UNA SERIE SULL'INTELLIGENZA ARTIFICIALE

Io, Gemini e altri quattro

Parte V

Il Grande Confronto: GPT, Gemini, Claude, LLaMA e Grok

ROCCO MASSIMILIANO MONDO

ABSTRACT

Nel panorama dell'intelligenza artificiale generativa, non esiste un solo protagonista. GPT di OpenAI, Gemini di Google, Claude di Anthropic, LLaMA di Meta e Grok di xAI sono le cinque grandi famiglie di modelli che si contendono la leadership tecnologica mondiale. Questo confronto prende in considerazione solo i grandi protagonisti occidentali, ma è doveroso sottolineare che ad oriente si è già sviluppato un ecosistema AI veramente competitivo e a tratti rivoluzionario. In questo quinto episodio confronteremo le AI secondo parametri tecnici e pratici: contesto massimo, capacità di ragionamento, multimodalità, sicurezza, modalità di accesso (open-source vs proprietario) e costo. L'obiettivo non è incoronare un vincitore assoluto — non esiste — ma fornire gli strumenti per scegliere il modello giusto per il compito giusto. Con una tabella comparativa dettagliata e considerazioni basate sull'uso reale, questa puntata vi aiuterà a orientarvi in un panorama in continua e rapida evoluzione.

Rocco M. Mondo, Maggio 2026

PARTE V

Il Grande Confronto: GPT, Gemini, Claude, LLaMA, Grok

PREMESSA: PERCHÉ CONFRONTARLI?

Nel precedente articolo abbiamo imparato che la qualità dell'output dipende dalla qualità del prompt. Ma c'è un secondo fattore altrettanto determinante: il modello che si usa. La stessa domanda, formulata allo stesso modo, può produrre risultati molto diversi a seconda del modello a cui la si pone.

Questo non è un giudizio di merito assoluto: ogni modello ha una storia, una filosofia progettuale, un ecosistema e dei casi d'uso in cui eccelle. Conoscere queste differenze permette di fare scelte consapevoli, invece di usare sempre lo stesso strumento per tutti i compiti.

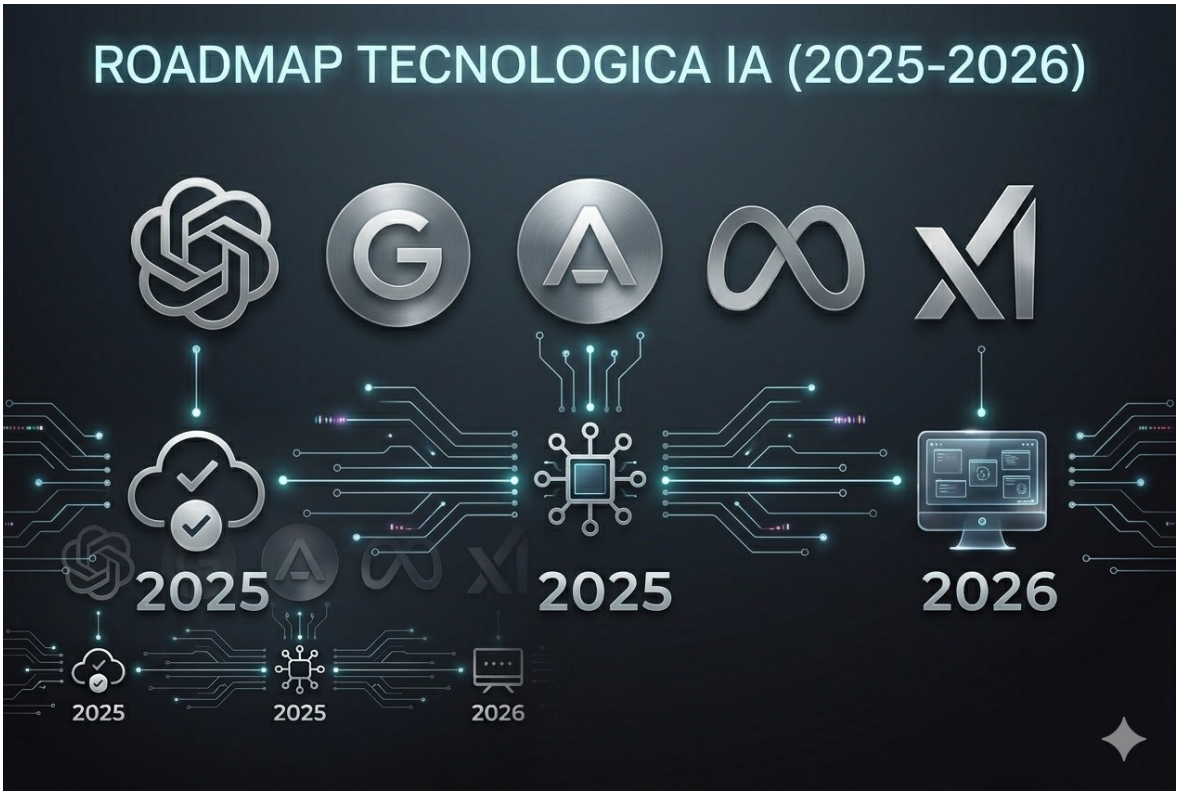
Ho personalmente utilizzato tutti e cinque i modelli descritti in modo intenso e su compiti tecnici reali: dall'analisi di bollette energetiche con Excel, alla revisione di guide al calcolo illuminotecnico, fino allo sviluppo di applicazioni in Python. Quello che segue non è teoria ma esperienza vissuta, con tutto il grado di soggettività che questo comporta.

PARTE V

Il Grande Confronto: GPT, Gemini, Claude, LLaMA, Grok

L'ECOSISTEMA AI: UNA MAPPA PER ORIENTARSI

Prima di entrare nel confronto tra i singoli modelli, è utile acquisire una visione d'insieme dell'ecosistema. I modelli LLM non sono oggetti isolati: vivono all'interno di un ecosistema tecnologico che ne definisce le modalità di accesso, di personalizzazione e di integrazione in prodotti reali.



Modelli Chiusi vs Open Weights

Esistono fondamentalmente due grandi categorie di modelli LLM, che si distinguono per una caratteristica fondamentale: ***la disponibilità dei pesi*** — cioè dei valori numerici appresi durante il training che determinano il comportamento del modello.

	Modelli Chiusi (Proprietari)	Open Weights
Esempi	GPT-4o, Gemini 2.0, Claude 3.7, Grok 3	LLaMA 3.3, Mistral, Phi-4, DeepSeek
Accesso	Solo via API / interfaccia web	Download diretto dei pesi (.gguf / .safetensors)
Modificabile?	No — architettura e pesi sigillati	Sì — fine-tuning, quantizzazione, fork
Privacy	Dati inviati ai server del provider	Elaborazione 100% locale, nessun dato trasmesso

Aggiornamenti	Automatici, a cura del provider	A carico dell'utente/operatore
Costo licenza	Abbonamento mensile o pay-per-token	Zero (Apache 2.0, MIT, LLAMA Community)
Hardware	Nessuno — tutto cloud	GPU dedicata consigliata (≥16 GB VRAM per 70B)

Come si usa un LLM?

Gli strumenti per accedere agli LLM spaziano dalle interfacce grafiche desktop per utenti non tecnici, fino alle ***API REST*** e agli ***SDK Python*** per sviluppatori. Le GUI desktop come ***Jan***, ***LM Studio*** e ***GPT4All*** offrono un'esperienza semplice per l'uso locale. Le ***Web UI*** come ***Claude.ai***, ***ChatGPT*** e ***Dify*** sono accessibili via browser senza alcuna configurazione. Gli strumenti ***CLI*** come ***ollama*** e ***llama.cpp*** garantiscono la massima flessibilità da terminale. Infine le estensioni per IDE come ***Continue*** e ***Cline*** integrano l'assistenza IA direttamente nell'ambiente di sviluppo.

[Nota]
*La lista degli strumenti può sembrare criptica. Il messaggio chiave è questo: oggi si usano gli LLM prevalentemente per sviluppo software e automazione IT. Ma si stanno moltiplicando le applicazioni per l'automazione di task d'ufficio e, più recentemente, per il controllo di dispositivi fisici tramite il protocollo **MCP (Model Context Protocol)**. L'LLM sta smettendo di essere un chatbot per diventare un operatore autonomo.*

LE TRE MODALITÀ FONDAMENTALI DI UTILIZZO

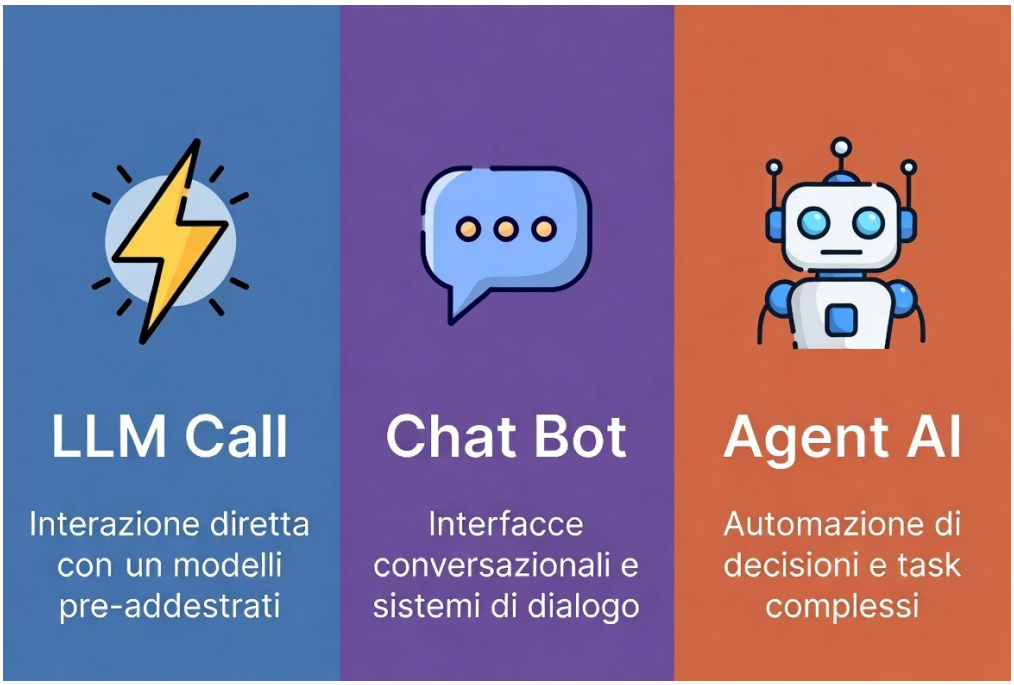
Indipendentemente dal modello scelto, esistono tre architetture fondamentali con cui un LLM può essere integrato in un sistema reale. La distinzione non è banale: determina la complessità implementativa, il costo computazionale, il livello di autonomia e i rischi connessi.

PARTE V

Il Grande Confronto: GPT, Gemini, Claude, LLaMA, Grok

Modalità	Carattere	Descrizione tecnica
LLM CALL	Stateless · Single-turn	Input → Output, una sola volta. Nessuna memoria tra le chiamate. Caso d'uso tipico: generazione di testo, classificazione, riassunto.
CHAT BOT	Stateful · Multi-turn	La cronologia dei messaggi viene mantenuta nel contesto. L'LLM non compie azioni autonome: risponde e attende. Caso d'uso: assistente, FAQ, supporto clienti.
AGENT AI	Agentic · Loop autonomo	LLM ragiona → decide azione → esegue tool → osserva risultato → rivaluta → continua o conclude. Caso d'uso: automazione task, coding agent, ricerca autonoma.

L'evoluzione verso l'AI agentica è la transizione più significativa in atto. Un agente non si limita a rispondere: pianifica, decide, agisce e osserva le conseguenze in un loop autonomo. Questo comporta capacità operative concrete: mandare email, compilare forms, navigare siti web, eseguire codice, scrivere su database. Da qui nascono sia le opportunità straordinarie che i rischi altrettanto reali di sistemi che operano per conto nostro senza supervisione continua.



IL DNA DI UN LLM: ANATOMIA TECNICA

Il mercato offre centinaia di modelli LLM. Perché così tanti? Perché ogni modello è il risultato di scelte progettuali precise — architettura, dati di training, tecnica di allineamento, dimensione — che lo rendono più o meno adatto a compiti specifici.

Le Capacità Principali

Sei capacità fondamentali caratterizzano i modelli moderni: ***Tool/Function Calling*** (invoca API esterne, esegue codice, legge file), ***Vision/Multimodalità*** (elabora immagini, grafici, video in input), ***Extended Thinking/CoT*** (ragionamento step-by-step esplicito e verificabile), ***Embedding*** (rappresentazione vettoriale per RAG e semantic search), supporto ***Multilingual*** (comprensione e generazione in oltre 30 lingue) e ***Coding*** (generazione, debug e refactoring del codice sorgente).

Anatomia di un Modello: Il Caso Gemma 4 (Google)

Per comprendere concretamente cosa significano i numeri associati a un LLM, analizziamo la scheda tecnica di ***Gemma 4 di Google***. La denominazione «Gemma-4-31B-IT» contiene già informazioni tecniche essenziali:

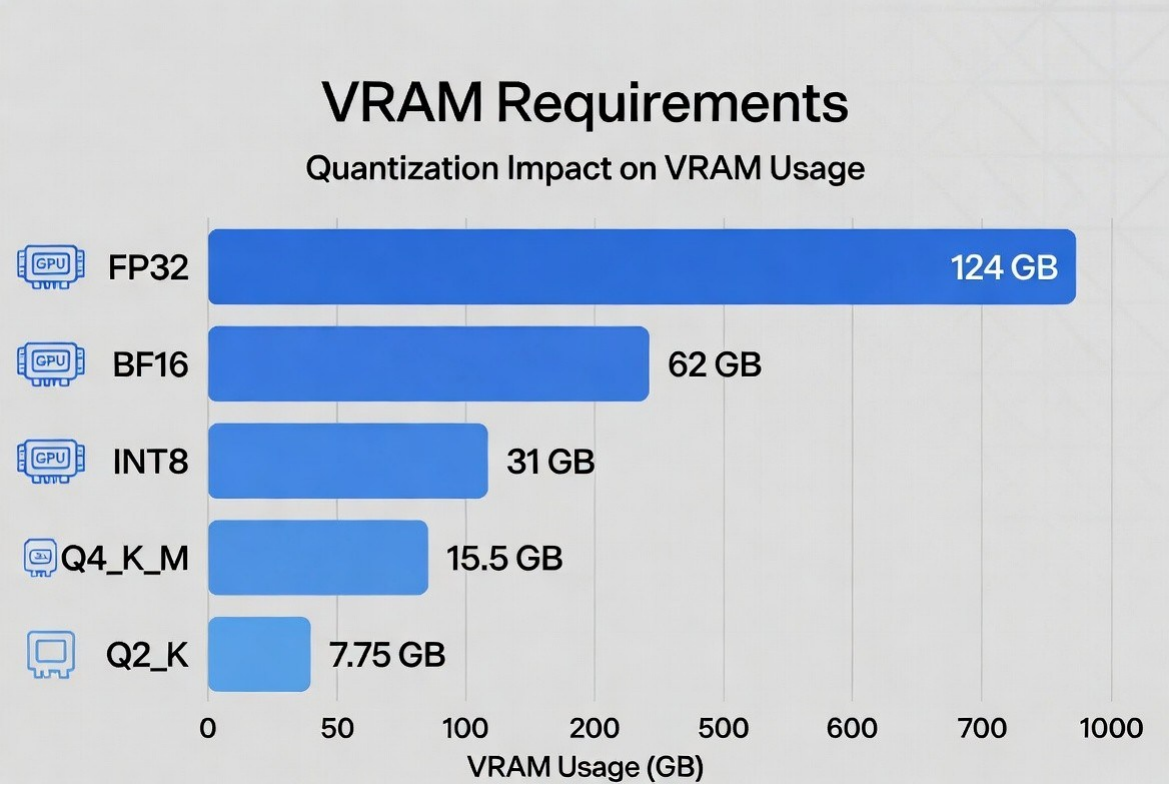
Elemento	Valore	Significato
Gemma	Nome famiglia	Serie di modelli leggeri di Google, ottimizzati per l'esecuzione locale
4	Versione	Quarta generazione della famiglia Gemma
31B	31 Miliardi	31.000.000.000 parametri — i «neuroni» della rete neurale interna
IT	Instruction Tuned	Fine-tuned per seguire istruzioni; la variante «PT» (Pre-Trained) è meno allineata

Parametri, Bit e Memoria: Il Calcolo della VRAM

La domanda pratica è: quanta memoria RAM o VRAM serve per eseguire un modello in locale? La risposta dipende dalla precisione numerica con cui i parametri vengono memorizzati. La formula di base è: VRAM necessaria (GB) = Numero di parametri × Bytes per parametro.

Precisione	Bit/param.	Bytes/param.	RAM/VRAM (31B)	Uso tipico
FP32	32	4	~124 GB	Training iniziale — cluster GPU
BF16 / FP16	16	2	~62 GB	Fine-tuning avanzato — GPU professionale (A100)
INT8	8	1	~31 GB	Inferenza server — NVIDIA RTX 4090 + offload
Q4_K_M	4	0.5	~15.5 GB	Inferenza locale — GPU consumer 16 GB ✓
Q2_K	2	0.25	~7.75 GB	CPU / iGPU — qualità ridotta, massima accessibilità

La quantizzazione è il processo di riduzione della precisione numerica per minimizzare l'occupazione di memoria con una perdita di qualità accettabile (una sorta di compressione del file). La notazione ***Q4_K_M*** indica una quantizzazione a 4 bit con strategia ***K-Means mixed*** — una delle tecniche più efficaci per il rapporto qualità/dimensione. Strumenti come ***Ollama*** e ***llama.cpp*** gestiscono la quantizzazione automaticamente, rendendo accessibile l'esecuzione locale anche su hardware consumer.



I CINQUE GRANDI: PROFILI

Prima di entrare nel vivo del confronto numerico, è utile capire chi c'è dietro ciascun modello: l'identità dell'azienda sviluppatrice riflette direttamente la filosofia, i punti di forza e le limitazioni del modello.

GPT-4o — OpenAI / Microsoft

Filosofia: «L'AI per tutti» — massima accessibilità, integrazione capillare con gli strumenti d'ufficio esistenti, ecosistema di plugin di terze parti.

Punto di forza: Ecosistema vastissimo: integrazione con Microsoft 365, Copilot, GitHub Copilot, Bing. Community più ampia al mondo. Supporto multimodale completo (testo, immagini, audio, video). Architettura MoE (Mixture of Experts) per efficienza computazionale.

Punto debole: Dipendenza totale dal cloud Microsoft. I piani avanzati hanno un costo significativo per uso intensivo professionale. Riduzione silenziosa delle capacità nelle versioni free.

Ideale per: Uso generale, automazione Office 365, sviluppo in ecosistema Microsoft, generazione immagini via DALL·E 3.

PARTE V

Il Grande Confronto: GPT, Gemini, Claude, LLaMA, Grok

Gemini 2.0 — Google DeepMind

Filosofia: «L'AI connessa al mondo» — nativa nel web, pensata per chi lavora nell'ecosistema Google con acce real-time alle informazioni.

Punto di forza: Context window da 1 milione di token — la più ampia disponibile (≈ 3.000 pagine A4). Accesso nativo a Google Search (Grounding) per verifica in tempo reale. Architettura nativa multimodale.

Punto debole: I dati inviati a Gemini alimentano l'ecosistema Google. Sconsigliato per informazioni aziendali riservate. La versione Ultra è accessibile solo tramite Google One AI Premium.

Ideale per: Ricerca avanzata, verifica in tempo reale, analisi di corpus molto lunghi, sviluppo in ecosistema Go Workspace.

Claude 3.7 — Anthropic

Filosofia: «L'AI sicura e precisa» — nata da ricercatori focalizzati sulla sicurezza e l'allineamento. Approccio Constitutional AI.

Punto di forza: Eccellente nel ragionamento su testi lunghi e nella coerenza narrativa (context 200K token). Extended Thinking (CoT esplicito e verificabile). Ottimale per analisi di documenti complessi, scrittura tecnica e coding Python. La modalità Sonnet offre il miglior rapporto qualità/costo disponibile.

Punto debole: Più lento nella modalità di ragionamento esteso. Non ha accesso nativo al web (richiede integrazione esplicita). Ecosistema di integrazioni più limitato rispetto a OpenAI.

Ideale per: Scrittura tecnica, analisi normative, coding Python, elaborazione di documenti molto lunghi, ragionamento multi-step verificabile.

LLaMA 3.3 — Meta AI

Filosofia: «L'AI per chiunque» — Meta crede che l'AI debba essere un bene comune, non proprietà di poche aziende. Open weights come atto politico.

Punto di forza: Open weights completo: scaricabile, modificabile, eseguibile localmente. Privacy totale — i dati non lasciano mai il proprio hardware. Fine-tuning (LoRA/QLoRA) su dataset propri. Licenza permissiva per uso commerciale.

Punto debole: Richiede hardware dedicato (GPU con ≥16 GB VRAM per la versione 70B). Setup tecnico non banale per utenti non esperti. Le versioni quantizzate mostrano degradazione della qualità su task complessi.

Ideale per: Privacy assoluta, personalizzazione su dominio tecnico specifico (fine-tuning), uso offline, integrazione in sistemi aziendali senza dipendenze esterne.

Grok 3 — xAI (Elon Musk)

Filosofia: «L'AI senza filtri» — velocità di risposta, accesso a informazioni di tendenza in tempo reale, approccio meno vincolato su temi controversi.

Punto di forza: Accesso in tempo reale ai dati della piattaforma X (ex Twitter) — unicità assoluta nel panorama. DeepSearch integrato per ricerca web. Architettura che punta all'alta velocità di inferenza.

Punto debole: Ecosistema di integrazioni ancora limitato rispetto ai competitor. Accesso completo richiede abbonamento X Premium+. Qualità del ragionamento tecnico inferiore a Claude e GPT-4o sui benchmark.

Ideale per: Monitoraggio notizie e tendenze social, analisi del sentiment su X, ricerca in tempo reale, domande su temi che altri modelli potrebbero limitare.

[Nota bene]
Questo panorama è in continua evoluzione. Mentre redigo questo articolo, ogni settimana vengono rilasciati aggiornamenti, nuove versioni e nuovi modelli. I dati tecnici si riferiscono a maggio 2026. La cosa più importante non è memorizzare i numeri esatti, ma comprendere i parametri di valutazione: sapere leggere una scheda tecnica sarà sempre più utile di qualsiasi classifica.

IL CONFRONTO TECNICO: TABELLA COMPARATIVA

La tabella seguente confronta i cinque modelli secondo i parametri più rilevanti per un uso professionale. Ho incluso sia dati tecnici oggettivi che valutazioni d'uso basate sulla mia esperienza diretta. Le stelle (★) rappresentano valutazioni qualitative soggettive, non benchmark ufficiali.

Caratteristica	GPT-4o	Gemini 2.0	Claude 3.7	LLaMA 3.3	Grok 3
Sviluppatore	OpenAI / MS	Google DeepMind	Anthropic	Meta AI	xAI
Anno rilascio	2024	2025	2025	2024	2025
Context Window	128K token	1M token ★	200K token	128K token	131K token
Parametri (est.)	~200B (MoE)	~1T (MoE)	~175B	70B (pub.)	~300B (est.)
Multimodalità	Testo, img, audio, video	Testo, img, audio, video	Testo, immagini	Testo (+ Vision)	Testo, immagini

Ragionamento	★★★★☆	★★★★☆	★★★★★	★★★☆☆	★★★★☆
Coding	★★★★☆	★★★☆☆	★★★★★	★★★☆☆	★★★★☆
Open Weights	✗ No	✗ No	✗ No	✓ Sì	✗ No
Uso Locale	✗ No	✗ No	✗ No	✓ Sì	✗ No
Privacy dati	⚠ Cloud MS	⚠ Google	✓ Uso limitato	★ Totale	⚠ Piattaforma X

[Precisazione tecnica: Context Window e Token]

*Con « **Context Window** » si intende la quantità massima di testo che il modello può elaborare in una singola sess (input + output combinati). L'unità di misura è il token — mediamente 0.75 parole in italiano. 1K token = circa parole = circa 3 pagine A4. Una **context** window da 200K token permette di elaborare circa 600 pagine di tes una sola sessione; quella da 1M token di Gemini 2.0 permette di caricare un'intera norma CEI o un capito d'appalto completo.*

IL PARADIGMA OPEN WEIGHTS VS PROPRIETARIO

La scelta tra modelli open-weights e modelli proprietari va ben oltre il costo implementativo e tocca temi fondamentali come sovranità tecnologica, privacy dei dati e personalizzazione del dominio.

Il caso per i modelli proprietari

I modelli cloud offrono potenza immediata senza investimento hardware: basta un browser e un abbonamento. Sono aggiornati continuamente senza che l'utente debba fare nulla. L'ecosistema di integrazioni — plugin, API, connettori verso applicativi di terze parti — è vastissimo. Per la maggior parte degli utenti e delle aziende, questa è la scelta naturale e più produttiva nel breve periodo.

Il caso per i modelli open weights

Nella Parte I di questa serie, ho parlato della «democratizzazione tecnologica» come il tema centrale di questa rivoluzione. L'open source è la forma più pura di questa democratizzazione: chiunque può scaricare il modello, studiarlo, modificarlo e addestrarlo su dati propri.

Il vantaggio più importante in ambito professionale è la privacy: i dati non lasciano mai il proprio computer. Per chi lavora con documentazione tecnica riservata, brevetti, dati aziendali o informazioni personali dei clienti, questo non è un dettaglio: è un requisito fondamentale. Il secondo vantaggio è la personalizzazione: attraverso il fine-tuning con tecniche come ***LoRA*** e ***QLoRA***, è possibile specializzare un modello base su un dominio tecnico ristretto — come le normative CEI o il calcolo illuminotecnico — ottenendo un modello specializzato che nessun provider cloud commerciale potrebbe offrire.

QUALE SCEGLIERE? UNA GUIDA PRATICA

La domanda «qual è il modello migliore?» è mal posta. La domanda corretta è: «migliore per cosa, con quale budget, con quale livello di privacy richiesto?». Dopo mesi di utilizzo parallelo nella mia attività professionale, ho sviluppato delle preferenze ben definite per tipo di compito — che riassumo nella tabella seguente.

Caso d'uso	Modello consigliato	Motivazione tecnica
Scrittura tecnica e divulgazione	Claude 3.7	Coerenza narrativa su contesti lunghi (200K token), riduzione delle allucinazioni, stile controllabile tramite prompt di sistema.
Ricerca e verifica in tempo reale	Gemini 2.0	Tool Grounding nativo su Google Search; context window da 1M token permette di caricare interi corpus di documenti.
Coding e sviluppo software	Claude 3.7 / GPT-4o	Claude eccelle in Python e ragionamento multi-step; GPT-4o sull'ecosistema Microsoft 365 e Copilot.
Analisi documenti lunghi (PDF, normative)	Claude 3.7	200K token ≈ 600 pagine A4 in una singola sessione. Coerenza su testi tecnici strutturati.

Privacy totale, uso locale	LLaMA 3.3 + Ollama	Elaborazione 100% on-premise. Possibilità di fine-tuning (LoRA/QLoRA) su dataset aziendali riservati.
Calcoli tecnici (ingegneria, illuminotecnica)	Claude 3.7 + CoT	Chain-of-Thought esplicito e verificabile. Ragionamento step-by-step affidabile. Verificare sempre il risultato.
Monitoraggio social e notizie in tempo reale	Grok 3	Accesso diretto ai dati della piattaforma X. Aggiornamenti in tempo reale senza latenza di indicizzazione.
Creatività e generazione multimodale	GPT-4o / Gemini 2.0	Supporto nativo per immagini, audio e video. DALL·E 3 integrato in GPT-4o; Imagen 3 in Gemini.

[Consiglio pratico]
Tutti i modelli descritti offrono un accesso gratuito di base. Il metodo più efficace per sviluppare un'intuizione propria è il test comparativo diretto: prendete una domanda tecnica che conoscete bene e sottoponetela a tre modelli diversi. Confrontate: struttura della risposta, correttezza tecnica, completezza, citazione delle norme di riferimento. Svilupperete rapidamente un'intuizione che vale più di qualsiasi classifica.

CONCLUSIONE

Non esiste il modello perfetto. Esiste il modello più adatto al compito specifico, al budget disponibile, al livello di privacy richiesto e alle competenze tecniche di chi lo usa. Un ingegnere che analizza documenti normativi lunghi non ha gli stessi bisogni di un redattore che deve scrivere articoli di blog, anche se entrambi usano strumenti basati su LLM.

Quello che questa serie ci ha insegnato, da Parte I in poi, è che la vera competenza non è saper usare un singolo strumento alla perfezione: è capire l'ecosistema, leggere una scheda tecnica, interpretare un benchmark e scegliere consapevolmente. La macchina più potente è quella che sappiamo usare nel modo più adatto al nostro contesto.