Scuola universitaria professionale della Svizzera italiana
Bachelor di Ingegneria Informatica

SUPSI

Machine Learning

Lezione 6 - Classificazione

Loris Cannelli, Ricercatore, IDSIA-SUPSI
loris.cannelli@supsi.ch

IDSIA-SUPSI, Polo universitario Lugano - Dipartimento Tecnologie Innovative

Classificazione

- Immaginiamo di avere N oggetti/dati $\mathbf{x}_1, \ldots, \mathbf{x}_N$ di dimensione D

Classificazione

- Immaginiamo di avere N oggetti/dati $\mathbf{x}_1, \ldots, \mathbf{x}_N$ di dimensione D
- Ipotizziamo che esistano C diverse Classi di appartenenza e che ogni oggetto $\mathbf{x}_n$ sia associato ad una delle classi $t_n \in \{1, \ldots, C\}$

Classificazione

- Immaginiamo di avere N oggetti/dati $\mathbf{x}_1, \ldots, \mathbf{x}_N$ di dimensione D
- Ipotizziamo che esistano C diverse Classi di appartenenza e che ogni oggetto $\mathbf{x}_n$ sia associato ad una delle classi $t_n \in \{1, \ldots, C\}$
- Vogliamo capire a quale classe t_{new} appartiene un nuovo oggetto $\mathbf{x}_{\text{new}}$

Classificazione

- Immaginiamo di avere N oggetti/dati $\mathbf{x}_1, \ldots, \mathbf{x}_N$ di dimensione D
- Ipotizziamo che esistano C diverse Classi di appartenenza e che ogni oggetto $\mathbf{x}_n$ sia associato ad una delle classi $t_n \in \{1, \ldots, C\}$
- Vogliamo capire a quale classe t_{new} appartiene un nuovo oggetto $\mathbf{x}_{\text{new}}$

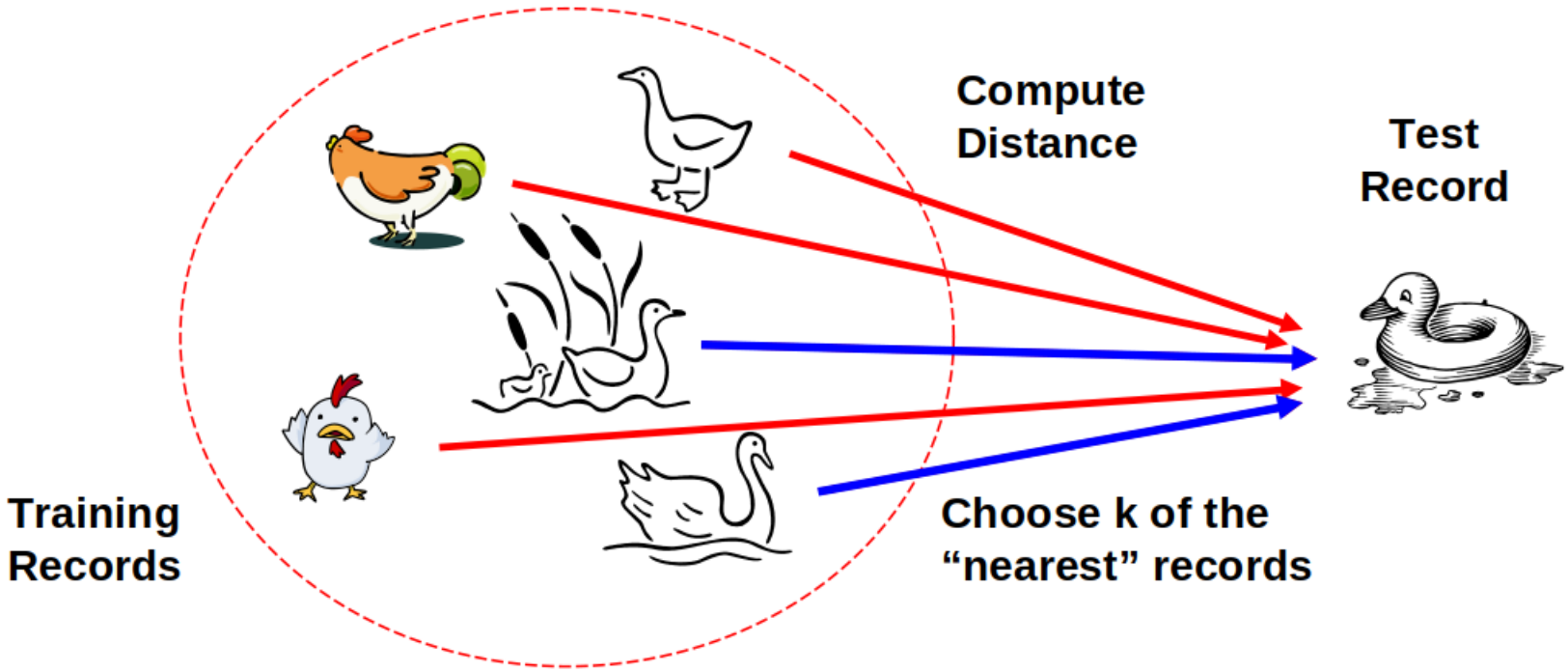
$\Rightarrow$ Classificazione

$K-$ Nearest Neighbors

- Idea chiave:
  - *Se cammina come un'anatra e fa quack come un'anatra, allora probabilmente è un'anatra*

$K-$ Nearest Neighbors

- Idea chiave:
  - *Se cammina come un'anatra e fa quack come un'anatra, allora probabilmente è un'anatra*



$K-$ Nearest Neighbors

Training instances					
Instance ID	*ATTRIB1*	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
#1	0.4	0.5	0.6	0.3	C_1
#2	0.1	0.6	0.9	0.3	C_0
#3	0.1	0.1	1.0	0.8	C_1

Test instance (Query)					
	ATTRIB1	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
q	0.6	0.4	0.0	0.3	?

- KNN: calcola la distanza Euclidea tra le voci di training e quella di test (la query) da classificare.

$K-$ Nearest Neighbors

Training instances					
Instance ID	*ATTRIB1*	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
#1	0.4	0.5	0.6	0.3	C_1
#2	0.1	0.6	0.9	0.3	C_0
#3	0.1	0.1	1.0	0.8	C_1

Test instance (Query)					
	ATTRIB1	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
q	0.6	0.4	0.0	0.3	?

- KNN: calcola la distanza Euclidea tra le voci di training e quella di test (la query) da classificare.

Esempio: Distanza tra la voce di test e la prima di training

$$d(\#1, q) = \sqrt{(0.6 - 0.4)^2 + (0.5 - 0.4)^2 + (0.6)^2 + 0^2} = 0.64$$

$K-$ Nearest Neighbors

- Salva tutti i dati di training

$K-$ Nearest Neighbors

- Salva tutti i dati di training
- Calcola la distanza tra ogni dato di training e il dato di test

$K-$ Nearest Neighbors

- Salva tutti i dati di training
- Calcola la distanza tra ogni dato di training e il dato di test
- Identifica i K dati di training che sono più vicini al dato di test

$K-$ Nearest Neighbors

- Salva tutti i dati di training
- Calcola la distanza tra ogni dato di training e il dato di test
- Identifica i K dati di training che sono più vicini al dato di test
- Predici la classe a maggioranza

$K-$ Nearest Neighbors

Training instances					
Instance ID	*ATTRIB1*	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
#1	0.4	0.5	0.6	0.3	C_1
#2	0.1	0.6	0.9	0.3	C_0
#3	0.1	0.1	1.0	0.8	C_1

Test instance (Query)					
	ATTRIB1	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
q	0.6	0.4	0.0	0.3	?

- Le distanze tra il dato di test e quelli di training:
  - $d(\#1, q) = 0.64$
  - $d(\#2, q) = 1.04$
  - $d(\#3, q) = 0.77$

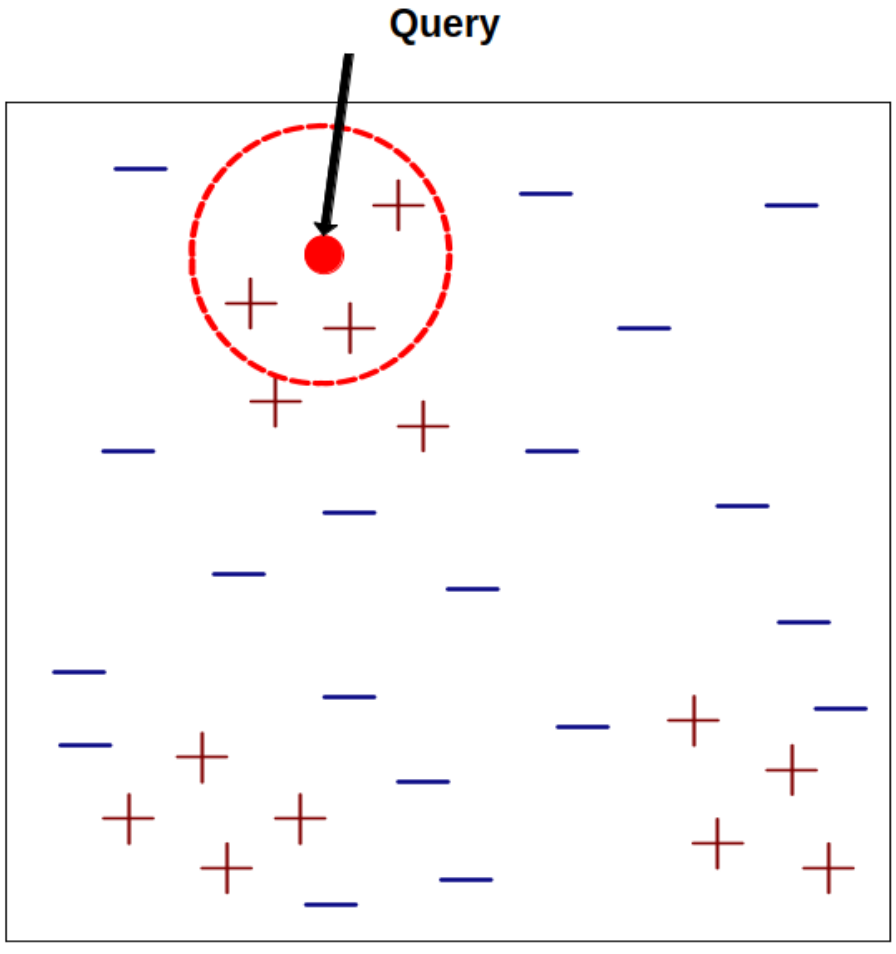
$K-$ Nearest Neighbors

Training instances					
Instance ID	*ATTRIB1*	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
#1	0.4	0.5	0.6	0.3	C_1
#2	0.1	0.6	0.9	0.3	C_0
#3	0.1	0.1	1.0	0.8	C_1

Test instance (Query)					
	ATTRIB1	*ATTRIB 2*	*ATTRIB 3*	*ATTRIB 4*	*CLASS*
q	0.6	0.4	0.0	0.3	?

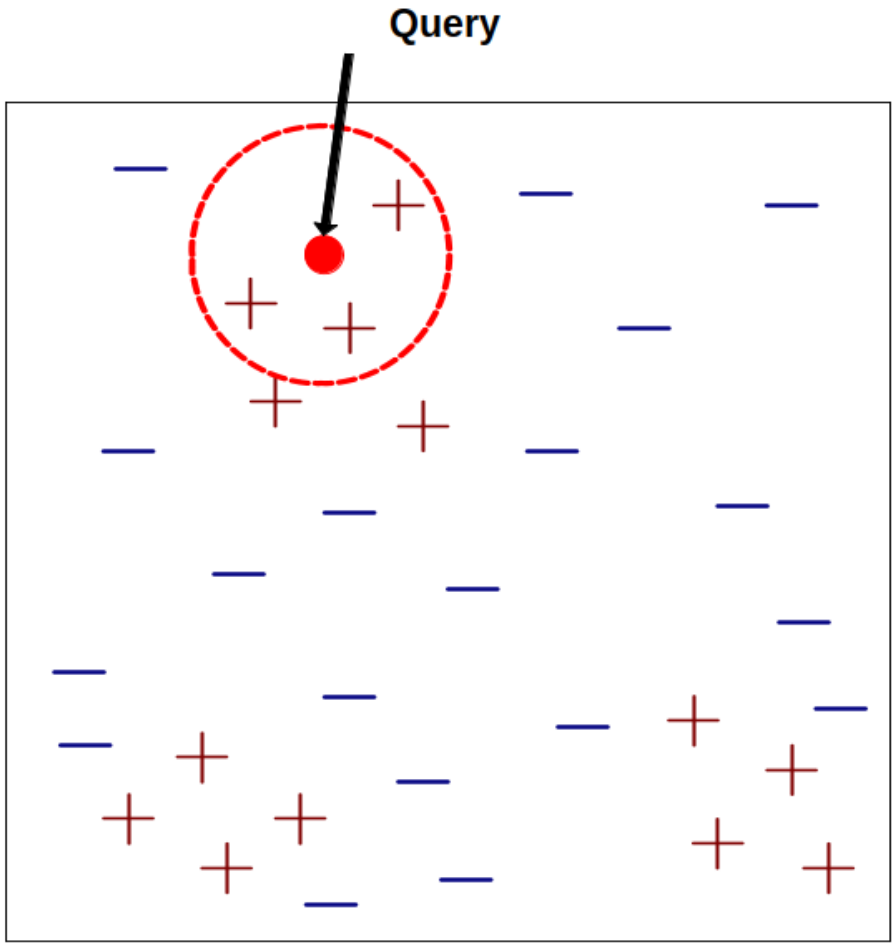
- Le distanze tra il dato di test e quelli di training:
  - $d(\#1, q) = 0.64$
  - $d(\#2, q) = 1.04$
  - $d(\#3, q) = 0.77$
- Con $K = 1$
  - Cerca la distanza minore ($\Rightarrow \#1$)
  - Assegna il dato di test alla classe C_1

$K-$ Nearest Neighbors: Esempio



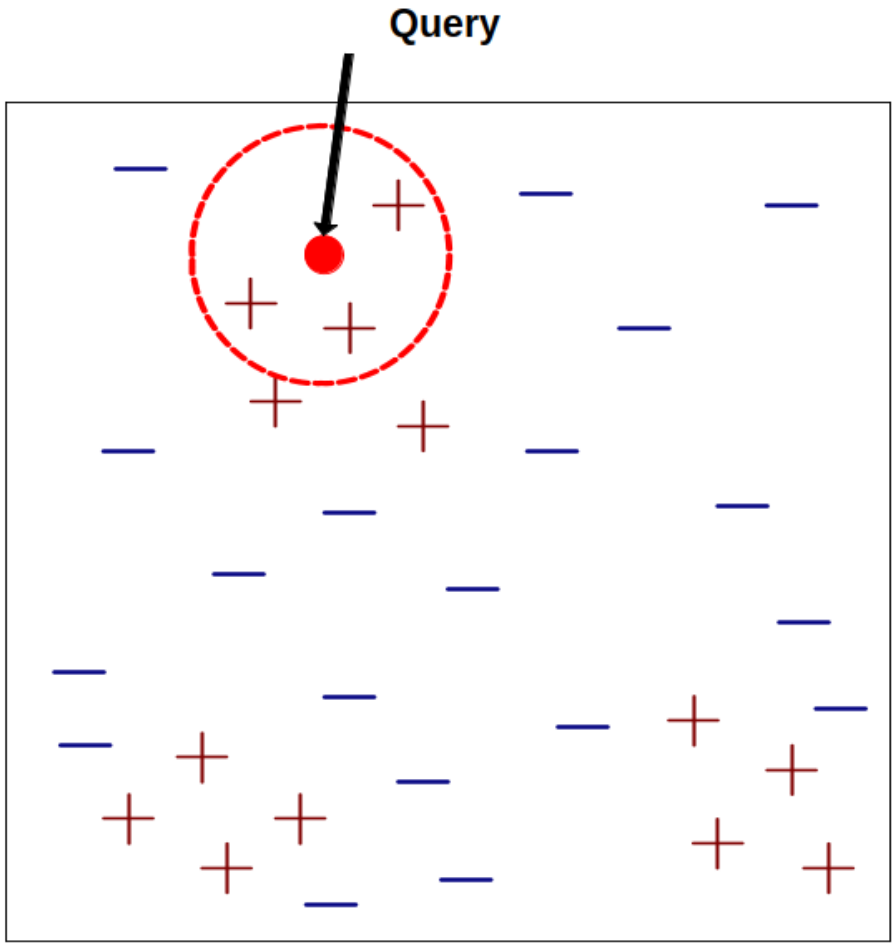
- Due classi; $K = 3$; distanza Euclidea

$K-$ Nearest Neighbors: Esempio



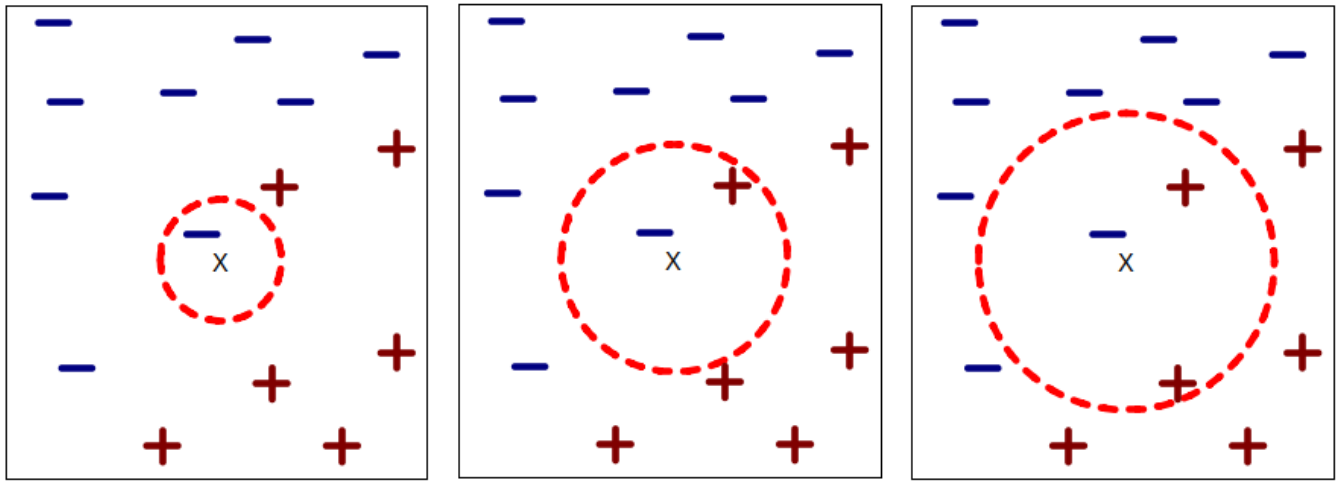
- Due classi; $K = 3$; distanza Euclidea
- Ogni simbolo $+$ rappresenta il training set di una classe; il - rappresenta il training set di un'altra classe

$K-$ Nearest Neighbors: Esempio



- Due classi; $K = 3$; distanza Euclidea
- Ogni simbolo $+$ rappresenta il training set di una classe; il - rappresenta il training set di un'altra classe
- In questo esempio i 3 oggetti più vicini al dato di test sono dei $+$, quindi il dato di test è assegnato a quella classe

$K-$ Nearest Neighbors: Esempio



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- La classificazione cambia a seconda di K: in questo esempio 1-NN classifica con il -, ma 3-NN con il $+$
- Si consiglia di scegliere K dispari in modo che non si verifichino classificazioni incerte come nel caso (b)

1– Nearest Neighbors: interpretazione geometrica

- Nel caso $K = 1$ si può dare un'interpretazione geometrica del metodo 1-NN rappresentando ogni dato di training come un punto nello spazio degli attributi

1– Nearest Neighbors: interpretazione geometrica

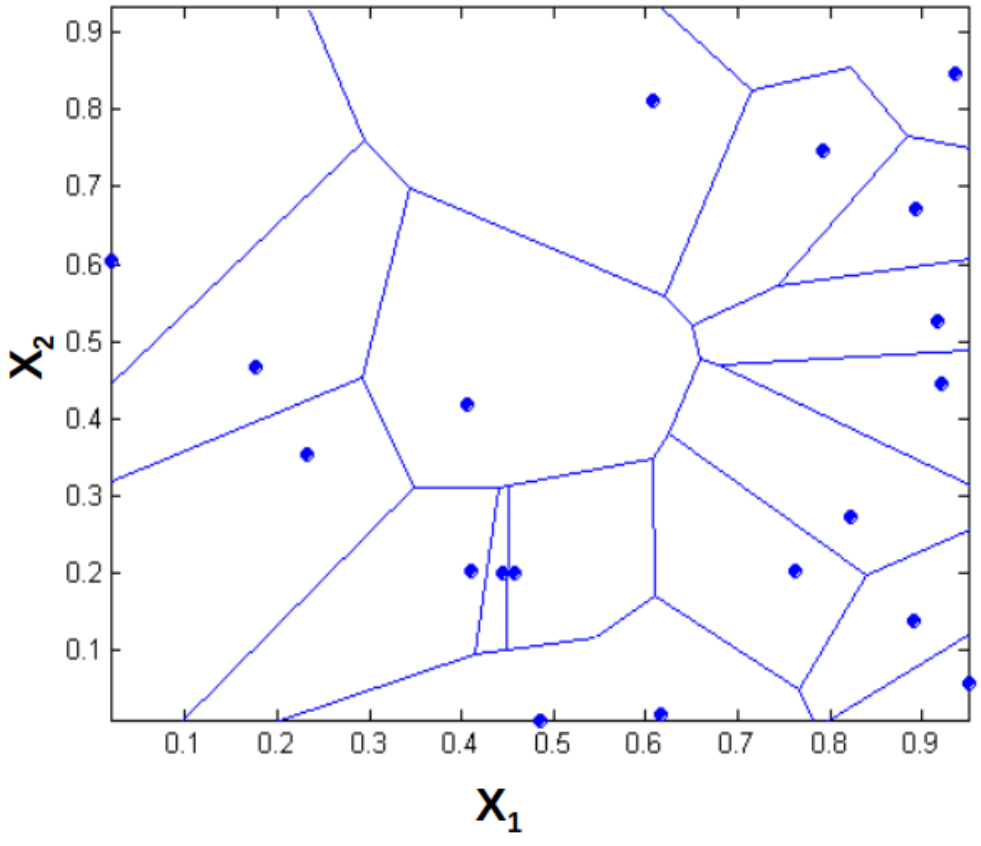
- Nel caso $K = 1$ si può dare un'interpretazione geometrica del metodo 1-NN rappresentando ogni dato di training come un punto nello spazio degli attributi

- Per ogni dato di training, si disegna una regione (cella di Voronoi) comprendente tutti i punti che sono più vicini a quel dato di training rispetto che a qualsiasi altro dato

1– Nearest Neighbors: interpretazione geometrica

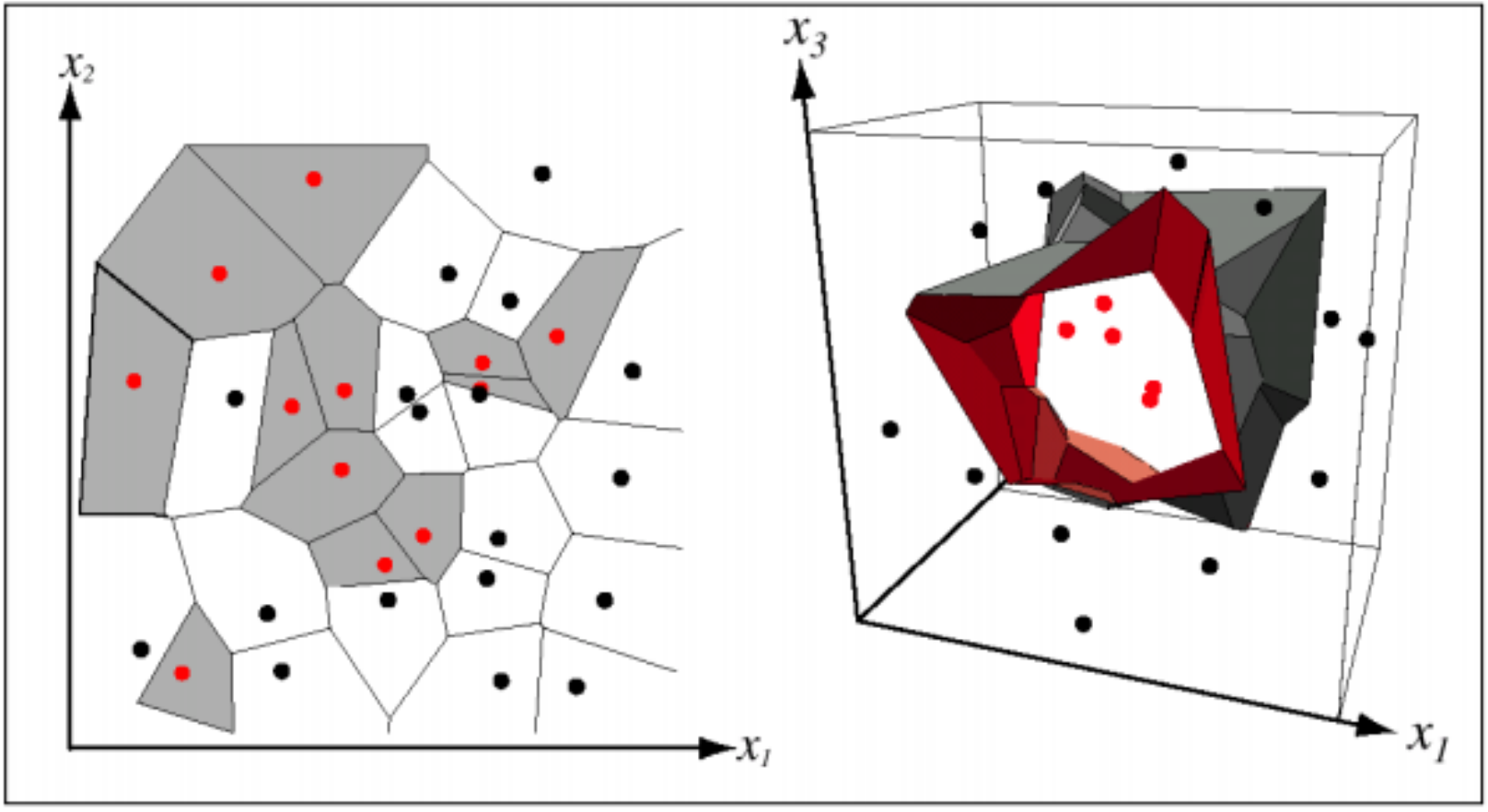
- Nel caso $K = 1$ si può dare un'interpretazione geometrica del metodo 1-NN rappresentando ogni dato di training come un punto nello spazio degli attributi
- Per ogni dato di training, si disegna una regione (cella di Voronoi) comprendente tutti i punti che sono più vicini a quel dato di training rispetto che a qualsiasi altro dato
- Le regioni di Voronoi creano una partizione dello spazio basata sulla distanza tra punti

1– Nearest Neighbors: interpretazione geometrica

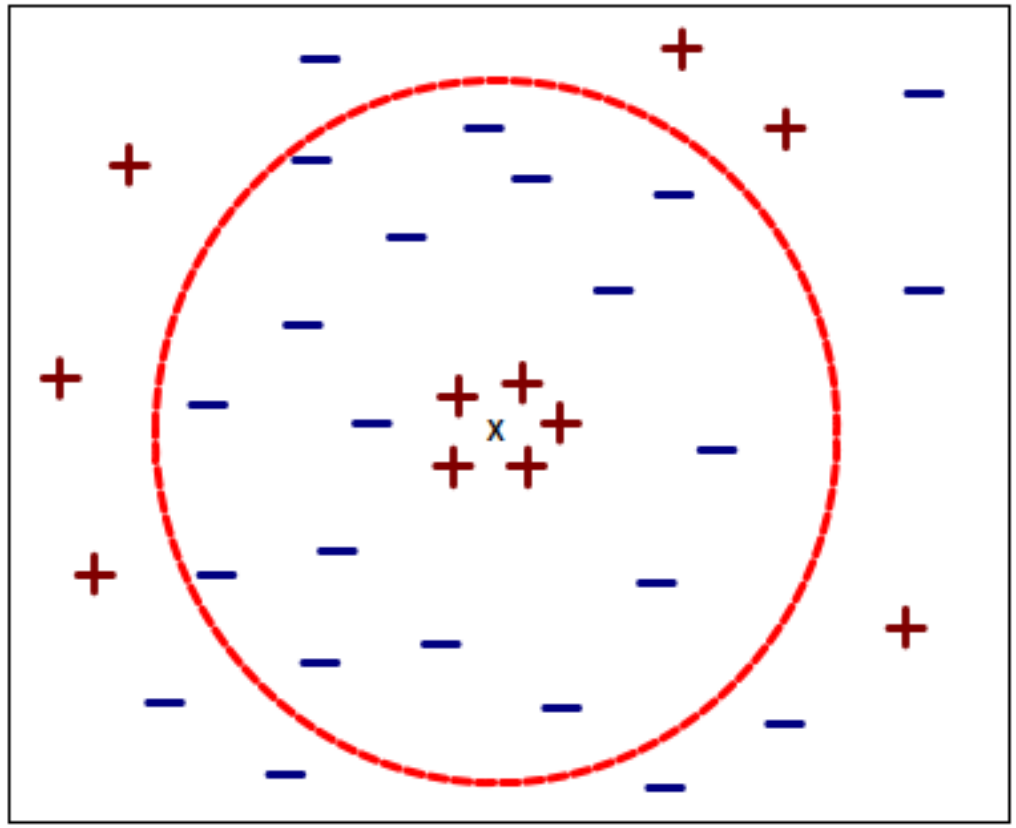


- Una regione di Voronoi racchiude tutti i punti che sono più vicini a un dato di training rispetto che a tutti gli altri
- Per fare classificazione è sufficiente vedere in quale regione cade il dato di test e controllare la classe del dato di training associato

1– Nearest Neighbors: interpretazione geometrica

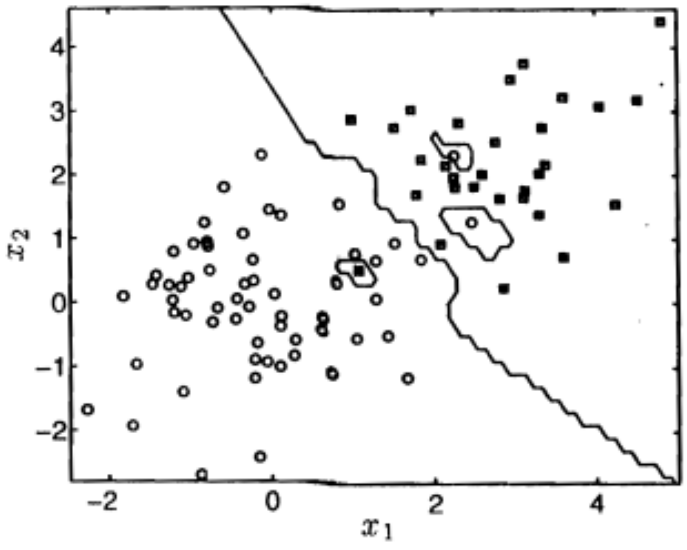


$K-$ Nearest Neighbors: scegliere K

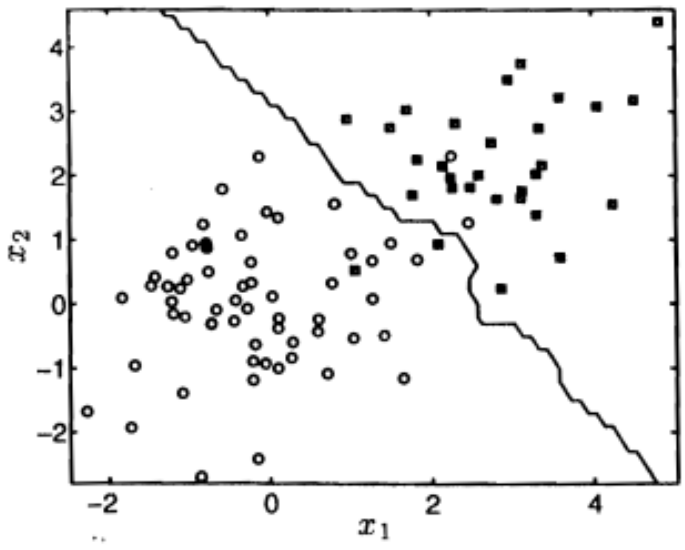


- Valori piccoli di K possono essere inaffidabili perché si basano su pochi elementi per classificare
- Valori alti di K hanno il problema che possono prendere in considerazione dati di training troppo lontani che appartengono alla classe sbagliata
- Scelte comuni: $K = 1, 5, 10$

$K-$ Nearest Neighbors: scegliere K



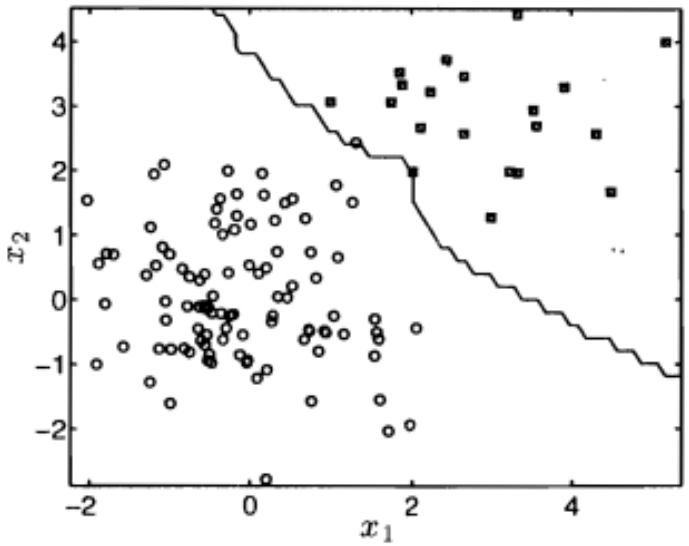
(a) Decision boundary when $K = 1$



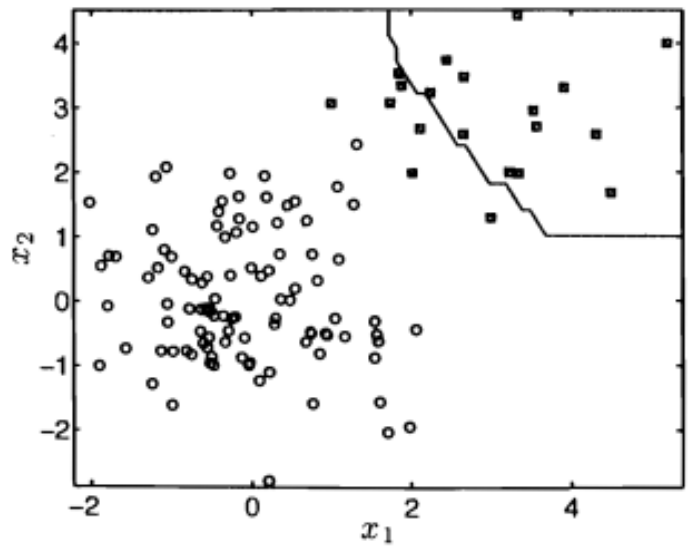
(b) Decision boundary when $K = 5$

Con $K = 1$ la classificazione è ancora inaccurata

$K-$ Nearest Neighbors: scegliere K



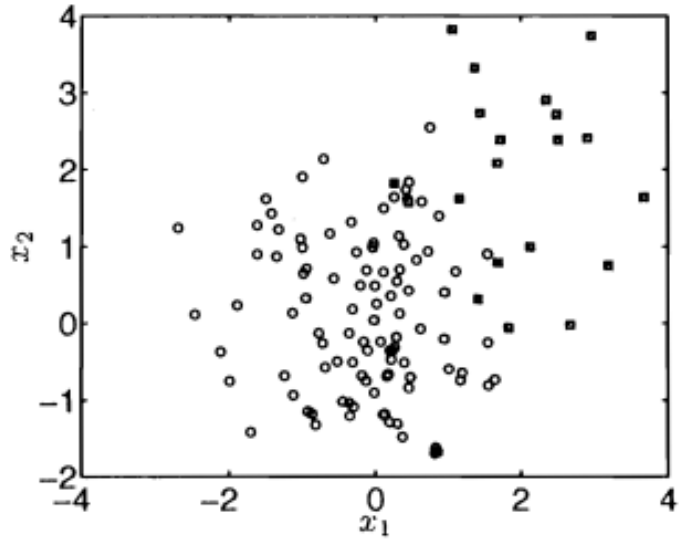
(a) Decision boundary when $K = 5$



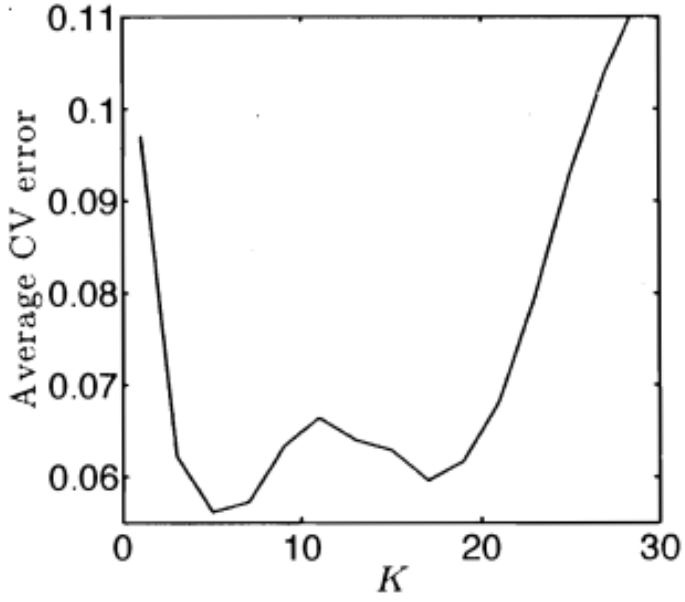
(b) Decision boundary when $K = 39$

- Quando K è troppo elevato la classificazione perde di significato
- Esempio: Immaginiamo un caso con $N_0 = 10$ dati di training per la classe 0 e $N_1 = 60$ per la classe 1. Se si scegliesse $K \geq 21$ qualsiasi dato di test sarebbe associato sempre alla classe 1

$K-$ Nearest Neighbors: scegliere K



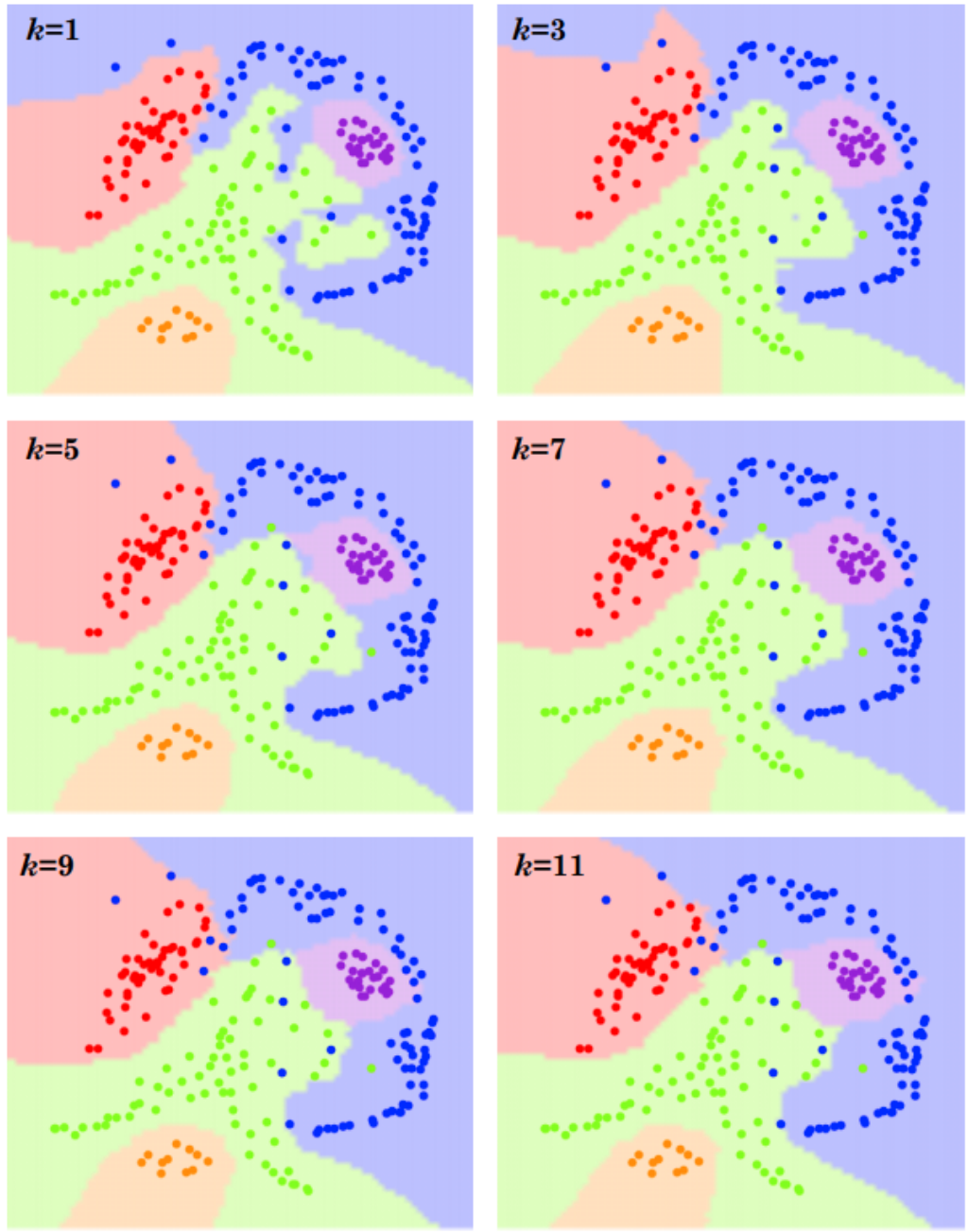
(a) Binary classification dataset. Note the class inbalance: the grey squares class has fewer members than the white circles



(b) Average cross-validation error as K is increased

$\Rightarrow$ cross-validation!

$K-$ Nearest Neighbors: scegliere K



Standardizzazione dei dati

- *Standardizzare* gli attributi è necessario per prevenire che le distanze calcolate siano affette dalle unità di misura

  Esempio

  - l'altezza di una persona può variare tra 1.50m e 1.80m
  - il peso di una persona può variare tra 40kg e 130kg
  - il reddito di una persona può variare tra \$10K e \$1M

Standardizzazione dei dati

- *Standardizzare* gli attributi è necessario per prevenire che le distanze calcolate siano affette dalle unità di misura

  Esempio

  - l'altezza di una persona può variare tra 1.50m e 1.80m
  - il peso di una persona può variare tra 40kg e 130kg
  - il reddito di una persona può variare tra \$10K e \$1M

- Standardizzare rende gli attributi numericamente confrontabili

Standardizzazione dei dati

- *Standardizzare* gli attributi è necessario per prevenire che le distanze calcolate siano affette dalle unità di misura

  Esempio

  - l'altezza di una persona può variare tra 1.50m e 1.80m
  - il peso di una persona può variare tra 40kg e 130kg
  - il reddito di una persona può variare tra \$10K e \$1M

- Standardizzare rende gli attributi numericamente confrontabili
- Una delle standardizzazioni più comuni rende tutti gli attributi a media 0 e deviazione standard 1 (ne esistono molte altre, come il min-max scaling)

Standardizzazione dei dati

- Per standardizzare un attributo si sottrae la media e si divide il risultato per la deviazione standard
- Dopo la standardizzazione, l'attributo ha media 0 e deviazione standard 1

$$[\mathbf{z}_n]_i = \frac{[\mathbf{x}_n]_i - \bar{x}_i}{\sigma_{x_i}}$$

Standardizzazione dei dati

- Per standardizzare un attributo si sottrae la media e si divide il risultato per la deviazione standard
- Dopo la standardizzazione, l'attributo ha media 0 e deviazione standard 1

$$[\mathbf{z}_n]_i = \frac{[\mathbf{x}_n]_i - \bar{x}_i}{\sigma_{x_i}}$$

- $[\mathbf{z}_n]_i$ è l'attributo i-esimo dell'oggetto n-esimo dopo la standardizzazione
- $[\mathbf{x}_n]_i$ è l'attributo i-esimo dell'oggetto n-esimo prima della standardizzazione
- $\bar{x}_i$ è la media dell'attributo i-esimo calcolata sul training set
- σ_{x_i} è la deviazione standard i-esimo calcolata sul training set

$$\bar{x}_i \triangleq \frac{1}{N}\sum_{t=1}^{N}[\mathbf{x}_t]_i; \quad \sigma_{x_i} \triangleq \sqrt{\frac{1}{N-1}\sum_{t=1}^{N}([\mathbf{x}_t]_i - \bar{x}_i)^2}$$

Standardizzazione dei dati

- Prima si standardizza ogni attributo di ogni dato del training set
- Poi si salva la media e la deviazione standard utilizzate per la standardizzazione
- Quando un dato da classificare viene ricevuto, questo va prima standardizzato
- La classificazione si effettua sul dato standardizzato utilizzando il training set standardizzato

Standardizzazione dei dati

	Unstandardized training set and query		
Instance	ATTRIB1	ATTRIB 2	CLASS
#1	0.3	50	C_1
#2	0.1	60	C_0
#3	-0.1	20	C_1
....	....	....	....
mean	0.1	50	
std. dev	0.1	15	
query	0.2	70	
	Standardized training set and query		
	ATTRIB1	ATTRIB 2	CLASS
#1	2	0	C_1
#2	0	0.66	C_0
#3	-2	-2	C_1
query	1	1.33	

Weighted NN

- Ad ogni dato del training set viene dato un peso inversamente proporzionale alla distanza dal dato da classificare

- Si classificano i dati di test come appartenenti alla classe con la somma di pesi più elevata tra i K elementi più vicini

Neighbor	Class	Distance from query	Inverse of distance (weight)
1	Yes	0.1	10
2	No	0.2	5
3	Yes	0.5	2

- Si possono scegliere molti modi diversi per costruire i pesi (inverso del quadrato delle distanze, ecc.)

Tempo Computazionale

Invece di mantenere tutti i dati del training set e calcolare la distanza da ciascuno di essi, per risparmiare tempo e costo computazionale si può derivare da essi dei prototipi e utilizzare questi ultimi per la classificazione

Tempo Computazionale

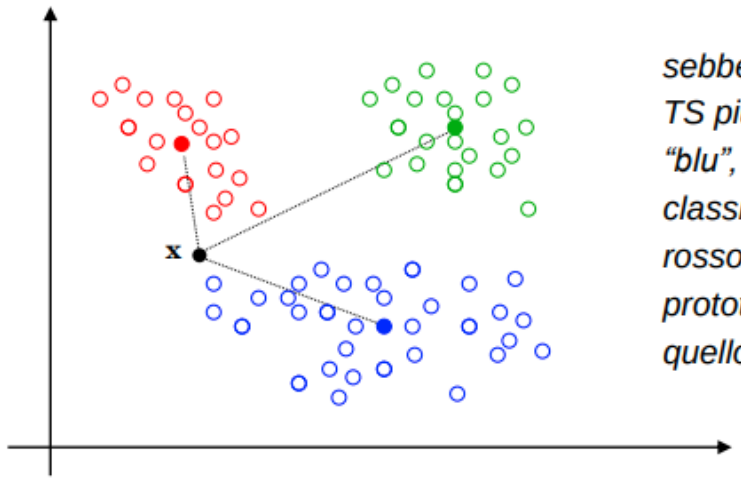
Invece di mantenere tutti i dati del training set e calcolare la distanza da ciascuno di essi, per risparmiare tempo e costo computazionale si può derivare da essi dei prototipi e utilizzare questi ultimi per la classificazione

- Editing:si cancellano dati/attributi senza derivarne di nuovi
- Condensing: si aggiungono nuovi dati, derivati da quelli già presenti

Tempo Computazionale

Invece di mantenere tutti i dati del training set e calcolare la distanza da ciascuno di essi, per risparmiare tempo e costo computazionale si può derivare da essi dei prototipi e utilizzare questi ultimi per la classificazione

- Editing:si cancellano dati/attributi senza derivarne di nuovi
- Condensing: si aggiungono nuovi dati, derivati da quelli già presenti

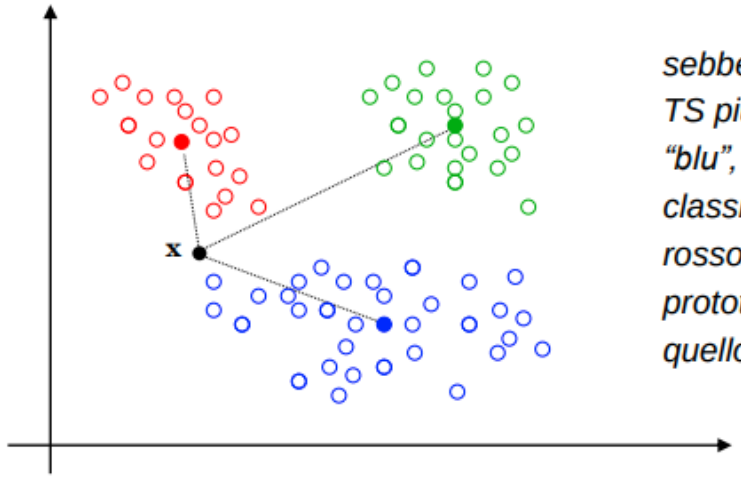


sebbene il pattern di TS più vicino a **x** *sia "blu",* **x** *viene classificato come rosso, in quanto il prototipo più vicino è quello rosso.*

Tempo Computazionale

Invece di mantenere tutti i dati del training set e calcolare la distanza da ciascuno di essi, per risparmiare tempo e costo computazionale si può derivare da essi dei prototipi e utilizzare questi ultimi per la classificazione

- Editing:si cancellano dati/attributi senza derivarne di nuovi
- Condensing: si aggiungono nuovi dati, derivati da quelli già presenti



sebbene il pattern di TS più vicino a **x** *sia "blu",* **x** *viene classificato come rosso, in quanto il prototipo più vicino è quello rosso.*

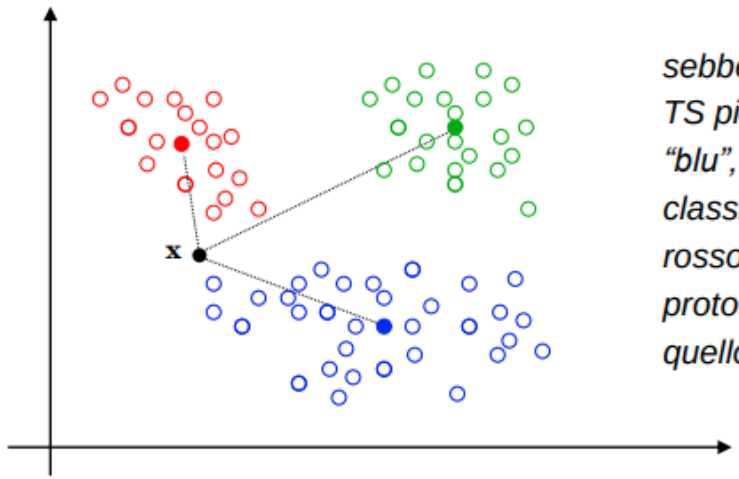
Vantaggi

- non è più necessario calcolare un elevato numero di distanze
- i dati ottenuti dal condensing sono spesso più affidabili e robusti (si riduce il rischio di affidarsi ad outliers)

Tempo Computazionale

Invece di mantenere tutti i dati del training set e calcolare la distanza da ciascuno di essi, per risparmiare tempo e costo computazionale si può derivare da essi dei prototipi e utilizzare questi ultimi per la classificazione

- Editing:si cancellano dati/attributi senza derivarne di nuovi
- Condensing: si aggiungono nuovi dati, derivati da quelli già presenti



sebbene il pattern di TS più vicino a **x** *sia "blu",* **x** *viene classificato come rosso, in quanto il prototipo più vicino è quello rosso.*

Vantaggi

- non è più necessario calcolare un elevato numero di distanze
- i dati ottenuti dal condensing sono spesso più affidabili e robusti (si riduce il rischio di affidarsi ad outliers)

Un singolo dato di condensing per una classe può essere derivato per esempio come vettore medio (*centroide*) dei vettori di quella classe nel training set. Altrimenti si applicano processi di *vector quantization* o *clustering*

0/1 loss

- Si conta il numero di errori di classificazione fatti e lo si divide per il numero totale di test effettuati

0/1 loss

- Si conta il numero di errori di classificazione fatti e lo si divide per il numero totale di test effettuati
- Il numero risultante indica la proporzione di test per cui il classificatore sbaglia

0/1 loss

- Si conta il numero di errori di classificazione fatti e lo si divide per il numero totale di test effettuati
- Il numero risultante indica la proporzione di test per cui il classificatore sbaglia
- Svantaggi: in un test binario in cui, ad esempio, un alto numero di dati appartiene alla Classe 0, un classificatore difettoso che classifica ogni dato di test come appartenente alla Classe 0 avrebbe buone performances secondo questa 0/1 loss

Sensitività e Specificità

- True Positives (TP): il numero di dati appartenente alla Classe 1 correttamente classificati come Classe 1

Sensitività e Specificità

- True Positives (TP): il numero di dati appartenente alla Classe 1 correttamente classificati come Classe 1
- True Negatives (TN): il numero di dati appartenente alla Classe 0 correttamente classificati come Classe 0

Sensitività e Specificità

- True Positives (TP): il numero di dati appartenente alla Classe 1 correttamente classificati come Classe 1
- True Negatives (TN): il numero di dati appartenente alla Classe 0 correttamente classificati come Classe 0
- False Positives (FP): il numero di dati appartenente alla Classe 0 erroneamente classificati come Classe 1

Sensitività e Specificità

- True Positives (TP): il numero di dati appartenente alla Classe 1 correttamente classificati come Classe 1
- True Negatives (TN): il numero di dati appartenente alla Classe 0 correttamente classificati come Classe 0
- False Positives (FP): il numero di dati appartenente alla Classe 0 erroneamente classificati come Classe 1
- False Negatives (TN): il numero di dati appartenente alla Classe 1 erroneamente classificati come Classe 0

Sensitività e Specificità

Sensitività

$$S_e \triangleq \frac{\text{TP}}{\text{TP}+\text{FN}}$$

La proporzione delle persone appartenenti alla Classe 1 (TP+FN) che è correttamente classificata (TP)

Specificità

$$S_p \triangleq \frac{\text{TN}}{\text{TN}+\text{FP}}$$

La proporzione delle persone appartenenti alla Classe 0 (TN+FP) che è correttamente classificata (TN)

Sensitività e Specificità

Sensitività

$$S_e \triangleq \frac{\text{TP}}{\text{TP+FN}}$$

La proporzione delle persone appartenenti alla Classe 1 (TP+FN) che è correttamente classificata (TP)

Specificità

$$S_p \triangleq \frac{\text{TN}}{\text{TN+FP}}$$

La proporzione delle persone appartenenti alla Classe 0 (TN+FP) che è correttamente classificata (TN)

- Idealmente vorremmo $S_e = S_p = 1$

Sensitività e Specificità

Sensitività

$$S_e \triangleq \frac{\text{TP}}{\text{TP+FN}}$$

La proporzione delle persone appartenenti alla Classe 1 (TP+FN) che è correttamente classificata (TP)

Specificità

$$S_p \triangleq \frac{\text{TN}}{\text{TN+FP}}$$

La proporzione delle persone appartenenti alla Classe 0 (TN+FP) che è correttamente classificata (TN)

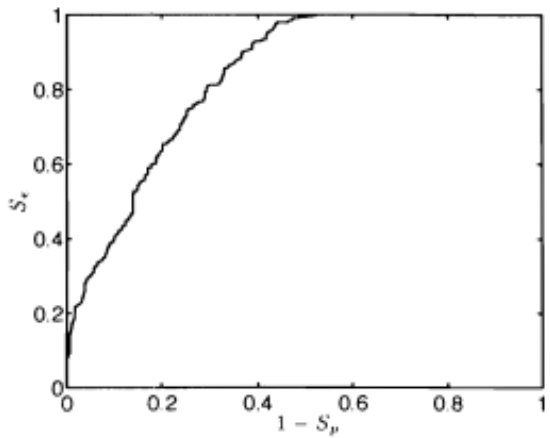
- Idealmente vorremmo $S_e = S_p = 1$
- Quale valore è più importante? Dipende dall'applicazione. Ad esempio, se stiamo facendo un test per scoprire se alcune persone sono malate (Classe 1), è più importante avere alti valori di S_e rispetto a quelli per S_p

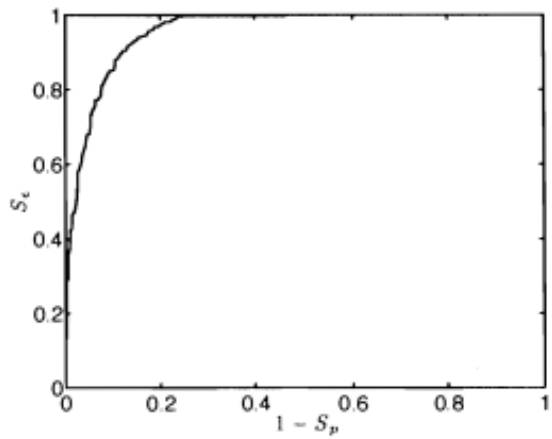
Curva ROC

La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p

Curva ROC

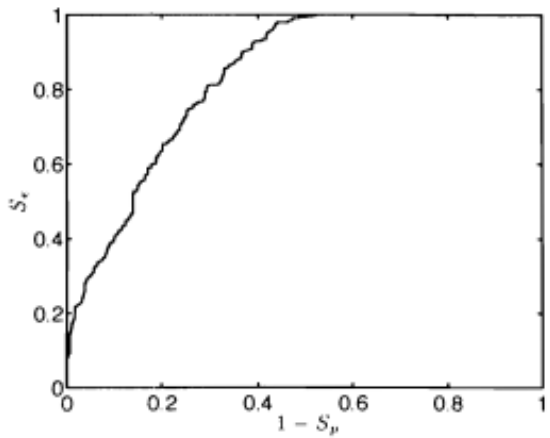
La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p

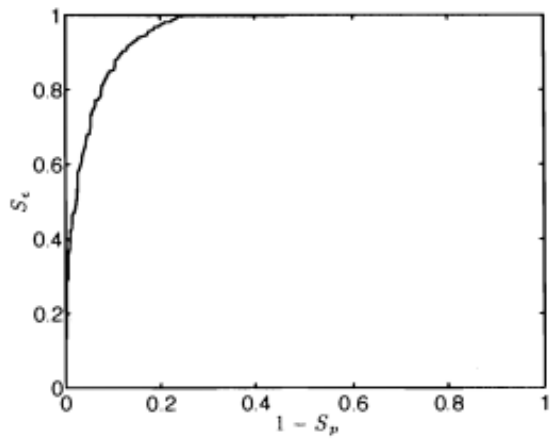




Curva ROC

La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p

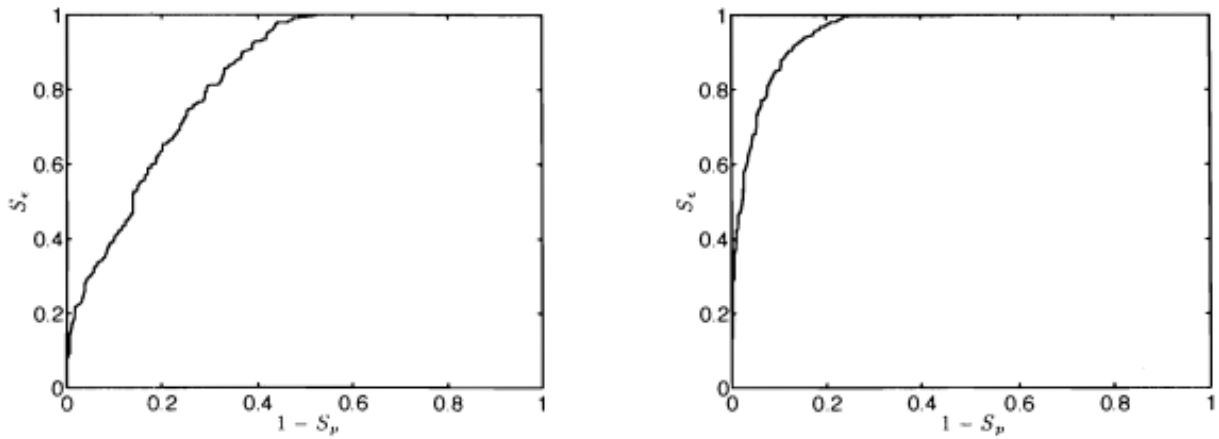




- I classificatori migliori hanno il picco della curva il più possibile vicino all'angolo in alto a sinistra

Curva ROC

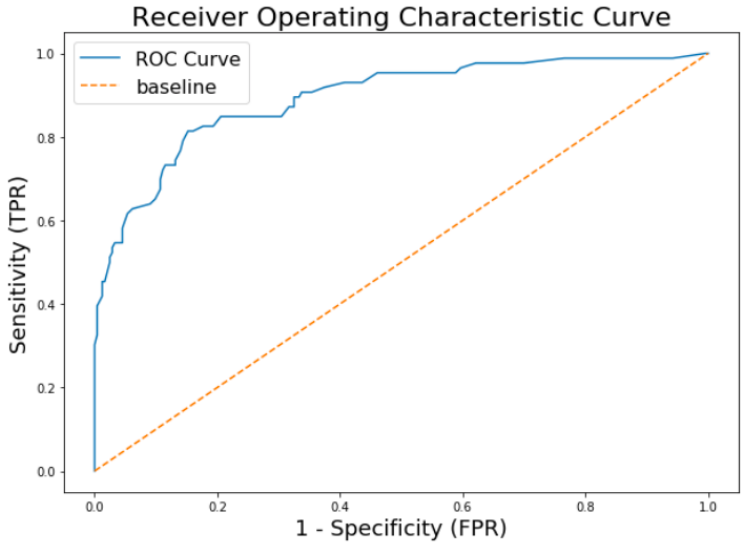
La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p



- I classificatori migliori hanno il picco della curva il più possibile vicino all'angolo in alto a sinistra
- Area Under the Curve (AUC) è un indicatore univoco che descrive la qualità di un classificatore.

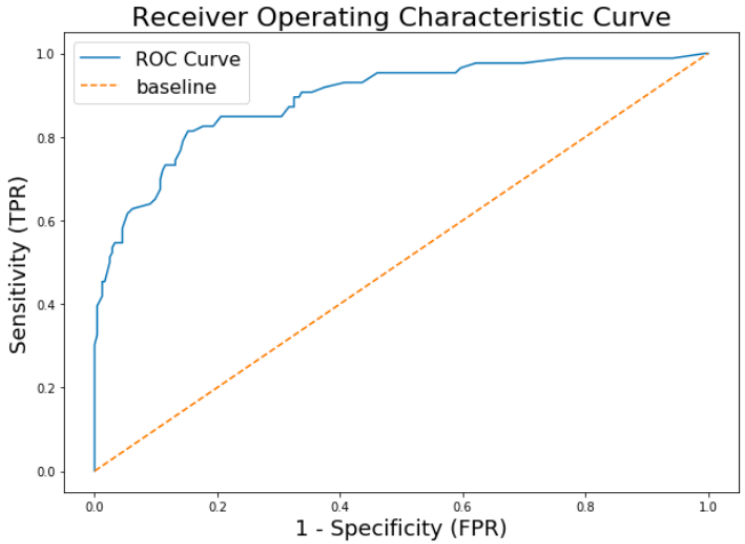
Curva ROC

La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p



Curva ROC

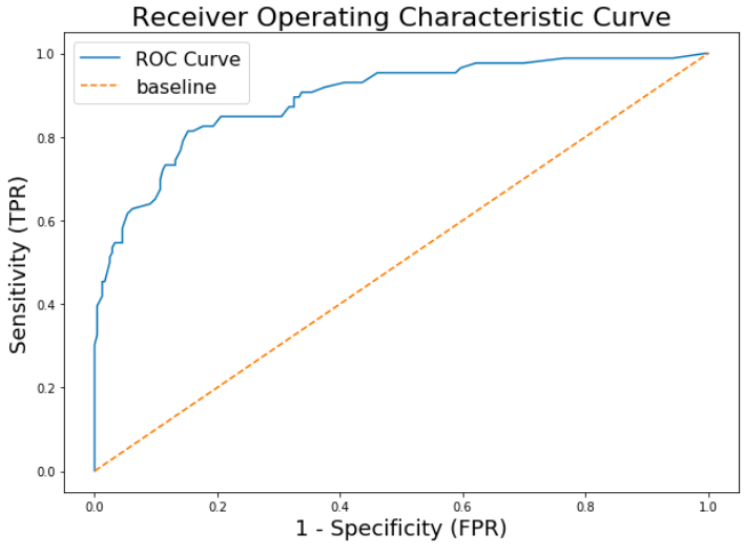
La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p



- Area Under the Curve (AUC) è un indicatore univoco che descrive la qualità di un classificatore.

Curva ROC

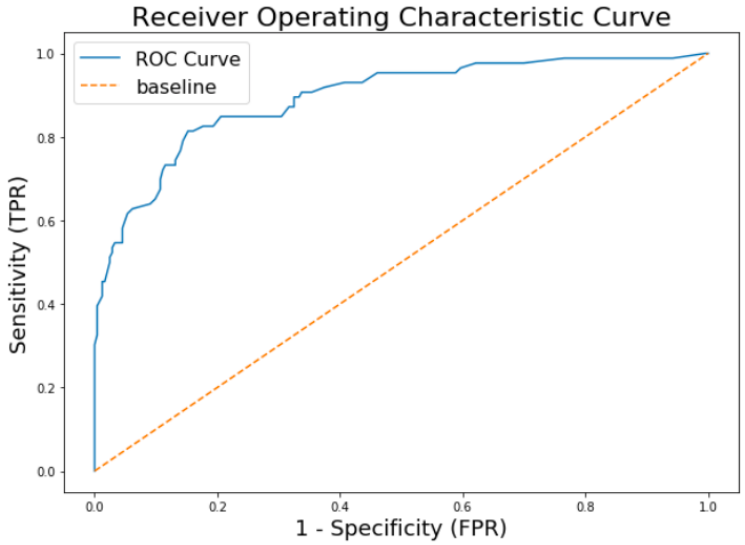
La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p



- Area Under the Curve (AUC) è un indicatore univoco che descrive la qualità di un classificatore.
  - AUC $= 0.5 \Rightarrow$ Peggior classificatore (è equivalente a lanciare una moneta e scegliere la classe a seconda del risultato)

Curva ROC

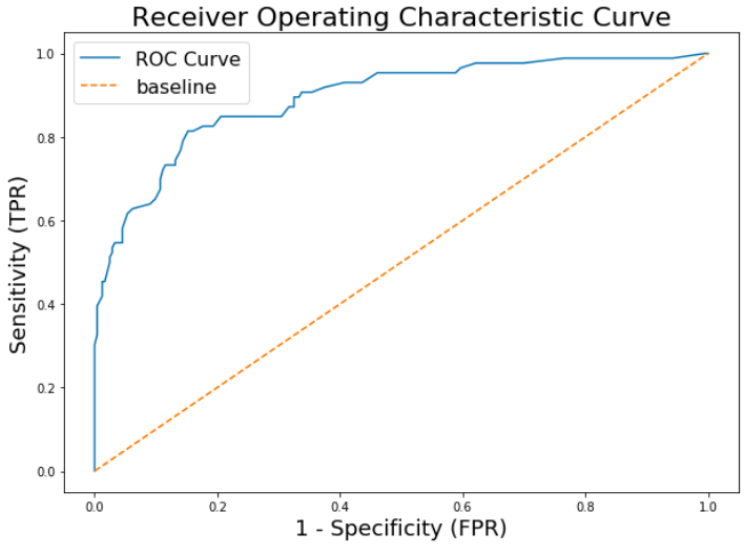
La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p



- Area Under the Curve (AUC) è un indicatore univoco che descrive la qualità di un classificatore.
  - AUC $= 0.5 \Rightarrow$ Peggior classificatore (è equivalente a lanciare una moneta e scegliere la classe a seconda del risultato)
  - AUC $= 1 \Rightarrow$ Miglior classificatore (0 errori)

Curva ROC

La curva Receiver Operating Characteristic (ROC) serve a dare un'interpretazione grafica della validità di un classificatore basandosi sui valori di S_e e S_p



- Area Under the Curve (AUC) è un indicatore univoco che descrive la qualità di un classificatore.
  - AUC $= 0.5 \Rightarrow$ Peggior classificatore (è equivalente a lanciare una moneta e scegliere la classe a seconda del risultato)
  - AUC $= 1 \Rightarrow$ Miglior classificatore (0 errori)
- Svantaggio: cosa fare quando ci sono più di due classi?

$\Rightarrow$ Confusion Matrix

Confusion Matrix

Struttura di una Confusion Matrix per un problema di classificazione binaria

		True class	
		1	0
Predicted class	1	TP	FP
	0	FN	TN

Confusion Matrix

Confusion Matrix per un problema multiclass con 20 classi

Predicted class \ True class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	18	19	20
1	242	3	3	0	1	0	0	1	0	4	2	0	2	10	4	7	1	12	7	47
2	0	296	33	8	8	42	9	1	1	0	0	4	18	7	8	2	0	1	1	3
3	0	6	209	15	9	8	4	0	0	0	0	1	0	1	0	1	0	0	0	0
4	0	12	60	303	36	12	46	2	0	1	0	1	28	3	0	0	0	0	0	0
5	0	8	10	22	277	2	21	0	0	1	0	2	7	0	0	1	1	0	0	0
6	1	21	30	2	2	304	0	1	0	3	0	1	3	0	1	2	0	0	1	0
7	0	1	0	5	5	1	235	5	1	2	0	1	1	0	0	0	1	0	0	0
8	0	3	1	6	4	0	31	356	25	3	1	0	9	4	0	0	2	2	1	0
9	0	2	2	0	1	2	5	4	353	1	0	0	2	0	1	0	1	1	0	1
10	0	0	2	0	1	1	0	2	2	348	4	0	0	1	0	0	1	1	0	0
11	1	0	1	1	0	0	1	0	0	16	382	0	1	0	1	0	1	1	0	0
12	1	16	16	5	4	10	3	1	1	2	0	360	45	0	4	1	3	4	3	1
13	1	4	1	24	16	0	9	5	1	2	0	3	260	3	4	0	0	0	0	0
14	2	3	4	0	8	0	2	0	1	0	2	2	6	324	4	1	1	0	3	3
15	3	7	4	1	2	3	3	2	0	0	1	0	4	3	336	0	2	0	7	5
16	39	4	5	0	0	1	3	1	1	3	2	2	5	17	4	376	3	7	2	68
17	4	0	0	0	3	1	1	5	4	1	0	9	0	3	1	3	325	3	95	19
18	7	1	0	0	0	1	3	1	2	2	1	0	2	6	2	1	2	325	4	5
19	7	2	9	0	6	2	5	8	5	8	4	8	0	10	21	1	16	19	185	7
20	10	0	1	0	0	0	1	0	0	0	0	1	0	1	1	2	4	0	1	92

- I valori elevati sulla diagonale principale indicano che il calssificatore in generale sta funzionando piuttosto bene
- Analizzando i valori off-diagonal si può intuire che le Classi 16 e 20 sono molto simili, così come 19 e 17