$$P(Y = y_k / X_1,\ldots,X_n) = \frac{P(Y = y_k)\cdot\prod_{i=1}^{n} P(X_i / H_i, Y = y_k)}{\sum_s P(Y = y_s)\cdot\prod_{i=1}^{n} P(X_i / H_i, Y = y_s)} = \frac{P(Y = y_k)\cdot\prod_{i=1}^{n}\sum_{j=1,j\neq i}^{n} w_{ij} P(X_i / X_j, Y = y_k)}{\sum_s P(Y = y_s)\cdot\prod_{i=1}^{n}\sum_{j=1,j\neq i}^{n} w_{ij} P(X_i / X_j, Y = y_s)}$$

Hidden Naïve Bayes (HNB)

Il termine w_{ij} fa in modo che il *genitore nascosto* H_i dell'attributo X_i sia una mistura delle influenze pesate da parte di tutti i restanti attributi X_j, $j \neq i$.

In particolare avremo

$$P(X_i / H_i, Y) = \sum_{j=1, j\neq i}^{n} w_{ij} P(X_i / X_j, Y)$$

dove

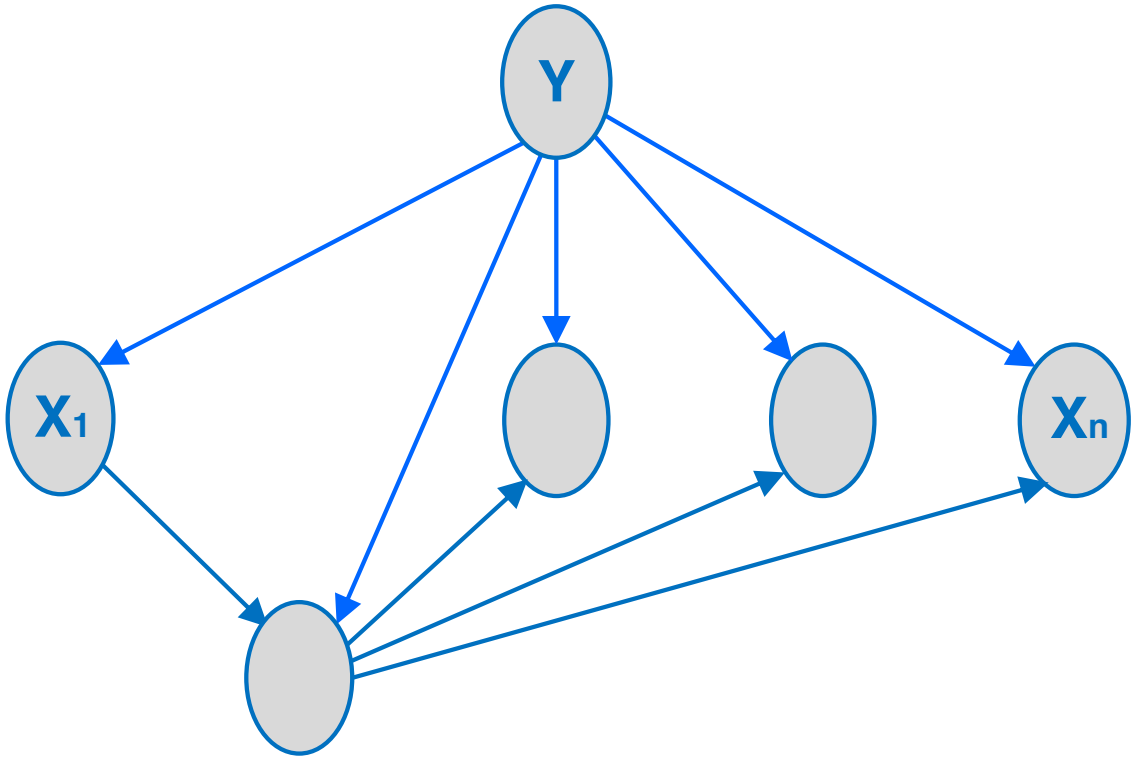
$$w_{ij} = \frac{I_P(X_i, X_j / Y)}{\sum_{j=1, j\neq i}^{n} I_P(X_i, X_j / Y)}$$

ed inoltre per ogni attributo X_i

$$\sum_{j=1, j\neq i}^{n} w_{ij} = 1$$

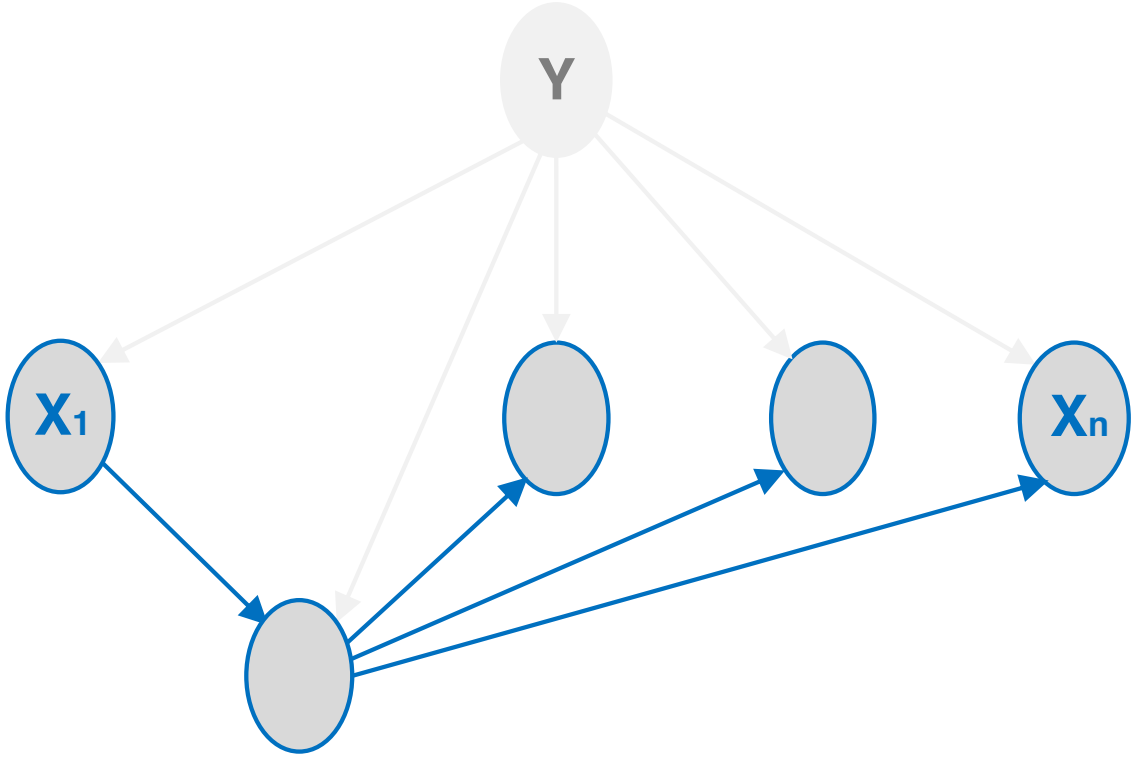
Tree Augmented Naïve Bayes (TANB)

Ammette che ogni attributo X_i abbia al più un altro nodo come genitore oltre al nodo classe. In questo modo è possibile tenere in considerazione le associazioni esistenti tra coppie di attributi, rimuovendo l'ipotesi di indipendenza condizionale associata al Naive Bayes.



Tree Augmented Naïve Bayes (TANB)

Ammette che ogni attributo X_i abbia al più un altro nodo come genitore oltre al nodo classe. In questo modo è possibile tenere in considerazione le associazioni esistenti tra coppie di attributi, rimuovendo l'ipotesi di indipendenza condizionale associata al Naive Bayes.



Tree Augmented Naïve Bayes (TANB)

La computazione della classe più probabile, la fase di classificazione di nuove istanze, viene effettuata in base alla seguente probabilità a posteriori

$$P(Y = y_k \mid X_1,\ldots,X_n) = \frac{P(Y = y_k)\cdot\prod_{i=1}^{n} P(X_i \mid pa(X_i) = pa(x_i), Y = y_k)}{\sum_j P(Y = y_j)\cdot\prod_{i=1}^{n} P(X_i \mid pa(X_i) = pa(x_i), Y = y_j)}$$

L'apprendimento di un TAN viene di norma condotto tramite l'algoritmo di Chow-Liu che è una particolare istanza dell'algoritmo di costruzione di un albero di supporto a costo minimo (Minimum weight Spannig Tree) applicato ad un grafo completo (tutte le variabili esplicative connesse tra loro con peso pari all'informazione mutua tra la coppia di variabili esplicative condizionatamente alla classe).

Averaged One-Dependence Estimators (AODE)

Condivide l'obiettivo dell'HNB e del TANB anche se cerca di raggiungerlo in un modo differente.

Nello specifico la computazione della probabilità a posteriori viene effettuata come segue:

$$P(Y = y_k \mid x_1,\ldots,x_n) = \frac{\sum_{s:1\le s\le n, F(x_s)\ge r} P(Y = y_k, x_s)\cdot\prod_{j=1}^{n} P(x_j \mid x_s, Y = y_k)}{\sum_{y_l\in Y}\sum_{s:1\le s\le n, F(x_s)\ge r} P(Y = y_l, x_s)\cdot\prod_{j=1}^{n} P(x_j \mid x_s, Y = y_l)}$$

Si aggregano modelli sulla base della numerosità minima "*r*" di casi necessari affinchè essi forniscano previsioni affidabili, nella fattispecie si escludono modelli per i quali siano disponibili meno di "*r*" osservazioni per il valore x_i assunto dall' attributo X_i.

In altre parole limitiamo l'accettazione della stima di una probabilità condizionata che se non verificata porta ad utilizzare come modello di classificazione, per quella particolare istanza, il modello NB.