

---

ANNO ACCADEMICO 2024/2025

---

## Apprendimento Automatico

---

Teoria

Altair's Notes



---

DIPARTIMENTO DI INFORMATICA

---



<b>CAPITOLO 1</b>	<b>INTRODUZIONE</b>	<b>PAGINA 5</b>
1.1	Le basi del machine learning Tasks — 7 • Modelli — 8 • Features — 9	5
<b>CAPITOLO 2</b>	<b>TASKS</b>	<b>PAGINA 11</b>
2.1	Classificazione Roc Plots Properties — 14 • Più di un Classificatore per una Singola Feature. — 15	11
2.2	Scoring e ranking Loss Function — 17 • Ranking — 20	16
2.3	Stima Probabilistica Squared Error — 22	21
2.4	Oltre la Classificazione Binaria Regressione — 26 • Apprendimento non Supervisionato — 27 • Subgroup-discovery — 29 • Regole di Associazione — 29	23



# Premessa

## Licenza

Questi appunti sono rilasciati sotto licenza Creative Commons Attribuzione 4.0 Internazionale (per maggiori informazioni consultare il link: <https://creativecommons.org/licenses/by/4.0/>).



## Formato utilizzato

Box di "Concetto sbagliato":

### Concetto sbagliato 0.1: Testo del concetto sbagliato

Testo contenente il concetto giusto.

Box di "Corollario":

### Corollario 0.0.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

### Definizione 0.0.1: Nome delle definizioni

Testo della definizione.

Box di "Domanda":

### Domanda 0.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

### Esempio 0.0.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

**Box di "Note":**

**Note:-**

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

**Box di "Osservazioni":**

**Osservazioni 0.0.1**

Testo delle osservazioni. Le osservazioni sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive. A differenza delle note le osservazioni sono più specifiche.



# 1

## Introduzione

### 1.1 Le basi del machine learning

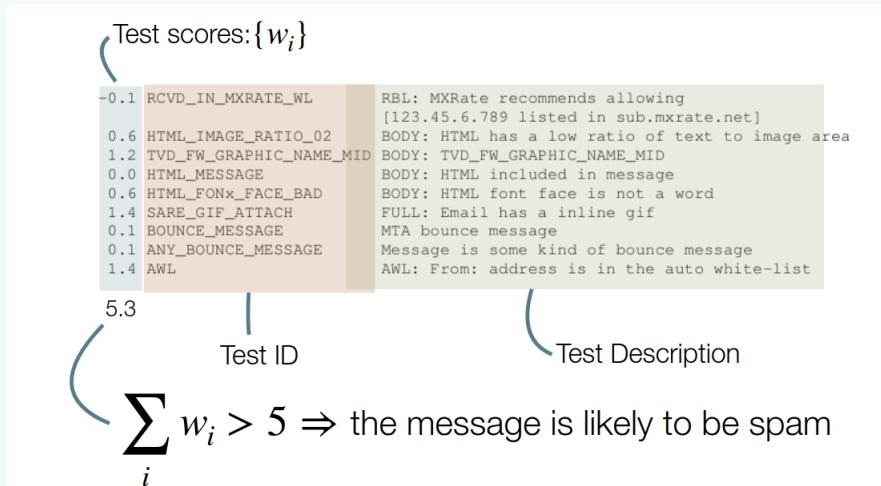
Gli ingredienti del machine learning:

- ⇒ *Task*: specifica di cosa si vuole fare;
- ⇒ *Modelli*: il modello matematico per affrontare un determinato task;
- ⇒ *Features*: il modo con cui sono descritti gli esempi.

**Note:-**

L'*apprendimento automatico* ruota attorno all'idea di estrarre una regola generale per risolvere un problema a partire da problemi già risolti.

#### Esempio 1.1.1 (Etichettatura delle email spam)



SpamAssassin è un filtro open-source usato per filtrare lo spam. Esso non lavora sul testo, ma su alcune *feature* della mail.

E-mail	$x_1$	$x_2$	Spam?	$4x_1 + 4x_2$
1	1	1	1	8
2	0	0	0	0
3	1	0	0	4
4	0	1	0	4

Discrimination rule example:  $\text{Spam}(x) = 4x_1 + 4x_2 > 5$

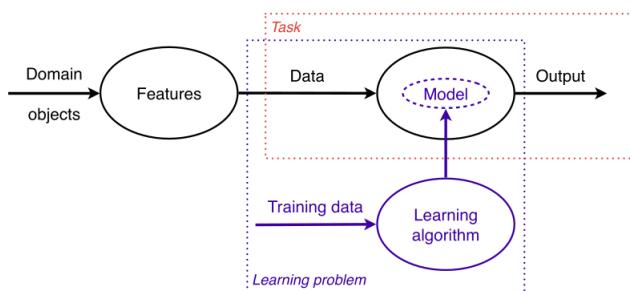
### Definizione 1.1.1: Apprendimento automatico

L'apprendimento automatico è lo studio sistematico di algoritmi e sistemi che migliorano le loro conoscenze e performance con l'esperienza.

L'apprendimento automatico è interessato a usare le giuste features per costruire il giusto modello per ottenere buone performance sul giusto task.

### Domanda 1.1

L'apprendimento automatico come può aiutarci a risolvere un task?



Dal dominio dell'applicazione arrivano degli oggetti descritti tramite features che vengono utilizzate per creare dei *training data* e un *dataset*. Questi vengono usati per costruire un modello per calcolare un output.

#### Note:-

Per risolvere un task bisogna sfruttare un modello. Per risolvere un problema di apprendimento bisogna trovare un algoritmo di apprendimento.

### 1.1.1 Tasks

#### Definizione 1.1.2: Tasks predittivi

Un task predittivo è focalizzato sul prevedere una variabile sulla base degli esempi. Si parte da problemi vecchi per trovare la soluzione a *nuovi* problemi.

#### Corollario 1.1.1 Overfitting

L'Overfitting è un adattamento eccessivo al dataset di allenamento per cui, messi di fronte a nuovi problemi, non si riesce a trovare una soluzione soddisfacente.

I tasks predittivi possono essere:

- *Binari e Multi-classe*: di categorizzazione.
- *Regressivi*: con un target numerico.
- *Clustering*: un target sconosciuto.

#### Note:-

IL Clustering fa anche parte dei tasks descrittivi.

#### Definizione 1.1.3: Tasks descrittivi

Un task descrittivo si concentra sul fornire regolarità nel dataset.

$$\begin{array}{c}
 \text{Films} \\
 \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \end{pmatrix} \\
 \text{Users} \\
 \begin{matrix} \text{The Shawshank Redemption} \\ \text{The Usual Suspects} \\ \text{The Godfather} \\ \text{The Big Lebowski} \end{matrix}
 \end{array}$$

Questa matrice rappresenta i voti dati da utenti a dei film. Si vogliono estrapolare le caratteristiche di questi film che hanno generato questi voti. Guardando questa matrice individualmente è difficile, per cui si compone con altre matrici.

$$\begin{array}{c}
 \text{Genres} \\
 \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 1 \\ 0 & 2 & 2 & 3 \end{pmatrix} = \text{Users} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 \begin{matrix} \text{drama} \\ \text{crime} \\ \text{comedy} \end{matrix} \\
 \begin{matrix} \text{The Shawshank Redemption} \\ \text{The Usual Suspects} \\ \text{The Godfather} \\ \text{The Big Lebowski} \end{matrix}
 \end{array}$$

### 1.1.2 Modelli

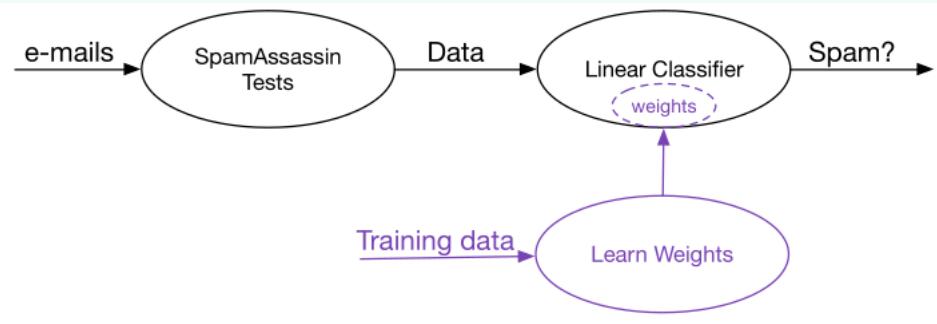
Ci sono 3 possibili tipi di modelli:

- *Geometrici*: modelli che usano l'intuizione dalla geometria per risolvere il problema;
- *Probabilistici*: usano il calcolo delle probabilità;
- *Logici*.

#### Definizione 1.1.4: Modelli geometrici

Nei modelli geometrici gli esempi sono punti di uno spazio vettoriale e la loro classificazione corrisponde a trovare un iperpiano che separi i punti positivi da quelli negativi.

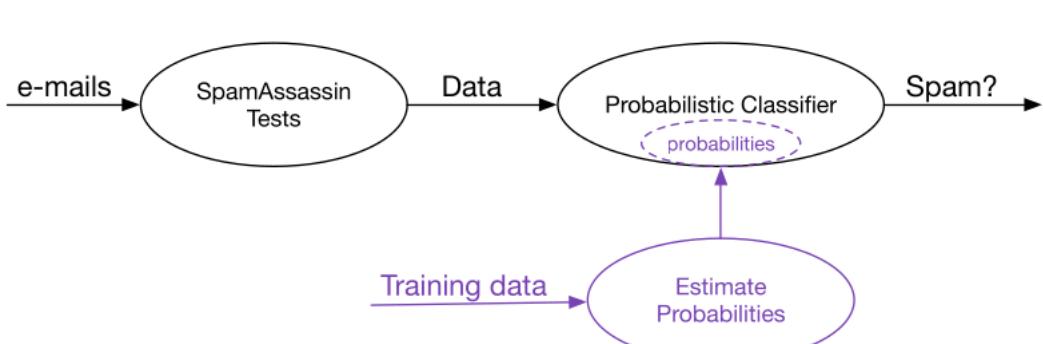
#### Esempio 1.1.2 (Modello geometrico)



#### Definizione 1.1.5: Modelli probabilistici

Nei modelli probabilistici si fanno delle stime con dei classificatori probabilistici. Dopo di che si usano delle regole di decisione.

#### Esempio 1.1.3 (Modello probabilistico)

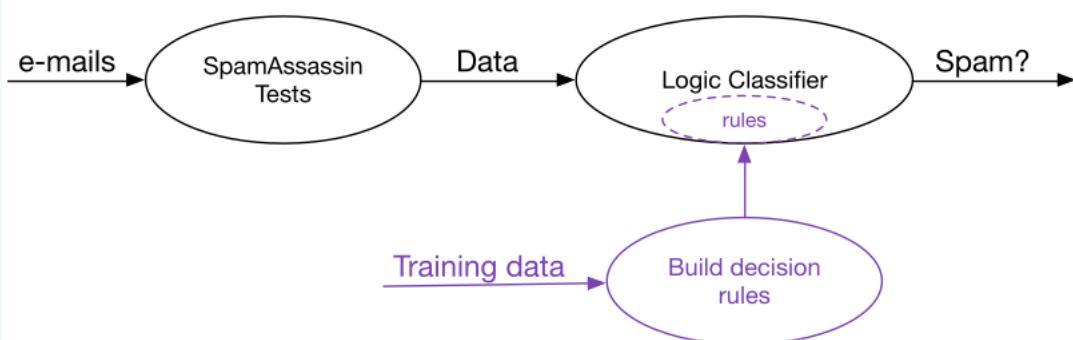


#### Note:-

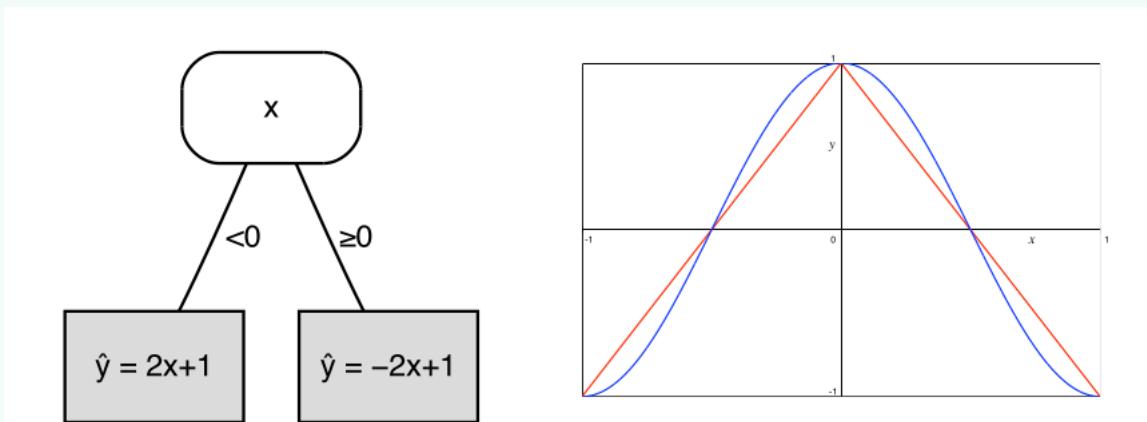
Uno degli algoritmi più semplici che si utilizza con i modelli probabilistici è l'assunzione di Naive Bayes. Si assume che  $x_1$  e  $x_2$  siano indipendenti tra loro per cui si possono calcolare solo i valori di  $x_1$  e di  $x_2$  individualmente.

**Definizione 1.1.6: Modelli logici**

Nei modelli logici si utilizza la logica. Si hanno una serie di regole.

**Esempio 1.1.4 (Modello logico)****1.1.3 Features****Definizione 1.1.7: Features**

Il modo in cui si descrivono i propri dati. Possono facilitare il lavoro di apprendimento se correttamente usate.

**Esempio 1.1.5 (Coseno)**

Due rappresentazioni della funzione coseno: a destra si utilizza una variabile di regressione, a destra un'approssimazione lineare.



# 2

## Tasks

I task più comuni sono:

- *Classificazione*.
- *Punteggio e classifica*.
- *Stima probabilistica*.
- *Regressione*.

### 2.1 Classificazione

#### Definizione 2.1.1: Classificazione

La classificazione è il task in cui si ha come obiettivo la costruzione di un modello  $\hat{c}: \mathbf{X} \rightarrow \mathbf{C}$  in cui  $\mathbf{C} = \{C_1, C_2, \dots, C_k\}$ . Questo modello è un'approssimazione del mondo reale.  
Un esempio è una coppia  $(x, c(x)) \in \mathbf{X} \times \mathbf{C}$ .

#### Osservazioni 2.1.1 Il problema dell'induzione

L'induzione partendo dai dati di un dataset è generalmente infondata senza ulteriori informazioni.

#### Note:-

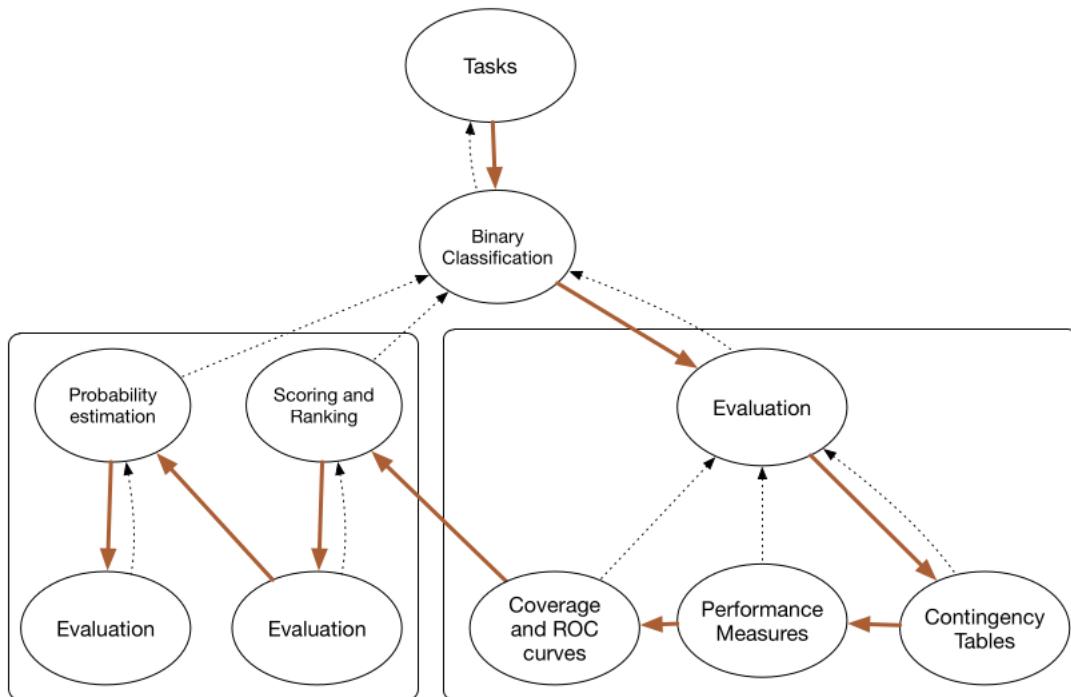
Il mondo non è semplice, per cui il rasoio di Occam non sempre funziona. Spesso però si utilizzano preconcetti e bias induttivi per avere apprendimento automatico.

#### Definizione 2.1.2: Classificazione binaria

La classificazione binaria è il caso in cui si hanno solo 2 opzioni (spesso 0 e 1).

#### Note:-

Dalla classificazione binaria si può passare alla classificazione multi-classe senza sviluppare nuovi algoritmi.

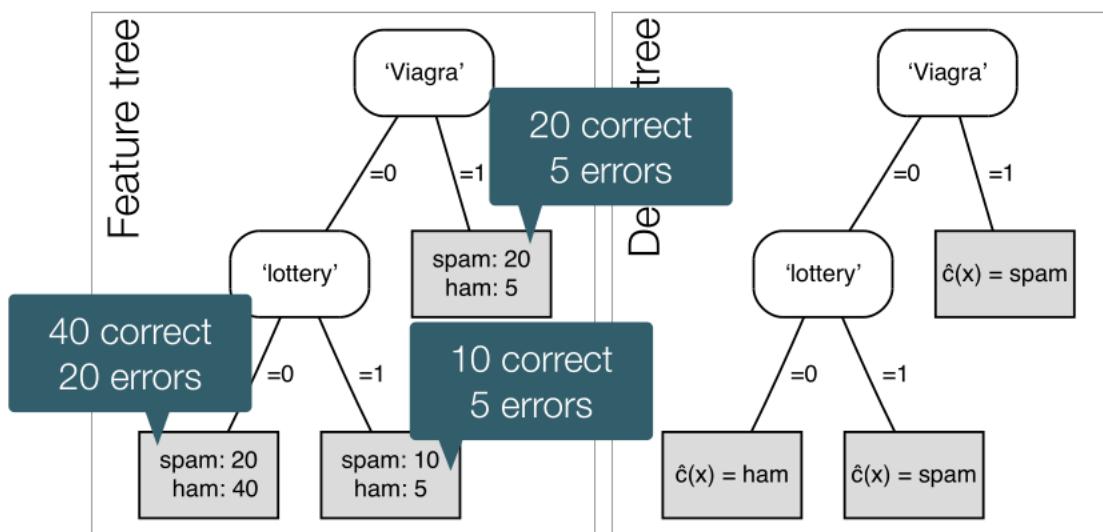


#### Definizione 2.1.3: Alberi di decisione

Alberi per visualizzare i dati. Ogni nodo corrisponde a una features.

#### Definizione 2.1.4: Alberi di Features

Alberi per visualizzare i dati. Si ha una suddivisione dei vari esempi divisi per etichette.



### Definizione 2.1.5: Tavola di contingenza

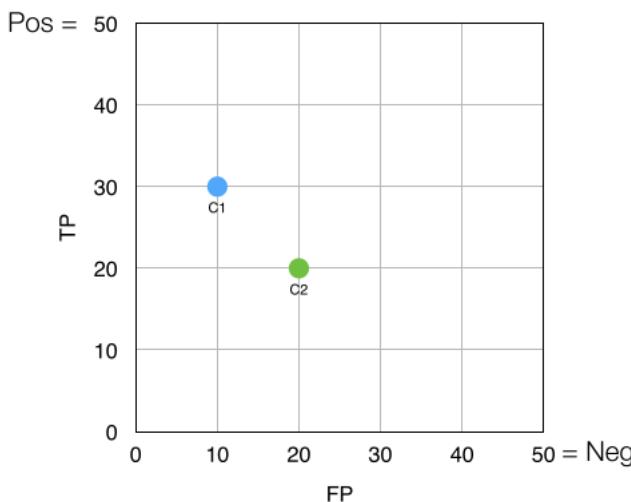
Tavola in cui le colonne corrispondono alle predizioni e le righe al mondo reale. Nella loro intersezione si ha il numero di esempi predetti in un certo modo e hanno una certa etichetta (TP, TN, FT, FN).

	<i>Predicted</i> $\oplus$	<i>Predicted</i> $\ominus$	
<i>Actual</i> $\oplus$	30	20	50
<i>Actual</i> $\ominus$	10	40	50
	40	60	100

### Definizione 2.1.6: Grafico di copertura

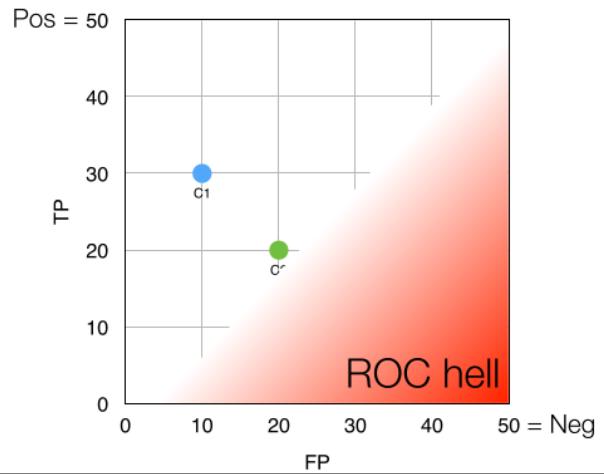
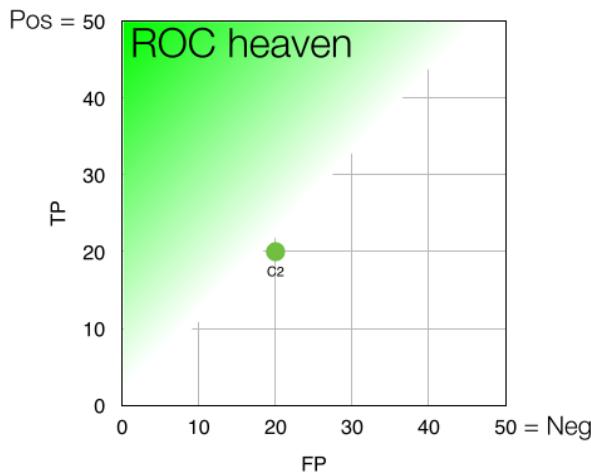
Grafico per visualizzare le informazioni della tavola di contingenza.

	<i>Predicted</i> $\oplus$	<i>Predicted</i> $\ominus$		<i>Predicted</i> $\oplus$	<i>Predicted</i> $\ominus$		
<i>Actual</i> $\oplus$	30	20	50	<i>Actual</i> $\oplus$	20	30	50
<i>Actual</i> $\ominus$	10	40	50	<i>Actual</i> $\ominus$	20	30	50
	40	60	100		40	60	100



**Note:-**

I classificatori che si trovano sulla bisettrice del piano cartesiano sono imprevedibili e quindi poco interessanti. Più un classificatore ha la coordinata x bassa e y alta più è preciso.

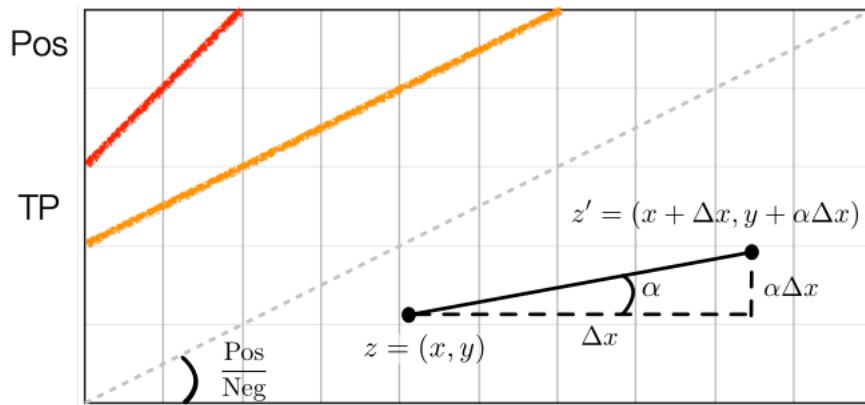
**Note:-**

Tutti i classificatori che stanno su una retta con pendenza 1 hanno la stessa *accuratezza*.

**Definizione 2.1.7: Avg recall**

$$\text{avg recall} = (\text{recall} + \text{specificity}) / 2 = (\text{TP}/\text{POS} + \text{TN}/\text{NEG})/2$$

Se due classificatori hanno la stessa avg recall allora sono su linee parallele alla diagonale principale.



$$\text{avgrec}(z) = \left( \frac{y}{\text{Pos}} + \frac{\text{Neg} - x}{\text{Neg}} \right) / 2$$

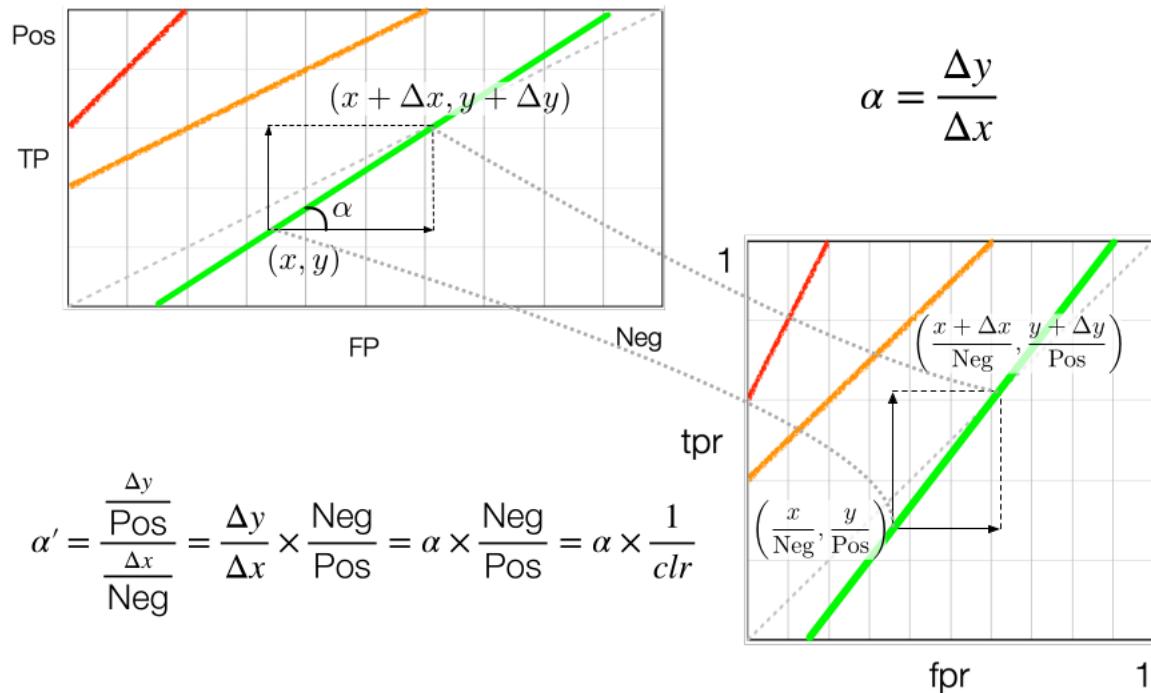
$$\text{avgrec}(z') = \left( \frac{y + \alpha\Delta x}{\text{Pos}} + \frac{\text{Neg} - x - \Delta x}{\text{Neg}} \right) / 2 = \left( \frac{y}{\text{Pos}} + \alpha \frac{\Delta x}{\text{Pos}} + \frac{\text{Neg} - x}{\text{Neg}} - \frac{\Delta x}{\text{Neg}} \right) / 2$$

$$z \text{ and } z' \text{ have the same avg-rec if and only if } \alpha = \frac{\text{Pos}}{\text{Neg}}$$

**2.1.1 Roc Plots Properties**

Se si vogliono confrontare le performance di un classificatore su un dataset o su un altro si deve *normalizzare* gli assi dividendo l'asse x per il numero di esempi negativi e l'asse y per il numero di esempi positivi. Così facendo

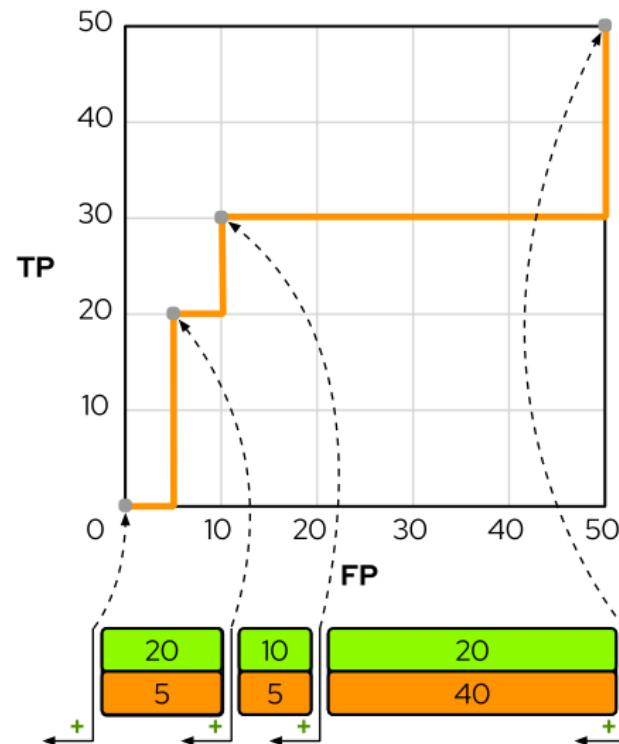
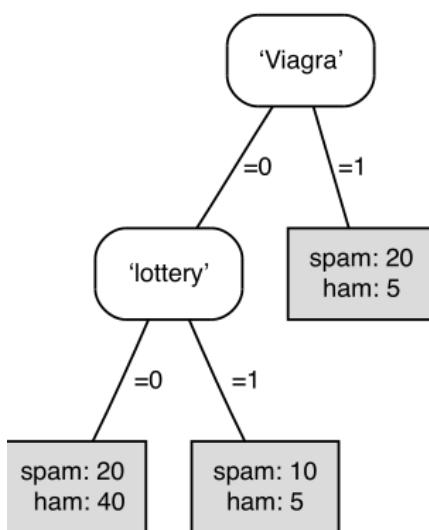
si otterrà un quadrato con gli assi compresi tra 0 e 1.



**Note:-**

Il clr è il class ratio.

### 2.1.2 Più di un Classificatore per una Singola Feature.



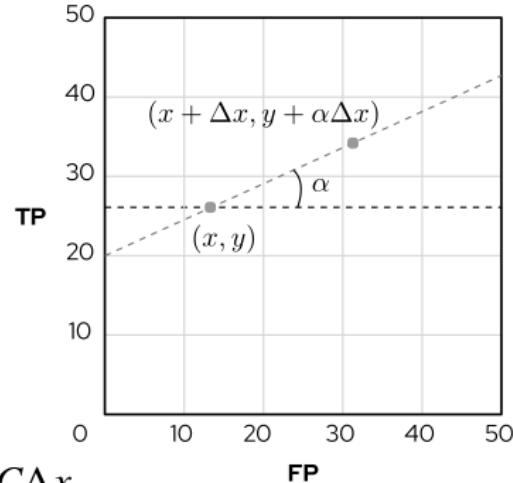
**Domanda 2.1**

Come si considera il caso in cui il costo per FP (falsi positivi) e FN (falsi negativi) sono differenti?

$$\text{acc}_C(x, y) = \frac{y + C \times \text{Neg} - Cx}{\text{Pos} + C \times \text{Neg}}$$

$$\begin{aligned}\text{acc}_C(x + \Delta x, y + \alpha \Delta x) &= \\ &= \frac{y + \alpha \Delta x + C \times \text{Neg} - C(x + \Delta x)}{\text{Pos} + C \times \text{Neg}} \\ &= \frac{y + C \times \text{Neg} - Cx}{\text{Pos} + C \times \text{Neg}} + \frac{\alpha \Delta x - C \Delta x}{\text{Pos} + C \times \text{Neg}}\end{aligned}$$

$$\text{if } \alpha = C, \text{ then: } \frac{\alpha \Delta x - C \Delta x}{\text{Pos} + C \times \text{Neg}} = \frac{C \Delta x - C \Delta x}{\text{Pos} + C \times \text{Neg}} = 0$$



In **ROC plots**, as is customary, one still needs to multiply by  $1/\text{clr}$ .

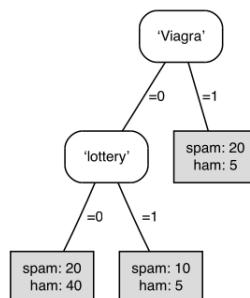
## 2.2 Scoring e ranking

### Definizione 2.2.1: Scoring classifier

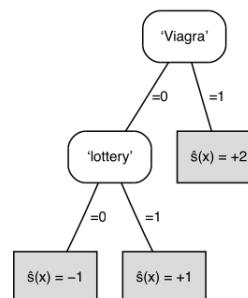
Uno scoring classifier è una mappatura  $\hat{s}: \mathbb{X} \rightarrow \mathbb{R}^k$  il cui output è un vettore  $(\hat{s}(x) = \hat{s}_1(x), \dots, \hat{s}_i(x))$  dove i-esimo componente è lo score assegnato alla classe  $C_i$  per l'istanza  $x$ .

#### Note:-

Se si hanno solo due classi si può considerare solo uno score. Gli score vanno interpretati nel contesto di un classificatore, sono misure della confidenza in una determinata predizione.



A **feature tree**  
with training set  
class distribution  
on the leaves



A **scoring tree**  
using the logarithm  
of the class ratio  
as scores

**Definizione 2.2.2: Margine**

Il margine assegnato dallo scoring classifier è positivo se  $\hat{s}$  è corretto, negativo altrimenti. Il margine è il prodotto tra la classe dell'esempio e lo score.

$$z(x) = c(x)\hat{s}(x) = \begin{cases} z(x) > 0 & \text{se la classificazione è corretta (cioè, } c(x) \text{ corrisponde alla classe prevista)} \\ z(x) < 0 & \text{se la classificazione è incorretta (cioè, } c(x) \text{ non corrisponde alla classe prevista)} \\ z(x) = 0 & \text{se lo score è esattamente al confine di decisione} \end{cases}$$

**2.2.1 Loss Function****Definizione 2.2.3: Loss Function**

La funzione di loss cerca di pesare l'impatto degli esempi negativi. In 0 la funzione di loss vale 1, tende a infinito con margini molto piccoli (molto negativi).

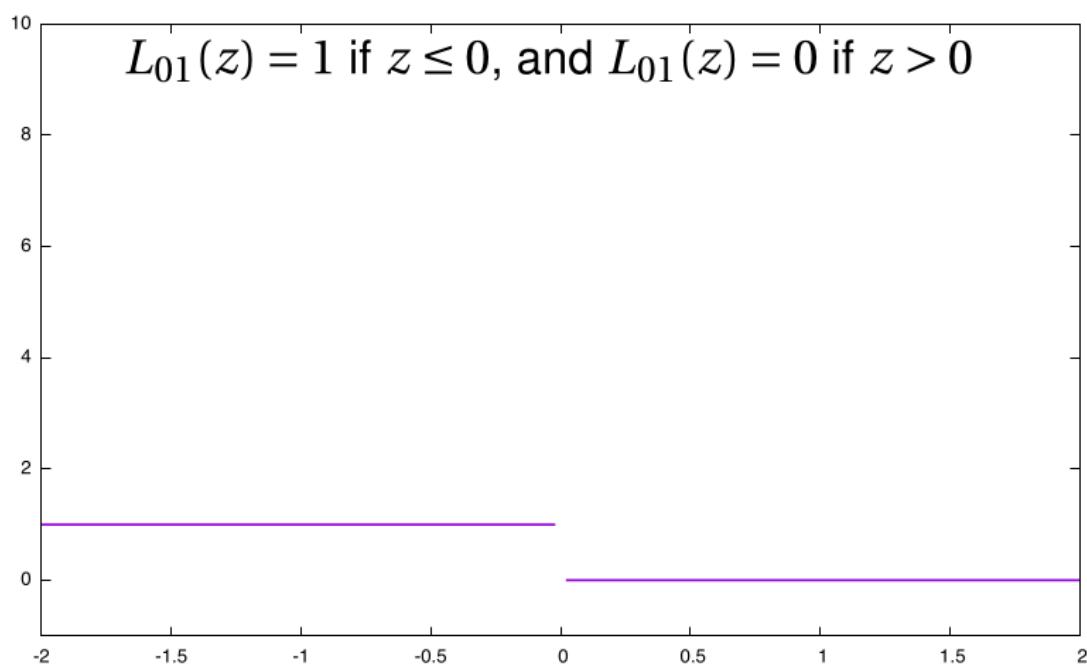
$$L : \mathbb{R} \rightarrow [0, \infty)$$

**Note:-**

Le loss function sono importanti durante l'apprendimento perché sono usate per guidare la ricerca della soluzione ottimale.

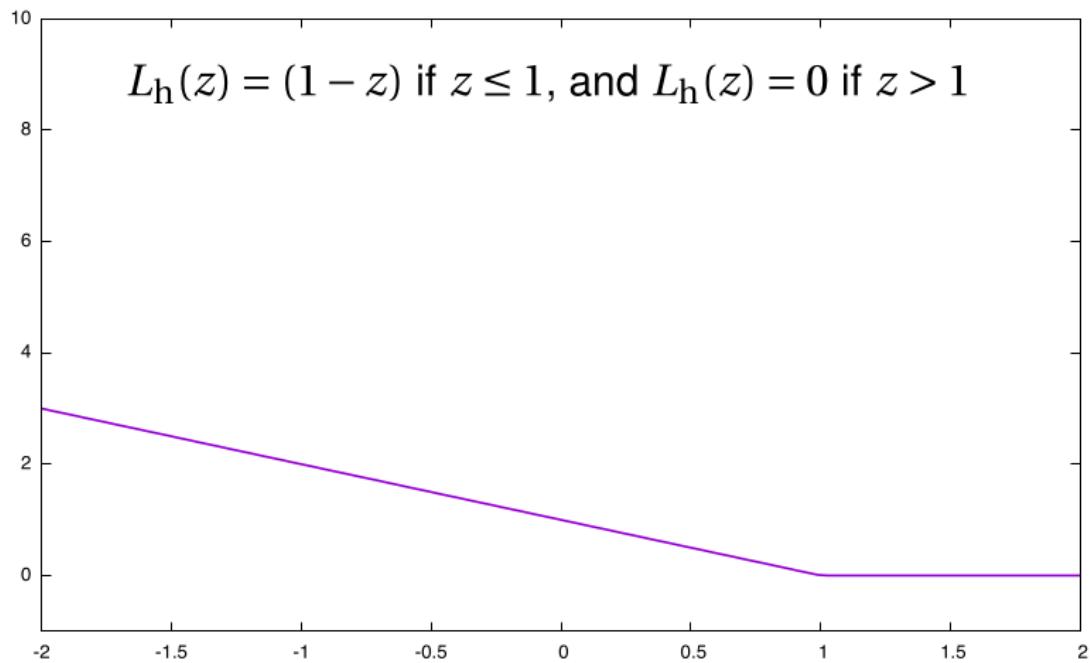
**Tipi di loss****Corollario 2.2.1 0-1 loss**

Si perde un'unità se si sbaglia e non si perde nulla se si indovina.

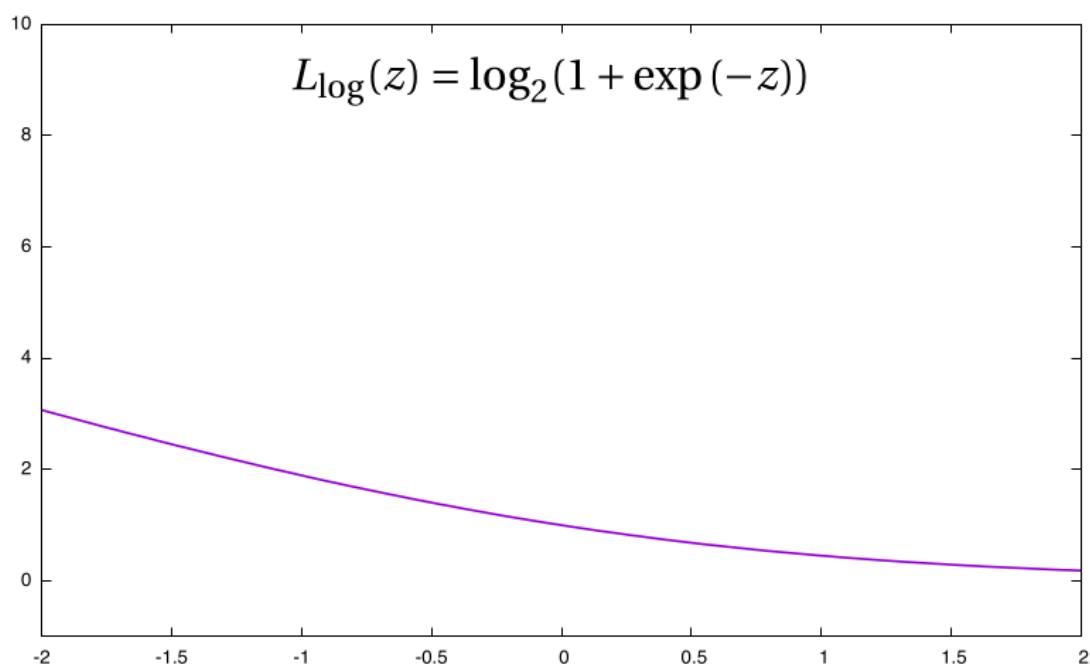


**Corollario 2.2.2 Hinge loss**

La Hinge loss è una loss che è lineare per valori minori di 1 e vale 0 per valori maggiori di 1.

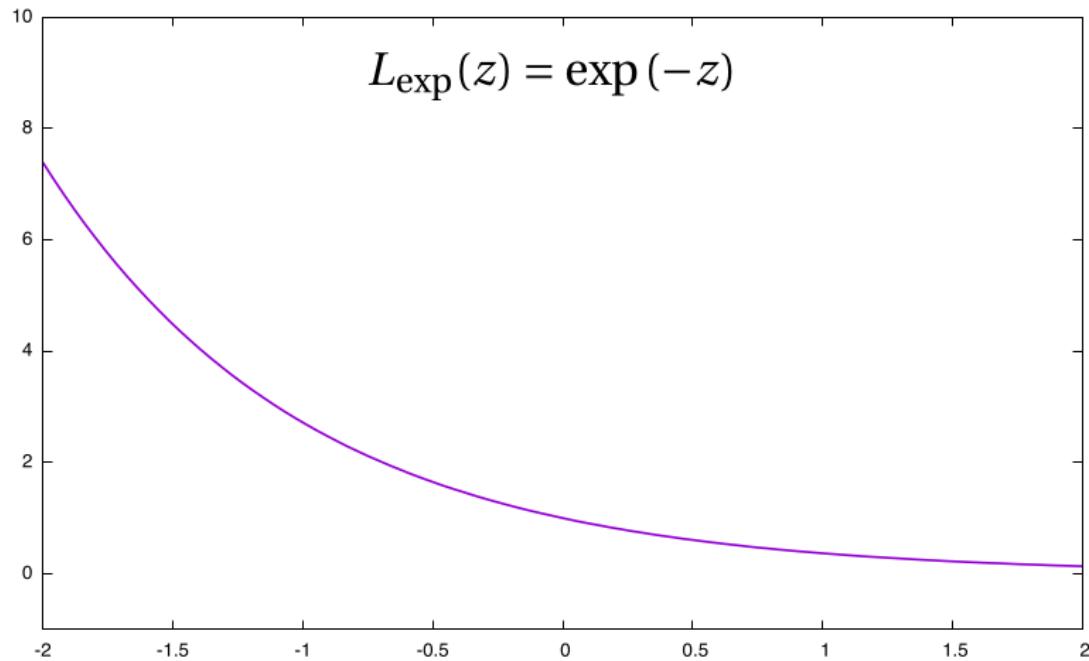
**Corollario 2.2.3 Logistic loss**

Approssimazione continua della Hinge loss.

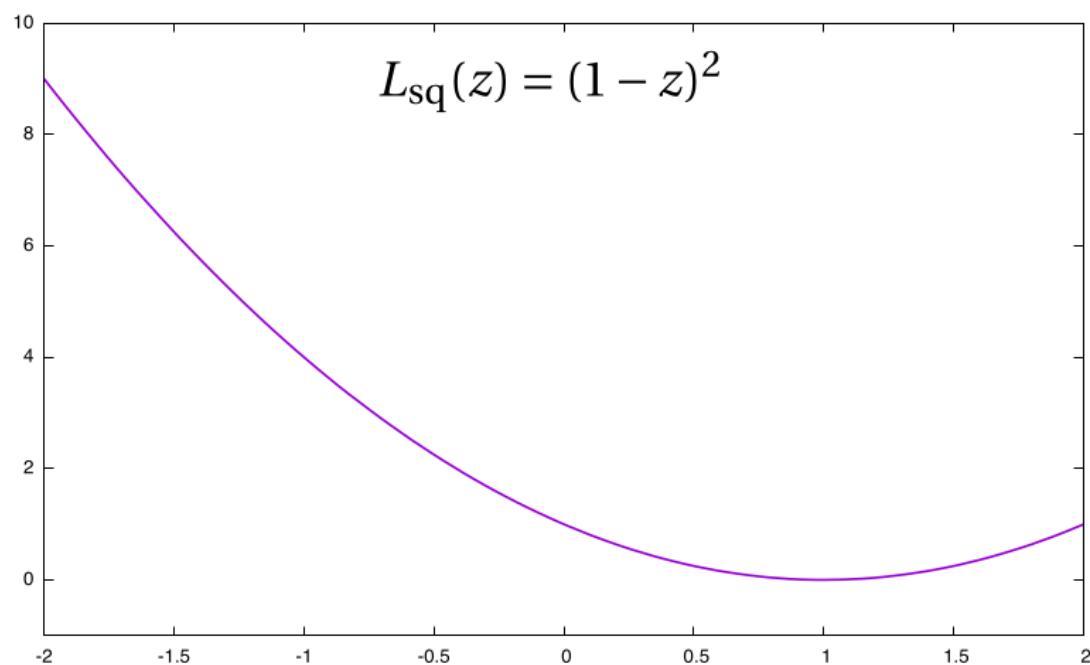


**Corollario 2.2.4 Loss esponenziale**

Cresce rapidamente quando si stanno facendo errori.

**Corollario 2.2.5 Loss quadratica**

Se viene ottimizzata troppo si hanno modelli incogniti, funziona meglio con la regressione.



### 2.2.2 Ranking

#### Definizione 2.2.4: Ranking

Ordina sulla base di uno score. Dall'esempio che è di classe più positiva a quello di classe meno positiva.

#### Corollario 2.2.6 Ranking Error Rate

The **ranking error rate** is defined as:

$$\text{rank-err} = \frac{\sum_{x \in Te^+, x' \in Te^-} I[\hat{s}(x) < \hat{s}(x')] + \frac{1}{2} I[\hat{s}(x) = \hat{s}(x')]}{Pos \cdot Neg}$$

1 point of penalty due to  
a ranking error: a  
positive example is  
ranked below a negative  
example

1/2 point of penalty for  
tying examples having  
different classes

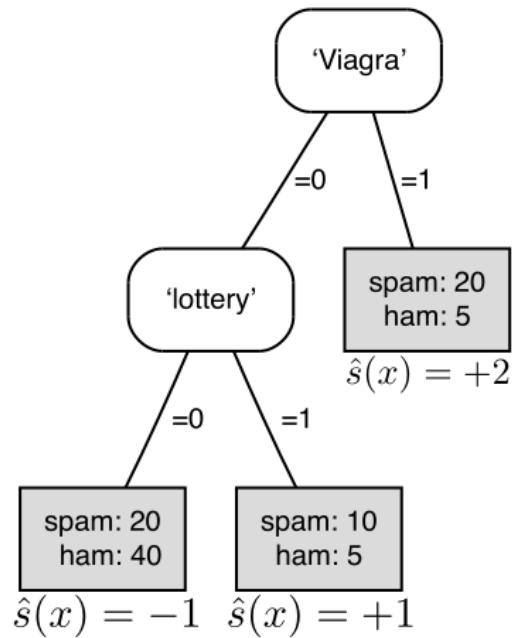
#### Esempio 2.2.1 (Ranking Error Rate)

$$\text{rank-err}(x_1^+, x_2^+, x_3^-, x_4^+, x_5^+, x_6^-, x_7^-, x_8^-) = \frac{2}{16} = \frac{1}{8}$$

$$\text{rank-err}(x_1^-, x_2^-, x_3^-, x_4^-, x_5^-, x_6^+, x_7^+, x_8^+) = \frac{15}{5 \times 3} = 1$$

**Esempio 2.2.2** (Spam)

- ◆ The 5 negatives in the right leaf are scored higher than the 10 positives in the middle leaf and the 20 positives in the left leaf, resulting in  $50 + 100 = 150$  ranking errors.
- ◆ The 5 negatives in the middle leaf are scored higher than the 20 positives in the left leaf, giving a further 100 ranking errors.
- ◆ In addition, the left leaf makes 800 half ranking errors (because 20 positives and 40 negatives get the same score), the middle leaf 50 and the right leaf 100.
- ◆ In total we have 725 ranking errors out of a possible  $50 \cdot 50 = 2500$ , corresponding to a ranking error rate of 29% or a ranking accuracy of 71%.



## 2.3 Stima Probabilistica

**Definizione 2.3.1:** Stimatore probabilistico di classi

Uno stimatore probabilistico di classi è un classificatore di scoring il cui output è un vettore di probabilità.

$$\hat{p} : \mathbb{X} \rightarrow [0, 1]^k$$

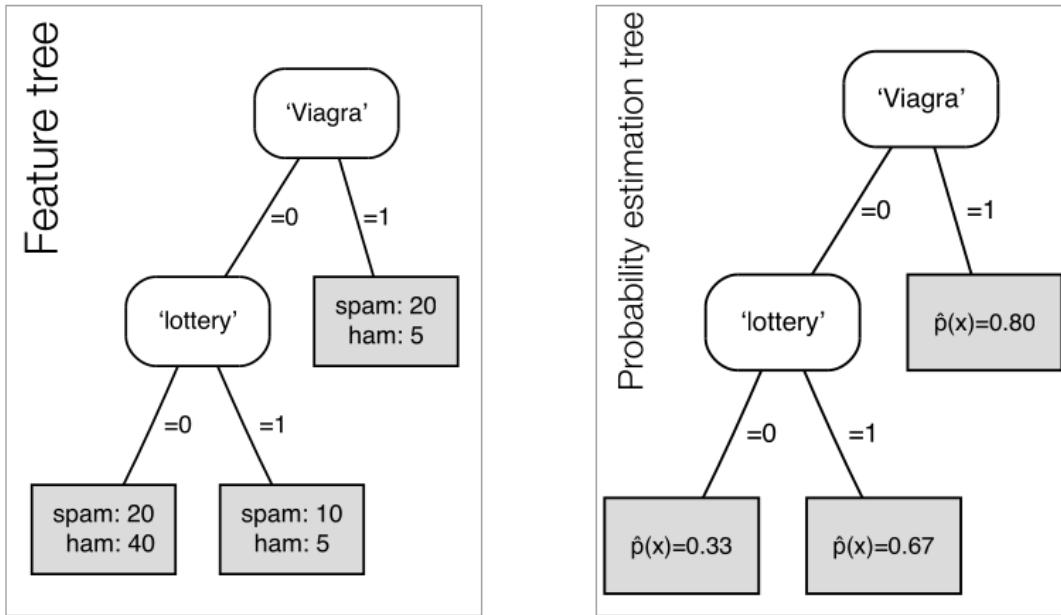
Scriviamo:

$$\hat{p}(x) = (\hat{p}_1(x), \dots, \hat{p}_k(x))$$

dove l'i-esimo componente è la probabilità assegnata alla classe  $C_i$  e  $\sum_{i=1}^k \hat{p}_i(x) = 1$

**Note:-**

Se si hanno solo 2 classi allora  $\hat{p}(x)$  denota la probabilità stimata per le classi positive.



### 2.3.1 Squared Error

#### Corollario 2.3.1 Squared Error

The **squared error** (SE) of the predicted probability vector on an example  $x$  is defined as:

$$\begin{aligned} \text{SE}(x) &= \frac{1}{2} \|\hat{\mathbf{p}}(x) - I_{c(x)}\|_2^2 \\ &= \frac{1}{2} \sum_{i=1}^k (\hat{p}_i(x) - I[c(x) = C_i])^2 \end{aligned}$$

where  $I_{c(x)}$  is a vector having 1 in the position corresponding to label  $c(x)$  and 0 in all other positions.

#### Corollario 2.3.2 Mean Squared Error

Il mean squared error è la media aritmetica degli squared error.

#### Definizione 2.3.2: Probabilità Empiriche

Le probabilità empiriche consentono di ottenere probabilità stimate da classificatori o rankers. Se si ha un insieme  $S$  di esempi etichettati e il numero di esempi in  $S$  di classe  $C_i$  è scritto  $n_i$  il vettore di probabilità empiriche associato ad  $S$  sarà:

$$\hat{\mathbf{p}}(S) = (n_1/|S|, \dots, n_k/|S|)$$

**Corollario 2.3.3 Correzione di Laplace**

Se si ha un insieme  $S$  e dentro si hanno  $n_i$  elementi di classe  $C_i$  per ogni classe si fa finta di avere un esempio aggiuntivo.

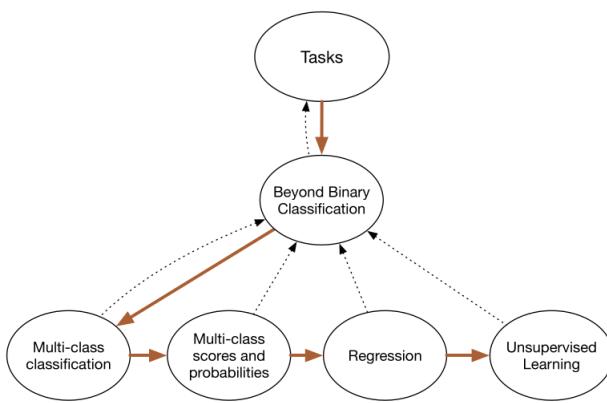
$$\hat{p}_i(S) = \frac{n_i + 1}{|S| + k}$$

Si può applicare anche:

$$\hat{p}_i(S) = \frac{n_i + m * \pi_i}{|S| + m}$$

La Correzione di Laplace è un caso speciale in cui  $m = k$  e la distribuzione è uniforme ( $\pi_i = \frac{1}{k}$ )

## 2.4 Oltre la Classificazione Binaria



**Schemi per estendere la classificazione binaria al caso multi-classe:**

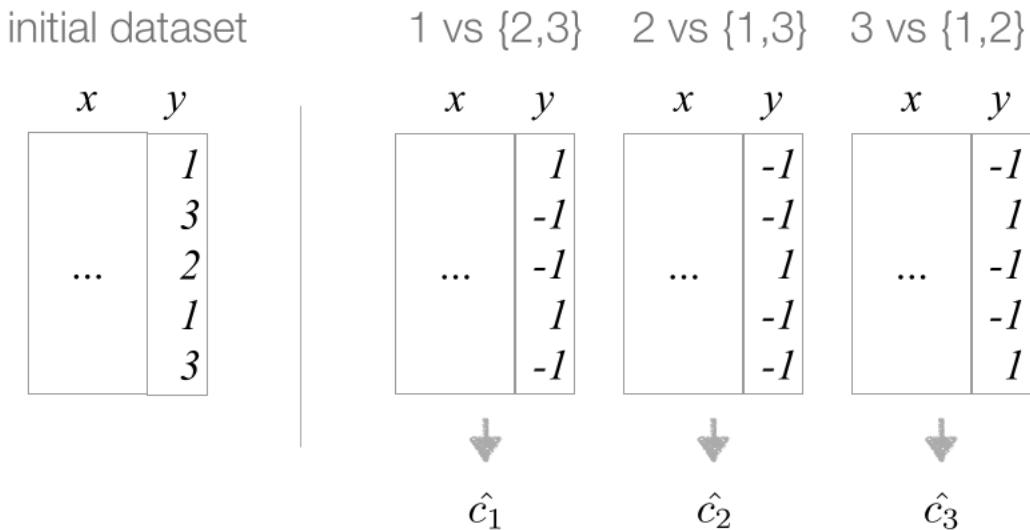
- one-vs-rest:

- apprendimento non ordinato;
- apprendimento in ordine fisso.

- one-vs-one:

- simmetrici;
- asimmetrici.

## One-vs-Rest (non ordinato)

Train  $k$  classifiers:**Note:-**

Si può costruire una matrice con il codice di output: in ogni riga si mette una classe, in ogni colonna si mette un classificatore che si vuole costruire e in ogni cella il valore che si vuole in output.

$$\begin{array}{ccc}
 & \begin{matrix} 1 & 2 & 3 \end{matrix} \\
 & \begin{matrix} vs \\ \{2,3\} \end{matrix} \quad \begin{matrix} vs \\ \{1,3\} \end{matrix} \quad \begin{matrix} vs \\ \{1,2\} \end{matrix} \\
 C_1 & \begin{pmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \end{pmatrix} \\
 C_2 & \\
 C_3 &
 \end{array}$$

## One-vs-Rest (ordinato)

$$\begin{array}{ccc}
 & \begin{matrix} 1 & 2 \end{matrix} \\
 & \begin{matrix} vs \\ \{2,3\} \end{matrix} \quad \begin{matrix} vs \\ \{3\} \end{matrix} \\
 C_1 & \begin{pmatrix} +1 & 0 \\ -1 & +1 \\ -1 & -1 \end{pmatrix} \\
 C_2 & \\
 C_3 &
 \end{array}$$

**One-vs-one (simmetrico)**

$$\begin{array}{ccc}
 & 1 & 1 & 2 \\
 & vs & vs & vs \\
 & 2 & 3 & 3 \\
 C_1 & \left( \begin{array}{ccc} +1 & +1 & 0 \\ -1 & 0 & +1 \\ 0 & -1 & -1 \end{array} \right) \\
 C_2 & \\
 C_3 &
 \end{array}$$

**One-vs-one (asimmetrico)**

$$\begin{array}{cccccc}
 & 1 & 2 & 1 & 3 & 2 & 3 \\
 & vs & vs & vs & vs & vs & vs \\
 & 2 & 1 & 3 & 1 & 3 & 2 \\
 C_1 & \left( \begin{array}{cccccc} +1 & -1 & +1 & -1 & 0 & 0 \\ -1 & +1 & 0 & 0 & +1 & -1 \\ 0 & 0 & -1 & +1 & -1 & +1 \end{array} \right) \\
 C_2 & \\
 C_3 &
 \end{array}$$

**Note:-**

Per classificare un nuovo esempio (vettore) si cerca la riga più simile.

**Osservazioni 2.4.1** Difficoltà nell'applicare one-vs-rest e one-vs-one

- Nel caso one-vs-rest il singolo classificatore vede un dataset molto sbilanciato sebbene il dataset di partenza fosse bilanciato.
- Nel caso one-vs-one il problema è mitigato assegnando 0 agli esempi che non appartengono alle due etichette scelte. Però è problematico quando si ha scarsità nei dati.

**Domanda 2.2**

Come si rompono i pareggi?

1. Si aggiungono classificatori.

2. Se si ha un algoritmo di apprendimento in grado di assegnare uno score si può scegliere il valore su cui si è più confidenti.

### 2.4.1 Regressione

#### Definizione 2.4.1: Stimatore di funzione

Uno stimatore di funzione, chiamato *regressore*, è una mappatura:

$$\hat{f} : \mathbb{X} \rightarrow \mathbb{R}$$

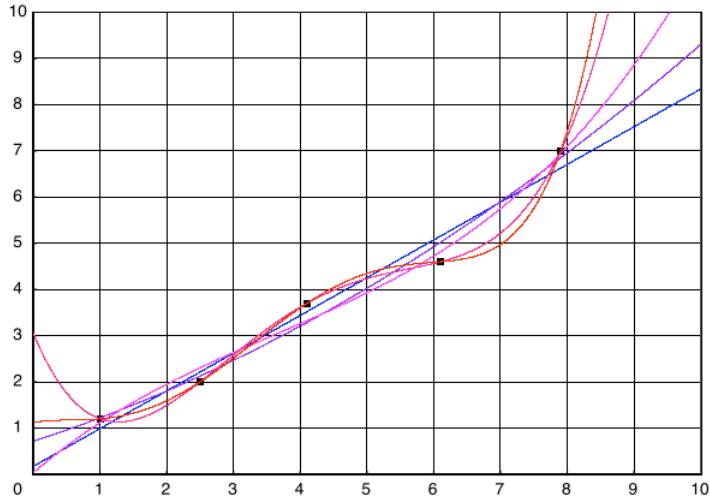
Il problema dell'apprendimento con la regressione è imparare uno stimatore di funzione dagli esempi  $(x_i, f(x_i))$

#### Note:-

In questo caso, a differenza dei precedenti, si passa a etichette di tipo reale (infinito non numerabile).

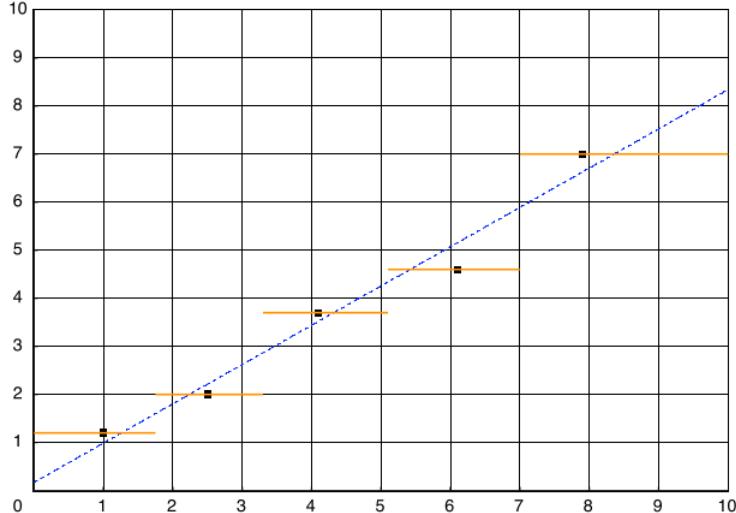
#### Funzione continua

$x$	$y$
1.0	1.2
2.5	2.0
4.1	3.7
6.1	4.6
7.9	7.0



#### Funzione costante a tratti

$x$	$y$
1.0	1.2
2.5	2.0
4.1	3.7
6.1	4.6
7.9	7.0



**Note:-**

Se si vuole "fittare"  $n + 1$  punti si può utilizzare un polinomio di grado  $n$  (quindi con  $n + 1$  parametri). Per evitare l'overfitting il numero dei parametri stimati dai dati deve essere molto minore del numero dei punti.

**Definizione 2.4.2: Bias-Variance Dilemma**

Un modello a bassa complessità si ha bassa variabilità a causa della variazione casuale nei dati. Però viene introdotto un bias sistematico che nemmeno un modello più grande può risolvere.

Un modello ad alta complessità elimina questo bias, ma è soggetto a errori dovuti alla varianza.

$$E[(f - \hat{f})^2] = (f - E[\hat{f}])^2 + E[(\hat{f} - E[\hat{f}])^2] = \text{Bias}^2(\hat{f}) + \text{Var}(\hat{f})$$

- $(f - E[\hat{f}])^2$ , zero se il regressore è mediamente corretto, altrimenti ha un bias sistematico.
- $E[(\hat{f} - E[\hat{f}])^2]$ , errore dovuto alla fluttuazione attorno alla media (errore dovuto alla varianza).

$$\begin{aligned} E[(f - \hat{f})^2] &= E[f^2 - 2f\hat{f} + \hat{f}^2] \\ &= f^2 - 2fE[\hat{f}] + E[\hat{f}^2] \\ &= f^2 - 2fE[\hat{f}] + E[\hat{f}^2] - E[\hat{f}]^2 + E[\hat{f}]^2 \\ &= E[\hat{f}^2] - E[\hat{f}]^2 + f^2 - 2fE[\hat{f}] + E[\hat{f}]^2 \\ &= E[(\hat{f} - E[\hat{f}])^2] + (f - E[\hat{f}])^2 \\ &= \text{Var}(\hat{f}) + \text{Bias}^2(\hat{f}) \end{aligned}$$

**Corollario 2.4.1 Bias**

Errore sistematico dovuto al fatto che il modello non è in grado di "fittare" perfettamente i dati (e.g. si sta cercando di "fittare" una parabola usando una retta).

**Corollario 2.4.2 Variance**

Errore per cui i modelli sono instabili (si ha del rumore). Anche con una minima variazione il modello può cambiare completamente.

**2.4.2 Apprendimento non Supervisionato****Definizione 2.4.3: Apprendimento non Supervisionato**

Nell'*apprendimento non supervisionato* non ci sono etichette associate ai dati. Il task consiste nel trovare *regolarità* nei dati usando solo i dati stessi.

**Note:-**

Un task non supervisionato è il Clustering che può essere sia predittivo che descrittivo.

### Definizione 2.4.4: Clustering

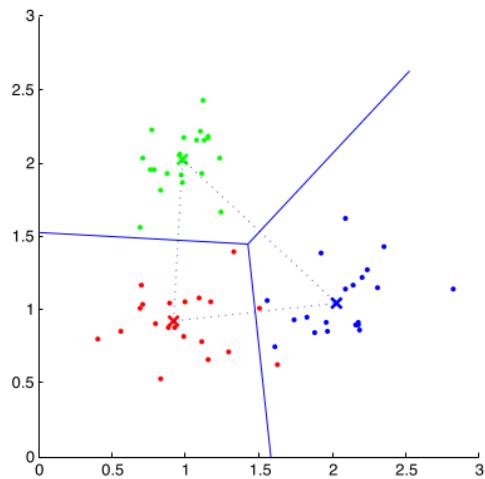
Il task del Clustering è guadagnare maggiore comprensione dei dati dividendo i dati stessi in gruppi tra loro omogenei.

#### Corollario 2.4.3 Clustering predittivo

Nel Clustering predittivo si può modellare il problema con la mappatura:

$$\hat{q} : \mathbb{X} \rightarrow \mathbb{C}$$

Dove  $\mathbb{C} = \{\mathbb{C}_1, \dots, \mathbb{C}_k\}$  è un nuovo insieme di etichette.

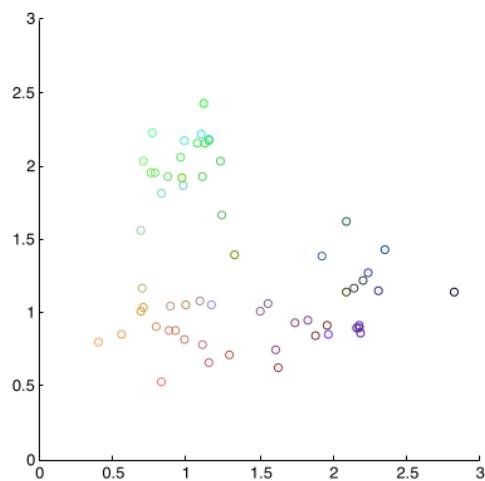


#### Corollario 2.4.4 Clustering descrittivo

Nel Clustering descrittivo si può modellare il problema con la mappatura:

$$\hat{q} : D \rightarrow \mathbb{C}$$

dove  $D$  è il dataset usato.



### 2.4.3 Subgroup-discovery

#### Definizione 2.4.5: Subgroup-discovery

Dato il dataset  $\{(x, l(x))\}$  si vuole trovare una funzione:

$$\hat{g} : D \rightarrow \{\text{true}, \text{false}\}$$

tale che  $G = \{x \in D | \hat{g}(x) = \text{true}\}$  ha una distribuzione di classe molto diversa dalla popolazione iniziale.

#### Note:-

$G$  è un *estensione* del sottogruppo.

#### Osservazioni 2.4.2 Subgroup-discovery

- In generale i Subgroup-discovery sono guidati da una valutazione delle misure;
- Tendono a favorire sottogruppi più grandi;
- Sono solitamente simmetrici (hanno lo stesso valore sia per il sottogruppo che per il suo completamento);
- Il risultato tenderà a dividere lo spazio in due parti uguali.

### 2.4.4 Regole di Associazione

#### Definizione 2.4.6: Regole di Associazione

Si ha un dataset non etichettato  $D$  e si vuole trovare un insieme di regole  $\{b \rightarrow h\}$  tale che l'oggetto  $b \cup h$  sia frequente e che  $h$  sia vera quando  $b$  è vera,

#### Note:-

$b$  e  $h$  sono insieme di coppie attributo/valore.

#### Esempio 2.4.1 (Supermercato)

If we set 0.6 as our support threshold (i.e., an itemset is frequent whenever it appears in 60% of the transactions). The following frequent itemsets can be extracted:

$\{\text{Bread}\}$  ( $\text{supp}: 0.8$ ),  $\{\text{Milk}\}$  ( $\text{supp}: 0.6$ ),  
 $\{\text{Water}\}$  ( $\text{supp}: 0.6$ ),  $\{\text{Bread, Milk}\}$  ( $\text{supp}: 0.6$ )

allowing one to generate the following rules:

$\text{Bread} \rightarrow \text{Milk}$  ( $\text{conf}: 0.6/0.8=0.75$ )

$\text{Milk} \rightarrow \text{Bread}$  ( $\text{conf}: 0.6/0.6=1$ )

id	Product
1	Bread
1	Milk
1	Water
2	Bread
2	Milk
3	Water
3	Bread
3	Ham
4	Water
4	Eggs
5	Bread
5	Milk

