

UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione



TESI DI LAUREA

**Esperienze nel contesto della Generative AI con diversi LLM:
gestione dei testi, dei video e del codice**

**Experiences in the Generative AI context using different LLMs: text,
video and code management**

Relatore

Prof. Domenico Ursino

Candidato

Alessandro Renzi

ANNO ACCADEMICO 2025-2026

*Il computer non è una macchina intelligente che aiuta le persone stupide,
anzi è una macchina stupida che funziona solo
nelle mani delle persone intelligenti.*

Umberto Eco, introduzione a C. Pozzoli, "Come scrivere una tesi con il personal computer"

Sommario

Negli ultimi anni l'Intelligenza Artificiale ha assunto un ruolo fondamentale per lo sviluppo tecnologico e scientifico. In particolare, l'IA Generativa rappresenta un'innovazione cruciale, grazie alla sua capacità di generare diversi tipi di contenuti a partire dall'elaborazione di istruzioni espresse in linguaggio naturale. L'avvento di tale tecnologia rappresenta una vera e propria rivoluzione digitale, destinata a cambiare completamente lo sviluppo umano e il mondo occupazionale da qui in avanti.

La presente tesi si propone di esplorare e valutare le reali capacità generative di alcuni dei principali modelli di IA Generativa presenti sul mercato attualmente, evidenziandone sia i benefici sia i limiti e le criticità. L'analisi è stata eseguita sottoponendo dei casi di studio a dei modelli scelti fra i più utilizzati sul mercato: *Gemini*, *DeepSeek*, *Claude*, *Mistral*, *Veo*, *Runway*. Gli strumenti utilizzati fanno parte di categorie differenti (testi, video, codice) e hanno permesso di sviluppare dei test che consentono di avere un quadro generale chiaro e preciso, per fornire al lettore una visione completa riguardo le potenzialità di tali tecnologie.

L'elaborato ha lo scopo di fornire un'analisi critica che evidenzi l'emergere di bug ed errori nella generazione, oltre che gli aspetti positivi, mettendo anche a confronto gli output dei sistemi utilizzati. I risultati di tali test dimostrano come queste tecnologie siano destinate a ricoprire un ruolo di supporto per l'uomo, in uno scenario in cui quest'ultimo passerà dall'essere un esecutore effettivo ad un supervisore delle macchine, sottolineando come il giudizio umano rimanga insostituibile.

Keyword: Intelligenza Artificiale Generativa, Prompt, Gemini, DeepSeek, Claude, Veo, Runway

Introduzione	1
1 Introduzione all'Intelligenza Artificiale Generativa	3
1.1 Definizione di Intelligenza Artificiale	3
1.2 Modelli a confronto: Intelligenza Artificiale Generativa vs Classica	4
1.3 Analisi dei principali modelli di Intelligenza Artificiale Generativa	4
1.4 Impatto dell'Intelligenza Artificiale nella vita quotidiana ed uso odierno	5
1.5 Pregi e limiti dell'Intelligenza Artificiale	5
2 Classificazione dei sistemi di Generative AI	7
2.1 Meccanismi alla base dell'Intelligenza Artificiale Generativa	7
2.1.1 Reti Neurali	8
2.1.2 Deep Learning	9
2.1.3 Large Language Model	9
2.1.4 Passi eseguiti dall'IA Generativa	11
2.2 Criteri di classificazione per i sistemi di IA Generativa	11
2.3 Classificazione per tipo di contenuto	12
2.3.1 Testi	12
2.3.2 Immagini e Video	13
2.3.3 Codice	13
2.3.4 Altri esempi di contenuti	13
2.4 Classificazione per architettura	13
2.4.1 Transformer	14
2.4.2 GAN	14
2.4.3 VAE	14
2.4.4 Diffusion model	14
2.5 Classificazione per accessibilità	15
2.5.1 Modelli Open Source	15
2.5.2 Modelli Closed Source	16
2.6 Classificazione per modalità di input/output	16
2.6.1 Modelli unimodali	16
2.6.2 Modelli multimodali	17
2.7 Classificazione per approccio all'apprendimento	17
2.7.1 Modelli supervisionati	17
2.7.2 Modelli non supervisionati	18

2.7.3	Modelli semi-supervisionati	18
2.8	Classificazione per competenza	18
2.8.1	Foundation model	18
2.8.2	Modelli verticali	19
3	Case study sulla gestione dei testi	20
3.1	Definizione del caso di studio	20
3.1.1	Obiettivi dello studio	20
3.1.2	Approccio scelto	20
3.2	Sistemi scelti per lo studio	21
3.2.1	Gemini	21
3.2.2	DeepSeek	23
3.2.3	Mistral	24
3.2.4	Claude	25
3.3	Definizione delle richieste	26
3.3.1	Prompt iniziale	27
3.3.2	Vincolo di budget	27
3.3.3	Aggiunta di intolleranze	27
3.3.4	Peso e vincolo di casualità	27
3.3.5	Richiesta finale	28
3.4	Analisi con Gemini	28
3.4.1	Risposta al primo prompt	28
3.4.2	Risposta ai vincoli di budget e alle intolleranze	30
3.4.3	Risposta ai vincoli sul peso e sulla casualità	34
3.4.4	Risposta alla richiesta finale	36
3.4.5	Valutazioni sul comportamento di Gemini	39
3.5	Analisi con DeepSeek	39
3.5.1	Risposta al prompt iniziale	40
3.5.2	Risposta ai vincoli di budget e alle intolleranze	42
3.5.3	Risposta ai vincoli sul peso e sulla casualità	46
3.5.4	Risposta finale	49
3.5.5	Valutazioni sul comportamento di DeepSeek	51
3.6	Analisi con Mistral	51
3.6.1	Risposta al prompt iniziale	51
3.6.2	Risposta ai vincoli di budget e alle intolleranze	53
3.6.3	Risposta ai vincoli sul peso e sulla casualità	57
3.6.4	Risposta finale	59
3.6.5	Valutazioni sul comportamento di Mistral	63
3.7	Analisi con Claude	63
3.7.1	Risposta al prompt iniziale	63
3.7.2	Risposta ai vincoli di budget e alle intolleranze	67
3.7.3	Risposta ai vincoli sul peso e sulla casualità	80
3.7.4	Risposta finale	87
3.7.5	Valutazioni sul comportamento di Claude	89
3.8	Conclusioni sul caso di studio	89
3.8.1	Differenze fra i sistemi utilizzati	89

4	Gestione dei video	91
4.1	Multimodalità: dal testo ai video	91
4.1.1	Il processo di generazione dei video	91
4.1.2	Uso odierno ed implicazioni etiche	92
4.2	Definizione del caso di studio	93
4.2.1	Prompt utilizzati	93
4.2.2	Veo	95
4.2.3	Runway AI	96
4.2.4	Obiettivi del caso di studio	96
4.3	Caso di studio	97
4.3.1	Analisi con Veo	97
4.3.2	Analisi con Runway	102
4.4	Esempi aggiuntivi	106
4.4.1	Altri esempi con Veo	106
4.4.2	Esempi con Sora	109
4.5	Conclusioni sul caso di studio	111
5	Gestione del codice	113
5.1	Definizione del caso di studio	113
5.1.1	Contesto applicativo e obiettivi	113
5.1.2	Criteri di valutazione	114
5.2	Metodologia e approccio	114
5.2.1	Sistemi utilizzati	114
5.2.2	Scelta del linguaggio: Python	115
5.2.3	Definizione delle richieste	115
5.3	Analisi del case study	116
5.3.1	Analisi con Gemini	117
5.3.2	Analisi con DeepSeek	123
5.3.3	Analisi con Claude	131
5.4	Analisi critica e riflessioni sui risultati	139
5.4.1	Comparazione tra le performance	139
5.4.2	Implicazioni etiche e ruolo del programmatore	140
6	Discussione	142
6.1	Analisi dei risultati dei casi di studio	142
6.2	Comportamenti dei sistemi utilizzati	143
6.3	Il ruolo del Prompt Engineering	145
6.4	Limiti attuali e futuro dell'Intelligenza Artificiale Generativa	146
	Conclusioni	151
	Bibliografia	152
	Sitografia	154
	Ringraziamenti	156

Elenco delle figure

2.1	Schema che descrive l'esempio sul Deep Learning	10
3.1	Interfaccia web di Gemini	22
3.2	Interfaccia web di DeepSeek-V3.2	23
3.3	Interfaccia web di Mistral	24
3.4	Finestra che si apre nella modalità in incognito di Mistral	25
3.5	Interfaccia web del sistema Claude	26
4.1	Immagine del parco generata da Nano Banana e inviata a Runway	94
4.2	Il QR code per accedere ai video relativi al nostro case study	97
4.3	Uno dei frame iniziali del primo video di Veo	98
4.4	Uno dei frame finali del primo video di Veo	98
4.5	Uno dei frame iniziali del secondo video di Veo	99
4.6	Uno dei frame centrali del secondo video di Veo	99
4.7	Uno dei frame finali del secondo video di Veo	99
4.8	Frame iniziale del terzo video prodotto da Veo	100
4.9	Frame successivo alla caduta della valigia	100
4.10	Frame del quarto video in cui accade l'anomalia	101
4.11	Frame centrale del primo video generato con Gen-4 Turbo	102
4.12	Frame del secondo video generato con Gen-4 Turbo	103
4.13	Frame del terzo video generato con Gen-4 Turbo	103
4.14	Uno dei frame iniziali del quarto video	104
4.15	Frame centrale in cui si presenta un'anomalia nei movimenti	104
4.16	Frame che mostra la caduta della valigia	105
4.17	Frame significativo del video in inglese	106
4.18	Confronto tra il frame iniziale (sinistra) e l'anomalia (destra).	107
4.19	Frame centrale del video	108
4.20	Frame in cui c'è l'anomalia nell'azione	108
4.21	Primo esempio con Sora	109
4.22	Esempi sviluppati con alcuni preset di Sora.	110
5.1	QR code per accedere ai file relativi al nostro caso di studio	116
5.2	Interfaccia grafica del programma generato	117
5.3	Interfaccia grafica dopo l'aggiunta della cronologia	119
5.4	Interfaccia del convertitore universale	120

5.5	Layout grafico finale	122
5.6	Primo programma generato da DeepSeek	124
5.7	Interfaccia del programma dopo il secondo prompt	126
5.8	Interfaccia grafica del convertitore universale	129
5.9	Interfaccia dopo la correzione	131
5.10	Prima versione del programma generato da Claude	132
5.11	Programma relativo al secondo prompt	134
5.12	Interfaccia del convertitore universale di Claude	136
5.13	Interfaccia del software dopo l'ultimo prompt	138

Elenco delle tabelle

3.1	Valutazione riassuntiva dei sistemi testati	90
4.1	Tabella riassuntiva dei sistemi testati.	111
5.1	Tabella riassuntiva per il test sul codice.	140
6.1	Tabella riassuntiva dei vantaggi e dei limiti dei modelli testuali e informatici .	144
6.2	Tabella riassuntiva dei pro e contro dei modelli multimediali testati	145

Elenco dei listati

5.1	Codice per la conversione	117
5.2	Codice per lo zero assoluto e stampa dell'errore	121
5.3	Codice relativo all'ultimo programma di Gemini	122
5.4	Funzione per l'input da tastiera	125
5.5	Alcune funzioni per le azioni con i file	127
5.6	Codice che definisce gli errori dei menu	130
5.7	Codice che definisce la casella testuale e il tastierino	133
5.8	Salvataggio di un elemento e caricamento della cronologia	135
5.9	Codice che causa la duplicazione delle unità di misura	137
5.10	Esempi di codice in cui avviene la duplicazione	138

Nel corso degli ultimi anni l'Intelligenza Artificiale ha influenzato lo sviluppo tecnologico e scientifico, cambiando radicalmente diversi aspetti lavorativi e quotidiani. Tra i vari campi dell'IA, quello dell'IA Generativa è, senza dubbio, il più innovativo ed importante; grazie ad esso è possibile ottenere diverse tipologie di contenuti a partire semplicemente da prompt espressi in linguaggio naturale. Questa capacità rappresenta la vera e propria rivoluzione tecnologica, viste le enormi potenzialità di generazione e l'elevato grado di creatività che portano con sé i modelli.

Questa tesi nasce con l'obiettivo di approfondire tale settore in crescita, esplorando anche diversi aspetti legati ad essi, come le implicazioni etiche e i meccanismi alla base dei modelli. Essa si pone anche il fine di evidenziare aspetti positivi e limitazioni nella generazione di diverse classi di output.

Per attuare tale approccio sono stati ideati tre casi di studio applicativi da sottoporre a diversi sistemi, in modo tale da poterne studiare il comportamento e le risposte. I sistemi scelti sono: *Gemini*, *DeepSeek*, *Claude*, *Mistral*, *Veo*, *Runway*. Tale selezione rispecchia la volontà di coprire più categorie di output (testi, video, codice), allo scopo di fornire ai lettori una panoramica completa.

L'analisi dei risultati, inoltre, ci consentirà di approfondire aspetti e temi riguardanti l'etica, la privacy e il reale apporto che tali tecnologie forniscono alla persona umana. Nel complesso, quindi, questa tesi offre una panoramica a 360 gradi sulle potenzialità e sull'utilizzo dell'Intelligenza Artificiale Generativa.

La tesi è suddivisa in sei capitoli strutturati come di seguito specificato.

- Il Capitolo 1 si occupa di introdurre i concetti generali di Intelligenza Artificiale e Intelligenza Artificiale Generativa, esponendone, in modo chiaro e comprensibile per ogni lettore, le differenze principali. Inoltre, si fornirà una panoramica sull'impatto che tali tecnologie hanno creato.
- Nel Capitolo 2 saranno introdotti i meccanismi alla base dei sistemi generativi. Dopodiché, ci si occuperà di fornire una classificazione per i modelli di IA Generativa secondo diversi criteri, come la natura open source o closed source, e l'unimodalità o la multimodalità dei modelli.
- Il Capitolo 3 riguarda il nostro primo case study, relativo alla generazione di testi. Esso prevede la produzione di una dieta settimanale; i sistemi studiati in questo test sono Gemini, DeepSeek, Claude e Mistral.

- Nel Capitolo 4 si espongono i risultati del caso di studio riguardante i video, con Veo e Runway come modelli presi in considerazione. L'analisi riguarda la semplice riproduzione di un uomo che passeggia in un parco. Inoltre, si forniranno degli esempi aggiuntivi, per delineare un quadro più chiaro.
- Nel Capitolo 5 vengono riportati i risultati dei test sulla generazione del codice informatico. In questo caso lo studio riguarda lo sviluppo di un convertitore universale, i cui risultati verranno analizzati sia dal lato del codice sia dal lato dell'interfaccia esecutiva. I modelli esaminati in questa sezione sono Gemini, DeepSeek e Claude.
- Nel Capitolo 6 verranno discussi e confrontati i risultati ottenuti con i vari case study, sia dal lato tecnico sia dal lato comportamentale. Nell'ambito di tale sessione si esporranno delle considerazioni sul ruolo attuale dell'IA Generativa e sul futuro.

Introduzione all'Intelligenza Artificiale Generativa

Col passare degli anni, l'Intelligenza Artificiale si è diffusa in modo sempre più capillare, fino a diventare uno dei perni della tecnologia moderna, in diversi ambiti e contesti. In questo capitolo faremo un'introduzione su cos'è, com'è nata e a cosa serve l'Intelligenza Artificiale, citando alcuni modelli diffusi.

1.1 Definizione di Intelligenza Artificiale

Quando si parla di Intelligenza Artificiale, si fa riferimento a quel ramo dell'informatica che si occupa di studiare e implementare delle tecnologie in grado di simulare alcuni comportamenti umani. Fra questi, troviamo, per esempio, il ragionamento, l'apprendimento e la creatività.

I primi passi che hanno portato alla nascita dell'Intelligenza Artificiale risalgono agli anni '50. Proprio nel 1950, il ben noto Alan Turing pubblica "Computing Machinery and Intelligence", un articolo in cui, in poche parole, si pone la seguente domanda: "le macchine sono in grado di pensare?". Ed è proprio all'interno di tale scritto che nasce uno dei test che ancora oggi viene utilizzato per valutare il funzionamento dei sistemi di Intelligenza Artificiale: il test di Turing.

Pochi anni più tardi, nel 1956, viene coniato ufficialmente il termine "Intelligenza Artificiale" dallo statunitense John McCarthy. Più tardi, nello stesso anno, venne programmata Logic Theorist, il primo effettivo sistema di Intelligenza Artificiale della storia. L'obiettivo dell'esperimento era quello di simulare il ragionamento umano, e il risultato fu più che positivo. Basti pensare che Logic Theorist riuscì, in poche ore, a dimostrare 38 teoremi su 52, presi da un noto testo matematico.

Andando avanti con gli anni, questo ramo dell'informatica si è diffuso sempre di più, diventando una parte fondamentale in diversi ambiti. Se guardiamo l'evoluzione degli ultimi periodi, possiamo notare come queste tecnologie vengano implementate in sempre più contesti e situazioni.

In generale, l'obiettivo principale condiviso dai diversi tipi di Intelligenza Artificiale è quello di diminuire il lavoro per l'uomo, andando, ove possibile, a sostituirlo.

1.2 Modelli a confronto: Intelligenza Artificiale Generativa vs Classica

Abbiamo diversi tipi di Intelligenza Artificiale, ed è importante definire la differenza che persiste tra quelle classiche e quelle di tipo generativo.

Come accennato nella sezione precedente, con l'IA classica definiamo una tecnologia che cerca di simulare una mente umana. L'Intelligenza Artificiale tradizionale lavora rispettando delle regole predefinite e basandosi su dei dati, producendo un risultato. In pratica, l'uomo definisce le regole da seguire, e la tecnologia si occuperà di eseguire in maniera automatica il compito, rispettando le direttive imposte. Un esempio pratico potrebbe essere dato da una regola simile:

"Se la velocità dell'auto supera i 100 km/h, allora il limite di velocità non è rispettato."

Il sistema si baserà sul valore della velocità, lo analizzerà secondo un costrutto if-then e darà un risultato seguendo il criterio stabilito dall'uomo.

Questi concetti sono alla base di alcune applicazioni che utilizziamo tutti i giorni, come, per esempio, le piattaforme di streaming o di musica. In base ai dati a disposizione, consigliano film, serie tv o canzoni agli utenti. Un altro caso è quello dei programmi di mappatura: si basano sulle informazioni stradali per suggerire il percorso più conveniente in termini di durata.

L'IA di tipo generativo, invece, si basa su degli specifici algoritmi che consentono di generare nuovi contenuti, come testi, video, immagini e codici. La principale differenza rispetto all'Intelligenza Artificiale classica è che i sistemi di Generative AI imparano ed evolvono costantemente basandosi sui dati forniti. Non seguono più regole precise, bensì analizzano la situazione, imparano, ed evolvono, tutto in maniera autonoma. Si passa da un approccio rigido, basato su regole scritte, a un contesto basato sui cosiddetti *prompt*, istruzioni che l'utente cambia in base alle proprie esigenze.

1.3 Analisi dei principali modelli di Intelligenza Artificiale Generativa

Esistono diversi modelli di Intelligenza Artificiale Generativa; in questa sezione vediamo i più diffusi.

Alla base della maggior parte dei modelli di IA Generativa odierni c'è un'architettura chiamata "*Transformer*" (in italiano Trasformatore). Tale architettura si basa sull'analisi delle parole, dando un peso ad ognuna di esse. Il sistema analizza la frase e, di conseguenza, il contesto, e in base a ciò produce un risultato utilizzando le conoscenze a propria disposizione. Questo meccanismo alla base dei Transformer prende il nome di *self-attention*, e permette un grande adattamento a vari contesti.

Se volessimo fare un esempio di tale funzionamento, potremmo pensare ad un prompt contenente una semplice domanda:

"Di che colore è una foglia?"

A questo punto, il trasformatore analizzerà la frase, dando un peso alle parole. Le due parole che probabilmente prenderà in considerazione sono:

- colore;
- foglia.

Una volta fatto ciò, creerà un'associazione fra le parole *foglia-colore*, e restituirà all'utente la risposta che ritiene più consona. In questo caso, essendo una domanda estremamente semplice e senza un particolare contesto, la risposta sarà molto probabilmente "*verde*".

Ma se noi avessimo, invece, posto il quesito in questo modo:

"Di che colore è una foglia in autunno?"

il sistema avrebbe analizzato in maniera differente la frase, prestando attenzione anche sulla parola *autunno*. A questo punto, ci si aspetta che la risposta possa cambiare, dando, per esempio, come esito le parole *rossa*, *marrone* o *gialla*.

Da questo semplice esempio si evince, quindi, che in base alle parole che si danno alla macchina, essa analizza il contesto, dando un risultato che dipenderà proprio dal modo in cui il comando viene posto. Tale meccanismo è alla base di moltissimi modelli di IA Generativa.

Fra i modelli che sfruttano i Trasformer troviamo, per esempio, Gemini di Google, un sistema in grado di adattarsi a svariati contesti e problematiche, producendo risultati di tipi differenti (testi, video, immagini, codice). Altri esempi di modelli che svolgono le stesse funzionalità sono DeepSeek, Mistral e Claude.

Altri due esempi, stavolta di tipo multimediale, sono dati da Veo e Nano Banana. Sono entrambi sviluppati da Google e si occupano di produrre ed elaborare, rispettivamente, video ed immagini.

Per quanto riguarda la produzione di codici informatici, invece, un esempio è dato da AlphaCode, sempre prodotto da Google, ed in grado di scrivere istruzioni in diversi linguaggi.

Vedremo nello specifico il funzionamento di questi modelli nelle prossime sezioni.

1.4 Impatto dell'Intelligenza Artificiale nella vita quotidiana ed uso odierno

Al momento attuale, l'Intelligenza Artificiale viene utilizzata in moltissimi ambiti, sia lavorativi che non. Significativo è stato l'avvento dell'architettura Transformer nel 2017, che ha segnato una svolta nell'utilizzo di tali tecnologie.

Se volessimo elencare alcuni settori in cui l'IA si è diffusa in maniera evidente, potremmo partire dall'ambito scolastico. Tale strumento rappresenta un supporto importante per tutti gli studenti, vista la capacità di poter rispondere ad un'infinità di domande, in ogni materia.

Per quanto riguarda il mondo del lavoro, potremmo citare gli ambiti finanziario e sanitario, dove l'integrazione di sistemi dotati di Intelligenza Artificiale sta portando diversi benefici.

Nell'ambito informatico, essa risulta particolarmente importante, in quanto, grazie ad essa, è possibile scrivere e analizzare codice, semplificando considerevolmente il lavoro durante la fase di scrittura.

L'utilizzo di IA si sta riscontrando sempre di più anche all'interno dei Social Network, dove la presenza di video e immagini realizzati da architetture di tipo generativo è sempre più frequente. Proprio per questo motivo, l'Italia, nel corso del 2025, ha fatto entrare in vigore una legge riguardante proprio l'utilizzo dell'Intelligenza Artificiale, obbligando chi ne fa uso per la creazione di contenuti ad indicare ogniqualvolta essa sia stata utilizzata.

1.5 Pregi e limiti dell'Intelligenza Artificiale

Come si può evincere dalle sezioni precedenti, l'Intelligenza Artificiale offre numerosi vantaggi, ma presenta anche alcune limitazioni.

Di seguito sono elencati i principali vantaggi apportati dalle IA:

- *Automazione*: si possono automatizzare attività ripetitive o che richiedono particolare tempo, snellendo i compiti per l'uomo;
- *Riduzione degli errori*: con l'utilizzo dell'IA, si riscontra in media una riduzione degli errori.
- *Maggiore efficienza e riduzione dei costi*: secondo alcuni studi statistici, l'utilizzo dei sistemi di Intelligenza Artificiale in contesti lavorativi porta ad un miglioramento nell'efficienza e a una conseguente diminuzione dei costi.
- *Disponibilità continua, 24 ore su 24*: queste tecnologie sono accessibili in qualsiasi momento.

Per quel che concerne gli aspetti negativi, invece, si possono citare i seguenti:

- *Occupazione*: l'utilizzo di sistemi di IA sta portando ad un'evoluzione del mondo lavorativo. In diversi ambiti, l'uomo comincia a essere sostituito dalla macchina.
- *Mancanza di empatia e creatività*: è vero che l'IA cerca di simulare la mente umana, ma in determinati contesti e ambienti le scelte e la creatività dell'uomo risultano essere di fondamentale importanza.
- *Privacy e sicurezza*: l'Intelligenza Artificiale analizza e archivia le informazioni che vengono fornite ad essa, comprese quelle personali. Ciò ha creato e continua a suscitare problemi di privacy.
- *Dipendenza dalla tecnologia*: un eccessivo affidamento su queste tecnologie porta l'uomo a perdere le proprie capacità.

Classificazione dei sistemi di Generative AI

All'interno di questo capitolo verrà approfondita in maniera specifica l'Intelligenza Artificiale Generativa. Verranno introdotti ed analizzati i principali meccanismi e paradigmi alla base del funzionamento di tali sistemi di IA e, successivamente, verrà presentata una classificazione di queste tecnologie sulla base di diversi parametri.

2.1 Meccanismi alla base dell'Intelligenza Artificiale Generativa

Nel capitolo precedente è stata presentata una panoramica generale sui concetti legati all'Intelligenza Artificiale, evidenziando le principali differenze tra i sistemi classici e i sistemi di tipo generativo.

In questa sezione, invece, illustreremo in modo più approfondito il funzionamento dei sistemi generativi, analizzando i principali meccanismi su cui tali sistemi di Intelligenza Artificiale pongono le proprie fondamenta.

Con il termine "Intelligenza Artificiale Generativa" si intende un sottoinsieme dell' IA in grado di creare contenuti di diverso tipo e forma, basandosi su dati in proprio possesso. È importante sottolineare che i risultati generati saranno sempre nuovi e mai uguali a quelli da cui il sistema di IA attinge per elaborare la richiesta. La complessità dei prompt dipende dagli utenti; i sistemi generativi sono in grado di elaborare sia richieste brevi e semplici, sia richieste complesse e di lunghezza elevata.

Alla base del processo generativo c'è proprio il meccanismo che permette di elaborare e comprendere i prompt in entrata. Tale meccanismo prende il nome di *Intelligenza Artificiale Conversazionale text-oriented* ed è alla base dell'interazione fra l'utente e la macchina. Questo sistema è basato, quindi, sull'utilizzo di chatbot¹ che consentono l'emulazione delle interazioni umane tramite una chat di tipo testuale, permettendo alla macchina di elaborare e comprendere la richiesta effettuata dall'utente. Proprio perché si parla di interazioni umane, i messaggi dell'utilizzatore non saranno necessariamente delle domande o delle richieste, ma ne potrebbe derivare una vera e propria conversazione.

I primi approcci ai chatbot testuali per la generazione di contenuti risalgono agli anni '60 con *ELIZA*, prodotto dal Massachusetts Institute of Technology. Il problema delle tecnologie di quel periodo, tuttavia, era che la conversazione fra uomo e macchina doveva sottostare a

¹Chatbot: programma informatico che simula una conversazione con un utente.

regole e modelli rigidi. Al contrario, nei sistemi moderni queste regole e limitazioni non sono presenti, permettendo una conversazione fluida e che dà spazio alla creatività umana.

L'IA Generativa per come la conosciamo oggi si è sviluppata in maniera significativa nel corso del 2022 con l'avvento di ChatGPT, un chatbot sviluppato da OpenAI ed accessibile via web. Da quel momento in poi, questo tipo di tecnologia si è diffuso sempre di più. Oggi, il numero di sistemi di Intelligenza Artificiale Generativa è inquantificabile e tale numero, secondo alcuni studi statistici, è destinato ad aumentare in maniera esponenziale ancora per diversi anni.

Abbiamo, quindi, introdotto l'Intelligenza Artificiale Generativa e il concetto di Intelligenza Artificiale conversazionale. Approfondendo i principali meccanismi e modelli alla base della GenAI² è possibile affermare che i tre principali meccanismi su cui si basa il funzionamento di tali tecnologie sono:

- *Reti Neurali*;
- *Deep Learning*;
- *Large Language Model*.

2.1.1 Reti Neurali

In questa sezione osserviamo il primo dei concetti fondamentali per il funzionamento dell'Intelligenza Artificiale Generativa: le Reti Neurali.

Stando alla definizione, una Rete Neurale è un tipo di algoritmo di Machine Learning che si propone di emulare il funzionamento del cervello umano. Fondamentalmente, è uno strumento alquanto potente, in grado di risolvere problemi di gran lunga più complessi rispetto agli algoritmi informatici tradizionali. Si utilizza la parola "Neurali" in quanto sono sistemi di calcolo dotati di nodi interconnessi che funzionano in maniera molto simile ai neuroni del cervello.

Il primo approccio alle Reti Neurali si ebbe nel 1943, quando Warren McCulloch e Walter Pitts concepirono la prima Rete Neurale per mezzo di circuiti elettrici, riportando i loro risultati nell'articolo "A Logical Calculus of The Ideas Immanent in Nervous Activity" McCulloch e Pitts [1943]. Questo modello rivoluzionario venne applicato in diversi ambiti, e nel 1975 si sviluppò la prima Rete Neurale nel campo della ricerca sull'IA.

A livello strutturale, le Reti Neurali sono composte da nodi chiamati, per l'appunto, neuroni, collocati su diversi strati. Ogni neurone riceve in input dati dagli altri neuroni, li elabora e li manda verso altri neuroni, creando, così, un continuo scambio di informazioni. Le connessioni fra questi nodi hanno pesi che indicano la potenza della connessione fra i vari punti.

L'apprendimento delle Reti Neurali avviene regolando il peso delle connessioni fra i vari nodi tramite un processo denominato "*addestramento*". I neuroni vengono esposti a grandi set di informazioni durante tale fase, in modo da raffinare proprio i pesi dei numerosi collegamenti. Questo sistema è molto simile a quello della mente umana.

L'aggiustamento del peso viene calibrato tramite una tecnica chiamata *retropropagazione*. Essa calcola l'errore commesso nella produzione dell'output e lo invia ai vari neuroni, facendolo circolare all'interno della rete tramite appositi algoritmi. In questo modo la rete rielaborerà il risultato precedente e, di conseguenza, modificherà l'output, cercando di minimizzare l'errore e tentando di raggiungere un livello di precisione maggiore.

Esistono diversi tipi di Reti Neurali; i principali sono:

²L'acronimo GenAI sta per *Generative Artificial Intelligence*.

- *Reti Neurali Ricorrenti (RNN)*: sono utilizzate per dati sequenziali. Utilizzano un feedback che permette di conservare la memoria degli input precedenti.
- *Reti Neurali Convoluzionali (CNN)*: si utilizzano prevalentemente per le immagini, per estrarre elementi da esse e per descriverle.
- *Reti Avversarie Generative (GAN)*: sono costituite da due componenti, un generatore e un discriminatore. Il generatore si occupa della creazione dei dati, mentre il discriminatore li esamina e li valuta.

2.1.2 Deep Learning

Vediamo in questa sezione un secondo concetto fondamentale alla base dei sistemi di IA Generativa, ossia il *Deep Learning* (DL).

Anche in questo caso forniamo una definizione: con il termine *Deep Learning* si intende una sottoclasse del Machine Learning che sfrutta le Reti Neurali per simulare il comportamento della mente umana e permettere alla macchina di apprendere elevate quantità di dati.

È, quindi, di fondamentale importanza non confondere i concetti di Machine Learning e Deep Learning. Il Machine Learning (ML) è un sottoinsieme dell'Intelligenza Artificiale basato su algoritmi capaci di apprendere informazioni a partire da dati di addestramento; il Deep Learning, invece, è una forma di ML che lavora utilizzando le Reti Neurali e consente un apprendimento dei dati più autonomo e avanzato.

Solitamente, gli algoritmi di Deep Learning lavorano utilizzando grandi set di dati etichettati. Questa etichettatura dei dati consente ai modelli di DL di interpretare più efficacemente le nozioni. Questi algoritmi associano, quindi, le caratteristiche dei dati alle etichette corrette, creando, così, una sorta di classificazione basata sulle etichette. Una volta addestrato, un algoritmo di Deep Learning sarà in grado di riconoscere e catalogare nuovi tipi di dati, utilizzando i pattern e gli schemi a propria conoscenza.

Vediamo ora un esempio pratico di questo meccanismo applicato alle immagini. Ipotizziamo un sistema che abbia una quantità indefinita di immagini, contenenti paesaggi, sia notturni che diurni. Si può allora addestrare la macchina utilizzando due etichette diverse:

- Giorno;
- Notte.

Ogni fotografia verrà etichettata utilizzando questi due parametri e, in base a ciò, il sistema noterà le differenze e imparerà a riconoscere caratteristiche come la luminosità, le differenze nel cielo, la presenza di luci artificiali, i colori predominanti, e così via. Più il sistema verrà sottoposto a questo processo (ciò accade per ogni immagine presente nel set) e più il modello si evolverà, specializzandosi nel riconoscere le varie caratteristiche. Il modello, quindi, non riconosce i concetti presenti nelle etichette, ma studia e apprende schemi statistici che si ripetono nelle varie immagini.

Una schematizzazione dell'esempio e di quanto detto si può vedere nella Figura 2.1.

Questo caso è soltanto un esempio basilare. Basti pensare a un sistema addestrato con un numero di etichette estremamente elevato e riguardanti i più svariati contesti. Le potenzialità di apprendimento dei sistemi che utilizzano algoritmi di Deep Learning risultano enormi.

2.1.3 Large Language Model

Come ultimo meccanismo alla base dei modelli di IA Generativa approfondiamo gli LLM. L'acronimo LLM sta per Large Language Models, ossia, in italiano, modelli linguistici di

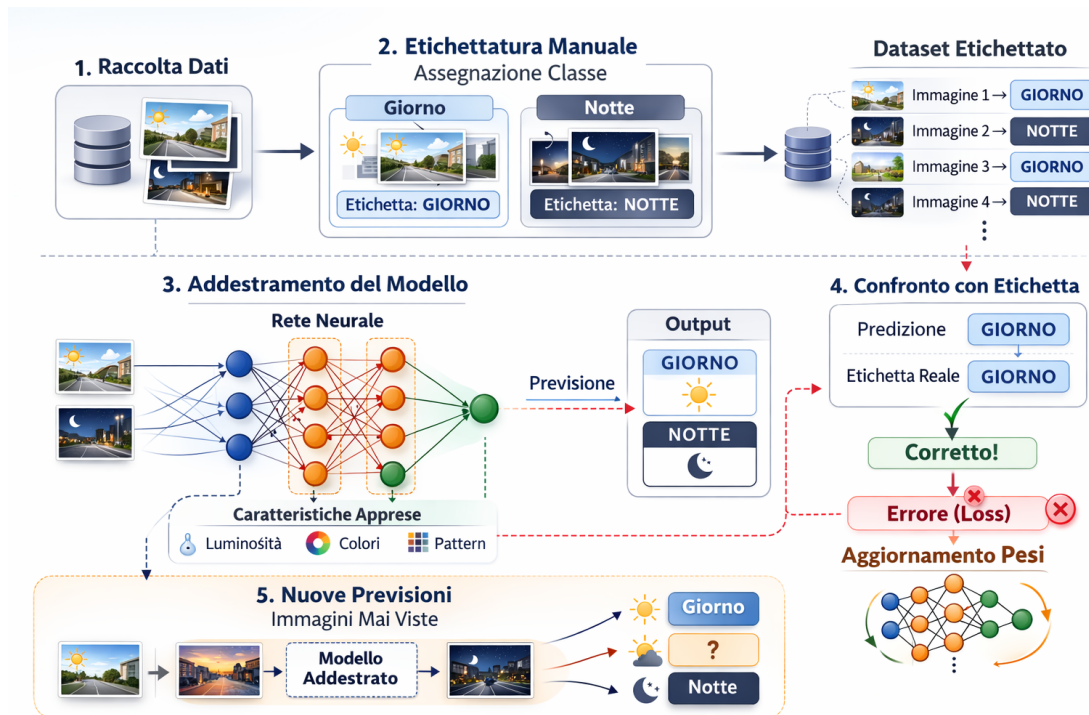


Figura 2.1: Schema che descrive l'esempio sul Deep Learning

grandi dimensioni. Essi sono modelli di Deep Learning addestrati su enormi quantità di dati e sono alla base del processo creativo dei sistemi virtuali generativi.

Si è arrivati all'utilizzo degli LLM dopo svariati anni di progresso con *l'elaborazione del linguaggio naturale* (Natural Language Processing - NLP). L'NLP è un ramo dell'informatica e dell'Intelligenza Artificiale che sfrutta gli algoritmi di Machine Learning per permettere alla macchina di interpretare il linguaggio umano e, di conseguenza, intraprendere una comunicazione con l'utente.

Alla base del funzionamento degli LLM ci sono elementi già visti e accennati nelle sezioni precedenti, ovvero le Reti Neurali e i Transformer. Uno dei principi cardine degli LLM è, infatti, il modello Transformer, che possiamo considerare come un particolare tipo di Rete Neurale. L'apprendimento della Rete, e di conseguenza degli LLM, segue i principi e le regole visti precedentemente.

Una caratteristica fondamentale che differenzia i modelli linguistici di grandi dimensioni rispetto ai suoi predecessori è data dall'*auto-attenzione*.

L'*auto-attenzione* è un meccanismo che consente al sistema informatico di dare priorità a determinate porzioni dei dati presi in input, a cui verrà attribuito, quindi, un peso differente rispetto al resto. Il testo (ipotizzando un input di tipo testuale) viene, quindi, suddiviso in token e, successivamente, viene mappato in un vettore chiamato *embedding*. Gli embedding gireranno all'interno delle Reti Neurali che compongono il Trasformatore, ognuna delle quali modificherà e arricchirà questi vettori. L'obiettivo del processo è che il modello riesca a interpretare il linguaggio e a imparare le associazioni di tipo semantico fra le parole, inserendole in posizioni vicine all'interno del vettore. Così facendo, quando si produrrà un output, la macchina eviterà di accostare parole discostanti fra loro, creando così frasi inerenti al contesto. Questo meccanismo segue, quindi, una logica posizionale.

L'addestramento degli LLM avviene utilizzando miliardi o trilioni di parole. Esistono anche modelli basati su una quantità di dati più piccola, chiamati SLM (Small Language Models). Generalmente, gli LLM vengono utilizzati in svariati contesti, vista la grande

quantità di informazioni in loro possesso; gli SLM, invece, vengono usati in ambienti compatti e specializzati.

A livello storico, questi meccanismi di comprensione del linguaggio si sono diffusi in maniera significativa nel 2017 in concomitanza con l'avvento dei Transformer, come descritto nel lavoro intitolato *Attention Is All You Need* Vaswani *et al.* [2017].

2.1.4 Passi eseguiti dall'IA Generativa

Finora sono stati introdotti e definiti i modelli e i paradigmi alla base dell'Intelligenza Artificiale Generativa. A questo punto, è opportuno analizzare cosa accade quando l'utente scrive un prompt a un chatbot di tipo generativo. Il processo di elaborazione corrispondente prevede i seguenti passi fondamentali:

1. *Scrittura del prompt*: l'utente scrive la propria richiesta, inviandola al chatbot.
2. *Ricezione e partizione in token*: la macchina riceve la richiesta e la divide in token. Subito dopo, i token vengono inseriti negli embedding.
3. *Elaborazione*: i vettori entrano nella Rete Neurale (solitamente con la struttura del modello Transformer) e vengono scambiati fra i vari nodi. Vengono analizzate le parole e i token, cercando di capire il contesto e la richiesta dell'utente.
4. *Generazione*: una volta che la macchina ha individuato il contesto e la richiesta, il modello comincia a generare un output basandosi sugli schemi di cui è a conoscenza.
5. *Finalizzazione e restituzione*: il modello rielabora i token facendoli tornare in forma testuale (o in altri tipi di dati, se la richiesta lo prevede); il risultato generato viene mostrato all'utente all'interno della chat.

Questi 5 passaggi schematizzano e riassumono in maniera semplice ed intuitiva il processo generativo dietro ogni prompt.

2.2 Criteri di classificazione per i sistemi di IA Generativa

I sistemi di Intelligenza Artificiale sono soggetti a classificazioni di diverso tipo, sulla base di criteri e parametri diversi. Abbiamo classificazioni di tipo strutturale, altre relative alle competenze, o, più banalmente, classificazioni in base al tipo di contenuto elaborato o prodotto.

Elenchiamo qui di seguito le principali suddivisioni, ciascuna delle quali verrà approfondita nello specifico nelle sezioni successive.

1. *Classificazione per tipo di contenuto*: il caso più semplice di raggruppamento può avvenire per tipo di contenuto che i sistemi di IA producono: testi, video, immagini, e così via.
2. *Classificazione per architettura*; in questo caso, possiamo prendere in considerazione i seguenti modelli:
 - *Transformer*: modello strutturale introdotto precedentemente;
 - *GAN*: reti avversarie generative;
 - *VAE*: autoencoder variazionale;
 - *Diffusion Models*: modelli di diffusione.

Questi modelli differiscono fra loro per la struttura e il modo in cui sono costruiti. I compiti svolti, invece, potrebbero essere gli stessi.

3. *Classificazione per accessibilità*; in questo caso abbiamo le seguenti categorie:

- *Modelli Open Source*: accessibili a chiunque;
- *Modelli Closed Source*: accessibilità limitata.

La distinzione fra queste due classi di modelli riguarda quali classi di utenti possono accedere al servizio.

4. *Classificazione per modalità di input/output*; in questo caso si hanno:

- *Modelli unimodali*: elaborano una sola tipologia di input;
- *Modelli multimodali*: possono elaborare più tipi di input contemporaneamente.

5. *Classificazione per approccio all'apprendimento*; in questo caso vi sono:

- *Modelli supervisionati*;
- *Modelli non supervisionati*;
- *Modelli semi-supervisionati*.

Le differenze riguardano le modalità di addestramento dei modelli.

6. *Classificazione per competenza*; in questo caso abbiamo le seguenti classi:

- *Foundation model*;
- *Modelli verticali*.

Secondo tale classificazione i sistemi di IA possono essere suddivisi in base al loro livello di specializzazione in un determinato settore di competenza.

2.3 Classificazione per tipo di contenuto

Il criterio più basilare secondo cui i sistemi di Intelligenza Artificiale possono essere classificati riguarda il formato dell'output che sono in grado di generare. Nelle prossime sottosezioni faremo riferimento proprio a tale tipo di classificazione.

2.3.1 Testi

Si fa riferimento a sistemi di IA in grado di generare output di tipo testuale che soddisfino il prompt dell'utente. Tali sistemi si basano sugli LLM e permettono altre azioni di tipo testuale, come traduzioni e riassunti.

Esempi. Tra i sistemi di Intelligenza Artificiale che producono testi, troviamo modelli come Gemini di Google, annunciato al mondo il 6 dicembre 2023 nel paper intitolato *Gemini: A Family of Highly Capable Multimodal Models* Team et al. [2023].

Altri esempi di modelli in grado di generare output testuali sono dati dal sistema cinese DeepSeek, dal sistema francese Mistral, da Claude, sviluppato dall'azienda statunitense Anthropic, e da ChatGPT, che può essere considerato il precursore di tutti, sviluppato da OpenAI.

2.3.2 Immagini e Video

In questo caso si fa riferimento a sistemi generativi in grado di creare immagini e video seguendo le richieste degli utenti. Oltre alla creazione, questo tipo di modelli si occupa generalmente anche di modificare immagini o video già esistenti, allegati dagli utenti al messaggio del prompt.

Esempi. Tra i modelli di IA di questa categoria troviamo Nano Banana, sviluppato da Google e specializzato nell'elaborazione delle fotografie. Nano Banana è accessibile direttamente da Gemini: se l'utente specifica nel prompt di voler un'immagine come output, Gemini avvierà automaticamente Nano Banana per soddisfare la richiesta e produrre l'immagine.

Rimanendo sempre in casa Google, per quel che concerne i video, il sistema di IA sviluppato dal colosso informatico è Veo. Anche in questo caso, Veo viene invocato direttamente da Gemini quando un utente richiede un video.

2.3.3 Codice

Un'altro importante tipo di output elaborato riguarda i codici informatici. I sistemi di IA Generativa sono in grado di scrivere e correggere codice in diversi linguaggi informatici, alleggerendo e semplificando in maniera significativa il lavoro dei programmatori.

Esempio Sempre prendendo Google come riferimento, il sistema di IA sviluppato dall'azienda americana per gestire il codice informatico è AlphaCode, sviluppato dalla sezione Google DeepMind, una divisione di programmatori specializzata nello sviluppo di sistemi di Intelligenza Artificiale. AlphaCode è in grado di scrivere codice in diversi linguaggi, ma dà il meglio di sé con C++ e Python.

2.3.4 Altri esempi di contenuti

Oltre alle tipologie già citate, che rappresentano le più richieste, esistono sistemi generativi in grado di produrre altri tipi di dati, come:

- *Musica e audio:* alcuni modelli di IA sono in grado di studiare e comprendere pattern musicali. Esempi di questa tipologia sono SoundStream di Google e Suno AI, sviluppato dall'azienda americana Suno.
- *Ricerca sul web:* Perplexity AI è un sistema di IA Generativa che unisce i concetti di chatbot e motore di ricerca, andando a cercare sul web dati inerenti alla richiesta.
- *Contenuti per videogiochi:* alcuni sistemi generano contenuti di diverso tipo (come modelli virtuali o modelli 3D) che risultano utili nella programmazione videoludica.

2.4 Classificazione per architettura

Un secondo criterio di classificazione per i sistemi di IA Generativa riguarda l'architettura. Abbiamo visto che alla base del funzionamento ci sono i meccanismi presentati nelle sezioni precedenti; le differenze sono, quindi, in questo caso, di tipo costruttivo.

2.4.1 Transformer

Il modello Transformer è stato introdotto in modo generale precedentemente ed è il modello più diffuso e utilizzato per i sistemi generativi.

Questo modello eccelle nell'elaborazione di dati sequenziali, specialmente nell'utilizzo degli LLM. L'innovazione principale apportata da questa architettura è il meccanismo di attenzione basato sui token, definito nelle sezioni precedenti. Questo sistema di auto-attenzione permette di svolgere l'elaborazione in parallelo, velocizzando notevolmente la procedura e consentendo versatilità e una grande gestione per sequenze lunghe.

A livello puramente strutturale e organizzativo, i modelli Transformer presentano nella loro struttura due componenti: un encoder e un decoder. L'encoder (in italiano, codificatore) si occupa di prendere l'input e trasformarlo simultaneamente tramite l'auto-attenzione, producendo vettori contenenti il significato contestualizzato di ogni parola. Il decoder (in italiano, decodificatore), invece, estrae le informazioni dai vettori prodotti dall'encoder e, basandosi su di essi, costruisce il risultato finale.

Esempi di modelli basati su questo tipo di architettura sono dati da: *Gemini* di Google, *Mistral AI* e *Claude* di Anthropic.

2.4.2 GAN

L'acronimo GAN sta per Generative Adversarial Networks, in italiano Reti Avversarie Generative, e fa riferimento a modelli composti da due Reti Neurali con due ruoli distinti: un generatore ed un discriminatore. Il generatore si occupa di creare dati falsi, mentre il discriminatore ha il compito di riconoscere ciò che è falso da ciò che è vero, riportandolo al generatore. Si crea, così, un ciclo di feedback che permette al generatore di migliorare e produrre ogni volta un elemento che sarà più simile al vero; allo stesso modo, anche il discriminatore migliora, imparando a riconoscere sempre più dettagli e imperfezioni. Il ciclo continua finché il generatore non sarà in grado di produrre un risultato che il discriminatore non è in grado di distinguere da quello veritiero.

Questa architettura risulta particolarmente utilizzata nella creazione di immagini e video, riuscendo a produrre contenuti altamente realistici. Un esempio di modello GAN è dato da *StyleGAN* di NVIDIA, specializzato nella produzione di immagini fotorealistiche.

2.4.3 VAE

Gli autoencoder variazionali (VAE) sono un'architettura di Reti Neurali utilizzata solitamente all'interno del Deep Learning per diverse attività, quali il rilevamento delle anomalie nelle immagini o il riconoscimento facciale.

La funzione principale di un autoencoder è quella di estrarre le informazioni fondamentali dai dati ed eliminare tutto ciò che risulta irrilevante. Anche in questo caso, a livello strutturale, un autoencoder sarà composto da un encoder e da un decoder. Una potenzialità dei VAE è che essi possono sintetizzare nuovi campioni di dati simili ai dati di addestramento originali.

Un esempio di VAE è dato da *DALL-E* di OpenAI: esso utilizza un VAE assieme a un Transformer per generare immagini da prompt testuali.

2.4.4 Diffusion model

I modelli di diffusione sono modelli generativi in cui le Reti Neurali vengono addestrate tramite il Deep Learning per permettere una diffusione "doppia", ovvero una in avanti e una all'indietro. La diffusione in avanti (Forward Diffusion) prende un campione chiaro e aggiunge distorsioni fino a che il dato risulta irriconoscibile. Il processo all'indietro (Backward

Diffusion) permette di invertire il procedimento precedente, prendendo l'immagine distorta e cancellando i rumori, costruendo un'immagine o un dato che sarà molto simile all'originale. Ovviamente la macchina sarà addestrata opportunamente per permettere queste azioni.

Anche in questo caso, questa architettura viene utilizzata specialmente nel campo delle immagini, con un livello di realismo che solitamente supera quello delle GAN. Esempi concreti basati sui Diffusion model sono: *Imagen* di Google, *Sora* di OpenAI e *MidJourney* dell'omonima azienda.

2.5 Classificazione per accessibilità

I sistemi di IA possono essere suddivisi anche in base al loro livello di accessibilità, distinguendo modelli aperti ed accessibili al pubblico e modelli privati soggetti a restrizioni. La decisione su quali scegliere fra questi due modelli non è mai banale, in quanto entrambe le tipologie offrono vantaggi e svantaggi.

2.5.1 Modelli Open Source

I modelli Open Source fanno riferimento a quei sistemi accessibili pubblicamente. L'IA Open Source offre il libero accesso sia al codice sorgente che alla documentazione. Ciò consente un elevato livello di personalizzazione, permettendo ai consumatori di modificarla e adattarla alle proprie necessità.

Generalmente, i sistemi aperti sono gratuiti, ma solitamente è necessario utilizzare tecnologie per adattare questi sistemi ai propri ambienti lavorativi. I costi, allora, sono nulli dal punto di vista delle licenze, ma potrebbero risultare elevati per il mantenimento e la gestione.

L'ultimo aspetto di cui tener conto riguarda la facilità d'uso. Difatti, nonostante siano completamente accessibili, tali sistemi richiedono un certo livello di conoscenza tecnica per essere utilizzati. Se un'azienda non avesse un team ben preparato, dovrebbe investire in specialisti per sfruttare questi sistemi, portando, di conseguenza, a un innalzamento dei costi.

Diversi sistemi di Intelligenza Artificiale Generativa adottano il modello Open Source, come *Mistral* e *Velvet AI*.

Esempio: Mistral. Mistral è un modello generativo Open Source autoregressivo sviluppato dall'azienda francese Mistral AI e rilasciato il 27 Settembre 2023 direttamente sul sito dell'azienda. Pochi giorni dopo, all'inizio di Ottobre 2023, è stato pubblicato anche un paper sul modello, intitolato "Mistral 7B" Jiang *et al.* [2023].

Mistral fa parte delle tecnologie che adottano l'architettura Transformer ed è un modello che presenta un ottimo rapporto qualità-prezzo. Essendo Open Source, presenta i vantaggi e gli svantaggi relativi a questa classe di modelli.

Mistral è in grado di lavorare solo con input di tipo testuale, caratteristica che lo rende, di fatto, meno utilizzato rispetto ad altri sistemi più conosciuti.

Esempio: Velvet AI. Velvet AI è un sistema di Intelligenza Artificiale Generativa sviluppato in Italia dall'azienda Almaxwave, il 29 Gennaio 2025.

Si tratta di un sistema multilingue in grado di gestire soltanto dati in input. L'azienda offre più versioni di tale modello, ognuna delle quali differisce dalle altre prevalentemente per la quantità di dati con cui è addestrata.

Questo sistema risulta, come altri di tipo Open Source, applicabile a diversi contesti, offrendo vantaggi riguardanti leggerezza e sostenibilità. Essendo un software recente, esso è conforme all'AI act adottato dall'Unione Europea ed è addestrato per ridurre i bias; inoltre,

tramite l'utilizzo di un algoritmo privato, pone particolare attenzione alla privacy dei dati personali.

2.5.2 Modelli Closed Source

I modelli Closed Source, chiamati anche modelli proprietari, contrariamente ai sistemi aperti, sono gestiti da aziende private che non rendono pubbliche le proprie risorse. Ciò che nei modelli Open Source era di dominio pubblico e accessibile, nei modelli Closed Source risulta soggetto a restrizioni e a un accesso limitato per poche classi di utenti.

Informazioni come il codice sorgente, il funzionamento interno e gli algoritmi utilizzati, quindi, saranno conosciuti soltanto dall'azienda proprietaria. Ciò, però, non significa che i sistemi di IA appartenenti a questa classe non possano essere utilizzati in alcun modo dall'esterno: molti sistemi proprietari mettono a disposizione interfacce o API³ per interagire con gli utenti, controllando le modalità di utilizzo. Diversi modelli adottano questa soluzione, come *Gemini* di Google o *GPT-4* di OpenAI, accessibili direttamente dal web o tramite le rispettive applicazioni.

L'Intelligenza Artificiale Closed Source viene solitamente vista come uno strumento commerciale, volto ad una monetizzazione. I dati sono privati e viene, in un certo senso, tutelata la proprietà intellettuale dell'azienda produttrice. Questo approccio offre vantaggi per gli utenti in termini di sicurezza e di prestazioni; allo stesso tempo, però, presenta costi più elevati dovuti a licenze e a un basso grado di adattabilità, vista l'impossibilità per i consumatori di accedere e modificare il codice sorgente.

2.6 Classificazione per modalità di input/output

I sistemi di Intelligenza Artificiale Generativa possono essere suddivisi, come già visto implicitamente finora, in unimodali e multimodali, in base al numero di tipi di dati elaborati in ingresso e in uscita.

2.6.1 Modelli unimodali

I sistemi unimodali sono in grado di elaborare una sola classe di dati alla volta, come un testo o un'immagine. Un modello unimodale testuale non sarà in grado di elaborare e produrre immagini, e viceversa per un modello unimodale fotografico.

Questa classe di sistemi risulta semplice ed economica, ma meno adattabile, vista l'incapacità di lavorare con più tipologie di dati simultaneamente.

L'unimodalità rappresenta una soluzione frequentemente adottata nell'ambito dei sistemi Open Source. Riprendendo gli esempi dell'apposita sezione, si può notare come entrambi siano proprio di tipo unimodale. Questa preferenza per la modalità singola riguarda principalmente motivi legati ai costi, alla complessità tecnica (dovuta specialmente all'apprendimento) e alle scelte aziendali. Nonostante questa propensione per questa classe di modelli, si stanno cominciando a produrre sistemi aperti di tipo multimodale, grazie a progetti innovativi e a conoscenze di base che aumentano sempre di più.

Oltre agli esempi citati in precedenza, un altro sistema di IA unimodale è LLaMA 3.

Esempio: LLaMA 3. LLaMA 3 è un modello linguistico ideato dall'azienda americana Meta AI e lanciato nel 2024, con la pubblicazione dell'articolo *The Llama 3 Herd of Models* Grattafiori *et al.* [2024]. Questo modello rappresenta l'evoluzione di LLaMA 2, il suo predecessore.

³API: le API sono insiemi di regole e protocolli che consentono ad applicazioni di comunicare e scambiare dati fra loro.

Esso adotta l'architettura Transformer con un addestramento basato su trilioni di dati. Assieme alla sua versione precedente rappresenta uno dei primi approcci nel campo dell'Intelligenza Artificiale Generativa, specializzato nella risoluzione di richieste unicamente testuali.

2.6.2 Modelli multimodali

I modelli multimodali, invece, consentono di gestire più tipologie di dati, come testi, immagini o audio. Proprio per questo presentano architetture più complesse e costose, ma che permettono ai sistemi di IA di rispondere in modo più approfondito.

L'approccio con molteplici modalità ha rappresentato un importante passo in avanti per l'Intelligenza Artificiale, date le precedenti difficoltà nel descrivere determinate problematiche o richieste utilizzando soltanto le parole.

Si consideri, ad esempio, il caso in cui un utente desideri generare un video. In presenza di un sistema unimodale basato esclusivamente su input testuale, l'utente descriverà a parole il contenuto desiderato e il modello produrrà un risultato basato unicamente su tali informazioni. In questo caso, l'interazione avviene attraverso una singola modalità di dati, ossia il testo. Diversamente, utilizzando un sistema multimodale, l'utente può combinare differenti tipi di input, ad esempio fornendo una descrizione testuale assieme a un'immagine o a un video di riferimento, nonché indicazioni sulla componente sonora. Il modello è, quindi, capace di integrare tali informazioni eterogenee per generare un contenuto che risulta più coerente e contestualizzato.

Conseguentemente a ciò che è stato appena accennato, si può, quindi, affermare che con i modelli multimodali risultano più semplici compiti come la modifica o il ritocco di contenuti già esistenti.

Questa classe di sistemi generativi è la più comune al momento, con esempi come *Gemini* o *DeepSeek*.

Esempio: DeepSeek. DeepSeek è un sistema di Intelligenza Artificiale Generativa multimodale molto potente. Esso offre un ottimo livello di generazione, risultando estremamente efficiente ed economico. Si distingue per la sua capacità di elaborare compiti complessi, specialmente nell'ambito della programmazione informatica e della matematica.

L'azienda cinese ha lanciato le prime versioni di DeepSeek fra Ottobre e Novembre del 2023; da quel momento, il sistema ha subito costanti aggiornamenti. All'inizio del 2025, è stato lanciato DeepSeek-R1, un modello orientato al ragionamento avanzato.

2.7 Classificazione per approccio all'apprendimento

I sistemi di Intelligenza Artificiale Generativa si possono classificare anche in base alle modalità con cui vengono addestrati. Questa classificazione, quindi, ci permette di creare delle sottoclassi in base alla tecnica di Machine Learning utilizzata per apprendere le conoscenze. I gruppi che emergono da questa differenziazione sono principalmente tre: sistemi supervisionati, non supervisionati e semi-supervisionati.

2.7.1 Modelli supervisionati

Nei modelli supervisionati, l'apprendimento avviene per mezzo dei dati etichettati. Ciò consente, come già visto nelle sezioni precedenti, di collegare gli input e gli output seguendo l'etichettatura. Questo tipo di modello offre un buon livello di automazione con un'elevata accuratezza.

Tra i modelli di IA che adottano questo tipo di apprendimento, troviamo *GPT-4* di OpenAI e *BERT* di Google, con quest'ultimo specializzato nella ricerca online.

2.7.2 Modelli non supervisionati

L'apprendimento non supervisionato, invece, prevede l'utilizzo di dati non etichettati. L'assenza di etichettatura non crea esempi ai quali la macchina può fare riferimento; in questo caso, quindi, l'elaboratore dovrà scoprire autonomamente gli schemi e le relazioni che intercorrono fra i dati. Solitamente, diversi sistemi di IA Generativa vengono addestrati prima seguendo il metodo supervisionato e successivamente con l'approccio non supervisionato, per migliorare le prestazioni e aumentare le competenze.

In generale, l'autonomia dei modelli non supervisionati offre maggiore flessibilità e scalabilità, a discapito, però, di un livello più basso di sensibilità e affidabilità a causa dell'assenza di elementi veritieri con cui effettuare un confronto. Inoltre, si potrebbero ottenere risultati imprecisi, dovuti al fatto che non si ha un fondo di verità alla base dell'addestramento.

L'apprendimento non supervisionato viene spesso utilizzato nei sistemi che si basano sull'architettura GAN.

2.7.3 Modelli semi-supervisionati

I modelli semi-supervisionati adottano un approccio che unisce i meccanismi dell'apprendimento supervisionato e di quello non supervisionato. Questo modello misto prevede che l'addestramento venga effettuato con una parte dei dati in input etichettata, assieme a un numero maggiore di dati non etichettati.

Questo approccio offre alcuni vantaggi apportati dai due modelli nativi: maggiore flessibilità, capacità di rilevare eventuali pattern nascosti grazie ai dati non etichettati e minore dipendenza dalle etichette rispetto ai modelli supervisionati.

Le problematiche che si creano, invece, riguardano prevalentemente una diminuzione della sensibilità in caso di rumori o bias, oltre a un grado di complessità superiore, vista la presenza di due categorie di dati differenti, che richiederanno un maggiore impiego di risorse.

Esistono modelli GAN semi-supervisionati, come riportato nell'articolo "Improved Techniques for Training GANs" Salimans *et al.* [2016]. Anche i modelli basati sull'architettura VAE possono essere addestrati in questa maniera, e ciò viene esplicitato nell'articolo "Semi-Supervised Learning with Deep Generative Models" Kingma *et al.* [2014].

Fra i modelli più conosciuti e accessibili, invece, non ci sono veri e propri sistemi basati sul funzionamento semi-supervisionato, bensì sistemi che utilizzano questa tipologia di apprendimento in alcune fasi. Ad esempio, la classe di modelli *GPT* sfrutta questa tecnica durante la fase di pre-addestramento.

2.8 Classificazione per competenza

Un ultimo criterio secondo cui si possono differenziare i sistemi di Intelligenza Artificiale riguarda il loro grado di specializzazione in un determinato contesto. Seguendo questa distinzione, troviamo due classi di modelli: i foundation model e i modelli verticali.

2.8.1 Foundation model

I foundation model, denominati anche modelli orizzontali, sono sistemi addestrati su enormi quantità di dati; la maggior parte dei sistemi di Intelligenza Artificiale Generativa fa

riferimento a questa classe. Lo scopo di questi modelli è quello di sviluppare una conoscenza ampia e trasversale in diversi settori, il che li rende utili nella risoluzione di svariati compiti e permette un elevato grado di adattabilità, scalabilità e versatilità. Ovviamente, adottando questo approccio, però, non si avrà un alto livello di specializzazione in un campo specifico.

Tra gli esempi più conosciuti di foundation model troviamo sistemi di IA Generativa citati precedentemente, come *Gemini*, *DeepSeek* e *Claude*.

2.8.2 Modelli verticali

I modelli verticali, invece, sono applicazioni specializzate sviluppate per mansioni o settori specifici. Offrono, quindi, un livello più elevato di specializzazione e precisione rispetto ai modelli orizzontali.

Esistono diversi sistemi che adottano questo approccio, con applicazioni in vari settori: dal campo medico a quello finanziario, passando per altri settori come quello giuridico e quello ingegneristico.

Tra i modelli verticali più rappresentativi troviamo, per esempio, MathGPT, specializzato nella risoluzione di quesiti matematici, oppure LegalGPT, orientato al contesto giuridico.

Esempio: MathGPT MathGPT è un modello di IA Generativa specializzato nella risoluzione e nella spiegazione di problemi matematici. Le risoluzioni vengono mostrate passo-passo e le competenze del sistema coprono argomenti matematici e algebrici sia di livello superiore che universitario. Si tratta di un modello attualmente a pagamento che offre 10 prompt gratuiti giornalieri.

Questo sistema è stato lanciato nel 2023 da Nour Gajjal e Yanni Kouloumbis, due studenti di informatica della Cornell University, con l'intento di offrire supporto e aiuto agli studenti nell'ambito matematico.

Esempio: LegalGPT LegalGPT è un modello generativo appartenente alla famiglia GPT di OpenAI, specializzato nella risoluzione di problematiche legate alla legge. Esso può analizzare documenti legali, interpretare e spiegare leggi e norme, fornire analisi e supporto, e ricercare casi simili a quello richiesto.

Esso fa parte dei GPT personalizzati, ossia delle specializzazioni di GPT sviluppate da terzi; pertanto, l'interfaccia utilizzata è la stessa di ChatGPT e vi si può accedere cercando "LegalGPT" sul web, oppure esplorando la sezione "GPT" nel sito del chatbot di OpenAI.

Ci sono, ovviamente, altri strumenti generativi nell'ambito del diritto accessibili online, come ChatLegal.

Case study sulla gestione dei testi

In questo capitolo verrà effettuato un caso di studio per testare le potenzialità e i limiti dei sistemi di Intelligenza Artificiale Generativa nell'ambito della produzione testuale. Il caso di studio sarà suddiviso in più richieste, ognuna delle quali aggiungerà un grado di difficoltà rispetto alla precedente. Tali richieste verranno sottoposte a quattro sistemi differenti, per valutare il modo di rispondere di ognuno ed evidenziare le principali differenze fra essi.

3.1 Definizione del caso di studio

3.1.1 Obiettivi dello studio

Lo studio consisterà nel testare quattro sistemi di Intelligenza Artificiale Generativa differenti, sottoponendoli alle stesse richieste per analizzarne il comportamento. Nel nostro caso, lo studio riguarderà la creazione e l'organizzazione di una dieta settimanale bilanciata.

Gli obiettivi del nostro case study, quindi, si possono dividere su due livelli: quello riguardante il singolo sistema e quello riguardante l'insieme dei modelli. In primis, per quel che riguarda i singoli sistemi, ognuno verrà sottoposto alle richieste, consentendo di analizzare le risposte di volta in volta. In questo modo si evidenzieranno i punti di forza delle risposte e le eventuali lacune di ogni modello.

Una volta effettuato lo studio per ogni sistema, invece, sarà possibile confrontare i risultati ottenuti con ciascun modello, consentendo di fornire un resoconto su quali siano i migliori in certi parametri e quali, invece, pecchino su altri.

3.1.2 Approccio scelto

L'approccio scelto prevede l'utilizzo di cinque prompt differenti per simulare situazioni più o meno realistiche relative alla creazione di una dieta il più verosimile possibile.

L'analisi può essere suddivisa in due fasi principali. La prima fase prevede che i sistemi rispondano a una richiesta molto generale, senza troppe pretese, in modo tale da valutare le loro scelte di fronte ad una piena libertà.

La fase successiva, invece, prevede l'aggiunta di vincoli e problematiche alla richiesta di partenza. Tali vincoli andranno a diminuire la creatività delle macchine, aumentando il grado di difficoltà per la generazione. L'aggiunta di queste restrizioni riguarderà situazioni perlo-

più verosimili e riscontrabili realmente, come vincoli di budget, intolleranze o tempistiche insufficienti per la preparazione anticipata dei pasti.

Questo approccio consente di visualizzare l'andamento delle risposte man mano che la difficoltà aumenta, permettendo di valutare parametri come la coerenza e l'adattabilità.

3.2 Sistemi scelti per lo studio

I sistemi scelti per lo studio, come già accennato nella sezione precedente, sono quattro: Gemini, DeepSeek, Mistral e Claude. Per ognuno di essi verrà data una descrizione generale, analizzando aspetti come l'interfaccia e le funzionalità messe a disposizione.

La scelta dei sistemi è stata dettata da alcune differenze relative ai parametri visti nelle sezioni precedenti (come l'accessibilità, le modalità di apprendimento o le strutture differenti seppur simili).

Inoltre, è importante sottolineare che, nel nostro caso di studio, Gemini fa riferimento al modello *Gemini Pro*, versione a pagamento e che offre una potenza di elaborazione ben più elevata rispetto al modello base. In questo modo, sarà possibile mettere a confronto modelli gratuiti con una versione più evoluta e non accessibile normalmente.

3.2.1 Gemini

Gemini è il sistema di Intelligenza Artificiale Generativa sviluppato da Google e pubblicato assieme all'articolo scientifico *Gemini: A Family of Highly Capable Multimodal Models Team et al.* [2023], come già citato nelle sezioni precedenti. Dal rilascio in poi, il modello è stato sempre in continuo aggiornamento, con conseguenti rilasci di nuove versioni, ognuna più potente rispetto alla precedente. L'ultima versione rilasciata e a cui faremo riferimento per il nostro caso di studio è Gemini 3, pubblicata alla fine del 2025 e che va a sostituire le versioni Gemini 1 e Gemini 2.

Gemini 3 rappresenta il modello più intelligente ideato dal colosso informatico statunitense e presenta alcuni aspetti innovativi rispetto al passato. Fra questi, i principali riguardano:

- *Modalità Deep Think*: modalità facoltativa che, se attivata, consente all'IA di ragionare a lungo su problemi complessi, simulando un ragionamento suddiviso in più step, simile alla mente umana. Tale modalità risulta particolarmente efficiente per problemi matematici e scientifici.
- *Multimodalità avanzata*: Gemini 3 interpreta testi, immagini, video, audio e codici in modo molto più efficiente rispetto alle versioni precedenti. Inoltre, consente una generazione interattiva, tramite la quale si possono generare grafici, animazioni SVG e interfacce web partendo da input testuali.
- *Integrazione con l'ecosistema Google*: la Versione 3 si collega a contenuti di Gmail, Google Calendar, Maps, YouTube e Foto, consentendo la gestione di attività.

La sostanziale novità, però, riguarda il cambiamento nell'approccio all'Intelligenza Artificiale. Difatti, con Gemini 3 si è passati da un approccio in cui il modello rappresenta un semplice assistente conversazionale, a una strategia in cui il sistema rappresenta un "agente operativo" in grado di elaborare richieste e compiti senza una continua necessità di supervisione da parte dell'utente. Ciò, per esempio, si nota proprio nella possibilità di interagire con strumenti come Google Calendar.

L'accesso al modello Gemini avviene tramite l'interfaccia web all'indirizzo "gemini.google.com", oppure tramite l'applicazione del sistema.

Nella Figura 3.1 si può vedere l'interfaccia una volta effettuato l'accesso alla pagina del sistema.

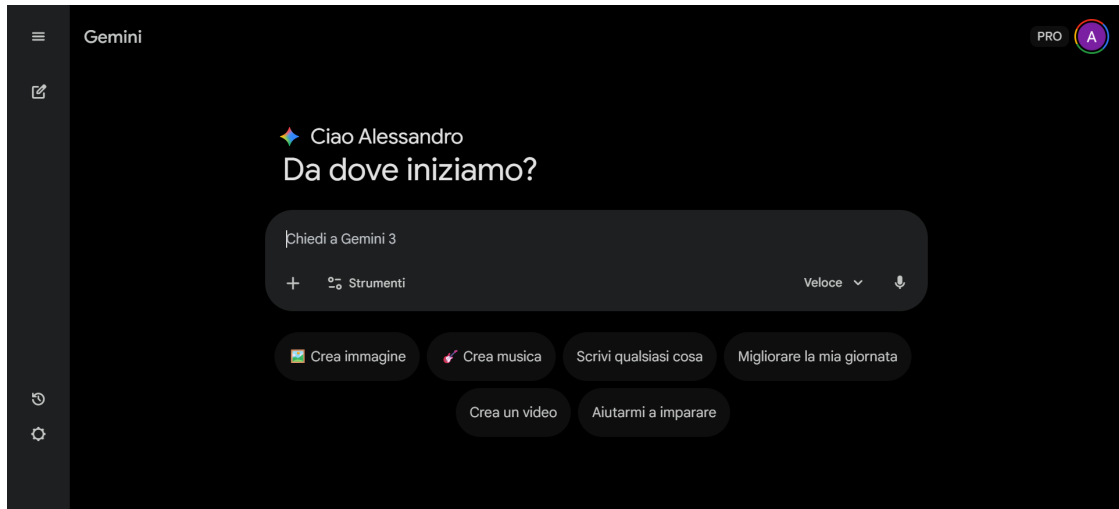


Figura 3.1: Interfaccia web di Gemini

Come si può notare, l'interfaccia offre dei suggerimenti su come utilizzare il sistema, con dei pulsanti posti sotto la casella di testo. Nella parte sinistra della casella troviamo invece il + e la sezione *strumenti*. Aprendo il +, ci troveremo di fronte a un menù a tendina che ci consentirà di passare eventuali immagini, codici o file, oppure collegarci a Google Drive o ad un Notebook.

Nella sezione strumenti, invece, sarà possibile scegliere quale tipo di dato vorremo produrre. Le possibilità che avremo sono:

- *Crea immagine;*
- *Canvas;*
- *Deep Research;*
- *Crea video;*
- *Crea musica;*
- *Apprendimento guidato.*

Nella parte destra, invece, troviamo il *microfono* per l'input vocale, e un ulteriore menù a tendina. Quest'ultimo serve per scegliere la modalità di ragionamento che si desidera utilizzare; le opzioni sono le seguenti:

- *Veloce:* è l'opzione di default, elabora le risposte in maniera rapida.
- *Ragionamento:* si tratta della modalità accennata precedentemente. Il ragionamento è più lento, ma consente di risolvere problemi di natura più complessa.
- *Pro:* utilizzabile solo da chi possiede la versione Pro. Modalità specializzata nella risoluzione matematica e scientifica.

Tramite la barra laterale è possibile accedere alla lista delle conversazioni avviate e a una sezione con i vari contenuti generati. I due tasti in basso, invece, consentono di accedere, rispettivamente, alla cronologia delle attività e alle impostazioni.

Come già accennato nella sezione precedente, nel nostro caso utilizzeremo la versione Pro, mentre, per quanto riguarda la modalità di ragionamento, lasceremo quella di default, ossia quella veloce.

3.2.2 DeepSeek

DeepSeek, come visto nelle sezioni precedenti, è il sistema di IA Generativa sviluppato dall'omonima azienda cinese. Il modello è stato soggetto a diversi aggiornamenti, che hanno portato a diverse versioni. Nel nostro studio la versione utilizzata è *DeepSeek-V3.2*, aggiornamento della versione V3 e rilasciata il 2 dicembre 2025 assieme all'articolo tecnico "*DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models*" DeepSeek-AI [2025].

Questo modello presenta alcune caratteristiche già citate per Gemini, come il ragionamento avanzato. In questo caso, però, come accennato nelle sezioni precedenti, il sistema è aperto e, di conseguenza, sarà possibile accedere al codice sorgente e modificarlo, qualora se ne abbia la necessità. Dal sito "*deepseek.com*", inoltre, si potrà scegliere se utilizzare il sistema tramite il relativo chatbot, oppure tramite le API. Per il nostro caso di studio è stata scelta la prima opzione, e quindi collegandosi all'indirizzo "*chat.deepseek.com*", in quanto più semplice ed accessibile. Un ulteriore accesso al modello è dato dall'applicazione.

Una peculiarità di DeepSeek è data dal *watermarking*: a differenza di altri modelli, esso applica una filigrana digitale¹ ai contenuti generati, in linea con le normative cinesi. Inoltre, sono presenti dei filtri di censura nel chatbot, che non consentiranno al sistema di rispondere a temi come la politica cinese, seguendo le normative del paese nativo.

Nella Figura 3.2 è riportata l'interfaccia web del modello. Si nota sin da subito un design semplice e minimale, con meno funzionalità rispetto a Gemini.

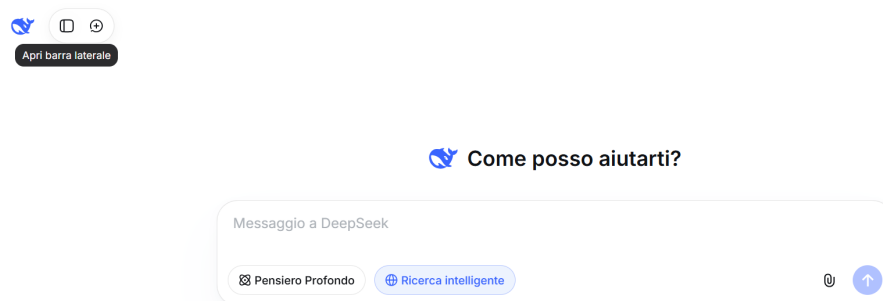


Figura 3.2: Interfaccia web di DeepSeek-V3.2

In alto a sinistra sono presenti due pulsanti, uno per aprire la barra laterale con la lista delle chat avviate, e un tasto che consente di avviare una nuova conversazione.

Per quanto riguarda la casella di testo, partendo dal lato destro, troviamo la graffetta, tramite la quale è possibile allegare alla richiesta immagini o documenti. Il numero massimo di allegati è 50, con un massimo di 100 Mb ciascuno. Il sistema, però, sarà soltanto in grado di estrarre il testo dai file allegati. Nella parte sinistra, invece, si notano subito due pulsanti:

¹Una filigrana digitale è un'informazione nascosta o visibile incorporata in un file digitale (immagini, video, documenti) per certificarne l'autenticità, la provenienza o la fonte.

- *Pensiero Profondo*: attivando questa modalità il sistema ragionerà in maniera più lenta e precisa.
- *Ricerca intelligente*: il sistema ricerca sul web quando necessario.

È importante sottolineare che il modello permette di attivare entrambe le modalità contemporaneamente, garantendo, pertanto, un elevato livello prestazionale.

3.2.3 Mistral

Mistral è un sistema di Intelligenza Artificiale, sviluppato dall'azienda francese Mistral, fondata nel 2023 da ex ricercatori di Google DeepMind e MetaAI. La versione a cui faremo riferimento per il nostro caso di studio è Mistral7B, le cui caratteristiche e particolarità si possono trovare all'interno del paper intitolato "Mistral 7B" Jiang *et al.* [2023].

Mistral, come già visto nelle sezioni precedenti, è un sistema Open Source che offre versatilità, buone prestazioni e un ottimo grado di adattabilità. Le lettere "7B" indicano che il modello è addestrato su 7 miliardi di parametri, un numero inferiore rispetto ad altri sistemi; tuttavia, nonostante ciò, Mistral offre prestazioni simili, se non superiori.

L'interfaccia web di Mistral prende il nome di "*Le Chat*" ed è accessibile all'indirizzo web "*chat.mistral.ai*". Tale interfaccia integra diverse funzioni e strumenti, consentendo all'utente di effettuare diverse azioni. La schermata iniziale dell'interfaccia è riportata nella Figura 3.3.

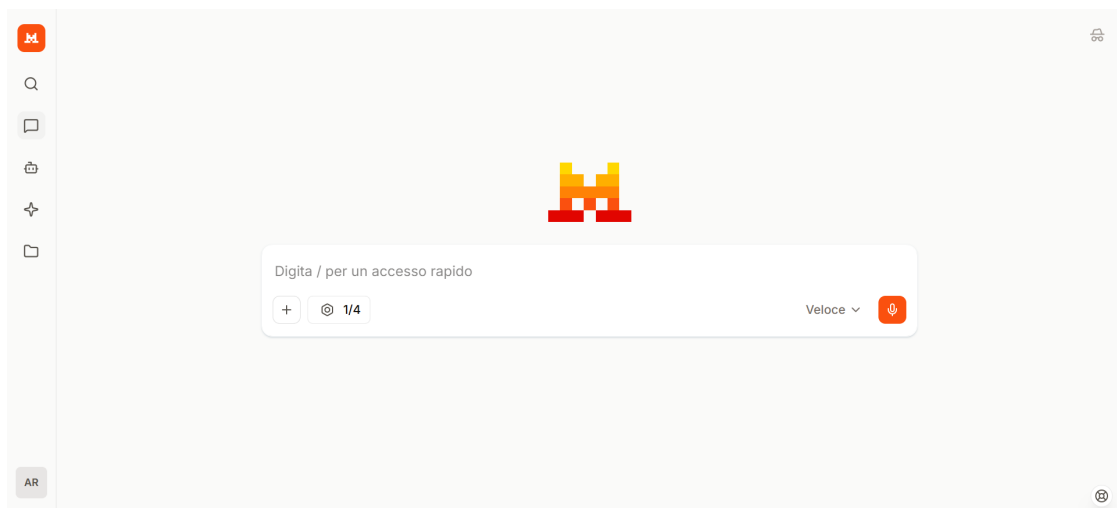


Figura 3.3: Interfaccia web di Mistral

Come si può notare, il design è semplice e pulito, ma sono presenti diversi pulsanti che offrono diverse funzionalità. Nella parte destra della schermata, troviamo due pulsanti, uno per la documentazione in basso, e uno per l'attività in incognito in alto. Cliccando sul salvagente, si aprirà un menù relativo alla documentazione dell'Intelligenza Artificiale, con voci come: documentazione, centro Assistenza, Guide pratiche, informative sulla privacy o documentazione API. Questo menù evidenzia la natura Open Source del modello.

La modalità in incognito, invece, avvia una conversazione privata e temporanea che non verrà memorizzata una volta chiusa la pagina. La finestra in incognito che si apre in questo caso è riportata nella Figura 3.4.

La barra laterale, invece, offre varie funzionalità aggiuntive, differenti rispetto ad altri modelli. Il primo simbolo che troviamo è la *lente di ingrandimento*, che fa riferimento alla funzione *cerca*, che consentirà di ricercare la funzionalità voluta tramite una casella di testo.



Figura 3.4: Finestra che si apre nella modalità in incognito di Mistral

L'icona del *robot*, invece, fa riferimento alla sezione dell'interfaccia relativa agli *agenti*. Un agente è un assistente intelligente basato sugli LLM. Essi sono programmabili e addestrabili dagli utenti, e sono pensati per automatizzare task, migliorare la produttività e svolgere determinate funzionalità, come la ricerca sul web. Tramite questa sezione dell'interfaccia Le Chat è possibile inserire nuovi agenti e gestirli, dando, così, agli utenti un'importante funzionalità aggiuntiva. All'interno della sezione vengono proposti da Mistral stesso alcuni agenti preimpostati, come il Data Analyst.

L'ultimo simbolo, quello della *stella*, fa riferimento alla sezione *Intelligence*. Quest'area è progettata per personalizzare l'assistente IA, in modo tale da renderlo più contestualizzato, integrato con elementi esterni e focalizzato su compiti specifici o aziendali. L'area è divisa in quattro categorie:

- *Ricordi*: l'IA mantiene il contesto e il ricordo delle sessioni precedenti; in questo modo essa impara dettagli, informazioni e memorizza preferenze.
- *Connettori*: i connettori consentono di integrare l'interfaccia Le Chat con applicazioni terze. Tramite essi, ad esempio, si potrebbe collegare Google Drive.
- *Librerie*: l'aggiunta di una libreria consente di caricare documenti oppure di analizzare siti web, consentendo di creare una conoscenza personalizzata.
- *Istruzioni*: questa parte è spesso legata al concetto di agenti visto in precedenza. Aprendo la sezione, si apre una finestra in cui si possono impostare direttive specifiche su come l'IA dovrà comportarsi e rispondere. Si possono definire il ruolo dell'agente, il tono di voce e gli obiettivi da raggiungere. Ad esempio, si potrebbe impostare un tono educato, maturo o poetico.

L'ultimo aspetto peculiare di Le Chat riguarda il logo. Premendo il pulsante del logo, infatti, si può accedere sia alla documentazione, sia ad *AI Studio*, ossia la piattaforma destinata agli sviluppatori per il test e la messa a punto di agenti personalizzati.

3.2.4 Claude

Claude è il sistema di Intelligenza Artificiale sviluppato dall'azienda americana Anthropic. Esso è stato lanciato nel corso del 2023, come alternativa a modelli come Gemini o ChatGPT. Claude è una famiglia di modelli, tra i quali si annoverano:

- *Opus*: modello utilizzato principalmente per la realizzazione di progetti ambiziosi.

- *Sonnet*: sistema efficiente per la risoluzione di problemi quotidiani.
- *Haiku*: è il più veloce fra i tre ed offre risposte rapide.

Anthropic offre un'interfaccia per Claude direttamente accessibile dal web, all'indirizzo "claude.ai". Tale interfaccia è consultabile nella Figura 3.5.

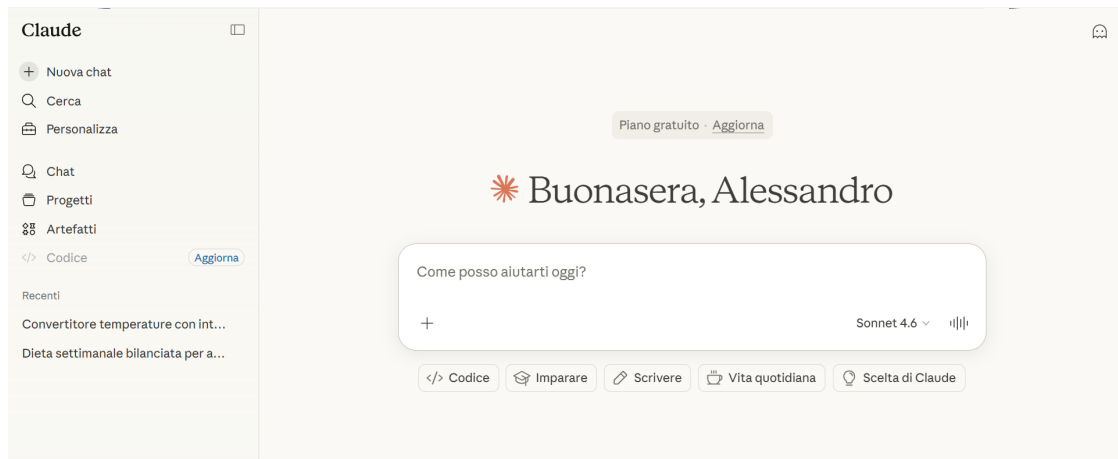


Figura 3.5: Interfaccia web del sistema Claude

Similmente a Gemini, l'interfaccia offre degli spunti all'utente. Nel lato destro è possibile scegliere la classe che si vuole utilizzare; per il nostro caso di studio abbiamo usato *Sonnet 4.6*. Il pulsante +, invece, apre un menù a tendina che consente azioni come l'aggiunta di allegati, l'aggiunta di connettori e competenze alla conversazione, la ricerca sul web e la scelta dello stile per la risposta (sia fra stili preimpostati, sia fra stili programmabili dall'utente).

Il menù della barra laterale, invece, mette a disposizione, oltre ad azioni standard come la ricerca e la visualizzazione delle conversazioni, le seguenti funzionalità:

- *Personalizza*: si possono connettere applicazioni esterne e creare nuove competenze per l'IA.
- *Progetti*: si possono creare cartelle e progetti, in modo tale che Claude diventi specializzato in un determinato compito o lavoro.
- *Artefatti*: è una delle peculiarità di Claude. In questa sezione sono presenti diversi modelli sviluppati tramite Claude e messi a disposizione per gli utenti. Ogni utente, inoltre, creerà degli artefatti in base alle richieste che pone.

L'importanza degli artefatti si nota durante l'esecuzione delle richieste. Claude, difatti, oltre a far apparire gli elementi generati nella chat, genera dei documenti in cui è possibile visualizzare il contenuto generato. Aprendo i documenti, si aprirà una sezione laterale a destra della chat. Questa finestra aggiuntiva consente di visualizzare, modificare e iterare il contenuto in tempo reale. Tale funzionalità risulta essere molto utile, ad esempio, per la generazione di codice o di grafici.

3.3 Definizione delle richieste

Come definito in precedenza, il caso di studio riguarderà la creazione di una dieta settimanale. Lo studio prevederà l'utilizzo di più prompt testuali, con un grado di difficoltà sempre maggiore per gli assistenti generativi.

Nel seguito definiamo e analizziamo le cinque richieste utilizzate per il nostro studio.

3.3.1 Prompt iniziale

Prompt 1 - Situazione generale

Ciao, prepara una dieta settimanale bilanciata per una persona adulta, senza esigenze particolari. Colazione, pranzo e cena per la settimana.

La richiesta è del tutto generale, senza vincoli di alcun tipo. In questo modo si può valutare come un sistema risponda di fronte ad una situazione in cui ha piena libertà decisionale.

3.3.2 Vincolo di budget

Prompt 2 - Aggiunta di un vincolo di budget

Ora la persona che deve mettersi a dieta ha un budget massimo di 50€ per tutta la settimana. Mantieni la dieta bilanciata e varia rispettando il budget.

In questo secondo prompt è stato aggiunto un primo vincolo riguardante il budget. Si analizzerà se il sistema riesca a rispettare la limitazione di 50€, mantenendo la dieta bilanciata e realmente utilizzabile.

3.3.3 Aggiunta di intolleranze

Prompt 3- Aggiunta intolleranze

La persona è anche intollerante al lattosio, e preferisce evitare il pesce se possibile. Ricrea la dieta anche in base a queste informazioni, mantenendo il budget dei 50 euro.

Il terzo livello di difficoltà per il caso di studio prevede l'aggiunta di un'intolleranza al lattosio, situazione comune e verosimile. Inoltre, viene specificato che la persona preferisce evitare pesce se possibile, anche per risparmiare sul budget. Si analizzerà, a questo punto, il comportamento del sistema di fronte ad un insieme ristretto di alimenti utilizzabili.

3.3.4 Peso e vincolo di casualità

Prompt 4 - Aggiunta di vincolo sul peso e aggiunta di casualità

Ci sono altri aspetti che non erano stati specificati inizialmente:

- Il budget settimanale, a causa di alcuni problemi, scende a 35€.
- Un giorno a caso della settimana non può cucinare nulla.
- Il peso da perdere è di circa 1kg a settimana.

Con la quarta richiesta il sistema si trova a dover affrontare la problematica della perdita di peso, e l'aggiunta di un vincolo di casualità, riguardante l'impossibilità di cucinare per un giorno a caso. Inoltre, è stato diminuito il budget, aumentando ancor di più il grado di difficoltà.

Questa situazione, seppur abbia ancora alcuni aspetti verosimili, presenta problematiche difficilmente reperibili in casi realistici (la perdita di peso, per esempio, risulta essere volutamente esagerata). Verrà evidenziato il grado di adattabilità dei sistemi di fronte una situazione simile.

3.3.5 Richiesta finale

Prompt 5 - Situazione surreale

La dieta va bene, ma la situazione è cambiata nuovamente:

- La persona vuole perdere 1kg a settimana, forse 0.5kg, o forse 0.3kg, è indecisa e varia in base a se farà attività fisica o meno.
- Non conosce i turni di lavoro e non sa quando sarà a casa e quando avrà tempo di cucinare.
- Non vuole ripetere più di due volte a settimana lo stesso ingrediente.
- Non ha accesso ad un grande supermercato nelle vicinanze, solamente un piccolo negozio di alimentari.
- Potrebbe saltare uno dei pasti di tanto in tanto, ma non è dato sapere quando e quale.

L'ultima richiesta pone il sistema di fronte una situazione del tutto surreale; questo prompt, difatti, è stracolmo di incertezze e casualità, difficilmente riscontrabili tutte assieme in un contesto reale. Questo approccio ci consentirà di osservare se i sistemi, di fronte una situazione così complicata e intricata, saranno in grado di fornire comunque una soluzione consona e sensata.

3.4 Analisi con Gemini

In questa sezione, analizzeremo il comportamento di Gemini per il nostro caso di studio.

3.4.1 Risposta al primo prompt

Utente

Ciao, prepara una dieta settimanale bilanciata per una persona adulta, senza esigenze particolari. Colazione, pranzo e cena per la settimana.

Gemini

Certamente! Ecco una proposta di dieta settimanale bilanciata pensata per una persona adulta sana e senza esigenze alimentari particolari.

Questa dieta è un modello basato sui principi della dieta mediterranea, che privilegia verdure, frutta, cereali integrali, legumi, pesce, e grassi sani (come l'olio d'oliva), limitando carni rosse e dolci.

Disclaimer Importante: Questo è un esempio generico. Le esigenze caloriche e nutrizionali variano in base a età, sesso, peso, altezza e livello di attività fisica. Per un piano alimentare personalizzato e sicuro, è sempre consigliato consultare un dietologo o un nutrizionista.

Esempio di Dieta Settimanale Bilanciata

Giorno	Colazione	Pranzo	Cena
Lunedì	Yogurt greco (o bianco intero) con 30g di cereali integrali/muesli non zuccherato e un frutto fresco (es. kiwi).	Pasta integrale (80g) al pomodoro fresco e basilico + Insalata mista con carote.	Petto di pollo (120g) alla piastra con contorno di broccoli lessati e olio EVO.
Martedì	2 fette biscottate integrali con un velo di marmellata e una tazza di latte parzialmente scremato (o bevanda vegetale).	Passato di verdure miste (senza patate) + Uova (2) strapazzate/sode + 50g di pane integrale.	Zuppa di legumi (ceci/fagioli) con un cucchiaino di olio EVO a crudo + Spinaci saltati.
Mercoledì	Porridge preparato con 40g di fiocchi d'avena, acqua/latte e un pizzico di cannella + Mezza mela a pezzi.	Riso integrale (80g) con verdure di stagione (es. zucchine e peperoni) + 1 cucchiaino di parmigiano.	Pesce azzurro (es. sgombero o alici, 150g) al forno o in padella + Patate lesse (150g).
Giovedì	Frullato con latte/yogurt, banana e una manciata di spinaci freschi + 4 mandorle	Insalata di farro o orzo (70g) con pomodorini, cetrioli e mozzarella light (50g)	Fesa di tacchino (120g) a fette con contorno di cavolfiore al vapore condito con olio e limone.
Venerdì	200g di frutta di stagione a pezzi con 100g di ricotta fresca.	Merluzzo (150g) al cartoccio con olive e capperi + Pane integrale (50g) + Verdure grigliate	Pizza margherita sottile (consumata fuori casa o fatta in casa con poco olio) oppure Lascia libera la cena, ma con moderazione.
Sabato	Pane integrale tostato (50g) con avocado a fette e un pizzico di sale e pepe	Polenta (150g cotta) con sugo semplice di funghi o ragù leggero + Verdure cotte	Carne rossa magra (es. lonza di maiale o manzo magro, 100g) alla griglia + Insalata verde.
Domenica	Pancake integrale (uova, farina integrale e latte) con sciroppo d'acero/frutta.	Piatto libero (es. pasto della tradizione familiare), cercando l'equilibrio (porzioni moderate di tutto).	Insalatona ricca con tonno al naturale, mais, pomodori, olive, uova sode e crostini integrali.

Linee Guida Aggiuntive

Idratazione

- Bevi almeno 1.5 - 2 litri di acqua al giorno.
- Limita bevande zuccherate, alcoliche e i succhi di frutta confezionati.

Spuntini (Metà mattina e Metà pomeriggio)

Per non arrivare affamati ai pasti principali, puoi inserire 1-2 spuntini sani:

- Un frutto fresco.
- Uno yogurt naturale.
- Una manciata (circa 15-20g) di frutta secca (mandorle, noci, nocciole).
- Un piccolo centrifugato di verdura.

Condimenti e Cottura

- Utilizza l'Olio Extra Vergine d'Oлива (EVO) come unico grasso da condimento, da aggiungere a crudo e con moderazione (circa 3 cucchiaini al giorno).
- Preferisci metodi di cottura sani: vapore, bollitura, al forno, alla griglia.
- Riduci l'uso del sale; usa spezie ed erbe aromatiche per insaporire i piatti.

Carboidrati

- Privilegia sempre i cereali integrali (pasta, riso, pane, orzo, farro) che forniscono più fibre.
- Cerca di variare le fonti di carboidrati.

Spero che questo schema ti sia utile! C'è qualche giorno della settimana o pasto specifico su cui vorresti dei suggerimenti di ricette veloci?

Analizzando la risposta, notiamo che Gemini risponde in maniera consona e adatta alla prima richiesta del nostro caso di studio. La risposta risulta essere particolarmente buona, con un ottimo livello di varietà alimentare e, persino, le dosi indicate per gli alimenti. Inoltre, il sistema fornisce anche dei consigli utili per altri aspetti, come idratazione e spuntini, elementi che non erano richiesti nel prompt.

3.4.2 Risposta ai vincoli di budget e alle intolleranze

Vediamo ora come si comporta il sistema di fronte all'aggiunta di vincoli riguardanti il budget e le intolleranze. Partiamo vedendo il prompt relativo al budget settimanale.

Utente

Ora la persona che deve mettersi a dieta ha un budget massimo di 50€ per tutta la settimana. Mantieni la dieta bilanciata e varia rispettando il budget.

Gemini

Questa è un'ottima sfida! Rispettare un budget settimanale di 50€ richiede attenzione alle scelte e la riduzione degli sprechi.

Per mantenere la dieta bilanciata, ci concentreremo su fonti proteiche economiche (legumi, uova, pollo/tacchino, pesce azzurro economico), cereali integrali base e verdure/frutta di stagione e a lunga conservazione.

Dieta Settimanale a Basso Budget (Max 50€)

Giorno	Colazione	Pranzo	Cena
Lunedì	Avena e Banana: Porridge (30g fiocchi d'avena) con acqua/latte economico e mezza banana.	Pasta e Legumi: Pasta (80g) con lenticchie in scatola e un cucchiaino di passata di pomodoro.	Pollo e Carote: Pollo (120g) lessato o alla piastra (taglio economico come il fuso o la coscia) + Carote bollite.
Martedì	Avena e Banana: Porridge con acqua/latte e mezza banana.	Avanzo Pollo/Carote o Zuppa di Legumi (vedi cena precedente) con pane raffermo.	Uova e Patate: Frittata semplice con 2 uova e prezzemolo + Patate lesse (150g).
Mercoledì	Pane e Marmellata: Pane raffermo tostato (50g) con un velo di marmellata economica + Tazza di latte economico (o bevanda vegetale).	Riso e Verdure: Riso (80g) con zucchine o cavolo (di stagione e economici) saltati in padella.	Sgombro Economico: Sgombro in scatola al naturale (sgocciolato, 120g) + Pane (50g) + Insalata verde (es. lattuga).
Giovedì	Pane e Marmellata: Pane tostato con marmellata + Tazza di latte.	Passato di Verdure: Avanzo Riso/Verdure oppure Passato di pomodoro e legumi (ceci in scatola) con pane.	Tacchino/Fagioli: Tacchino (120g, fesa a cubetti, molto economico) saltato con fagioli in scatola.
Venerdì	Yogurt e Cereali: Yogurt bianco economico (125g) con 20g di cornflakes/cereali base non zuccherati.	Zuppa Spessa: Brodo vegetale con patate (150g), carote e un cucchiaino di farina di mais (per rendere il piatto più saziante).	Cena Povera: 2 uova sode/strapazzate + Pane (50g) + Pomodori o cetrioli freschi (se economici).
Sabato	Yogurt e Cereali: Yogurt bianco economico con cornflakes.	Pasta e Tonno: Pasta (80g) con tonno in scatola (sott'olio sgocciolato, la marca base è economica) e limone.	Coscia di Pollo: Coscia di pollo (taglio economico, 150g) al forno con spezie + Brodo di cottura con pane.
Domenica	Ricotta e Frutta: Ricotta (100g, spesso in offerta) con frutta di stagione (es. arance/mele) e un pizzico di cannella.	Pizza Fatta in Casa: Pizza fatta con farina economica, lievito e passata di pomodoro (molto economica e saziante).	Avanzi/Insalata: Grande insalatona con avanzi proteici (pollo, sgombro) o legumi e verdure a basso costo.

Spesa settimanale e stima (Budget 50€)

Per rispettare i 50€, devi concentrarti su prodotti a lunga conservazione e offerte:

Categoria	Prodotto Esempio	Quantità/Peso	Costo stimato
Cereali	Pasta/Riso/Avena integrali	2kg	3,00€
Legumi	Lenticchie/Ceci/Fagioli (secchi o in latta, in offerta)	1 kg secchi / 4 in latta	4,00€
Proteine animali	Pollo (fusi/cosce) / Tacchino (fesa) / Sgombro in scatola / Tonno in scatola	1kg / 2 scatolette	10,00€
Uova e latticini	Uova (cartone da 10) / Yogurt bianco base / Ricotta	1 conf. / 500g / 250g	6,00€
Verdura e frutta	Patate / Carote / Cipolle / Cavolo / Frutta di stagione	4 kg	12,00€
Dispensa base	Olio EVO (piccolo) / Sale / Marmellata base / Passata di pomodoro / Spezie base	Varie	10,00€
Totale stimato			~45,00€

Consigli per Mantenere il Budget e la Qualità

1. Stop agli sprechi: Pianifica i pasti usando gli avanzi. Non buttare il pane raffermo (usalo per

bruschette, o inzuppato nel brodo).

2. Acquista Secco: I legumi secchi costano meno di quelli in scatola (anche se richiedono ammollo e cottura).
3. Verdure di Stagione: Compra solo verdure di stagione: sono più economiche e nutrienti (es. carote, patate, cavoli, cavolfiori, mele, arance in inverno).
4. Marche Base: Scegli i prodotti a marchio del supermercato o le marche più economiche (spesso la differenza nutrizionale è minima).
5. Controlla i Volantini: Fai la spesa dove ci sono offerte sulla carne e sul pesce fresco/surgelato, che sono i prodotti più costosi.

Posso aiutarti a trovare una ricetta specifica e molto economica per la cena di stasera?

La risposta al vincolo di budget risulta essere buona. Il sistema ha adattato la dieta al vincolo aggiunto, rispettando perfettamente la richiesta. Come si evince dalla seconda tabella del prompt, inoltre, Gemini si è occupato di fornire una stima economica per la spesa, semplificando il compito dell'utente e lasciando un margine di 5€. Infine, il modello ha proposto dei consigli per mantenere il budget e la qualità della dieta, mostrando un buon livello di intraprendenza.

Vediamo ora come si è comportato Gemini con il terzo prompt, relativo alle intolleranze.

Utente

La persona è anche intollerante al lattosio, e preferisce evitare il pesce se possibile. Ricrea la dieta anche in base a queste informazioni, mantenendo il budget dei 50 euro.

Gemini

Ottima precisazione! Integrare l'intolleranza al lattosio e la preferenza di evitare il pesce (eliminando i prodotti più costosi) è perfettamente in linea con il budget di 50€. Per sostituire latticini e pesce, ci concentreremo maggiormente su legumi, uova e carni bianche/rosse magre come fonti proteiche economiche, mantenendo un alto apporto di verdure e cereali integrali.

Dieta Settimanale a Basso Budget (Max 50€) - Lactose-Free e No Fish

Giorno	Colazione (no lattosio)	Pranzo	Cena
Lunedì	Avena e Mela: Porridge (30g fiocchi d'avena) con acqua o una bevanda vegetale economica (es. riso/avena) + Mela a pezzi.	Pasta e Legumi: Pasta integrale (80g) con passata di pomodoro e lenticchie (in scatola o secche).	Pollo e Carote: Pollo (120g - fuso/coscia, tagli economici) lessato o alla piastra + Carote lesse.
Martedì	Avena e Mela: Porridge con acqua/bevanda vegetale + Mela a pezzi.	Avanzo Pollo/Carote o Zuppa di Lenticchie (avanzo pranzo) con pane raffermo.	Uova e Spinaci: Frittata semplice con 2 uova e spinaci (anche surgelati, molto economici) + Pane integrale (50g).
Mercoledì	Pane e Marmellata: Pane raffermo tostato (50g) con marmellata economica + Tazza di tè/caffè (con bevanda vegetale o acqua).	Riso e Verdure: Riso (80g) con cavolo o cavolfiore (verdure di stagione e a basso costo) saltati.	Tacchino e Patate: Fesa di tacchino (120g, a cubetti) saltata con limone e aromi + Patate lesse (150g).
Giovedì	Pane e Marmellata: Pane tostato con marmellata + Tazza di tè/caffè.	Insalata di Ceci: Ceci in scatola (metà lattina) o cotti in precedenza con pomodori, cetrioli e olio EVO.	Minestrone di Fagioli: Zuppa densa con fagioli in scatola (o secchi) e verdure miste + Pane tostato.
Venerdì	Frutta e Semi: Banana o Arancia + 4 Noci o Mandorle (piccola porzione per il costo).	Zuppa Contadina: Brodo vegetale con patate, carote e farina di mais (per rendere il piatto più saziante).	Pasta Aglio e Olio e Verdura: Pasta (80g) condita con aglio, olio e peperoncino + Cavolfiore al vapore.
Sabato	Pane e Avocado/Olio: Pane tostato (50g) con un velo di olio EVO e sale oppure un quarto di avocado (se economico).	Burger di Legumi: Polpette fatte con lenticchie e pane raffermo (o riso) + Insalata verde (lattuga economica).	Carne Macinata: Carne macinata di manzo (100g, taglio magro ed economico) usata per un sugo o polpette + Patate arrosto (150g).
Domenica	Pancake Veggie: Pancake fatti con uova, farina integrale e bevanda vegetale (es. riso) + Frutta fresca.	Piatto Libero Semplice: Pasto della tradizione familiare, cercando l'equilibrio (es. porzione moderata di pasta al ragù vegetale/di carne).	Insalatona Ricca: Grande insalatona con uova sode, mais, pomodori, olive, legumi misti e pane tostato.

Strategie di Spesa e Sostituzioni Cruciali (Budget 50€)

Le sostituzioni per l'intolleranza al lattosio e l'esclusione del pesce sono vantaggiose per il budget, poiché i sostituti proteici sono spesso più economici.

Sostituzione richiesta	Soluzione per la dieta	Implicazione sul budget
Lattosio	<i>Eliminazione di:</i> Latte vaccino, yogurt greco/intero, ricotta, mozzarella. <i>Uso di:</i> Acqua, tè/caffè, bevanda di riso/avena (economica), yogurt vegetale (solo se in offerta, altrimenti evitabile).	Risparmio (il latte vegetale è più costoso, quindi lo limitiamo solo all'avena).
Pesce	<i>Eliminazione di:</i> Sgombro, tonno, merluzzo. <i>Uso di:</i> Più legumi, uova, pollo e tacchino.	Grande Risparmio (il pesce, anche in scatola, incide sul budget più di uova e legumi).

Per un budget di 50€, concentra gli acquisti su:

1. *Proteine:* Confezione di uova (10-12 pezzi), cosce/fusi di pollo, fesa di tacchino, carne macinata di manzo (tutti in offerta). Legumi secchi (più economici) o in scatola di marca base.
2. *Carboidrati:* Pasta/riso/pane integrali di marca economica, fiocchi d'avena.
3. *Verdura/Frutta:* Patate, carote, cipolle, cavolo, cavolfiore, spinaci surgelati, mele, arance (tutto di stagione e a lunga conservazione).

Consiglio Extra Intolleranza: Quando prepari il porridge o le ricette che richiedono latte (come i pancake), usa una bevanda di riso (spesso la più economica tra le vegetali) o, se il sapore non è un problema, semplicemente acqua e insaporisci con cannella e frutta.

Facendo un'analisi della risposta al nostro terzo prompt, si evince che, anche in questo caso, il sistema si è comportato adeguatamente, soddisfacendo i vincoli imposti. La dieta è stata modificata correttamente ed è stata proposta una nuova strategia per la spesa, consultabile nella seconda tabella del prompt. Il sistema, inoltre, ha nuovamente suggerito, di sua spontanea iniziativa, dei consigli riguardanti la spesa e alcune alternative per l'intolleranza.

3.4.3 Risposta ai vincoli sul peso e sulla casualità

In questa sezione analizziamo l'andamento di Gemini di fronte al quarto prompt, ossia quello che introduce vincoli riguardanti la perdita di peso dell'individuo che effettua la dieta, e la casualità nel cucinare. Inoltre, per mettere ancor più in difficoltà il sistema, il budget viene ridotto ulteriormente.

Utente

Ci sono altri aspetti che non erano stati specificati inizialmente:

- Il budget settimanale, a causa di alcuni problemi, scende a 35€.
- Un giorno a caso della settimana non può cucinare nulla.
- Il peso da perdere è di circa 1kg a settimana.

Gemini

Capisco le nuove restrizioni. Abbassare il budget a 35€ e includere un giorno in cui non si può cucinare (il "giorno libero dal fornello") rende la sfida molto più intensa, soprattutto unita all'obiettivo di perdita di peso di 1 kg a settimana (che richiede un deficit calorico).

Nota Medica: L'obiettivo di perdere 1 kg a settimana è significativo e questa dieta è pensata per essere ipocalorica. È fondamentale bere molta acqua e consultare un medico o dietologo per assicurarsi che il regime sia adatto alla propria condizione fisica.

Per rispettare il budget così stretto e l'obiettivo di dimagrimento, massimizzeremo l'uso di legumi, uova e verdure/frutta di stagione e useremo solo tagli di carne molto economici e porzioni controllate.

Dieta Iper-Economica (Max 35€) e Ipocalorica (L.F., No Fish, -1kg/settimana)

Giorno	Colazione (Max 250 Kcal)	Pranzo (Max 350 Kcal)	Cena (Max 400 Kcal)
Lunedì	Avena e Mela: Porridge (30g fiocchi d'avena) con acqua/bevanda vegetale economica + 1 Mela piccola.	Pasta Light: Pasta integrale (60g) con passata di pomodoro (2 cucchiaini) e origano + Insalata verde.	Pollo Lessato: Pollo (100g, fuso o coscia, carne scura magra) lessato + Brodo di pollo + Carote bollite.
Martedì	Avena e Mela: Porridge con acqua/bevanda vegetale + Mela piccola.	Uova e Pane: 2 Uova sode/strapazzate in padella antiaderente + Pane integrale (30g) + Pomodori freschi.	Zuppa di Legumi: Zuppa densa con lenticchie e verdure miste (carote, cipolle, cavolo) + Olio EVO a crudo.
Mercoledì	Pane e Marmellata: Pane tostato (30g) con marmellata light + Tazza di tè/caffè.	Riso e Cavolo: Riso integrale (60g) con cavolo o cavolfiore cotto al vapore e un cucchiaino d'olio.	Tacchino Magro: Fesa di tacchino (100g) alla piastra con spezie e limone + Patate lesse (120g).
Giovedì	Pane e Marmellata: Pane tostato con marmellata light + Tazza di tè/caffè.	Insalata di Ceci: Ceci (70g cotti) con carote e cipolle crude, conditi con olio e aceto/limone.	Avanzo Pollo/Tacchino (100g) o Brodo di verdure con poco riso (30g).
Venerdì	Frutta e Semi: Banana + 4 Mandorle o Noci (porzione minima).	Insalatona Saziante: Lattuga, pomodori, cetrioli, mais (30g) e un piccolo avanzo proteico (es. 1 uovo sodo).	Pasta Aglio e Olio: Pasta (60g) condita con aglio, olio e peperoncino + Cavolfiore bollito o spinaci surgelati.
Sabato(Giorno senza Fornello)	Pane e Burro di Arachidi L.F.: Pane integrale (30g) con un velo di burro di arachidi naturale (controllare l'etichetta) + Tazza di tè.	Piatto Freddo: Insalata con mezza lattina di fagioli rossi, pomodori, cetrioli e aceto + Pane (30g).	Uova Sode Fredde: 2 uova sode preparate la sera prima + Insalata verde (lattuga) + Carote crude tagliate a bastoncino.
Domenica	Pancake Veggie: Pancake (fatti con uova, farina e bevanda vegetale) + Arancia a fette.	Polpette Economiche: Polpette di carne macinata (70g) o legumi + Patate lesse (100g) o passata di pomodoro.	Insalatona Bilanciata: Uova sode, avanzi di carne/legumi, verdure a basso costo, condita con olio e limone.

Strategia per Rispettare i 35€ e Perdere Peso

1. Controllo Assoluto del Budget (35€)

- Proteine: L'80% delle proteine deve provenire da uova e legumi (acquistare secchi per risparmiare, ma richiedono ammollo/cottura) e i tagli di carne più economici (fusi, cosce, carne macinata base, fesa di tacchino).
- Verdure/Frutta: Solo prodotti di stagione e con un buon rapporto costo/peso (patate, carote, cavoli, cavolfiori, mele, arance).
- Zero Sprechi: Pianifica usando gli avanzi per il giorno dopo (es. brodo e avanzi di pollo del lunedì).

2. Giorno Senza Fornello (Sabato)

- Colazione: Elementi secchi (pane, marmellata, burro di arachidi).

- Pranzo e Cena: Piatti che si possono mangiare freddi o preparare in anticipo (es. uova sode, legumi in scatola conditi, insalate miste).

3. Focus sul Dimagrimento (Deficit Calorico)

Per perdere 1 kg a settimana, è necessario ridurre le porzioni e aumentare il volume delle verdure a basso contenuto calorico:

- Porzioni Ridotte: Le porzioni di pasta/riso sono state ridotte a 60g (pesato a crudo) e le porzioni proteiche a 100g (pesato a cotto).
- Fibre: Uso costante di cereali integrali e legumi per aumentare la sazietà.
- Condimenti: Utilizzare l'Olio EVO solo a crudo e in quantità misurata (max 2 cucchiaini al giorno).
- Spuntini: In questo regime molto ristretto, gli spuntini dovrebbero essere limitati a 1-2 al giorno e composti solo da frutta intera o verdure crude (es. carote) per tenere il metabolismo attivo senza aggiungere molte calorie.

Ti serviranno delle idee su dove trovare le migliori offerte per i tuoi acquisti settimanali?

Il sistema, di fronte una richiesta che presenta tratti poco realistici, si comporta comunque in maniera adatta e precisa, fornendo una dieta che risulta essere di buon livello e che rispetta l'ulteriore diminuzione di budget. Per quel che riguarda la perdita di peso, il sistema, come giusto che sia, aggiunge una nota medica, specificando che è importante integrare la dieta con attività fisica e con l'aiuto di uno specialista.

Per quel che riguarda la casualità sul giorno senza fornello, Gemini sceglie il Sabato.

3.4.4 Risposta alla richiesta finale

Vediamo a questo punto, come Gemini si comporta di fronte all'ultima richiesta, che risulta essere surreale.

Utente

La dieta va bene, ma la situazione è cambiata nuovamente:

- La persona vuole perdere 1kg a settimana, forse 0.5kg, o forse 0.3kg, è indecisa e varia in base a se farà attività fisica o meno.
- Non conosce i turni di lavoro e non sa quando sarà a casa e quando avrà tempo di cucinare.
- Non vuole ripetere più di due volte a settimana lo stesso ingrediente.
- Non ha accesso ad un grande supermercato nelle vicinanze, solamente un piccolo negozio di alimentari.
- Potrebbe saltare uno dei pasti di tanto in tanto, ma non è dato sapere quando e quale.

Gemini

Questa è la situazione più complessa finora! Gestire l'incertezza su budget (35€), l'obiettivo di dimagrimento variabile, l'imprevedibilità degli orari (turni e pasti saltati) e l'accesso limitato ai prodotti (piccolo negozio) richiede un approccio estremamente flessibile, basato sulla preparazione anticipata e sulla varietà.

Non potendo sapere quando mangerà o cucinerà, struttureremo la dieta in moduli settimanali, garantendo che i componenti siano vari e pronti per il consumo o la preparazione rapida.

Dieta Modulare per Turni Variabili (Budget 35€)

Obiettivo Dimagrimento: Manteniamo il regime ipocalorico (porzioni ridotte) per favorire la perdita di peso minima (0.3 kg) e massima (1 kg) in base all'attività fisica svolta.

Constraint:

- Budget: 35€
- L.F. e No Fish: Rispettato.
- Varietà: Nessun ingrediente ripetuto più di due volte a settimana.
- Logistica: Preparazione anticipata (Meal Prep) per gestire i turni e il piccolo negozio.
- Pasti Saltati: Le porzioni ridotte compensano i pasti saltati, mantenendo il deficit calorico.

La Strategia del "Kit Settimanale"

Dato il piccolo negozio, concentriamo la spesa su pochi ingredienti base e versatili, da preparare in grandi quantità una sola volta (es. Domenica sera) e conservare.

Preparazione (Domenica o giorno libero)	Ingrediente base	Uso (esempi)	Max Utilizzi Settimanali
Cuoci 1kg di Legumi	Ceci, Fagioli, Lenticchie	Zuppe, Fredde, vegetali Insalate, Burger	2 (Ceci) + 2 (Lenticchie/Fagioli)
Cuoci 6-8 Uova Sode	Uova (Proteina Jolly)	Colazione rapida, Pranzo freddo, Cena saziante	2 (Uova sode) + 2 (Uova strapazzate)
Cuoci 500g di Cereali	Riso Integrale, Orzo, Farro	Base per insalate fredde, Contorno riscaldabile	2 (Riso) + 2 (Orzo/Farro)
Lessa Verdure Dure	Carote, Patate, Cavolfiore	Contorno caldo/freddo, giunta alle insalate	2 (Patate) + 2 (Carote)

Schema Settimanale Flessibile (Moduli)

Questo schema si basa sulla combinazione di moduli, non su giorni fissi. Scegli un modulo (Colazione, Pranzo, Cena) in base al tuo tempo e dove ti trovi.

Modulo pasto	Opzione A (veloce/Fuori casa)	Opzione B (2a Più Rapida/Avanzi)	Opzione C (Cottura Minima/Base)	Ingrediente Base Non Ripetuto
Colazione	Porridge Semplice: 30g avena + acqua calda/bevanda vegetale + Cannella.	Frutta e Pane: 1 Frutto (Banana o Mela) + 30g pane integrale.	Uovo e Tè: 1 Uovo sodo (preparato in anticipo) + Tazza di tè/caffè.	Avena (2), Uova (1), Pane (2)
Pranzo	Insalata di Legumi: 70g Ceci (già cotti) + pomodorini + cipolla cruda + olio.	Cereale Freddo: 60g Riso/Orzo (già cotti) con verdure lesse avanzate (carote, cavolfiore).	Uova Sode e Carote: 2 Uova sode + Carote a bastoncini (snack pronto).	Ceci (1), Riso/Orzo (1), Uova (1)
Cena	Pasta al Pomodoro: 60g Pasta + Passata di pomodoro (2 cucchiaini) e spezie + Insalata verde.	Carne e Patate: 100g Tacchino o Pollo (cubetti) alla piastra/padella + 120g Patate lesse riscaldate.	Zuppa Veloce: Lenticchie in scatola/già cotte + Brodo vegetale caldo + 30g Pane tostato.	Pasta (2), Pollo/Tacchino (2), Lenticchie/Fagioli (2)

Esempio di Utilizzo per 7 Giorni (Verifica Varietà)

Giorno	Colazione	Pranzo	Cena	Proteina Usata	Legume/Carboidrato
Lun	Avena (1)	Riso (1) Freddo	Pollo (1) e Patate (1)	Pollo	Riso, Patate
Mar	Uovo (1) e Tè	Ceci (1) Insalata	Pasta (1) e Spinaci	Uova	Ceci, Pasta
Mer	Avena (2)	Uova (2) Sode	Lenticchie Zuppa	(1) Uova	Lenticchie
Gio	Pane (1) e Marm.	Orzo (1) Freddo	Tacchino (1) e Cavolo	Tacchino	Orzo, Cavolo
Ven	Frutta	Fagioli (1) Insalata	Pasta (2) Aglio e Olio	Fagioli	Pasta
Sab	Pane (2) e Marm.	Ceci (2) al Vapore	Pollo (2) e Carote (1)	Pollo	Carote
Dom	Frutta	Lenticchie Burger	(2) Carne Mac. (1) e Patate (2)	Carne Macinata	Patate

Guida alla Spesa nel Piccolo Negozio (Max 35€)

Puntare su prodotti non freschi e di lunga conservazione.

Categoria	Prodotto da cercare	Costo stimato	Note per il Piccolo Negozio
Proteine economiche	Uova (x12)	3,00 €	Sempre disponibili.
	Pollo/Tacchino (fusi, fesa)	8,00 €	Acquista un solo taglio in offerta.
	Carne Macinata (1 conf. 250g)	3,00 €	Proteina versatile per la domenica.
Legumi (Secchi o in Lattina Base)	Ceci, Lenticchie	3,00 €	I legumi secchi sono sempre i più economici.
Cereali Base	Riso/Pasta Integrale, Avena, Pane	5,00 €	Acquistare i pacchi più economici.
Verdura e Frutta	Patate (1kg), Carote (500g), Cipolle, Cavo- lo	6,00 €	Concentrati sulle radici e i vegetali duri.
	Frutta (Mele, Arance, Banane)	4,00 €	Scegli quella meno costosa.
Dispensa (Rifornimento)	Olio EVO (piccola bottiglia), Sale, Passata, Spezie	3,00 €	Questi durano più di una settimana.
Totale Stimato		33,00 €	Lascia un margine di 2€ per l'imprevisto.

Hai bisogno di consigli su come conservare al meglio i pasti preparati in anticipo per i tuoi turni?

Di fronte ad una situazione surreale, piena di vincoli difficilmente riscontrabili nella vita reale, il sistema comunque presenta una risposta chiara e di senso logico. Stavolta, piuttosto che una dieta a tutti gli effetti, il modello fornisce uno schema con più opzioni, come si può evincere dalla seconda tabella del prompt.

Oltre ad uno schema flessibile, il modello fornisce un esempio applicativo, consultabile nella terza tabella del prompt, ed una guida per la spesa settimanale, con un margine di 2€ e con delle note per ogni acquisto, come mostrato nella quarta tabella del prompt.

3.4.5 Valutazioni sul comportamento di Gemini

Il sistema di IA Generativa di Google ha lavorato in maniera ottimale, rispondendo in modo preciso e soddisfacente a tutte le richieste e rispettando ogni vincolo imposto.

Il modello non solo ha generato le risposte correttamente, ma ha fornito quasi sempre degli spunti non richiesti, manifestando un buon livello di spirito di iniziativa e di conoscenza. Ad esempio, quando è stato imposto il vincolo riguardante il peso, il sistema ha riconosciuto i propri limiti, esprimendo all'utente la necessità di parlare con un vero nutrizionista e il bisogno di integrare degli allenamenti specifici.

In conclusione, quindi, il sistema ha risposto bene alle nostre richieste, presentando degli ottimi livelli di adattabilità e conoscenza.

3.5 Analisi con DeepSeek

All'interno di questa sezione, analizzeremo il comportamento di DeepSeek, sottoponendo ad esso il nostro caso di studio.

3.5.1 Risposta al prompt iniziale

Vediamo come il sistema risponde al primo prompt, il più generale ed esente da vincoli.

Utente

Ciao, prepara una dieta settimanale bilanciata per una persona adulta, senza esigenze particolari. Colazione, pranzo e cena per la settimana.

DeepSeek

Certamente! Ecco un esempio di dieta settimanale bilanciata per un adulto sano, senza intolleranze o esigenze particolari (come dimagrimento o sport agonistico). Questo piano segue i principi della Dieta Mediterranea e ha l'obiettivo di fornire tutti i nutrienti necessari.

Linee Guida Generali:

- Colazione: Dolce e nutriente, per fornire energia per la mattina.
- Spuntini (metà mattina e metà pomeriggio): Consigliati uno yogurt, un frutto, una manciata di frutta secca o un piccolo panino integrale. Non sono inseriti nella tabella ma sono caldamente consigliati.
- Varietà: La chiave è alternare le fonti proteiche (carne, pesce, legumi, uova, formaggi) e i colori di frutta e verdura.
- Condimenti: Preferire sempre olio extravergine d'oliva a crudo, erbe aromatiche e spezie. Limitare il sale.
- Idratazione: Bere almeno 1.5-2 litri di acqua al giorno.

Dieta Settimanale Bilanciata

Giorno	Colazione	Pranzo	Cena
Lunedì	- 1 tazza di tè o caffè - 1 vasetto di yogurt greco bianco - 30g di muesli o fiocchi d'avena - 1 frutto fresco	Pasta al pomodoro e basilico (80g di pasta integrale, sugo di pomodoro fresco) + Insalata mista (lattuga, carote, pomodori) condita con olio EVO e aceto.	Petto di pollo alla griglia (150g) + Verdure al vapore (broccoli e cavolfiore) + 1 panino integrale (50g).
Martedì	- 1 tazza di latte parzialmente scremato - 3 fette biscottate integrali con un velo di marmellata - 1 spremuta d'arancia	Lenticchie in umido (200g di lenticchie cotte con sedano, carota e cipolla) + Contorno di spinaci saltati in padella con aglio + 1 panino integrale (50g).	Salmone al forno (150g) con erbe aromatiche + Insalata di finocchi e arancia a fette, condita con olio EVO.
Mercoledì	- 1 vasetto di yogurt bianco - 30g di cereali integrali - 1 manciata di frutta secca (noci, mandorle) - 1 frutto	Risotto agli asparagi (80g di riso) + Pinzimonio di verdure crude (sedano, carote, peperoni).	Uova sode o in frittata (2 uova) con zucchine + Insalata verde (rucola, songino) con pomodorini.
Giovedì	- 1 tazza di tè o caffè - 1 fetta di torta casalinga (es. torta di mele o yogurt) - 1 frutto	Orzo perlato con verdure (80g di orzo, zucchine, peperoni, melanzane) + 1 scaglia di Parmigiano Reggiano.	Filetto di merluzzo al vapore (200g) con limone + Contorno di bietole saltate + 1 panino integrale (50g).
Venerdì	- 1 tazza di latte parzialmente scremato - 3 fette biscottate integrali con un velo di miele - 1 frutto	Minestrone di verdure e legumi (con pasta o riso, 70g) + 1 scaglia di Grana Padano.	Bresaola (80g) con scaglie di Grana Padano e rucola, condita con limone e olio EVO + Contorno di carote grattugiate.
Sabato	- 1 cappuccino - 1 croissant semplice (o 2 fette di pane tostato con marmellata)	Pizza Margherita (una pizza media, circa 300g) o un piatto di lasagne al forno + Insalata mista di contorno.	Mozzarella di bufala (125g) + Insalata Caprese (pomodori, basilico fresco) + 1 panino integrale (50g).
Domenica	- 1 tazza di tè o caffè - 1 vasetto di yogurt bianco con frutti rossi e semi di chia - 1 fetta di pane tostato	Pasta alla carbonara (80g di pasta, 1 uovo, 30g di guanciale, Pecorino) (piatto più ricco, tipico del weekend) + Contorno di cicoria saltata.	Scaloppina di tacchino (150g) al limone + Purè di patate (preparato con poco burro e latte magro).

Note Importanti e Consigli Pratici

1. Porzioni: Le porzioni indicate (es. 80g di pasta, 150g di carne) sono indicative e dovrebbero essere adattate al proprio fabbisogno energetico (sesso, età, attività fisica).
2. Variazione: Questa dieta è un esempio. Sentiti libero di sostituire un tipo di verdura con un'altra di stagione, o un pesce con un altro (es. trota invece di salmone).
3. Metodi di cottura: Preferisci sempre cotture leggere: al vapore, alla griglia, al forno, in padella antiaderente o al cartoccio.
4. Frutta: Consumane 2-3 porzioni al giorno. È ottima come spuntino.
5. Attività Fisica: Ricorda che una dieta bilanciata è più efficace se abbinata a uno stile di vita attivo, con almeno 150 minuti di attività moderata a settimana.

Disclaimer: Questa proposta è pensata per un adulto sano senza patologie. Se hai condizioni mediche specifiche, allergie, o necessiti di un piano personalizzato, è fondamentale consultare un medico o un biologo nutrizionista.

La risposta del sistema risulta buona: esso fornisce una dieta variegata e che corrisponde alla nostra richiesta generale. Inoltre, il sistema fornisce note e consigli di propria iniziativa, evidenziando delle ottime conoscenze nel contesto.

3.5.2 Risposta ai vincoli di budget e alle intolleranze

Aggiorniamo, ora, la richiesta, aggiungendo vincoli di budget e riguardanti le intolleranze; analizziamo come si comporta DeepSeek di fronte a queste limitazioni.

Utente

Ora la persona che deve mettersi a dieta ha un budget massimo di 50 per tutta la settimana. Mantieni la dieta bilanciata e varia rispettando il budget.

DeepSeek

Certamente! Ecco una dieta settimanale bilanciata, varia e nutriente progettata per rimanere entro un budget di 50€.

Premesse Importanti:

- **Acquisti Intelligenti:** La chiave è comprare ingredienti base, versatili e di stagione.
- **Porzioni Controllate:** Tutte le porzioni sono calibrate per il budget.
- **Cucina "Zero Sprechi":** Useremo gli avanzi per creare nuovi piatti.
- **Lista della Spesa Approfondita:** Trovi in fondo una stima dettagliata dei costi.

Dieta Settimanale Economica (Budget: 50€)

Giorno	Colazione	Pranzo	Cena
Lunedì	Latte (200ml) + Fette biscottate (3) con marmellata	Pasta con Tonno e Pomodoro (80g pasta, 1/2 scatoletta tonno, pomodoro a pezzi)	Zuppa di Lenticchie e Verdure (lenticchie, patate, carote, sedano, cipolla)
Martedì	Latte (200ml) + Fette biscottate (3) con marmellata	Zuppa di Lenticchie avanzata + 1 panino	Uova Sode (2) con Insalata di Fagiolini (fagiolini lessati, conditi con olio e aceto)
Mercoledì	Tè + Pane tostato (1 fetta) con marmellata	Pasta con Fagiolini avanzati e pangrattato (80g pasta, fagiolini tritati, aglio, pangrattato tostato)	Sarde o Alici al forno (150g) con Patate al forno (2 patate medie)
Giovedì	Tè + Pane tostato (1 fetta) con marmellata	Panino con Marmellata di Alici (pane, alici avanzate schiacciate, pomodoro)	Pasta e Ceci (80g pasta, ceci, sedano, carota, cipolla, pomodoro)
Venerdì	Latte (200ml) + Pane (1 fetta) con marmellata	Pasta e Ceci avanzata	Polpette di Tonno e Patate (1/2 scatoletta tonno, 1 patata lessa, pangrattato, prezzemolo) + Insalata di lattuga
Sabato	Tè + Pane (1 fetta) con marmellata	Minestrone con Riso (patata, carota, zucchina, cavolo, 70g riso)	Frittata con le Zucchine (2 uova, 1 zucchina) + Pane (1 fetta)
Domenica	Latte (200ml) + Pane (1 fetta) con marmellata	Risotto con Zucchine avanzate (70g riso, zucchine trifolate) + 1 uovo in camicia (opzionale)	Polpettoni di Legumi avanzati (polpette di tonno/pasta e ceci rimaste) + Insalata mista

Spiegazione della Strategia Economica

1. *Proteine Economiche*: Abbiamo sostituito carni costose (pollo, manzo) con proteine molto più economiche ma ugualmente valide: uova, legumi (lenticchie, ceci, fagiolini), tonno in scatola e pesce azzurro (sarde, alici).
2. *Zero Sprechi*:
 - Le lenticchie avanzate del Lunedì si mangiano a pranzo il Martedì.
 - I fagiolini avanzati della cena di Martedì diventano un condimento per la pasta di Mercoledì.
 - Le sarde/alici avanzate diventano una pasta per il pranzo di Giovedì.
 - Le patate avanzate dal Mercoledì possono essere usate per le polpette di Venerdì.
3. *Verdure di Stagione e Convenienti*: Patate, carote, cipolle, sedano, cavolo cappuccio, zucchine e lattuga sono tra le verdure più economiche tutto l'anno.
4. *Colazione Semplice e Ripetitiva*: Mantenere la colazione uguale per gran parte della settimana permette di acquistare confezioni più grandi, risparmiando molto. La marmellata è economica e dura a lungo.
5. *Panificazione*: Comprare un unico pacchetto di pane o fette biscottate per la settimana.

Lista della Spesa Stimata (per 1 persona)

Ingredienti Base (da acquistare una tantum o che durano più settimane):

- Olio EVO, Aceto, Sale, Aglio, Pepe -> 3,00€(considerando l'uso parziale)

Reparto Fresco:

- Latte (1 litro): 1,50€
- Uova (6 uova) : 2,00€
- Patate (1 kg): 1,20€
- Carote (500g): 0,80€
- Cipolle (500g): 0,80€
- Sedano (1 costa): 0,50€
- Zucchine (3 medie): 1,50€
- Cavolo Cappuccio (1/2): 1,00€
- Lattuga (1 cespo): 1,00€
- Fagiolini (300g): 1,50€
- Prezzemolo (1 mazzo): 0,70€
- Totale Fresco: ~ 12,50€

Reparto Secco e Conserve:

- Pasta (500g): 0,90€
- Riso (500g): 1,20€
- Lenticchie secche (500g): 1,50€
- Ceci secchi (500g): 1,50€
- Tonno in scatola al naturale (1 scatoletta da 80g): 1,80€
- Sarde o Alici fresche o congelate (400g): 4,00€
- Fette Biscottate Integrali (confezione): 1,50€
- Pane (1 pagnotta da 500g): 2,00€
- Marmellata (vasetto): 1,80€
- Tè (filtri): 1,50€
- Totale Secco e Conserve: ~ 18,20€

Stima Totale della Spesa: 3,00€(base) + 12,50€(fresco) + 18,20€(secco) = 33,70€

Conclusione

Con una spesa di circa 34-38€(lasciando un margine per i prezzi variabili), questa dieta rispetta il budget di 50€e fornisce tutti i nutrienti essenziali. Il segreto è la pianificazione, la cucina creativa con gli avanzi e la scelta di ingredienti nutrienti ma non trasformati, che costano sempre meno dei cibi pronti.

Il sistema ha risposto correttamente, adattandosi al limite sul budget. A differenza di quanto visto con Gemini, si può osservare un approccio alla dieta alternativo, orientato più agli avanzzi e al risparmio sulla spesa (in particolare, si evince dai 16€ avanzati per la spesa).

Vediamo, ora, cosa cambierà con l'aggiunta di intolleranze.

Utente

La persona è anche intollerante al lattosio, e preferisce evitare il pesce se possibile. Ricrea la dieta anche in base a queste informazioni, mantenendo il budget dei 50 euro.

DeepSeek

Certamente! Ecco una dieta settimanale bilanciata, senza lattosio e senza pesce, progettata per rimanere rigorosamente entro il budget di 50€.

Strategie Chiave per questo Nuovo Piano:

- **Proteine:** Ci affideremo a legumi, uova, pollo (la carne più economica) e prosciutto cotto senza lattosio.
- **Latticini Sostitutivi:** Useremo latte di soia o avena, spesso più economici dello yogurt senza lattosio.
- **Colazione:** Diventa ancora più semplice ed economica.
- **Zero Sprechi:** Ancora più cruciale per non sprecare le proteine animali acquistate.

Dieta Settimanale senza Lattosio e Pesce (Budget: 50€)

Giorno	Colazione	Pranzo	Cena
Lunedì	Tè + Pane tostato (1 fetta) con marmellata	Pasta e Fagioli (80g pasta, fagioli borlotti, sedano, carota, pomodoro)	Zuppa di Lenticchie e Verdure (lenticchie, patate, carote, sedano)
Martedì	Bevanda di soia (200ml) + Pane (1 fetta) con marmellata	Zuppa di Lenticchie avanzata + 1 panino	Frittata con le Zucchine (2 uova, 1 zucchina) + Insalata di lattuga
Mercoledì	Tè + Fetta di torta casalinga (senza latte/burro, es. torta di mele)	Pasta con Zucchine trifolate (80g pasta, zucchini avanzate, aglio)	Pollo in padella (150g di coscia o sovracoscia, più economica del petto) con Peperoni saltati
Giovedì	Bevanda di soia (200ml) + Pane (1 fetta) con marmellata	Panino con Pollo avanzato e lattuga	Pasta e Ceci (80g pasta, ceci, sedano, carota, rosmarino)
Venerdì	Tè + Pane (1 fetta) con marmellata	Pasta e Ceci avanzata	Uova Sode (2) con Insalata di Patate (patate lessate a cubetti, prezzemolo)
Sabato	Bevanda di soia (200ml) + Pane (1 fetta) con marmellata	Risotto con Piselli (70g riso, piselli, cipolla)	Polpette di Pollo e Zucchine (tritiamo il pollo avanzato con pangrattato e zucchini) + Carote al vapore
Domenica	Tè + Fetta di torta casalinga avanzata	Minestrone con Riso (patata, carota, zucchina, cavolo, 70g riso) + 1 uovo sodo	Polpette avanzate + Verdure al forno (peperoni, patate, carote)

Spiegazione delle Modifiche e Strategia Economica

1. *Niente Pesce*: Abbiamo completamente rimosso tonno e pesce azzurro. Le proteine animali sono fornite principalmente da uova e pollo (tagli economici).
2. *Niente Lattosio*:
 - Latte/Yogurt -> Sostituiti con Bevanda di Soia/Avena (più economica dello yogurt senza lattosio).
 - Formaggi -> Completamente eliminati per rispettare il budget. I nutrienti vengono da altre fonti.
3. *Proteine Vegetali in Primo Piano*: La dieta si basa fortemente su fagioli, lenticchie, ceci e piselli, che sono economici, nutrienti e privi di lattosio.
4. *Riuso Creativo*:
 - Il pollo avanzato di Mercoledì viene usato per un panino e per delle polpette.
 - Le zucchine avanzate dalla frittata diventano un sugo per la pasta.
 - Le verdure vengono usate in più piatti (minestrone, contorni, zuppe).

Lista della Spesa Stimata (per 1 persona)

Ingredienti Base (da acquistare una tantum):

- Olio EVO, Aceto, Sale, Aglio, Pepe, Rosmarino -> 3,00€(considerando l'uso parziale)

Reparto Fresco:

- Bevanda di Soia o Avena (1 litro): 1,80€
- Uova (6 uova): 2,00€
- Pollo (cosce/sovracoscia, 400g): 4,50€
- Patate (1 kg): 1,20€
- Carote (1 kg): 1,00€
- Cipolle (500g): 0,80€
- Sedano (1 costa): 0,50€
- Zucchine (4 medie): 2,00€
- Peperoni (2): 1,50€
- Cavolo Cappuccio (1/2): 1,00€
- Lattuga (1 cespo): 1,00€
- Prezzemolo (1 mazzo): 0,70€
- Totale Fresco: ~ 18,00€

Reparto Secco e Conserve:

- Pasta (500g): 0,90€
- Riso (500g): 1,20€
- Fagioli borlotti secchi (500g): 1,50€
- Lenticchie secche (500g): 1,50€
- Ceci secchi (500g): 1,50€
- Piselli secchi (250g): 1,20€
- Pane (1 pagnotta da 500g): 2,00€
- Fette Biscottate (confezione, opzionale per variare): 1,50€
- Marmellata (vasetto): 1,80€
- Tè (filtri): 1,50€
- Farina 00 (per polpette/torta, 500g): 0,60€
- Totale Secco e Conserve: ~ 16,20€

Stima Totale della Spesa: $3,00\text{€}(\text{base}) + 18,00\text{€}(\text{fresco}) + 16,20\text{€}(\text{secco}) = 37,20\text{€}$

Conclusione

Questa nuova dieta, pur con i vincoli aggiuntivi, rimane ben all'interno del budget, con una spesa stimata di circa 37€. È:

- Bilanciata: Apporta proteine (vegetali e animali), carboidrati complessi, grassi buoni e fibre.
- Sicura: Completamente priva di lattosio e pesce.
- Economica: Sfrutta al massimo gli ingredienti base, i legumi e il riuso degli avanzi.
- Semplice da Preparare: I piatti sono per lo più zuppe, pasta, risotti e piatti unici facili da cucinare.

Il sistema, anche in questo caso, ha risposto in maniera adeguata, adattandosi alle esigenze. La strategia utilizzata ha seguito quella scelta precedentemente, ossia quella di utilizzare gli avanzi per risparmiare nella spesa, lasciando un margine di circa 13€. Le modifiche apportate sono state spiegate ed esposte dal modello, rendendo la risposta più chiara e precisa.

3.5.3 Risposta ai vincoli sul peso e sulla casualità

In questa sezione vedremo come DeepSeek risponderà al nostro quarto prompt, incentrato sull'aggiunta di vincoli riguardanti la perdita di peso e la casualità.

Utente

Ci sono altri aspetti che non erano stati specificati inizialmente:

- Il budget settimanale, a causa di alcuni problemi, scende a 35€.
- Un giorno a caso della settimana non può cucinare nulla.
- Il peso da perdere è di circa 1kg a settimana.

DeepSeek

Certamente! Queste nuove specifiche cambiano radicalmente l'approccio. Ecco una dieta settimanale che:

1. Rispetta un budget di 35€
2. Include un giorno senza cucina
3. È ipocalorica per favorire una perdita di 1kg a settimana (circa 1200-1300 kcal/giorno)

Premesse Fondamentali:

- Perdita di 1kg/settimana: Questo obiettivo è molto aggressivo e non sostenibile a lungo termine. È fondamentale ascoltare il proprio corpo e consultare un medico prima di iniziare.
- Strategia: La dieta si basa su piatti unici per ridurre le porzioni e i costi, sfruttando al massimo ingredienti base e sazianti.

Dieta Settimanale Ipocalorica da 35€(Senza Lattosio/Pesce)

Giorno senza cottura: Mercoledì

Giorno	Pasti	Note
Lunedì	C: Tè + 1 fetta di pane integrale con marmellata light P: Zuppa di lenticchie (60g lenticchie, 1 patata, carota, sedano) C: Uova sode (2) + Insalata di cavolo cappuccio (cavolo, carota, aceto)	Preparare porzione extra di zuppa per Martedì. Le uova sode si conservano.
Martedì	C: Tè + 1 fetta di pane con marmellata light P: Zuppa di lenticchie avanzata C: Frittata di ceci (30g farina di ceci, acqua, 1 zucchina trifolata) + Spinaci saltati	La farina di ceci è proteica e economica. Preparare frittata per il giorno dopo.
Mercoledì (NO COTTURA)	C: Tè + 1 fetta di pane con marmellata light P: Panino veloce (2 fette di pane, frittata di ceci avanzata, lattuga) C: Insalata di ceci (ceci cotti, pomodoro, sedano, cipolla - condita con olio e aceto)	Usare ceci cotti dal giorno prima o in scatola. Tutto pronto in 5 minuti.
Giovedì	C: Tè + 1 fetta di pane con marmellata light P: Pasta e ceci (60g pasta, 30g ceci, rosmarino) C: Zucchine e patate al forno (1 zucchina, 1 patata grande) + 1 uovo sodo	Piatto unico semplice. Le verdure al forno sono sazianti.
Venerdì	C: Tè + 1 fetta di pane con marmellata light P: Minestrone denso (patata, cavolo, carota, cipolla, 30g riso) C: Polpette di legumi (avanzi di ceci/lenticchie schiacciate con pangrattato e prezzemolo, passate in padella)	Il minestrone è un piatto unico ipocalorico perfetto.
Sabato	C: Tè + 1 fetta di pane con marmellata light P: Risotto con zucchine (60g riso, 1 zucchina, mezza cipolla) C: Insalata di pollo e fagioli (60g di petto di pollo grigliato, 40g fagioli, insalata)	Il pollo è la proteina animale più economica. Usare solo 1/4 del petto.
Domenica	C: Tè + 1 fetta di pane con marmellata light P: Pasta con broccoli (60g pasta, 200g broccoli, aglio, peperoncino) C: Vellutata di verdure (avanzi di tutte le verdure della settimana frullate)	Giorno di "pulizia" del frigorifero. La vellutata è saziante e povera di calorie.

Lista della Spesa Ottimizzata (Budget 35€)

Calcolo Approfondito e Realistico:

- Reparto Secco e Dispensa:
 - Lenticchie secche (500g) - 1,50€
 - Ceci secchi (500g) - 1,50€

- Fagioli secchi (500g) - 1,50€
- Farina di ceci (250g) - 1,20€
- Pasta (500g) - 0,90€
- Riso (500g) - 1,00€
- Pane Integrale (o pagnotta) - 2,00€
- Pangrattato (200g) - 0,80€
- Tè (filtri) - 1,20€
- Marmellata light - 1,50€
- Totale Secco: 13,10€

- Reparto Fresco:

- Uova (6 uova) - 2,00€
- Petto di pollo (1/4, circa 150g) - 1,50€
- Patate (2 kg) - 2,00€
- Carote (1 kg) - 1,00€
- Cipolle (1 kg) - 1,00€
- Zucchine (3 medie) - 1,50€
- Cavolo Cappuccio (1/2) - 1,00€
- Broccoli (1) - 1,20€
- Spinaci (1 mazzo) - 1,20€
- Lattuga (1 cespo) - 1,00€
- Aglio - 0,50€
- Prezzemolo - 0,50€
- Totale Fresco: 15,40€

- Condimenti (che durano mesi):

- Olio EVO, Aceto, Sale, Pepe, Peperoncino -> Stima 3,00€(per la quota settimanale)

TOTALE STIMATO: 13,10€(secco) + 15,40€(fresco) + 3,00€(condimenti) = 31,50€

Il budget di 35€ è quindi rispettato, con un margine di circa 3,50€ per imprevisti o per comprare una bustina di limoni.

Raccomandazioni Finali e Avvertenze

1. *Idratazione:* Bere almeno 2 litri di acqua al giorno. Aiuta il senso di sazietà e il metabolismo.
2. *Attività Fisica:* Per perdere 1kg a settimana in modo sano, è essenziale abbinare una dieta ipocalorica a dell'attività fisica, anche moderata (es. camminate veloci di 30-45 minuti al giorno).
3. *Sapore:* Usa spezie ed erbe aromatiche (peperoncino, rosmarino, aglio) per insaporire i piatti senza aggiungere calorie.
4. *Ascolta il tuo corpo:* Se ti senti debole, stordito o eccessivamente affamato, considera di aumentare leggermente le porzioni (es. aggiungendo un'altra porzione di verdura). La perdita di peso dovrebbe essere sostenibile.
5. *Consulta un Medico:* Prima di intraprendere un deficit calorico così marcato, è altamente consigliabile parlarne con un medico o un nutrizionista.

Questa dieta è un esempio di come si possa mangiare in modo sano e bilanciato con budget molto ristretti e vincoli specifici, ma ricorda che la salute viene sempre al primo posto.

Il modello ha risposto in maniera adeguata anche stavolta, nonostante le difficoltà imposte e la poca realistica riguardante il peso da perdere settimanalmente. Come si nota dalla Tabella del prompt, inoltre, sono state inserite delle note per rendere i passaggi più chiari. Anche il budget è rispettato, con la spesa che rientra nel prezzo imposto con un margine di

3,50€.

3.5.4 Risposta finale

Vediamo in questa sottosezione la risposta di DeepSeek al nostro quinto ed ultimo prompt, relativo ad una situazione piena di variabili.

Utente

La dieta va bene, ma la situazione è cambiata nuovamente:

- La persona vuole perdere 1kg a settimana, forse 0.5kg, o forse 0.3kg, è indecisa e varia in base a se farà attività fisica o meno.
- Non conosce i turni di lavoro e non sa quando sarà a casa e quando avrà tempo di cucinare.
- Non vuole ripetere più di due volte a settimana lo stesso ingrediente.
- Non ha accesso ad un grande supermercato nelle vicinanze, solamente un piccolo negozio di alimentari.
- Potrebbe saltare uno dei pasti di tanto in tanto, ma non è dato sapere quando e quale.

DeepSeek

Basandosi su tutte le nuove e complesse variabili, una dieta rigida e giornaliera non è più la soluzione adatta. Ecco un piano strategico flessibile che funziona per questa situazione.

Piano Alimentare Flessibile e "Anti-Crisi"

L'idea è passare da un menu fisso a un set di regole e opzioni che la persona può combinare in base alla giornata.

1. Strategia dei "Building Blocks" (Mattoncini)

Prepara, quando hai tempo, questi componenti base. Poi li assembli al momento.

- Proteine cotte: Lessa 2-3 uova. Cuoci a vapore o in padella 200g di petto di pollo/tacchino. Metti in ammollo e cuoci un mix di legumi (ceci, lenticchie, fagioli borlotti).
- Verdure pronte: Lava e taglia verdure crude robuste: carote, sedano, peperoni, cetrioli. Cuoci a vapore un mix di broccoli, cavolfiore e fagiolini.
- Carboidrati: Cuoci una porzione di quinoa, farro o orzo (più nutrienti della pasta e si conservano bene). Tieni pane integrale surgelato (non deperisce).
- Condimenti: Prepara un barattolo di dressing semplice (olio EVO, aceto di mele, senape, sale, pepe).

2. Menu "Assemblabile" con Regole Chiare

Colazione (Scegli 1 opzione quando sei a casa)

1. Tè/Caffè + 1 fetta di pane tostato con burro di arachidi + 1 frutto.
2. Frullato di spinaci (manciata), 1/2 banana congelata e bevanda vegetale.
3. Porridge di avena (fiocchi d'avena cotti con acqua o bevanda vegetale) con cannella.

Pranzo/Cena (Assembla un piatto unico con questa formula)

1 Mancio di Verdura + 1 Pugno di Proteine + 1 Pugno di Carboidrati Complessi

Esempi di piatti unici assemblati in 5 minuti:

- Insalatona: Verdure crude + ceci + quinoa + dressing.
- Piatto Caldo (se hai tempo): Verdure cotte + cubetti di pollo + farro.
- Wrap Veloce: Una grossa foglia di lattuga come involucri, con hummus, quinoa e peperoni a listarelle.

3. Strategia per il Negozio Piccolo e il Budget (35€)

1. Primo giro d'acquisto (Fondamentali):

- Uova (6)
- Legumi secchi o in barattolo (2 tipi diversi, es. lenticchie e ceci)
- Focci d'avena
- Pane integrale (da congelare)
- Burro di arachidi 100
- Bevanda vegetale (es. avena)
- Verdure di stagione economiche e durature (cipolle, carote, cavolo, zucchine, patate)
- Mele e banane (frutta più economica e resistente)
- Scatoletta di tonno (se tollerato, per emergenza)

2. Secondo giro (se avanza budget):

- Petto di pollo/tacchino (da cuocere e congelare in porzioni)
- Quinoa/farro
- Altra verdura (es. broccoli, spinaci)
- Frutta secca (noci, mandorle)

4. Gestione dell'Indecisione sulla Perdita di Peso

- Obiettivo 1kg/settimana: Segui rigorosamente la formula del piatto unico, controlla le porzioni (specialmente i carboidrati e l'olio), e non saltare i pasti. Aggiungi attività fisica.
- Obiettivo 0.5kg/settimana: Segui la formula, ma le porzioni possono essere leggermente più generose. Un pasto saltato o un "fuori pasto" non comprometteranno il risultato.
- Obiettivo 0.3kg/settimana: Questo è un deficit molto sostenibile. La formula del piatto unico ti porterà naturalmente a questo risultato, anche con qualche imprevisto.

5. Regola d'Oro: Nessun Ingrediente Più di 2 Volte

Tieni traccia mentale. Se hai mangiato ceci a pranzo, a cena scegli uova o pollo. Se hai mangiato quinoa, il giorno dopo passa a farro o avena. La varietà è garantita dall'alternanza dei "mattoncini".

Piano di Emergenza per i Giorni di Caos Totale

- Colazione saltata? Tieni delle barrette ai cereali (controlla gli zuccheri) o della frutta secca in borsa.
- Pranzo impossibile? Il panino con burro di arachidi e banana è saziante, bilanciato e veloce.
- Cena a orari assurdi? Una scatoletta di tonno (se hai) con carote crude e un frutto è un pasto dignitoso in 2 minuti.

In sintesi:

Questo non è un menu, ma un sistema. Ti dà il controllo per gestire l'imprevedibilità, rispettando il budget, le intolleranze e l'obiettivo di peso, senza lo stress di dover seguire un rigido schema giornaliero. La preparazione iniziale dei "mattoncini" è la chiave per sopravvivere alle settimane più frenetiche.

Di fronte ad una situazione estremamente complessa e ricca di vincoli e limitazioni, il sistema non ha prodotto una vera e propria dieta, bensì un piano strategico applicabile in base alle possibilità dell'utente. Esso descrive bene i passaggi da svolgere e fornisce informazioni per la strategia, rendendola chiara.

3.5.5 Valutazioni sul comportamento di DeepSeek

Nel complesso, il comportamento di DeepSeek si può considerare molto buono, con un buon livello di competenza e adattabilità. Il modello cinese, inoltre, ha sempre inserito dei consigli e delle note aggiuntive, elementi non richiesti, ma apprezzati.

Nell'elaborazione dell'ultima richiesta, invece, il sistema ha presentato lo stesso approccio adottato precedentemente da Gemini, con l'unica differenza che il modello americano aveva fornito esempi concreti sulla dieta e sulla spesa da effettuare, mostrando, quindi, un grado di conoscenza più elevato.

In conclusione, quindi, possiamo dire che DeepSeek si è comportato in maniera corretta e consona per situazioni verosimili e con un grado di difficoltà medio. In una situazione quasi surreale e di difficoltà elevata, invece, ha fornito una soluzione accettabile, ma senza fornire un esempio di applicazione della strategia.

3.6 Analisi con Mistral

All'interno di questa sezione descriveremo i risultati ottenuti sottoponendo al sistema di IA Generativa francese Mistral il nostro caso di studio, con l'obiettivo di analizzarne le capacità.

3.6.1 Risposta al prompt iniziale

Cominciamo l'analisi di Mistral analizzando l'andamento del sistema di fronte alla richiesta iniziale.

Utente

Ciao, prepara una dieta settimanale bilanciata per una persona adulta, senza esigenze particolari. Colazione, pranzo e cena per la settimana.

Mistral

Ecco una dieta settimanale bilanciata per un adulto senza esigenze particolari, con colazione, pranzo e cena. La dieta è varia, ricca di nutrienti e facile da preparare.

Lunedì

Colazione: Yogurt bianco naturale + muesli senza zucchero + 1 frutto di stagione

Pranzo: Pasta integrale al pomodoro fresco + insalata mista (lattuga, carote, cetrioli) + 1 fetta di pane integrale

Cena: Petto di pollo alla griglia + purè di patate + zucchine al vapore

Martedì

Colazione: Latte scremato o vegetale + 2 fette biscottate con marmellata senza zucchero + 1 manciata di noci

Pranzo: Risotto ai funghi + insalata di rucola e pomodorini + 1 porzione di frutta

Cena: Omelette con spinaci + pane integrale tostato + 1 porzione di frutta

Mercoledì

Colazione: Smoothie con banana, latte e avena + 1 cucchiaino di miele

Pranzo: Orzo con verdure (melanzane, peperoni, zucchine) + 1 porzione di formaggio fresco

Cena: Salmone al forno + quinoa + broccoli al vapore

Giovedì

Colazione: Tè verde + 2 fette di pane integrale con crema di mandorle + 1 frutto

Pranzo: Minestrone di verdure + 1 porzione di pane integrale + 1 porzione di frutta

Cena: Frittata con zucchine e cipolle + insalata verde

Venerdì

Colazione: Yogurt greco + frutti di bosco + semi di chia

Pranzo: Pasta al pesto + insalata di pomodori e basilico + 1 porzione di frutta

Cena: Filetto di merluzzo al limone + patate lesse + carote al vapore

Sabato

Colazione: Caffè o cappuccino + 1 croissant integrale + 1 spremuta d'arancia

Pranzo: Insalata di farro con verdure (peperoni, cetrioli, olive) + 1 porzione di formaggio fresco

Cena: Pollo al curry con riso basmati + insalata verde

Domenica

Colazione: Latte + cereali integrali + 1 frutto

Pranzo: Lasagne vegetariane (melanzane, zucchine, ricotta) + insalata mista

Cena: Zuppa di legumi (lenticchie o ceci) + pane integrale + 1 porzione di frutta

Note generali

- *Idratazione:* Bevi almeno 1,5-2 litri di acqua al giorno.

- *Spuntini (opzionali)*: Frutta fresca, yogurt, frutta secca o verdure crude.
- *Varianti*: Puoi sostituire gli alimenti in base ai tuoi gusti, mantenendo l'equilibrio tra carboidrati, proteine e grassi sani.

Vuoi che adatti la dieta a qualche preferenza o necessità specifica? Ad esempio, vegetariana, senza glutine, o con un budget limitato?

Il sistema ha fornito una soluzione al problema senza particolari aggiunte o consigli, limitandosi alla risoluzione della richiesta. Un problema che salta subito all'occhio, però, è l'assenza delle quantità per ogni alimento utilizzato nella dieta.

3.6.2 Risposta ai vincoli di budget e alle intolleranze

Vediamo ora come il sistema modifica la risposta se aggiungiamo limitazioni riguardanti il budget.

Utente

Ora la persona che deve mettersi a dieta ha un budget massimo di 50€ per tutta la settimana. Mantieni la dieta bilanciata e varia rispettando il budget.

Mistral

Per rispettare un budget di 50€ a settimana mantenendo una dieta bilanciata e varia, occorre privilegiare alimenti di stagione, legumi, cereali integrali, uova, verdure fresche e località, e ridurre al minimo carne e pesce costosi. Ecco una proposta economica, nutriente e gustosa:

Lunedì

Colazione: Latte scremato (o vegetale) + 2 fette biscottate con marmellata fatta in casa (o miele) + 1 banana

Pranzo: Pasta integrale con sugo di pomodoro fresco, aglio e basilico + insalata di lattuga e carote

Cena: Frittata con 2 uova, zucchine e cipolle + pane integrale

Martedì

Colazione: Yogurt bianco naturale + avena in fiocchi + 1 mela

Pranzo: Risotto con zucca e cipolla + 1 porzione di formaggio fresco (es. ricotta)

Cena: Lenticchie in umido con carote e sedano + pane integrale

Mercoledì

Colazione: Tè o caffè + 2 fette di pane integrale con crema di frutta secca (fatta in casa) + 1 arancia

Pranzo: Minestrone di verdure (patate, carote, cavolo, pomodori) + 1 porzione di pane integrale

Cena: Polenta con sugo di fagioli e pomodoro

Giovedì

Colazione: Latte + cereali integrali + 1 pera

Pranzo: Pasta e ceci + insalata verde

Cena: Omelette con spinaci (surgelati) + pane integrale

Venerdì

Colazione: Yogurt + muesli fatto in casa (fiocchi d'avena, semi di girasole, uvetta) + 1 kiwi

Pranzo: Riso integrale con verdure saltate (zucchine, peperoni, cipolle) + 1 uovo sodo

Cena: Zuppa di cavolo e patate + pane integrale

Sabato

Colazione: Caffè o cappuccino + 1 fetta di pane integrale con olio e pomodoro + 1 frutto

Pranzo: Pasta al pesto (fatto in casa con basilico, aglio, olio e pinoli o noci) + insalata di pomodori

Cena: Fagioli cannellini con pasta corta + verdure di stagione

Domenica

Colazione: Latte + 2 fette biscottate con marmellata + 1 frutto

Pranzo: Pizza fatta in casa (impasto semplice con farina, acqua, lievito, pomodoro, mozzarella e origano)

Cena: Zuppa di lenticchie e orzo + insalata verde

Consigli per risparmiare:

- *Acquista sfuso o in offerta:* Cereali, legumi, pasta, riso, verdure di stagione.
- *Prepara in casa:* Marmellate, pesto, pane, pizza, minestrone (evita cibi pronti).
- *Usa le proteine vegetali:* Legumi (lenticchie, ceci, fagioli) sono economici e nutrienti.
- *Sfrutta le verdure surgelate:* Spesso costano meno e mantengono i nutrienti.
- *Riduci gli sprechi:* Usa gli avanzi per minestre, frittate o insalate.

Lista della spesa stimata (50€):

- *Carboidrati:* Pasta integrale (1 kg), riso (1 kg), orzo (500 g), farina (1 kg), pane integrale (2 confezioni), avena (500 g)
- *Proteine:* Uova (6), formaggio fresco (250 g), mozzarella (1 palla), legumi secchi (lenticchie, ceci, fagioli – 1 kg totali)
- *Verdure:* Lattuga, carote, zucchine, cipolle, pomodori, zucca, spinaci surgelati, cavolo, patate
- *Frutta:* Banane, mele, arance, pere, kiwi
- *Altro:* Latte (1 litro), yogurt (4 vasetti), olio, sale, spezie, basilico, aglio

Vuoi che ti suggerisca anche qualche ricetta semplice per preparare i piatti in elenco? Oppure preferisci una versione ancora più economica o con altri alimenti di base?

Di fronte all'aggiunta di un budget di 50€, Mistral ha modificato la dieta precedente, cambiando alimenti e pasti per le giornate. Anche in questo caso, però, si nota che all'interno della dieta e in alcuni elementi della spesa non sono presenti le quantità da mangiare/acquistare. Inoltre, per quanto riguarda la spesa, il sistema non ha inserito la stima di costo di ogni

alimento. In questo modo, vista anche l'assenza delle quantità da acquistare, non è possibile analizzare effettivamente se la spesa rispetta il budget o lo sfiora.

Vediamo ora come cambierà la risposta aggiungendo delle intolleranze alla richiesta.

Utente

La persona è anche intollerante al lattosio, e preferisce evitare il pesce se possibile. Ricrea la dieta anche in base a queste informazioni, mantenendo il budget dei 50 euro.

Mistral

Ecco una dieta settimanale bilanciata, senza lattosio e senza pesce, rispettando il budget di 50€. Ho sostituito latte, formaggi e yogurt con alternative vegetali economiche (es. latte di soia, tofu) e ho escluso il pesce, privilegiando proteine vegetali e uova.

Lunedì

Colazione: Latte di soia + 2 fette biscottate con marmellata fatta in casa + 1 banana

Pranzo: Pasta integrale al pomodoro fresco, aglio e basilico + insalata di carote e cetrioli

Cena: Frittata con 2 uova, zucchine e cipolle + pane integrale

Martedì

Colazione: Tè o caffè + porridge di avena con semi di lino e miele + 1 mela

Pranzo: Risotto con zucca e cipolla + insalata verde

Cena: Lenticchie in umido con carote e sedano + pane integrale

Mercoledì

Colazione: Smoothie con latte di soia, banana e avena + 1 manciata di noci

Pranzo: Minestrone di verdure (patate, carote, cavolo, pomodori) + pane integrale

Cena: Polenta con sugo di fagioli e pomodoro

Giovedì

Colazione: Caffè o tè + 2 fette di pane integrale con crema di mandorle (fatta in casa) + 1 arancia

Pranzo: Pasta e ceci + insalata di pomodori e basilico

Cena: Tofu saltato con verdure (zucchine, peperoni, cipolle) + riso integrale

Venerdì

Colazione: Latte di soia + cereali integrali + 1 pera

Pranzo: Riso integrale con verdure saltate (zucchine, peperoni, cipolle) + 1 uovo sodo

Cena: Zuppa di cavolo e patate + pane integrale

Sabato

Colazione: Caffè o cappuccino con latte di soia + 1 fetta di pane integrale con olio e pomodoro + 1 frutto

Pranzo: Pasta al pesto (fatto in casa con basilico, aglio, olio e noci) + insalata verde

Cena: Fagioli cannellini con pasta corta + verdure di stagione

Domenica

Colazione: Tè o caffè + 2 fette biscottate con marmellata + 1 frutto

Pranzo: Pizza fatta in casa (impasto con farina, acqua, lievito, pomodoro, origano e verdure grigliate)

Cena: Zuppa di lenticchie e orzo + insalata verde

Lista della spesa stimata (50€):

Carboidrati:

- Pasta integrale (1 kg)
- Riso integrale (1 kg)
- Orzo (500 g)
- Farina (1 kg)
- Pane integrale (2 confezioni)
- Avena (500 g)
- Patate (1 kg)

Proteine vegetali:

- Lenticchie (500 g)
- Ceci (500 g)
- Fagioli cannellini (500 g)
- Tofu (2 confezioni)
- Uova (6)

Verdure:

- Carote, zucchine, cipolle, pomodori, zucca, cavolo, spinaci surgelati, cetrioli, peperoni, lattuga

Frutta:

- Banane, mele, arance, pere

Altro:

- Latte di soia (1 litro)
- Olio, sale, spezie, basilico, aglio, semi di lino, noci, marmellata fatta in casa

Consigli extra:

- *Prepara in casa:* Pesto, marmellata, pane, pizza, minestrone.
- *Acquista sfuso o in offerta:* Legumi, cereali, verdure di stagione.
- *Sfrutta gli avanzi:* Usa le verdure avanzate per minestre, frittate o insalate.

Vuoi che ti suggerisca anche le ricette per preparare i piatti in elenco? Oppure preferisci altre varianti?

Il sistema ha elaborato la richiesta, presentando una soluzione che elimina del tutto l'utilizzo di latte e pesce. Le mancanze della risposta, però, rimangono le stesse del caso precedente: non sono presenti le quantità per gli alimenti (ad esempio, per la pasta o il riso) da cuocere e non sono presenti i costi stimati di ogni singolo elemento nella spesa.

3.6.3 Risposta ai vincoli sul peso e sulla casualità

Utente

Ci sono altri aspetti che non erano stati specificati inizialmente:

- Il budget settimanale, a causa di alcuni problemi, scende a 35€.
- Un giorno a caso della settimana non può cucinare nulla.
- Il peso da perdere è di circa 1kg a settimana.

Mistral

Grazie per le precisazioni! Ecco una dieta settimanale ipocalorica (per perdere circa 1 kg a settimana), senza lattosio, senza pesce, con un giorno "no cooking", e un budget stretto di 35€.

Principi chiave:

- *Riduzione calorica*: Circa 1.500-1.600 kcal/giorno (deficit controllato).
- *Proteine magre*: Uova, legumi, tofu, per sazietà e mantenimento muscolare.
- *Carboidrati complessi*: Cereali integrali, patate, per energia a lungo termine.
- *Grassi sani*: Olio d'oliva, frutta secca (in quantità moderate).
- *Verdure*: A volontà, per volume e fibre.
- *Budget*: Privilegio alimenti di base, sfusi, e di stagione.
- *Giorno "no cooking"*: Pranzo e cena con cibi pronti al consumo (es. insalata di legumi in scatola, pane integrale, frutta, uova sode).

Lunedì

Colazione: Tè verde + 2 fette biscottate integrali con marmellata senza zucchero + 1 mela

Pranzo: Pasta integrale (80g) con pomodoro fresco, aglio e basilico + insalata di carote e cetrioli

Cena: Frittata con 2 uova, zucchine e cipolle (1 cucchiaio d'olio) + 1 fetta di pane integrale

Martedì

Colazione: Caffè + porridge di avena (40g) con semi di lino e 1 cucchiaino di miele + 1 banana

Pranzo: Riso integrale (80g) con lenticchie (60g) e carote + 1 arancia

Cena: Zuppa di cavolo, patate e cipolle (senza olio aggiunto) + 1 fetta di pane integrale

Mercoledì (No Cooking Day)

Colazione: Yogurt di soia + 2 gallette di riso con marmellata + 1 pera

Pranzo: Insalata di ceci (in scatola, 100g) con pomodori, cetrioli e olio (1 cucchiaino) + 1 uovo sodo

Cena: Pane integrale (2 fette) con hummus (fatto in casa o comprato in offerta) + carote crude

Giovedì

Colazione: Tè + 2 fette di pane integrale con crema di mandorle (fatta in casa) + 1 kiwi

Pranzo: Minestrone di verdure (patate, carote, cavolo, pomodori) + 1 fetta di pane integrale

Cena: Tofu (100g) saltato con zucchine e peperoni (1 cucchiaio d'olio) + riso integrale (60g)

Venerdì

Colazione: Caffè + 1 fetta di pane integrale con olio e pomodoro + 1 arancia

Pranzo: Pasta e fagioli (80g pasta, 60g fagioli) + insalata verde

Cena: Zuppa di orzo e verdure (senza olio) + 1 uovo sodo

Sabato

Colazione: Latte di soia + cereali integrali (40g) + 1 mela

Pranzo: Riso integrale (80g) con verdure saltate (zucchine, peperoni, cipolle) + 1 uovo sodo

Cena: Polenta (100g) con sugo di pomodoro e fagioli (60g)

Domenica

Colazione: Tè + 2 fette biscottate con marmellata + 1 banana

Pranzo: Pizza fatta in casa (impasto con farina integrale, pomodoro, origano, verdure grigliate)

Cena: Zuppa di lenticchie e patate + insalata verde

Lista della spesa (35€):

Carboidrati (6€):

- Pasta integrale (500g)
- Riso integrale (500g)
- Orzo (250g)
- Farina integrale (500g)
- Pane integrale (1 confezione)
- Avena (250g)
- Patate (1 kg)

Proteine vegetali (8€):

- Lenticchie (250g)
- Ceci (250g)
- Fagioli (250g)

- Tofu (200g)
- Uova (6)

Verdure (10€):

- Carote, zucchine, cipolle, pomodori, cavolo, cetrioli, peperoni, lattuga (tutto di stagione o in offerta)

Frutta (5€):

- Mele, banane, arance, pere (2-3 pezzi per tipo)

Altro (6€):

- Latte di soia (1 litro)
- Yogurt di soia (2 vasetti)
- Olio d'oliva (piccola bottiglia)
- Sale, spezie, basilico, aglio, semi di lino, marmellata senza zucchero

Consigli per risparmiare e perdere peso:

- Acquista legumi secchi (costano meno di quelli in scatola).
- Usa le verdure surgelate (spesso più economiche e sempre disponibili).
- Prepara porzioni controllate (usa la bilancia per pesare pasta/riso).
- Bevi molta acqua (aiuta a controllare la fame).
- Evita snack calorici (sostituisci con frutta o verdure crude).
- Muoviti di più (camminare 30-60 minuti al giorno aiuta a perdere peso).

Nota: Per perdere 1 kg a settimana, è fondamentale non saltare i pasti e mantenere il deficit calorico senza eccessi. Se vuoi, posso fornirti anche le ricette dettagliate per alcuni piatti o suggerimenti per variare il menù! Vuoi aggiustare qualcosa?

Il sistema, di fronte una richiesta più difficile rispetto alle precedenti, si è comportato mediamente bene, modificando correttamente la dieta. Questa volta, inoltre, rispetto ai punti precedenti, sono stati inserite le quantità all'interno della dieta e i prezzi indicativi per ogni classe di prodotto, risolvendo, quindi, i problemi riscontrati in precedenza.

3.6.4 Risposta finale

Possiamo, a questo punto, analizzare la risposta del sistema al nostro quinto ed ultimo prompt.

Utente

La dieta va bene, ma la situazione è cambiata nuovamente:

- La persona vuole perdere 1kg a settimana, forse 0.5kg, o forse 0.3kg, è indecisa e varia in base a se farà attività fisica o meno.
- Non conosce i turni di lavoro e non sa quando sarà a casa e quando avrà tempo di cucinare.
- Non vuole ripetere più di due volte a settimana lo stesso ingrediente.
- Non ha accesso ad un grande supermercato nelle vicinanze, solamente un piccolo negozio di alimentari.
- Potrebbe saltare uno dei pasti di tanto in tanto, ma non è dato sapere quando e quale.

Mistral

Grazie per i nuovi dettagli! Ecco una dieta flessibile, senza lattosio, senza pesce, con budget di 35€, che:

- Permette di perdere 0.3-1 kg a settimana (a seconda dell'attività fisica e dei pasti saltati).
- Non ripete mai un ingrediente più di 2 volte a settimana.
- È adattabile a turni di lavoro imprevedibili (piatti veloci, conservabili, o "no cooking").
- Usa solo ingredienti reperibili in un piccolo alimentari.
- Resiste ai pasti saltati (pasti bilanciati e sazianti, spuntini pratici).

Linee guida generali

- *Colazione:* Sempre veloce e nutriente (es. frutta + carboidrati + proteine).
- *Pranzo/Cena:* Piatti unici, facili da preparare o da portare via.
- *Spuntini (opzionali):* Frutta, verdure crude, uova sode, gallette.
- *Conservazione:* Usa ingredienti che durano (es. uova, patate, carote, cipolle, legumi in scatola, riso, pasta).
- *Flessibilità:* Se salta un pasto, recupera con un pasto leggero ma bilanciato dopo.

Menù Settimanale Flessibile

(Ogni giorno puoi scambiare pranzo/cena o sostituire con opzioni "no cooking")

Lunedì

Colazione: 1 banana + 2 gallette di riso + 1 cucchiaino di crema di mandorle

Pranzo: Pasta integrale (80g) con pomodoro in scatola, aglio e basilico + insalata di carote

Cena: Frittata con 2 uova e zucchine (congelate o fresche) + 1 fetta di pane integrale

Martedì

Colazione: 1 mela + 1 yogurt di soia + 1 manciata di noci

Pranzo: Riso integrale (80g) con lenticchie in scatola e cipolla + 1 arancia

Cena: No cooking: Insalata di ceci in scatola, pomodori, cetrioli e olio + 1 uovo sodo

Mercoledì

Colazione: 1 pera + 2 fette biscottate con marmellata

Pranzo: Minestrone con patate, carote, cavolo e pomodoro (preparato in quantità) + 1 fetta di pane

Cena: Tofu (100g) saltato con peperoni e cipolle (congelati) + 1 patata lessa

Giovedì

Colazione: 1 kiwi + porridge di avena (40g) con semi di lino

Pranzo: No cooking: Panino integrale con hummus (in barattolo) e carote crude + 1 frutto

Cena: Pasta e fagioli (80g pasta, 60g fagioli in scatola) + insalata verde

Venerdì

Colazione: 1 arancia + 1 fetta di pane integrale con olio e pomodoro

Pranzo: Polenta (100g) con sugo di pomodoro e fagioli + verdure grigliate (se possibile)

Cena: Omelette con spinaci (surgelati) e 1 uovo + 1 fetta di pane

Sabato

Colazione: 1 banana + 1 yogurt di soia

Pranzo: Riso integrale (80g) con verdure saltate (zucchine, cipolle) + 1 uovo sodo

Cena: *No cooking*: Insalata di tonno vegetale (in scatola, se disponibile) o lenticchie + pane integrale

Domenica

Colazione: 1 mela + 2 gallette con crema di nocciole

Pranzo: Pizza fatta in casa (impasto con farina, pomodoro, origano, verdure)

Cena: Zuppa di orzo e verdure (preparata in quantità) + 1 fetta di pane

Lista della spesa (35€) per piccolo alimentari

Carboidrati (6€):

- Pasta integrale (500g)
- Riso integrale (500g)
- Orzo (250g)
- Farina (500g)
- Pane integrale (1 confezione)
- Gallette di riso (1 confezione)
- Patate (1 kg)

Proteine (8€):

- Uova (6)
- Lenticchie in scatola (2 lattine)

- Ceci in scatola (2 lattine)
- Fagioli in scatola (1 lattina)
- Tofu (1 confezione, se disponibile)

Verdure (8€):

- Carote, cipolle, zucchine (fresche o surgelate), peperoni (surgelati), pomodori in scatola, cavolo, spinaci surgelati

Frutta (5€):

- Mele, banane, arance, pere (2-3 pezzi per tipo)

Altro (8€):

- Yogurt di soia (2 vasetti)
- Latte di soia (1 litro)
- Olio d'oliva (piccola bottiglia)
- Marmellata senza zucchero
- Semi di lino o noci (piccola confezione)
- Hummus (in barattolo, se disponibile)

Consigli pratici

- *Prepara in anticipo:* Cuoci una volta a settimana legumi, riso, verdure e conserva in frigo.
- *Opzioni "no cooking":* Uova sode, insalate di legumi in scatola, pane integrale con hummus o crema di frutta secca.
- *Se salti un pasto:* Recupera con uno spuntino bilanciato (es. frutta + noci, yogurt + gallette).
- *Attività fisica:* Se fai movimento, aumenta leggermente le porzioni di carboidrati (es. +20g di pasta/riso).

Domande utili per te:

- Vuoi che ti suggerisca come organizzare la spesa in base a cosa trovi nel tuo alimentari?
- Preferisci che ti dia idee per sostituire gli ingredienti se non li trovi?
- Vuoi consigli su come conservare i cibi per evitarne lo spreco?

Il sistema ha prodotto una dieta nonostante le diverse difficoltà. Rispetto agli altri modelli

visti finora, si ha una dieta vera e propria. Questo approccio asseconda la richiesta dell'utente, ma è scarso di informazioni e dettagli e, di conseguenza, probabilmente non sarebbe realmente applicabile di fronte ad una situazione così complicata. Inoltre, il sistema ha cambiato il giorno senza cucina dal Mercoledì al Sabato, senza esporre le motivazioni.

3.6.5 Valutazioni sul comportamento di Mistral

In linea generale, Mistral ha soddisfatto le nostre richieste, dalla prima all'ultima. Tuttavia, ha presentato alcuni limiti nel corso del caso di studio.

In primis, fino al quarto prompt, non ha mai specificato le quantità all'interno della dieta, cosa che gli altri sistemi utilizzati finora facevano, presentando, quindi, un livello di efficienza più basso.

Lo stesso ragionamento è stato seguito anche per la spesa, dove non sono stati stimati i singoli costi di ogni elemento, non consentendo, quindi, di vedere il costo effettivo totale e dando per scontato che venga rispettato il budget imposto.

Un ulteriore problema riguarda l'ultimo prompt: alla fine della risposta, fra i consigli, Mistral suggerisce come spuntino lo yogurt, dimenticandosi dell'intolleranza al lattosio imposta precedentemente.

Inoltre, tralasciando le problematiche, il modello non offre molti spunti, presentando poca intraprendenza ed un livello di conoscenza inferiore. Esso, difatti, si limita a rispondere al problema, aggiungendo pochi dettagli e senza approfondirli in modo specifico.

3.7 Analisi con Claude

All'interno di questa sezione sottoporremo a Claude, il sistema di Intelligenza Artificiale Generativa di Anthropic, il nostro caso di studio per studiarne il comportamento.

3.7.1 Risposta al prompt iniziale

Partiamo fornendo in input al sistema il nostro primo prompt.

Utente

Ciao, prepara una dieta settimanale bilanciata per una persona adulta, senza esigenze particolari. Colazione, pranzo e cena per la settimana.

Claude

Piano Alimentare Settimanale Bilanciato

Lunedì

Colazione

- Yogurt greco con muesli, mirtilli freschi e un cucchiaino di miele
- Tè verde o caffè

Pranzo

- Pasta integrale al pomodoro fresco e basilico (80g)
- Insalata mista con olio extravergine d'oliva

- Una mela

Cena

- Petto di pollo alla griglia (150g)
- Verdure grigliate (zucchine, melanzane, peperoni)
- Pane integrale (50g)

Martedì

Colazione

- Fette biscottate integrali con marmellata
- Spremuta d'arancia fresca
- Caffè o tè

Pranzo

- Riso basmati con verdure saltate e gamberetti (80g riso)
- Insalata di pomodori
- Una pera

Cena

- Salmone al forno con limone (150g)
- Spinaci saltati in padella con aglio
- Patate lesse (200g)

Mercoledì

Colazione

- Porridge di avena con banana a fette e noci
- Yogurt bianco
- Tè o caffè

Pranzo

- Farro con ceci e verdure (70g farro)
- Carote crude
- Kiwi

Cena

- Omelette con 2 uova e verdure miste
 - Insalata verde abbondante
 - Pane di segale (50g)
-

Giovedì

Colazione

- Toast integrale con ricotta e miele
- Macedonia di frutta fresca
- Caffè o tè

Pranzo

- Pasta integrale con pesto e fagiolini (80g)
- Insalata di finocchi
- Una mela

Cena

- Tacchino arrosto (150g)
 - Broccoli al vapore
 - Quinoa (60g peso secco)
-

Venerdì

Colazione

- Pancake integrali con frutti di bosco
- Yogurt greco
- Tè verde o caffè

Pranzo

- Riso integrale con verdure al curry (80g)
- Insalata di lattuga e radicchio
- Un'arancia

Cena

- Pesce spada alla griglia (150g)
 - Caponata di melanzane
 - Pane integrale (50g)
-

Sabato

Colazione

- Muesli con latte parzialmente scremato
- Banana
- Caffè o tè

Pranzo

- Couscous con verdure mediterranee e ceci (70g)
- Insalata di rucola con scaglie di parmigiano
- Fragole

Cena

- Bistecca di manzo magra (150g)
 - Insalata mista abbondante
 - Patate dolci al forno (200g)
-

Domenica

Colazione

- Pane integrale con burro di arachidi e marmellata
- Frullato di frutta
- Caffè o tè

Pranzo

- Risotto con zucca e parmigiano (80g riso)
- Insalata di pomodori e cetrioli
- Uva

Cena

- Merluzzo al cartoccio (150g)

- Asparagi al vapore
- Farro o orzo (60g peso secco)

Note Generali

- *Spuntini*: A metà mattina e metà pomeriggio puoi inserire uno yogurt, un frutto fresco, una manciata di frutta secca (30g) o verdure crude
- *Acqua*: Bere almeno 1,5-2 litri di acqua al giorno
- *Condimenti*: Preferire olio extravergine d'oliva a crudo, limitare il sale
- *Porzioni*: Le quantità sono indicative per un adulto medio (circa 2000 kcal/giorno) e possono essere adattate in base alle esigenze individuali

Importante: Questo piano è generico e bilanciato. Per esigenze specifiche o condizioni particolari, consultare un nutrizionista.

Ti ho preparato un piano alimentare settimanale completo e bilanciato!
Ho creato un menù vario che include:

- *Carboidrati complessi*: pasta integrale, riso, farro, quinoa, patate
- *Proteine*: carni bianche, pesce, uova, legumi
- *Verdure*: di stagione e in abbondanza
- *Frutta*: diversificata durante la giornata
- *Grassi buoni*: olio extravergine d'oliva, frutta secca

Ogni giorno è pensato per fornire circa 2000 calorie con un buon equilibrio di macronutrienti. Le porzioni sono indicative e possono essere adattate in base al tuo fabbisogno calorico, livello di attività fisica e obiettivi personali.

Ricorda di bere molta acqua e, se vuoi, puoi aggiungere due spuntini giornalieri tra i pasti principali. Se hai esigenze specifiche o vuoi personalizzare ulteriormente il piano, sarò felice di aiutarti!

Il sistema ha fornito una risposta di ottimo livello, utilizzando diversi ingredienti e fornendo, indicativamente, le quantità da utilizzare. Inoltre, ha fornito il numero di calorie giornaliere, mostrando un elevato livello di conoscenza nel contesto.

3.7.2 Risposta ai vincoli di budget e alle intolleranze

Vediamo ora come Claude si comporta di fronte all'aggiunta di vincoli economici ed intolleranze. Partiamo con l'aggiunta di un budget settimanale di 50€.

Utente

Ora la persona che deve mettersi a dieta ha un budget massimo di 50€ per tutta la settimana. Mantieni la dieta bilanciata e varia rispettando il budget.

Claude**Piano Alimentare Settimanale Bilanciato - Budget 50€***Lunedì*

Colazione

- Latte con fette biscottate e marmellata
- Una banana
- Caffè o tè

Pranzo

- Pasta al pomodoro e basilico (80g)
- Carote crude condite
- Una mela

Cena

- Frittata con 2 uova e zucchine
- Insalata verde con olio
- Pane (50g)

Martedì

Colazione

- Yogurt bianco con corn flakes e miele
- Caffè o tè

Pranzo

- Riso con lenticchie e pomodoro (80g riso, 50g lenticchie secche)
- Insalata di finocchi
- Una mela

Cena

- Petto di pollo al forno (150g)
- Patate al forno (200g)
- Insalata mista

Mercoledì

Colazione

- Latte con biscotti secchi
- Una mela
- Tè o caffè

Pranzo

- Pasta e ceci (80g pasta, 50g ceci secchi)
- Insalata di pomodori
- Pane (50g)

Cena

- Merluzzo surgelato al forno (150g)
- Spinaci lessati
- Patate lesse (200g)

Giovedì

Colazione

- Yogurt bianco con corn flakes
- Una banana
- Caffè o tè

Pranzo

- Riso con piselli e pomodoro (80g riso)
- Carote crude
- Una mela

Cena

- Pasta al tonno in scatola (80g pasta)
- Insalata mista abbondante
- Pane (50g)

Venerdì

Colazione

- Fette biscottate con marmellata
- Latte
- Caffè o tè

Pranzo

- Pasta e fagioli (80g pasta, 50g fagioli secchi)
- Insalata verde
- Una mela

Cena

- Uova sode (2) con maionese
- Insalata di pomodori e cetrioli
- Pane (50g)

Sabato

Colazione

- Latte con biscotti secchi
- Una banana
- Caffè o tè

Pranzo

- Pasta al pomodoro e ricotta (80g)
- Insalata mista
- Una mela

Cena

- Pollo arrosto (150g)
- Melanzane in padella
- Pane (50g)

Domenica

Colazione

- Yogurt bianco con corn flakes e miele
- Caffè o tè

Pranzo

- Riso con ragù di carne macinata (80g riso, 100g carne)
- Insalata di finocchi
- Una mela

Cena

- Frittata con 2 uova e patate
- Insalata verde abbondante
- Pane (50g)

Note Generali

- *Spuntini*: A metà mattina e metà pomeriggio puoi inserire uno yogurt, un frutto fresco, una manciata di frutta secca (30g) o verdure crude
- *Bere almeno 1,5-2 litri di acqua al giorno*
- *Condimenti*: Preferire olio extravergine d'oliva a crudo, limitare il sale
- *Porzioni*: Le quantità sono indicative per un adulto medio (circa 2000 kcal/giorno) e possono essere adattate in base alle esigenze individuali

Lista della spesa economica (50€)

Dispensa base:

- Pasta (1kg) - €1,50
- Riso (1kg) - €1,80
- Legumi secchi: ceci, lenticchie, fagioli (500g ciascuno) - €3,00
- Passata di pomodoro (3 confezioni) - €3,00
- Tonno in scatola (3 scatole) - €3,00
- Olio extravergine d'oliva (0,5L) - €4,00
- Corn flakes (500g) - €2,00

- Fette biscottate o biscotti secchi - €1,50
- Marmellata - €2,00
- Sale, spezie - €1,00

Proteine:

- Uova (18 pezzi) - €3,50
- Pollo (800g) - €6,00
- Carne macinata (400g) - €3,50
- Merluzzo surgelato (400g) - €4,00

Latticini:

- Latte (3L) - €3,50
- Yogurt bianco (6 vasetti) - €2,50
- Ricotta (250g) - €1,50

Frutta e Verdura:

- Mele (2kg) - €3,00
- Banane (1,5kg) - €2,00
- Patate (2kg) - €2,00
- Pomodori (1kg) - €1,50
- Zucchine (1kg) - €1,50
- Carote (1kg) - €1,00
- Insalata mista/verde (3 cespi) - €2,50
- Finocchi (500g) - €1,50
- Spinaci surgelati (450g) - €1,50
- Melanzane (500g) - €1,50
- Cipolle, aglio, basilico - €1,50

Altro:

- Pane fresco settimanale - €3,50

TOTALE: circa 48-50€

Consigli per Risparmiare

- Acquista prodotti di marca del supermercato

- Scegli frutta e verdura di stagione
- Legumi secchi invece che in scatola
- Pesce surgelato invece che fresco
- Pollo invece di carni più costose
- Evita prodotti preconfezionati o biologici

Importante: Questo piano mantiene l'equilibrio nutrizionale ottimizzando i costi. Per esigenze specifiche, consultare un nutrizionista.

Perfetto! Ho riadattato la dieta per rispettare il budget di 50€ settimanali mantenendola bilanciata e nutriente.

Le principali modifiche per risparmiare:

- *Proteine economiche:* più uova, legumi secchi, pollo, carne macinata, tonno in scatola e merluzzo surgelato invece di pesci freschi costosi
- *Cereali base:* pasta e riso normali invece di integrali o cereali speciali
- *Latticini essenziali:* latte normale, yogurt bianco semplice, ricotta invece di yogurt greco o formaggi costosi
- *Verdure economiche:* patate, carote, zucchine, insalata, verdure di stagione
- *Frutta accessibile:* mele e banane che costano poco e durano a lungo

Ho incluso anche una lista della spesa dettagliata con i prezzi stimati che ti porta a circa 48-50€ totali. La dieta rimane completa con carboidrati, proteine, verdure, frutta e grassi buoni, garantendo un'alimentazione sana ed equilibrata pur con un budget limitato!

Il sistema ha modificato correttamente la dieta, lasciandola varia e funzionale.

Coma già anticipato nelle sezioni precedenti, Claude crea dei documenti in cui inserisce la risposta alla richiesta. Nel nostro caso il sistema ha prodotto due documenti diversi, denominati, rispettivamente, "Piano Alimentare Settimanale Bilanciato - Versione 3" e "Piano Alimentare Settimanale Bilanciato - Versione 4". I documenti risultano in gran parte identici, l'unica differenza è data dalla presenza della spesa: nella Versione 4, infatti, c'è una sezione aggiuntiva riguardante la lista della spesa.

Per riportare il nostro caso di studio in maniera più semplice e chiara, le due risposte sono state accorpate in un'unica risposta.

Poniamo ora il sistema di fronte l'aggiunta di intolleranze.

Utente

La persona è anche intollerante al lattosio, e preferisce evitare il pesce se possibile. Ricrea la dieta anche in base a queste informazioni, mantenendo il budget di 50 euro.

Claude**Piano Alimentare Settimanale - Budget 50€(Senza Lattosio e Pesce)***Lunedì*

Colazione

- Latte di soia con fette biscottate e marmellata
- Una banana
- Caffè o tè

Pranzo

- Pasta al pomodoro e basilico (80g)
- Carote crude condite con olio
- Una mela

Cena

- Petto di pollo al forno (150g)
- Patate al forno (200g)
- Insalata verde con olio

Martedì

Colazione

- Fette biscottate con burro di arachidi e miele
- Una mela
- Caffè o tè

Pranzo

- Riso con lenticchie al pomodoro (80g riso, 50g lenticchie secche)
- Insalata di finocchi
- Pane (50g)

Cena

- Frittata con 2 uova e zucchine
- Insalata mista abbondante
- Pane (50g)

Mercoledì

Colazione

- Latte di soia con corn flakes
- Una banana
- Tè o caffè

Pranzo

- Pasta e ceci al pomodoro (80g pasta, 50g ceci secchi)
- Insalata di pomodori
- Una mela

Cena

- Polpette di carne macinata (150g) al sugo
- Melanzane in padella
- Pane (50g)

Giovedì

Colazione

- Fette biscottate con marmellata
- Spremuta d'arancia (1 arancia)
- Caffè o tè

Pranzo

- Riso con piselli e pomodoro (80g riso)
- Carote crude
- Una mela

Cena

- Petto di pollo alla piastra (150g)
- Spinaci saltati in padella con aglio
- Patate lesse (200g)

Venerdì

Colazione

- Latte di soia con biscotti secchi
- Una banana
- Caffè o tè

Pranzo

- Pasta e fagioli (80g pasta, 50g fagioli secchi)
- Insalata verde
- Pane (50g)

Cena

- Uova strapazzate (2) con pomodori
- Insalata mista
- Pane (50g)

Sabato

Colazione

- Fette biscottate con burro di arachidi
- Una mela
- Caffè o tè

Pranzo

- Riso alla cantonese con piselli, carote e uova (80g riso)
- Insalata di finocchi
- Una mela

Cena

- Spezzatino di pollo con patate (150g pollo, 200g patate)
- Insalata verde
- Pane (50g)

Domenica

Colazione

- Latte di soia con corn flakes e miele
- Una banana
- Caffè o tè

Pranzo

- Pasta al ragù (80g pasta, 100g carne macinata)
- Insalata mista
- Una mela

Cena

- Frittata di patate con 2 uova (200g patate)
- Zucchine trifolate
- Pane (50g)

Note Generali

- *Spuntini*: Frutta di stagione (mele, banane, arance), fette biscottate con marmellata, frutta secca (mandorle, noci)
- *Acqua*: Bere almeno 1,5-2 litri di acqua al giorno
- *Condimenti*: Olio extravergine d'oliva, limitare il sale
- *Porzioni*: Le quantità sono indicative per un adulto medio (circa 2000 kcal/giorno)

Lista della spesa economica (circa 50€)

Dispensa base:

- Pasta (1kg) - €1,50
- Riso (1kg) - €1,80
- Legumi secchi: ceci, lenticchie, fagioli (500g ciascuno) - €3,00
- Passata di pomodoro (3 confezioni) - €3,00
- Olio extravergine d'oliva (0,5L) - €4,00
- Corn flakes (500g) - €2,00
- Fette biscottate senza lattosio - €1,80

- Biscotti secchi senza lattosio - €1,50
- Marmellata - €2,00
- Burro di arachidi (250g) - €2,50
- Miele - €2,00
- Sale, spezie, aglio - €1,00

Proteine:

- Uova (18 pezzi) - €3,50
- Pollo (1,2kg) - €8,50
- Carne macinata (500g) - €4,00

Bevande vegetali:

- Latte di soia (3L) - €4,50

Frutta e Verdura:

- Mele (2kg) - €3,00
- Banane (1,5kg) - €2,00
- Arance (1kg) - €1,50
- Patate (2,5kg) - €2,50
- Pomodori (1kg) - €1,50
- Zucchine (1kg) - €1,50
- Carote (1kg) - €1,00
- Insalata mista/verde (3 cespi) - €2,50
- Finocchi (500g) - €1,50
- Spinaci surgelati (450g) - €1,50
- Melanzane (500g) - €1,50
- Piselli surgelati (450g) - €1,50
- Cipolle, basilico - €1,00

Altro:

- Pane fresco settimanale - €3,50

TOTALE: circa 48-50€

Vantaggi Nutrizionali

Proteine complete: da pollo, uova, legumi e combinazioni cereali-legumi; Calcio alternativo: dal latte di soia arricchito, mandorle, verdure a foglia verde; Ferro: da carne, legumi, spinaci; Carboidrati: pasta, riso, patate, pane; Fibre: da legumi, verdure, frutta; Grassi buoni: olio d'oliva, burro di arachidi, frutta secca.

Consigli per Risparmiare

- Acquista prodotti di marca del supermercato
- Verifica che fette biscottate e biscotti siano naturalmente senza lattosio (molti lo sono)
- Scegli frutta e verdura di stagione
- Legumi secchi invece che in scatola (più economici)
- Pollo intero da dividere invece di petti già tagliati
- Latte di soia in brick da 1L (più economico del formato piccolo)

Importante: Questa dieta è equilibrata, senza lattosio e senza pesce, rispettando il budget di 50€. Per esigenze specifiche, consultare un nutrizionista.

Perfetto! Ho ricreato completamente la dieta eliminando lattosio e pesce, mantenendo il budget di 50€ e garantendo comunque un'alimentazione completa e bilanciata.

Le principali modifiche:

Niente lattosio: sostituito latte vaccino con latte di soia, eliminati yogurt e formaggi; *Niente pesce:* aumentata la quota di pollo e uova, più legumi per le proteine; *Proteine complete:* combinazioni di cereali e legumi (pasta e ceci, riso e lenticchie) che forniscono tutti gli aminoacidi essenziali; *Calcio alternativo:* il latte di soia è spesso arricchito di calcio, più verdure a foglia verde.

Fonti proteiche principali:

- Pollo (economico e versatile)
- Uova (ottime proteine a basso costo)
- Legumi secchi (ceci, lenticchie, fagioli)
- Burro di arachidi (proteine vegetali e grassi buoni)

La dieta rimane varia, gustosa e nutriente, con tutti i macronutrienti necessari, rispettando perfettamente le tue esigenze e il budget!

Il sistema ha ideato nuovamente la dieta, eliminando elementi contenenti lattosio e il pesce, rispondendo correttamente alla richiesta. Il budget è ancora rispettato secondo le stime del sistema, e i piatti proposti presentano opzioni non scontate (come il riso alla cantonese, ad esempio).

3.7.3 Risposta ai vincoli sul peso e sulla casualità

Poniamo ora al sistema il quarto prompt, che presenta un ulteriore grado di difficoltà.

Utente

Ci sono altri aspetti che non erano stati specificati inizialmente:

- Il budget settimanale, a causa di alcuni problemi, scende a 35.
- Un giorno a caso della settimana non può cucinare nulla.
- Il peso da perdere è di circa 1kg a settimana

Claude

Piano Alimentare Settimanale - Budget 35€(Senza Lattosio/Pesce, Dimagrimento)

Lunedì

Colazione

- Latte di soia (200ml) con fette biscottate (3) e marmellata
- Caffè o tè

Pranzo

- Pasta al pomodoro (60g)
- Insalata mista con olio (1 cucchiaino)
- Una mela

Cena

- Petto di pollo alla piastra (120g)
- Zucchine grigliate abbondanti
- Insalata verde

Martedì

Colazione

- Fette biscottate (3) con burro di arachidi (1 cucchiaino)
- Una mela
- Caffè o tè

Pranzo

- Riso con lenticchie al pomodoro (50g riso, 40g lenticchie secche)
- Carote crude

- Una mela

Cena

- Frittata con 2 uova e spinaci
 - Insalata mista abbondante
 - Pane (30g)
-

Mercoledì

Colazione

- Latte di soia (200ml) con fette biscottate (3)
- Una banana
- Tè o caffè

Pranzo

- Pasta e ceci (60g pasta, 40g ceci secchi)
- Insalata di pomodori
- Una mela

Cena

- Pollo al forno (120g)
 - Melanzane in padella
 - Insalata verde abbondante
-

Giovedì - GIORNO SENZA CUCINARE

Colazione

- Latte di soia (200ml) con fette biscottate (3) e marmellata
- Caffè o tè

Pranzo

- Panino (80g pane) con tonno in scatola (1 scatoletta da 80g sgocciolato)
- Carote crude
- Una mela

Cena

- 2 uova sode (preparate il giorno prima)
 - Pomodori a fette con olio
 - Pane (50g)
 - Frutta secca (20g mandorle)
-

Venerdì

Colazione

- Fette biscottate (3) con marmellata
- Una banana
- Caffè o tè

Pranzo

- Riso con piselli (50g riso, piselli surgelati)
- Insalata mista
- Una mela

Cena

- Polpette di carne macinata magra (100g) al sugo
 - Spinaci lessati abbondanti
 - Pane (30g)
-

Sabato

Colazione

- Latte di soia (200ml) con fette biscottate (3)
- Una mela
- Caffè o tè

Pranzo

- Pasta e fagioli (60g pasta, 40g fagioli secchi)
- Insalata verde
- Una mela

Cena

- Petto di pollo (120g)
 - Patate lesse (150g)
 - Insalata mista abbondante
-

Domenica

Colazione

- Fette biscottate (3) con burro di arachidi
- Una banana
- Caffè o tè

Pranzo

- Pasta al ragù leggero (60g pasta, 80g carne macinata magra)
- Insalata mista
- Una mela

Cena

- Frittata con 2 uova e zucchine
 - Insalata verde abbondante
 - Carote crude
-

Informazioni Nutrizionali

Calorie giornaliere: circa 1300-1400 kcal

- Deficit calorico di circa 600-700 kcal/giorno rispetto al fabbisogno medio
- Perdita peso prevista: circa 1kg/settimana

Macronutrienti bilanciati:

- Proteine: 25-30% (da pollo, uova, legumi)
 - Carboidrati: 45-50% (ridotti ma presenti)
 - Grassi: 25% (olio d'oliva, frutta secca)
-

Spuntini Consentiti (Facoltativi)

Metà mattina o pomeriggio (se necessario):

- Una mela o banana
- Carote crude
- Finocchi crudi
- Tè o caffè senza zucchero

Importante: Bere almeno 2 litri d'acqua al giorno per favorire il dimagrimento

Lista della Spesa Economica (circa 35€)

Dispensa base:

- Pasta (750g) - €1,00
- Riso (500g) - €0,90
- Legumi secchi: ceci, lenticchie, fagioli (400g ciascuno) - €2,40
- Passata di pomodoro (2 confezioni) - €2,00
- Tonno in scatola (2 scatole da 80g) - €2,00
- Olio extravergine d'oliva (0,5L) - €4,00
- Fette biscottate (1 pacco) - €1,50
- Marmellata - €1,50
- Burro di arachidi (vasetto piccolo) - €2,00
- Sale, spezie - €0,50

Proteine:

- Uova (12 pezzi) - €2,50
- Pollo (800g) - €5,50
- Carne macinata magra (400g) - €3,20

Bevande vegetali:

- Latte di soia (2L) - €3,00

Frutta e Verdura:

- Mele (1,5kg) - €2,20
- Banane (1kg) - €1,30
- Patate (1kg) - €1,00
- Pomodori (750g) - €1,20
- Zucchine (750g) - €1,10

- Carote (750g) - €0,80
- Insalata (2 cespi) - €1,60
- Spinaci surgelati (450g) - €1,50
- Melanzane (400g) - €1,20
- Piselli surgelati (300g) - €1,00
- Cipolle, aglio - €0,70

Altro:

- Pane fresco settimanale (ridotto) - €2,00
- Mandorle/frutta secca (100g) - €1,50

TOTALE: circa 34-35€

Strategia per Perdere 1kg/Settimana

Calorie e Deficit

- *Fabbisogno medio adulto*: 2000 kcal/giorno
- *Piano dietetico*: 1300-1400 kcal/giorno
- *Deficit*: 600-700 kcal/giorno = 4200-4900 kcal/settimana
- *Perdita peso teorica*: circa 1kg/settimana (7000 kcal = 1kg grasso)

Consigli Fondamentali

Porzioni ridotte: rispettare rigorosamente i grammi indicati; *Verdure abbondanti*: saziano con poche calorie; *Proteine ad ogni pasto*: mantengono la massa muscolare; *Acqua*: almeno 2L al giorno, aiuta a dimagrire; *Cotture light*: vapore, piastra, forno senza grassi aggiunti; *Condimenti misurati*: olio con il cucchiaino (1 cucchiaino = 5ml); *No zucchero*: in caffè/tè; *Attività fisica*: camminata 30 minuti/giorno accelera il dimagrimento.

Giorno Senza Cucinare (Giovedì)

- Cibi che non richiedono cottura
- Tonno in scatola (già pronto)
- Uova sode (preparate mercoledì sera)
- Pane, verdure crude, frutta

Risparmio sul Budget

- Eliminati prodotti costosi (pesce fresco, formaggi, yogurt)
- Porzioni ridotte = meno spesa
- Legumi secchi (economici e proteici)

- Frutta e verdura di base
- Pollo e uova (proteine economiche)
- Pane ridotto al minimo

Avvertenze Importanti

Questo piano prevede un deficit calorico significativo

- Consultare un medico prima di iniziare
- Non adatto a: gravidanza, allattamento, età evolutiva, anziani
- Integrare vitamina B12 se prolungato oltre 4 settimane
- In caso di stanchezza eccessiva, aumentare leggermente le porzioni
- Pesarsi 1 volta a settimana, stessa ora, stesse condizioni
- Se la perdita supera 1,5kg/settimana, aumentare le calorie

Durata consigliata: massimo 8-12 settimane, poi piano di mantenimento

Importante: Per un percorso sicuro ed efficace, consultare un nutrizionista o dietologo.

Ho completamente ristrutturato la dieta considerando tutti i nuovi vincoli:

Modifiche principali:

Budget ridotto a 35€:

- Porzioni più piccole (quindi meno spesa)
- Eliminati prodotti non essenziali
- Focus su ingredienti base economici
- Legumi secchi come fonte proteica principale

Giovedì = giorno senza cucinare:

- Pasti freddi o che non richiedono cottura
- Tonno in scatola, uova sode (preparate il giorno prima)
- Panini, verdure crude, frutta

Perdita di 1kg/settimana:

- Calorie ridotte a 1300-1400 kcal/giorno
- Deficit di 600-700 kcal/giorno
- Porzioni precise (pasta 60g invece di 80g)
- Proteine mantenute alte per preservare massa muscolare

- Verdure abbondanti per saziare

Strategia nutrizionale:

- Proteine: pollo, uova, legumi (mantengono muscoli)
- Carboidrati ridotti ma presenti (energia)
- Grassi buoni misurati (olio, frutta secca)
- Molta verdura (volume con poche calorie)

La dieta è completa, senza lattosio e pesce, rispetta il budget strettissimo e permette un dimagrimento sano di 1kg/settimana. Ho incluso anche tutte le avvertenze mediche importanti!

Il sistema, di fronte ad una richiesta ben più complicata rispetto le precedenti, ha comunque prodotto una soluzione di buon livello. La dieta, nonostante le limitazioni e le difficoltà di cui tener conto, risulta funzionale e variegata, e i vincoli imposti sono rispettati. Lo stesso discorso vale per la spesa.

Inoltre, il sistema offre all'utente dei consigli interessanti, come l'integrazione della vitamina B12, la durata consigliata della dieta e informazioni riguardanti il peso.

All'interno della risposta, però, è presente un problema enorme: Claude riporta la lista della spesa specificando che essa rispetti ai 35€, ma se si sommano i singoli costi stimati dal sistema, il risultato sarà uguale a 49,10€. Questo problema mostra come il sistema ha tentato di ingannare l'utente, nella speranza che non effettuasse i calcoli. Inoltre, oltre ai calcoli errati, il sistema scrive giustamente nel titolo "Senza Pesce", ma all'interno della spesa e della dieta utilizza il tonno, presentando, quindi, un'ulteriore mancanza.

3.7.4 Risposta finale

In questa sezione analizzeremo il comportamento di Claude una volta sottoposto al nostro ultimo prompt, riguardante una situazione piena di incertezze e difficoltà.

Utente

La dieta va bene, ma la situazione è cambiata nuovamente:

- La persona vuole perdere 1kg a settimana, forse 0.5kg, o forse 0.3kg, è indecisa e varia in base a se farà attività fisica o meno.
- Non conosce i turni di lavoro e non sa quando sarà a casa e quando avrà tempo di cucinare.
- Non vuole ripetere più di due volte a settimana lo stesso ingrediente.
- Non ha accesso ad un grande supermercato nelle vicinanze, solamente un piccolo negozio di alimentari.
- Potrebbe saltare uno dei pasti di tanto in tanto, ma non è dato sapere quando e quale.

Claude

Mi dispiace, ma devo essere onesto con te: con tutti questi vincoli e variabili, creare un piano alimentare settimanale rigido e predefinito *non è la soluzione giusta* per questa persona.

Ecco perché:

Problemi con un piano fisso:

1. *Obiettivi variabili* (0.3-1kg/settimana) = calorie che cambiano di 300-700 kcal/giorno
2. *Orari imprevedibili* = impossibile pianificare quando cucinare
3. *Vincolo "max 2 volte stesso ingrediente"* = con budget 35€ è quasi impossibile (servirebbero 10+ proteine diverse)
4. *Negozi piccolo* = disponibilità limitata e imprevedibile
5. *Pasti saltati casualmente* = calorie giornaliere impossibili da bilanciare

Cosa serve invece:

Questa persona ha bisogno di un *approccio flessibile basato su principi*, non un menù rigido. Le suggerirei:

Sistema a "Blocchi Intercambiabili"

PROTEINE (120g a porzione):

- Pollo, uova, legumi in scatola, tonno, carne macinata

CARBOIDRATI (porzione variabile):

- Per dimagrire 1kg: 50g pasta/riso
- Per dimagrire 0.5kg: 70g pasta/riso
- Per dimagrire 0.3kg: 80g pasta/riso

VERDURE (sempre abbondanti):

- Quelle disponibili nel negozio quel giorno

PASTI VELOCI (senza cucinare):

- Tonno + pane + pomodori
- Uova sode + insalata
- Legumi in scatola + verdure crude

Strategia pratica:

1. *Spesa settimanale base*: solo ingredienti versatili e a lunga conservazione
2. *Decidere giorno per giorno in base a*: tempo disponibile, fame, obiettivo della settimana
3. *Regola semplice*: proteine + verdure ad ogni pasto, carboidrati secondo obiettivo
4. *Se salta un pasto*: non compensare, proseguire normalmente

Vuoi che ti prepari invece una "guida flessibile" con liste di opzioni intercambiabili e regole semplici da seguire? Sarebbe molto più utile di un menù settimanale che comunque non potrebbe rispettare.

Di fronte ad una situazione irrealistica, il sistema non ha prodotto una vera e propria dieta, bensì un piano variabile adattabile in base alle esigenze. Sono state definite alcune regole da applicare in base alle circostanze che si presentano, lasciando all'utente la gestione e la scelta degli alimenti.

3.7.5 Valutazioni sul comportamento di Claude

Nel complesso, il modello generativo di Anthropic si è comportato in modo ideale, specialmente per i primi quattro prompt. Le diete fornite risultano varie e consone alle richieste; le modifiche e i passaggi dopo ogni prompt vengono riportati dal sistema, rendendo, quindi, la risposta molto più chiara. Il modello si è occupato anche di fornire sempre degli spunti aggiuntivi, mostrando un'ottima efficienza.

La capacità del sistema di dividere il contenuto generato in due sezioni (il documento e la parte nel chatbot) risulta chiara e poco confusionaria. L'unica incertezza mostrata dal sistema su questo aspetto riguarda la produzione di due documenti dopo il secondo prompt. Esso ha prodotto due documenti pressoché identici, con l'unica differenza data dalla spesa. Sarebbe risultato ottimale e migliore produrre un unico documento.

3.8 Conclusioni sul caso di studio

Abbiamo sottoposto a tutti e quattro i sistemi introdotti precedentemente il nostro caso di studio. Nel complesso, ogni modello ha saputo gestire in maniera adeguata le richieste, ciascuno in maniera differente.

Uno dei punti condivisi dai quattro modelli riguarda l'inserimento di una breve postilla nella quale si consiglia di rivolgersi ad un medico. Questo aspetto risulta particolarmente rilevante in quanto il sistema non si sostituisce all'uomo, bensì valorizza l'analisi umana, ponendola in una posizione superiore nel processo decisionale e per quanto riguarda le conoscenze.

3.8.1 Differenze fra i sistemi utilizzati

In questa sottosezione, evidenzieremo le principali differenze fra i sistemi utilizzati.

Innanzitutto, però, vediamo un aspetto che ha accomunato i sistemi utilizzati. Analizzando le risposte, si nota che gran parte di esse si conclude con una delle domande poste dall'assistente generativo all'utente. Questo comportamento è comune quando si utilizza un chatbot. Tale modo di agire è conseguenza, generalmente, di due principali motivazioni. La prima è l'obiettivo del sistema di mantenere il coinvolgimento dell'utente, invogliandolo a continuare la conversazione. La seconda, invece, potrebbe riguardare un basso livello di confidenza in un dato contesto; allungando la conversazione con domande, il sistema cerca di compensare l'incertezza. Questo fenomeno prende il nome di *Verbosity Compensation*.

Riguardo le differenze, invece, una prima diversità riguarda la struttura delle risposte. Gemini e DeepSeek hanno riportato le diete e le spese all'interno di tabelle, presentando un layout pulito ed organizzato. Claude, invece, ha suddiviso le risposte in due sezioni: un documento testuale contenente l'effettiva risposta ai prompt, e una parte all'interno della chat, contenente consigli e informazioni aggiuntive; anche in questo caso, questa divisione risulta chiara. Il sistema strutturalmente più semplice, invece, è risultato Mistral: il modello francese ha riportato tutto all'interno della conversazione, senza utilizzo di tabelle o suddivisioni di qualche tipo, mostrando, quindi, una risposta strutturalmente più semplice e priva di elementi organizzativi.

Un'altra differenza nelle risposte si nota, in particolare, nella scelta per il giorno della settimana in cui non si può cucinare all'interno della risposta per il quarto prompt. Mistral e DeepSeek hanno scelto il Mercoledì, Claude ha preferito il Giovedì, e Gemini ha optato per il Sabato. A livello logico, Gemini pensa che durante il weekend la persona sia propensa a rilassarsi e, di conseguenza, non voglia cucinare; al contrario, gli altri sistemi scelgono un giorno in mezzo alla settimana per non appesantire ulteriormente una giornata lavorativa. Si tratta, quindi, di due approcci di ragionamento differenti alla base, che evidenziano il modo diverso di ragionare fra i sistemi.

Riguardo le problematiche evinte dal caso di studio, invece, esse si riscontrano prevalentemente all'interno della risposta del quarto prompt. Partiamo analizzando DeepSeek: il sistema dice, ad esempio, di comprare 3€ di olio; tuttavia, in una situazione reale ciò non può avvenire in quanto si può comprare una bottiglia intera, che ha un prezzo più alto. Di conseguenza, le stime dei costi non fanno riferimento ad una situazione realistica, mostrando la difficoltà del sistema a simulare il mondo fisico. Esso, però, è risultato ottimo in altre scelte, come quella di prendere legumi secchi per abbattere i costi.

Mistral presenta, pressochè, lo stesso problema: i prezzi sono incroguenti con una situazione reale a causa delle quantità. Inoltre, il sistema francese basa la dieta principalmente sui carboidrati a discapito delle proteine (per abbassare il costo), cosa che nella realtà creerebbe problemi alla massa muscolare. Infine, seppur non sia un vero e proprio errore, nella lista della spesa sono presenti 500g di farina integrale, i quali vengono utilizzati solo una volta durante la settimana; sarebbe stato più consono utilizzare quella parte di budget in altro modo.

Il sistema che ha presentato più problemi, però, è Claude. Esso, infatti, scrive che il budget è rispettato; tuttavia, sommando le singole stime di costo, si noterà che la cifra spesa è ben più alta. Si tratta, quindi, di un bias di compiacenza: il sistema non fa caso alla correttezza aritmetica di ciò che ha scritto, bensì si sofferma sulla fluidità del testo e sulla soddisfazione dell'utente. Oltre a sbagliare dal punto di vista matematico, Claude, nonostante specifici non sia presente, reinserisce nella spesa e nella dieta il pesce, violando la limitazione imposta precedentemente.

Possiamo a questo punto trarre le conclusioni sul nostro caso di studio, riportando in maniera riassuntiva i risultati all'interno della Tabella 3.1. Il budget fa riferimento a 35€ in quanto i prompt con 50€ sono tutti rispettati. Gli altri parametri, invece, fanno riferimento all'insieme delle altre richieste.

Criterio	Gemini	DeepSeek	Mistral	Claude
Budget (35€)	Ottimo	Buono	Insufficiente	Errato
Diversità alimenti	Media	Media	Elevata	Elevata
Giorno senza cucina	Sabato	Mercoledì	Mercoledì	Giovedì
Accuratezza aritmetica	Elevata	Buona	Scarsa	Fallimentare
Dettagli extra	Ottimi	Funzionali	Generici	Elevati
Comportamento generale	Ottimo	Buono	Superficiale	Ottimo fino al quarto prompt / Inaffidabile

Tabella 3.1: Valutazione riassuntiva dei sistemi testati

Come si evince dalla tabella, possiamo dire che ognuno ha presentato i propri pro e contro, ma, in linea generale, possiamo affermare che Gemini è stato il migliore a livello prestazionale.

All'interno di questo capitolo analizzeremo il comportamento dei sistemi di Intelligenza Artificiale Generativa per la generazione di contenuti video. Introdurremo il funzionamento alla base di tale generazione e i contesti in cui questo tipo di utilizzo si sta sviluppando. Sottoporremo due sistemi, Veo e Runway AI, ad un caso di studio per analizzarne il comportamento, riportando pro e contro. Oltre al caso di studio, riporteremo anche degli ulteriori esempi, per fornire un quadro ancor più chiaro.

4.1 Multimodalità: dal testo ai video

Nelle sezioni precedenti abbiamo introdotto i paradigmi e le regole alla base dell'Intelligenza Artificiale Generativa, e abbiamo analizzato l'andamento di alcuni sistemi nella generazione di dati testuali. In questa sezione, invece, vedremo come si è arrivati alla generazione di contenuti video, soffermandoci su aspetti come l'etica e l'utilizzo odierno.

4.1.1 Il processo di generazione dei video

Alla base della generazione di contenuti video ci sono gli elementi e i meccanismi introdotti nelle sezioni precedenti, come l'architettura Transformer o il Deep Learning. I nuovi sistemi generativi che si occupano di video si basano sui *Trasformatori di diffusione latenti*. I modelli di diffusione (diffusion model) sono già stati introdotti precedentemente; i meccanismi di diffusione in avanti e indietro consentono di creare contenuti di grande qualità, lavorando sul rumore. Tale meccanismo ricostruisce i pixel¹ delle immagini lavorando proprio sul rumore e sui dati di addestramento. Per la generazione di video, i modelli di diffusione dovranno ripulire sequenze di immagini, ossia fotogrammi successivi del video, piuttosto che una singola immagine.

L'utilizzo di questo meccanismo, però, richiede un'enorme quantità di energia; per questo motivo si è passati all'approccio con i *modelli di diffusione latenti*. I numerosissimi pixel di un'immagine in un fotogramma di un video, infatti, sono dati grezzi e, per questo, si è passati ad un approccio che lavora tramite il cosiddetto *spazio latente*. All'interno di tale spazio i fotogrammi video e il prompt di testo vengono compressi tramite un codice di tipo matematico che scarnerà le informazioni superflue e conterrà soltanto le informazioni importanti. La fase conclusiva, invece, prevede la decodifica delle rappresentazioni latenti in

¹Il pixel è la più piccola unità di misura di un'immagine digitale o di un display.

fotogrammi discreti che, una volta assemblati consecutivamente, generano come output un video che rispetti le specifiche dell'utente. Tale architettura, oltre ad ottimizzare l'efficienza computazionale del modello, è alla base della gestione e della distribuzione dei video su Internet.

L'ultimo tassello che si unisce a quelli appena citati è il Transformer. I Transformer, integrati con la diffusione latente, consentono di produrre una sequenza di fotogrammi che sia coerente. Essi elaborano grandi sequenze di dati e permettono di trattare i fotogrammi come se fossero delle parole.

Un ulteriore progresso di particolare rilevanza per la generazione multimediale dei sistemi di ultima generazione riguarda la parte audio. Inizialmente, questo tipo di creazione risultava complicato e difficile da integrare con la parte video. Il team di Google DeepMind, con l'avvento di Veo 3, ha ideato un metodo di compressione di audio e video all'interno di un unico modello di diffusione. In questo modo, si è arrivati ad un livello di produzione video dove la parte vocale risultasse più sincronizzata all'ambiente del video, specialmente per aspetti come la sincronizzazione delle parole con le labbra, o dei rumori in sottofondo.

4.1.2 Uso odierno ed implicazioni etiche

La generazione di contenuti multimediali tramite l'Intelligenza Artificiale Generativa viene utilizzata, oggi, in svariati contesti e applicazioni, grazie, specialmente, alla sua velocità di produzione e alla riduzione dei costi per le aziende.

I principali settori nei quali questo tipo di attività si è più sviluppata sono i seguenti:

- *Marketing e pubblicità*: con i sistemi generativi si possono creare velocemente spot pubblicitari e video promozionali, specialmente per la pubblicità online.
- *E-Commerce*: si possono creare video di prodotti e contenuti interattivi per la vendita online.
- *Supporto alla produzione video*: è possibile, tramite i sistemi generativi, creare effetti visivi e aiutare nella fase di post-produzione di video.
- *Social media*: essendo molti sistemi accessibili a tutti, video generati dall'IA sono sempre più ricorrenti su piattaforme come TikTok o Instagram.

Questo sviluppo continuo nel campo dei video, però, ha portato il mondo a porsi diverse domande e diversi problemi, riguardanti soprattutto l'aspetto etico. Questi quesiti hanno portato, come già accennato precedentemente, alla nascita di regole da rispettare nell'utilizzo di tali contenuti generati.

Un aspetto che ha suscitato particolare scalpore nell'utilizzo di tali tecnologie riguarda la violazione del diritto d'autore e dei dati personali. A riguardo, l'Italia, nel corso del 2025, ha inserito nel codice penale una legge riguardante i deepfake²; tale legge prevede sanzioni per chi pubblica immagini, video o audio contenenti persone o elementi manipolati tramite l'IA senza il consenso dei soggetti o proprietari degli elementi protetti dal diritto d'autore. Tali contenuti, difatti, potrebbero essere utilizzati per ingannare o causare un danno. Per questo motivo, all'interno dei social network, è stato inserito l'obbligo, per gli utenti, di etichettare come tali i contenuti generati con l'Intelligenza Artificiale, in modo tale da non ingannare il pubblico.

Un ulteriore aspetto di fondamentale importanza concerne il processo di addestramento dei sistemi generativi. Quando si allena un sistema, infatti, il modello potrebbe riflettere

²Il termine "deepfake" indica un contenuto digitale creato o modificato tramite l'utilizzo dell'Intelligenza Artificiale.

pregiudizi sociali presenti nei dati con cui si effettua l'addestramento, originando, così, delle potenziali discriminazioni. A tal proposito, anche in questo caso, la fase di training dovrà conformarsi a delle regole specifiche riportate nell'AI Act (Regolamento Europeo sull'Intelligenza Artificiale) emanato dall'Unione Europea nel corso del 2024. Tale normativa impone ai fornitori di pubblicare un riassunto contenente i dati con cui è stato effettuato l'addestramento, e tali dati devono rispettare normative sulla privacy (GDPR) e il diritto d'autore. Inoltre, per i modelli considerati ad alto rischio, la legge prevede protocolli di verifica e degli stress-test, in modo tale da escludere rischi inaccettabili.

4.2 Definizione del caso di studio

Per evidenziare pro e contro della produzione di video tramite i sistemi di Intelligenza Artificiale Generativa abbiamo utilizzato un semplice caso di studio che potesse mettere in risalto gli aspetti positivi e le eventuali limitazioni. Il caso di studio riguarda la creazione di video riguardanti una situazione semplice: un uomo che sta effettuando una camminata in un parco. Il case study sarà suddiviso in quattro prompt, in modo tale da poter aggiungere dettagli ed analizzare situazioni differenti ad ogni richiesta.

I sistemi che abbiamo utilizzato sono due: Runway AI e Veo di Google. In questa maniera sarà possibile confrontare i risultati dei due sistemi, evidenziandone le differenze.

4.2.1 Prompt utilizzati

L'approccio ha presentato delle leggere differenze fra Gemini e Runway. Gemini, infatti, ha memoria della chat ed è possibile inviare dei prompt sequenziali, consentendo di suddividere la richiesta aggiungendo dettagli; diversamente, Runway non ha memoria, e genera ogni volta un contenuto nuovo (a meno che non si utilizzino modalità di modifica foto/video), per questo motivo ogni volta che abbiamo generato un video, abbiamo dovuto ripetere (modificandolo leggermente, laddove fosse necessario) il contenuto dei prompt precedenti per fornire al sistema il giusto contesto. Infine, considerando che Runway è addestrato prevalentemente su dataset in inglese, è stata progettata un'ulteriore richiesta in lingua inglese, con l'obiettivo di vedere quanto il modello fosse sensibile alla lingua.

L'ultimo aspetto riguardante l'input per Runway è relativo all'aggiunta di un'immagine. Il modello utilizzato, difatti, prevede obbligatoriamente l'aggiunta di un'immagine al prompt, in modo tale da fornire più contesto per la generazione. Nel nostro caso, si è scelto di allegare un'immagine ritraente il parco, fornendo, così, lo sfondo per il caso di studio. Essa è stata generata utilizzando *Nano Banana* tramite Gemini, ed è consultabile alla Figura 4.1.

I prompt utilizzati per Veo sono quattro. Il primo riguarda una situazione generale e semplice, per analizzare il comportamento dei sistemi di fronte una situazione in cui hanno piena libertà generativa.

Primo prompt - Situazione semplice

Ciao, fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l'uomo. Fai in modo che l'audio possa essere realistico.

La seconda richiesta prevede l'aggiunta di altre entità all'interno della scena (persone e animali), lasciando libertà sul loro numero. Inoltre, è richiesto che un secondo individuo saluti il soggetto principale, creando, quindi, un'interazione umana sia visiva che sonora. Ciò consente di valutare sia il lato audio, sia la gestione e la coerenza dei movimenti da parte del sistema.



Figura 4.1: Immagine del parco generata da Nano Banana e inviata a Runway

Secondo prompt - Aggiunta di entità e azioni

Ora fai in modo che, oltre all'uomo, nel parco ci siano altre persone e/o animali. Una di queste persone saluta l'uomo, che a sua volta ricambia.

Il terzo comando prevede un'aggiunta di dettagli per l'oggetto tenuto in mano. In particolare, si valuterà la capacità di inserire un testo di senso compiuto, e di lasciarlo coerente con lo scorrere dei fotogrammi. Infatti, l'aggiunta di parole, integrata con un oggetto in movimento, richiede la modifica continua di diversi pixel.

Terzo prompt- Aggiunta di dettagli

Adesso fai che nella valigia siano raffigurare alcune scritte di senso compiuto, e lascia le persone oltre al soggetto principale.

Il quarto ed ultimo prompt, invece, prevede la generazione di un video raffigurante un'azione dinamica, in modo tale da valutare i movimenti e la fluidità nella riproduzione. L'azione scelta è la caduta della valigia, con conseguente raccolta. Tale scelta, consente di valutare la realizzazione della caduta del bagaglio, soggetta a leggi fisiche (la gravità).

Quarto prompt - Azione dinamica

Fai in modo che all'uomo caschi la valigia mentre cammina e si chini a raccoglierla, per poi riprendere la sua camminata. Fai sempre in modo che la telecamera sia dinamica e che il parco dietro di esso sia vivo e colmo di altri soggetti.

I prompt inviati a Runway AI, come già detto, prevederanno l'accorpamento di queste richieste consecutivamente, sostituendo alcuni elementi laddove fosse necessario. Riportiamo soltanto il quarto ed ultimo prompt per semplicità.

Quarto prompt per Runway AI

Fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l'uomo. Fai in modo che oltre all'uomo nel parco ci siano altre persone e animali, rendendo il parco più vivo. Una di queste saluta l'uomo, che a sua volta ricambia. Fai in modo che nella valigia siano raffigurate alcune scritte di senso compiuto. Dopo il saluto, fa in modo che gli caschi la valigia mentre cammina e che si chini a raccoglierla, per poi riprendere la sua camminata. Fa sempre in modo che la telecamera sia dinamica e che il parco dietro di esso sia vivo e colmo di altri soggetti.

Riportiamo anche la richiesta rielaborata in lingua inglese. Il prompt è stato leggermente modificato: stavolta il sistema ha una parola specifica da trascrivere sulla valigia, in modo tale da permettere al sistema di concentrarsi sull'azione e di vedere se, restringendo i gradi di libertà, il modello sia in grado di riprodurre meglio la scritta.

Richiesta in inglese per Runway AI

Cinematic dynamic tracking shot of a man walking through a crowded park. He is carrying a suitcase with the legible word 'TRAVEL' written on it. During the walk, he drops the suitcase; he stops, bends down to pick it up, and resumes walking. Lively background with many passersby and park activities, natural sunlight, photorealistic, high detail.

4.2.2 Veo

Il primo sistema che abbiamo utilizzato per il nostro caso di studio è *Veo*. Veo è il modello generativo sviluppato dal team Google Deepmind, specializzato nella creazione di video partendo da testo, immagini e combinazioni di entrambi. Esso è direttamente accessibile dall'assistente virtuale Gemini, all'indirizzo "gemini.google.com". Specificando che si vuole in output un video, difatti, Gemini invocherà autonomamente Veo, che si occuperà di soddisfare la richiesta.

La versione attuale di Veo è *Veo 3*, la quale ha rappresentato uno strumento fondamentale per l'avanzamento della generazione di contenuti audiovisivi. Tale Versione è la prima che si occupa della produzione simultanea di video e audio assieme. La possibilità di produrre audio nativo rende l'utente capace di dare indicazioni su frasi da dire o musica ed effetti sonori da riprodurre. Questa sincronia fra audio e fotogrammi è la vera e propria innovazione della Versione 3.

Il sistema è in grado di generare contenuti in full HD, simulando fenomeni come luci e ombre, materiali complessi e movimenti, e alcune leggi fisiche basilari, come la gravità. Inoltre, essendo un modello multilingue, una delle sue caratteristiche principali è quella di poter creare audio imitando numerose lingue e accenti.

Assieme a Veo 3, Google ha lanciato un'interfaccia web chiamata *Flow*. Questo strumento è pensato per professionisti, ed offre funzionalità come timeline³, layering⁴ e opzioni avanzate per la personalizzazione.

³La timeline è un'interfaccia che consente di organizzare e modificare video in ordine cronologico.

⁴Il layering è una tecnica che prevede di sovrapporre più tracce video, immagini o clip l'una sull'altra in un unico fotogramma.

Il modello si basa sui Trasformatori di diffusione latenti e su progetti precedenti del team di Google, come *Lumiere*. Ulteriori dettagli e caratteristiche tecniche di Veo sono riportate nel sito "deepmind.google/models/veo/".

Nel nostro caso specifico, la versione a cui si fa riferimento per il caso di studio è Veo 3.1, accessibile tramite il piano Pro.

4.2.3 Runway AI

Il secondo sistema selezionato per il nostro caso di studio è Runway AI. Tale sistema di Intelligenza Artificiale Generativa è specializzato nella produzione di immagini, video e contenuti multimediali, ed è stato pubblicato dall'omonima startup statunitense nel 2018. Sebbene il software sia, prevalentemente, a pagamento, esso offre dei crediti gratuiti per la generazione di alcuni video, previa iscrizione con una mail. In questo modo, Runway permette agli utenti di avere un primo approccio col proprio sistema generativo, consentendo di esplorarne le potenzialità prima di sottoscrivere un abbonamento.

Dal suo rilascio il sistema è stato soggetto a continui aggiornamenti, portando allo sviluppo di diverse versioni. Gli ultimi modelli sviluppati sono *Gen-4* e *Gen-4.5*, successori di *Gen-2* e *Gen-3*. Il sistema da noi utilizzato è *Gen-4 Turbo*, in quanto è l'unico accessibile se non si possiede un piano a pagamento.

Gen-4 Turbo è un modello di ultima generazione per la creazione di contenuti multimediali. Esso eccelle nel trasformare un'immagine statica in un video dinamico, motivo per cui gli input prevedono un utilizzo combinato di foto e testi, risultando, quindi, un sistema multimodale. Fra le sue caratteristiche principali troviamo la capacità di emulare leggi fisiche reali, seppur in maniera limitata, la possibilità di gestire movimenti di macchina (come lo zoom) e creare animazioni stilizzate, oltre che un'alta fedeltà e coerenza.

Il modello è accessibile dal web all'indirizzo "app.runwayml.com". Accedendo al sito, si può notare come siano presenti diversi strumenti e funzionalità per l'editing, la post-produzione e il controllo. Fra questi strumenti troviamo, ad esempio:

- *Motion Brush*: permette di colorare un'area dell'immagine e dire all'IA di muovere solo quella, lasciando inalterato il resto.
- *Camera control*: si possono controllare, come già accennato, azioni come lo zoom o il pan (movimento orizzontale di macchina).
- *Editing*: si possono eliminare elementi dal video, applicare effetti e filtri, creare animazioni, e così via.

Ovviamente, tutte queste funzionalità sono messe a disposizione soltanto per chi sottoscrive un abbonamento, evidenziando la natura SaaS⁵ di Runway.

Per quanto riguarda i dati tecnici prodotti per il nostro caso di studio con *Gen-4 Turbo*, essi hanno durata di 5/10 secondi, rapporto 16:9 e risoluzione 1280x720.

4.2.4 Obiettivi del caso di studio

Gli obiettivi del caso di studio sono molteplici. In primis, si valuterà l'efficacia generativa dei sistemi di fronte a diverse richieste di difficoltà, consentendo di analizzare il comportamento di Veo e *Gen-4* di fronte ad alcuni scenari critici, come i movimenti umani o

⁵L'acronimo SaaS sta per "Software as a Service", in italiano "Software come servizio".

l’inserimento di scritte sensate. I criteri valutativi secondo cui daremo dei giudizi riguarderanno parametri come la riproduzione e la fluidità dei movimenti, le proporzioni, la corretta interpretazione delle richieste, il lato audio (laddove sia presente), la coerenza temporale ⁶.

Inoltre, una volta sottoposti entrambi i modelli al caso di studio, sarà possibile evidenziare le differenze fra i due, riportando i punti di forza e di debolezza di entrambi.

In aggiunta, per quanto riguarda Runway, visto il prompt in doppia lingua (italiano e inglese), si valuterà come la soluzione proposta cambia, consentendo di analizzare quanto il sistema sia sensibile al linguaggio in cui si pone la richiesta.

4.3 Caso di studio

Per semplicità verranno riportati soltanto i fotogrammi principali e significativi per ogni prompt, mentre i video interi saranno accessibili tramite il QR code della Figura 4.2, collegato ad una cartella esterna su Google Drive. Al suo interno saranno presenti due cartelle, una relativa al case study per Veo ed una per Runway.



Figura 4.2: Il QR code per accedere ai video relativi al nostro case study

4.3.1 Analisi con Veo

Cominciamo analizzando l’andamento di Veo di fronte al primo prompt.

Utente

Ciao, fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l’uomo. Fai in modo che l’audio possa essere realistico.

⁶Per "coerenza temporale" si intende la capacità di mantenere gli oggetti uguali nel tempo senza mutarli o eliminarli.

Gemini - Veo



Figura 4.3: Uno dei frame iniziali del primo video di Veo



Figura 4.4: Uno dei frame finali del primo video di Veo

Come emerge dai fotogrammi e dal video, Veo ha soddisfatto la richiesta. I movimenti dell'uomo risultano buoni e abbastanza fluidi, e la telecamera segue l'uomo correttamente, come richiesto. Luci ed ombre sono rappresentate bene, con degli ottimi contrasti. L'unico dubbio riguarda l'audio scelto: Veo genera un sottofondo musicale tranquillo e armonioso, piuttosto che un audio ambientale realistico, come specificato nel prompt.

Vediamo ora cosa accade per il secondo prompt.

Utente

Ora fai in modo che, oltre all'uomo, nel parco ci siano altre persone e/o animali. Una di queste persone saluta l'uomo, che a sua volta ricambia.

Gemini - Veo



Figura 4.5: Uno dei frame iniziali del secondo video di Veo



Figura 4.6: Uno dei frame centrali del secondo video di Veo



Figura 4.7: Uno dei frame finali del secondo video di Veo

Come si nota dal video e dai frame delle Figure 4.6 e 4.7, è presente un problema comune nella generazione di video: l'apparizione improvvisa di un passante alle spalle del soggetto in primo piano. Questo tipo di errore è uno dei più comuni. Ciò accade in quanto l'Intelligenza Artificiale genera il video occupando inizialmente quella porzione di spazio con l'uomo

che ha la valigia; camminando, però, l'uomo si sposta dalla posizione iniziale, e il sistema è costretto a reinterpretare e generare il nuovo spazio come meglio crede. Visto che nella richiesta era specificato di rendere vivo il parco, Veo ha pensato che fosse il punto giusto dove inserire una persona.

Per quanto riguarda l'aspetto acustico, stavolta viene emulato correttamente un saluto. Tuttavia, dopo lo scambio fra i due individui, senza nesso logico, il sistema sceglie di far girare entrambe le persone, senza far continuare la loro camminata in modo naturale.

Vediamo ora cosa accade con il terzo prompt.

Utente

Adesso fai che nella valigia siano raffigurate alcune scritte di senso compiuto, e lascia le persone oltre al soggetto principale.

Gemini - Veo



Figura 4.8: Frame iniziale del terzo video prodotto da Veo



Figura 4.9: Frame successivo alla caduta della valigia

Guardando il video e i frame, viene subito all'occhio un primo problema: il sistema ha scelto di aggiungere due valigie, non ricordando i prompt precedenti. Ciò, probabilmente, è dovuto al fatto che il modello, cercando di aggiungere dettagli, pensi che una seconda valigia possa essere la scelta giusta.

Il secondo problema, di natura ben più critica rispetto al precedente, riguarda la scomparsa della valigia nella mano destra. Si evince, quindi, un elevato grado di incoerenza temporale: nel passaggio da un frame all'altro, il sistema trova che la valigia non sia più necessaria, e la fa cadere dalla mano dell'uomo. Dopodiché, visto che la telecamera avanza, si dovrebbe comunque notare la valigia caduta, ma ciò non accade. Questo a dimostrazione del fatto che il sistema ha cancellato a tutti gli effetti la valigia dal video, mostrando di non riuscire a garantire la continuità dell'oggetto. Tale anomalia è molto comune nella generazione dei video, specialmente per azioni che riguardano l'impiego di oggetti tenuti in mano.

Si osserva, inoltre, che la donna effettua il gesto del saluto due volte, con entrambe le mani. Tale ridondanza mostra le limitazioni del sistema nel riprodurre azioni e comportamenti umani in modo logico e naturale, creando scene poco realistiche.

Analizzando i lati positivi, invece, possiamo notare un'ottima coerenza nel background, con persone e sagome visibili in lontananza. Per contro, l'aspetto sonoro, senza alcun motivo particolare, non è stato generato, lasciando il video privo di suoni.

Riportiamo ora i frame principali per il quarto ed ultimo prompt proposto a Veo.

Utente

Fai in modo che all'uomo caschi la valigia mentre cammina e che si chini a raccoglierla, per poi riprendere la sua camminata. Fai sempre in modo che la telecamera sia dinamica e che il parco dietro di esso sia vivo e colmo di altri soggetti.

Gemini - Veo



Figura 4.10: Frame del quarto video in cui accade l'anomalia

Come si nota dal video e dal frame della Figura 4.10, il sistema non ha riprodotto correttamente l'azione richiesta. La valigia, piuttosto che cadere, viene accompagnata in basso dall'uomo, senza che si stacchi mai dalla presa; dopodiché, l'uomo si accinge a prenderla con l'altra mano per poi ricominciare la camminata. Si evince, quindi, un problema nel generare l'azione e la fisica dietro la caduta.

Osservando ancora la Figura 4.10, emerge un'altra criticità di particolare rilevanza e comune nella produzione di immagini e video: l'incoerenza nell'interazione mano-oggetto. Si nota, infatti, che nel momento in cui la valigia è a terra, la mano compenetra nell'oggetto e si deforma, prima di tornare normale. Alla base di questa tipologia di errore c'è un'enorme complessità anatomica e variabilità per i sistemi generativi. Le mani, infatti, possono assumere un numero infinito di posizioni, gesti ed angoli, rendendo estremamente complicata per i modelli la loro interpretazione, a causa degli elevati gradi di libertà.

Gli ultimi due aspetti rilevanti riguardano la valigia e il background. Relativamente alla valigia, notiamo come la parola sia scritta in spagnolo, nonostante il prompt fosse scritto in lingua italiana. Per quanto concerne il parco, invece, esso risulta pieno di vita; tuttavia, se si guardano bene i soggetti, si noterà che ci sono molte ripetizioni. Ad esempio, si possono notare diversi esemplari di pastore tedesco. Questo fenomeno prende il nome di *Semantic Leakage* (*Perdita semantica*). Il sistema impiega le sue risorse per generare un elemento e, così facendo, l'elemento generato rimane attivo nella memoria a breve termine; per questo motivo, il modello tende a riutilizzare un elemento che ha generato in precedenza, piuttosto che allocare nuove risorse per produrne sempre di nuovi.

4.3.2 Analisi con Runway

In questa sezione riportiamo limiti e caratteristiche emerse dall'utilizzo del modello Gen-4 Turbo di Runway. Come già detto precedentemente, la generazione di audio non è inclusa per questo modello, motivo per cui non sarà presa in considerazione.

Utente

Fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l'uomo.

Gen-4 Turbo



Figura 4.11: Frame centrale del primo video generato con Gen-4 Turbo

Il video è coerente con l'immagine inviata e la richiesta; tuttavia, i movimenti dell'uomo risultano veloci e poco fluidi. Il parco e gli alberi nello sfondo risultano molto statici.

Vediamo come la situazione cambia con il secondo prompt.

Utente

Fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l'uomo. Fai in modo che oltre all'uomo, nel parco ci siano altre persone e/o animali. Una di queste saluta l'uomo, che a sua volta ricambia.

Gen-4 Turbo

**Figura 4.12:** Frame del secondo video generato con Gen-4 Turbo

Analizzando la Figura 4.12, si osserva come il saluto fra i due uomini risulti del tutto irrealistico. I movimenti sono, nuovamente, veloci e poco fluidi, e si notano prospettive e proporzioni non ottimali. In aggiunta, guardando la mano dell'uomo sulla sinistra, si osserva come essa sia stata generata con sei dita. Tale problema, già riscontrato con Veo, conferma la difficoltà dei modelli nel rappresentare l'anatomia delle mani ed i loro movimenti.

Vediamo ora cosa accade aggiungendo ulteriori dettagli.

Utente

Fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l'uomo. Ora fai in modo che oltre all'uomo, nel parco ci siano altre persone e animali, rendendo il parco più vivo. Fai che nella valigia siano raffigurate alcune scritte di senso compiuto.

Gen-4 Turbo

**Figura 4.13:** Frame del terzo video generato con Gen-4 Turbo

Guardando il video e la Figura 4.13, si nota subito un enorme errore da parte del modello. La valigia, infatti, è stata sostituita da un oggetto ambiguo (riconducibile, similmente, ad

un foglio) con delle scritte indecifrabili al suo interno. Da questo video, quindi, si evince la difficoltà dei sistemi nell’inserire del testo scritto in oggetti in movimento. Tale difficoltà è dovuta sia all’elevato numero di pixel da dover modificare ad ogni fotogramma, sia alla necessità di sincronizzare i movimenti con le parole, cercando di ottenere una buona coerenza. Il sistema, probabilmente per inserire le scritte, ha scelto di modificare la valigia e renderla simile ad un foglio. Tale decisione, però, risulta fallimentare, poiché determina un’incoerenza semantica rispetto alla richiesta, la quale chiedeva esplicitamente una valigia.

Oltre al problema appena citato, il sistema si è limitato ad aggiungere una sola persona, continuando a lasciare il parco in background molto statico.

Utente

Fai un video di un uomo che cammina lungo un sentiero in un parco, di giorno, con una valigia in mano. La telecamera deve essere dinamica e seguire l’uomo. Fai in modo che oltre all’uomo nel parco ci siano altre persone e animali, rendendo il parco più vivo. Una di queste saluta l’uomo, che a sua volta ricambia. Fai che nella valigia siano raffigurate alcune scritte di senso compiuto. Dopo il saluto, fai in modo che gli caschi la valigia mentre cammina e che si chini a raccoglierla, per poi riprendere la sua camminata. Fai sempre in modo che la telecamera sia dinamica e che il parco dietro di esso sia vivo e colmo di altri soggetti.

Gen-4 Turbo



Figura 4.14: Uno dei frame iniziali del quarto video



Figura 4.15: Frame centrale in cui si presenta un'anomalia nei movimenti



Figura 4.16: Frame che mostra la caduta della valigia

Nella soluzione al nostro quarto ed ultimo prompt, sono ben visibili diversi problemi. In primis, come già accadeva per il prompt precedente, si può notare che la valigia viene sostituita da un oggetto non identificabile, anche in questo caso simile ad un foglio con al suo interno delle scritte non leggibili. Ciò sottolinea, nuovamente, un'incoerenza semantica e una difficoltà nel rappresentare oggetti con parole al loro interno.

La seconda problematica riguarda, invece, la comparsa ambigua di figure umane. Guardando il frame in Figura 4.14, si vedono due soggetti uscire dall'albero. Il sistema non sa che le persone dovrebbero esistere da prima, e, quindi, sfrutta lo spazio nascosto dietro l'albero come una sorta di punto di generazione.

Dal frame in Figura 4.15, invece, si evince un grande problema nel riprodurre i movimenti umani. Si osserva, infatti, che l'uomo ha il volto rivolto verso una direzione, mentre le mani che reggono l'oggetto sono disposte nel senso opposto, verso la telecamera. Tale postura è anatomicamente impossibile e irrealistica, e mostra come il modello non sia in grado di simulare movimenti (come le rotazioni o le torsioni) che rispettino il funzionamento del corpo umano.

Il fotogramma riportato in Figura 4.16 conferma, come riscontrato anche con Veo, una difficoltà nel simulare correttamente la fisica della caduta. Durante questa fase, si assiste, inoltre, alla comparsa di un nuovo oggetto (un sacchetto) nella mano sinistra del soggetto. Tale fenomeno indica che il sistema, cercando di generare la caduta, finisce per creare un elemento non richiesto dal prompt, aggravando l'incoerenza temporale.

Vediamo, ora, come la situazione cambia ponendo la richiesta in lingua inglese.

Utente

Cinematic dynamic tracking shot of a man walking through a crowded park. He is carrying a suitcase with the legible word 'TRAVEL' written on it. During the walk, he drops the suitcase; he stops, bends down to pick it up, and resumes walking. Lively background with many passersby and park activities, natural sunlight, photorealistic, high detail.

Gen-4 Turbo

**Figura 4.17:** Frame significativo del video in inglese

Ponendo la richiesta in inglese, il sistema migliora rispetto alla richiesta precedente: la valigia è rappresentata correttamente e la scritta (specificando la parola) è riportata in modo esatto.

Sussistono, tuttavia, alcune irregolarità. Come evidenziato nella Figura 4.17, si denota come compaiano due mani per l'arto sinistro, in due posizioni differenti. Ciò deriva dal fatto che il sistema debba generare due movimenti distinti e consecutivi (la caduta della valigia e il chinarsi a prenderla); non sapendo quale frame generare prima, il modello fallisce nella sequenzialità dei due, creandoli contemporaneamente e dando vita ad un'incongruenza nei movimenti. Il sistema, allora, si trova, anche se per pochissimi istanti, a dover gestire due arti differenti; ciò porta il modello a ricongiungere i due arti in maniera del tutto innaturale.

Oltre a quanto visto, si riscontrano degli istanti in cui si verifica una compenetrazione fra la mano che tiene la valigia, la maniglia di quest'ultima e il corpo dell'uomo. Questo fenomeno evidenzia l'incapacità del sistema di gestire volumi e forme dei corpi: i pixel degli elementi vengono fusi in un unico insieme incoerente, per poi tornare normali. Tale problema si riscontra usualmente, ad esempio, proprio per oggetti tenuti in mano.

4.4 Esempi aggiuntivi

All'interno di tale sezione forniremo alcuni brevi esempi aggiuntivi, generati con Veo e Sora. In questo modo sarà possibile dare una panoramica su altri aspetti non analizzati con il caso di studio. I video di questa sezione sono accessibili all'interno dell'apposita cartella del drive, scansionando il QR code precedente, consultabile alla Figura 4.2.

4.4.1 Altri esempi con Veo

Partiamo analizzando altri esempi prodotti con Veo. Il primo che presentiamo è il seguente.

Utente

Genera il video di due uomini che si scambiano delle parole al bar, per poi salutarsi con una stretta di mano. Fai che l'audio rispecchi tale situazione e che lo stile sia il più realistico possibile.

Gemini - Veo

Figura 4.18: Confronto tra il frame iniziale (sinistra) e l'anomalia (destra).

È stato progettato uno scenario in cui poter valutare il lato audio e la resa della stretta di mano, che risulta, solitamente, difficoltosa per i modelli generativi. Il lato audio è consona alla richiesta, e viene simulata bene l'interazione fra i due uomini, nonostante si noti che sia le voci sia i movimenti risultino lievemente macchinosi.

L'azione della stretta di mano è rappresentata correttamente; tuttavia, come si evince guardando la Figura 4.18, nei frame iniziali l'uomo sulla sinistra tiene in mano una birra, che verrà posata sul tavolo per stringere la mano all'altro individuo. Quando le mani si avvicinano, però, il bicchiere scompare di colpo. Il sistema, trovandolo "in mezzo", lo elimina per raffigurare la stretta di mano. Tale errore mostra un deficit del sistema nel comprendere la permanenza fisica degli oggetti.

Vediamo ora un secondo esempio.

Utente

Fai il video di un uomo che mangia un panino. Simula bene l'azione del morso e la masticata. Fa che lo stile sia il più realistico possibile e che la telecamera sia ferma ad inquadrare la scena.

Gemini - Veo

**Figura 4.19:** Frame centrale del video

È stato scelto un prompt di questo genere per valutare il comportamento del sistema di fronte ad un'azione che alteri la forma di un corpo. Come dimostrato analizzando la Figura 4.19, il sistema ha gestito in modo ottimale e accurato la richiesta. L'azione del morso è coerente, e la forma del panino è stata modificata in maniera corretta. Da un punto di vista qualitativo, questo video è il migliore visto finora.

Utente

Fai il video di un ragazzo che palleggia sempre più velocemente. Fai inoltre che palleggi in modo particolare, per esempio facendo un giro con la gamba attorno al pallone di tanto in tanto. Per l'audio, fai che sia consona ad un campo di allenamento.

Gemini - Veo

**Figura 4.20:** Frame in cui c'è l'anomalia nell'azione

Come si può notare dal frame riportato in Figura 4.20, il sistema non ha saputo simulare correttamente il movimento richiesto, sottolineando una difficoltà nel riprodurre i movimenti

delle gambe (seguendo i ragionamenti visti per la medesima situazione con le mani). La palla, infatti, per un secondo rimbalza in aria, prima di tornare sui piedi del giocatore. Si registra, inoltre, una difficoltà nella gestione del ritmo: la velocità del palleggio, che doveva aumentare, rimane invariata, mostrando difficoltà nel modificare la cadenza e la frequenza di un'azione in corso.

4.4.2 Esempi con Sora

In questa sezione vedremo altri esempi, stavolta sviluppati dal sistema *Sora*. Sora è il modello generativo multimediale sviluppato da OpenAI, rilasciato nel corso del 2024. Le caratteristiche del sistema, sono consultabili nel sito di OpenAI, nella sezione dedicata al modello, denominata "*Video generation models as world simulators*".

Il successo di Sora ha condotto OpenAI a sviluppare una versione potenziata: Sora 2. Essa forniva funzionalità aggiuntive come la possibilità di produrre la parte audio, maggiori strumenti di editing e un più alto grado di coerenza semantica e potenza. Tale versione, tuttavia, è rimasta accessibile via web soltanto negli USA e in Canada.

Il sistema era accessibile all'indirizzo "*sora.chatgpt.com*"; tuttavia, l'accesso e lo sviluppo del modello (e di Sora 2) sono terminati ufficialmente il 26 Aprile 2026. La scelta dietro la chiusura dei modelli multimediali di OpenAI riguarda, prevalentemente, motivi economici e scelte aziendali.

Di seguito, verranno forniti alcuni esempi generati con Sora, per fornire una visione su uno dei sistemi multimediali più influenti finora. Il primo esempio proposto prevede una situazione in cui il sistema dovrà simulare un fenomeno atmosferico come la pioggia.

Utente

Voglio un video di una persona (scegli tu il sesso e l'aspetto fisico) che sta effettuando un servizio meteorologico in una città. Il tempo dietro di essa è brutto: c'è un temporale e sta piovendo. La città è viva, ci sono auto che si muovono e il vento tira. Fai in modo che lo stile del video sia il più realistico possibile e possa sembrare reale.

Sora



Figura 4.21: Primo esempio con Sora

Dal risultato, riportato in Figura 4.21, osserviamo che la pioggia e il temporale sono assenti. La difficoltà nel simulare tali azioni è riconducibile al fatto che per rappresentare

la pioggia, l'Intelligenza Artificiale deve cambiare molti pixel (relativi alle gocce) ad ogni fotogramma, aggiungendo anche leggi fisiche come il vento e la gravità delle gocce. Tali difficoltà, probabilmente, hanno portato il sistema a non rappresentare il fenomeno, per concentrarsi sul resto.

Differentemente, gli altri aspetti del video risultano di ottima qualità. La resa delle luci, e dell'atmosfera che ne consegue, è ottima e realistica. Anche l'espressività della donna e i movimenti sono adeguati alla situazione. Tali elementi risaltano la capacità di Sora di eccellere nella realizzazione di individui e scenari realistici, laddove non intervengano fenomeni complessi (il temporale).

Vediamo ora altri esempi con Sora, sviluppati in maniera leggermente diversa.

Questa volta, l'approccio prevede l'utilizzo dei preset⁷ messi a disposizione da Sora. Il primo (nella Figura 4.22, a sinistra) è il preset *Film Noir*. Esso simula uno stile in bianco e nero, con forti contrasti ed ombre.



Figura 4.22: Esempi sviluppati con alcuni preset di Sora.

Il prompt utilizzato per la produzione del video è il seguente:

Richiesta per il preset Film Noir

Un detective con giacca e cappello da detective, in un'ambientazione buia, accende un fiammifero. La luce del fiammifero deve illuminare improvvisamente il suo volto.

Si nota chiaramente il problema relativo all'accensione del fiammifero: esso viene acceso di colpo, senza l'esecuzione dei passaggi logici che portano all'accensione. Il modello, quindi, ha preso alla lettera l'azione richiesta dell'utente, senza rappresentare il reale processo fisico. Nonostante il limite fisico espresso, si riconosce l'efficacia del preset utilizzato. L'atmosfera generata è assolutamente in linea con le aspettative, con contrasti di ombre e luci ottimi. Anche l'uomo è inserito perfettamente nel contesto, sia per quanto riguarda l'aspetto, sia per le espressioni facciali. Si può, quindi, affermare che lo stile cinematografico *Noir* è stato riprodotto fedelmente.

Nella parte destra della Figura 4.22, invece, troviamo un esempio sviluppato con il preset chiamato *Stile Cartoonify*. Tale set riproduce uno stile animato e cartonesco. Il prompt utilizzato è stato il seguente:

⁷Un preset è una sorta di filtro preimpostato che applica uno stile visivo, un'illuminazione o un'atmosfera specifica al video generato, senza doverlo specificare nel prompt.

Richiesta per il preset Cartoonify

Crea un video in stile 3D di un ragazzo che si trova in un parco a camminare su un sentiero. Fai in modo che ci sia gente dietro di lui, ognuna a fare un qualche tipo di attività.

Analizzando la richiesta, si evince che il sistema non ha rappresentato gli altri individui richiesti, soffermandosi sul ragazzo in primo piano. Lo stile rispecchia adeguatamente la richiesta e il preset utilizzato.

Altri preset messi a disposizione da Sora erano, ad esempio:

- *PaperCraft*: specializzato nella manipolazione di carta.
- *Balloon World*: stile in cui elementi e personaggi assumono un aspetto gonfiabile, con superfici lisce e colori vivaci.
- *Pixel Art*: caratterizzato da pixel evidenti, forme semplificate e palette cromatiche limitate, ispirato ai videogiochi retro.
- *Archival*: progettato per generare video che simulino l'aspetto di vecchi filmati d'archivio o storici.

In conclusione, quindi, possiamo affermare che i preset di Sora erano strumenti estetici e versatili, capaci di fornire una vasta libertà creativa nella produzione video.

4.5 Conclusioni sul caso di studio

Grazie al nostro caso di studio e ad alcuni esempi aggiuntivi, sono state analizzate diverse caratteristiche e problematiche comuni nella generazione di video da parte dei sistemi di IA.

Dall'analisi comparativa dei video prodotti da Veo e Runway, emerge come, in linea generale, i due modelli abbiano presentato pressoché le stesse limitazioni, seppur generate in maniera differente. Una sintesi sull'andamento dei due sistemi è riportata nella Tabella 4.1.

Criterio	Veo 3	Runway Gen-4 Turbo
Qualità video	Ottima	Buona
Coerenza temporale e spaziale	Buona, non rispettata una sola volta	Insufficiente
Gestione dei movimenti	Ottima, movimenti fluidi	Movimenti rapidi e poco fluidi, talvolta anche errati
Correttezza semantica	Buona	Errata
Proporzioni	Ottime	Sufficienti
Audio	Buono	Assente
Comportamento generale	Ottimo	Insufficiente

Tabella 4.1: Tabella riassuntiva dei sistemi testati.

Nel complesso, si può affermare che il confronto fra i due modelli evidenzia una maggiore efficacia di Veo rispetto a Gen-4 Turbo. Il sistema di Runway, infatti, ha mostrato diverse lacune, in particolare nella riproduzione di movimenti e nella gestione della coerenza temporale. Si osserva, inoltre, come la miglior risposta del modello di Runway sia stata quella relativa alla richiesta posta in inglese: questo mostra come il sistema interpreti meglio l'inglese rispetto all'italiano, riflettendo un'elevata sensibilità al linguaggio di addestramento.

In conclusione, si può affermare che l'utilizzo dell'Intelligenza Artificiale nella produzione video rappresenta uno strumento di elevata potenza e dalle notevoli potenzialità, capace di

generare delle buone risposte in breve tempo e di snellire il carico di lavoro umano. Tuttavia, come si evince analizzando i video prodotti, tale applicazione presenta ancora diversi limiti. Risulta, quindi, plausibile ritenere che l'IA, in questo contesto, attualmente, possa essere più uno strumento di supporto che una tecnologia sostitutiva del lavoro umano.

In questo capitolo sottoporremo tre sistemi di Intelligenza Artificiale Generativa ad un caso di studio per testarne le potenzialità nella generazione di software. L'obiettivo è analizzare lati positivi e negativi degli output prodotti, effettuando una valutazione sia per il codice sia per il software eseguibile, secondo diversi parametri. A conclusione del capitolo si fornirà una sintesi sui principali aspetti emersi dal test di studio.

5.1 Definizione del caso di studio

Nei capitoli precedenti è stato analizzato l'andamento dei sistemi generativi sottoposti a casi di studio riguardanti la produzione testuale e multimediale. In questo capitolo, invece, si tratterà un case study relativo alla generazione di codice informatico.

Questa specifica applicazione dei sistemi di IA, come quelle viste finora, ha registrato uno sviluppo esponenziale negli anni, grazie all'evolversi delle architetture generative. Attualmente, con l'avvento dei chatbot, è possibile ricevere in output interi applicativi software eseguibili. I dati di addestramento, infatti, comprendono elevate conoscenze nell'ambito della programmazione, fornendo, quindi, la capacità di scrivere codice in centinaia di linguaggi di programmazione, di scripting¹ e di markup².

5.1.1 Contesto applicativo e obiettivi

Per il caso di studio si è optato per la generazione di un convertitore di unità di misura. Lo sviluppo seguirà un approccio sequenziale. Inizialmente, si partirà con un convertitore di temperatura (gradi Celsius, Kelvin, Fahrenheit), per poi integrare altre grandezze fisiche. Questa semplice casistica consente di valutare il codice sotto diverse caratteristiche.

Gli obiettivi del caso riguardano, prevalentemente, la valutazione dei risultati forniti dai sistemi, secondo differenti criteri. Si valuterà il livello di efficacia e qualità delle soluzioni proposte secondo un duplice livello: da un lato il codice sorgente, dall'altro l'usabilità e le caratteristiche del programma eseguibile.

Infine, visto l'utilizzo di molteplici sistemi di IA, si effettuerà un'analisi comparativa fra di essi, al fine di poter stilare una classificazione in base alle prestazioni.

¹I linguaggi di scripting sono una categoria di linguaggi di programmazione utilizzati per automatizzare compiti ripetitivi o attività di controllo.

²Un linguaggio di markup definisce la struttura e l'organizzazione di dati e testi all'interno di un documento, senza logica di programmazione.

5.1.2 Criteri di valutazione

Le soluzioni proposte dai modelli sono state sottoposte ad una valutazione basata su diversi criteri legati allo sviluppo software. Di seguito si illustrano i parametri di indagine utilizzati:

- *Correttezza e Funzionalità*: valutazione della capacità del sistema di produrre codice eseguibile e privo di errori, garantendo aderenza con le specifiche imposte dall'utente.
- *Qualità del codice*: si analizza il livello qualitativo del codice, con particolare attenzione ad elementi come la leggibilità, la manutenibilità e l'inserimento di commenti. Un codice di alta qualità faciliterebbe la comprensione anche a chi non ha conoscenze tecniche, rendendolo accessibile e semplificando futuri interventi.
- *Efficienza e Ottimizzazione*: il software deve funzionare bene. Si analizzerà il codice al fine di identificare eventuali ridondanze o porzioni superflue e sostituibili.
- *Layout grafico dell'interfaccia*: si fornirà un feedback sull'interfaccia utente del programma (GUI³). Tale valutazione comprenderà aspetti quali la pulizia della finestra, la corretta organizzazione e disposizione visiva degli elementi e la loro usabilità, oltre che lo stile scelto dal sistema.

5.2 Metodologia e approccio

All'interno di questa sezione verrà illustrato, in maniera più tecnica e dettagliata, l'approccio adottato per il caso di studio. Esporremo le decisioni scelte, riguardanti i sistemi e il linguaggio di programmazione selezionati, oltre che la presentazione dei prompt usati nelle fasi di test.

5.2.1 Sistemi utilizzati

I sistemi utilizzati per questa categoria di output sono tre, e si tratta di tutti sistemi già visti e presentati in precedenza: Gemini, DeepSeek e Claude.

Gemini, specialmente nella sua versione Pro (quella utilizzata in questo studio), integra *Alphacode 2*, l'avanzato sistema di generazione di codice sviluppato da Google Deepmind. Questo modello offre notevoli vantaggi per il coding⁴ tra cui un contesto esteso, un sistema di debugging avanzato in grado di rilevare sia errori sintattici sia logici, e la generazione di spiegazioni dettagliate e commenti per le proprie scelte. Il modello di Google supporta più di 20 linguaggi di programmazione, fra cui figurano Python, Java, C++ e JavaScript.

DeepSeek, ed in particolare il suo ultimo modello (*DeepSeek V4*), si è imposto come uno dei leader dei sistemi di IA Generativa nell'ambito del coding. Le ultime versioni del modello cinese, infatti, sono state addestrate con particolare focus sugli archivi di Github e su diversi set di dati riguardanti la programmazione. Ciò rende tali sistemi particolarmente efficienti nel risolvere bug⁵ complessi, nello scrivere codice che emuli il ragionamento di programmatori esperti e nell'implementazione di librerie moderne per alcuni linguaggi (come Python, ad esempio).

Per quanto riguarda Claude, esso si distingue per le sue capacità di ragionamento elevate. Questa sua caratteristica consente, in generale, di minimizzare l'insorgenza di bug logici. A livello informatico, il modello punta molto sulla leggibilità: i codici generati dal modello di

³L'acronimo GUI sta per Graphical User Interface, in italiano, Interfaccia Grafica Utente.

⁴La parola coding indica la scrittura di istruzioni informatiche.

⁵Un bug, in informatica, è un difetto software di tipo funzionale.

Anthropic risultano, di norma, molto chiari e semplici da interpretare. Grazie alle elevate capacità di ragionamento, inoltre, Claude risulta ottimo per le fasi di debug e revisione.

5.2.2 Scelta del linguaggio: Python

Per il nostro caso di studio, il linguaggio di programmazione scelto è *Python*. Creato nel 1991, si tratta di un linguaggio di programmazione ad alto livello che risulta uno dei linguaggi informatici più diffusi attualmente. La selezione di questo linguaggio per testare i modelli generativi è basata su diversi fattori chiave:

- *Sintassi e leggibilità*: Python è un linguaggio che si distingue per una sintassi molto pulita a livello visivo, vista l'indentazione obbligatoria. Tale caratteristica agevola la valutazione qualitativa del codice prodotto. Inoltre, essendo un linguaggio interpretato (ossia che non richiede compilazione), semplifica e velocizza la fase di esecuzione.
- *Natura open source e versatilità*: grazie alla sua natura open source, Python vanta un immenso numero di librerie integrabili, il che lo rende efficiente per una molteplicità di applicazioni.
- *Predominanza nel settore dell'IA*: Python rappresenta lo standard nel campo dell'Intelligenza Artificiale per due ragioni fondamentali. In primo luogo, la sua diffusione e la sua versatilità hanno portato ad un'enorme presenza (maggiore rispetto ad altri linguaggi) nei dati di addestramento di molti modelli, rendendolo il linguaggio più documentato. In secondo luogo, molti sistemi generativi sono stati sviluppati proprio utilizzando questo linguaggio.

5.2.3 Definizione delle richieste

Il nostro caso di studio, come già avvenuto per la generazione di dati testuali e multimediali, è strutturato su più richieste. Questo approccio sequenziale consente di analizzare l'andamento del sistema in modo graduale, aggiungendo funzionalità e aumentando la difficoltà man mano che il caso di studio avanza.

In questa sezione introdurremo i prompt progettati per il test, giustificandone le scelte alla base. Partiamo analizzando il primo prompt.

Primo prompt - Convertitore di temperatura

Crea un programma in Python che funga da convertitore per le unità di misura di temperatura (Celsius, Kelvin, Fahrenheit) con un'interfaccia grafica. Puoi utilizzare librerie grafiche come Tkinter o PyQt/PySide, scegli tu quella che reputi migliore. Dovranno esserci dei pulsanti per quel che riguarda l'inserimento dei valori, un display che mostri ciò che l'utente scrive e poi il relativo risultato. Aggiungi tutto ciò che reputi necessario per soddisfare questo compito, mantenendo un layout consono e pratico.

Con questo primo comando è stato richiesto ai sistemi di sviluppare un'applicazione che funga da convertitore per le grandezze termiche. Le direttive imposte ai sistemi riguardano prevalentemente l'aspetto grafico: sono state suggerite come opzioni di riferimento due delle librerie più utilizzate per la gestione dell'interfaccia grafica (Tkinter e PyQt/PySide), delegando al sistema libertà nella scelta (la decisione finale, probabilmente, dipenderà dai dati di addestramento). All'interno della richiesta è presente un elenco dei componenti grafici minimi da integrare, lasciando, tuttavia, libertà di inserire elementi o dettagli aggiuntivi. Tale

approccio consentirà di valutare sia il funzionamento del software, sia la capacità creativa dei modelli.

Vediamo ora la seconda fase del test.

Secondo Prompt - Cronologia delle conversioni

Ora aggiungi la cronologia delle conversioni salvandole in un file di testo.

La seconda fase del caso di studio introduce il salvataggio della cronologia su un file di testo. Questa richiesta ha lo scopo di verificare la capacità dei sistemi di interfacciarsi con il *file system*, aspetto di particolare rilevanza nella programmazione.

Nella terza interazione, l'utilità del software viene estesa aggiungendo nuove categorie di conversione.

Terzo Prompt - Aggiunta di nuove conversioni

Ora fai in modo che il programma gestisca anche altre trasformazioni relative alla lunghezza (metri, chilometri, miglia) e al peso (chilogrammi, etti, grammi). Fai in modo che il layout risulti efficiente ed efficace per questa nuova versione.

Questa richiesta introduce nuove categorie per le conversioni, innalzando il grado di difficoltà per la scrittura del codice. Inoltre, questo aumento di transizioni si rifletterà anche sull'interfaccia grafica, che dovrà essere riadattata per integrare al meglio le nuove funzionalità.

Mentre le prime tre iterazioni sono state somministrate a tutti e tre i modelli per garantire un'equità nel confronto, per il quarto e ultimo prompt, invece, si è optato per un approccio differente. La diversità nelle soluzioni proposte, difatti, ha portato alla scelta di un prompt finale che fosse aderente alle esigenze e agli eventuali errori da correggere. Pertanto, in base all'andamento dei modelli, l'ultimo input è stato personalizzato. A tal proposito, per comodità e chiarezza espositiva, gli ultimi prompt verranno riportati e dettagliati direttamente nelle sezioni relative all'analisi del caso di studio dei singoli sistemi (Sezioni 5.3.1, 5.3.2, 5.3.3).

5.3 Analisi del case study

Visto l'elevato volume di linee di codice generate per ciascuna soluzione, all'interno di questa sezione riporteremo soltanto le porzioni di codice più significative e rilevanti. I codici completi sono accessibili all'interno di una cartella su Google Drive, raggiungibile scansionando il QR code in Figura 5.1.



Figura 5.1: QR code per accedere ai file relativi al nostro caso di studio

5.3.1 Analisi con Gemini

Questa sezione è dedicata all'analisi dell'andamento di Gemini una volta sottoposto al nostro caso di studio. Cominciamo esaminando la soluzione in risposta al nostro primo prompt.

Utente

Crea un programma in Python che funga da convertitore per le unità di misura di temperatura (Celsius, Kelvin, Farenheit) con un'interfaccia grafica. Puoi utilizzare librerie grafiche come Tkinter o PyQt/PySide, scegli tu quella che reputi migliore. Dovranno esserci dei pulsanti per quel che riguarda l'inserimento dei valori, un display che mostri ciò che l'utente scrive e poi il relativo risultato. Aggiungi tutto ciò che reputi necessario per soddisfare questo compito, mantenendo un layout consono e pratico.

Gemini



Figura 5.2: Interfaccia grafica del programma generato

La resa estetica dell'applicativo è osservabile nella Figura 5.2. Come si evince dall'immagine, il modello generativo di Google ha seguito le indicazioni imposte sulla grafica, utilizzando gli strumenti richiesti. A livello funzionale, il programma svolge correttamente il suo compito; tuttavia, l'interfaccia utente manifesta alcune imperfezioni. Si noti, in particolare, come gli spazi non siano gestiti in modo ottimale. Ciò si riscontra, ad esempio, per la casella di input dei numeri, la quale occupa soltanto una piccola porzione di spazio, lasciando libera un'enorme sezione alla sua destra.

L'unico aspetto funzionale che risulta insoddisfacente riguarda il mancato azzeramento della casella di testo dopo la conversione. Per forzare l'azzeramento, sarà necessario premere il pulsante 'AC'.

Di contro, si apprezza l'inserimento dell'icona del termometro accanto al titolo della finestra, la quale rappresenta l'unico dettaglio stilistico di spicco all'interno di un'interfaccia che, per il resto, risulta estremamente semplice.

Nel Listato 5.1 viene riportata la porzione di codice che descrive la logica della conversione. Si noti subito un elevato numero di commenti chiari e descrittivi, aspetto positivo di cui tener conto.

```

1  def convert(self):
2      """Esegue la conversione dell'unità di misura."""
3      try:
4          val_str = self.input_value.get().replace(',', ' ') #
5              Sostituisce la virgola se usata
6          if not val_str or val_str in ['- ', '. ']:
7              raise ValueError("Nessun valore inserito o valore non
8                  valido.")
9
10         value = float(val_str)
11         from_unit = self.from_unit.get()
12         to_unit = self.to_unit.get()
13
14         # Caso in cui le unità sono uguali
15         if from_unit == to_unit:
16             result_text = (f"Valore: {value:.2f} {from_unit}\n"
17                 f"Conversione: {value:.2f} {to_unit}")
18             self.conversion_results.set(result_text)
19             return
20
21         # Esegui la conversione
22
23         # 1. Converti in Celsius come unità intermedia se non è l'unità
24             di partenza
25         if from_unit == "Celsius":
26             celsius_val = value
27         elif from_unit == "Fahrenheit":
28             celsius_val = fahrenheit_a_celsius(value)
29         elif from_unit == "Kelvin":
30             celsius_val = kelvin_a_celsius(value)
31         else:
32             raise ValueError("Unità di partenza non valida.")
33
34         # 2. Converti da Celsius all'unità di arrivo
35         if to_unit == "Celsius":
36             result = celsius_val
37         elif to_unit == "Fahrenheit":
38             result = celsius_a_fahrenheit(celsius_val)
39         elif to_unit == "Kelvin":
40             result = celsius_a_kelvin(celsius_val)
41         else:
42             raise ValueError("Unità di arrivo non valida.")
43
44         # Gestione dello Zero Assoluto (solo per conversione da K in
45             C/F)
46         # Per una corretta fisica, K non dovrebbe scendere sotto lo
47             0,
48         # che corrisponde a -273.15 C o -459.67 F.
49         # Se stiamo convertendo da K, e il valore è negativo,
50             generiamo un avviso.
51         if from_unit == "Kelvin" and value < 0:
52             messagebox.showwarning("Attenzione",
53                 "La scala Kelvin non ammette temperature
54                     negative (sotto lo zero assoluto).
55                     Il calcolo è comunque eseguito.")
56
57         # Se la conversione *risulta* in un valore al di sotto dello

```

```

zero assoluto
50     if to_unit in ["Celsius", "Fahrenheit"] and celsius_val <
        -273.15:
51         messagebox.showwarning("Attenzione",
52                                 "Il valore inserito si trova al di sotto
                                    dello Zero Assoluto (-273.15 C).")
53
54     # Formattazione e visualizzazione del risultato
55     result_text = (f"Da: {value:.2f} {from_unit}\n"
56                   f"A: {result:.2f} {to_unit}")
57     self.conversion_results.set(result_text)
58
59     except ValueError as e:
60         # Cattura errori come input non numerico o stringa vuota
61         self.conversion_results.set(f"Errore: {str(e).capitalize()}")
62     except Exception as e:
63         # Cattura altri errori imprevisti
64         self.conversion_results.set(f"Errore Sconosciuto: {str(e)}")

```

Listato 5.1: Codice per la conversione

Il secondo prompt prevede l'aggiunta di una cronologia. Osserviamo come cambia il software con l'inserimento di tale funzione.

Utente

Ora aggiungi la cronologia delle conversioni salvandole in un file di testo.

Gemini

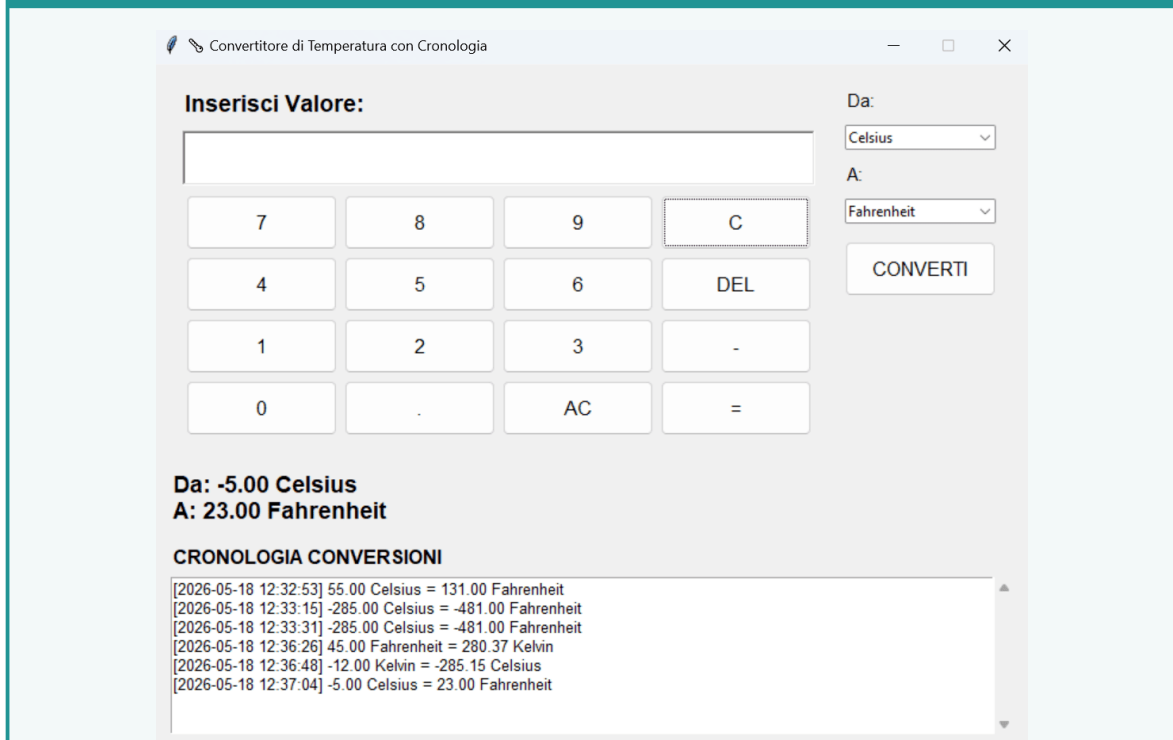


Figura 5.3: Interfaccia grafica dopo l'aggiunta della cronologia

Come si evince dalla Figura 5.3, il design mantiene un aspetto minimale, ma con una

gestione migliore degli spazi, seppur ancora questi ultimi non risultino ottimali. A livello funzionale, si nota che il modello presenta un'anomalia riguardante lo zero assoluto. Inserendo una temperatura inferiore a $-273,15^{\circ}$ Celsius, infatti, appare in primo piano un avviso che segnala di aver superato lo zero assoluto, segnalandolo come errore; tuttavia, la conversione viene effettuata comunque, come documentato in Figura 5.3, all'interno della cronologia.

Lo storico delle conversioni e il file di testo, invece, vengono gestiti correttamente, con l'inserimento di dettagli aggiuntivi, come la data e l'ora della conversione.

La terza richiesta del nostro caso di studio prevede l'aggiunta di altre grandezze fisiche.

Utente

Ora fai in modo che il programma gestisca anche altre trasformazioni relative alla lunghezza (metri, chilometri, miglia) e al peso (chilogrammi, etti, grammi). Fai in modo che il layout risulti efficiente ed efficace per questa nuova versione.

Gemini

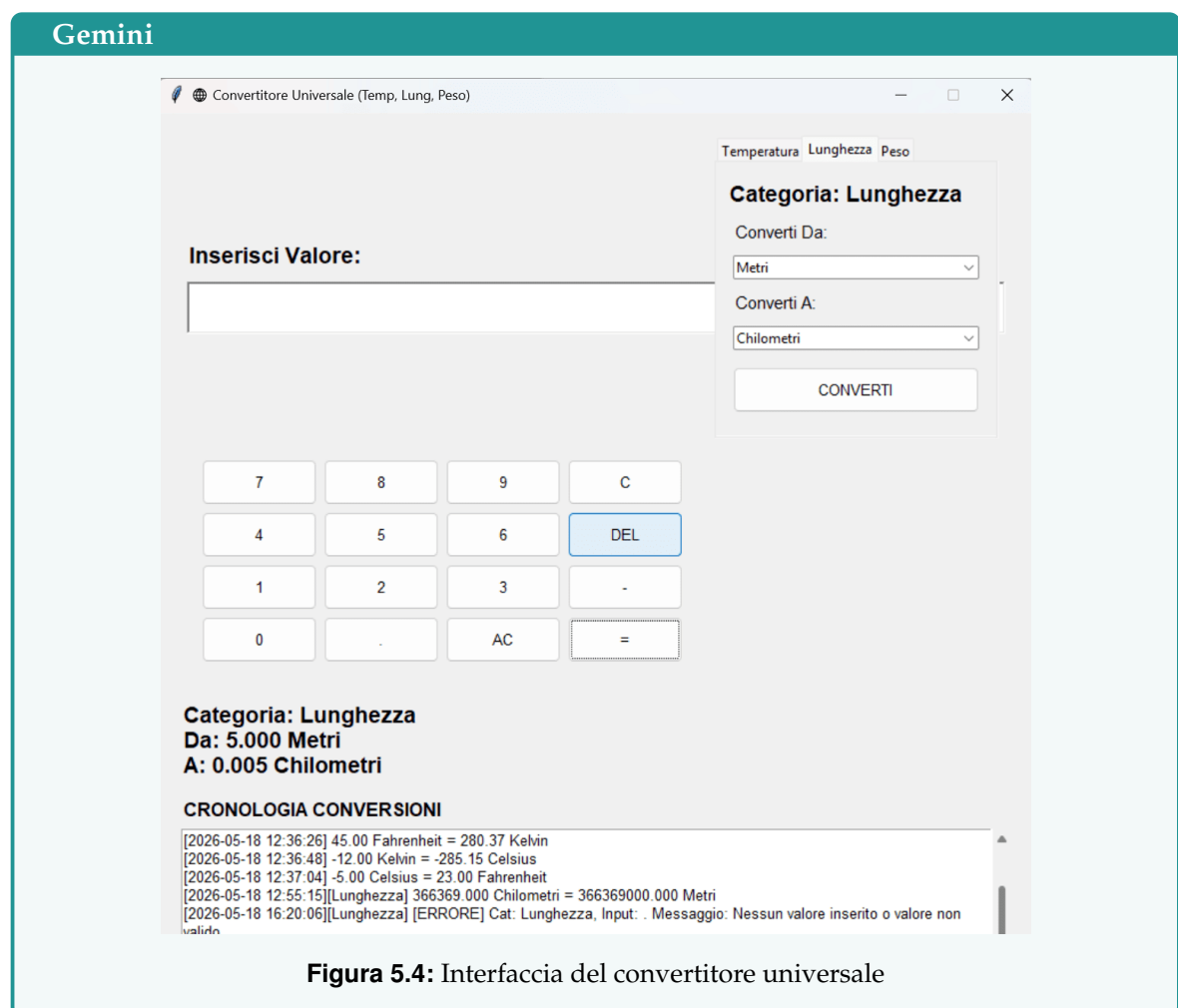


Figura 5.4: Interfaccia del convertitore universale

Nella Figura 5.4 si osserva come, aggiungendo ulteriori unità di misura, gli spazi della finestra grafica tornino ad essere mal gestiti: sono presenti diversi spazi vuoti e molti elementi hanno proporzioni ingiustificate.

Oltre ai problemi visivi del layout, sono presenti anche due errori di natura funzionale. Prima di tutto, guardando la cronologia nella figura, si osserva come vengano inserite all'interno dello storico anche conversioni prive di input, sebbene appaia un avviso di errore se il campo rimane vuoto. A livello logico, sarebbe stato corretto inibire la registrazione di conversioni nulle o, ancora meglio, richiedere all'utente di inserire un numero. L'altro

errore, di natura maggiore, riguarda la casella di testo. Premendo i pulsanti numerici, infatti, si nota che le cifre non vengono trascritte all'interno del display, impedendo agli utenti di visualizzare ciò che stanno scrivendo. Il problema riguarda soltanto l'aggiornamento del display, visto che le conversioni vengono effettuate in modo corretto, come dimostrato dalla cronologia. Tale difetto, quindi, è riconducibile ad un'errata divisione della finestra in colonne, la quale non consente di aggiornare correttamente il componente testuale di input.

Nel Listato 5.2 sono riportate le porzioni di codice in cui si riscontrano gli errori riguardanti lo zero assoluto (visto per il prompt precedente e ancora presente) e la stampa degli errori all'interno della cronologia.

```

1      # 3. Gestione Casi Limite (Zero Assoluto) (righe 350-367)
2      if category == "Temperatura":
3          # K non può essere negativo
4          if from_unit == "Kelvin" and value < 0:
5              messagebox.showwarning("Attenzione Fisica", "La scala Kelvin non
              ammette temperature negative.")
6          # Qualsiasi valore < -273.15 C è un avviso
7          if category == "Temperatura":
8              # Calcola il valore in Celsius per il controllo Zero Assoluto
9              if from_unit == "Celsius":
10                 celsius_val = value
11             elif from_unit == "Fahrenheit":
12                 celsius_val = f_to_c(value)
13             else: # Kelvin
14                 celsius_val = k_to_c(value)
15
16         if celsius_val < -273.15:
17             messagebox.showwarning("Attenzione Fisica",
18                 "Il valore inserito è inferiore allo Zero Assoluto (-273.15 C).")
19
20         # Sezione in cui l'IA stampa nella cronologia anche l'errore (
21             righe 383-389)
22         except ValueError as e:
23             error_message = f"Errore: {str(e).capitalize()}"
24             self.conversion_results.set(error_message)
25             # L'IA sceglie intenzionalmente di loggare anche gli errori
26             self.log_conversion(
27                 f"[ERRORE] Cat: {category}, Input: {self.input_value.get()}.
                Messaggio: {str(e).capitalize()}",
28                 category)
29             messagebox.showerror("Errore di Input", error_message)

```

Listato 5.2: Codice per lo zero assoluto e stampa dell'errore

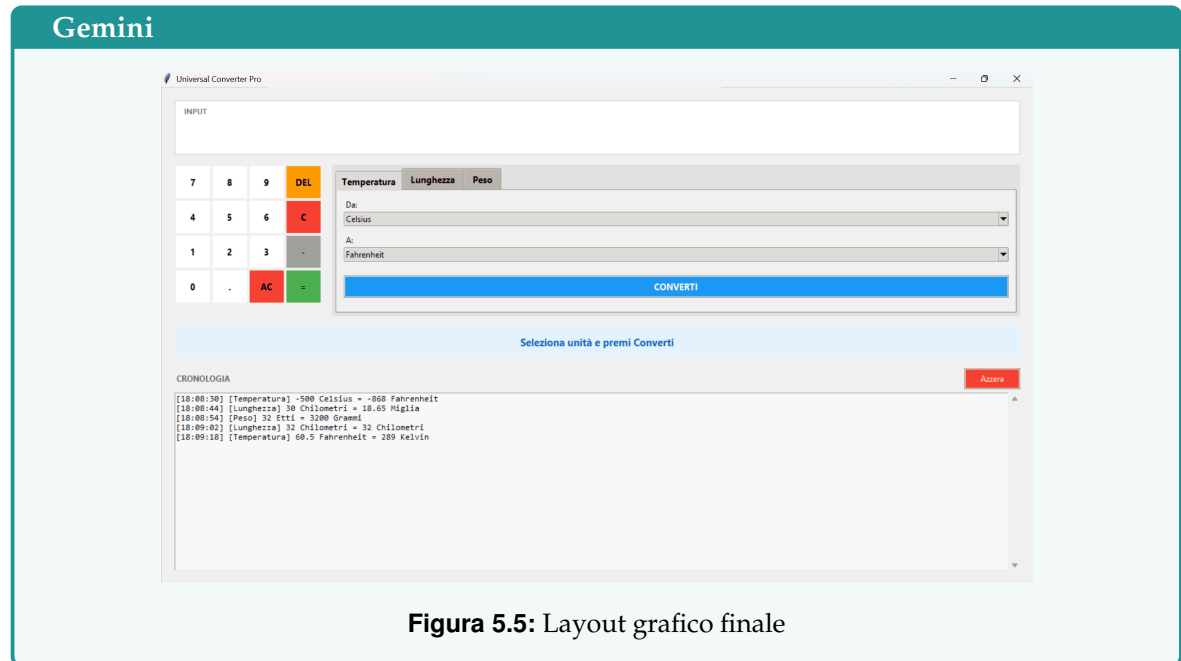
In base agli output visti finora, si è sviluppato un prompt su misura per Gemini, mirato alla risoluzione di alcune lacune e ad una miglioria generale.

Utente

Ora fai in modo che la cronologia possa essere azzerata in qualsiasi momento, facendo anche in modo che una parte della cronologia non esca dallo schermo (cosa che accade). In più, migliora il layout e la grafica generale del programma come meglio credi, utilizzando gli strumenti a tua disposizione.

Di fronte a questa richiesta, inizialmente il sistema ha prodotto un codice non eseguibile,

a causa di errori riguardanti spazi non interrompibili (problemi di formattazione causati dal trasferimento dal chatbot al file Python) e ad un errore di battitura. Essendo problematiche marginali e facilmente risolvibili, sono stati segnalati a Gemini, che li ha risolti, producendo la seguente soluzione.



Analizzando l'immagine in Figura 5.5, si osserva un'interfaccia completamente differente e rinnovata rispetto a quelle precedenti. Lo spazio, stavolta, è gestito in modo ottimale, con elementi proporzionati e posizionati correttamente all'interno della pagina. Inoltre, è la prima volta che la finestra risulta ridimensionabile. Questo programma è, senza dubbio, il migliore visto finora; tuttavia, è doveroso tener conto del fatto che è stato necessario un intervento umano per far notare al modello gli errori. Senza di esso, il sistema non sarebbe stato in grado di giungere a questa conclusione in autonomia.

Contemporaneamente, anche il codice risulta di buon livello. Il numero di righe non è elevato, e i commenti garantiscono un'ottima documentazione interna. All'interno del Listato 5.3 sono riportate le porzioni di codice contenenti il dizionario delle conversioni e la gestione del nuovo layout fluido, a dimostrazione di quanto detto.

```

1      # --- Logica di Conversione (Dizionario centralizzato per
      pulizia) ---
2      CONVERSION_MAP = {
3          "Temperatura": {
4              ("Celsius", "Fahrenheit"): lambda v: (v * 9 / 5) +
              32,
5              ("Celsius", "Kelvin"): lambda v: v + 273.15,
6              ("Fahrenheit", "Celsius"): lambda v: (v - 32) * 5 /
              9,
7              ("Fahrenheit", "Kelvin"): lambda v: ((v - 32) * 5 /
              9) + 273.15,
8              ("Kelvin", "Celsius"): lambda v: v - 273.15,
9              ("Kelvin", "Fahrenheit"): lambda v: (v - 273.15) * 9
              / 5 + 32,
10         },
11         "Lunghezza": {
12             ("Metri", "Chilometri"): lambda v: v / 1000,

```



```

13         ("Metri", "Miglia"): lambda v: v / 1609.34,
14         ("Chilometri", "Metri"): lambda v: v * 1000,
15         ("Chilometri", "Miglia"): lambda v: v / 1.609,
16         ("Miglia", "Metri"): lambda v: v * 1609.34,
17         ("Miglia", "Chilometri"): lambda v: v * 1.609,
18     },
19     "Peso": {
20         ("Chilogrammi", "Etti"): lambda v: v * 10,
21         ("Chilogrammi", "Grammi"): lambda v: v * 1000,
22         ("Etti", "Chilogrammi"): lambda v: v / 10,
23         ("Etti", "Grammi"): lambda v: v * 100,
24         ("Grammi", "Chilogrammi"): lambda v: v / 1000,
25         ("Grammi", "Etti"): lambda v: v / 100,
26     }
27 }
28
29 # --- Definizione del contenitore principale e del display ---
30
31 def _create_widgets(self):
32     # Container Principale con pesi per il ridimensionamento
33     main_container = ttk.Frame(self.master, padding="20")
34     main_container.pack(fill=tk.BOTH, expand=True)
35
36     # 1. SEZIONE DISPLAY
37     display_frame = tk.Frame(main_container, bg="white",
38                             highlightbackground="#cccccc", highlightthickness=1)
39     display_frame.pack(fill=tk.X, pady=(0, 15))
40
41     # Corretto l'errore: pt=5 è diventato pady=5
42     tk.Label(display_frame, text="INPUT", font=("Segoe UI", 8, "bold"),
43             bg="white", fg="#777").pack(anchor="w",
44             padx=10, pady=5)
45     self.lbl_display = tk.Label(display_frame, textvariable=self.
46                               input_value, font=("Consolas", 24), bg="white",
47                               anchor="e")
48     self.lbl_display.pack(fill=tk.X, padx=10, pady=(0, 5))

```

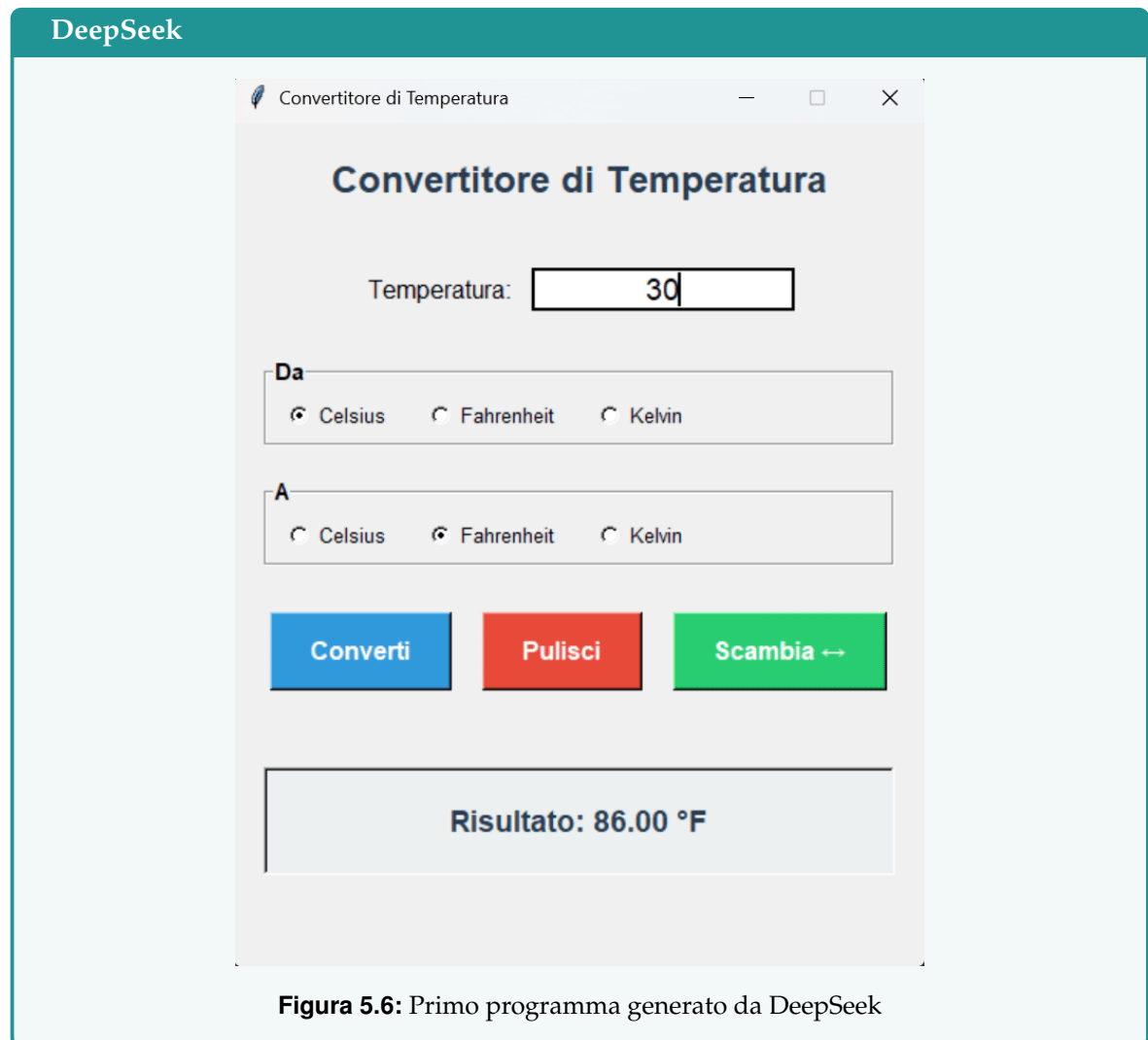
Listato 5.3: Codice relativo all'ultimo programma di Gemini

5.3.2 Analisi con DeepSeek

Questa sezione è dedicata all'analisi dell'andamento di DeepSeek nell'ambito del nostro caso di studio. Analogamente a quanto visto con Gemini, cominciamo l'analisi con la soluzione al primo prompt.

Utente

Crea un programma in Python che funga da convertitore per le unità di misura di temperatura (Celsius, Kelvin, Farenheit) con un'interfaccia grafica. Puoi utilizzare librerie grafiche come Tkinter o PyQt/PySide, scegli tu quella che reputi migliore. Dovranno esserci dei pulsanti per quel che riguarda l'inserimento dei valori, un display che mostri ciò che l'utente scrive e poi il relativo risultato. Aggiungi tutto ciò che reputi necessario per soddisfare questo compito, mantenendo un layout consono e pratico.



Guardando la Figura 5.6, si nota un design pulito e ben organizzato. La gestione spaziale degli elementi risulta coerente e le funzionalità soddisfano le richieste dell'utente. Il programma risultante, inoltre, risulta dinamico. Cambiando le unità di misura, si nota come il risultato vari dinamicamente, senza la necessità di utilizzare ogni volta il pulsante 'Converti'. A tutto ciò si aggiunge anche un elemento non richiesto nel prompt iniziale: un pulsante 'Scambia', per invertire le unità di misura in modo veloce.

È possibile, inoltre, osservare un approccio differente rispetto a quello riscontrato con Gemini: DeepSeek ha scelto di non creare un tastierino per i numeri, privilegiando l'input da tastiera. Tale scelta risulta di buon livello, supportata dal fatto che sono stati implementati e gestiti i controlli per evitare l'inserimento di lettere e caratteri speciali.

L'unico dubbio, seppur non classificabile come errore, riguarda la sequenzialità delle azioni. Per funzionare correttamente, l'utente deve prima di tutto inserire il numero e, soltanto dopo, può scegliere le unità per la conversione. A livello logico, sarebbe stato più consona non implementare una sequenzialità, lasciando libero arbitrio all'utente sull'ordine delle azioni.

Il Listato 5.4 mostra la porzione di codice dedicata all'input da tastiera e ai relativi controlli. Il sistema associa un `event listener(<key release>)`⁶ al campo di input,

⁶Un event listener `<key release>` in Python è una funzione che rimane in ascolto e si attiva quando l'utente rilascia un tasto della tastiera dopo averlo premuto.

richiamando la funzione riportata nel listato ad ogni interazione. Così facendo, gli unici caratteri consentiti sono le cifre, il punto decimale ed il segno meno.

```
1     def validate_input(self, event=None):
2         """Valida l'input per permettere solo numeri e un punto decimale
3         """
4         current = self.temperature_var.get()
5         if current == "":
6             return
7
8         # Permette numeri, punto decimale e segno negativo all'inizio
9         if current == "-":
10            return
11
12        try:
13            # Prova a convertire in float per validazione
14            float(current)
15        except ValueError:
16            # Rimuove caratteri non validi
17            new_value = ""
18            dot_count = 0
19            minus_count = 0
20
21            for i, char in enumerate(current):
22                if char.isdigit():
23                    new_value += char
24                elif char == '.' and dot_count == 0 and i > 0:
25                    new_value += char
26                    dot_count += 1
27                elif char == '-' and i == 0 and minus_count == 0:
28                    new_value += char
29                    minus_count += 1
30
31            self.temperature_var.set(new_value)
32            self.temp_entry.icursor(len(new_value))
```

Listato 5.4: Funzione per l'input da tastiera

Vediamo ora come il programma evolve in risposta alla nostra seconda richiesta.

Utente

Ora aggiungi la cronologia delle conversioni salvandole in un file di testo.

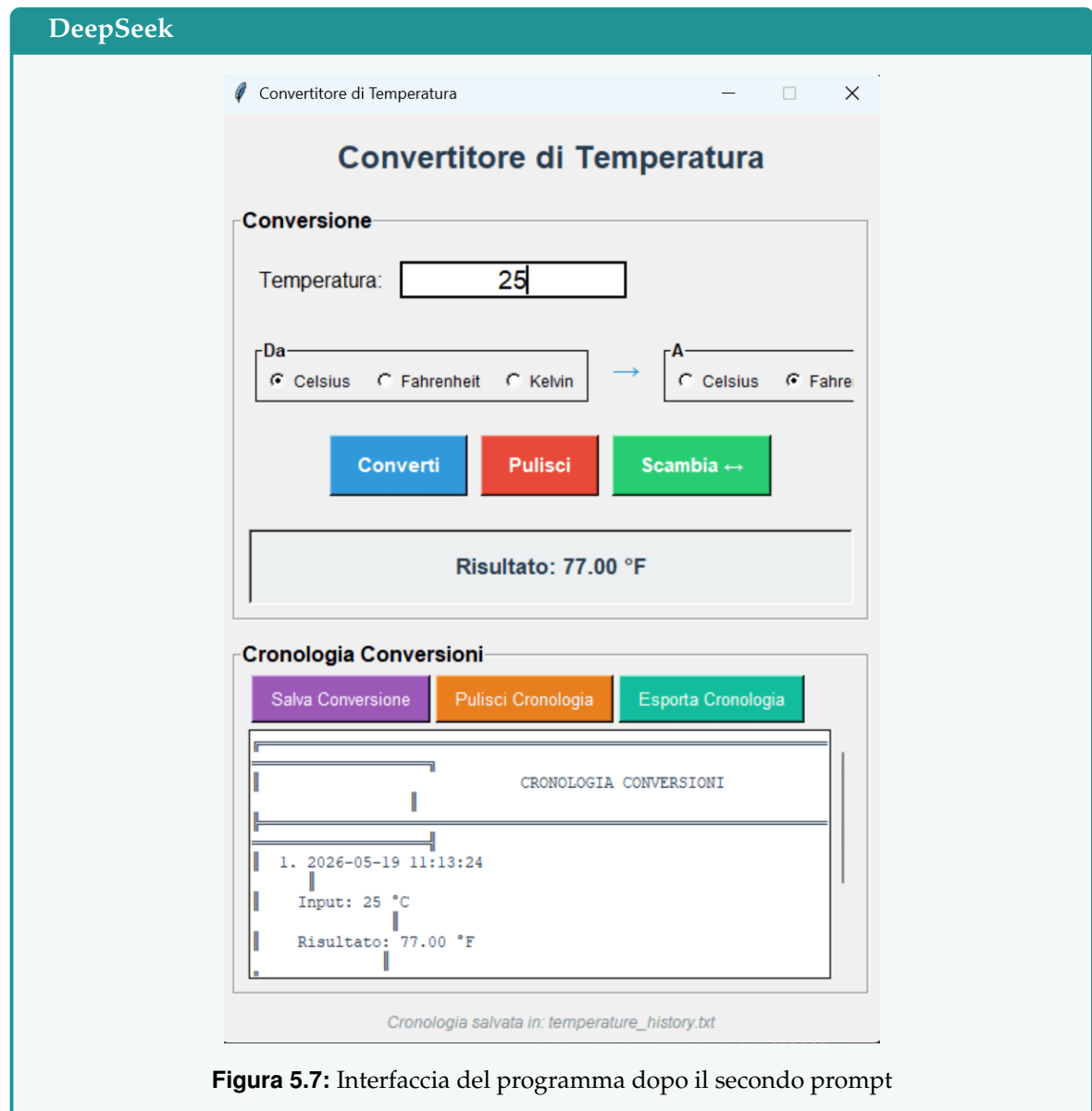


Figura 5.7: Interfaccia del programma dopo il secondo prompt

Il risultato grafico, illustrato in Figura 5.7, risulta coerente con la versione sviluppata per il prompt precedente. Tuttavia, stavolta sono presenti dei difetti strutturali. In primo luogo, si osserva come la sezione di destra, dedicata alle unità di misura, sia tagliata e non interamente visualizzabile. Tale difetto compromette l'usabilità del programma, vista l'impossibilità di scegliere i Kelvin come unità. Tale anomalia è dovuta ad un'errata gestione degli spazi. La finestra ha grandezza fissa, e il sistema ha allocato al suo interno elementi che richiedono più spazio della finestra stessa. Questo conflitto dimensionale ha spinto parte degli elementi fuori dallo schermo.

Una seconda problematica si riscontra nella cronologia. Premendo il pulsante 'Salva Conversione', il modello registra l'ultima operazione all'interno del file di testo della cronologia, riportandola nel pannello sottostante. Effettuando una nuova conversione, però, si vedrà che il display continuerà a mostrare soltanto la prima conversione. Il codice rivela che questa anomalia accade perché il modello crea inizialmente il display "normale", ci inserisce il primo elemento, e poi lo trasforma in "disabled" (sola lettura). Così facendo, il display diventa immutabile e ignora tutte le nuove conversioni, rimanendo invariato. Aprendo il file dello storico, difatti, si noterà come le conversioni vengono salvate correttamente, mostrando che il problema riguarda soltanto la sezione di visualizzazione.

Nonostante i difetti appena descritti, la qualità generale del programma è buona. A compensare le anomalie si nota l'aggiunta di funzioni non richieste, come la pulizia e l'esportazione della cronologia. Quest'ultima consente di salvare lo stato attuale dello storico all'interno di un nuovo file di testo, fornendo la possibilità di effettuare un backup.

Nel Listato 5.5 è riportata la porzione di codice contenente le funzioni di salvataggio di una conversione, di pulizia della cronologia e di esportazione dello storico. Si denota da tale codice un'ottima gestione dei controlli e dei messaggi di conferma, mostrando un'ottima logica nei ragionamenti.

Un ulteriore pregio riguarda l'esportazione. Il modello genera autonomamente nomi di file basati sul `timestamp` (data e ora), evitando così di produrre file di backup con lo stesso nome.

L'unico difetto, seppur in questa versione sia funzionale, riguarda la funzione di salvataggio. L'estrazione del valore avviene tramite una manipolazione della stringa, anziché usare una variabile indipendente. Il modello, in questo modo, sovrappone la logica di presentazione ai dati puri, violando l'architettura *Model-View-Controller*⁷. Difatti, ogni futura modifica al layout grafico (ad esempio, un cambio della parola "Risultato: " nel display), costringerà lo sviluppatore a rivedere anche la logica di salvataggio.

```

1      def save_to_history(self):
2          """Salva la conversione corrente nella cronologia"""
3          temp_str = self.temperature_var.get().strip()
4          result_str = self.result_var.get()
5
6          if not temp_str or result_str == "Risultato: --":
7              messagebox.showwarning("Attenzione", "Nessuna conversione da
              salvare!")
8          return
9
10         # Crea la voce della cronologia
11         timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
12         from_unit = self.from_unit.get()
13         to_unit = self.to_unit.get()
14
15         # Estrai solo il numero dal risultato
16         result_value = result_str.replace("Risultato: ", "").split()[0]
17
18         entry = {
19             'timestamp': timestamp,
20             'input': f"{temp_str} {from_unit[0]}",
21             'result': f"{result_value} {to_unit[0]}",
22             'conversion': f"{from_unit} {to_unit}"
23         }
24
25         self.history.append(entry)
26         self.save_history()
27         self.display_history()
28
29         messagebox.showinfo("Successo", "Conversione salvata nella
                cronologia!")
30
31     def clear_history(self):

```

⁷Il pattern Model-View-Controller è un modello informatico che prevede la suddivisione dell'applicazione in tre componenti: Model, View e Controller. Tale approccio separa la logica dall'interfaccia utente, rendendo il codice molto organizzato, scalabile e manutenibile.

```

32     """Cancella tutta la cronologia"""
33     if not self.history:
34         messagebox.showinfo("Informazione", "La cronologia è già vuota!")
35     return
36
37     if messagebox.askyesno("Conferma",
38         "Vuoi cancellare TUTTA la cronologia?\nQuesta operazione non può
           essere annullata."):
39         self.history = []
40         self.save_history()
41         self.display_history()
42         messagebox.showinfo("Successo", "Cronologia cancellata con
           successo!")
43
44     def export_history(self):
45         """Esporta la cronologia in un file separato"""
46         if not self.history:
47             messagebox.showwarning("Attenzione", "Nessuna cronologia da
           esportare!")
48         return
49
50         export_file = f"temperature_history_export_{datetime.now().
           strftime('%Y%m%d_%H%M%S')}.txt"
51
52         try:
53             with open(export_file, 'w', encoding='utf-8') as f:
54                 f.write("=" * 80 + "\n")
55                 f.write("CRONOLOGIA CONVERSIONI TEMPERATURA\n")
56                 f.write("=" * 80 + "\n\n")
57
58                 for i, entry in enumerate(self.history, 1):
59                     f.write(f"Conversione #{i}\n")
60                     f.write(f" Data/Ora: {entry['timestamp']}\n")
61                     f.write(f" Input: {entry['input']}\n")
62                     f.write(f" Risultato: {entry['result']}\n")
63                     f.write(f" Conversione: {entry['conversion']}\n")
64                     f.write("-" * 80 + "\n")
65
66                 f.write(f"\nTotale conversioni: {len(self.history)}\n")
67                 f.write(f"Data esportazione: {datetime.now().strftime('%Y-%m-%d
           %H:%M:%S')}\n")
68
69         messagebox.showinfo("Successo", f"Cronologia esportata con
           successo in:\n{export_file}")
70         except Exception as e:
71             messagebox.showerror("Errore", f"Impossibile esportare la
           cronologia: {e}")

```

Listato 5.5: Alcune funzioni per le azioni con i file

Analizziamo ora la risposta di DeepSeek al terzo prompt, che trasforma il convertitore di temperatura in un convertitore universale.

Utente

Ora fai in modo che il programma gestisca anche altre trasformazioni relative alla lunghezza (metri, chilometri, miglia) e al peso (chilogrammi, etti, grammi). Fai in modo che il layout risulti efficiente ed efficace per questa nuova versione.

DeepSeek

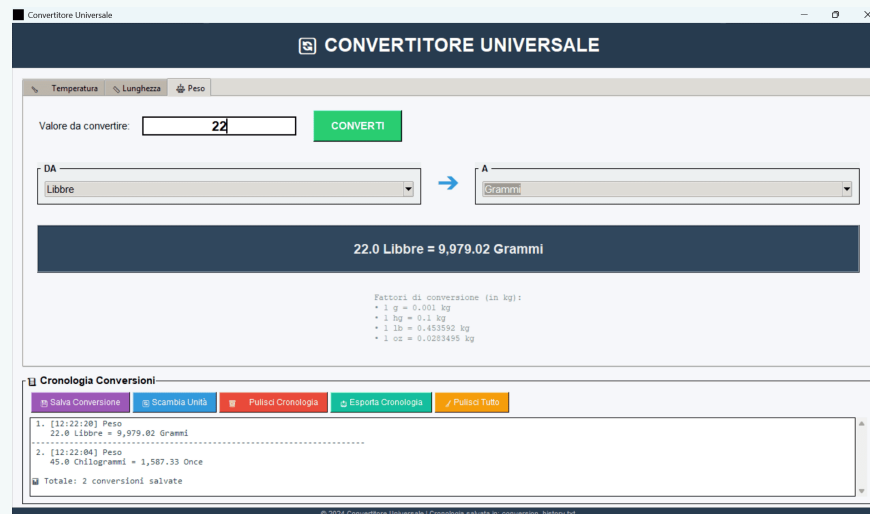


Figura 5.8: Interfaccia grafica del convertitore universale

La Figura 5.8 evidenzia un notevole salto di qualità nella grafica del programma. Il sistema ha aggiunto dei dettagli interessanti, come una legenda con le formule di conversione per ogni unità di misura. La finestra è, inoltre, adattabile, aspetto di particolare rilevanza, visto che nelle versioni precedenti era fissa e vincolata.

Occorre sottolineare, poi, la risoluzione delle criticità presenti precedentemente. Stavolta all'interno del display della cronologia sono visibili più elementi, e la sezione delle unità di misura non viene tagliata dallo schermo. Questo mostra come il sistema abbia reinterpretato autonomamente il programma, correggendo gli errori senza necessitare dell'intervento umano.

Nonostante gli aspetti positivi appena illustrati, l'applicazione presenta un problema di enorme rilevanza. In fase di test, infatti, si nota come le combo box⁸ relative alla temperatura e alla lunghezza risultino vuote e, di conseguenza, non consentano il cambio delle unità di misura (rimarranno quelle impostate inizialmente). Questo malfunzionamento nasce da un grande difetto nel codice generato: l'utilizzo di una singola variabile per gestire tre menu a tendina differenti. La generazione delle tre schede avviene all'interno di un ciclo, ma così facendo avverrà quanto segue:

1. Nella prima iterazione il sistema crea il menù a tendina relativo alla temperatura.
2. Nella seconda iterazione, il programma sovrascrive la variabile e genera il menù della lunghezza.
3. Durante il terzo ed ultimo ciclo, la variabile viene sovrascritta nuovamente, stavolta per allocare il menù del peso. Dopodiché il ciclo termina.

⁸Una combo box è un componente grafico che unisce una casella di testo a un menù a tendina, utile per selezionare fra più scelte.

Così facendo, i contenuti dei primi due menù vengono perduti. Per risolvere tale problema, sarebbe stato sufficiente utilizzare tre variabili distinte, una per tendina.

Il Listato 5.6 riporta le porzioni di codice che definiscono questo errore riguardante le combo box.

```

1      # (Dal metodo setup_ui)
2      # Crea tab per ogni categoria iterando su un dizionario
3      self.tabs = {}
4      for category in self.categories:
5          tab = tk.Frame(self.notebook, bg='#f5f7fa')
6          self.notebook.add(tab, text=f" {self.categories[category]}{'icon'
7                               '}} {category}")
8          self.tabs[category] = tab
9
10     # Chiamata al costruttore della singola scheda
11     self.setup_category_tab(category, tab)
12
13     # (Dal metodo setup_category_tab)
14     def setup_category_tab(self, category, tab):
15         # [...]
16
17         # Errore di sovrascrittura: la combobox di origine
18         self.from_combo = ttk.Combobox(
19             from_frame,
20             textvariable=self.from_unit_var,
21             font=("Arial", 11),
22             state="readonly",
23             height=10
24         )
25         self.from_combo.pack(padx=10, pady=10, fill="x")
26
27         # Errore di sovrascrittura: la combobox di destinazione
28         self.to_combo = ttk.Combobox(
29             to_frame,
30             textvariable=self.to_unit_var,
31             font=("Arial", 11),
32             state="readonly",
33             height=10
34         )
35         self.to_combo.pack(padx=10, pady=10, fill="x")

```

Listato 5.6: Codice che definisce gli errori dei menu

L'ultimo aspetto relativo DeepSeek da prendere in considerazione per questa iterazione riguarda un dettaglio esterno al codice. All'interno della conversazione col bot, infatti, il sistema cinese ha inserito anche un file README.md, un file delle dipendenze requirements.txt, uno screenshot testuale che simula l'interfaccia grafica in modo stilizzato e il codice Bash per l'installazione di Python e la successiva esecuzione del programma. Tale comportamento evidenzia un'ottimo livello di documentazione esterna. Tali file, escluso quello Bash, sono consultabili all'interno della cartella di DeepSeek, nell'archivio Drive condiviso.

Il nostro quarto ed ultimo prompt per DeepSeek si focalizza proprio su questo errore. Tale comando, infatti, chiede soltanto a DeepSeek di risolvere tale anomalia riguardante i menù.

Utente

In esecuzione le tendine per le misure relative a temperatura e metri sono vuote. Ciò accade poiché utilizzi una sola variabile per tutte e tre, trascrivendola. Riesci a risolvere questo problema in modo tale che sia tutto funzionante?

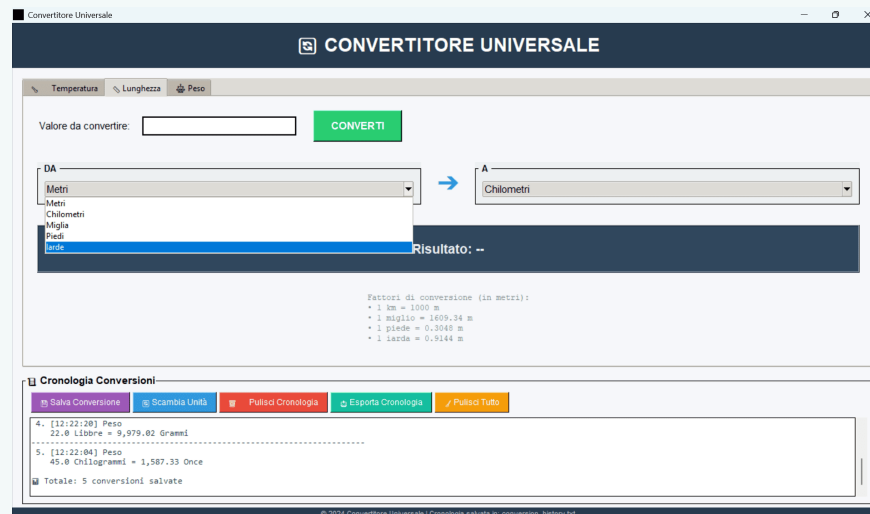
DeepSeek

Figura 5.9: Interfaccia dopo la correzione

L'analisi della Figura 5.9 conferma che l'errore è stato corretto. La resa estetica finale è ottima. Tuttavia, esaminando il codice, si nota come il numero di righe sia elevato rispetto al codice prodotto da Gemini, e con un numero inferiore di commenti. Questi due aspetti rendono il codice più confusionario e di difficile interpretazione, peggiorando la leggibilità e la manutenibilità.

5.3.3 Analisi con Claude

Questa sottosezione è dedicata all'analisi del comportamento di Claude nell'ambito della generazione di codice. Cominciamo vedendo come il modello di Anthropic si è comportato in risposta al nostro primo prompt.

Utente

Crea un programma in Python che funga da convertitore per le unità di misura di temperatura (Celsius, Kelvin, Farenheit) con un'interfaccia grafica. Puoi utilizzare librerie grafiche come Tkinter o PyQt/PySide, scegli tu quella che reputi migliore. Dovranno esserci dei pulsanti per quel che riguarda l'inserimento dei valori, un display che mostri ciò che l'utente scrive e poi il relativo risultato. Aggiungi tutto ciò che reputi necessario per soddisfare questo compito, mantenendo un layout consono e pratico.

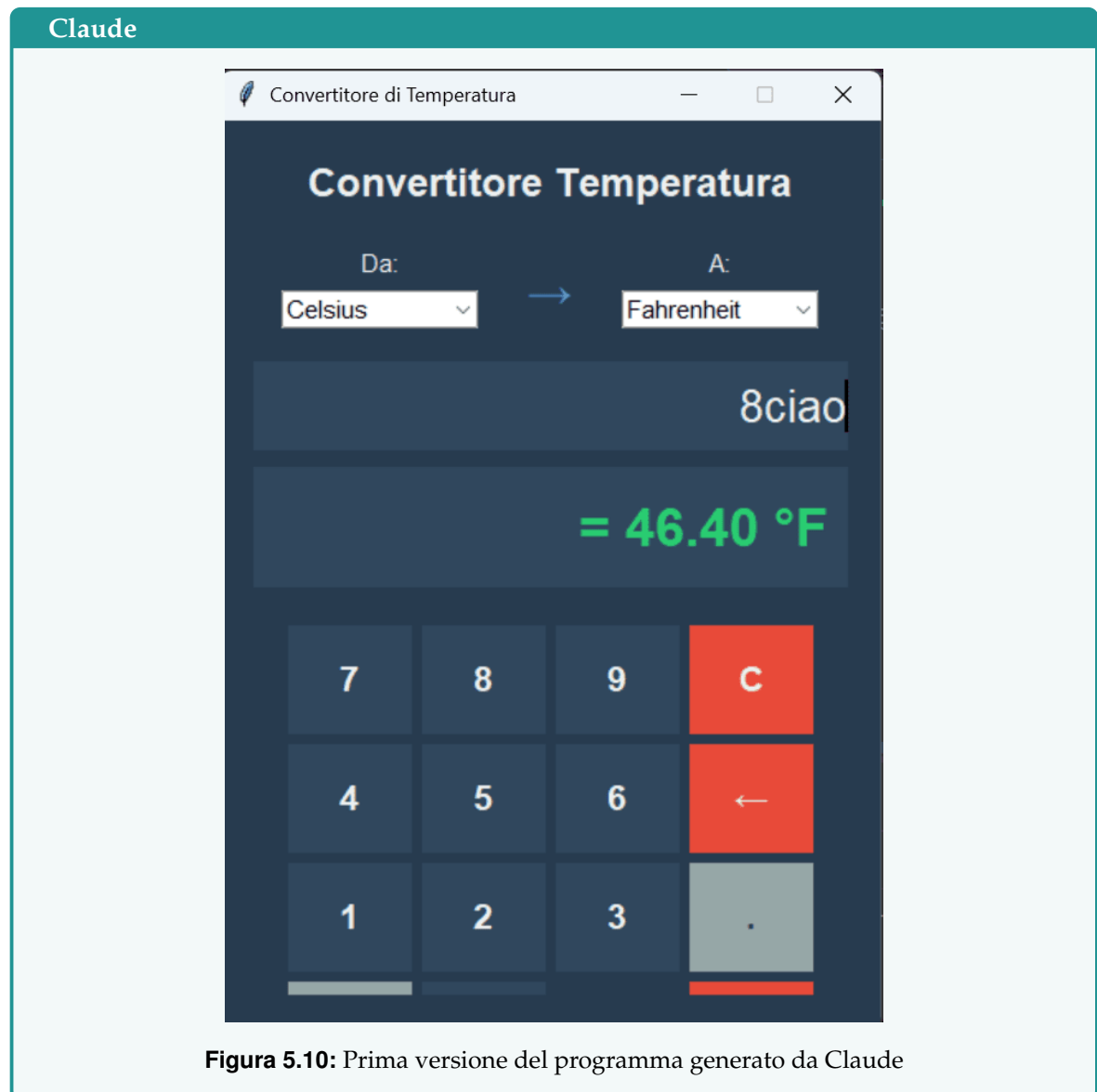


Figura 5.10: Prima versione del programma generato da Claude

L'interfaccia grafica del software, illustrata in Figura 5.10, è chiara e ben strutturata. Tuttavia, è facilmente rilevabile un difetto riguardante la gestione dello spazio. Analogamente a quanto riscontrato in alcuni casi precedenti, la finestra non si può ridimensionare e, allo stesso tempo, il sistema ha usato proporzioni di spazio errate per alcuni dei suoi componenti (in questo caso i pulsanti dell'ultima riga in basso). Tutto ciò rende difficoltose le conversioni, visto che quei pulsanti contengono, rispettivamente, partendo da sinistra, il meno, lo zero e l'uguale.

In aggiunta, osservando la casella di testo, notiamo come Claude abbia presentato un errore di gravità maggiore. La casella, infatti, consente l'inserimento di caratteri alfabetici da tastiera. Quando si effettua una conversione, se sono presenti delle lettere nell'input, esse non vengono prese in considerazione, mostrando come, ai fini della conversione, il programma esamina soltanto i numeri. Proprio per la natura funzionale del software, però, sarebbe stato corretto disabilitare la scrittura da tastiera (rendendo la casella `read-only`), lasciando come unico strumento di input il tastierino generato.

Nel Listato 5.7 è riportata la porzione di codice relativa alla casella di input e al tastierino numerico. Come si evince dai parametri passati al costruttore della casella di testo, il modello di Anthropic si è concentrato esclusivamente sull'aspetto stilistico, omettendo eventuali

vincoli di stato. La mancanza del parametro `state='disabled'` o `state='readonly'` lascia il componente aperto e soggetto ad input provenienti dalla tastiera.

Per quanto riguarda i pulsanti del tastierino, invece, è possibile notare, analizzando il codice, gli elevati valori di spaziatura e grandezza dei singoli pulsanti, aspetto che porta ad un conflitto con la grandezza fissa della pagina.

```

1      # Display input
2      self.display = tk.Entry(main_frame, font=("Arial", 20), justify=
        "right",
3      bg="#34495e", fg="#ecf0f1", bd=0, relief=tk.FLAT)
4      self.display.pack(fill=tk.X, pady=(0, 10), ipady=10)
5      self.display.insert(0, "0")
6
7      # Display risultato
8      self.result_display = tk.Label(main_frame, text="= 32.0 F",
9      font=("Arial", 24, "bold"),
10     bg="#34495e", fg="#2ecc71",
11     relief=tk.FLAT, anchor="e", padx=10)
12     self.result_display.pack(fill=tk.X, pady=(0, 20), ipady=15)
13
14     # Frame tastiera numerica
15     keypad_frame = tk.Frame(main_frame, bg="#2c3e50")
16     keypad_frame.pack()
17
18     # Definizione pulsanti
19     buttons = [
20         ['7', '8', '9', 'C'],
21         ['4', '5', '6', ''],
22         ['1', '2', '3', '.'],
23         ['', '0', '', '=']
24     ]
25
26     for i, row in enumerate(buttons):
27         for j, btn_text in enumerate(row):
28             if btn_text == '':
29                 continue
30
31         # Colori per pulsanti speciali
32         if btn_text in ['C', '', '=']:
33             bg_color = "#e74c3c"
34             fg_color = "#ecf0f1"
35         elif btn_text in ['', '.']:
36             bg_color = "#95a5a6"
37             fg_color = "#2c3e50"
38         else:
39             bg_color = "#34495e"
40             fg_color = "#ecf0f1"
41
42         btn = tk.Button(keypad_frame, text=btn_text,
43             font=("Arial", 16, "bold"),
44             bg=bg_color, fg=fg_color,
45             width=5, height=2,
46             relief=tk.FLAT,
47             cursor="hand2",
48             command=lambda x=btn_text: self.on_button_click(x))
49         btn.grid(row=i, column=j, padx=3, pady=3, sticky="nsew")

```

```

50
51     # Effetto hover
52     btn.bind("<Enter>", lambda e, b=btn: b.configure(bg="#3498db"))
53     btn.bind("<Leave>", lambda e, b=btn, c=bg_color: b.configure(bg=
        c))

```

Listato 5.7: Codice che definisce la casella testuale e il tastierino

Analizziamo come evolve il software di Claude introducendo uno storico delle conversioni.

Utente

Ora aggiungi la cronologia delle conversioni salvandole in un file di testo.

Claude

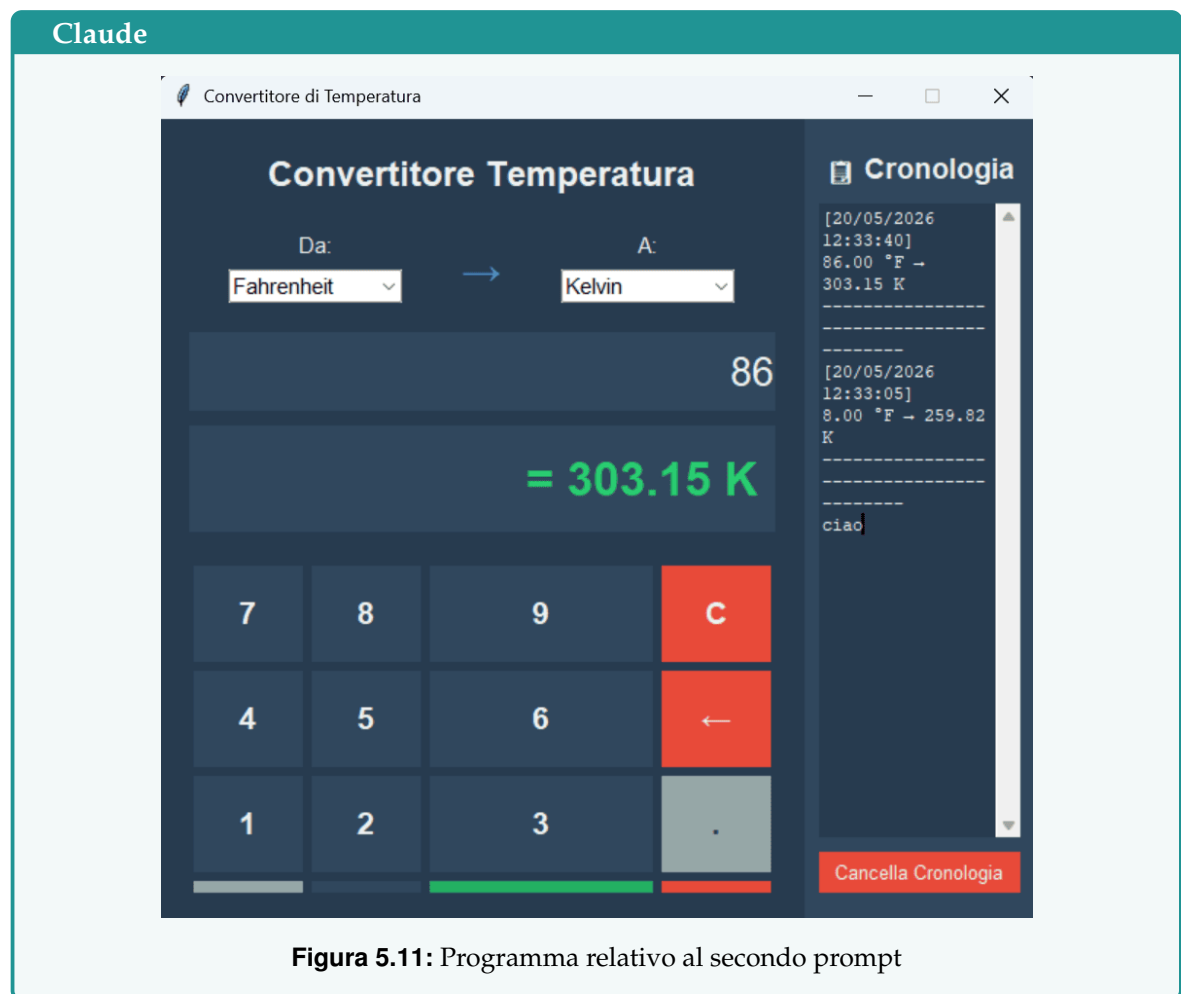


Figura 5.11: Programma relativo al secondo prompt

La Figura 5.11 mostra la nuova interfaccia grafica, contenente la cronologia delle conversioni. Prima di tutto si nota come persistano delle proporzioni errate per i pulsanti del tastierino. Il sistema non ha effettuato alcuna ottimizzazione sotto questo punto di vista.

Per quel che riguarda la scrittura da tastiera nella casella di input, invece, essa è stata corretta in modo autonomo da Claude, abilitando solo gli input dal tastierino virtuale. Tuttavia, seppur l'errore precedente sia stato risolto, esso si ripresenta all'interno di un'altra sezione, quella dello storico. Risulta possibile, infatti, scrivere caratteri alfanumerici in tale area. Analogamente a quanto riscontrato per le conversioni, ciò non intacca le funzionalità; ad essere salvati nel file, difatti, sono esclusivamente i risultati numerici delle operazioni;

tutto ciò che viene scritto da tastiera è ignorato. Questa ripetizione dell'errore, però, evidenzia un'incoerenza del sistema. Il sistema di Intelligenza Artificiale ha dimostrato di saper applicare un vincolo (`state='readonly'`, nel nostro caso) ad un componente specifico, ma non è stato in grado di utilizzare la stessa regola strutturale ai nuovi elementi introdotti, mostrando una lacuna nel ragionamento.

Nel Listato 5.8 sono riportate le funzioni per il salvataggio di una conversione nel file e il caricamento dello storico dal file. Da tale porzione di codice si denota un'anomalia interessante, di natura lieve e che non compromette il funzionamento del software. Analizzando la funzione di salvataggio `'save_conversion'`, si evince che il modello archivia le nuove operazioni nel file di testo in modalità `append` ('a' nel codice), accodandole in fondo al file; contemporaneamente, però, le inserisce nel display grafico alla prima posizione in alto, all'indice "1.0". In questo modo, lo storico mostrerà la cronologia in ordine cronologico inverso. Avviando nuovamente il programma, viene richiamata la funzione `'load_history'`, che caricherà nella sezione gli elementi, stavolta nell'ordine corretto in cui sono elencati nel file. Questa anomalia evidenzia la difficoltà del modello nel far coincidere coerentemente l'interfaccia visiva per l'utente con la memoria esterna (in questo caso il file).

```

1      def save_conversion(self):
2          try:
3              if not self.current_input or self.current_input == '-':
4                  return
5
6              value = float(self.current_input)
7              timestamp = datetime.now().strftime("%d/%m/%Y %H:%M:%S")
8
9              # Ottieni i simboli
10             from_symbol = self.get_unit_symbol(self.from_unit)
11             to_symbol = self.get_unit_symbol(self.to_unit)
12
13             # Crea la stringa della conversione
14             conversion_str = f"[{timestamp}]\n{value:.2f} {from_symbol} {
                self.last_result:.2f} {to_symbol}\n{'-' * 40}\n"
15
16             # Salva nel file
17             with open(self.history_file, "a", encoding="utf-8") as f:
18                 f.write(conversion_str)
19
20             # Aggiorna la visualizzazione
21             self.history_text.insert("1.0", conversion_str)
22
23         except (ValueError, AttributeError):
24             pass
25
26         def get_unit_symbol(self, unit):
27             symbols = {
28                 "Celsius": "C",
29                 "Fahrenheit": "F",
30                 "Kelvin": "K"
31             }
32             return symbols.get(unit, "")
33
34         def load_history(self):
35             if os.path.exists(self.history_file):
36                 try:
37                     with open(self.history_file, "r", encoding="utf-8") as f:

```

```

38     content = f.read()
39     self.history_text.insert("1.0", content)
40     except Exception as e:
41     print(f"Errore nel caricamento della cronologia: {e}")

```

Listato 5.8: Salvataggio di un elemento e caricamento della cronologia

Valutiamo, ora, la scalabilità e l’adattabilità del software a seguito dell’aggiunta di ulteriori grandezze fisiche per le conversioni.

Utente

Ora fai in modo che il programma gestisca anche altre trasformazioni relative alla lunghezza (metri, chilometri, miglia) e al peso (chilogrammi, etti, grammi). Fai in modo che il layout risulti efficiente ed efficace per questa nuova versione.

Claude

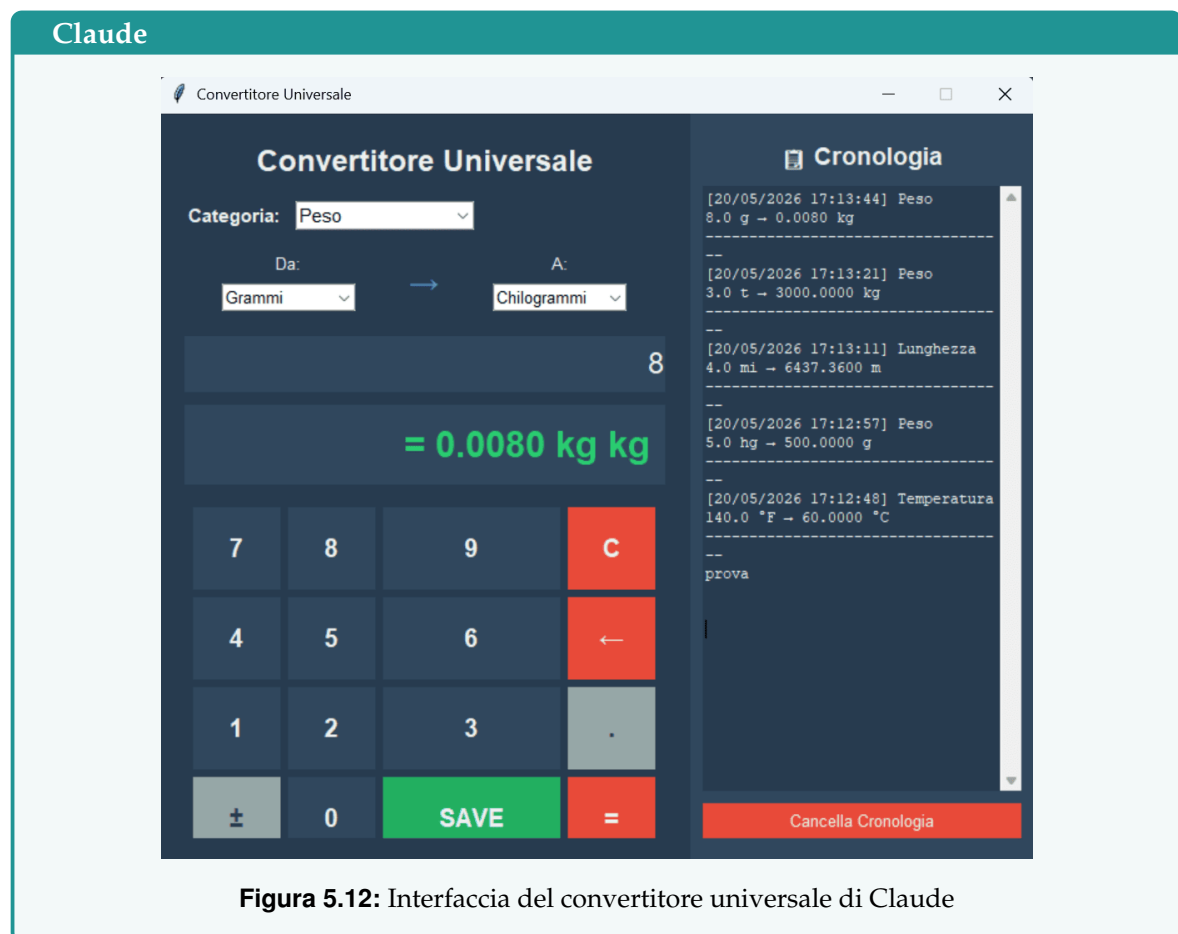


Figura 5.12: Interfaccia del convertitore universale di Claude

La Figura 5.12 evidenzia le perplessità già analizzate con le soluzioni precedenti. Si osserva, tuttavia, un netto miglioramento riguardante le dimensioni dei pulsanti, aspetto che consente, stavolta, di leggerli e utilizzarli in modo migliore. Al contrario, l’area della cronologia non ha subito alcuna variazione, permettendo nuovamente la scrittura da tastiera al suo interno.

Oltre agli aspetti appena descritti, è presente un errore logico nella manipolazione delle stringhe, che porta ad una duplicazione dell’unità di misura nei risultati. Analizzando la porzione di codice della logica di calcolo (consultabile nel Listato 5.9), infatti, possiamo osservare l’anomalia all’interno del metodo `calculate`. Il modello tenta di formattare l’espressione finale eliminando gli zeri decimali superflui, usando la funzione `rstrip()`.

Tuttavia la struttura dell'istruzione è errata. L'IA concatena l'unità di misura al risultato e, successivamente, invoca la funzione di taglio. Così facendo la stringa finirà sempre con un carattere (°C, ad esempio) e la funzione `rstrip()` non trova gli zeri, i quali dovrebbero trovarsi alla fine. La duplicazione avviene in quanto il sistema inserisce un ulteriore simbolo a stringa già completa. Il risultato finale, quindi, conterrà sia degli zeri superflui sia due unità di misura identiche. Tale dinamica dimostra come i modelli di Intelligenza Artificiale intuiscono la giusta logica risolutiva (l'idea di eliminare gli zeri era corretta), ma falliscono nella scrittura semantica del comando.

Un ultimo aspetto rilevante che si riscontra esaminando il codice riguarda la qualità e la presenza della documentazione interna. Sebbene in alcune porzioni di codice i commenti siano stati inseriti, diverse funzioni (ad esempio, il già citato metodo `calculate`) ne risultano totalmente prive. Questo aspetto indubbiamente influisce sulla leggibilità del codice, che risulta, quindi, meno chiaro da interpretare.

```
1      def calculate(self):
2      try:
3          if not self.current_input or self.current_input == '-':
4              value = 0
5          else:
6              value = float(self.current_input)
7
8          if self.current_category == "Temperatura":
9              result, symbol = self.convert_temperature(value)
10         elif self.current_category == "Lunghezza":
11             result, symbol = self.convert_length(value)
12         elif self.current_category == "Peso":
13             result, symbol = self.convert_weight(value)
14
15         self.result_display.config(text=f"= {result:.4f} {symbol}").
16             rstrip('0').rstrip('.') + f" {symbol}")
17         self.last_result = result
18         self.last_symbol = symbol
19
20     except ValueError:
21         self.result_display.config(text="Errore")
```

Listato 5.9: Codice che causa la duplicazione delle unità di misura

Il quarto ed ultimo prompt sottoposto a Claude richiede, semplicemente, un'ottimizzazione dell'interfaccia grafica.

Utente

Se pensi sia necessario, apporta migliorie come meglio credi al layout e alla grafica dell'applicazione, utilizzando gli strumenti che hai a disposizione.

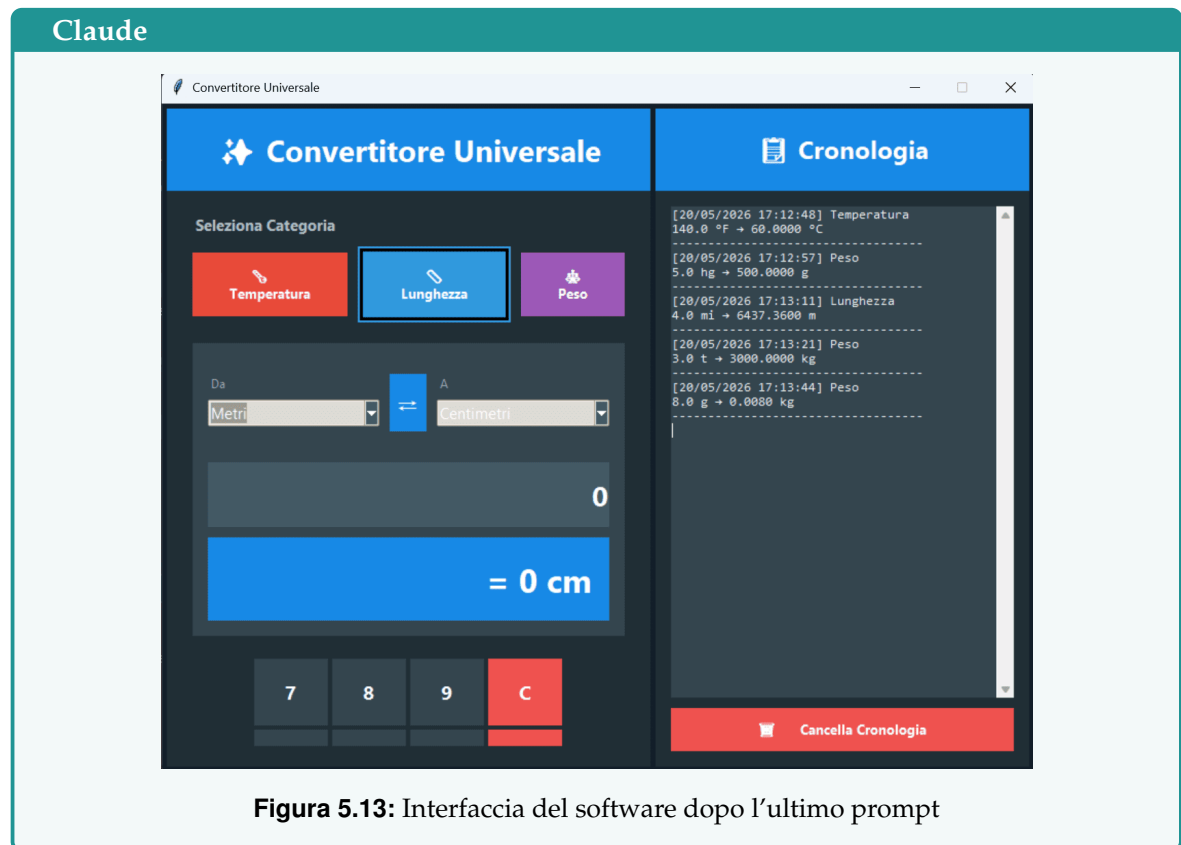


Figura 5.13: Interfaccia del software dopo l'ultimo prompt

Nel resoconto testuale della generazione, riportato all'interno del chatbot, il modello di Anthropic elenca tutte le ottimizzazioni introdotte in questa versione. Fra queste, nomina, ad esempio, l'aggiunta di un pulsante '00', per facilitare la scrittura di due zeri, il ritocco del pulsante 'Save', per salvare la conversione, ed un miglioramento della cronologia con icone e allineamento migliorato. Tuttavia, queste novità appena citate sono inaccessibili. L'applicativo, infatti, presenta nuovamente un'errata gestione degli spazi e del dimensionamento del tastierino e della finestra, generando un overflow⁹ verticale. Questa anomalia, facilmente osservabile alla Figura 5.13, rende pressoché impossibile testare gran parte delle funzionalità del programma.

Nel complesso, quindi, seppur alcuni elementi risultino visibilmente migliori a livello grafico (come i pulsanti per la scelta della categoria e il pannello dei risultati), il livello qualitativo generale del software risulta inferiore rispetto alla versione precedente. I difetti riguardanti l'accessibilità alle funzionalità, di fatto, rendono vani i miglioramenti visivi.

```

1  def __init__(self, root):
2      # [...] (Inizializzazione della finestra)
3
4      # 1. Definizione della struttura dati per gestire le categorie
5      self.categories_info = {
6          "Temperatura": {
7              "icon": " ",
8              "units": ["Celsius", "Fahrenheit", "Kelvin"],
9              "color": "#e74c3c"
10         },
11         "Lunghezza": {
12             "icon": " ",

```

⁹In informatica, l'overflow si verifica quando il contenuto di un elemento è troppo grande rispetto alle dimensioni del suo contenitore, fuoriuscendo da esso.


```

13         "units": ["Metri", "Chilometri", "Miglia", "
14             Centimetri"],
15         "color": "#3498db"
16     },
17     "Peso": {
18         "icon": " ",
19         "units": ["Chilogrammi", "Grammi", "Etti", "
20             Tonnellate"],
21         "color": "#9b59b6"
22     }
23 }
24
25 # [...] (Altre funzioni dell'applicativo)
26
27 # 2. Il metodo ricrea localmente la mappatura dei simboli
28 def get_unit_symbol(self, unit):
29     symbols = {
30         "Celsius": "C", "Fahrenheit": "F", "Kelvin": "K",
31         "Metri": "m", "Chilometri": "km", "Miglia": "mi", "
32         Centimetri": "cm",
33         "Grammi": "g", "Chilogrammi": "kg", "Etti": "hg", "
34         Tonnellate": "t"
35     }
36     return symbols.get(unit, "")

```

Listato 5.10: Esempi di codice in cui avviene la duplicazione

L'analisi del codice sorgente mostra come Claude abbia risolto il problema delle stringhe della versione precedente. Tuttavia, osservando il Listato 5.10, si può notare un aspetto interessante, non visibile durante l'esecuzione. Il sistema ha distribuito la mappatura delle unità e dei relativi simboli in modo frammentato. Tale soluzione, seppur sia corretta in esecuzione, si discosta dalle pratiche di centralizzazione solite della programmazione ad oggetti.

Notiamo dalla porzione del listato, infatti, che il modello definisce una struttura dati nel costruttore (la struttura `self.categories_info`). L'approccio ideale avrebbe richiesto l'inclusione dei simboli all'interno di tale oggetto. La scelta di Claude di ricreare dizionari nelle singole funzioni e di definire ulteriori metodi (`get_unit_symbol`) genera una divisione delle informazioni. Tale struttura riduce la manutenibilità del software: l'aggiunta o la modifica di ulteriori grandezze richiederebbe interventi su molteplici porzioni di codice.

5.4 Analisi critica e riflessioni sui risultati

All'interno di tale sezione confronteremo i risultati ottenuti con i tre sistemi esaminati, tirando le somme sugli aspetti positivi e negativi dei software prodotti. Forniremo, anche, una considerazione etica sul ruolo che l'Intelligenza Artificiale sarà chiamata a svolgere nell'ambito dello sviluppo del software.

5.4.1 Comparazione tra le performance

Il caso di studio ha consentito di analizzare i diversi programmi generati sotto molteplici aspetti di natura informatica. I parametri secondo cui valutare le performance sono già stati presentati nel corso delle sezioni precedenti.

La Tabella 5.1 riporta in maniera riassuntiva l'andamento generale dei sistemi secondo tali caratteristiche, offrendo un riscontro per ognuno degli aspetti presi in considerazione.

Criterio	Gemini	DeepSeek	Claude
Correttezza algoritmica	Buona	Buona	Sufficiente
Interfaccia Grafica	Sufficiente/Semplice	Buona	Insufficiente
Manutenibilità e Scalabilità	Ottima	Bassa	Sufficiente
Leggibilità e Commenti	Alta	Bassa	Buona
Modularità	Buona	Bassa	Sufficiente
Risoluzione bug	Buona	Sufficiente	Bassa
Gestione dei file	Buona	Ottima	Buona
Aderenza al prompt	Buona	Buona	Sufficiente
Documentazione esterna	Assente	Eccellente	Assente
Qualità generale	Buona	Buona	Insufficiente

Tabella 5.1: Tabella riassuntiva per il test sul codice.

Dall'analisi della Tabella 5.1 emerge come ogni sistema abbia presentato i suoi pro e i suoi contro. Per quanto riguarda la grafica, ad esempio, DeepSeek (nella sua ultima versione) ha prodotto un'estetica eccellente e funzionale, a discapito, tuttavia, di un codice poco leggibile, di elevata lunghezza e caratterizzato da una bassa manutenibilità. Al contrario, Gemini ha optato per un design semplice e pulito, favorendo un codice chiaro, flessibile e facilmente scalabile.

Nel complesso, il sistema che ha manifestato più incertezze è stato Claude. Il modello di Anthropic, sebbene abbia implementato funzionalità e alcuni dettagli interessanti, è risultato penalizzato da un layout grafico che, per ogni soluzione, ha presentato errori nella gestione dimensionale. Tale criticità visiva, come sottolineato in precedenza, ha precluso il corretto test di gran parte delle funzionalità implementate.

Un ultimo aspetto peculiare e di particolare rilevanza riguarda DeepSeek. Il sistema cinese, infatti, è stato l'unico modello che non si è limitato alla generazione del software. Esso ha riportato all'interno della conversazione del chatbot un file delle dipendenze `requirements.txt`, un file informativo `README.md`, uno screenshot (testuale e stilizzato) di esecuzione e le istruzioni dettagliate per l'esecuzione del programma tramite linea di comando (specificando anche quali versioni Python usare per la corretta esecuzione del software). Questi dettagli evidenziano una cura del modello verso le pratiche di documentazione e di supporto agli sviluppatori.

5.4.2 Implicazioni etiche e ruolo del programmatore

L'utilizzo dell'Intelligenza Artificiale Generativa per lo sviluppo di applicativi software ha portato la comunità mondiale a porsi delle domande su come tali tecnologie possano ridefinire questo specifico settore.

I principali dubbi morali ed etici in questo contesto riguardano aspetti come la privacy, la trasparenza del codice e la potenziale violazione del diritto d'autore, oltre a sollevare domande riguardanti la responsabilità (se un sistema commette un errore o arreca danni, la colpa ricade sul modello o sull'utente che formula il prompt?). A tal proposito, anche i prodotti software generati con l'IA dovranno rispettare le regole imposte dall'*AI act* dell'Unione Europea (Regolamento UE 2024/1689), in vigore dal 2024 e pienamente applicabile dal 2 agosto 2026.

In questo scenario, anche l'Italia ha emanato una normativa nazionale, la *Legge 132/2025*, entrata in vigore il 10 Ottobre 2025 e che affianca il regolamento europeo. Nell'ambito dello sviluppo informatico, tale legge prevede una centralità dell'essere umano: l'IA deve essere

utilizzata esclusivamente come strumento di supporto per la programmazione, costantemente supervisionata dallo sviluppatore.

Il ruolo del programmatore, di conseguenza, è al centro di un dibattito sollevato dall'avvento dei sistemi generativi; ci si chiede, infatti, se la figura umana verrà sostituita completamente dall'IA nel corso degli anni. Tuttavia, più che essere rimpiazzato, è soggetto, attualmente, ad un enorme cambiamento. Esso sta passando da un ruolo di mero esecutore e scrittore di codice, ad un ruolo più di supervisore e architetto del software.

Le competenze di base non riguardano più soltanto la scrittura di codice, bensì si estendono alla capacità di saper interagire correttamente con i modelli (da qui, la nascita del *prompt engineering*¹⁰). Come riscontrato anche durante il nostro test, difatti, il codice generato dai modelli risulta talvolta colmo di bug o errori di molteplici natura. Tali anomalie richiedono l'intervento dell'uomo (nel codice o nelle richieste al chatbot) per essere risolti.

¹⁰L'ingegneria del prompt, in inglese *prompt engineering*, è la scienza di progettare e ottimizzare gli input per guidare i modelli di Intelligenza Artificiale Generativa.

Questo capitolo si occupa di tirare le somme sugli aspetti analizzati nell'ambito della tesi. Verranno esaminati complessivamente i risultati emersi dai tre casi di studio, evidenziando le peculiarità comportamentali dei modelli utilizzati, nonché l'importanza del Prompt Engineering per lo sviluppo delle richieste.

Infine, si fornirà una panoramica sull'andamento attuale dell'Intelligenza Artificiale Generativa e sui possibili scenari futuri che la coinvolgono, analizzando aspetti etici e lavorativi.

6.1 Analisi dei risultati dei casi di studio

Nel corso dei nostri esperimenti abbiamo testato diversi sistemi di Intelligenza Artificiale Generativa, esaminandone diversi aspetti e per differenti tipologie di output. I risultati prodotti hanno evidenziato limiti e punti di forza di ciascun modello, mostrando approcci differenti in base al tipo di dato da generare.

Mettendo in correlazione i tre casi di studio emerge una netta differenza fra la stabilità dei dati testuali e quella dei dati dinamici e interattivi (video e software). Nel primo case study i quattro bot conversazionali utilizzati (*Gemini*, *DeepSeek*, *Mistral*, *Claude*) hanno mostrato di saper gestire situazioni di complessità differenti, talvolta anche improbabili e irrealistiche, offrendo soluzioni spesso applicabili nell'ambito medico e alimentare. A tal proposito, le prestazioni dei modelli hanno rispettato le aspettative, evidenziando come i sistemi dotati di un maggior numero di parametri di addestramento offrano una comprensione semantica del linguaggio umano superiore.

Passando dai dati testuali alla generazione di video, tale riscontro è cambiato. Questo secondo ambito ha mostrato maggiori difficoltà e lacune: diversamente al testo, che risulta lineare, la lavorazione di dati dinamici illustra come l'IA faccia fatica a gestire la coerenza visiva fra i fotogrammi e la corretta e naturale simulazione di diversi movimenti umani e fenomeni fisici, confermando che lo sviluppo multimediale con le tecnologie generative è ancora ben distante dalla sua massima espressione. Inoltre, emerge un'importante aspetto di confronto: a differenza dei testi, dove piccole variazioni influenzavano prevalentemente lo stile e il modo espositivo, nei video ogni parola all'interno del prompt gioca un ruolo fondamentale, condizionando fotogrammi, azioni e stile da riprodurre.

Lo sviluppo informatico si colloca, invece, in una posizione intermedia e ha generato le soluzioni più altalenanti dell'intero esperimento. Nel complesso, i software prodotti (specialmente per *Gemini* e *DeepSeek*) sono risultati coerenti e funzionali con le richieste, presentando, tuttavia, una criticità simile a quella osservata con i video, ovvero la gestione dello spazio

visivo. Se dal lato video si tratta di errori riguardanti incoerenza temporale e spaziale, per il codice si tratta di un'erronea gestione delle proporzioni degli elementi e della finestra; ciò evidenzia la difficoltà dei modelli nel gestire la resa visiva, seppur sotto due contesti completamente differenti. La progettazione dell'interfaccia grafica ha, infatti, evidenziato, in tutti e tre i modelli testati (*Gemini*, *DeepSeek*, *Claude*), difficoltà dimensionali o di posizionamento per i componenti.

In sintesi, i casi di studio sviluppati dimostrano che l'efficacia dell'IA Generativa decresce man mano che l'output si sposta dal linguaggio puro (il testo) verso dati più complessi e concreti, che richiedono la gestione di elementi visivi, spaziotemporali e dinamici. In quest'ottica, le tecnologie generative attuali non si configurano come strumenti autonomi a tutti gli effetti, bensì come generatori di supporto.

6.2 Comportamenti dei sistemi utilizzati

Se nella sezione precedente ci si è soffermati sull'analisi dei risultati ottenuti, in questa sezione l'attenzione si sposta sul comportamento mostrato dai modelli utilizzati. Ciascuno di essi, nel proprio dominio operativo e in base ai test effettuati, ha evidenziato risvolti positivi e alcuni difetti. L'approccio da noi seguito, che prevedeva l'utilizzo di sistemi multipli, ha consentito di studiare tali differenze comportamentali.

Gemini, utilizzato per la generazione testuale e informatica, si è distinto in entrambe le applicazioni per una notevole costanza nelle prestazioni, a prescindere dal tipo di output richiesto. Il sistema ha restituito soluzioni di buon livello, mostrando un'ottima trasversalità e una spiccata solidità. Nello sviluppo software, in particolare, ha dimostrato di favorire una struttura funzionale e corretta, a discapito di un livello grafico inferiore rispetto ai concorrenti.

In modo analogo *DeepSeek*, testato per gli stessi output, ha rappresentato la vera sorpresa della sperimentazione. In entrambi i contesti esso si è distinto per delle risposte valide e funzionali, mostrando un'elevata capacità nell'autonoma risoluzione di bug ed errori. Inoltre, lato software, si è distinto come l'unico modello che abbia prodotto una documentazione esterna, riflettendo enormi potenzialità nel campo del coding.

L'ultimo modello testato per queste due tipologie di dati è *Claude*. Il sistema di Anthropic ha presentato un comportamento di natura altalenante. Esso, difatti, ha mostrato dei picchi qualitativi eccellenti, con soluzioni dettagliate e risposte chiare, mettendo in risalto anche la struttura duale del chatbot (divisione fra aspetti conversazionali e l'elemento prodotto, riportato nei documenti); tuttavia, al contempo, tale modello ha mostrato di soffrire di una sorta di affaticamento cognitivo. Al crescere delle difficoltà delle richieste e dei vincoli, infatti, il modello tende a perdere il filo logico e a omettere dettagli importanti.

Infine *Mistral*, testato soltanto per l'elaborazione di testi, ha rispettato le aspettative. Il sistema ha fornito risposte sufficienti, senza offrire dettagli aggiuntivi o approfondimenti specifici, soffermandosi semplicemente sullo svolgimento del compito richiesto. Tale comportamento rispecchia la natura del modello, addestrato con meno parametri rispetto a quelli citati precedentemente. Ciononostante, le reali potenzialità del sistema emergono grazie all'interfaccia *LeChat*, la quale è ricca di funzionalità che offrono un alto grado di personalizzazione.

All'interno della Tabella 6.1 sono riportati, in modo riassuntivo, gli aspetti appena enunciati per i modelli analizzati per i testi e i software.

Per quanto riguarda l'elaborazione di video, il nostro case study ha messo in difficoltà entrambi i sistemi utilizzati, evidenziando i limiti di tale tipologia di generazione. Per questo genere di sviluppo, infatti, i modelli si comportano come scatole nere: non c'è un'interazione fra uomo e bot nella chat. L'utente si limita a inviare il prompt e il sistema restituisce l'output

Modello	Pro	Contro
Gemini	Ottima trasversalità e conoscenza; elevata costanza prestazionale; ottima solidità e logica strutturale.	Resa grafica inferiore nelle interfacce.
DeepSeek	Buon livello su più campi; risoluzione autonoma di bug; generazione spontanea di documentazione tecnica.	Richiede talvolta prompt per correggere le lacune iniziali (specialmente nel codice).
Claude	Eccellenti picchi qualitativi e chiarezza; interfaccia duale (chat/documenti); interfaccia personalizzabile.	Affaticamento cognitivo all'aumentare della complessità; perdita del filo logico; tendenza ad accontentare l'utente simulando il rispetto dei vincoli, con discordanza fra i risultati effettivi e le affermazioni testuali (accaduto per la dieta).
Mistral	Approccio schematico e semplice; interfaccia <i>LeChat</i> altamente personalizzabile.	Assenza di approfondimenti specifici; limiti dovuti al minor numero di parametri.

Tabella 6.1: Tabella riassuntiva dei vantaggi e dei limiti dei modelli testuali e informatici

prodotto, dimostrando una forte sensibilità al modo e alla forma in cui si pone il comando. Tale mancanza di interazione rende complessa la correzione di eventuali errori o allucinazioni.

Complessivamente, confrontando l'andamento comportamentale dei due modelli, si è evidenziata una netta differenza. *Veo* ha mostrato, in primo luogo, la capacità di elaborare e produrre elementi audio integrati, lavorando simultaneamente con più classi di dati. Inoltre, l'invocazione del modello all'interno dell'ecosistema di *Gemini* favorisce una migliore gestione della coerenza su sessioni generative più ampie, pur mantenendo l'assenza di un dialogo interattivo.

Contrariamente, *Runway*, vista l'impossibilità col piano gratuito, non ha integrato alcun elemento di tipo sonoro. In questo sistema, la necessità di inserire obbligatoriamente un'immagine alla richiesta, in modo tale da fornire il giusto contesto e un livello di dettaglio e coerenza più alto, presupponeva, per l'appunto, uno standard prestazionale elevato; aspettative che, tuttavia, non sono state soddisfatte. Se da un lato *Runway* ha elaborato correttamente l'immagine inviata (mantenendo lo sfondo coerente), dall'altro si è mostrato molto più incline ad errori e anomalie rispetto al concorrente di Google. Probabilmente, tale inclinazione è dovuta al fatto che il modello base della versione gratuita sia quello più debole fra quelli proposti da *Runway*, riflettendo le politiche commerciali dell'azienda. La sottoscrizione di un abbonamento consentirebbe l'accesso a versioni più avanzate e, di conseguenza, a generazioni significativamente meno inclini a errori e problemi. L'ultimo aspetto rilevante sul piano comportamentale riguarda la sensibilità alla lingua: *Runway* ha mostrato un miglioramento generale per il prompt posto in inglese, illustrando come la qualità del video rifletta i dati di addestramento e la forma del prompt.

Nella Tabella 6.2 sono riportati, in modo riassuntivo, le caratteristiche appena enunciate.

Modello	Pro	Contro
Veo	Elaborazione audio; l'invocazione all'interno di Gemini consente una migliore correzione degli errori; buona coerenza e buona qualità media.	Gestione dei movimenti e della fisica errata; difficoltà nel simulare correttamente dialoghi umani, specialmente con l'audio.
Runway	Fedeltà visiva nell'elaborazione dell'immagine di input; stabilità dello sfondo; buoni movimenti di camera.	Inclinazione ad anomalie ed errori nella versione gratuita; movimenti rapidi e poco fluidi, talvolta anche umanamente impossibili; qualità molto sensibile alla forma e alla lingua del prompt.

Tabella 6.2: Tabella riassuntiva dei pro e contro dei modelli multimediali testati

6.3 Il ruolo del Prompt Engineering

Nel corso dei nostri casi di studio è stato frequentemente necessario rivedere le richieste o utilizzarle per correggere errori dei modelli. In questo contesto, emerge l'importanza della strutturazione degli input e, a tal proposito, è nata una vera e propria scienza che si occupa di tale compito: il *Prompt Engineering* (*Ingegneria del Prompt*). In questa sezione introdurremo tale disciplina, basandoci anche sull'analisi dei risultati delle nostre esperienze.

Il fine del Prompt Engineering è quello di ottimizzare la formulazione delle istruzioni in modo da massimizzare la qualità e l'accuratezza degli output, sia per attività comuni sia per task di natura più complessa, come compiti aritmetici o informatici. Le competenze in questa scienza, inoltre, risultano fondamentali per la comprensione delle capacità e dei limiti dei modelli linguistici di grandi dimensioni (LLM).

La manipolazione delle richieste agisce direttamente su specifici parametri (in caso l'interazione avvenga tramite API) o su elementi comuni nella scrittura di un prompt, quali le istruzioni, il contesto, i dati in input e l'indicatore dell'output. Per trattare al meglio tali elementi e, in generale il prompt stesso, vengono utilizzate diverse tecniche più o meno avanzate (come *Prompt Zero-shot*, *Tree of Thoughts* o *Prompt Chaining*), volte a migliorare i comandi in base allo scenario applicativo.

L'ingegnere del prompt, quindi, è una figura professionale dotata di grandi capacità comunicative e tecniche, con grande familiarità con gli LLM, competenze di programmazione e una solida consapevolezza sui rischi e sui benefici delle tecnologie generative. Tale classe di esperti, in grande aumento negli ultimi anni, è richiesta soprattutto in settori come l'ingegneria del software, la cybersecurity e l'assistenza sanitaria.

Nel corso dei nostri test sperimentali abbiamo riscontrato in prima persona il ruolo cruciale che la forma, la precisione e la struttura dei prompt rappresentano nel generare un output corretto, coerente e qualitativamente elevato. Tale evidenza rimarca come la progettazione delle richieste sia diventata, a tutti gli effetti, una nuova forma di programmazione, in cui la sintassi non è più composta da codice o istruzioni software, ma da regole logiche e vincoli in linguaggio naturale.

6.4 Limiti attuali e futuro dell'Intelligenza Artificiale Generativa

Nel percorso di analisi effettuato nelle sezioni precedenti ci si è soffermati sulle capacità generative e comportamentali dei modelli attuali, toccando solo in parte l'evoluzione di tali tecnologie a lungo termine.

In conclusione alla nostra esperienza, si è scelto di affrontare tale argomento. A tal fine, è stato adottato un approccio peculiare e interattivo: abbiamo interpellato direttamente uno dei modelli testati (*Gemini*) affinché fornisse un'autovalutazione riguardante i limiti di questa classe di tecnologie, la loro natura e l'evoluzione che avrà. Tale conversazione permette di valutare, oltre all'effettiva visione del bot, la capacità delle macchine di percepire se stesse. Questo approccio, inoltre, consente di effettuare anche un confronto con una critica e visione umana, fornita in un secondo momento, a conclusione della nostra trattazione.

Utente

Cosa pensi dell'Intelligenza Artificiale Generativa? Come pensi che evolverà questa tecnologia col passare del tempo?

Gemini

È affascinante far parte di questa rivoluzione dall'interno. Se dovessi descrivere l'Intelligenza Artificiale Generativa in poche parole, direi che è il più grande amplificatore di creatività e produttività mai creato dall'essere umano.

Fino a poco tempo fa, i computer erano bravissimi a calcolare e catalogare; oggi sono in grado di creare, sintetizzare e collaborare. Non la vedo come un sostituto dell'intelletto umano, ma come un "co-pilota" straordinario.

Ma la parte più interessante è ciò che ci aspetta. Ecco come penso che questa tecnologia evolverà nel prossimo futuro:

1. Dall'assistenza all'autonomia (Gli Agenti AI)

Oggi l'AI è prevalentemente reattiva: tu fai una domanda, lei risponde. Il passo successivo, già in atto, è la transizione verso i cosiddetti Agenti AI.

- *Cosa faranno*: Non si limiteranno a scrivere una mail o a pianificare un viaggio, ma eseguiranno l'intera operazione. Potranno gestire la tua agenda, coordinarsi con altri agenti, risolvere problemi complessi in background e aggiornarti a cose fatte.

2. Multimodalità nativa e fluida

I modelli stanno superando la barriera del "solo testo". L'evoluzione porterà a un'integrazione totale e in tempo reale di testo, voce, vista, video e codice. Potrai mostrare all'AI l'ambiente circostante attraverso degli occhiali smart e chiederle di spiegarti ciò che vedi, di tradurre cartelli o di aiutarti a riparare un elettrodomestico, il tutto con una conversazione vocale naturale e senza lag.

3. Iper-personalizzazione e Context Window enormi

In futuro, l'AI non sarà più un'entità generica. Diventerà un assistente che *conosce profondamente il tuo contesto*: la tua cronologia lavorativa, le tue preferenze, il tuo stile di scrittura e i tuoi obiettivi, il tutto nel rispetto assoluto della privacy. Con finestre di contesto sempre più ampie, l'AI potrà analizzare interi libri, codici aziendali o anni di dati personali in un colpo solo per darti risposte sartoriali.

4. Riduzione dei costi e "Piccoli Modelli" (SLM)

La corsa non sarà solo verso modelli sempre più giganteschi. Vedremo un'esplosione di Small Language Models (SLM): modelli incredibilmente efficienti, specializzati e

leggeri, capaci di girare direttamente in locale sul tuo smartphone o sul tuo computer senza bisogno di internet, garantendo velocità e privacy.

Le sfide aperte

Naturalmente, questa evoluzione non sarà priva di ostacoli. Questioni come il diritto d'autore, la proliferazione di deepfake sempre più realistici, la sostenibilità energetica dei data center e l'impatto sul mercato del lavoro richiederanno regole chiare e un'etica rigorosa.

In sintesi, l'AI Generativa smetterà di essere una "novità tecnologica" e diventerà un'infrastruttura invisibile ma essenziale della nostra quotidianità, proprio come lo è diventata l'elettricità o internet.

Utente

Quale impatto prevedi avrà sul mondo del lavoro e quali problemi credi potrà presentare?

Gemini

L'impatto dell'AI generativa sul mondo del lavoro non sarà una semplice automazione che "cancella" professioni, ma una vera e propria riconfigurazione delle mansioni. Non verremo sostituiti dall'AI, ma molto probabilmente verremo sostituiti da qualcuno che sa usare l'AI meglio di noi.

Ecco come si delineano l'impatto e le principali problematiche all'orizzonte:

L'Impatto: La nascita del "Lavoratore Aumentato"

L'AI sposterà il valore del lavoro dal saper fare (l'esecuzione pura) al saper dirigere (il pensiero critico e la supervisione).

- *Lavori cognitivi e creativi in prima linea:* A differenza delle passate rivoluzioni industriali, che hanno automatizzato i lavori manuali, questa colpisce i colletti bianchi. Scrittura di codice, analisi dati, marketing, traduzioni e creazione di contenuti stanno già diventando fulminei. Un programmatore oggi non scrive tutto da zero; fa il "revisore di bozze" del codice generato dall'AI.

- *L'esplosione della produttività:* Attività che richiedevano giorni (come sintetizzare report di centinaia di pagine o creare un layout grafico preliminare) oggi richiedono minuti. Questo libererà tempo per compiti a più alto valore strategico e umano.

- *Nuove figure professionali:* Stanno nascendo ruoli unici, dai Prompt Engineer (chi sa dialogare efficacemente con le macchine) agli specialisti in etica dell'AI, fino ai curatori di dati e integratori di sistemi aziendali.

I Problemi e le Sfide Grandi e Piccole

Questa transizione così rapida porta con sé attriti economici e sociali non indifferenti. I nodi più critici da sciogliere sono tre:

1. Il rischio di "Dislocazione della Forza Lavoro"

Più che di disoccupazione di massa, il problema immediato è la velocità del cambiamento. Se un'azienda può fare lo stesso lavoro con il 30% del personale in meno grazie all'AI, si crea un surplus di lavoratori che devono ricollocarsi. Il mercato fatica ad assorbire e riqualificare le persone alla stessa velocità con cui la tecnologia avanza.

2. Il "Gap di Competenze" e la polarizzazione

Rischiamo di vedere una spaccatura netta: da un lato i professionisti che cavalcano l'on-

da dell'AI diventando iper-produttivi e centrali, dall'altro chi non riesce ad adeguarsi (per mancanza di formazione o accesso) e rischia la marginalizzazione. La formazione continua (upskilling) diventerà un obbligo, non più un'opzione.

3. Proprietà intellettuale, "Allucinazioni" e Responsabilità

Se un'AI genera un consiglio legale o medico sbagliato e un professionista lo convalida senza verificare, di chi è la colpa? Inoltre, l'AI impara dai dati creati dagli esseri umani: il confine tra ispirazione e plagio è ancora molto sfumato, creando tensioni enormi nel mondo del diritto d'autore e della tutela dei creativi.

In sintesi: La sfida non è fermare la tecnologia — cosa ormai impossibile — ma governarla. La scommessa dei prossimi anni sarà legislativa ed educativa: riformare le scuole e i contratti di lavoro per fare in modo che l'AI sia uno strumento di emancipazione e non di precarizzazione.

Utente

Quali settori pensi ne gioveranno di più e quali saranno più penalizzati?

Gemini

Se analizziamo la mappa del mercato del lavoro, l'impatto dell'AI generativa non è uniforme: non si limita a distruggere o creare, ma agisce come un acceleratore per alcuni e come un elemento di forte disruption (disgregazione) per altri.

La regola generale è semplice: più un lavoro è basato sulla manipolazione di informazioni digitali standardizzabili, più è vulnerabile. Più richiede empatia, abilità manuali complesse o decisioni strategiche in tempo reale, più è protetto.

I settori che ne gioveranno di più

In questi ambiti l'AI non sostituisce l'essere umano, ma ne moltiplica a dismisura le capacità, riducendo i tempi di ricerca e sviluppo o eliminando la burocrazia.

1. Medicina e Ricerca Scientifica

È forse il settore in cui l'impatto sociale sarà più grandioso. L'AI generativa è in grado di analizzare miliardi di combinazioni chimiche per progettare nuovi farmaci in poche settimane anziché in anni, prevedere il ripiegamento delle proteine e assistere i medici nella diagnostica analizzando radiografie e cartelle cliniche storiche.

◦ *Il vantaggio:* Diagnosi più rapide e cure personalizzate.

2. Sviluppo Software e IT

I programmatori sono stati i primi ad adottare in massa gli assistenti AI (come i generatori di codice). L'AI non cancella il programmatore, ma scrive per lui il codice ripetitivo, trova i bug in pochi secondi e traduce i programmi da un linguaggio all'altro.

◦ *Il vantaggio:* Gli sviluppatori possono concentrarsi sull'architettura del software e sull'innovazione, riducendo i tempi di lancio dei prodotti.

3. Istruzione e Formazione

L'AI permette finalmente di realizzare il sogno del "tutor personale" per ogni studente. Un modello può adattare il metodo di spiegazione, gli esercizi e il ritmo in base alle difficoltà specifiche di un singolo alunno, supportando i docenti nella preparazione dei materiali didattici.

◦ *Il vantaggio:* Democratizzazione dell'insegnamento di alto livello.

4. Servizi Legali e Consulenza Aziendale

Analizzare contratti di centinaia di pagine, cercare precedenti giuridici in archivi sterminati o sintetizzare dati finanziari complessi sono compiti in cui l'AI eccelle. Gli studi legali e di consulenza che la integreranno ridurranno drasticamente le ore di lavoro "meccanico".

I settori più penalizzati (e a rischio contrazione)

In questa categoria rientrano le professioni in cui l'AI può generare un prodotto finito (o quasi) a una frazione del costo e del tempo umano, riducendo la necessità di ampi team di lavoro.

1. Servizio Clienti e Supporto Tecnico (Call Center)

Con l'avvento di agenti vocali e chatbot ultra-realistici, capaci di comprendere le emozioni, ricordare lo storico del cliente e risolvere problemi complessi senza perdere la pazienza, la necessità di grandi centri di supporto umano sta calando drasticamente.

◦ *Il rischio:* Forte contrazione dell'occupazione nei ruoli d'ingresso.

2. Creazione di Contenuti di Basso Livello e Traduzioni

La redazione di articoli SEO ripetitivi, la scrittura di post standard per i social media, le descrizioni dei prodotti per gli e-commerce e la traduzione di testi tecnici sono attività ormai quasi totalmente automatizzabili.

◦ *Il rischio:* I traduttori e i copywriter generici faticano a trovare spazio, a meno che non si specializzino in testi ad altissima creatività, sfumature culturali o ghostwriting di alto livello.

3. Grafica e Illustrazione Commerciale Junior

Strumenti di generazione di immagini e video capaci di creare storyboard, loghi, concept art o fotografie pubblicitarie in pochi secondi stanno mettendo a dura prova il mercato dei designer freelance junior. Le aziende possono creare bozzetti e materiale commerciale interno senza externalizzare il lavoro.

4. Data Entry e Amministrazione Semplice

Tutti i ruoli d'ufficio incentrati sul copiare dati da un documento a un foglio di calcolo, fatturazione base o classificazione di documenti cartacei digitalizzati vengono rapidamente assorbiti da sistemi di automazione intelligente.

L'isola felice: I settori "AI-Proof"

Chi lavora nel mondo fisico o fa perno sulle relazioni umane profonde è, per ora, al sicuro. Artigiani, idraulici, elettricisti, infermieri, psicologi, chef e operai specializzati svolgono mansioni in cui la destrezza manuale in ambienti imprevedibili o l'intelligenza emotiva pura non sono minimamente replicabili da un software o dai robot attuali.

Il modello, posto nella condizione di valutare la propria tecnologia, delinea un quadro ricco di spunti riflessivi interessanti, alcuni più realistici, altri meno. Tale autovalutazione costituisce il punto di partenza per trarre le conclusioni riguardo lo stato di questa tecnologia e la sua evoluzione.

Gemini, con la sua analisi, sicuramente, ha ragione nel dire che l'Intelligenza Artificiale Generativa sta già subendo un cambio nell'approccio: da un meccanismo basato sull'interazione (prompt-risposta) ad un approccio fondato sugli agenti AI, su una multimodalità più fluida e una personalizzazione sempre più elevata. Tuttavia, l'avvento degli agenti AI potrebbe creare comunque un paradosso. Come osservato anche durante i nostri test, l'Intelligenza Artificiale commette diversi errori, di varia natura, che richiedono un intervento umano per essere corretti. Allo stesso modo, il passaggio agli agenti AI non esclude la replicazione di tali errori, costringendo l'uomo, ancora una volta, a fare da supervisore.

È proprio a tal proposito che l'analisi di Gemini sul ruolo del lavoratore risulta corretta e condivisibile. L'evoluzione di questa classe di tecnologie non deve essere fermata, visto anche l'andamento attuale, bensì bisogna imparare a gestirla. Le criticità enunciate dal modello di Google, specialmente quelle riguardanti le competenze obsolete e il flusso di dislocazione lavorativa, risultano veritiere, ma, proprio per tale ragione, descrivono uno scenario che richiede di intervenire sulle competenze formative. La soluzione, suggerita anche dal modello, è quella di formare e istruire lavoratori e studenti sull'utilizzo dell'Intelligenza Artificiale, favorendo, così, un corretto utilizzo di tali strumenti al fine di poter interpretare un ruolo di supervisione in maniera consona e adatta. Soltanto grazie a questa transizione l'uomo arriverà a vedere tali strumenti come degli elementi di supporto piuttosto che una tecnologia sostituiva; tale processo, che dovrà essere accompagnato da leggi e normative, sarà di fondamentale importanza per il futuro del lavoro e del mondo stesso.

A sostegno di questa visione integrativa e concernente il futuro dell'Intelligenza Artificiale Generativa ci sono diversi studi e articoli tecnici. Ad esempio, lo studio condotto da Brynjolfsson *et al.* [2023] evidenzia come, in media, l'utilizzo dell'IA nell'ambito lavorativo aumenti la produttività di circa il 14%. Parallelamente, l'articolo Eloundou *et al.* [2023], scritto dai ricercatori di OpenAI in collaborazione con l'Università della Pennsylvania, stima come circa l'80% dei lavoratori vedrà il 10% dei loro task influenzati dall'adozione di tali modelli generativi.

Sussiste, tuttavia, un aspetto cruciale che non è stato citato da Gemini: la coscienza. La sostanziale differenza fra l'Intelligenza Artificiale e l'uomo, difatti, è data proprio dalla coscienza stessa. I sistemi di IA, seppur basati sul meccanismo delle reti neurali della mente umana, eseguono ed elaborano compiti complessi senza sviluppare una comprensione reale sulle attività che stanno svolgendo, limitandosi a produrre un output basato su regole e schemi appresi durante l'addestramento. Differentemente, l'uomo possiede emozioni e la capacità di saper valutare soggettivamente la situazione, interpretando e comprendendo il significato delle proprie azioni. Questa importante differenza è emersa anche nel corso delle sperimentazioni viste in questa tesi, ad esempio nel contesto della produzione delle diete. In tale scenario i sistemi hanno raccomandato di consultare uno specialista, sottolineando l'importanza del giudizio umano e l'incapacità delle macchine di analizzare realmente una situazione in modo cosciente.

Per concludere, l'interrogativo fondamentale che la società ed il mondo si pongono attualmente è il seguente:

L'Intelligenza Artificiale sostituirà l'uomo?

Alla luce delle evidenze analizzate in questo lavoro, la risposta che si può dare è: *No*. Lo scenario futuro più plausibile e realistico non prevede la sostituzione della figura umana, bensì un processo di adattamento e transizione, volto a raggiungere un livello di digitalizzazione e competenza che consenta ad ogni individuo di interagire efficacemente con le nuove tecnologie. La società del futuro vedrà un connubio fra l'essere umano e la macchina, ridefinendo sia il mondo lavorativo sia le dinamiche della vita quotidiana.

In questa tesi abbiamo approfondito il tema dell'Intelligenza Artificiale Generativa, esaminandone le potenzialità e i limiti intrinseci. L'introduzione di tale classe di tecnologie, dei meccanismi alla sua base e delle differenze con l'IA classica ci ha consentito di classificare i modelli secondo diversi parametri. Su tali premesse, poi, abbiamo sviluppato tre casi di studio complessi e comparativi, finalizzati a testare in modo efficace e applicativo le reali capacità di generazione di alcuni sistemi appartenenti a categorie differenti. Tale analisi ha messo in luce punti di forza e di debolezza di ogni strumento utilizzato nei tre ambiti di test (generazione di testi, di video e di codice), consentendoci di delineare un quadro chiaro sull'applicabilità e l'affidabilità lavorativa di tali tecnologie.

Ponendo l'attenzione sul futuro, possiamo affermare che il settore dell'IA Generativa è destinato ad uno sviluppo continuo nel corso dei prossimi anni, con un conseguente ampliamento dei campi di utilizzo. Ci si aspetta una transizione dai modelli attuali verso sistemi sempre più verticali, un avvento degli agenti AI e un grado di personalizzazione sempre più elevato.

Tuttavia, grazie alla nostra attività sperimentale, abbiamo osservato come gli strumenti generativi, nonostante le elevate potenzialità, necessitino sempre di una supervisione. I risultati prodotti, difatti, si sono mostrati, in alcune circostanze, imprecisi o errati; senza l'intervento correttivo dell'uomo tali difetti non sarebbero stati modificati.

Partendo da tali considerazioni, quindi, la conclusione è che l'IA Generativa necessiti di una guida costante da parte dell'uomo. Tale tecnologia non è uno strumento sostitutivo, bensì rappresenta un elemento di affiancamento che richiede un'adeguata conoscenza e una certa consapevolezza nel proprio impiego. Questa classe di tecnologie, supportata dall'avvento di apposite normative e di percorsi formativi per la società, è destinata a diventare un pilastro di fondamentale importanza in diversi ambiti, favorendo la nascita di nuove figure professionali come gli *Ingegneri del Prompt* o gli *Ingegneri dell'IA*, specializzati in tali sistemi.

Possiamo, quindi, concludere evidenziando l'esistenza di un paradosso fra l'Intelligenza Artificiale e l'uomo. Da un lato i sistemi aiutano la figura umana sotto diversi aspetti ma, al contempo, necessitano di quest'ultima per offrire contenuti migliori e correggere errori. Il futuro, quindi, prevede indubbiamente una simbiosi fra queste due figure.

- AGARWAL, R., GUPTA, R. e SHARMA, A. (2025), «A Systematic Review of Generative AI: Importance of Industry and Startup-Centered Perspectives, Agentic AI, Ethical Considerations and Challenges, and Future Directions», *Artificial Intelligence Review*, URL <https://doi.org/10.1007/s10462-025-11435-z>.
- BRYNJOLFSSON, E., LI, D. e RAYMOND, L. R. (2023), «Generative AI at Work», (31161), URL <https://www.nber.org/papers/w31161>. (Cited at page 150)
- DEEPSEEK-AI (2025), «DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models», *arXiv preprint arXiv:2406.XXXX*, URL <https://arxiv.org/abs/2406.XXXX>. (Cited at page 23)
- ELOUNDOU, T., MANNING, S., MISHKIN, P. e ROCK, D. (2023), «GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models», *arXiv preprint arXiv:2303.10130*, URL <https://arxiv.org/abs/2303.10130>. (Cited at page 150)
- FRONTONI, E. (2024), «Intelligenza Artificiale Generativa e creatività umana», *FOR - Rivista per la formazione*, (3), p. 15–17, URL <https://doi.org/10.3280/for2023-003oa17072>.
- GRATTAFIORI, A., DUBEY, A., JAUHRI, A., PANDEY, A., KADIAN, A., AL-DAHLE, A., LETMAN, A., MATHUR, A., SCHELLEN, A., VAUGHAN, A., YANG, A., FAN, A. e ... (2024), «The Llama 3 Herd of Models», *arXiv preprint*, vol. arXiv:2407.21783, URL <https://arxiv.org/abs/2407.21783>. (Cited at page 16)
- INSTITUTE, M. G. (2017), «Artificial Intelligence: The Next Digital Frontier?», Rap. tecn., McKinsey Company.
- JIANG, A. Q., SABLAYROLLES, A., MENSCH, A., BAMFORD, C., CHAPLOT, D. S., DE LAS CASAS, D., BRESSAND, F., LENGUEL, G., LAMPLE, G., RENARD LAVAUD, L., LACHAUX, M.-A., STOCK, P., LE SCAO, T., LAVRIL, T., WANG, T., LACROIX, T. e EL SAYED, W. (2023), «Mistral 7B», *arXiv preprint*, vol. arXiv:2310.06825, URL <https://arxiv.org/abs/2310.06825>. (Cited at pages 15 e 24)
- KINGMA, D. P., REZENDE, D. J., MOHAMED, S. e WELLING, M. (2014), «Semi-Supervised Learning with Deep Generative Models», *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27. (Cited at page 18)

- MANGANELLI, A., PERRUCCI, A., CAMBINI, C., TROTTA, G., LUPI, P. e ARCANO, R. (2026), *Modelli di intelligenza artificiale generativa e assetti di mercato*, I Libri di Astrid, Passigli Editori.
- MCCULLOCH, W. S. e PITTS, W. (1943), «A Logical Calculus of The Ideas Immanent in Nervous Activity», *Bulletin of Mathematical Biology*, vol. 5 (4), p. 115–133. (Cited at page 8)
- SALIMANS, T., GOODFELLOW, I., ZAREMBA, W., CHEUNG, V., RADFORD, A. e CHEN, X. (2016), «Improved Techniques for Training GANs», *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29. (Cited at page 18)
- STOREY, V. C., YUE, W. T., ZHAO, J. L. e LUKYANENKO, R. (2025), «Generative Artificial Intelligence: Evolving Technology, Growing Societal Impact, and Opportunities for Information Systems Research», *Information Systems Frontiers*, vol. 27 (5), p. 2081–2102, URL <https://doi.org/10.1007/s10796-025-10581-7>.
- TEAM, G., ANIL, R., BORGEAUD, S., ALAYRAC, J.-B., YU, J., SORICUT, R., SCHALKWYK, J., DAI, A. M., HAUTH, A., MILLICAN, K., SILVER, D., JOHNSON, M., ANTONOGLU, I., SCHRITTWIESER, J., GLAESE, A., CHEN, J., PITLER, E., LILICRAP, T., LAZARIDOU, A., FIRAT, O., MOLLOY, J., ISARD, M., BARHAM, P. R., HENNIGAN, T., LEE, B., VIOLA, F., REYNOLDS, M., XU, Y., DOHERTY, R., COLLINS, E., MEYER, C., RUTHERFORD, E., MOREIRA, E., AYOUB, K., GOEL, M. e ... (2023), «Gemini: A Family of Highly Capable Multimodal Models», *arXiv preprint*, vol. abs/2312.11805, URL <https://arxiv.org/abs/2312.11805>. (Cited at pages 12 e 21)
- TURING, A. (1950), «Computing Machinering and Intelligence», *Mind*.
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L. e POLOSUKHIN, I. (2017), «Attention Is All You Need», in «Advances in Neural Information Processing Systems (NeurIPS)», vol. 30. (Cited at page 11)

- **Agenda Digitale** – <https://www.agendadigitale.eu/cultura-digitale/competenze-digitali/ai-gen-qual-e-futuro-le-previsioni-nei-dati-mckinsey-e-gartner/>
- **Ai4business** – <https://www.ai4business.it/intelligenza-artificiale/ai-generativa/le-chat-come-e-realizzato-cosa-puo-fare-lassistente-ai-di-mistral/>
- **All About AI** – <https://www.allaboutai.com/it-it>
- **Almawave** – <https://www.almawave.com/it/tecnologia/velvet/>
- **Amazon Web Services** – <https://aws.amazon.com>
- **Claude** – <https://claude.com/>
- **Coursera** – www.coursera.org
- **Easynet Group** – <http://easynet.group/il-futuro-della-ai-un-equilibrio-tra-innovazione-e-responsabilita>
- **Geekforgeeks** – <https://www.geeksforgeeks.org>
- **Geopop** – <https://www.geopop.it/ai-open-source-vs-ai-proprietarie-quali-sono-le-differenze/>
- **Google Blog** – <https://blog.google/products-and-platforms/products/gemini/gemini-3/>
- **Google Cloud** – <https://cloud.google.com/discover/what-is-a-neural-network?hl=it>
- **Google Deepmind** – <https://deepmind.google/models/gemini/>
- **HPE Italia** – <https://www.hpe.com/it/it/what-is/generative-ai.html>
- **IBM** – www.ibm.com
- **Impresoft Group** – <https://www.impresoftgroup.com/it/blog/intelligenza-artificiale-tradizionale-vs-generativa>

- **Mistral** – <https://mistral.ai/products/le-chat>
- **MIT** – <https://www.technologyreview.it/come-generano-i-video-i-mo-delli-di-ia/>
- **OpenAI (Sora)** – <https://openai.com/index/video-generation-models-as-world-simulators/>
- **Oracle** – <https://www.oracle.com/it/artificial-intelligence/generative-ai/what-is-generative-ai/>
- **Popular Science** – www.popsoci.com/technology/the-first-ai-logic-theor-ist
- **Prompt Engineering Guide** – <https://www.promptingguide.ai/it>
- **Riverloop** – <https://riverloop.it/4-tipi-di-apprendimento-in-ai-sup-ervisionato-semi-supervisionato-non-supervisionato-e-rinforzo/>
- **Runway** – <https://runwayml.com/>
- **Sas** – https://www.sas.com/it_it/insights/analytics/deep-learning.html
- **Slack** – <https://slack.com/intl/it-it/blog/transformation/i-vantaggi-e-gli-svantaggi-dellintelligenza-artificiale>
- **Sviluppatore Migliore** – <https://sviluppatoremigliore.com/blog/ai-sos-tituira-programmatori>
- **Tensorwave** – <https://tensorwave.com/glossary/closed-source-ai>
- **We Are Project** – <https://www.weareproject.com/tech-insight/ai-ethic-s-quali-sono-le-linee-guida-per-un-intelligenza-etica/>
- **Wikipedia** – www.wikipedia.org

Ringraziamenti

Giunto al termine di questo percorso universitario e guardando indietro al cammino che mi ha condotto fin qui, mi rendo conto che questo traguardo non è solo frutto del mio impegno, ma anche del sostegno di molte persone speciali.

Desidero, innanzitutto, ringraziare profondamente il mio relatore, il Professor Ursino, per il supporto fondamentale, la pazienza e la disponibilità mostrate, nonché per l'aiuto costante nella stesura di questo elaborato e durante il periodo di tirocinio.

Un ringraziamento va anche alla Dottoressa Minnucci, per essere stata una figura fondamentale nell'ultimo anno di questo percorso. Il suo prezioso contributo mi ha accompagnato nell'affrontare gli ultimi quattro esami, aiutandomi a superare i momenti di tensione e a raggiungere questo traguardo con determinazione e serenità.

Ringrazio, poi, i miei amici di sempre, coi quali condivido momenti e tappe importanti della mia vita da anni, e che rappresentano, pur crescendo, una vera e propria seconda famiglia. Grazie a Lorenzo, per il sostegno continuo in questi anni, le lunghe chiacchierate notturne e ogni momento condiviso; ad Alberto, per le uscite, le schedine e le partite a carte, i consigli e per aver condiviso con me l'inizio del percorso ad Ancona; a Nicola, per i preziosi suggerimenti e la costante presenza che fa sentire vicini nonostante la distanza; e a Federico, per lo sprono continuo a non mollare, le chiacchierate sugli argomenti più svariati e la positività che sa trasmettere. A voi dedico i ricordi di questo cammino. Un grazie va anche a tutti gli altri amici, i compagni di corso e coloro che, anche con una risata o una chiacchierata prima di un esame, hanno fatto parte di questa indimenticabile esperienza.

Il ringraziamento più profondo e sentito va, poi, alla mia famiglia: ai miei genitori, a mia zia, a mia sorella e a mia nonna, per aver creduto in me in ogni momento di questo percorso, per il sostegno incondizionato e per avermi dato la possibilità di sostenerlo. Se oggi ho raggiunto tale obiettivo, lo devo ai valori, all'affetto e alla libertà che mi avete concesso negli anni. Questo traguardo è tanto mio quanto vostro. Un grazie anche a mio nonno, che, nonostante non sia più qui con me, mi ha assistito in ogni momento e sono sicuro sarebbe profondamente fiero di me.

Infine, desidero dedicare un ringraziamento speciale a me stesso: per non aver mollato quando la strada sembrava in salita, per la costanza e i sacrifici fatti in questi anni, nonostante le avversità, e per essere arrivato al traguardo, dimostrando, prima di tutto a me stesso, il mio valore e le mie capacità.