Docente: **Giorgio Valentini**
Tutor: **Matteo Re**

UNIVERSITÀ DEGLI STUDI DI MILANO

Insegnamento: **Bioinformatica**
A.A. 2011-2012 semestre II

C.d.l. **BIOTECNOLOGIE DEL FARMACO**

Apprendimanto automatico

Giorgio Valentini
e –mail: *valentini@dsi.unimi.it*
http://homes.dsi.unimi.it/~valenti

Matteo Re
e –mail: *re@dsi.unimi.it*
http://homes.dsi.unimi.it/~re

DSI – Dipartimento di Scienze dell' Informazione
Università degli Studi di Milano

Apprendimento automatico

Obiettivi:

Scrivere programmi che siano in grado di analizzare dei campioni e fornire, per ognuno di essi, un'opinione (usiamo il termine «opinione» in senso molto ampio).

I «campioni» possono essere geni, molecole, proteine, immagini ecografiche, analisi biochimico-cliniche, ecc. …

Apprendimento automatico

Perché è necessario?

- Non esistono «esperti» umani (navigazione su Marte)
- Operatori umani non sono in grado di spiegare come svolgono una determinata operazione (es. riconoscimento del parlato)
- La soluzione cambia nel tempo (il sistema deve essere in grado di adattarsi in modo estremamente rapido)
- La soluzione deve essere adattata a casi specifici (es. rilevatori biometrici: scansione retinica)

Apprendimento automatico

Diverse classi di metodi di apprendimento automatico:

- Il tipo di metodo dipende essenzialmente dal tipo di **opinione** che vogliamo esprimere (o dal tipo di problema che si vuole risolvere).

L'apprendimento automatico si divide classicamente in due grosse aree (anche se, di recente, ne stanno emergendo di nuove) :

- apprendimento **SUPERVISIONATO** ed
- apprendimento **NON SUPERVISIONATO**

Cosa si intende per apprendimento?

- Costruzione di modelli «generali» a partire da un ristretto insieme di esempi
- I **dati** costano poco e sono abbondanti (es statistiche di navigazione in rete); la **conoscenza** dei fenomeni che vogliamo analizzare è spesso **scarsa, incompleta (e costosa da produrre)**

OBIETTIVO: costruire modelli che siano utili a fini pratici e che rappresentino una **approssimazione** accettabile delle informazioni (conoscenza) a disposizione riguardanti il fenomeno che vogliamo studiare.

Apprendimento automatico dal punto di vista operativo

- Definizione di un **obiettivo**
- Definizione di un **criterio di ottimizzazione** che permetta di verificare se l'obiettivo è stato raggiunto (e di **quantificare** le performance)

Sono necessarie competenze da diverse aree di ricerca (statistica, informatica)

Ruolo dell'informatica: sviluppo di programmi efficienti per **risolvere il problema di ottimizzazione** (utilizzo del criterio di ottimizzazione per testare molte possibili soluzioni e trovare la migliore). Creazione di un **modello**. Utilizzo del modello per produrre un'**opinione**.

(alcune) Aree di applicazione

- **Finanza:** analisi di mercato, analisi e previsioni fluttuazioni azionarie, rilevamento di frodi
- **Astronomia:** classificazione galassie, classificazione stelle
- **Medicina:** supporto alla diagnosi
- **Bioinformatica:** analisi di motivi in sequenze di DNA/RNA, identificazione di nuovi geni, …
- **Sviluppo farmaci:** riposizionamento, predizione interazione farmaco-target

… e molte altre

(alcune) Aree di applicazione

- **Finanza:** analisi di mercato, analisi e previsioni fluttuazioni azionarie, rilevamento di frodi
- **Astronomia:** classificazione galassie, classificazione stelle
- **Medicina:** supporto alla diagnosi
- **Bioinformatica:** analisi di motivi in sequenze di DNA/RNA, identificazione di nuovi geni, …
- **Sviluppo farmaci:** riposizionamento, predizione interazione farmaco-target

… e molte altre

Metodi supervisionati

Ne esistono molti tipi ma hanno tutti alcune caratteristiche in comune:

- Operano in **fasi distinte**: **addestramento e test**
- Durante la fase di addestramento **ad ogni esempio sono associate delle informazioni aggiuntive** (spesso nella forma di un'etichetta che definisce la classe dell'esempio. Ad es. sano/malato).
- Durante la fase di addestramento viene prodotto un **modello** che, a partire dall'esempio permette di predire l'etichetta (o classe) dell'esempio.
- Durante la fase di test si usa il modello per predire la classe di esempi **non utilizzati durante l'addestramento** (ma per i quali è disponibile l'etichetta) e si quantificano le performance del modello.

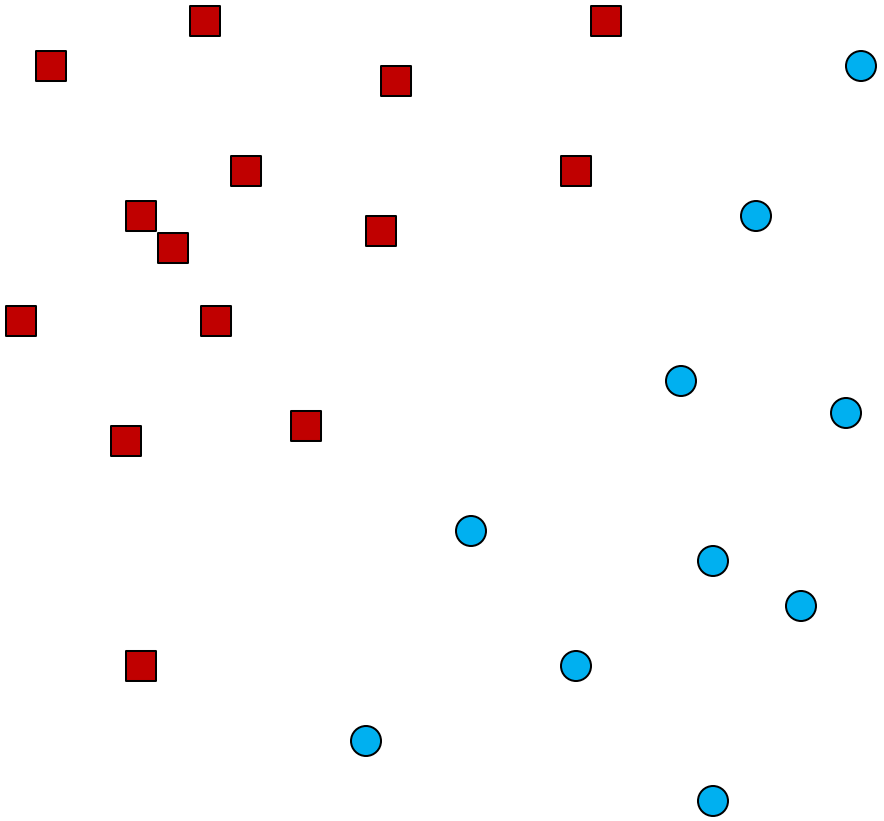
Librerie R

- In **R** esistono diverse librerie contenenti strumenti utilizzabili in esperimenti di apprendimento automatico (machine learning) . In particolare noi utilizzeremo:

- e1071 (CRAN)

- caret (CRAN)

Metodi supervisionati

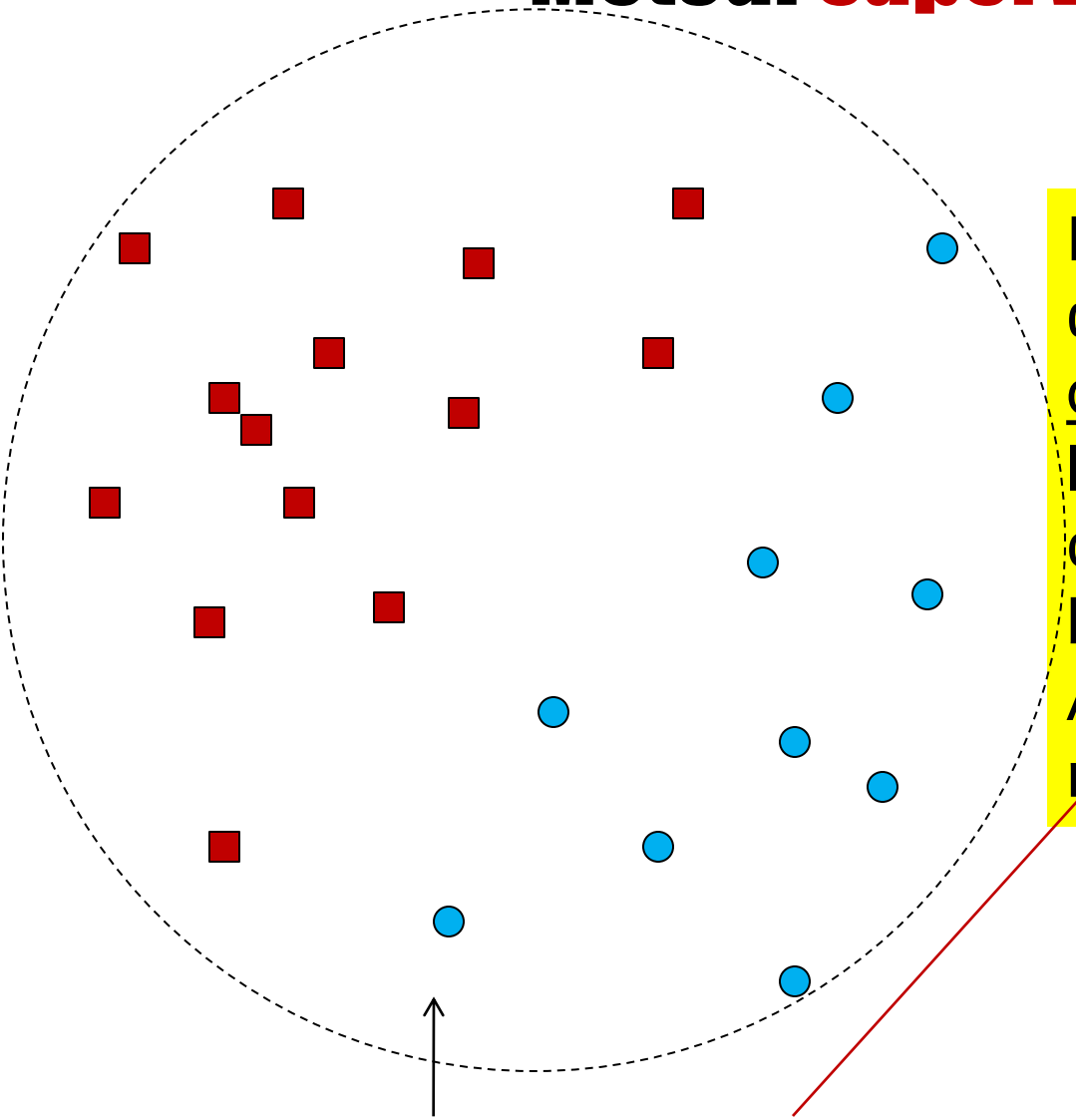


SCENARIO:

- Due classi (rosso/blu)
- Collezione di esempi
- **Obiettivo:** creazione di un **modello** che permetta di predire se la classe di un esempio è rosso o blu

TIPO DI PROBLEMA:
Classificazione binaria

Metodi supervisionati

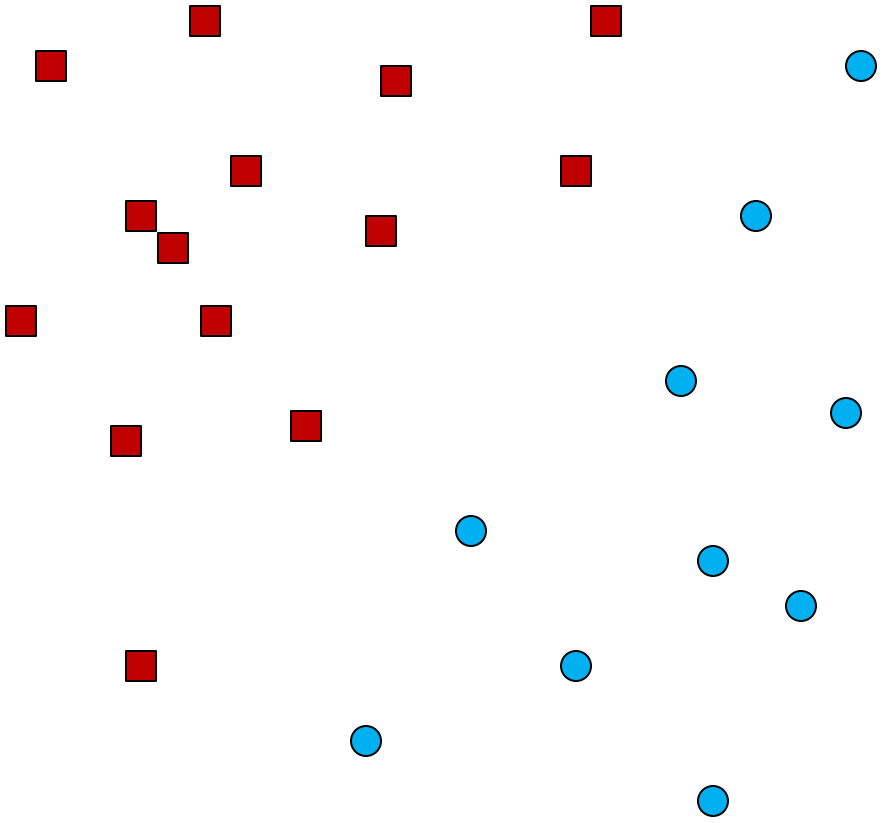


PUNTO FONDAMENTALE:

Per ogni esempio abbiamo a disposizione la **VERA** classe dell'esempio (sappiamo se l'esempio appartiene alla classe rosso o alla classe blu).
Abbiamo delle "etichette" di riferimento

DATI + CONOSCENZA → MODELLO

Metodi supervisionati



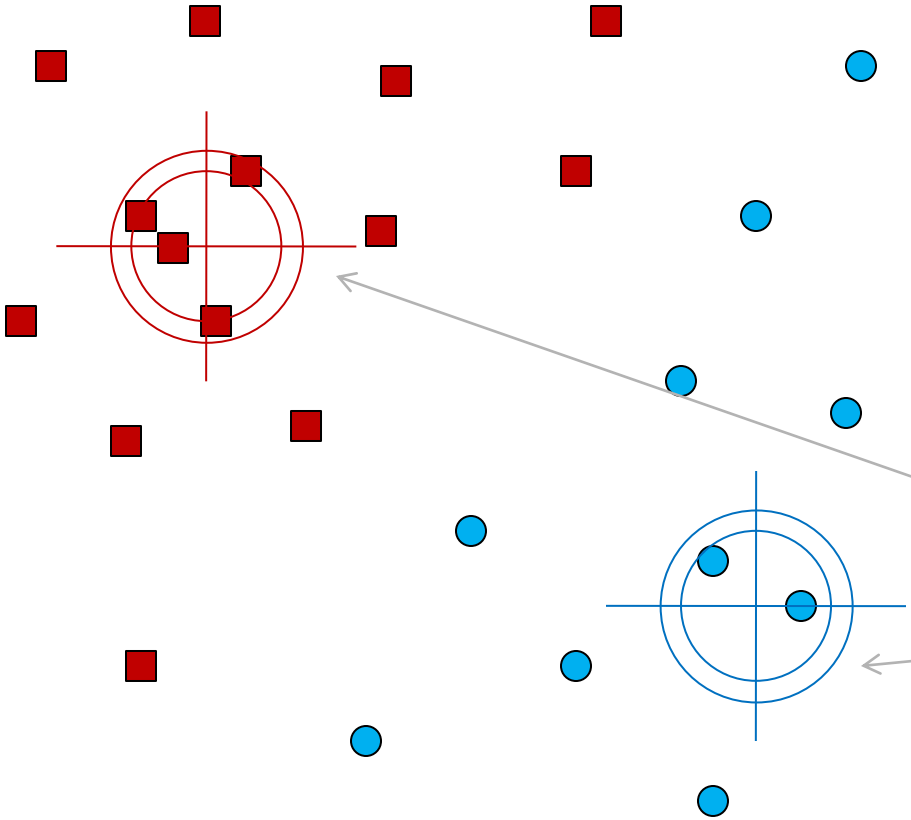
PROBLEMA:

L'obiettivo è la creazione di un modello …
Ma cosa è un modello?

"Modello" è un termine generico … ne esistono moltissimi tipi diversi!!!

Metodi supervisionati

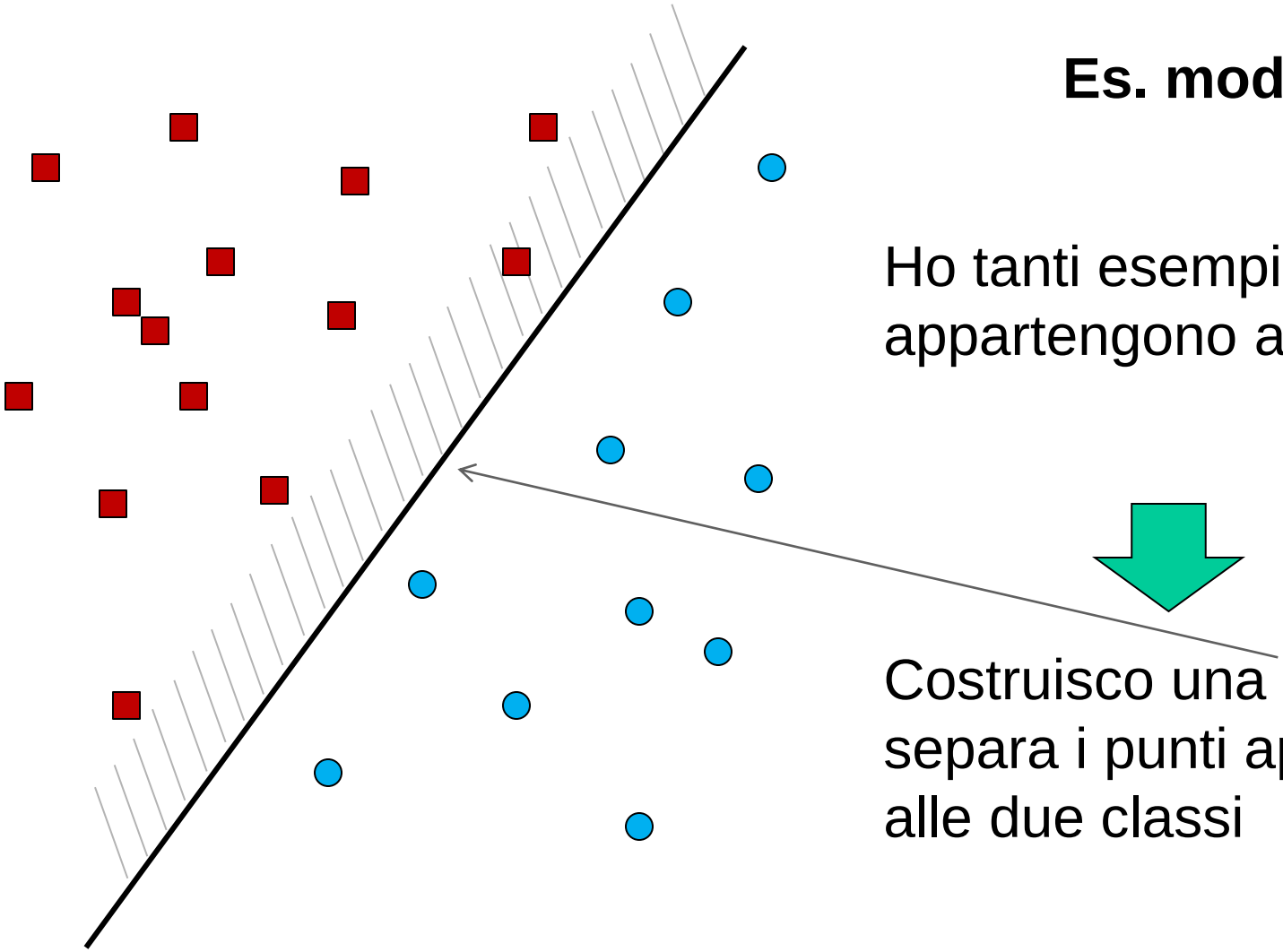
Es. modello 1 :



Ho tanti esempi (punti) che appartengono a due classi.

Calcolo un **“punto” medio** per ognuna delle due classi

Metodi supervisionati



Es. modello 2 :

Ho tanti esempi (punti) che appartengono a due classi.

Costruisco una **retta** che separa i punti appartenenti alle due classi

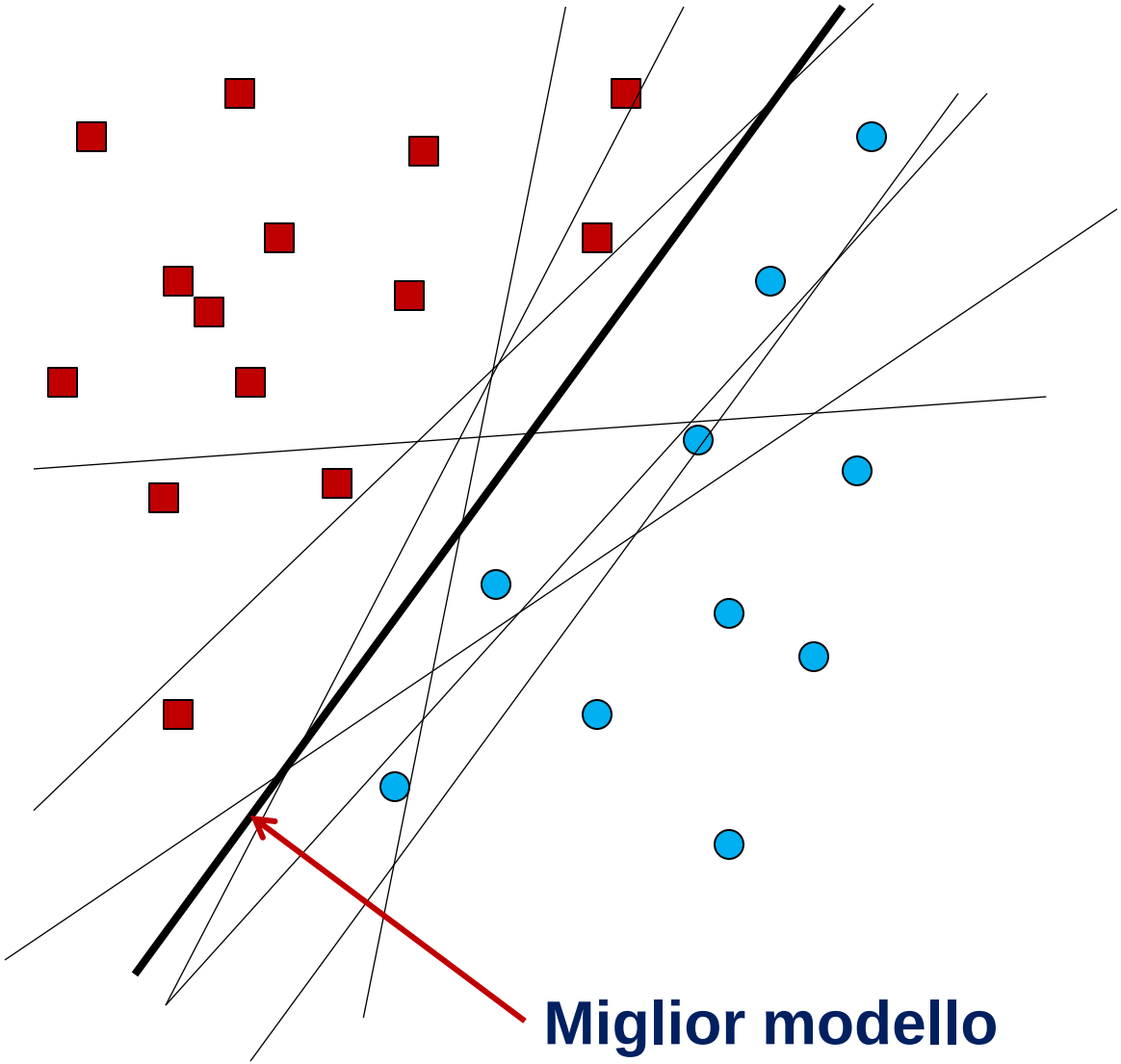
Metodi supervisionati

NB: La costruzione dei tipi di modello che abbiamo visto nelle slide precedenti è possibile solo se siamo in grado di riconoscere la classe degli esempi (punti).

Questo vuol dire che possiamo addestrare il modello **SOLO SE ABBIAMO LE ETICHETTE DEI PUNTI CHE VEDIAMO DURANTE L'ADDESTRAMENTO**.

Tra tutti i possibili modelli (ad es. tra tutte le possibili rette separatrici) sceglieremo quello che separa meglio i punti …

Metodi supervisionati



Esistono infinite rette

- Voglio scegliere quella che separa meglio i punti delle due classi
- Posso testare molti modelli (rette) e scegliere quello che separa meglio le classi **SOLO SE CONOSCO LE CLASSI DEI PUNTI**.

Metodi supervisionati

• Scelgo il “tipo” di modello che voglio addestrare

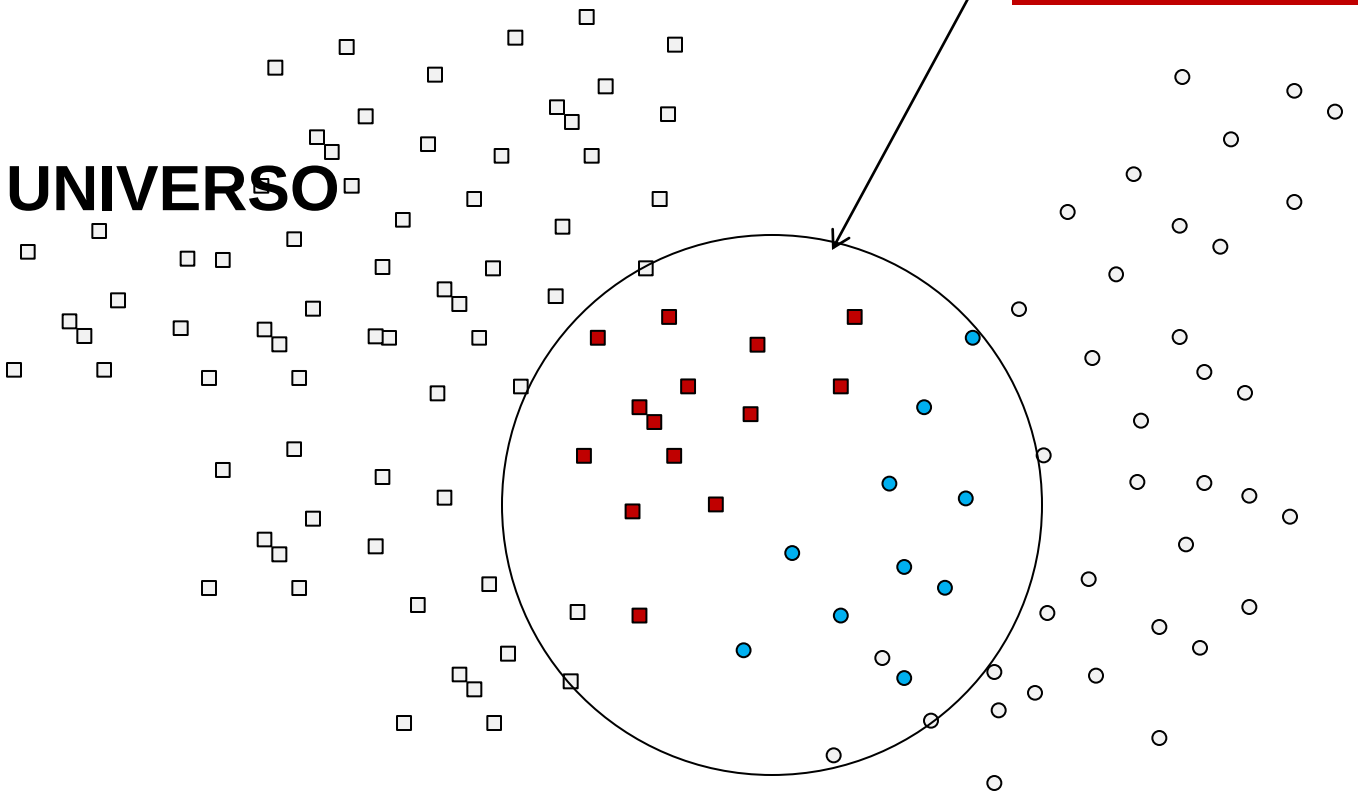
• Testo varie versioni del modello : **OTTIMIZZAZIONE del modello** , per ogni versione del modello quantifico l’errore nella classificazione dei punti

• Posso ottimizzare perché conosco la vera classe dei punti di addestramento

Questi metodi si chiamano “supervisionati” perché **durante il training la scelta del modello finale è effettuata sulla base di performance quantificabili mediante il confronto tra classi predette (dal modello) e classi reali (definite dalle etichette).**

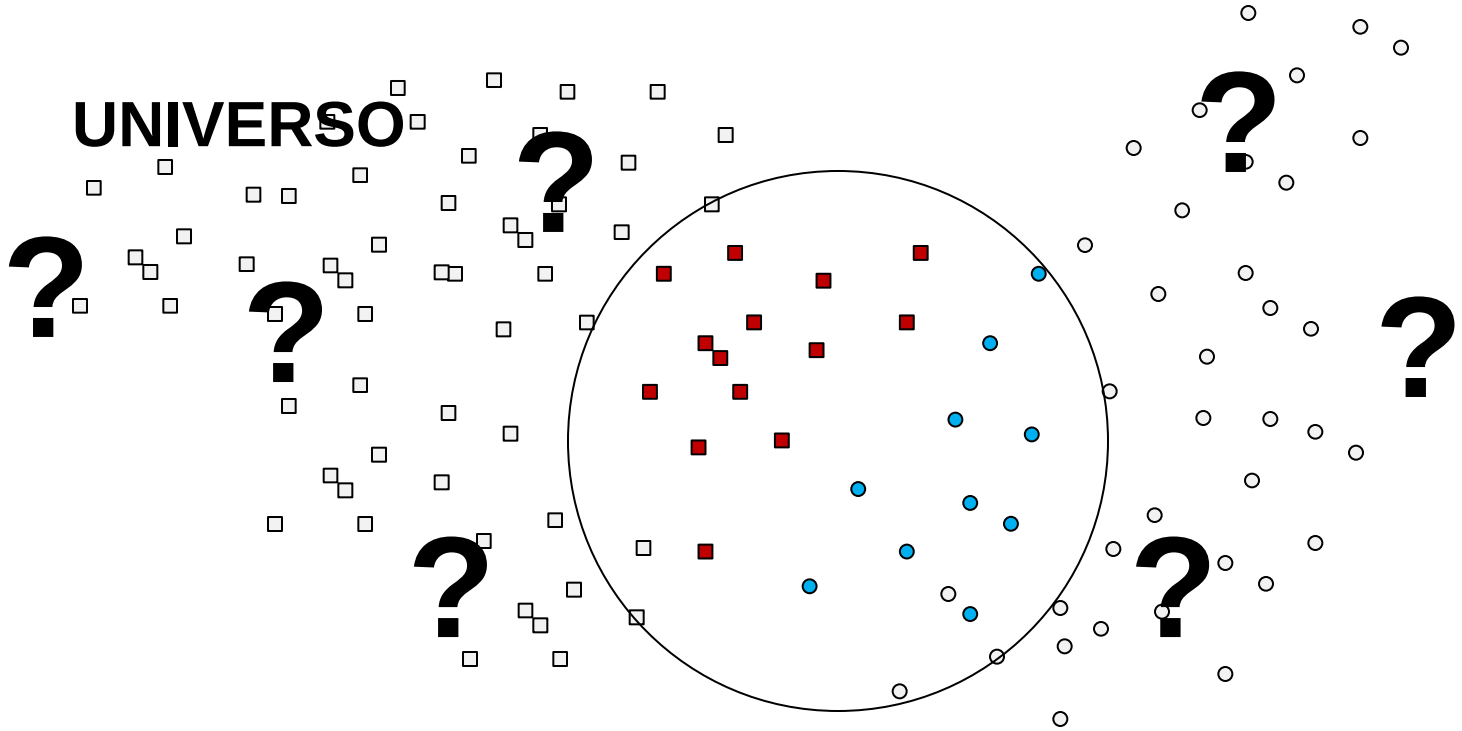
Metodi supervisionati

I metodi supervisionati sono tra i più potenti per la soluzione di problemi di classificazione.
Tuttavia non sono esenti da difetti.



Metodi supervisionati

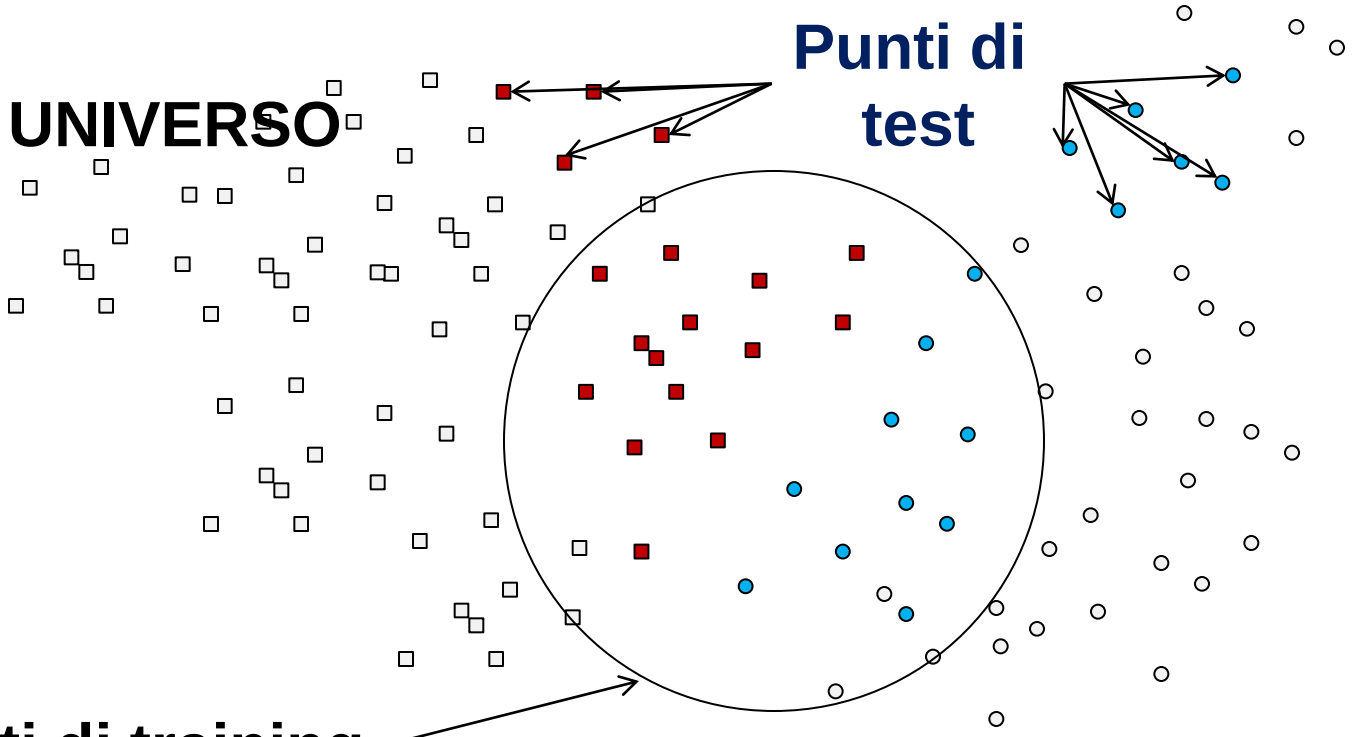
Chi mi garantisce che un modello che ha performance buone sui dati di training non avrà performance mediocri su tutti i possibili punti che non abbiamo usato durante l'addestramento?



Metodi supervisionati

Possibile soluzione: Testare il modello addestrato su punti

- per i quali è **disponibile un'etichetta**
- **non utilizzati** in fase di addestramento



Metodi supervisionati

Un problema cruciale per tutti i metodi di apprendimento automatico (indipendentemente dal fatto che siano supervisionati o non supervisionati) è la quantificazione delle performance.

Anche in questo caso il tipo di metodo di quantificazione dipende strettamente dal problema affrontato.

Vediamo un esempio che riguarda:

- un metodo di apprendimento **supervisionato**
- un problema di **classificazione** (decidere se un punto è di classe “rosso”

Metodi supervisionati

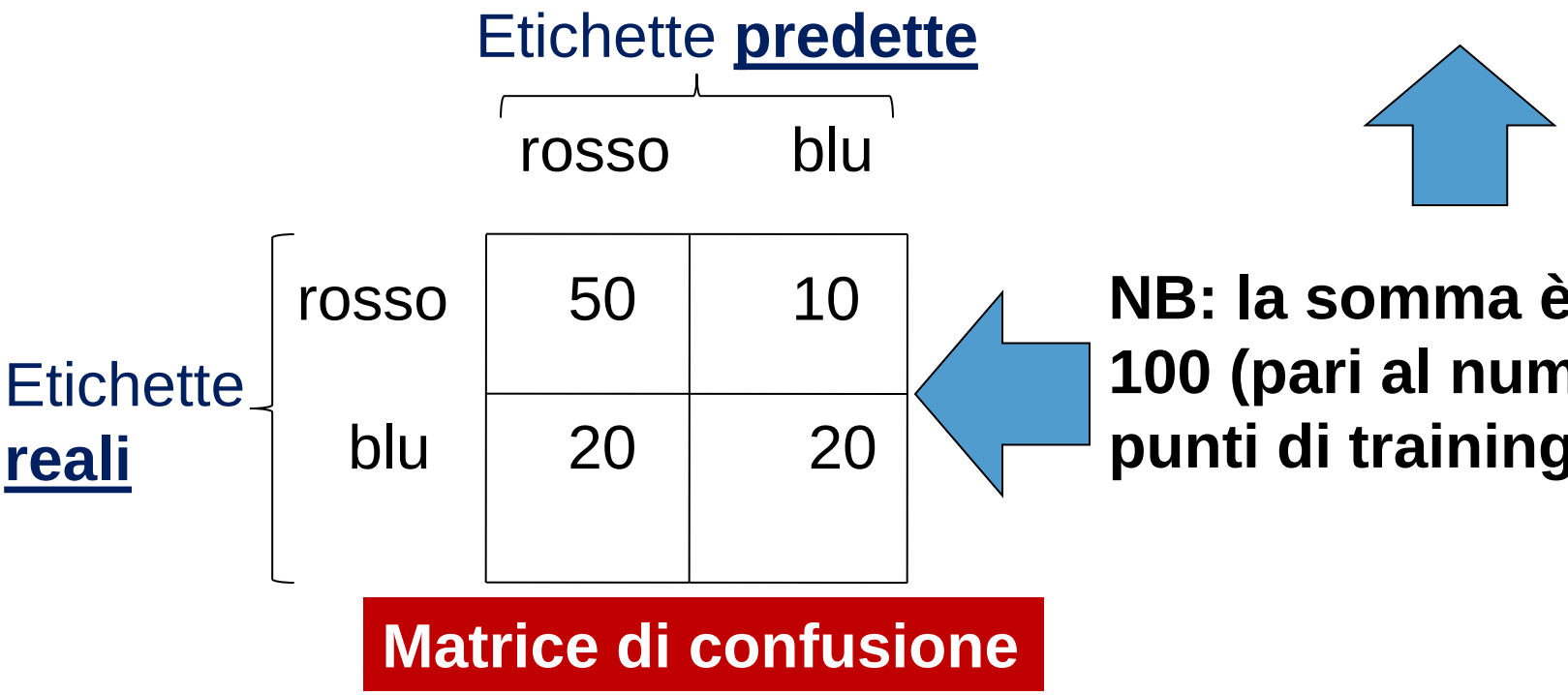
Supponiamo di aver addestrato un modello utilizzando 100 esempi di training. Di questi 100, 60 sono di classe “rosso” e 40 sono di classe “blu”.

Il modello ha effettuato le seguenti classificazioni:

		Etichette **predette**: rosso	Etichette **predette**: blu
Etichette **reali**	rosso	50	10
	blu	20	20

Metodi supervisionati

Etichetta reale	Etichetta predetta	Tipo predizione
rosso	rosso	True positive (TP) (50)
rosso	blu	False negative (FN) (10)
blu	blu	True negative (TN) (20)
blu	rosso	False positive (FP) (20)



Metodi supervisionati

Etichetta reale	Etichetta predetta	Tipo predizione
rosso	rosso	True positive (TP) (50)
rosso	blu	False negative (FN) (10)
blu	blu	True negative (TN) (20)
blu	rosso	False positive (FP) (20)

Misure di performance :

Precisione: «Quanti dei punti che ho predetto di tipo **rosso** sono **davvero** punti di **classe «rosso»** ?

$$\text{Precisione: } \frac{TP}{TP + \mathbf{FP}} = 50 / (50+20) = 0.7142$$

Metodi supervisionati

Etichetta reale	Etichetta predetta	Tipo predizione
rosso	rosso	True positive (TP) (50)
rosso	blu	False negative (FN) (10)
blu	blu	True negative (TN) (20)
blu	rosso	False positive (FP) (20)

Misure di performance :

Sensibilità: «Dei punti che **dovrei** predire come «rosso» quanti ne ho predetti di classe «**rosso**»?

$$\text{Sensibilità} : \frac{TP}{TP + \mathbf{FN}} = 50 / (50+10) = 0.8333$$

somma dei positivi nel training set

Metodi supervisionati

Etichetta reale	Etichetta predetta	Tipo predizione
rosso	rosso	True positive (TP) (50)
rosso	blu	False negative (FN) (10)
blu	blu	True negative (TN) (20)
blu	rosso	False positive (FP) (20)

Misure di performance :

Specificità: «Dei punti che **ho predetto** come «blu» (negativo) quanti sono in realtà di classe «**rosso» (positivo)**?

$$\text{Specificità} : \frac{TN}{TN + \mathbf{FN}} = 20 / (20+10) = 0.6666$$

Curva Receiver Operating Characteristic (ROC)

La curva **ROC** è una tecnica statistica attualmente utilizzata in una grande varietà di campi scientifici.

Questa tecnica trae origine nell'ambito della **teoria della rivelazione del segnale**. Si tratta di una metodologia che è stata utilizzata per la prima volta da alcuni ingegneri, durante la seconda guerra mondiale, per l'analisi delle immagini radar e lo studio del rapporto segnale/disturbo.

E' possibile usare la curva ROC per valutare le performance di un modello di classificazione.

Curva Receiver Operating Characteristic (ROC)

Razionale:

Supponiamo di aver addestrato un classificatore che, per ogni esempio, esprima un giudizio nella forma di un numero reale compreso tra 0 e 1 (inclusi) .

Per ogni esempio abbiamo a disposizione una etichetta che ci informa rispetto alla classe dell'esempio.

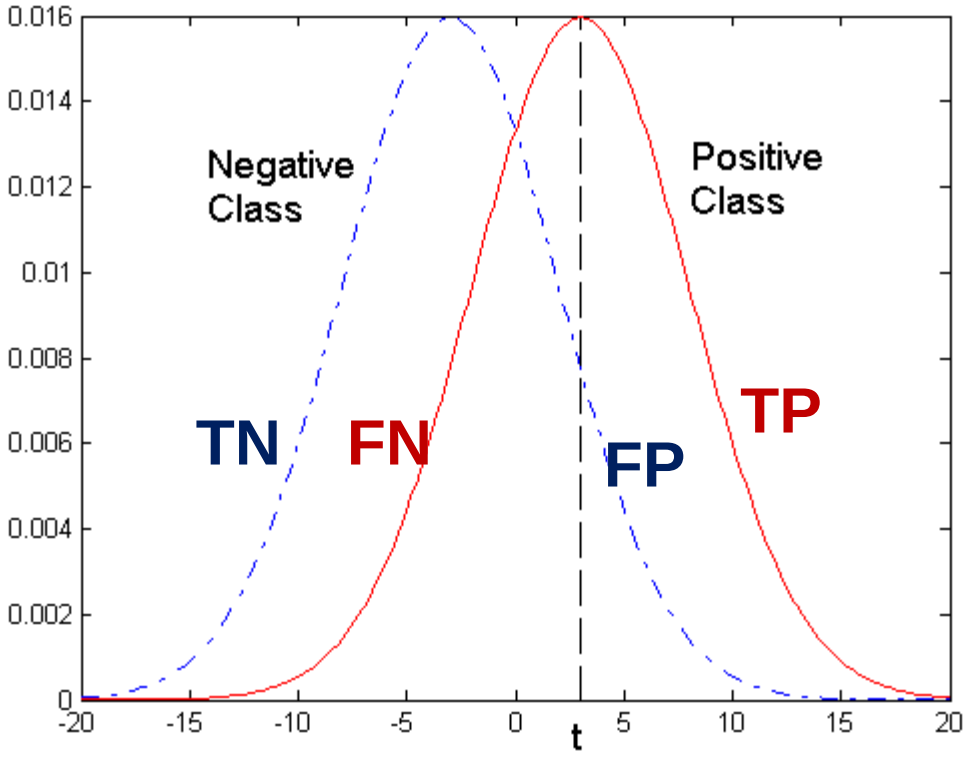
Possiamo definire una **soglia t** e decidere che se lo score prodotto dal classificatore è **>t** allora l'opinione espressa è «**positivo**», in caso contrario l'opinione espressa è «**negativo**».

Curva Receiver Operating Characteristic (ROC)

Razionale:

Fissata una soglia t, tutti gli esempi per i quali il modello (classificatore) genera uno score >t vengono predetti positivi.

Questo ci permette di quantificare, per ogni scelta del valore soglia, **TP, TN, FP e FN.**

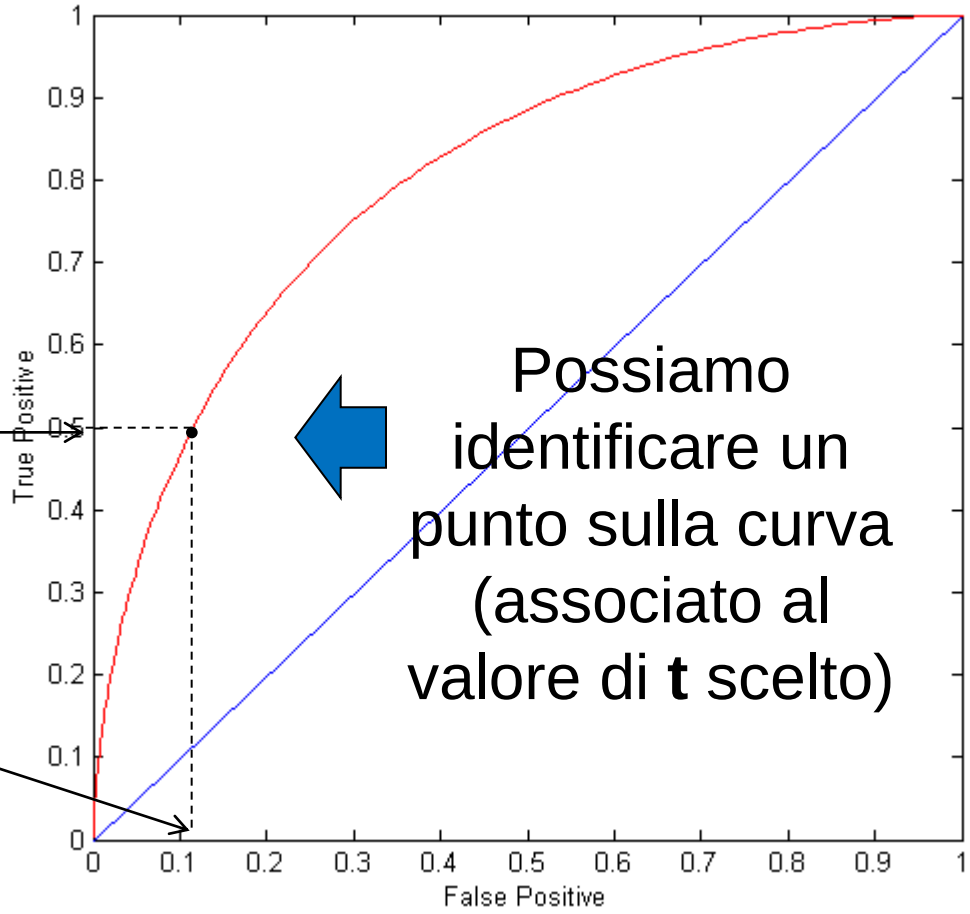


Curva Receiver Operating Characteristic (ROC)

Razionale:

Fissato un valore di t possiamo calcolare, ad esempio:

TP=0.5
FN=0.5
FP=0.12
FN=0.88

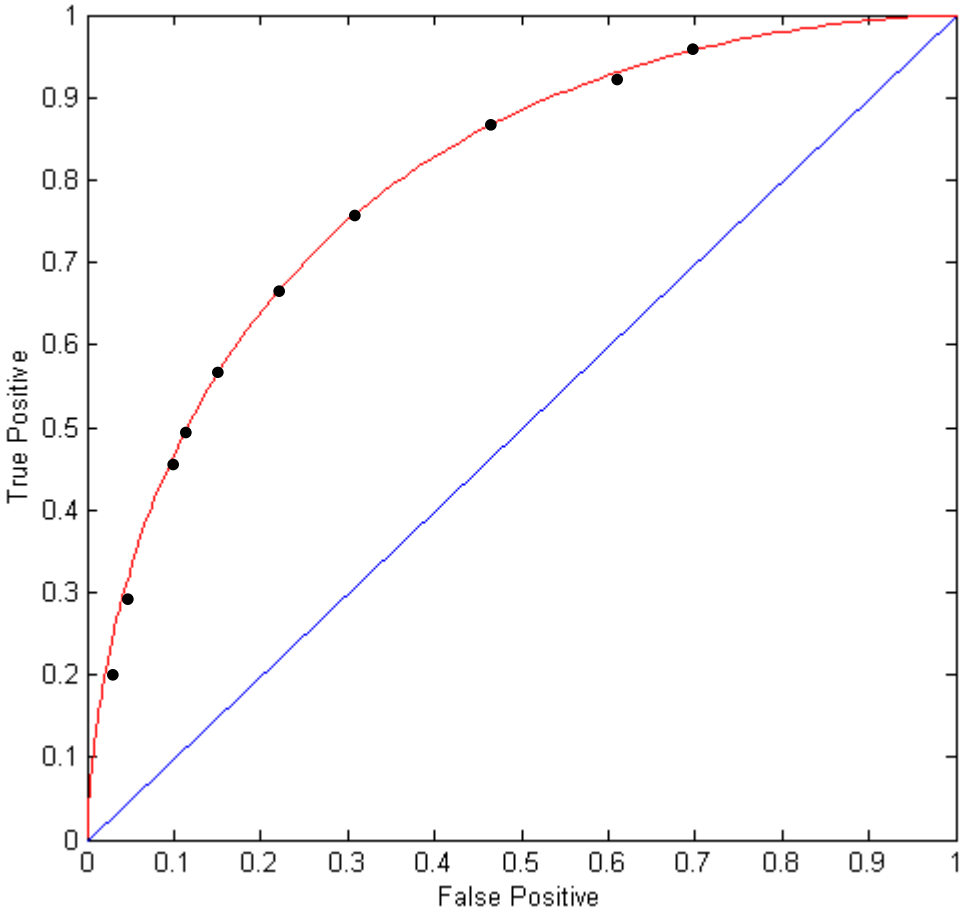


Possiamo identificare un punto sulla curva (associato al valore di **t** scelto)

Curva Receiver Operating Characteristic (ROC)

Razionale:

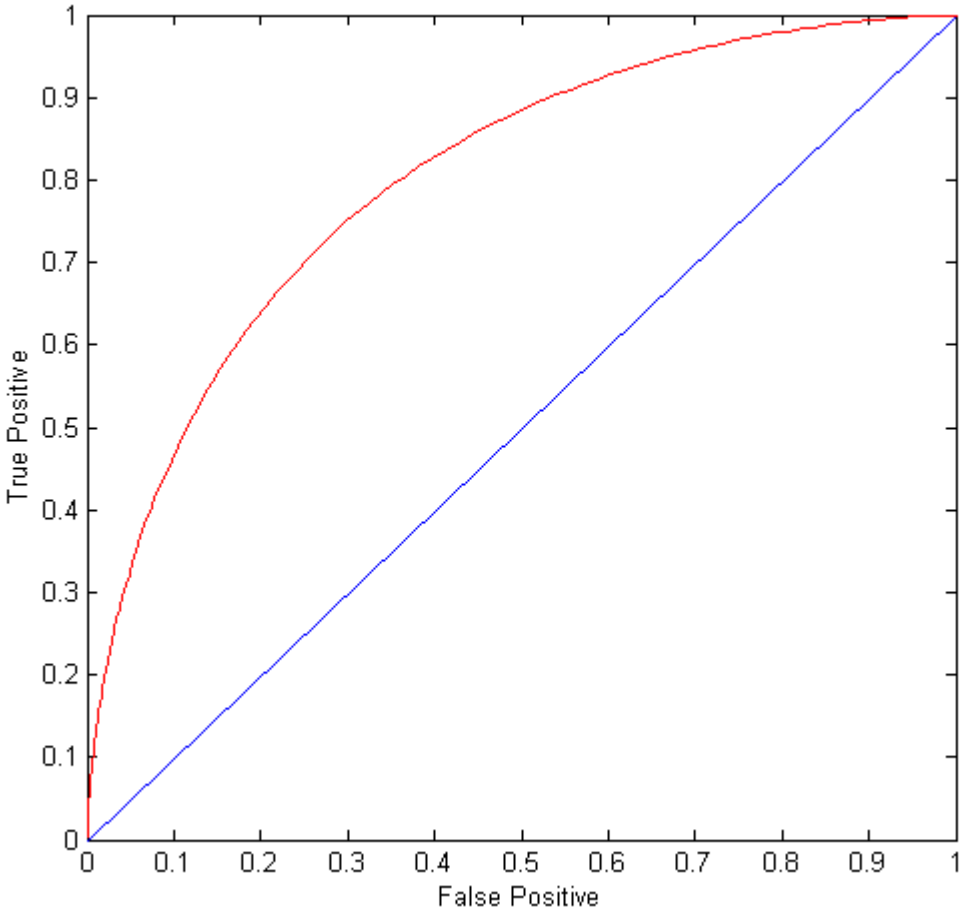
Il valore degli score generati dal classificatore può variare tra 0 e 1 . Possiamo calcolare TP e FP per una serie di valori di t (es. 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, …, 1.0). In questo modo otteniamo diversi punti che compongono la curva ROC.



Curva Receiver Operating Characteristic (ROC)

Razionale:

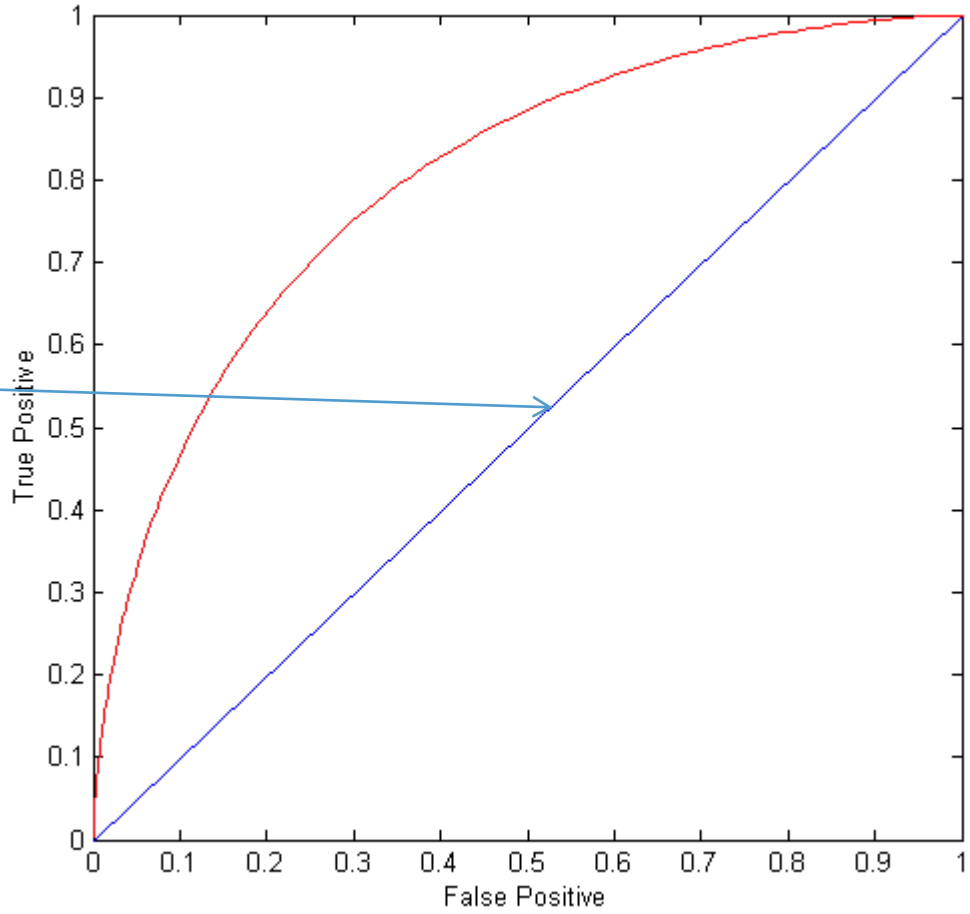
Ciò che è importante non è la curva di per sé … ma **l'area sotto la curva**. Questa quantità è indicata con il termine Area Under the ROC curve (**AUC**)



Curva Receiver Operating Characteristic (ROC)

AUC = 1 :
Il classificatore è perfetto

AUC = 0.5 :
Il classificatore è totalmente casuale (lancio di una moneta)



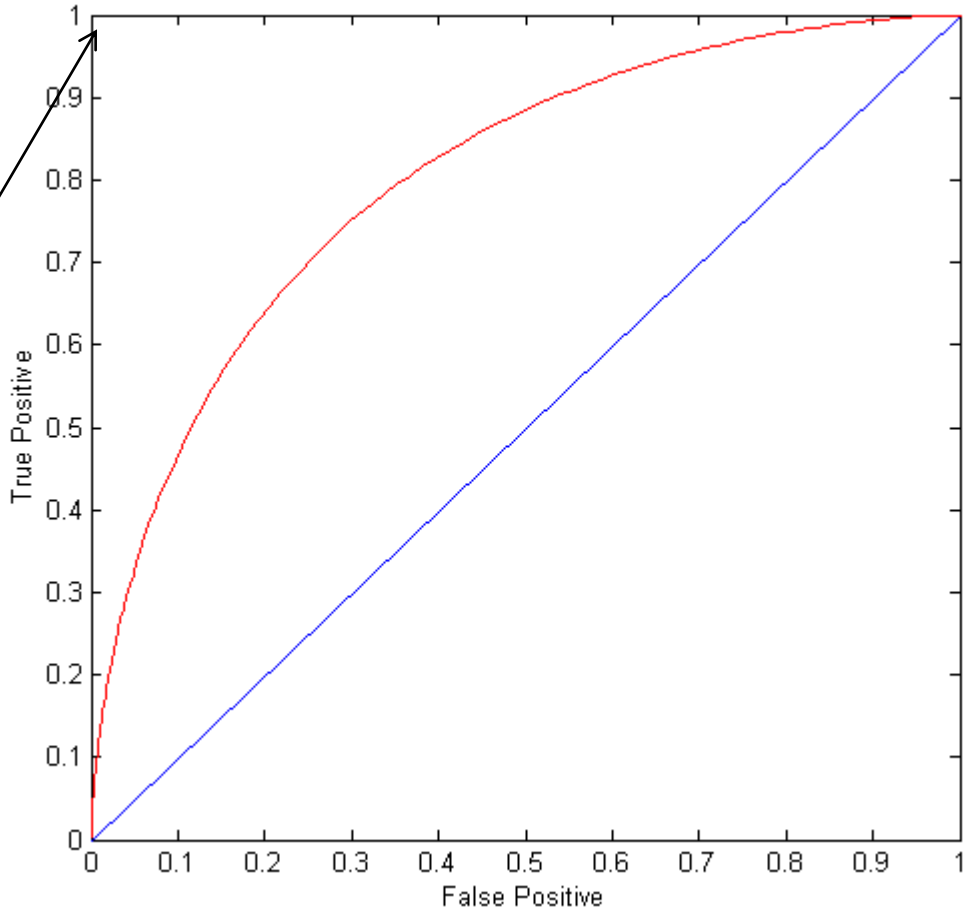
Curva Receiver Operating Characteristic (ROC)

Rispetto a TP e FP :

TP:0, FP:0
Predice sempre «negativo»

TP:1, FP:1
Predice sempre «positivo»

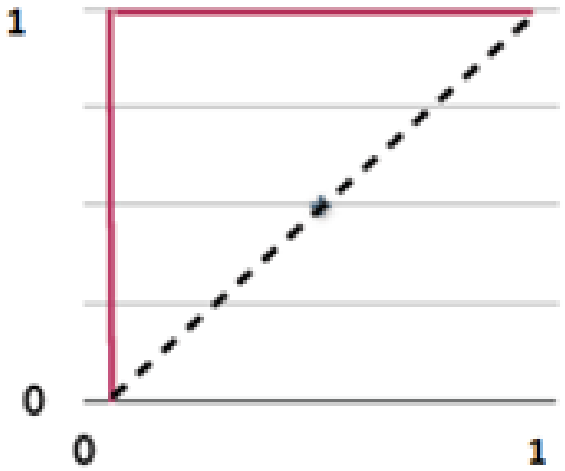
TP:1, FP:0
Classificatore ideale



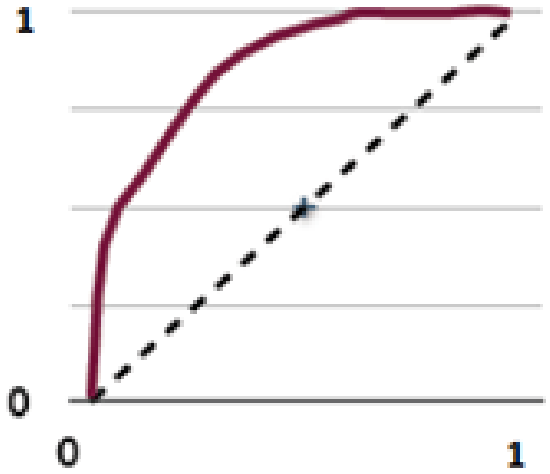
Curva ROC

AUC=1 AUC=0,8 AUC=0,5

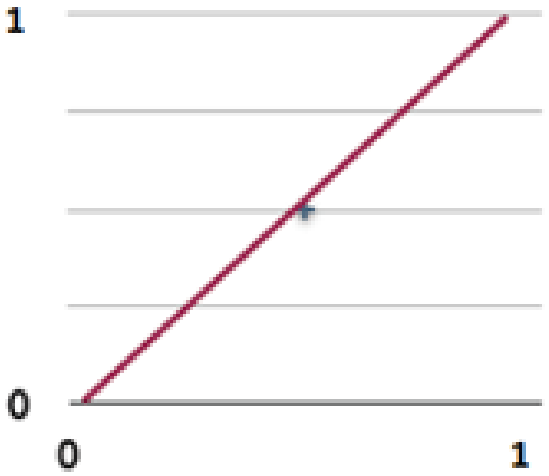
A migliore di B milgiore di C (lancio moneta)



A



B



C

Addestramento di un classificatore in R

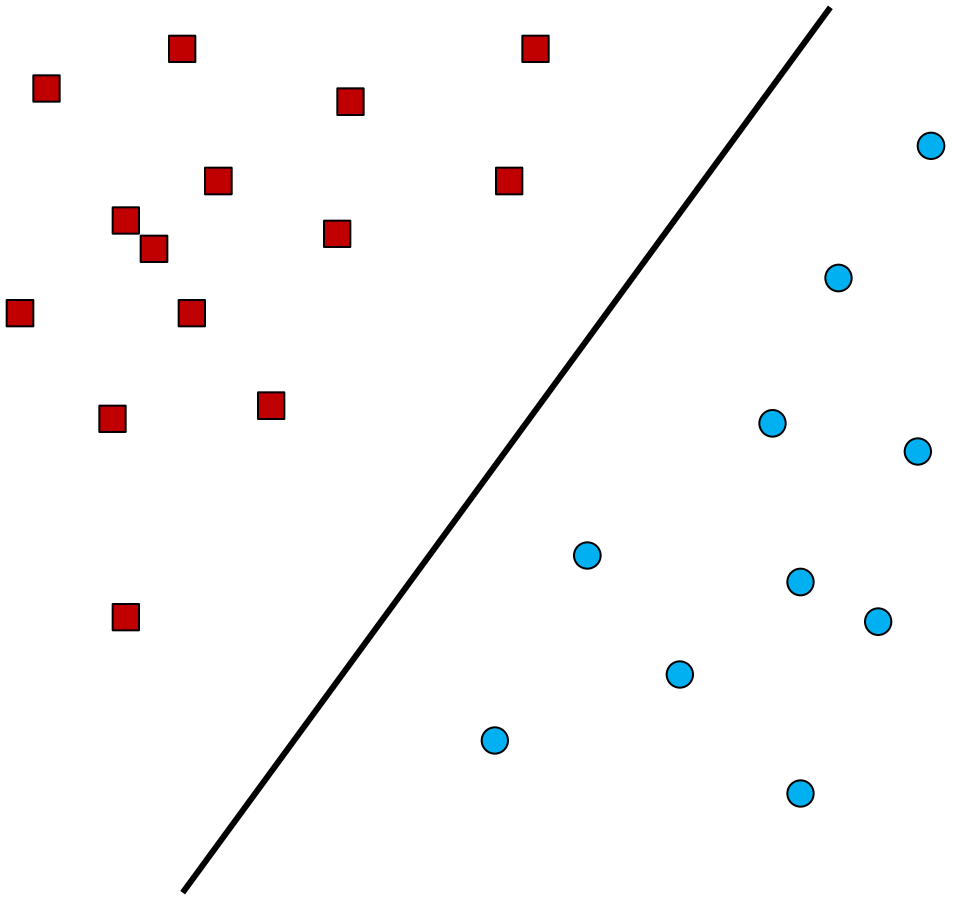
In questo esempio addestreremo un classificatore in R. Il tipo di classificatore che utilizzeremo è una

Support Vector Machine, o SVM (macchina a vettori di supporto)

Una implementazione di questo metodo è disponibile nella libreria **e1071** di R (installatela).

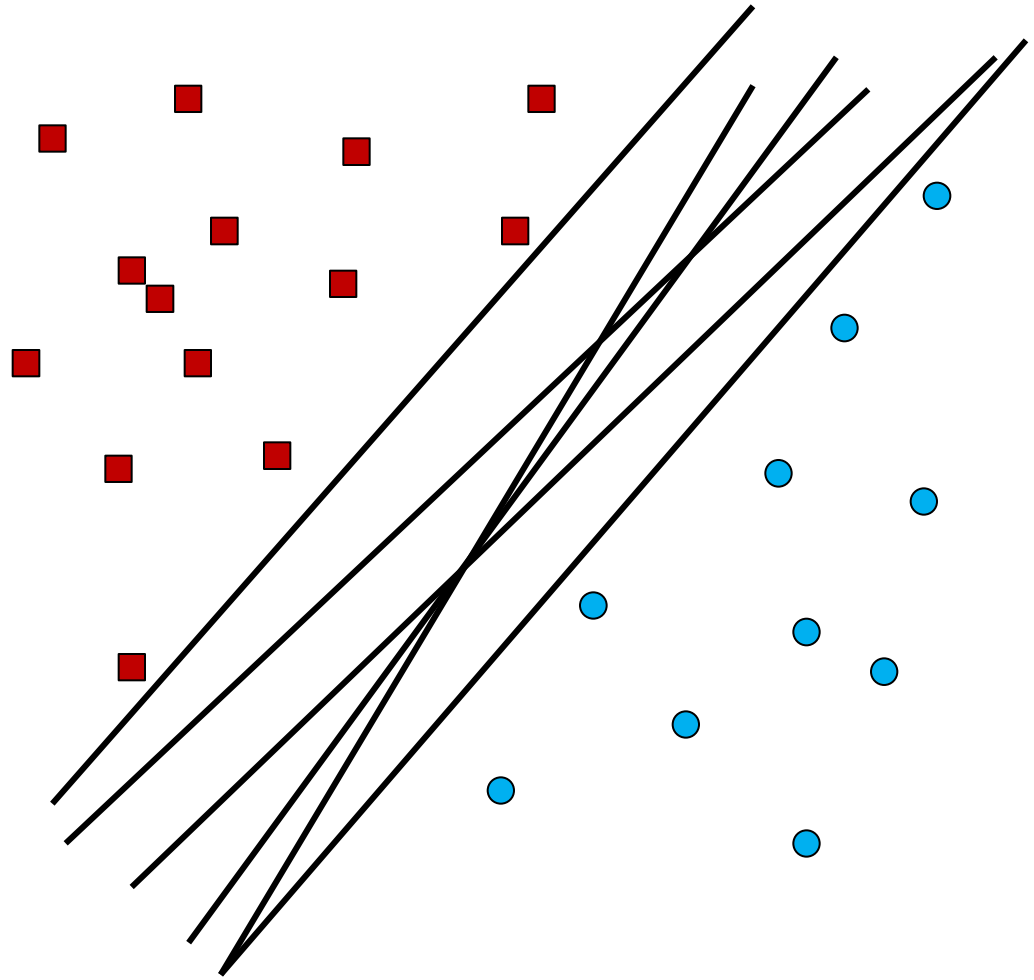
Support Vector Machine (SVM)

Classificatore che cerca di separare due classi di esempi mediante un piano di separazione (in 2D, una retta)



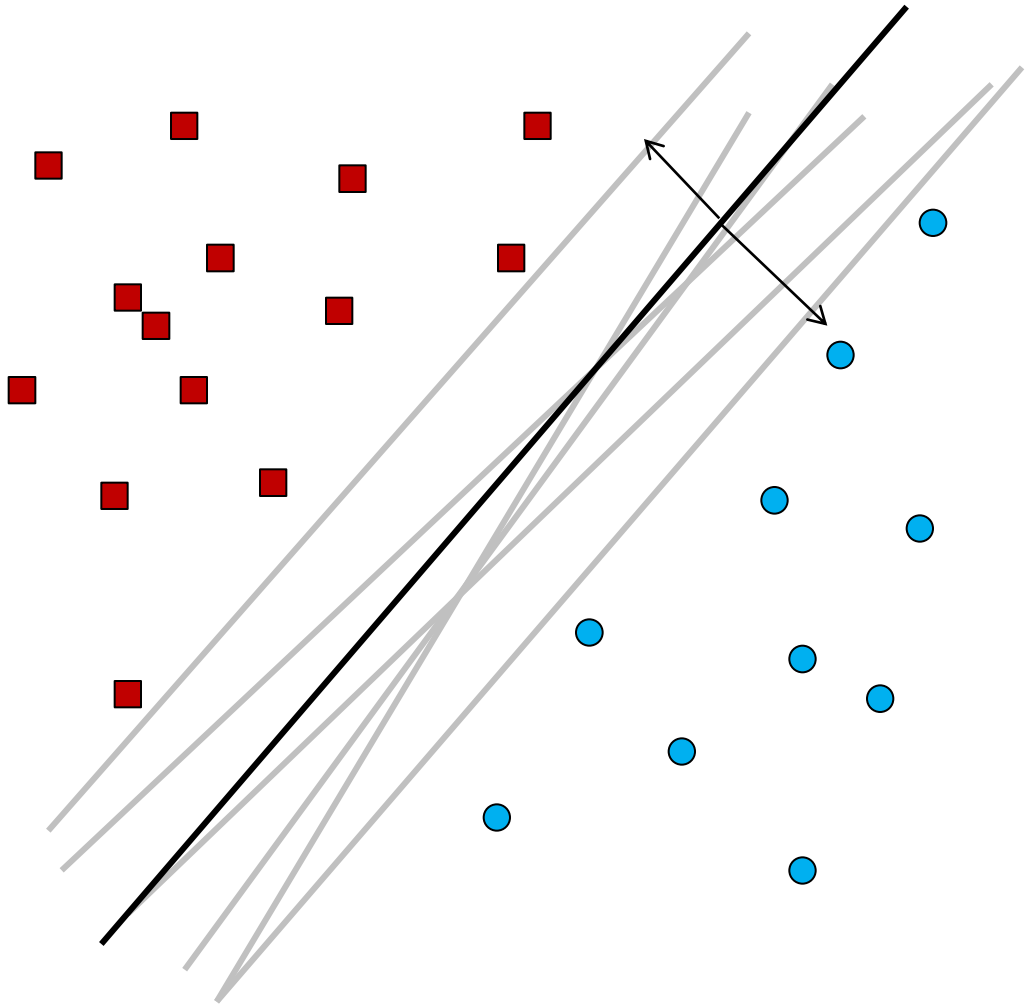
Support Vector Machine (SVM)

Ci sono diverse opzioni (rette) in grado di separare in modo corretto i punti … quale sarà la retta «migliore»?



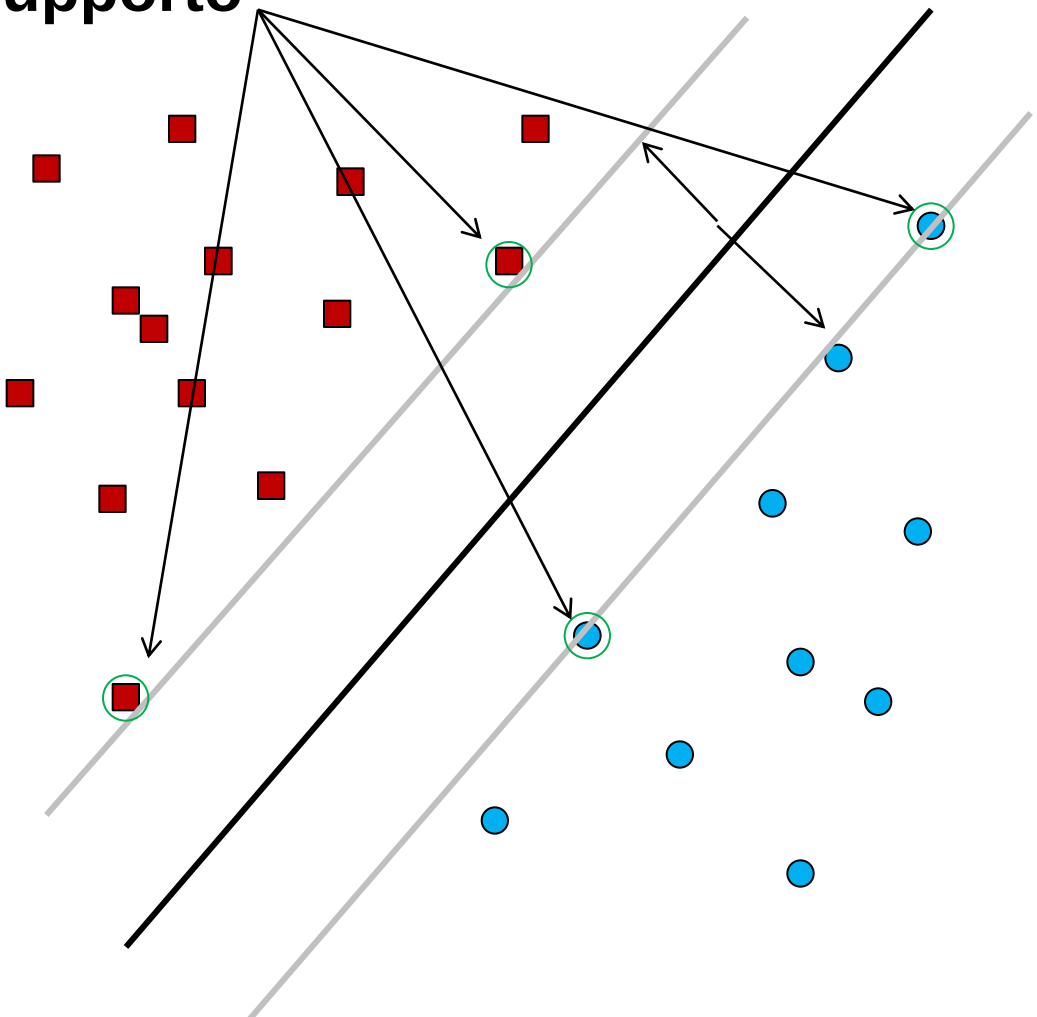
Support Vector Machine (SVM)

Quella che rende **più ampio il margine che separa le due classi**.



Support Vector Machine (SVM)

Questa retta è identificata da alcuni punti (esempi) detti **vettori di supporto**



Support Vector Machine (SVM)

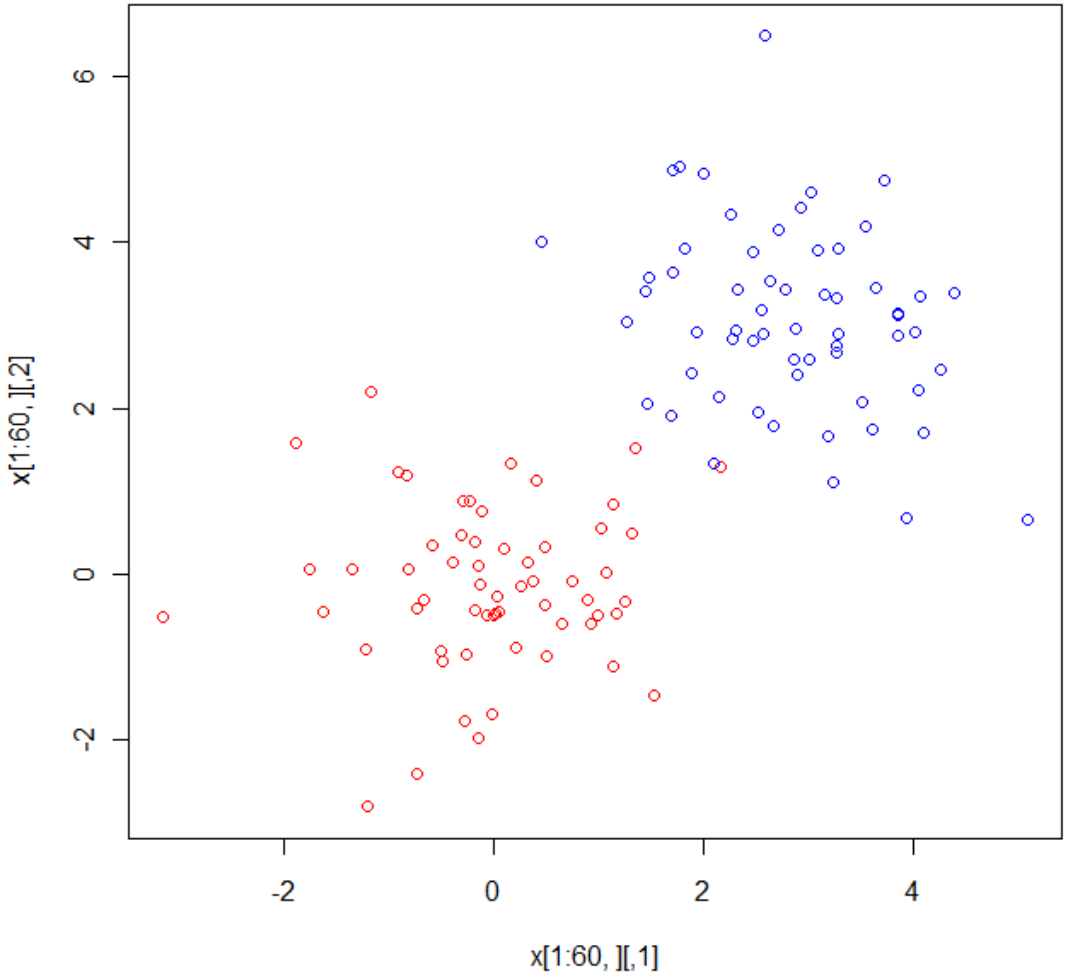
Razionale SVM:

- Devo addestrare su utilizzando una quantità finita di esempi
- Non so nulla rispetto a tutti gli altri esempi (punti) presenti nell'universo (ma dovrò cercare di predire correttamente la loro classe)
- La soluzione **MENO RISCHIOSA** è quella di creare un modello che rende massima la separazione tra gli esempi delle due classi che ho a disposizione (il che equivale a rendere **il più ampio possibile** il margine di separazione tra le classi)

Addestramento di un classificatore in R

```
# Creazione di un dataset sintetico:

data <- rbind(matrix(rnorm(120, mean=0), , 2),
matrix(rnorm(120,mean = 3), , 2))

# grafico del dataset:

plot(x[1:60,],col="red",
xlim=c(min(x[,1]),max(x[,1])),
ylim=c(min(x[,2]),max(x[,2])) )
points(x[61:120,],col="blue")

# creazione etichette:
labels <- matrix(c(rep(1, 60), rep(-1, 60)))
```

Addestramento di un classificatore in R



Addestramento di un classificatore in R

```
# UTILIZZO LIBRERIA e1071 per SVM:
library(e1071)

#Addestramento:

model <- svm(x=data, y=labels, type = "C-classification", kernel="linear", cost=10)

summary(model)
```

Addestramento di un classificatore in R

```
# Test:
# Creazione di un dataset sintetico:

data2 <- rbind(matrix(rnorm(120, mean=0), , 2),
matrix(rnorm(120,mean = 3), , 2))

# creazione etichette:
labels2 <- matrix(c(rep(1, 60), rep(-1, 60)))


#PREDIZIONE:
pred <- predict(model, data2)
pred2 <- predict(model, data2, decision.values = TRUE)
pred.numeric <- as.numeric(as.character(pred))

# matrice di confusione:
table(num.pred, labels2)
```

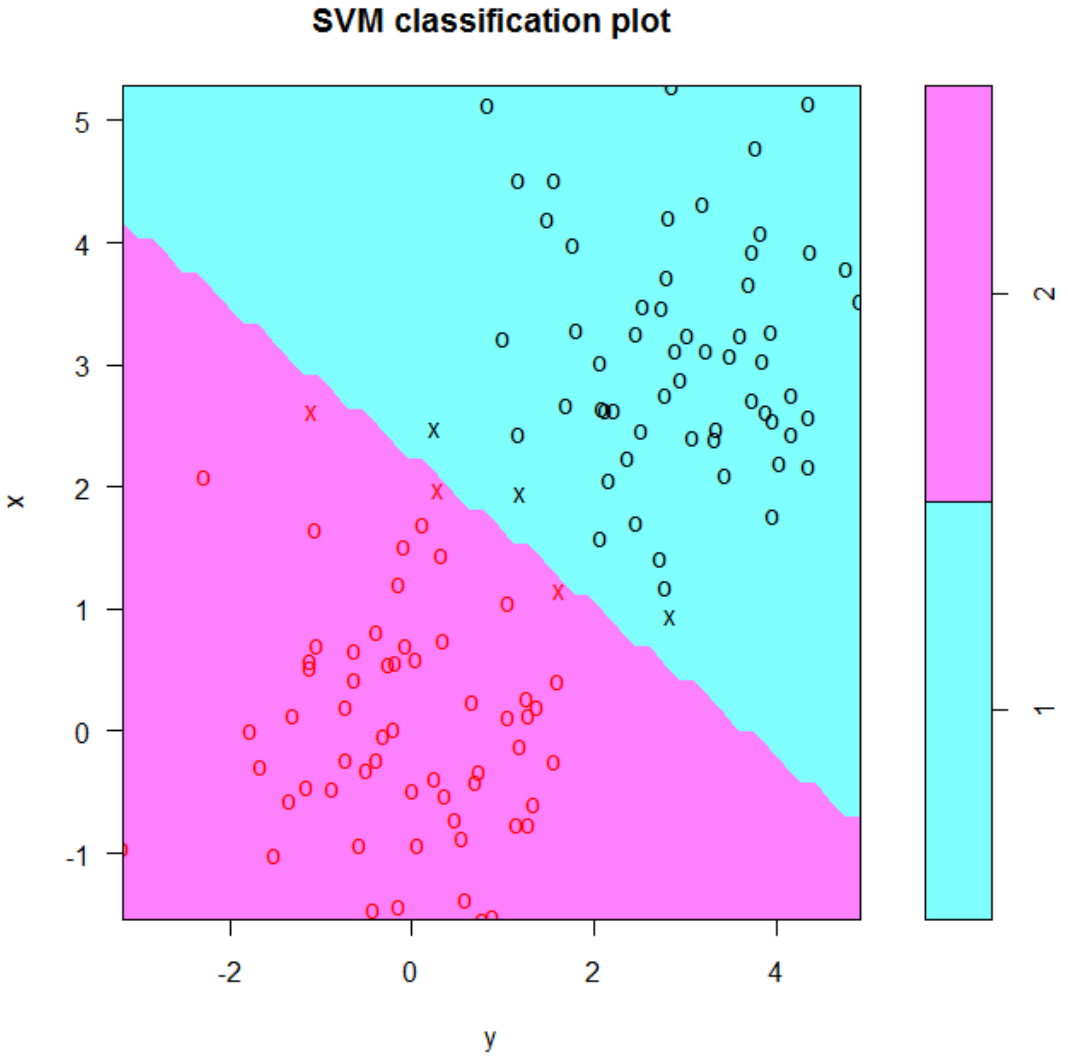
Addestramento di un classificatore in R

```
############## visualizzazione del modello addestrato
##############à

trainingdata <- cbind(data,as.factor(labels))
trainingdata<-as.data.frame(trainingdata)
colnames(trainingdata)<-c("x","y","dataLabels")
model3 <- svm(dataLabels ~ ., data = trainingdata,
type = "C-classification", kernel = "linear", cost =
10)

plot(model3,trainingdata)
```

Addestramento di un classificatore in R

```
############## visualizzazione del modello addestrato
##############à

trainingdata <- cbind(data,as.factor(labels))
trainingdata<-as.data.frame(trainingdata)
colnames(trainingdata)<-c("x","y","dataLabels")
model3 <- svm(dataLabels ~ ., data = trainingdata,
type = "C-classification", kernel = "linear", cost =
10)

plot(model3,trainingdata)
```

Addestramento di un classificatore in R



Addestramento di un classificatore in R

```
# kernel radiale

trainingdata <- cbind(data,as.factor(labels))
trainingdata<-as.data.frame(trainingdata)
colnames(trainingdata)<-c("x","y","dataLabels")
model3 <- svm(dataLabels ~ ., data = trainingdata,
type = "C-classification", kernel = "radial", cost =
10)

plot(model3,trainingdata)
```

Addestramento di un classificatore in R