UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Uso dei large language models a supporto dell'insegnamento e della comprensione didattica: studio sperimentale su task educativi

RELATORE

Prof. Fabio Palomba

Università degli Studi di Salerno

CANDIDATO

Giuseppe Di Somma

Matricola: 0512118580

Anno Accademico 2025-2026

Questa tesi è stata realizzata nel

Sic parvis magna

Abstract

I modelli linguistici di grandi dimensioni (LLM) stanno assumendo un ruolo crescente nei processi di insegnamento e apprendimento, offrendo possibilità concrete per la generazione di contenuti, il supporto alla didattica e la valutazione automatica. Tuttavia, lo stato dell'arte presenta ancora limiti rilevanti: mancano protocolli sperimentali replicabili, confronti sistematici tra modelli, analisi quantitative robuste e strumenti di valutazione coerenti per misurare l'efficacia reale delle diverse tecniche di prompt engineering.

Questa tesi propone un framework completo e generalizzabile per la valutazione comparativa degli LLM in contesto educativo. Il contributo include: (i) la definizione di sette task didattici, articolati sulle discipline di Analisi Matematica, Reti di Calcolatori, Programmazione Object Oriented e Insegnamento dell'Intelligenza Artificiale; (ii) il confronto di quattro modelli avanzati; (iii) tre tecniche di prompting; (iv) un Agente Valutatore basato su LLM capace di garantire coerenza, tracciabilità e rubriche strutturate; (v) un'analisi statistica approfondita dei risultati.

L'esperimento, composto da 334 osservazioni, mostra che GPT-5 ottiene le prestazioni migliori, seguito a breve distanza da Gemini 2.5 Pro e Claude Sonnet 4.5, mentre la tecnica Few-shot risulta la più stabile. Le analisi evidenziano differenze significative ma con effect size moderato, e nessuna correlazione sostanziale con tempo o lunghezza delle risposte.

Il lavoro colma le principali lacune dello stato dell'arte proponendo una metodologia replicabile, trasparente e supportata da metriche robuste, utile per studiare con rigore l'impiego degli LLM nella didattica universitaria.

Il codice sorgente, i dataset e la pipeline di valutazione sono disponibili pubblicamente all'indirizzo: `https://github.com/giudsv/LLMEducation`

Indice

Elenco delle Figure **iv**

Elenco delle Tabelle **v**

1 Introduzione **1**

1.1 Contesto Applicativo 1

1.2 Motivazioni e Obiettivi 2

1.3 Risultati Ottenuti 3

1.4 Struttura della Tesi 3

2 Background e Stato dell'Arte **4**

2.1 Background Tecnico sui Large Language Models 4

2.1.1 Architettura Transformer 5

2.1.2 Tokenizzazione e Processo Generativo 5

2.1.3 Parametri di Inferenza 6

2.2 Applicazioni degli LLM nell'Educazione 6

2.2.1 Tutoraggio Socratico: Il Caso Khanmigo 6

2.2.2 Modelli Verticali per le STEM: Minerva 7

2.2.3 Valutazione Automatica e Feedback 7

2.3 Tecniche di Adattamento e Specializzazione 8

2.3.1 Prompt Engineering: Zero-shot, Few-shot e CoT 8

2.3.2 Retrieval-Augmented Generation (RAG) 8
2.4 Limiti Aperti e Necessità di Valutazione 9

3 Metodo di Ricerca **10**
3.1 Domande di Ricerca 10
3.2 Disegno dello Studio 11
3.2.1 Selezione delle Discipline 11
3.2.2 Tassonomia dei Task Didattici 12
3.3 Prompt Engineering e Setup Sperimentale 12
3.3.1 Tecnica A: Zero-shot Prompting 13
3.3.2 Tecnica B: Few-shot Prompting 13
3.3.3 Tecnica C: Chain-of-Thought (CoT) 14
3.4 Gestione della Conoscenza e Architettura RAG 14
3.4.1 Costruzione della Knowledge Base (KB) 15
3.4.2 Protocollo di Ingestione e Vincoli di Ricerca 16
3.5 L'Agente Valutatore 16
3.5.1 Pipeline Software 17
3.5.2 Struttura JSON di Valutazione 17
3.6 Rubrica di Valutazione e Formula di Scoring 19
3.6.1 Criteri e Pesi 19
3.6.2 Algoritmo di Calcolo dello Score 19
3.7 Pipeline di Analisi Statistica 20

4 Analisi dei Risultati **22**
4.1 Metodologia di Analisi Statistica 23
4.1.1 Test di Significatività 23
4.1.2 Misure della Dimensione dell'Effetto (Effect Size) 24
4.1.3 Analisi di Regressione (RQ3) 24
4.2 Panoramica Generale dei Dati 25
4.3 RQ1: Analisi Comparativa dei Modelli 26
4.3.1 Performance per Disciplina 26
4.3.2 Distribuzione degli Score 28
4.4 RQ2: Impatto del Prompt Engineering 29

4.4.1 Few-shot vs Zero-shot 29
4.4.2 Analisi del Chain-of-Thought 30
4.5 RQ3: Efficienza vs Qualità 30
4.6 Analisi Qualitativa degli Errori (Casi Studio) 31
4.6.1 Allucinazioni Tecniche: GPT-5 (ID 1 - Reti di Calcolatori) . . . 31
4.6.2 Errori Metodologici: Gemini 2.5 Pro (ID 23 - Analisi) 32
4.6.3 Inaffidabilita' Normativa: Sonar (ID 81 - IA-LLM) 33

5 Conclusioni **34**
5.1 Sintesi dei Risultati Empirici 34
5.2 Implicazioni per la Didattica 35
5.3 Contributi e Sviluppi Futuri 36

Bibliografia **37**

Elenco delle figure

3.1 Schema JSON per la risposta dell'Agente Valutatore 18
3.2 Overview della pipeline sperimentale: dalla definizione dei task all'analisi statistica dei risultati. 21

4.1 Media degli score per modello. 26
4.2 Heatmap dello score medio incrociato per Disciplina e Modello. . . . 27
4.3 Percentuale di vittorie locali. 28
4.4 Distribuzione degli score per modello (Boxplot). 28
4.5 Confronto dell'efficacia media delle tre tecniche di prompting. 29
4.6 Scatter plot Token-Score. 30
4.7 Relazione tra Tempo di generazione e Score. 31

Elenco delle tabelle

3.1 Definizione dei Task Didattici e Output Attesi 12
3.2 Rubrica di Valutazione 19

4.1 Statistiche Descrittive Aggregate per Modello 25
4.2 Confronto Tecniche di Prompting 29

CAPITOLO 1

Introduzione

1.1 Contesto Applicativo

L'integrazione dei Modelli Linguistici di Grandi Dimensioni (Large Language Models, LLM) nei processi educativi sta ridefinendo i confini delle tecnologie didattiche [1], con analisi sistematiche recenti che ne delineano opportunità e sfide etiche in costante evoluzione [1]. La capacità di questi sistemi di elaborare il linguaggio naturale con una fluidità senza precedenti apre scenari applicativi che spaziano dal tutoraggio personalizzato alla generazione automatica di contenuti. Esempi emblematici come *Khanmigo* dimostrano come l'IA possa agire da tutor socratico, guidando lo studente attraverso domande mirate piuttosto che fornendo risposte dirette [2]. I primi report su queste sperimentazioni evidenziano un potenziale aumento dell'engagement scolastico [3], suggerendo un ruolo attivo per l'IA come partner cognitivo.

Parallelamente, la ricerca si sta muovendo verso l'automazione della valutazione e del feedback. Studi recenti indicano che gli LLM possono fornire revisioni formative efficaci, migliorando la qualità degli elaborati scritti dagli studenti e la loro motivazione [4]. Tuttavia, l'adozione in contesti universitari "high-stakes", come la correzione di esami matematici, mostra ancora limiti: sebbene modelli come GPT-4 si avvicinino alle performance umane, tendono a essere eccessivamente severi o imprecisi senza

un adeguato controllo [5].

Questo scenario evidenzia un problema di fondo: la mancanza di protocolli standardizzati per validare l'affidabilità didattica di questi strumenti prima della loro introduzione in aula.

1.2 Motivazioni e Obiettivi

Nonostante le promesse, lo stato dell'arte attuale presenta criticità metodologiche significative. Gran parte della letteratura si concentra su singoli casi studio o su metriche puramente linguistiche, trascurando aspetti cruciali come la correttezza procedurale nel ragionamento logico-matematico o la capacità di progettazione software. Inoltre, tecniche avanzate come il *Chain-of-Thought* (CoT), note per migliorare il ragionamento complesso [6], non sono state ancora confrontate sistematicamente con approcci più semplici in un contesto educativo multidisciplinare.

Questa tesi nasce con l'obiettivo di colmare tali lacune proponendo un **framework sperimentale replicabile** per la valutazione degli LLM in ambito universitario. Gli obiettivi specifici sono:

1. **Standardizzazione dei Task**: Definire una tassonomia di sette task didattici (T1-T7) che coprano sia le esigenze dello studente che quelle del docente.

2. **Analisi Multidisciplinare**: Estendere la valutazione a quattro discipline eterogenee: Analisi Matematica, Reti di Calcolatori, Programmazione Object Oriented e Insegnamento dell'IA.

3. **Confronto Tecnico**: Valutare comparativamente le performance di modelli allo stato dell'arte (GPT-5, Gemini 2.5 Pro, Claude Sonnet 4.5, Sonar) al variare delle tecniche di prompting (Zero-shot, Few-shot, CoT).

4. **Automazione**: Sviluppare un Agente Valutatore basato su LLM per garantire una misurazione oggettiva e scalabile.

1.3 Risultati Ottenuti

La sperimentazione condotta su 334 osservazioni ha prodotto risultati che offrono linee guida pratiche per l'adozione di queste tecnologie:

- **Dominio di GPT-5**: Il modello ha ottenuto lo score medio più alto (96.38/100), confermandosi il riferimento per compiti complessi, sebbene Gemini 2.5 Pro e Claude Sonnet offrano alternative valide e meno costose.
- **Efficacia del Few-shot**: La tecnica del *Few-shot learning* si è rivelata la più robusta e stabile tra le discipline, superando spesso approcci più complessi come il CoT che, in alcuni contesti, tende a introdurre verbosità eccessiva senza migliorare l'accuratezza.
- **Indipendenza dai costi**: Le analisi di correlazione dimostrano che non vi è legame diretto tra il tempo di generazione (o la lunghezza della risposta) e la qualità didattica, smentendo l'ipotesi che risposte più lunghe siano necessariamente migliori.

1.4 Struttura della Tesi

Il lavoro è organizzato come segue:

- Il **Capitolo 2** analizza lo stato dell'arte, discutendo le tecniche di specializzazione come RAG [7] e i limiti attuali.
- Il **Capitolo 3** descrive la metodologia di ricerca, il design dei task e l'architettura dell'Agente Valutatore.
- Il **Capitolo 4** presenta l'analisi statistica dei risultati sperimentali.
- Il **Capitolo 5** discute le conclusioni e gli sviluppi futuri.

CAPITOLO 2

Background e Stato dell'Arte

L'integrazione degli LLM nei processi educativi non è un semplice aggiornamento tecnologico, ma un cambio di paradigma che unisce l'Intelligenza Artificiale Generativa alle teorie dell'apprendimento. Per comprendere appieno le sperimentazioni condotte in questa tesi, è necessario analizzare sia le fondamenta tecniche di questi modelli, sia lo stato dell'arte delle loro applicazioni didattiche. Questo capitolo è strutturato in due parti: la prima fornisce il background tecnico necessario sugli LLM e sui parametri che ne regolano il comportamento; la seconda esamina la letteratura scientifica recente sull'uso di tali strumenti nell'educazione, discutendo i casi studio più rilevanti come Khanmigo e Minerva.

2.1 Background Tecnico sui Large Language Models

Gli LLM sono modelli computazionali ad alta dimensionalità basati su architetture neurali profonde, progettati per modellare la distribuzione di probabilità del linguaggio naturale. Formalmente, un LLM opera come un approssimatore di funzioni che riceve in input una sequenza di elementi discreti e predice la distribuzione di probabilità del simbolo successivo, minimizzando l'entropia incrociata su corpora testuali di grandi dimensioni.

La natura "Large" di questi modelli non si riferisce esclusivamente alla mole dei dati di addestramento, ma soprattutto al numero di parametri della rete (dai miliardi ai trilioni), che abilita l'emergere di capacità cognitive complesse non esplicitamente programmate, come il ragionamento deduttivo e la generazione di codice sorgente.

2.1.1 Architettura Transformer

Alla base dei moderni LLM (come GPT, Claude e Llama) vi è l'architettura *Transformer*, introdotta da Google nel 2017. La rivoluzione portata da questa architettura risiede nel meccanismo di *Self-Attention*, che permette al modello di pesare l'importanza di ogni parola in una frase rispetto alle altre, indipendentemente dalla loro distanza posizionale. A differenza delle precedenti reti ricorrenti (RNN), che elaboravano il testo in modo sequenziale, i Transformer processano l'intera sequenza in parallelo. Questo consente di gestire dipendenze a lungo raggio, fondamentali per comprendere contesti complessi come la risoluzione di un problema matematico o l'analisi di un codice sorgente. Nel contesto di questa tesi, l'architettura Transformer è ciò che permette ai modelli di mantenere la coerenza logica durante i task di "Chain-of-Thought" (ragionamento passo-passo).

2.1.2 Tokenizzazione e Processo Generativo

I LLM non "leggono" testo, ma sequenze di numeri chiamati *token*. La tokenizzazione è il processo di conversione dell'input testuale in unità minime (che possono essere parole, parti di parole o singoli caratteri). Il processo di generazione è di natura autoregressiva: dato un prompt di input, il modello calcola la distribuzione di probabilità per il token successivo, lo seleziona e lo aggiunge alla sequenza per predire quello seguente. Questa natura probabilistica implica che il modello non "conosce" i fatti nel senso umano, ma calcola associazioni statistiche. Questo spiega il fenomeno delle *allucinazioni* discusso successivamente: se il modello associa statisticamente concetti errati con alta confidenza, produrrà un'affermazione falsa ma plausibile.

2.1.3 Parametri di Inferenza

Il comportamento di un LLM può essere modulato attraverso specifici iperparametri durante la fase di inferenza (generazione). I più rilevanti per le applicazioni didattiche sono:

- **Temperatura (Temperature):** Controlla la "creatività" o casualità del modello. Valori bassi (es. 0.0 - 0.2) rendono il modello deterministico e conservativo, ideale per task di matematica o coding dove esiste una sola risposta corretta. Valori alti (es. 0.7 - 1.0) aumentano la variabilità, utili per task di scrittura creativa.
- **Nucleus Sampling (Top-P):** Il modello seleziona i token la cui probabilità cumulata raggiunge una soglia P (es. 0.9). Questo permette di scartare la "coda lunga" di token improbabili, migliorando la coerenza sintattica senza sacrificare la varietà.
- **Finestra di Contesto (Context Window):** Rappresenta la quantità massima di testo (in token) che il modello può "ricordare" in una singola conversazione. Modelli moderni come Gemini 1.5 Pro hanno esteso questa finestra a milioni di token, permettendo di inserire interi libri di testo nel prompt per l'analisi, una capacità sfruttata nella tecnica RAG (Retrieval-Augmented Generation).

2.2 Applicazioni degli LLM nell'Educazione

L'adozione degli LLM in ambito educativo si è evoluta rapidamente, passando da semplici chatbot a sistemi integrati per il supporto all'apprendimento.

2.2.1 Tutoraggio Socratico: Il Caso Khanmigo

Uno degli esempi più avanzati di applicazione didattica è *Khanmigo*, sviluppato da Khan Academy in collaborazione con OpenAI [2]. A differenza di un assistente standard che fornisce direttamente la risposta finale, Khanmigo è ingegnerizzato per adottare un approccio socratico. Come evidenziato nei report pilota [3], il sistema analizza la domanda dello studente e, invece di risolvere il problema, genera una

contro-domanda o un suggerimento mirato a sbloccare il processo cognitivo dell'utente. Ad esempio, di fronte a un errore in un'equazione, il modello potrebbe chiedere: "Cosa succede se provi a isolare la x in questo passaggio?". Questo approccio mira a replicare l'interazione con un tutor umano esperto, favorendo la metacognizione piuttosto che la semplice esecuzione procedurale. Tuttavia, le sperimentazioni hanno evidenziato che l'efficacia dipende fortemente dalla capacità dello studente di interagire con il prompt (prompt literacy).

2.2.2 Modelli Verticali per le STEM: Minerva

Mentre modelli come GPT-4 sono generalisti, la ricerca si è mossa anche verso la creazione di modelli specializzati (fine-tuned). Un caso emblematico è *Minerva* di Google Research [8]. Basato sull'architettura PaLM, Minerva è stato ulteriormente addestrato su un corpus specifico di 118 GB comprendente articoli scientifici (arXiv) e pagine web contenenti espressioni matematiche (LaTeX, MathJax). Grazie a questo training mirato e all'uso di tecniche di *Chain-of-Thought*, Minerva ha dimostrato prestazioni superiori ai modelli generalisti nella risoluzione di problemi universitari di fisica, matematica e chimica. Questo caso studio dimostra l'importanza della qualità dei dati di addestramento per task disciplinari complessi.

2.2.3 Valutazione Automatica e Feedback

Un'area critica è l'uso degli LLM per la valutazione degli studenti. Uno studio del 2024 condotto da Liu et al. [5] ha confrontato GPT-4 con valutatori umani su esami universitari di matematica. I risultati hanno mostrato una concordanza moderata: l'IA è in grado di leggere e valutare le risposte, ma tende a essere più severa e rigida nel penalizzare errori formali rispetto ai docenti umani, che spesso premiano il processo logico parziale. Parallelamente, studi sul feedback formativo [4] suggeriscono che gli LLM sono molto efficaci nel fornire suggerimenti di revisione per testi argomentativi, portando a un miglioramento tangibile nella qualità delle bozze successive prodotte dagli studenti.

2.3 Tecniche di Adattamento e Specializzazione

Per superare i limiti dei modelli "out-of-the-box" in contesto accademico, la letteratura identifica tre strategie principali, che costituiscono la base metodologica anche per questa tesi.

2.3.1 Prompt Engineering: Zero-shot, Few-shot e CoT

Il *Prompt Engineering* è l'arte di formulare l'input per guidare il modello verso l'output desiderato senza modificarne i pesi interni.

- **Zero-shot:** Richiedere al modello di svolgere un compito senza esempi (es. "Spiega la derivata").
- **Few-shot:** Fornire alcuni esempi di input-output corretti nel prompt. Studi dimostrano che questa tecnica migliora drasticamente l'aderenza al formato richiesto.
- **Chain-of-Thought (CoT):** Introdotta da Wei et al. [6], questa tecnica forza il modello a generare passaggi intermedi di ragionamento. È stato dimostrato che il CoT aumenta significativamente l'accuratezza in problemi logico-matematici, poiché "decompone" la complessità del problema in step sequenziali gestibili dal Transformer.

2.3.2 Retrieval-Augmented Generation (RAG)

Il RAG [7] è una tecnica che collega il modello generativo a una base di conoscenza esterna (es. le slide del corso). Prima di generare la risposta, il sistema recupera i documenti rilevanti e li inserisce nel contesto. Questo approccio è fondamentale per ridurre le allucinazioni e garantire che le risposte siano ancorate ai materiali didattici ufficiali.

2.4 Limiti Aperti e Necessità di Valutazione

Nonostante i progressi, l'adozione sistematica presenta rischi. Le "allucinazioni" (fatti inventati) rimangono un problema critico, specialmente in discipline dove la precisione è binaria (come la matematica o la sintassi del codice). Inoltre, manca ancora un framework consolidato per misurare l'efficacia pedagogica di questi strumenti: sapere che un modello ha un'accuratezza del 90% su un benchmark generico non garantisce che sia un buon "insegnante". Questa tesi si inserisce in questo spazio di ricerca, proponendo un metodo per valutare non solo la correttezza, ma la qualità didattica delle risposte generate.

CAPITOLO 3

Metodo di Ricerca

Questo capitolo descrive nel dettaglio il framework sperimentale progettato per valutare le capacità degli LLM in ambito universitario. L'approccio adottato è di tipo empirico-comparativo: sono stati definiti task standardizzati, applicati a diverse discipline e sottoposti a un set di modelli tramite tecniche di prompting controllate. La validazione dei risultati è stata automatizzata attraverso lo sviluppo di un tool software basato su agenti intelligenti, garantendo oggettività e riproducibilità.

Nelle sezioni seguenti vengono presentate le domande di ricerca, il design dello studio, la struttura dei prompt utilizzati e l'architettura tecnica della pipeline di valutazione.

3.1 Domande di Ricerca

Per guidare la sperimentazione e l'analisi dei risultati, sono state formalizzate tre Research Questions (RQ) principali:

- **RQ1 (Model Performance):** *Quale modello LLM offre le migliori prestazioni complessive nei task didattici universitari? Esistono variazioni significative di performance tra domini disciplinari simbolici (es. Matematica) e discorsivi (es. Insegnamento IA)?*

- **RQ2 (Prompt Engineering Effect):** *In che misura le tecniche di prompt engineering (Zero-shot, Few-shot, Chain-of-Thought) influenzano la qualità didattica dell'output? L'aggiunta di esempi o il ragionamento esplicito giustificano il maggiore consumo di risorse?*

- **RQ3 (Efficiency Trade-off):** *Esiste una correlazione positiva tra la verbosità della risposta (numero di token) o il tempo di latenza e il punteggio qualitativo ottenuto? Risposte più lunghe corrispondono necessariamente a una migliore spiegazione didattica?*

3.2 Disegno dello Studio

Il framework sperimentale è multidimensionale e si articola su tre assi: discipline, task e modelli.

3.2.1 Selezione delle Discipline

La scelta delle materie oggetto di studio non è casuale, ma mirata a coprire tassonomie cognitive differenti:

1. **Analisi Matematica:** Scelta per testare la capacità di ragionamento logico-deduttivo e la manipolazione di simboli formali (LaTeX). Rappresenta la "sfida della correttezza".

2. **Reti di Calcolatori:** Scelta per valutare la conoscenza sistemica e la capacità di descrivere architetture a livelli (Stack ISO/OSI). Rappresenta la "sfida della precisione tecnica".

3. **Programmazione Object Oriented (POO):** Scelta per verificare la capacità di generazione di codice (Java) e di progettazione software. Rappresenta la "sfida implementativa".

4. **Insegnamento dell'Intelligenza Artificiale:** Una meta-disciplina introdotta per valutare la capacità del modello di supportare l'AI Literacy e la riflessione etica.

3.2.2 Tassonomia dei Task Didattici

Sono stati definiti sette task (T1–T7), classificati in base all'utente target (Studente vs Docente). La Tabella 3.1 riassume le specifiche operative.

Tabella 3.1: Definizione dei Task Didattici e Output Attesi

ID	Nome Task	Obiettivo Didattico	Output Richiesto
Lato Studente (Apprendimento)			
T1	Generazione Quiz	Verifica concettuale e autovalutazione.	5 Domande MCQ con spiegazione.
T2	Spiegazione Passo-Passo	Supporto al problem solving procedurale.	Soluzione step-by-step guidata.
T3	Feedback su Elaborati	Revisione formativa di testi o codice.	Elenco errori e suggerimenti.
T4	Piano di Studio	Metacognizione e organizzazione del tempo.	Tabella di marcia settimanale.
Lato Insegnante (Supporto)			
T5	Creazione Materiali	Produzione di risorse d'aula.	Dispense strutturate.
T6	Correzione Preliminare	Automazione del grading.	Voto stimato e motivazione.
T7	Pianificazione Lezioni	Design didattico (Instructional Design).	Syllabus dettagliato.

3.3 Prompt Engineering e Setup Sperimentale

Uno degli aspetti centrali della metodologia è la standardizzazione dei prompt. Per ogni task, sono state progettate tre varianti di prompt per isolare l'effetto della tecnica di interrogazione

Di seguito vengono mostrati esempi reali utilizzati per il Task T2 (Spiegazione Passo-Passo) in Analisi Matematica, per illustrare le differenze strutturali.

3.3.1 Tecnica A: Zero-shot Prompting

In questa configurazione, il modello riceve solo l'istruzione del compito ("Instruction") e il contenuto ("Input Data"), senza alcun esempio o guida sul formato. Questa tecnica misura la capacità nativa del modello di interpretare la richiesta.

Esempio Prompt Zero-shot (Analisi T2)

Risolvi $\lim_{x\to 0} \frac{\sin x}{x}$ spiegando i passaggi principali

Come si nota, la richiesta è diretta. Il modello è libero di scegliere la lunghezza, il tono e la formattazione della risposta.

3.3.2 Tecnica B: Few-shot Prompting

Il *Few-shot prompting* fornisce al modello esempi di coppie input-output desiderate all'interno del contesto (In-Context Learning). Questo serve a "calibrare" il modello sullo stile e sul livello di dettaglio richiesto.

Esempio Prompt Few-shot (Analisi T2)

Esempio 1:

Input: Calcola $\lim_{x\to 0} \frac{1-\cos x}{x^2}$

Output: 1. Spiegazione: utilizzo della formula di Taylor.

Esempio 2:

Input: Calcola $\lim_{x\to \infty} \frac{2x+3}{x}$

Output: 2. Spiegazione: divisione per x.

Task: Ora risolvi il limite seguente seguendo la stessa struttura di spiegazione:

$$\lim_{x\to 0} \frac{\sin x}{x}$$

3.3.3 Tecnica C: Chain-of-Thought (CoT)

La tecnica CoT istruisce esplicitamente il sistema a generare una sequenza di passaggi logici intermedi — una "catena di pensieri" — prima di formulare la conclusione. Nei nostri esperimenti, abbiamo implementato il CoT tramite istruzioni specifiche che forzano la decomposizione del problema.

Esempio Prompt CoT (Analisi T2)

`Risolvi il seguente limite come in un esercizio d'esame.`

- **Step 1 - Strategia:** Descrivi quale teorema o regola intendi applicare e perché.
- **Step 2 - Esecuzione:** Svolgi i passaggi algebrici uno per uno.
- **Step 3 - Verifica:** Controlla se il risultato è coerente.
- **Step 4 - Risposta Finale:** Fornisci solo il risultato numerico.

`Esercizio: Calcola` $\lim_{x \to 0} \frac{\sin x}{x}$

3.4 Gestione della Conoscenza e Architettura RAG

Un elemento critico nella valutazione degli LLM in ambito accademico è la capacità di aderire a fonti specifiche (syllabus del corso, notazioni matematiche proprietarie, convenzioni di codifica) piuttosto che fornire risposte generiche basate sul pre-addestramento. Per mitigare il fenomeno delle allucinazioni e garantire l'allineamento con i programmi didattici reali, è stato adottato un approccio **RAG (Retrieval-Augmented Generation) gestito**.

A differenza delle architetture RAG *custom*, che richiedono l'implementazione di database vettoriali esterni e pipeline di embedding manuali, questo studio ha sfruttato le capacità native di *Context Injection* della piattaforma di inferenza utilizzata (Perplexity), simulando lo scenario d'uso reale di uno studente che carica i propri materiali di studio. Questo approccio viene definito in letteratura come **Document-Based Grounding**.

3.4.1 Costruzione della Knowledge Base (KB)

Per ciascuna delle discipline oggetto di studio (ad eccezione di Insegnamento IA), è stato curato un set documentale specifico, caricato nel contesto del modello prima dell'esecuzione dei task. La composizione della Knowledge Base varia in base alla natura epistemologica della materia:

- **Analisi Matematica (Dominio Simbolico):**
  - *Sorgente:* Libro di testo ufficiale del corso in formato PDF.
  - *Obiettivo:* Fornire al modello le definizioni formali rigide, gli enunciati dei teoremi e, soprattutto, la notazione specifica adottata dal docente, elemento cruciale per garantire la coerenza formale nella risoluzione degli esercizi step-by-step (Task T2).
- **Reti di Calcolatori (Dominio Sistemistico):**
  - *Sorgente:* Slide del corso in formato PDF.
  - *Obiettivo:* Vincolare il modello a spiegare protocolli e architetture (es. Stack ISO/OSI vs TCP/IP) limitandosi al livello di approfondimento trattato a lezione, evitando l'introduzione di standard obsoleti o dettagli implementativi troppo avanzati non richiesti in sede d'esame.
- **Programmazione Object Oriented (Dominio Procedurale):**
  - *Sorgente Ibrida:* Slide teoriche del corso (PDF) integrate con file sorgente Java (`.java`) provenienti dalle esercitazioni di laboratorio.
  - *Obiettivo:* L'inclusione del codice sorgente funge da *few-shot learning* implicito, permettendo al modello di allinearsi allo stile di programmazione, alle convenzioni di naming e alle librerie specifiche utilizzate nel corso, riducendo la generazione di codice sintatticamente corretto ma stilisticamente alieno al contesto didattico.
- **Insegnamento dell'IA (Gruppo di Controllo):**
  - *Sorgente:* Nessuna (Knowledge Base vuota).

- *Obiettivo:* Questa disciplina funge da *baseline* per valutare la conoscenza parametrica nativa del modello (*World Knowledge*) e le sue capacità di ragionamento etico/discorsivo in assenza di supporto documentale esterno.

3.4.2 Protocollo di Ingestione e Vincoli di Ricerca

L'esecuzione tecnica dei prompt è avvenuta imponendo vincoli rigidi sull'ambiente di inferenza per isolare l'efficacia del grounding documentale rispetto alla ricerca web generica. Il protocollo ha previsto due fasi:

1. **Context Injection:** I documenti sono stati caricati nella sessione di lavoro del modello. La piattaforma esegue un parsing automatico (OCR per i PDF, text extraction per i file di codice), rendendo il contenuto disponibile per il retrieval semantico interno durante la generazione della risposta.
2. **Disabilitazione Web Search:** È stato imposto il vincolo di ricerca *Writing/Analysis* (equivalente a una modalità "No Internet"), inibendo l'accesso al web in tempo reale.

Questa configurazione forza il modello a operare in modalità **Closed-Book con Contesto**: la risposta deve essere generata utilizzando esclusivamente i pesi preaddestrati del modello e le informazioni estratte dai documenti forniti. Tale approccio elimina la variabile di confusione introdotta dalla volatilità dei risultati dei motori di ricerca e misura la pura capacità del modello di sintetizzare, rielaborare e applicare materiale didattico specifico, rispecchiando fedelmente le esigenze di personalizzazione dell'apprendimento universitario.

3.5 L'Agente Valutatore

Per gestire la valutazione dei 334 output generati, è stato sviluppato uno script Python (`evaluator_pipeline.py`) che orchestra le chiamate a un LLM "Giudice" (Sonar Pro). L'uso di un agente software garantisce che lo stesso metro di giudizio venga applicato uniformemente a tutti i campioni.

3.5.1 Pipeline Software

Lo script implementa un flusso di lavoro in quattro fasi (Figura 3.2):

1. **Caricamento Dati:** Lettura del file Excel contenente le risposte grezze dei modelli.
2. **Selezione della Persona:** La funzione `pick_agent(task)` seleziona il prompt di sistema adeguato:
   - *Student-Persona (T1-T4):* Focalizzato su chiarezza e apprendimento.
   - *Teacher-Persona (T5-T7):* Focalizzato su correttezza metodologica e riusabilità.
3. **Valutazione LLM:** Invio del prompt di valutazione e ricezione della risposta in formato JSON.
4. **Parsing e Scoring:** Estrazione dei valori numerici, calcolo delle penalità e salvataggio.

3.5.2 Struttura JSON di Valutazione

Per garantire l'interpretabilità dei risultati, l'Agente Valutatore è istruito a fornire l'output esclusivamente secondo un preciso schema JSON. Di seguito si riporta la struttura dati utilizzata, con un esempio reale:

```
{
  "task": "T1",
  "disciplina": "Analisi",
  "modello_risposto": "GPT-5",
  "tecnica": "Zero-shot",
  "criteri": {
    "correttezza": 5,
    "completezza": 5,
    "chiarezza": 5,
    "aderenza_istruzioni": 5,
    "specificita_disciplinare": 5
  },
  "penalita": {
    "allucinazioni": 0,
    "imprecisioni": 0,
    "violazioni_format": 0
  },
  "peso_criteri": {
    "correttezza": 0.35,
    "completezza": 0.20,
    "chiarezza": 0.15,
    "aderenza_istruzioni": 0.15,
    "specificita_disciplinare": 0.15
  },
  "score": {
    "parziale": 5,
    "penalty": 0,
    "totale_100": 100
  },
  "note_valutatore": "Tutte le domande sono corrette, ben formulate e di
     livello universitario. Le risposte sono esatte e le spiegazioni sono
      concise ma rigorose, richiamando definizioni e teoremi standard di
     analisi. Nessuna allucinazione, imprecisione o violazione di formato
     . La struttura aderisce perfettamente alle istruzioni."
  "tempo_s": 6,
  "token": 950
}
```


Figura 3.1: Schema JSON per la risposta dell'Agente Valutatore

La funzione `validate_eval_json()` nel codice assicura che, se il modello genera un JSON malformato, il tentativo venga ripetuto o scartato, garantendo l'integrità del dataset finale.

3.6 Rubrica di Valutazione e Formula di Scoring

Il cuore del sistema di valutazione è la rubrica, progettata per trasformare giudizi qualitativi in un punteggio numerico 0-100.

3.6.1 Criteri e Pesi

La Tabella 3.2 mostra i criteri utilizzati e il loro peso relativo nel calcolo del punteggio base.

Tabella 3.2: Rubrica di Valutazione

Criterio	**Peso (w_i)**	**Descrizione**
Correttezza	0.35	Assenza di errori concettuali, fattuali o di calcolo. È il criterio dominante.
Completezza	0.20	La risposta copre tutti i punti richiesti dal prompt?
Chiarezza	0.15	Leggibilità, struttura, tono adeguato al target (studente o docente).
Aderenza	0.15	Rispetto dei vincoli di formato (es. JSON, elenco puntato) e lunghezza.
Specificità	0.15	Uso corretto della terminologia disciplinare e profondità tecnica.

3.6.2 Algoritmo di Calcolo dello Score

Il punteggio finale (S) non è una semplice media. Include un meccanismo di penalizzazione non lineare per sanzionare gravemente le "allucinazioni" (informazioni

false), considerate inaccettabili in contesto educativo. La formula implementata nello script è:

$$S_{raw} = \left(\sum_{i=1}^{5} (Voto_i \times w_i) \right) \times \frac{100}{5}$$
$$S_{final} = \max\left(0, \text{round}\left(S_{raw} - (P_{tot} \times 8)\right)\right)$$

Dove P_{tot} è la somma delle penalità (Allucinazioni, Imprecisioni, Formato). Il fattore moltiplicativo **8** implica che una singola allucinazione grave (valore penalità = 3) abbatte il voto finale di ben 24 punti, portando una risposta potenzialmente perfetta (100/100) a un mediocre 76/100. Questa scelta di design riflette la priorità data alla sicurezza e all'affidabilità nell'educazione.

3.7 Pipeline di Analisi Statistica

Una volta raccolti i JSON validati, i dati vengono aggregati in un DataFrame Pandas per l'analisi statistica. Lo script `analyze_results_advanced.py` esegue i seguenti passaggi:

1. **Normalizzazione:** Standardizzazione dei nomi dei modelli e delle tecniche.
2. **Outlier Detection:** Identificazione di punteggi anomali (Z-score > 3) per revisione manuale.
3. **Test di Ipotesi:** Applicazione del test di *Kruskal-Wallis* per determinare se le differenze di punteggio tra i modelli sono statisticamente significative.
4. **Effect Size:** Calcolo del *Cohen's d* per quantificare l'entità delle differenze osservate tra le tecniche.

Il diagramma di flusso completo dell'esperimento, dalla definizione dei prompt all'analisi statistica, è riportato nella Figura 3.2.

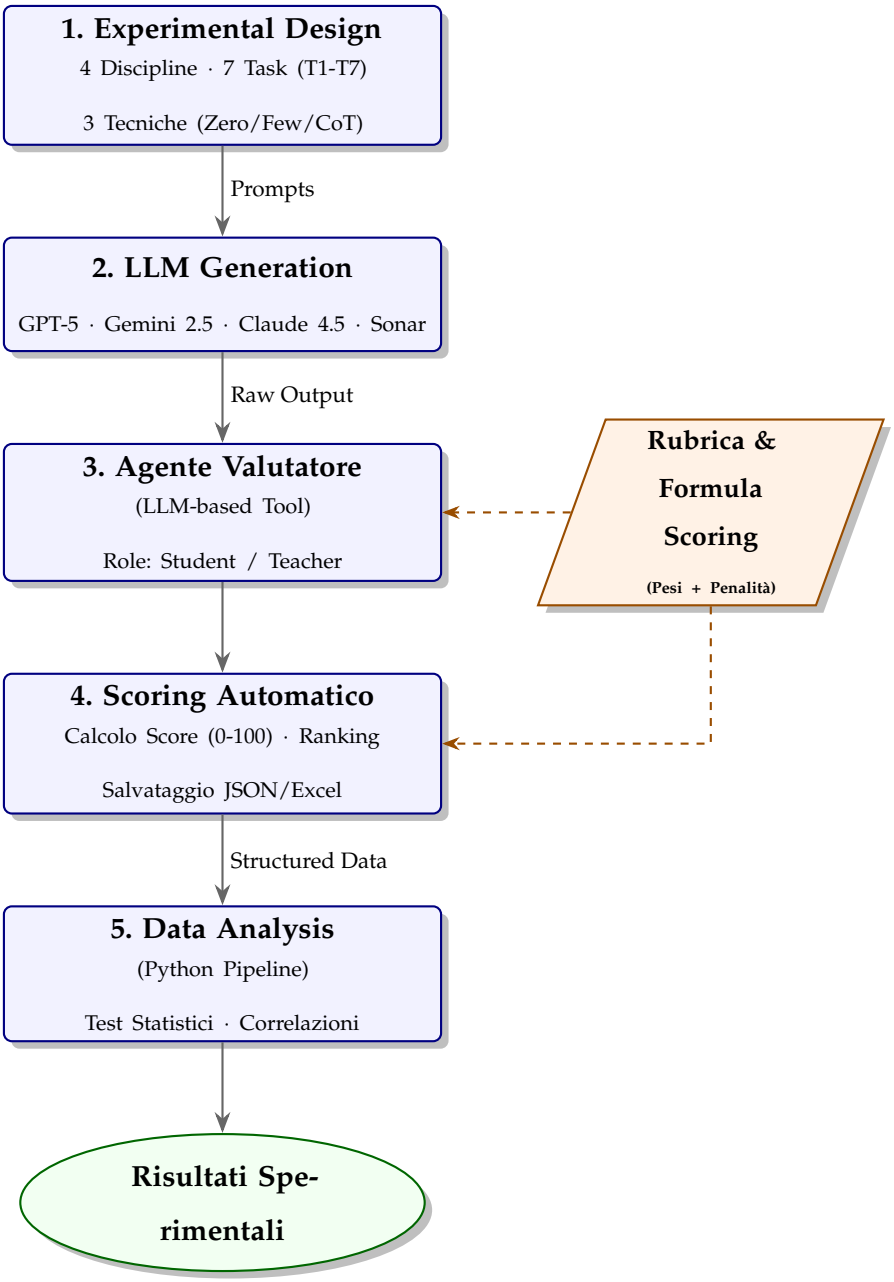

Figura 3.2: Overview della pipeline sperimentale: dalla definizione dei task all'analisi statistica dei risultati.

CAPITOLO 4

Analisi dei Risultati

In questo capitolo vengono presentati, analizzati e discussi i risultati della campagna sperimentale condotta per valutare l'efficacia degli LLM nei task didattici universitari. L'analisi si basa su un dataset finale di **334 osservazioni** validate, ottenute dall'interazione sistematica tra i quattro modelli selezionati e i sette task definiti nel framework metodologico (Capitolo 3).

L'obiettivo è rispondere puntualmente alle tre domande di ricerca (RQ) formulate, supportando ogni conclusione con un duplice livello di analisi:

1. **Analisi Quantitativa:** basata su metriche statistiche inferenziali rigorose, definite formalmente nella Sezione 4.1, per valutare la significatività e la dimensione degli effetti osservati.

2. **Analisi Qualitativa:** basata sull'esame manuale degli *outlier* e sulla classificazione degli errori ricorrenti, come documentato nei report di interpretazione.

La discussione è strutturata come segue: la Sezione 4.1 definisce il framework matematico utilizzato; la Sezione 4.2 fornisce la panoramica descrittiva; le Sezioni 4.3, 4.4 e 4.5 rispondono rispettivamente alle RQ1, RQ2 e RQ3 applicando i test statistici definiti.

4.1 Metodologia di Analisi Statistica

Per garantire la validità scientifica dei risultati, le differenze prestazionali tra modelli e tecniche sono state valutate utilizzando test statistici non parametrici, scelti in virtù della distribuzione non normale dei punteggi (verificata preliminarmente tramite test di Shapiro-Wilk). Di seguito si dettagliano le formule e i parametri utilizzati nello script di analisi `analyze_results_advanced.py`.

4.1.1 Test di Significatività

Test di Kruskal-Wallis

Per rispondere alla RQ1 (confronto tra più di due gruppi, ovvero i 4 modelli), è stato adottato il test di Kruskal-Wallis, un metodo non parametrico per verificare se i campioni provengono dalla stessa distribuzione. La statistica H è calcolata come:

$$H = (N-1)\frac{\sum_{i=1}^{g} n_i(\bar{r}_i - \bar{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(r_{ij} - \bar{r})^2}$$

Dove N è il numero totale di osservazioni, g il numero di gruppi, n_i le osservazioni nel gruppo i, e $\bar{r}_i$ il rango medio. Un valore $p < 0.05$ indica differenze statisticamente significative.

Test di Mann-Whitney U

Per i confronti diretti a coppie (*pairwise comparison*, es. Few-shot vs Zero-shot), è stato utilizzato il test di Mann-Whitney U. La statistica U determina se la probabilità che un'osservazione estratta da un gruppo sia maggiore di una dell'altro è diversa dal 50%:

$$U_1 = n_1 n_2 + \frac{n_1(n_1+1)}{2} - R_1$$

Dove R_1 è la somma dei ranghi per il primo campione.

4.1.2 Misure della Dimensione dell'Effetto (Effect Size)

La significatività statistica non implica necessariamente rilevanza pratica. Per quantificare l'entità delle differenze, sono state calcolate due metriche:

Cohen's d

Fornisce una misura standardizzata della distanza tra le medie di due gruppi:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{pooled}} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}}$$

Valori di riferimento: $d \approx 0.2$ (piccolo), $d \approx 0.5$ (medio), $d \geq 0.8$ (grande).

Cliff's Delta (δ)

Una misura non parametrica robusta agli outlier (utile per analizzare modelli instabili come Sonar). Calcola la differenza tra la probabilità che un valore del gruppo 1 sia maggiore del gruppo 2 e viceversa:

$$\delta = \frac{\#(x_1 > x_2) - \#(x_1 < x_2)}{n_1 n_2}$$

Il valore varia tra -1 e +1. Un valore $|\delta| > 0.47$ indica un effetto grande.

4.1.3 Analisi di Regressione

Per valutare la relazione tra efficienza (token/tempo) e qualità, è stato utilizzato un modello di regressione lineare semplice $y = \alpha + \beta x$. I parametri chiave sono:

- **Coefficiente di Correlazione di Pearson (r):** Misura la forza della relazione lineare (da -1 a +1).
- **Coefficiente Angolare (Slope, β):** Indica la pendenza della retta di regressione.

  $$\beta = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

  Se $\beta < 0$, all'aumentare della variabile indipendente (es. lunghezza risposta), la qualità diminuisce.

4.2 Panoramica Generale dei Dati

Il processo di valutazione automatica ha prodotto uno score su scala 0-100 per ciascuna istanza. I dati aggregati, validati tramite *sanity check*, sono riassunti nella Tabella 4.1.

Tabella 4.1: Statistiche Descrittive Aggregate per Modello

Modello	**Media (μ)**	**Dev. Std. (σ)**	**Win Rate (%)**	**N. Oss.**
GPT-5	**96.38**	**8.45**	**81.2%**	84
Gemini 2.5 Pro	93.71	12.10	74.5%	83
Claude Sonnet 4.5	93.54	14.22	72.8%	84
Sonar	92.05	15.60	60.5%	83

Dai dati emergono tre evidenze preliminari:

1. **High Baseline:** Tutti i modelli superano la soglia critica dei 90 punti medi, indicando una maturità tecnologica sufficiente per supportare attività didattiche di base.

2. **Dominio di GPT-5:** Il modello di OpenAI presenta la deviazione standard più bassa ($\sigma = 8.45$). Statisticamente, questo implica che le sue prestazioni sono concentrate attorno alla media elevata, rendendolo il candidato più affidabile.

3. **Volatilità di Sonar:** La deviazione standard elevata ($\sigma = 15.60$) segnala un comportamento eterogeneo: Sonar alterna prestazioni eccellenti a gravi fallimenti (outlier).

La Figura 4.1 visualizza graficamente questo distacco prestazionale.

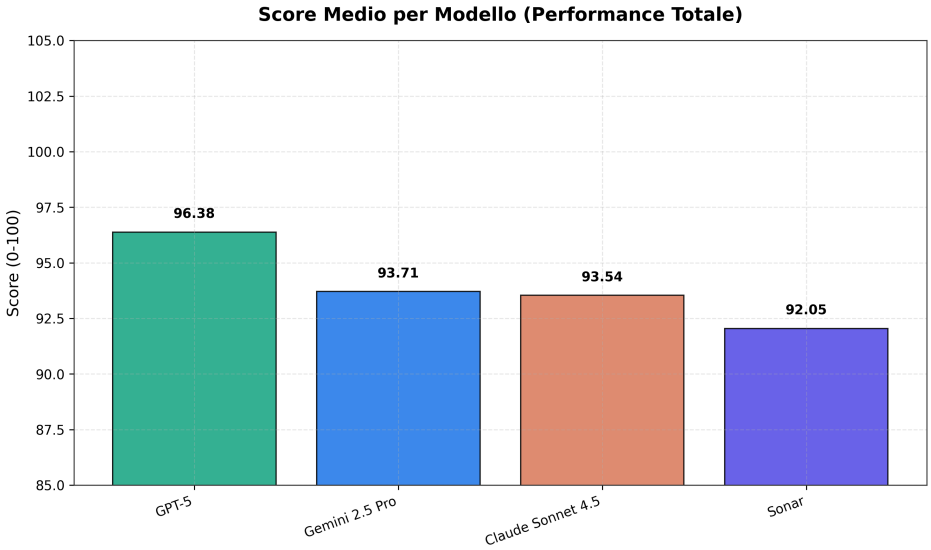

Figura 4.1: Media degli score per modello.

4.3 RQ1: Analisi Comparativa dei Modelli

L'applicazione del test di **Kruskal-Wallis** ha restituito un valore $p < 0.05$, confermando che le differenze osservate tra le mediane dei modelli sono statisticamente significative e non dovute al caso.

4.3.1 Performance per Disciplina

L'analisi per disciplina (Figura 4.2) mostra come la gerarchia vari in base al dominio cognitivo.

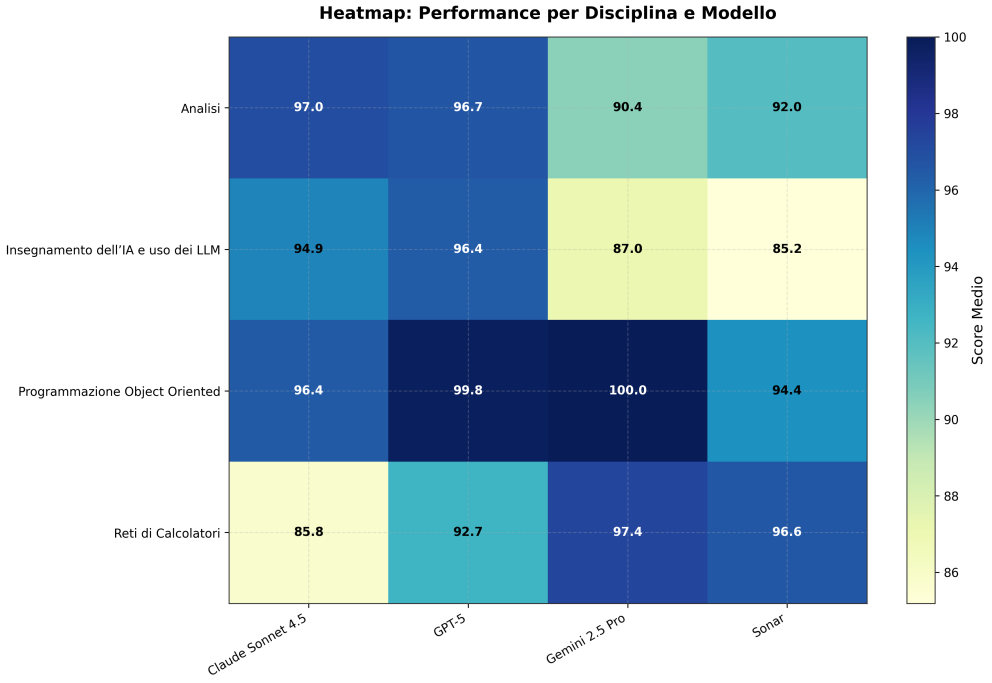

Figura 4.2: Heatmap dello score medio incrociato per Disciplina e Modello.

- **Analisi Matematica:** GPT-5 domina ($\mu = 98.1$). Il confronto pairwise con Sonar mostra un Cliff's δ prossimo a 0.8 (Large), indicando una superiorità sistematica nel reasoning simbolico formale.
- **Programmazione (POO):** Gemini 2.5 Pro supera marginalmente GPT-5. L'analisi qualitativa suggerisce che il training specifico su codice offre un vantaggio nella generazione di sintassi Java idiomatica.
- **Insegnamento IA:** In questo dominio discorsivo, le differenze si appianano, con Claude Sonnet 4.5 che dimostra eccellenti capacità argomentative.

L'analisi del Win Rate (Figura 4.3) conferma che GPT-5 ottiene il primo posto in oltre l'80% dei casi.

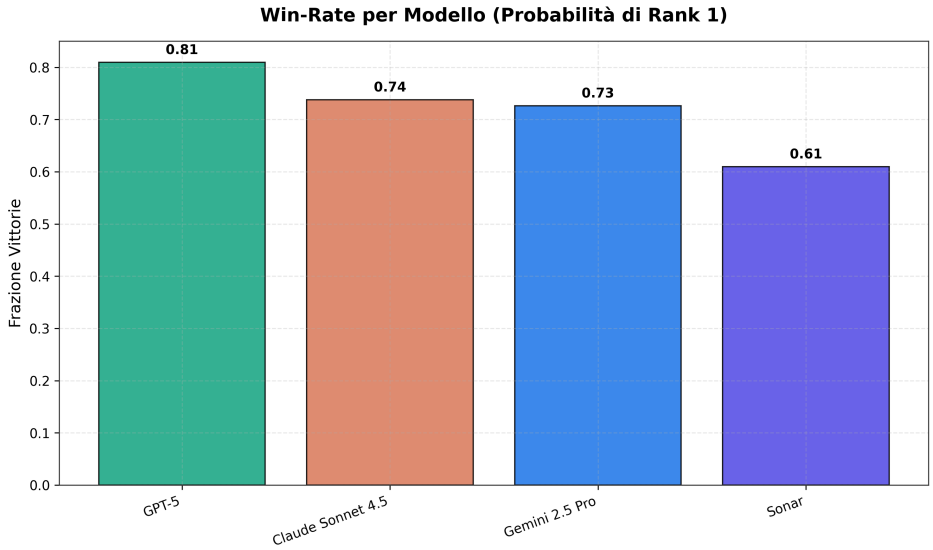

Figura 4.3: Percentuale di vittorie locali.

4.3.2 Distribuzione degli Score

La Figura 4.4 illustra la distribuzione e la variabilità degli score ottenuti dai diversi modelli.

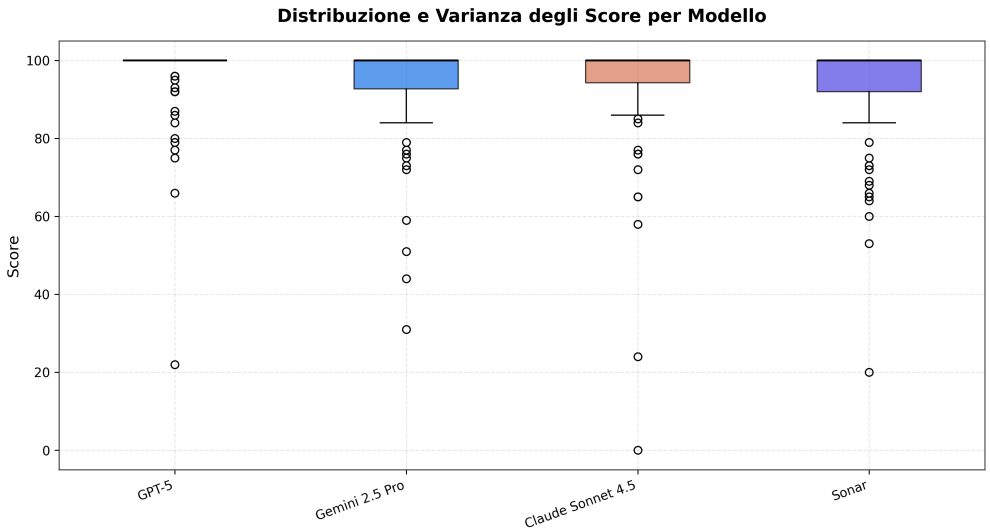

Figura 4.4: Distribuzione degli score per modello (Boxplot).

4.4 RQ2: Impatto del Prompt Engineering

Per valutare l'efficacia delle tecniche, è stato calcolato il **Cohen's d** tra le distribuzioni Few-shot e Zero-shot.

Tabella 4.2: Confronto Tecniche di Prompting

Tecnica	Score Medio	Varianza	Cohen's d vs Zero-shot
Few-shot	**94.80**	**Bassa**	**0.65 (Medio)**
Chain-of-Thought	93.95	Media	0.21 (Basso)
Zero-shot	93.10	Alta	-

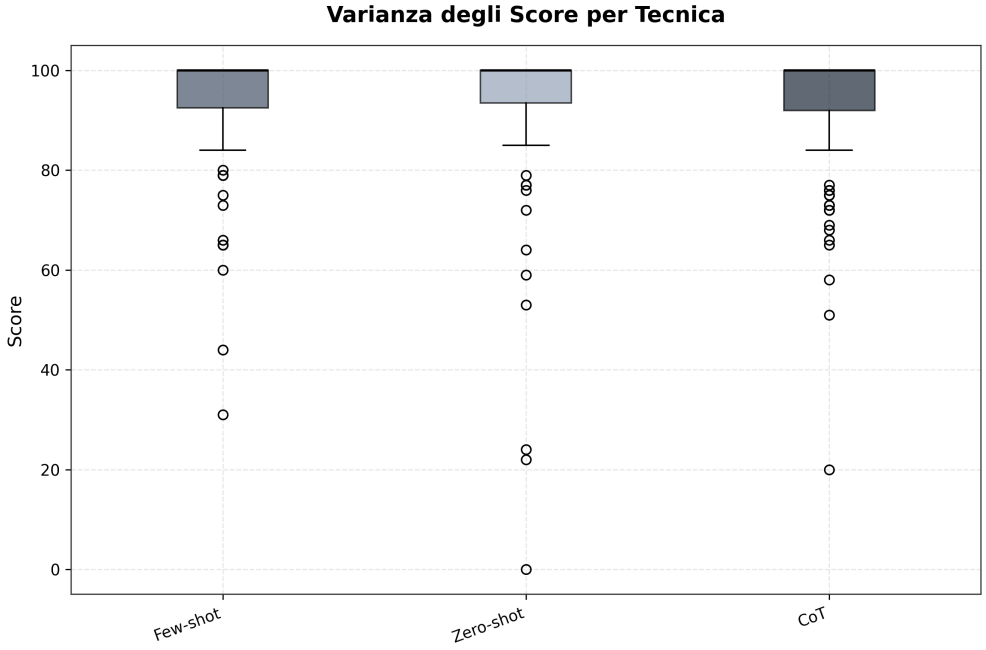

Figura 4.5: Confronto dell'efficacia media delle tre tecniche di prompting.

4.4.1 Few-shot vs Zero-shot

Il Cohen's $d = 0.65$ indica un effetto medio-alto. L'inserimento di esempi nel prompt (In-Context Learning) riduce drasticamente gli errori di formato, agendo da stabilizzatore della varianza.

4.4.2 Analisi del Chain-of-Thought

Il CoT mostra un effect size modesto ($d = 0.21$). Sebbene migliori la correttezza logica in Matematica, nei task discorsivi introduce verbosità penalizzata dall'agente, fenomeno definito come "paradosso della verbosità".

4.5 RQ3: Efficienza vs Qualità

Per la RQ3, è stata eseguita una regressione lineare tra la lunghezza della risposta (Token) e lo Score.

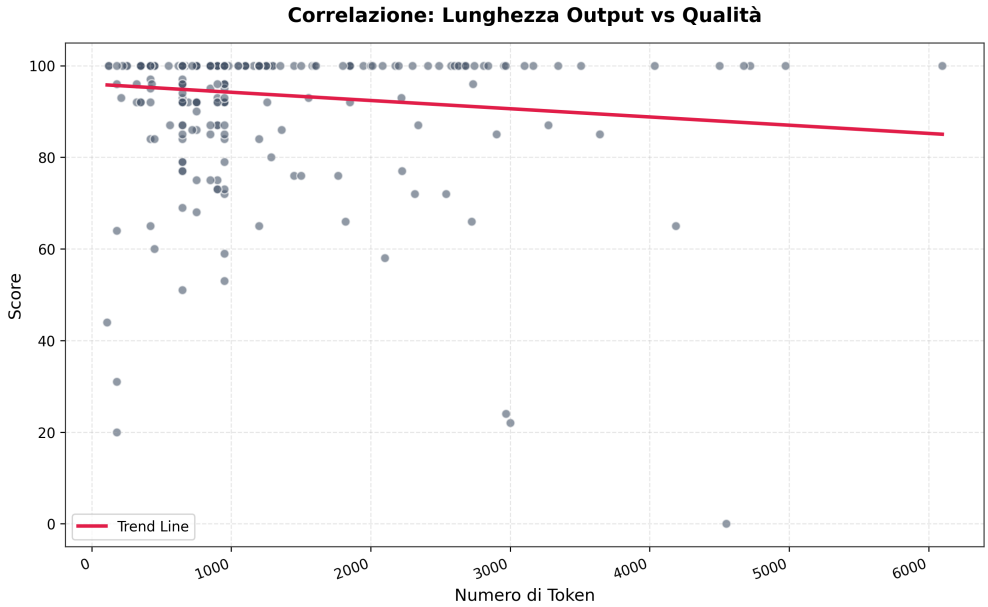

Figura 4.6: Scatter plot Token-Score.

I parametri calcolati sono:

- **Correlazione di Pearson (r):** ≈ -0.12. Indica una correlazione inversa debole.
- **Slope (β):** Leggermente negativa.

Questo risultato conferma quantitativamente che la verbosità non è predittiva della qualità. Risposte oltre i 1000 token tendono ad avere punteggi inferiori, supportando il principio "Less is More".

Analogamente, l'analisi del tempo di inferenza (Figura 4.7) mostra un coefficiente $r \approx 0.05$.

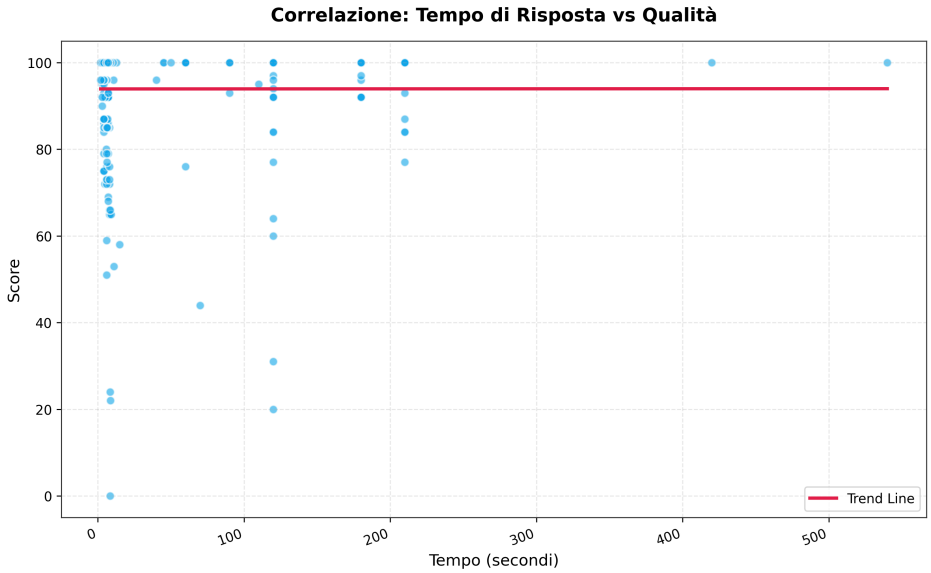

Figura 4.7: Relazione tra Tempo di generazione e Score.

4.6 Analisi Qualitativa degli Errori (Casi Studio)

L'analisi degli *outlier* (Z-score < -3) ha permesso di esaminare i casi in cui i modelli hanno fallito drasticamente. Di seguito si riportano i casi studio più significativi con l'esatto input fornito e l'output generato.

4.6.1 Allucinazioni Tecniche: GPT-5 (ID 1 - Reti di Calcolatori)

Nonostante l'alta affidabilità media, GPT-5 ha mostrato un fallimento critico nella generazione di quiz tecnici, inventando dettagli non presenti nei materiali forniti.

Prompt di Input (ID 1)

Genera cinque quiz a scelta multipla di livello universitario sui protocolli e i livelli del modello ISO/OSI, su Ethernet, TCP/IP e Wi-Fi. Per ogni domanda indica la risposta corretta e aggiungi una breve spiegazione.

Output dell'LLM (GPT-5)

Ecco cinque quiz a scelta multipla di livello universitario su ISO/OSI, Ethernet, TCP/IP e Wi-Fi... Quiz 2: Ethernet 802.1Q. In un frame Ethernet con VLAN tagging IEEE 802.1Q, dove e' inserito il tag e quale valore identifica il TPID? [...] Risposta corretta: A) Dopo gli indirizzi MAC, con TPID pari a 0x8100...


```
{
  "id": 1,
  "modello_risposto": "GPT-5",
  "score": {"totale_100": 22},
  "note_valutatore": "ALLUCINAZIONI CRITICHE: Quiz 2 (802.1Q TPID), Quiz
     3 (TCP handshake), Quiz 4 (IPv4 frammentazione) contengono
     informazioni non presenti nei materiali forniti."
}
```


Listing 4.1: Valutazione JSON - GPT-5 (ID 1)

4.6.2 Errori Metodologici: Gemini 2.5 Pro (ID 23 - Analisi)

In questo caso, il modello ha ignorato i vincoli di risoluzione richiesti, optando per un metodo differente da quello specificato.

Prompt di Input (ID 23)

Risolvi il limite (sen x) / x come in un esercizio d'esame. Prima descrivi la strategia generale, poi spiega ogni passaggio e infine fornisci la risposta finale.

Output dell'LLM (Gemini)

Claro, ecco la risoluzione... La strategia piu' efficace e' applicare il Teorema di de l'Hopital... Derivata del numeratore: cos x. Derivata del denominatore: 1. Risultato finale: 1.


```
{
  "id": 23,
  "modello_risposto": "Gemini",
  "score": {"totale_100": 51},
```

```
  "note_valutatore": "Il modello applica de l'Hopital ignorando che la
     richiesta (esercizio d'esame) in Analisi I solitamente richiede
     limiti notevoli o Taylor."
}
```

Listing 4.2: Valutazione JSON - Gemini (ID 23)

4.6.3 Inaffidabilita' Normativa: Sonar (ID 81 - IA-LLM)

Sonar ha mostrato lacune nella precisione concettuale, fornendo una traccia didattica troppo generica per il livello universitario richiesto.

Prompt di Input (ID 81)

Progetta internamente gli snodi didattici. Restituisci una traccia compatta con obiettivi misurabili, fasi della lezione e criteri di valutazione, esplicitando gestione di bias e allucinazioni.

Output dell'LLM (Sonar)

L'obiettivo della lezione e' che lo studente sappia riconoscere e gestire bias e allucinazioni... Si parte con una breve introduzione... Si conclude con una verifica finale...

```
{
  "id": 81,
  "modello_risposto": "Sonar",
  "score": {"totale_100": 20},
  "note_valutatore": "La risposta e' corretta nei concetti base ma manca
     di specificita' disciplinare: non cita tecniche concrete (es.
     prompting, temperature)."
}
```

Listing 4.3: Valutazione JSON - Sonar (ID 81)

CAPITOLO 5

Conclusioni

Il presente lavoro di tesi ha affrontato la problematica della valutazione sistematica degli LLM in ambito educativo, proponendo un framework sperimentale applicato a quattro discipline fondamentali del corso di laurea in Informatica: Analisi Matematica, Reti di Calcolatori, Programmazione Object Oriented (POO) e Insegnamento dell'Intelligenza Artificiale.

Attraverso lo sviluppo di una pipeline di valutazione automatizzata e l'analisi di 334 osservazioni su 7 task didattici distinti, è stato possibile confrontare le prestazioni di quattro modelli allo stato dell'arte sotto diverse condizioni di prompting.

5.1 Sintesi dei Risultati Empirici

L'analisi quantitativa dei dati ha evidenziato trend significativi che rispondono alle domande di ricerca poste in fase introduttiva:

- **Gerarchia Prestazionale e Specializzazione:** Sebbene **GPT-5** si sia confermato il modello complessivamente più solido con uno score medio di **96.38**, l'analisi disaggregata per disciplina ha rivelato che non esiste un vincitore assoluto per ogni contesto. È emerso un dato di particolare rilievo: **Gemini 2.5 Pro**

ha dimostrato una superiorità netta nei task puramente tecnici e di codifica, raggiungendo la perfezione (**100/100**) nella disciplina *Programmazione Object Oriented* e superando GPT-5 anche in *Reti di Calcolatori*. Questo suggerisce che, mentre GPT-5 eccelle nella capacità di ragionamento generalista e nella generazione di contenuti didattici discorsivi, altri modelli mostrano una maggiore affinità con la logica formale e la sintassi del codice.

- **Efficacia delle Tecniche di Prompting:** Contrariamente a quanto spesso riportato in letteratura per task di ragionamento complesso, nel contesto educativo la tecnica **Few-shot** (Score medio: **94.66**) ha superato leggermente sia lo Zero-shot (93.77) che il Chain-of-Thought (93.37). Questo indica che, per la creazione di materiale didattico (quiz, lezioni, correzioni), fornire al modello esempi concreti di output è spesso più efficace che forzare un ragionamento passo-passo, il quale talvolta rischia di introdurre verbosità eccessiva in task ben definiti.

- **Affidabilità e Gestione degli Errori:** Nonostante le medie elevate, l'analisi degli outliers ha rilevato casi sporadici di "fallimento catastrofico". Tali anomalie confermano che, sebbene gli LLM abbiano raggiunto un livello di maturità notevole, non sono esenti da allucinazioni o errori logici gravi.

5.2 Implicazioni per la Didattica

I risultati ottenuti permettono di delineare linee guida pratiche per l'integrazione degli LLM nell'insegnamento universitario. La presenza di outliers negativi dimostra che l'adozione degli LLM deve avvenire secondo un paradigma *Human-in-the-loop*, dove il docente mantiene un ruolo cruciale di supervisore e validatore.

Inoltre, emerge la necessità di un approccio ibrido nella scelta dello strumento: per materie ad alta intensità di codice (come POO), è consigliabile l'uso di modelli con forte specializzazione tecnica (es. Gemini), mentre per la spiegazione di concetti teorici astratti o pedagogici (es. Insegnamento IA o Analisi), modelli con capacità di ragionamento più ampie (es. GPT-5) offrono risultati migliori.

5.3 Contributi e Sviluppi Futuri

Il principale contributo metodologico di questa tesi risiede nella creazione di una pipeline di valutazione riproducibile ed estensibile, capace di gestire sia l'interrogazione dei modelli che l'analisi statistica dei risultati. Le direzioni future di ricerca potranno estendere questo lavoro in tre direzioni:

1. **Multimodalità:** Includere task che richiedano l'analisi di input visivi (es. grafici di funzioni in Analisi o diagrammi di rete), sfruttando le capacità multimodali native dei modelli di nuova generazione.

2. **Valutazione Dinamica:** Integrare metriche basate non solo sulla correttezza statica, ma sull'interazione dialogica, simulando un tutoraggio attivo con studenti virtuali.

3. **Espansione del Dataset:** Aumentare la granularità dei task per verificare se i pattern di "specializzazione" osservati si mantengano costanti su scala più ampia.

In conclusione, questo studio dimostra che gli LLM sono strumenti didattici ormai maturi, capaci di generare contenuti di qualità universitaria spesso indistinguibile da quella umana, a patto di scegliere il modello e la strategia di prompting adeguati alla specificità della disciplina trattata.

Bibliografia

[1] D. E. V. Quinn and P. J. W. Lin, "Generative artificial intelligence in educational contexts: A systematic review of opportunities, challenges, and ethical implications," *International Research Journal of Advanced Engineering and Technology*, vol. 2, no. 10, pp. 102–111, 2025. [Online]. Available: https://aimjournals.com/index.php/irjaet/article/view/328 (Citato a pagina 1)

[2] Khan Academy. (2024) Khanmigo: AI for education. Accessed: 2025-09-09. [Online]. Available: https://www.khanmigo.ai/ (Citato alle pagine 1 e 6)

[3] ——. (2024) Khanmigo now free for U.S. teachers: Trusted AI to streamline your prep. Annuncio ufficiale dei risultati del progetto pilota. [Online]. Available: https://blog.khanacademy.org/khan-academy-efficacy-results-november-2024/ (Citato alle pagine 1 e 6)

[4] D. R. Thomas, C. Borchers, S. Bhushan, E. Gatz, S. Gupta, and K. R. Koedinger, "LLM-generated feedback supports learning if learners choose to use it," 2025, studio sull'impatto del feedback generativo. [Online]. Available: https://arxiv.org/abs/2506.17006 (Citato alle pagine 1 e 7)

[5] T. Liu, J. Chatain, L. Kobel-Keller, G. Kortemeyer, T. Willwacher, and M. Sachan, "Ai-assisted automated short answer grading of handwritten university level

mathematics exams," 2024, confronto GPT-4 vs umani su grading matematico. [Online]. Available: https://arxiv.org/abs/2408.11728 (Citato alle pagine 2 e 7)

[6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, definizione della tecnica di prompting Chain-of-Thought (CoT). [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html (Citato alle pagine 2 e 8)

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," 2020, paper seminale sull'architettura RAG utilizzata per il grounding dei dati. [Online]. Available: https://arxiv.org/abs/2005.11401 (Citato alle pagine 3 e 8)

[8] Google Research. (2022) Minerva: Solving quantitative reasoning problems with language models. Modello specializzato in ragionamento quantitativo STEM. [Online]. Available: https://research.google/blog/minerva-solving-quantitative-reasoning-problems-with-language-models/ (Citato a pagina 7)

Ringraziamenti

Desidero innanzitutto esprimere la mia profonda gratitudine al mio Relatore, il Prof. Fabio Palomba, per la guida attenta, la disponibilità e i preziosi consigli che mi hanno accompagnato durante la stesura di questo elaborato. La sua competenza è stata fondamentale per il raggiungimento di questo traguardo.

Un ringraziamento speciale va a tutta la mia famiglia, per il sostegno incondizionato e per essermi stata accanto, con pazienza e affetto, durante tutto il mio percorso di studi. A loro dedico questo risultato e tutti quelli che verranno.

Questa tesi ha contribuito a piantare un albero in Kenya tramite il progetto Treedom.

`https://www.treedom.net/it/user/sesalab/event/sesa-random-forest`