SUPSI

Data Science

Introduzione al machine learning

Classificazione

Alessandro Giusti
IDSIA – SUPSI, Galleria 1, Manno
alessandro.giusti@supsi.ch

Parte del materiale tratto da:
F. Stella, "Business Intelligence", corso di Laurea in Informatica, Università degli studi di Milano - Bicocca
L. Azzimonti. Slide per il corso di "Business Intelligence". Corso di Laurea in Ing. Gestionale. SUPSI
E. Keogh. Introduction to Machine Learning.
G. Corani

Supervised learning

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

Supervised learning

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.

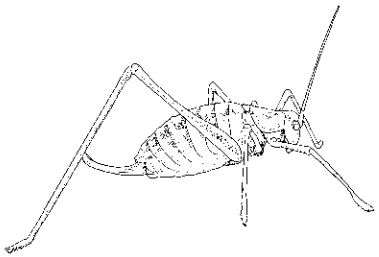
Il Supervised Learning utilizza le variabili esplicative (continue, categoriche ordinali o nominali) per risolvere problemi di

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

La Classificazione

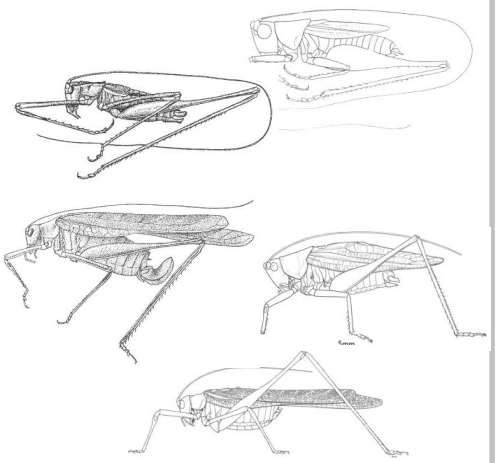
(definizione informale)

E’ data una collezione di dati annotati. In questo caso, 5 osservazionoi di **Katydid** e 5 di **Grasshopper.** Decidi a quale classe appartiene l’osservazione sotto.

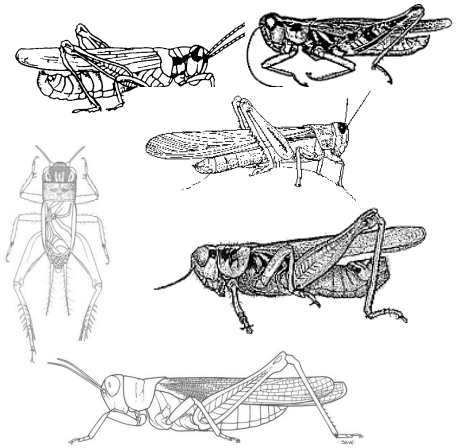


Katydid or **Grasshopper?**

Katydids



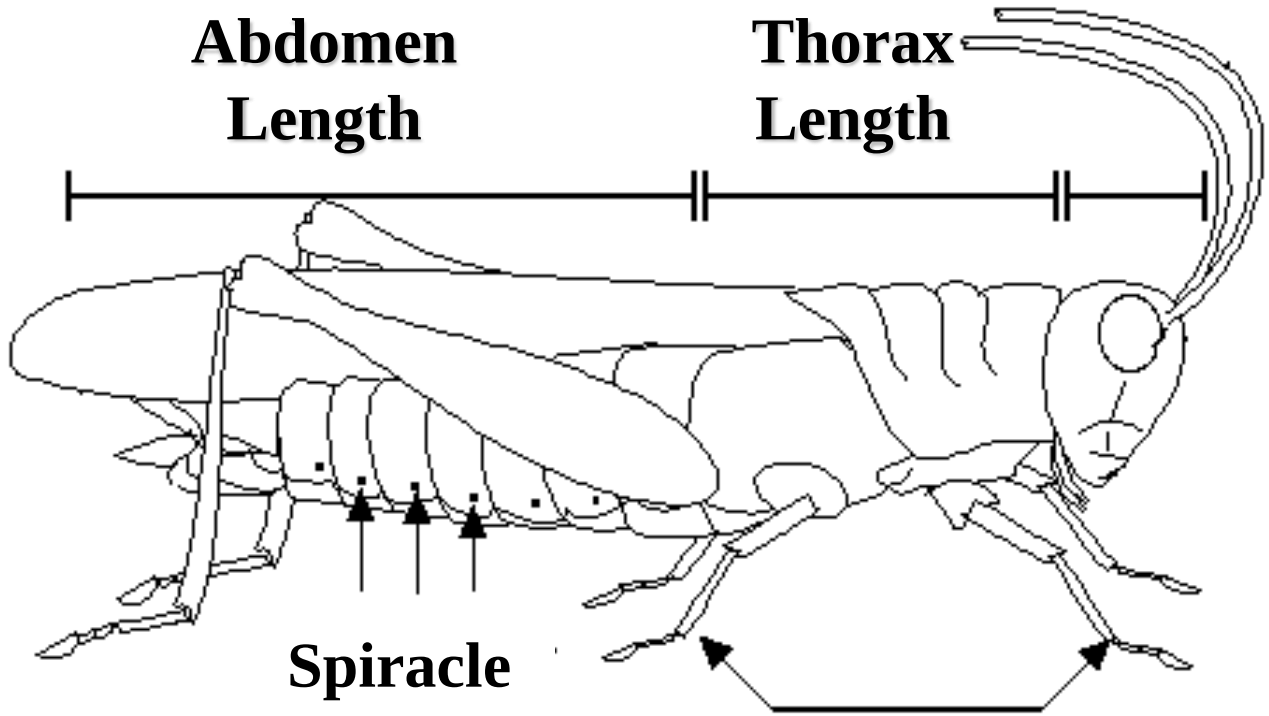
Grasshoppers



Inipendentemente dal dominio in oggetto, possiamo calcolare delle *features*

Color {Green, Brown, Gray, Other}

Has Wings?



Abbiamo un dataset di osservazioni con le features (variabili esplicative) e la classe (variabile target)

Dato questo dataset (**My_Collection**), predici la **classe** di una osservazione mai vista prima

My_Collection

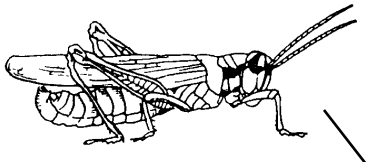
Insect ID	Abdomen Length	Antennae Length	Insect Class
1	2.7	5.5	**Grasshopper**
2	8.0	9.1	**Katydid**
3	0.9	4.7	**Grasshopper**
4	1.1	3.1	**Grasshopper**
5	5.4	8.5	**Katydid**
6	2.9	1.9	**Grasshopper**
7	6.1	6.6	**Katydid**
8	0.5	1.0	**Grasshopper**
9	8.3	6.6	**Katydid**
10	8.1	4.7	**Katydids**

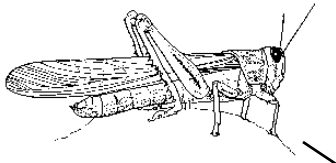
previously unseen instance =

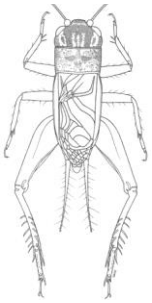
11	5.1	7.0	**???????**

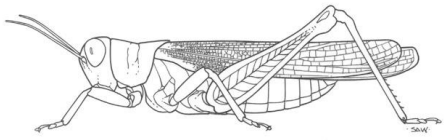
Grasshoppers

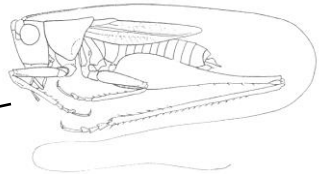
Katydids

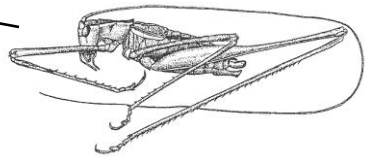


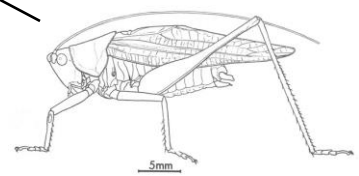


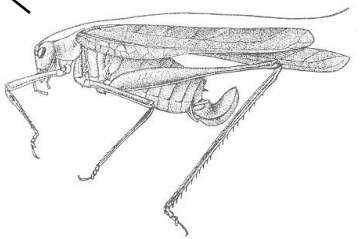


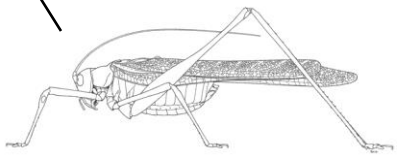


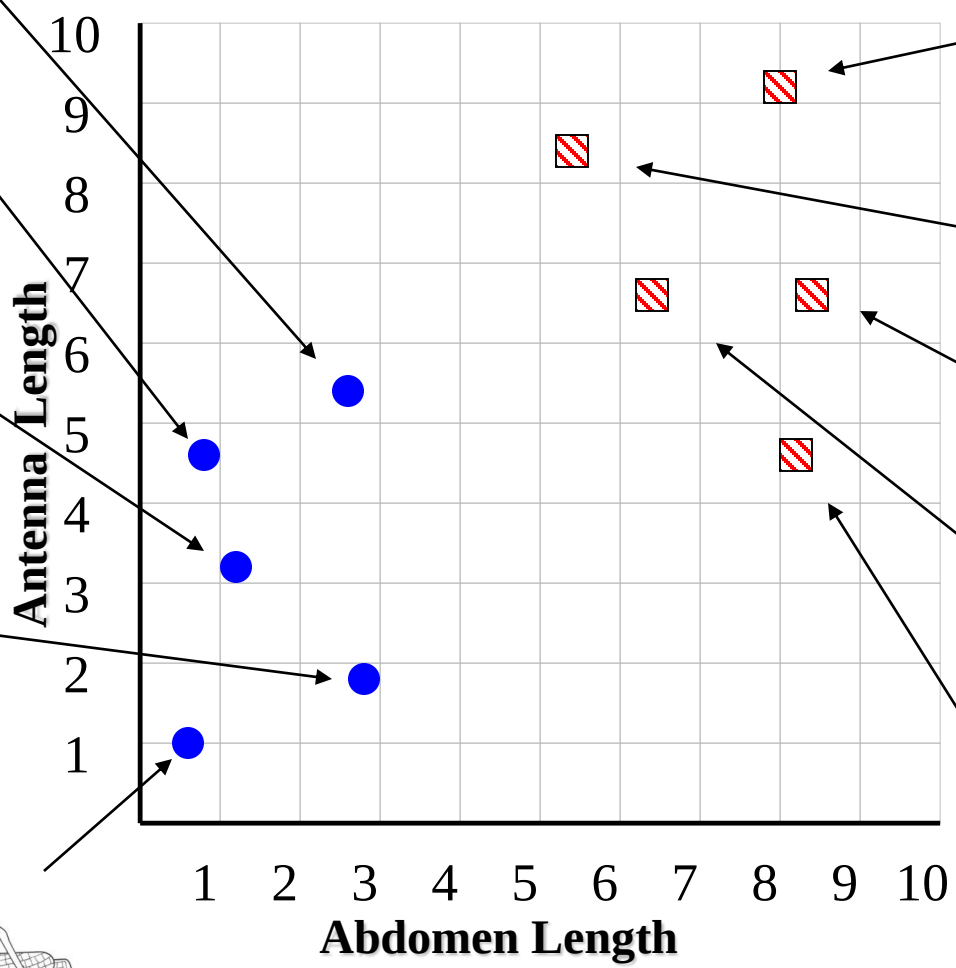




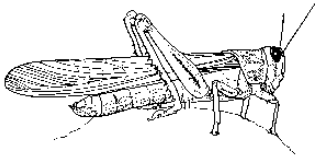




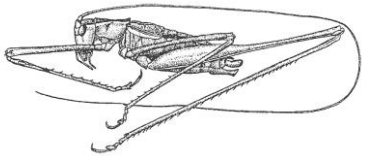




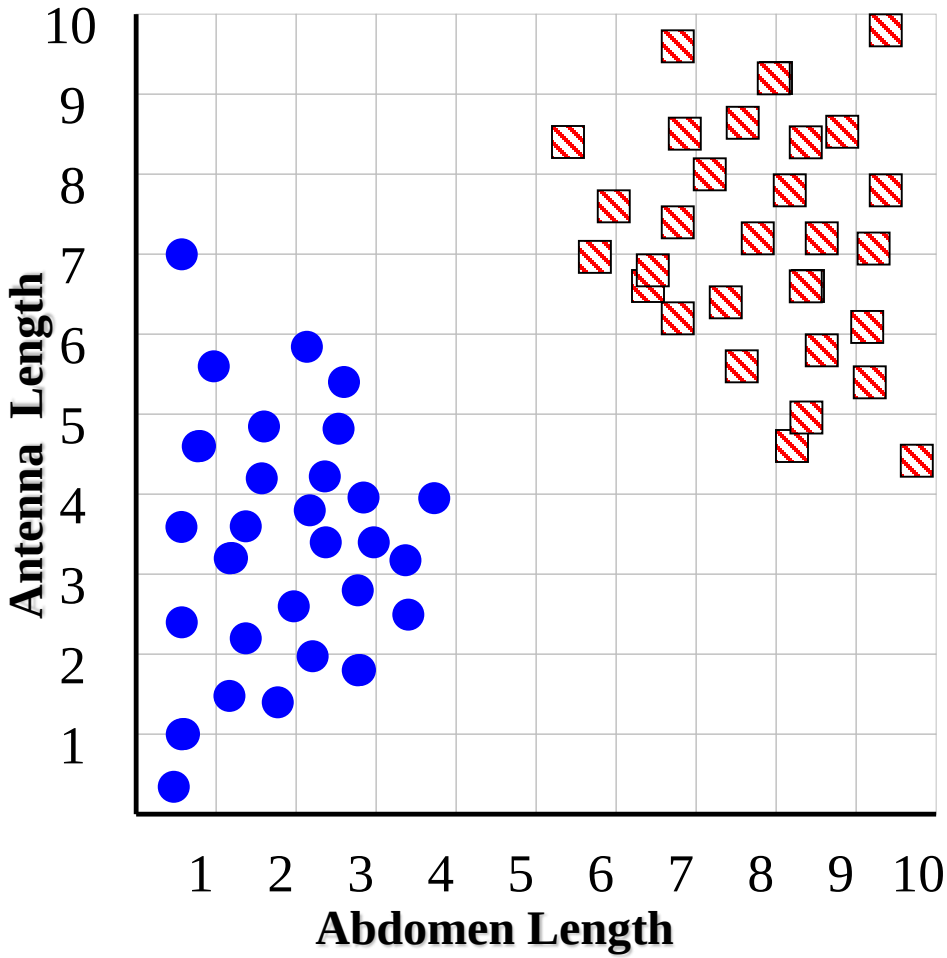
Grasshoppers



Katydids



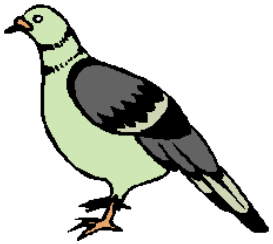
Se ho piu’ dati:



Ognuno di questi punti si puo’ chiamare:

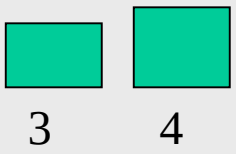
- istanza
- esempio
- record
- osservazione
- …

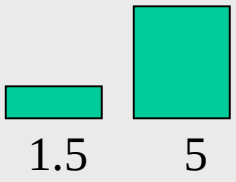
Anche i piccioni sanno risolvere i problemi di classificazione!

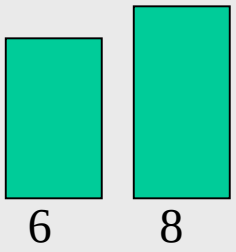


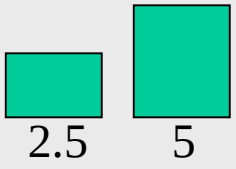
Pigeon Problem 1

Esempi classe A

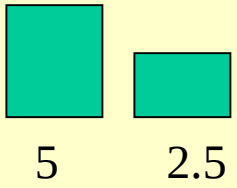


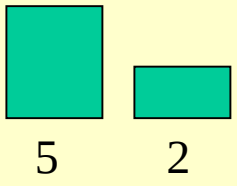


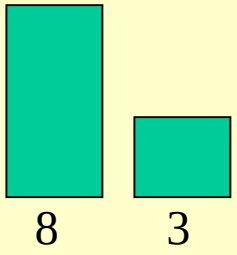


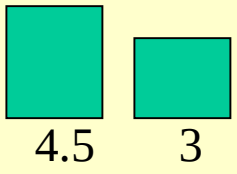


Esempi classe B



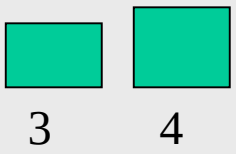


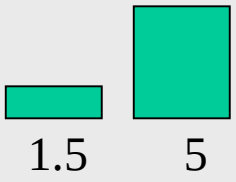


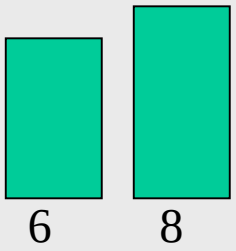


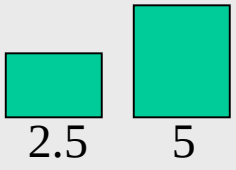
Pigeon Problem 1

Esempi classe A

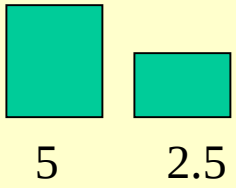


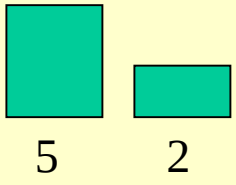


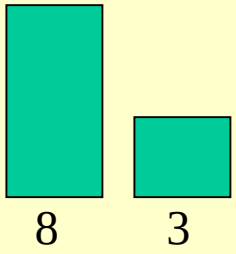


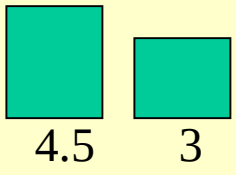


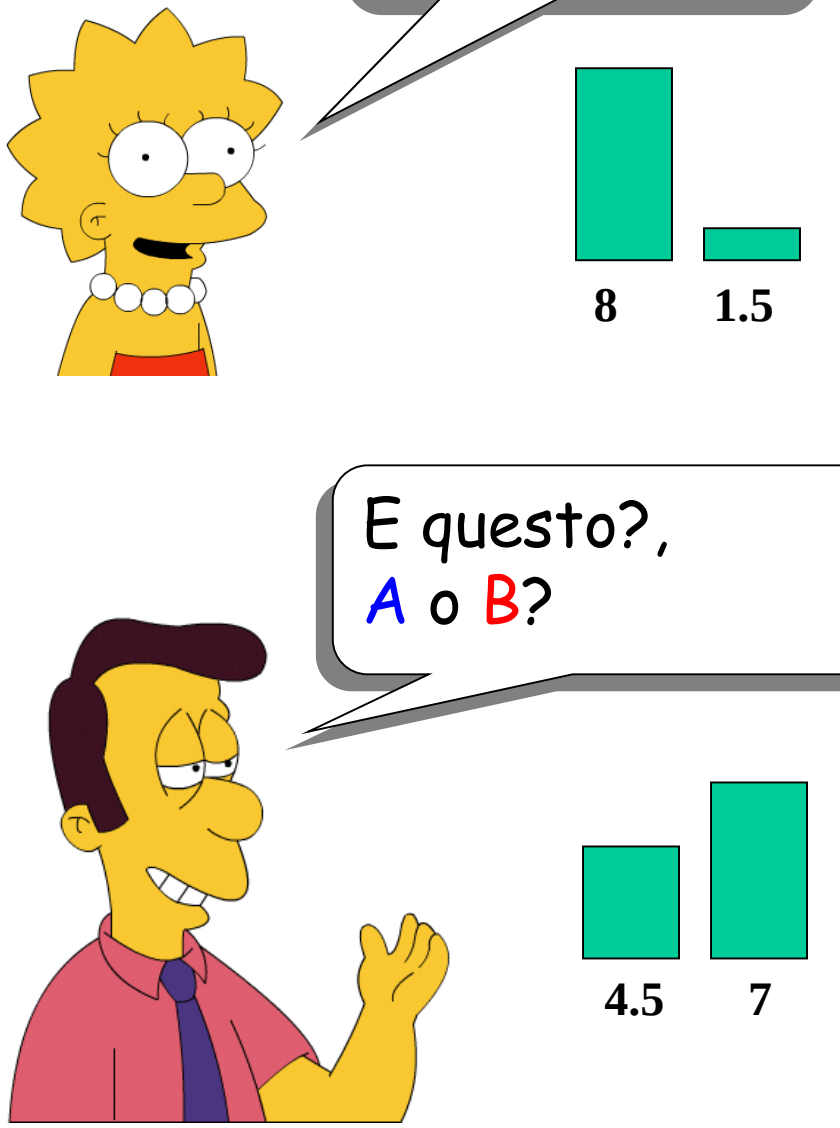
Esempi classe B





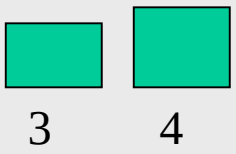


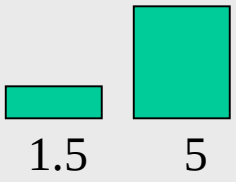


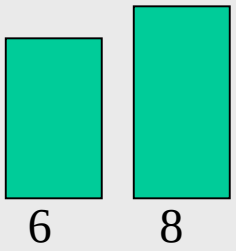


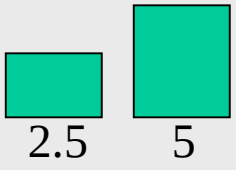
Pigeon Problem 1

Esempi classe A

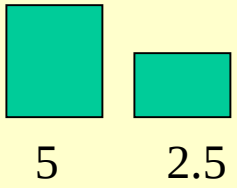


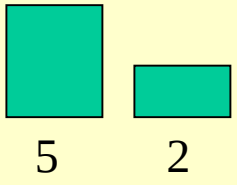


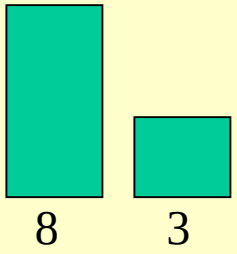


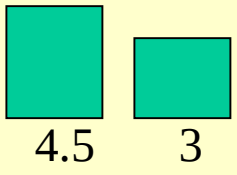


Esempi classe B

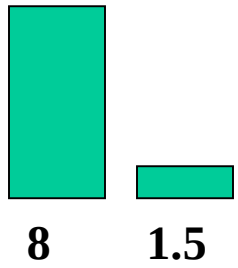








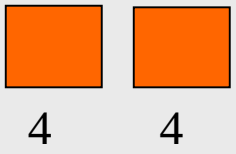
Questo e' B!

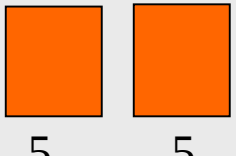


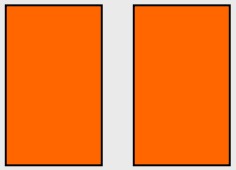
La regola: se la barra a sinistra e' piu' bassa di quella a destra, e' A, altrimenti, B.

Pigeon Problem 2

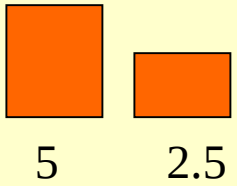
Esempi classe A

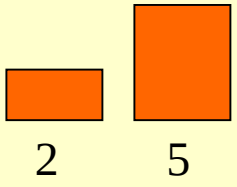


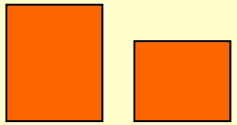


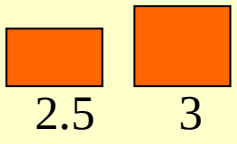


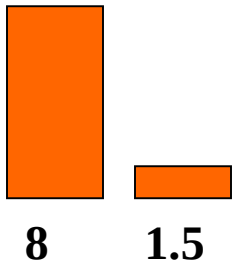
Esempi classe B

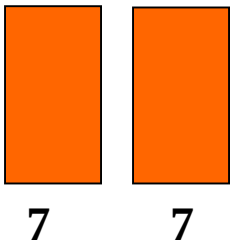






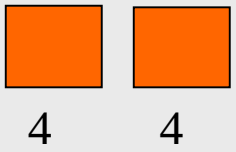


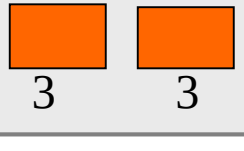




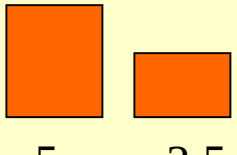
Pigeon Problem 2

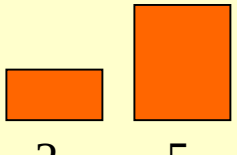
Esempi classe A

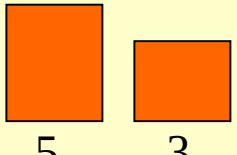


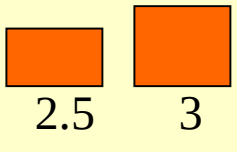


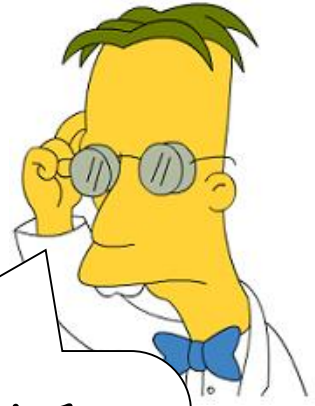
Esempi classe B





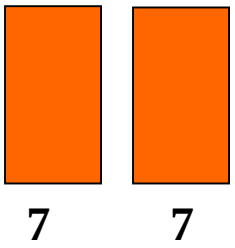






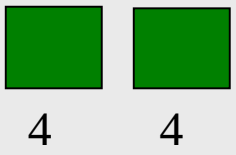
La regola e' questa: se le due barre sono uguali, e' A. Altrimenti B.

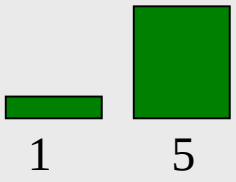
Questa e' A.

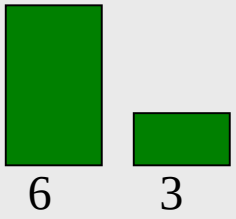


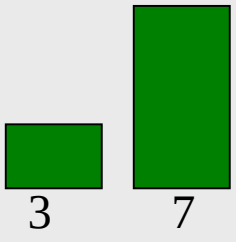
Pigeon Problem 3

Esempi classe A

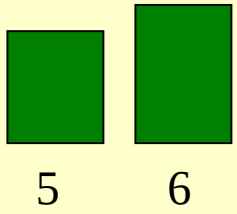


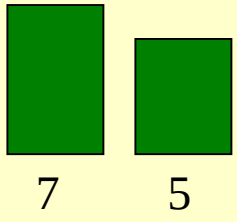


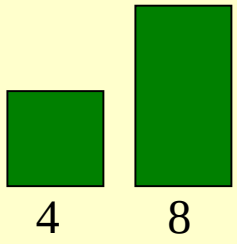


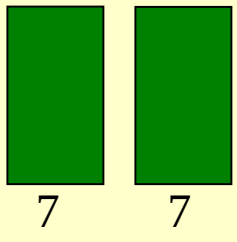


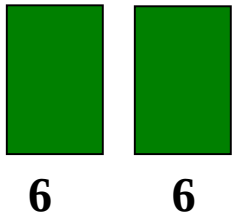
Esempi classe B







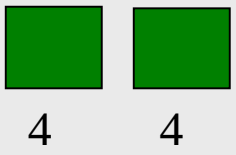


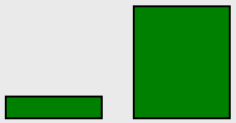


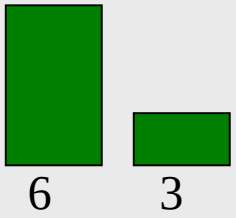
Ahi, questa e' dura!
Cos' e', A o B?

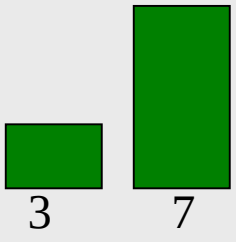
Pigeon Problem 3

Esempi classe A

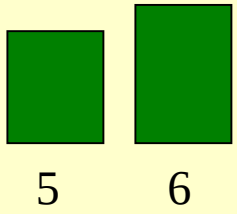


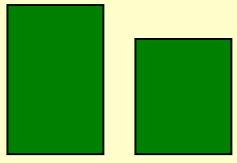


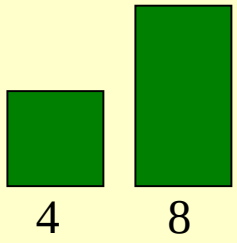


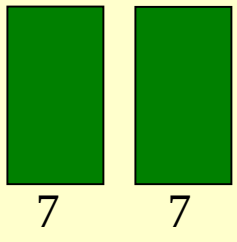


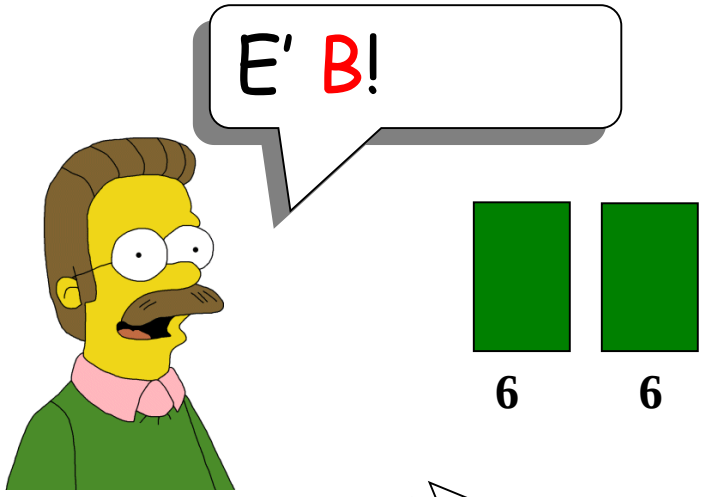
Esempi classe B







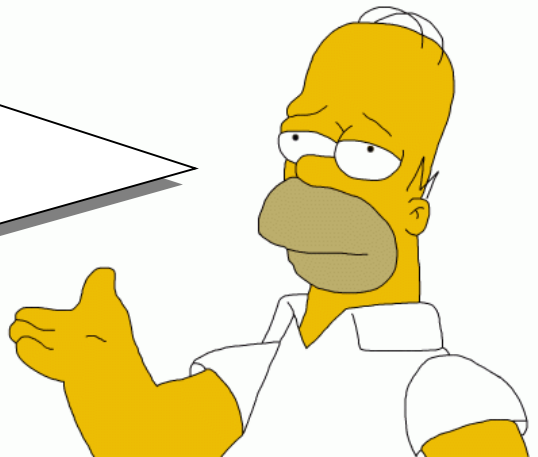




La regola e' questa: se la somma dei quadrati e' minore o uguale a 60, e' A. Altrimenti e' B.

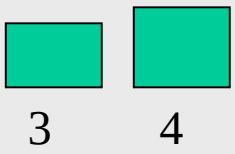
Perche' abbiamo perso tutto questo tempo con questo giochino stupido?

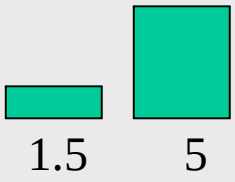
Perche' volevamo mostrare che i problemi di classificazione hanno un'interpretazione geometrica! Guarda le prossime slide!

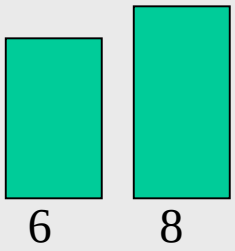


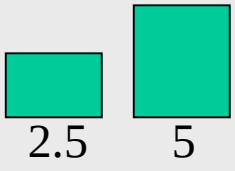
Pigeon Problem 1

Esempi classe A

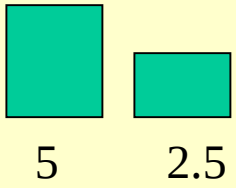


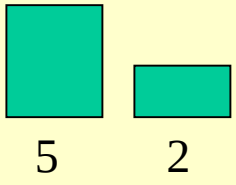


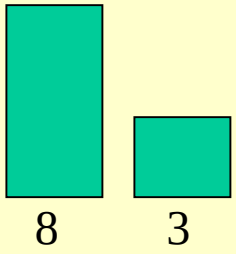


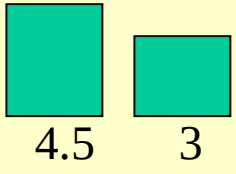


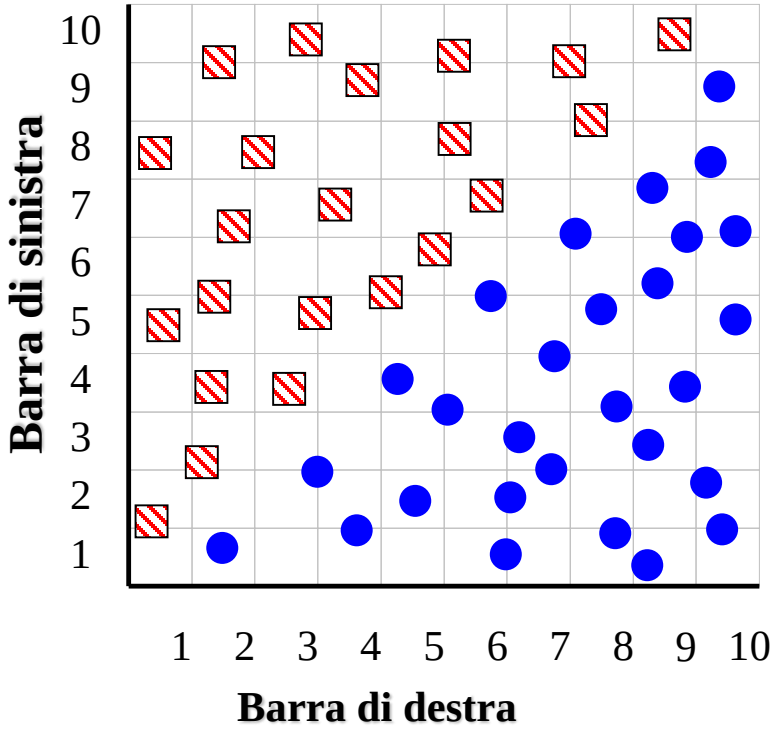
Esempi classe B







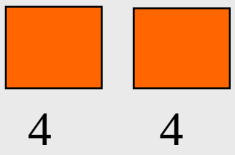


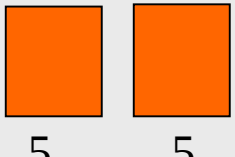


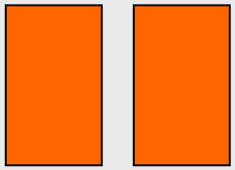
La regola era: se la riga di sinistra e' piu' bassa di quella di destra, e' A, altrimenti B.

Pigeon Problem 2

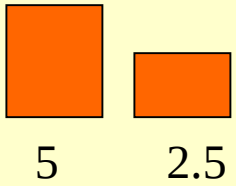
Esempi classe A

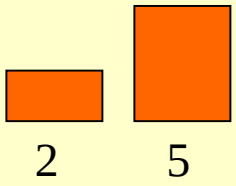


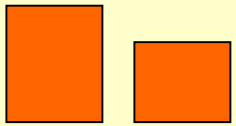


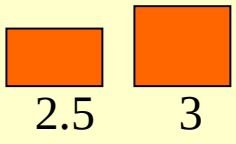


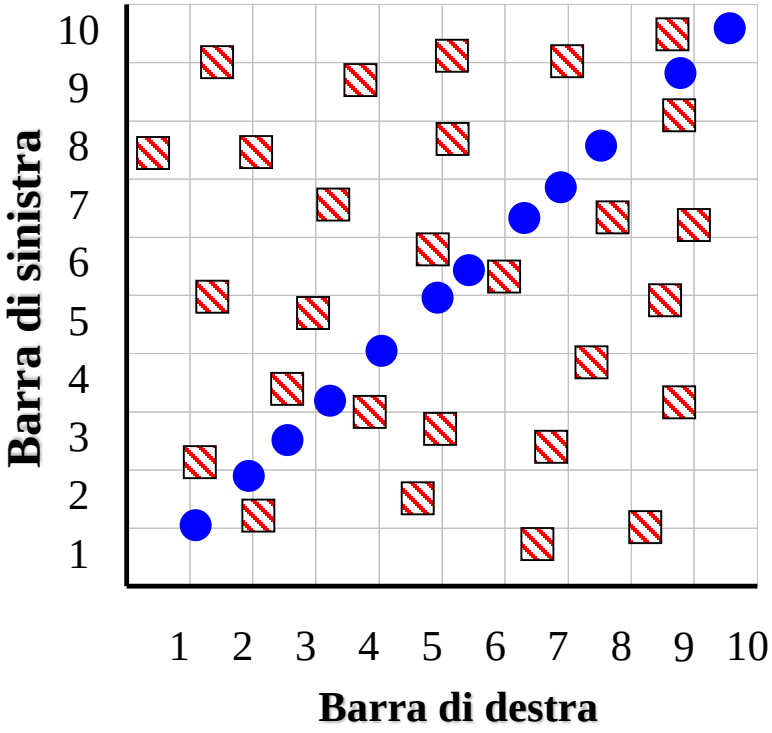
Esempi classe B

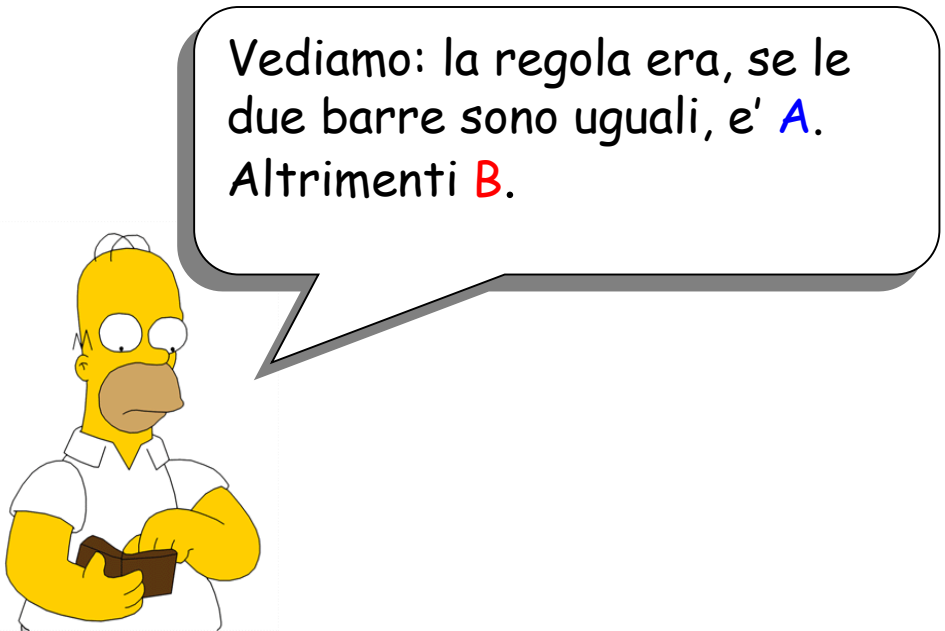






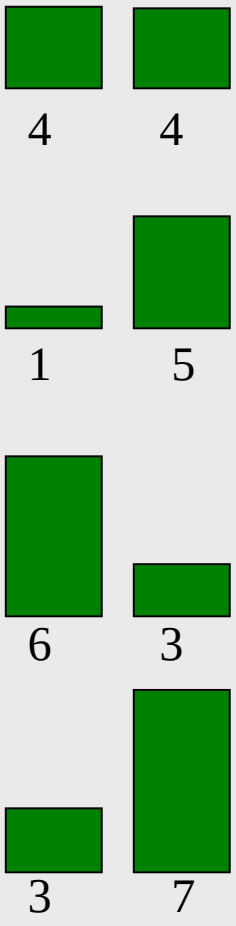




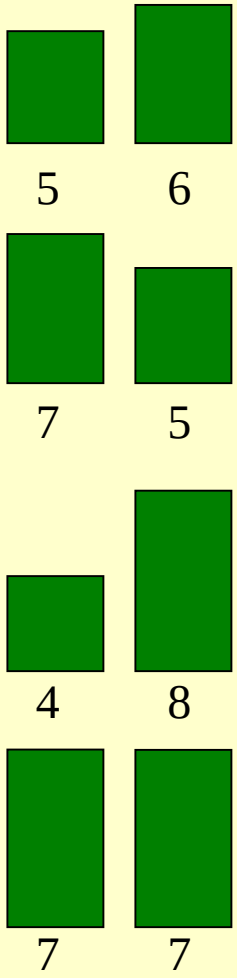


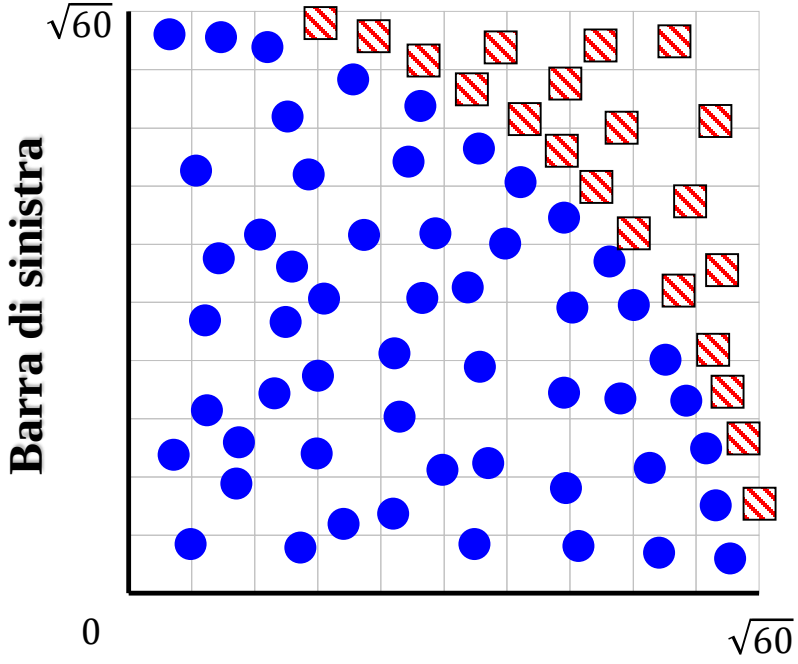
Pigeon Problem 3

Esempi classe A



Esempi classe B

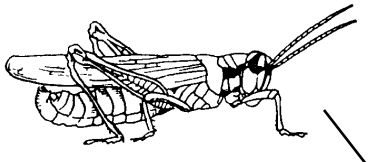


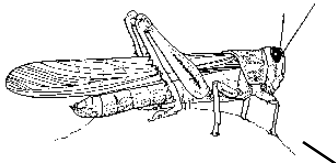


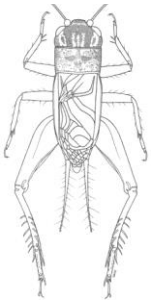
La regola:
se la somma dei quadrati e' minore o uguale a 60, e' A.
Altrimenti e' B.

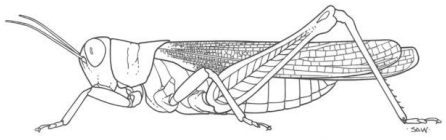
Grasshoppers

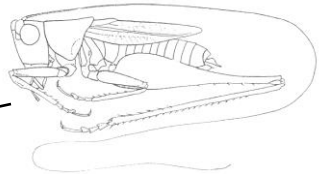
Katydids

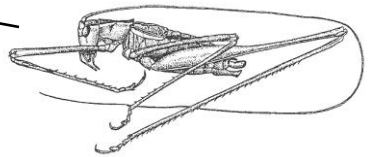


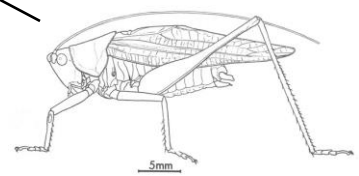


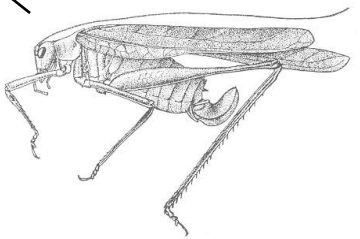


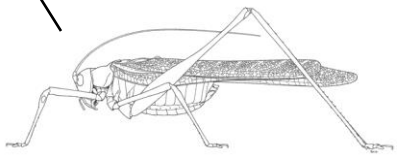


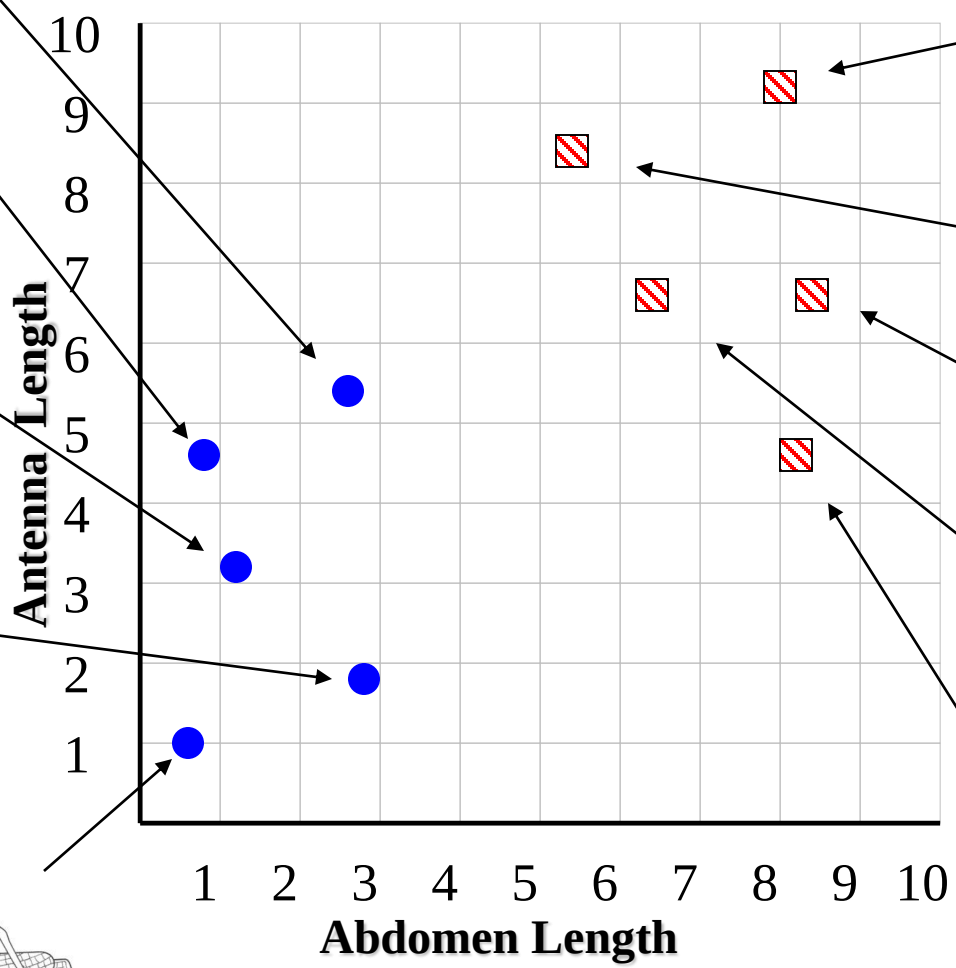




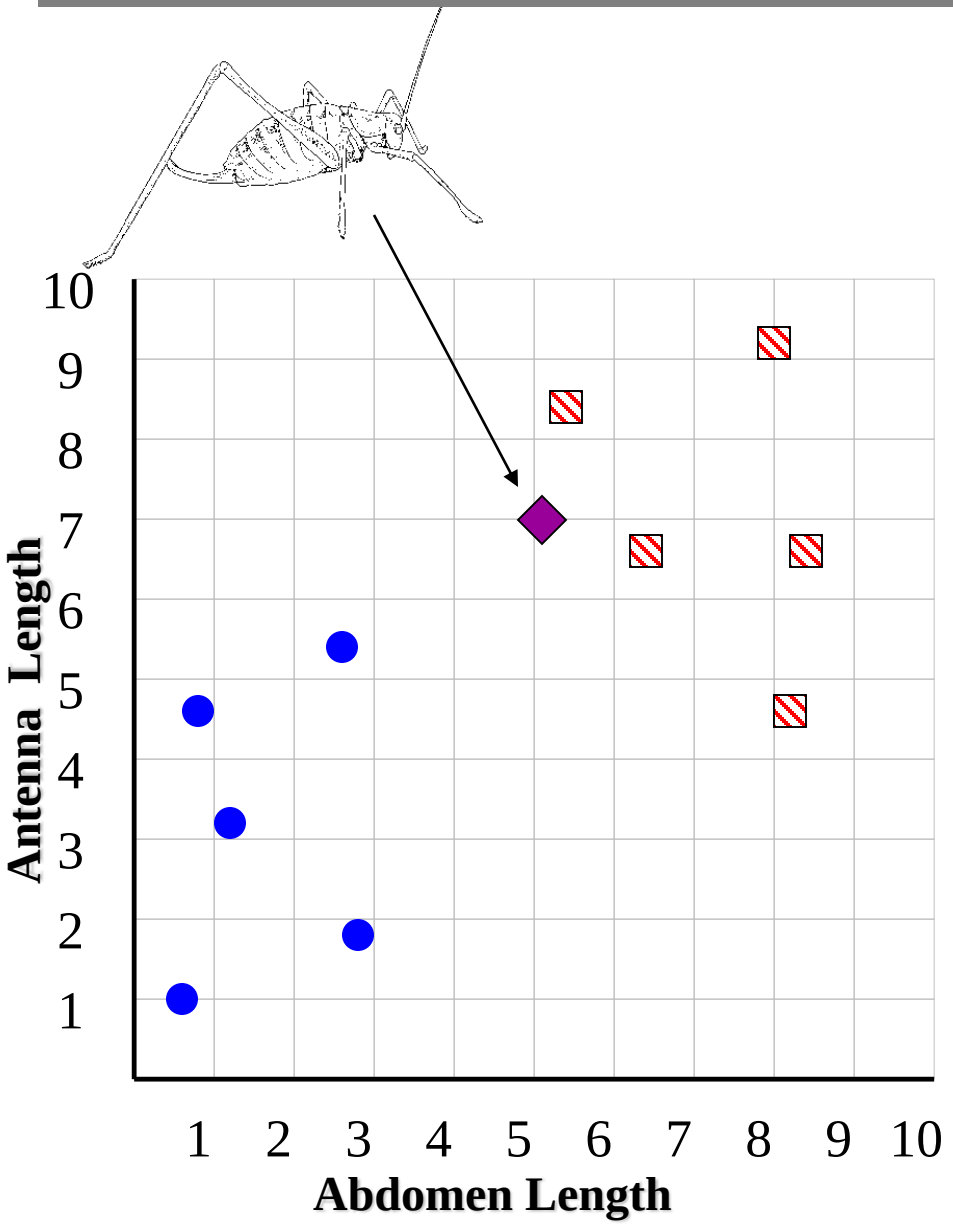








previously unseen instance =	11	5.1	7.0	???????



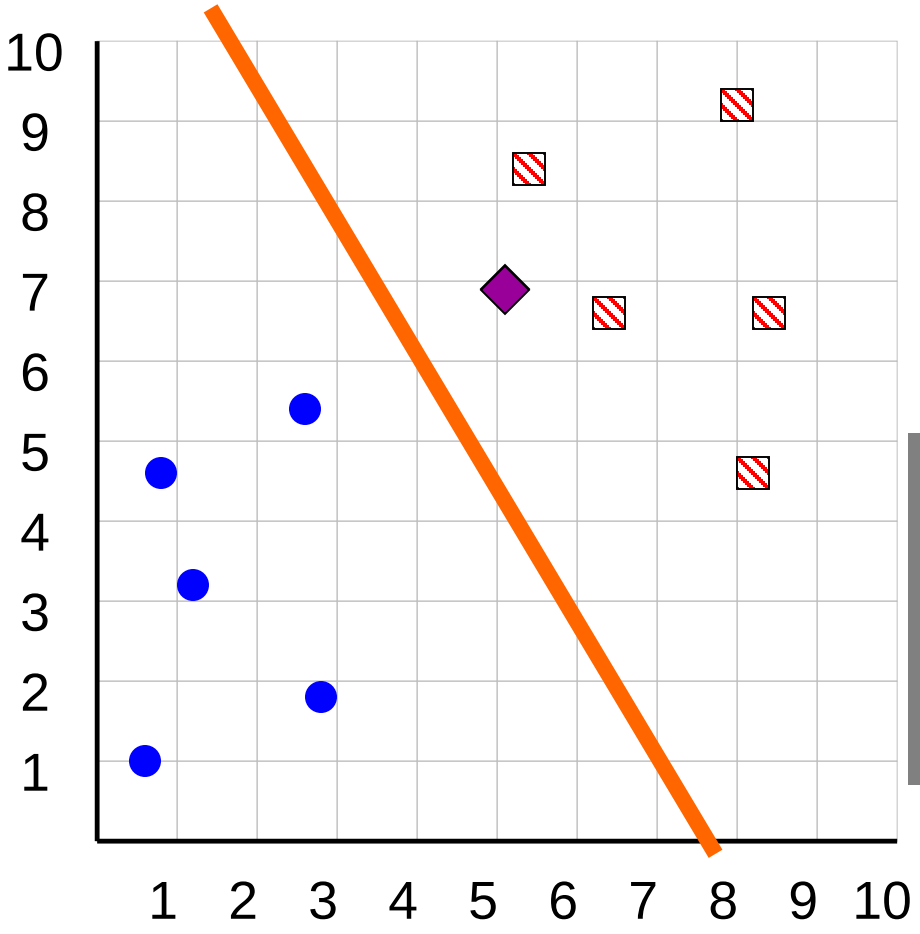
- Possiamo "proiettare" l'osservazione che non abbiamo ancora visto nello stesso spazio del dataset

- Ora possiamo dimenticare i dettagli specifici del problema, e ragionare solo con i punti nello spazio!

▨ **Katydids**

● **Grasshoppers**

Classificatore lineare semplice

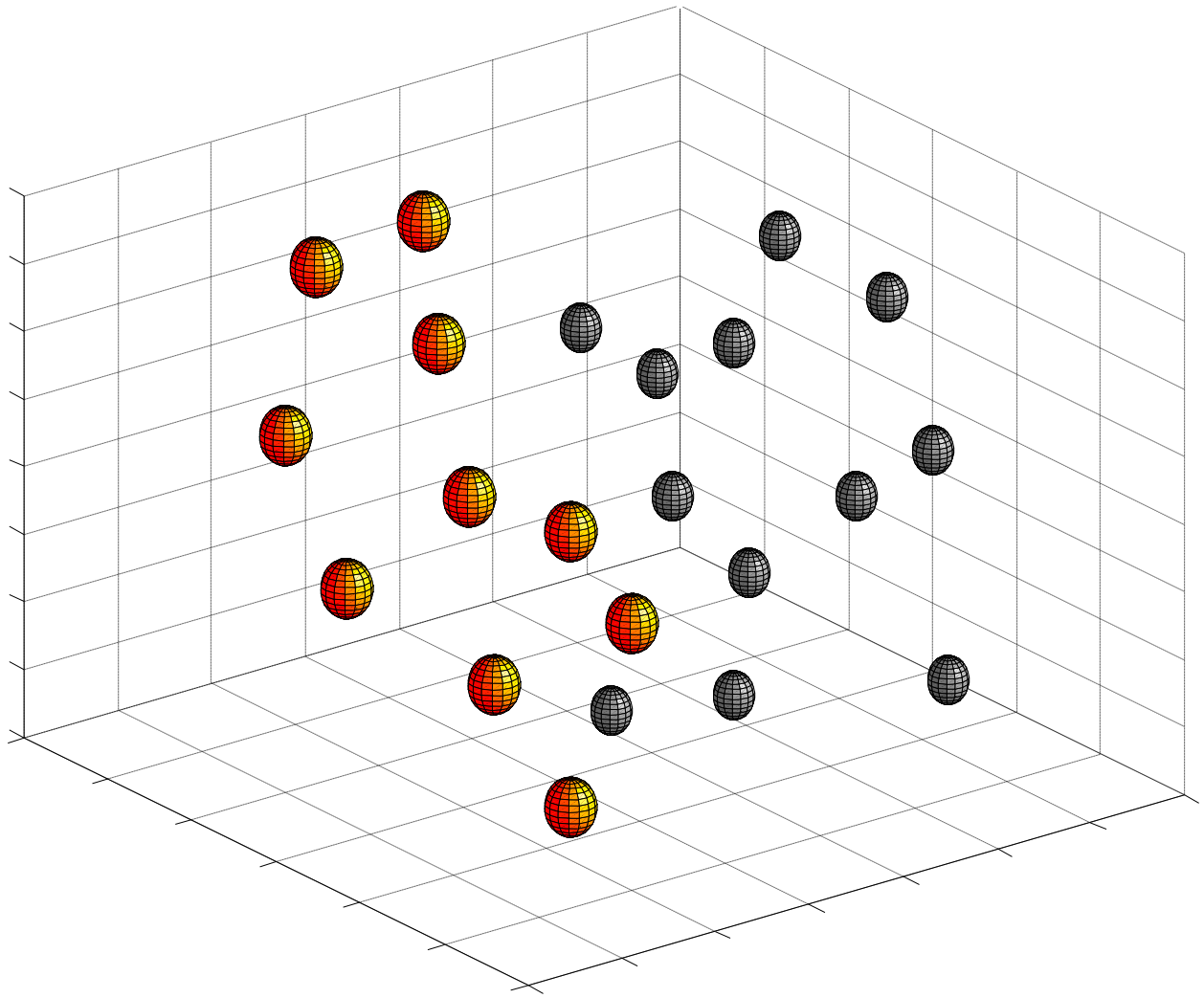


R.A. Fisher
1890-1962

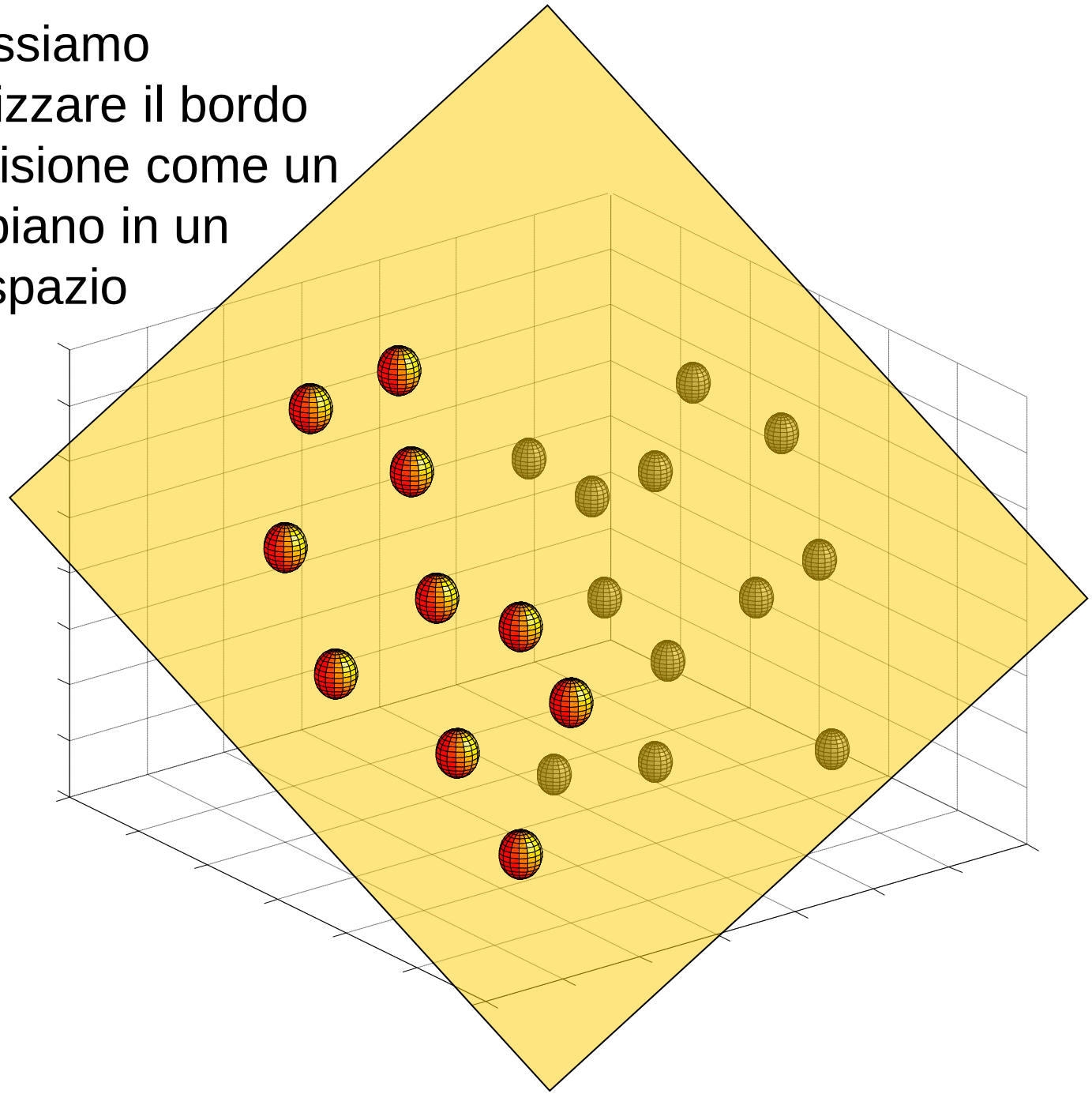
If nuova osservazione sopra la linea
then
class is **Katydid**
else
class is **Grasshopper**

Katydids
Grasshoppers

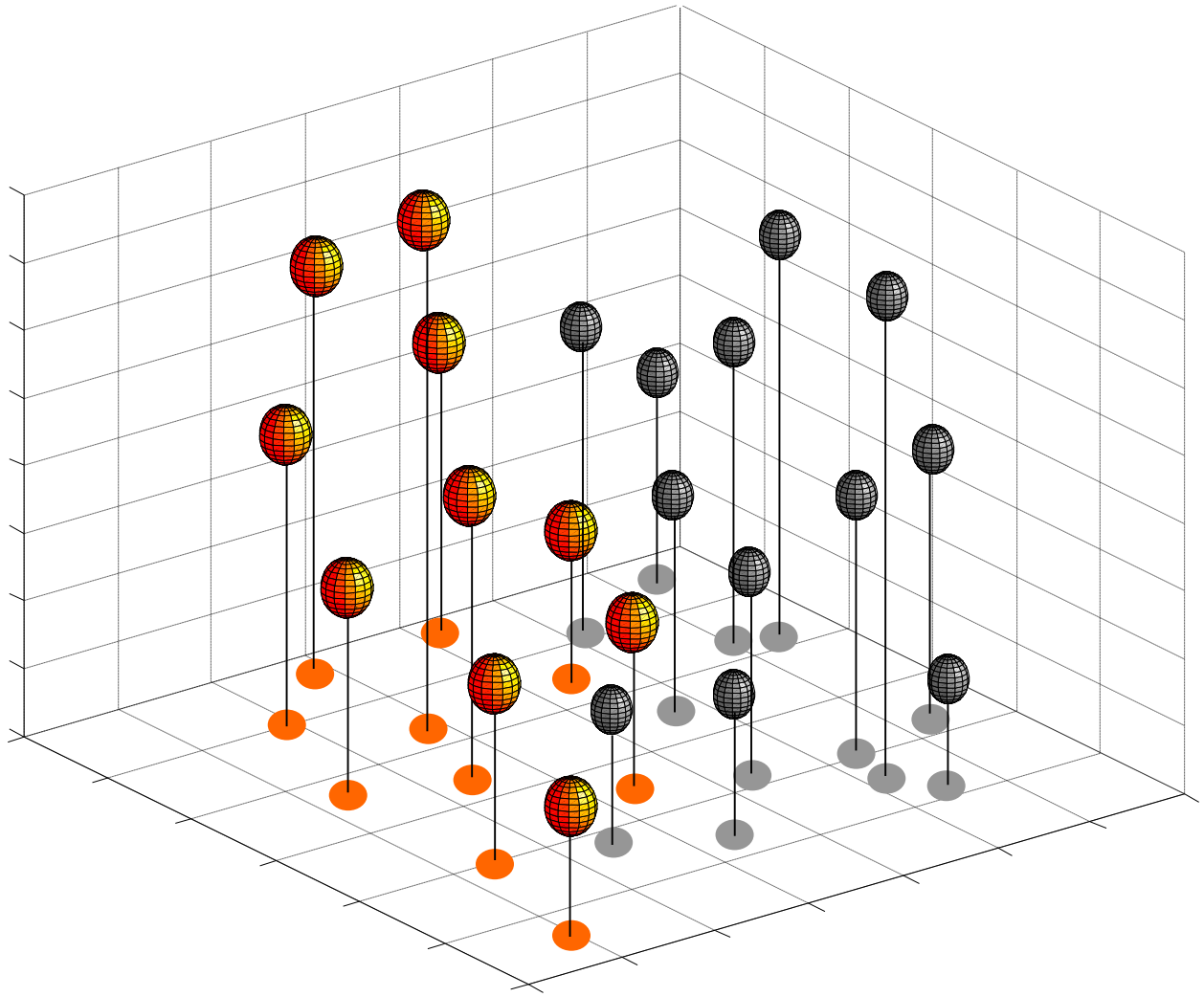
E se ci sono piu’ di 2 variabili esplicative (features)?

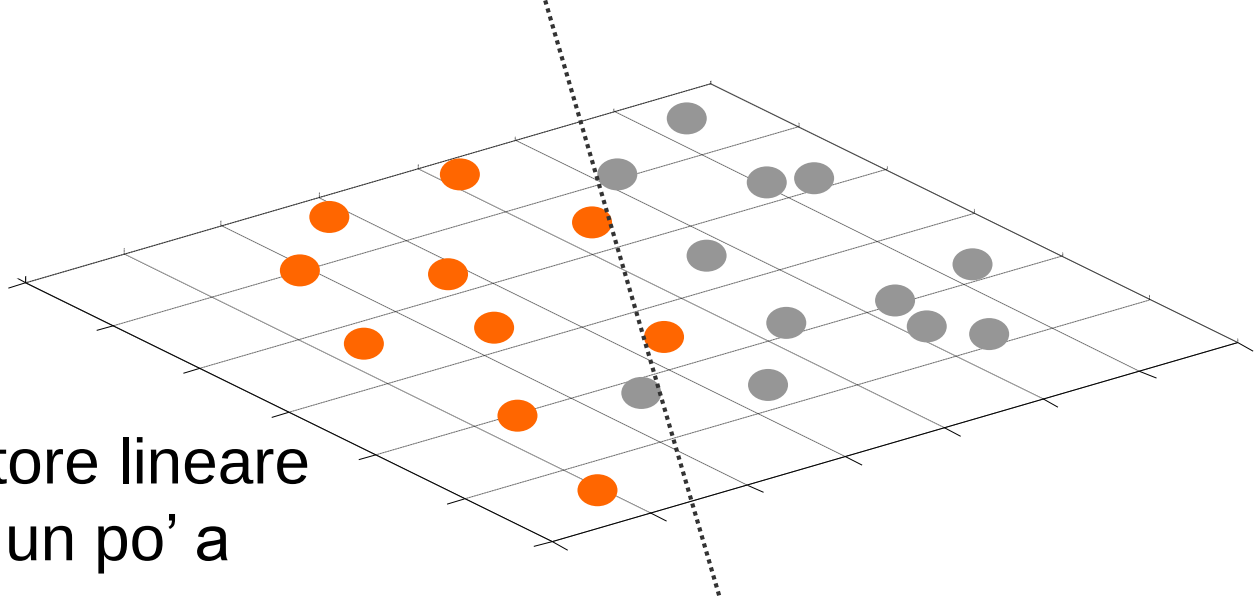


… Possiamo visualizzare il bordo di decisione come un (iper)piano in un (iper)spazio



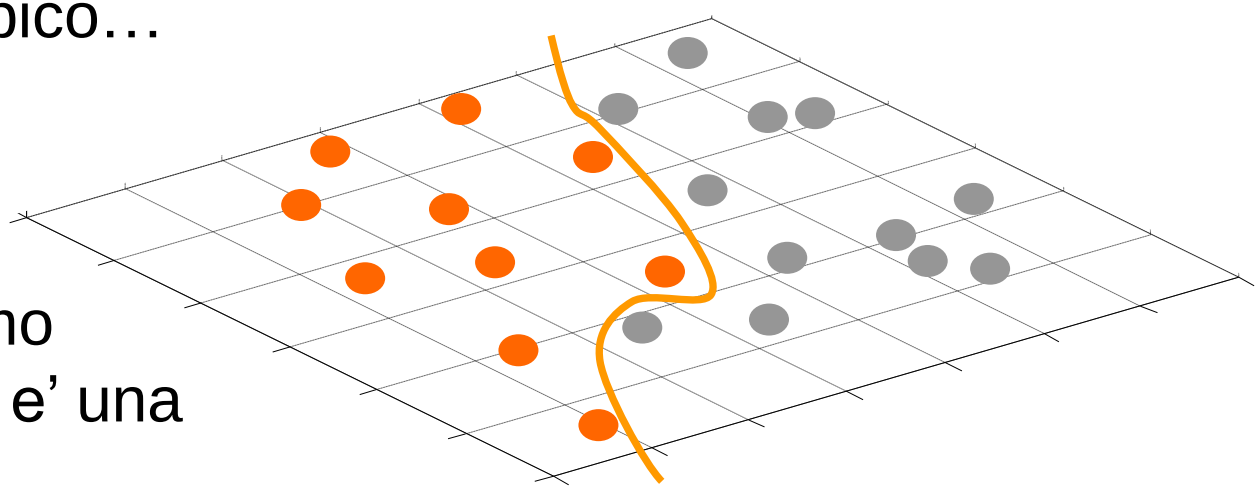
Cosa succederebbe in questo caso se non avessimo la terza dimensione?





Ahi! Il classificatore lineare semplice lascia un po' a desiderare!

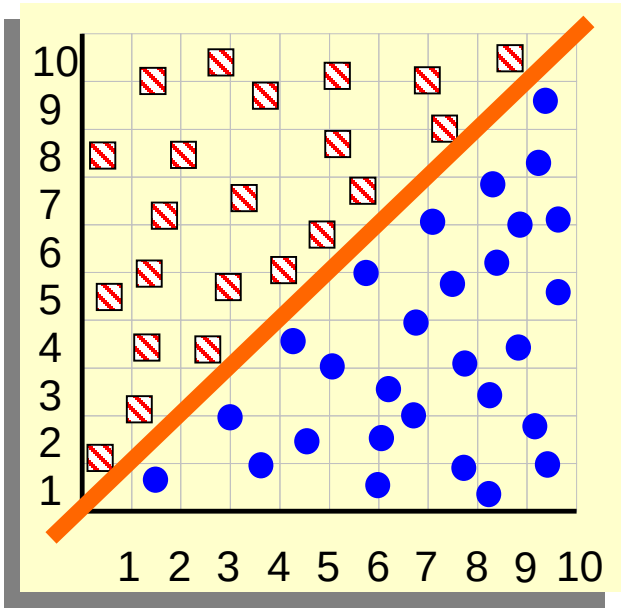
Potremmo piuttosto usare un bordo di decisione quadratico o cubico…

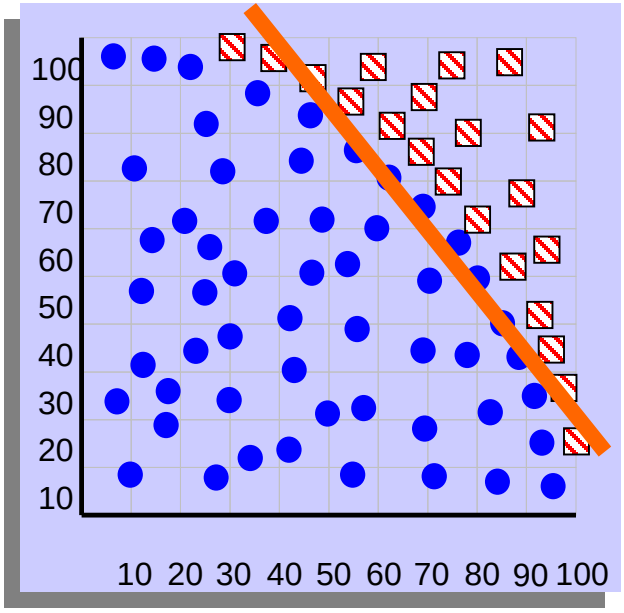


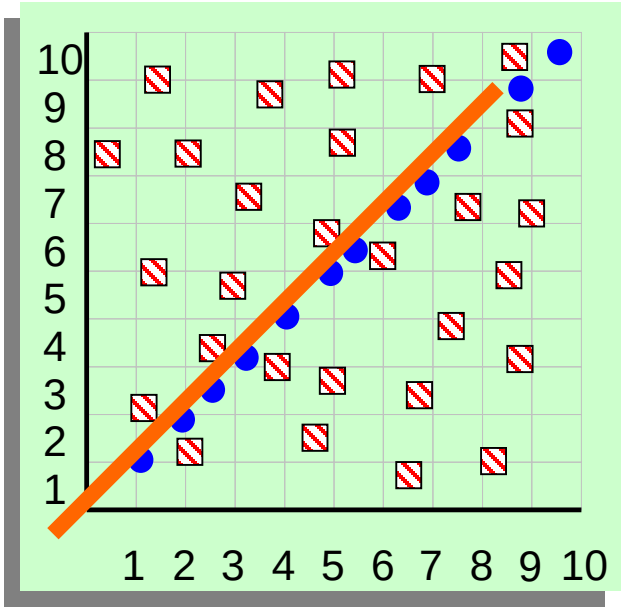
Tuttavia, vedremo che spesso non e' una grande idea…

Quali “Pigeon Problems” si risolvono bene con un classificatore lineare semplice?

1) Perfetto
2) Inutile!
3) Mica male







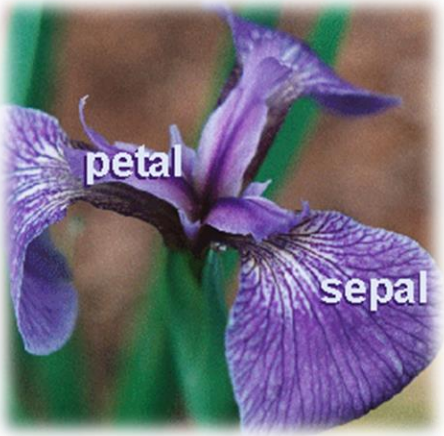
I problemi che si risolvono con un classificatore lineare si dicono **linearmente separabili**

Un altro problema di classificazione

SETOSA

VERSICOLOR

VIRGINICA

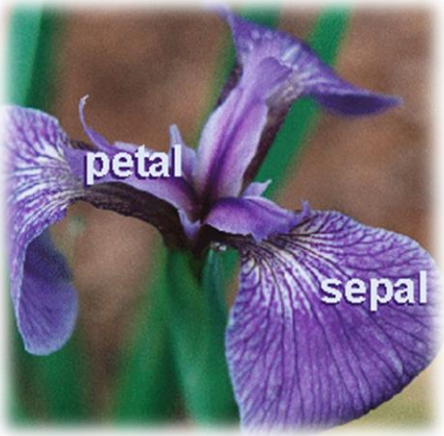


Un altro problema di classificazione

Classe: SETOSA

VERSICOLOR

VIRGINICA



Variabili esplicative

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

Un altro problema di classificazione

Classe: SETOSA

VERSICOLOR

VIRGINICA

Obiettivo:

Dato un insieme di casi o osservazioni per le quali sono noti i valori assunti dalle *variabili esplicative* o *features* e dalla *variabile target* o *classe*, vogliamo sviluppare un modello in grado di fornire previsioni circa la variabile target, per delle nuove osservazioni mai viste prima.

Classificazione: un po' di notazione

Consideriamo un dataset

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

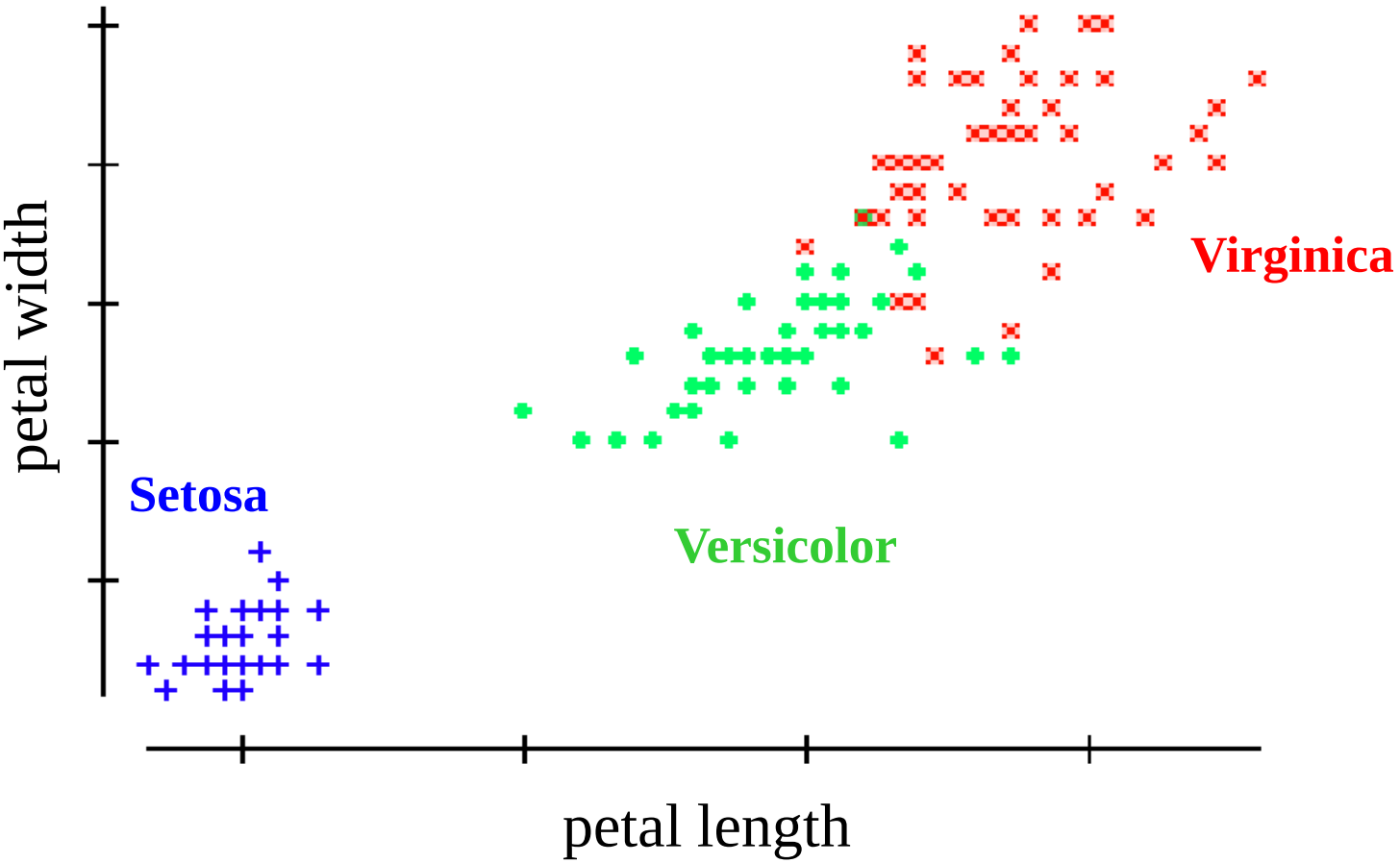
contenente m osservazioni (definite *esempi* o *istanze*) relative a:

- n variabili esplicative $\mathbf{X_j}$, j=1,…, n, che possono essere *continue*, *ordinali* o *nominali*
- una variabile target Y, definita anche *classe* o *etichetta*, che assume un numero finito di valori, ovvero ha supporto finito

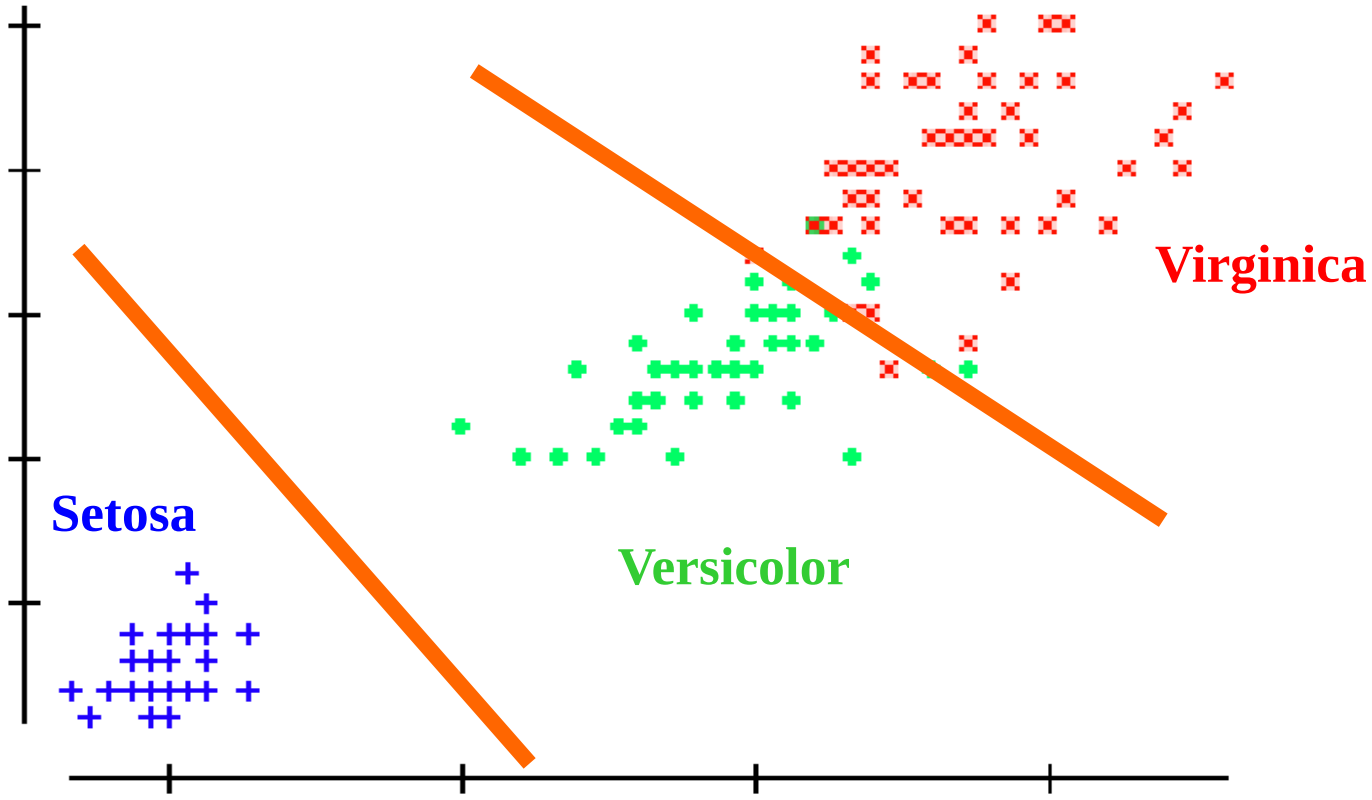
$$H=\{v_1,\ldots v_H\}.$$

Se H=2 si parla di *classificazione binaria* e le due classi possono essere indicate come {0,1} o {-1, +1}.

Se H>2 si parla di *classificazione multi-classe* (multi-categorica).


petal width
petal length
Setosa
Versicolor
Virginica

Possiamo generalizzare il classificatore lineare binario al caso multi-classe: per H classi, troviamo H-1 rette: per esempio, con la prima impariamo a distinguere **Setosa** da **Virginica**/**Versicolor**. Con la seconda, ci concentriamo su queste ultime due classi e disciminiamo **Virginica** e **Versicolor**.



If petal width < b + k * petal length **then** class = **Setosa**
Elseif petal width…

Classificazione: obiettivo

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.

Per fare ciò indicheremo con F una famiglia di funzioni f definite sullo spazio delle variabili esplicative e con valori sul supporto della variabile target, ovvero

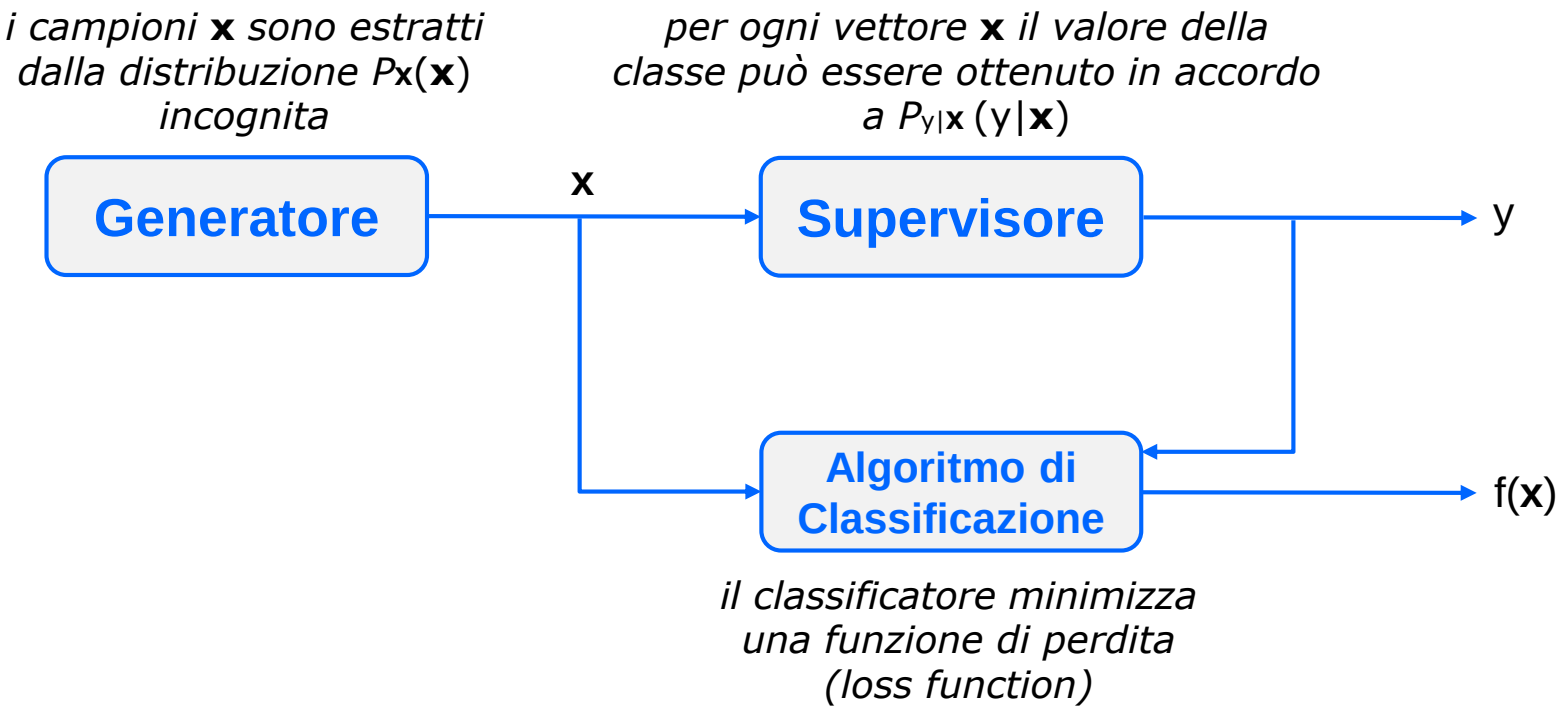
$$f: R^n \rightarrow H$$

Queste funzioni rappresentano possibili relazioni esistenti tra le variabili esplicative e la variabile di classe e vengono anche chiamate *ipotesi*.

Il *problema di classificazione* consiste nel:

- *definire di uno spazio delle ipotesi (F)*
- *progettare o scegliere 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 $\mathbf{X}_j$, j=1,…m, e la variabile di classe Y.

Ipotizziamo che le coppie $(\mathbf{x}_i, y_i)$ provengano da una distribuzione di probabilità $P_{\mathbf{x},Y}(\mathbf{x},y)$ incognita. Solitamente i modelli di classificazione non fanno assunzioni sulla forma di $P_{\mathbf{x},Y}(\mathbf{x},y)$. Le uniche assunzioni fatte sono descritte nel diagramma:



Il dataset $D=\{(\mathbf{x}_1,y_1), \ldots (\mathbf{x}_m,y_m)\}$ viene in parte utilizzato per l'apprendimento del classificatore, (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 (fase di *test* del modello).

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 predetta.

Modelli di classificazione

- **Modelli di regressione**, ipotizzano un'esplicita forma funzionale per la probabilità condizionata $P_{y|\mathbf{x}}(y|\mathbf{x})$. Esempio: *regressione logistica*.

- **Modelli euristici**, utilizzano procedure di classificazione basate su schemi algoritmici elementari ed intuitivi. Esempi: *nearest neighbor*, *decision trees*.

- **Modelli probabilistici**, formulano un'ipotesi circa la forma funzionale delle probabilità condizionate delle osservazioni data la classe target di appartenenza, $P_{\mathbf{x}|y}(\mathbf{x}|y)$, chiamate probabilità condizionate alle classi. Si sfrutta una stima della probabilità a priori $P_y(y)$ e il teorema di Bayes per ricavare la probabilità a posteriori della variabile di classe. Esempi: *Naïve Bayes*, *Bayesian Networks*.

- **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). Esempi: *analisi discriminante*, *Support Vector Machines*, *Neural Networks*.

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

$$g: R^n \rightarrow R$$

che associa ad ogni osservazione **x** un valore interpretabile come una stima della probabilità che la classe predetta dal classificatore per l'osservazione **x** sia effettivamente corretta.

Nei modelli probabilistici g(**x**) è esattamente la probabilità che la classe predetta sia giusta. Nei modelli di separazione g(**x**) può rappresentare per esempio la distanza di **x** dal bordo della regione di separazione.

La *score function* consente di ricavare una regola di classificazione per prevedere la classe target associata all'osservazione **x**. Per esempio, scegliendo f(**x**)=sgn(g(**x**)) si ottiene l'assegnamento alle classi {-1,+1}.

Valutazione dei modelli di classificazione

Nel corso di un'analisi di classificazione è opportuno sviluppare diversi modelli (sia in termini di algoritmo selezionato sia in termini di valori dei relativi parametri) al fine di selezionare un modello di classificazione, un algoritmo e un set di parametri che garantiscano la *massima accuratezza predittiva*.

I modelli di classificazione vengono valutati in base a:

- ***Accuratezza***
- ***Velocità***
- ***Robustezza***
- ***Scalabilità***
- ***Intepretabilità***

Accuratezza

- 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.

Siano

$$D_T = \text{training set contenente t osservazioni}$$

$$D_V = \text{test set contenente v osservazioni}$$

tali che

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

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

Indichiamo con y_i la classe di appartenenza associata all'osservazione $\mathbf{x}_i \in D_V$ e con $f(\mathbf{x}_i)$ la classe prevista per mezzo di $f \in F$, funzione che implementa l'algoritmo di classificazione $A=A_F$. Possiamo allora definire la seguente funzione di perdita (loss function)

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

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

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

In alternativa si può utilizzare la *percentuale di errori* commessi dal classificatore

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

Velocità e scalabilità

La complessità degli algoritmi di classificazione può essere misurata in termini di:

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

La scelta di un modello di classificazione dipende fortemente dalle caratteristiche del problema da affrontare.

Un classificatore, il cui addestramento è molto oneroso computazionalmente, può comunque essere comunque addestrato riducendo la dimensionalità dei dati disponibili, per esempio per mezzo di un campionamento casuale delle osservazioni.

La scalabilità di un classificatore è legata alla predisposizione ad apprendere da grandi quantità di dati. Questa proprietà è intrinsecamente collegata alla velocità di apprendimento.

Robustezza e interpretabilità

La robustezza di un algoritmo di classificazione si riferisce alla robustezza in termini di

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

Nei casi in cui l'analisi sia orientata alla comprensione del problema, oltre all'accuratezza, alla velocità e alla robustezza, gioca un ruolo fondamentale anche l'*interpretabilità* dei risultati. Le regole prodotte devono infatti essere semplici e comprensibili per l'*esperto di dominio*.

Metodo Holdout

Siano

D_T = training set contenente t osservazioni

D_V = test set contenente v osservazioni

tali che

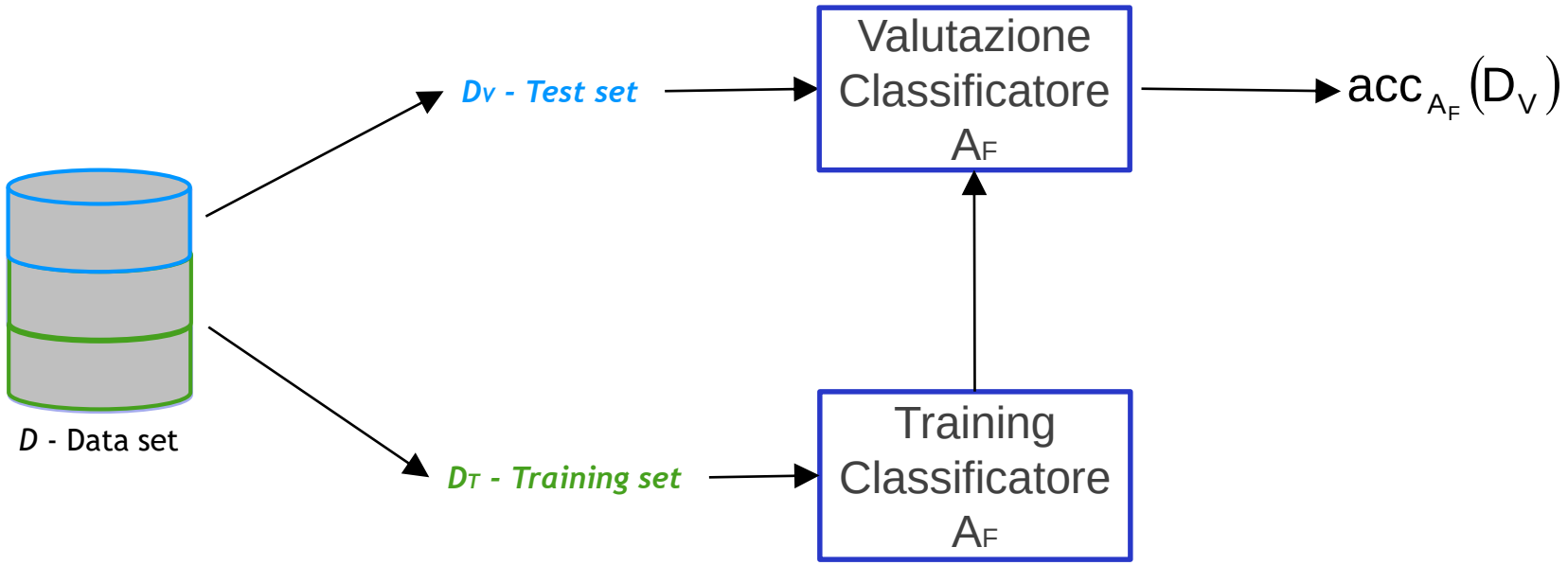
$$D = D_T \cup D_V,\ D_T \cap D_V = \emptyset,\ m=t+v$$

L'accuratezza di un modello di classificazione viene stimata calcolando

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

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.

Il metodo *Hold-Out* consiste nel dividere il dataset disponibile in due parti ed usare una sola di esse per stimare il modello di classificazione. L'altra parte del dataset verrà utilizzata per stimare il grado di affidabilità del modello.



La stima dell'accuratezza dipende dalla scelta del test set, pertanto è possibile sovrastimare o sottostimare il reale valore dell'accuratezza. Una stima più robusta può essere ottenuta per mezzo dell'*Iterated Hold-Out* e della *Cross Validation*.

Metodo Iterated Holdout

Il metodo dei campionamenti casuali ripetuti o *Iterated Hold-Out* consiste nel replicare R volte l'applicazione del metodo Hold-Out.

Per ogni iterazione r=1,…, R:

1. si estrae un campione casuale indipendente, indicato con D_{Tr}, di cardinalità pari a t osservazioni, da cui si apprende il modello di classificazione;
2. si valuta l'accuratezza del modello su $D_{Vr}=D-D_{Tr}$.

L'accuratezza del classificatore A_F viene stimata tramite la media campionaria delle accuratezze per le singole repliche:

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

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

Osservazioni

a) Il metodo Iterated Hold-Out è decisamente preferibile rispetto al metodo Hold-Out in quanto in grado di ottenere una stima maggiormente attendibile.

b) Il metodo Iterated Hold-Out non consente però 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 esistano osservazioni dominanti, inusuali, anomale, o degli 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.

Metodo Cross Validation

Il metodo 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.

1. Si effettua una ripartizione del dataset D in K sottoinsiemi disgiunti, esaustivi (partizione di D) e di cardinalità il più possibile uguale D_1, D_2, …, D_K;
2. per ogni iterazione k=1,…K, si effettua:
   a) una fase di apprendimento sul dataset $D_{T_k} = \{D_1,\ldots,D_{k-1},D_{k+1},\ldots,D_K\}$
   b) una fase di test sul dataset $D_{V_k} = D_k$ calcolando l'accuratezza del modello
3. si calcola l'accuratezza del modello come media aritmetica dell'accuratezza ottenuta nelle singole ripetizioni

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

Osservazioni:

a) L'insieme DT_k viene utilizzato come insieme di training mentre l'insieme D_k viene utilizzato come insieme di test per l'iterazione "k-ma".

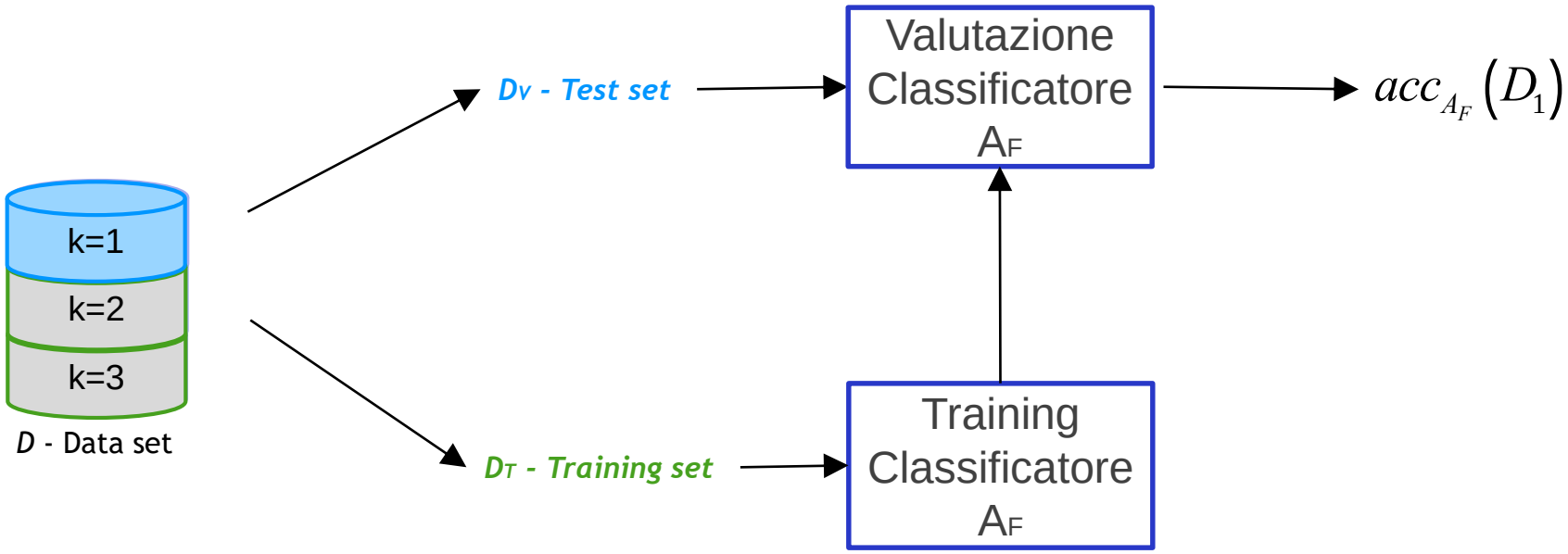
b) Lo stimatore dell'accuratezza ottenuto col metodo Cross Validazione è uno stimatore più robusto degli stimatori visti precedentemente per stimare l'effettiva accuratezza del classificatore.

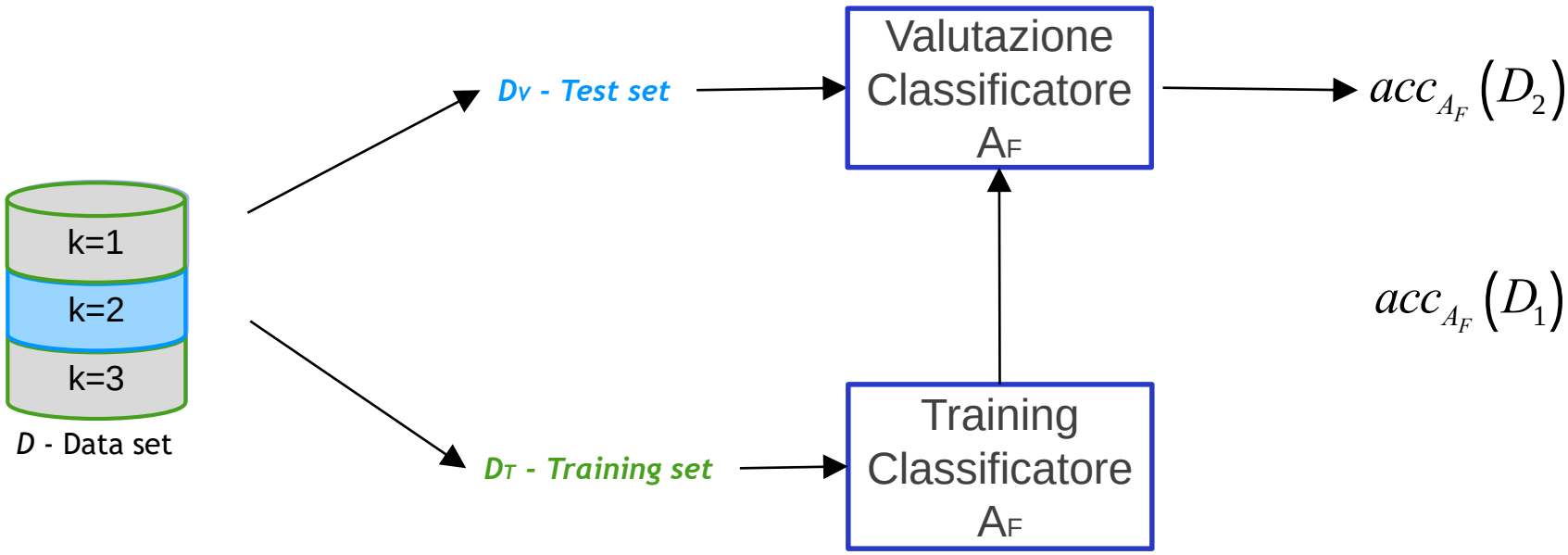
Esistono diverse possibilità per la scelta del valore del parametro K nello schema di cross validation.

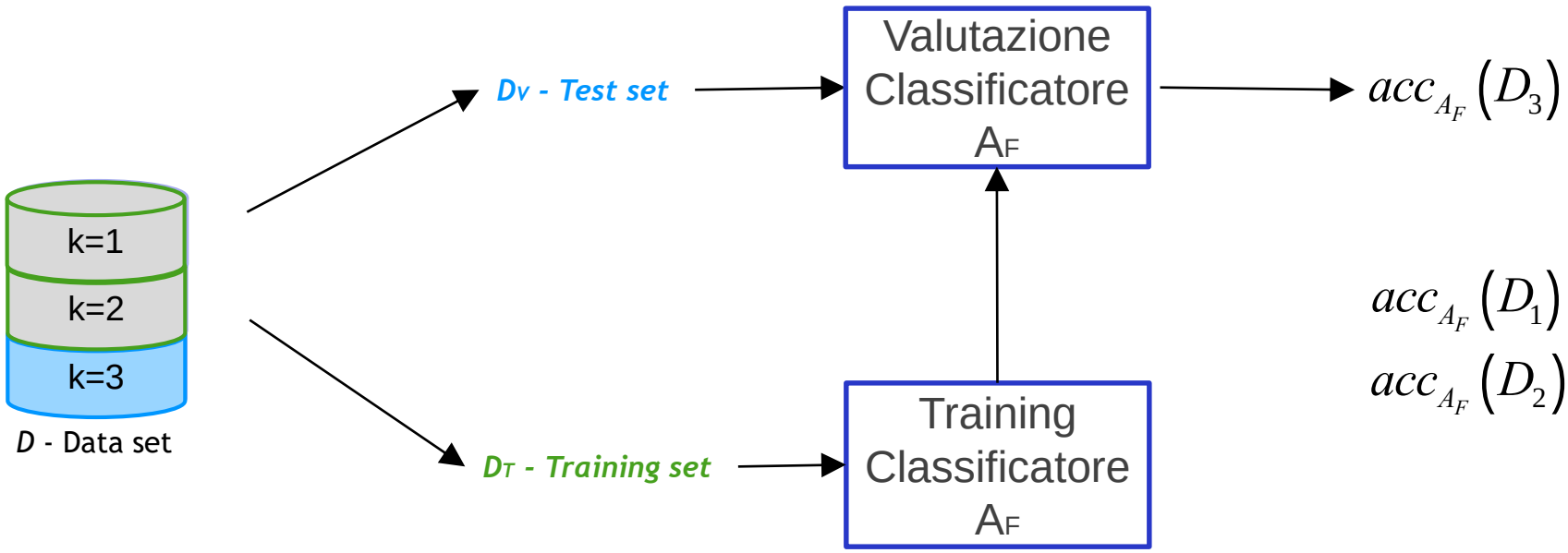
Valori usuali sono K=3, 5, 10, per K=10 il metodo si chiama Tenfold Cross Validation.

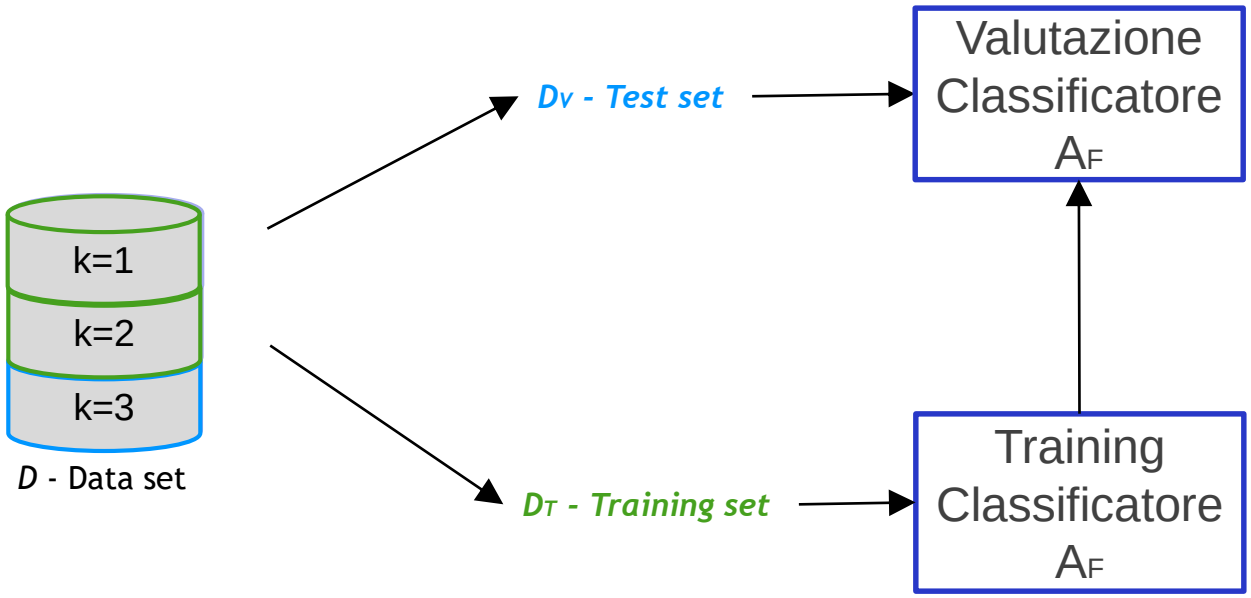
Una scelta spesso utilizzata nella letteratura specializzata, soprattutto nel caso in cui la numerosità dei dati disponibili sia bassa, è nota con il nome di *Leave One Out Cross Validation*.

Leave One Out 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.


k=1
k=2
k=3
D - Data set
Dv - Test set
DT - Training set
Valutazione Classificatore AF
Training Classificatore AF
$acc_{A_F}(D_1)$


k=1
k=2
k=3
D - Data set
Dv - Test set
DT - Training set
Valutazione Classificatore AF
Training Classificatore AF
$acc_{A_F}(D_2)$
$acc_{A_F}(D_1)$


k=1
k=2
k=3
D - Data set
Dv - Test set
DT - Training set
Valutazione Classificatore AF
Training Classificatore AF
$acc_{A_F}(D_3)$
$acc_{A_F}(D_1)$
$acc_{A_F}(D_2)$



$$acc_{A_F}(D_1)$$
$$acc_{A_F}(D_2)$$
$$acc_{A_F}(D_3)$$

$$acc_A = acc_{A_F} = \frac{1}{3}\sum_{k=1}^{3} acc_{A_F}(D_k)$$

Nota bene:

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.

Altri metodi di valutazione: Matrici di Confusione

Accuratezza ed errore non sempre sono esaustive ed adeguate per valutare la qualità di un modello di classificazione e per confrontare diversi modelli di classificazione.

Esempio:

Supponiamo di voler sviluppare un modello di classificazione per diagnosticare una malattia genetica a partire dalla misurazione dei livelli di espressione genetica di un individuo. Supponiamo inoltre che all'interno del nostro dataset solo il 2% dei pazienti sia affetto da tale malattia.

Il classificatore che assegna sempre la classe "non malato" a tutti pazienti ha un'accuratezza pari al 98%.

Il 2% dei dati misclassificati corrisponde però ai pazienti che si vuole classificare correttamente per diagnosticare la malattia!

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**).

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

Gli *elementi 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**).

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

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

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

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

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

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

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

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

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

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

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

F-Measure $\mathrm{F} = \frac{(\beta^2 - 1) \cdot \%\mathrm{TP} \cdot \mathrm{prc}}{\beta^2 \cdot \mathrm{prc} + \%\mathrm{TP}}$ $\beta \in [0, +\infty)$

regola l'importanza relativa di **prc** rispetto a **%TP**.

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

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

matrice dei costi di misclassificazione

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*) deve essere 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.

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)

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

Accuratezza = 0.8

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

Accuratezza = 0.9

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

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

Accuratezza = 0.8

Costo = 40*100+60=4060

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

Accuratezza = 0.9

Costo = 45*100+5=4505

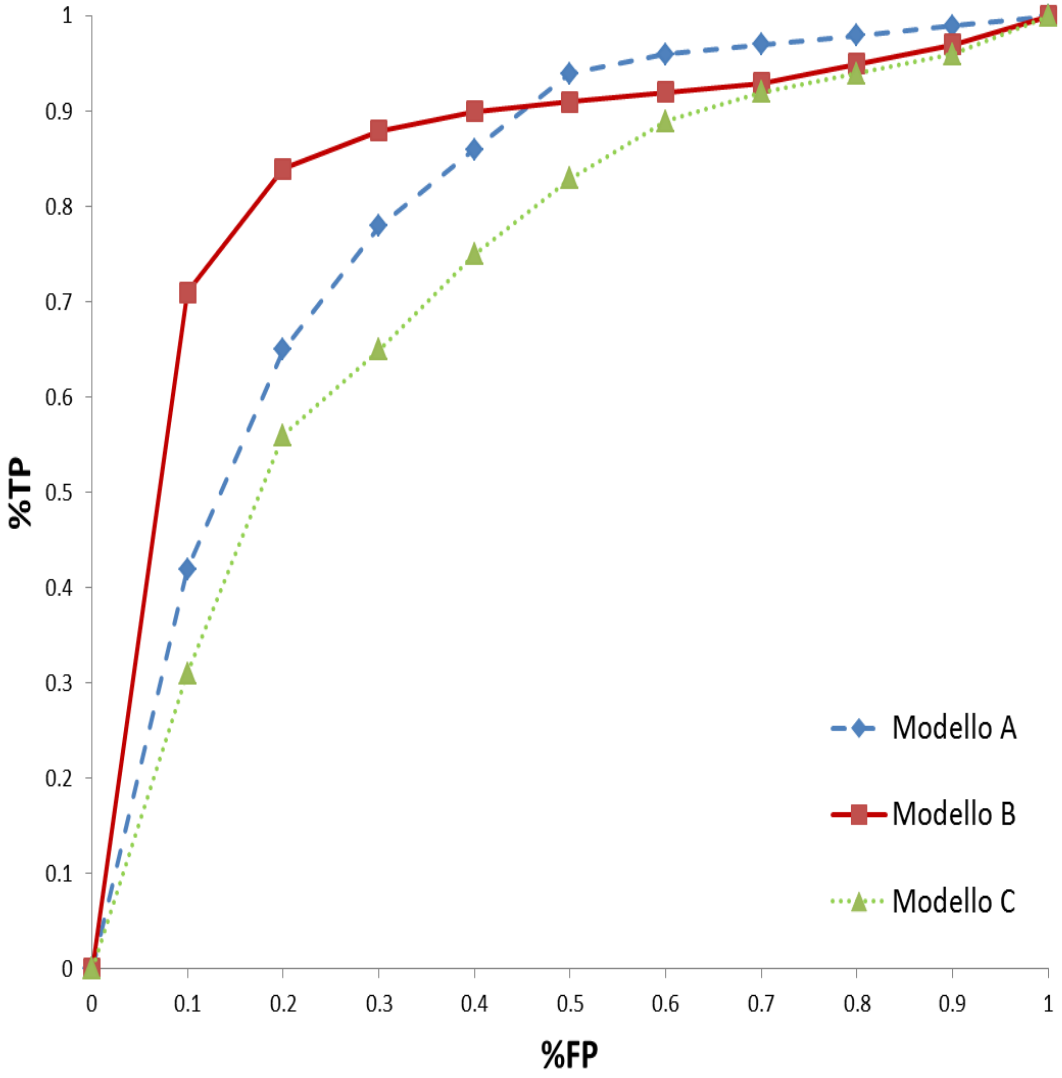
Altri metodi di valutazione: grafico ROC

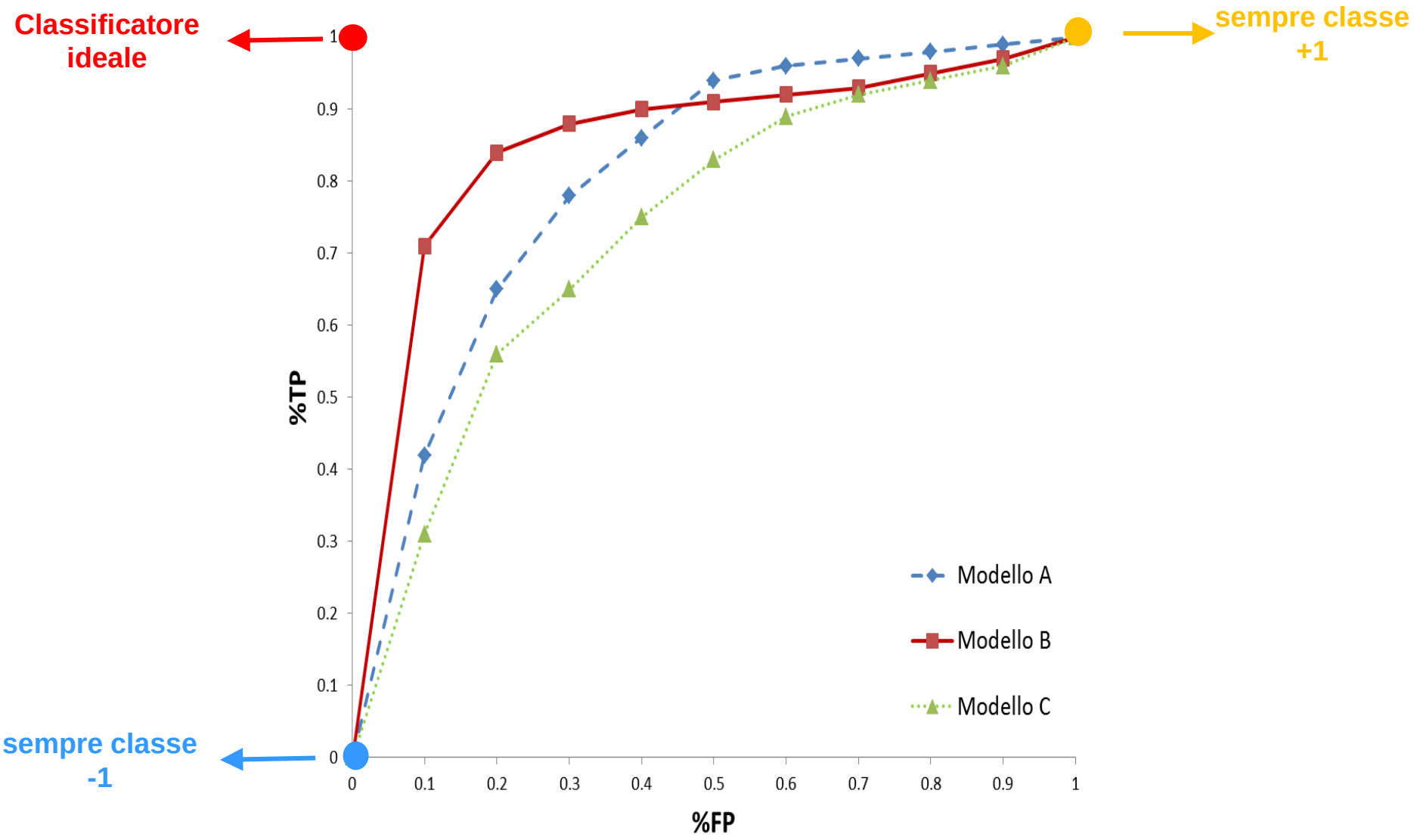
I grafici *Receiving Operating Characteristic* curve (**ROC**) consentono di valutare visivamente la qualità di un classificatore ed al contempo consentono di comparare tra loro diversi modelli di classificazione.

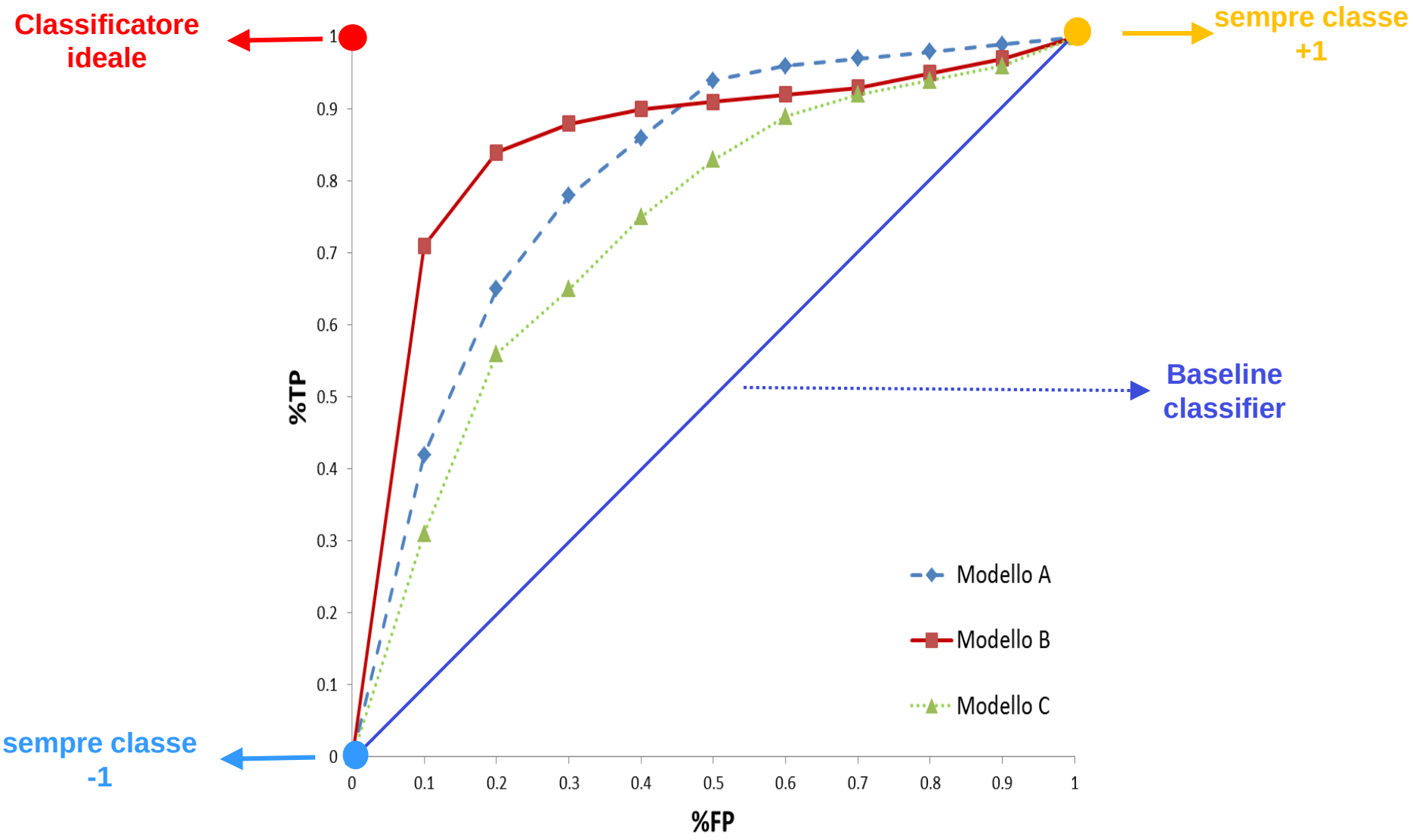
Forniscono la sintesi delle informazioni ricavabili tramite una sequenza di matrici di confusione e consentono di determinare il trade-off ottimale tra il numero di osservazioni positive classificate correttamente (**TP**) ed il numero di osservazioni negative classificate in modo errato (*FP*), offrendo un'alternativa all'assegnazione dei costi di misclassificazione.

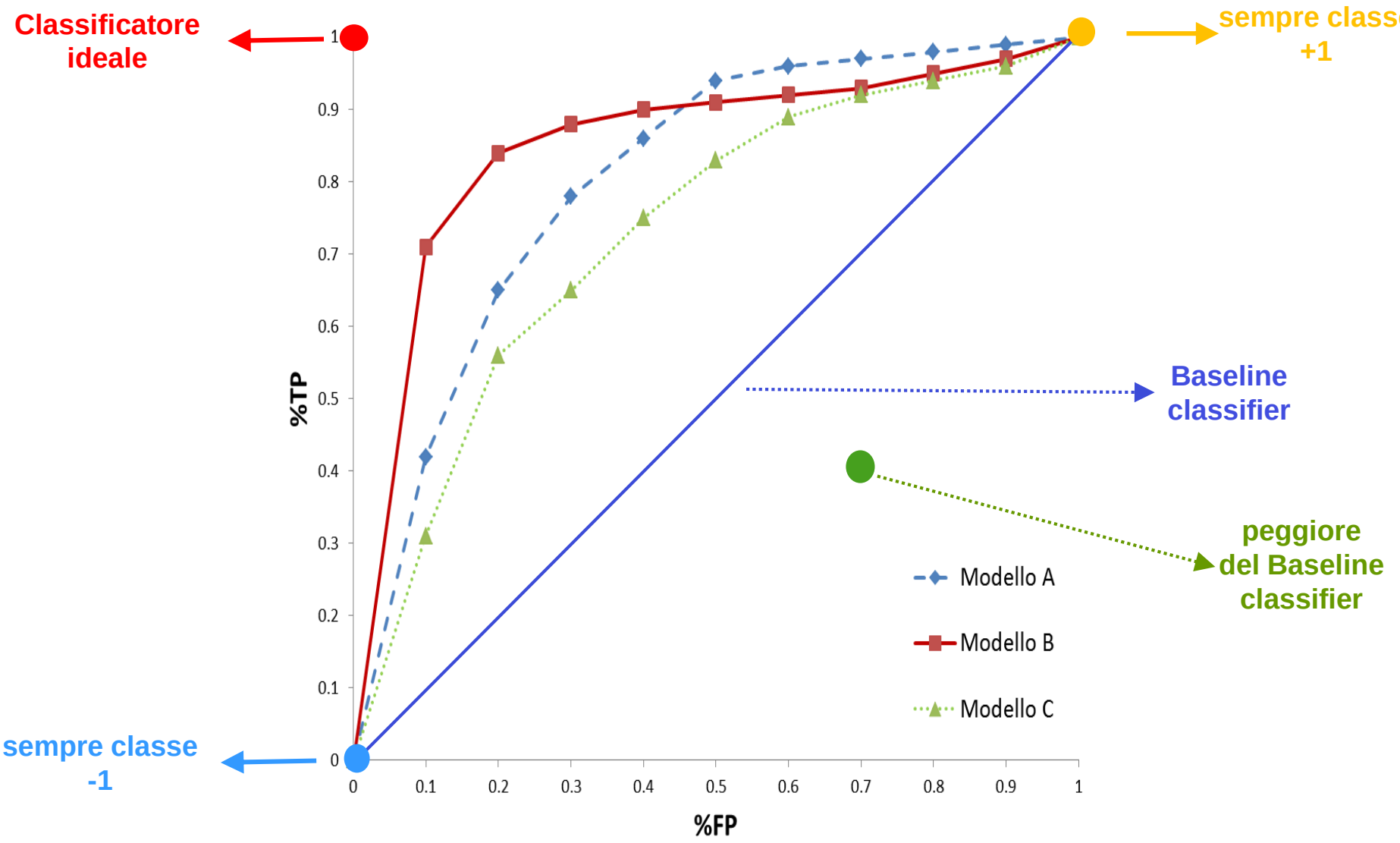
Il grafico ROC è un riporta:

- sull'asse delle ascisse la percentuale di false positive (*%FP o FPR*)
- sull'asse delle ordinate la percentuale di true positive (**%TP o TPR**)


%TP
%FP
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Modello A
Modello B
Modello C


Classificatore ideale
sempre classe +1
sempre classe -1
%TP
%FP
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Modello A
Modello B
Modello C


Classificatore ideale
sempre classe +1
Baseline classifier
sempre classe -1
%TP
%FP
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Modello A
Modello B
Modello C


Classificatore ideale
sempre classe +1
Baseline classifier
peggiore del Baseline classifier
sempre classe -1
%TP
%FP
Modello A
Modello B
Modello C
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

La maggior parte dei classificatori permette di effettuare tuning di alcuni parametri (ad esempio la soglia di probabilita' oltre la quale viene restituita una classe piuttosto che l'altra) in modo da aumentare il numero di true positive (**TP**) a discapito di un conseguente aumento del numero di false positive (*FP*).

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

(**TPR**, *FPR*)

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.

True Positive Rate (o Recall)
TPR = TP/(TP+FN)

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

- (TP+FN) e' il numero totale di osservazioni di classe +1
- Data un'istanza di classe +1, TPR e' la probabilita' di classificarla positiva
- Un airport scanner (bomb/ no bomb) richiede un TPR molto alto: se c'e' una bomba, vogliamo rilevarla!
- Di solito ci serve TPR alto quando ci sono poche osservazioni di classe +1

False Positive Rate = FP/(FP+TN)

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

- (FP+TN) e' il numero di osservazioni di classe -1
- Data un'istanza di classe -1, FPR e' la probabilita' di classificarla positiva (quindi di sbagliare)
- Uno spam filter ("spam" ha classe +1), richiede FPR basso: non vogliamo mandare dello spam (classificare come positivo) della mail valida (classe vera -1)!

Receiver – Operating curve (ROC)

- Vogliamo sempre massimizzare TPR e mimizzare FPR.
- Ma e' un tradeoff!
- Esempio: un classificatore che predice sempre +1 che valori ha di TPR e FPR?

Receiver – Operating curve (ROC)

- Vogliamo sempre massimizzare TPR e mimizzare FPR.
- Ma e' un tradeoff!
- Esempio: un classificatore che predice sempre +1 che valori ha di TPR e FPR?
- TPR=1 (il massimo) ma purtroppo anche FPR=1 (il massimo)

- La curva ROC ci permette di studiare il tradeoff tra TPR ed FPR
- Si applica solo a problemi di classificazione binari (2 classi)

Come costruire la curva ROC

Osservaz	$P(c_1|\mathbf{x})$	True Class
1	0.95	c_1
2	0.93	c_1
3	0.87	c_0
4	0.85	c_0
5	0.85	c_0
6	0.85	c_1
7	0.76	c_0
8	0.53	c_1
9	0.43	c_0
10	0.25	c_1

- Due classi: c_1 e c_0
- Per ogni osservazione abbiamo valori distinti degli attributi esplicativi
- Per ogni istanza abbiamo la probabilita' della classe c_1 restuita dal classificatore (scoring function)

Come costruire la curva ROC

Osservaz	$P(c_1|\mathbf{x})$	True Class
1	0.95	c_1
2	0.93	c_1
3	0.87	c_0
4	0.85	c_0
5	0.85	c_0
6	0.85	c_1
7	0.76	c_0
8	0.53	c_1
9	0.43	c_0
10	0.25	c_1

- Uno per uno, considero ogni valore distinto di $P(c_1| x)$ come possibile soglia di classificazione
- Per ogni valore della soglia:
  - Classifico ogni osservazione come c_1 se e solo se $P(c_1| x)$ e' maggiore o uguale alla soglia. Altrimenti c_0
  - calcolo FPR e TPR

Come costruire la curva ROC

Actual class	C_1	C_0	C_1	C_0	C_0	C_0	C_1	C_0	C_1	C_1	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

TPR= TP/(TP+FN) FPR= FP/(FP+TN)

Nella prima colonna usiamo 0.25 come soglia di classificazione: tutte le osservazioni con score maggiore o uguale a 0.25 saranno classificate come positive. Cotniamo TP,FP,TN,FN e calcoliamo TPR ed FPR

Nella seconda colonna usiamo 0.43 come soglia di classificazione e facciamo lo stesso. E cosi' via

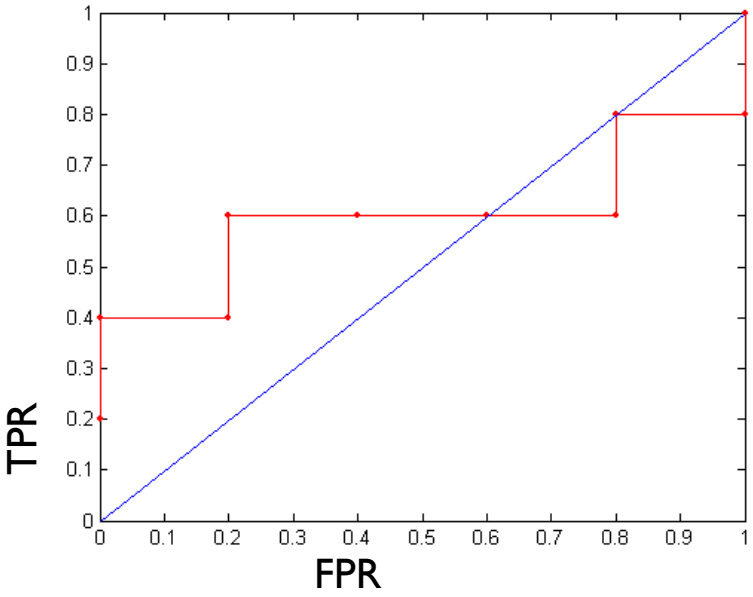
Nell'ultima colonna (soglia 1.00) consideriamo tutte le istanze come negative.

Come costruire la curva ROC

Ogni punto della curva e' definito da una coppia (TPR, FPR)

Se disegnamo questi punti nello spazio (FPR,TPR) otteniamo la ROC

TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0



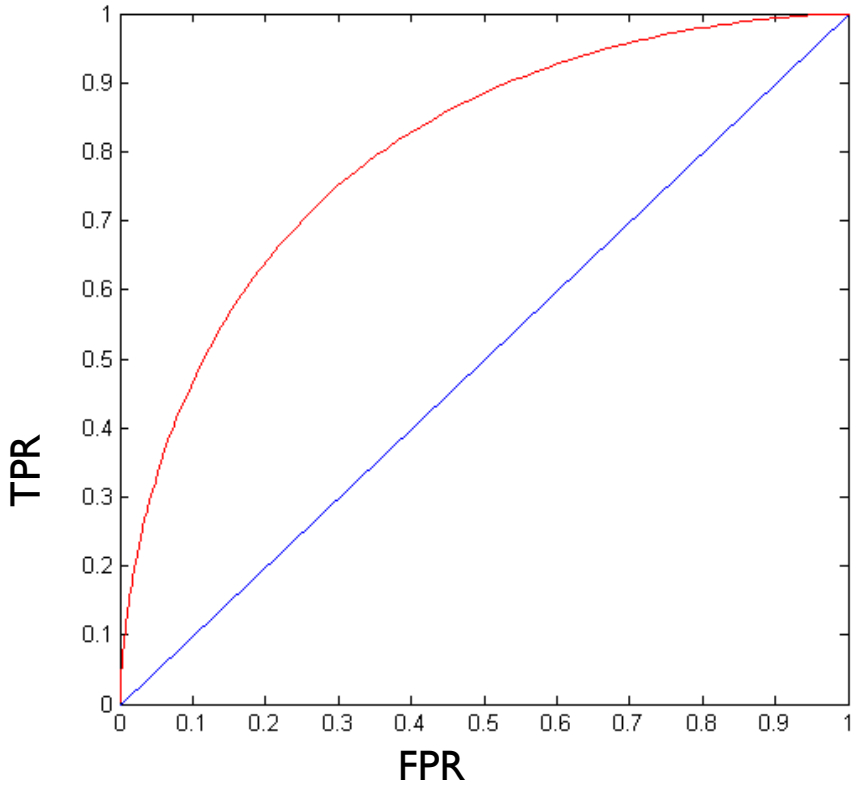
In rosso la ROC del nostro classificatore. In blu la diagonale, che e' la ROC di un classificatore casuale

Il punto a sx della tabella e' il punto in alto a dx della curva rossa

Punti chiave

(TPR,FPR):

- (0,0): classifico ogni osservazione come negativa
- (1,1): classifico ogni osservazione come positiva
- (1,0): classificatore ideale



- Sulla diagonale: la performance di un classificatore che risponde a caso
- Sotto la diagonale: peggio che rispondere a caso

Il classificatore che risponde a caso

- Un *random guesser* e' un classificatore che predice positivo con probabilita' p indipendentemente dai valori delle feature. Casi limite: rispondi sempre positivo (p=1) o sempre negativo (p=0).

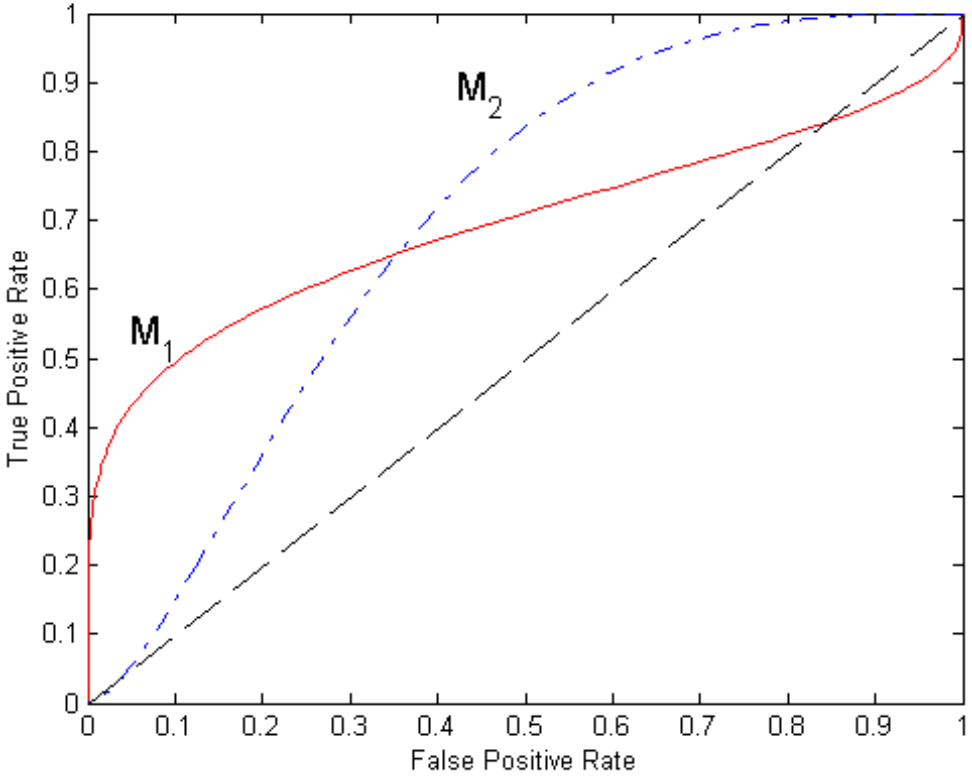
- Siano il numero di istanze positive e negative rispettivamente n_+ e n_-

$$\mathrm{TPR} = \mathrm{TP} / \left(\mathrm{TP} + \mathrm{FN}\right) \Rightarrow \mathrm{TPR} = \frac{pn_+}{pn_+ + (1-p)n_+} = p$$

$$\mathrm{FPR} = \mathrm{FP} / \left(\mathrm{FP} + \mathrm{TN}\right) \Rightarrow \mathrm{FPR} = \frac{pn_-}{pn_- + (1-p)n_-} = p$$

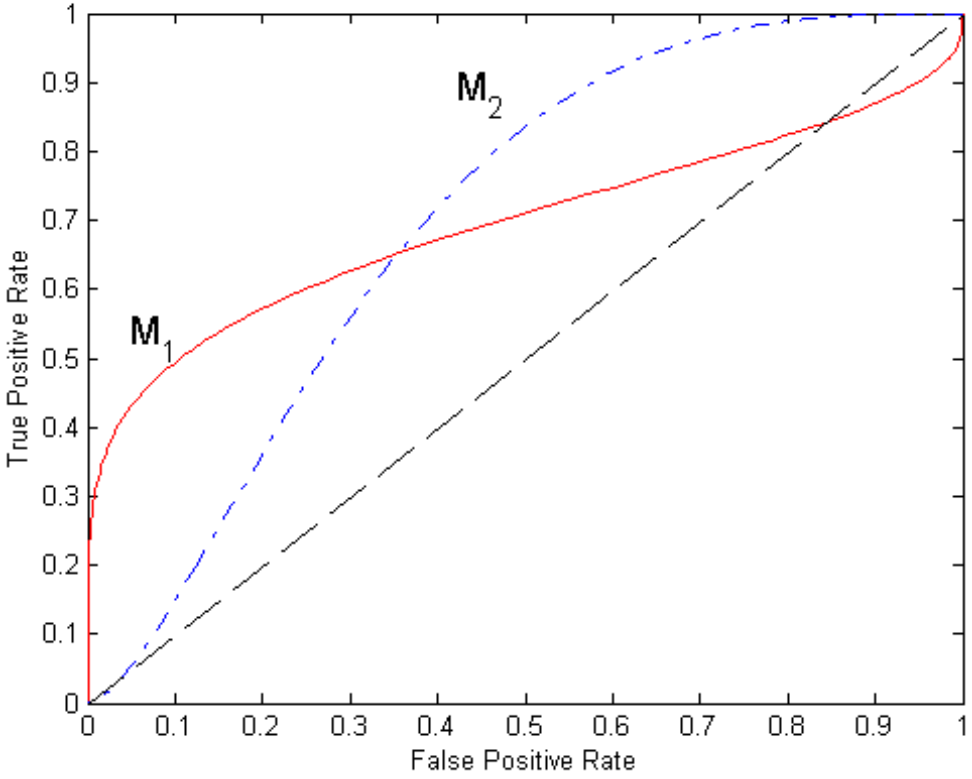
- Quindi Il random guesser ottiene approssimativamente TPR = FPR. Variando p nell'intervallo [0,1], si ottiene la roc del random guesser: la diagonale principale.

La ROC serve a confrontare i modelli



- Dato un FPR, preferiamo il metodo con il TPR piu' alto
- M1 e' un modello migliore se l'obiettivo e' avere un FPR basso. Esempio: filtri anti-spam (FP: mail vere classificate erroneamente come spam)
- M2 e' migliore se l'obiettivo e' avere TPR alto: scanner di sicurezza negli aeroporti: accettiamo un po' di FP, ma e' indispensabile avere TPR altissimo (se c'e' una bomba, la devo trovare)

La ROC serve a confrontare i modelli



- Non c'e' un vincitore consistente tra M1 ed M2 per tutti i valori di FPR
- Possiamo calcolare l'area AUC sottesa alla curva (valore compreso tra 0 e 1), che riassume la qualita' del classificatore ma perde informazioni sul comportamento per FPR bassi e alti.