

# Non aver paura del Machine Learning!

(oppure: il tuo primo modello di ML)



**Giuseppe Mastrandrea**  
Machine Learning Specialist @ Data Masters

**Linux Day Matera**  
28/10/2023

AI&ML



# Introduzione

## Argomenti del modulo:

- Introduzione al ML
- Tipi di apprendimento
- Processo di addestramento
- Hands on!



# Chi sono

- Ingegnere informatico
- Sviluppatore web @ Frankhood dal 2011
- Docente di informatica @ ITT Panetti Pitagora
- Machine Learning Specialist @ Datamasters dal 2020



# IA vs ML vs Deep Learning



## INTELLIGENZA ARTIFICIALE

Un software in grado di agire ed adattarsi in maniera autonoma

Ad esempio AIML, agenti intelligenti, sistemi basati sulla conoscenza, ...

## MACHINE LEARNING

Algoritmi le cui performance  
Migliorano con l'esperienza  
accumulata confrontandosi con  
più dati nel tempo

Ad esempio alberi decisionali,  
regressioni, SVM, ...

## DEEP LEARNING

Subset del Machine  
Learning dove reti neurali  
multistrato apprendono da  
set di dati molto ampi

# Definizione

*Il **Machine Learning** è un campo di studio che offre a un computer la capacità di apprendere qualcosa senza esserne esplicitamente programmato.*

Arthur Samuel, esperto statunitense di intelligenza artificiale e videogames, coniò il termine «Machine Learning» e la relativa definizione nel 1959.



*...il meccanismo principale della macchina si basava sull'analisi probabilistica delle posizioni raggiungibili dalla posizione attuale. Siccome la macchina disponeva di una quantità di memoria molto limitata, Samuel decise di implementare l'algoritmo di ricerca potatura alfa-beta. Invece di cercare in una volta sola ogni possibile strada per arrivare all'altra sponda, e conseguentemente vincere il gioco, Samuel sviluppò una funzione in grado di analizzare la posizione della dama in ogni istante della partita. Questa funzione provava a calcolare le possibilità di vittoria per ogni lato nella posizione attuale, agendo di conseguenza. Prendeva in considerazione diverse variabili tra cui il numero di pezzi per lato, il numero di dame e la distanza dei pezzi 'mangiabili'. Il programma sceglieva le sue mosse basandosi sulla strategia minimax, ovvero agendo in modo da ottimizzare il valore della sua funzione, assumendo che l'avversario agisse e ragionasse nel medesimo modo...*

— Wikipedia —

# Definizione

Si dice che un software impari dall'esperienza **E** rispetto ad alcune classi di attività **T** e misura delle prestazioni **P**, se la sua prestazione in compiti in **T** misurata da **P** migliora con l'esperienza **E**.

**Tom Mitchell** - Informatico e professore universitario – 1998  
Rilevante poiché per la prima volta una definizione operativa  
dell'apprendimento automatico

Es.  
E = esperienza nel giocare a scacchi  
T = compito di giocare a scacchi  
P = probabilità che il programma  
vinca la partita successiva



# Machine Learning

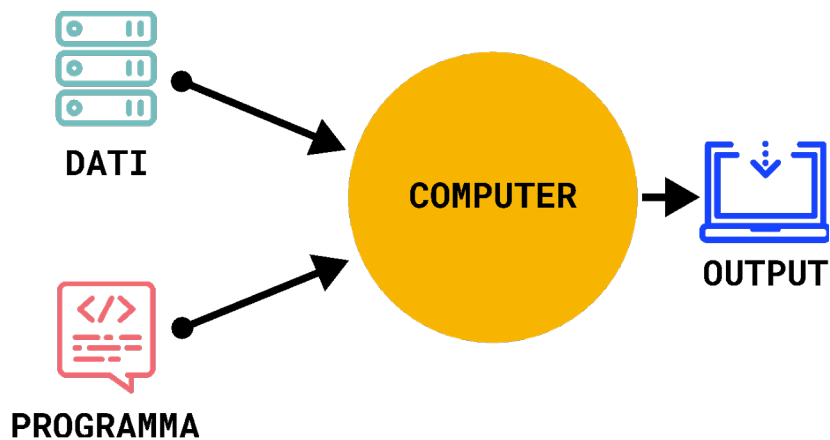


**Usare dati** per **rispondere a domande**

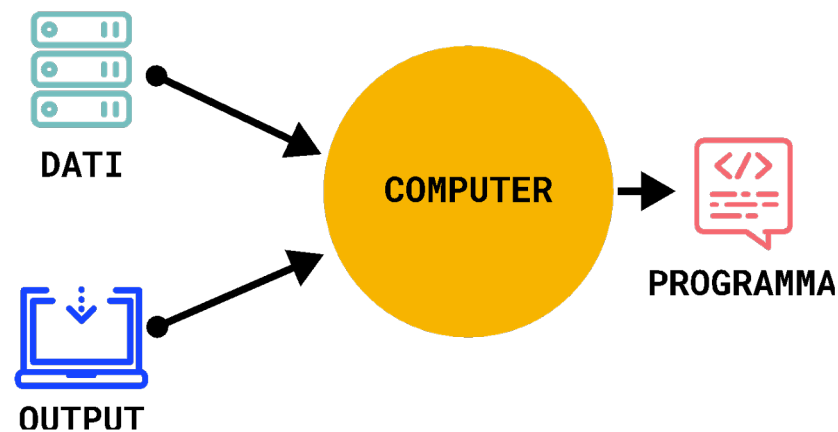
addestramento

predizione / classificazione

## Programmazione tradizionale



## Machine Learning



# Perché oggi: l'era dei Big Data

## DATI

Dati disponibili ovunque

Bassi costi per  
l'archiviazione dei dati

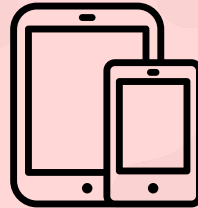
Hardware più potente e più  
economico



## DISPOSITIVI

Chiunque ha dispositivi  
elettronici con connettività  
internet e sensoristica che  
raccolge dati

- GPS
- Fotocamera
- Microfono



## SERVIZI

Cloud computing

- archiviazione online
- infrastrutture disponibili  
come servizi

Applicazioni disponibili

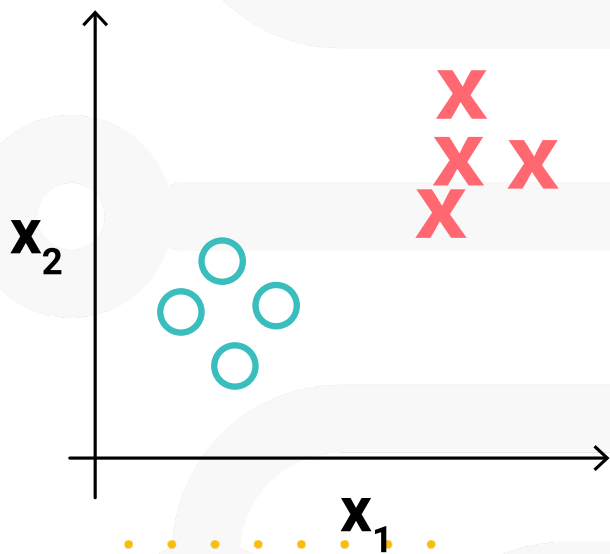
- YouTube
- Gmail
- Facebook
- Twitter
- ...



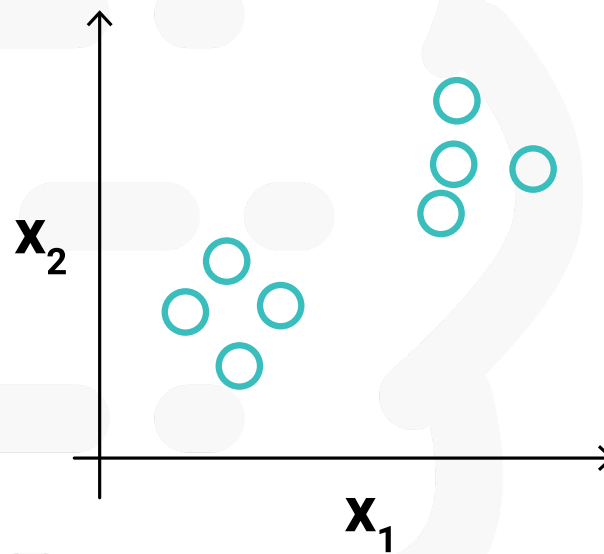


# Classificazione algoritmi di M.L.

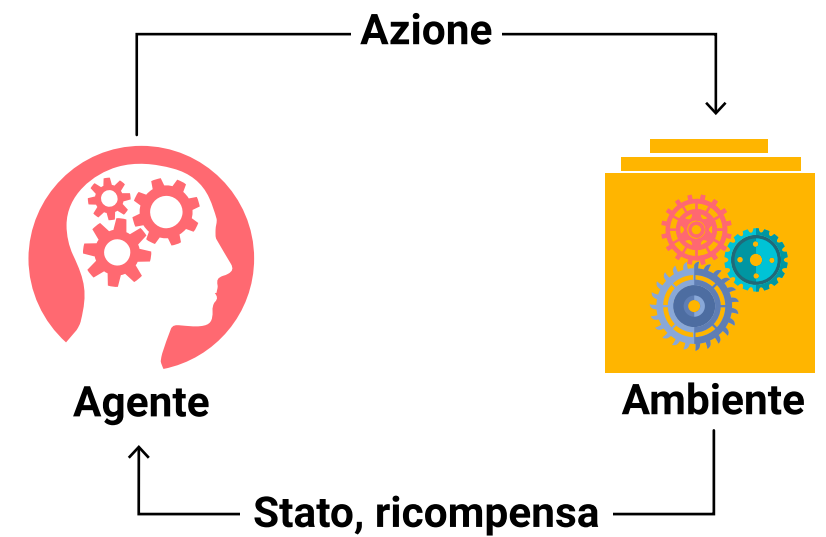
In generale, qualsiasi problema di apprendimento automatico può essere ricondotto a una delle **seguenti classi di algoritmi**:



**Apprendimento  
supervisionato**



**Apprendimento  
non supervisionato**



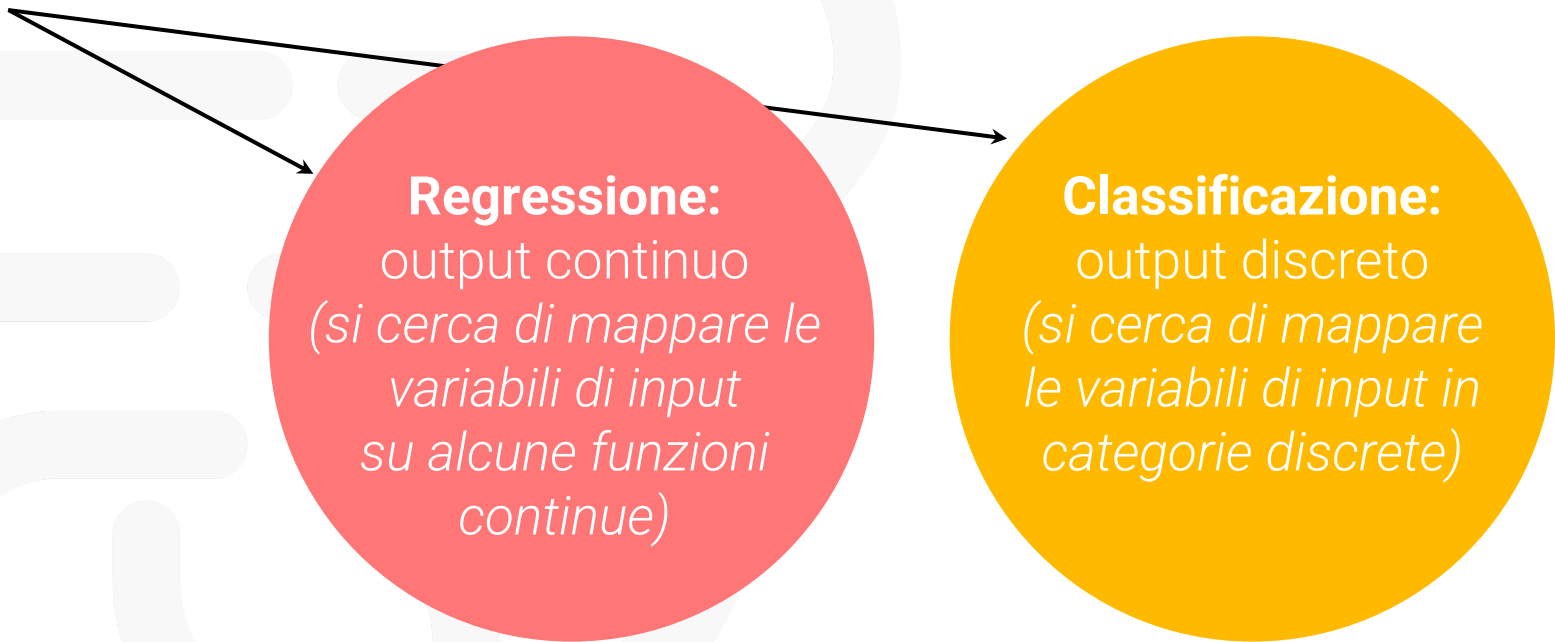
**Apprendimento  
per rinforzo**



# Apprendimento supervisionato

Viene fornito un set di dati e si sa come dovrebbe essere il nostro output corretto, supponendo che ci sia una relazione tra input e output.

I problemi di **apprendimento supervisionato** sono classificati in:



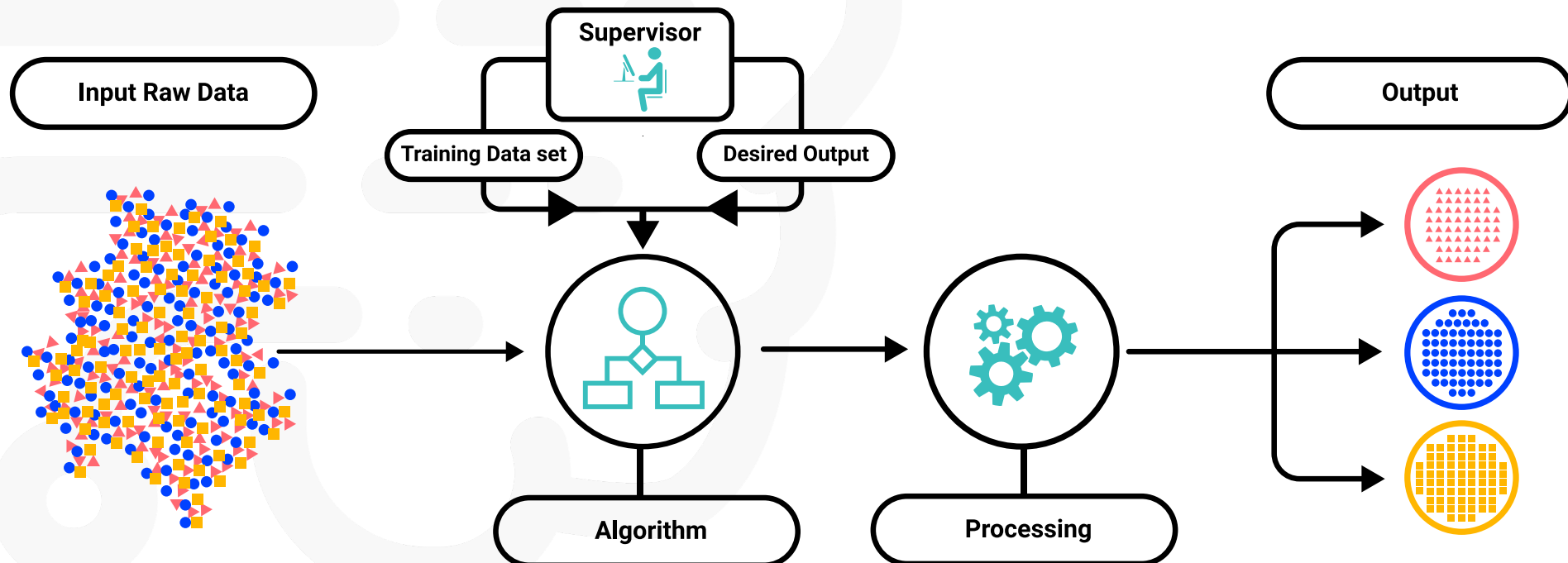
**Regressione:**  
output continuo  
*(si cerca di mappare le  
variabili di input  
su alcune funzioni  
continue)*

**Classificazione:**  
output discreto  
*(si cerca di mappare  
le variabili di input in  
categorie discrete)*



# Esempi di apprendimento supervisionato

- Da dati sulla dimensione delle case sul mercato immobiliare, si prova a **prevederne il prezzo** (regressione) o la fascia di prezzo (classificazione)
- **Prevedere l'età di una persona** basandosi su una sua fotografia (regressione)
- **Stabilire se un tumore** è benigno o maligno (classificazione)



# Apprendimento non supervisionato

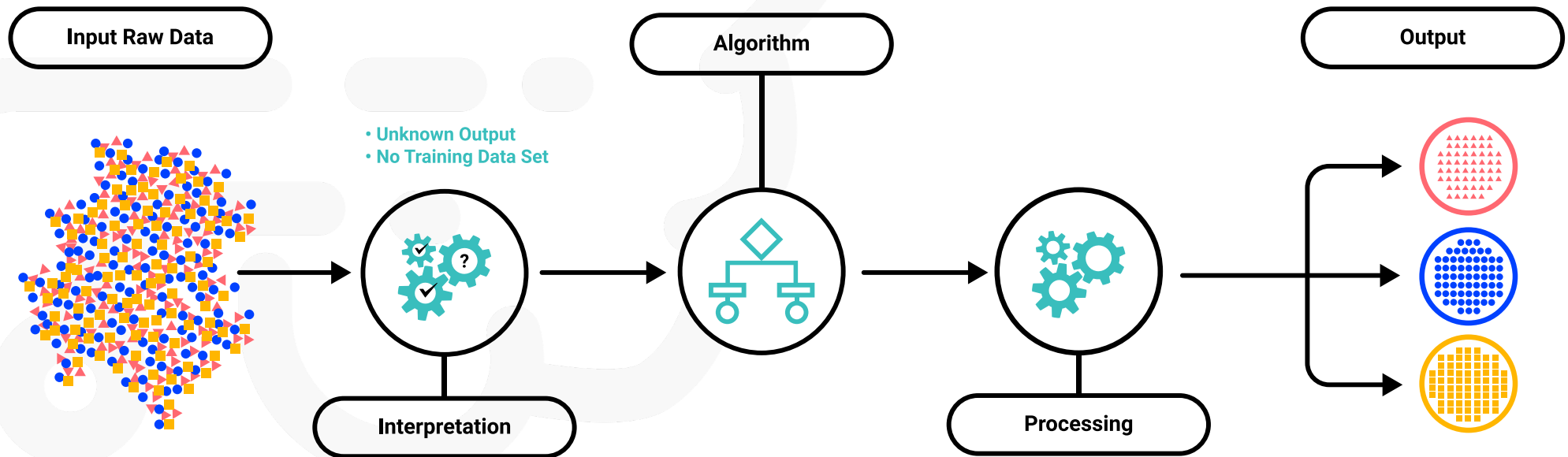
- L'apprendimento non supervisionato si applica in contesti con poche o nessuna idea relativamente ai risultati
- Possiamo derivare la struttura di un modello da dati in cui non conosciamo necessariamente l'effetto delle variabili
- Possiamo ricavare la struttura del modello raggruppando i dati in base alle relazioni tra le variabili nei dati
- Con l'apprendimento senza supervisione non esiste alcun feedback basato sui risultati della previsione



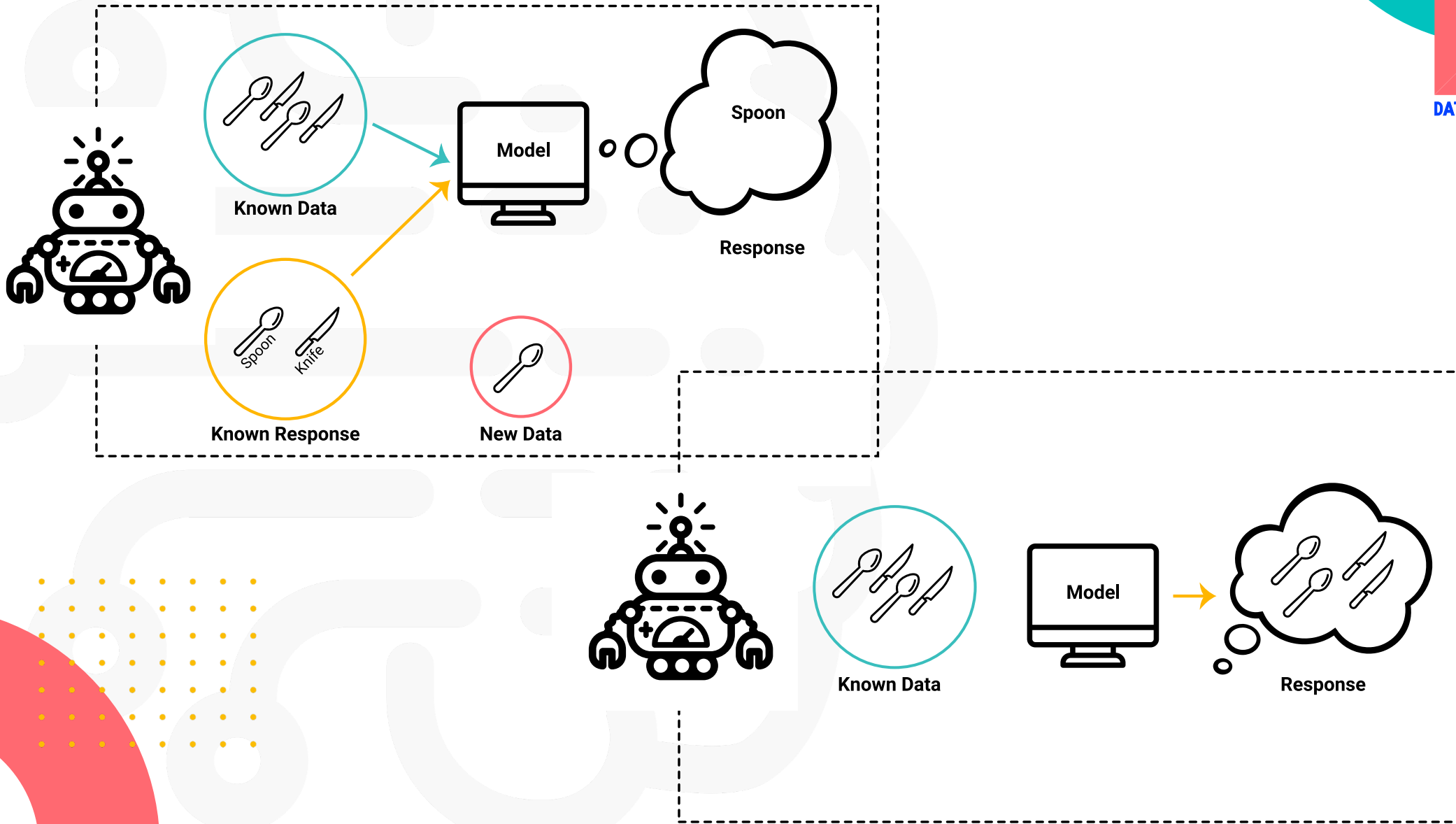
# Esempi di apprendimento non supervisionato

- Da una raccolta di 1.000.000 di geni diversi si trova un modo per raggruppare automaticamente questi geni in gruppi che sono in qualche modo simili o correlati da variabili diverse, come posizione, ruoli e così via.
- Identificare singole voci e musica da un insieme di suoni in un bar.

 [https://cnl.salk.edu/~tewon/Blind/blind\\_audio.html](https://cnl.salk.edu/~tewon/Blind/blind_audio.html)



# Supervisionato vs Non Supervisionato



# Apprendimento per rinforzo

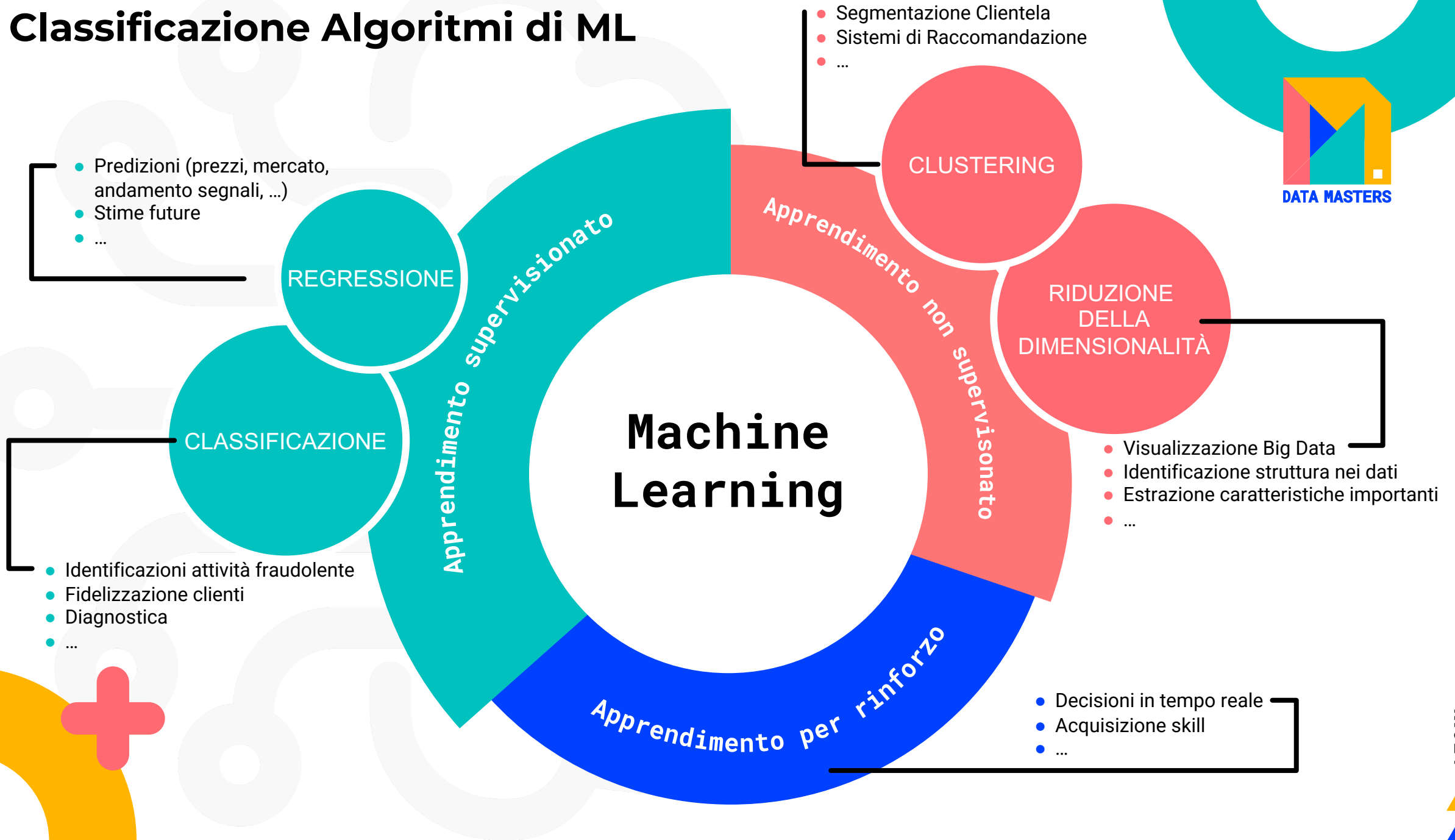
Apprendimento **tramite interazione con l'ambiente** e le conseguenze delle proprie azioni.

## Step:

1. Osservazione dello stato in cui l'ambiente si trova
2. Decisione
3. Passaggio in un nuovo stato
4. Ricompensa



# Classificazione Algoritmi di ML



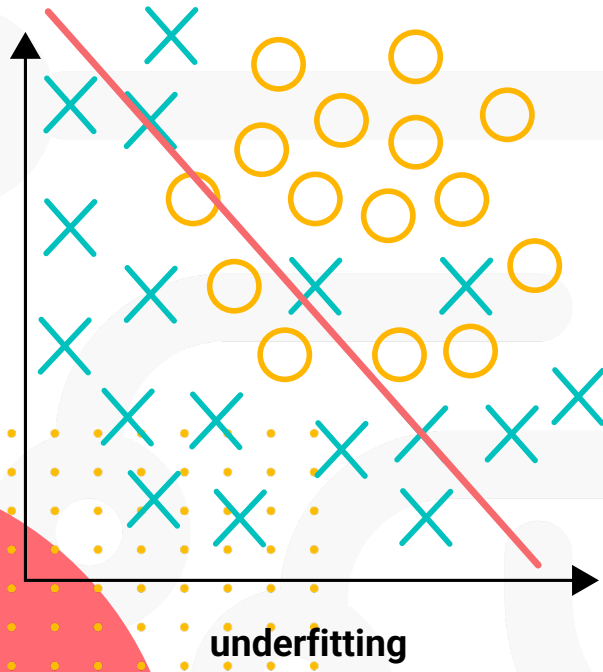


# Overfitting

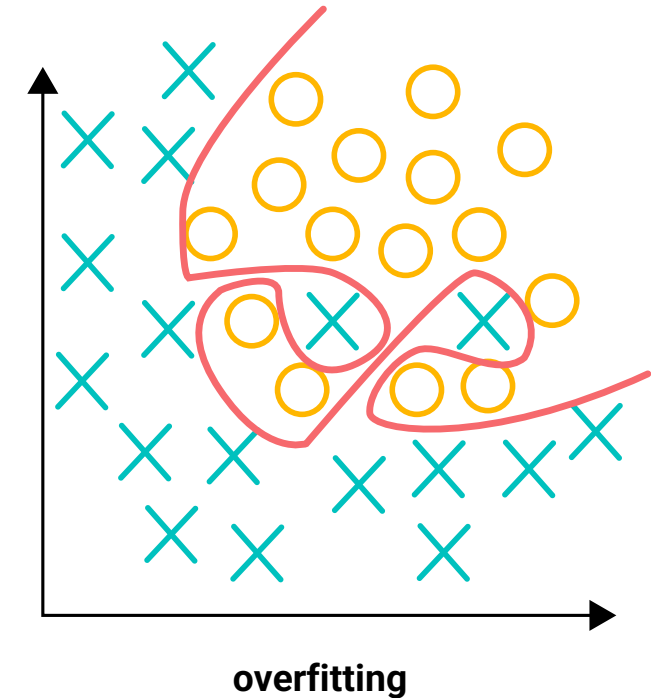
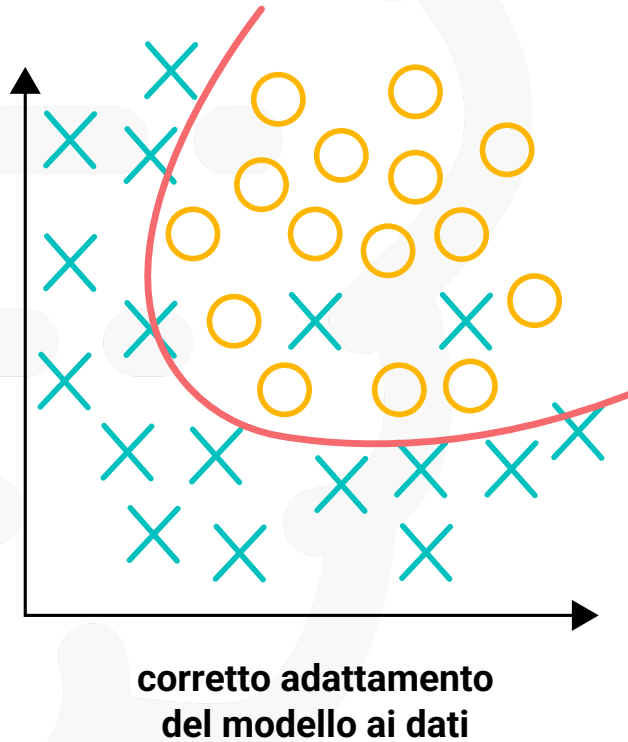
## OBIETTIVO

Trovare un equilibrio tra l'adattamento ai dati di addestramento e la capacità di generalizzare su nuovi dati.

modello debole



modello complesso



# Overfitting - Esempi

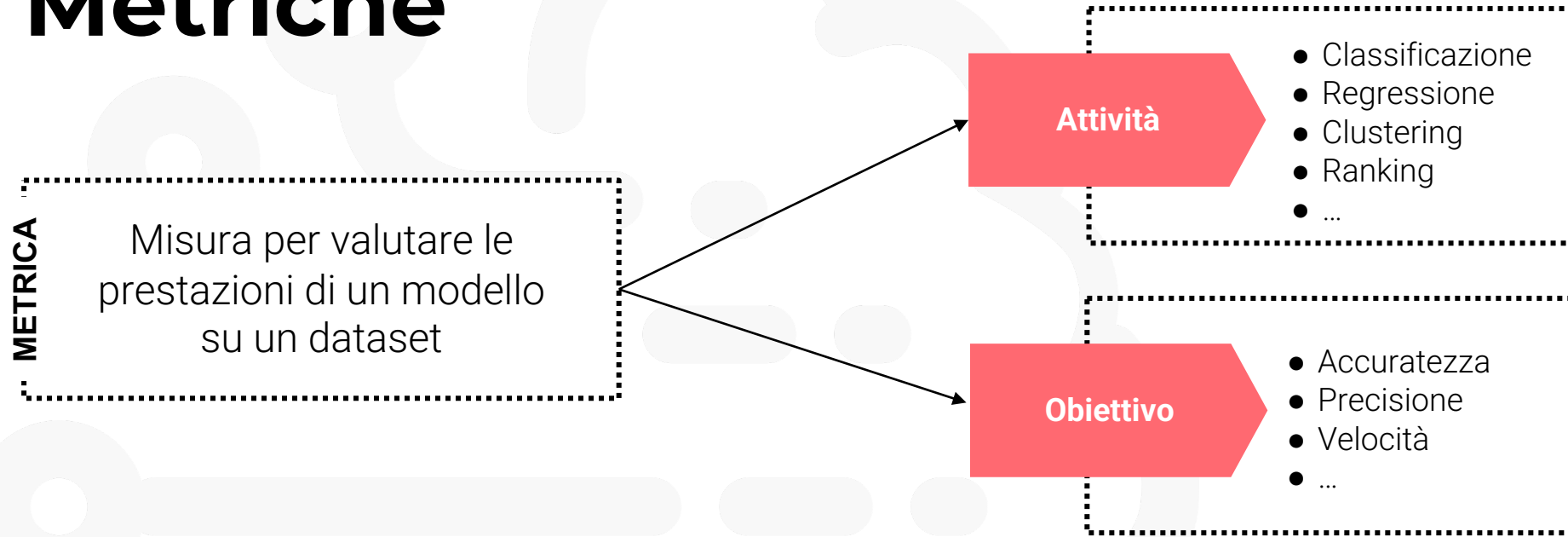
- Elevato numero di attributi o attributi non necessari:
  - House Pricing - colore pareti e pavimento
  - Lancio della monetina - piove sì / no
- Elevato numero di parametri -> simile ad avere troppi attributi per il problema in esame

**Tecniche**

**Convalida incrociata dei dati**



# Metriche



**Misurare le prestazioni del modello**

**Prendere decisioni informate**

La scelta della metrica corretta dipende:

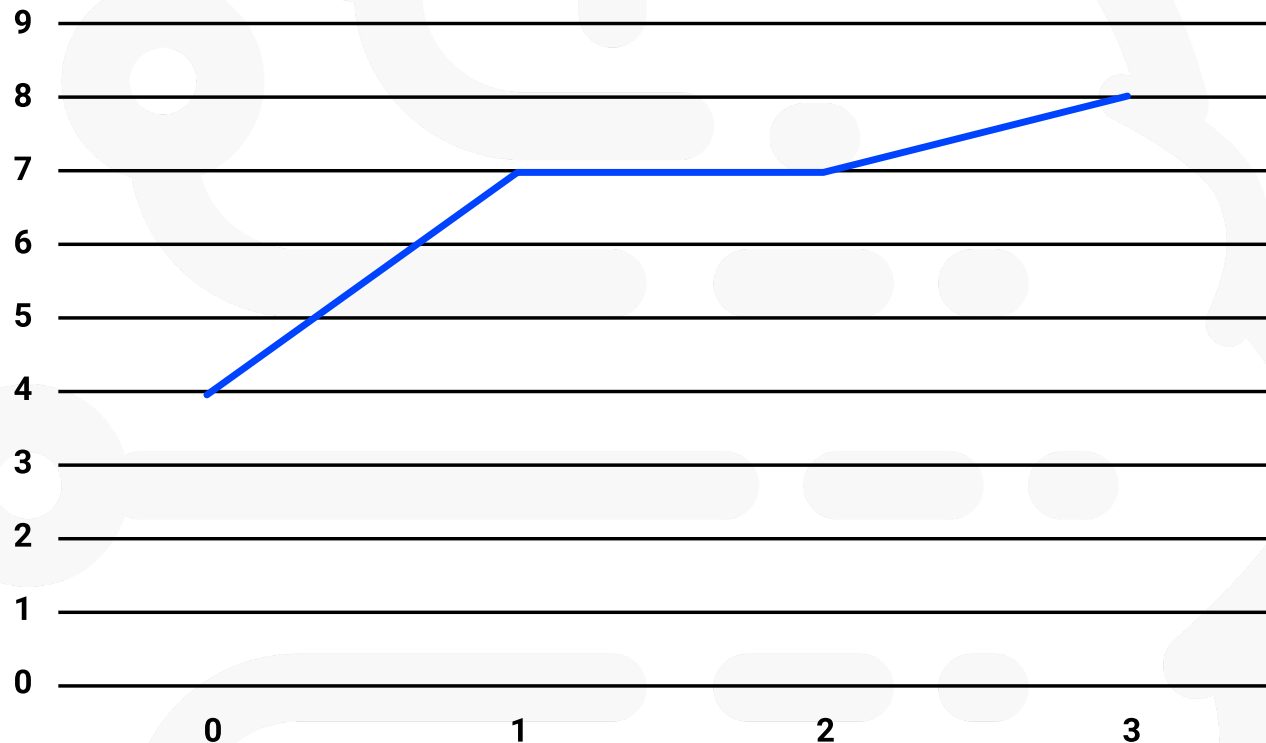
- Natura del problema (dataset)
- Conseguenze delle previsioni errate

# Algoritmi di Machine Learning



- Regressione Lineare
- Regressione Logistica
- K-Nearest Neighbors
- Random Forest
- XG-Boost
- Naive Bayes
- Support Vector Machine
- K-Means
- Alberi Decisionali
- Reti Neurali Artificiali
- Q-Learning
- Clustering Gerarchico
- PCA

# Processo di addestramento



INPUT (x)	OUTPUT (y)
0	4
1	7
2	7
3	8

Da questo dataset, **come fare** per predire l'output di un nuovo input non presente nel dataset fornito, come ad esempio **42** ?

# Processo di addestramento



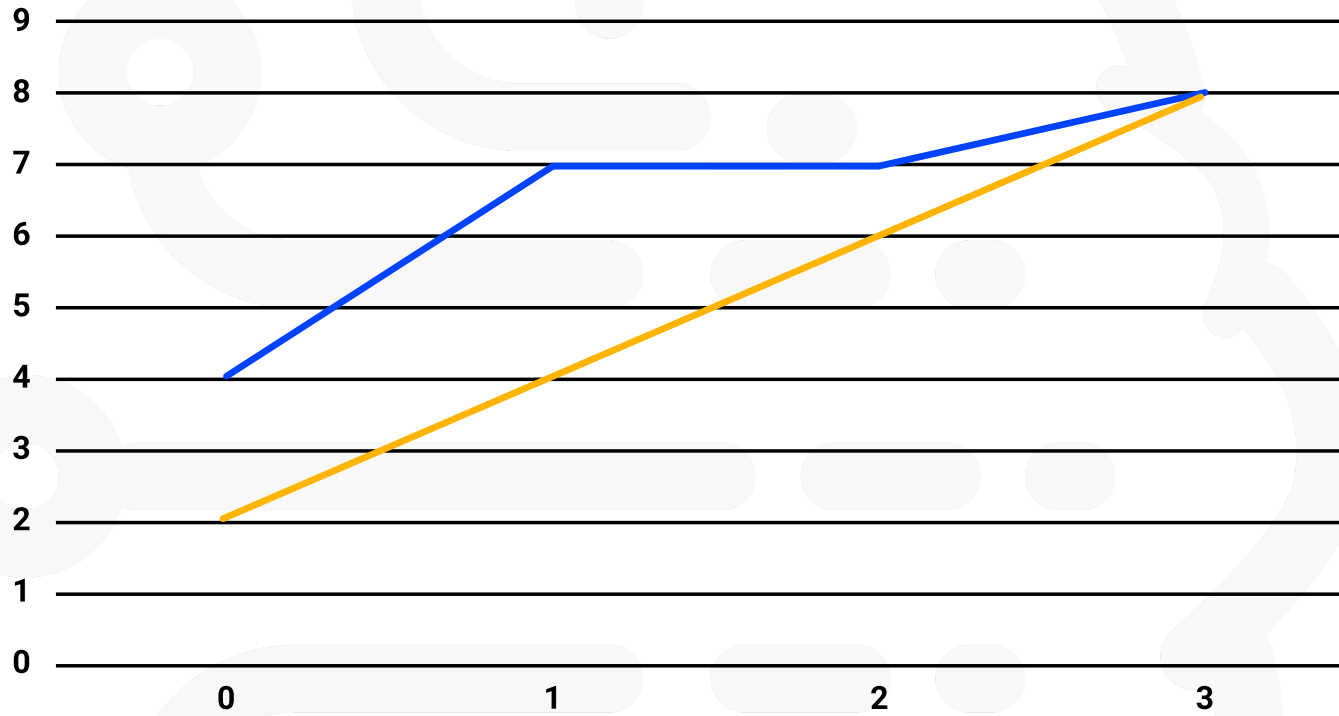
La **regressione lineare univariata** viene utilizzata quando si desidera prevedere un singolo valore di output y da un singolo valore di input x

E' una tipologia di **apprendimento supervisionato**

$$\hat{y} = h_{\theta}(x) = \theta_0 + \theta_1 x$$



# Processo di addestramento



INPUT (x)	OUTPUT (y)
0	4
1	7
2	7
3	8

Ipotesi:  $\theta_0 = 2$   
 $\theta_1 = 2$   $h_{\theta}(x) = 2 + 2x$

# Processo di addestramento

- Necessaria per misurare l'accuratezza di un modello
- Dipendente dal modello utilizzato
- Utile per stimare i parametri del modello
- E' una funzione che mappa un evento su un numero reale che rappresenta un "**costo**" associato all'evento
- Normalmente si cerca di **minimizzare** una funzione di costo
- Dipende dalla differenza tra i valori attesi e quelli reali, su un determinato dataset





# Processo di addestramento



- Permette di misurare l'accuratezza del nostro modello
- Effettua una media tra i risultati delle ipotesi del modello rispetto ai valori riscontrati nella realtà
- E' chiamata Errore Quadratico Medio (**Mean Square Error - MSE**)

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

# Processo di addestramento

- Abbiamo una funzione di ipotesi
- Sappiamo misurare quanto tale funzione rappresenta dati reali



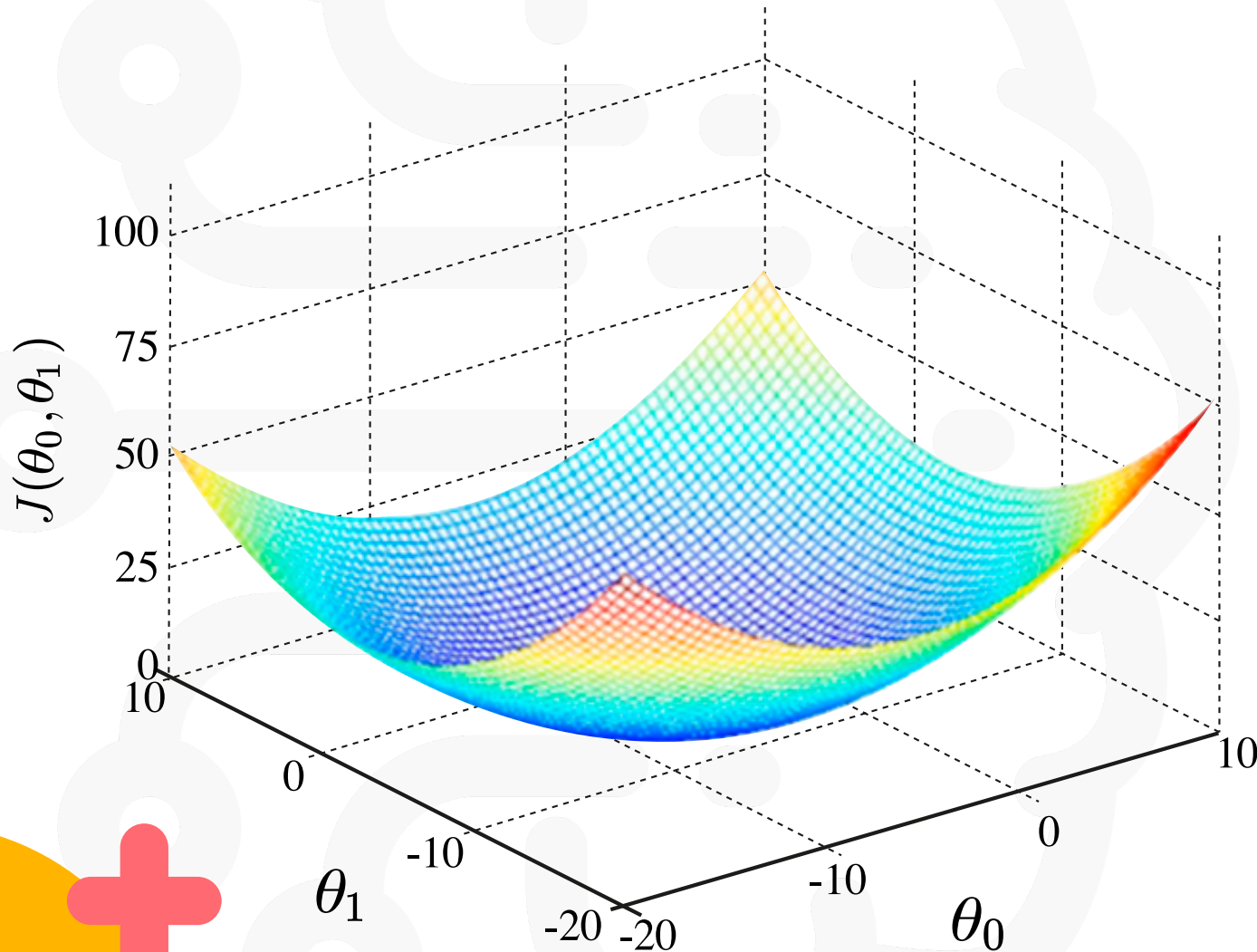
Cosa  
fare?

Dobbiamo **stimare al meglio** i parametri della funzione di ipotesi per minimizzare l'errore prodotto dalla nostra ipotesi rispetto ai casi reali

- In modo casuale
- Seguendo un metodo

Come?

# Processo di addestramento



Rappresentiamo graficamente la **funzione di costo** in base alla variazione dei parametri della funzione ipotesi



# Processo di addestramento

Il modo in cui ottimizziamo i parametri della funzione di ipotesi è prendendo la **derivata** (=tangente alla funzione) della nostra funzione di costo.

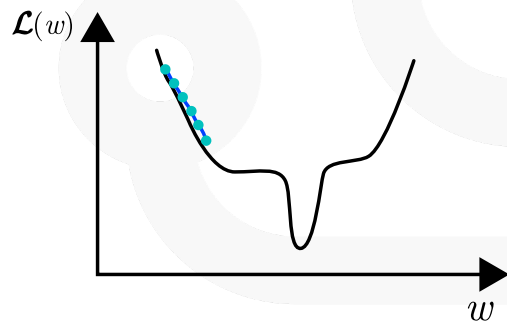


La pendenza del piano tangente in un punto è data dalla derivata in quel punto e fornisce la direzione su cui muoverci.

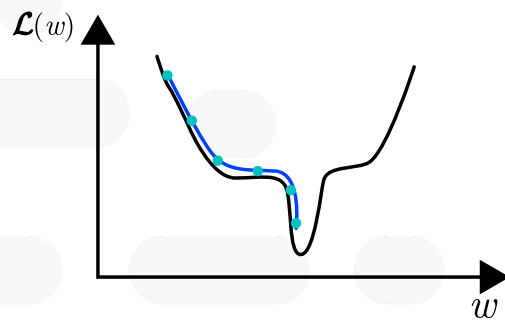
Riduciamo il costo della funzione ipotesi seguendo la direzione trovata, compiendo passi determinati da un parametro  $\alpha$ , che è chiamato «tasso di apprendimento» (**learning rate**).



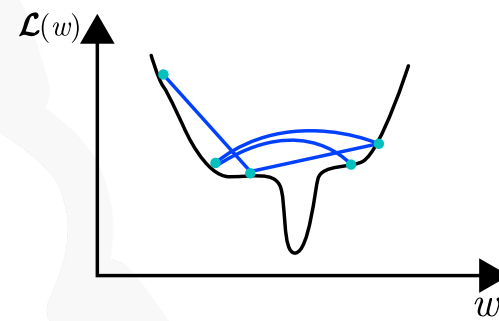
# Processo di addestramento



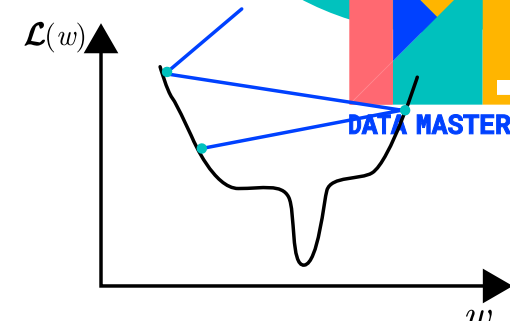
Learning rate too low



Good learning rate

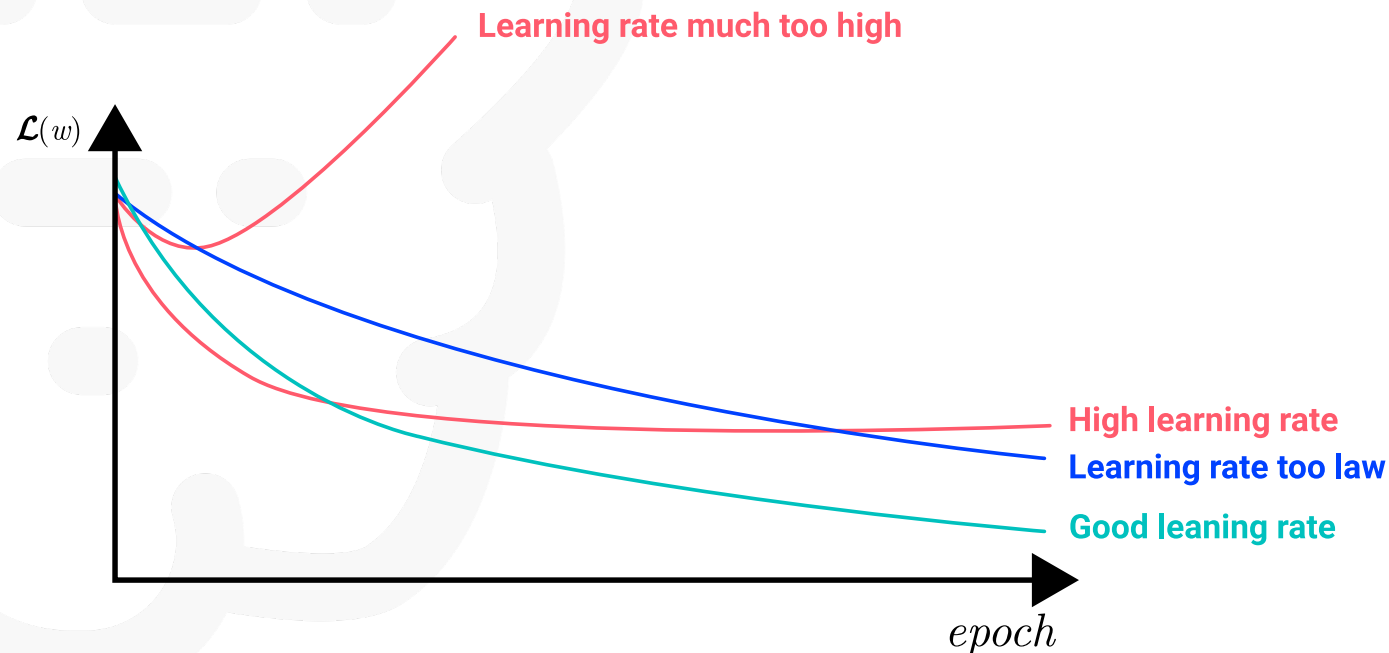


High learning rate



Learning rate much too high

**Variare** il learning rate al massimo di un 30% tra un test ed un'altro



# Processo di addestramento

Repeat  
until  
Convergence:

$$\left\{ \begin{aligned} \theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i) \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m ((h_{\theta}(x_i) - y_i) x_i) \end{aligned} \right\}$$



# Processo di addestramento

Scegliere una **funzione di costo**

Dare dei valori iniziali ai parametri del modello

Iniziare il loop della **discesa del gradiente**

- Fare le predizioni con i parametri correnti
- Calcolare la funzione di costo
- Calcolare il gradiente della funzione di costo
- Aggiornare i parametri di un fattore scalato del **learning rate**
- Andare avanti per un numero prefissato di epoche oppure **fino a convergenza**



# Hands on!

Vai su

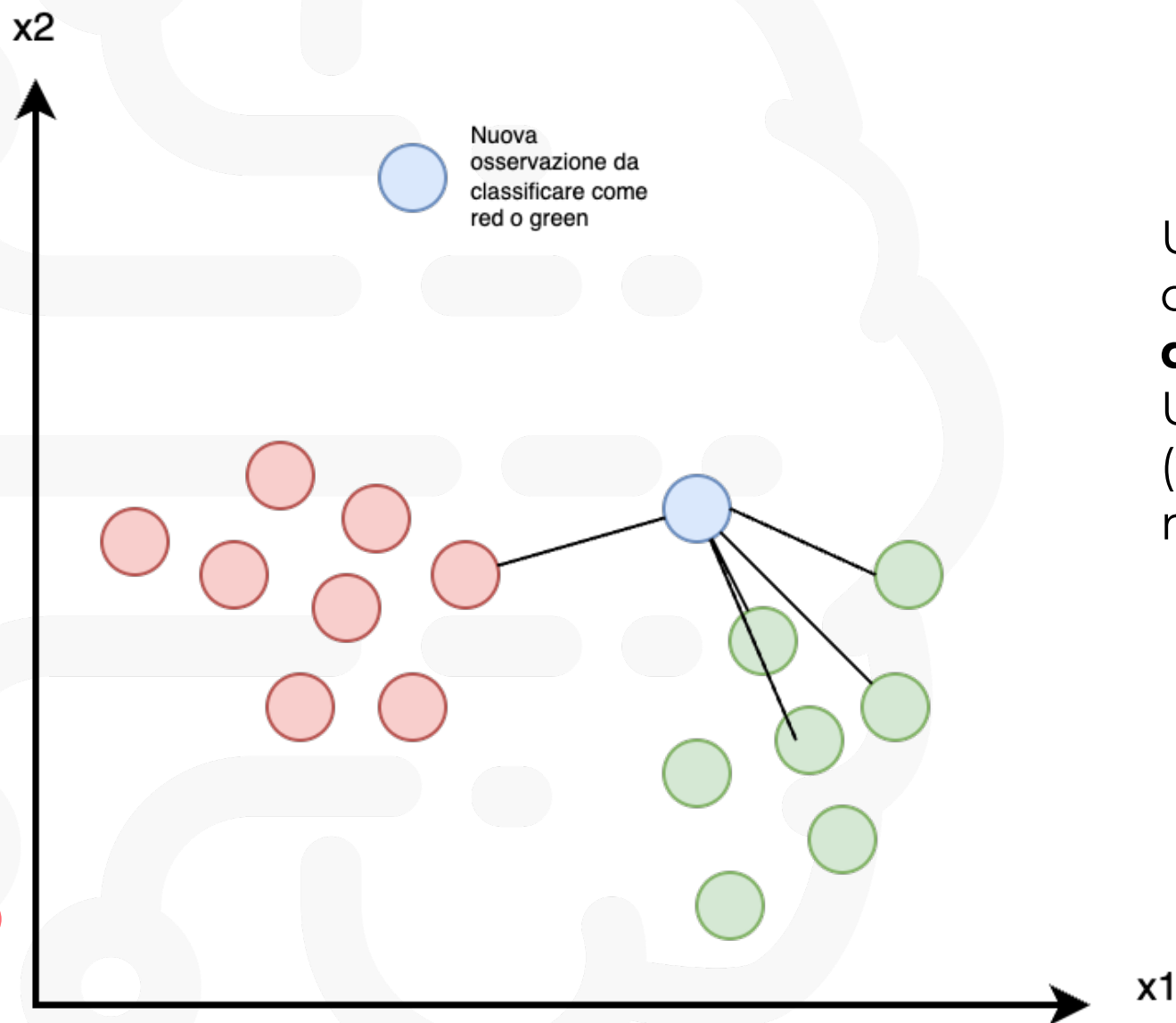
<https://bit.ly/dont-fear-ml>

E segui le istruzioni!





# Bonus Track - KNN



Usata per fare predizioni di osservazioni basandosi sui **valori dei K «vicini più vicini»**  
Usato sia in task di classificazione (voto a maggioranza) che di regressione (media dei K vicini)



# Let's keep in touch!



[launchpass.com/datamasters](https://launchpass.com/datamasters)

- LinkedIn: [Giuseppe Mastrandrea](#)
- Instagram: [giu.mast](#)
- Facebook: [Giu Mast](#)
- Substack: [giumast.substack.com](#)

