

Machine learning: modelli e algoritmi

STATISTICA NUMERICA 2024-25

ELENA LOLI PICCOLOMINI

Capire l'AI: modelli e algoritmi alla portata di tutti

Corso 2: algoritmi e previsioni

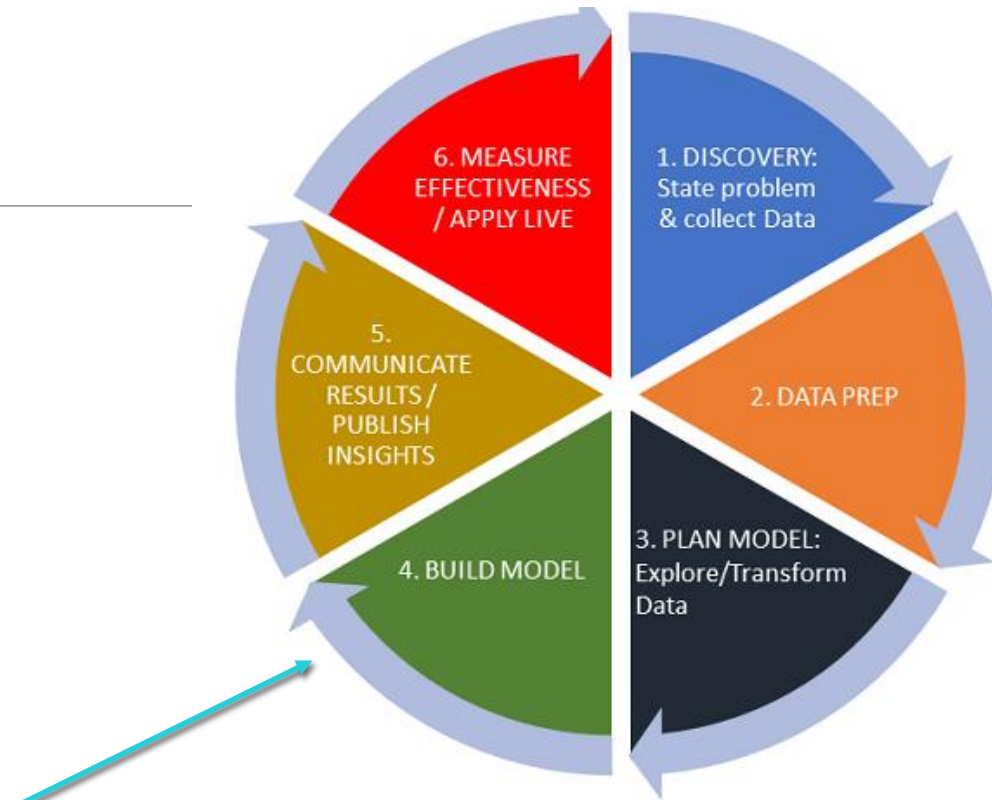
Elena Morotti

7 marzo 2025

MOHO, Science & Technology

A solid blue horizontal bar at the bottom of the slide.

Data Science

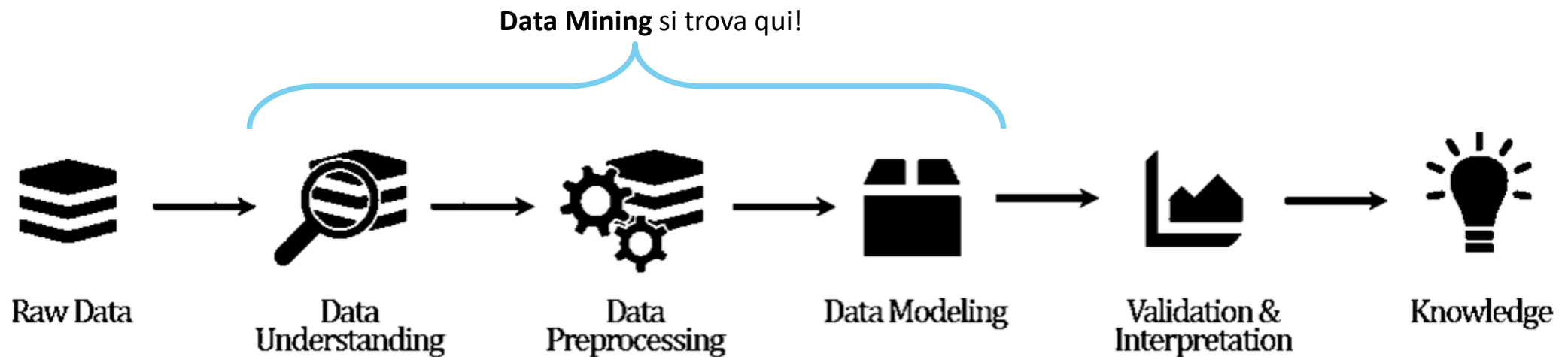


MACHINE LEARNING ALGORITHMS

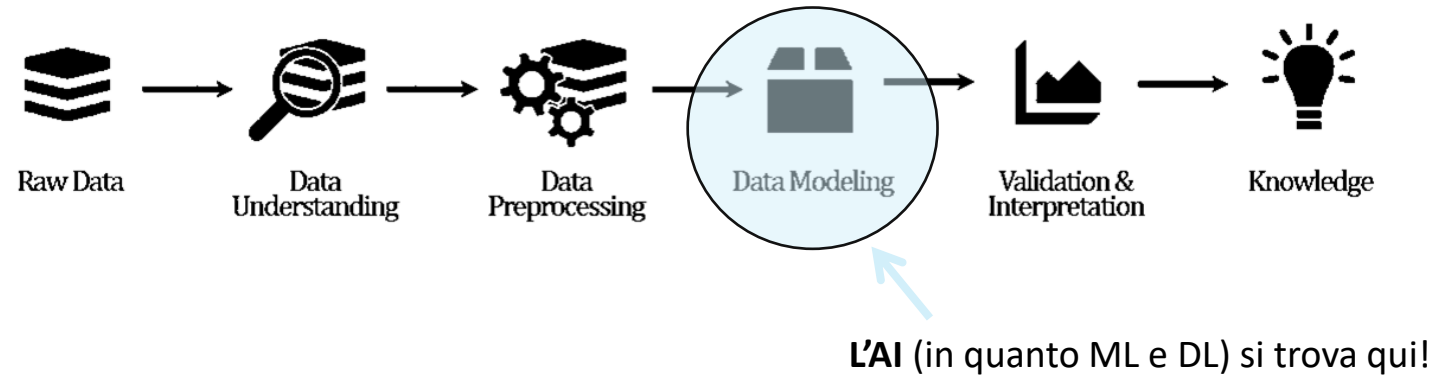
Data Science life cycle

L'AI richiede l'intelligenza umana

Il ciclo di vita della Data Science è lungo e articolato (questo è il *Knowledge Discovery process*, una visione tecnica del ciclo di vita della DS).



L'AI richiede l'intelligenza umana-bis

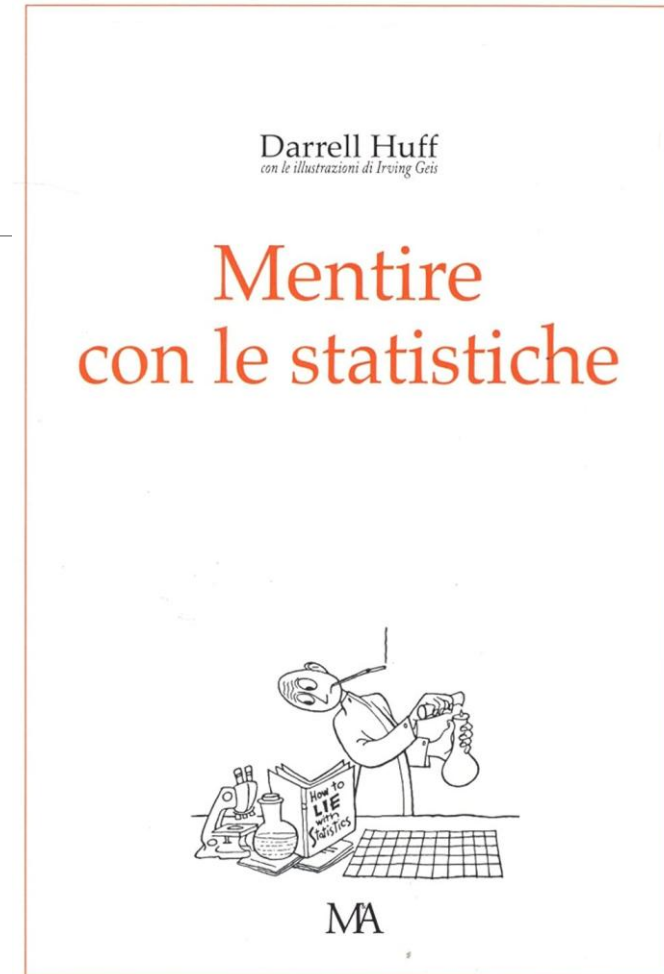


GIGO è il concetto che un'informazione errata, distorta o di scarsa qualità ("spazzatura" come input) produce un risultato di qualità simile ("spazzatura" come output).

Alcune letture
interessanti

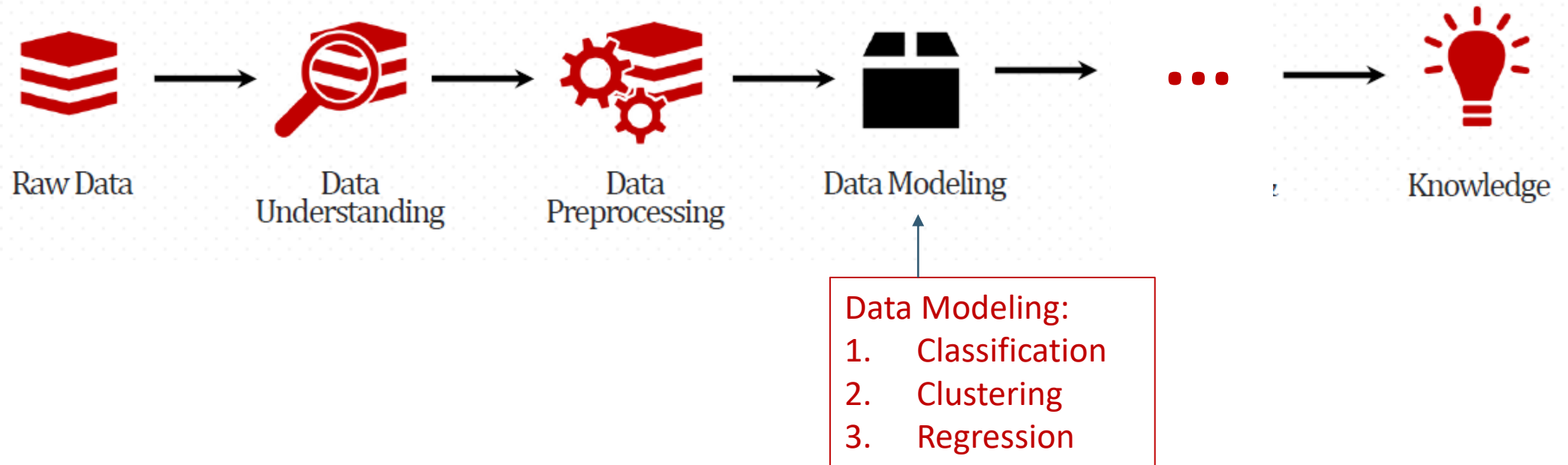


"Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy", 2016

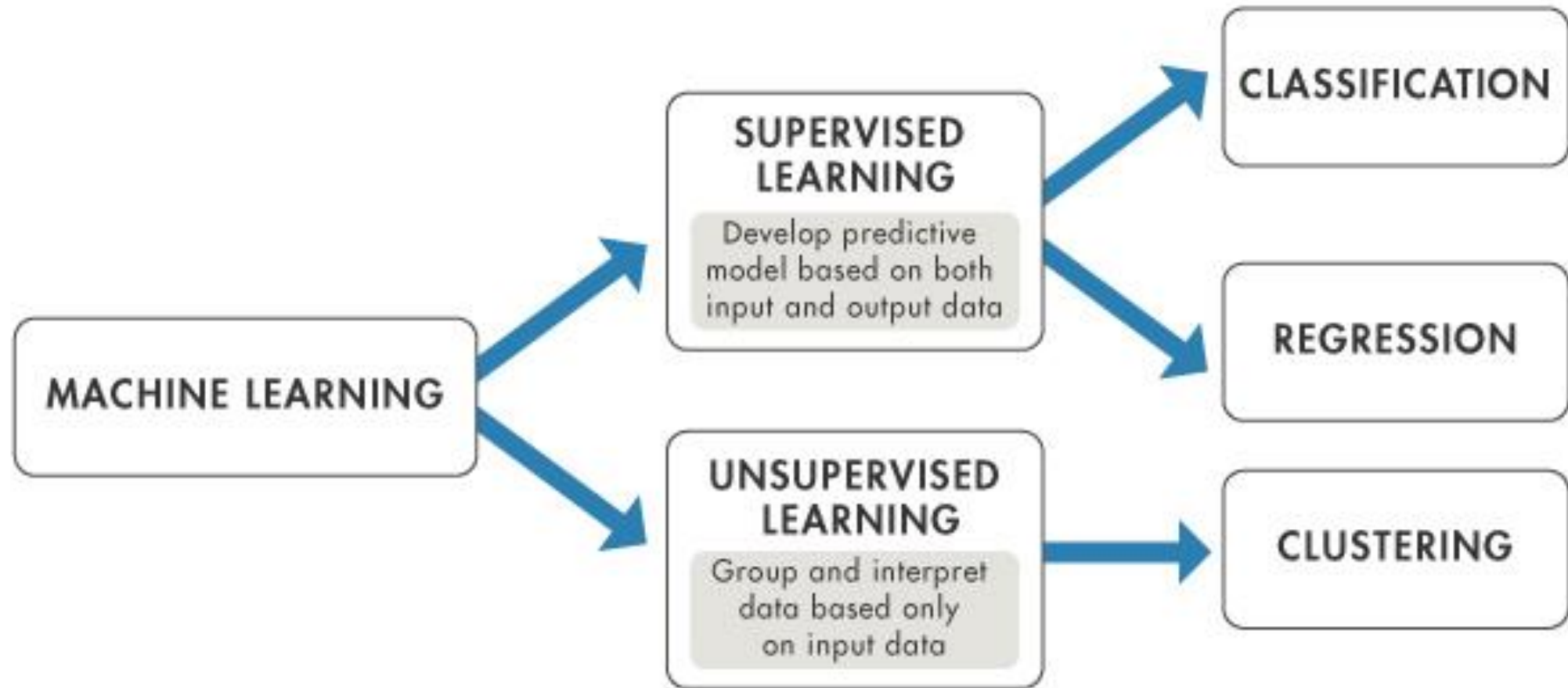


"How to Lie with Statistics", 1954

Processo per l'analisi dati



Algoritmi di ML (ovvero, per imparare dai dati disponibili)



Fasi di un algoritmo di Machine Learning supervisionato

1. Suddivisione dei dati in train e test (e validazione come vedremo in seguito)
2. Scelta del modello che dipende da un insieme di parametri Θ
3. Identificazione dei parametri durante la fase di training (imparando dai dati) utilizzando i dati target (o ground truth)
4. Applicazione del modello con i parametri stimati ai dati di test per verificare, tramite misure di errore sempre utilizzando i dati target, l'efficacia del modello ottenuto.
5. Applicazione del modello a dati non appartenenti al data set di cui non ho la ground truth e per i quali quindi non posso stimare errore.

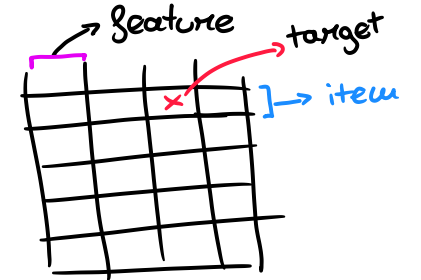
Applico il modello su dati non appartenenti al data set → Ottengo una generalizzazione

CLASSIFICAZIONE

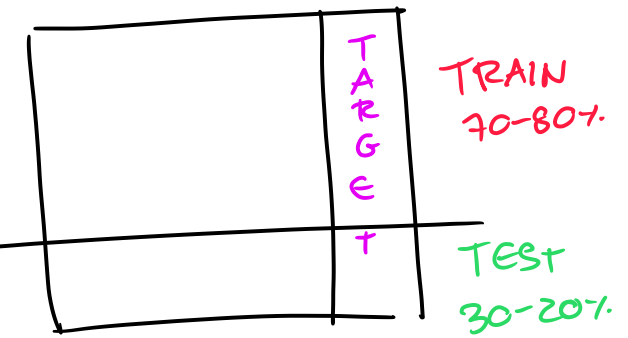
↳ target (variabile categorica)

REGRESSIONE

↳ target (variabile numerica)

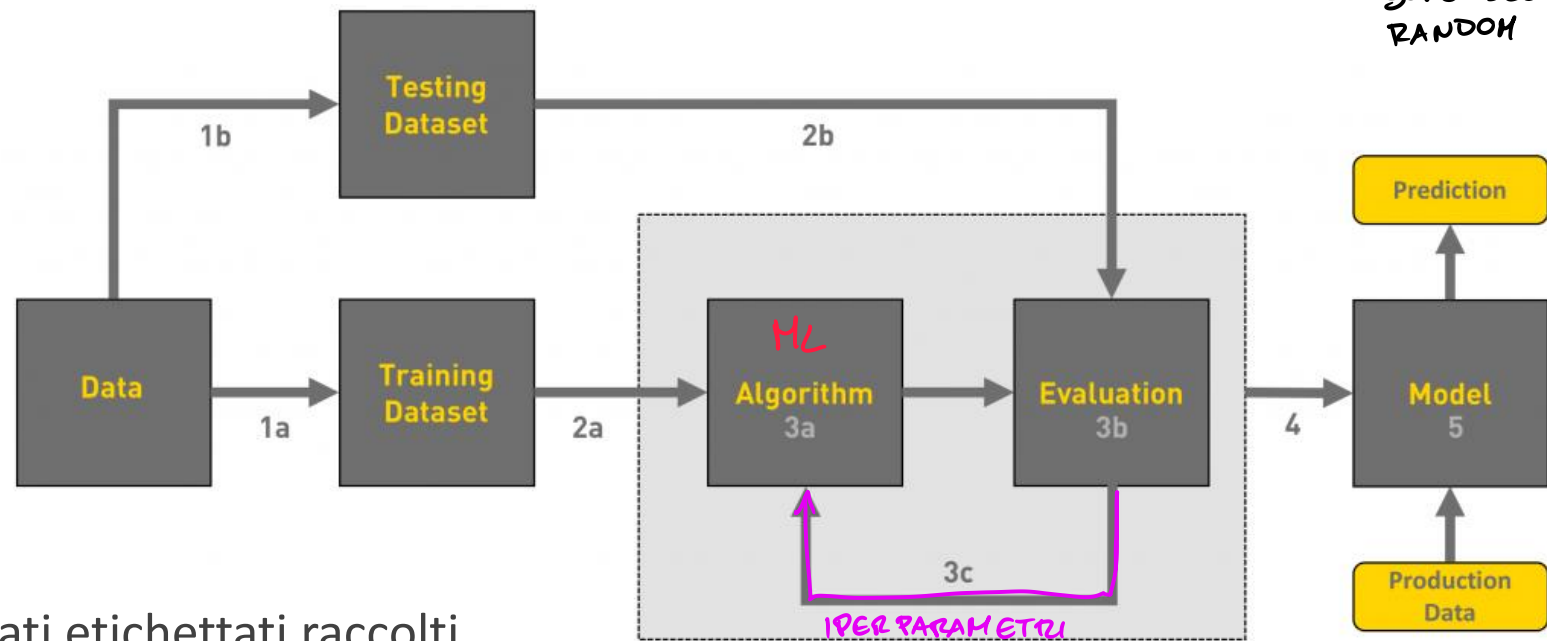


Divisione in «Train and Test»



Parte dei dati disponibili si usano per valutare le capacità predittive del classificatore.

Train e test
sono scelti in modo
RANDOM



Nei progetti di ML:

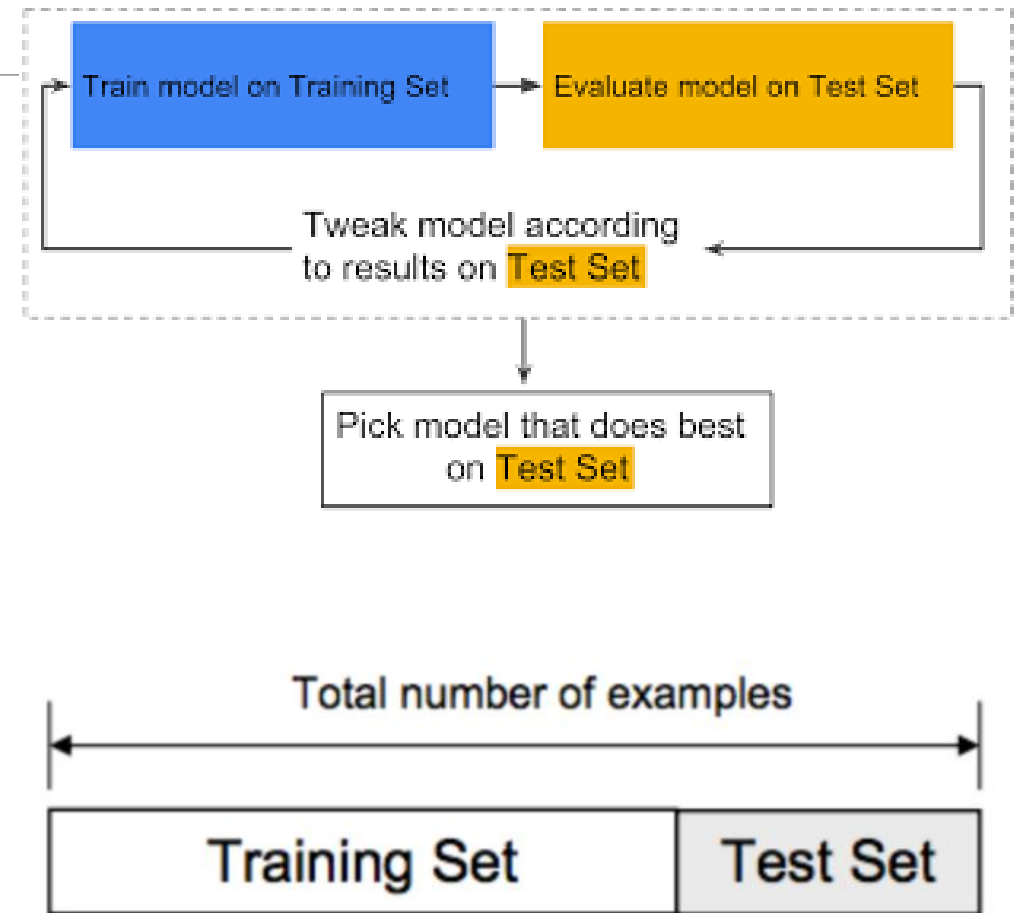
- 1) Si dividono i dati etichettati raccolti
- 2) Si addestra il classificatore sul sottoinsieme di *training*
- 3) Si usa il sottoinsieme di *testing* per il calcolo delle metriche di qualità

Divisione dei dati

- **Training Set:**
è il sottoinsieme di dati utilizzato per addestrare il modello di ML. Il modello osserva e apprende da questi dati e ottimizza i suoi parametri.
- **Test set:**
è il campione di dati utilizzato per fornire una valutazione imparziale dell'adattamento del modello finale al set di dati di addestramento. Generalmente è utilizzato per valutare diversi modelli nelle competizioni.
La fase di testing replica il tipo di situazione che si incontrerà una volta che il modello sarà distribuito per l'uso in tempo reale.

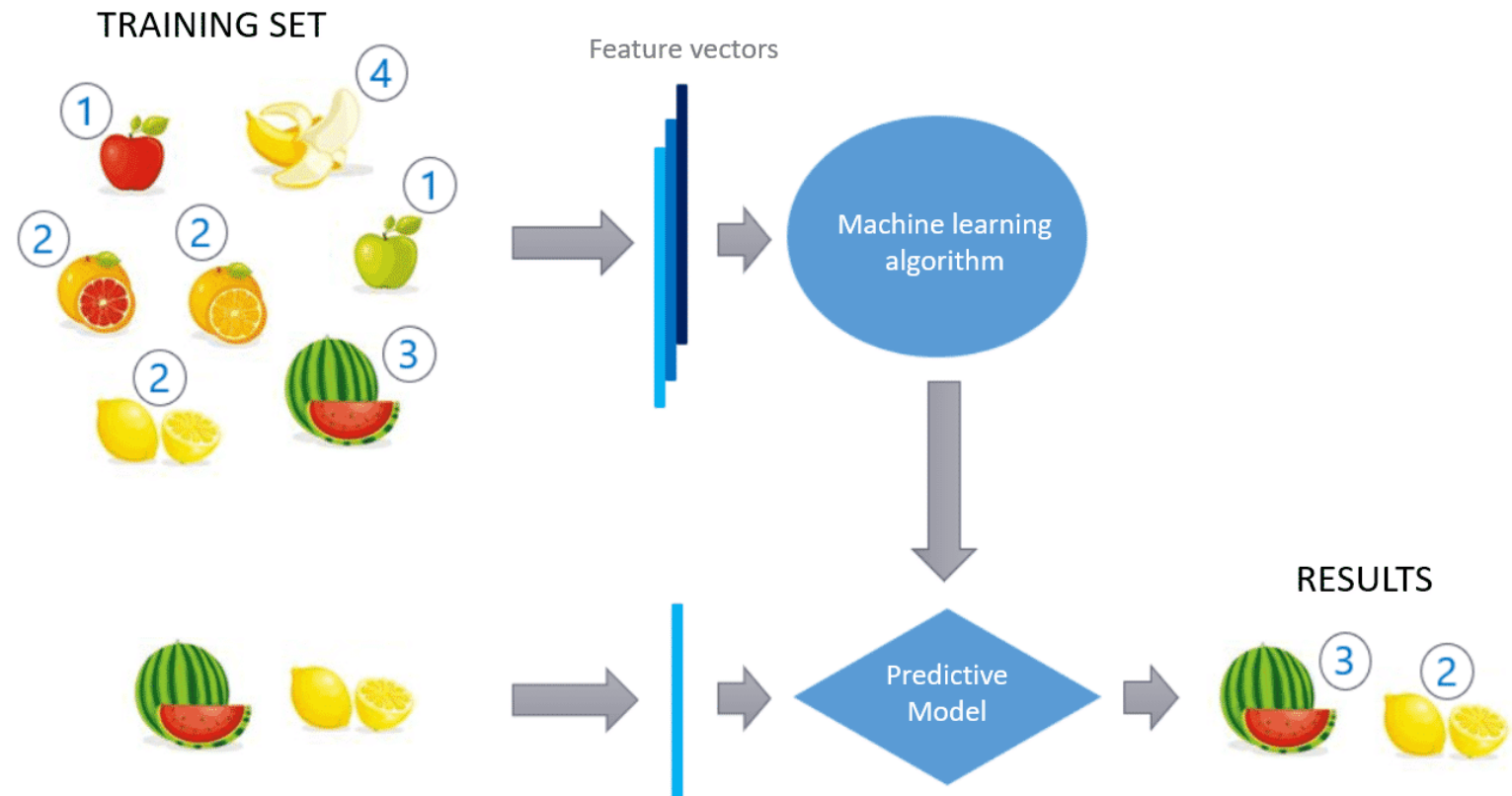
Tipicamente si divide con rapporto di training/testing dato da:

- 70/30 o 80/20 per dataset piccoli;
- 50/50 o 95/5 per dataset grandi.

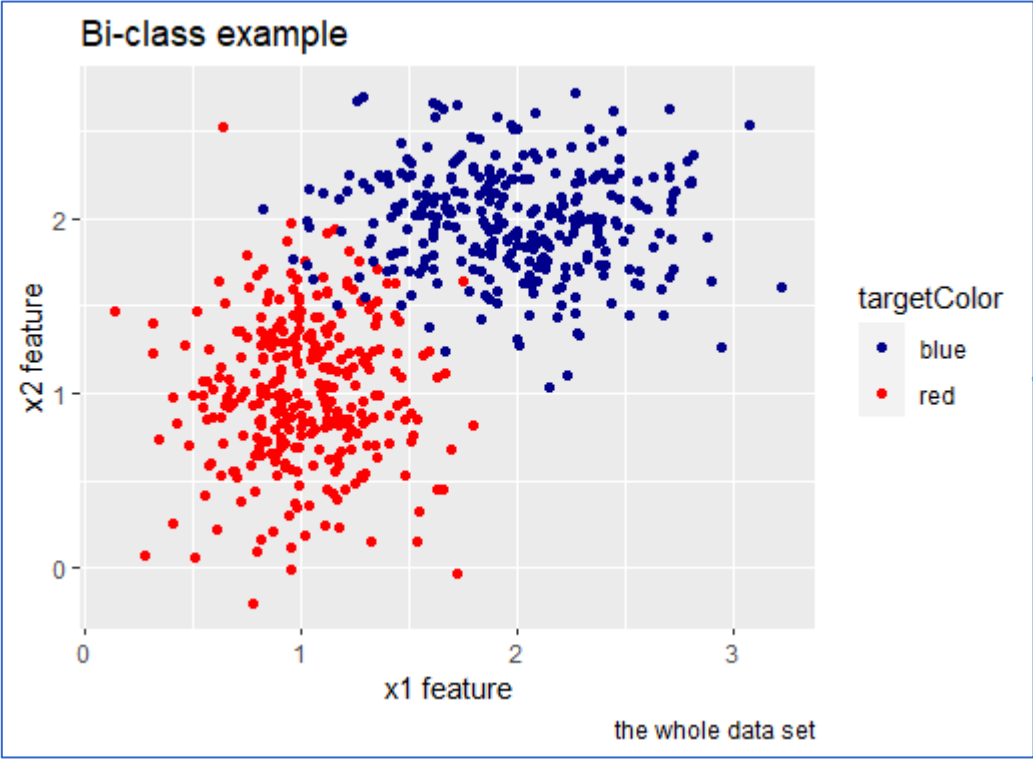


Classificazione

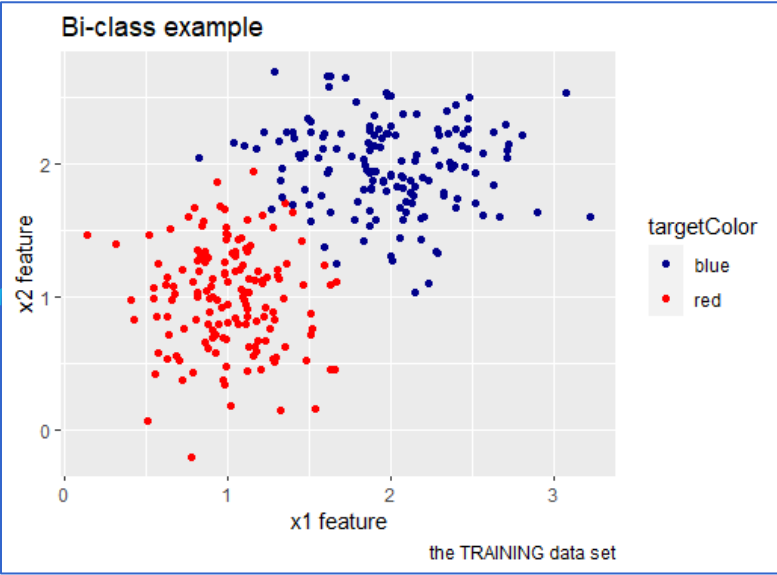
Con
«classificazione»
intendiamo un
insieme di
metodologie
utilizzate per
prevedere una
variabile *categorica*
di risposta da una o
più variabili
predittive



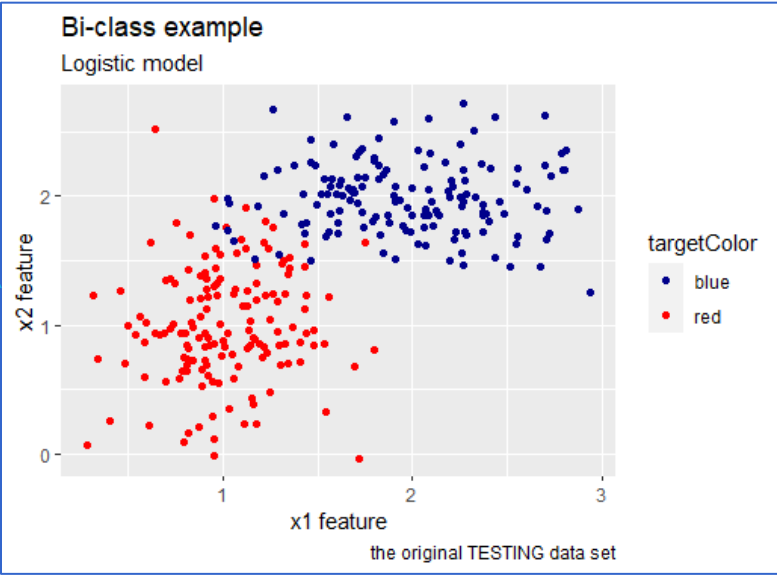
Occorre divider i sottoinsiemi mantenendo la rappresentatività delle classi da predire!



blue red
300 300



blue red
154 146



blue red
146 154

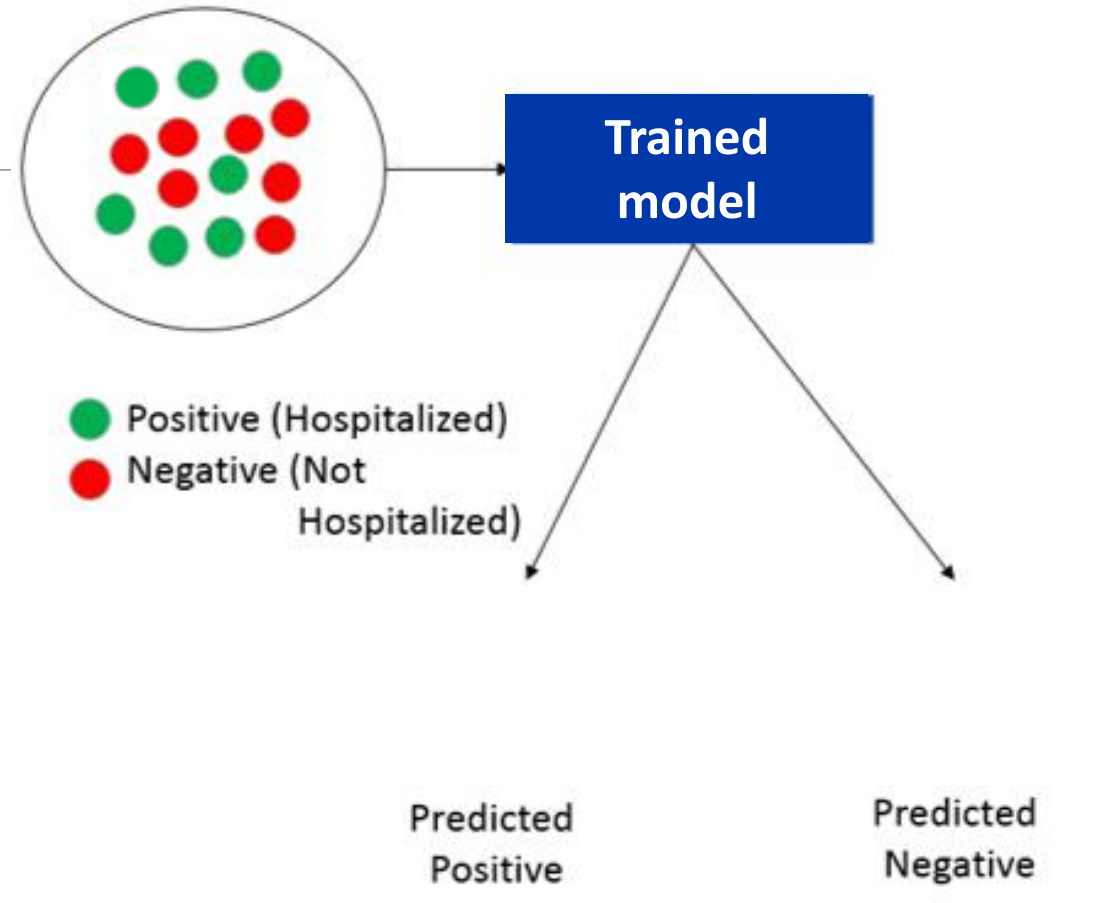
Esempio

	Actual POSITIVE (verde)	Actual NEGATIVE (rosso)	tot
Predicted POSITIVE (verde)	4	2	6
Predicted NEGATIVE (rosso)	3	5	8
tot	7	7	14

Abbiamo:

- 4 True Positive
 - 5 True Negative
 - 2 False Positive
 - 3 False Negative
- } *predizioni corrette*
- } *predizioni errate*

Test set



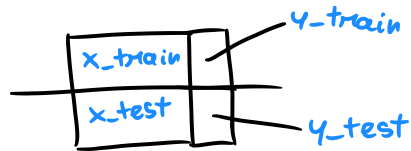
Terminologia derivata dalla matrice di confusione

- **Accuratezza (ACC)**
percentuale di predizioni corrette (positive e negative che siano), sul totale;
- **errore di misclassificazione**
percentuale di errori, sul totale.
- **Sensitività, recall, hit rate, o true positive rate (TPR)**
percentuale di casi positivi predetti correttamente;
- **Specificità, o true negative rate (TNR)**
percentuale di casi negativi predetti correttamente.
- **Precisione, o positive predictive value (PPV)**
percentuale di predizioni positive corrette;
- **Negative predictive value (NPV)**
percentuale di predizioni negative corrette.

		True class		Measures
		Positive	Negative	
Predicted class	Positive	True positive TP	False positive FP	Positive predictive value (PPV) $\frac{TP}{TP+FP}$
	Negative	False negative FN	True negative TN	Negative predictive value (NPV) $\frac{TN}{FN+TN}$
Measures		Sensitivity $\frac{TP}{TP+FN}$	Specificity $\frac{TN}{FP+TN}$	Accuracy $\frac{TP+TN}{TP+FP+FN+TN}$

divido dataset \rightarrow X (features)
 \rightarrow Y (target)

split $\begin{cases} X_{\text{train}} \text{ (features train)} \text{ (matrice)} \\ Y_{\text{train}} \text{ (target train)} \text{ (array)} \\ X_{\text{test}} \text{ (features test)} \text{ (matrice)} \\ Y_{\text{test}} \text{ (features test)} \text{ (array)} \end{cases}$



- addestramento (training set)

$\text{model.fit}(X_{\text{train}}, Y_{\text{train}})$

- predizione (test set)

$Y_{\text{pred}} = \text{model.predict}(X_{\text{test}})$

\downarrow
array

- Valutazione con la matrice di confusione

$\text{confusion_matrix}(Y_{\text{test}}, Y_{\text{pred}})$

	Actual POSITIVE (verde)	Actual NEGATIVE (rosso)	tot
Predicted POSITIVE (verde)	4	2	6
Predicted NEGATIVE (rosso)	3	5	8
Tot	7	7	14

		True class		Measures
		Positive	Negative	
Predicted class	Positive	True positive <i>TP</i>	False positive <i>FP</i>	Positive predictive value (PPV) $\frac{TP}{TP+FP}$
	Negative	False negative <i>FN</i>	True negative <i>TN</i>	Negative predictive value (NPV) $\frac{TN}{FN+TN}$
Measures		Sensitivity $\frac{TP}{TP+FN}$	Specificity $\frac{TN}{FP+TN}$	Accuracy $\frac{TP+TN}{TP+FP+FN+TN}$

Abbiamo:

- 4 True Positive
- 5 True Negative
- 2 False Positive
- 3 False Negative

Quindi:

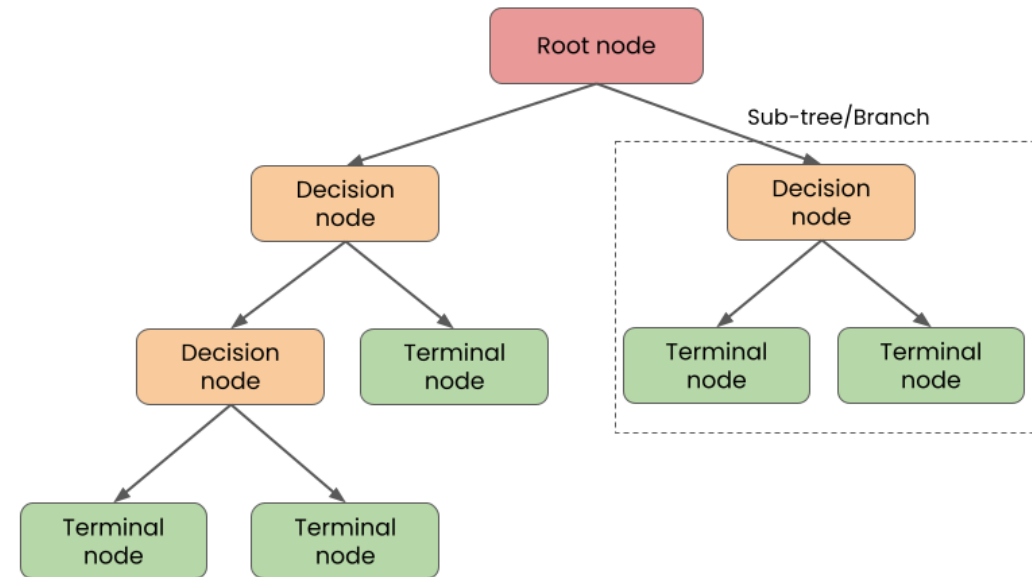
- Accuratezza = 64% $(9/14 = 0,64)$
- Misclassification Error = 36% $(5/14 = 0,36)$
- Sensitività = 57% $(4/7 = 0,57)$
- Specificità = 71% $(5/7 = 0,71)$
- Precisione = 66% $(4/6 = 0,66)$

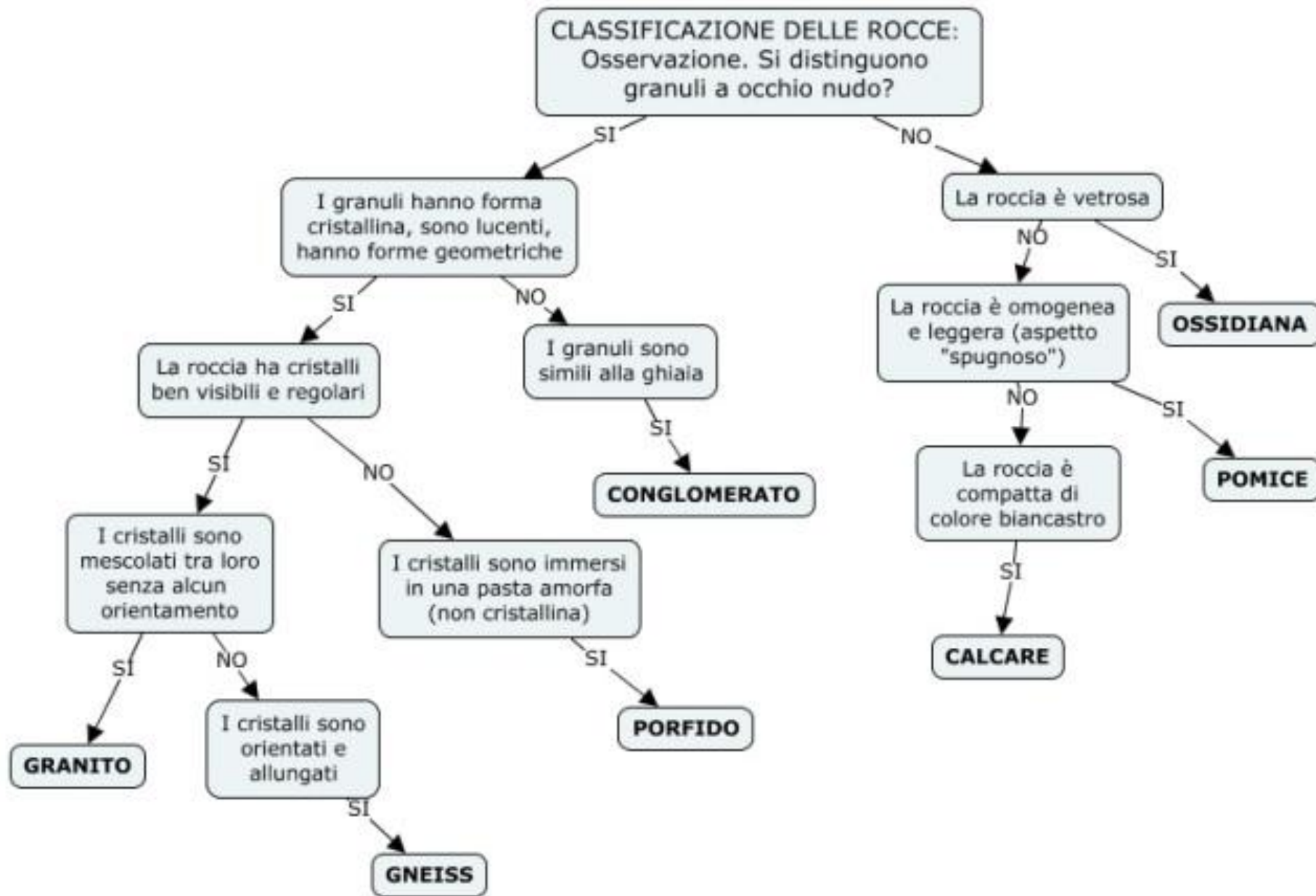
Albero decisionale come classificatore

Suddivide un set di dati in sottoinsiemi sempre più piccoli; allo stesso tempo l'albero cresce incrementalmente.
Il risultato finale è un albero con nodi decisionali e nodi terminali (foglie).

Il nodo più in alto (nodo radice) di un albero corrisponde all'intero set di dati.

Un nodo decisionale ha due o più rami e un nodo foglia rappresenta una classificazione o una decisione.

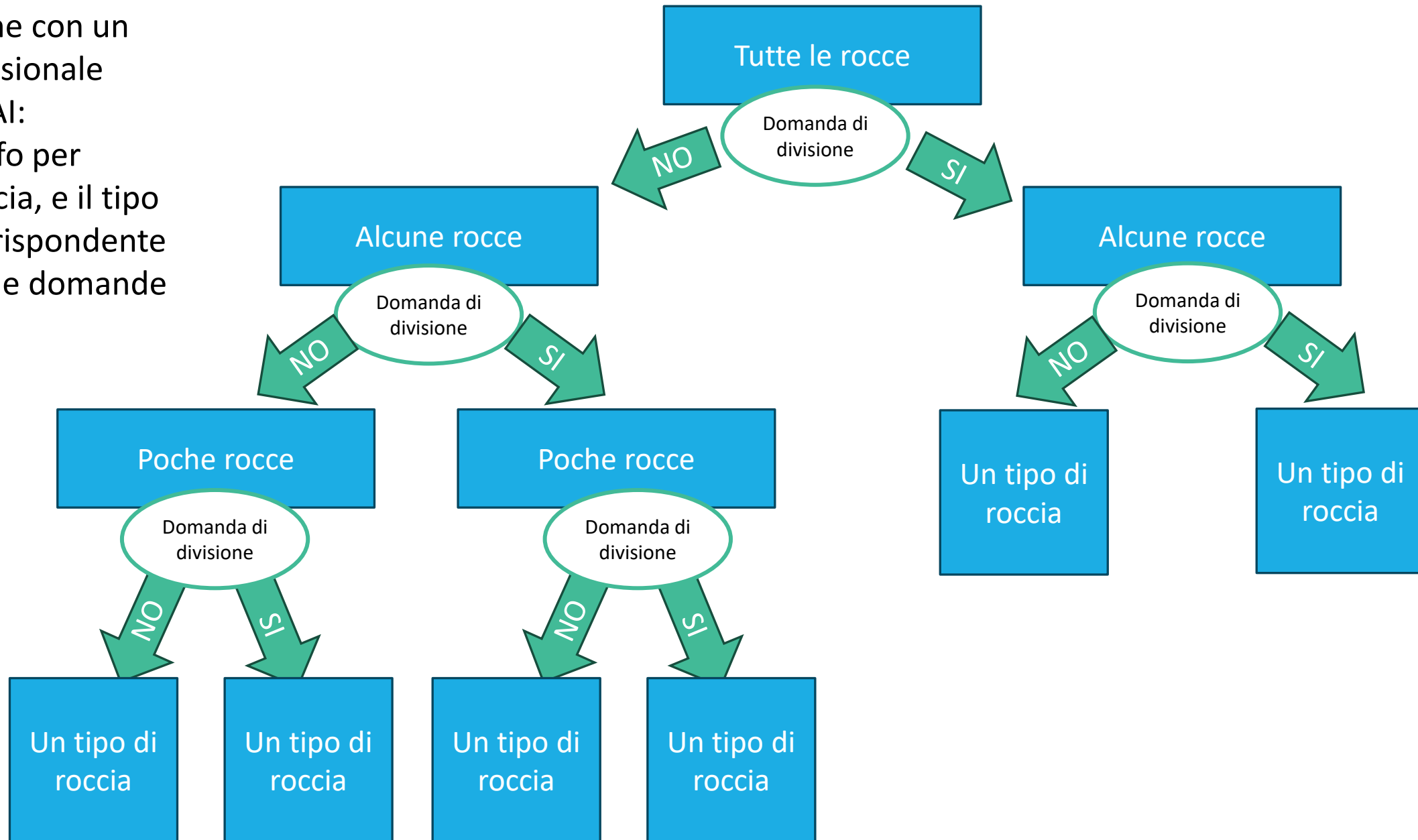


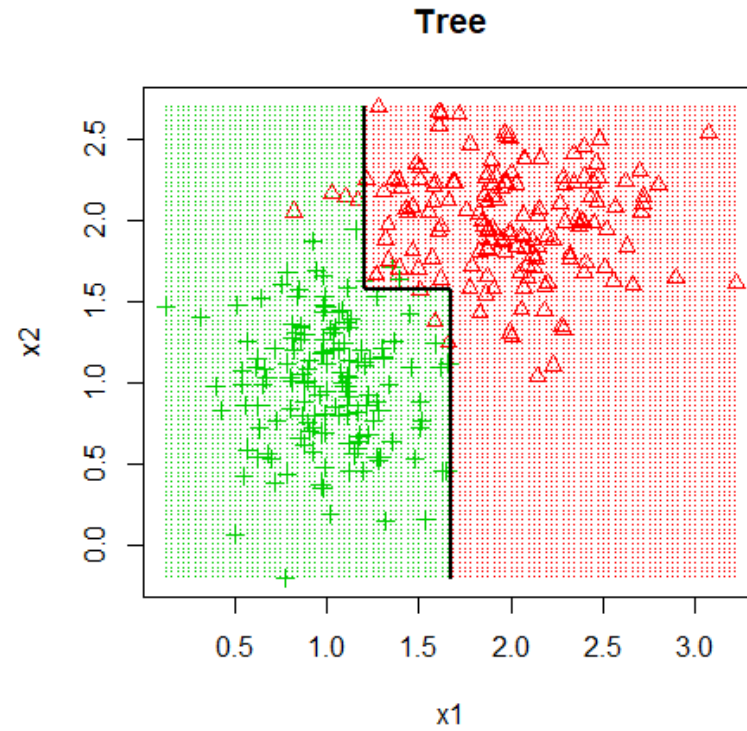


Classificazione con
un albero
decisionale
...anche senza AI!

Classificazione con un
albero decisionale
per l'AI:

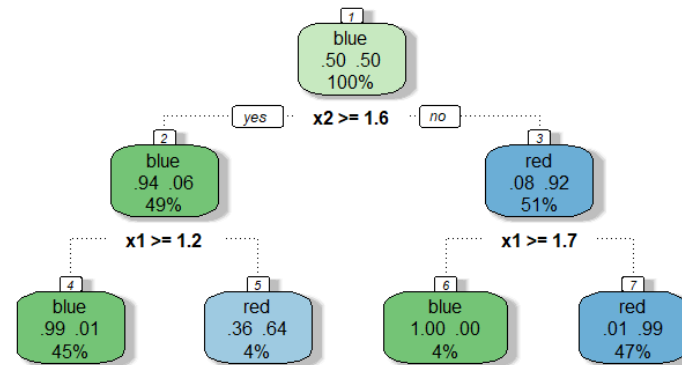
- Ho alcune info per
ciascuna roccia, e il tipo
di roccia corrispondente
- Devo capire le domande
migliori!





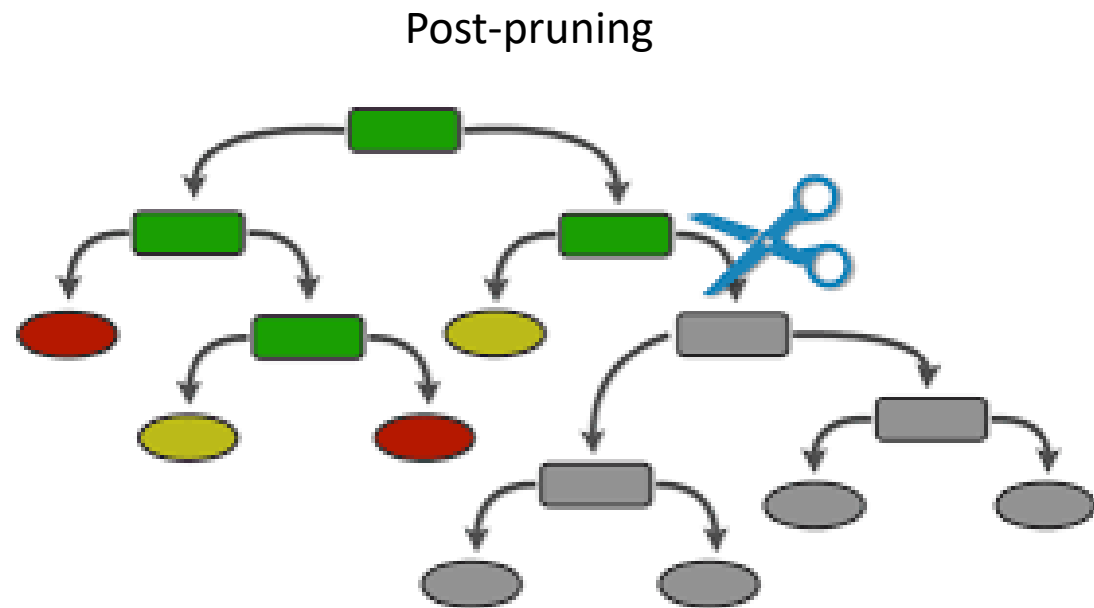
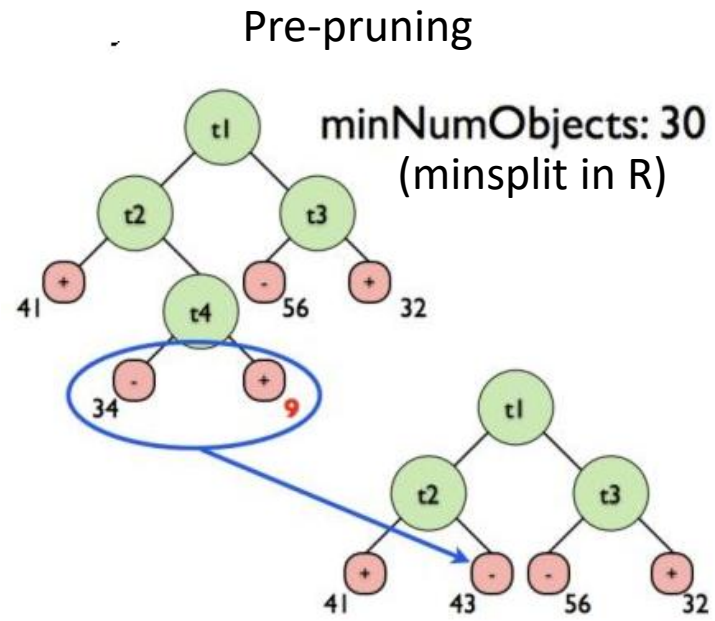
Un albero «buono»:

- ho abbastanza dati delle due classi;
- ho abbastanza nodi nell'albero;
- il *confine di decisione* è ragionevole



Quando il dataset è ampio e completo, gli alberi possono essere molto profondi e fare overfitting. Per evitare l'overfitting:

- ~~pre-pruning~~
Prima di addestrare, imposto dei limiti «dimensionali» alla struttura dell'albero e alla sua generazione
- ~~post-pruning~~
~~Guardando alla struttura addestrata, poto rami poco significativi~~



Over e Underfitting

L'overfitting → si verifica quando un modello ML apprende troppo bene i dettagli presenti nei dati di addestramento, adattandosi in modo eccessivo a questi dati.

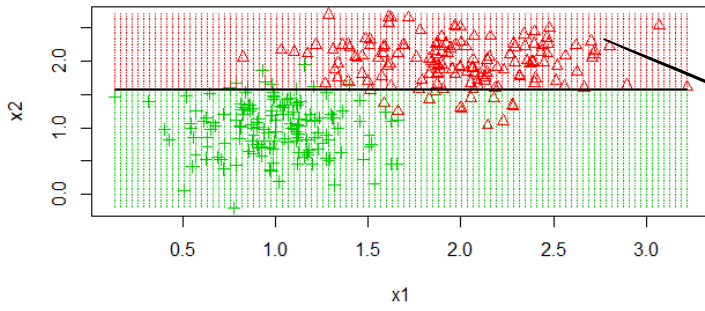
Di conseguenza, il modello ottiene prestazioni eccellenti sui dati di training, ma fallisce nel generalizzare su dati nuovi o di test.

L'underfitting → avviene quando un modello troppo semplice o non riesce a cogliere la complessità e le relazioni sottostanti nei dati.

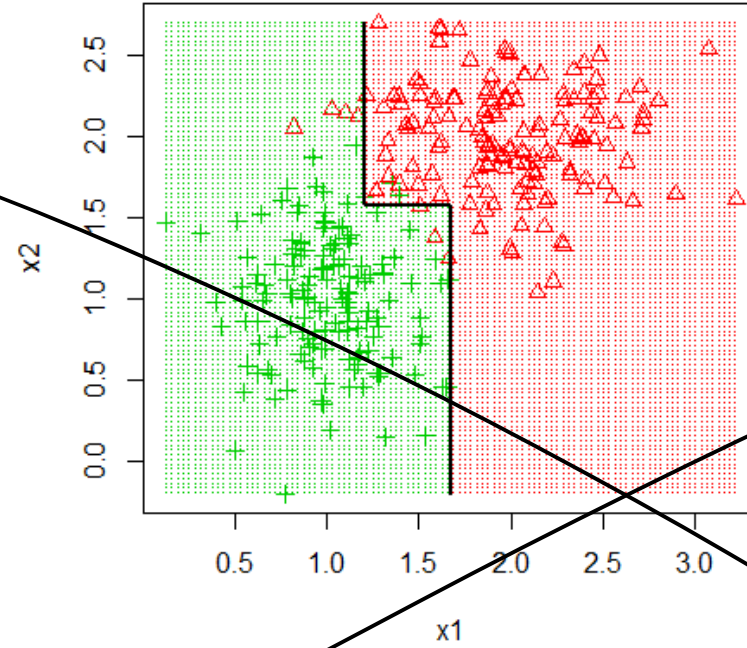
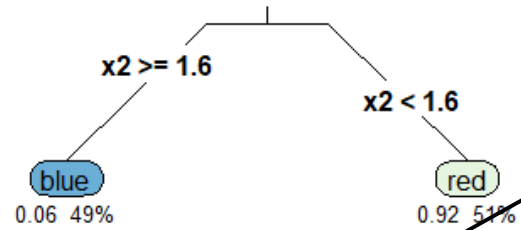
In questo caso, il modello mostra prestazioni scarse sui dati di addestramento sia su quelli di test.

Tree

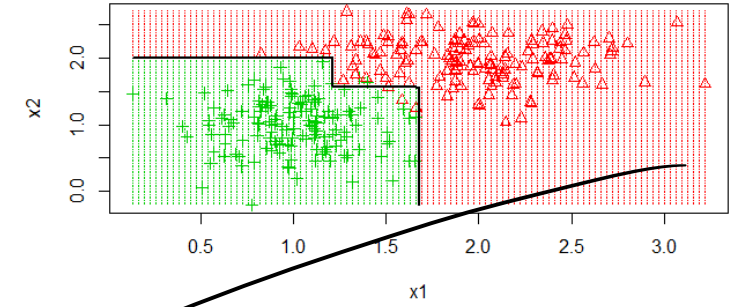
tree1



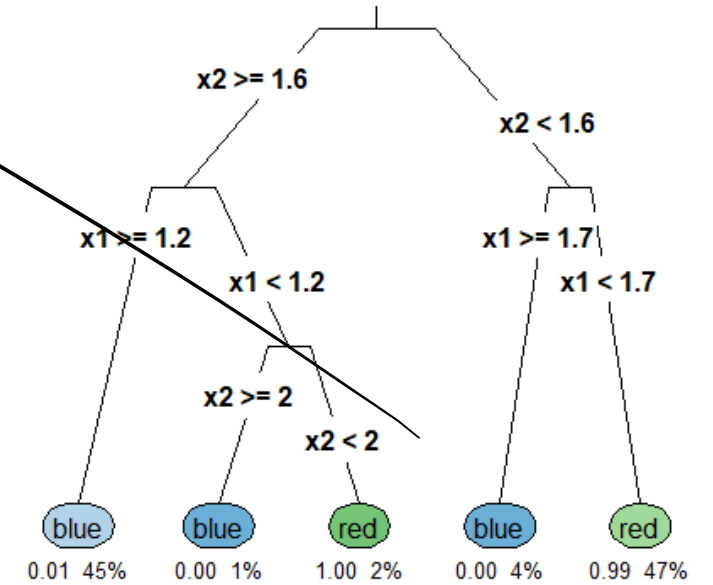
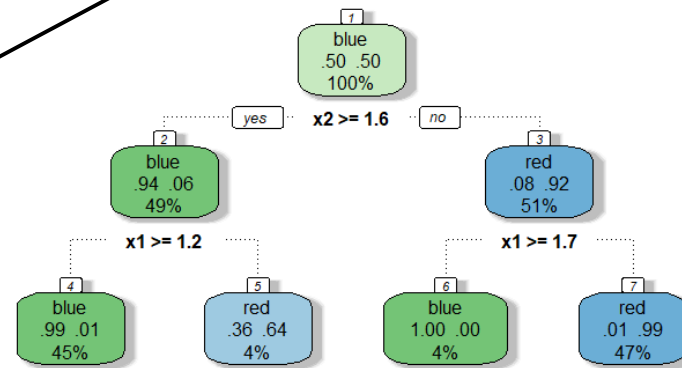
maxdepth = 1



tree3



minbucket = 1, minsplit=2

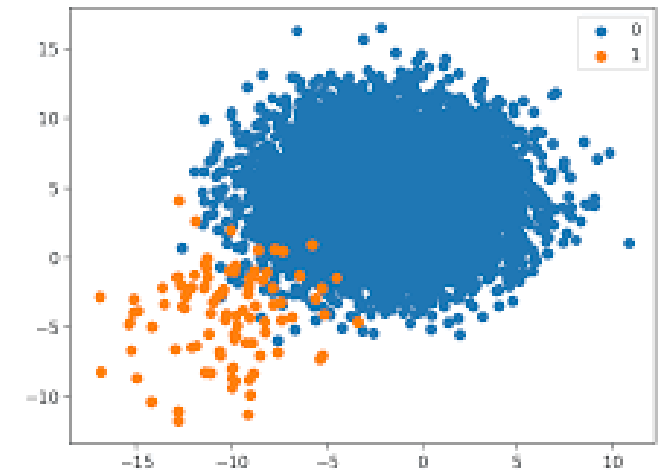


Sbilanciamento delle classi

La **class imbalance** (sbilanciamento delle classi) accade quando una delle due classi è significativamente meno rappresentata rispetto all'altra.

In questi casi, attenzione alle metriche!

- un'alta *accuratezza* può essere ingannevole perché un classificatore può ottenere un valore elevato semplicemente predicando sempre la classe dominante.
- La *sensibilità* è particolarmente importante quando la classe minoritaria è quella di interesse (es. diagnosi di malattia). Un classificatore sbilanciato potrebbe avere una sensibilità molto bassa, indicando che non sta rilevando correttamente la classe meno frequente.



	Actual POSITIVE	Actual NEGATIVE	tot
Predicted POSITIVE	1	0	1
Predicted NEGATIVE	2	997	999
tot	3	997	1000

		True class		Measures
		Positive	Negative	
Predicted class	Positive	True positive <i>TP</i>	False positive <i>FP</i>	Positive predictive value (PPV) $\frac{TP}{TP+FP}$
	Negative	False negative <i>FN</i>	True negative <i>TN</i>	Negative predictive value (NPV) $\frac{TN}{FN+TN}$
Measures		Sensitivity $\frac{TP}{TP+FN}$	Specificity $\frac{TN}{FP+TN}$	Accuracy $\frac{TP+TN}{TP+FP+FN+TN}$

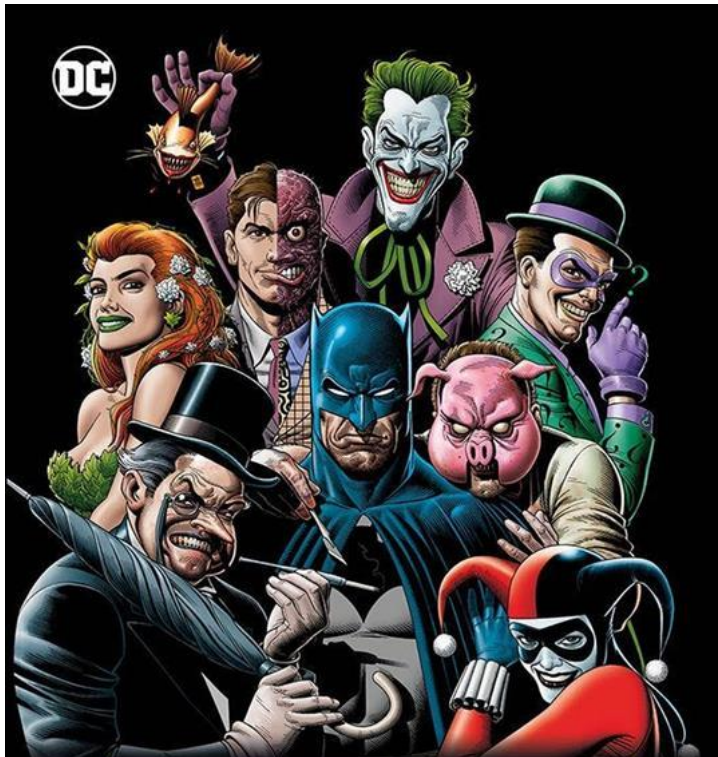
Abbiamo:

- 1 True Positive
- 997 True Negative
- 0 False Positive
- 2 False Negative

Quindi:

- Accuracy = 99,80% (998/1000)
- Misclassification Error = 0,2% (2/1000)
- Sensitivity = 33% (1/3)
- Specificity = 100% (997/997)
- Precision = 100% (1/1)

Esempio di albero decisionale

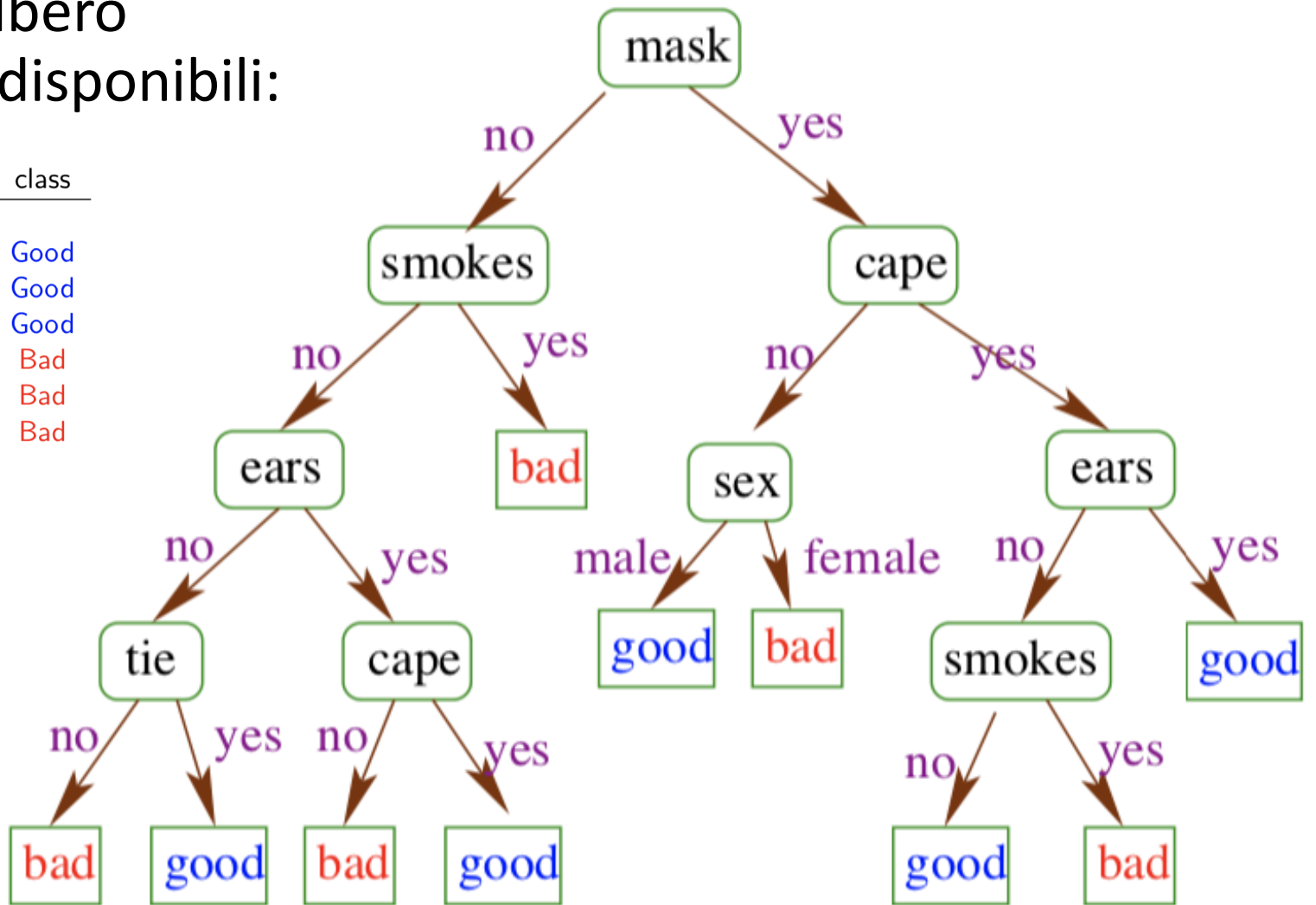


Vogliamo identificare i personaggi come buoni o cattivi dal loro aspetto in base a:

	sex	mask	cape	tie	ears	smokes	class
	training data						
batman	male	yes	yes	no	yes	no	Good
robin	male	yes	yes	no	no	no	Good
alfred	male	no	no	yes	no	no	Good
penguin	male	no	no	yes	no	yes	Bad
catwoman	female	yes	no	no	yes	no	Bad
joker	male	no	no	no	no	no	Bad

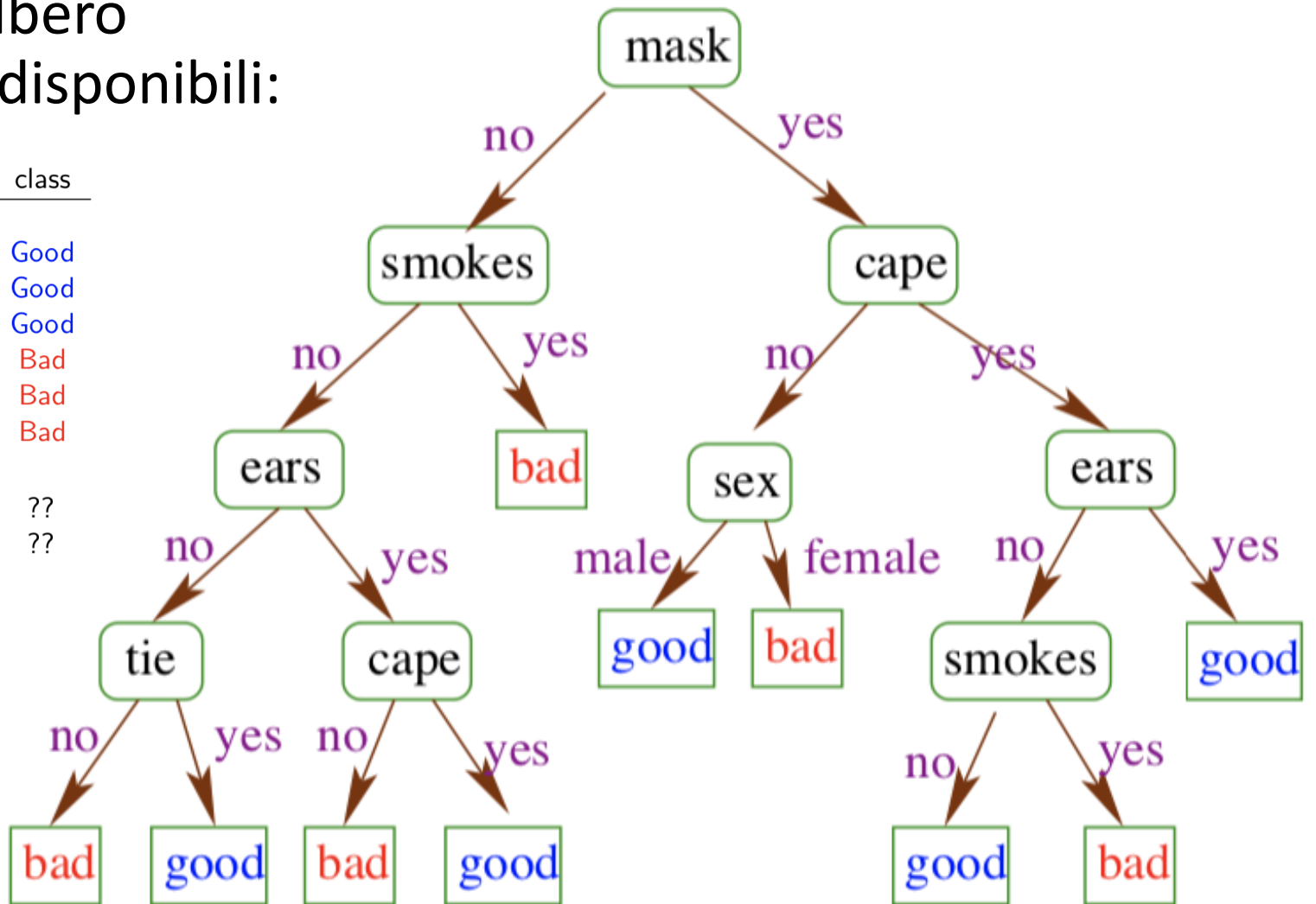
Dopo aver addestrato un albero decisionale sui 6 campioni disponibili:

	sex	mask	cape	tie	ears	smokes	class
training data							
batman	male	yes	yes	no	yes	no	Good
robin	male	yes	yes	no	no	no	Good
alfred	male	no	no	yes	no	no	Good
penguin	male	no	no	yes	no	yes	Bad
catwoman	female	yes	no	no	yes	no	Bad
joker	male	no	no	no	no	no	Bad



Dopo aver addestrato un albero decisionale sui 6 campioni disponibili:

	sex	mask	cape	tie	ears	smokes	class
training data							
batman	male	yes	yes	no	yes	no	Good
robin	male	yes	yes	no	no	no	Good
alfred	male	no	no	yes	no	no	Good
penguin	male	no	no	yes	no	yes	Bad
catwoman	female	yes	no	no	yes	no	Bad
joker	male	no	no	no	no	no	Bad
test data							
batgirl	female	yes	yes	no	yes	no	??
riddler	male	yes	no	no	no	no	??



Test:

- Batgirl:
good (classificato correttamente)
- Riddler:
good (classificato non correttamente)