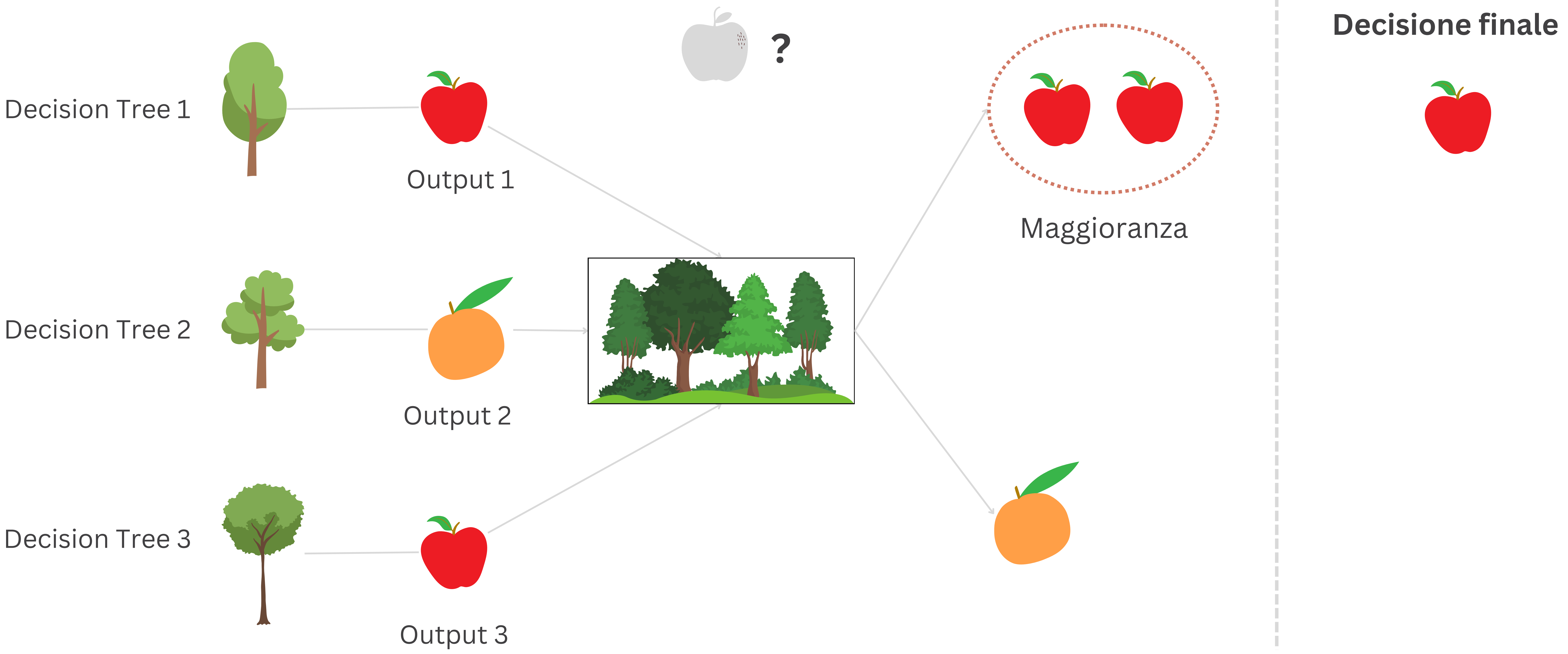
Cos'è Random Forest?

Random Forest è un algoritmo di Machine Learning che combina l'output di più alberi decisionali (Decision Trees) per raggiungere un unico risultato.

Trova applicazione in:

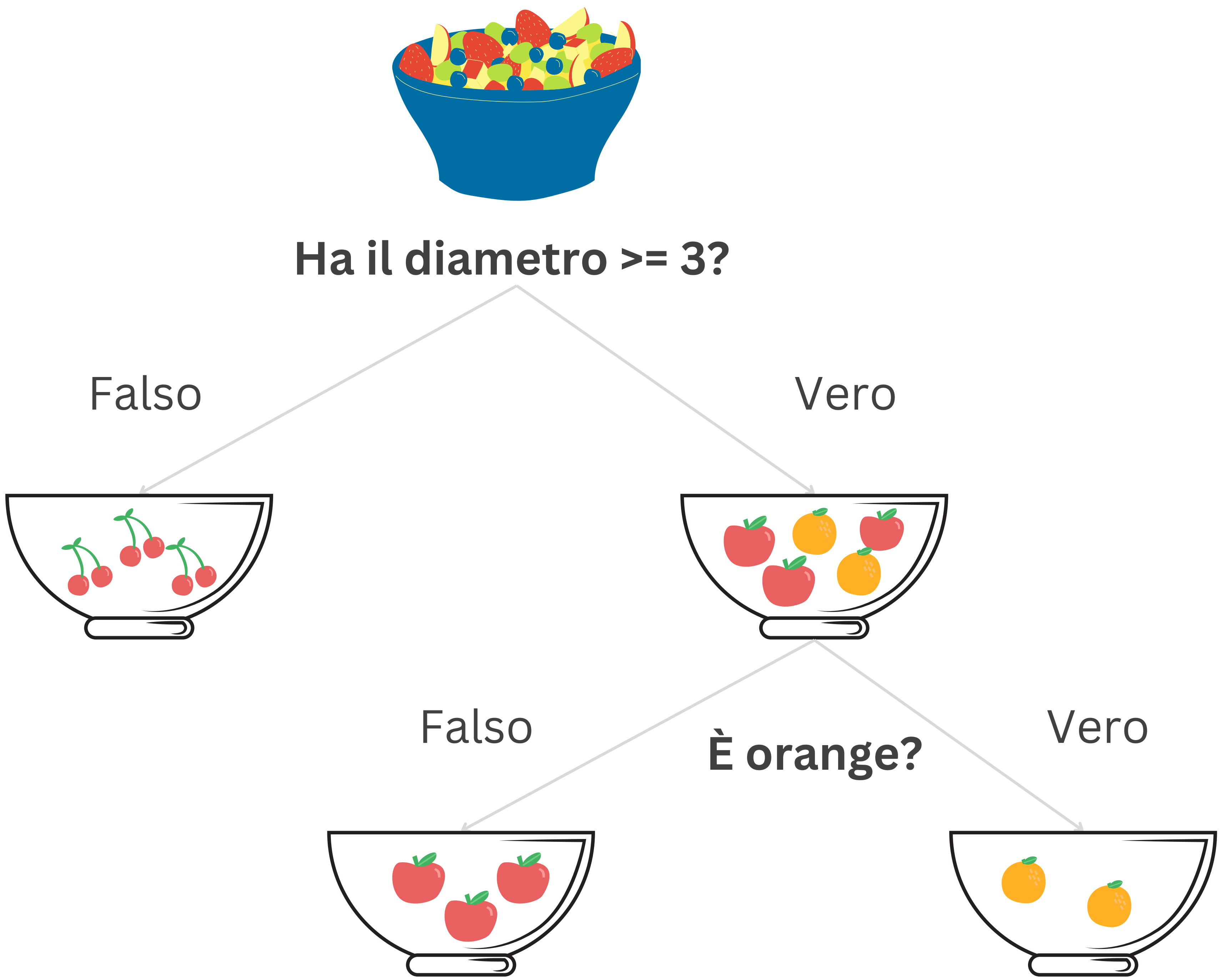
- compiti di classificazione che di regressione,
- metodi per la riduzione della dimensionalità,
- gestire dati mancanti,
- valori degli outlier ed altri passaggi essenziali di esplorazione dei dati.

Cos'è Random Forest?

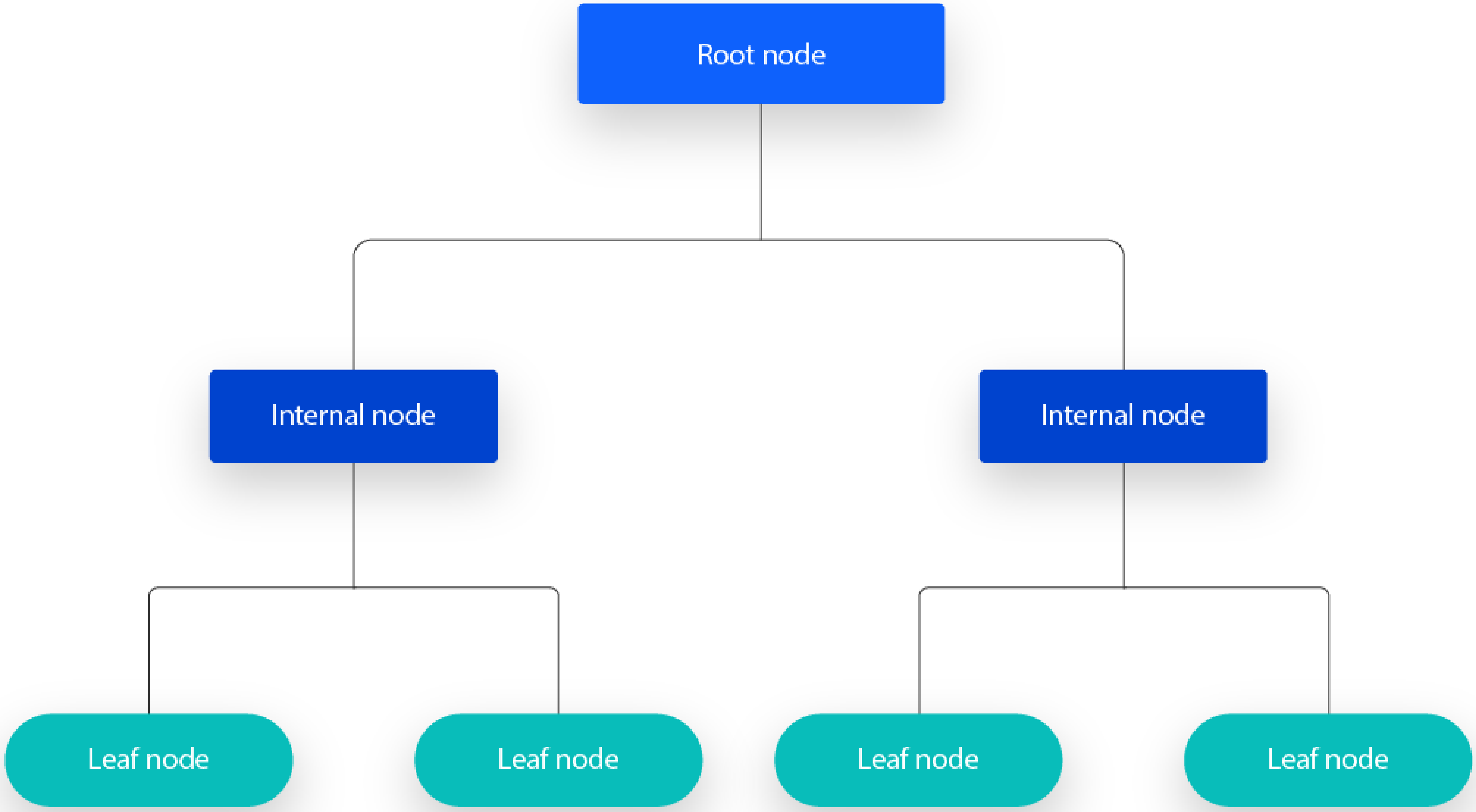


Cos'è un Decision Tree?

Un Decision Tree è un algoritmo non parametrico, con una struttura gerarchica a forma di albero, che consiste in un nodo radice *(root node)*, rami *(branches)*, nodi interni *(internal nodes)* e nodi foglia *(leaf nodes)*.



Cos’è un Decision Tree?



Decision Tree - Concetti

Entropy

Information gain

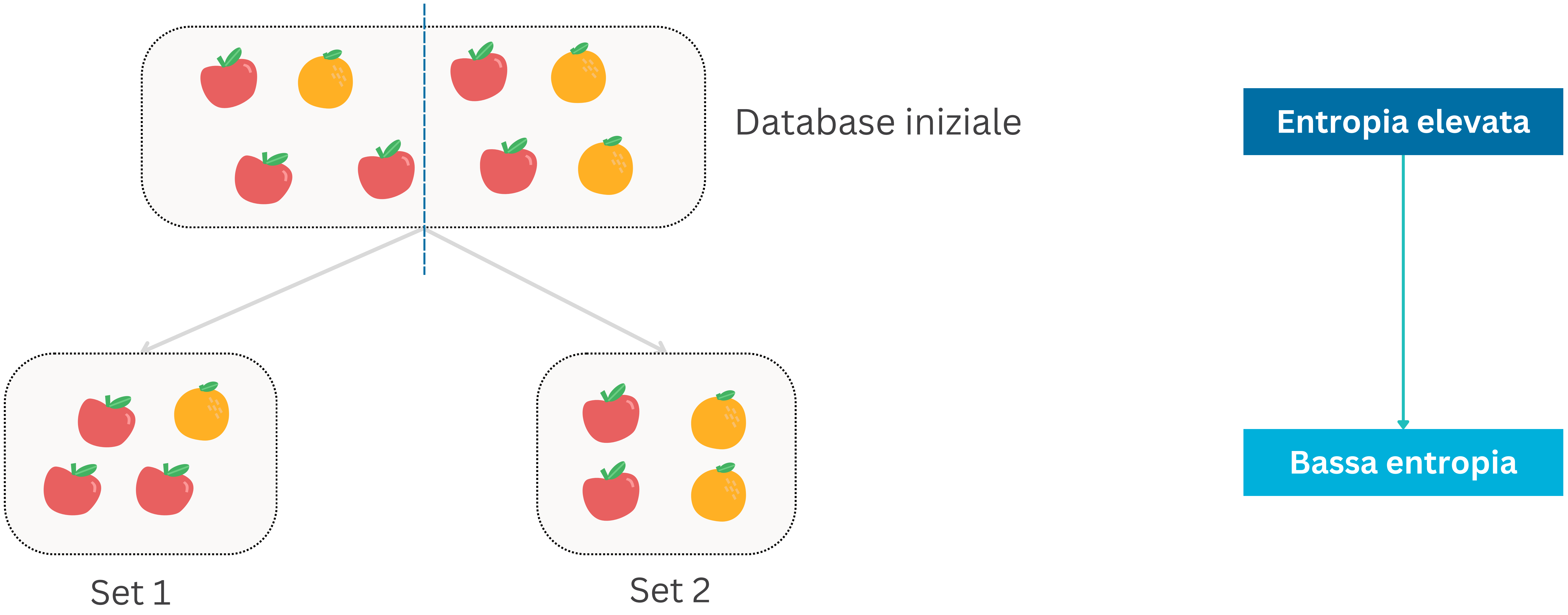
Leaf Node

Decision Node

Root Node

Entropy

Misura il livello di incertezza di un insieme di dati o di un sistema.



Decision Tree - Concetti

Entropy

Information gain

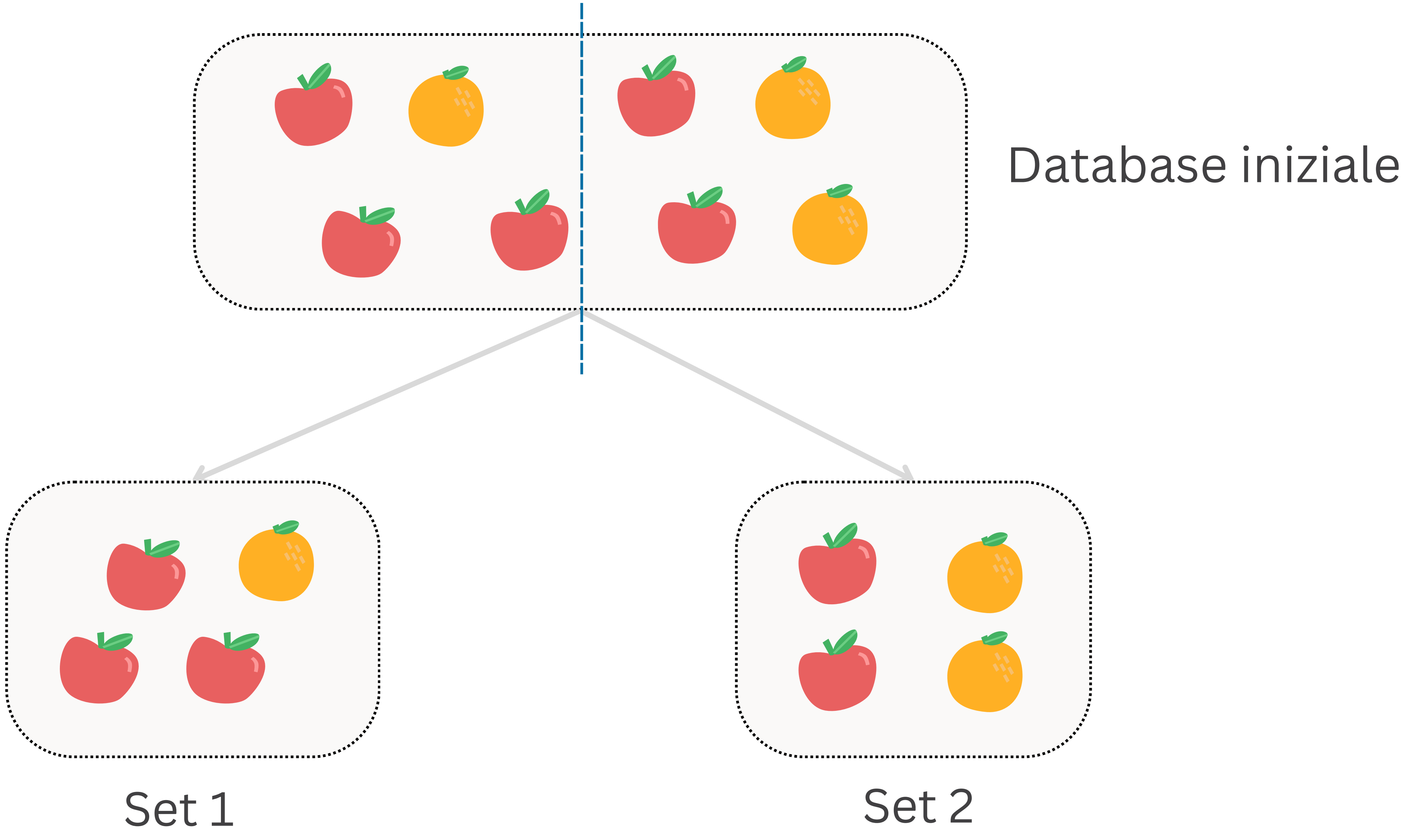
Leaf Node

Decision Node

Root Node

Information gain

La misura della diminuzione dell'entropia dopo la divisione del database.



Entropia elevata E1

Bassa entropia E2

Information Gain
E1-E2

Decision Tree - Concetti

Entropy

Information gain

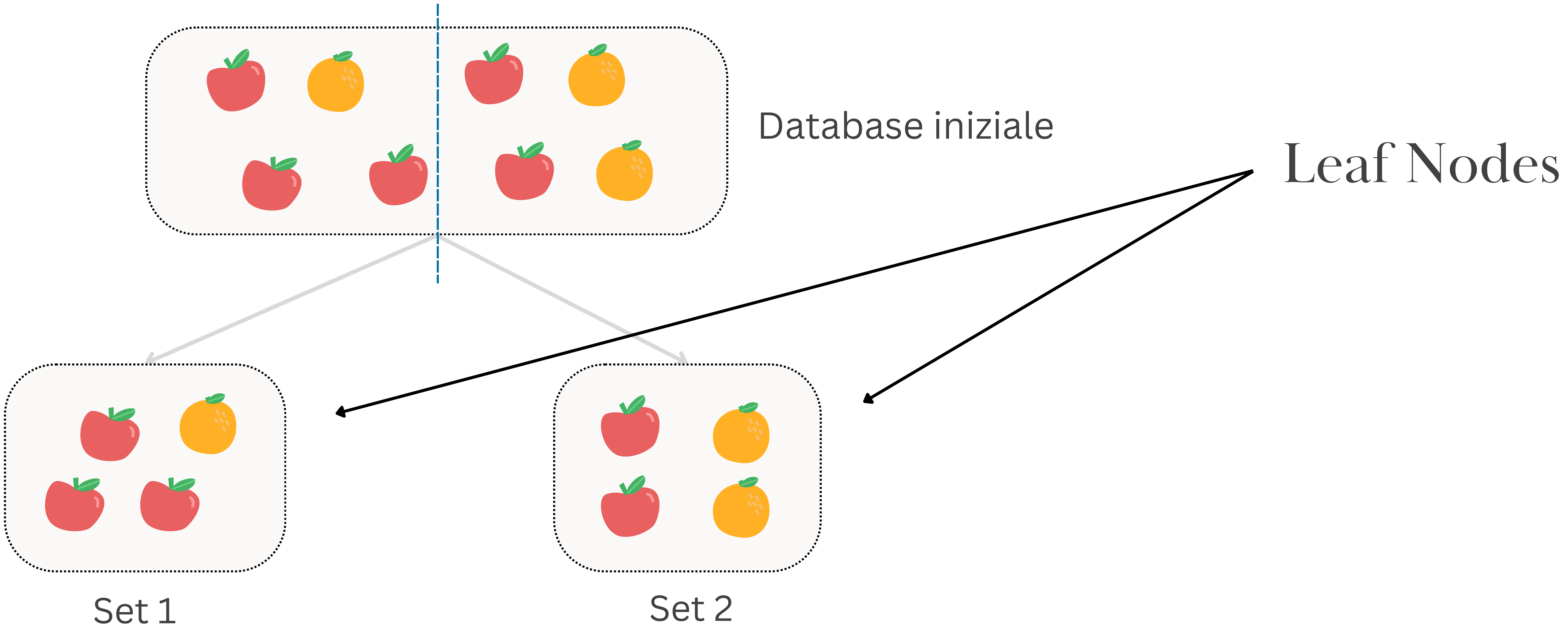
Leaf Node

Decision Node

Root Node

Leaf Node

Il nodo foglia porta la classificazione o la decisione.



Decision Tree - Concetti

Entropy

Information gain

Leaf Node

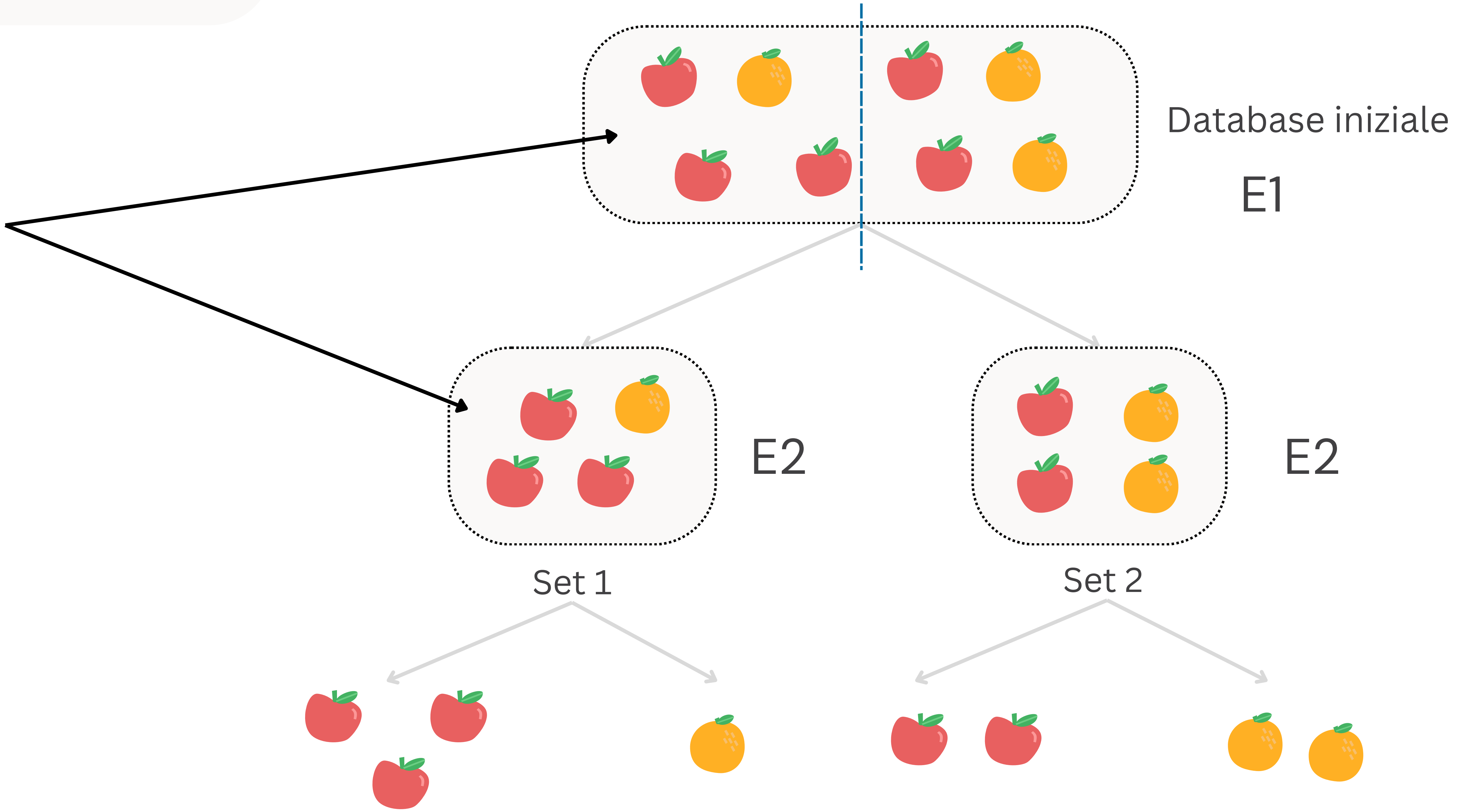
Decision Node

Root Node

Decision Node

Il nodo decisionale ha due o più rami.

Decision
Nodes



Decision Tree - Concetti

Entropy

Information gain

Leaf Node

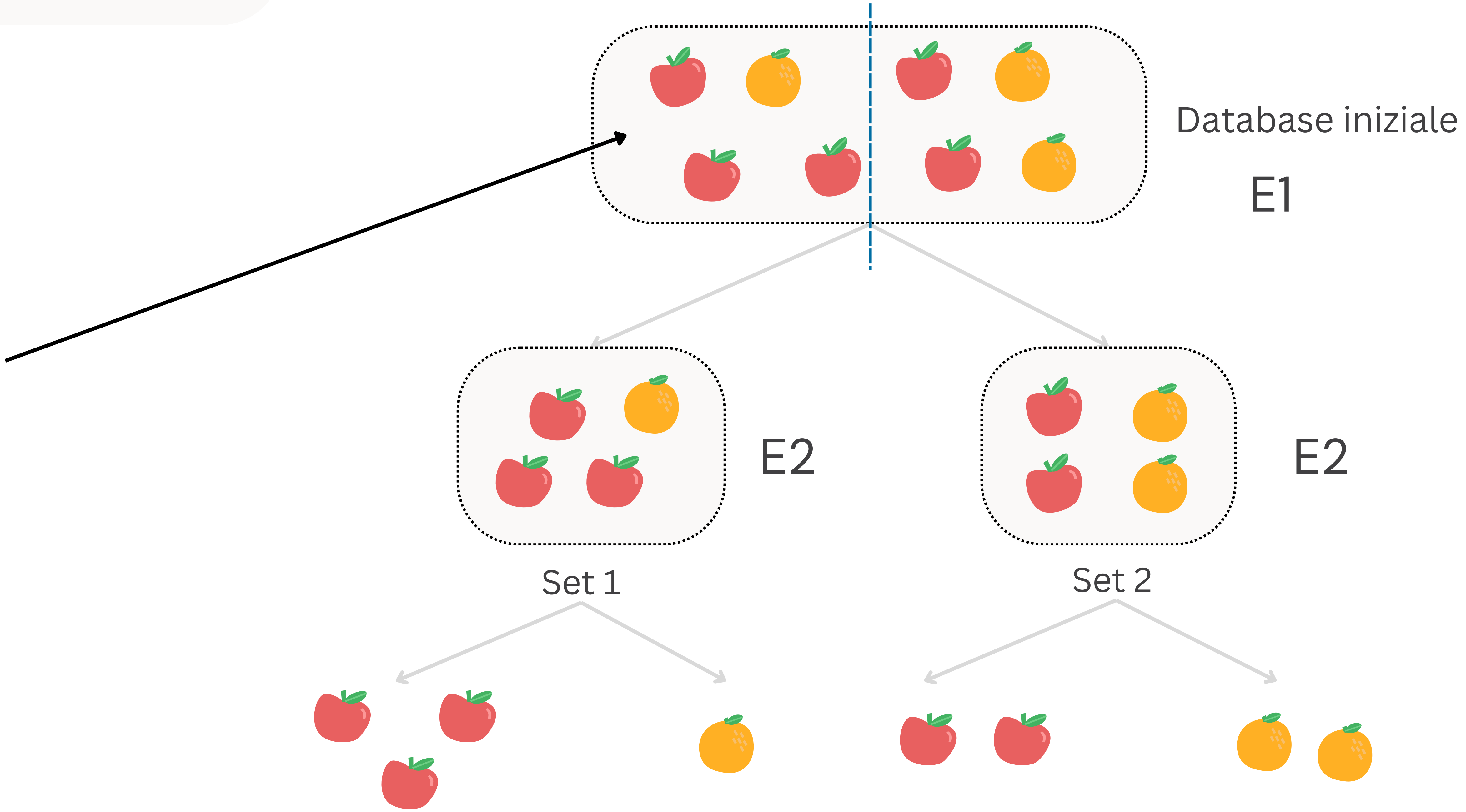
Decision Node

Root Node

Root Node

Il nodo decisionale più alto.

Root Node



Come funziona il Random Forest?

Le foreste casuali sono una modifica del metodo di **bagging** che costruisce una grande raccolta di alberi de-correlati, e quindi ne calcola la media.

Bagging = **B**ootstrap + **Agg**regation

Bagging = Bootstraping dei dati + Utilizzare Aggregation per arrivare in una decisione

Bagging

Il **bagging** è una tecnica per ridurre la varianza di una funzione di previsione stimata. Il bagging sembra funzionare specialmente bene con procedure ad alta varianza e bassa distorsione, quali gli alberi.

Bootstrap

Il bootstrapping ricampiona il database originale con sostituzione migliaia di volte per creare diversi set di dati.

Aggregation

Aggrega i risultati degli alberi.

Bagging

Altri esempi:

- Per la **regressione**, si adatta semplicemente molte volte lo stesso albero di regressione su versioni campionate via bootstrap dei dati di training, e si calcola una *media* dei risultati.

- Per la **classificazione**, si adatta un “comitato” di alberi, ognuno dei queli esprime *un voto* per la classe prevista.

Come funziona il Random Forest?

Random *perché:*

Bootstraping:
Assicura che non vengano utilizzati gli stessi dati per ogni albero -> *meno sensibile*

Random Feature Selection:
Scelta casuale del sottoinsieme di caratteristiche per ogni albero e utilizzo solo di queste per l'addestramento -> *minore correlazione tra gli alberi*

Come funziona il Random Forest?

Random *perché:*

Bootstraping

Random Feature Selection

Out of Bag Dataset: Valori che non sono stati inseriti nel set di dati di bootstrap

Misurare l'accuratezza della nostra foresta casuale in base alla percentuale di campioni fuori sacco classificati correttamente dalla foresta casuale.

Come funziona il Random Forest?

Random perché:

Bootstraping

Random Feature Selection

Out of Bag Dataset: Valori che non sono stati inseriti nel set di dati di bootstrap

Out of Bag Error (OOB): Percentuale classificata erroneamente

Cenni di teoria

L'idea fondamentale del bagging:

calcolare la media di molti modelli contenenti errore

↓

ridurre la varianza

Gli alberi sono candidati ideali per il bagging, poiché:

- catturano strutture di interazione complesse
- beneficiano dal calcolo della media poiché sono soggetti ad errore

Cenni di teoria

La distorsione di alberi “bagged” è la stessa dei singoli alberi —— miglioramento solo tramite → la riduzione della varianza.

Varianza media

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

dove:

ρ *correlazione a coppie positiva*

σ^2 *varianza*

B *numero di alberi generati*

Cenni di teoria

Quando si fa crescere un albero su un dataset “bootstrapped”:

- Prima di ogni split, seleziona a caso **$m \leq p$** delle variabili di input come candidate per lo split.

m = numero massimo di variabili di input/features
p = numero totale di variabili

Valore usualmente preferibile è pari a **p/3**.

Cenni di teoria

Quando si fa crescere un albero su un dataset “bootstrapped”:

- *Il predittore basato su foresta casuale sarà:*

$$\hat{f}^{B}_{rf}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x; \Theta_b)$$

dove:

Θ_b *b-mo albero della foresta casuale in termini di variabili di split, punto di split in ogni nodo, valori del nodo terminale*

$T_b(x; \Theta_b)$ *previsione del b-mo*

Riassunto del algoritmo

Per b = 1…B:

1. Estrai un campione bootstrap *Z** di dimensione *N* dai dati di training.
2. Fai crescere un albero
3. Seleziona ***m*** variabili a caso tra le ***p*** variabili.
4. Seleziona la migliore variabile e punto di split tra le ***m***.
5. Dividi (split) il nodo in due nodi figli.
6. Ripetendo i passaggi per ciascun nodo terminale dell'albero, fino al raggiungimento dell dimensione minima dei nodi ***n***–min.
7. Ritorna l'insieme degli alberi $\{T_b\}_1^B$

Esempio di BostonHousing in R

BostonHousing {mlbench} R Documentation

Boston Housing Data

Description

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe `BostonHousing` contains the original data by Harrison and Rubinfeld (1979), the dataframe `BostonHousing2` the corrected version with additional spatial information (see references below).

Usage

```
data(BostonHousing)
data(BostonHousing2)
```

Abbrevazioni dei variabili

The original data are 506 observations on 14 variables, `medv` being the target variable:

crim	per capita crime rate by town
zn	proportion of residential land zoned for lots over 25,000 sq.ft
indus	proportion of non-retail business acres per town
chas	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
nox	nitric oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted distances to five Boston employment centres
rad	index of accessibility to radial highways
tax	full-value property-tax rate per USD 10,000
ptratio	pupil-teacher ratio by town
b	$1000(B - 0.63)^2$ where B is the proportion of blacks by town
lstat	percentage of lower status of the population
medv	median value of owner-occupied homes in USD 1000's

Esempio di BH in R

Scarichiamo i packages che ci serviranno:

```
install.packages("randomForest")
install.packages("mlbench")
install.packages("dplyr")

library(mlbench)
library(dplyr)
library(randomForest)
```

Esempio di BH in R

Riepilogo dei dati di esempio:

```
summary(BostonHousing)
```

```
> summary(BostonHousing)
      crim                zn             indus      
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46  
 1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19  
 Median : 0.25651   Median :  0.00   Median : 9.69  
 Mean   : 3.61352   Mean   : 11.36   Mean   :11.14  
 3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10  
 Max.   :88.97620   Max.   :100.00   Max.   :27.74  
 chas         nox               rm       
 0:471   Min.   :0.3850   Min.   :3.561  
 1: 35   1st Qu.:0.4490   1st Qu.:5.886  
         Median :0.5380   Median :6.208  
         Mean   :0.5547   Mean   :6.285  
         3rd Qu.:0.6240   3rd Qu.:6.623  
         Max.   :0.8710   Max.   :8.780  
      age              dis              rad        
 Min.   :  2.90   Min.   : 1.130   Min.   : 1.000  
 1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000  
 Median : 77.50   Median : 3.207   Median : 5.000  
 Mean   : 68.57   Mean   : 3.795   Mean   : 9.549  
 3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000  
 Max.   :100.00   Max.   :12.127   Max.   :24.000  
```

Esempio di BH in R

Prepariamo i dati per l’analisi (continua...):

```
set.seed(100)

bostonhousing$chas <- factor(bostonhousing$chas, levels = 0:1,
labels = c("no", "yes"))
bostonhousing$rad <- factor(bostonhousing$rad, ordered = TRUE)

bostonhousing <- bostonhousing %>%
  select(medv, age, lstat, rm, zn, indus, chas, nox, age, dis,
rad, tax, crim, b, ptratio)
```

Esempio di BH in R

Prepariamo quindi i dati per l’analisi:

```
train <- sample(nrow(bostonhousing), 400)

bh_train <- bostonhousing[train,]
bh_test <-  bostonhousing[-train,]

(rf_fit <- randomForest(medv ~ ., data = bh_train))
```

Esempio di BH in R

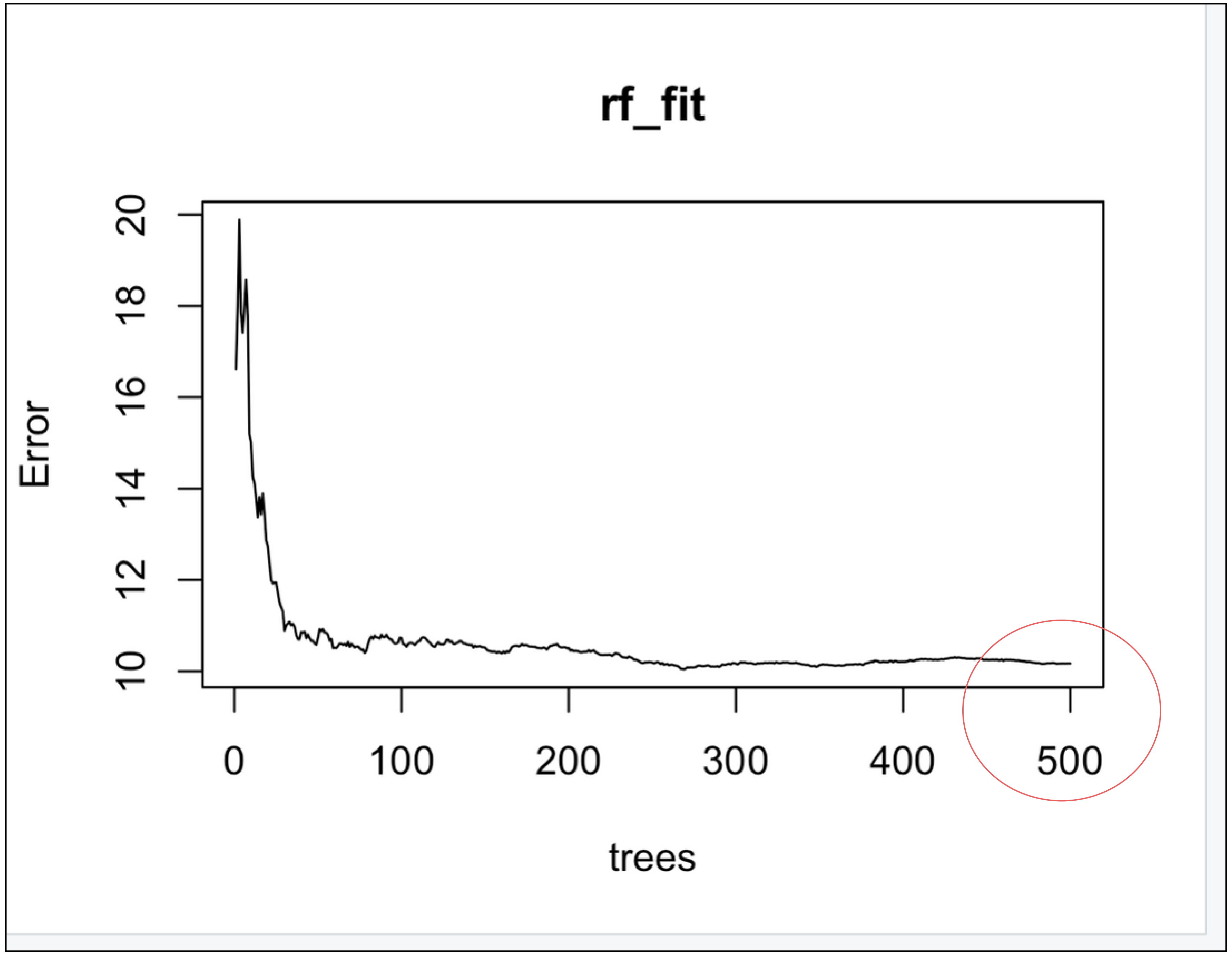
L'output delle 2 slide sopra:

```
Call:
 randomForest(formula = medv ~ ., data = bh_train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 10.17175
                    % Var explained: 87.27
```

Esempio di BH in R

```
plot(rf_fit)
```



- La linea continua mostra **l'errore complessivo** della foresta casuale quando il numero di alberi *B* cresce.
- In questo caso, un numero di alberi < 500 è **sufficiente** a minimizzare l'errore global.

Esempio di BH in R

Produciamo un grafico dei *valori osservati* vs. *i valori previsti* (continua...)

```
data_gr <- bh_train %>%
mutate(set="train") %>%
bind_rows(bh_test %>% mutate(set="test"))

data_gr$fit <- predict(rf_fit, data_gr)

library(ggplot2)
```

Esempio di BH in R

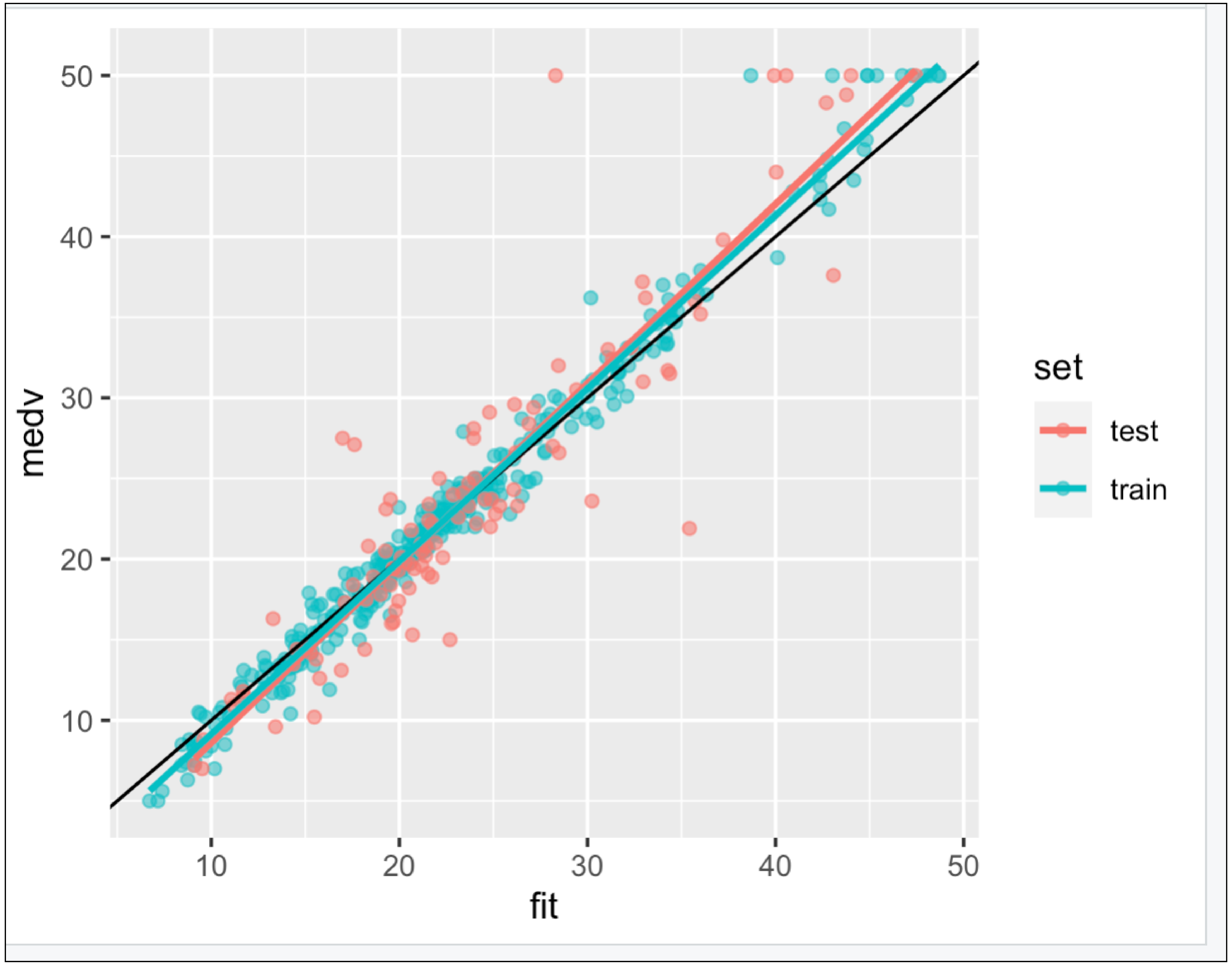
Produciamo un grafico dei *valori osservati* vs. *i valori previsti*

```
ggp <- ggplot(data = data_gr, mapping = aes(x=fit, y=medv)) +
  geom_point(aes(colour=set), alpha=0.6) +
  geom_abline(slope=1, intercept = 0) +
  geom_smooth(method = "lm", se = FALSE, aes(colour=set),
alpha=0.6)

print(ggp)
```

Esempio di BH in R

Il modello di foresta casuale mostrato estrae casualmente per ogni split 4 variabili da considerare per lo split stesso.



Esempio di BH in R

Cosa accade facendo variare il numero di predittori estratti tra 1 e tutti i 13 possibili predittori utilizzabili ad ogni split?

```
set.seed(100)
oob_err <- double(13)
test_err <- double(13)
```

Esempio di BH in R

```
#mtry è il numero di variabili scelte casualmente ad ogni split
for(mtry in 1:13) {
rf <- randomForest(medv ~ . , data = bh_train, mtry=mtry,
ntree=400)
oob_err[mtry] <- rf$mse[400] #Errore per tutti gli alberi adattati
pred <- predict(rf,bh_test) #Previsioni sul set di test per ciascun
albero
test_err[mtry] <- with(bh_test, mean( (medv - pred)^2)) #Mean
Squared Error per l'insieme di test
}
#Errore sull'insieme di Test
test_err
```

Esempio di BH in R

L’output delle 2 slide sopra:

```
> test_err
 [1] 28.21875 19.55995 17.51116 16.42723 16.63094 15.60248
 [7] 16.48503 16.14524 17.02976 16.84301 17.57899 17.91067
[13] 17.81792
```

Esempio di BH in R

Stima dell'errore Out of Bag

```
oob_err
```

```
> oob_err
 [1] 19.310462 12.309479 11.158004  9.985156  9.395468
 [6]  9.972013  9.516923  9.839743  9.792508  9.876895
[11]  9.724254 10.430090 10.716280
```

Esempio di BH in R

Il grafico dell'**Errore dell'insieme di test** e l'**Errore Out of Bag** (continua...)

```
ds_gr <- data_frame(type=c(rep("test", length(test_err)),
rep("oob", length(oob_err))),
mtry = c(1:length(test_err), 1:length(oob_err)),
error=c(test_err, oob_err))
```

Esempio di BH in R

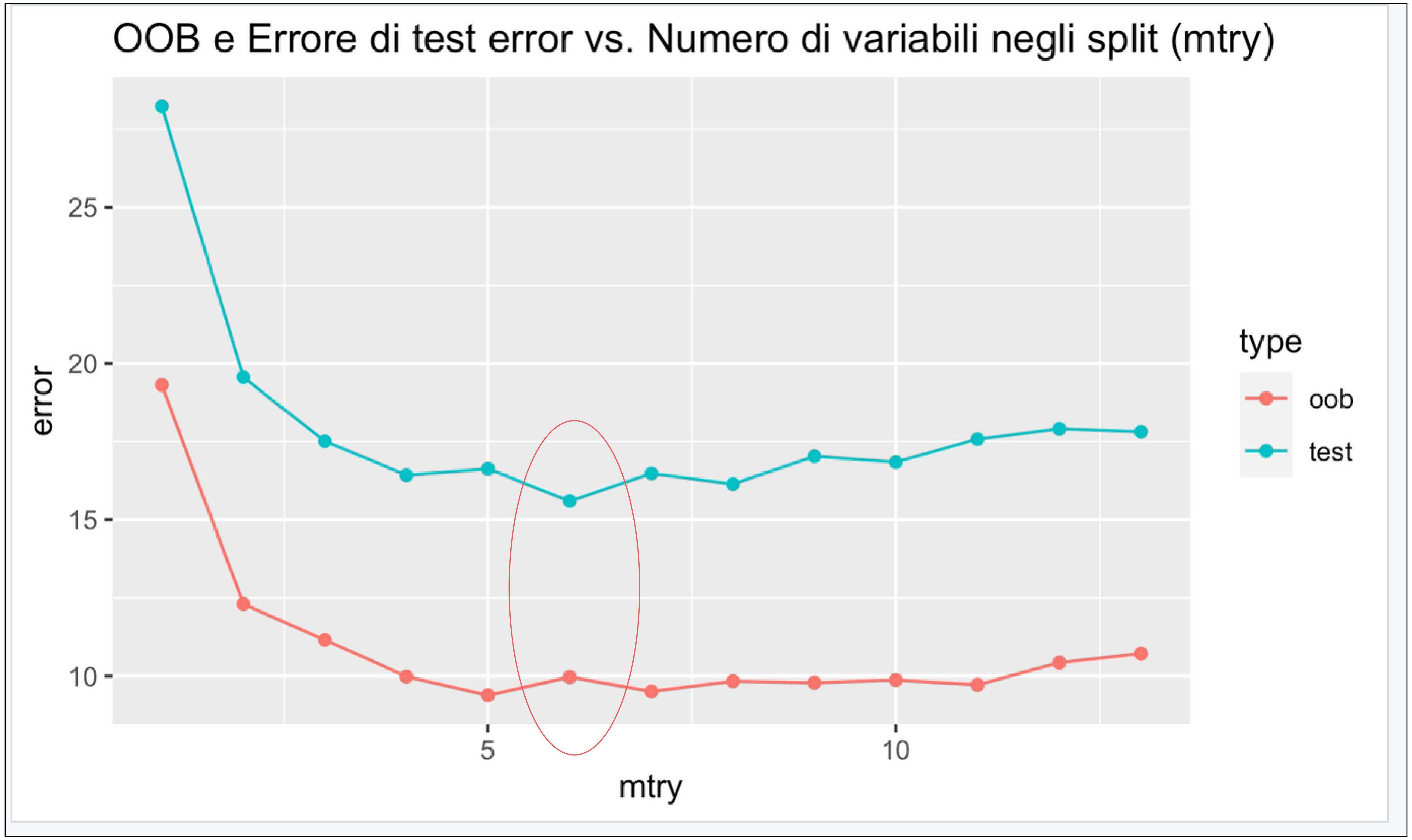
Il grafico dell'**Errore dell'insieme di test** e l'**Errore Out of Bag**

```
ggp <- ggplot(data = ds_gr, mapping = aes(x=mtry,y=error)) +
  geom_line(aes(colour=type)) +
  geom_point(aes(colour=type)) +
  ggtitle("OOB e Errore di test error Vs. Numero di variabili
negli split (mtry)")

print(ggp)
```

Esempio di BH in R

Sia l'errore di Test, sia l'errore OOB, tendono a minimizzarsi intorno a mtry = 6.



Esempio di BH in R

Ora possiamo quindi provare il modello con **mtry = 6**

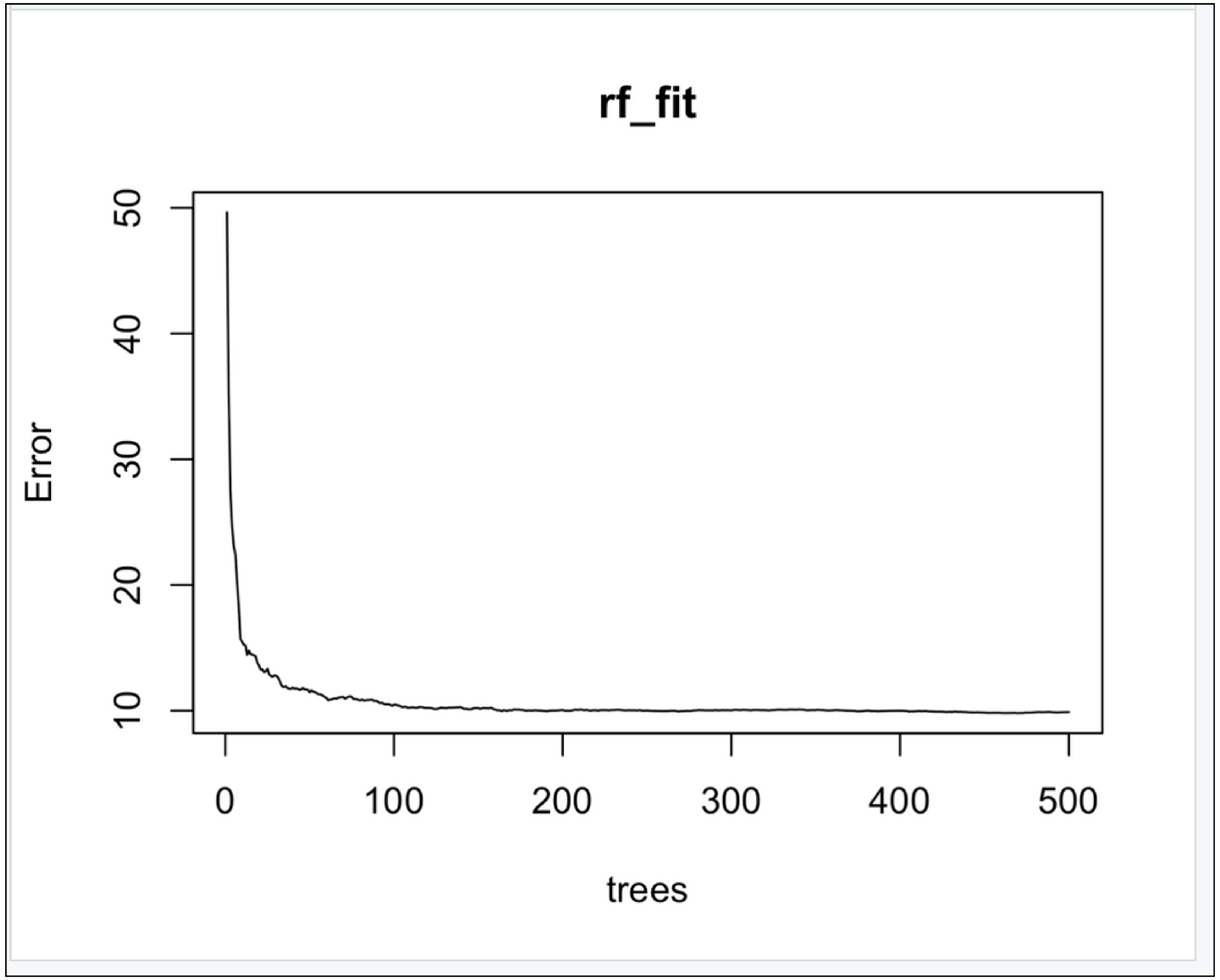
```
set.seed(100)
(rf_fit <- randomForest(medv ~ ., data = bh_train, mtry=6)
```

```
Call:
 randomForest(formula = medv ~ ., data = bh_train, mtry = 6)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 9.879587
                    % Var explained: 87.64
```

Esempio di BH in R

```
plot(rf_fit)
```



Esempio di BH in R

```
data_gr <- bh_train %>%
  mutate(set="train") %>%
  bind_rows(bh_test %>% mutate(set="test"))

data_gr$fit <- predict(rf_fit, data_gr)

mse <- data_gr %>%
  filter(set=="test") %>%
  summarise(mse = mean((fit-medv)^2)) %>%
  pull()

print(mse)
```

```
[1] 16.24758
```

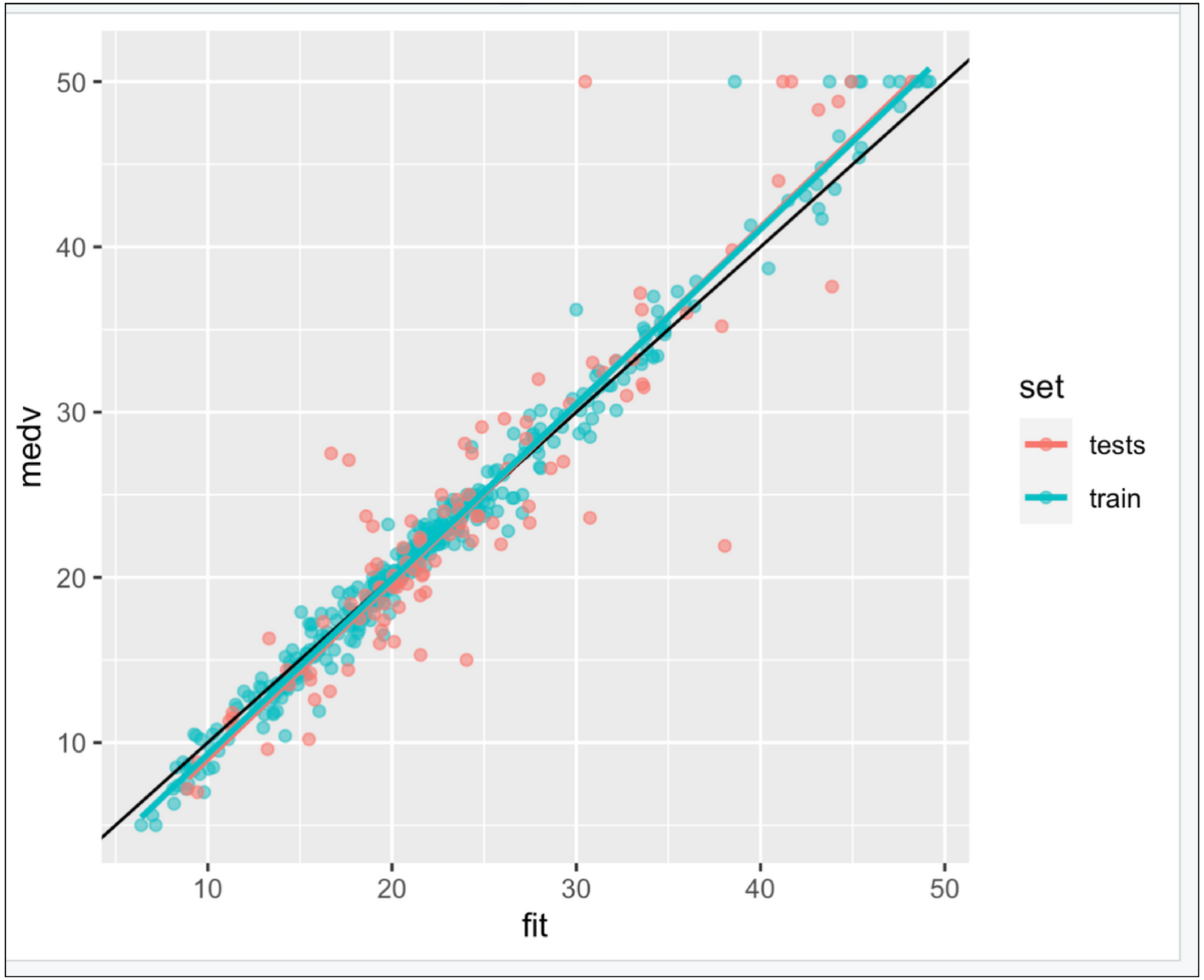
Esempio di BH in R

```
ggp <- ggplot(data = data_gr, mapping = aes(x=fit, y=medv)) +
  geom_point(aes(colour=set), alpha=0.6) +
  geom_abline(slope=1, intercept = 0) +
  geom_smooth(method = "lm", se = FALSE, aes(colour=set),
alpha=0.6)

print(ggp)
```

Esempio di BH in R

- Non sono evidenti grandi differenze rispetto al modello precedente.
- L'errore quadratico medio per l'insieme di test è pari a 16.2475782.



Vantaggi vs. Svantaggi

Vantaggi

- Riduce il rischio di overfitting
- Fornisce flessibilità
- Facile determinazione dell'importanza delle caratteristiche

Svantaggi

- Processo che richiede tempo
- Richiede più risorse
- Più complesso

Fonti e Appunti per me

- https://www.ibm.com/topics/random-forest
- https://search.r-project.org/CRAN/refmans/mlbench/html/BostonHousing.html
- https://cran.r-project.org/web/packages/randomForest/index.html
- http://www.r-project.it/_book/random-forest-rf-1.html