Introduzione all'Intelligenza Artificiale

Prof. ssa Romina Eramo
Università degli Studi di Teramo
Dipartimento di Scienze della Comunicazione
reramo@unite.it

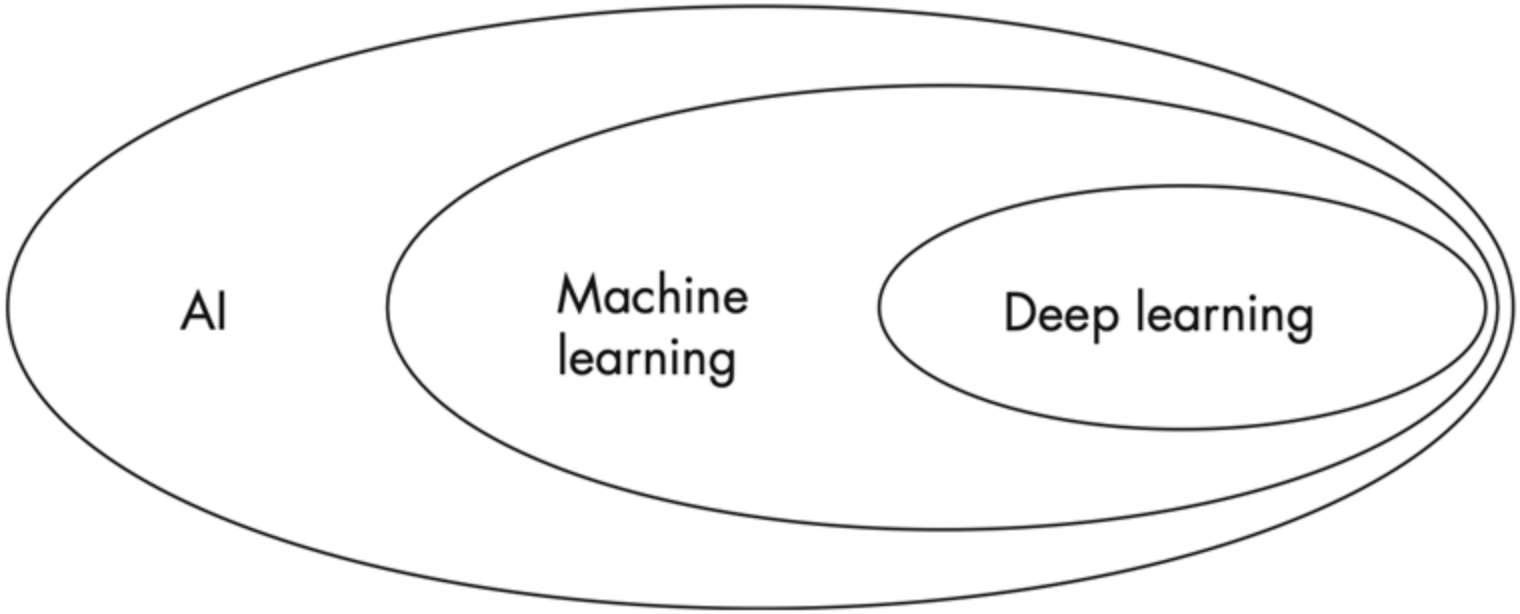
Panoramica sull'IA

L'intelligenza artificiale (IA) tenta di convincere una macchina, in genere un computer, a comportarsi in modi che gli umani giudicano intelligenti. La frase è stata coniata negli anni '50 dal famoso informatico John McCarthy (1927-2011).

Panoramica sull'IA

- I *computer* sono programmati per svolgere un compito specifico, fornendo loro una sequenza di istruzioni, un programma, che incarna un algoritmo, o la ricetta che il programma fa eseguire al computer.
- L'*algoritmo* è un elenco di istruzioni dettagliate, elaborate per svolgere una determinata attività o risolvere un problema specifico.
- Un essere umano concepisce un algoritmo, quindi traduce l'algoritmo in una sequenza di passaggi (un *programma*). La macchina esegue il programma, implementando così l'algoritmo. La macchina non capisce cosa sta facendo; sta semplicemente eseguendo una serie di istruzioni primitive.

Relazione tra Intelligenza Artificiale, Machine Learning e Deep Learning



- Il Deep Learning è un sottocampo del Machine Learning, che è un sottocampo dell'Intelligenza Artificiale
- Questa relazione implica che l'IA implichi concetti che non sono né ML né DL

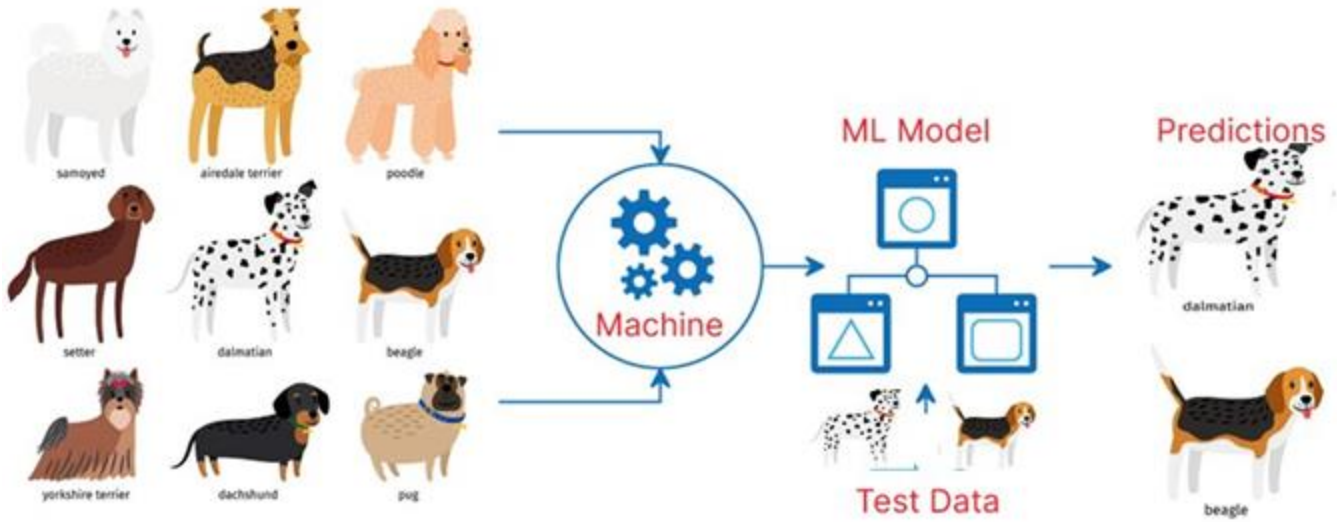
Panoramica sull'IA

Machine Learning (ML) costruisce modelli a partire dai dati.

- In ML, un *modello* è una nozione astratta di qualcosa che accetta input e genera output, dove input e output sono correlati in qualche modo significativo.
- L'obiettivo principale del ML è condizionare un modello utilizzando **dati noti** (*known data*) in modo che il modello produca output significativi quando vengono forniti **dati sconosciuti** (*unknown data*).

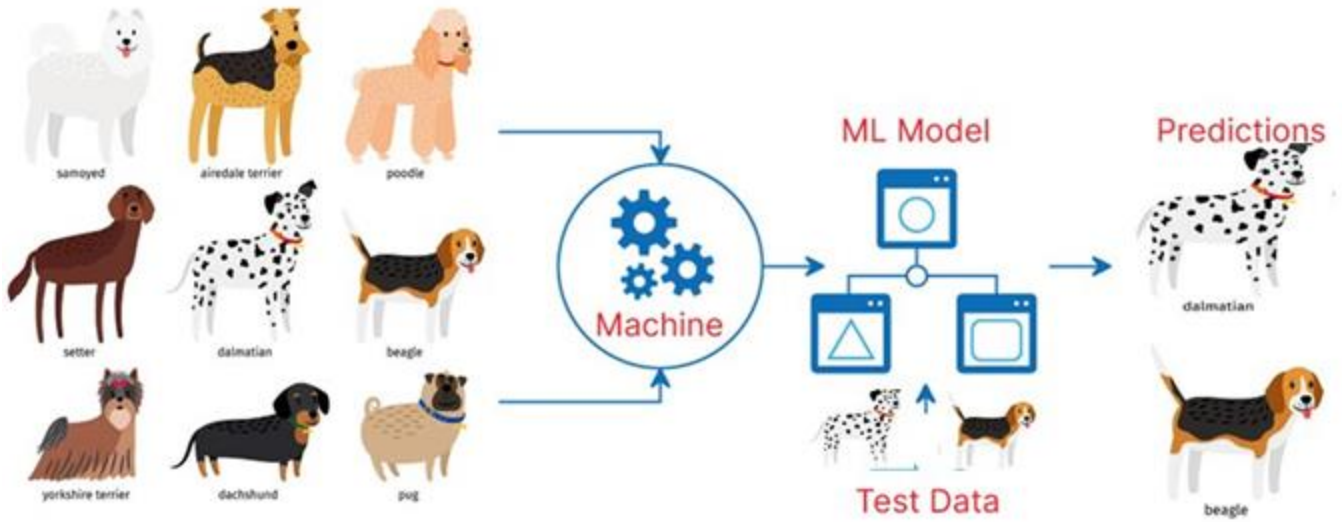
Deep learning si avvale di modelli di grandi dimensioni, che in passato erano troppo grandi per essere utilizzati.

Panoramica sull'IA



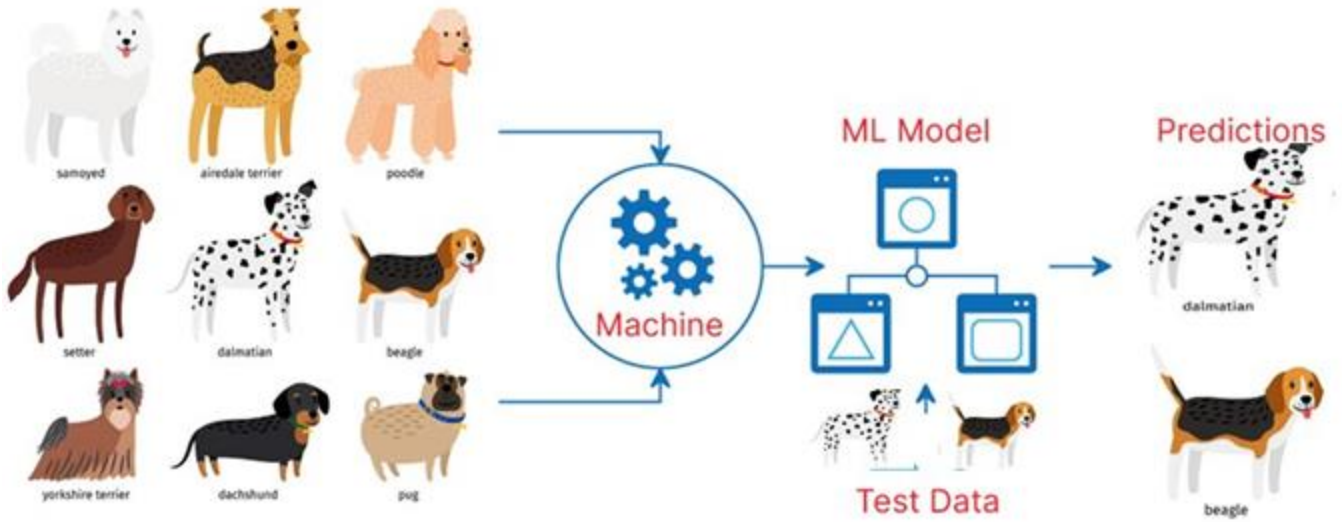
» I dati sono tutto nell'intelligenza artificiale.
  - Il modello è una tabula rasa che i dati devono condizionare per renderlo adatti a un compito.
  - Se i dati sono scadenti, il modello è scadente (da qui dati "buoni" e "cattivi").

Panoramica sull'IA



- Un modello di ML è una scatola nera che
  - accetta un input, solitamente una raccolta di numeri,
  - e produce un output, solitamente un'etichetta come "cane" o "gatto", o un valore continuo come la probabilità di essere un "cane" o il valore di una casa con le caratteristiche fornite al modello (dimensioni, numero di bagni, codice postale e così via).

Panoramica sull'IA



» Il modello ha dei **parametri**, che controllano l'output del modello. Il condizionamento di un modello, noto come addestramento (*training*), cerca di impostare i parametri del modello in modo tale che producano l'output corretto per un dato input.

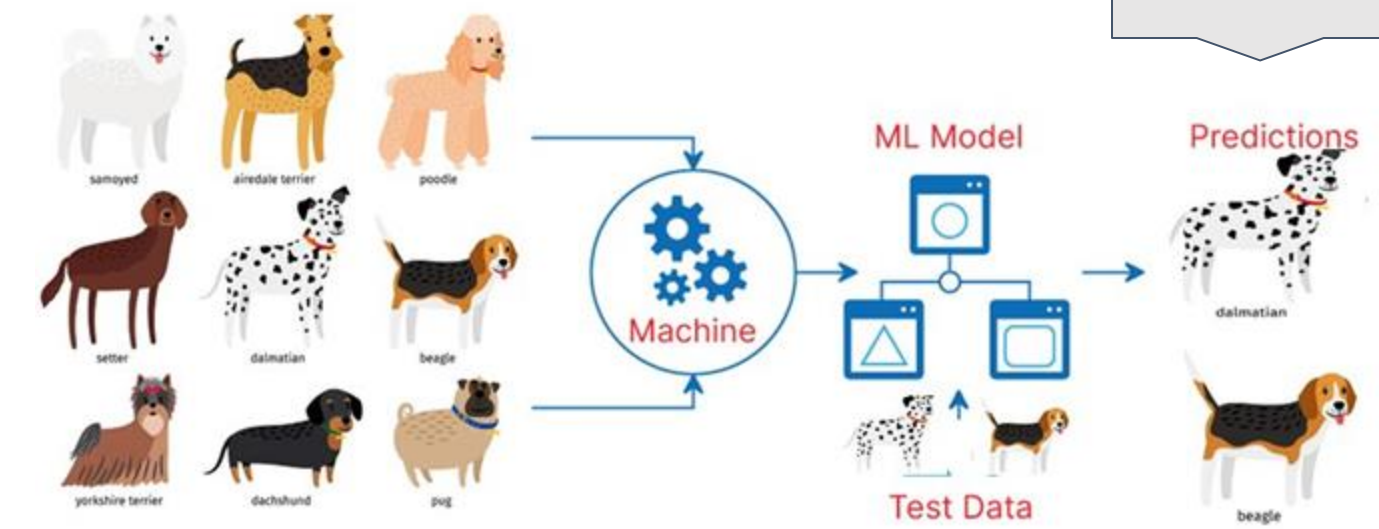
Panoramica sull'IA

1. Raccogli un set di dati di addestramento costituito da una raccolta di input per il modello e dagli output che ci aspettiamo dal modello per quegli input.
2. Seleziona il tipo di modello che vogliamo addestrare.
3. Addestra il modello presentando gli input di addestramento e regolando i parametri del modello quando sbaglia gli output.
4. Ripeti il passaggio 3 finché non siamo soddisfatti delle prestazioni del modello.
5. Utilizza il modello ora addestrato per produrre output per nuovi input sconosciuti.

Supervised learning: Utilizziamo dati etichettati noti per addestrare il modello. "Supervisioniamo" il modello mentre impara a produrre output corretti.

Panoramica sull'IA

SISTEMA



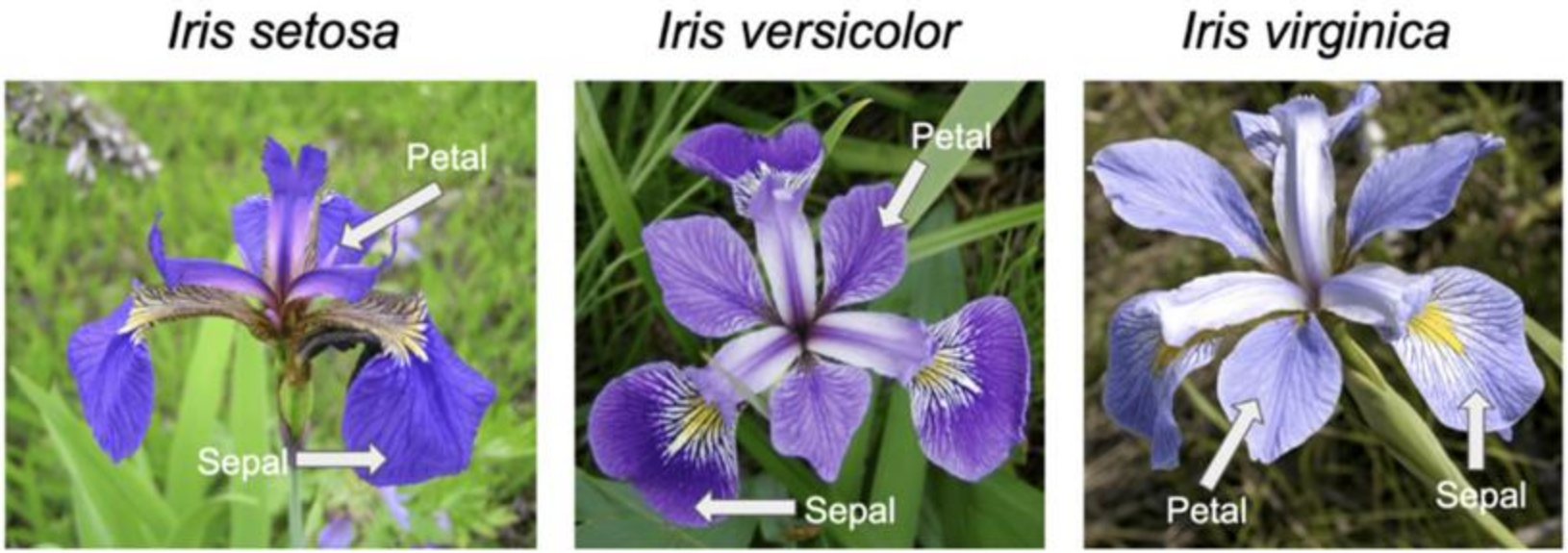
OUTPUT SIGNIFICATIVO

MODELLO ADDESTRATO CON INPUT E OUTPUT CONOSCIUTI

INPUT SCONOSCIUTO

ESEMPIO

Iris Flower Classification



Problema: Il fiore di iris si divide in tre specie primarie (Iris setosa, Iris versicolor e Iris virginica) che differiscono per lunghezza e larghezza del sepalo, lunghezza e larghezza del petalo.

Obiettivo: Sviluppare un modello di ML in grado di apprendere dalle misurazioni dei fiori di iris e automatizzare il processo di classificazione in base alle caratteristiche distinte di ciascuna specie di iris.

https://github.com/Apaulgithub/oibsip_taskno1

ESEMPIO

Iris Flower Classification

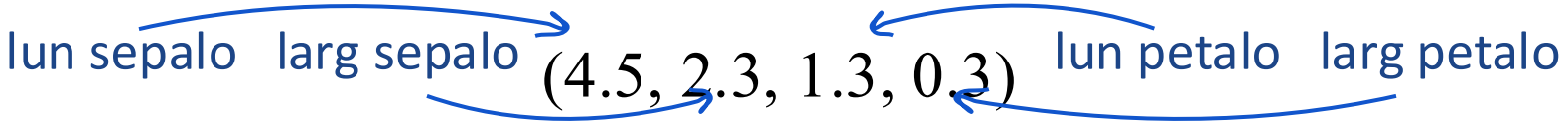
Vettore: è una stringa di valori trattati come un'unica entità.
Es, possiamo raggruppare le misure di un Iris

$$(4.5, 2.3, 1.3, 0.3)$$

ESEMPIO

Iris Flower Classification

Vettore: è una stringa di valori trattati come un'unica entità.
Es, possiamo raggruppare le misure di un Iris



ESEMPIO

Iris Flower Classification

Vettore: è una stringa di valori trattati come un'unica entità. Es, possiamo raggruppare le misure di un Iris

lun sepalo larg sepalo $(4.5, 2.3, 1.3, 0.3)$ lun petalo larg petalo

Matrice: in ML, spesso i set di dati sono rappresentati come matrici, dove le righe sono vettori che rappresentano gli elementi del set di dati, come un fiore di iris. e le colonne sono le misurazioni.

$$\begin{bmatrix} 4.5 & 2.3 & 1.3 & 0.3 \\ 5.6 & 2.9 & 3.6 & 1.3 \\ 5.7 & 4.4 & 1.5 & 0.4 \\ 6.7 & 3.1 & 4.4 & 1.4 \\ 4.6 & 3.1 & 1.5 & 0.2 \end{bmatrix}$$

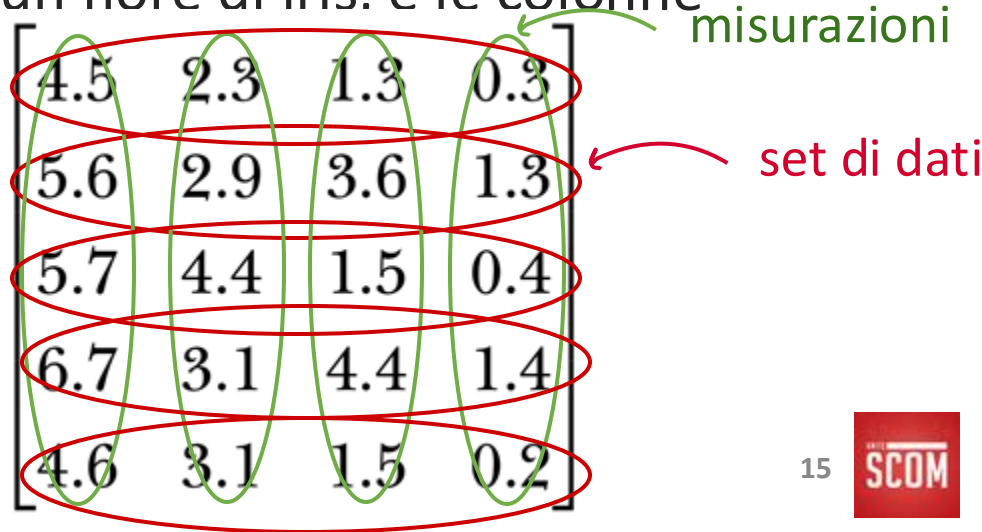
ESEMPIO

Iris Flower Classification

Vettore: è una stringa di valori trattati come un'unica entità.
Es, possiamo raggruppare le misure di un Iris

lun sepalo larg sepalo $(4.5, 2.3, 1.3, 0.3)$ lun petalo larg petalo

Matrice: in ML, spesso i set di dati sono rappresentati come matrici, dove le righe sono vettori che rappresentano gli elementi del set di dati, come un fiore di iris. e le colonne sono le misurazioni.



ESEMPIO

Iris Flower Classification

Vettore: è una stringa di valori trattati come un'unica entità. Es, possiamo raggruppare le misure di un Iris

lun sepalo larg sepalo $(4.5, 2.3, 1.3, 0.3)$ lun petalo larg petalo

Matrice: in ML, spesso i set di dati sono rappresentati come matrici, dove le righe sono vettori che rappresentano gli elementi del set di dati, come un fiore di iris. e le colonne sono le misurazioni.

- Il numero di elementi in un vettore determina la sua dimensionalità

vettore quadridimensionali (quattro misurazioni del fiore)

misurazioni

set di dati

$$\begin{bmatrix} 4.5 & 2.3 & 1.3 & 0.3 \\ 5.6 & 2.9 & 3.6 & 1.3 \\ 5.7 & 4.4 & 1.5 & 0.4 \\ 6.7 & 3.1 & 4.4 & 1.4 \\ 4.6 & 3.1 & 1.5 & 0.2 \end{bmatrix}$$

ESEMPIO

Iris Flower Classification

Semplificazione

Iris setosa

Iris versicolor

Misurazioni:

- larghezza del petalo
- lunghezza del petalo

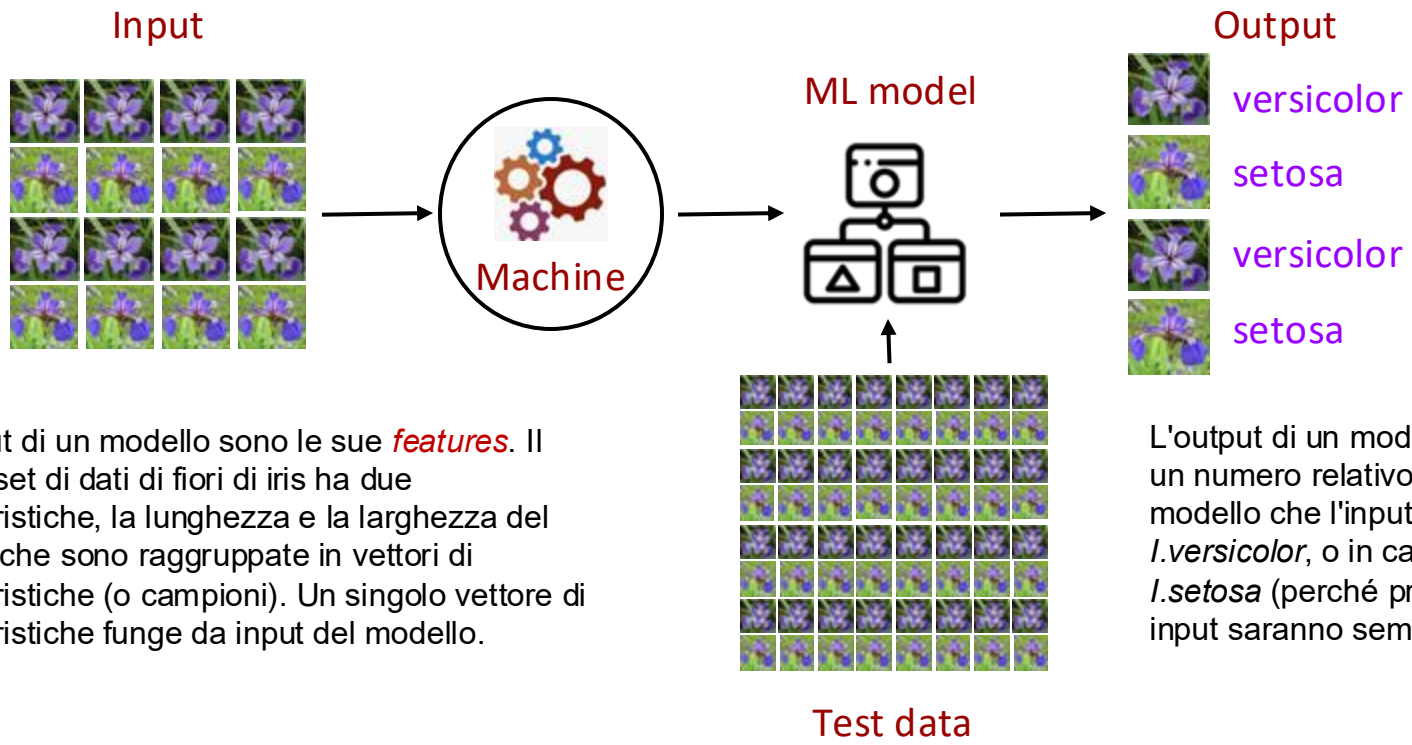
Specie:

- *I.setosa*
- *I.versicolor*

Pertanto, vogliamo che il modello accetti due misurazioni come input e ci fornisca un output che possiamo interpretare come *I.setosa* o *I.versicolor*.

- Modelli binari (come questo) decidono tra due possibili output e sono comunemente usati in IA
- Se il modello decide tra più di due categorie, è un modello multiclasse (multiclass model).

Iris Flower Classification

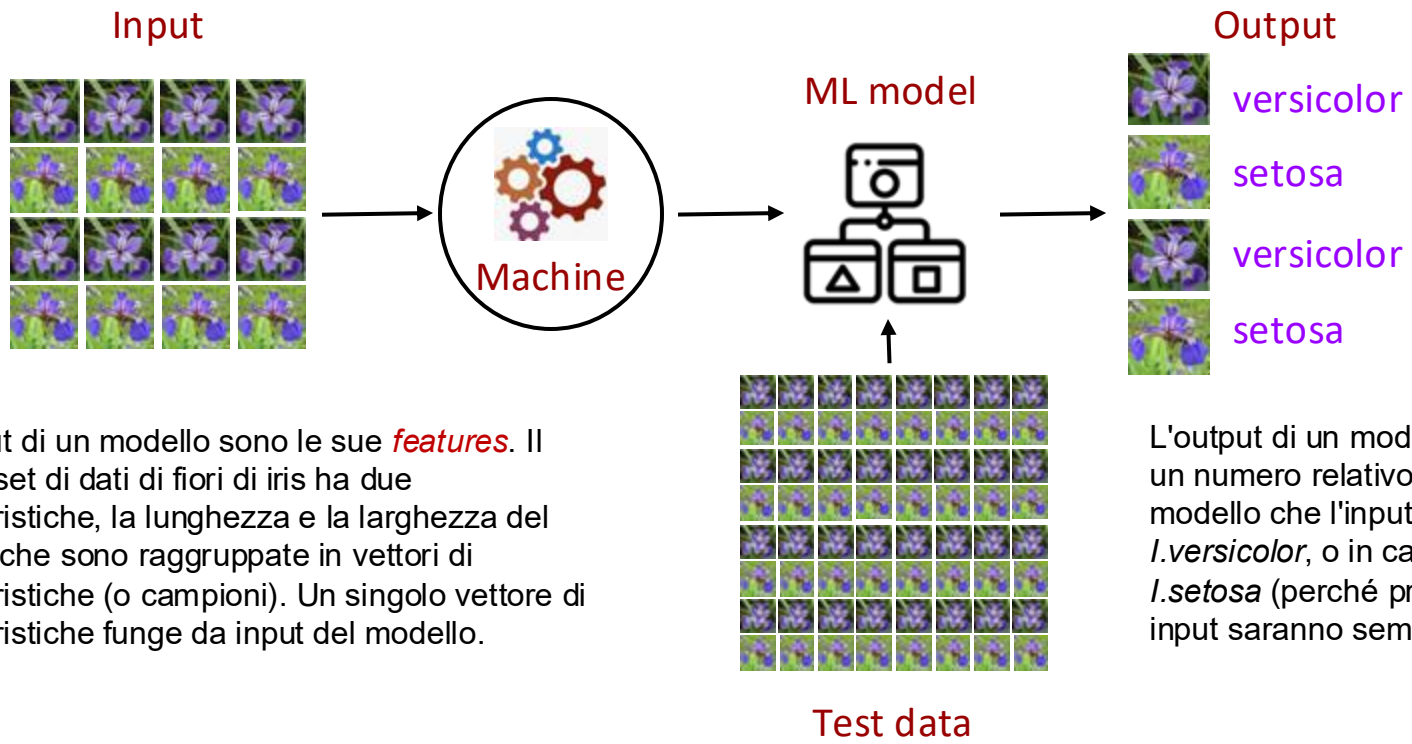


Gli input di un modello sono le sue *features*. Il nostro set di dati di fiori di iris ha due caratteristiche, la lunghezza e la larghezza del petalo, che sono raggruppate in vettori di caratteristiche (o campioni). Un singolo vettore di caratteristiche funge da input del modello.

L'output di un modello binario è in genere un numero relativo alla convinzione del modello che l'input appartenga alla classe *I.versicolor*, o in caso contrario, l'input è *I.setosa* (perché presumiamo che gli input saranno sempre uno o l'altro).

ESEMPIO

Iris Flower Classification



Gli input di un modello sono le sue *features*. Il nostro set di dati di fiori di iris ha due caratteristiche, la lunghezza e la larghezza del petalo, che sono raggruppate in vettori di caratteristiche (o campioni). Un singolo vettore di caratteristiche funge da input del modello.

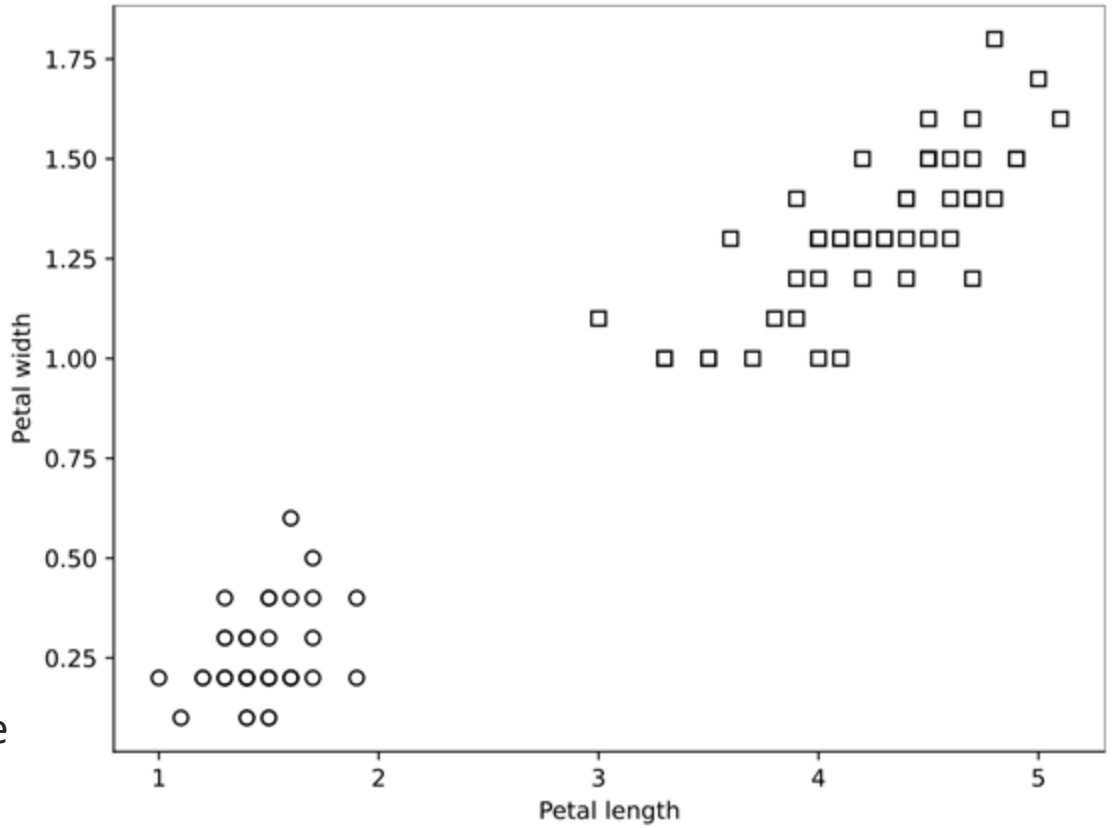
L'output di un modello binario è in genere un numero relativo alla convinzione del modello che l'input appartenga alla classe *I.versicolor*, o in caso contrario, l'input è *I.setosa* (perché presumiamo che gli input saranno sempre uno o l'altro).

Il modo corretto per testare un modello è conservare alcuni dei dati di *training* da utilizzare dopo. Utilizzeremo 80 campioni etichettati per per il training e ne terremo 20 per i test, assicurandoci che sia i set di addestramento che quelli di test contengano un mix approssimativamente uniforme di entrambe le classi (tipi di fiori).

Si vuole un modello che apprenda le caratteristiche generali dei dati di addestramento per generalizzare a nuovi dati. Quindi per il modello, il set di test contiene dati nuovi e invisibili che non ha utilizzato per modificare i suoi parametri. Le prestazioni del modello sul set di test sono un indizio delle sue capacità di generalizzazione

Iris training data (bidimensionali)

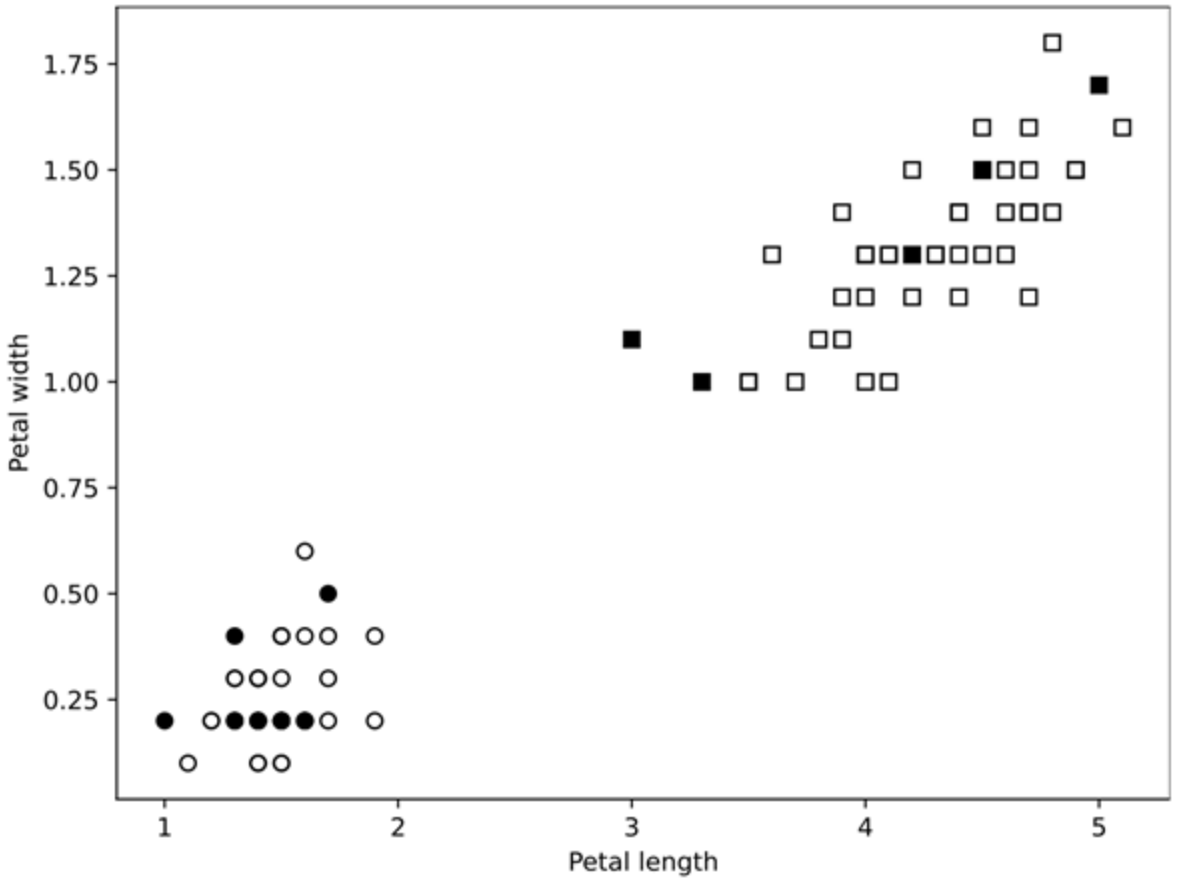
- L'asse x è la lunghezza del petalo e l'asse y è la larghezza del petalo.
- I cerchi corrispondono alle istanze di *I.setosa* e i quadrati a *I.versicolor*.
- Ogni cerchio o quadrato rappresenta un singolo campione di training, la lunghezza e la larghezza del petalo per un fiore specifico.
- Non c'è sovrapposizione tra le classi, sono separate nello spazio delle caratteristiche.



Come classificare? La lunghezza dei petali è inferiore a 2,5 cm? Se la risposta è "sì", allora restituisci classe 0, *I.setosa*; altrimenti, restituisci classe 1, *I.versicolor.*

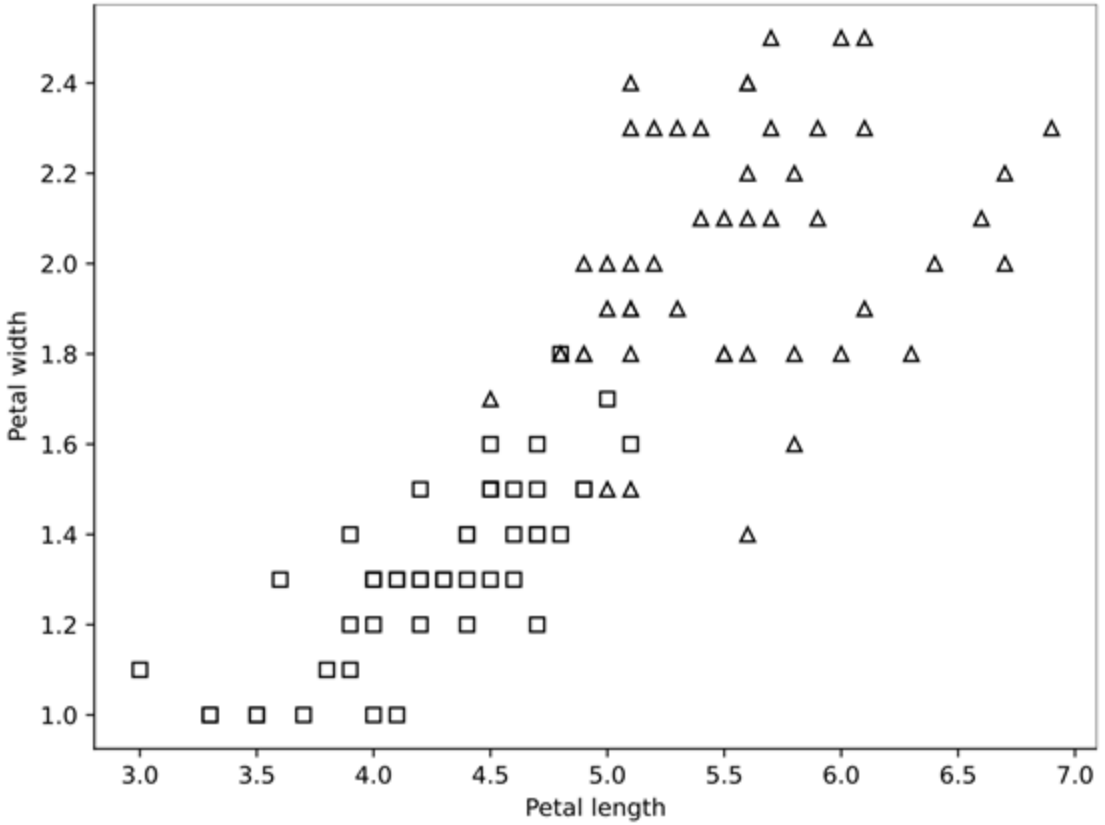
Iris training data

- Si aggiungono i dati di test (cerchi e i quadrati pieni) che non abbiamo utilizzato per creare il classificatore a domanda singola.
- Nessuno dei dati di test viola la nostra regola; otteniamo comunque etichette di classe corrette chiedendo se la lunghezza del petalo è inferiore a 2,5 cm. Pertanto, il nostro modello è perfetto; non commette errori!



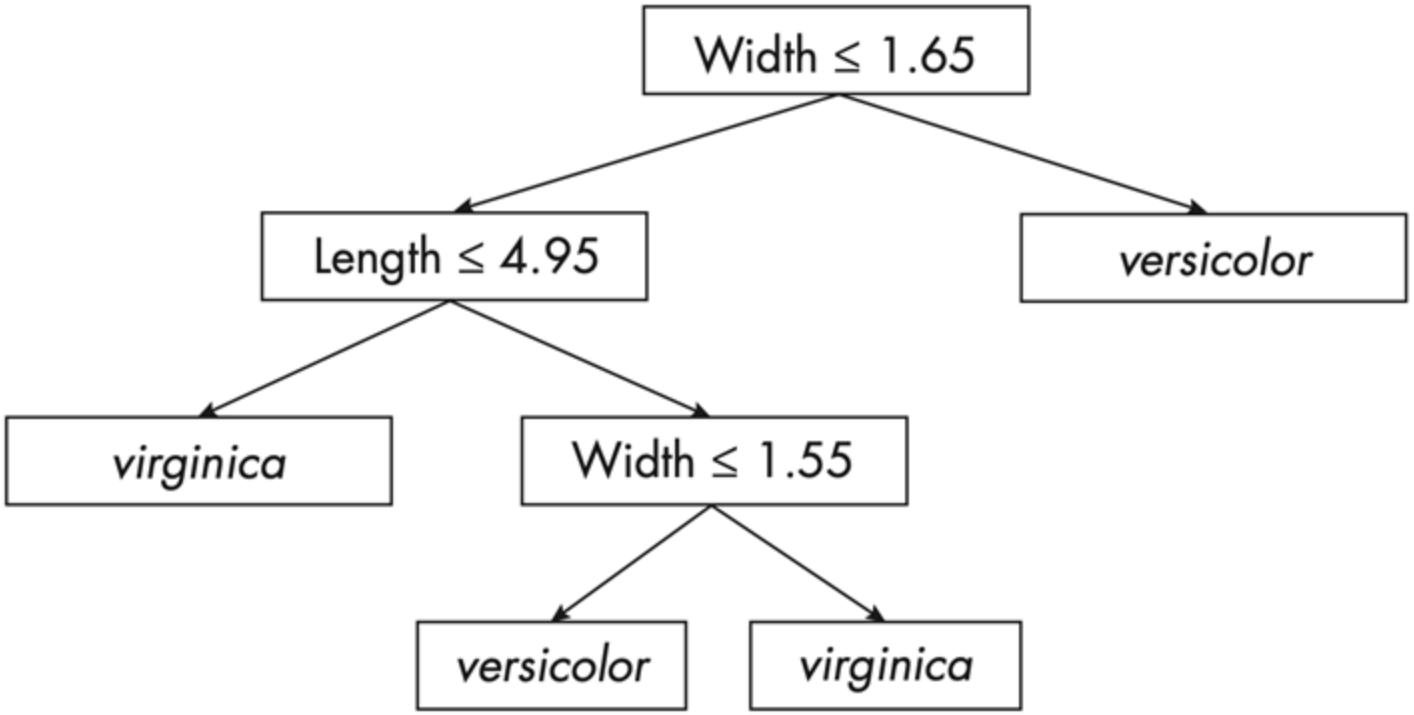
Iris training data

- Ripetiamo l'esercizio, sostituendo *I.setosa* con la specie di iris rimanente, *I.virginica* (triangoli). L'evidente divario tra le classi è scomparso e si sovrappongono.
- Come prima, c'erano 80 campioni per l'addestramento e 20 trattenuti per i test. Questa volta, il modello non era perfetto. Ha etichettato correttamente 18 dei 20 campioni, per una precisione di 9 su 10, o il 90 percento.
- Ciò significa approssimativamente che quando questo modello assegna un fiore a una particolare classe, c'è una probabilità del 90 percento che sia corretto.



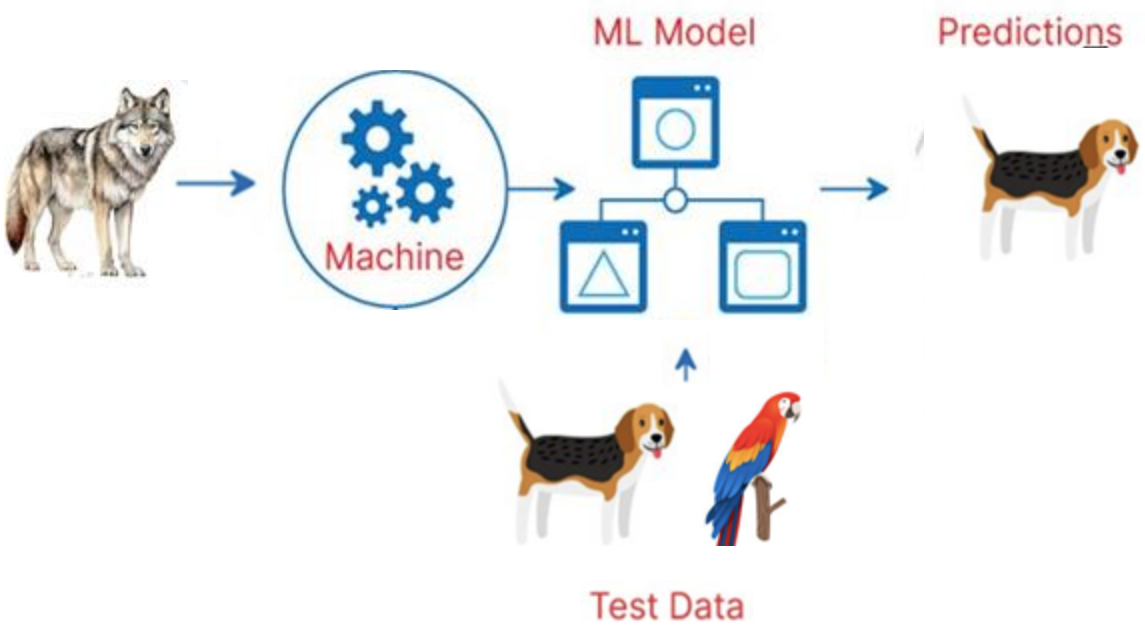
I modelli di apprendimento automatico non sono sempre perfetti; (abbastanza frequentemente) commettono errori.

Iris training data (Albero decisionale)



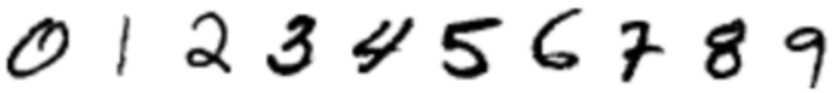
Gamma di input

» Affinché un modello funzioni bene "in natura", ovvero quando viene utilizzato nel mondo reale, i dati utilizzati per addestrare il modello devono coprire l'intera gamma di input che il modello potrebbe incontrare



MNIST dataset

- Dataset di decine di migliaia di piccole immagini contenenti cifre scritte a mano, da 0 a 9.
  - MNIST (Modified NIST) perché derivato da un set di dati costruito dal National Institute of Standards and Technology (NIST)



- **Obiettivo**: Costruire una rete neurale che impari a identificare le cifre 0, 1, 3 e 9.
- Possiamo addestrare Reti Neurali senza sapere come funzionano grazie a potenti toolkit open source (es., scikit-learn)

ESEMPIO

MNIST (Matrice di confusione)

- Le righe della matrice rappresentano le etichette reali per i campioni forniti al modello.
- Le colonne sono le risposte del modello
- I valori nella tabella sono conteggi, il numero di volte in cui si è verificata ogni possibile combinazione di classe di input ed etichetta assegnata dal modello.
- Nel complesso, il modello delle cifre è accurato al 99%!
- **Ma cosa succede se l'input è diverso da 0, 1, 3 o 9?**

Confusion Matrix

	0	**1**	**3**	**9**
0	978	0	1	1
1	2	1,128	3	2
3	5	0	997	8
9	5	1	8	995

MNIST (Cambio di input)

» Dando al modello 982 quattro, si ottiene:

0	1	3	9
48	9	8	917

» E dando sette?

0	1	3	9
19	20	227	762

» Il modello non è stato addestrato a riconoscere i quattro o i sette, quindi ha fatto la cosa migliore e li ha inseriti in una categoria vicina.

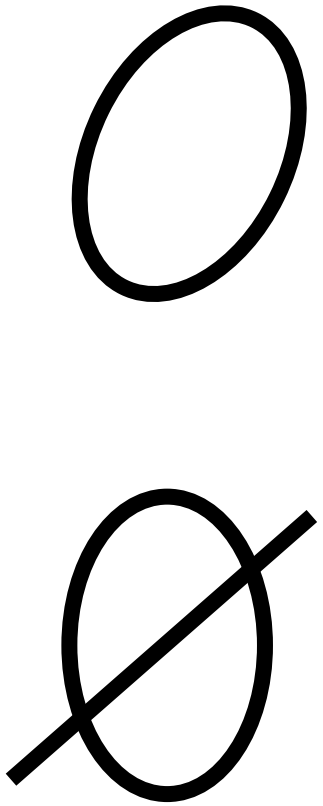
Interpolazione ed estrapolazione

- ***L'interpolazione*** **approssima entro l'intervallo di dati noti e** ***l'estrapolazione*** **va oltre i dati noti.**
- Quando da una legge sperimentale si ricava un valore non misurato, si esegue una:
  - Interpolazione se il valore ricade nell'intervallo dei valori misurati
  - Estrapolazione se il valore ricade al di fuori dell'intervallo dei valori misurati

ESEMPIO

Interpolazione ed estrapolazione

- L'**interpolazione** potrebbe riferirsi all'incontro con uno **zero inclinato** in natura quando nessuno degli zeri nel set di addestramento era particolarmente inclinato
  - il modello deve interpolare, in un certo senso, per rispondere correttamente
- L'**estrapolazione** è più simile alla classificazione di uno **zero con una barra che lo attraversa**
  - che è qualcosa di invisibile durante il periodo di addestramento



ESEMPIO

Interpolazione ed estrapolazione

- Lo stesso processo di generazione dei dati ha creato entrambi.
  - Il modello stava, in un certo senso, interpolando nei primi casi; ma quando abbiamo forzato il modello a prendere decisioni su 4 e 7, stavamo estrapolando facendo in modo che il modello prendesse decisioni su dati che non aveva mai visto durante l'addestramento.
- Va detto che: interpolazione buona, estrapolazione cattiva. Set di dati cattivi portano a modelli cattivi; set di dati buoni portano a modelli buoni, che si comportano male quando sono costretti a estrapolare. E, per buona misura: tutti i modelli sono sbagliati, ma alcuni sono utili.

ESEMPIO

Falsi positivi e falsi negativi

» Nuovo modello: “La cifra di input è un nove?”
  - Il modello è la stessa rete neurale usata in precedenza. Se addestrato su un set di dati in cui ogni immagine è un nove (ovvero, nei dati di addestramento non ci sono quattro o sette), allora il modello è accurato al 99%. come mostra la matrice di confusione:

	Not 9	9
Not 9	9,754	23
9	38	1,362

falsi positivi

falsi negativi

  - In questo caso, la matrice di confusione è piccola perché il modello ha solo due classi: nove o non nove.
  - In altre parole, questo è un modello binario.

ESEMPIO

Falsi positivi e falsi negativi

» Diamo in input gli sconosciuti quattro e sette

	Not 9	9
Not 9	5,014	9,103

Il modello ha etichettato 2/3 come un nove

» Aggiungiamo quattro e sette al set di addestramento (3 percento dei dati di addestramento quattro e sette)

	Not 9	9
Not 9	9,385	3,321

Il modello ha etichettato solo 1/4 come un nove

» Se aumentiamo la proporzione al 18%, il modello classifica erroneamente in meno dell'1% dei casi.

Falsi positivi e falsi negativi

» Poiché i modelli imparano dai dati, **dobbiamo usare set di dati che siano il più completi possibile in modo che i nostri modelli interpolino e non estrapolino.**

Nota:

» Recenti ricerche dimostrano che i moderni modelli di Deep Learning **sono quasi sempre estrapolatori**, ma più gli input sono simili ai dati su cui è stato addestrato il modello, migliori sono le prestazioni!

Explainable AI

- Chiunque cerchi di comprendere, o di lavorare con, l'IA deve prendere a cuore gli avvertimenti sulla **qualità dei dati utilizzati per addestrare** i modelli di IA.
- *Nature Machine Intelligence da Michael Roberts et al., "Common Pitfalls and Recommendations for Using Machine Learning to Detect and Prognosticate for COVID-19 Using Chest Radiographs and TC Scans", 2021.*
  - Gli autori hanno valutato le prestazioni dei modelli di ML progettati per rilevare il COVID-19 nelle radiografie del torace e nelle TC, riducendo il pool iniziale di candidati di oltre 2.000 studi (modelli) a 62 per test rigorosi. Alla fine, gli autori hanno dichiarato che nessuno dei modelli era adatto all'uso clinico a causa di difetti nella costruzione, distorsioni nei set di dati o di entrambi.
- Risultati come questi hanno portato alla creazione di ***explainable AI*** (**IA spiegabile**), un sottocampo che cerca di dare ai modelli la capacità di spiegare se stessi.
  - *Guarda i tuoi dati e cerca di capire, per quanto umanamente possibile, **cosa sta facendo il tuo modello e perché**.*

Tipi di apprendimento automatico

Tipo di apprendimento	Scenario di utilizzo	Esempi di algoritmo
Supervised learning ("Riconosco connessioni che legano i dati in input a una colonna obiettivo")	Regressione ("L'obiettivo è un numero") Classificazione ("L'obiettivo è una categoria")	Linear/logistic regression Decision tree Random forest Neural network
Unsupervised learning ("Riconosco strutture nei dati")	Clustering ("Voglio raggruppare in gruppi omogenei") Riduzione della dimensionalità ("Voglio ridurre il numero di colonne")	K-means Hierarchical clustering Latent Dirichlet Allocation (LDA) Principal Component Analysis (PCA) Singular Value Decomposition (SVD)
Reinforcement learning ("Faccio tentativi e imparo dagli errori")	Agente autonomo ("Voglio imparare a rispondere in base a come varia l'ambiente")	Q-Learning Monte Carlo

Supervised Learning

- La macchina impara da **esempi passati** in cui si conosce già il risultato (target).
- **Obiettivo -** Trovare la relazione tra:
  - **Input** (caratteristiche)
  - **Output** (target / colonna obiettivo)
- **Esempi d'uso**
  - Predire il **prezzo** di una casa
  - Classificare un email come **spam/non spam**
  - Stimare il rischio di credito
- **Tipi**
  - **Regressione** → target numerico
  - **Classificazione** → target categoriale

Condizione chiave del supervised learning

- Serve un **maestro** (dataset etichettato):
  - Ogni esempio deve avere il **risultato corretto**
  - La macchina impara quali input producono quali output
  - È l'analogia dello **studente che apprende dagli esempi del maestro**

Unsupervised Learning

- La macchina **non ha un target**.
- Deve scoprire **strutture nascoste** nei dati.
- **Obiettivo**
  - Trovare pattern, gruppi, similarità, ricorrenze.
- **Esempi d'uso**
  - Raggruppare clienti con comportamenti simili (**clustering**)
  - Ridurre il numero di colonne (**dimensionality reduction**)
  - Trovare segmenti di mercato
- **Nessun "maestro"**
  - La macchina imparare **esplorando i dati da sola**.
- Il caso più comune di apprendimento non supervisionato è quello del ***clustering***

Esempio di clustering

- Obiettivo: creare gruppi omogenei (cluster).
- Esempio immobiliare:
  - Raggruppare appartamenti simili per aiutarne la valutazione
  - La macchina trova da sola pattern in base a caratteristiche come:
    - numero di stanze
    - zona
    - prezzo
    - superficie

Reinforcement Learning

- La macchina **impara dai propri errori** interagendo con un ambiente.
- **Come funziona?**
  - Tenta un'azione
  - Riceve **ricompensa** o **penalità**
  - Impara a scegliere le mosse migliori nel tempo
- **Analogia:**
  - Un bambino che impara a camminare provando e cadendo.
- **Esempi d'uso**
  - Robotica
  - Giochi (scacchi, Go)
  - Agenti autonomi di decisione

Confronto tra i tre approcci

Tipo	Cosa ha?	Obiettivo	Esempi
Supervised	Dati + Target	Predire un risultato	Regressione, Classificazione
Unsupervised	Solo dati (no target)	Trovare strutture	Clustering, PCA
Reinforcement	Ambiente + Ricompensa	Imparare strategie	Q-learning, decision making

Quando usare quale tipo?

- Hai risultati storici noti? → **Supervised**
- Vuoi trovare gruppi o pattern nascosti? → **Unsupervised**
- La macchina deve interagire col mondo e imparare dalle ricompense? → **Reinforcement**

Underfitting e Overfitting

» Due problemi fondamentali nel machine learning che influenzano la **capacità di generalizzazione** del modello.

» **Underfitting** → modello troppo semplice → non apprende abbastanza.

» **Overfitting** → modello troppo complesso → apprende troppo, anche il rumore.

Problema	Modello	Effetto
Underfitting	Troppo semplice	Bassa accuratezza ovunque
Overfitting	Troppo complesso	Memorizza, non generalizza
Modello ideale	Complessità bilanciata	Generalizza bene

Underfitting

- Il modello è **troppo semplice** per apprendere la relazione tra input e output.
- **Sintomi**
  - Altezza errori su **training** e **test**
  - Modello poco accurato
  - Pattern importanti ignorati
- **Esempio**
  - Una retta usata per dati che seguono una curva complessa.

👉 *Il modello non impara abbastanza.*

Underfitting

- **Cause principali dell'Underfitting**
  - Modello troppo rigido (es. regressione lineare su dati non lineari)
  - Troppi pochi predittori
  - Pochi epoche/iterazioni di training
  - Dati insufficienti o troppo rumorosi
  - Eccessiva regolarizzazione
- **Come prevenire l'Underfitting**
  - Usare modelli più complessi
  - Aggiungere nuove feature
  - Aumentare la durata del training
  - Ridurre la regolarizzazione
  - Applicare feature engineering

Overfitting

- Il modello è **troppo complesso** e impara **anche il rumore** invece della relazione reale.
- **Sintomi**
  - Errore **bassissimo** sul training set
  - Errore **alto** sul test set
  - Scarsa generalizzazione
  - Modello “memorizza” i dati

👉 *Il modello impara troppo (anche ciò che non serve).*

Overfitting

- **Cause principali dell'Overfitting**
  - Modello troppo complesso (es. reti profonde, alberi non potati)
  - Troppi predittori inutili
  - Dataset piccolo rispetto alla complessità
  - Troppo training (over-training)
  - Rumore presente nei dati
- **Come prevenire l'Overfitting**
  - Cross-validation
  - Ridurre la complessità del modello
  - Early stopping
  - Regularizzazione (L1, L2, dropout)
  - Aumentare i dati (data augmentation)
  - Rimozione delle feature irrilevanti

Partitioning

- Il **Partitioning** è il processo con cui si **suddivide un dataset** in più parti distinte, normalmente:
  - **Training set** → usato per addestrare il modello
  - **Test set** → usato per valutarne l'accuratezza
  - (opzionale) **Validation set** → per ottimizzare i parametri
- **Obiettivo**
  - Garantire che il modello venga valutato in modo **corretto**, **onesto** e **generalizzabile**.
- **Fondamentale perchè:**
  - Evita l'**overfitting**
  - Simula dati "mai visti" dal modello
  - Permette di valutare la **capacità di generalizzazione**
  - Rende gli esperimenti **riproducibili**
  - Migliora la credibilità scientifica dei risultati

Esempio classico di partitioning

- Suddividere i dati immobiliari in:
  - **80% training** → per addestrare il modello
  - **20% test** → per verificarne l'accuratezza

Valutazione del modello

- **Perché è importante?**
  - Capire **quanto il modello descrive la realtà**
  - Evitare l'**overfitting** e valutare che il modello funzioni su dati mai visti
  - Scegliere il modello migliore tra più alternative
  - Garantire che le predizioni siano **affidabili** per il business

👉 *Non esiste un'unica metrica perfetta: la scelta dipende dal tipo di problema.*

Valutazione nei modelli supervisionati

- **Dipende dalla natura del problema:**
  - **Regressione** → si valuta l'errore sui numeri
  - **Classificazione** → si valutano le etichette previste vs reali

Valutazione nei modelli di regressione

- **Strumenti essenziali**
  - RMSE (errore medio quadratico)
  - R^2 (varianza spiegata)

RMSE (Root Mean Squared Error)

- Una misura dell'**errore medio** tra valori osservati e valori predetti.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (r_i - p_i)^2}{N}}$$

r_i è il valore osservato per l'i-esima riga
p_i è il corrispondente valore predetto
N è il numero delle righe

- **Come interpretarlo**
  - Valori **più bassi** → modello migliore
  - Sensibile agli **errori grandi**
  - Utile quando si predicono **numeri continui** (prezzi, temperature, quantità…)
- **Pro**
  - Intuitivo
  - Stessa unità di misura del target
- **Contro**
  - Dipende dalla scala del fenomeno
  - Non confrontabile tra problemi diversi

R^2 (Coefficiente di determinazione)

» Misura **quanta parte della variabilità del target** è spiegata dal modello.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (r_i - p_i)^2}{\sum_{i=1}^{N} (r_i - \overline{r})^2}$$

r_i è il valore osservato per l'i-esima riga
p_i è il corrispondente valore predetto
$\overline{r}$ è la media aritmetica dei valori osservati
N è il numero delle righe

R^2 (Coefficiente di determinazione)

- **Interpretazione**
  - **$0 \leq R^2 \leq 1$**
  - **0** → il modello non spiega nulla
  - **1** → il modello spiega tutto (attenzione all'overfitting!)
  - Esempi tipici:
  - 0.65 → il modello spiega il 65% della variabilità
  - 0.90 → modello molto buono
  - 0.25 → può essere comunque utile in fenomeni molto complessi o rumorosi
- **Pro**
  - Indipendente dalla scala
  - Facilmente confrontabile tra modelli sullo stesso problema
- **Contro**
  - Può essere "ingannevole" con modelli troppo complessi
  - Non confrontabile tra problemi diversi

Valutazione nei modelli di classificazione

- Capire *quanto* un classificatore riconosce correttamente le classi.
- **Strumenti principali**
  - Confusion Matrix
  - Accuracy
  - Precision
  - Sensitivity / Recall
- **Perché non basta l'accuracy?**
  - Perché **non tutti gli errori hanno lo stesso peso** (es. falsi negativi in un test medico).

Confusion Matrix

» La confusion matrix mostra:

Reale \ Predetto	Positivo	Negativo
Positivo	TP (true positive)	FN (false negative)
Negativo	FP (false positive)	TN (true negative)

» **A cosa serve?**

- A capire *quali tipi di errori* il modello commette
- A calcolare tutte le metriche principali della classificazione

Errori di classificazione

❌ **False Positive (FP)**

» Predetto “positivo” ma in realtà è negativo

→ Esempio: TAC non necessaria

❌ **False Negative (FN)**

» Predetto “negativo” ma in realtà è positivo

→ Esempio: tumore *non* diagnosticato

→ Errore più pericoloso nella medicina

Accuracy

» Percentuale di **previsioni corrette** del modello.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

» **Interpretazione**
- 1.0 (100%) → modello perfetto
- 0.69 (69%) → 11 previsioni corrette su 16, ecc.

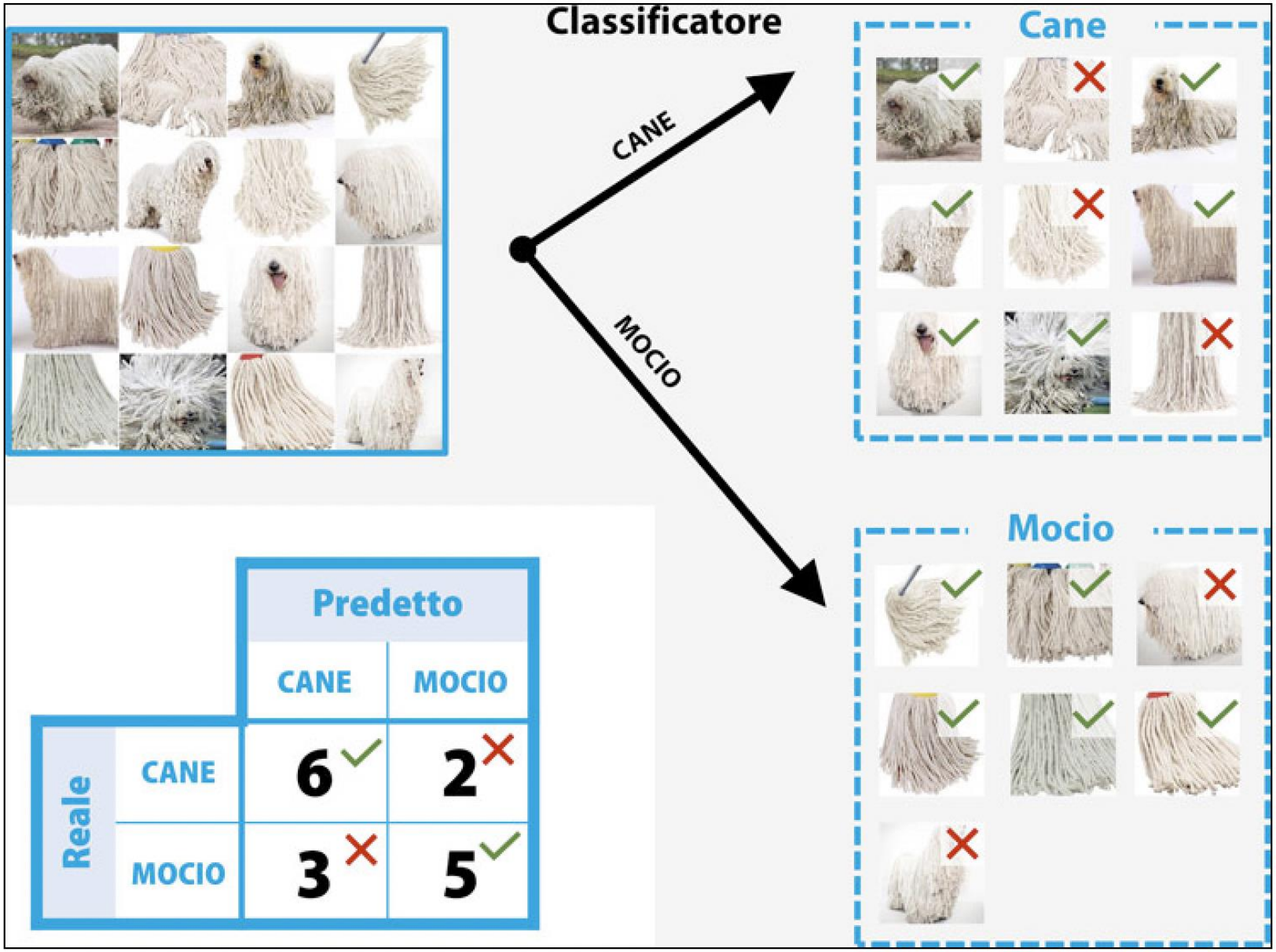
» **Pro**
- Facilissima da interpretare
- Ottima per set bilanciati

» **Contro**
- Fuorviante se le classi sono sbilanciate
- (es. 95% di una classe → modello "accurato" al 95% anche se non impara nulla)

Precision

$$Precision = \frac{TP}{TP + FP}$$

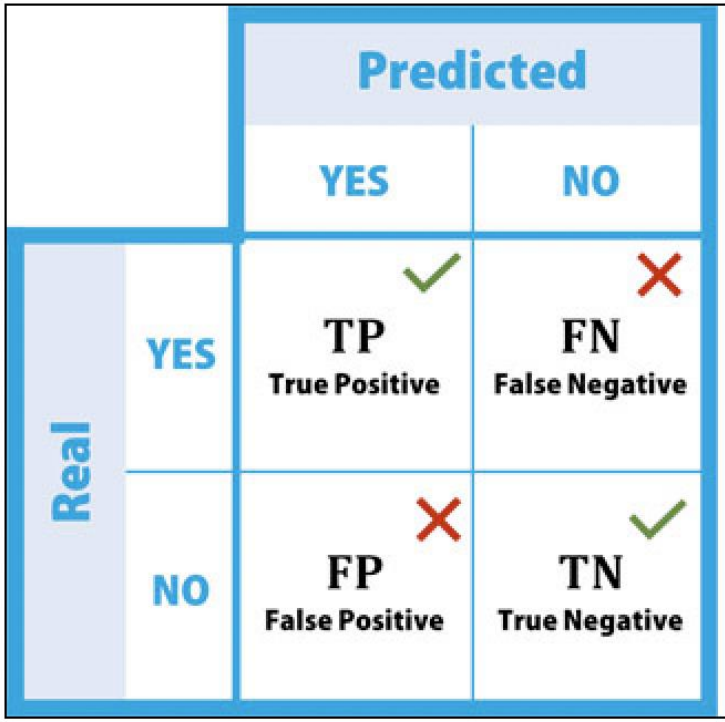
- Tra tutti i **positivi predetti**, quanti sono *realmente positivi*?
- **Quando è importante?**
  - Quando evitare falsi positivi è cruciale
  - (es. evitare di allarmare un paziente sano, evitare TAC non necessarie)
- **Esempio "cane vs mocio"**
  - Precision class "cane" = frazione di predizioni "cane" che sono davvero cani.


Classificatore
CANE
MOCIO
Cane
Mocio
Predetto
CANE
MOCIO
Reale
CANE
MOCIO
6
2
3
5

Sensitivity (Recall)

$$Sensitivity = \frac{TP}{TP + FN}$$

- Tra tutti i **veri positivi**, quanti vengono riconosciuti dal modello?
- **Quando è importante?**
  - Quando i **falsi negativi sono molto pericolosi**
  - (es. un tumore *non* riconosciuto → paziente non curato in tempo)
- **Esempio test medico**
  - Un FN = paziente malato classificato come sano → errore gravissimo.



$$\text{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

$$\text{Sensitivity} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

F1-score

- Metrica per la classificazione che combina **precision** e **recall** in un unico valore.
- **Perché serve?**
  - Utile quando le classi sono **sbilanciate**
  - Utile quando **precision** e **recall** sono entrambe importanti
  - Evita di concentrarsi su una sola delle due misure

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F1-score

- **Interpretazione**
  - **Alto F1** → il modello bilancia bene precision e recall
  - **Basso F1** → almeno una tra precision o recall è scarsa
  - Valori compresi tra **0 e 1** (1 è perfetto)
- **Quando usarlo**
  - Classi sbilanciate
  - Situazioni dove FP e FN hanno pesi simili
  - Problemi di classificazione binaria e multi-classe (one-vs-rest)

Quando usare quale metrica?

Scenario	Metrica consigliata	Perché
Regressione	**RMSE**	Misura l'errore medio in unità reali; intuitivo e confrontabile fra modelli sullo stesso problema
	R^2	Spiega quanta parte della variabilità del fenomeno viene "catturata" dal modello
Classificazione	**Accuracy**	Da usare quando le classi sono **bilanciate** e i costi di FP/FN sono simili
	Precision	Utile quando bisogna evitare **falsi positivi** (allarmi inutili, diagnosi errate "per eccesso")
	Recall (Sensitivity)	Utile quando bisogna evitare **falsi negativi** (mancate diagnosi, frodi non rilevate)
	F1-score	Metrica unica che bilancia precision e recall; ottima per **dataset sbilanciati**