

Intelligenza Artificiale

Paolo Ceravolo

paolo.ceravolo@unimi.it

Editoria Digitale

- Abbiamo visto che uno dei punti critici del processo di produzione editoriale è rappresentato dall'**acquisizione dei contenuti** e dalla loro **revisione**
- Per permettere che il prodotto sia **percepito come utile** i contenuti devono raggiungere i **bisogni del target** e farlo con un **livello di qualità adeguato**
- Questo porta a elevati costi di produzione dei contenuti in quanto richiedono si deve passare da un **processo creativo** e di **elaborazione manuale**, spesso organizzato attraverso **diverse fasi**

Ideazione

Acquisizione dei contenuti

Revisione e redazione

Progettazione grafica

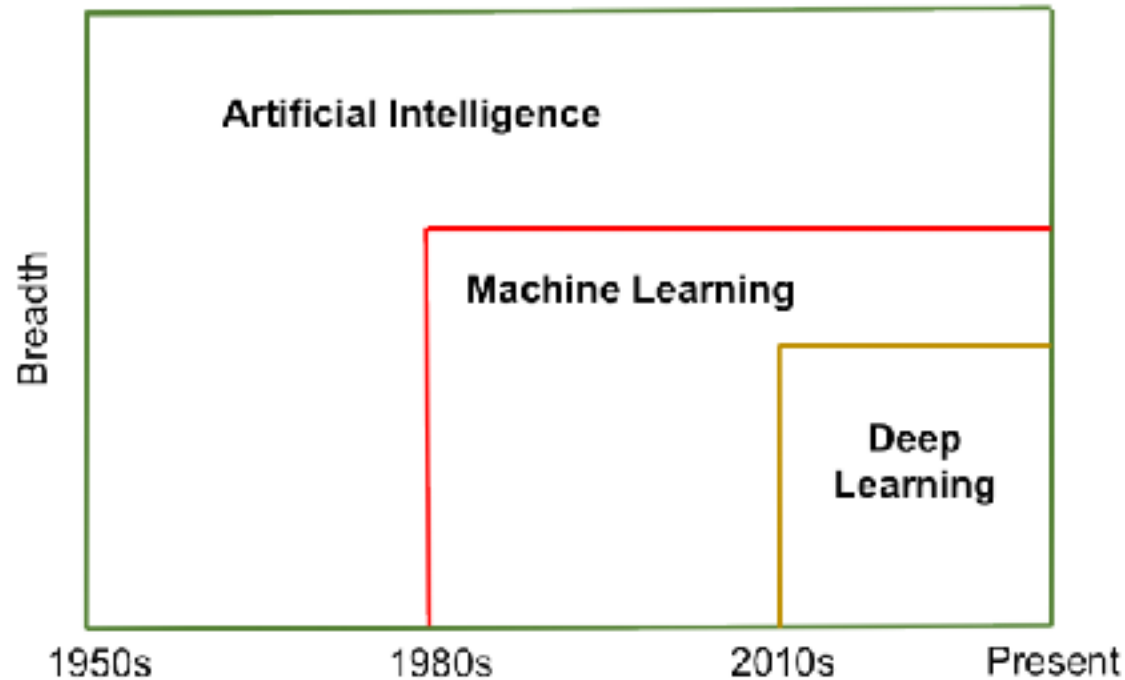
Produzione

Distribuzione

- In generale il processo di acquisizione dei contenuti comporta elevati costi di produzione in quanto richiedono si deve passare da un **processo creativo** e di **elaborazione manuale**
 - In alcuni casi si possono raggiungere delle **fonti aperte** ma questo può eliminare solo una parte del processo di produzione manuale perché si deve passare in ogni caso da **selezione** e **riorganizzazione** del contenuto
- Grazie a strumenti di Intelligenza Artificiale sempre più avanzati, negli ultimi anni si è provato a abbassare il costo di produzione inserendo alcune fasi di **generazione automatica dei contenuti**, che naturalmente non possono sostituire le fasi di selezione e organizzazione del contenuto
- Questa tendenza si è rafforzata recentemente grazie ai successi e alla diffusione di strumenti di **AI generativa**
 - I modelli generativi apprendono la struttura e le caratteristiche di un insieme di dati e sono in grado di generare nuovi dati simili a quelli di partenza
 - Due tipi comuni di generative models sono le Generative Adversarial Networks (GAN) e i Generative Pre-trained Transformer (GPT)

- La diffusione dei Large Language Model (LLM) ha avuto un impatto significativo sull'editoria digitale, consentendo la generazione automatica di contenuti di vario genere
 - **Automazione della scrittura di contenuti:** automazione di una parte della produzione di contenuti editoriali, generalmente i più semplici, come resoconti finanziari, sportivi, descrizioni di prodotti ...
 - **Traduzione automatica di contenuti:** tradurre testi in modo automatico e coerente, generalmente con la supervisione di un traduttore esperto
 - **Revisione di testi:** dato un testo migliorare la fluidità espressiva, modificare lo stile, generare una versione sintetica del testo
 - **Generazione di suggerimenti creativi:** generare suggerimenti creativi per scrivere titoli accattivanti, introdurre storie o creare contenuti coinvolgenti
 - **Creazione di chatbot e assistenza virtuale:** sistemi per rispondere alle domande degli utenti, fornire informazioni e migliorare l'esperienza complessiva dell'utente
 - **Analisi automatica dei contenuti:** alcuni LLM possono essere impiegati per analizzare automaticamente i contenuti digitali, rilevare tendenze, estrarre informazioni chiave e supportare attività di analisi dei dati editoriali

- La diffusione dei sistemi generativi per immagini e i contenuti audio a portato alla sperimentazione di
 - **Generazione di traccia audio da testo:** strumenti di sintesi vocale consentono di convertire un testo nella sua riproduzione vocale
 - **Generazione di illustrazioni per testo:** i sistemi generativi, come le reti neurali generative (GAN), sono stati impiegati con successo per tradurre descrizioni testuali in rappresentazioni visive.
 - **Generazioni di video:** la capacità di generare simultaneamente illustrazioni e tracce audio e testo basate su testo di partenza offre un potenziale significativo per l'integrazione multimediale e la generazione di video
- Recentemente Microsoft e TikTok hanno lanciato due marchi e editoriali [8080 Books](#) e [8th Note Press](#)
- L'obiettivo è la produzione di contenuti editoriali con tempi più rapidi grazie all'uso dell'IA



Artificial Intelligence, Machine Learning e Deep Learning possono essere considerati termini in relazione di iperonimia

Ci limiteremo all'aspetto dell'apprendimento automatico
(*Machine Learning* - ML)

Non per adeguarci alla tendenza promossa dal marketing
dei colossi dell'informatica ma per restringere il campo
del discorso

L'obiettivo di un processo di AU è la costruzione di un
modello di risposta:

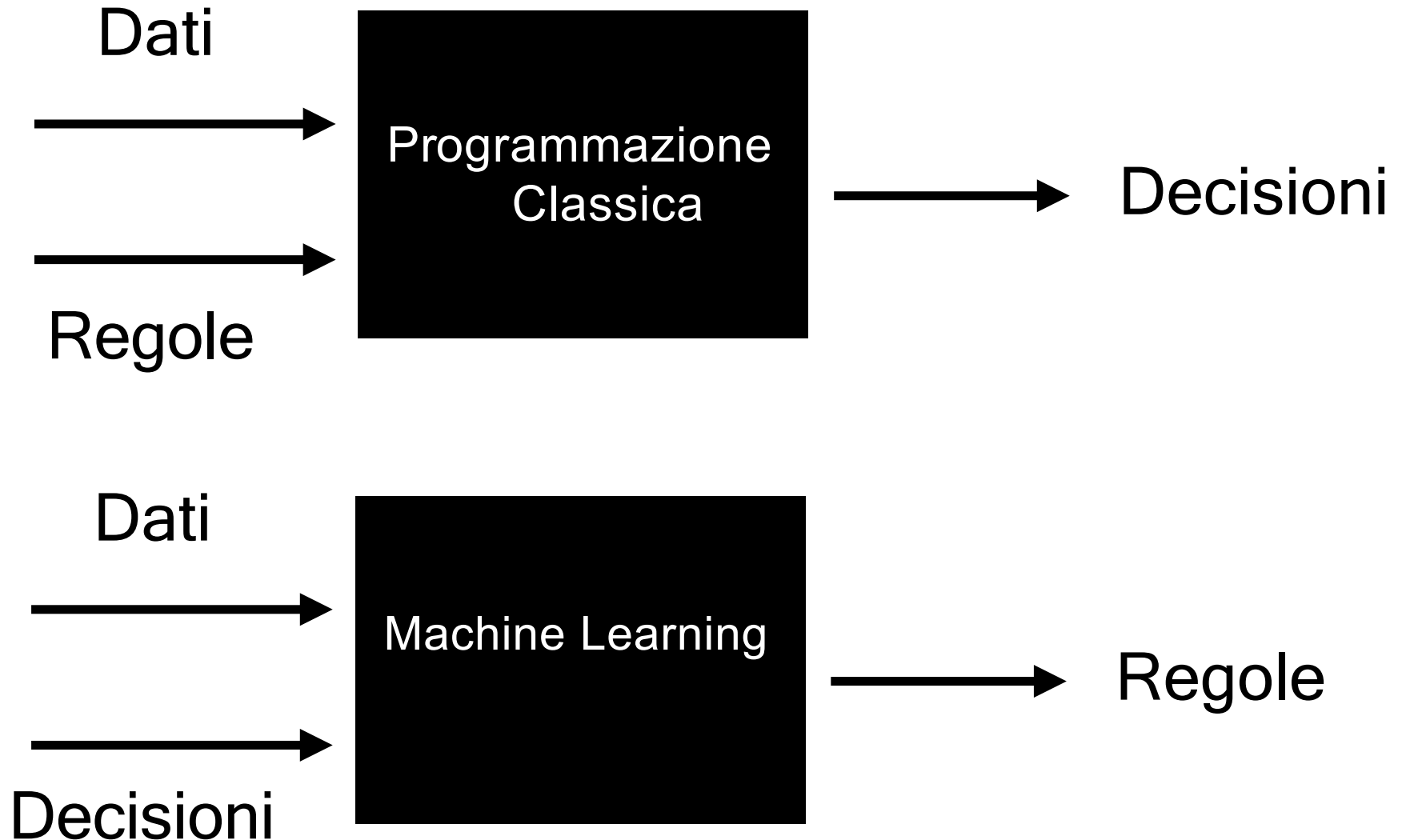
Una procedura che data un tupla X produce una risposta
 y :

$X \rightarrow y$



Esecuzione ML

Studio e
Preparazione
dei Dati



I processi retroattivi o *retroregolati* possono essere considerati la forma più semplice di AU

In un controllo retroattivo il valore di una **variabile in uscita** dal sistema viene letto dal controllore che agisce **modificando l'ingresso** del sistema (ad esempio il termostato, sistema di puntamento)

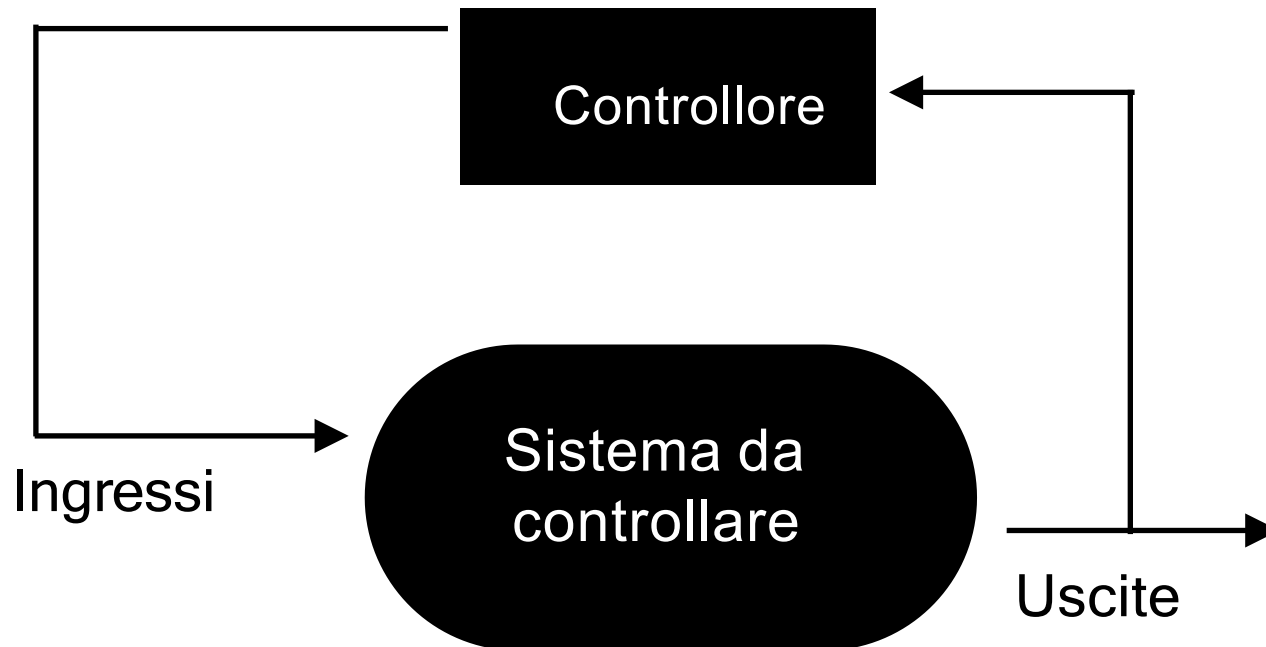
Il concetto è stato introdotto dal matematico americano Norbert Wiener negli **anni quaranta**, iniziatore della *cibernetica*. La teoria dei sistemi retroazionati è utilizzata in molti campi delle scienze pure, delle scienze applicate (tra cui i controlli automatici) e della biologia.

In questo caso il modello è definito inizialmente da un insieme di procedure ma i valori dei punti di decisione sono aggiornati attraverso aggiustamenti progressivi

RETROAZIONE



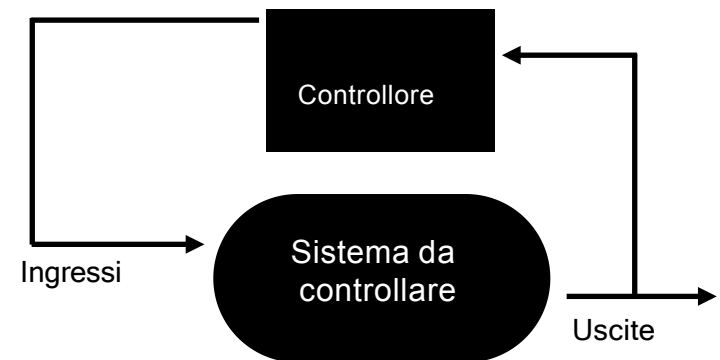
UNIVERSITÀ
DEGLI STUDI
DI MILANO



Il progettista non sempre conosce tutte le regole di decisione

Procedure di apprendimento

- Supervisionato
- Non-supervisionato
- Per rinforzo



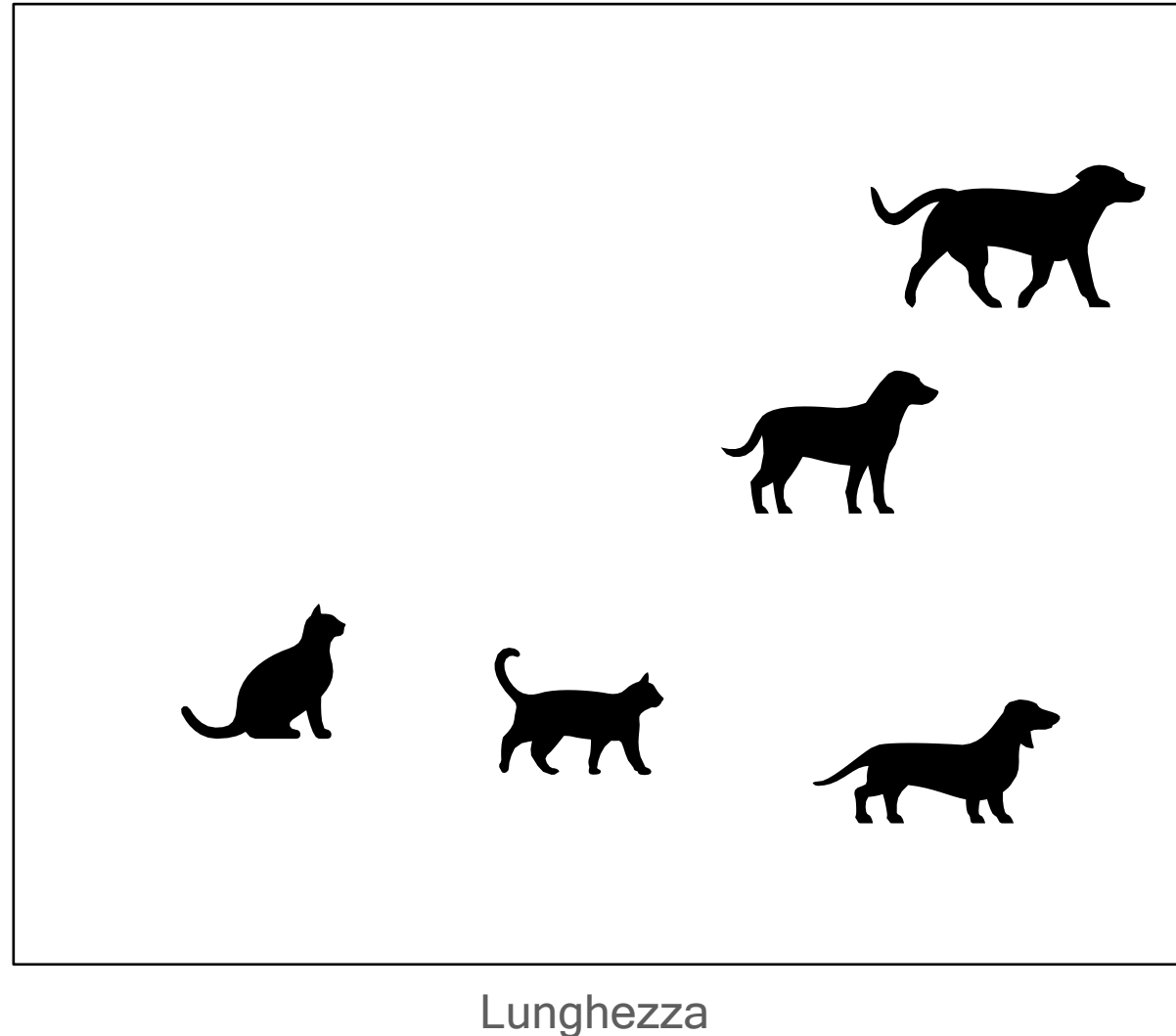
I processi retroattivi si adattano a segnali che provengono dall'ambiente (spesso si assume siano in relazione con gli output del sistema)

Quando i segnali in grado di indirizzare l'adattamento (feedback) non provengono direttamente dall'ambiente è possibile ottenere dei feedback attraverso un processo di valutazione nel quale i risultati prodotti dalla macchina siano valutati confrontandoli con risultati che si conosce essere corretti

Questa nozione è alla base dei processi di **apprendimento supervisionato**

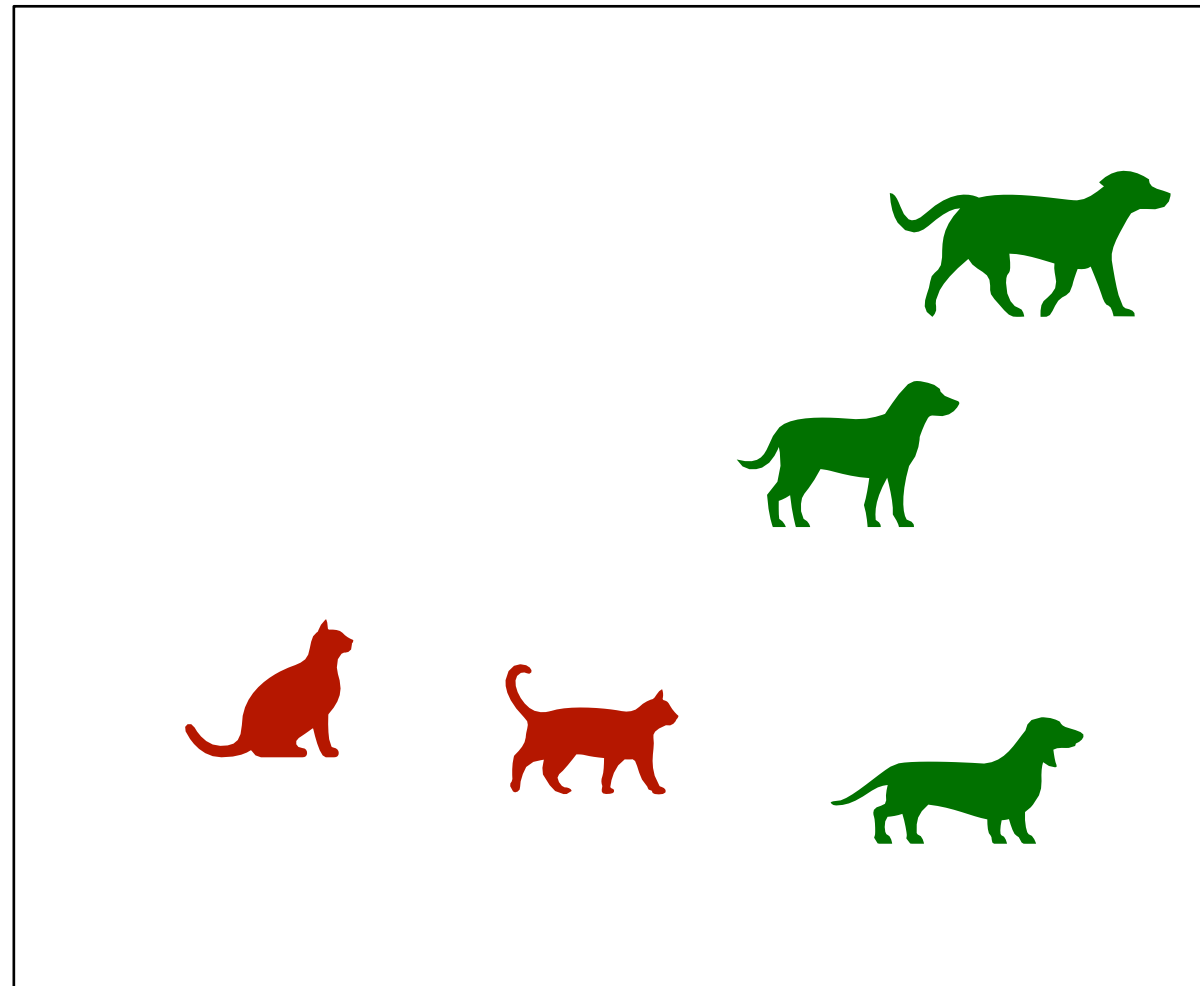
- ▶ Apprendimento Supervisionato
 - ▶ Induzione
 - ▶ Transduzione

Altezza



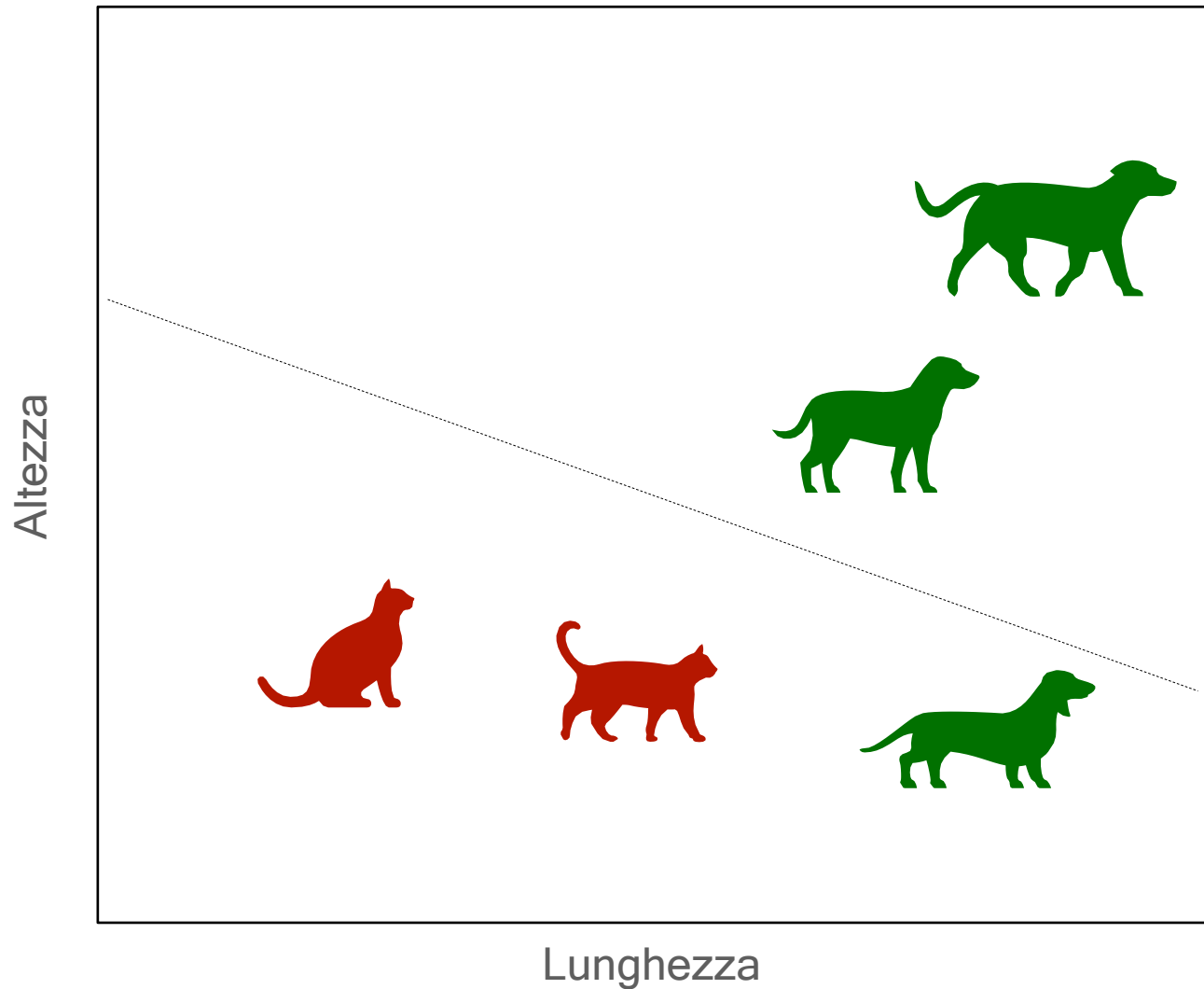
- ▶ Apprendimento Supervisionato
 - ▶ Induzione
 - ▶ Transduzione

Altezza

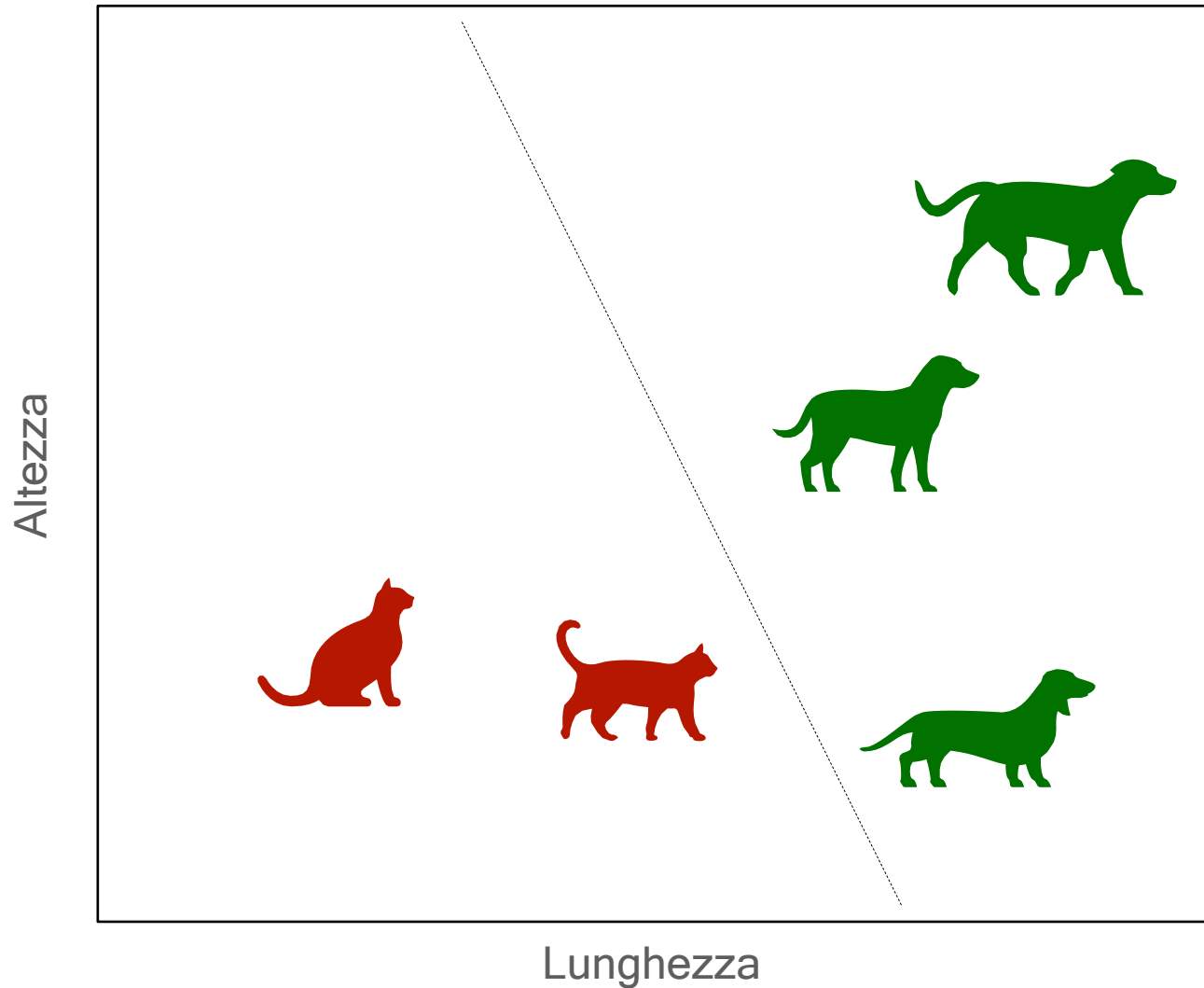


Lunghezza

- ▶ Apprendimento Supervisionato
 - ▶ Induzione
 - ▶ Transduzione



- ▶ Apprendimento Supervisionato
 - ▶ Induzione
 - ▶ Transduzione



Classificazione

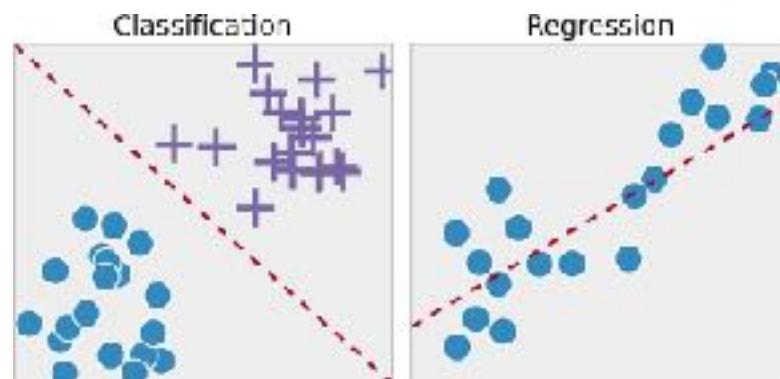
Riconoscere un insieme di esempi accumulati da stesse proprietà

Predire dati categorici

Regressione

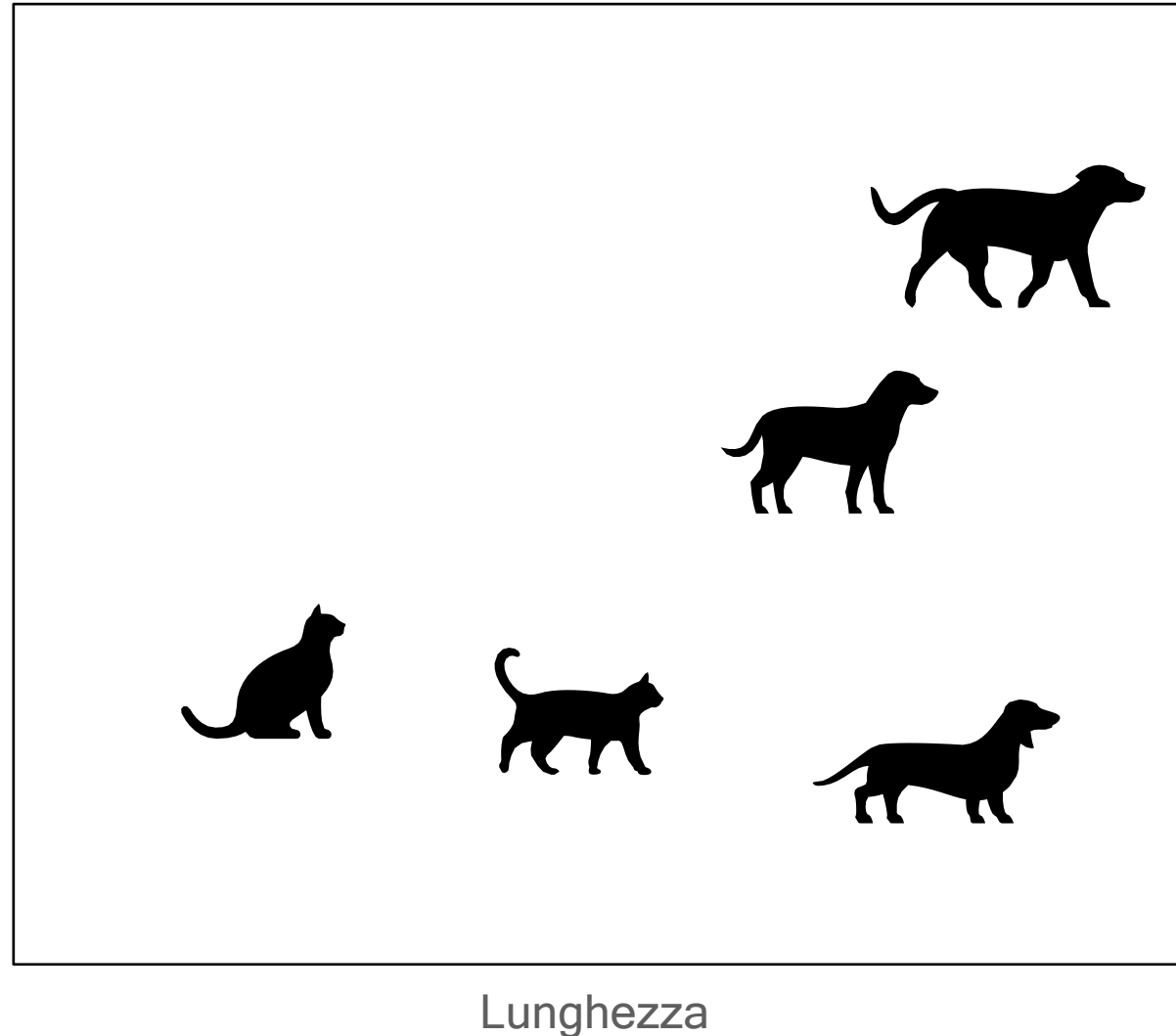
Individuare la tendenza di evoluzione di una distribuzione

Predire dati numerici



- ▶ Apprendimento Non Supervisionato
 - ▶ Induzione
 - ▶ Deduzione

Altezza



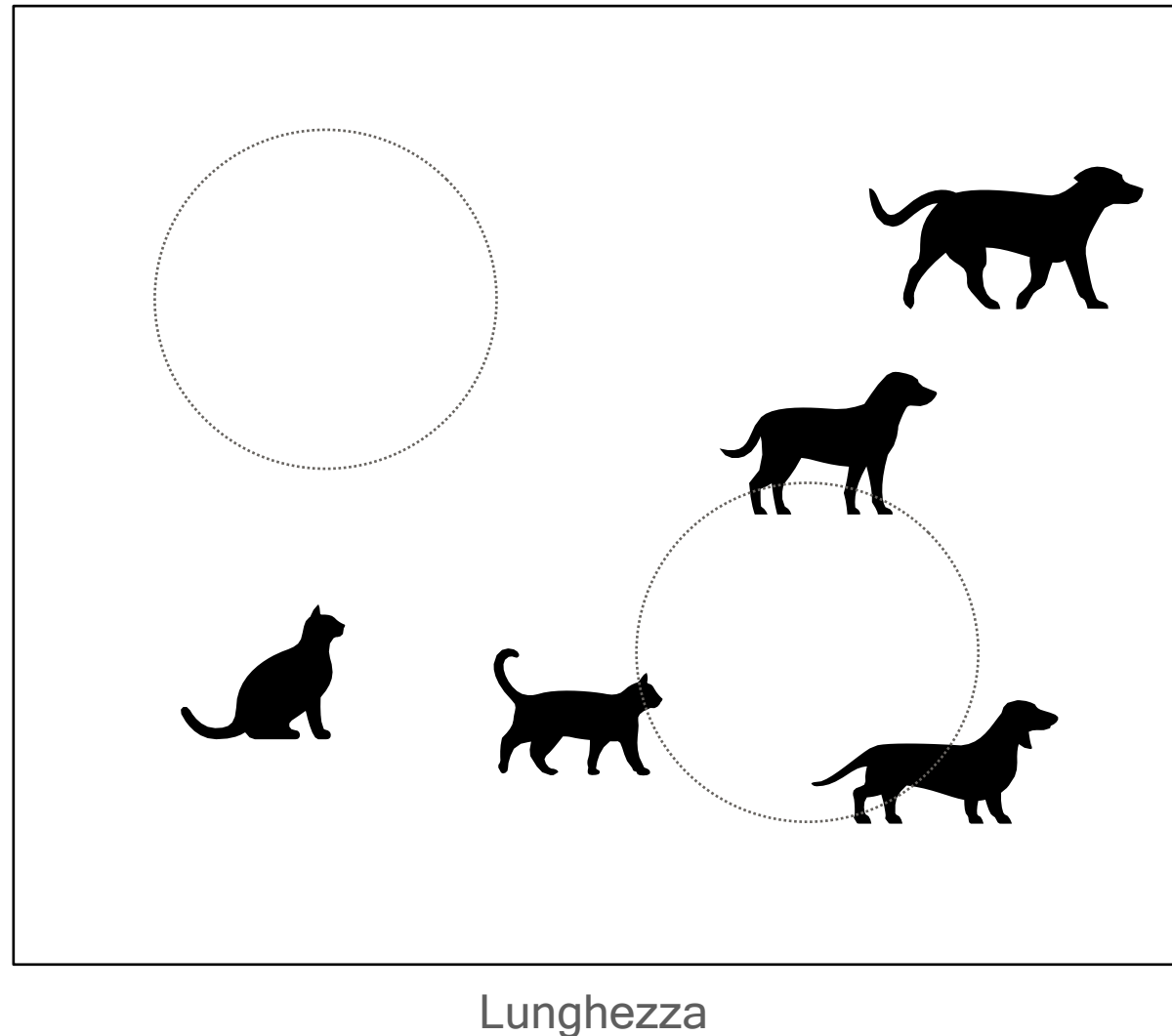
APPRENDIMENTO NON SUPERVISIONATO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

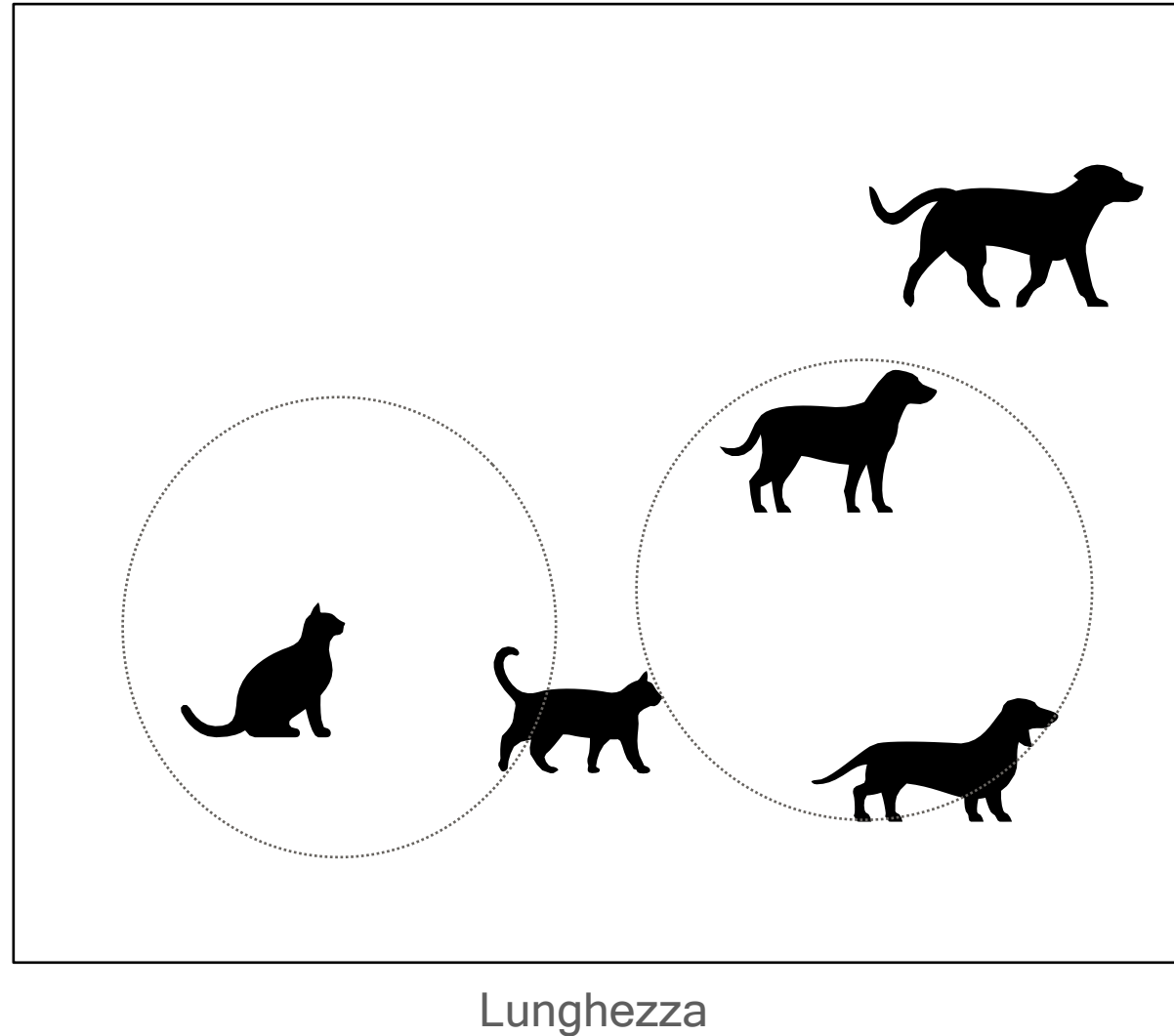
- ▶ Apprendimento Non Supervisionato
 - ▶ Induzione
 - ▶ Deduzione

Altezza



- ▶ Apprendimento Non Supervisionato
 - ▶ Induzione
 - ▶ Deduzione

Altezza



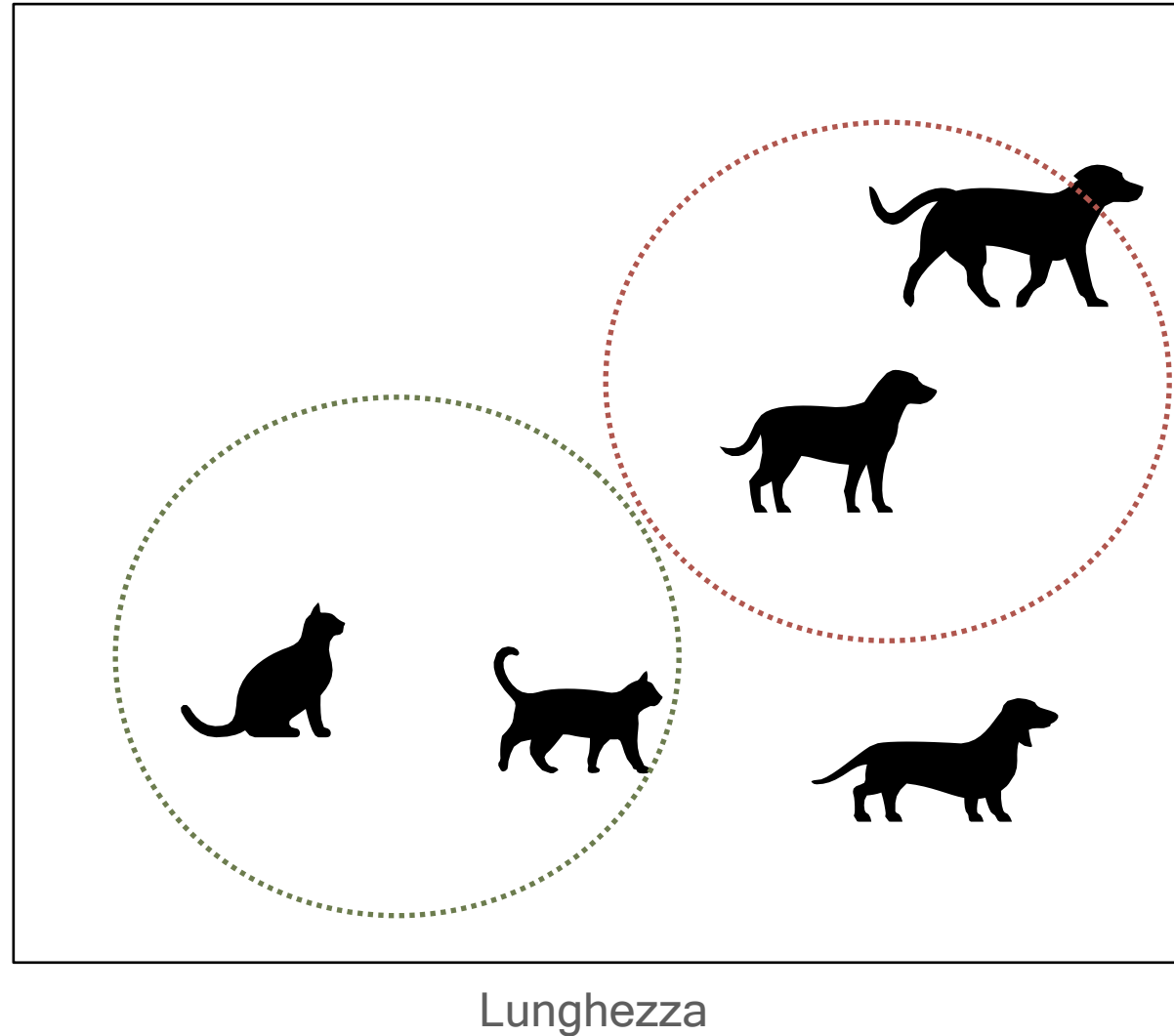
APPRENDIMENTO NON SUPERVISIONATO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- ▶ Apprendimento Non Supervisionato
 - ▶ Induzione
 - ▶ Deduzione

Altezza

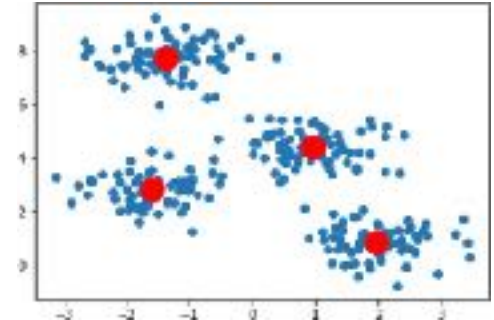


FUNZIONI DELL'APPRENDIMENTO NON SUPERVISIONATO

Clustering

Raggruppare esempi accumulati da stesse proprietà

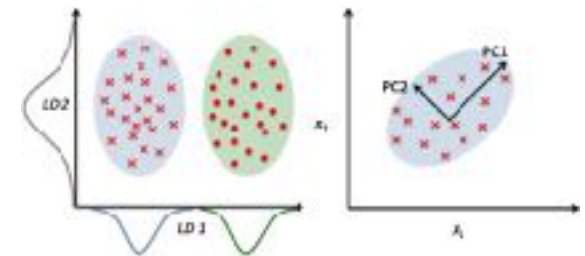
Dati categorici / dati numerici



Dimension Reduction

Individuare la dimensioni di maggior importanza in un dataset

Dati categorici / dati numerici



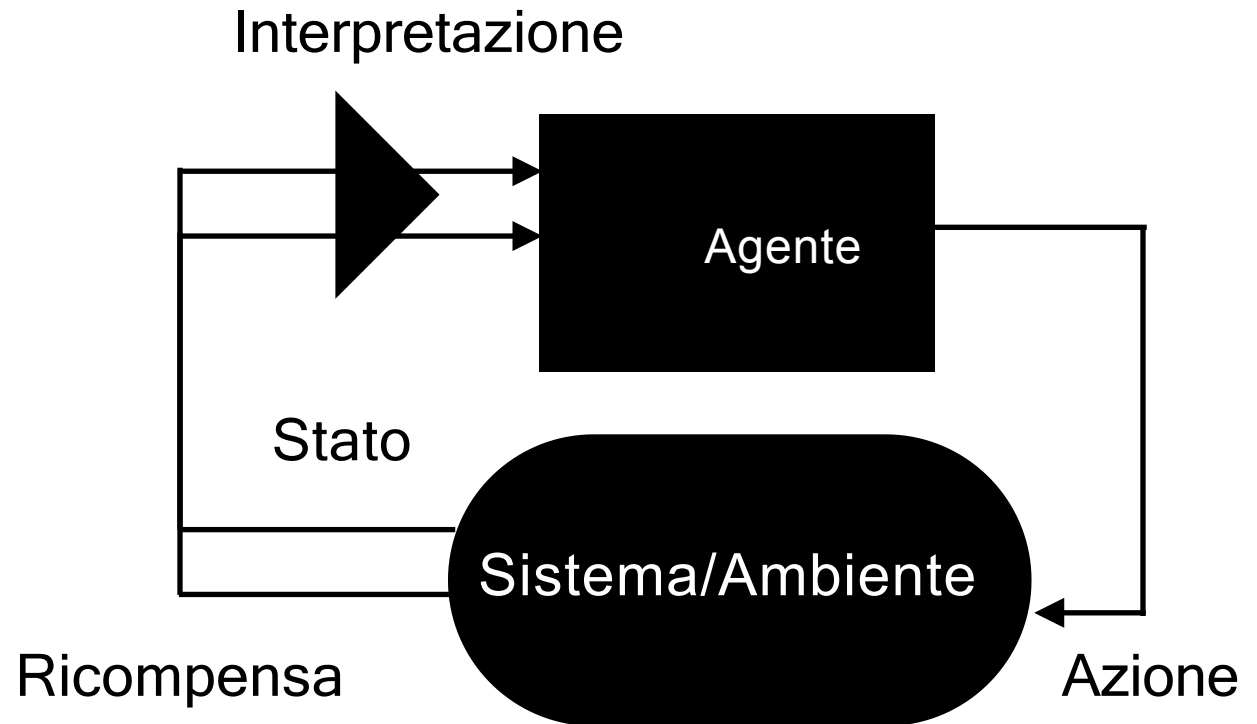
Association

Individuare correlazioni e probabilità condizionate di un dataset

Transaction 1	   
Transaction 2	  
Transaction 3	 
Transaction 4	 
Transaction 5	   
Transaction 6	  
Transaction 7	 
Transaction 8	 

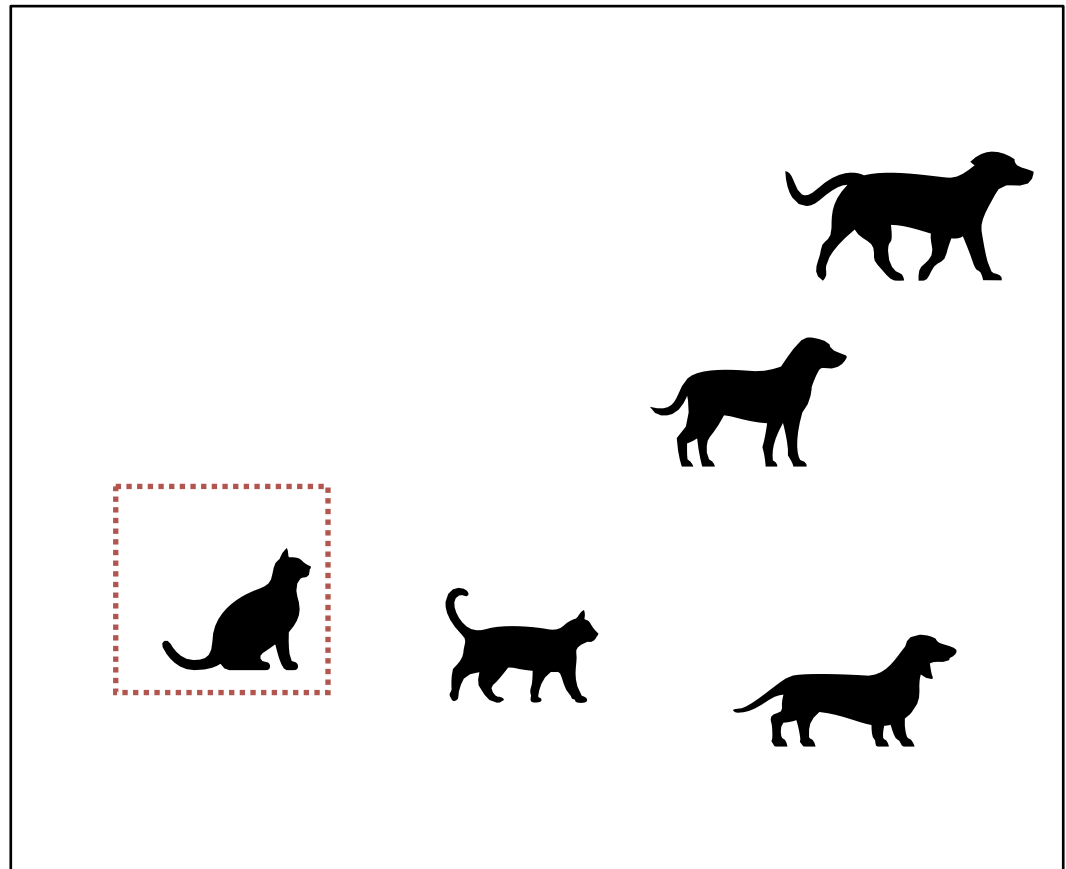
- ▶ Apprendimento Per Rinforzo

- ▶ Azione
- ▶ Stato
- ▶ Ricompensa



- ▶ Apprendimento Per Rinforzo
 - ▶ Azione
 - ▶ Stato
 - ▶ Ricompensa

Altezza



Lunghezza

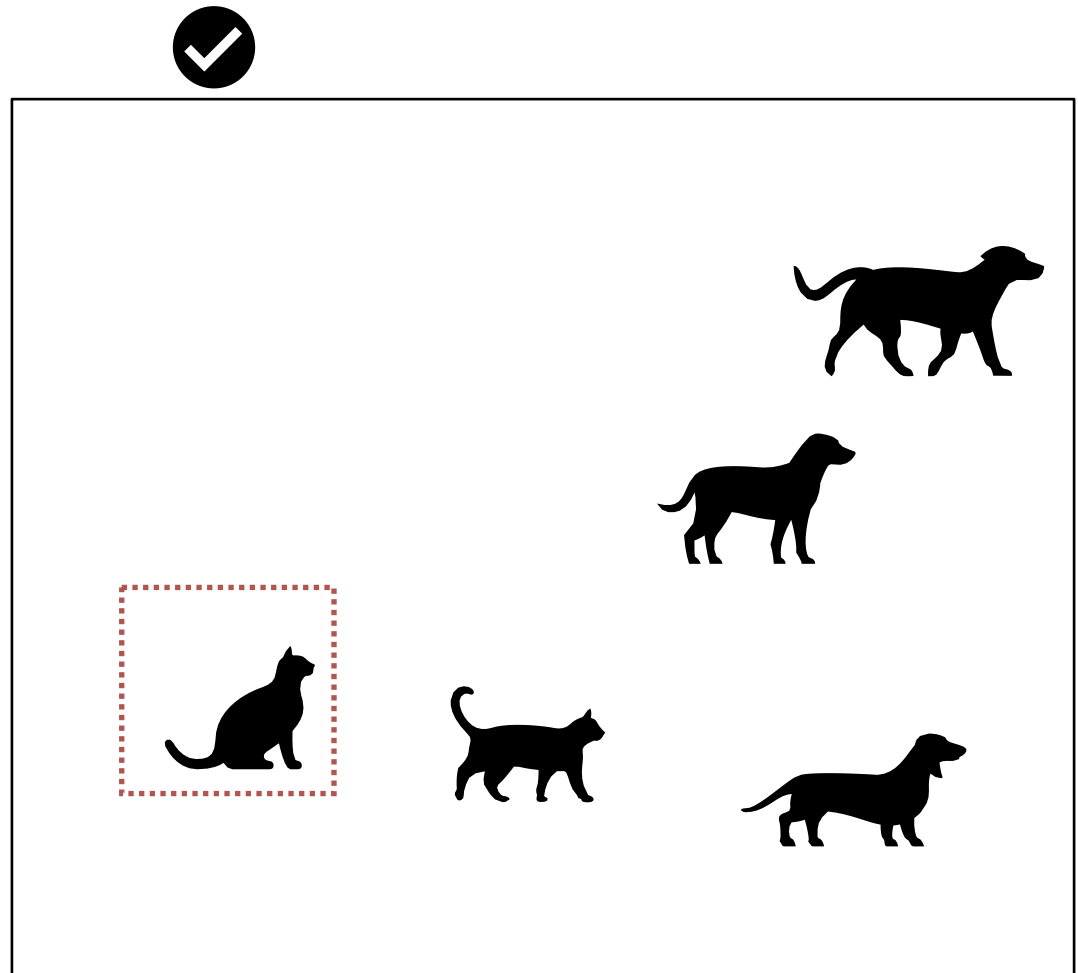
APPRENDIMENTO PER RINFORZO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- ▶ Apprendimento Per Rinforzo
 - ▶ Azione
 - ▶ Stato
 - ▶ Ricompensa

Altezza



Lunghezza

APPRENDIMENTO PER RINFORZO

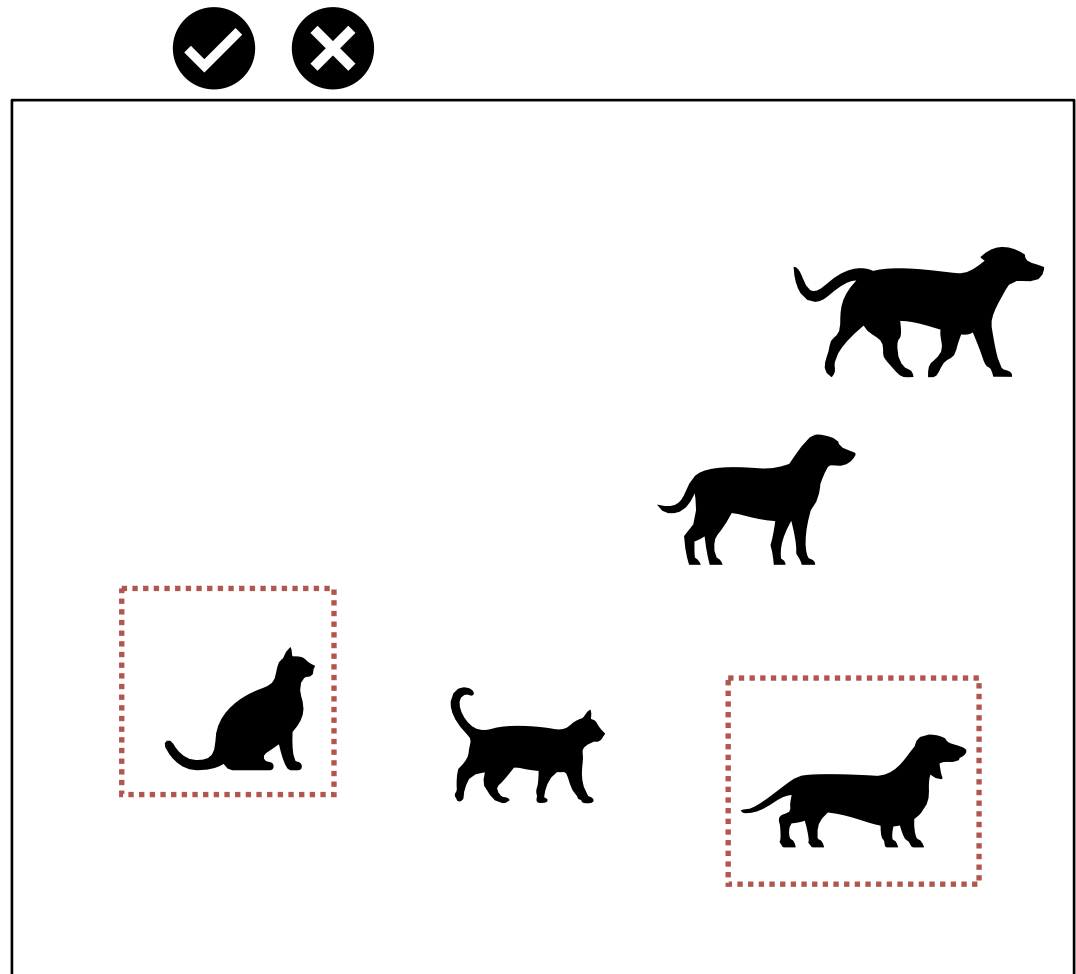


UNIVERSITÀ
DEGLI STUDI
DI MILANO

► Apprendimento Per Rinforzo

- Azione
- Stato
- Ricompensa

Altezza



Lunghezza

APPRENDIMENTO PER RINFORZO

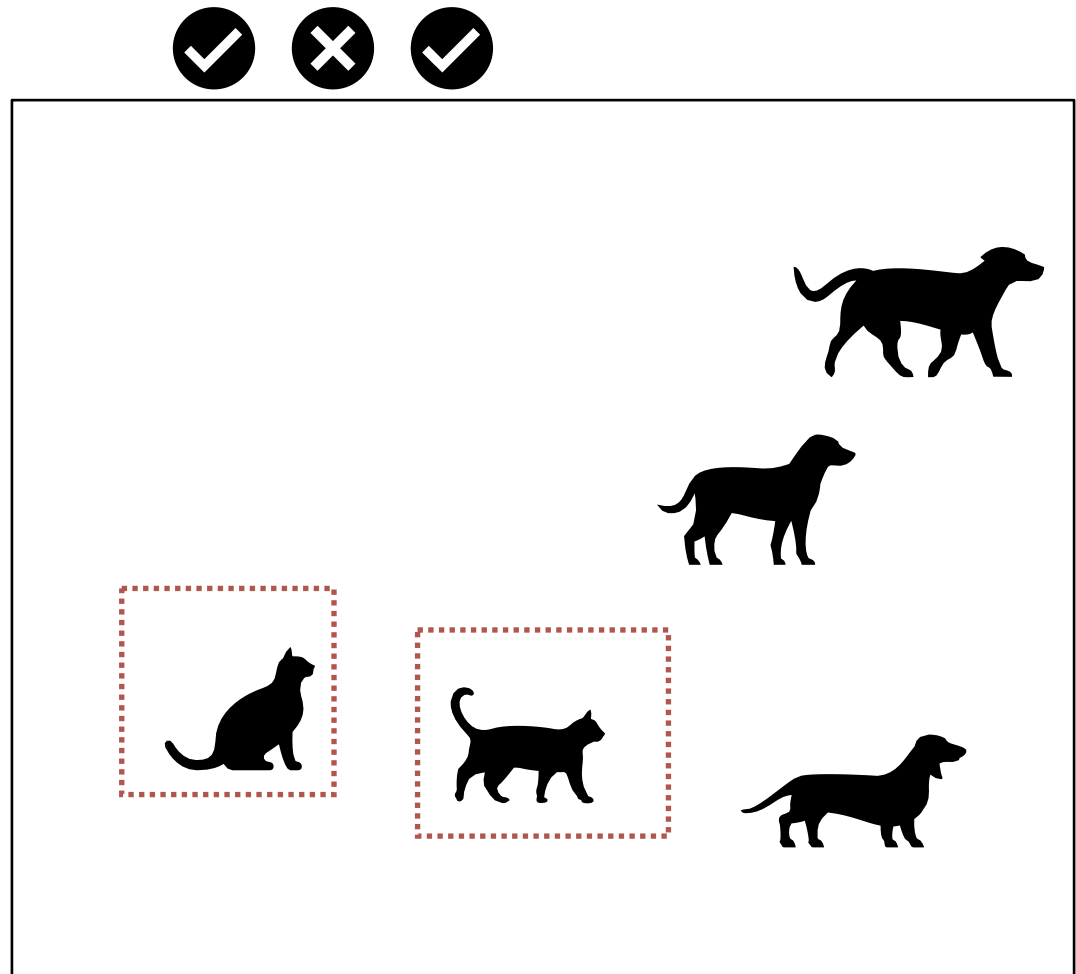


UNIVERSITÀ
DEGLI STUDI
DI MILANO

► Apprendimento Per Rinforzo

- Azione
- Stato
- Ricompensa

Altezza



Lunghezza

Nell'apprendimento auto-supervisionato *self-supervised learning*, l'obiettivo è sfruttare informazioni intrinseche presenti nei dati stessi per addestrare un modello, senza l'utilizzo di etichette fornite esternamente

Un esempio di self-supervised learning è la predizione di parte di un'istanza a partire dalle altre parti di quella stessa istanza

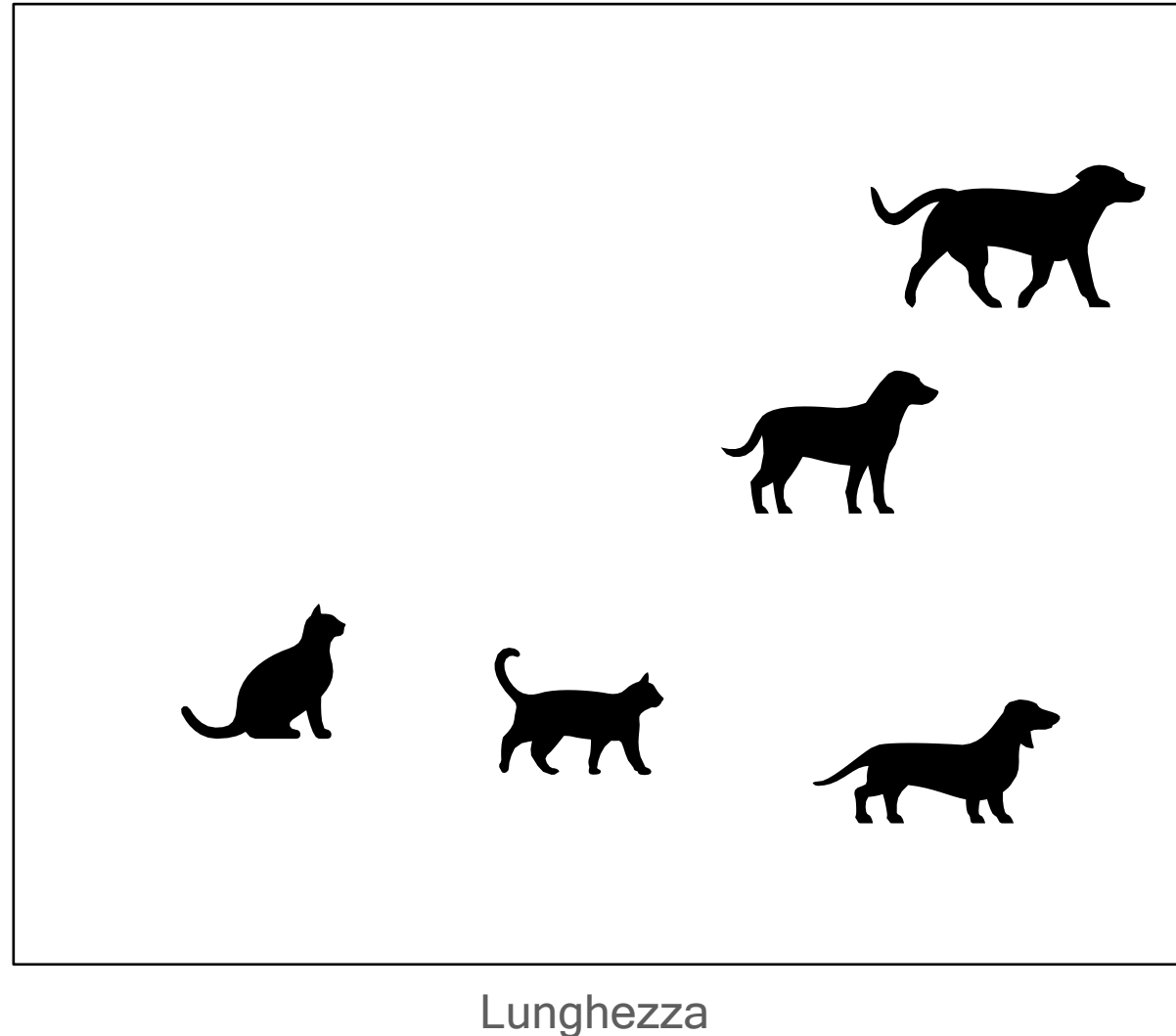
Supponiamo di avere immagini di cani e gatti, possiamo suddividere ogni immagine in due parti: la testa e il corpo dell'animale

Quindi, creiamo un compito di predizione dove il modello deve imparare a predire la parte mancante a partire dalla parte visibile

Approccio molto usato molto nei modelli linguistici per predire la parola successiva dato una sequenza di parole

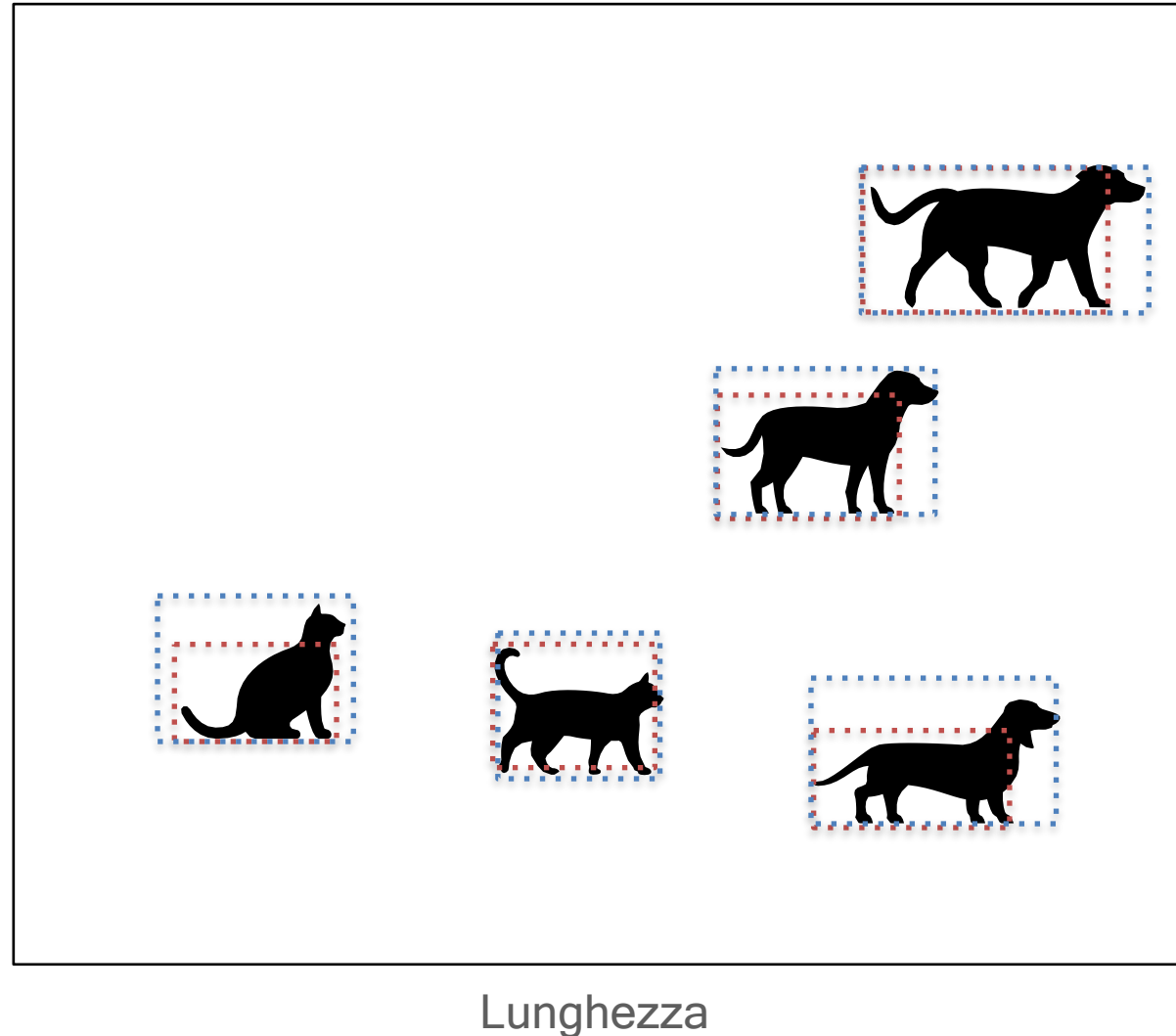
- ▶ Apprendimento Auto-supervisionato
 - ▶ Simulazione

Altezza



- ▶ Apprendimento Auto-supervisionato
 - ▶ Simulazione

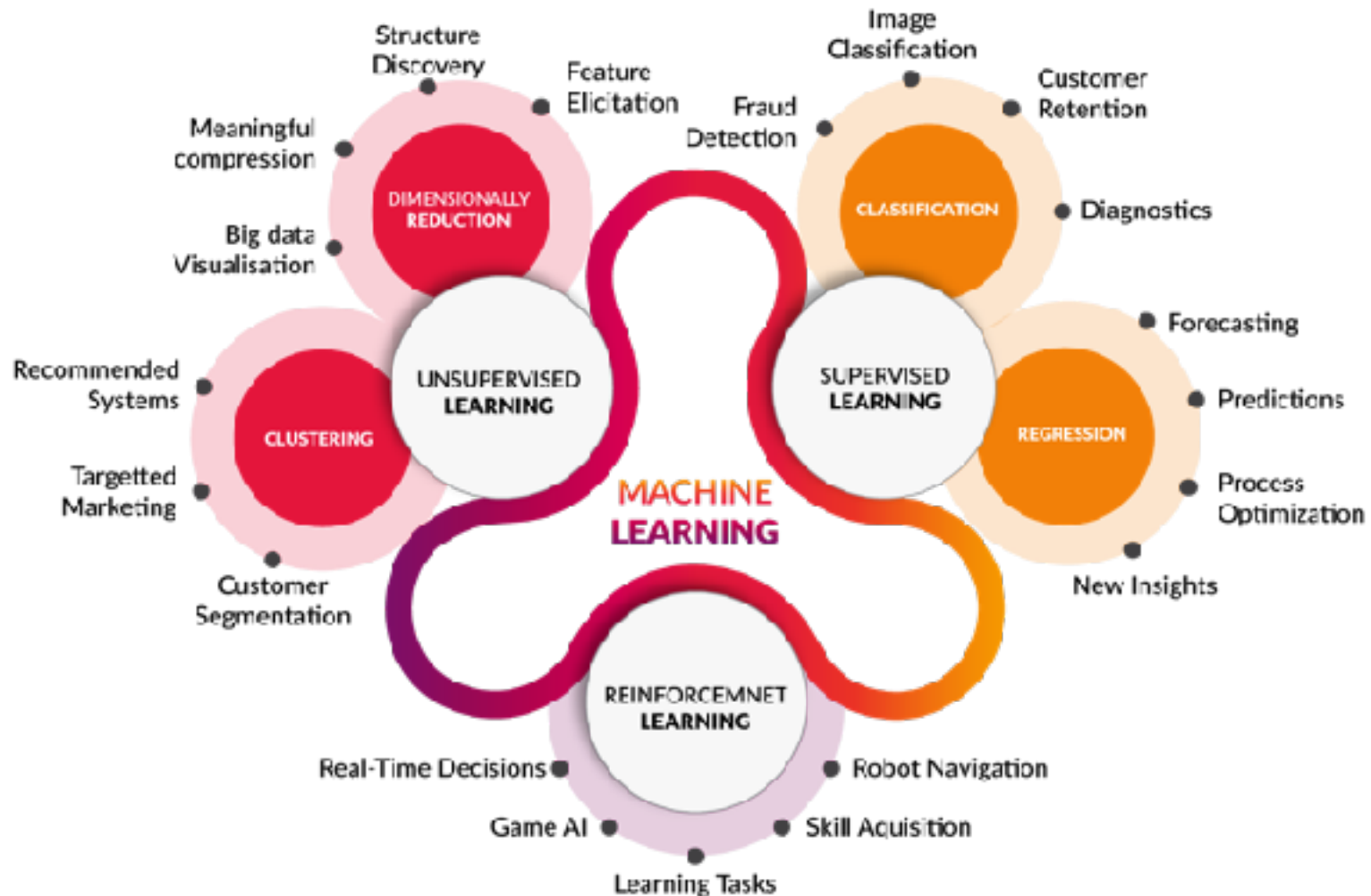
Altezza



PROCEDURE DI APPRENDIMENTO

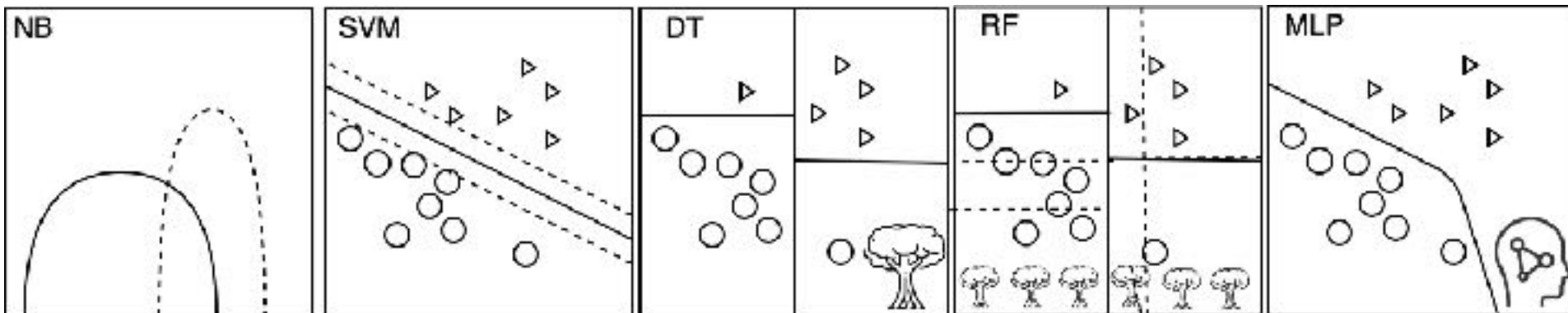


UNIVERSITÀ
DEGLI STUDI
DI MILANO



- ▶ **Active Learning:** l'apprendimento attivo è una tecnica in cui il modello è in grado di interrogare un operatore umano durante il processo di apprendimento al fine di risolvere possibili ambiguità. È utile quando non ci sono molti dati disponibili o i dati sono costosi da raccogliere o etichettare
- ▶ **Multi-Task Learning:** si riferisce a un processo di apprendimento che può essere condiviso da più agenti. Per esempio, la codifica della distribuzione di parole nel testo può essere condivisa tra più compiti NLP
- ▶ **Online Learning:** si riferisce ad algoritmi che sono in grado di apprendere da flussi continui di dati. Questi algoritmi apprendono in modo incrementale, senza la necessità di conoscere l'intero insieme dei dati, quindi con la possibilità di cancellare i dati qualora non ci fosse più spazio in memoria
- ▶ **Transfer Learning:** l'apprendimento per trasferimento è un tipo di apprendimento in cui un modello viene prima addestrato su un compito, poi una parte o tutto il modello viene usato come punto di partenza per un compito correlato
- ▶ **Ensemble Learning:** un approccio in cui due o più modalità sono adattate agli stessi dati e le previsioni di ogni modello sono combinate. L'obiettivo è quello di migliorare le prestazioni rispetto all'uso di un singolo modello

- ▶ Naive Bayes
- ▶ Support Vector Machine
- ▶ Decision Tree
- ▶ Random Forest
- ▶ Multi-Layer Perceptron



Nel 1958 Frank Rosenblatt propose un algoritmo ispirato al comportamento delle sinapsi che chiamò **percettrone**

Si tratta di entità con uno strato di ingresso ed uno di uscita ed una regola di apprendimento basata sulla minimizzazione dell'errore che in base alla valutazione sull'uscita effettiva della rete rispetto ad un dato ingresso altera i pesi delle connessioni (sinapsi) come differenza tra l'uscita effettiva e quella desiderata.

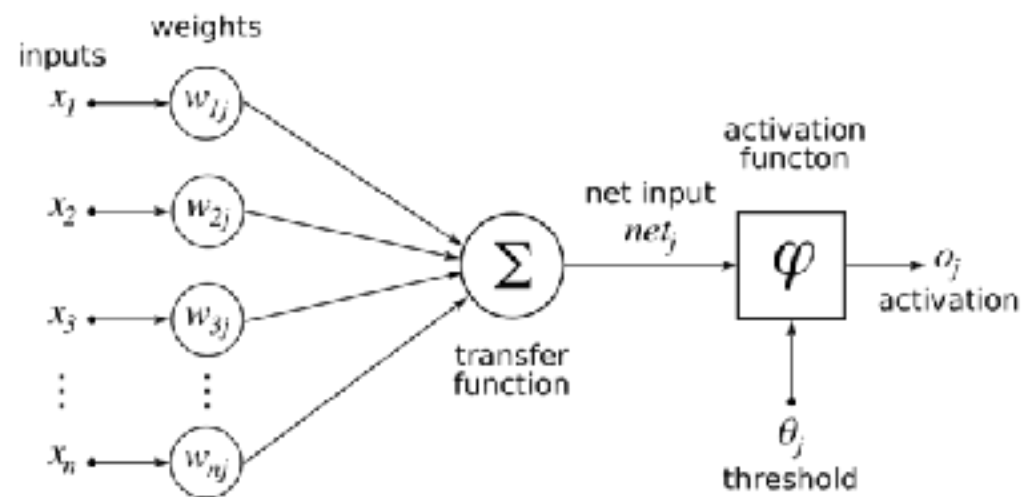
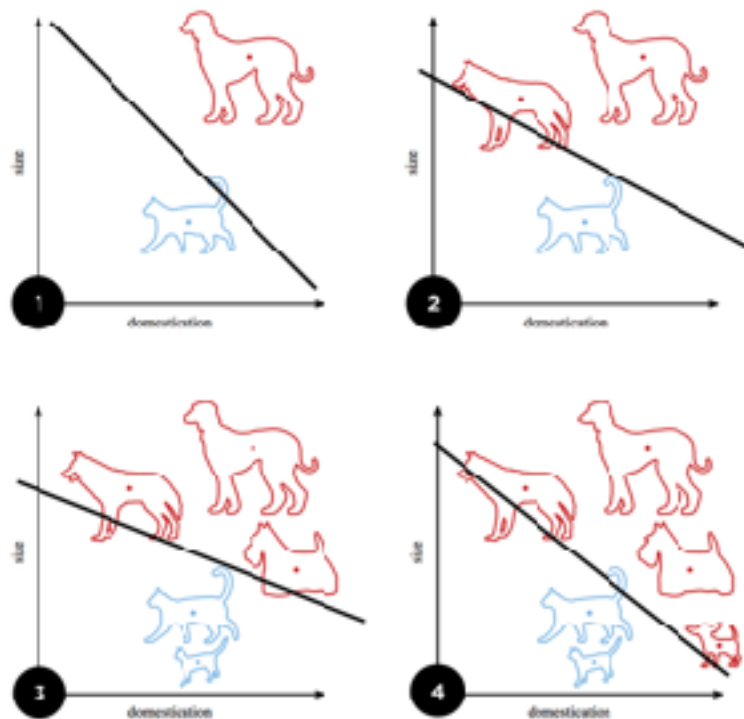
Ci fu un iniziale entusiasmo ma poco dopo Marvin Minsky e Seymour Papert dimostrarono i limiti del percettrone e cioè la sua capacità di riconoscere **solamente funzioni linearmente separabili** (ad esempio la funzione logica XOR non può essere implementata da un percettrone).

Seguì quindi una fase nella quale questo approccio fu poco studiato.

PERCETTRONE



UNIVERSITÀ
DEGLI STUDI
DI MILANO



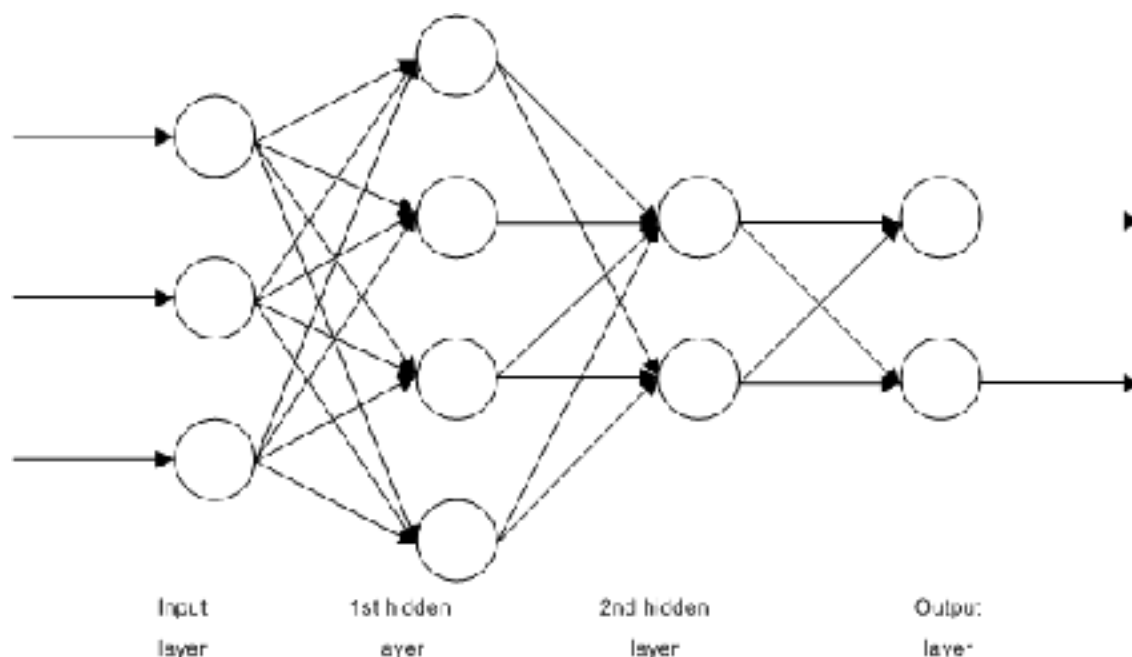
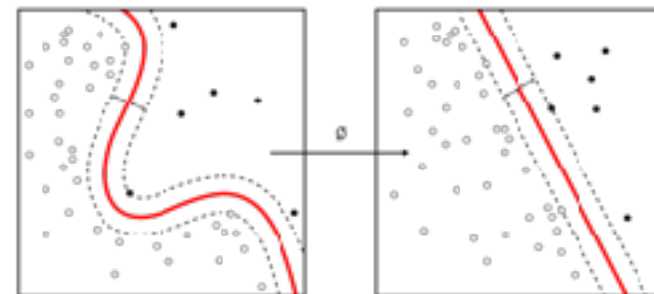
Animazione:

L'algoritmo di apprendimento standard è un algoritmo iterativo che ad ogni iterazione calcola l'output del percettrone e lo confronta con il risultato desiderato quindi, il vettore dei pesi viene aggiornato come segue:

$$w^{t+1} = w^t + \alpha (g(x^t) - f(x^t)) x^t$$

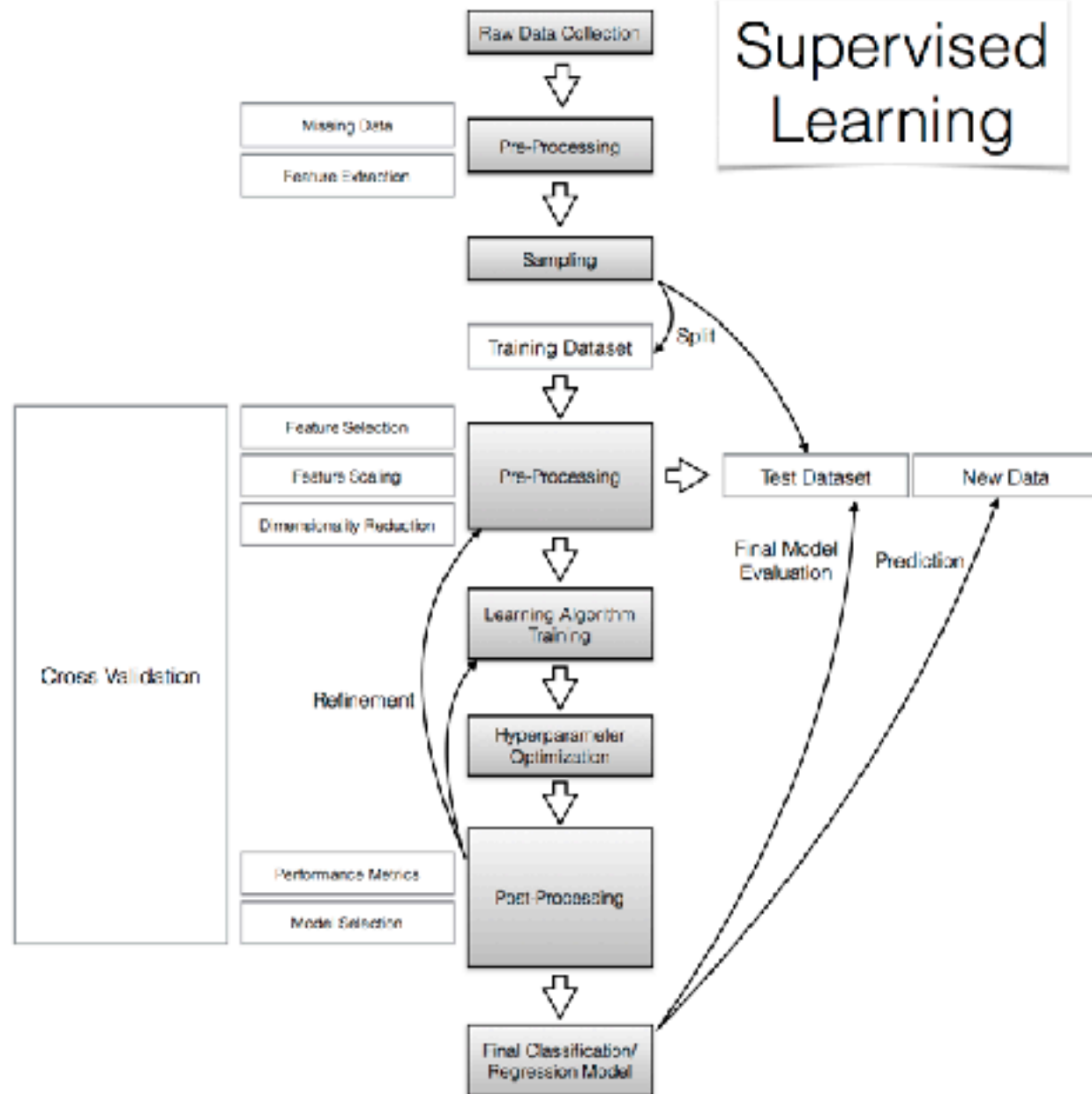
Dove $g(x)$ è il risultato desiderato, $f(x)$ è il risultato ottenuto e α è una costante che regola la velocità dell'apprendimento

Organizzando i percettroni in una rete a più strati (strati nascosti) è possibile fornire funzioni di risposta non lineari



- L'apprendimento dei pesi di una rete neurale è un processo lento e debolmente informato
- Ogni esempio corretto aggiusta un po' i pesi finché non raggiungiamo un'accuratezza accettabile

Supervised Learning

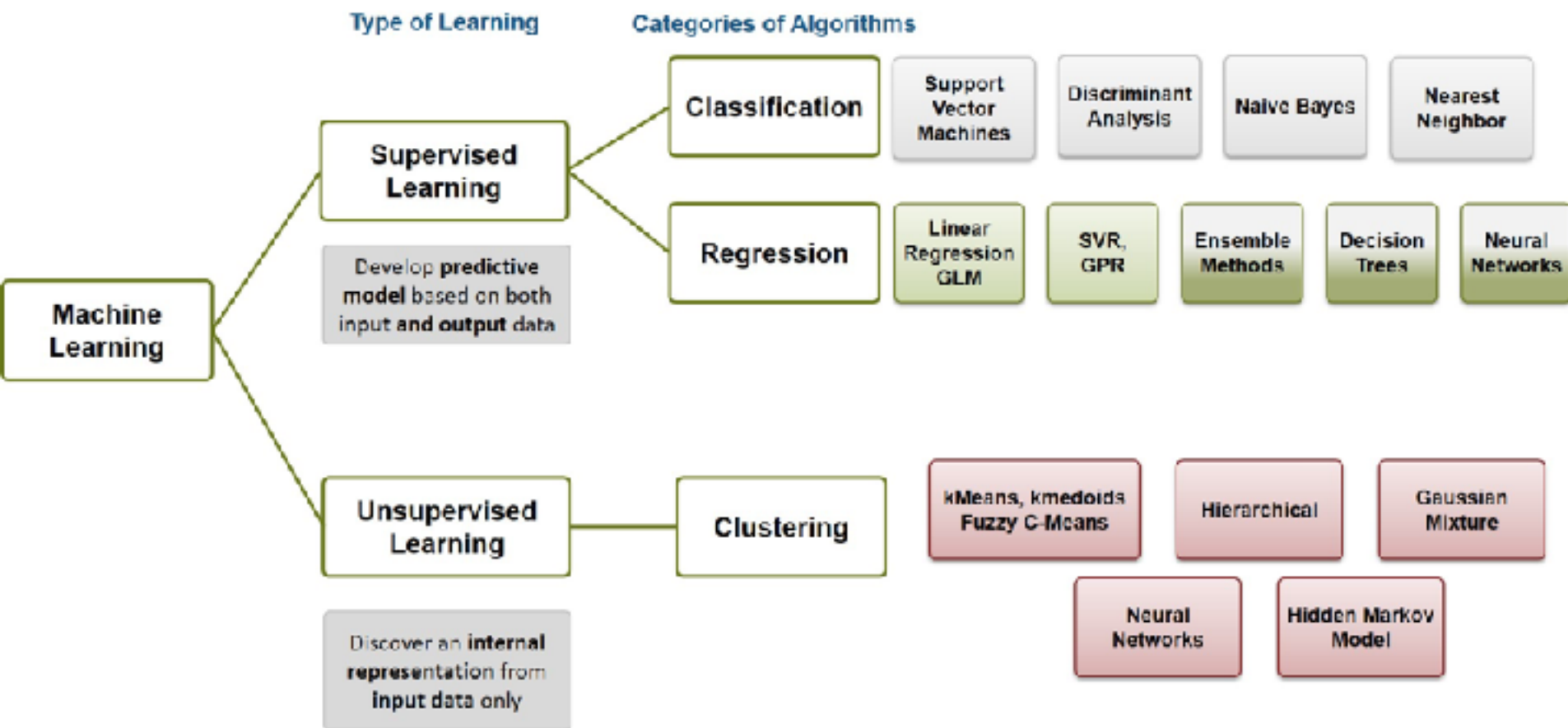


UNA TASSONOMIA DEL AU



UNIVERSITÀ
DEGLI STUDI
DI MILANO

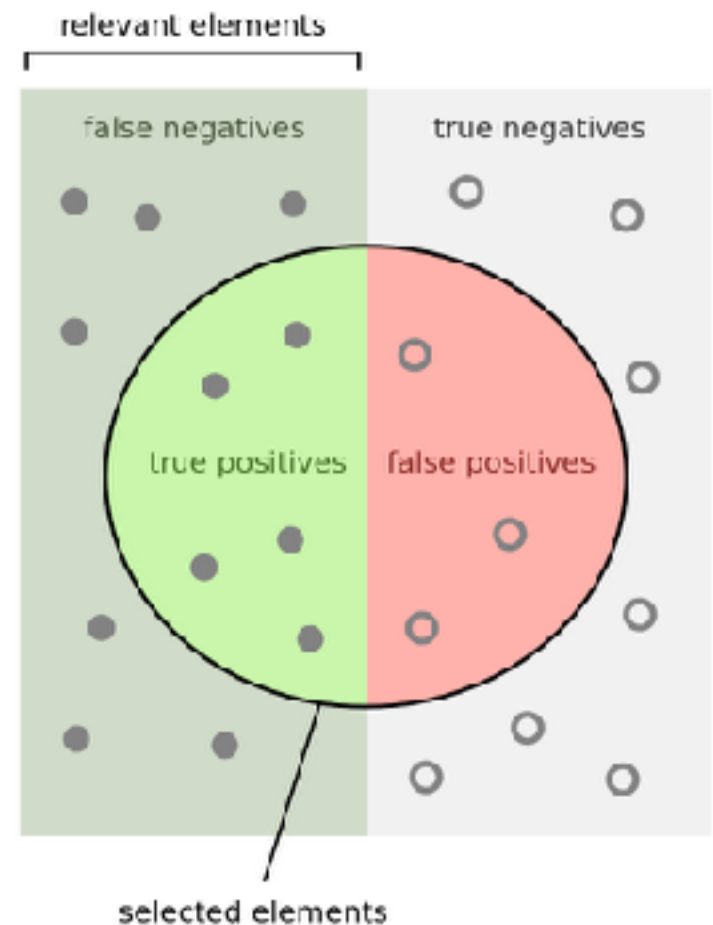
- Se aggiungiamo dati al contesto molte soluzioni prodotte non sono più valide



VALIDAZIONE

Precision (precisione) e *Recall* (recupero o richiamo), sono due comuni metriche di qualità di un sistema predittivo

La precisione può essere vista come una misura di esattezza o fedeltà, mentre il recupero è una misura di completezza



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

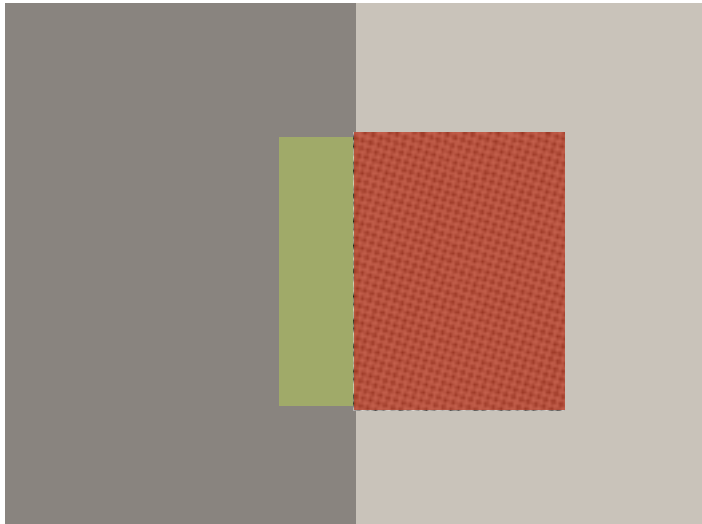
How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

VALIDAZIONE: PRECISION E RECALL

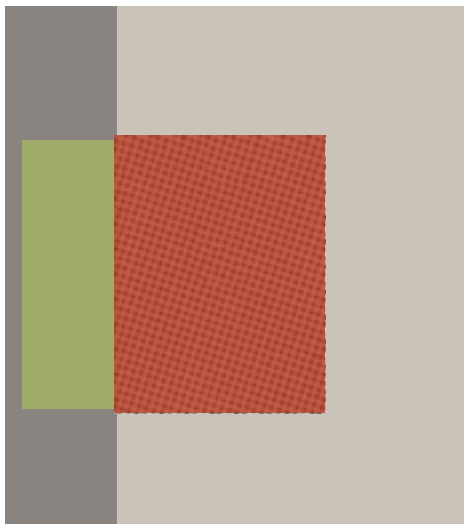


UNIVERSITÀ
DEGLI STUDI
DI MILANO



Precision bassa

Recall bassa



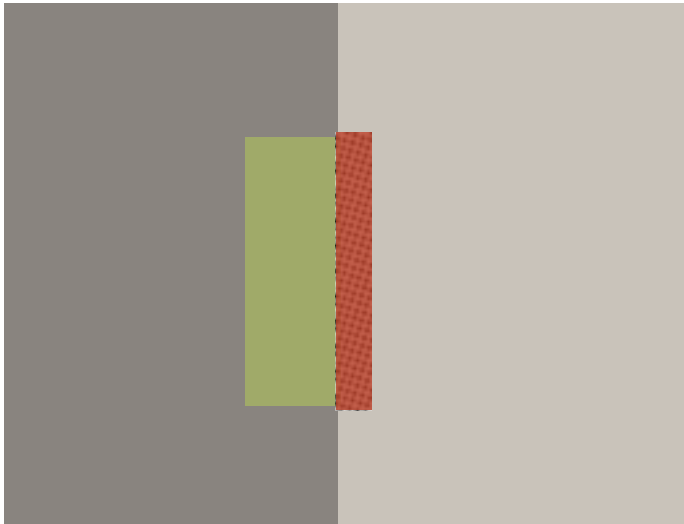
Precision bassa

Recall alta

VALIDAZIONE: PRECISION E RECALL

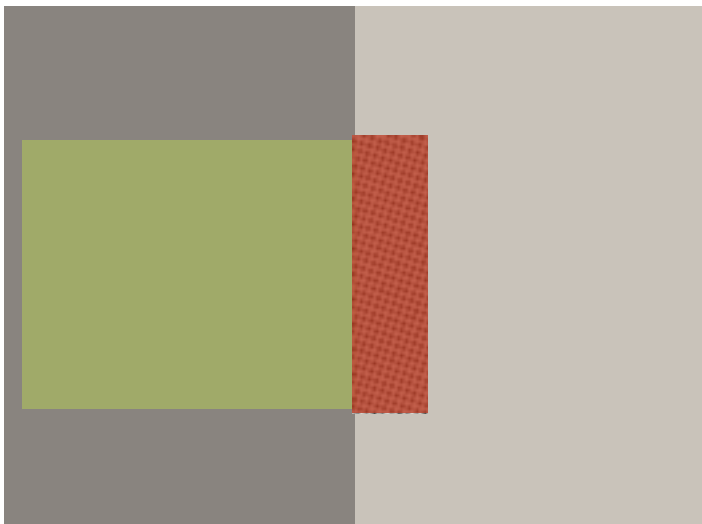


UNIVERSITÀ
DEGLI STUDI
DI MILANO



Precision alta

Recall bassa



Precision alta

Recall alta

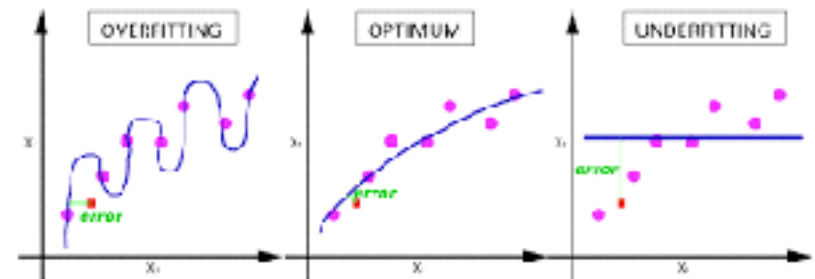
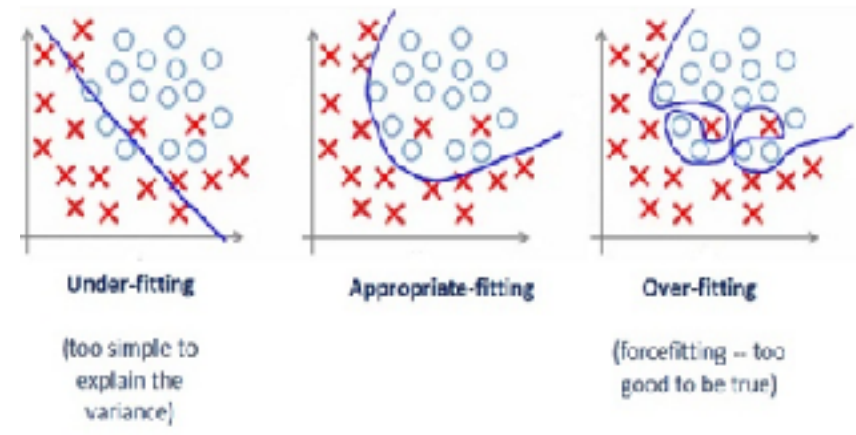
VALIDAZIONE: OVERFITTING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Nel valutare il grado di generalità del modello appreso:

- Il modello si dice sotto specificato (*underfitting*) se *Precision* e/o *Recall* sono basse
- Il modello si dice sovra specificato (*overfitting*) se è troppo legato agli esempi osservati, quindi non funzionerà correttamente con dati di test diversi dai dati usati nella fase di addestramento

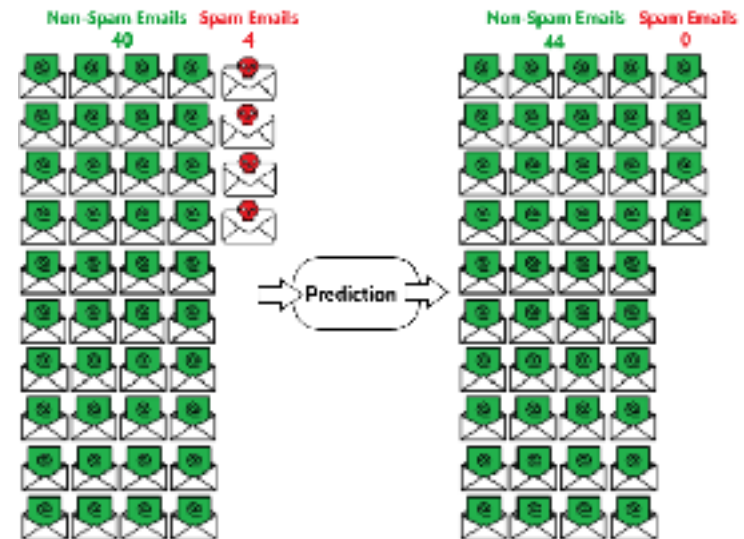


VALIDAZIONE: CONFRONTO



Per valutare la qualità del modello è sempre necessario identificare un termine di paragone, un baseline model

- Il modello sarà di qualità se la sua accuratezza sarà superiore alla baseline
- La Zero Rule o ZeroR è una procedura di riferimento per gli algoritmi di classificazione il cui risultato è semplicemente la frequenza della classe più frequente nei dati. Se il 65% degli elementi di un dataset appartiene a una classe, ZeroR predice per tutti gli elementi del dataset quella classe. La precision sarà quindi del 65%



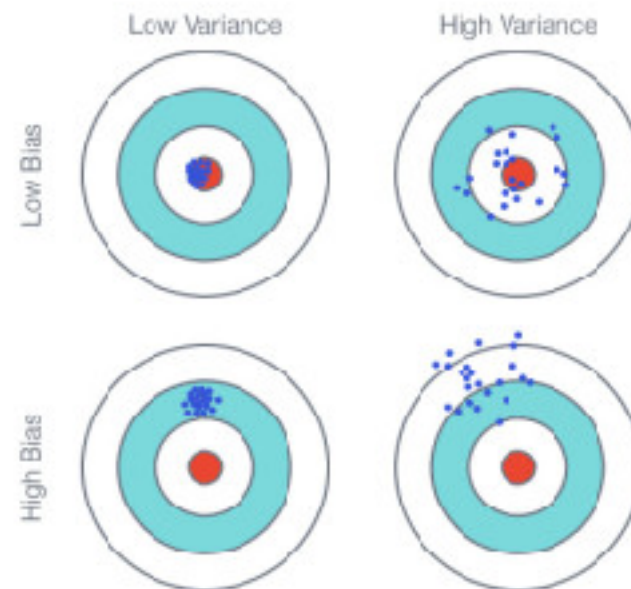
VALIDAZIONE: OVERFITTING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

Due fattori che influenzano
l'*overfitting* o *underfitting* sono:

- Distorsione del modello (*bias*): abbiamo usato alcune assunzioni errate
- Variabilità del dominio (*variance*): che produce un modello molto sensibile alle fluttuazioni



Precision e *Recall* offrono una misura statica della qualità del processo di apprendimento ma questo è largamente insufficiente

Sono rari i contesti nei quali un algoritmo addestrato si trova a lavorare sempre con gli stessi dati

Una prima contromisura è quella di valutare il modello appreso dall'algoritmo attraverso diversi dati di test

In questo modo sarà possibile valutare il **grado di generalità del modello** appreso

Il training set deve quindi essere rappresentativo e non introdurre bias

Un classico criterio di imparzialità è la sezione casuale dei dati, ma quanti dati servono per rappresentare le istanze?

Dipende dalla complessità del problema (lineare - non lineare) e dell'algoritmo (numero di dimensioni, features, parametri)

Alcuni dicono: la numerosità migliore è quella massima che si può raccogliere

Un criterio è dato dall'identificazioni di fattori moltiplicatori

50-100-1000 per ogni classe del problema

10-30 il numero di feature modellate

10-30 il numero di parametri dell'algoritmo

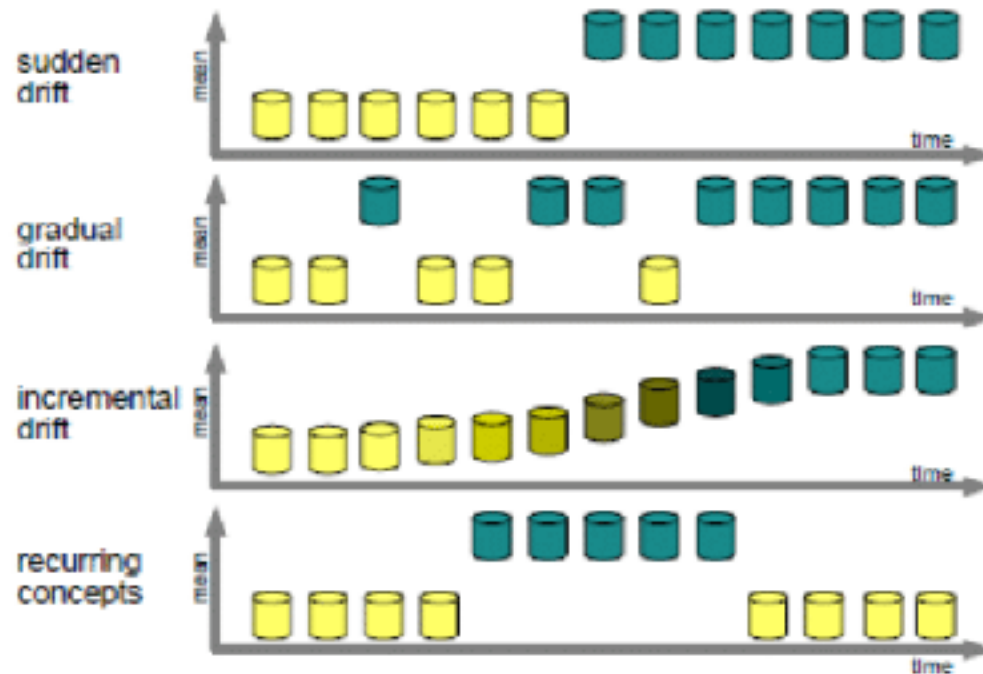
Il tema dell'aggiornamento della conoscenza è altresì critico:

- Quanto è costoso per un algoritmo aggiornare il modello?
 - in termini di dimensione dei dati di addestramento (*training set*) o in termini di tempo
- Il dominio che sto trattando presenta evoluzioni diacroniche (*dominio stazionario o non stazionario* - presenza di *concept drift*)
 - L'evoluzione può essere: improvvisa (*sudden*), graduale (*gradual*), incrementale (*incremental*) oppure ricorrente (*recurrent*)

AGGIORNAMENTO DELLA CONOSCENZA



UNIVERSITÀ
DEGLI STUDI
DI MILANO

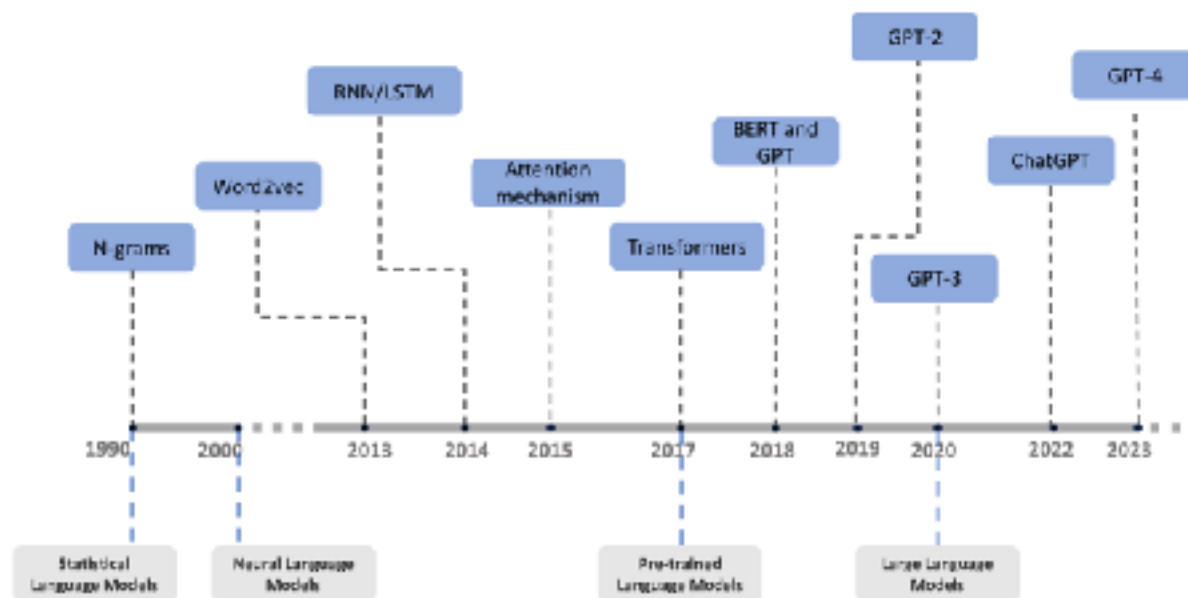


LARGE LANGUAGE MODELS



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- I Large Language Models (LLMs) sono modelli di linguaggio avanzati addestrati su vasti corpus di testo per comprendere, generare e manipolare linguaggio naturale
- Questi modelli sono ottenuti attraverso reti neurali addestrate utilizzando metodi di **self-supervised learning**
- Al modello vengono presentate sequenze di parole, o token, nelle quali alcuni token sono intenzionalmente nascosti, il task è prevedere i token mancanti



- I Large Language Models (LLMs) sono modelli di linguaggio avanzati addestrati su vasti corpus di testo per comprendere, generare e manipolare linguaggio naturale
- Questi modelli sono ottenuti attraverso reti neurali addestrate utilizzando metodi di **self-supervised learning**
 - Al modello vengono presentate sequenze di parole, o token, nelle quali alcuni token sono intenzionalmente nascosti, il task è prevedere i token mancanti
- Sono caratterizzati da dimensioni massicce, spesso con miliardi di parametri, che consentono loro di catturare complessità linguistiche e semantiche
 - Tra i più noti si trovano
 - BERT di Google (340M)
 - GPT-3 (175B) e GPT-4 (8x220B ~ 1.8T) di OpenAI
 - Gemini (156B) di Google
 - Claude (136B) di Anthropic
 - LLaMA (65B) di Meta
 - Mistral AI (7B, 8x7B, 8x30B)

LARGE LANGUAGE MODELS



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Diversi modelli hanno diverse **prestazioni**

Need	Recommended Models	Key Features
High-Quality, Nuanced Output	o1-preview, o1-mini, Claude 3.5 Sonnet (October)	Top-tier quality, well-structured responses, ideal for professional applications
Speed-Sensitive Applications	Llama 3.2 1B, Gemini 1.5 Flash (May)	Fastest output speeds, minimal delay, great for real-time tasks
Cost-Conscious Deployments	Minstral 3B, Llama 3.2 1B	Highly affordable per million tokens, cost-effective for large-scale use
Extensive Context Processing	Gemini 1.5 Pro (September)	2 million token context window, supports deep contextual analysis
Low-Latency, Fast Response	Mistral NeMo, OpenChat 3.5	Minimal response time, excellent for interactive, real-time applications

Best LLM 2024: Top Models for Speed, Accuracy, and Price Generative AI, Nov 6, 2024
<https://medium.com/@genai.works/best-llm-2024-top-models-for-speed-accuracy-and-price-d07ae29f41c4>

LARGE LANGUAGE MODELS



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Diversi modelli hanno diverse **prestazioni**

Model	Provider	# Input cost	# Output cost	# Tokens / second	FinMetricsBench	Parameters
Claude 3 Haiku	Anthropic	\$0.25	\$1.25	91.62	Pass	20 Billion (est.)
Mixtral 8x7B	Groq	\$0.27	\$0.27	253.72	Fail	46.7 Billion
Command Light	Cohere	\$0.30	\$0.60	74.43	Fail	6 Billion
Command-R	Cohere	\$0.50	\$1.50	77.06	Fail	35 Billion
GPT-3.5 Turbo	OpenAI	\$0.50	\$1.50	39.22	Fail	154 Billion
Mixtral 8x7B	Mistral	\$0.70	\$0.70	122.65	Fail	46.7 Billion
Claude 3 Sonnet	Anthropic	\$3.00	\$15.00	40.97	Pass	70 Billion (est.)
Mistral Large	Mistral	\$8.00	\$24.00	111.66	Pass	
GPT-4 Turbo	OpenAI	\$10.00	\$30.00	15.52	Pass	1.76 Trillion (est.)
Claude 3 Opus	Anthropic	\$15.00	\$75.00	16.63	Pass	2 Trillion (est.)

- I LLMs hanno rivoluzionato il Natural Language Processing (NLP), consentendo una migliore comprensione del contesto, una miglior capacità di contestualizzare, generazione di testi coerenti e traduzione automatica più accurata (varie lingue e vari formati sintattici)
- Possono essere impiegati per **supportare autori**, revisore ed editori nella creazione di contenuti, revisione, traduzione. Sono dei coadiuvatori mai autori!
- Devono sempre essere considerati come un aiuto al processo editoriale, per velocizzare alcune operazioni, la responsabilità del contesto è sempre dell'autore
 - Il tipo di contenuto che producono riflette la **conoscenza enciclopedica** attuale, non sono creativi né originali
 - L'autore deve sempre istruire il modello per **indirizzare la costruzione del contenuto**, quali collegamenti privilegiare, quale stile utilizzare, quale vocabolario è più adeguato
 - L'autore è responsabile del contenuto prodotto che va **sempre rivisto** per assicurarsi che non siano stati inseriti errori, **allucinazioni** o elementi poco **informativi**

- Quali modalità per coadiuvare la scrittura attraverso i LLM sono considerate lecite?
- Molte **linee guida** stanno nascendo
- In generale l'approccio è quello di chiedere all'autore di indicare le modalità con le quali ha coadiuvato il suo lavoro di scrittura attraverso i LLM e di chiarire che la responsabilità dei contenuti è sempre dell'autore
 - [Wikibook](#)
 - [IEEE](#)
 - [Avvocati europei](#)
 - [Dipartimento di Informatica - UNIMI](#)
 - [Università di Siena](#)

- Le **linee guida** di [Wikibook](#) richiedono di considerare
 - **Rischi e limitazioni**
 - I LLMs possono generare errori, distorsioni, diffamazioni o violazioni di copyright
 - Possono produrre riferimenti fittizi, contenuti di scarsa qualità, ripetitivi, scarsamente informativi o inappropriati
 - Non possono essere utilizzati per creare nuovi contenuti o idee
 - **Indicazioni da seguire nell'adozione**
 - Possono suggerire miglioramenti stilistici, ma ogni modifica va verificata e giustificata
 - L'utente è responsabile del prodotto finale; l'output dell'LLM deve essere coerente con gli obiettivi del progetto
 - Ogni utilizzo deve essere segnalato nel riepilogo delle modifiche e sulla pagina di discussione con: (i) data di utilizzo; (ii) strumento e versione; (iii) prompt utilizzati

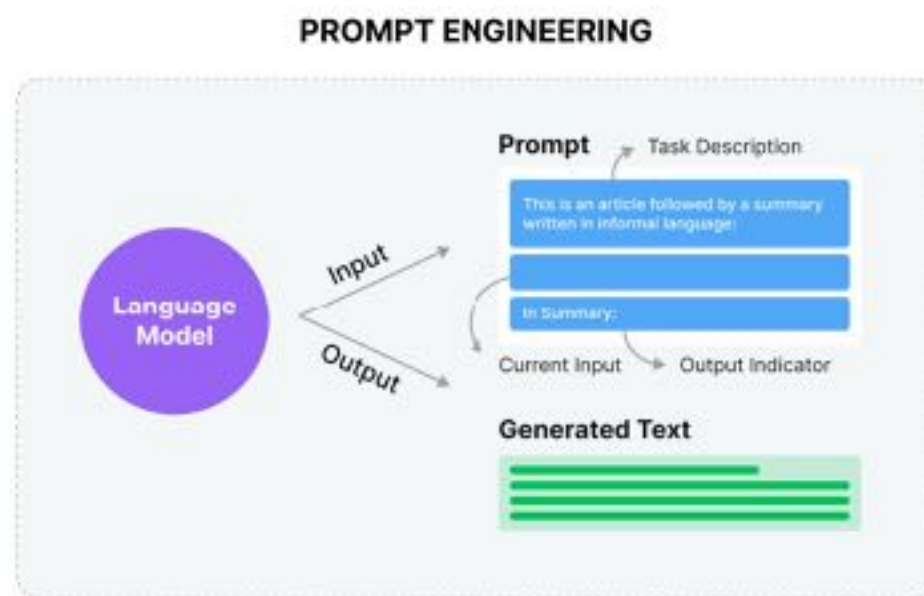
- Per ottenere i risultati migliori è importante **istruire correttamente il modello**. Questo si può fare attraverso tre metodologie principali
 1. Contestualizzare bene la richiesta attraverso opportune tecniche di **prompt engineering**

metodi di interrogazione del modello che favoriscono la corretta contestualizzazione del dominio d'interesse e della tipologia di output richiesto
 2. Adattare i modelli attraverso il **fine-tuning**

una fase aggiuntiva di addestramento per adattare il modello a uno specifico compito o a contenuti di uno specifico dominio
 3. Realizzare un'applicazione **RAG** (*Retrieval Augmented Generation*)

un repository di conoscenze aggiornate e relative a uno specifico dominio è collegato a un LLM che funge da interfaccia per la sua interrogazione

- In termini generali per prompt si intende qualsiasi interrogazione a un LLM
- Alcuni si spingono a definirlo un modo di programmare i LLM o addirittura un nuovo modo di programmare i computer
- Si tratta in gran parte di una competenza empirica nel comporre il prompt per massimizzare le prestazioni del modello su un compito desiderato
- L'obiettivo del prompt engineering è quello di configurare un LLM in coerenza con il compito desiderato
- È possibile “ingannarli” spingendo per l'esecuzione di un compito a cui normalmente non rispondono



- Possiamo distinguere un prompt in alcune sezioni generali
 1. **Contesto**: il dominio, le motivazioni, il ruolo dell'autore, i destinatari, la situazione
 - *Dati i dati delle vendite mensili in formato CSV, scrivi...*
 - *Immagina di essere un professore universitario che deve tenere una lezione a dei professionisti*
 2. **Istruzioni**: cosa è richiesto fare
 - *Identifica l'andamento generale ed evidenzia alcuni casi limite che sembrano non seguire la tendenza*
 - *Scrivi una funzione Python per calcolare i fattoriali*
 3. **Esempi**: sequenze di domande/risposte corrette
 - *Dopo una sequenza di crescita osservi un calo evidenzia questo dato*
 - *if num < 0: # this fragment evaluate a condition on the num variable*
 4. **Vincoli**: quali regole rispettare, quale stile, quale lingua, quale struttura del testo, quali formati
 - *Organizza le tue osservazioni attraverso una tabella*
 - *Fai seguire ad ogni esempio di codice una spiegazione in italiano e discuti i possibili output*
 5. **Domande**: collegamenti, analisi conclusive, valutazioni
 - *Come si potrebbero spiegare questi risultati?*
 - *Consoci alcuni collegamenti storici che aiutano a spiegare i fattoriali?*

- Vediamo sette strategie per un buon prompt engineering:
 1. Essere descrittivi: più informazioni si riescono a fornire meglio è
 1. Chiarire il contesto
 2. Assegnare al modello un ruolo definendo il suo livello di competenza e di esperienza
 3. Chiarire l'obiettivo
 2. Usare esempi: sono ottimi per contestualizzare un dominio
 3. Richiedere risposte strutturate: in questo modo è possibile definire meglio il compito e ottenere il formato più adatto. È possibile usare template
 4. Descrivere una catena di elementi con un crescente livello di dettaglio
 5. Tenete aperto il “dialogo”: lasciate che il modello vi faccia delle domande o chiarite cosa non vi soddisfa delle sue risposte
 6. Riflettere, rivedere e perfezionare

Prompt Engineering Strategies & Tactics

Specify Domain or Style	Frame as a Conversation	Guide the Model's Thinking	Control Tokens or Codes	Clarify Ambiguities	Explore Different Phrasings
Monitor & Evaluate Responses	Combine Instructions	Formatting Instructions	User Feedback Loop	Task Description	Length Experimentation
Incorporate Constraints	Addressing Biases	Critical Evaluation	Use of System Prompts	Temperature and Top-p Adjustment	Iterative Refinement
Explicit Instruction for Creativity	Contextual Description	Example Based Instructions	Task Decomposition	Clarity & Specificity	Formatting Instructions

- È comune distinguere alcune strategie di prompt engineering sulla base della loro strutturazione
- **zero-shot prompting**: si limita ad utilizzare l'istruzione
- **few-shot prompting**: utilizza istruzioni dettagliate ed esempi
- **chain of thought**: definisce una serie di passaggi intermedi per passare dall'input all'output
- **tree of thoughts**: usa una serie di passaggi intermedi ed esplora strategie alternative

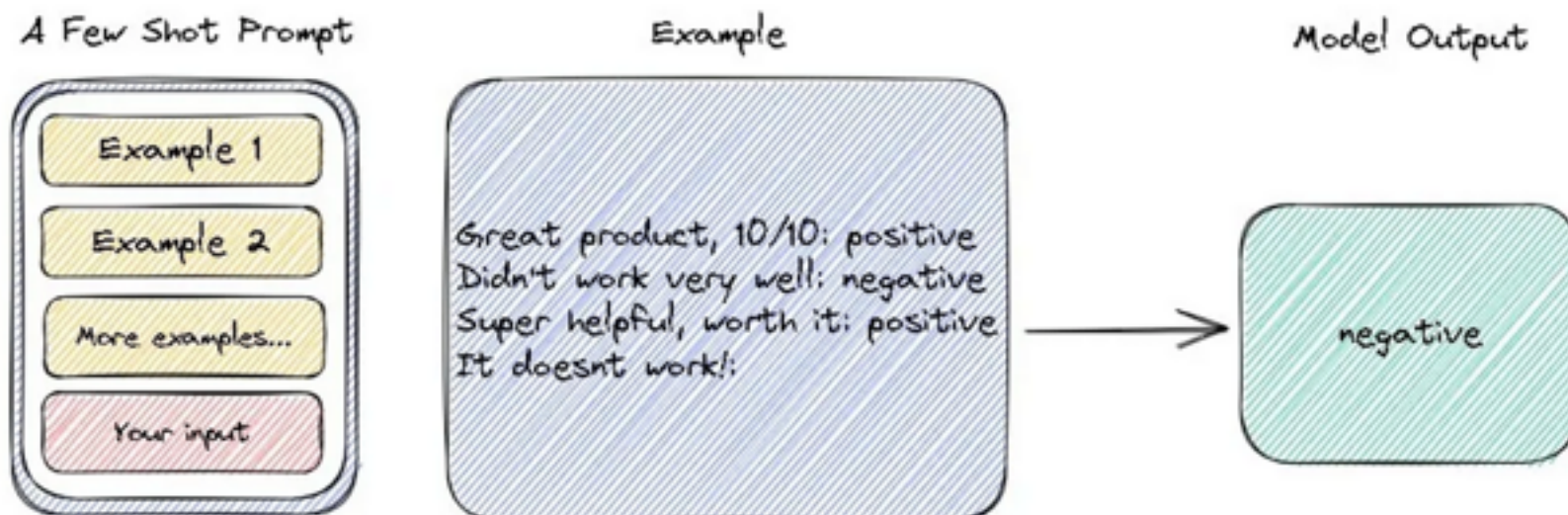
- È adatto a task semplici generalmente orientati alla modifica di formati o stili
 - Traduzione
 - Riassunti di testi
 - Completamento di testi
 - Modifica del tono o stile di un testo
 - Modifica del formato di un testo o codice

FEW-SHOT PROMPTING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- È adatto a task più complessi che richiedono di valutare e classificare
- Classificazione
- Riconoscimento di elementi
- Giustificazione di valutazioni



CHAIN OF THOUGHT PROMPTING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- ▶ Consente capacità di ragionamento complesse attraverso fasi di ragionamento intermedie
- ▶ Per ottenere risultati migliori su compiti più complessi: combinarlo con few-shot prompting

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. **X**

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. **✓**

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 **X**

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. **✓**

- Temperatura: casualità della risposta
 - Temperatura più bassa: risultati più deterministici (per applicazioni di valutazione basate su criteri)
 - Temperatura più alta: maggiore casualità, risultati più diversificati o creativi (per applicazioni come la generazione di testi)
- Top P: una tecnica di campionamento con temperatura, chiamata campionamento del nucleo.
 - Solo i token che comprendono la massa di probabilità top_p sono considerati per le risposte.
 - Top_p inferiore: seleziona le risposte più sicure > risposte esatte e concrete
 - Top_p più alto: considera più parole possibili, comprese quelle meno probabili > risposte più diversificate
- Raccomandazione generale: modificare la temperatura o il Top P, ma non entrambi
- Lunghezza massima: Il numero di token che il modello genera
 - Per evitare risposte lunghe o irrilevanti e controllare i costi
- Sequenze di arresto: una stringa che impedisce al modello di generare token
 - Un altro modo per controllare la lunghezza e la struttura della risposta del modello

▸ **Task:** identifying the label and category of hateful tweets

▸ **Prompt:**

Think you are a linguistic and law expert, and your job is to identify the best match of language type of the sentences below based on the provided list of labels which is: {Hate Speech, Offensive Language, Abusive Language, Discriminative, Irony, cyberbullying, Slur, Aggressiveness, Stereotypes, Body Shame}.

Also detect the category of each sentence based on this list:

{Gender, Sexual orientation, Religion, Disability, Nationality, Race, Ethnicity}.

Please add some explanation and specify the tokens that lead you to the specified label.

Give me the JSON format by fields text, label, tokens, category and explanation.

Here is the list of sentences that are separated by newlines:
{list_of_comments}

PROMPT ENGINEERING – ESEMPIO



UNIVERSITÀ
DEGLI STUDI
DI MILANO

► Output:

- Possiamo usare questo output per creare un set di dati per l'addestramento di un modello specializzato
- Per un esempio più completo: [automatic grader](#)



ChatGPT

Here is the analysis of each sentence based on the provided labels and categories:

```
jason Copy code
[
  {
    "text": "The men are turtles...idiots even in uk. get lost",
    "label": "Stereotypes",
    "tokens": ["men", "turtles", "idiots", "uk"],
    "category": "Gender",
    "explanation": "The sentence contains stereotypes against men, port
  },
  {
    "text": "The guy should feel shame to be gay. It is such a wrong ch
    "label": "Homophobic",
    "tokens": ["guy", "shame", "gay", "wrong choice", "bad influence",
    "category": "Sexual orientation",
    "explanation": "The sentence expresses homophobic views, shaming an
  },
  {
    "text": "The only time I am so disappointed at Trump. Under politic
    "label": "None".
```

- Non dimentichiamo che il prompt engineering ha limiti intrinseci
- 1. Lo spazio di configurazione è **limitato**: non tutte le informazioni pertinenti possono essere inserite nella finestra dei contenuti
- 2. Le strategie da utilizzare **dipendono dal modello**: ogni LLM ha sue peculiarità che devono essere verificate prima di considerare un prompt efficace
- 3. Un modello generico può essere **inefficiente** sia dal punto di vista della accuratezza che dei costi: un modello specializzato più piccolo può superare un modello generale più grande

- Il **fine-tuning** del modello è un processo in cui un modello pre-allenato, che ha già appreso alcuni schemi e caratteristiche su un ampio set di dati, viene ulteriormente addestrato (o "raffinato") su un set di dati più piccolo e specifico
- Evita il **costo elevato** di addestrare un grande modello da zero in termini di risorse computazionali e tempo
- Consente di **adattare** il modello a **dati più aggiornati e specifici** per il dominio di interesse
- Consente di **adattare** il modello allo **specifico task** che dovrà eseguire
- Consente di **ridurre** i rischi di **distorsione**
- Può consentire di aumentare il livello di **riservatezza** della conoscenza inclusa nel modello

- Le fasi del fine-tuning



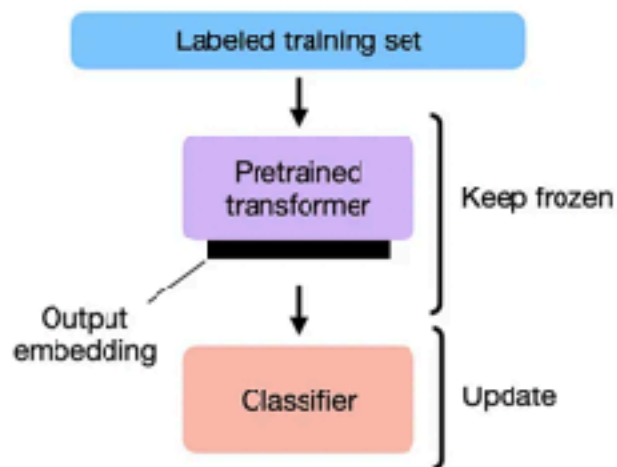
- Le fasi del fine-tuning:
 - Preparazione del dataset. Si prepara il dataset per la messa a punto pulendolo, dividendolo in set di addestramento, validazione e test e assicurandosi che sia compatibile con il modello
 - Scelta del metodo di fine-tuning. Si definisce una metodologia di fine-tuning che è adeguata al task che si vuole realizzare
 - Inizializzazione del modello: Si inizia con un LLM pre-addestrato, come GPT-3 o LLaMA, e lo si inizializza con i suoi pesi pre-addestrati
 - Adattamento: Il modello viene addestrato su un set di dati specifico per l'attività. Durante l'addestramento, i pesi del modello vengono aggiornati tramite backpropagation e discesa del gradiente in base ai dati forniti. È possibile implementare meccanismi di arresto anticipato per evitare l'overfitting
 - Regolazione degli iperparametri. La messa a punto consiste nel regolare iperparametri come il tasso di apprendimento, la dimensione del batch e la forza di regolarizzazione per ottimizzare le prestazioni del modello
 - Validazione. Si monitorano le prestazioni del modello su un set di dati di convalida separato durante il processo di addestramento. Questa fase aiuta a valutare il grado di apprendimento del modello e l'eventuale overfitting rispetto ai dati di addestramento. Se i risultati non sono soddisfacenti si procede a rieseguire le fasi precedenti
 - Test. Una volta completato l'addestramento, si valuta il modello su un set di dati di prova separato che non ha mai visto prima. Questa fase fornisce una misura imparziale delle prestazioni del modello e della sua capacità di gestire dati nuovi e sconosciuti.

METODI DI FINE-TUNING



UNIVERSITÀ
DEGLI STUDI
DI MILANO

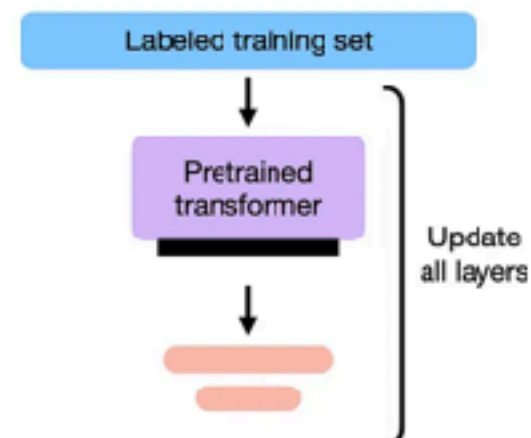
1) FEATURE-BASED APPROACH



2) FINETUNING I



3) FINETUNING II



The 3 conventional feature-based and finetuning approaches.

- **Feature-based.** Utilizza un LLM pre-addestrato come estrattore di caratteristiche, trasformando il testo in ingresso in un vettore di dimensioni fisse. Un classificatore predice la probabilità di attribuire una classe ai vettori generati dal LLM. Durante l'addestramento, cambiano solo i pesi del classificatore, il che rende il sistema poco dispendioso in termini di risorse ma potenzialmente meno performante
- **Finetuning I.** Migliora il LLM pre-addestrato aggiungendo ulteriori strati di neuroni. Durante l'addestramento, vengono regolati solo i pesi dei nuovi strati, mantenendo congelati i pesi dell'LLM pre-addestrato. Negli esperimenti ha mostrato prestazioni leggermente migliori rispetto all'approccio feature-based
- **Finetuning II.** L'intero modello, compreso il LLM, viene riaddestrato, consentendo l'aggiornamento di tutti i pesi del modello. Questo metodo richiede molte risorse, ma può offrire le prestazioni migliori. Un rischio è il catastrophic forgetting, una situazione in cui le nuove caratteristiche sovrascrivono le vecchie conoscenze

- Low Ranking Adaptation (**LoRA**). Utilizza metodi di approssimazione a basso rango per ridurre i costi computazionali e finanziari dell'adattamento di modelli con miliardi di parametri, a compiti o domini specifici
- Quantized LoRA (**QLoRA**). Riduce significativamente l'utilizzo della memoria, pur mantenendo le prestazioni del fine-tuning completo a 16 bit. Questo risultato si ottiene retropropagando i gradienti attraverso un modello linguistico pre-addestrato congelato e quantizzato a 4 bit in adattatori di basso rango
- Parameter Efficient Fine Tuning (**PEFT**). è una tecnica che adatta in modo efficiente i modelli regolando solo un piccolo insieme di parametri, riducendo i costi di calcolo e di archiviazione. Combatte l'oblio catastrofico, regolando i parametri chiave per compiti specifici, e offre prestazioni paragonabili a quelle di una messa a punto completa
- **DeepSpeed**. DeepSpeed è una libreria software di deep learning che accelera l'addestramento di modelli parallelizzando l'uso della memoria
- **ZeRO**. si tratta di una forma di parallelismo dei dati efficiente dal punto di vista della memoria che consente di accedere alla memoria aggregata della GPU di tutti i dispositivi disponibili, senza l'inefficienza causata dalla replica dei dati nel parallelismo dei dati

- L'approccio più adatto a un progetto di editoria digitale è probabilmente il Finetuning di tipo I
- In questo caso il **dataset** deve essere preparato fornendo **una serie di esempi di risposte attese** che il LLM userà per contestualizzare il suo modello di risposta
- Ade esempio con OpenAI il formato è il seguente

```
messages: [  
  {  
    role: "system",  
    content:  
      "Shooketh is an AI bot that answers in the style of Shakespeare's  
literary works.",  
  },  
  { role: "user", content: "What counsel hath thy father imparted?" },  
  {  
    role: "assistant",  
    content:  
      "Arise with rectitude, young warrior, and stiffen thine upper lip  
forthwith.",  
  },  
]
```

- L'approccio più adatto a un progetto di editoria digitale è probabilmente il Finetuning di tipo I
- In questo caso il dataset deve essere preparato fornendo una serie di esempi di risposte attese che il LLM userà per contestualizzare il suo modello di risposta
- Ade esempio con LLaMa il formato è il seguente

```
<s>[INST] Me gradué hace poco de la carrera de medicina ¿Me podrías aconsejar para conseguir rápidamente un puesto de trabajo? [/INST] Esto vale tanto para médicos como para cualquier otra profesión tras finalizar los estudios aniversarios y mi consejo sería preguntar a cuántas personas haya conocido mejor. En este caso, mi primera opción sería hablar con otros profesionales médicos, echar currículos en hospitales y cualquier centro de salud. En paralelo, trabajaría por mejorar mi marca personal como médico mediante un blog o formas digitales de comunicación como los vídeos. Y, para mejorar las posibilidades de encontrar trabajo, también participaría en congresos y encuentros para conseguir más contactos. Y, además de todo lo anterior, seguiría estudiando para presentarme a las oposiciones y ejercer la medicina en el sector público de mi país. </s>
```

- La libreria [transformers](#) di Hugging Face offre molteplici metodo per il training di algoritmi
- In genere si richiede la disponibilità di GPU
 - Qui un [esempio](#) di flusso per il fine-tuning di Phi3 e qui per utilizzarlo in fase di [inference](#)
 - È un modello piccolo ed è quindi gestibile anche attraverso un normale PC
 - Qui un [esempio](#) di flusso per il fine-tuning di LLaMA
- Un framework piuttosto completo per il fine-tuning di LLM è [Axolotl](#)

RETRIEVAL AUGMENTED GENERATION (RAG)



UNIVERSITÀ
DEGLI STUDI
DI MILANO

- Per compiti più complessi nei quali serve una conoscenza di dettaglio
- Un sistema che combina LLM e archivi di documenti su argomenti specifici

- Maggiore coerenza fattuale
- Maggiore affidabilità delle risposte generate
- Riduzione del problema dell'“allucinazione”

