Statistical learning and deep learning: theoretical background and hands-on sessions

Lezione 2 - LAB

S. Biffani

IBBA/CNR

27 gennaio, 2023

Compromesso tra Bias e Varianza - Simulazione 1

Obiettivo: Usare dati simulati per *capire* meglio:

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

Perchè usiamo dati simulati ?

Compromesso tra Bias e Varianza - Simulazione 2

Cosa simuliamo ?

$$Y = f(X) + \epsilon$$

dove,

$Y =$ variabile continua

$f =$ è una funzione (nota nella nostra simulazione ma non nota nella realtà) che lega le $X_1, ..., X_p$

$\epsilon =$ errore *irriproducibile*

Compromesso tra Bias e Varianza - Simulazione 3

- Ipotizziamo una X da 0 a 10 con incrementi di 0.5

```
##  [1]  0.0  0.5  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0  6.5
## [16]  7.5  8.0  8.5  9.0  9.5 10.0
```

  - un andamento *quadratico* $X \rightarrow X^2$
  - un $\epsilon \sim \mathcal{N}(0, 25)$

Compromesso tra Bias e Varianza - Simulazione 4

Come simuliamo questa funzione in R?

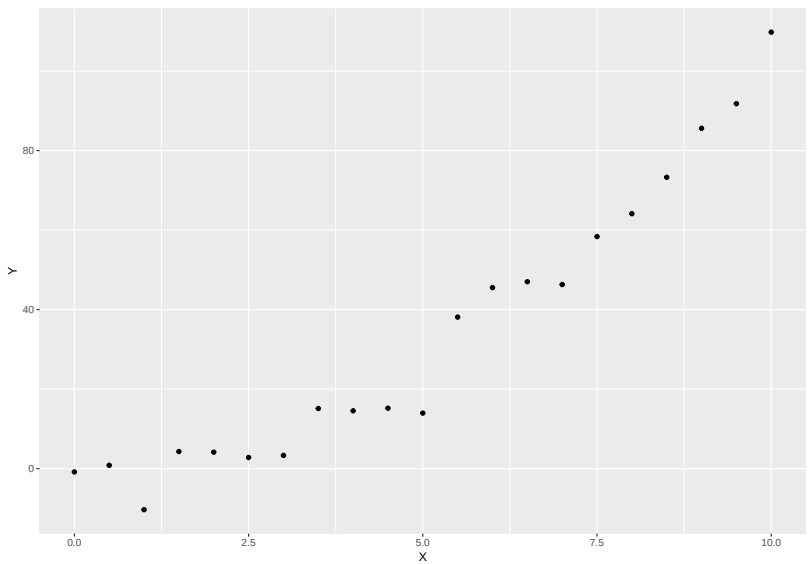
- dobbiamo simulare un andamento tendezialmente *quadratico* con del *rumore*

```
# ds "rumore"
std<- 25^.5
# sequenza di X
seq(0,10,.5)-> X
# andamento quadratico con rumore
Y<- X**2 + rnorm(length(X), 0, std)
```

- sd:

```
## [1] 5
```

- X: [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 [16] 7.5 8.0 8.5 9.0 9.5 10.0
- Y: [1] -0.839180 0.835186 -10.331931 4.299906 4.133900 2.798931 [7] 3.326632 15.108450 14.537152 15.182408 13.956867 38.105002 [13] 45.517517 46.995594 46.291890 58.363982 64.115717 73.273962 [19] 85.586873 91.773225 109.770733



Compromesso tra Bias e Varianza - Simulazione 5

- e se volessi assegnare la costruzione di Y ad una funzione di R?
- ‘nomeFunzione <- function(lista_argomenti){
  comando1
  return(valore) }‘
- ???

Compromesso tra Bias e Varianza - Simulazione 5

- e se volessi assegnare la costruzione di Y ad una funzione di R?
- ‘nomeFunzione <- function(lista_argomenti){
  comando1
  return(valore) }‘

```
set.seed(123) # seme fisso per riproducibilità
get_y <- function(X, std=5){

  return(X**2 + rnorm(length(X), 0, std))

}
```

[1] -2.8023782 -0.9008874 8.7935416 2.6025420 4.6464387 14.8253249 [7] 11.3045810 5.9246938 12.5657357 18.0216901 31.1204090 32.0490691 [13] 38.0038573 42.8034136 46.2207943 65.1845657 66.4892524 62.4169142 [19] 84.5067795 87.8860430 94.6608815

- commentate il comando set.seed() e ripetete il calcolo, cosa succede?

Compromesso tra Bias e Varianza - Simulazione 6

A questo punto vogliamo testare 3 diversi modelli, che si differenziano per la loro complessità:

1. regressione lineare

```
mod1 <- lm(y ~ x,              data = dat)
```

2. regressione quadratica

```
mod2 <- lm(y ~ poly(x, 2),  data = dat)
```

3. polinimio di 10° grado

```
mod3 <- lm(y ~ poly(x, 10), data = dat)
```

Compromesso tra Bias e Varianza - Simulazione 7

```
X<-seq(0,10,.5)
dat<-data.frame(x=X,
                y=get_y(X))
mod1 <- lm(y ~ x,          data = dat)
mod2 <- lm(y ~ poly(x, 2), data = dat)
mod3 <- lm(y ~ poly(x, 10), data = dat)
```

```
ggplot(dat,aes(x,y))+
  geom_point()+
  geom_line(aes(x,predict(mod1)))->g1

ggplot(dat,aes(x,y))+
  geom_point()+
  geom_line(aes(x,predict(mod2)))->g2

ggplot(dat,aes(x,y))+
  geom_point()+
  geom_line(aes(x,predict(mod3)))->g3
```

Compromesso tra Bias e Varianza - Simulazione 8

Quale è il migliore?

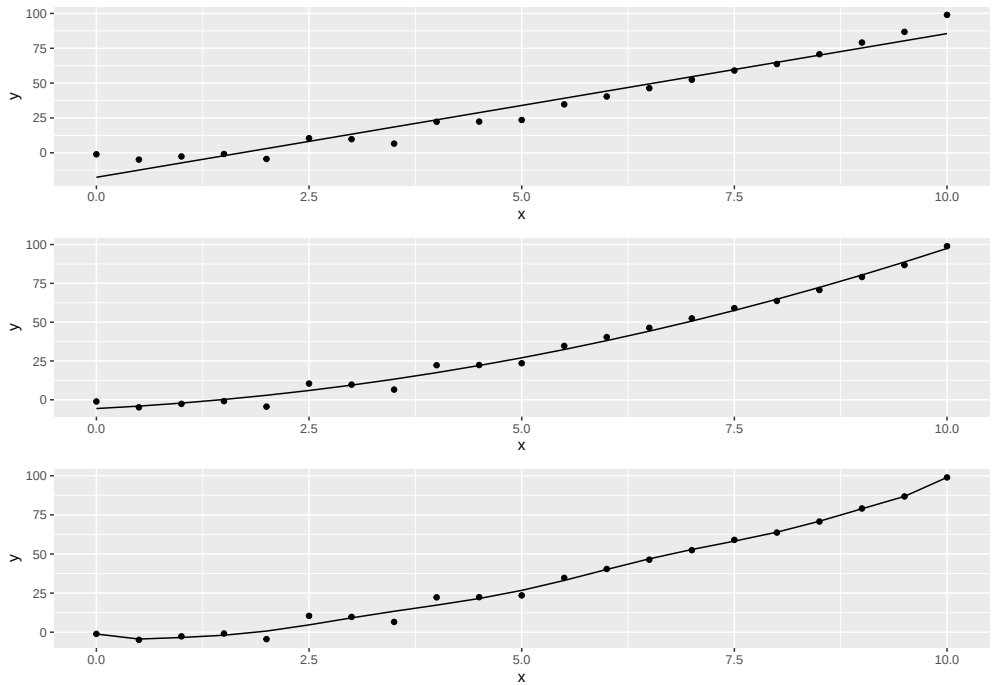


Figura 1: Risultati

Compromesso tra Bias e Varianza - Simulazione 9

Per rispondere a questa domanda dobbiamo ricordarci che il nostro dataset è un **campione** casuale di tutti i possibili dati che corrispondono alla nostra funzione iniziale

- chi ha più bias?
- chi ha più varianza?

Possiamo rispondere a queste domande se:

1. generiamo un numero elevato di *train data* (e.g. 1000)
2. applichiamo i 3 modelli
3. usiamo i 3 modelli per predire il valore di y per un valore noto (e.g. $x = 4$)

Compromesso tra Bias e Varianza - Simulazione 10

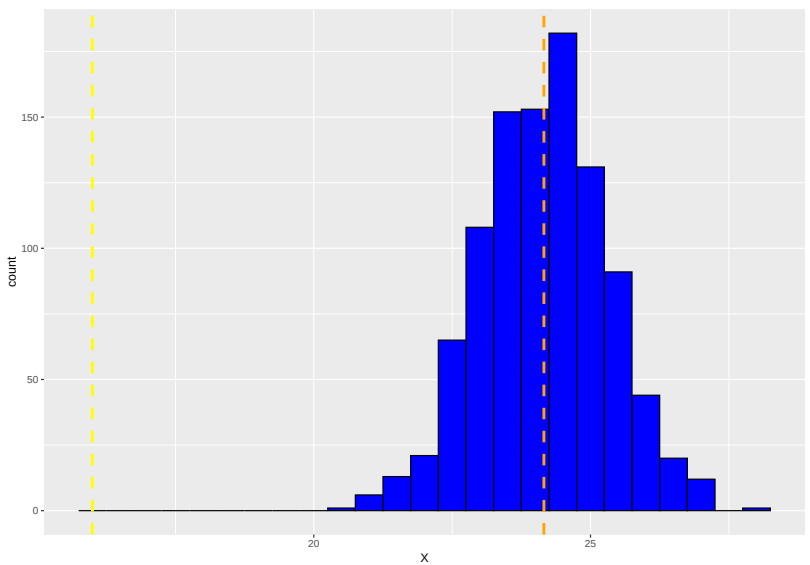
- sfruttiamo quello che abbiamo visto: **funzioni di R** e **iterazioni**

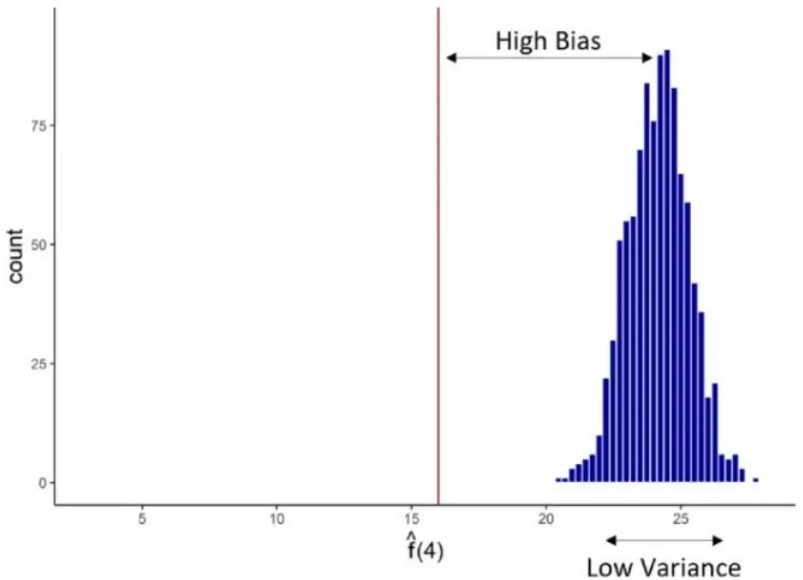
```
# dati
X <- seq(0, 10, 0.5)
# funzione che genera i valori casuali y
get_y <- function(X, std = 5){

  return(X**2 + rnorm(length(X), 0, std))

}
# seed for reproducibility
set.seed(12345)
f_hat_1 <- numeric(0)
f_hat_2 <- numeric(0)
f_hat_3 <- numeric(0)
# genero 1000 datasets, applico modelli, predico y quando X è 4 f^(4)
for (i in 1:1000){
  dat <- data.frame(x = X, y = get_y(X))

  mod1 <- lm(y ~ x,           data = dat)
  mod2 <- lm(y ~ poly(x, 2),  data = dat)
  mod3 <- lm(y ~ poly(x, 10), data = dat)

  f_hat_1 <- c(f_hat_1, predict(mod1, data.frame(x = 4))[[1]])
  f_hat_2 <- c(f_hat_2, predict(mod2, data.frame(x = 4))[[1]])
  f_hat_3 <- c(f_hat_3, predict(mod3, data.frame(x = 4))[[1]])

}
```

Compromesso tra Bias e Varianza - Simulazione 11

I 3 oggetti **f_hat_1**, **f_hat_2** e **f_hat_3**, contengono le stime dei nostri modelli quando $x_0 = 4$ (che dovrebbe essere 16)

- che succede se guardo come sono distribuite nel modello 1?

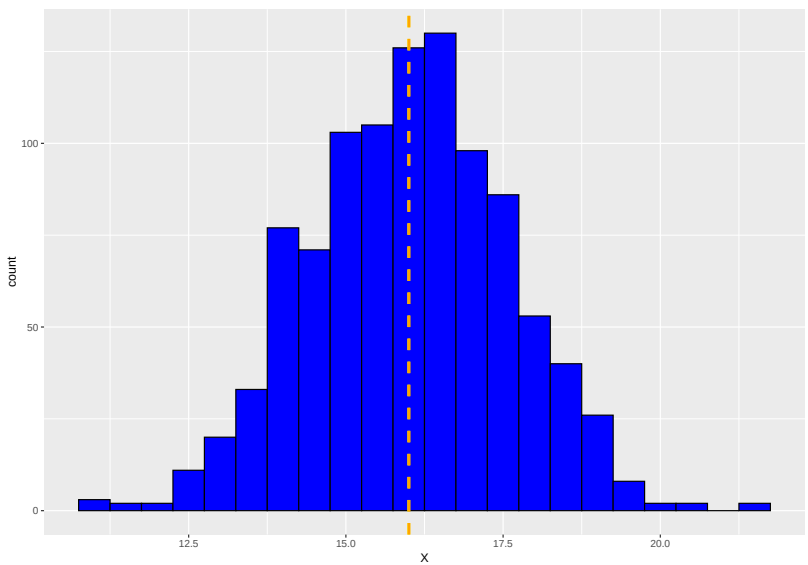


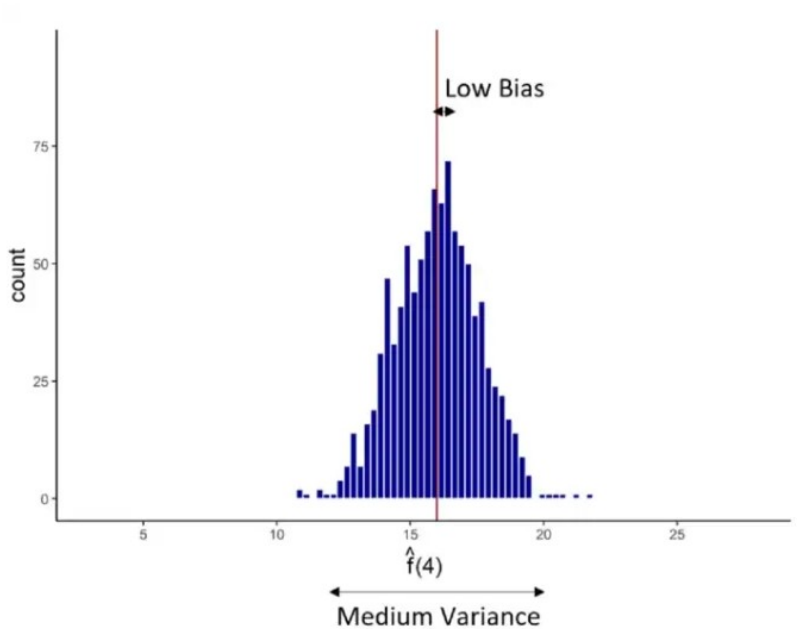


Compromesso tra Bias e Varianza - Simulazione 12

I 3 oggetti **f_hat_1**, **f_hat_2** e **f_hat_3**, contengono le stime dei nostri modelli quando $x_0 = 4$ (che dovrebbe essere 16)

- che succede se guardo come sono distribuite nel modello 2?

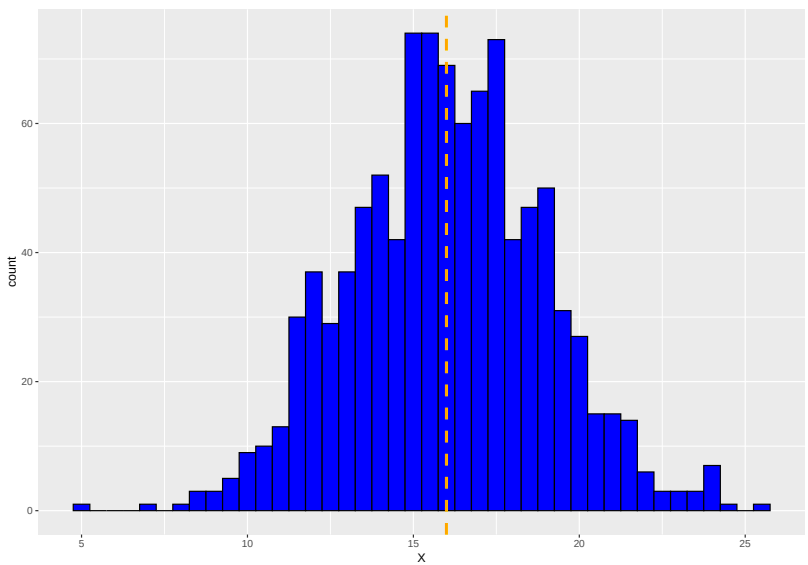


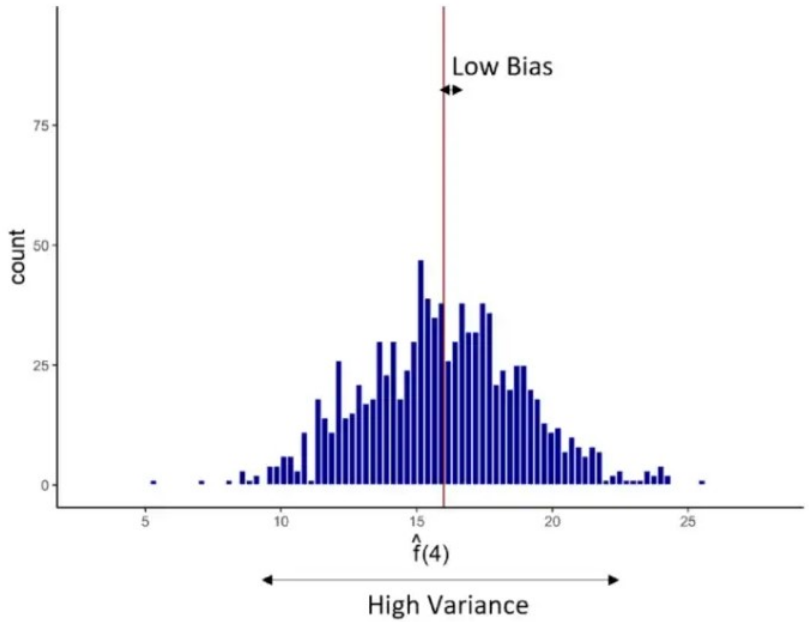


Compromesso tra Bias e Varianza - Simulazione 13

I 3 oggetti **f_hat_1**, **f_hat_2** e **f_hat_3**, contengono le stime dei nostri modelli quando $x_0 = 4$ (che dovrebbe essere 16)

- che succede se guardo come sono distribuite nel modello 3?





Compromesso tra Bias e Varianza - Simulazione 14

Ora posso calcolare il **bias** $((\mathbb{E}_\tau[\hat{f}(x_0)] - f(x_0))^2)$ per i 3 modelli (4 corrisponde a 4)

```
# Bias^2
b1 <- (mean(f_hat_1) - 4^2)^2
b1
```

```
## [1] 66.5535
```

```
b2 <- (mean(f_hat_2) - 4^2)^2
b2
```

```
## [1] 0.000006693562
```

```
b3 <- (mean(f_hat_3) - 4^2)^2
b3
```

```
## [1] 0.00003767206
```

Compromesso tra Bias e Varianza - Simulazione 15

. . . e anche la **varianza** $(\mathbb{E}_\tau[(\hat{f}(x_0) - \mathbb{E}_\tau[\hat{f}(x_0)])^2])$

```
# Variance
v1 <- var(f_hat_1)
v1
```

```
## [1] 1.291835
```

```
v2 <- var(f_hat_2)
v2
```

```
## [1] 2.496312
```

```
v3 <- var(f_hat_3)
v3
```

```
## [1] 8.604308
```

Compromesso tra Bias e Varianza - Simulazione 16

ricordandoci che il *test MSE* è funzione di 3 componenti:

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$$

e ricordando che $Var(\epsilon) = 25$ possiamo calcolare il *test MSE* per i 3 modelli:

```
# EPE right side
epe_1 <- b1 + v1 + 25
epe_2 <- b2 + v2 + 25
epe_3 <- b3 + v3 + 25
```

- modello 1: 92.8453367
- modello 2: 27.4963189
- modello 3: 33.6043453

Compromesso tra Bias e Varianza - Simulazione 17

- ...ma il *test MSE* è anche pari a: $E(y_0 - \hat{f}(x_0))^2$, lo posso calcolare sempre usando una simulazione
  1. generando n nuovi *train data*
  2. in ogni *train data* generando delle p possibili y_0 e le relative predizioni $(\hat{f}(x_0))$
  3. calcolando la media del Errore Atteso (come differenza tra y_0 e $\hat{f}(x_0)$)

Compromesso tra Bias e Varianza - Simulazione 18

(9 corrisponde all'elemento 9 del mio vettore di dati)

```
set.seed(12345)
EPE_1 <- 0
EPE_2 <- 0
EPE_3 <- 0
# per ciascun train data .
for(i in 1:1000){

  # ...genero 1000 nuovi valori y_0
  for(j in 1:1000){

    y_0 <- get_y(X)[9]

    # calcolo la differenza al quadrato tra y_0 e la predizione
    EPE_1 <- EPE_1 + (y_0 - f_hat_1[i])^2
    EPE_2 <- EPE_2 + (y_0 - f_hat_2[i])^2
    EPE_3 <- EPE_3 + (y_0 - f_hat_3[i])^2

  }

}
# calcolo la media di EPE
EPE_1 <- EPE_1 / 1000000
EPE_2 <- EPE_2 / 1000000
EPE_3 <- EPE_3 / 1000000
```

Compromesso tra Bias e Varianza - Simulazione 19

- modello 1: 92.7696591
- modello 2: 27.4821872
- modello 3: 33.6033783

. . . corrispondono a quanto calcolato prima: **il test MSE** è davvero composto dalle 3 componenti!