Statistica II (750AA)

Lezione 6

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

27/10/2025

Regressione

Regressione

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

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

dove

Regressione

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

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

dove

- $x_i \in F$ sono **variabili di input** (fattori di ingresso, regressori, predittori, covariate, variabili esplicative, variabili indipendenti)

Regressione

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

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

dove

- $x_i \in F$ sono **variabili di input** (fattori di ingresso, regressori, predittori, covariate, variabili esplicative, variabili indipendenti)
- $y_i \in C$ sono **variabili di output** (fattori di uscita, risposte, variabili dipendenti)

Regressione

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

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

dove

  - $x_i \in F$ sono **variabili di input** (fattori di ingresso, regressori, predittori, covariate, variabili esplicative, variabili indipendenti)
  - $y_i \in C$ sono **variabili di output** (fattori di uscita, risposte, variabili dipendenti)

- determinare un **modello di regressione**:

$$h : F \to C, \quad h(x) \approx y$$

Regressione

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

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

dove

  - $x_i \in F$ sono **variabili di input** (fattori di ingresso, regressori, predittori, covariate, variabili esplicative, variabili indipendenti)
  - $y_i \in C$ sono **variabili di output** (fattori di uscita, risposte, variabili dipendenti)

- determinare un **modello di regressione**:

$$h : F \to C, \quad h(x) \approx y$$

- **Osservazione:** se $C = \{-1, 1\} \to$ classificazione (binaria).

Regressione

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

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

dove

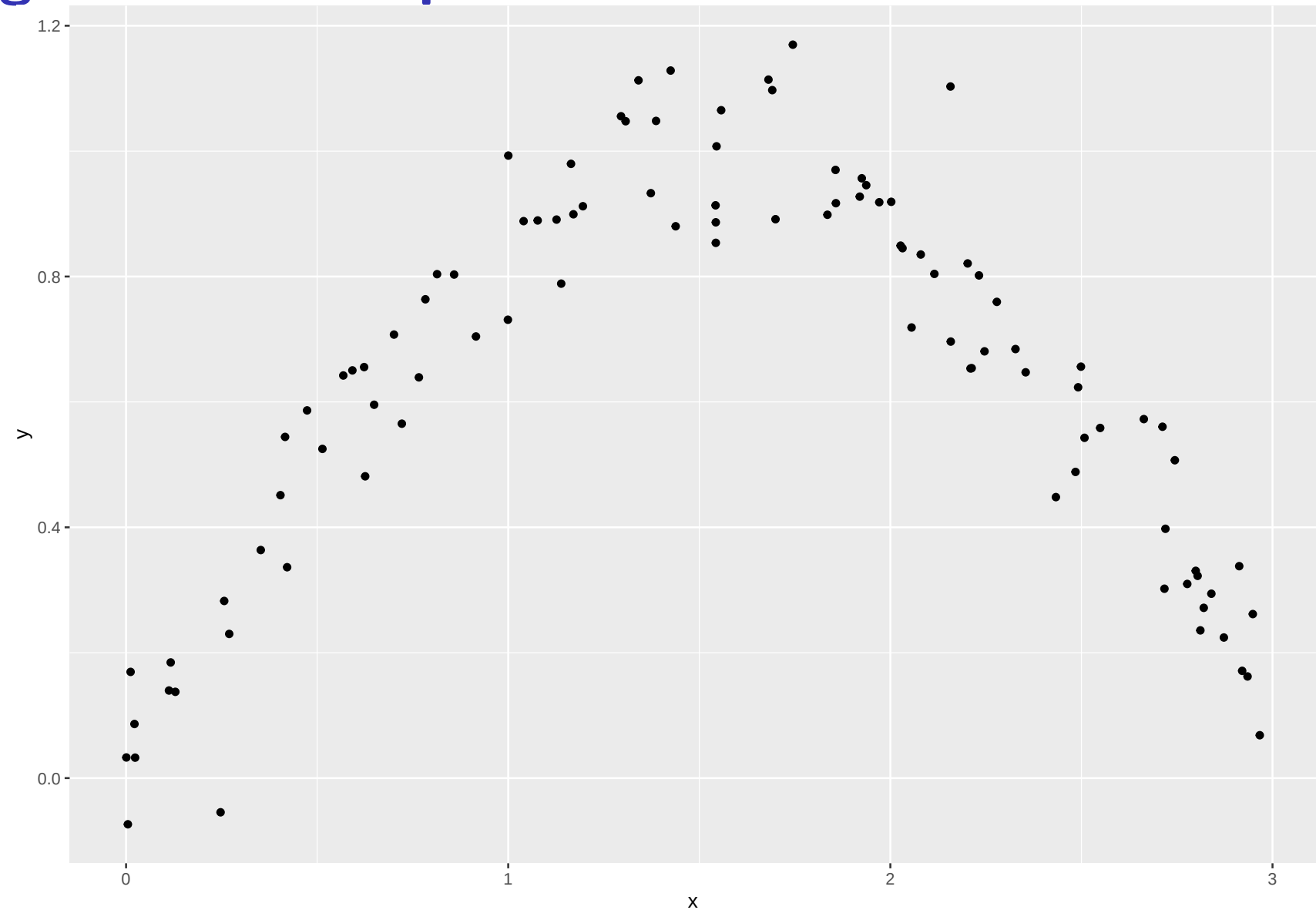
  - $x_i \in F$ sono **variabili di input** (fattori di ingresso, regressori, predittori, covariate, variabili esplicative, variabili indipendenti)
  - $y_i \in C$ sono **variabili di output** (fattori di uscita, risposte, variabili dipendenti)

- determinare un **modello di regressione**:

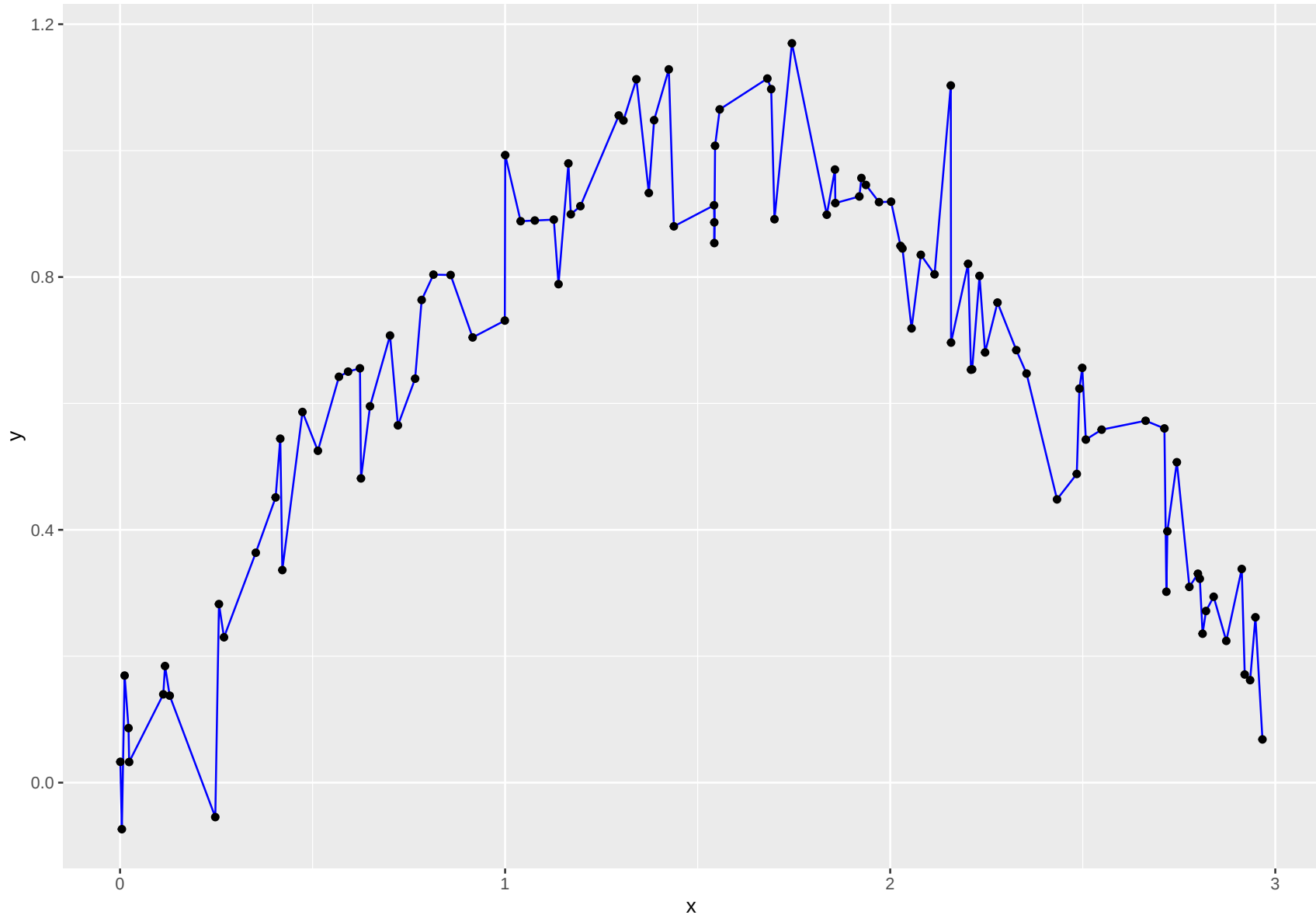
$$h : F \to C, \quad h(x) \approx y$$

- **Osservazione:** se $C = \{-1, 1\} \to$ classificazione (binaria).
- **Ipotesi:** $F \subseteq \mathbb{R}^d$ e $C = \mathbb{R}^k$ (per semplicità, $k = 1$).

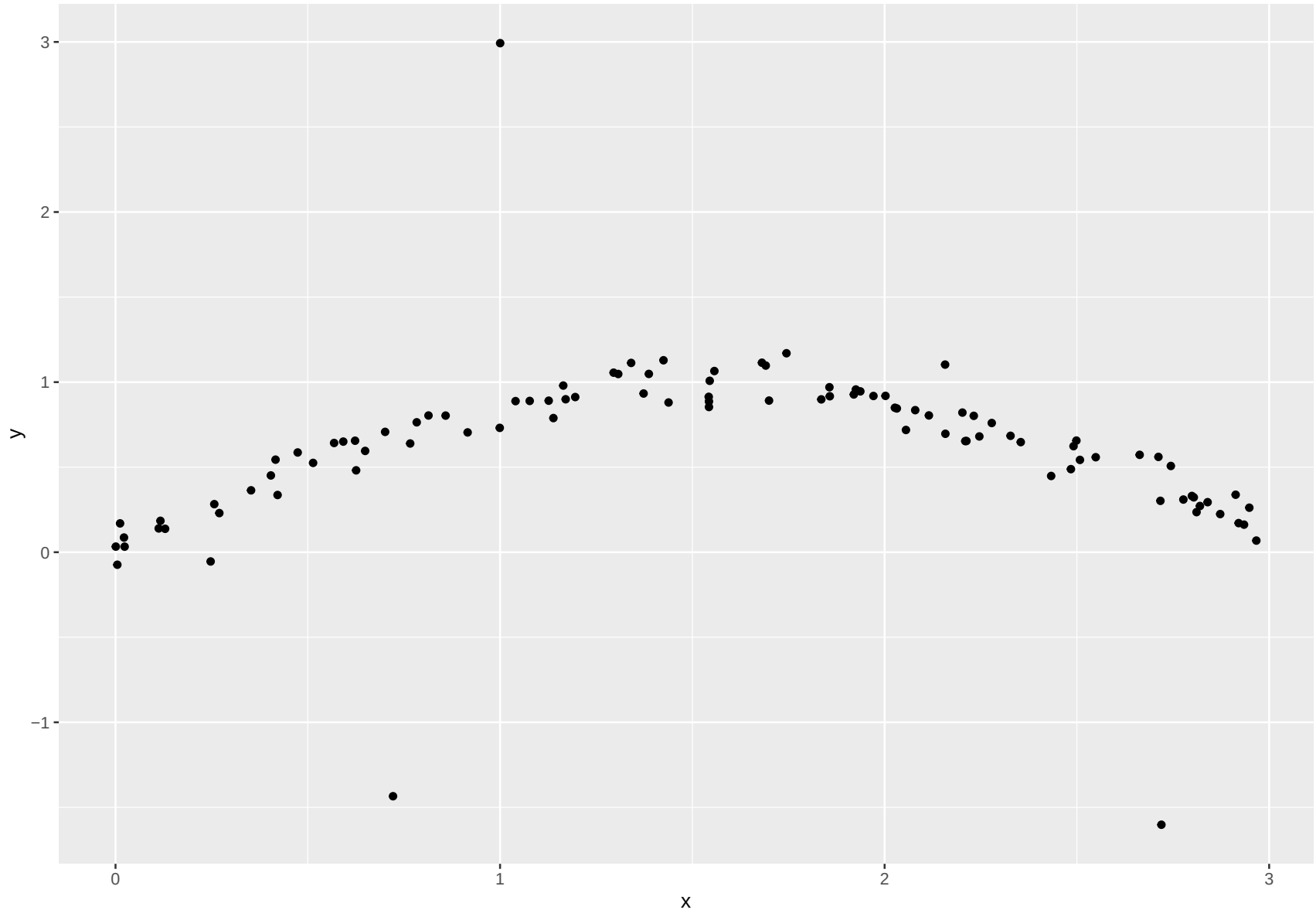
Legame con interpolazione

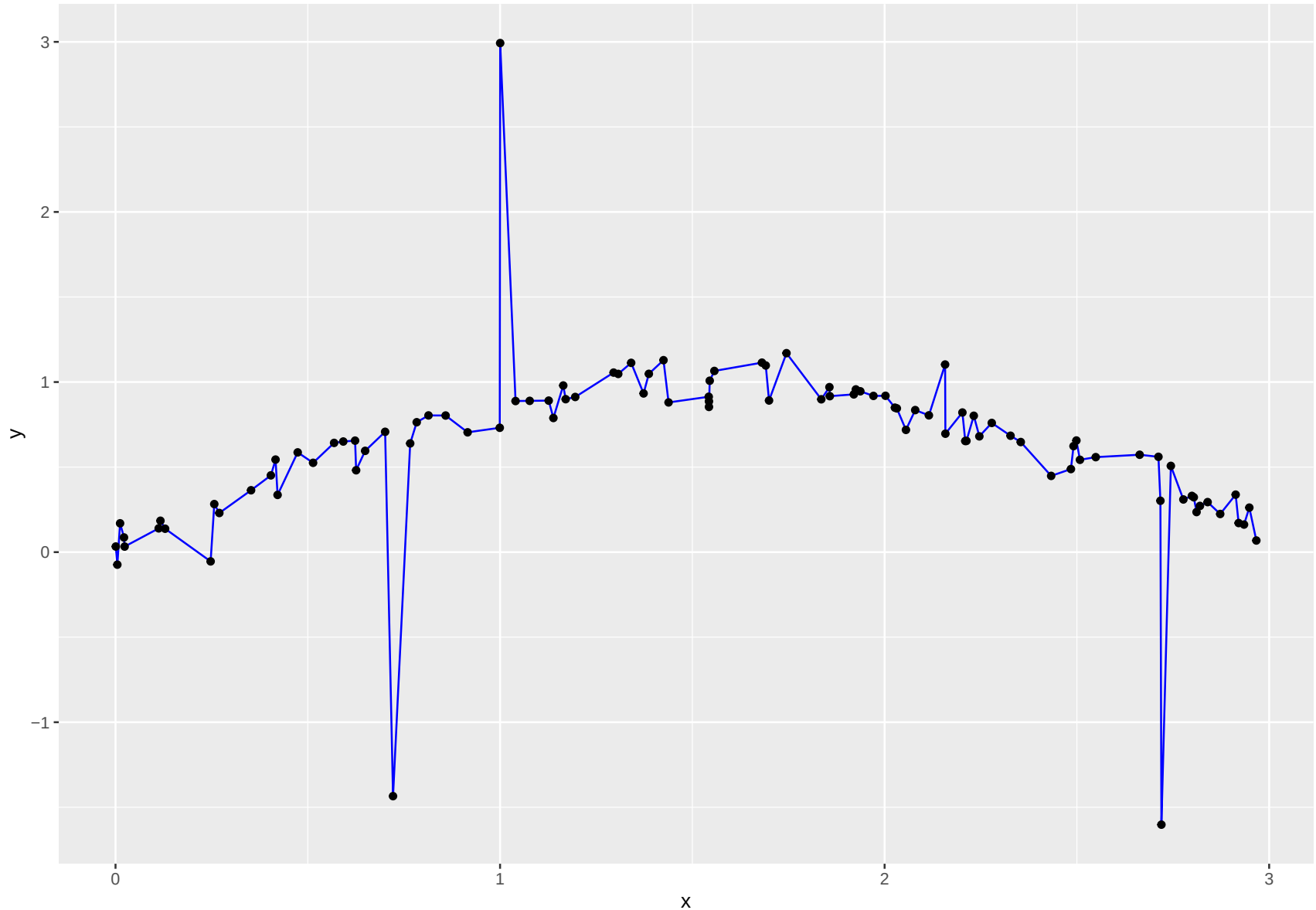


- Se $h(x_i) = y_i$ per ogni $i = 1, \ldots, n \rightarrow$ **interpolazione**.


1.2
0.8
0.4
0.0
y
0
1
2
3
x

Esempio con outliers




3
2
1
0
−1
y
0
1
2
3
x

K-NN per regressione

- **Idea**: estendiamo KNN per la regressione.

K-NN per regressione

- **Idea**: estendiamo KNN per la regressione.
- *Regressione ai k-primi vicini*: 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 caratteristiche **più vicine** a x e definiamo

$$h(x) = \frac{y_{i_1(x)} + y_{i_2(x)} + \ldots + y_{i_k(x)}}{k}.$$

K-NN per regressione

- **Idea**: estendiamo KNN per la regressione.
- *Regressione ai k-primi vicini*: 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 caratteristiche **più vicine** a x e definiamo

$$h(x) = \frac{y_{i_1(x)} + y_{i_2(x)} + \ldots + y_{i_k(x)}}{k}.$$

- **Osservazione:** per $k = 1$, $h(x)$ è il valore di output del punto più vicino a x.

K-NN per regressione

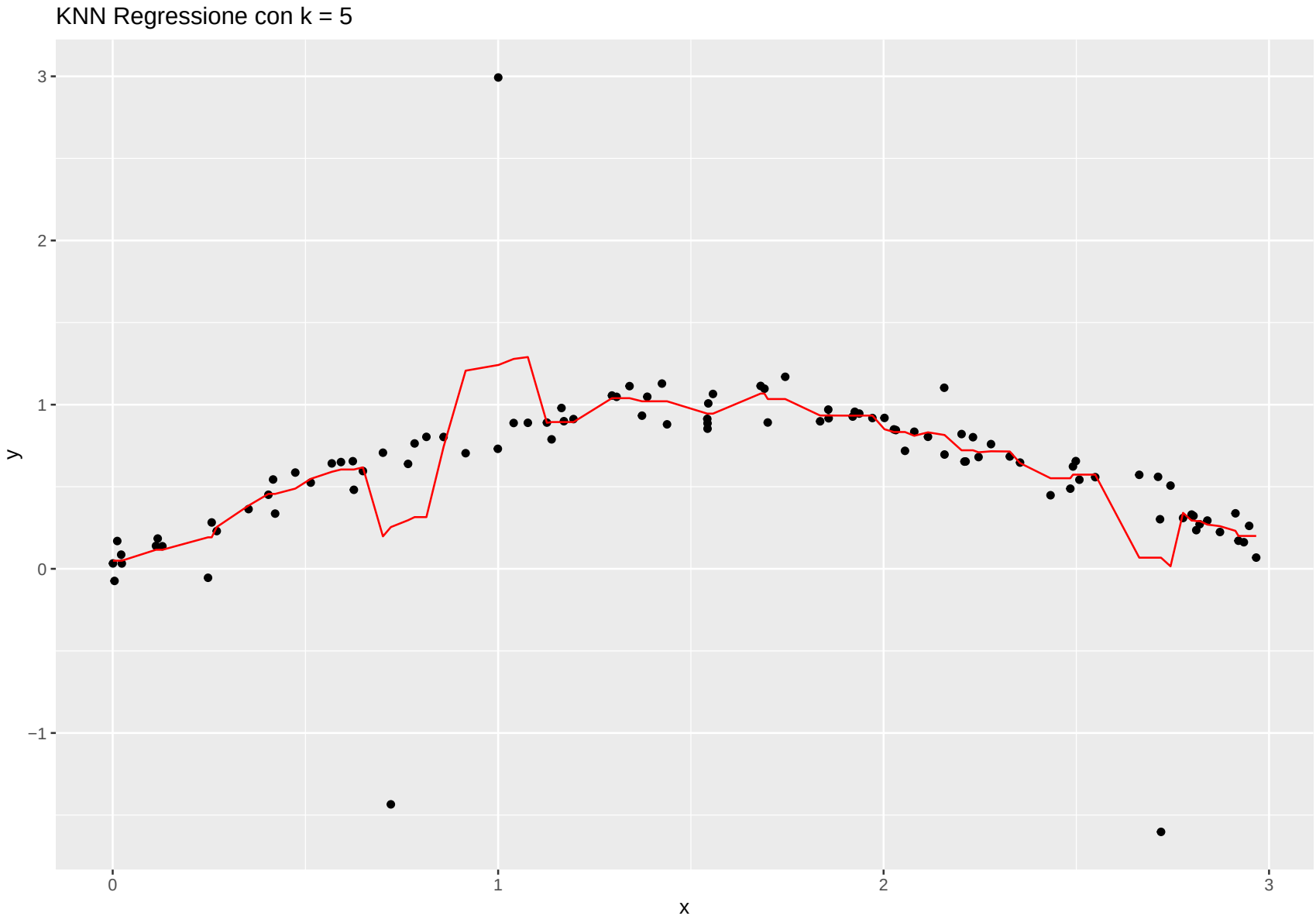
- **Idea**: estendiamo KNN per la regressione.
- *Regressione ai k-primi vicini*: dato $x \in F \subseteq \mathbb{R}^d$, troviamo i k punti nel *training set*

$$i_1(x), i_2(x), \ldots, i_k(x)$$

con caratteristiche **più vicine** a x e definiamo

$$h(x) = \frac{y_{i_1(x)} + y_{i_2(x)} + \ldots + y_{i_k(x)}}{k}.$$

- **Osservazione:** per $k = 1$, $h(x)$ è il valore di output del punto più vicino a x.
- Vantaggi (semplicità) e svantaggi (scelta di k, curse of dimensionality) come di KNN per classificazione.


KNN Regressione con k = 5
3
2
1
0
−1
y
0
1
2
3
x

Indicatori di performance

Per la classificazione l'accuratezza è un indicatore di performance:

$$Acc(h) = \sum_i 1_{\{h(x_i)=y_i\}}$$

Per regressione: quale funzione di perdita (loss) $\ell(h(x), y)$ utilizzare?

- **Errore quadratico medio (MSE)**: loss quadratica

$$MSE(h) = \frac{1}{n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

Root Mean Squared Error (RMSE) $:= \sqrt{MSE}$.

Indicatori di performance

Per la classificazione l'accuratezza è un indicatore di performance:

$$Acc(h) = \sum_i 1_{\{h(x_i)=y_i\}}$$

Per regressione: quale funzione di perdita (loss) $\ell(h(x), y)$ utilizzare?

- **Errore quadratico medio (MSE)**: loss quadratica

$$MSE(h) = \frac{1}{n}\sum_{i=1}^{n}(h(x_i) - y_i)^2$$

  Root Mean Squared Error (RMSE) $:= \sqrt{MSE}$.

- **Errore assoluto medio (MAE)**:

$$MAE(h) = \frac{1}{n}\sum_{i=1}^{n}|h(x_i) - y_i|$$

Errori di train, validation, test e generalizzazione

Fissata la loss $\ell(h(x), y)$, valgono le *stesse osservazioni della classificazione*: l'obiettivo finale non è minimizzare l'errore sul training set, ma **generalizzare** bene su dati nuovi!

- **Ipotesi**: le osservazioni (x_i, y_i) sono un campione i.i.d. con legge $P(X, Y) \rightarrow$ l'*errore di generalizzazione* (o *rischio*) è

$$L(h) := \mathbb{E}[\ell(h(X), Y)]$$

Errori di train, validation, test e generalizzazione

Fissata la loss $\ell(h(x), y)$, valgono le *stesse osservazioni della classificazione*: l'obiettivo finale non è minimizzare l'errore sul training set, ma **generalizzare** bene su dati nuovi!

- **Ipotesi**: le osservazioni (x_i, y_i) sono un campione i.i.d. con legge $P(X, Y) \rightarrow$ l'*errore di generalizzazione* (o *rischio*) è

$$L(h) := \mathbb{E}[\ell(h(X), Y)]$$

- L'errore di generalizzazione non è calcolabile (non conosciamo $P(X, Y)$) $\rightarrow$ *errore di test*:

$$L_{test}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i^{test}), y_i^{test})$$

Errori di train, validation, test e generalizzazione

Fissata la loss $\ell(h(x), y)$, valgono le *stesse osservazioni della classificazione*: l'obiettivo finale non è minimizzare l'errore sul training set, ma **generalizzare** bene su dati nuovi!

- **Ipotesi**: le osservazioni (x_i, y_i) sono un campione i.i.d. con legge $P(X, Y) \rightarrow$ l'*errore di generalizzazione* (o *rischio*) è

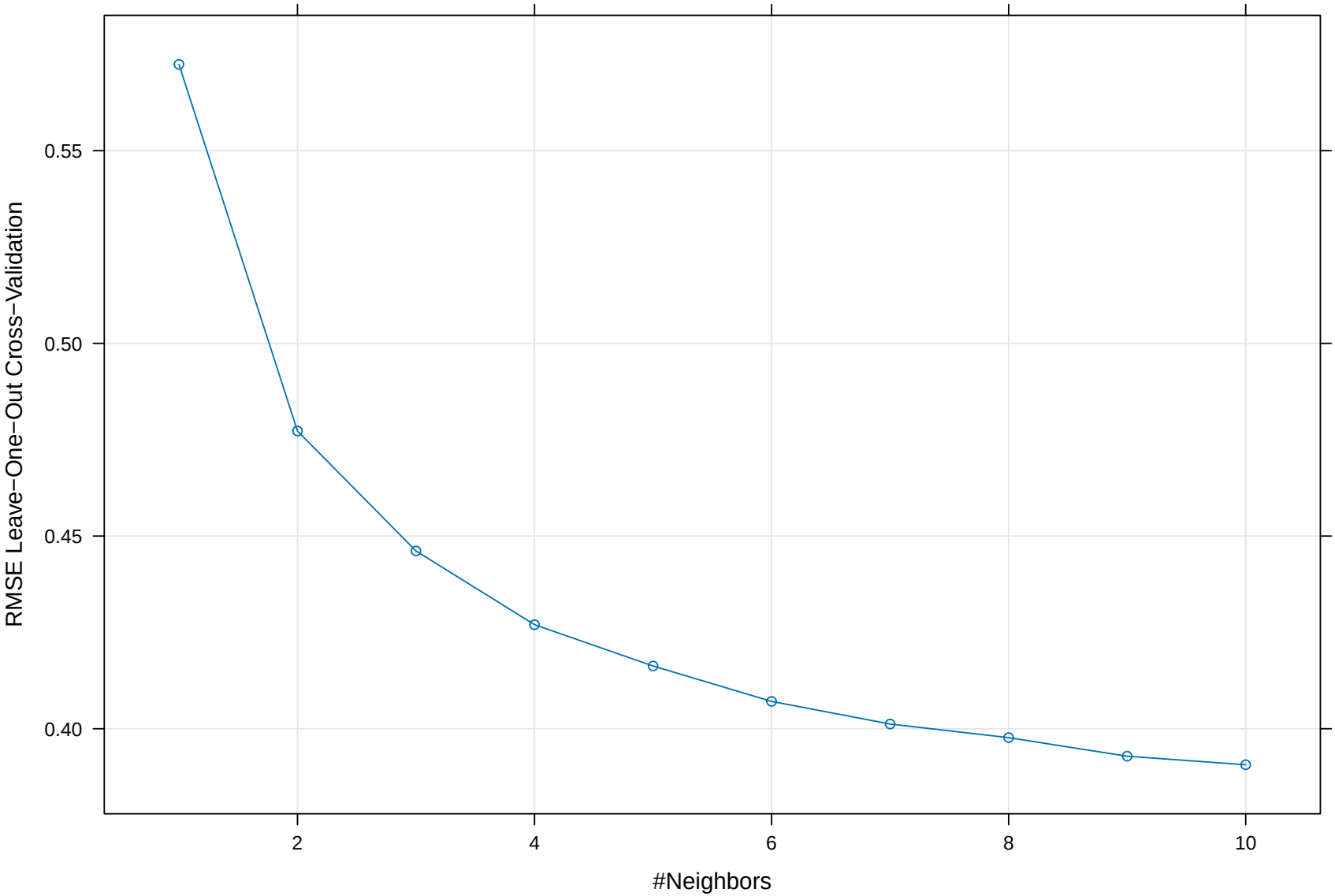
$$L(h) := \mathbb{E}[\ell(h(X), Y)]$$

- L'errore di generalizzazione non è calcolabile (non conosciamo $P(X, Y)$) $\rightarrow$ *errore di test*:

$$L_{test}(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i^{test}), y_i^{test})$$

- Per stime più affidabili $\rightarrow$ schema di *cross-validation* (k-fold, LkO-CV) sul training set.

Esempio di cross validation sul dataset generato



Altri indicatori di performance

Sia la RMSE che la MAE sono indicatori *in scala* (dipendono dall'unità di misura delle variabili di output). Possiamo riscalarli?

- **Errore percentuale assoluto medio (MAPE)**:

$$MAPE(h) = \frac{1}{n} \sum_{i=1}^{n} \frac{|h(x_i) - y_i|}{|y_i|}$$

Altri indicatori di performance

Sia la RMSE che la MAE sono indicatori *in scala* (dipendono dall'unità di misura delle variabili di output). Possiamo riscalarli?

- **Errore percentuale assoluto medio (MAPE)**:

$$MAPE(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{|h(x_i) - y_i|}{|y_i|}$$

- **R-quadro (coefficiente di determinazione)**:

$$R^2(h) = 1 - \frac{\sum_{i=1}^{n}(h(x_i) - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

dove $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

Altri indicatori di performance

Sia la RMSE che la MAE sono indicatori *in scala* (dipendono dall'unità di misura delle variabili di output). Possiamo riscalarli?

- **Errore percentuale assoluto medio (MAPE)**:

$$MAPE(h) = \frac{1}{n}\sum_{i=1}^{n}\frac{|h(x_i) - y_i|}{|y_i|}$$

- **R-quadro (coefficiente di determinazione)**:

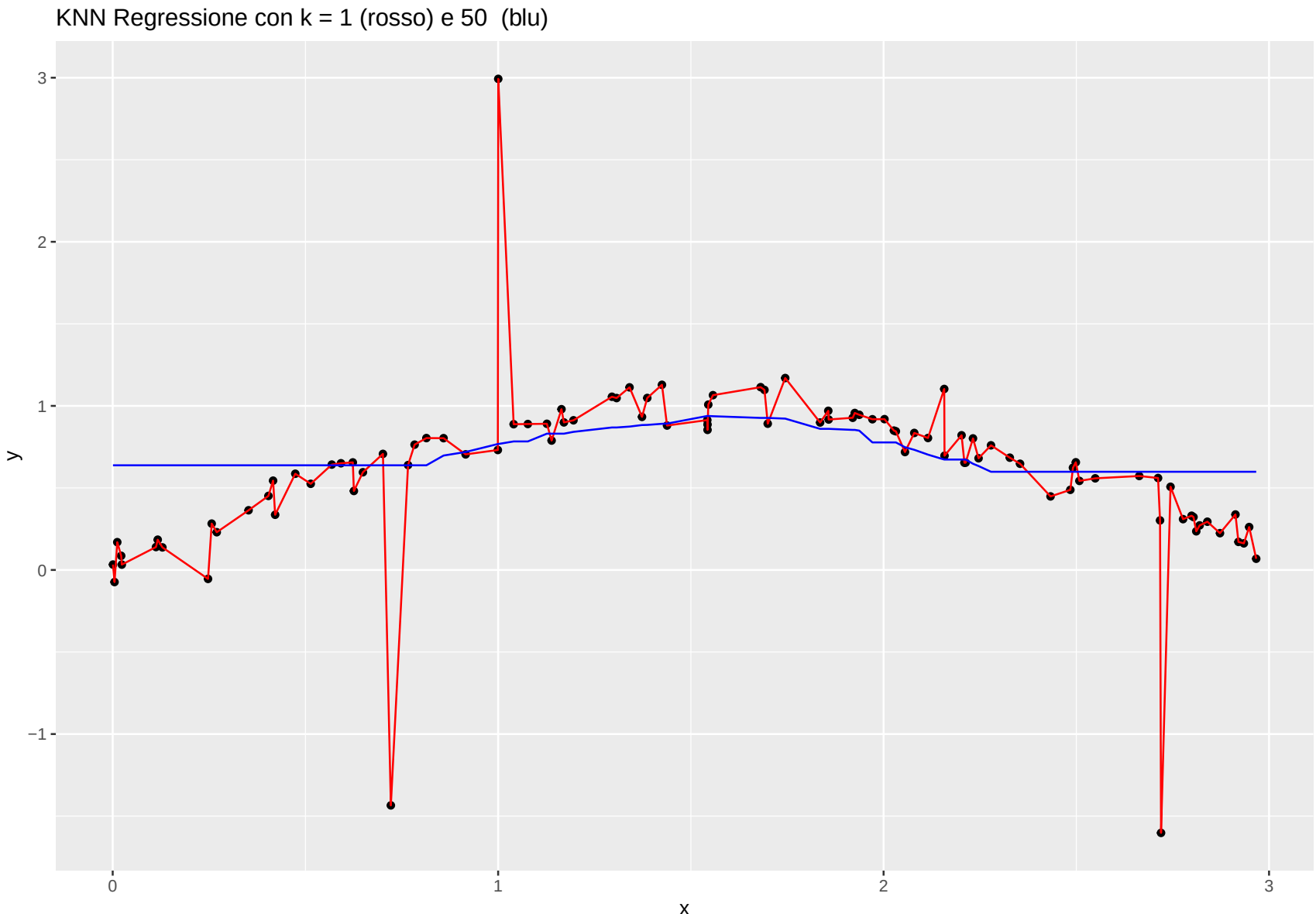
$$R^2(h) = 1 - \frac{\sum_{i=1}^{n}(h(x_i) - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

  dove $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

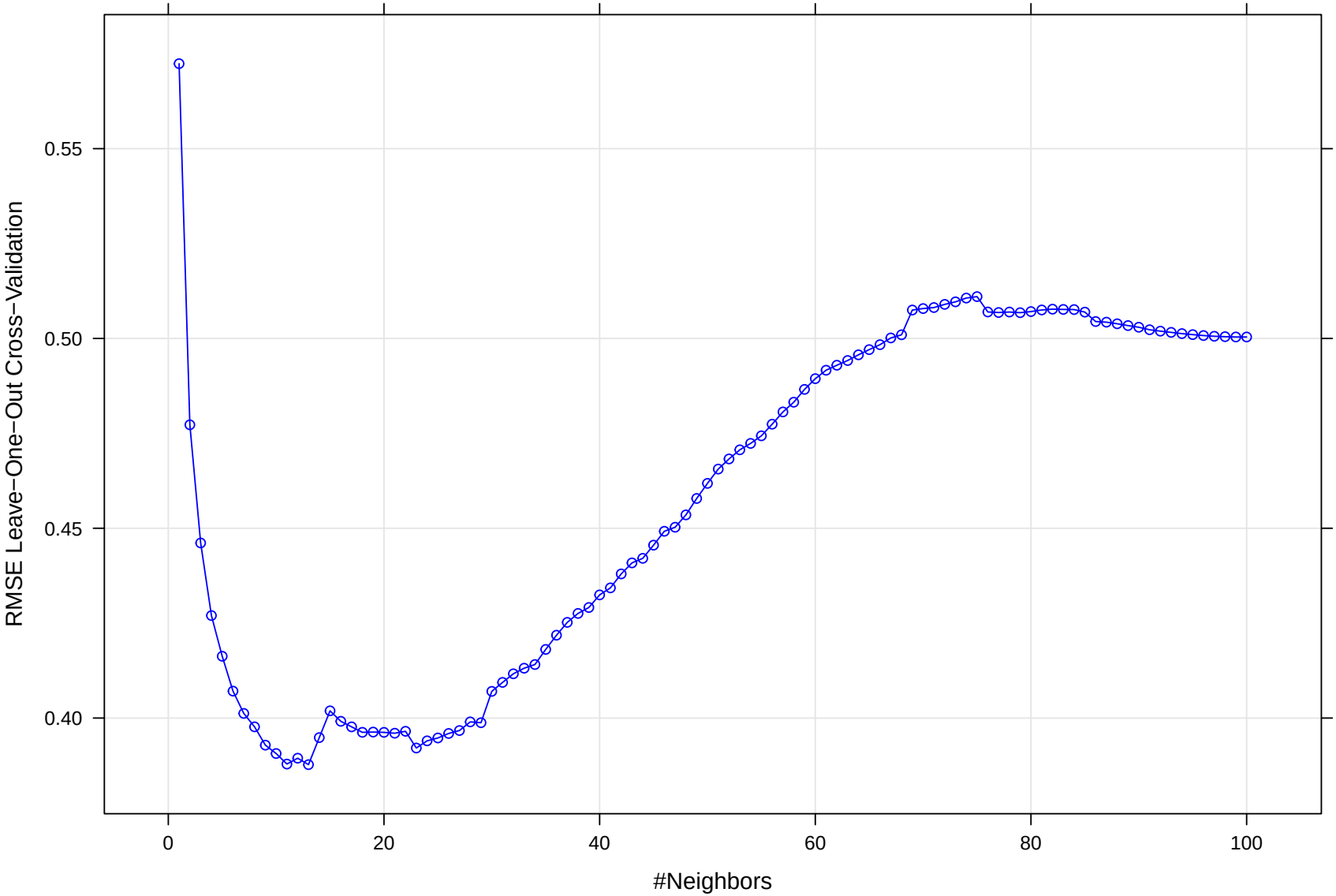
- **Osservazione:** R^2 misura la frazione di varianza spiegata dal modello (massimo 1, negativo se il modello è peggiore della media).

Decomposizione dell'errore

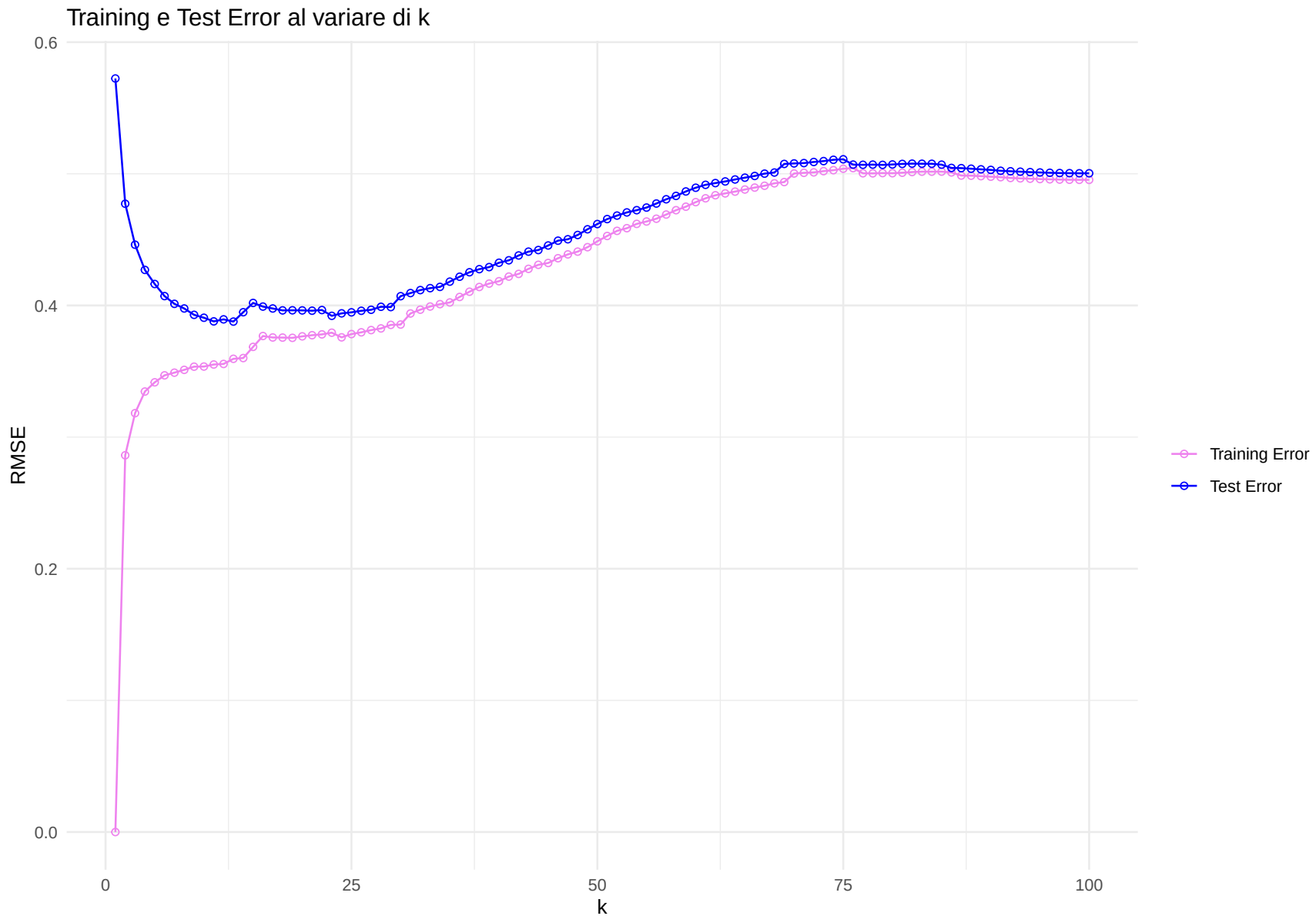
Cosa succede in k-NN per valori di k estremi (troppo grandi/piccoli)?



Leave-one-out CV error al variare di k



Confronto tra training e CV error



Dilemma bias/varianza

- Per $k = 1$ (modello molto complesso) l'errore di training è minimo, ma l'errore di test è elevato (*overfitting*).

Dilemma bias/varianza

- Per $k = 1$ (modello molto complesso) l'errore di training è minimo, ma l'errore di test è elevato (*overfitting*).
- Per $k = 50$ (modello molto semplice) l'errore di training è elevato, e anche l'errore di test è elevato (*underfitting*).

Dilemma bias/varianza

- Per $k = 1$ (modello molto complesso) l'errore di training è minimo, ma l'errore di test è elevato (*overfitting*).
- Per $k = 50$ (modello molto semplice) l'errore di training è elevato, e anche l'errore di test è elevato (*underfitting*).
- Troviamo un valore intermedio di k che minimizza l'errore di test (bias-variance tradeoff).

Decomposizione dell'errore

Per la perdita quadratica possiamo scrivere una decomposizione generale dell'errore di generalizzazione in tre termini:

$$L(h) = \underbrace{\text{Bias}^2}_{\text{errore sistematico}} + \underbrace{\text{Varianza}}_{\text{errore da varianza}} + \underbrace{\text{Rumore}}_{\text{errore intrinseco}}$$

- **Bias**^{2}: incapacità del modello di rappresentare la relazione tra input e output.

Decomposizione dell'errore

Per la perdita quadratica possiamo scrivere una decomposizione generale dell'errore di generalizzazione in tre termini:

$$L(h) = \underbrace{\text{Bias}^2}_{\text{errore sistematico}} + \underbrace{\text{Varianza}}_{\text{errore da varianza}} + \underbrace{\text{Rumore}}_{\text{errore intrinseco}}$$

- **Bias**^{2}: incapacità del modello di rappresentare la relazione tra input e output.
- **Varianza**: sensibilità del modello alle variazioni nei dati di training.

Decomposizione dell'errore

Per la perdita quadratica possiamo scrivere una decomposizione generale dell'errore di generalizzazione in tre termini:

$$L(h) = \underbrace{\text{Bias}^2}_{\text{errore sistematico}} + \underbrace{\text{Varianza}}_{\text{errore da varianza}} + \underbrace{\text{Rumore}}_{\text{errore intrinseco}}$$

- **Bias**^{2}: incapacità del modello di rappresentare la relazione tra input e output.
- **Varianza**: sensibilità del modello alle variazioni nei dati di training.
- **Rumore**: intrinseco nei dati, dovuto a fattori non modellati.

Dimostrazione: preliminari

- **Ipotesi**:

Dimostrazione: preliminari

- **Ipotesi**:
  - il modello di regressione $h(x) = h(x; D)$ dipende dai dati di training $D = \{(x_i, y_i)\}_{i=1}^{n}$.

Dimostrazione: preliminari

- **Ipotesi**:
  - il modello di regressione $h(x) = h(x; D)$ dipende dai dati di training $D = \{(x_i, y_i)\}_{i=1}^{n}$.
  - i dati (train/test) sono variabili aleatorie indipendenti e identicamente distribuite (i.i.d.) con legge $P(X, Y)$.

Dimostrazione: preliminari

- **Ipotesi**:
  - il modello di regressione $h(x) = h(x; D)$ dipende dai dati di training $D = \{(x_i, y_i)\}_{i=1}^{n}$.
  - i dati (train/test) sono variabili aleatorie indipendenti e identicamente distribuite (i.i.d.) con legge $P(X, Y)$.
- **Lemmi**:

Dimostrazione: preliminari

- **Ipotesi**:
  - il modello di regressione $h(x) = h(x; D)$ dipende dai dati di training $D = \{(x_i, y_i)\}_{i=1}^{n}$.
  - i dati (train/test) sono variabili aleatorie indipendenti e identicamente distribuite (i.i.d.) con legge $P(X, Y)$.
- **Lemmi**:
  1. **Teorema di Pitagora** per variabili aleatorie: se U, V sono variabili aleatorie e U è costante conoscendo l'informazione I

$$\mathbb{E}[(V - U)^2|I] = \mathbb{E}[(V - \mathbb{E}[V|I])^2|I] + (\mathbb{E}[V|I] - U)^2$$

Dimostrazione: preliminari

- **Ipotesi**:
  - il modello di regressione $h(x) = h(x; D)$ dipende dai dati di training $D = \{(x_i, y_i)\}_{i=1}^{n}$.
  - i dati (train/test) sono variabili aleatorie indipendenti e identicamente distribuite (i.i.d.) con legge $P(X, Y)$.
- **Lemmi**:
  1. **Teorema di Pitagora** per variabili aleatorie: se U, V sono variabili aleatorie e U è costante conoscendo l'informazione I

$$\mathbb{E}[(V - U)^2|I] = \mathbb{E}[(V - \mathbb{E}[V|I])^2|I] + (\mathbb{E}[V|I] - U)^2$$

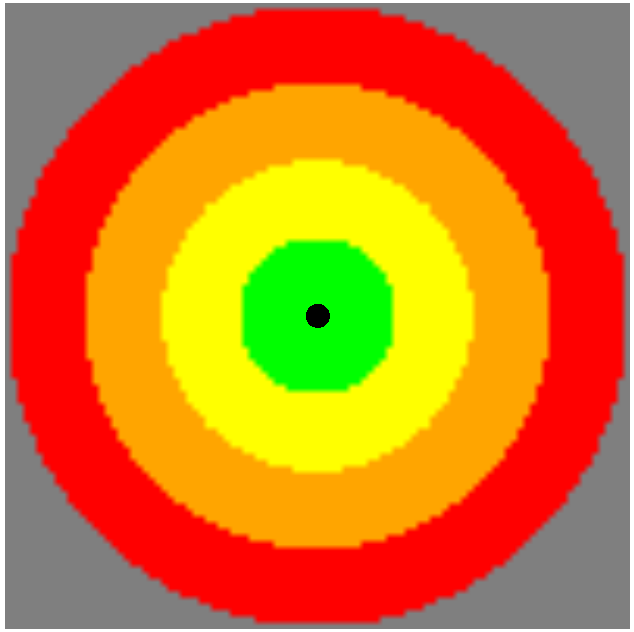
  2. **Legge della probabilità totale** per il valor medio: se V è variabile aleatoria e I informazione,

$$\mathbb{E}[V] = \mathbb{E}[\mathbb{E}[V|I]]$$

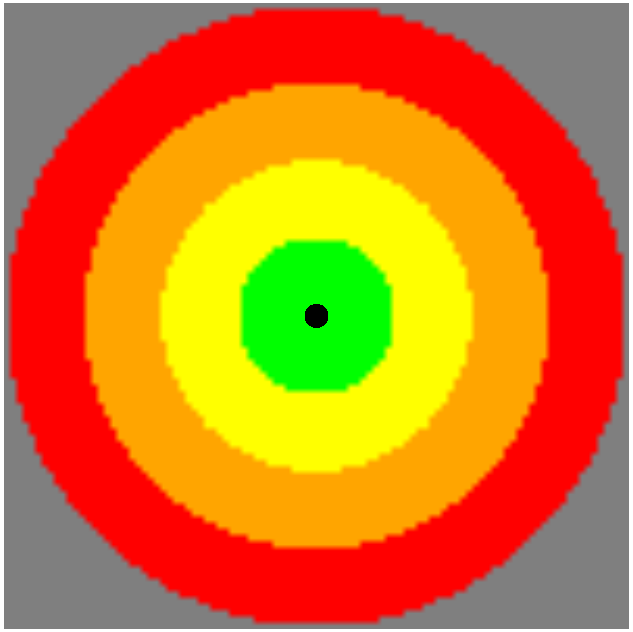
Passo 1: estrazione del Rumore

Passo 2: decomposizione Bias+Varianza

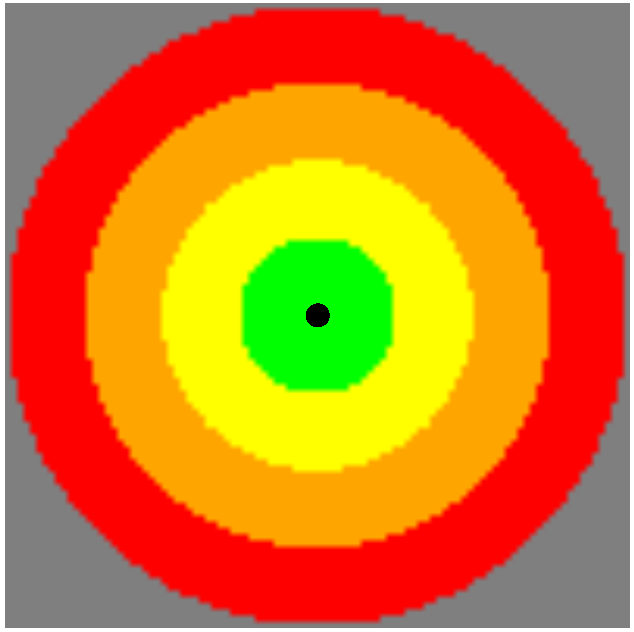
Bias: Alto Varianza: Alto



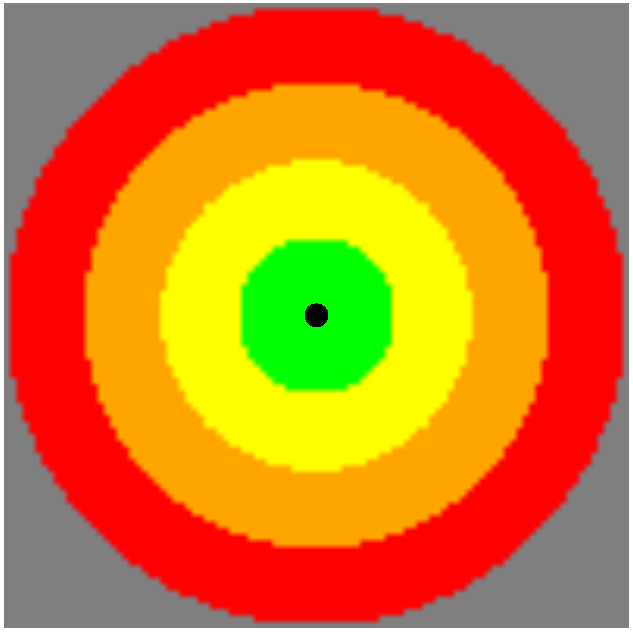
Bias: Alto Varianza: Basso



Bias: Basso Varianza: Alto



Bias: Basso Varianza: Basso



Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN
  - *bagging*, *dropout*, ecc.

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN
  - *bagging*, *dropout*, ecc.
- Se l'errore di training è **alto**, e l'errore di test è **alto** $\rightarrow$ underfitting (modello troppo semplice).

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN
  - *bagging*, *dropout*, ecc.
- Se l'errore di training è **alto**, e l'errore di test è **alto** $\rightarrow$ underfitting (modello troppo semplice).
  - aumentare il numero di parametri

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN
  - *bagging*, *dropout*, ecc.
- Se l'errore di training è **alto**, e l'errore di test è **alto** $\rightarrow$ underfitting (modello troppo semplice).
  - aumentare il numero di parametri
  - ridurre il valore di regolarizzazione – il valore di k in KNN.

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN
  - *bagging*, *dropout*, ecc.
- Se l'errore di training è **alto**, e l'errore di test è **alto** $\rightarrow$ underfitting (modello troppo semplice).
  - aumentare il numero di parametri
  - ridurre il valore di regolarizzazione – il valore di k in KNN.
  - *boosting*

Indicazioni pratiche dalle curve di errore

- Se l'errore di training è *basso*, e l'errore di test è *basso* $\rightarrow$ buon modello (modello ben bilanciato).
- Se l'errore di training è **molto basso**, ma l'errore di test è *alto* $\rightarrow$ rischio di overfitting (modello troppo complesso).
  - ridurre il numero di parametri
  - aumentare iperparametri di regolarizzazione – il valore di k in KNN
  - *bagging*, *dropout*, ecc.
- Se l'errore di training è **alto**, e l'errore di test è **alto** $\rightarrow$ underfitting (modello troppo semplice).
  - aumentare il numero di parametri
  - ridurre il valore di regolarizzazione – il valore di k in KNN.
  - *boosting*
- **Attenzione:** molto in pratica dipende dal modello utilizzato, ci sono casi che non presentano il dilemma (reti neurali profonde, random forests, ecc.).

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni
  - curse of dimensionality (dati sparsi in spazi ad alta dimensione)

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni
  - curse of dimensionality (dati sparsi in spazi ad alta dimensione)
  - difficile intepretazione del modello.

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni
  - curse of dimensionality (dati sparsi in spazi ad alta dimensione)
  - difficile intepretazione del modello.
- **Modelli parametrici:** assumiamo che il modello di regressione h appartenga a una famiglia di funzioni parametrizzate da un vettore di parametri $\theta \in \mathbb{R}^p$:

$$h(x; \theta), \quad \theta \in \mathbb{R}^p$$

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni
  - curse of dimensionality (dati sparsi in spazi ad alta dimensione)
  - difficile intepretazione del modello.
- **Modelli parametrici:** assumiamo che il modello di regressione h appartenga a una famiglia di funzioni parametrizzate da un vettore di parametri $\theta \in \mathbb{R}^p$:

$$h(x; \theta), \quad \theta \in \mathbb{R}^p$$

- Due fasi:

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni
  - curse of dimensionality (dati sparsi in spazi ad alta dimensione)
  - difficile intepretazione del modello.
- **Modelli parametrici:** assumiamo che il modello di regressione h appartenga a una famiglia di funzioni parametrizzate da un vettore di parametri $\theta \in \mathbb{R}^p$:

$$h(x; \theta), \quad \theta \in \mathbb{R}^p$$

- Due fasi:
  1. stima dei parametri (training del modello)

Modelli parametrici

- k-NN per regressione è *non parametrico*. **Problemi:**
  - quando la taglia n è grande $\rightarrow$ costi computazionali elevati per fare previsioni
  - curse of dimensionality (dati sparsi in spazi ad alta dimensione)
  - difficile intepretazione del modello.
- **Modelli parametrici:** assumiamo che il modello di regressione h appartenga a una famiglia di funzioni parametrizzate da un vettore di parametri $\theta \in \mathbb{R}^p$:

$$h(x; \theta), \quad \theta \in \mathbb{R}^p$$

- Due fasi:
  1. stima dei parametri (training del modello)
  2. previsione sui nuovi dati (test del modello)

Metodo dei minimi quadrati (OLS)

Dato un training set, si sceglie il modello di regressione che minimizza l'errore quadratico medio sui dati di training:

$$\theta_{OLS} = \arg\min_{\theta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (h(x_i; \theta) - y_i)^2$$

Per la previsione su un nuovo dato x, si usa il modello con i parametri stimati:

$$x \mapsto h(x; \theta_{OLS})$$

- **Osservazione:** il metodo dei minimi quadrati può essere usato con qualsiasi modello parametrico (lineare, polinomiale, reti neurali, ecc.).

Interpretazione probabilistica del metodo OLS

Ipotesi: relazione tra input e output con un termine di rumore additivo (**residuo**) gaussiano:

$$Y = h(X;\theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Supponendo ϵ, X indipendenti (noti i parametri), dato un training set $D = \{(x_i, y_i)\}_{i=1}^n$ i.i.d. la *verosimiglianza* dei parametri θ è

$$\begin{aligned}
\mathcal{L}(\theta; D) &= \prod_{i=1}^{n} p(Y = y_i, X = x_i|\theta) \\
&= \prod_{i=1}^{n} p(h(x_i;\theta) + \epsilon = y_i | X = x_i, \theta) p(X = x_i|\theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - h(x_i;\theta))^2}{2\sigma^2}\right) p(X = x_i)
\end{aligned}$$

Interpretazione probabilistica del metodo OLS

Ipotesi: relazione tra input e output con un termine di rumore additivo (**residuo**) gaussiano:

$$Y = h(X;\theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Supponendo ϵ, X indipendenti (noti i parametri), dato un training set $D = \{(x_i, y_i)\}_{i=1}^n$ i.i.d. la *verosimiglianza* dei parametri θ è

$$\begin{aligned}
\mathcal{L}(\theta; D) &= \prod_{i=1}^{n} p(Y = y_i, X = x_i|\theta) \\
&= \prod_{i=1}^{n} p(h(x_i;\theta) + \epsilon = y_i|X = x_i, \theta) p(X = x_i|\theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - h(x_i;\theta))^2}{2\sigma^2}\right) p(X = x_i)
\end{aligned}$$

- La stima di *massima verosimiglianza* (MLE) dei parametri è

Problemi del metodo

1. la minimizzazione può essere difficile (non convessa, molti minimi locali, ecc.).

Problemi del metodo

1. la minimizzazione può essere difficile (non convessa, molti minimi locali, ecc.).
2. rischio di overfitting se modello troppo complesso, come regolarizzare?

Problemi del metodo

1. la minimizzazione può essere difficile (non convessa, molti minimi locali, ecc.).
2. rischio di overfitting se modello troppo complesso, come regolarizzare?
3. $h(x; \theta_{OLS})$ è una previsione puntuale, come valutare l'incertezza?

Problemi del metodo

1. la minimizzazione può essere difficile (non convessa, molti minimi locali, ecc.).
2. rischio di overfitting se modello troppo complesso, come regolarizzare?
3. $h(x; \theta_{OLS})$ è una previsione puntuale, come valutare l'incertezza?

- **Una soluzione**: modello di regressione *lineare*.

Modello di regressione lineare

Definizione: un modello di regressione parametrico $h(x;\theta)$ è detto **lineare** se è della forma

$$h(x;\theta) = \theta_0 + \sum_{j=1}^{p} \theta_j \phi_j(x)$$

dove $\phi_j : F \to \mathbb{R}$ sono funzioni *note* (funzioni base, features, trasformazioni delle variabili di input).

- **Attenzione:** il modello è **lineare nei parametri** θ_j, ma le funzioni ϕ_j possono essere **non lineari**.

Esempi di modelli di regressione lineare

- modello di regressione polinomiale di grado p in una variabile:

$$h(x;\theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p$$

$$h(x;\theta) = \theta_0 + \sum_{j=1}^{d} \theta_j x_j$$

Esempi di modelli di regressione lineare

- modello di regressione polinomiale di grado p in una variabile:

$$h(x;\theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p$$

- Caso $p = 1 \rightarrow$ modello di **regressione lineare semplice**.

$$h(x;\theta) = \theta_0 + \sum_{j=1}^{d} \theta_j x_j$$

Esempi di modelli di regressione lineare

- modello di regressione polinomiale di grado p in una variabile:

$$h(x;\theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p$$

- Caso $p = 1 \rightarrow$ modello di **regressione lineare semplice**.
- modello di regressione lineare multipla in d variabili:

$$h(x;\theta) = \theta_0 + \sum_{j=1}^{d} \theta_j x_j$$

Esempi di modelli di regressione lineare

- modello di regressione polinomiale di grado p in una variabile:

$$h(x;\theta) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_p x^p$$

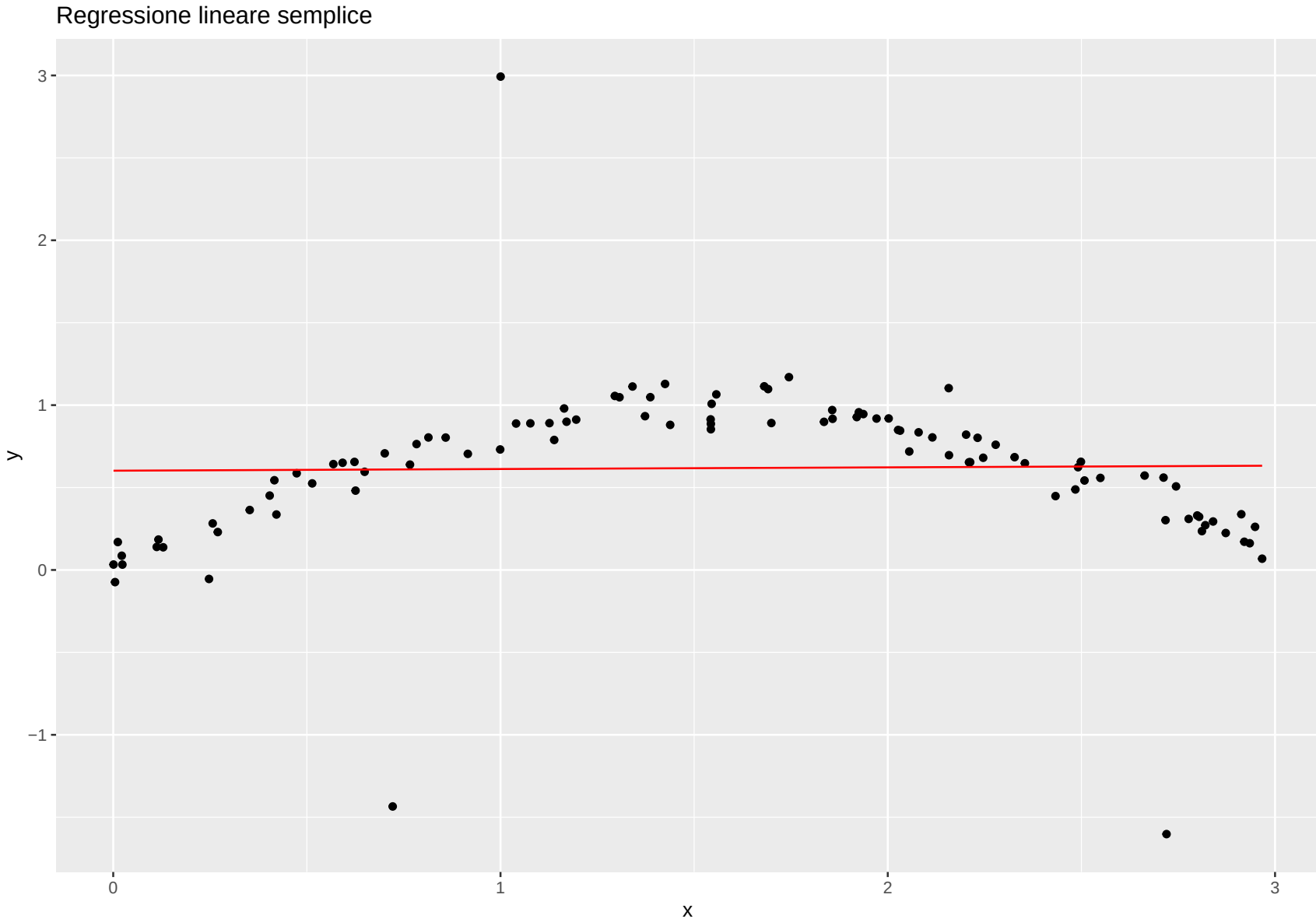
- Caso $p = 1 \rightarrow$ modello di **regressione lineare semplice**.
- modello di regressione lineare multipla in d variabili:

$$h(x;\theta) = \theta_0 + \sum_{j=1}^{d} \theta_j x_j$$

- modello di regressione con interazioni tra variabili:

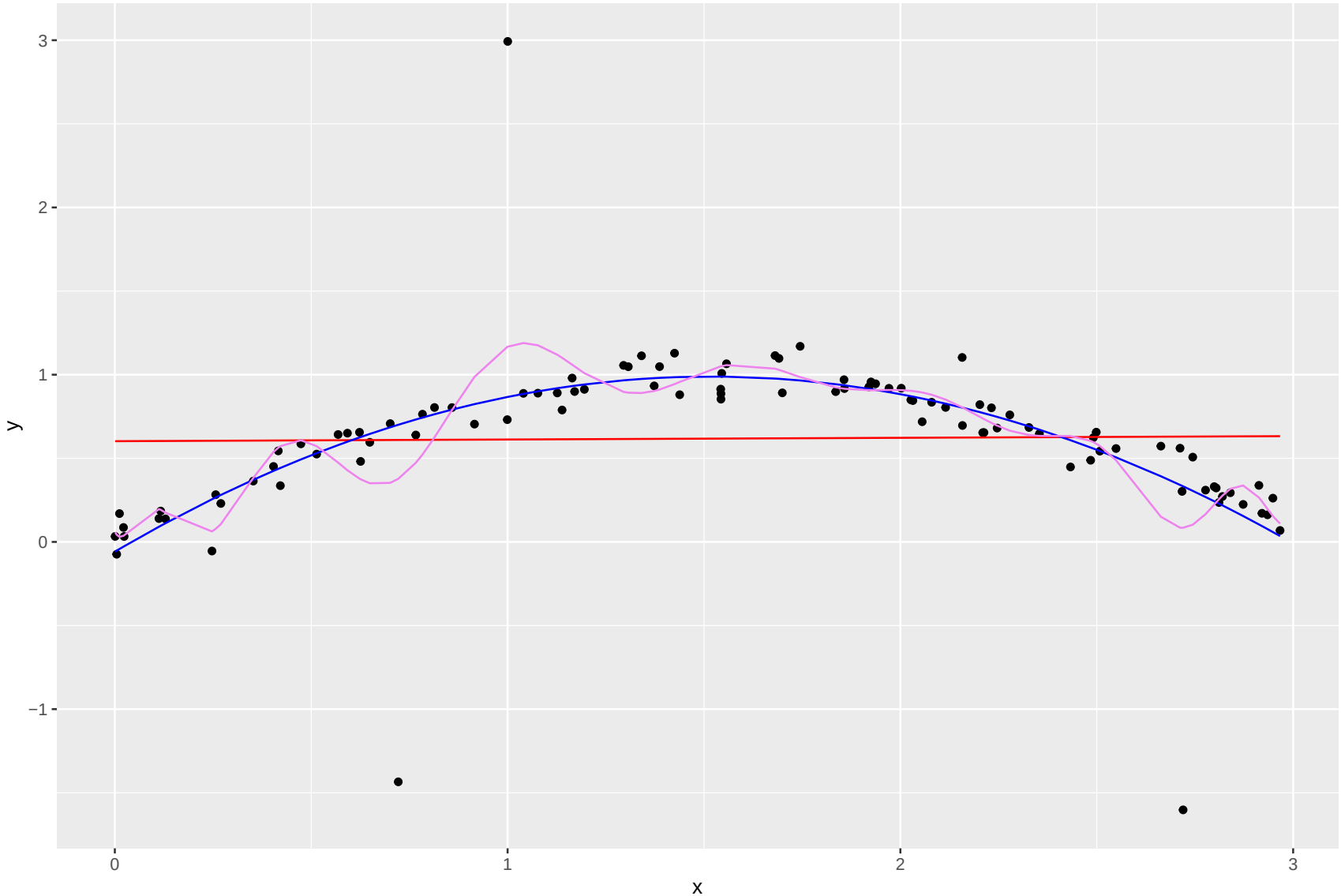
$$h(x;\theta) = \theta_0 + \sum_{j=1}^{d} \theta_j x_j + \sum_{j=1}^{d} \sum_{k=j+1}^{d} \theta_{jk} x_j x_k$$

Esempio sul dataset generato: regressione lineare semplice

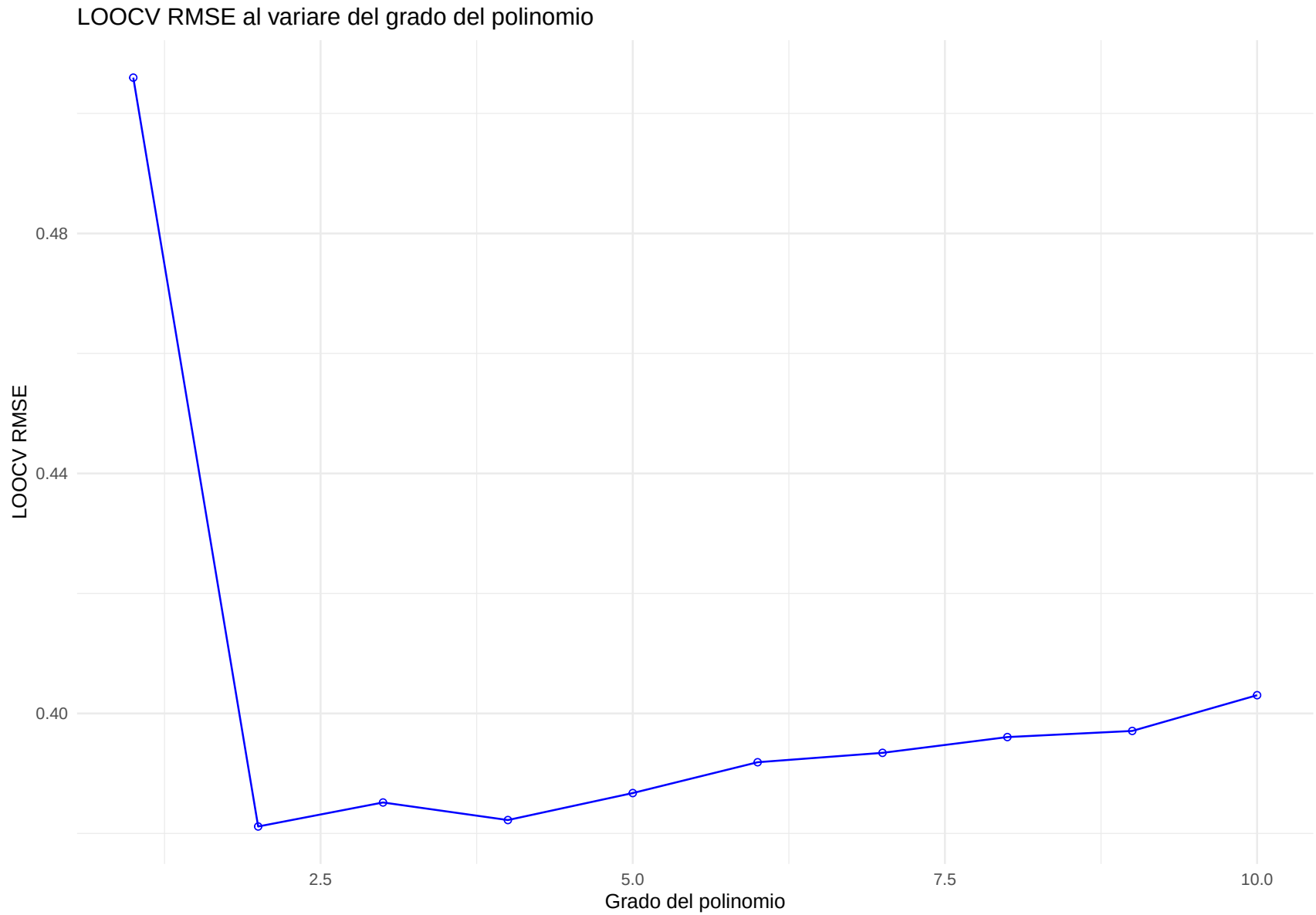


Esempio sul dataset generato: regressione polinomiale

Regressione polinomiale di gradi 1 (rosso), 2 (blu) e 30 (viola)



Errore di validazione crociata al variare del grado massimo p



Interpretazione dei parametri nella regressione lineare semplice

- Nel modello di regressione lineare semplice

$$h(x;\theta) = \theta_0 + \theta_1 x$$

Interpretazione dei parametri nella regressione lineare semplice

- Nel modello di regressione lineare semplice

$$h(x;\theta) = \theta_0 + \theta_1 x$$

- il parametro θ_0 è l'*intercetta* (valore di $h(x;\theta)$ per $x = 0$).

Interpretazione dei parametri nella regressione lineare semplice

- Nel modello di regressione lineare semplice

$$h(x;\theta) = \theta_0 + \theta_1 x$$

- il parametro θ_0 è l'*intercetta* (valore di $h(x;\theta)$ per $x = 0$).
- il parametro θ_1 è la *pendenza* (variazione di $h(x;\theta)$ al variare di x).

Interpretazione dei parametri nella regressione lineare semplice

- Nel modello di regressione lineare semplice

$$h(x;\theta) = \theta_0 + \theta_1 x$$

- il parametro θ_0 è l'*intercetta* (valore di $h(x;\theta)$ per $x = 0$).
- il parametro θ_1 è la *pendenza* (variazione di $h(x;\theta)$ al variare di x).
- Entrambe si ricavano dai dati di training tramite il metodo dei minimi quadrati:

$$\theta_{OLS,1} := \mathrm{cor}(x,y)\frac{\sigma_y}{\sigma_x}, \quad \theta_{OLS,0} := \bar{y} - \theta_1 \bar{x}$$

Interpretazione dei parametri nella regressione lineare semplice

- Nel modello di regressione lineare semplice

$$h(x;\theta) = \theta_0 + \theta_1 x$$

- il parametro θ_0 è l'*intercetta* (valore di $h(x;\theta)$ per $x = 0$).
- il parametro θ_1 è la *pendenza* (variazione di $h(x;\theta)$ al variare di x).
- Entrambe si ricavano dai dati di training tramite il metodo dei minimi quadrati:

$$\theta_{OLS,1} := \text{cor}(x,y)\frac{\sigma_y}{\sigma_x}, \quad \theta_{OLS,0} := \bar{y} - \theta_1 \bar{x}$$

- dove $\text{cor}(x,y)$ è la correlazione campionaria tra x e y, σ_x, σ_y sono le deviazioni standard campionarie e $\bar{x}$, $\bar{y}$ sono le medie campionarie.

Dimostrazione

Problema:

$$\min_{\theta_0,\theta_1} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2 = \min_{\theta_0} \min_{\theta_1} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

- Dato $\theta_1 \rightarrow \theta_0$ è la media degli $y_i - \theta - ix_i$

$$\rightarrow \quad \theta_0 = \bar{y} - \theta_1 \bar{x}$$

Dimostrazione

Problema:

$$\min_{\theta_0,\theta_1} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2 = \min_{\theta_0} \min_{\theta_1} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2$$

- Dato $\theta_1 \rightarrow \theta_0$ è la media degli $y_i - \theta - ix_i$

$$\rightarrow \quad \theta_0 = \bar{y} - \theta_1 \bar{x}$$

- Per trovare θ_1 imponiamo che la derivata (in θ_1 si annulli:

$$0 = \frac{d}{d\theta_1} \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i)^2 = -2 \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 x_i) x_i$$