Statistica II (750AA)

Lezione 4

Dario Trevisan – *https://web.dm.unipi.it/trevisan*

13/10/2025

Classificazione

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

dove

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

dove

- $x_i \in F$ sono *caratteristiche* (features) dei dati

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

dove

- $x_i \in F$ sono *caratteristiche* (features) dei dati
- $\ell_i \in C$ sono etichette (classi, *labels*)

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

dove

  - $x_i \in F$ sono *caratteristiche* (features) dei dati
  - $\ell_i \in C$ sono etichette (classi, *labels*)

- determinare un *classificatore*:

$$h : F \to C, \quad h(x) = \ell$$

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

  dove

  - $x_i \in F$ sono *caratteristiche* (features) dei dati
  - $\ell_i \in C$ sono etichette (classi, *labels*)

- determinare un *classificatore*:

$$h : F \to C, \quad h(x) = \ell$$

- **Obiettivi della lezione**:

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

dove

  - $x_i \in F$ sono *caratteristiche* (features) dei dati
  - $\ell_i \in C$ sono etichette (classi, *labels*)

- determinare un *classificatore*:

$$h : F \to C, \quad h(x) = \ell$$

- **Obiettivi della lezione**:
  - Comprendere i metodi di classificazione *classici* (statistici).

Introduzione alla classificazione

- **Problema:** dato un insieme di osservazioni

$$\{(x_i, \ell_i)\}_{i=1,\ldots,n}$$

  dove

  - $x_i \in F$ sono *caratteristiche* (features) dei dati
  - $\ell_i \in C$ sono etichette (classi, *labels*)

- determinare un *classificatore*:

$$h : F \to C, \quad h(x) = \ell$$

- **Obiettivi della lezione**:
  - Comprendere i metodi di classificazione *classici* (statistici).
  - Conoscere le metriche di performance.

La classificazione è fondamentale in statistica e machine learning.

- Si tratta di un problema di apprendimento *supervisionato*:

$$\{(x_i, \ell_i)\}_{i=1,\dots,n} \Rightarrow h : x \to h(x).$$

Esempi:

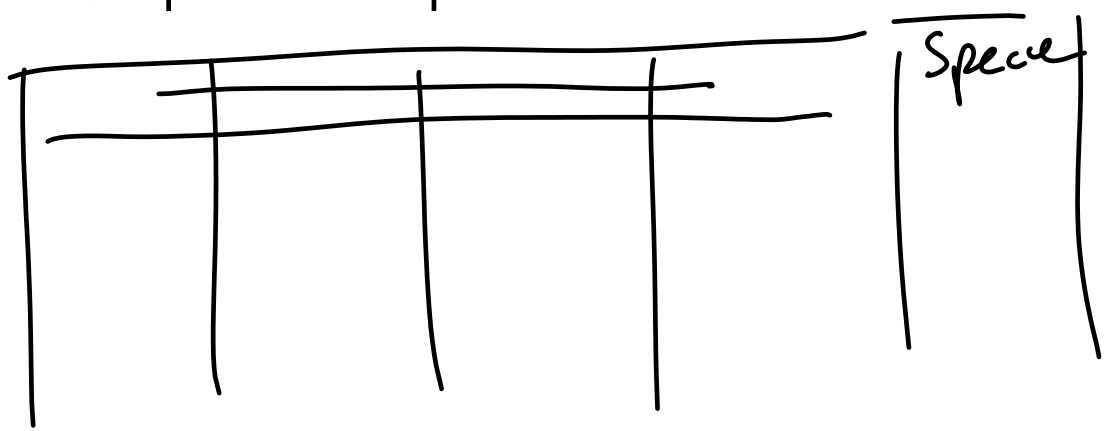
La classificazione è fondamentale in statistica e machine learning.

- Si tratta di un problema di apprendimento *supervisionato*:

$$\{(x_i, \ell_i)\}_{i=1,\dots,n} \Rightarrow h : x \to h(x).$$

Esempi:

- Determinare la specie dei fiori del dataset Iris in base a lunghezza e ampiezza dei petali.



La classificazione è fondamentale in statistica e machine learning.

- Si tratta di un problema di apprendimento *supervisionato*:

$$\{(x_i, \ell_i)\}_{i=1,\dots,n} \Rightarrow h : x \to h(x).$$

Esempi:

- Determinare la specie dei fiori del dataset Iris in base a lunghezza e ampiezza dei petali.
- Determinare se un messaggio di posta è spam oppure no.

La classificazione è fondamentale in statistica e machine learning.

- Si tratta di un problema di apprendimento *supervisionato*:

$$\{(x_i, \ell_i)\}_{i=1,\dots,n} \Rightarrow h : x \to h(x).$$

Esempi:

- Determinare la specie dei fiori del dataset Iris in base a lunghezza e ampiezza dei petali.
- Determinare se un messaggio di posta è spam oppure no.
- Determinare se un tumore è benigno o maligno.

La classificazione è fondamentale in statistica e machine learning.

- Si tratta di un problema di apprendimento *supervisionato*:

$$\{(x_i, \ell_i)\}_{i=1,\dots,n} \Rightarrow h : x \to h(x).$$

Esempi:

- Determinare la specie dei fiori del dataset Iris in base a lunghezza e ampiezza dei petali.
- Determinare se un messaggio di posta è spam oppure no.
- Determinare se un tumore è benigno o maligno.
- Determinare per quale candidato voterà un individuo in base alle sue attività online. . . .

K-Nearest Neighbors (KNN): caso $k = 1$

- **Idea**: punti con caratteristiche simili hanno una etichetta simile.

K-Nearest Neighbors (KNN): caso $k = 1$

- **Idea**: punti con caratteristiche simili hanno una etichetta simile.
- *Classificatore al primo vicino*: dato $x \in F \subseteq \mathbb{R}^d$, troviamo il punto nel *training set* $i(x)$ con caratteristiche **più vicine** a x

$$i(x) \in \arg\min_{i=1,\dots,n} \|x_i - x\|$$

e classifichiamo $h(x) := \ell_{i(x)}$.

K-Nearest Neighbors (KNN): caso $k = 1$

- **Idea**: punti con caratteristiche simili hanno una etichetta simile.
- *Classificatore al primo vicino*: dato $x \in F \subseteq \mathbb{R}^d$, troviamo il punto nel *training set* $i(x)$ con caratteristiche **più vicine** a x

$$i(x) \in \arg \min_{i=1,\dots,n} \|x_i - x\|$$

  e classifichiamo $h(x) := \ell_{i(x)}$.
- **Vantaggi**: intiuitivo, semplice da implementare, anche con metriche diverse dall'Euclidea (Manhattan,...)

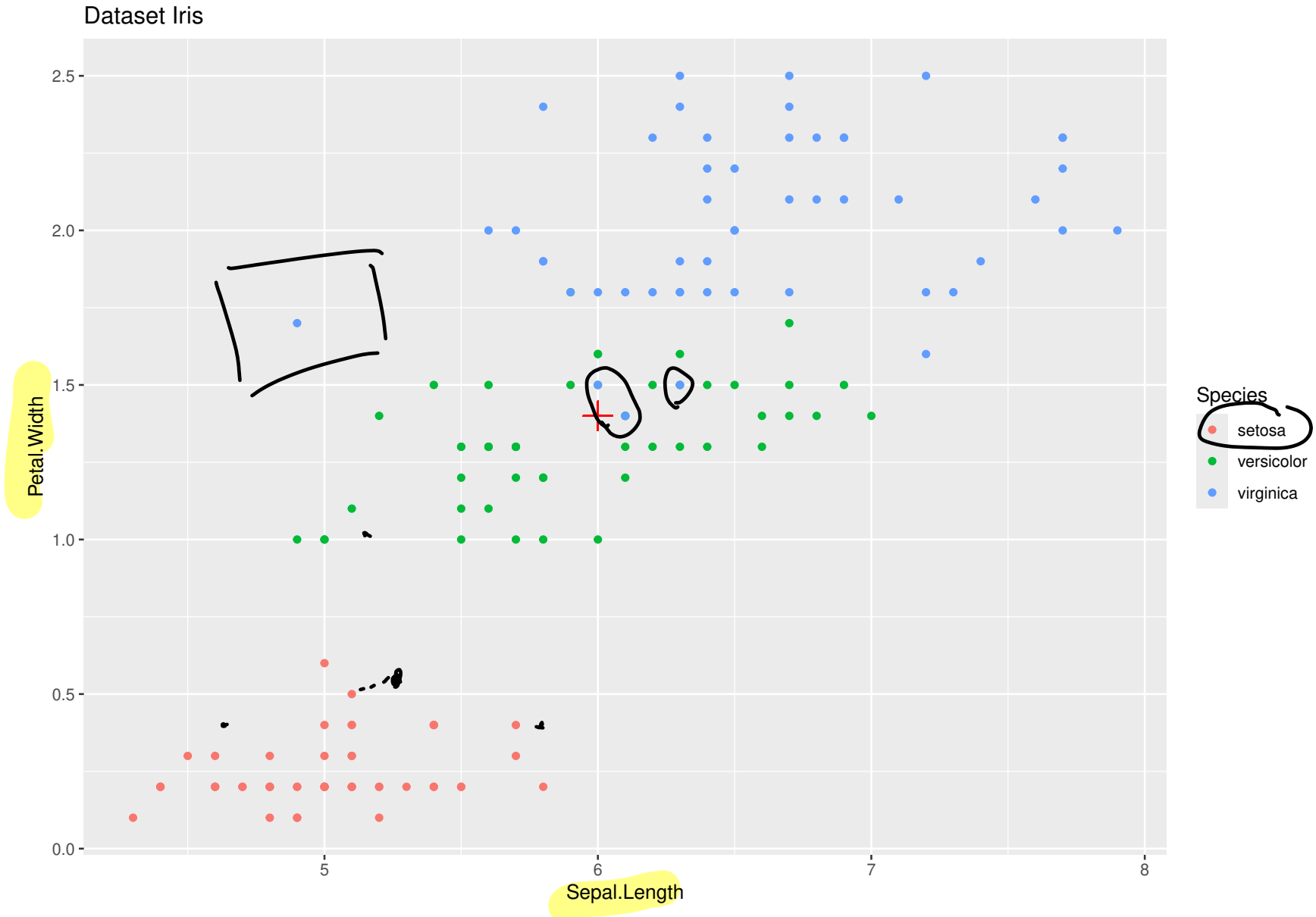
K-Nearest Neighbors (KNN): caso $k = 1$

- **Idea**: punti con caratteristiche simili hanno una etichetta simile.
- *Classificatore al primo vicino*: dato $x \in F \subseteq \mathbb{R}^d$, troviamo il punto nel *training set* $i(x)$ con caratteristiche **più vicine** a x

$$i(x) \in \arg \min_{i=1,\dots,n} \|x_i - x\|$$

  e classifichiamo $h(x) := \ell_{i(x)}$.
- **Vantaggi**: intiuitivo, semplice da implementare, anche con metriche diverse dall'Euclidea (Manhattan,...)
- **Svantaggi**: sensibile al *rumore* (varianza), costo computazionale se $n \gg 1$, curse of dimensionality.

Esempio (iris)



KNN: Caso k generale

Per **alleviare** il problema della varianza:

- fissiamo un parametro k, ad es, $k = 2, 3, 4, \ldots$ (meglio se dispari)

KNN: Caso k generale

Per **alleviare** il problema della varianza:

- fissiamo un parametro k, ad es, $k = 2, 3, 4, \ldots$ (meglio se dispari)
- dato $x \in F \subseteq \mathbb{R}^d$, troviamo i k punti nel *training set* $i_1(x), i_2(x), \ldots, i_k(x)$ con *features* **più vicine** a x e le rispettive *labels*

$$\ell_{i_1}, \ell_{i_2}, \ldots, \ell_{i_k}$$

KNN: Caso k generale

Per **alleviare** il problema della varianza:

- fissiamo un parametro k, ad es, $k = 2, 3, 4, \ldots$ (meglio se dispari)
- dato $x \in F \subseteq \mathbb{R}^d$, troviamo i k punti nel *training set* $i_1(x), i_2(x), \ldots, i_k(x)$ con *features* **più vicine** a x e le rispettive *labels*

$$\ell_{i_1}, \ell_{i_2}, \ldots, \ell_{i_k}$$

- definiamo $h_{knn}(x)$ pari alla label più frequente nelle k.

KNN: Caso k generale

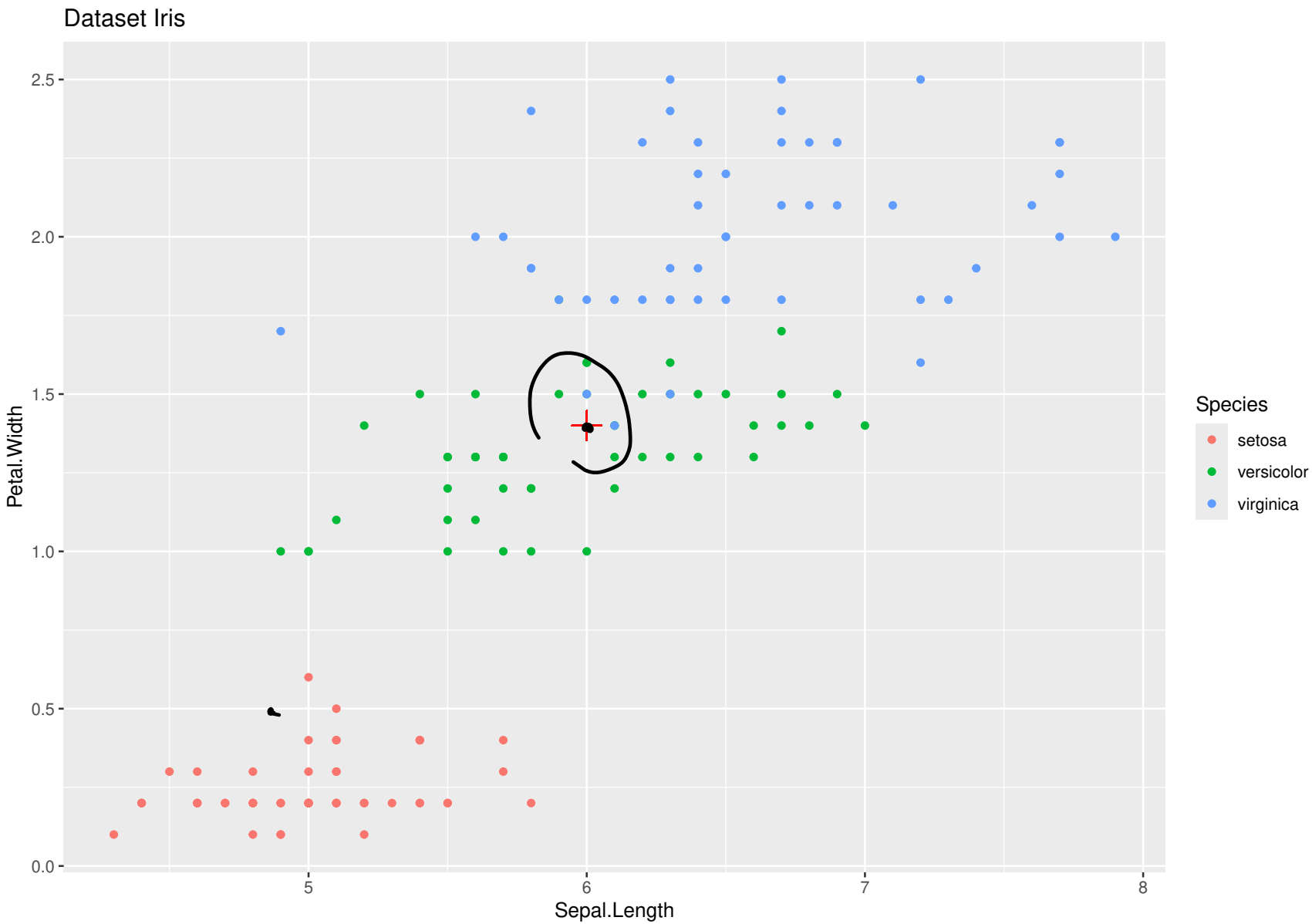
Per **alleviare** il problema della varianza:

- fissiamo un parametro k, ad es, $k = 2, 3, 4, \ldots$ (meglio se dispari)
- dato $x \in F \subseteq \mathbb{R}^d$, troviamo i k punti nel *training set* $i_1(x), i_2(x), \ldots, i_k(x)$ con *features* **più vicine** a x e le rispettive *labels*

$$\ell_{i_1}, \ell_{i_2}, \ldots, \ell_{i_k}$$

- definiamo $h_{knn}(x)$ pari alla label più frequente nelle k.
- **Problema:** come determinare il migliore valore di k?

Esempio


Dataset Iris
Petal.Width
Sepal.Length
2.5
2.0
1.5
1.0
0.5
0.0
5
6
7
8
Species
setosa
versicolor
virginica

Hello KNN!

- Giocate voi!
  https://dario-trevisan.shinyapps.io/HelloNeighborKNN/

Hello KNN!

- Giocate voi!
  https://dario-trevisan.shinyapps.io/HelloNeighborKNN/
- Codice (R shiny) disponibile nel team e nella pagina del corso

Accuratezza/errore

L'*accuratezza* (accuracy) di un classificatore è una metrica di performance. - Indica la percentuale di dati classificati in modo corretto:

$$\text{Acc(h)} = \frac{\#\text{ classificazioni corrette}}{\#\text{ esempi totali}}$$

- **Problema**: su quali dati calcolare l'accuratezza?

  - **Training Error**: calcolato sui dati originali (*training set*)

$$\frac{1}{n}\sum_{i=1}^{n} 1_{\{h(x_i)\neq \ell_i\}}$$

1-NN è ottimale!

Accuratezza/errore

K-NN

L'*accuratezza* (accuracy) di un classificatore è una metrica di performance. - Indica la percentuale di dati classificati in modo corretto:

$$\text{Acc(h)} =$$

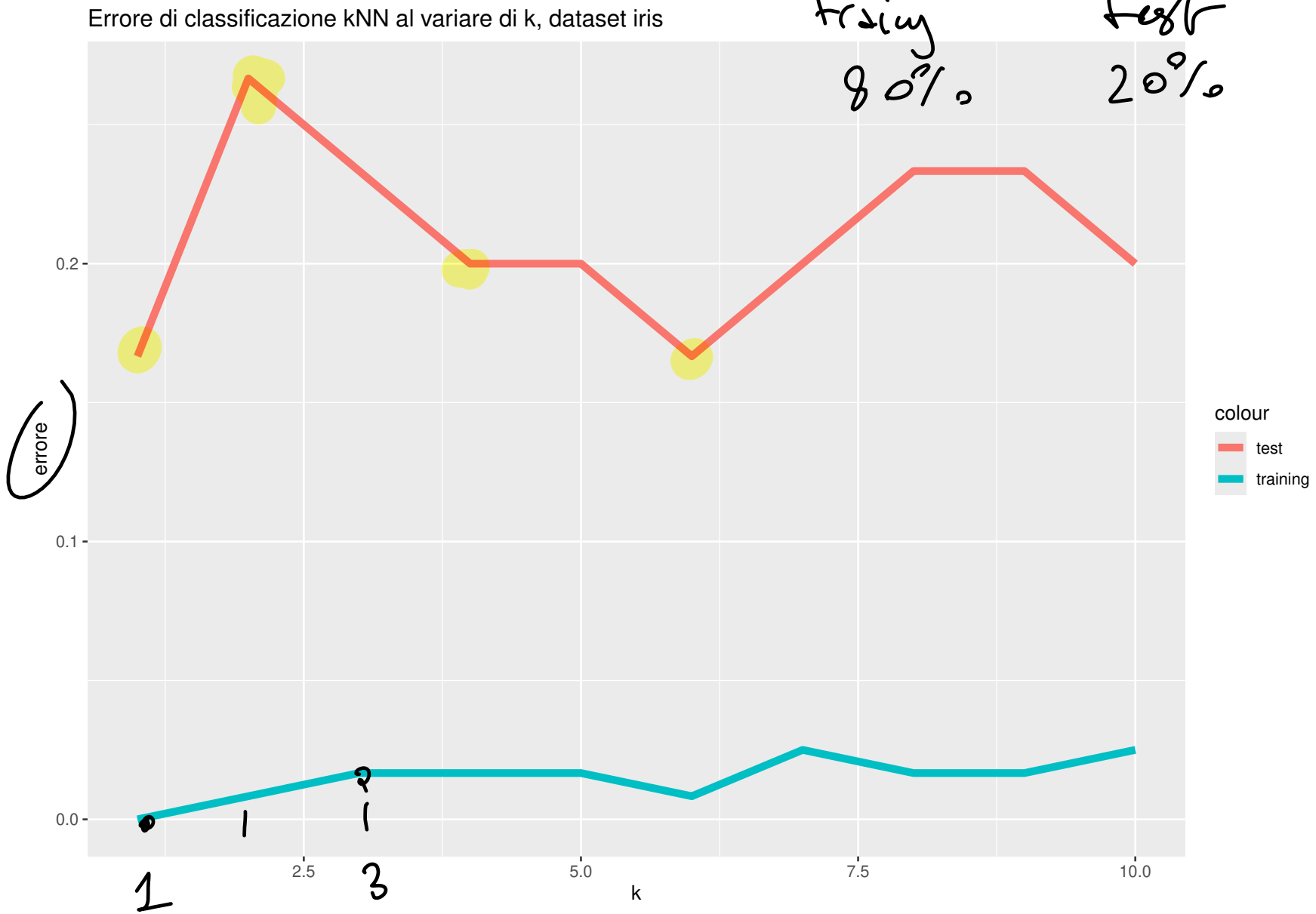
- **Problema**: su quali dati calcolare l'accuratezza?

  - **Training Error**: calcolato sui dati originali (*training set*)

$$\sum_{i=1}^{n} 1_{\{h(x_i) \neq \ell_i\}}$$

  - **Test Error**: calcolato su un *insieme di verifica* $\{(x_j, \ell_j)\}_{j=1,\ldots,m}$ (*test set*) per valutare la *generalizzazione* del modello.

Esempio (iris)



Errore di generalizzazione

Problema: La decomposizione del dataset in *train* e *test* è del tutto arbitraria,

- si *sprecano* dati per valutare la performance (ad esempio per selezionare k)

Errore di generalizzazione

Problema: La decomposizione del dataset in *train* e *test* è del tutto arbitraria,

- si *sprecano* dati per valutare la performance (ad esempio per selezionare k)
- il *vero* obiettivo è di valutare l'accuratezza su *nuovi* dati.

Errore di generalizzazione

Problema: La decomposizione del dataset in *train* e *test* è del tutto arbitraria,

- si *sprecano* dati per valutare la performance (ad esempio per selezionare k)
- il *vero* obiettivo è di valutare l'accuratezza su *nuovi* dati.
- **Modello probabilistico** dei dati non osservati: $(X, L) \in F \times C$ è aleatorio, l'**errore di generalizzazione** di h è

$$P(h(X) \neq L)$$

Errore di generalizzazione

Problema: La decomposizione del dataset in *train* e *test* è del tutto arbitraria,

- si *sprecano* dati per valutare la performance (ad esempio per selezionare k)
- il *vero* obiettivo è di valutare l'accuratezza su *nuovi* dati.
- **Modello probabilistico** dei dati non osservati: $(X, L) \in F \times C$ è aleatorio, l'**errore di generalizzazione** di h è

$$P(h(X) \neq L)$$

- I dati $\{(x_i, \ell_i)\}_{i=1\ldots,n}$ sono un campione di *variabili aleatorie indipendenti*, tutte con la stessa *legge* di (X, L).

Richiami di probabilità

Date affermazioni (*eventi*) A, I, la probabilità

$$P(A|I) \in [0, 1]$$

indica il grado di validità di A (supponendo di aver) osservato I.

- $P(A|I) \approx 1 \Rightarrow A$ è ritenuta vera

Richiami di probabilità

Date affermazioni (*eventi*) A, I, la probabilità

$$P(A|I) \in [0, 1]$$

indica il grado di validità di A (supponendo di aver) osservato I.

- $P(A|I) \approx 1 \Rightarrow A$ è ritenuta vera
- $P(A|I) \approx 0 \Rightarrow A$ è ritenuta falsa

Parametrizzazioni alternative

- $P(A|I) \in [0, 1]$

Parametrizzazioni alternative

- $P(A|I) \in [0, 1]$
- logaritmo: $\log(P(A|I))$

Parametrizzazioni alternative

- $P(A|I) \in [0,1]$
- logaritmo: $\log(P(A|I))$
- quote (*odds*):

$$\text{Odds}(A|I) = \frac{P(A|I)}{P(\text{non } A|I)} \in [0,\infty]$$

Odds(A) = 1 : 1 ⟶ P(A) = 50%

Parametrizzazioni alternative

- $P(A|I) \in [0,1]$
- logaritmo: $\log(P(A|I))$
- quote (*odds*):

$$\text{Odds}(A|I) = \frac{P(A|I)}{P(\text{non } A|I)} \in [0,\infty]$$

- logit

$$\text{Logit}(A|I) = \log(\text{Odds}(A|I)) \in [-\infty,\infty]$$

Regole di calcolo (eventi)

- **Regola della somma**: $P(A \text{ oppure } B|I) = P(A|I) + P(B|I)$, se A, B *incompatibili*. $P(A \text{ e } B \,|\, I) = 0$

Regole di calcolo (eventi)

- **Regola della somma**: $P(A \text{ oppure } B|I) = P(A|I) + P(B|I)$, se A, B *incompatibili*.
- **Probabilità della negazione**: $P(\text{non } A|I) = 1 - P(A|I)$

Regole di calcolo (eventi)

- **Regola della somma**: $P(A \text{ oppure } B|I) = P(A|I) + P(B|I)$, se A, B *incompatibili*.
- **Probabilità della negazione**: $P(\text{non } A|I) = 1 - P(A|I)$
- **Regola del prodotto**: $P(A \text{ e } B|I) = P(A|I)P(B|A \text{ e } I)$

Indipendenza $\quad P(A \text{ e } B) = P(A)P(B)$

Regole di calcolo (eventi)

- **Regola della somma**: $P(A \text{ oppure } B|I) = P(A|I) + P(B|I)$, se A, B *incompatibili*.
- **Probabilità della negazione**: $P(\text{non } A|I) = 1 - P(A|I)$
- **Regola del prodotto**: $P(A \text{ e } B|I) = P(A|I)P(B|A \text{ e } I)$
- **Formula di Bayes**: $P(A|B) = P(A)P(B|A)/P(B)$, oppure

$$\text{Odds}(A|B) = \text{Odds}(A)\frac{\mathcal{L}(A;B)}{\mathcal{L}(\text{non } A;B)}$$

(applico di verosimiglianza

Regole di calcolo (eventi)

- **Regola della somma**: $P(A \text{ oppure } B|I) = P(A|I) + P(B|I)$, se A, B *incompatibili*.
- **Probabilità della negazione**: $P(\text{non } A|I) = 1 - P(A|I)$
- **Regola del prodotto**: $P(A \text{ e } B|I) = P(A|I)P(B|A \text{ e } I)$
- **Formula di Bayes**: $P(A|B) = P(A)P(B|A)/P(B)$, oppure

$$\text{Odds}(A|B) = \text{Odds}(A)\frac{\mathcal{L}(A;B)}{\mathcal{L}(\text{non } A;B)}$$

- **Verosimiglianza**: $\mathcal{L}(A;B) = P(B|A)$.

Regole di calcolo (eventi)

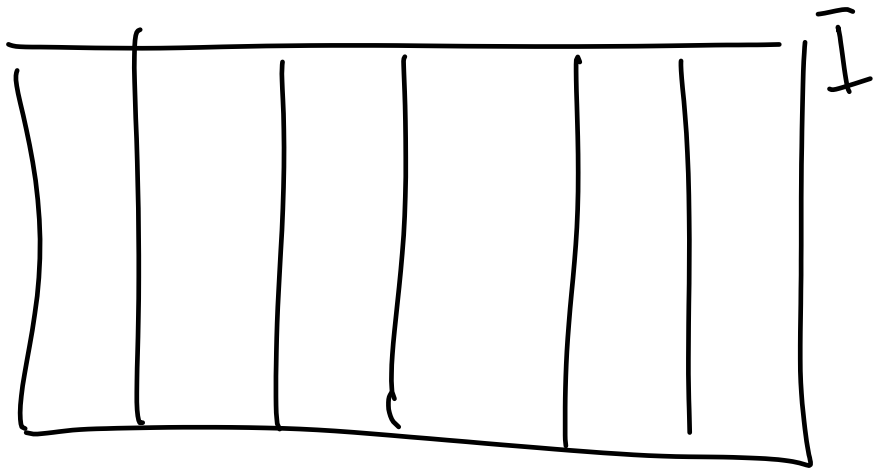
- **Regola della somma**: $P(A \text{ oppure } B|I) = P(A|I) + P(B|I)$, se A, B *incompatibili*.
- **Probabilità della negazione**: $P(\text{non } A|I) = 1 - P(A|I)$
- **Regola del prodotto**: $P(A \text{ e } B|I) = P(A|I)P(B|A \text{ e } I)$
- **Formula di Bayes**: $P(A|B) = P(A)P(B|A)/P(B)$, oppure

$$\text{Odds}(A|B) = \text{Odds}(A)\frac{\mathcal{L}(A;B)}{\mathcal{L}(\text{non } A;B)}$$

- **Verosimiglianza**: $\mathcal{L}(A;B) = P(B|A)$.
- **Indipendenza**: $P(A|B) = P(A)$. $P(B|A) = P(B)$

Variabili aleatorie

- **Sistema di alternative:** $(A_\ell)_{\ell \in E}$ eventi tali che *esattamente uno è sempre vero* (non si sa quale)



Variabili aleatorie

- **Sistema di alternative:** $(A_\ell)_{\ell \in E}$ eventi tali che *esattamente uno è sempre vero* (non si sa quale)
- **Esempio**: $A_\ell =$ *il dato test è di classe* ℓ, con $E = C$ (possibili classi)

Variabili aleatorie

- **Sistema di alternative:** $(A_\ell)_{\ell \in E}$ eventi tali che *esattamente uno è sempre vero* (non si sa quale)
- **Esempio**: $A_\ell =$ *il dato test è di classe* ℓ, con $E = C$ (possibili classi)
- Caratteristica aleatoria $X \in E$ è un sistema di alternative

$$A_x = \{X = x\} = \text{la caratteristica } X \text{ vale } x$$

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$
- Densità continua: $P(X \in [a, b]) = \int_a^b p(X = x)dx$

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$
- Densità continua: $P(X \in [a, b]) = \int_a^b p(X = x)dx$
- Notazione (*imprecisa ma comoda*) $P(X|I) = P(X = \cdot|I)$.

$P(X = \cdot)$

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$
- Densità continua: $P(X \in [a, b]) = \int_a^b p(X = x)dx$
- Notazione (*imprecisa ma comoda*) $P(X|I) = P(X = \cdot|I)$.
- **Regola del prodotto**:
  $P(X = x, L = \ell) = P(X = x|I)P(L = \ell|X = x)$.

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$
- Densità continua: $P(X \in [a, b]) = \int_a^b p(X = x)dx$
- Notazione (*imprecisa ma comoda*) $P(X|I) = P(X = \cdot|I)$.
- **Regola del prodotto**:
  $P(X = x, L = \ell) = P(X = x|I)P(L = \ell|X = x)$.
- **Formula di Bayes**:
  $P(L = \ell|X = x) = P(L = \ell)P(X = x|L = \ell)/P(X = x)$,
  oppure

$$\text{Odds}(L = \ell|X = x) = \text{Odds}(L = \ell)\frac{\mathcal{L}(\ell; x)}{\mathcal{L}(L \neq \ell; x)}$$

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$
- Densità continua: $P(X \in [a, b]) = \int_a^b p(X = x)dx$
- Notazione (*imprecisa ma comoda*) $P(X|I) = P(X = \cdot|I)$.
- **Regola del prodotto**:
  $P(X = x, L = \ell) = P(X = x|I)P(L = \ell|X = x)$.
- **Formula di Bayes**:
  $P(L = \ell|X = x) = P(L = \ell)P(X = x|L = \ell)/P(X = x)$,
  oppure

$$\text{Odds}(L = \ell|X = x) = \text{Odds}(L = \ell)\frac{\mathcal{L}(\ell; x)}{\mathcal{L}(L \neq \ell; x)}$$

- **Verosimiglianza**: $\mathcal{L}(L = \ell; x) = P(X = x|L = \ell)$.

Regole di calcolo (variabili)

- Densità discreta: $P(L \in U) = \sum_{\ell \in U} P(L = \ell)$
- Densità continua: $P(X \in [a, b]) = \int_a^b p(X = x)dx$
- Notazione (*imprecisa ma comoda*) $P(X|I) = P(X = \cdot|I)$.
- **Regola del prodotto**:
  $P(X = x, L = \ell) = P(X = x|I)P(L = \ell|X = x)$.
- **Formula di Bayes**:
  $P(L = \ell|X = x) = P(L = \ell)P(X = x|L = \ell)/P(X = x)$,
  oppure

$$\text{Odds}(L = \ell|X = x) = \text{Odds}(L = \ell)\frac{\mathcal{L}(\ell; x)}{\mathcal{L}(L \neq \ell; x)}$$

- **Verosimiglianza**: $\mathcal{L}(L = \ell; x) = P(X = x|L = \ell)$.
- **Indipendenza**: $P(X = x|Y = y) = P(X = x)$.

Esempi

- Densità *Bernoulli*: $X \in \{0, 1\}$,

$$P(X = x) = q^x (1 - q)^{1-x} = \begin{cases} q & \text{se } x = 1 \\ 1 - q & \text{se } x = 0 \end{cases}$$

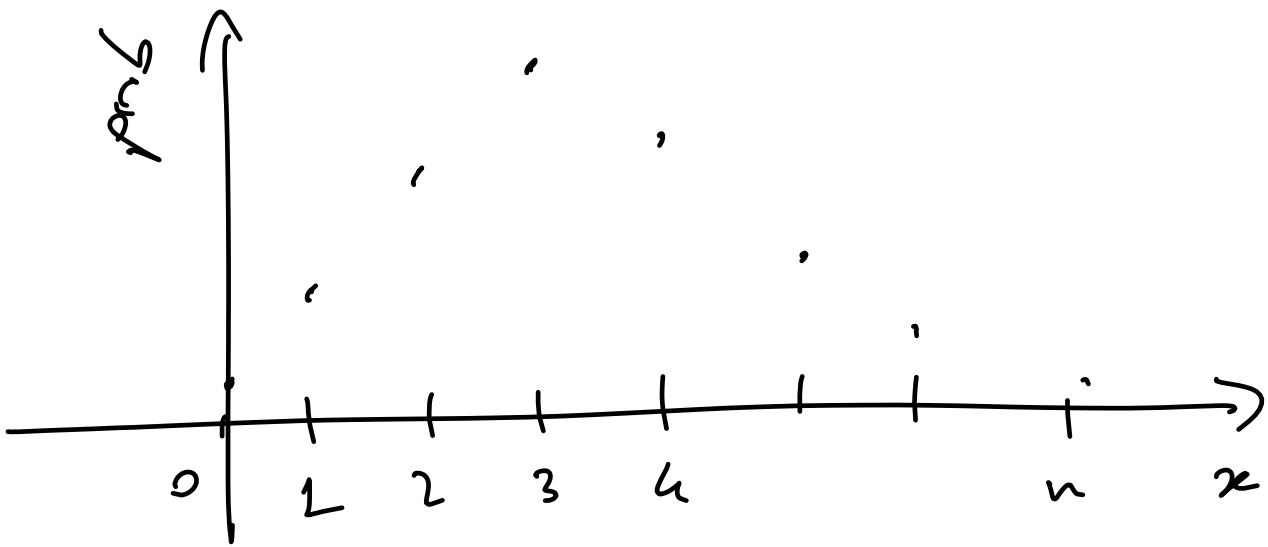
Esempi

- Densità *Bernoulli*: $X \in \{0, 1\}$,

$$P(X = x) = q^x (1 - q)^{1-x}$$

- Densità *Binomiale*: $X \in \{0, 1, \ldots, n\}$,

$$P(X = x) = \binom{n}{x} q^x (1 - q)^{n-x}$$



Esempi

- Densità *Bernoulli*: $X \in \{0, 1\}$,

$$P(X = x) = q^x (1 - q)^{1-x}$$

- Densità *Binomiale*: $X \in \{0, 1, \ldots, n\}$,

$$P(X = x) = \binom{n}{x} q^x (1 - q)^{n-x}$$

- Densità *Gaussiana*: $X \in \mathbb{R}$,

$$p(X = x) = \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \cdot \frac{1}{\sqrt{2\pi\sigma^2}}$$

Modello statistico per la classificazione

- Supponiamo che:

Modello statistico per la classificazione

- Supponiamo che:
  - i dati del training set $\{z_i = (x_i, \ell_i)\}_{i=1,\ldots,n}$ sono osservazioni di *variabili aleatorie indipendenti* tutte con la stessa legge (densità)

$$P(Z_1, \ldots, Z_n) = P(Z_1)P(Z_2)\ldots, P(Z_n)$$

Modello statistico per la classificazione

- Supponiamo che:
  - i dati del training set $\{z_i = (x_i, \ell_i)\}_{i=1,\ldots,n}$ sono osservazioni di *variabili aleatorie indipendenti* tutte con la stessa legge (densità)

$$P(Z_1, \ldots, Z_n) = P(Z_1)P(Z_2)\ldots, P(Z_n)$$

  - il dato (o i dati) del test set sono pure i.i.d. con la stessa legge $Z = (X, L)$.

Modello statistico per la classificazione

- Supponiamo che:
  - i dati del training set $\{z_i = (x_i, \ell_i)\}_{i=1,\dots,n}$ sono osservazioni di *variabili aleatorie indipendenti* tutte con la stessa legge (densità)

$$P(Z_1, \dots, Z_n) = P(Z_1)P(Z_2)\dots, P(Z_n)$$

  - il dato (o i dati) del test set sono pure i.i.d. con la stessa legge $Z = (X, L)$.
- Un classificatore $h : F \to C$ dipende da $\mathcal{D} = (Z_i)_{i=1}^n$ e l'*errore di generalizzazione* è il rischio atteso

$$P(h(X) \neq L) = \mathbb{E}[1_{h(X)\neq L}]$$

Come stimare ?

Nozione statistica di stimatore $T(z_1 \dots z_n)$

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h
  - *Test set* (il rimanente) per verificare l'accuratezza

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h
  - *Test set* (il rimanente) per verificare l'accuratezza
- **Problema**: troppo *rigida*, rischiamo di scegliere il classificatore in base a come abbiamo diviso.

Convalida incrociata (cross-validation) $\subset\!\!\surd$

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h
  - *Test set* (il rimanente) per verificare l'accuratezza
- **Problema**: troppo *rigida*, rischiamo di scegliere il classificatore in base a come abbiamo diviso.
- **Soluzione** è la convalida incrociata, in cui variamo la decomposizione *Train*/*Test* in modo da mitigare il rischio.

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h
  - *Test set* (il rimanente) per verificare l'accuratezza
- **Problema**: troppo *rigida*, rischiamo di scegliere il classificatore in base a come abbiamo diviso.
- **Soluzione** è la convalida incrociata, in cui variamo la decomposizione *Train*/*Test* in modo da mitigare il rischio.
- **In pratica**:

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h
  - *Test set* (il rimanente) per verificare l'accuratezza
- **Problema**: troppo *rigida*, rischiamo di scegliere il classificatore in base a come abbiamo diviso.
- **Soluzione** è la convalida incrociata, in cui variamo la decomposizione *Train*/*Test* in modo da mitigare il rischio.
- **In pratica**:
- *Train*/*Validation set* (80% dei dati) per calcolare vari h, stimare errori di generalizzazione, selezionare parametri ecc.

Convalida incrociata (cross-validation)

Obiettivo: *stimare* dai dati osservati l'errore di generalizzazione $P(h(X) \neq L)$

- **Idea naive**:
  - *Train set* (circa 80% dei dati) per calcolare il classificatore h
  - *Test set* (il rimanente) per verificare l'accuratezza
- **Problema**: troppo *rigida*, rischiamo di scegliere il classificatore in base a come abbiamo diviso.
- **Soluzione** è la convalida incrociata, in cui variamo la decomposizione *Train*/*Test* in modo da mitigare il rischio.
- **In pratica**:
- *Train/Validation set* (80% dei dati) per calcolare vari h, stimare errori di generalizzazione, selezionare parametri ecc.
- *Test set* per verifica finale sul modello selezionato.

Leave p-out (Lp0) CV

- Un metodo di CV:

.

Leave p-out (Lp0) CV

- Un metodo di CV:
  - fissare p (e.g., $p = 1$ o $p = 2 \ldots$)

Leave p-out (Lp0) CV

- Un metodo di CV:
  - fissare p (e.g., $p = 1$ o $p = 2 \ldots$)
  - scartare p elementi dal training set

Leave p-out (Lp0) CV

- Un metodo di CV:
  - fissare p (e.g., $p = 1$ o $p = 2 \ldots$)
  - scartare p elementi dal training set
  - calcolare l'accuratezza sui p elementi rimossi del classificatore h ottenuto (dal training set senza i p)

Leave p-out (Lp0) CV

- Un metodo di CV:
  - fissare p (e.g., $p = 1$ o $p = 2\ldots$)
  - scartare p elementi dal training set
  - calcolare l'accuratezza sui p elementi rimossi del classificatore h ottenuto (dal training set senza i p)
  - *mediare* l'accuratezza ottenuta variando su tutte (o molte) scelte di p elementi

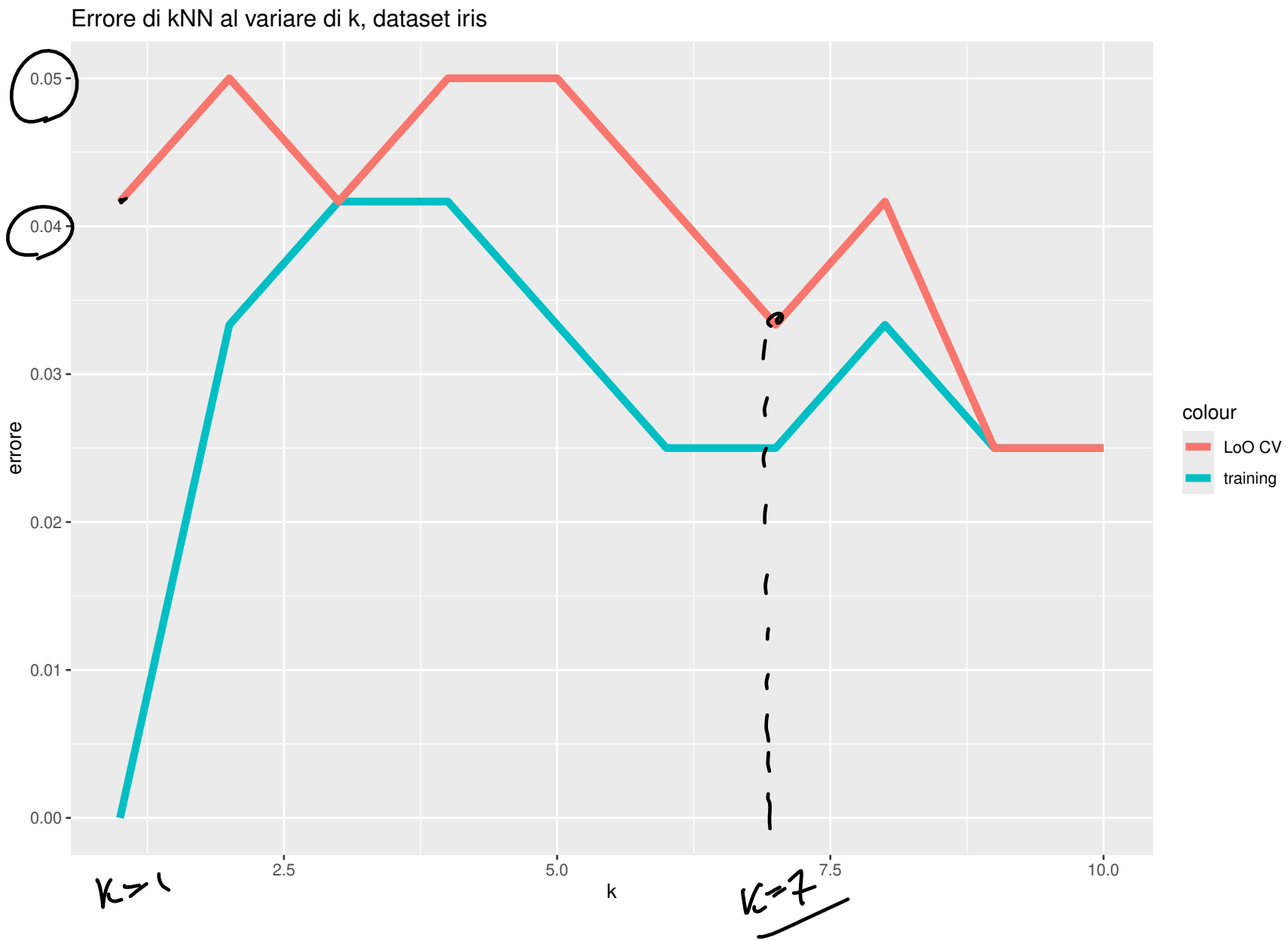
Leave p-out (Lp0) CV

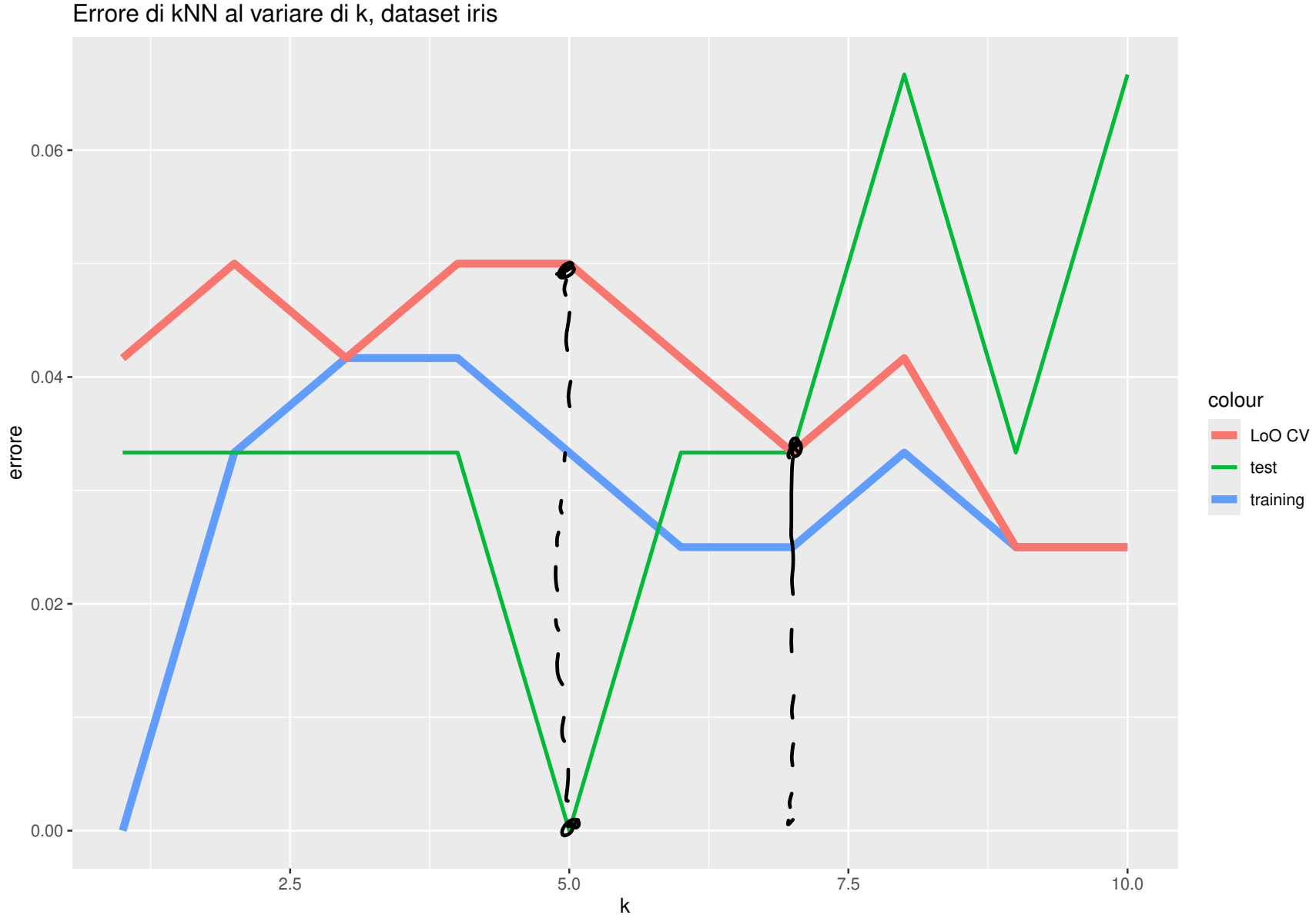
- Un metodo di CV:
  - fissare p (e.g., $p = 1$ o $p = 2 \ldots$)
  - scartare p elementi dal training set
  - calcolare l'accuratezza sui p elementi rimossi del classificatore h ottenuto (dal training set senza i p)
  - *mediare* l'accuratezza ottenuta variando su tutte (o molte) scelte di p elementi
- **Problema:** il numero *totale* di possibili scelte di p elementi da rimuovere su n osservazioni è

$$\binom{n}{p} \approx n^p.$$

L10 $p=1$ $\longrightarrow$ n

Esempio

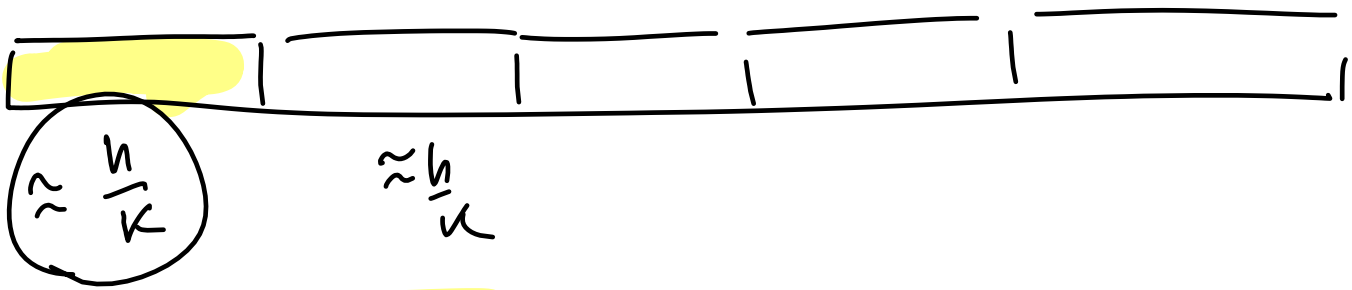

Errore di kNN al variare di k, dataset iris
errore
0.05
0.04
0.03
0.02
0.01
0.00
2.5
5.0
7.5
10.0
k
colour
LoO CV
training


Errore di kNN al variare di k, dataset iris
errore
0.06
0.04
0.02
0.00
2.5
5.0
7.5
10.0
k
colour
LoO CV
test
training

k-Fold CV

LpO CV richiede $\approx n^p$ iterazioni

- **Procedura**:



K

$\frac{n}{K} = p$

k-Fold CV

- **Procedura**:
  - fissare k (nulla a che vedere con il k di kNN)

k-Fold CV

- **Procedura**:
  - fissare k (nulla a che vedere con il k di kNN)
  - Dividere il dataset in k parti uguali (*folds*)

k-Fold CV

- **Procedura**:
  - fissare k (nulla a che vedere con il k di kNN)
  - Dividere il dataset in k parti uguali (*folds*)
  - Per ogni fold:

k-Fold CV

- **Procedura**:
  - fissare k (nulla a che vedere con il k di kNN)
  - Dividere il dataset in k parti uguali (*folds*)
  - Per ogni fold:
    - Utilizzare gli altri $k - 1$ folds per calcolare il classificatore

k-Fold CV

- **Procedura**:
  - fissare k (nulla a che vedere con il k di kNN)
  - Dividere il dataset in k parti uguali (*folds*)
  - Per ogni fold:
    - Utilizzare gli altri $k-1$ folds per calcolare il classificatore
    - Utilizzare il fold selezionato come validation set

k-Fold CV

- **Procedura**:
  - fissare k (nulla a che vedere con il k di kNN)
  - Dividere il dataset in k parti uguali (*folds*)
  - Per ogni fold:
    - Utilizzare gli altri $k-1$ folds per calcolare il classificatore
    - Utilizzare il fold selezionato come validation set
- *Esempio*: Con $k = 5$, il dataset è diviso in 5 parti; il modello viene addestrato su 4 parti e testato su 1, ripetendo l'operazione 5 volte.

Pro e contro di k-Fold CV

Vantaggi:

- **Meno costoso di LOOCV**: Richiede solo k passaggi

Svantaggi:

Pro e contro di k-Fold CV

Vantaggi:

- **Meno costoso di LOOCV**: Richiede solo k passaggi
- **Maggiore generalizzazione**: Ogni fold rappresenta una porzione del dataset.

Svantaggi:

Pro e contro di k-Fold CV

Vantaggi:

- **Meno costoso di LOOCV**: Richiede solo k passaggi
- **Maggiore generalizzazione**: Ogni fold rappresenta una porzione del dataset.

Svantaggi:

- **Scelta di k** può influenzare le stime

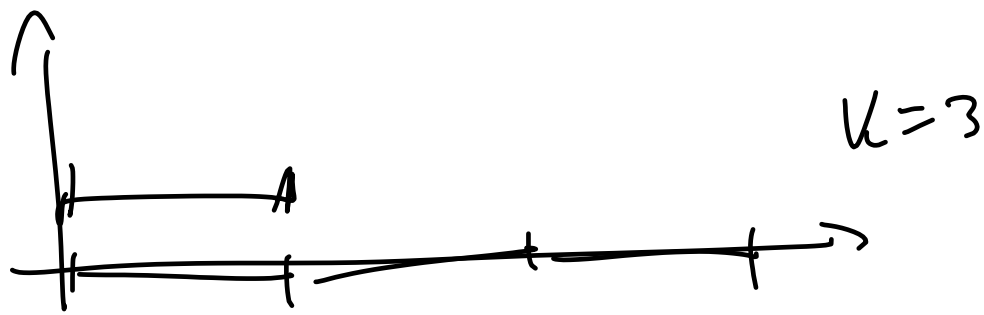
Pro e contro di k-Fold CV

Vantaggi:

- **Meno costoso di LOOCV**: Richiede solo k passaggi
- **Maggiore generalizzazione**: Ogni fold rappresenta una porzione del dataset.

Svantaggi:

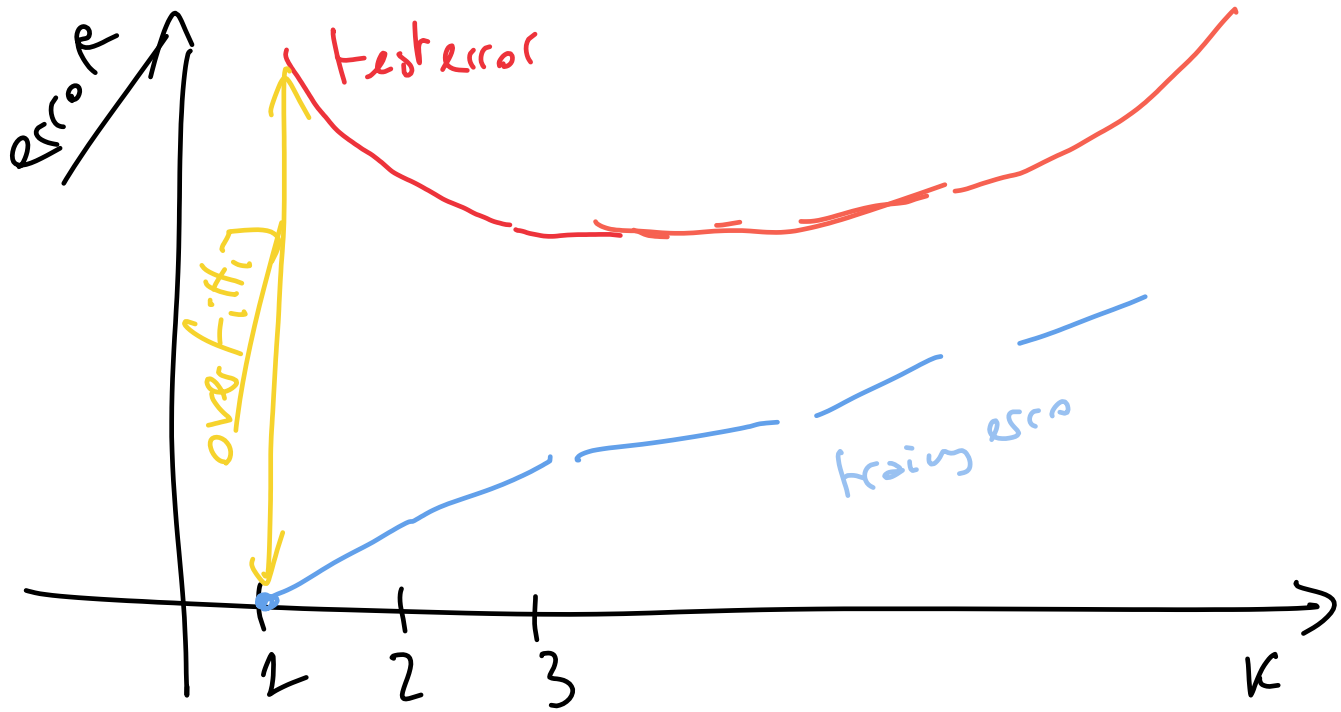
- **Scelta di k** può influenzare le stime
- **Possibili fluttuazioni** a seconda della suddivisione dei dati.



Compromesso bias-varianza

Per molti modelli le curve di errore al variare di un parametro di *regolarizzazione* interpolano tra due *estremi*:

- **Overfitting**: il classificatore h si adatta troppo ai dati di addestramento (k-NN con k piccolo) $\Rightarrow$ il *training error* è basso, il *test/validation* error è alto.



Compromesso bias-varianza

Per molti modelli le curve di errore al variare di un parametro di *regolarizzazione* interpolano tra due *estremi*:

- **Overfitting**: il classificatore h si adatta troppo ai dati di addestramento (k-NN con k piccolo) $\Rightarrow$ il *training error* è basso, il *test/validation* error è alto. VARIANZA
- **Underfitting**: il classificatore è troppo semplice per catturare la struttura dei dati (k-NN con k grande) $\Rightarrow$ il *training* error è alto (vicino al *test/validation* error) BIAS

Compromesso bias-varianza

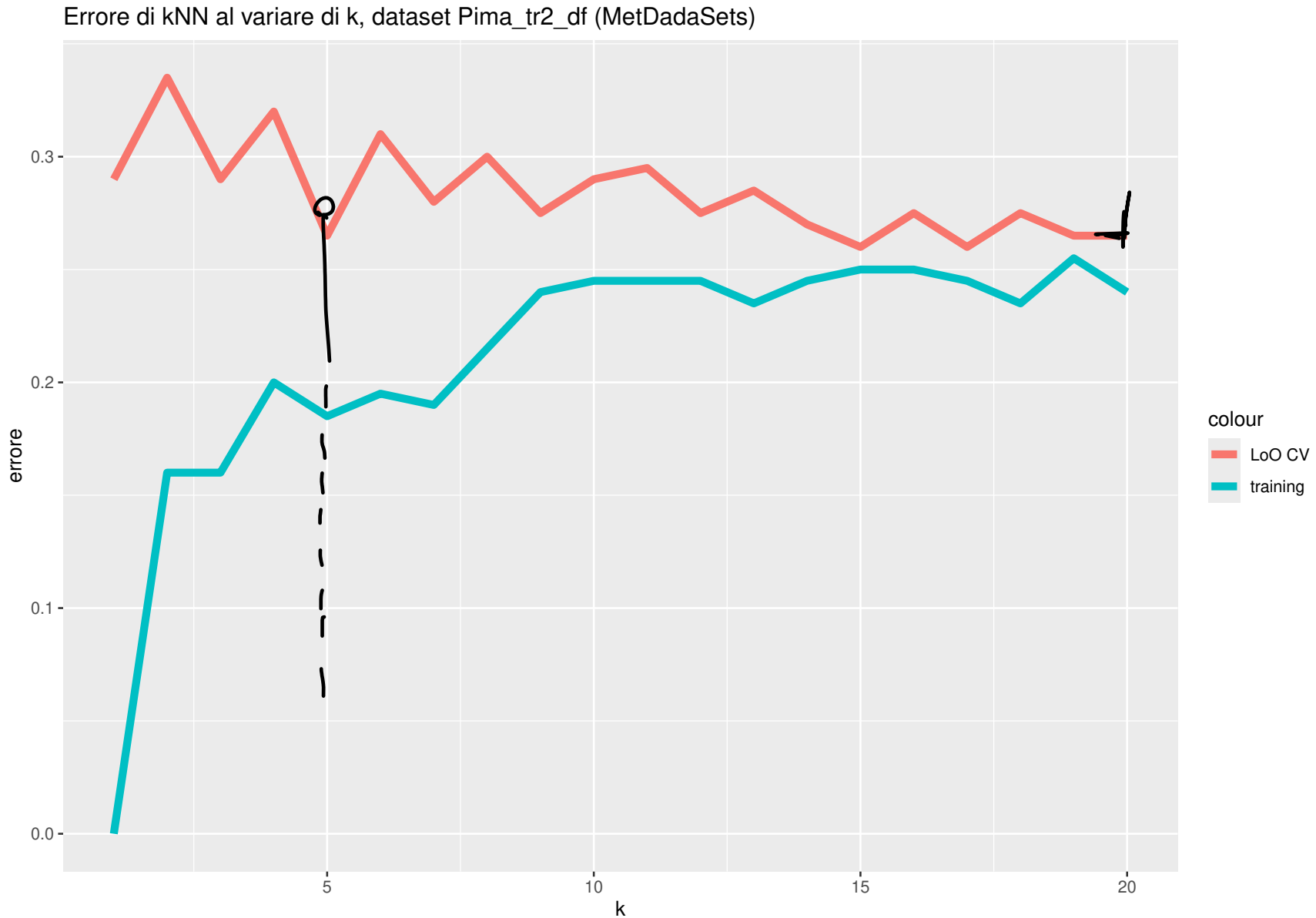
Per molti modelli le curve di errore al variare di un parametro di *regolarizzazione* interpolano tra due *estremi*:

- **Overfitting**: il classificatore h si adatta troppo ai dati di addestramento (k-NN con k piccolo) $\Rightarrow$ il *training error* è basso, il *test/validation* error è alto.
- **Underfitting**: il classificatore è troppo semplice per catturare la struttura dei dati (k-NN con k grande) $\Rightarrow$ il *training* error è alto (vicino al *test/validation* error)
- **Attenzione:** vedremo la decomposizione

$$Errore = Bias + Varianza + Rumore$$

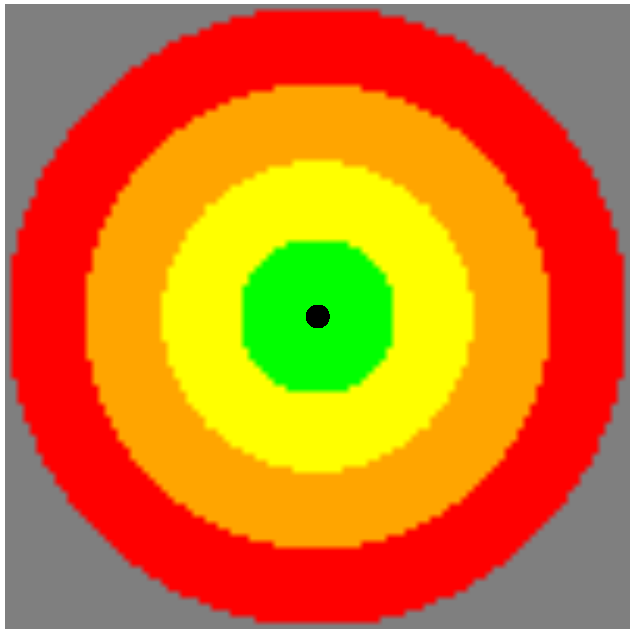
non è *sempre* vero in tutti i modelli il *compromesso* tra bias/varianza.

Esempio

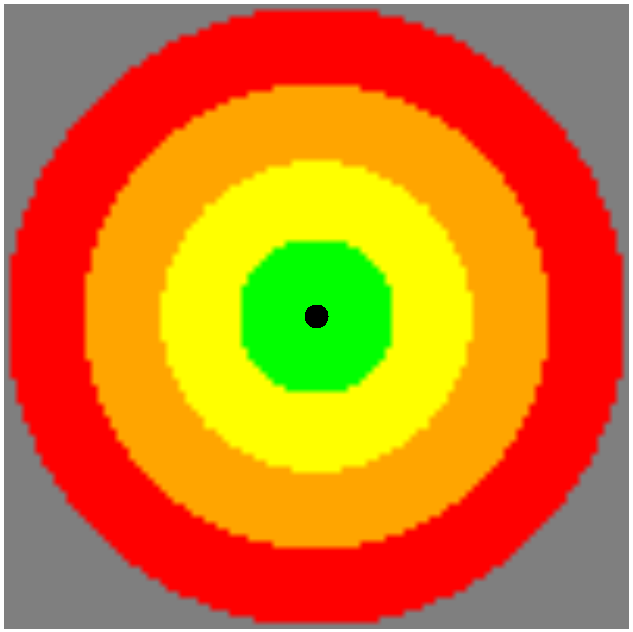

Errore di kNN al variare di k, dataset Pima_tr2_df (MetDadaSets)
errore
0.0
0.1
0.2
0.3
5
10
15
20
k
colour
LoO CV
training

Bias e varianza

Bias: Alto Varianza: Alto

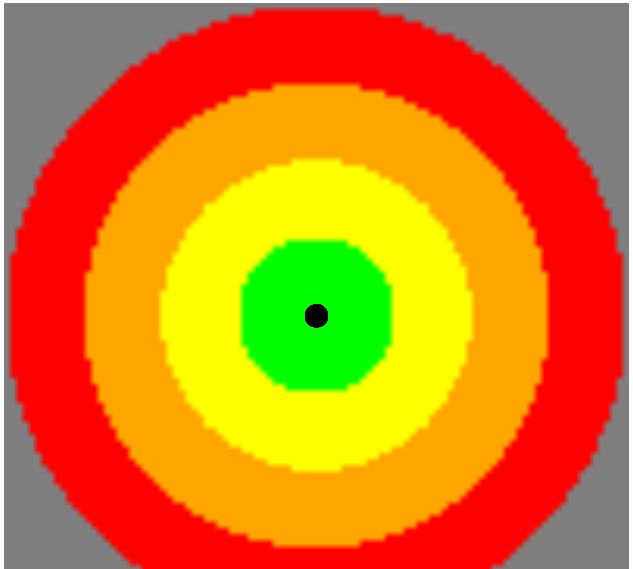


Bias: Alto Varianza: Basso



Bias: Basso Varianza: Alto

Bias: Basso Varianza: Basso



Classificatori binari:

- **Problema**: l'accuratezza potrebbe non essere un buon indicatore di prestazione.

Classificatori binari:

- **Problema**: l'accuratezza potrebbe non essere un buon indicatore di prestazione.
- Studiamo nel dettaglio il caso di *classi binarie* $C = \{-1, 1\}$ (Negativi, Positivi).

Classificatori binari:

- **Problema**: l'accuratezza potrebbe non essere un buon indicatore di prestazione.
- Studiamo nel dettaglio il caso di *classi binarie* $C = \{-1, 1\}$ (Negativi, Positivi).
- **Esempio**: se la **prevalenza** dei positivi $p(L = 1) \approx 1\%$ è piccola, allora $h(x) = -1$ per ogni $x \in F$ ha accuratezza grande: 99%

Matrice di confusione (confusion matrix)

- **Definizione** è una tabella di contingenza per valutare le prestazioni di un classificatore h.

	Predetto Positivo	Predetto Negativo
Positivo (Reale)	TP	FN
Negativo (Reale)	FP	TN

Matrice di confusione (confusion matrix)

- **Definizione** è una tabella di contingenza per valutare le prestazioni di un classificatore h.
- **Elementi della matrice**:

	Predetto Positivo	Predetto Negativo
Positivo (Reale)	*TP*	*FN*
Negativo (Reale)	*FP*	*TN*

Matrice di confusione (confusion matrix)

- **Definizione** è una tabella di contingenza per valutare le prestazioni di un classificatore h.
- **Elementi della matrice**:
  - **Vero Positivo (TP)**,

	Predetto Positivo	Predetto Negativo
Positivo (Reale)	*TP*	*FN*
Negativo (Reale)	*FP*	*TN*

Matrice di confusione (confusion matrix)

- **Definizione** è una tabella di contingenza per valutare le prestazioni di un classificatore h.
- **Elementi della matrice**:
  - **Vero Positivo (TP)**,
  - **Falso Positivo (FP)**, o *falso allarme*

	Predetto Positivo	Predetto Negativo
Positivo (Reale)	*TP*	*FN*
Negativo (Reale)	*FP*	*TN*

Matrice di confusione (confusion matrix)

- **Definizione** è una tabella di contingenza per valutare le prestazioni di un classificatore h.
- **Elementi della matrice**:
  - **Vero Positivo (TP)**,
  - **Falso Positivo (FP)**, o *falso allarme*
  - **Vero Negativo (TN)**,

	Predetto Positivo	Predetto Negativo
Positivo (Reale)	*TP*	*FN*
Negativo (Reale)	*FP*	*TN*

Matrice di confusione (confusion matrix)

- **Definizione** è una tabella di contingenza per valutare le prestazioni di un classificatore h.
- **Elementi della matrice**:
  - **Vero Positivo (TP)**,
  - **Falso Positivo (FP)**, o *falso allarme*
  - **Vero Negativo (TN)**,
  - **Falso Negativo (FN)**, o *mancato allarme*

	Predetto Positivo	Predetto Negativo
Positivo (Reale)	*TP*	*FN*
Negativo (Reale)	*FP*	*TN*

Esempio

Consideriamo il dataset `Pima_tr2_df` dal pacchetto `MedDataSets` e applichiamo k-nn su una decomposizione *Train*+*Test* di 100 + 100 osservazioni. Le classi sono l'ottava osservazione (presenza o assenza di diabete).

```
##       Classificato
## Reale No Yes
##    No 55  11
##    Yes 16 18
```

55+11 = N

16+18 = Yes

55+16 29

55+18+11+16 = n_{test} = 100

Calcolo delle Metriche

- **Accuratezza**:

$$\text{Acc} = \frac{TP + TN}{P + N} \approx P(h(X) = L)$$

Calcolo delle Metriche

- **Accuratezza**:

$$\text{Acc} = \frac{TP + TN}{P + N} \approx P(h(X) = L)$$

- **Prevalenza**:

$$\text{Prevalenza} = \frac{P}{P + N} \approx P(L = +1)$$

Calcolo delle Metriche

- **Accuratezza**:

$$\text{Acc} = \frac{TP + TN}{P + N} \approx P(h(X) = L)$$

- **Prevalenza**:

$$\text{Prevalenza} = \frac{P}{P + N} \approx P(L = +1)$$

- **Precisione**:

$$\text{Precisione} = \frac{TP}{TP + FP} \approx P(L = +1 | h(X) = +1)$$

Calcolo delle Metriche

- **Accuratezza**:

$$\text{Acc} = \frac{TP + TN}{P + N} \approx P(h(X) = L)$$

- **Prevalenza**:

$$\text{Prevalenza} = \frac{P}{P + N} \approx P(L = +1)$$

- **Precisione**:

$$\text{Precisione} = \frac{TP}{TP + FP} \approx P(L = +1 | h(X) = +1)$$

- **Potenza/tasso dei veri positivi/sensitività/richiamo**:

$$\text{TPR} = \frac{TP}{P} \approx P(h(X) = +1 | L = +1)$$

Calcolo delle Metriche

- **Accuratezza**:

$$\text{Acc} = \frac{TP + TN}{P + N} \approx P(h(X) = L)$$

- **Prevalenza**:

$$\text{Prevalenza} = \frac{P}{P + N} \approx P(L = +1)$$

- **Precisione**:

$$\text{Precisione} = \frac{TP}{TP + FP} \approx P(L = +1 | h(X) = +1)$$

- **Potenza/tasso dei veri positivi/sensitività/richiamo**:

$$\text{TPR} = \frac{TP}{P} \approx P(h(X) = +1 | L = +1)$$

- **Errore di tipo I/tasso dei falsi positivi**:

$$\text{FPR} = \frac{FP}{N} \approx P(h(X) = +1 | L = -1)$$

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

- **Calcolo delle Metriche**:

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

- **Calcolo delle Metriche**:
  - Accuratezza: $\frac{18+55}{100} = 73\%$

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

- **Calcolo delle Metriche**:
  - Accuratezza: $\frac{18+55}{100} = 73\%$
  - Prevalenza: $\frac{18+16}{100} = 34\%$

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

- **Calcolo delle Metriche**:
  - Accuratezza: $\frac{18+55}{100} = 73\%$
  - Prevalenza: $\frac{18+16}{100} = 34\%$
  - Precisione: $\frac{18}{18+11} = 62\%$

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

- **Calcolo delle Metriche**:
  - Accuratezza: $\frac{18+55}{100} = 73\%$
  - Prevalenza: $\frac{18+16}{100} = 34\%$
  - Precisione: $\frac{18}{18+11} = 62\%$
  - Potenza: $\frac{18}{18+16} = 53\%$

Esempio di Confusion Matrix

- Consideriamo il classificatore per il diabete trovato con k-NN

	Predetto Positivo	Predetto Negativo
Reale Positivo	18 (TP)	16 (FN)
Reale Negativo	11 (FP)	55 (TN)

- **Calcolo delle Metriche**:
  - Accuratezza: $\frac{18+55}{100} = 73\%$
  - Prevalenza: $\frac{18+16}{100} = 34\%$
  - Precisione: $\frac{18}{18+11} = 62\%$
  - Potenza: $\frac{18}{18+16} = 53\%$
  - Errore tipo I: $\frac{11}{11+55} = 17\%$

Commenti sulla Confusion Matrix

- **Vantaggi**:

Commenti sulla Confusion Matrix

- **Vantaggi**:
  - Fornisce informazioni dettagliate sulle prestazioni del classificatore.

Commenti sulla Confusion Matrix

- **Vantaggi**:
  - Fornisce informazioni dettagliate sulle prestazioni del classificatore.
  - Permette di identificare dove fallisce (FP e FN).

Commenti sulla Confusion Matrix

- **Vantaggi**:
  - Fornisce informazioni dettagliate sulle prestazioni del classificatore.
  - Permette di identificare dove fallisce (FP e FN).
- **Svantaggi**:

Commenti sulla Confusion Matrix

- **Vantaggi**:
  - Fornisce informazioni dettagliate sulle prestazioni del classificatore.
  - Permette di identificare dove fallisce (FP e FN).
- **Svantaggi**:
  - Non fornisce informazioni sulla probabilità di errore (passare ai tassi di falsi positivi/negativi).

Commenti sulla Confusion Matrix

- **Vantaggi**:
  - Fornisce informazioni dettagliate sulle prestazioni del classificatore.
  - Permette di identificare dove fallisce (FP e FN).
- **Svantaggi**:
  - Non fornisce informazioni sulla probabilità di errore (passare ai tassi di falsi positivi/negativi).
  - Difficile confrontare matrici di classificatori diversi.

Grafico FPR/TPR

Idea: dato un classificatore h, plottiamo la coppia di metriche (FPR, TPR) (*errore I specie*, *potenza*) nel quadrato $[0,1]^2$.

- Un classificatore ideale ha valori $(0,1)$ (angolo in alto a sx).

Grafico FPR/TPR

Idea: dato un classificatore h, plottiamo la coppia di metriche (FPR, TPR) (*errore I specie*, *potenza*) nel quadrato $[0, 1]^2$.

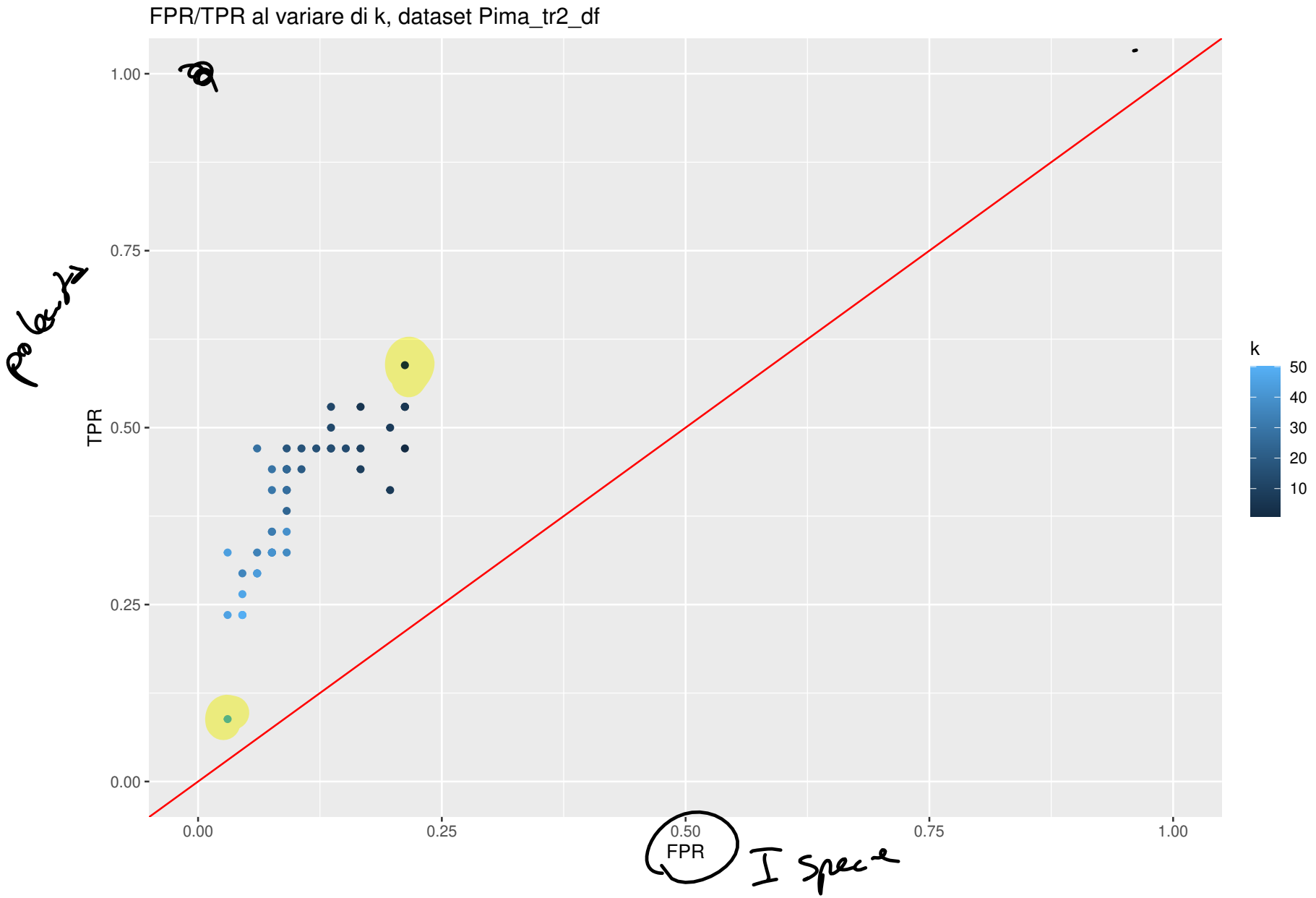
- Un classificatore ideale ha valori $(0, 1)$ (angolo in alto a sx).
- Un classificatore completamente random ha valori sulla diagonale.

Grafico FPR/TPR

Idea: dato un classificatore h, plottiamo la coppia di metriche (FPR, TPR) (*errore I specie*, *potenza*) nel quadrato $[0, 1]^2$.

- Un classificatore ideale ha valori $(0, 1)$ (angolo in alto a sx).
- Un classificatore completamente random ha valori sulla diagonale.
- Più il punto è in alto a sinistra, meglio è la performance del modello.

Esempio (k-NN al variare di k)



Curva ROC e AUC

- *Spesso* un classificatore h permette di introdurre un parametro t di soglia.

Curva ROC e AUC

- *Spesso* un classificatore h permette di introdurre un parametro t di soglia.
- Esempio: k-NN con k-fissato: si valuta $h(x) = +1$ se il numero di vicini positivi tra i k è $> t$ (k-NN standard si ottiene per $t = k/2$)

Curva ROC e AUC

- *Spesso* un classificatore h permette di introdurre un parametro t di soglia.
- Esempio: k-NN con k-fissato: si valuta $h(x) = +1$ se il numero di vicini positivi tra i k è $> t$ (k-NN standard si ottiene per $t = k/2$)
- Si plotta allora la curva ottenuta al variare del parametro detta *curva ROC*.

Curva ROC e AUC

- *Spesso* un classificatore h permette di introdurre un parametro t di soglia.
- Esempio: k-NN con k-fissato: si valuta $h(x) = +1$ se il numero di vicini positivi tra i k è $> t$ (k-NN standard si ottiene per $t = k/2$)
- Si plotta allora la curva ottenuta al variare del parametro detta *curva ROC*.
- L'area sotto la curva (*AUC*) è spesso usato come (singolo) indicatore della performance del modello.