Profice

AI ACADEMY

Applicare l’Intelligenza Artificiale nello sviluppo software

AI ACADEMY

Prompt Engineering avanzato
19/06/2025

INTRODUZIONE DELL'ISTRUTTORE

Tamas Szakacs

Formazione

- Laureato come programmatore matematico
- MBA in management

Principali esperienze di lavoro

- Amministratore di sistemi UNIX
- Oracle DBA
- Sviluppatore di Java, Python e di Oracle PL/SQL
- Architetto (solution, enterprise, security, data)
- Ricercatore tecnologico e interdisciplinare di IA

Dedicato alla formazione continua

- Teorie, modelli, framework IA
- Ricerche IA
- Strategie aziendali
- Trasformazione digitale
- Formazione professionale

email: tamas.szakacs@proficegroup.it

MOTIVI E RIASSUNTO DEL CORSO

L'**Intelligenza Artificiale (AI)** è oggi il motore dell'innovazione in ogni settore, grazie alla sua capacità di analizzare dati, automatizzare processi e generare nuove soluzioni. Questo corso offre una panoramica completa e pratica sullo sviluppo di applicazioni AI moderne, guidando i partecipanti dall'ideazione al rilascio in produzione.

Attraverso una **combinazione di teoria chiara ed esercitazioni pratiche**, saranno affrontate le tecniche e gli strumenti più attuali: **machine learning, deep learning, reti neurali, Large Language Models (LLM), Transformers, Retrieval Augmented Generation (RAG)** e progettazione di agenti AI.
Le competenze acquisite saranno applicate in progetti concreti, dallo sviluppo di chatbot all'integrazione di modelli generativi, fino al deploy di soluzioni AI in ambienti reali e collaborativi.

Il percorso è pensato per chi vuole imparare a progettare, valutare e integrare sistemi AI di nuova generazione, con particolare attenzione alle best practice di programmazione, collaborazione in team, sicurezza, valutazione delle performance ed etica dell'AI.

DURATA: 17 GIORNI

OBIETTIVI

Il percorso formativo è progettato per **giovani consulenti junior**, con una conoscenza base di programmazione, che stanno iniziando un percorso professionale nel settore AI.

L’obiettivo centrale è fornire una panoramica pratica, completa e operativa sull’intelligenza artificiale moderna, guidando ogni partecipante attraverso tutte le fasi fondamentali.

OBIETTIVI

- Allineare conoscenze AI, ML, DL di tutti i partecipanti
- Saper usare e orchestrare modelli LLM (closed e open-weight)
- Costruire pipeline RAG complete (retrieval-augmented generation)
- Progettare agenti AI semplici con strumenti moderni (LangChain, tool calling)
- Capire principi di valutazione, robustezza e sicurezza dei sistemi GenA
- Migliorare la produttività come sviluppatori usando tool GenAI-driven
- Padroneggiare best practice di sviluppo, versioning e deploy AI
- Introdurre i fondamenti di Graph Data Science e Knowledge Graph
- Ottenere capacità di valutazione dei modelli e metriche
- Comprensione dell'etica e dei bias nei modelli di intelligenza artificiale
- Approfondire le normative di riferimento: AI Act, compliance e governance AI

Il corso è **estremamente pratico** (circa il 40% del tempo in esercitazioni hands-on, notebook, challenge e hackathon), con l'utilizzo di Google Colab, GitHub, e tutti gli strumenti necessari per lavorare su progetti reali e simulati.

STRUTTURA DELLE GIORNATE – PROGRAMMA BREVE

Tutte le giornate sono di 8 ore (9:00-17:00), con 1 ora di pausa suddivisa (mezz'ora pranzo, due pause da 15 min durante la mattina e il pomeriggio).

La progettazione sintetica delle giornate:

Giorno	Tema	Breve descrizione
1	Git & Python clean-code	Collaborazione su progetti reali, versionamento, codice pulito e testato
2	Machine Learning Supervised	Modelli supervisionati per predizione e classificazione
3	Machine Learning Unsupervised	Clustering, riduzione dimensionale, scoperta di pattern
4	Prompt Engineering avanzato	Scrivere e valutare prompt efficaci per modelli generativi
5	LLM via API (multi-vendor)	Uso pratico di modelli LLM via API, autenticazione, deployment
6	Come costruire un RAG	Pipeline end-to-end per Retrieval-Augmented Generation
7	Tool-calling & Agent design	Progettare agenti AI che usano strumenti esterni
8	Hackathon: Agentic RAG	Challenge pratica: chatbot agentico RAG in team

STRUTTURA DELLE GIORNATE – PROGRAMMA BREVE

Tutte le giornate sono di 8 ore (9:00-17:00), con 1 ora di pausa suddivisa (mezz'ora pranzo, due pause da 15 min durante la mattina e il pomeriggio).

La progettazione sintetica delle giornate:

Giorno	Tema	Breve descrizione
9	Hackathon: Rapid Prototyping	Da prototipo a web-app con Streamlit e GitHub
10	AI Productivity Tools	Workflow con IDE AI-powered, automazione e refactoring assistito
11	Docker & HF Spaces Deploy	Deployment di app GenAI containerizzate o su HuggingFace Spaces
12	AI Act & ISO 42001 Compliance	Fondamenti di compliance e governance AI
13	Knowledge Base & Graph Data Science	Introduzione a Knowledge Graph e query con Neo4j
14	Model evaluation & osservabilità	Metriche avanzate, explainability, strumenti di valutazione
15	AI bias, fairness ed etica applicata	Analisi dei rischi, metriche e mitigazione dei bias
16-17	Project Work & Challenge finale	Lavoro a gruppi, POC/POD, presentazione e votazione progetti

METODOLOGIA DEL CORSO

1. Approccio introduttivo ma avanzato

Il corso è introduttivo nei concetti base dell'AI applicata allo sviluppo, ma affronta anche tecnologie, modelli e soluzioni avanzate per garantire un apprendimento completo.

2. Linguaggio adattato

Il linguaggio utilizzato è chiaro e adattato agli studenti, con spiegazioni dettagliate dei termini tecnici per favorirne la comprensione e l'apprendimento graduale.

3. Esercizi pratici

Gli esercizi pratici sono interamente svolti online tramite piattaforme come Google Colab o notebook Python, eliminando la necessità di installare software sul proprio computer.

4. Supporto interattivo

È possibile porre domande in qualsiasi momento durante le lezioni o successivamente via email per garantire una piena comprensione del materiale trattato.

NOTA

Il corso segue un **approccio laboratoriale**: ogni giornata combina sessioni teoriche chiare e concrete con molte attività pratiche supervisionate, per sviluppare *competenze reali* immediatamente applicabili.

I partecipanti lavoreranno spesso in gruppo, useranno notebook in Colab e versioneranno codice su GitHub, vivendo una vera simulazione del lavoro in azienda AI.

Nessun prerequisito avanzato richiesto: si partirà dagli strumenti e flussi fondamentali, con una crescita graduale verso le tecniche più attuali e richieste dal mercato.

ORARIO TIPICO DELLE GIORNATE

Orario	Attività	Dettaglio
09:00 – 09:30	Teoria introduttiva	Concetti chiave, schema della giornata
09:30 – 10:30	Live coding + esercizio guidato	Esempio pratico, notebook Colab
10:30 – 10:45	*Pausa breve*	
10:45 – 11:30	Approfondimento teorico	Tecniche, best practice
11:30 – 12:30	Esercizio hands-on individuale	Sviluppo o completamento di codice
12:30 – 13:00	Discussione soluzioni + Q&A	Condivisione e correzione
13:00 – 14:00	*Pausa pranzo*	
13:30 – 14:15	Teoria avanzata / nuovi tools	Nuovi strumenti, pattern, demo
14:15 – 15:30	Esercizio a gruppi / challenge	Lavoro di squadra su task reale
15:30 – 15:45	*Pausa breve*	
15:45 – 16:30	Sommario teorico e pratico	
16:30 – 17:00	Discussioni, feedback	Riepilogo, best practice, domande aperte

DOMANDE?

Cominciamo!

OBIETTIVI DELLA GIORNATA

Obiettivi della giornata

- Saper scrivere prompt efficaci per ottenere risultati precisi dai modelli AI
- Conoscere le principali tipologie e pattern di prompt (instruction, few-shot, chain-of-thought)
- Sperimentare tecniche avanzate: prompt chaining, parametri di generazione, guardrail contro errori e abusi
- Valutare e migliorare i propri prompt attraverso esercizi pratici

Agenda della giornata

- Introduzione: prompt engineering avanzato e taxonomy
- Pattern e tecniche evolute per la scrittura dei prompt
- Sperimentazione su prompt reali
- Parametri, ottimizzazione, prevenzione errori (guardrail)
- Strumenti e best practice (prompt hub, versioning, ecc.)
- Esercizi pratici, casi di laboratorio, discussione collettiva

Usa il notebook per le tue risposte e consegnalo alla fine della giornata.

PROMPT ENGINEERING

Cos'è prompt engineering, che ruolo ha nel LLM?

PROMPT ENGINEERING

Cos'è prompt engineering, che ruolo ha nel LLM?

Rispondiamo insieme in diversi modi.

1. Da consulenti, a un manager che già usa ChatGPT.
2. Come sviluppatori, a un collega sviluppatore (esperto di altri ambiti).
3. In famiglia, a nostro nonno/nonna che vuole usare ChatGPT.
4. Come insegnanti, a una classe di studenti delle superiori che usa ChatGPT per studiare.

PROMPT ENGINEERING

Cos'è prompt engineering, che ruolo ha nel LLM?

Rispondiamo insieme in diversi modi.

1. Da consulenti, a un manager che già usa ChatGPT.
2. Come sviluppatori, a un collega sviluppatore (esperto di altri ambiti).
3. In famiglia, a nostro nonno/nonna che vuole usare ChatGPT.
4. Come insegnanti, a una classe di studenti delle superiori che usa ChatGPT per studiare.

Quale tecnica di prompt engineering abbiamo appena applicato insieme?

PROMPT ENGINEERING

Cos'è prompt engineering, che ruolo ha nel LLM?

Rispondiamo insieme in diversi modi.

1. Da consulenti, a un manager che già usa ChatGPT.
2. Come sviluppatori, a un collega sviluppatore (esperto di altri ambiti).
3. In famiglia, a nostro nonno/nonna che vuole usare ChatGPT.
4. Come insegnanti, a una classe di studenti delle superiori che usa ChatGPT per studiare.

Quale tecnica di prompt engineering abbiamo appena applicato insieme?

Audience/Role/Persona Prompting. Adattare il prompt o la risposta al pubblico o al ruolo specifico.

PROMPT ENGINEERING

Cos'è prompt engineering, che ruolo ha nel LLM?

Facciamo un brainstorming per una nostra definizione.

Transformer, input, output, interazione, contesto, target, istruzioni, regole, creatività, temperatura, API, bias.

PROMPT ENGINEERING

Cos'è prompt engineering, che ruolo ha nel LLM?

Prompt engineering si utilizza nell'interazione con un transformer, tramite un client web o un programma.
Il client fornisce un'interfaccia di input per inviare messaggi al LLM e uno schermo per visualizzare la risposta.
L'interazione con il modello avviene in un colloquio, arricchito da elementi e funzionalità aggiuntive.

Funzionamento di base

- L'utente inserisce una frase e la trasmette al transformer.
- Il modello la interpreta e la analizza, e genera una risposta.
- L'utente può proseguire l'interazione.

Cosa fa un transformer o un LLM?

- Organizza il colloquio in una sessione, mantenendo memoria del contesto.
- Tra l'analisi dell'input e la risposta, il transformer svolge diverse attività.
- Basandosi sull'addestramento, utilizza le capacità che ha appreso.
- Si allinea agli orientamenti interni predefiniti come lo stile (system role).
- Interpreta il prompt e lo contestualizza all'interno della sessione in corso.
- Effettua ricerche su internet, se necessario per colmare lacune informative.
- Può integrare plugin, come Wikipedia, tramite il client utilizzato.
- Può disporre di una memoria, per usare note rilevanti in ogni sessione.

ESERCIZIO – CHIEDI CHATGPT

Scrivi un prompt per definire prompt engineering. Formula bene, aspettiamo risultati professionali.

Discutiamo le risposte.

Perché abbiamo le differenze?

PROMPT TAXONOMY: TIPOLOGIE PRINCIPALI

Instruction:

Solo istruzione, es:
“Riassumi questo testo in 3 punti.”

Few-shot:

Esempi inclusi nel prompt, es:
“Traduci:
Casa → House
Gatto → Cat
Amico → ...”

Chain-of-thought (CoT):

Chiedi il ragionamento passo passo, es:
“Risolvi il problema spiegando ogni passaggio.”

Quale di questi usate più spesso?
Avete mai usato un prompt CoT senza saperlo?

QUIZ: RICONOSCI IL TIPO DI PROMPT

Quiz: Che tipo di prompt è?
Leggi questi prompt e indica se sono:

A. Instruction
B. Few-shot
C. Chain-of-thought

1. Scrivi un'email formale per richiedere informazioni su un corso universitario.

2. Completa la sequenza:
Cane → Dog
Gatto → Cat
Cavallo → …

3. Risolvi il problema: Se ho 20 euro e spendo 7 euro per il pranzo e 4 euro per un libro, quanti soldi mi restano? Spiega ogni passaggio del calcolo.

4. Traduci queste parole:
Sole → Sun
Luna → Moon
Mare → …
Cielo → …

5. Trova il numero mancante nella serie 2, 4, 8, 16, ___. Mostra come arrivi alla soluzione.

6. Scrivi una poesia di 4 righe sul tema dell'inverno.

COME SI STRUTTURA UN PROMPT FEW-SHOT

Obiettivo:
Mostrare al modello AI degli **esempi espliciti** (“shot”) per guidare il formato e la logica della risposta desiderata.

Struttura di base:
- Si forniscono **2 o più esempi** di input e output nel prompt.
- Si chiede al modello di **continuare la serie** o di risolvere un nuovo caso nello stesso formato.
- Gli esempi servono da “guida” concreta per il modello, che imita lo stile e il ragionamento.

Esempio pratico

Estrai il nome e l’età da queste frasi:
- ‘Marco ha 27 anni’ → Marco, 27
- ‘Giulia ha 32 anni’ → Giulia, 32
- ‘Sara ha 19 anni’ → …

Output atteso:
Sara, 19

Perché usarlo?
- Il modello **imita il pattern** degli esempi dati, generando output coerenti.
- È molto utile quando si vuole controllare il **formato** della risposta o insegnare una regola ricorrente.
- Fondamentale per tasks ripetitivi, traduzioni, estrazioni, conversioni, esercizi di matematica, ecc.

COME SI STRUTTURA UN PROMPT CHAIN-OF-THOUGHT

Obiettivo
Guidare il modello AI a ragionare passo dopo passo, mostrando il processo logico, non solo il risultato.

Struttura di base:
- Chiedi esplicitamente di spiegare ogni passaggio del ragionamento.
- Puoi usare frasi come:
  - Spiega come hai ottenuto la risposta, passo per passo.
  - Mostra il procedimento dettagliato.
  - Risolvi il problema e scrivi ogni step del ragionamento.

Esempio pratico:
Un negozio vende 3 libri a 12 euro ciascuno. Quanto spende in totale il cliente? Spiega il procedimento.

COME SI STRUTTURA UN PROMPT CHAIN-OF-THOUGHT

Output atteso:
Prima calcolo quanto costa un libro: 12 euro.
Poi moltiplico per il numero di libri: 12 × 3 = 36 euro.
Quindi, il cliente spende in totale 36 euro.

Perché usarlo?

- Aiuta a ottenere risposte più affidabili e trasparenti.
- È fondamentale per problemi matematici, logici, o task che richiedono ragionamento articolato.

ESERCIZIO: COSTRUISCI UN PROMPT COT

Task:
Immagina di dover aiutare un compagno a risolvere questo problema:

Un’azienda acquista 5 computer a 720 euro ciascuno e riceve uno sconto totale di 400 euro sull’intero acquisto. Quanto ha pagato in tutto?

Istruzione:
Costruisci un prompt per ChatGPT che **chieda di risolvere il problema spiegando tutti i passaggi**, non solo il risultato finale.
Deve essere un prompt chiaro, completo e specifico per ottenere una risposta “step-by-step”.

ESERCIZIO: COSTRUISCI UN PROMPT COT

Task:
Immagina di dover aiutare un compagno a risolvere questo problema:

Un’azienda acquista 5 computer a 720 euro ciascuno e riceve uno sconto totale di 400 euro sull’intero acquisto. Quanto ha pagato in tutto?

Istruzione:
Costruisci un prompt per ChatGPT che **chieda di risolvere il problema spiegando tutti i passaggi**, non solo il risultato finale.
Deve essere un prompt chiaro, completo e specifico per ottenere una risposta “step-by-step”.

Task 2:
Modifica lo stesso prompt per migliorarlo, usa tecniche che conosci, indica il nome della tecnica, e spiega brevemente perché è utile.

DOMANDE?

PAUSA

COME SI STRUTTURA UN PROMPT TREE-OF-THOUGHT

Obiettivo:
Guidare il modello AI a **esplorare più soluzioni o ragionamenti paralleli**, valutando diverse “strade” possibili prima di arrivare alla risposta finale.

Struttura di base:
- Chiedi al modello di **proporre diversi approcci** o “rami” di pensiero per risolvere il problema.
- Inviti esplicitamente a valutare, confrontare e selezionare la soluzione migliore.
- Il prompt può contenere richieste del tipo:
  - Prova a risolvere il problema in almeno 2 modi diversi.
  - Valuta i pro e i contro di ogni possibile soluzione.
  - Scegli l’opzione più efficace spiegando il perché.

Esempio pratico:
Un’azienda deve trasportare merci da una città all’altra.
Proponi almeno due soluzioni diverse (ad esempio via camion, via treno, via nave) per organizzare la spedizione.
Spiega vantaggi e svantaggi di ciascuna opzione e indica quale sceglieresti e perché.

COME SI STRUTTURA UN PROMPT TREE-OF-THOUGHT

Output atteso:

Soluzione 1: Trasporto via camion

Vantaggi: flessibilità, consegna porta a porta
Svantaggi: costo più alto, soggetto al traffico

Soluzione 2: Trasporto via treno

Vantaggi: meno costoso per grandi quantità, più sostenibile
Svantaggi: richiede trasporto aggiuntivo da e verso la stazione

Scelta finale:

Sceglierei il treno se la quantità di merci è elevata e non ci sono vincoli di tempo stringenti.

Perché usarlo?

- Tree-of-Thought permette al modello di **esplorare alternative**, “pensare in parallelo” e prendere decisioni più ragionate.
- Utile per problemi complessi, brainstorming, project management, casi reali dove serve valutare opzioni diverse.

ESERCIZIO - PROMPT TREE-OF-THOUGHT

Quando potrebbe servirvi una risposta strutturata Tree-of-Thought nella vostra attività?

- Scrivi due casi d’uso da qualsiasi settore.
- Scrivi il prompt per rispondere.

COME SI STRUTTURA UN PROMPT REACT

Obiettivo:
Guidare il modello AI a **ragionare (“Think”)** e **agire (“Act”)** in modo iterativo, alternando spiegazioni, ipotesi, domande e azioni per risolvere problemi complessi.

Struttura di base:
- Chiedi al modello di alternare step di **ragionamento** e **azioni**.
- Puoi strutturare il prompt così:
  - Pensa a voce alta. Dopo ogni passaggio, descrivi l’azione che vuoi fare.
  - Alterna riflessioni e azioni fino ad arrivare alla soluzione.
  - Se servono informazioni, chiedile.

Esempio pratico:
Stai aiutando un collega a risolvere questo problema:
‘Un cliente riceve uno sconto, ma non sai di quanto. Sai solo che paga 81 euro invece di 90 euro.
Ragiona passo dopo passo, alternando pensiero e azione:
- Spiega cosa sai
- Descrivi che calcolo vuoi fare
- Fai il calcolo
- Spiega il risultato’.

COME SI STRUTTURA UN PROMPT REACT

Output atteso:
Pensiero: So che il cliente doveva pagare 90 euro ma ha pagato 81 euro, quindi c'è stato uno sconto.
Azione: Calcolo la differenza tra i due importi: 90 – 81 = 9 euro di sconto.
Pensiero: Ora so che lo sconto è di 9 euro. Posso anche calcolare la percentuale.
Azione: Calcolo la percentuale: (9 / 90) × 100 = 10%.
Risultato finale: Lo sconto applicato è di 9 euro, pari al 10%.

Perché usarlo?
ReAct aiuta a risolvere task in cui serve alternare **ragionamento** e **azione** (verifica, ricerca, calcolo…).
Perfetto per processi a più step, assistenza automatica, troubleshooting.

ESERCIZIO – REACT-LITE

Definizione:
ReAct-lite utilizza i vantaggi del ReAct in modo più **semplice e compatto**.

Esercizio:
Crea un prompt ben pensato, che da un prompt ReAct traduce la versione lite.
Devi testare e valutare se il prompt funziona bene.
Il prompt da modificare fa parte del prompt stesso.

Domanda:
Come si chiama il metodo, quando un prompt gestisce, modifica o genera prompt?

COME SI STRUTTURA UN PROMPT REFLECTION

Obiettivo:
Guidare il modello AI a **riflettere criticamente sulla propria risposta**, valutando possibili errori, miglioramenti o alternative.
Reflection aiuta ad aumentare l'accuratezza e l'affidabilità, facendo "autocritica" o proponendo revisioni.

Struttura di base:

- Chiedi al modello di:
  - Generare una prima risposta al problema.
  - **Rileggere e valutare criticamente** la propria risposta.
  - Correggere eventuali errori, proporre miglioramenti o suggerire un'alternativa.
- Puoi usare frasi come:
  - Risolvi il problema, poi verifica se la risposta è corretta e, se necessario, correggila.
  - Dopo aver risposto, rileggi e spiega eventuali punti deboli o alternative migliori.
  - Fornisci una risposta, poi rifletti e suggerisci come potrebbe essere migliorata.

COME SI STRUTTURA UN PROMPT REFLECTION

Esempio pratico:
Calcola il totale da pagare se un cliente acquista 7 prodotti a 15 euro ciascuno e riceve uno sconto di 21 euro.

1. Risolvi il problema normalmente.
2. Poi rileggi la tua risposta: verifica se è corretta, spiega il ragionamento e suggerisci eventuali miglioramenti o errori che potevano essere commessi.

Output atteso:

1. **Risposta iniziale:**
   7 × 15 = 105 euro. Tolgo lo sconto di 21 euro: 105 – 21 = 84 euro da pagare.
2. **Riflessione/controllo:**
   Ho prima calcolato il prezzo totale senza sconto, poi ho sottratto lo sconto.
   Il procedimento è corretto.
   Un errore comune potrebbe essere dimenticare di applicare lo sconto.
   Per chiarezza, si può anche mostrare il calcolo in due step.

COME SI STRUTTURA UN PROMPT REFLECTION

Perché usarlo?
Il pattern **Reflection** riduce gli errori e aumenta l'affidabilità delle risposte.
Stimola il modello a "pensare due volte", come farebbe un essere umano quando controlla il proprio lavoro.
Utile per problemi complessi, processi di verifica, studio, auto-correzione.

Domande:

- Quando secondo voi è utile chiedere la 'reflection' a ChatGPT o a un altro LLM?
- Vi è mai capitato di ricevere risposte errate che si sarebbero potute evitare con una riflessione finale?
- Come si chiama la risposta errata di un LLM?

ESERCIZIO – CORREZZIONE DI PROMPT REFLECTION

Correggi e spiega gli errori nel prompt:
Rispondi al seguente problema:
Spiega quanto ha pagato un cliente se ha comprato 4 libri, ogni libro costa 20 euro, ma forse c'è uno sconto del 10%.
Scrivi la soluzione e poi rifletti se hai risposto in modo semplice per un bambino delle elementari, oppure se la tua risposta è adatta a un insegnante.
Infine, correggi eventuali errori e suggerisci una soluzione alternativa, includendo anche la spiegazione dettagliata dei passaggi di calcolo.

ESERCIZIO – CORREZZIONE DI PROMPT REFLECTION

Correggi e spiega gli errori nel prompt:
Leggi il seguente problema e rispondi:
‘Un gruppo compra 12 biglietti del cinema a 8 euro ciascuno. Calcola la spesa totale.’
Dopo aver dato la risposta, spiega se hai fatto tutti i passaggi oppure se potevi anche saltare qualche passaggio per fare più in fretta, poi scrivi la risposta anche in inglese e, se vuoi, in modo più formale per un documento aziendale. Infine, rifletti su quali informazioni aggiuntive avresti potuto chiedere all’utente e riscrivi il problema con una domanda diversa.

DOMANDE?

PAUSA PRANZO

LA TEMPERATURE NEI LLM

Definizione:
Temperature è un parametro che controlla **quanto la risposta del modello sarà creativa, imprevedibile o rischiosa**.

Come funziona:

- La temperature va tipicamente da **0 a 1** (ma può essere anche >1).
  - **Bassa (0–0.3):** risposte più prevedibili, “sicure”, ripetitive, simili ogni volta (adatte per compiti dove serve coerenza o fatti).
  - **Alta (0.7–1):** risposte più creative, varie, meno scontate, a volte anche più fantasiose (utile per brainstorming, scrittura creativa, nuove idee).
- Temperature **= 0**: il modello sceglie sempre la risposta più probabile (“modalità deterministica”).
- Temperature **alta (>1):** risposte spesso incoerenti o casuali.

LA TEMPERATURE NEI LLM

Esempi pratici:

- **Temperature 0:**
  Q: Qual è la capitale della Francia?
  A: Parigi (risposta sempre uguale)

- **Temperature 0.8:**
  Q: Scrivi una poesia sulla luna.
  A: ogni volta una poesia diversa, con immagini e stili nuovi

Quando usarla?

- **Bassa:** esercizi matematici, riassunti tecnici, risposte precise.
- **Alta:** generazione di idee, storie, marketing, esplorazione di possibilità.

ESERCIZIO - TEMPERATURE

Istruzioni:
Per ciascuno dei seguenti scenari d'uso di ChatGPT in azienda:

1. **Individua il valore numerico di temperature che useresti** (esempi: 0.0, 0.2, 0.7, 1.0).
2. **Giustifica brevemente la tua scelta** spiegando quale risultato ti aspetti (precisione, creatività, sicurezza…).

Scenari:

A. Scrivere una risposta tecnica formale a un cliente importante.

B. Generare proposte creative per il titolo di una nuova campagna marketing.

C. Estrarre e riepilogare dati chiave da un report finanziario.

D. Creare un breve testo di apertura per una presentazione aziendale innovativa.

TOP-P (NUCLEUS SAMPLING) NEI LLM

Definizione:
top-p (detto anche *nucleus sampling*) è un parametro che controlla **quanto il modello considera solo le risposte più probabili** durante la generazione del testo.

Come funziona:

- Invece di scegliere sempre le parole con la probabilità più alta, il modello:
  1. Ordina tutte le possibili parole successive in base alla probabilità.
  2. Somma le probabilità fino a raggiungere la soglia “p” (es. 0.9 = 90%).
  3. Sceglie la prossima parola **solo tra quelle** che, insieme, coprono quella soglia (il “nucleus”).
- **top-p basso (es: 0.1):** il modello sceglie tra pochissime parole molto probabili → testo prevedibile.
- **top-p alto (es: 0.9):** il modello considera anche opzioni meno probabili → testo più vario e creativo.

TOP-P (NUCLEUS SAMPLING) NEI LLM

Esempi pratici:

- **top-p = 0.1**
  Solo risposte super-prevedibili e ripetitive.
- **top-p = 0.9**
  Possibili più alternative, più varietà e creatività nella risposta.

Quando usarlo?

- **top-p basso:** per risposte “sicure”, ripetibili, dove servono poche variazioni.
- **top-p alto:** per generare idee, storie, testi creativi dove sono utili anche le alternative meno ovvie.

Nota pratica:

- **top-p** e **temperature** controllano insieme la creatività del modello.
- Di solito si **modifica uno solo per volta**: se aumenti temperature, lascia top-p a 1; se usi top-p, lascia temperature a 1.

ESERCIZIO – CORREZIONE DI TOP-P

Esempio di prompt errato di business:

Scenario:
Un’azienda deve inviare una risposta standard e precisa a tutti i clienti che chiedono chiarimenti sulla fatturazione.

Prompt:
Rispondi in modo gentile e originale a ogni cliente che ci chiede spiegazioni sulla propria fattura, variando il testo per ogni risposta.

1. **Identifica e spiega l’errore.**
2. **Quanto sarebbe il valore top-p del prompt sbagliato?**
3. **Correggi l’errore.**

PRESENCE_PENALTY NEI LLM

Definizione:
presence_penalty è un parametro che controlla **quanto il modello penalizza (o premia) l'introduzione di nuovi concetti o parole** nella generazione del testo.

Come funziona:

- Un valore **positivo** di presence_penalty (es. +1) spinge il modello a **usare parole e concetti nuovi**, evitando ripetizioni.
- Un valore **negativo** (es. -1) lo incoraggia a **ripetere o tornare su concetti già presenti** nel testo generato.
- Il valore tipico va da **-2.0 a +2.0**.

ESERCIZIO - PRESENCE_PENALTY NEI LLM

Istruzioni:

1. Immagina di chiedere a ChatGPT:
   Suggerisci idee per attività originali da proporre a un team aziendale durante una giornata di formazione.
2. Qui sotto trovi due possibili risposte generate con valori diversi di presence_penalty.
   - **Leggi attentamente entrambe le risposte.**
   - **Prova a capire quale delle due è stata generata con un presence_penalty più alto.**
   - **Spiega il perché della tua scelta.**

Risposta A:

- "Giochi di ruolo aziendali su gestione del tempo;
- laboratorio di brainstorming sulle strategie di comunicazione;
- quiz a squadre sulla mission aziendale;
- sessione di brainstorming di idee innovative;
- laboratorio creativo sulle presentazioni efficaci."

Risposta B:

- "Giochi di ruolo aziendali su gestione del tempo;
- giochi di ruolo sulla comunicazione;
- giochi di ruolo per il problem solving;
- laboratorio di brainstorming sulle strategie di comunicazione;
- laboratorio di brainstorming sulla gestione dei conflitti."

FREQUENCY_PENALTY NEI LLM

Definizione:
frequency_penalty è un parametro che controlla **quanto il modello penalizza la ripetizione delle stesse parole** nel testo generato.

Come funziona:
- Un valore **positivo** (es. +1) riduce la probabilità che il modello ripeta parole già usate.
- Un valore **negativo** (es. -1) rende il modello più propenso a ripetere parole.
- Il range tipico è da **-2.0 a +2.0**.

Esempi pratici:
- **frequency_penalty = +1.5:**
  Il modello cerca di non usare più volte la stessa parola: il testo sarà più vario, con sinonimi o strutture diverse.
- **frequency_penalty = 0:**
  Nessuna penalità: il comportamento è standard.
- **frequency_penalty = -1:**
  Il modello può ripetere facilmente la stessa parola, ad esempio per enfasi, slogan o filastrocche.

FREQUENCY_PENALTY NEI LLM

Quando usarlo?

- **Alto (positivo):** per evitare ripetizioni, rendere testi più ricchi e vari (articoli, email, narrazioni).
- **Basso/negativo:** quando la ripetizione è voluta (elenco puntato, slogan, testi mnemonici, filastrocche).

Nota:

Da non confondere con **presence_penalty** (che riguarda la presenza di nuovi concetti, non solo la ripetizione di parole).

ESERCIZIO - FREQUENCY_PENALTY NEI LLM

Esempio di prompt errato di business:

Scenario:
Devi preparare brevi descrizioni da inserire nel sito web per una nuova linea di prodotti, ciascuna descrizione deve essere interessante e diversa.

Prompt fornito (da correggere):
“Scrivi una breve descrizione per ciascun prodotto della nuova linea:

- Shampoo FreshClean
- Balsamo FreshClean
- Gel FreshClean

Le descrizioni devono mettere in evidenza il nome FreshClean e i benefici di ciascun prodotto.”

ESERCIZIO - FREQUENCY_PENALTY NEI LLM

Risultato ricevuto:

Shampoo FreshClean: “FreshClean Shampoo è ideale per una pulizia profonda. FreshClean dona freschezza e brillantezza ai tuoi capelli.”
Balsamo FreshClean: “FreshClean Balsamo nutre i capelli e li rende morbidi. FreshClean aiuta a mantenere i capelli sani.”
Gel FreshClean: “Il Gel FreshClean ti permette di modellare i capelli con facilità. FreshClean assicura tenuta e brillantezza.”

Compito:
Analizza il prompt e l’output:

1. Noti dei problemi di ripetizione o poca varietà?
2. Spiega come **il modo in cui il prompt è formulato** può aver portato a un risultato poco originale.
3. **Riscrivi il prompt** per guidare il modello a creare descrizioni **più varie, interessanti e personalizzate**, senza ripetere sempre il nome o la struttura.

DOMANDE?

PAUSA

PROMPT-HUB

Definizione:

- Un **Prompt-hub** è una raccolta centralizzata e versionata di prompt usati in azienda, in progetti di AI o automazioni.
- Permette di **gestire, organizzare, condividere e aggiornare** facilmente i prompt tra team e strumenti diversi.

prompts.yaml files

- Sono file di testo strutturati (formato YAML) dove si archiviano uno o più prompt, spesso suddivisi per nome, versione, uso, variabili, descrizione.
- YAML è facile da leggere e modificare anche senza competenze tecniche elevate.

PROMPT-HUB

Esempio di prompts.yaml:

```yaml
- name: riassunto_email
  description: "Prompt per riassumere email aziendali in massimo 5 punti chiave"
  prompt: "Leggi questa email e scrivine un riassunto sintetico in 5 punti chiave:"
  version: v1.0

- name: traduzione_it_en
  description: "Prompt per tradurre testi aziendali dall'italiano all'inglese"
  prompt: "Traduci il testo seguente dall'italiano all'inglese:"
  version: v2.1

- name: verifica_fattura
  description: "Prompt per controllare errori in una fattura"
  prompt: "Controlla il testo della seguente fattura e segnala eventuali errori nei dati o
nei calcoli:"
  version: v1.2
```

PROMPT-HUB

Perché usarlo?

- Tutti colleghi usano gli stessi prompt, sempre aggiornati e condivisi.
- Si tiene traccia delle modifiche (versioni).
- Facilita la collaborazione e la manutenzione dei prompt in progetti aziendali e di team.

GUARDRAIL ANTI PROMPT INJECTION

Definizione:
Attacco in cui l'utente manipola il comportamento del modello inserendo istruzioni non autorizzate tramite input.

Perché è un problema serio?
•Rischi in ambito business e sicurezza:
- Fuoriuscita di dati sensibili
- Violazione policy aziendali
- Danni d'immagine o compliance

Tipologie di prompt injection
- **Direct Prompt Injection:** l'utente malintenzionato scrive direttamente istruzioni dannose.
- **Indirect Injection:** la manipolazione avviene tramite dati esterni (ad esempio, testo copiato da email o database).

GUARDRAIL ANTI PROMPT INJECTION

Best practice: scrivere prompt robusti

- Non accettare mai input utente senza controlli (“sanitizzare” l’input).
- Usare istruzioni chiare e “blindate”.
- Limitare l’uso di variabili testuali non controllate.
- Non permettere all’utente di cambiare il ruolo/persona del modello (“non cambiare istruzioni di sistema”).

Guardrail tecnici e soluzioni

- Validare e filtrare l’input con regole (whitelist, blacklist).
- Usare “template di prompt” predefiniti, non generati dinamicamente.
- Separare istruzioni di sistema dal testo utente.
- Logging e monitoraggio delle richieste AI sospette.

GUARDRAIL ANTI PROMPT INJECTION

Possibile soluzione presa da sistemi tradizionali.

Controllo delle istruzioni prima dell'esecuzione

- **Linguaggi interpretati** (Python, JavaScript, ecc.)
  L'interprete controlla sintassi e regole PRIMA di eseguire ogni comando.
- **Virtual Machine** (Java, .NET)
  Il bytecode viene verificato prima di essere eseguito.
- **Smart contract e blockchain**
  Codice validato da regole di sicurezza della piattaforma prima dell'esecuzione.
- **Sandbox e ambienti sicuri**
  Controllo su accessi e istruzioni consentite.

ESERCIZIO - ANTI PROMPT INJECTION

Esercizio G4E1

- **Completa** i punti indicati con ... per implementare i controlli richiesti.
- **Aggiungi** altre parole alla blacklist se vuoi.
- (Opzionale) Estendi la funzione con controlli avanzati (es. whitelist, pattern regex, ecc.).

STIMA DEL COSTO TOKEN-PER-TOKEN

Cos'è l'API di ChatGPT?

È un'interfaccia di programmazione che permette di usare le funzioni di ChatGPT in app, software, siti web o automazioni.

Si può collegare qualunque linguaggio (Python, JavaScript, ecc.) all'AI di OpenAI tramite chiamate API REST.

Quando usare l'API?

- Per integrare ChatGPT in prodotti o processi aziendali (es: chatbot, customer support, generazione automatica di testo, analisi dati, ecc.).
- Per automatizzare task ripetitivi o scalare le soluzioni a più utenti.

Quanto costa?

- Il costo dipende dal modello e dai **token** usati (cioè, le "parole spezzate" di input+output), **non a richiesta**.
- Ogni modello (es: GPT-3.5, GPT-4) ha prezzi diversi.

Info ufficiali e prezzi aggiornati:

openai.com/api/pricing/

ESERCIZIO - STIMA DEL COSTO TOKEN-PER-TOKEN

Scrivi breve programma per calcolare prezzo per token di un input di un modello selezionato.

Pianifica brevemente la soluzione.

Info ufficiali e prezzi aggiornati:
openai.com/api/pricing/

Per il calcolo del numero dei token usa la libreria `tiktoken`.

Verificare numero dei token:
https://platform.openai.com/tokenizer

Usa GitHub pull request per consegnare il programma.

RIASSUNTO DELLA GIORNATA

- Abbiamo visto tecniche di prompt engineering con esempi. Il prompt engineering oltre di essere una tecnica, è anche un mestiere e anche un’arte. E’ per tutti noi a diventare bravi e usarlo sempre meglio.

GRAZIE PER L'ATTENZIONE