

# Fondamenti di Machine Learning

Laurea Triennale in Ingegneria delle Comunicazioni

## 1 - Introduzione

---

Docente: S. Scardapane



**SAPIENZA**  
UNIVERSITÀ DI ROMA

# Introduzione

---

Che cosa è il machine learning?

Un neolaureato in una filiale di una banca si vede assegnati due task da risolvere:

### Task 1

---

Filtrare un insieme di **email** in base alla **dimensione dei loro allegati**.

### Task 2

---

Filtrare un insieme di **clienti** in base alla **probabilità che vadano in default**.

Il primo task può essere risolto facilmente, mentre il secondo è un buon esempio di **apprendimento automatico**.

Il concetto di *probabilità di default* non è facile da definire, ma diventa possibile avendo uno storico: ad esempio, se i clienti sono andati in negativo in passato, sono (probabilmente) *inattendibili*.

**L'apprendimento automatico (ML)**, ed in particolare **l'apprendimento supervisionato**, permettono di sfruttare questo concetto *inferendo e generalizzando* la relazione da dati storici a *nuovi potenziali clienti*.

Questo è solo un esempio didattico, ed il **credit scoring** tramite ML è una pessima idea per varie ragioni, come vedremo più avanti.

Un **dataset** è un insieme di coppie (input, output) che descrivono **esempi** di quanto vorremmo ottenere (es., stimare il credit score).

Età	Cliente da (anni)	Disponibilità	Prestito	Default?
18	2	3000	1000	0
42	5	10000	2000	1
...	...	...	...	...

Date le prime 4 colonne (**input**), vogliamo predire la colonna finale (**output**). Ogni colonna di input viene detta una **feature**, mentre l'ultima colonna viene detta **target**. Il dataset viene detto **tabellare**.

- ▶ In questo corso consideriamo due tipi di feature molto semplici: feature **numeriche** (numeri reali), e feature **categoriche** (valori discreti).
- ▶ Il target, invece, può essere a sua volta un valore continuo (es., prezzo di una casa) o discreto (es., classe di un cliente). Parliamo rispettivamente di **regressione** e **classificazione**.
- ▶ Le feature possono essere direttamente in nostro possesso, oppure estratte in qualche modo dai dati (**feature engineering**).

Un **modello** è una funzione che prende in input le feature e produce un output desiderato (es., la classe):

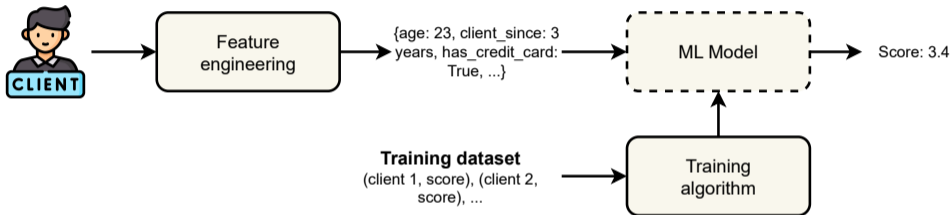
$$\hat{y} = f(x) \tag{1}$$

Un **algoritmo di apprendimento** è un algoritmo che, dato un dataset, produce una istanza particolare della classe di modelli scelta: diciamo che stiamo **addestrando** il modello.

Esistono numerosissimi modelli e algoritmi di apprendimento per dataset tabellari, ciascuno con le sue proprietà e limiti, che vedremo più avanti.

- ▶ Il modello descrive in che modo *rappresentare* la relazione tra input ed output: alberi decisionali, regole logiche, reti neurali...
- ▶ L'algoritmo di apprendimento descrive come costruire un modello che abbia buone prestazioni sul proprio dataset.

Come vedremo, scegliere questi due aspetti dipende da tanti fattori: capacità di rappresentazione, tipo di dato, efficienza computazionale. Dobbiamo anche avere buone **metriche** per valutare le performance.



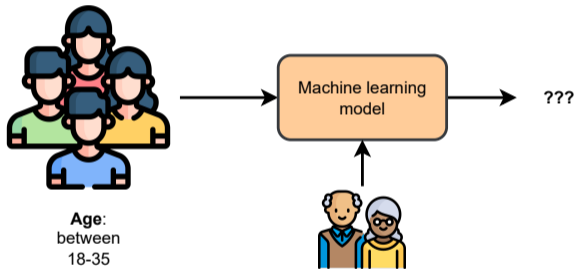
**Figure 1:** Componenti essenziali di un sistema di apprendimento automatico: definizione delle **feature**, **dataset**, **modelli**, e **algoritmi di apprendimento**.

Un **dataset** è un insieme di coppie (input, output) che descrivono **esempi** di quanto vorremmo ottenere (es., stimare il credit score). Costruire un buon dataset non è banale ed è solitamente la parte più complessa del problema:

- ▶ Deve essere *rappresentativo* di tutti i possibili clienti (passati e futuri).
- ▶ Deve essere sufficiente *numeroso* per permettere di generalizzare a nuovi clienti.

Ritorniamo su questi aspetti più avanti quando definiremo il problema in maniera formale.

Spesso i dati su cui vengono allenati i modelli non corrispondono ai dati su cui vengono richieste predizioni: in questo caso, si parla di **domain shift**.



*Images from Flaticon.com*

**Figure 2:** Un esempio artificiale di domain shift: una banca con clienti prevalentemente giovani allena un modello per uso interno, ma non può applicarlo su nuovi clienti che non rientrano in queste caratteristiche.

Questo semplice esempio chiarisce anche che in molti casi è necessaria una fase di **pre-processing** dei dati per renderli più facilmente manipolabili, es.:

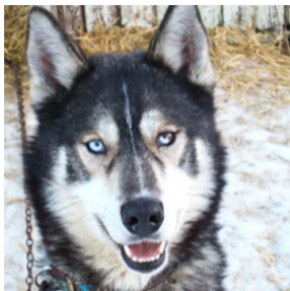
- ▶ Colonne diverse hanno diversi range (**normalization**);
- ▶ Alcuni dati potrebbero essere mancanti (**missing data imputation**);
- ▶ Alcuni valori potrebbero essere incorretti o sbagliati (**outlier detection** or **anomaly detection**);
- ▶ Nuove feature possono essere estratte da feature in nostro possesso, es., il CAP a partire dall'indirizzo (**feature engineering**).

Un famoso aneddoto (con buona probabilità apocrifo)<sup>1</sup> descrive una rete neurale allenata a riconoscere carri armati. Per errore, tutte le foto di carri armati nel training set sono scattate in giorni nuvolosi, e la rete ‘impara’ questa correlazione. Le prestazioni in fase di training sono eccellenti, mentre in fase di *deployment* sono pessime.

Questo è un esempio molto artificiale di **bias** nella collezione dei dati. Lo scarto di prestazione viene detto **overfitting**.

---

<sup>1</sup><https://gwern.net/tank>



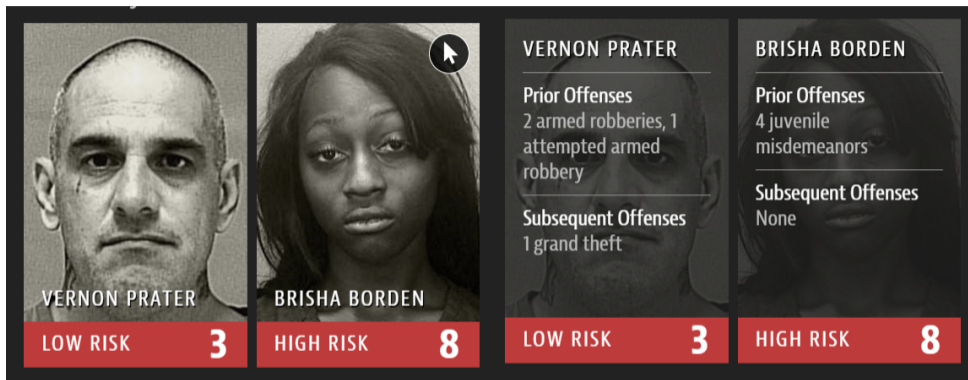
(a) Husky classified as wolf



(b) Explanation

**Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.**

**Figure 3:** Un famoso esempio (artificiale) di bias (riprodotto da Ribeiro et al., 2016): un sistema allenato a riconoscere gli husky impara in realtà a riconoscere la neve sullo sfondo.



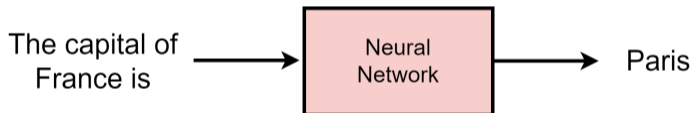
**Figure 4:** COMPAS è un esempio importante di bias in un sistema usato in ambito giudiziario, a seguito di una indagine di Pro Publica nel 2016. L'inchiesta ha avuto un ampio seguito mediatico ed accademico, in quanto le conclusioni possono variare a seconda di quale metrica di 'unfairness' scegliamo di usare.

# Introduzione

---

Machine learning e reti neurali

Un modello come ChatGPT genera una distribuzione sul **prossimo blocco di testo** (token), quindi viene chiamato un **language model**. Usandolo ripetutamente possiamo generare testi molto lunghi (**autoregressione**).

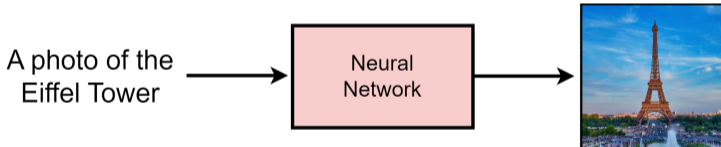


Nella terminologia di prima, il prompt è l'*input*, mentre la generazione è l'*output*, e la rete neurale è il *modello*.



Figure 5: MCU Characters as 80s Wrestlers [Reddit]

Un modello generativo come Stable Diffusion rientra sempre nell'ambito dell'apprendimento supervisionato, ma in questo caso l'input è una descrizione testuale e l'output una immagine:



Stable Diffusion e ChatGPT sono a loro volta esempi di apprendimento automatico, essendo allenati a partire da dataset di testo o di coppie (immagine, descrizione). Hanno però diverse caratteristiche peculiari:

- ▶ I loro input ed output possono essere estremamente complessi (testo, immagini, video, ...).
- ▶ Di conseguenza, i dataset richiesti possono a loro volta essere molto grandi (milioni di immagini, PB di testo).

Di tutti i modelli di apprendimento supervisionato, le **reti neurali** sono l'unico metodo noto che possono scalare a questi contesti. In questo corso vedremo solo le basi per quanto riguarda le reti neurali.

# Introduzione

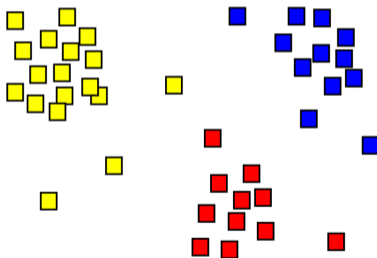
---

Altri argomenti

L'apprendimento supervisionato è solo un sottoinsieme del campo del machine learning, seppur il più comune. Altre categorie interessanti di apprendimento automatico sono:

- ▶ **Unsupervised learning**: in questo caso il nostro obiettivo non dipende da una etichetta (*label*) associato agli input. Ad esempio nel **clustering** vogliamo raggruppare i nostri dati in insiemi sufficientemente omogenei. Nella **dimensionality reduction**, vogliamo proiettarli in uno spazio a 2D o 3D per la visualizzazione.
- ▶ **Reinforcement learning** (apprendimento con rinforzo): in questo caso il modello deve prendere diverse decisioni in sequenza, e riceve un reward solo al termine di ogni sequenza (es., giocare a scacchi).

Il clustering può essere inteso come un problema di classificazione per il quale non conosciamo le etichette in fase di training, es., segmentare i clienti di una azienda in gruppi non noti a priori per fini di marketing.



**Figure 6:** Esempio di clustering (riprodotto da Wikipedia).

Esistono poi argomenti ortogonali alla scelta del modello e del tipo di apprendimento, che approfondiremo solo in parte:

- ▶ **Misura delle performance** (valutazione), soprattutto in situazioni con classi sbilanciate o costi diversi a seconda del tipo di errore.
- ▶ **Interpretabilità** delle predizione (*perché* è stata scelta una certa classe).
- ▶ **Robustezza** dei modelli e **costo computazionale** per l'apprendimento e per l'esecuzione.
- ▶ **Deployment** dei modelli e verifica costante del loro corretto funzionamento.