UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Large Language Models & HealthCare: "AICare", una piattaforma integrata di supporto decisionale per i medici

Relatore
Prof. Rocco Zaccagnino

Candidata
Daria Simonetti
Matricola: 0512114704

Anno Accademico 2023/2024

"Lo sai che i papaveri son alti alti alti,
e tu sei piccolina, e tu sei piccolina..."

A nonna Titi, la mia seconda mamma,
che sarà sempre parte di me.

Abstract

Il recente sviluppo e avanzamento dell'Intelligenza Artificiale ha favorito la diffusione di modelli conversazionali avanzati basati su tecniche di Machine Learning, in grado di comprendere e generare linguaggio umano. I Large Language Models (LLM) rappresentano un'evoluzione significativa in questo campo: si tratta di modelli di intelligenza artificiale, creati su algoritmi di deep learning, capaci di riconoscere, generare, riassumere, tradurre e prevedere contenuti elaborando grandi set di dati. Questi modelli linguistici stanno guadagnando sempre maggiore popolarità in numerosi ambiti, tra cui quello dell'Healthcare, grazie alla loro capacità di analizzare rapidamente informazioni cliniche e fornire supporto ai professionisti della salute.

In questo decennio di progressi e innovazioni, un altro concetto che sta rapidamente guadagnando terreno nel contesto della sanità digitale è il Digital Twin (DT). Un Digital Twin è una replica virtuale di un'entità fisica, costantemente aggiornata grazie alla condivisione dei dati in tempo reale. Sebbene il potenziale di questa tecnologia in ambito sanitario sia ancora in fase di esplorazione, il concetto di Patient Digital Twin si propone come un approccio innovativo per migliorare la medicina personalizzata, predittiva e preventiva.

L'intelligenza artificiale sta emergendo come una tecnologia chiave per il supporto ai professionisti sanitari. L'uso degli LLM in ambito sanitario rappresenta un'area ancora in fase esplorativa, con il potenziale di facilitare l'interpretazione e la comunicazione delle informazioni cliniche. Comprendere fino a che punto questi modelli possano affiancare il medico nella gestione dei pazienti è un aspetto cruciale per valutarne l'applicabilità pratica. Un'ulteriore direzione di sviluppo è rappresentata dall'integrazione degli LLM con i Digital Twin. Tuttavia, prima di considerare un loro impiego in sistemi complessi, è fondamentale studiare le capacità e i limiti degli LLM nel contesto clinico attuale.

Questa tesi esplora il potenziale e l'impatto degli LLM nell' Healthcare, attraverso lo sviluppo di un'applicazione web, "AICare", che consente ai medici di accedere ai dati clinici dei pazienti e interagire con un modello conversazionale per supportare l'analisi delle informazioni sanitarie. Invece di addestrare un modello su dati sanitari, si è scelto di integrare *Mistral 7B OpenOrca* un LLM open-source, ottimizzato tramite Prompt Engineering. Il lavoro svolto non mira a fornire una soluzione definitiva, ma rappresenta un primo passo nell'indagine sulle potenzialità degli LLM in ambito sanitario. I risultati ottenuti possono costruire una base per sviluppi futuri, tra cui il miglioramento dell'applicazione e l'ampliamento della rappresentazione del paziente.

Indice

Capitolo 1

Introduzione

Il presente lavoro di tesi si concentra sullo studio e sull'applicazione dei *Large Language Models* (LLM) in ambito *Healthcare*.

Negli ultimi anni l'intelligenza artificiale ha rivoluzionato diversi settori, con un impatto significativo nell'ambito sanitario. Tra le tecnologie più avanzate in questo contesto troviamo i Large Language Models (LLM) e i Digital Twin (DT). Queste due innovazioni, apparentemente distinte, stanno convergendo per creare strumenti sempre più sofisticati, come il Patient Digital Twin (PDT), in grado di migliorare l'assistenza sanitaria e la gestione dei pazienti.

Cosa sono gli LLM e i Digital Twin? I Large Language Models sono modelli di intelligenza artificiale basati su tecniche di deep learning, progettati per comprendere e generare linguaggio naturale. Il termine *large* (grande) si riferisce al numero di parametri che compongono il modello. Questi parametri rappresentano i *pesi* all'interno delle reti neurali che consentono al modello di apprendere le relazioni e i pattern presenti nei dati di addestramento. Gli LLM vengono addestrati su enormi quantità di dati testuali provenienti da libri, articoli, pagine web e altre fonti linguistiche, acquisendo la capacità di svolgere compiti come traduzioni, sintesi, analisi e completamento del testo e risposte a domande.

L'applicazione degli LLM in ambito sanitario è una direzione di ricerca ancora in evoluzione. Tuttavia, il loro impiego per supportare i medici nell'analisi dei dati clinici, nella generazione di referti e nel supporto decisionale terapeutico, sta attirando crescente interesse. La capacità di questi modelli di elaborare grandi volumi di dati in tempi ridotti potrebbe rapppresentare un vantaggio significativo nell'ottimizzazione dei flussi di lavoro clinici.

Accanto all'evoluzione degli LLM, ha preso piede il concetto di Digital Twin (DT), ovvero una replica virtuale di un oggetto, un sistema o persino una persona, progettata per rappresentare fedelmente la sua controparte fisica in un ambiente digitale. Questi modelli possono essere aggiornati dinamicamente grazie ai dati raccolti in tempo reale da sensori, dispositivi IoT (Internet of Things) o sistemi informatici, consentendo una simulazione dinamica e accurata del sistema rappresentato. La tecnologia dei DT trova applicazione in diversi ambiti, come

l'ingegneria, la produzione industriale e l'energia [17].

In ambito sanitario, i Digital Twin evolvono nel concetto di Patient Digital Twin (PDT): repliche virtuali di pazienti umani reali. L'idea alla base dei PDT è quella di creare modelli che integrino dati clinici del paziente, come parametri vitali, anamnesi medica, immagini diagnostiche e risultati di laboratorio, al fine di simulare il decorso di malattie e prevedere la risposta ai trattamenti medici [10]. Sebbene questo approccio sia promettente, la sua applicazione pratica è ancora in fase di sviluppo. Nella letteratura emergente, i PDT vengono studiati per il loro potenziale in diversi ambiti della medicina, tra cui:

- Simulazione di scenari clinici complessi: la possibilità di prevedere come un paziente potrebbe rispondere a un trattamento farmacologico o a un intervento chirurgico [7].
- Predizione dell'evoluzione di malattie: l'integrazione di algoritmi di machine learning potrebbe consentire ai PDT di anticipare complicazioni o progressioni patologiche, aiutando i medici a intervenire in modo preventivo [10].
- Personalizzazione delle cure: la capcità di testare diversi approcci terapeutici in un ambiente virtuale, potrebbe contribuire a ridurre il rischio per il paziente e a migliorare la precisione delle decisioni cliniche [7].

Uno modello affine ai PDT è la *"P4 Medicine"* (Predictive, Preventive, Personalized, and Participatory Medicine), un modello di medicina che punta a prevenire le malattie, personalizzare i trattamenti in base al singolo paziente e coinvolgerlo attivamente nel processo di cura [19]. L'integrazione tra LLM e PDT potrebbe, in futuro, rappresentare un'area di ricerca promettente per migliorare l'interazione con le simulazioni cliniche. Tuttavia, affinchè questi due strumenti possando diventare utilizzabili sinergicamente nella pratica clinica, sono necessarie ulteriori ricerche per valutarne l'accuratezza, la sicurezza e l'affidabilità. Per questo motivo, questa tesi si concentra sugli LLM, esplorandone il potenziale come strumenti di supporto per i medici.

Motivazioni della tesi. La scelta di scrivere questa tesi è stata motivata da tre fattori principali:

1. *L'impatto degli LLM nell'Healthcare.* I Large Language Models stanno dimostrando un potenziale straordinario nel campo della sanità. Questi modelli sono strumenti in grado di analizzare grandi volumi di dati clinici e fornire supporto decisionale ai medici, ma il loro reale valore nell'assistenza sanitaria

richiede ulteriori studi e applicazioni pratiche. È fondamentale comprendere come gli LLM possano migliorare l'efficienza e la precisione nella pratica medica quotidiana, dal supporto diagnostico alla generazione di consigli terapeutici.

2. *L'innovazione del Patient Digital Twin.* L'idea di creare una replica digitale di un paziente per simulare scenari clinici rappresenta un passo avanti verso la medicina personalizzata. Studiare come integrare gli LLM nei Patient Digital Twin (PDT) offre l'opportunità di esplorare soluzioni innovative per migliorare la gestione dei pazienti e personalizzare i trattamenti. Sebbene il mio lavoro si concentri maggiormente sull'impatto degli LLM, i PDT rappresentano un concetto strettamente correlato, che guida lo sviluppo di tecnologie future per la sanità.

3. *La necessità di validare l'efficacia degli LLM nelle applicazioni pratiche.* Nonostante il crescente interesse per gli LLM nell'ambito sanitario, molte delle loro applicazioni rimangono teoriche o non sufficientemente validate. Il mio lavoro mira a sviluppare e testare un'applicazione web, "AICare", che integra un LLM per supportare i medici nella gestione e nell'analisi dei pazienti. L'obiettivo è dimostrare concretamente quanto questi strumenti possano essere efficaci nel migliorare il processo decisionale clinico e semplificare l'accesso alle informazioni essenziali, pur garantendo sicurezza e privacy dei dati.

Scopo e organizzazione della tesi. Lo scopo principale di questa tesi è valutare le capacità e i limiti dei Large Language Models nel contesto clinico attuale. Attraverso l'analisi di casi d'uso e lo sviluppo di un prototipo di applicazione web, si cercherà di rispondere a domande fondamentali: quanto sono validi questi strumenti? Possono davvero rappresentare un valido supporto per i medici?

All'introduzione qui conclusa seguono altri cinque capitoli:

2. *Stato dell'arte.* Fornisce una breve descrizione degli LLM e della loro storia evolutiva. Viene quindi esplorato l'uso di questi modelli in ambito Healthcare.

3. *Background.* Spiegazione dettagliata delle tecnologie utilizzate, con focus sull'LLM scelto: Mistral 7B OpenOrca. Vengono illustrati la sua storia, le sue caratteristiche e il suo corretto utilizzo in ambito sanitario.

4. *Realizzazione di "AICare".* Presentazione della piattaforma web "AICare". Questo capitolo analizza i punti salienti che hanno portato alla realizzazione del framework e saranno descritti i criteri di progettazione.

5. *Analisi dei risultati.* In questo capitolo vengono presentati e discussi i risultati ottenuti durante la valutazione della piattaforma da parte di cinque medici. Vengono analizzate le prestazioni del sistema, evidenziando i punti di forza del modello così come gli aspetti in cui pecca.

6. *Conclusioni.* Nell'ultimo capitolo viene valutato l'impatto dell'LLM scelto nel contesto della medicina digitale e personalizzata. Vengono infine suggeriti possibili sviluppi futuri per il miglioramento del sistema, con particolare attenzione verso il concetto di Patient Digital Twin.

Capitolo 2

Stato dell'arte

In questo capitolo viene fornita una descrizione dettagliata dei *Large Language Models* insieme a una panoramica dello stato dell'arte di questa tecnologia. Viene infine illustrata la loro potenzialità in ambito *Healthcare*.

2.1 I Large Language Models

I Large Language Models (LLM) rappresentano una delle invenzioni più significative nel campo dell'intelligenza artificiale. Questi modelli sono basati su reti neurali profonde con lo scopo di analizzare e generare linguaggio naturale in modo sofisticato e contestuale. Grazie al loro addestramento su dataset di enormi dimensioni, che includono libri, articoli, pagine web e numerose altre fonti linguistiche, gli LLM sono in grado di comprendere, sintetizzare e produrre testo in diverse lingue e su numerosi argomenti [9].

Una delle caratteristiche principali degli LLM è il numero di parametri che li compongono. Questi parametri rappresentano i *pesi* all'interno della rete neurale, i quali determinano come il modello apprende dai dati. I modelli più avanzati, come *GPT-4* di OpenAI e *LLama 3* di Meta, contano decine o centinaia di miliardi di parametri, consentendo loro di eseguire compiti estremamente complessi mantenendo elevata accuratezza [9, 37].

Un'altra caratteristica importante degli LLM è la tecnologia chiave sulla quale si basano: i *Transformers*. Introdotti per la prima volta nel modello BERT (Bidirectional Encoder Representations from Transformers) e successivamente perfezionati nei modelli della serie GPT (Generative Pre-trained Transformer), i Transformers sono reti neurali progettate per elaborare il contesto globale di un testo. Ciò è possibile grazie a meccanismi come *l'attenzione* e *l'auto-attenzione*, che consentono al modello di attribuire pesi diversi a parole o frasi in base al contesto [38].

Gli LLM possono essere classificati in base a caratteristiche come il tipo di addestramento, il focus applicativo e le dimensioni. Solitamente gli LLM vengono distinti nelle seguenti categorie:

1. *Modelli Generativi*. Questi modelli, come GPT-3 e GPT-4, sono progettati per generare nuovo testo basandosi su un input iniziale. Sono utilizzati in

una vasta gamma di applicazioni tra cui la scrittura automatica, la creazione di contenuti e i chatbot avanzati.

2. *Modelli di Comprensione.* Esempi di questi modelli includono BERT e RoBERTa [14], essi sono ottimizzati per comprendere il significato di un testo e vengono utilizzati principalmente per compiti di classificazione del testo, estrazione di informazioni e risposta alle domande.

3. *Modelli Multimodali.* I modelli multimodali, tra cui GPT-4 e Gemini 1.5 di Google, hanno ampliato le loro capacità includendo la multimodalità, cioè la capacità di processare input sia testuali che visivi. Questa caratteristica li rende particolarmente utili in applicazioni come il riconoscimento di immagini e l'elaborazione di documenti complessi.

Nonostante i progressi straordinari, gli LLM presentano alcune limitazioni importanti:

1. *Costo computazionale.* L'addestramento e l'implementazione degli LLM richiedono enormi risorse computazionali, sia in termini di potenza di calcolo che di memoria.

2. *Bias nei dati.* Questa è probabilmente la limitazione più importante. Essendo addestrati su dataset provenienti da fonti pubbliche, gli LLM posso ereditare pregiudizi presenti nei dati con il rischio di generare risposte non etiche o inappropriate [6].

3. *Mancanza di comprensione reale.* Sebbene gli LLM siano in grado di simulare la comprensione del linguaggio, in realtà si basano su correlazioni statistiche piuttosto che su una vera comprensione semantica.

I Large Language Models rappresentano comunque una pietra miliare nello sviluppo dell'intelligenza artificiale, con applicazioni che spaziano dalla generazione di contenuti alla comprensione del linguaggio naturale.

2.2 Storia ed Evoluzione degli LLM

Nonostante gli LLM abbiano una storia recente, la loro evoluzione ha rivoluzionato il campo dell'intelligenza artificiale in pochi anni.

Il termine *"Large Language Models"* è stato introdotto per indicare modelli basati su reti neurali profonde, progettati per elaborare il linguaggio naturale (Natural Language Processing, NLP) su scala senza precedenti. Questa tecnologia si basa su concetti chiave come l'autoapprendimento supervisionato e l'elaborazione contestuale.

Dalle origini ai primi modelli. L'idea di creare modelli di linguaggio risale agli anni '50 con l'introduzione dei primi approcci basati su regole grammaticali e modelli statistici.

Tuttavia, è solo con l'avvento delle *reti neurali ricorrenti* (RNN) negli anni '80 e '90 che il campo del linguaggio naturale ha fatto un primo grande passo avanti [16]. Infatti, le RNN hanno introdotto la capacità di elaborare sequenze di dati con dipendenze temporali, rendendole particolarmente adatte per compiti come la modellazione del linguaggio, dove il contesto di parole precedenti è cruciale per comprendere o generare testo coerente. Questo approccio ha superato i limiti dei modelli statici, permettendo una rappresentazione più dinamica e contestuale del linguaggio.

Un altro cambiamento fondamentale è arrivato con l'introduzione dei *modelli di word embedding*, come Word2Vec (2013) [24], che hanno permesso ai modelli di rappresentare il significato delle parole come vettori in uno spazio multidimensionale. Questo approccio ha aperto la strada all'elaborazione semantica e al superamento della semplice analisi basata su frequenza.

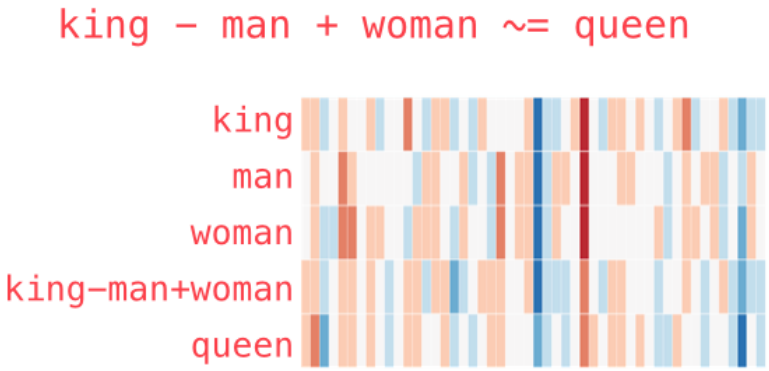

Figura 2.1: Rappresentazione vettoriale delle parole con Word2Vec. L'analogia *king - man + woman* ≈ *queen* evidenzia come il modello cattura le relazioni semantiche tra parole. [4]

L'era dei Transformers. Il progresso decisivo nella storia degli LLM è avvenuto con l'introduzione dei Transformers nel 2017, grazie al lavoro di Vaswani et al. [38]. Il modello *"Attention Is All You Need"* ha introdotto il concetto di *self-attention*, che permette di attribuire pesi differenti alle parole in base al loro contesto globale. Questo meccanismo ha risolto molti dei limiti delle RNN, come la difficoltà di elaborare lunghe sequenze di testo. I transformers hanno dato vita a una serie di modelli innovativi come BERT, creato da Google (2018), è stato il primo modello basato su Transformers a pre-addestrare le sue rappresentazioni linguistiche considerando il contesto bidirezionale di un testo [14], e i modelli GPT, introdotti da OpenAI (2018) ogni versione successiva ha incrementato drasticamente la com-

plessità e le capacità dei modelli, rendendoli leader nelle applicazioni generative [28].

L'espansione verso i modelli multimodali. Nel 2020, con l'introduzione di CLIP e DALL·E da parte di OpenAI, è stato possibile combinare il linguaggio naturale con l'elaborazione visiva, aprendo la strada ai modelli multimodali [29].

Lo stato attuale degli LLM. Oggi gli LLM sono utilizzati in una vasta gamma di applicazioni e i principali sviluppi riguardano:

1. *Efficienza computazionale:* l'introduzione di tecniche come *sparse attention* e *quantizzazione* dei parametri mira a ridurre i costi computazionali mantenendo alte le prestazioni.Un esempio recente è DeepSeek, un LLM sviluppato in Cina che utilizza un'architettura Mixture of Experts (MoE) per attivare solo una parte del modello durante l'elaborazione, ottimizzando il consumo di risorse [1]. Sebbene MoE non sia una novità, l'efficienza con cui DeepSeek lo implementa rappresenta un progresso significativo.

2. *Personalizzazione:* modelli come LLama 3 di Meta offrono la possibilità di essere adattati per applicazioni specifiche, consentendo un addestramento più rapido su dataset mirati [37].

3. *Sicurezza ed etica:* cresce l'attenzione verso i problemi di bias, trasparenza e impatto sociale degli LLM, con ricerche che cercano di affrontare queste sfide [6].

Gli LLM rappresentano un'evoluzione straordinaria nell'intelligenza artificiale, con una crescita che continua ad accelerare. Grazie a tecnologie come i Transformers e alla crescente potenza computazionale, questi modelli stanno ridefinendo il modo in cui interagiamo con le macchine, offrendo opportunità senza precedenti per l'innovazione nei più diversi settori, incluso quello sanitario.

2.3 Gli LLM per la Sanità del Futuro

I Large Language Models stanno trasformando il settore sanitario, aprendo nuove possibilità per migliorare la qualità e l'efficienza delle cure. Questi modelli trovano applicazione in molteplici ambiti, dimostrando un potenziale straordinario in compiti che vanno dalla diagnosi alla gestione dei dati clinici.

Uno degli utilizzi principali è il supporto diagnostico: grazie alla loro capacità di analizzare grandi volumi di dati clinici, gli LLM assistono i medici nella formulazione di diagnosi più accurate. Ad esempio, il modello *Med-PaLM 2* di

Google ha raggiunto competenze paragonabili a quelle umane, ottenendo risultati di livello "esperto" in test medici complessi come l'US Medical Licensing Examination. Ciò dimostra come gli LLM possano contribuire a migliorare il processo decisionale clinico, supportando i professionisti nell'individuazione di patologie e nella definizione di trattamenti [31].

Un altro settore che trae grandi benefici dall'introduzione degli LLM è la generazione di documentazione clinica. Scrivere report medici, riassunti delle cartelle cliniche e note di dimissione richiede tempo e risorse; i modelli linguistici, automatizzando queste attività, liberano il personale sanitario da parte del carico amministrativo consentendo loro di dedicare maggiore attenzione ai pazienti.

Inoltre, gli LLM sono utilizzati per alimentare assistenti virtuali e chatbot in grado di interagire con i pazienti. Questo tipo di applicazione migliora l'accessibilità alle informazioni sanitarie e fornisce un supporto immediato, particolarmente utile in contesti con risorse limitate.

Anche nella ricerca e nella formazione medica gli LLM stanno giocando un ruolo cruciale. Questi modelli sono in grado di analizzare rapidamente enormi quantità di letteratura scientifica, sintetizzando informazioni rilevanti e individuando tendenze emergenti. In ambito formativo, i modelli possono simulare scenari clinici complessi, offrendo ai medici un modo innovativo per affinare le proprie competenze.

Tra gli esempi più rilevanti di applicazioni reali troviamo MedLM di Google, utilizzato in ospedali e strutture sanitarie per migliorare l'efficienza operativa e supportare le decisioni cliniche [11]. Ad esempio, l'Istituto Europeo di Oncologia (IEO) e il Centro Cardiologico Monzino hanno adottato la Clinical Data Platform (CDP), basata sulle soluzioni di Intelligenza Artificiale di Google Cloud, per classificare e analizzare i dati clinici anonimizzati a una velocità significativamente superiore rispetto ai metodi tradizionali. Questa piattaforma utilizza strumenti come MedLM per facilitare l'analisi dei dati e supportare la ricerca medica. Questo progetto dimostra come l'adozione degli LLM stia già generando un impatto tangibile nel settore.

Nonostante le grandi opportunità offerte dagli LLM in ambito sanitario, permangono diverse limitazioni e sfide che devono ancora essere affrontate. L'accuratezza e l'affidabilità degli LLM, per esempio, rappresentano un problema. Studi hanno evidenziato che tendono a incontrare difficoltà nei colloqui clinici, dove è fondamentale catturare informazioni dettagliate ed evitare errori interpretativi. Inoltre, l'uso di dati clinici sensibili solleva preoccupazioni significative sulla privacy e sulla sicurezza delle informazioni, aspetti che richiedono una gestione estremamente attenta [15].

Un'altra sfida importante è legata ai bias presenti nei modelli. Gli LLM, essendo addestrati su grandi quantità di dati, possono riflettere pregiudizi esistenti,

rischiando di perpetuare disuguaglianze nelle cure mediche. A ciò si aggiunge la necessità di integrare queste tecnologie nei flussi di lavoro clinici, un processo che richiede formazione e adattamenti significativi per garantire un utilizzo efficace e sicuro.

Dal punto di vista etico, invece, l'uso degli LLM in sanità solleva questioni legate alla depersonalizzazione delle cure, all'autonomia decisionale dei pazienti e all'equità nell'accesso ai trattamenti.

Nonostante questi problemi, l'adozione di modelli linguistici avanzati continua a crescere, spinta dal desiderio di migliorare le prestazioni del sistema sanitario e di affrontare le sfide globali della medicina moderna.

Capitolo 3

Background

Per comprendere appieno il funzionamento della piattaforma, sviluppata in questa tesi, è necessario introdurre le principali tecnologie utilizzate. Questo capitolo fornisce una panoramica sugli strumenti adottati, con un focus specifico su *Mistral*, il modello di linguaggio impiegato per elaborare le richieste dei medici, nostri utenti.

L'obiettivo di questo capitolo è fornire un quadro chiaro delle tecnologie alla base di *"AICare"*, delineando il loro ruolo nella gestione e nell'elaborazione dei dati sanitari, e mettendo in evidenza le sfide e le soluzioni adottate durante lo sviluppo.

3.1 Synthea: Generazione di Dati Sanitari Sintetici

Nell'ambito della ricerca sanitaria, l'accesso ai dati clinici è spesso limitato da vincoli etici, legali e pratici. Per superare queste restrizioni, si ricorre frequentemente a dati sintetici che riproducono le caratteristiche statistiche dei dati reali senza compromettere la privacy dei pazienti. In questo contesto, Synthea si distingue come uno strumento open-source di rilievo per la generazione di dati sanitari sintetici.

3.1.1 Cos'è Synthea?

Synthea, acronimo di "Synthetic Health", è un simulatore di popolazione paziente progettato per produrre dati sanitari realistici ma fittizi.

Il suo obiettivo principale è quello di fornire dati completi di cartelle cliniche elettroniche (Electronic Health Records, EHR) che riflettano fedelmente le dinamiche di una popolazione reale, garantendo però l'assenza di informazioni riconducibili a individui esistenti. Ciò è reso possibile attraverso la modellazione di percorsi di vita dei pazienti, dalla nascita alla morte, includendo eventi sanitari come diagnosi, allergie, trattamenti e risultati clinici [12].

3.1.2 Utilizzo di Synthea

Nel presente lavoro, Synthea è stato utilizzato per generare i pazienti da inserire all'interno della piattaforma, in modo da poter valutare l'applicazione sviluppata e il Large Language Model integrato.

La scelta di Synthea è stata dettata da due motivi:

1. *La difficoltà nel trovare e accedere a dataset reali.* Come detto precedentemente l'accesso a dati clinici reali è fortemente limitato da vincoli etici e normative sulla privacy, rendendo l'uso di dati sintetici una soluzione efficace.

2. *La capacità di produrre dati dettagliati e strutturati.* Synthea genera informazioni su dati demografici, condizioni mediche, farmaci assunti, procedure sanitarie, piani di cura e tanto altro. Inoltre, questi dati sono esportabili in formati standardizzati come HL7 FHIR, C-CDA e CSV, ma anche in file testuali.

In questa tesi, ho scelto di utilizzare i file testuali generati da Synthea per creare i documenti da inserire nel database *MongoDB* della piattaforma.

Dopo aver analizzato il formato e il contenuto dei dati prodotti, ho deciso di escludere alcune informazioni non rilevanti per gli obiettivi dell'applicazione, selezionando solo quelle più utili per la gestione dei pazienti virtuali.

Per realizzare questa fase di trasformazione e strutturazione dei dati, ho sviluppato una funzione in Python, utilizzata in un notebook Jupyter esterno alla piattaforma, per estrarre e convertire i dati contenuti nei file testuali in un formato JSON compatibile con il database. Questa funzione legge il file di testo generato da Synthea; esclude le informazioni non necessarie; estrae e organizza i dati utili; infine, salva i dati in formato JSON, pronti per essere caricati nel database.

L'approccio adottato ha permesso di filtrare le informazioni superflue e di organizzare i dati in una struttura più chiara e facilmente gestibile, ottimizzando così il loro utilizzo all'interno della piattaforma sviluppata.

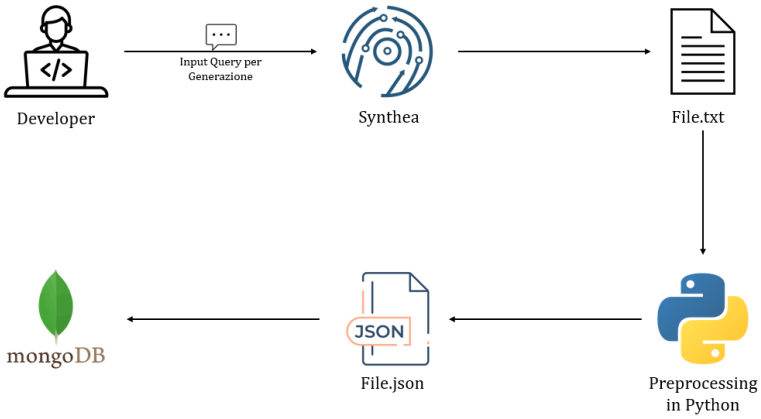

Figura 3.1: Flusso di Elaborazione dei Dati Generati da Synthea.

3.1.3 Limitazioni

Nonostante i vantaggi offerti, è importante riconoscere alcune limitazioni nell'uso di Synthea. Poiché i dati generati sono basati su modelli e statistiche predefinite, potrebbero non catturare tutte le complessità e le variabilità presenti nei dati clinici reali.

Inoltre, la loro natura sintetica sta a significare che non rappresentano scenari realmente osservati, il che può influenzare la loro affidabilità in contesti di ricerca clinica.

Nel contesto di questa tesi, i dati generati da Synthea sono stati usati esclusivamente per il testing della piattaforma e per valutare il comportamento del Large Language Model integrato. L'uso di dati sintetici per il testing non rappresenta un problema, in quanto consente di verificare le funzionalità della piattaforma in un ambiente controllato, senza compromettere la privacy dei pazienti. Tuttavia, in scenari reali, sarebbe opportuno validare il sistema anche su dati clinici reali, per garantire un'affidabilità completa nelle applicazioni pratiche.

3.2 Tecniche di Utilizzo degli LLM

Per sfruttare al meglio il potenziale dei Large Language Models sono state sviluppate diverse tecniche che ne ottimizzano l'efficacia e l'efficienza. Tra le principali metodologie si annoverano il Fine-Tuning, la Retrieval-Augmented Generation (RAG) e il Prompt Engineering.

3.2.1 Fine-Tuning

Il *Fine-Tuning* è un processo che implica l'ulteriore addestramento di un modello pre-addestrato su un dataset specifico al fine di adattarlo a compiti o domini particolari. Questa tecnica consente al modello di apprendere sfumature e dettagli rilevanti per applicazioni specifiche, migliorando la pertinenza e la precisione delle risposte.

Una volta preparato il dataset di addestramento, esso viene diviso in tre parti: una per il training, una per la validation e una per il test. Durante il Fine-Tuning, il modello riceve prompt dal dataset di addestramento e genera risposte. Successivamente, un algoritmo di ottimizzazione, come la *discesa del gradiente*, calcola l'errore tra l'output generato e il valore atteso. Questo errore viene utilizzato per aggiornare iterativamente i pesi del modello in un processo che si ripete per più cicli, detti *epoche*, fino a minimizzare l'errore di predizione [23].

Il Fine-Tuning consente di personalizzare l'LLM per un determinato contesto, migliorandone le capacità rispetto a un modello generalista. Tuttavia, questo processo richiede risorse computazionali significative e un dataset di alta qualità; quest'ultimo va bilanciato con il numero di epoche al fine di evitare l'*overfitting*, il quale potrebbe compromettere la capacità del modello di generalizzare a nuovi dati.

3.2.2 Retrieval-Augmented Generation

La *Retrieval-Augmented Generation* (RAG) è una tecnica che combina le capacità generative degli LLM con meccanismi di recupero di informazioni da fonti esterne.

In questo approccio, l'LLM usa l'input dell'utente per effettuare ricerche in database o documenti specifici, recuperando informazioni pertinenti che vengono poi utilizzate, insieme all'input iniziale, per generare risposte più aggiornate e accurate. Studi recenti hanno evidenziato l'efficacia dell'approccio RAG in vari contesti applicativi [20].

Sebbene il RAG e il Fine-Tuning possano essere percepiti come metodi concorrenti, in realtà possono essere utilizzati in modo complementare. L'Integrazione delle due tecniche offre, infatti, un approccio sinergico; il Fine-Tuning può essere utilizzato per specializzare LLM in un dominio specifico, mentre il RAG può fornire accesso a informazioni aggiornate e dettagliate all'interno di quel dominio, portando a prestazioni significativamente migliori nel tempo [30].

3.2.3 Prompt Engineering

Il *Prompt Engineering* è una disciplina emergente nell'ambito dell'elaborazione del linguaggio naturale (NLP) e consiste nella progettazione dei prompt per guidare

gli LLM nella generazione. Questa tecnica non richiede la modifica dei parametri interni del modello, ma si basa sulla formulazione strategica delle richieste per ottenere risposte pertinenti e accurate.

Le tecniche di Prompt Engineering sono numerose[34], tra le principali abbiamo:

1. *Zero-Shot Prompting.* Consiste nel formulare un prompt che permette al modello di affrontare un compito senza fornire esempi specifici.

2. *Few-Shot Prompting.* In questo approccio, il prompt include alcuni esempi del compito desiderato, fornendo al modello un contesto su come dovrebbe rispondere.

3. *Chain-of-Tought Prompting.* Questa tecnica incoraggia il modello a generare una sequenza di passaggi logici, o "catena di pensieri", per giungere alla risposta finale, migliorando le capacità di ragionamento del modello [39].

4. *Instruction-Based Prompting.* In questo caso, il prompt fornisce al modello istruzioni esplicite su come deve rispondere.

3.2.4 Approccio Scelto

In questo progetto è stato scelto di utilizzare il Prompt Engineering invece del Fine-Tuning, poiché quest'ultimo richiede risorse computazionali elevate e un dataset annotato. Il Prompt Engineering offre invece un metodo più flessibile ed efficiente, permettendo di guidare il comportamento dell'LLM tramite input ben strutturati senza necessità di riaddestramento.

La sezione 3.5 approfondisce come questa tecnica sia stata concretamente applicata nel progetto.

3.3 Mistral 7B OpenOrca, una Panoramica sul Modello Dietro "AICare"

Di seguito viene presentato *Mistral 7B OpenOrca*, il modello selezionato, a seguito dell'analisi di numerosi modelli, per il progetto *"AICare"*. Verrà analizzata l'origine, l'architettura, le capacità, le limitazioni e le modalità di installazione e utilizzo, con particolare attenzione alla versione quantizzata a 8 bit.

3.3.1 Origine

Il modello Mistral 7B OpenOrca è il risultato di una collaborazione tra MistralAI e il progetto OpenOrca. MistralAI ha rilasciato il modello originale *Mistral 7B* il

27 settembre 2023 [2].

Successivamente, il team OpenOrca ha effettuato un Fine-Tuning su questo modello utilizzando il proprio dataset, denominato *OpenOrca*, per migliorarne le prestazioni in compiti specifici di elaborazione del linguaggio naturale.

Il modello risultante, Mistral 7B OpenOrca, è stato rilasciato come open-source sotto licenza Apache 2.0, rendendolo accessibile per applicazioni accademiche e di ricerca. Questo rilascio rappresenta una novità: è, infatti, un modello completamente aperto con eccellenti prestazioni, capace di funzionare in modo accelerato anche su GPU consumer di fascia media.

3.3.2 Architettura e caratteristiche principali

Mistral 7B OpenOrca mantiene l'architettura del modello Mistral 7B originale; sono entrambi modelli con 7 miliardi di parametri, basati sull'architettura Transformer. Questo tipo di architettura si basa su meccanismi di *self-attention*, che permettono al modello di valutare l'importanza di diverse parole in una sequenza durante l'elaborazione del testo. Mistral adotta, inoltre, una configurazione decoder-only, ottimizzata per la generazione di testo. Questa scelta architetturale consente al modello di prevedere la parola successiva in una sequenza, rendendolo efficace in compiti come la scrittura automatica, la traduzione e la risposta a domande.

Una caratteristica importante di Mistral è l'implementazione dell'*attenzione con query raggruppate* (GQA). Tale tecnica migliora l'efficienza computazionale riducendo la complessità dell'operazione di attenzione, permettendo al modello di gestire sequenze più lunghe con un consumo di risorse inferiore.

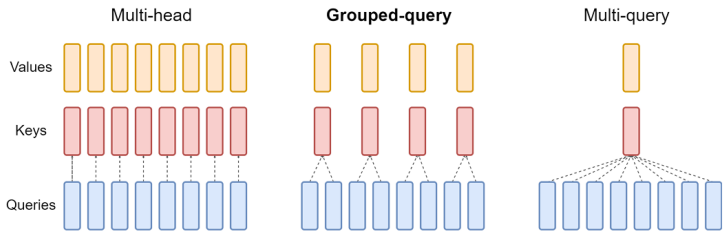

Figura 3.2: Panoramica del metodo di query raggruppate. [3]

Inoltre il modello usa la tecnica della *Sliding Window Attention* (SWA), che consente di elaborare testi più estesi senza aumentare la complessità computazionale.

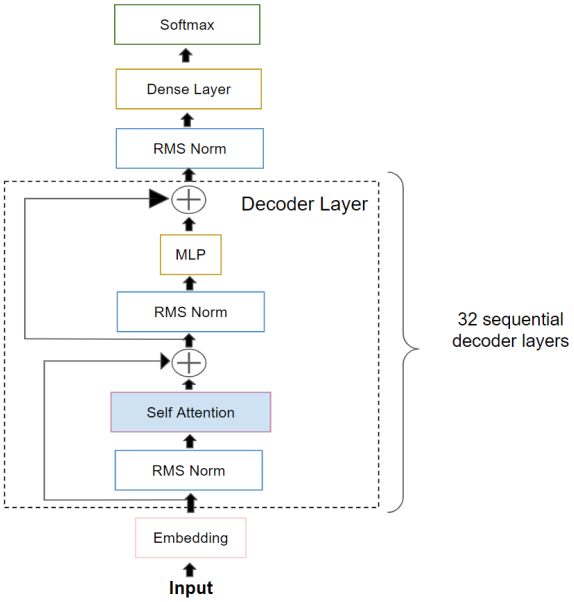

Figura 3.3: Visione di alto livello del modo in cui il modello Mistral elabora l'input per generare il linguaggio..

Il fine-tuning effettuato da OpenOrca ha comportato l'addestramento del modello su un dataset arricchito, migliorando le sue capacità di comprensione e generazione del linguaggio. Il dataset usato è una collezione di dati derivata dalla FLAN Collection [22], arricchita con completamenti generati da modelli avanzati come GPT-4 e GPT-3.5 [26]. Esso è strutturato in istanze che includono un identificatore unico, il prompt di sistema utilizzato, la domanda originale tratta dalla FLAN Collection e la risposta generata dal modello. Questa organizzazione facilita l'uso del dataset per compiti di addestramento e valutazione in vari ambiti dell'elaborazione del linguaggio naturale.

Per ottimizzare ulteriormente l'efficienza, è stata applicata una quantizzazione a 8 bit al modello. La quantizzazione è una tecnica che riduce la precisione dei pesi del modello, diminuendo quindi le risorse necessarie per l'inferenza, senza però compromettere significativamente le prestazioni. Questo approccio rende Mistral 7B OpenOrca più accessibile per l'implementazione su hardware con risorse limitate, mantenendo al contempo elevate capacità di elaborazione del linguaggio naturale. È per questi motivi che, nel progetto "AICare", è stata integrata la versione quantizzata a 8 bit.

3.3.3 Task, punti di forza e limitazioni

Il modello Mistral 7B OpenOrca è in grado di svolgere numerosi task:

1. *Generazione di Testo.* È in grado di produrre contenuti coerenti e pertinenti, utili per la scrittura creativa, la redazione di articoli e la generazione di codice;

2. *Risposta a Domande.* Fornisce risposte dettagliate e informative su una vasta gamma di argomenti, dimostrando una comprensione approfondita del linguaggio;

3. *Conversazione.* Può partecipare a dialoghi aperti, mantendo il contetso e la coerenza in più turni di conversazione;

4. *Riassunto.* Sintetizza testi lunghi distillando le informazioni chiave in riassunti concisi;

5. *Analisi del Sentimento.* È in grado di valutare il tono emotivo di un testo, identificando sentimenti positivi, negativi o neutri.

I punti di forza del modello sono sicuramente: l'efficienza computazionale, poichè la versione quantizzata a 8 bit offre tempi di inferenza più rapidi con un uso della memoria ridotto, e le prestazioni elevate; nonostante la quantizzazione, il modello mantiene elevate capacità di comprensione e generazione del linguaggio.

Tuttavia, è importante notare che, nonostante le capacità avanzate, il modello potrebbe ereditare bias dai dati di addestramento e non possedere una conoscenza aggiornata ad eventi recenti. Inoltre, la quantizzazione a 8 bit può comportare una leggera perdita di precisione nelle risposte generate, sebbene non sia significativa nella maggior parte dei casi d'uso. Un'altra limitazione da considerare è la sua dipendenza dai prompt; come per la maggior parte degli LLM, prompt poco chiari o ambigui possono portare a risposte meno accurate.

3.3.4 Installazione

La versione quantizzata a 8 bit di Mistral 7B OpenOrca è disponibile su HugginFace nella repository di TheBloke [8] come file .GGUF.
Una volta scaricato il file e messo nella cartella "models", bisognerà installare la libreria *llama-cpp* con il seguente comando: "pip install llama-cpp-python". Infine, sarà possibile caricare il modello utilizzando il seguente frammento di codice.

```
from llama_cpp import Llama

model_path = "./model/mistral-7b-openorca.Q8_0.gguf"
model = Llama(model_path=model_path, n_ctx=4096)
```

Listing 3.1: Snippet di codice per caricare Mistral 7B OpenOrca

È importante notare che al momento del caricamento del modello è possibile personalizzare determinati parametri, tra cui il parametro *"n_ctx"*, il quale determina la finestra di contesto. Questo parametro rappresenta il numero massimo di token (input + output) che il modello può considerare in una singola richiesta. Mistral 7B OpenOrca può supportare una finestra di contesto di 32.768 token. Tuttavia, quando viene caricato senza esplicitare il valore di *"n_ctx"*, esso viene automaticamente impostato a 512 token. Bisogna bilanciare la scelta di questo valore, poiché un valore troppo alto potrebbe portare a errori o rallentamenti, mentre un valore troppo basso potrebbe ridurre la comprensione del testo o, in caso di input lunghi, il modello potrebbe non avere abbastanza contesto per generare risposte coerenti.

3.4 Mistral in "AICare"

3.4.1 Integrazione

Il modello Mistral 7B OpenOrca è stato integrato nell'applicazione *"AICare"*, una piattaforma progettata per supportare i medici nell'analisi e nell'interpretazione dei dati clinici dei pazienti. L'implementazione del modello nel sistema è stata realizzata attraverso un'interfaccia RESTful, consentendo la comunicazione tra il backend dell'applicazione e il modello linguistico.

L'integrazione di Mistral è avvenuta sfruttando un server Flask, il quale gestisce le richieste in ingresso e inoltra le query all'LLM. Quando un medico invia una richiesta tramite l'interfaccia della piattaforma, il sistema elabora il prompt e lo trasmette al modello per la generazione della risposta. Questo processo avviene attraverso le seguenti fasi principali:

1. *Acquisizone dell'input.* Il messaggio inviato dal medico viene acquisito e arricchito con il contesto clinico del paziente, che include dati demografici, anamnesi ed eventuali risultati di esami diagnostici.

2. *Formattazione del prompt.* Il messaggio viene strutturato secondo il paradigma del Prompt Engineering per ottimizzare la risposta dell'LLM.

3. *Inferenza del modello.* Il prompt viene passato al modello, il quale genera una risposta coerente basata sulle informazioni fornite.

4. *Post-processing e restituzione della risposta.* La riposta generata viene elaborata per garantire chiarezza e coerenza, quindi viene restituita all'interfaccia utente.

```
@auth_bp.route('/api/send-category', methods=['POST'])
def send_category_to_llm():
    try:
        data = request.json
        patient_id = data.get('patient_id')
        category_name = data.get('category_name')
        category_text = data.get('category_text')

        if not patient_id or not category_name or not
        category_text:
            return jsonify({"error": "patient_id, category_name,
            and category_text are required"}), 400


        patients_collection = auth_bp.db["Patients"]
        patient = patients_collection.find_one({"_id":
        ObjectId(patient_id)})

        if not patient:
            return jsonify({"error": "Patient not found"}), 404

        patient_details = f"""
        Name: {patient['details'].get('name', 'N/A')}
        Age: {patient['details'].get('age', 'N/A')}
        Gender: {patient['details'].get('gender', 'N/A')}
        """

        conversation_context = f"""
        You are an assistant helping a doctor analyze patient
        data.
        Below is the patient's demographic information and
        the category data to analyze.
        If there are missing details or data not provided,
        clearly state "Data not available" and do not assume
        or invent information, avoid unnecessary repetitions.

        {patient_details}

        Task: Provide a detailed summary and analysis of the
        patient's {category_name}.
        Content: {category_text}

        Assistant: Respond in a complete and detailed manner,
        avoiding unnecessary repetitions.
        """
```

```
46
47          response = auth_bp.model(
48              conversation_context.strip(),
49              max_tokens=2048,
50              stop=["User:", "Assistant:"],
51              temperature=0.7
52          )
53
54          bot_response = response['choices'][0]['text'].strip()
55
56          if patient_id not in chat_histories:
57              chat_histories[patient_id] = []
58          chat_histories[patient_id].append(f"User:
59          {category_name}: {category_text}")
60          chat_histories[patient_id].append(f"Assistant:
61          {bot_response}")
62
63          return jsonify({"response": bot_response}), 200
64
65      except Exception as e:
66          return jsonify({"error": str(e)}), 500
```

Listing 3.2: API per far analizzare una categoria di dati clinici a Mistral

L'adozione della versione quantizzata a 8 bit ha permesso di mantenere elevate capacità di elaborazione, riducendo al contempo i requisiti hardware del sistema.

L'architettura adottata ha dimostrato un'elevata efficienza nel gestire le richieste degli utenti, consentendo una comunicazione fluida tra medico e modello linguistico, senza necessità di un addestramento supplementare.

La scelta della *finestra di contesto* a 4096 token, che ha aumentato leggermente i tempi di generazione delle risposte, è stata dettata dalla necessità di garantire token sufficienti per l'output, prevedendo l'uso di numerosi token in input presi dai dati medici.

3.4.2 Tecniche di Prompt Engineering Adottate

L'addestramento e l'ottimizzazione degli LLM possono avvenire attraverso diverse tecniche, come anticipato alla fine del sottocapitolo 3.2. Nel contesto del progetto "AICare", la scelta è ricaduta sul Prompt Engineering per diverse ragioni.

Motivazioni. Il Fine-Tuning prevede l'addestramento del modello su un dataset specifico per adattarlo a un dominio particolare, come quello medico. Tuttavia, questo processo richiede grandi quantità di dati annotati, risorse computazionali elevate e tempi di addestramento prolungati. Inoltre, i modelli Fine-Tunati rischiano di perdere capacità generali, diventando eccessivamente specializzati e

meno adattabili a compiti diversi.

D'altra parte, il Prompt Engineering consente di ottimizzare il comportamento del modello senza modificarne i pesi, utilizzando semplicemente istruzioni ben formulate. Questa tecnica si è rivelata particolarmente adatta per "AICare", poiché permette di ottenere risposte più precise senza dover effettuare un costoso riaddestramento.

Vantaggi nel Progetto. L'uso del Prompt Engineering offre numerosi benefici nel contesto dell'analisi di dati clinici

1. *Efficienza Computazionale.*Non c'è bisogno di avere GPU potenti o lunghi tempi di addestramento.

2. *Flessibilità.* Permette di adattare rapidamente il comportamento dell'LLM a nuovi contesti senza modificare il modello.

3. *Facilità di Implementazione.* La struttura delle richieste all'LLM può essere ottimizzata e migliorata iterativamente con test pratici.

4. *Minore Rischio di Overfitting.* A differenza del Fine-Tuning, il Prompt Engineering non limita il modello a un dominio ristretto, lasciandolo capace di rispondere a una varietà di domande.

Strategie Utilizzate. Per massimizzare l'efficacia delle risposte generate da Mistral, sono state adottate e combinate le seguenti tecniche:

1. *Instruction-Based Prompting.* Questa tecnica prevede l'inserimento di istruzioni chiare e dettagliate per guidare il modello verso una risposta adeguata. Specificare chiaramente il ruolo dell'LLM e il tipo di risposte richieste migliora la pertinenza delle informazioni generate. Di seguito sono presenti degli esempi, per due task leggermente diversi, che mostrano l'utilizzo di questa strategia .

```
conversation_context = f"""
        You are an assistant to a medical doctor. Your task
        is to analyze patient data, summarize the most
        important information, and provide a brief overview
        of the patient's general health condition. Provide
        recommendations if
        applicable."""
```

Listing 3.3: Prompt per far fare un riassunto di tuta la storia clinica.

```
conversation_context = f"""
        You are an assistant to a medical doctor. Your task
        is to help with patient data analysis, summarization,
        and answering questions in a precise and professional
        manner. Provide responses suitable for a clinical
        setting."""
```

Listing 3.4: Prompt per fare in modo che Mistral risponda a domande generali di medicina.

```
conversation_context += "\nAssistant: Respond in a complete
   and detailed manner, avoiding unnecessary repetitions."
```

Listing 3.5: Aggiunta di contesto dettagliato.

2. *Contextual Dynamic Prompting.* Nel sistema sviluppato, il prompt non è statico ma viene dinamicamente aggiornato in base al contesto della conversazione. Questa tecnica, simile a quella descritta in *Swamy et al.* [35], consente di migliorare significativamente la coerenza e la pertinenza delle risposte. Il contesto utilizzato include sia lo storico della conversazione, ovvero i precedenti turni di dialogo tra medico e modello, sia i dati del paziente, come informazioni anagrafiche e cliniche. L'uso di prompt dinamici contestuali consente di migliorare la qualità delle interazioni, evitando risposte generiche e garantendo una migliore continuità nel dialogo.

```
if patient_id not in chat_histories:
    chat_histories[patient_id] = []

conversation_context = "\n".join(chat_histories[patient_id])

while calculate_tokens(conversation_context) > 2000:
    chat_histories[patient_id].pop(0)
    conversation_context = "\n".join(chat_histories
    [patient_id])
```

Listing 3.6: Aggiunta delle conversazioni precedenti al prompt da dare input all'LLM.

3.4.3 Problemi riscontrati e soluzioni adottate

L'integrazione di Mistral 7B OpenOrca non ha comportato molte sfide tecniche e operative, ciò è stato possibile grazie all'uso della versione quantizzata a 8 bit che ha evitato problemi come latenza eccessiva e uso di memoria eccessiva. In questa sezione viene analizzato il principale problema riscontrato durante lo sviluppo e le soluzioni adottate per garantire un'inferenza efficiente e risposte affidabili all'interno del sistema.

Generazione di Risposte Non Sempre Pertinenti. Nonostante la qualità del modello, in alcuni casi le risposte generate erano generiche o non strettamente pertinenti al contesto medico. Questo è un problema comune negli LLM non Fine-Tunati su dati specifici, che potrebbero fornire risposte vaghe o poco contestualizzate.

Soluzione: Sono state ottimizzate le richieste al modello attraverso l'uso del Prompt Engineering, specificando il ruolo del modello e fornendo istruzioni dettagliate. Inoltre, è stata aggiunta un'istruzione per evitare risposte basate su dati incerti: *"If there are missing details or data not provided, clearly state "Data not available" and do not assume or invent information."* 3.2.

3.5 Osservazioni Finali

L'integrazione di Mistral 7B OpenOrca nell'applicazione "AICare" ha permesso di sfruttare un modello avanzato di elaborazione del linguaggio naturale per supportare i medici nell'analisi e nell'interpretazione dei dati clinici. I risultati ottenuti verranno approfonditi nel capitolo 5.

Capitolo 4

Realizzazione di "AICare"

L'obiettivo di questo lavoro è quello di realizzare una piattaforma in grado di offrire supporto ai medici, nell'analisi e nella cura dei pazienti, tramite l'integrazione di Mistral 7B OpenOrca.

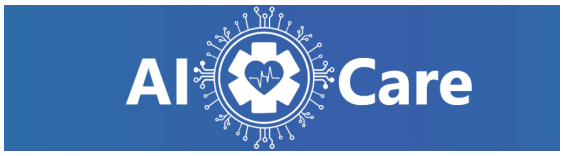

Figura 4.1: Logo di "AICare".

"AICare" [33] è una web application sviluppata utilizzando le principali tecniche che sono presentate nel seguente capitolo.

4.1 Architettura della Piattaforma

L'architettura di "AICare" è stata progettata per garantire modularità, scalabilità ed efficienza, sfruttando una separazione chiara tra frontend e backend. Questa scelta architetturale consente una maggiore flessibilità nello sviluppo, facilitando eventuali aggiornamenti e miglioramenti futuri.

L'applicazione è composta da tre livelli principali:

1. *Frontend (React + Tailwind CSS)* - L'interfaccia con cui il medico interagisce.
2. *Backend (Flask + Python)* - Il server che gestisce le richieste, comunica con il modello AI e fornisce i dati elaborati.
3. *Modulo LLM (Mistral 7B OpenOrca Q8)* - Il modello linguistico che genera risposte a partire dai dati forniti.

Questa separazione consente al frontend e al backend di operare indipendentemente, comunicando tramite API REST.

4.1.1 Struttura Generale

La piattaforma segue un'architettura client-server, dove il frontend si occupa della gestione dell'interfaccia utente, mentre il backend fornisce i dati richiesti e gestisce l'interazione con il modello AI.

Frontend (React + Tailwind CSS). Il frontend di “AICare” è stato sviluppato utilizzando React, una delle librerie più popolari per la creazione di interfacce utente dinamiche e reattive. React è stato scelto per diversi motivi:

1. *Componentizzazione.* Permette di suddividere l'interfaccia utente in componenti riutilizzabili, migliorando la manutenibilità del codice.

2. *Efficienza.* Grazie al *Virtual DOM*, React aggiorna solo le parti necessarie all'interfaccia, migliorando le prestazioni rispetto ai framework tradizionali.

3. *Modularità.* React si integra facilmente con altri strumenti e librerie, facilitando l'integrazione con il backend e con l'LLM.

Per lo stile e il design è stato utilizzato Tailwind CSS direttamente *inline* all'interno del codice React. Questa scelta ha permesso di:

1. *Evitare la creazione di file separati*, mantenendo gli stili direttamente nei componenti React per una migliore leggibilità e organizzazione.

2. *Ridurre il rischio di conflitti tra stili*, poiché ogni componente definisce i propri stili inline, evitando problemi di override tra classi globali.

3. *Ottenere facilmente un design responsivo.* Le classi di Tailwind facilitano la creazione di interfacce adattabili a diversi dispositivi.

In “AICare”, React e Tailwind CSS sono stati utilizzati per creare un'interfaccia pulita, intuitiva e facilmente navigabile dai medici, con particolare attenzione alla semplicità d'uso e alla chiarezza nella visualizzazione dei dati clinici.

Backend (Flask + Pyhton). Il backend di “AICare” è stato sviluppato con Flask, un micro-framework basato su Python, scelto per la sua leggerezza e semplicità d'uso nella gestione di API REST. Flask è stato preferito rispetto ad altri framework più complessi per diversi motivi:

1. *Semplicità.* La sua struttura minimale consente di sviluppare REST rapidamente, senza una configurazione eccessivamente rigida.

2. *Integrazione con modelli AI.* Flask è ampiamente utilizzato per deploy di modelli di Machine Learning e si integra facilmente con le librerie AI.

3. *Scalabilità.* Pur essendo un micro-framework, Flask può essere scalato per gestire un numero elevato di richieste tramite l’uso di sistemi di caching e load balancing.

Nel contesto del progetto, il backend gestisce le richieste provenienti dal frontend attraverso API REST, ricevendo i messaggi inviati dal medico e processandoli per ottenere risposte dall’LLM. Infatti, si interfaccia con Mistral per formulare richieste e ottenere risposte contestualizzate.

Interfacciamento con MongoDB. Un aspetto fondamentale della progettazione di “AICare” è la gestione dei dati clinici, provenienti da Synthea, attraverso MongoDB, un database NoSQL scelto per la sua scalabilità e capacità di gestire dati non strutturati in modo efficiente. Il backend della piattaforma funge da intermediario tra il database e le richieste provenienti dal frontend, garantendo un accesso rapido e sicuro alle informazioni cliniche.

Quando un medico interagisce con “AICare”, il backend si occupa di recuperare dal database tutti i dati rilevanti all’operazione da svolgere, che essa sia un’operazione di visualizzazione dei dati oppure di consultazione del modello.

Il database MongoDB contiene i record dei pazienti, organizzati in formato *document-based* (JSON-like). Ogni documento memorizza informazioni dettagliate, come dati demografici, condizioni cliniche, farmaci prescritti e piani di cura.

Oltre al recupero dei dati, il backend è responsabile anche dell’inserimento e dell’aggiornamento delle informazioni nel database. Ad esempio, quando avviene un inserimento o una modifica dei pazienti, il sistema aggiorna dinamicamente il database per mantenere la coerenza e la tracciabilità delle informazioni.

L’adozione di MongoDB in “AICare” si è rivelata una scelta ottimale per garantire flessibilità e scalabilità nella gestione delle informazioni sanitarie. La sua capacità di supportare query rapide e ottimizzate ha permesso di mantenere un flusso di dati efficiente, senza compromettere le prestazioni della piattaforma.

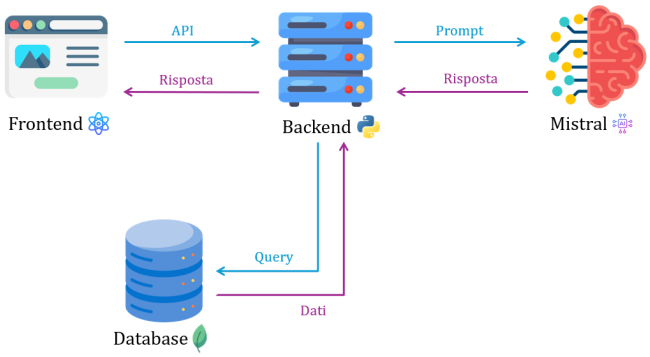

Figura 4.2: Diagramma di flusso che mostra l’interazione i componenti del sistema.

4.1.2 Integrazione con il Modello Mistral

L’integrazione di Mistral 7B OpenOrca è avvenuta attraverso una serie di API REST che permettono al frontend di inviare richieste e ricevere risposte in modo efficiente. Quando un medico effettua una domanda attraverso l’interfaccia utente, il sistema segue i successivi passaggi:

1. Il frontend React invia la richiesta al backend tramite una chiamata API (presente nel file api.js);
2. Il backend Flask riceve la richiesta, estrae i dati clinici pertinenti e costruisce un prompt da inviare al modello AI;
3. Mistral 7B OpenOrca elabora la richiesta e genera una risposta testuale;
4. Il backend riceve la risposta dall’LLM, la filtra e la invia al frontend;
5. Il frontend visualizza la risposta all’interno dell’interfaccia utente in un formato chiaro e leggibile.

Grazie a questa integrazione, “AICare” garantisce che il medico veda solo dati clinici in linguaggio naturale, senza dover interpretare informazioni tecniche o grezze. Questo migliora la leggibilità delle informazioni e facilita la presa di decisioni cliniche.

4.2 Interfaccia Utente e Funzionalità

L’interfaccia utente di “AICare” è stata progettata per offrire un’esperienza d’uso intuitiva ed accessibile ai medici, consentendo loro di interagire con i dati clinici e con il modello Mistral in modo efficiente. Data la natura clinica dell’applicazione, l’interfaccia non solo deve risultare intuitiva, ma deve anche minimizzare il rischio di errori nell’interpretazione delle informazioni e ottimizzare il flusso di lavoro del medico.

Figura 4.3: Palette dei colori di “AICare”.

La piattaforma segue un'architettura modulare, in cui ogni componente React è stato sviluppato per gestire una specifica funzionalità, garantendo un'esperienza fluida e senza interruzioni. A differenza di un'interfaccia tradizionale basata su una dashboard centralizzata, “AICare” utilizza una struttura più flessibile.

In questa sezione verranno illustrati i principi di progettazione dell'interfaccia e le funzionalità chiave della piattaforma, con un focus sulla navigazione e sull'accesso ai dati clinici.

4.2.1 Principi di Progettazione dell'UI

La progettazione dell'interfaccia di “AICare” si basa su alcuni principi fondamentali di usabilità e *user experience* (UX), con l'obiettivo di semplificare il lavoro del medico e migliorare l'accessibilità dei dati clinici.

Architettura Modulare. La struttura dell'interfaccia si basa su componenti React separati, ognuno con una funzione specifica. Questo approccio ha diversi vantaggi: in primis, ogni componente è indipendente e può essere facilmente aggiornato o modificato senza impattare l'intero sistema. Inoltre, l'uso di *stati globali* e *props* consente di mantenere una comunicazione fluida tra i vari componenti. Infine, grazie a questo tipo di architettura, la piattaforma può essere facilmente estesa aggiungendo nuove funzionalità senza stravolgere l'interfaccia.

Usabilità e User Experience. L'uso di “AICare” è pensato per minimizzare il carico cognitivo del medico, rendendo l'accesso ai dati immediato e chiaro. Alcuni elementi chiave della UX includono:

1. *Interazione rapida.* Il sistema è stato progettato per ridurre al minimo il numero di click necessari per accedere alle informazioni chiave, senza risultare però troppo pesante.

2. *Struttura intuitiva.* I dati vengono presentati in modo gerarchico e ben organizzato, evitando un sovraccarico informativo.

3. *Minimizzazione degli errori.* Le informazioni vengono presentate in modo strutturato e leggibile, riducendo il rischio di interpretazioni errate.

4. *Compatibilità con dispositivi diversi.* Grazie a Tailwind CSS, l'interfaccia è completamente responsiva, adattandosi a schermi di diverse dimensioni (desktop, tablet).

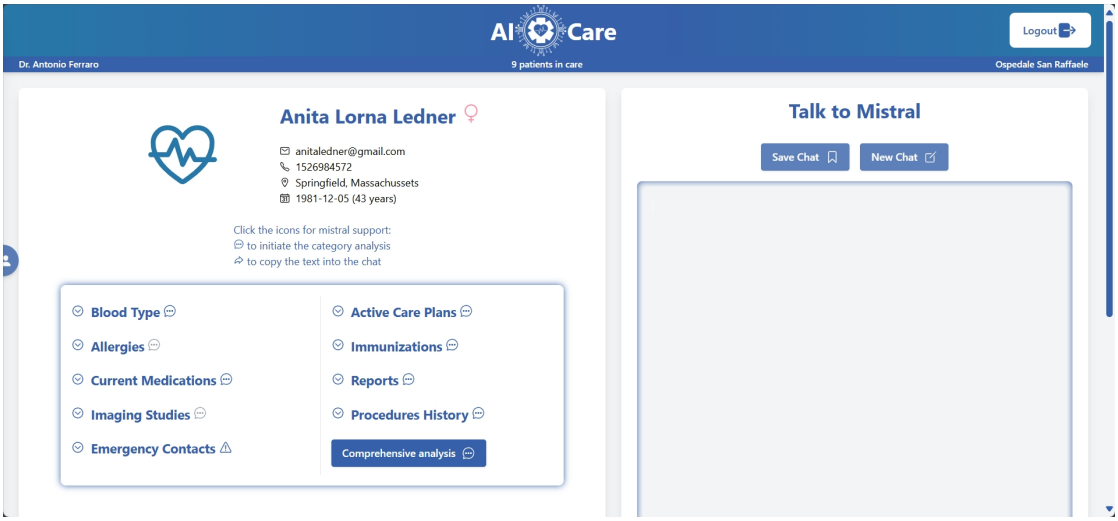

Figura 4.4: Visualizzazione parziale della pagina Home di “AICare”.

Design Pulito e Intuitivo. L’utilizzo di Tailwind CSS ha permesso di sviluppare un’interfaccia moderna e minimale, garantendo leggibilità e organizzazione chiara dei contenuti ed evitando elementi ridondanti o distrazioni. I vantaggi di questa scelta includono:

1. *Forte personalizzazione.* Poichè l’utilizzo di Tailwind CSS avviene direttamente inline, nei componenti React, è stato possibile definire stili diversi per componenti uguali in modo rapido ed efficace.

2. *Velocità di sviluppo.* Ciò ha accelerato il processo di sviluppo e migliorato la manutenzione dell’interfaccia.

4.2.2 Navigazione nella Piattaforma

L’interfaccia di “AICare” è stata progettata per garantire una navigazione fluida e intuitiva, permettendo al medico di accedere rapidamente alle funzionalità essenziali dell’applicazione.

Schermata di Login. L’accesso alla piattaforma avviene attraverso una pagina di login sicura, che richiede le credenziali dell’utente al fine di garantire che solo personale autorizzato possa accedere ai dati clinici. Il processo di autenticazione è stato implementato con un sistema di gestione delle sessioni, che mantiene l’utente connesso per un periodo di tempo limitato o finché non chiude la scheda, migliorando così la sicurezza.

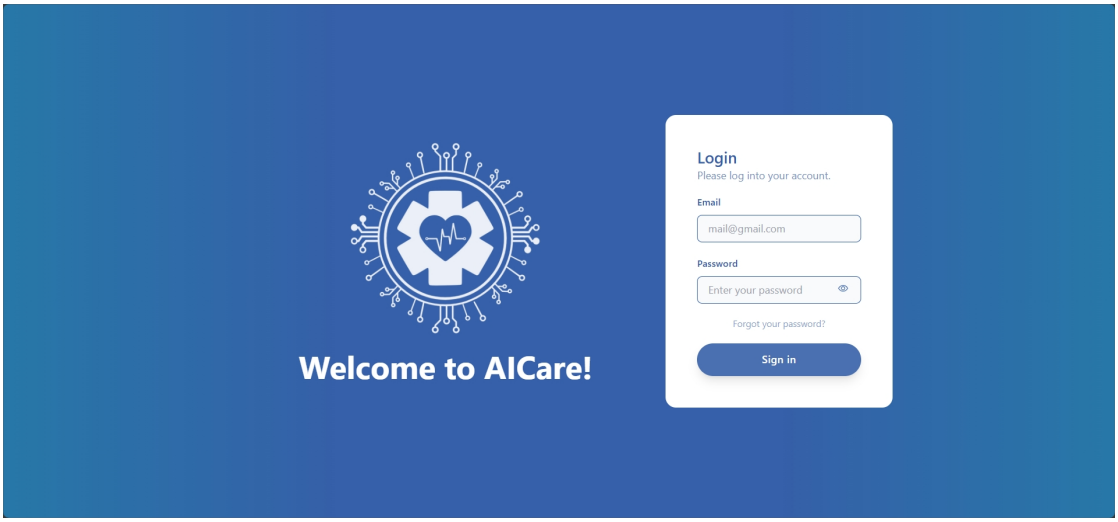

Figura 4.5: Schermata di Login.

Home Page e Componenti Principali. Dopo il login, il medico viene reindirizzato alla Home, che rappresenta il cuore dell’applicazione. A differenza di un’interfaccia basata su più pagine separate, “AICare” sfrutta una Home divisa in più componenti interattivi, ciascuno con una funzione specifica:

1. *Navbar e Subnavbar.* Nella Home sono stati integrati due componenti navbar. La prima, ovvero la più grande, è composta semplicemente dal logo di “AICare” e dal pulsante per effettuare il logout. La Subnavbar, più sottile, è stata pensata per mostrare al dottore, oltre al suo nome e il nome dell’ospedale o del distretto in cui lavora, un conteggio reale dei pazienti a cui fornisce assistenza.

2. *Sidebar.* La Sidebar a comparsa contiene un elenco di *card* che mostrano nome, età e genere dei pazienti in cura, fornendo così una panoramica degli

assistiti. È presente anche una *search bar* che consente la ricerca rapida, tramite nome, di uno specifico paziente. Nella Sidebar è stato scelto di inserire un pulsante per permettere al medico di aggiungere un paziente alla piattaforma tramite l'inserimento dei dati in una modale a comparsa. Quando viene cliccata la *card* di un paziente, esso viene visualizzato nella PatientCard.

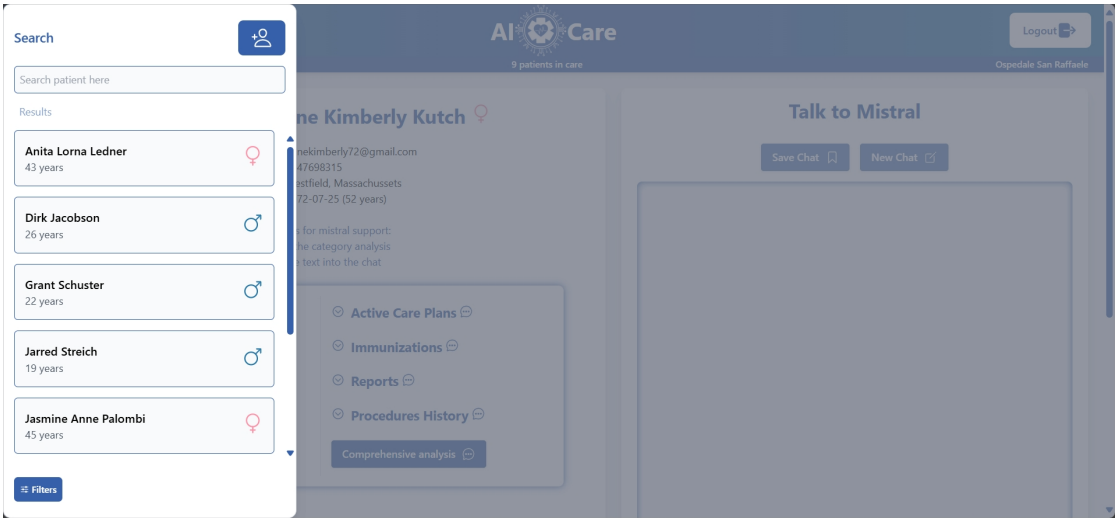

Figura 4.6: Sidebar a comparsa.

3. *PatientCard.* Questo componente mostra le informazioni cliniche del paziente selezionato. Al suo interno è presente una sezione in rilievo che consente di visualizzare i dati non anagrafici del soggetto tramite tendine espandibili. Accanto a ogni macrocategoria è presente un pulsante che invia automaticamente le informazioni contenute in essa al backend, avviando così il riassunto e l'analisi dei dati da parte dell'LLM.
   Un carosello con scroll orizzontale consente di visualizzare i documenti, in formato *PDF* o *.txt*, caricati dal medico tramite un apposito pulsante.
   Infine, è presente un pulsante che abilita la comparsa di una modale che consente al medico di aggiungere o aggiornare le informazioni del paziente.

4. *LLMInterface.* Questo componente consente di interagire direttamente tramite input testuale con Mistral 7B OpenOrca. L'interfaccia permette di visualizzare i messagi scambiati tra il medico, in azzurro, e l'LLM, in verde. In alto sono presenti due pulsanti: uno consente di avviare una nuova chat, mentre l'altro abilita il salvataggio dei messaggi presenti nella chat all'interno del database. I nomi delle conversazioni salvate vengono mostrati nella sezio-

ne *“Chat History”*; tali conversazioni possono essere cancellate dal database o visualizzate nuovamente nell’interfaccia tramite dei pulsanti appositi.

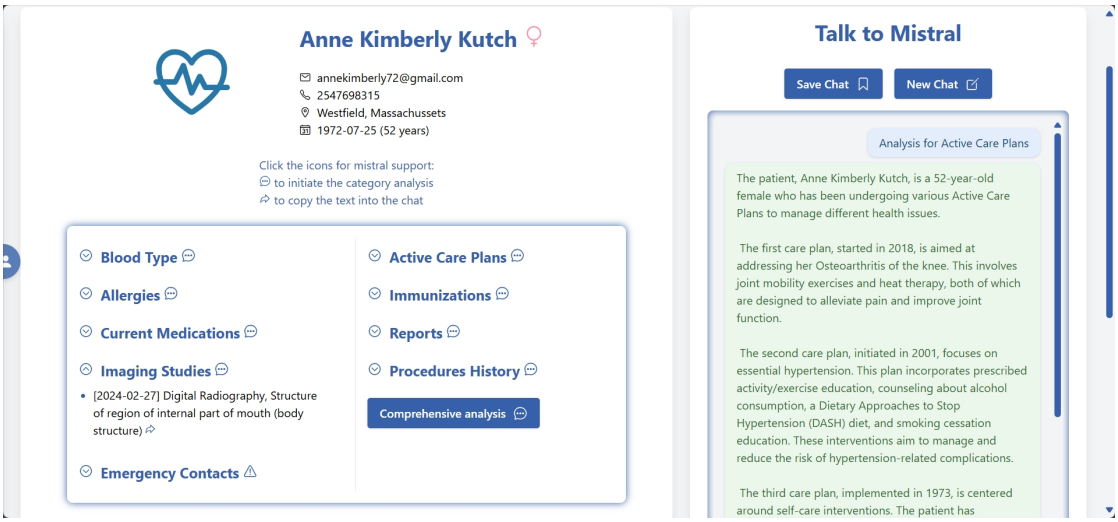

Figura 4.7: Visualizzazione di una conversazione. I messaggi in azzuro sono quelli inviati in automatico dal sistema o dal medico, i messaggi in verde sono le risposte di Mistral.

4.2.3 Accesso ai Dati del Paziente

Quando un medico seleziona un paziente, nella *PatientCard* vengono visualizzate tutte le informazioni cliniche disponibili. I dati demografici del paziente sono presentati direttamente, fornendo al medico l’accesso immediato alle informazioni di contatto.

Le condizioni mediche pregresse e attuali, le terapie in corso, le prescrizioni di farmaci, gli esami diagnostici e altri dati clinici sono, invece, organizzati in tendine espandibili in modo da non sovraccaricare l’utente di informazioni.

Modalità di Interazione con i Dati. Uno degli aspetti più innovativi di “AI-Care” è l’integrazione diretta tra i dati clinici e il modello AI. Il medico non solo può consultare le informazioni in modo tradizionale, ma può anche:

1. *Ottenere spiegazioni dettagliate sui dati.* Tramite il pulsante della *bubble-chat*, le informazioni presenti nella macrocategoria vengono automaticamente inviate dal sistema all’LLM; inoltre, viene mostrato un messaggio preimpostato come se fosse stato inviato dal medico. Quindi, il backend processa i dati, tramite le tecniche di Prompt Engineering già viste, e avvia la conversazione con Mistral. La risposta, consistente in un riassunto e un’ana-

lisi delle informazioni, viene mostrata anch'essa nell'interfaccia dedicata allo scambio di messaggi.

2. *Generare un analisi della storia clinica.* Tramite il pulsante “Comprehensive Analysis”, seguendo lo stesso meccanismo spiegato prima, vengono mandati tutti i dati clinici del paziente a Mistral, il quale genera un parere personale sulla condizione attuale.

3. *Copiare rapidamente i dati.* Per sollevare il medico dal compito oneroso del copia-incolla, vicino a tutti i singoli campi dei dati è stato introdotto un pulsante di inoltro che copia i dati scelti direttamente nella text-area dedicata all'invio di messaggi a Mistral.

Sicurezza e Protezione dei Dati Clinici. Un aspetto essenziale, per qualsiasi applicazione che tratta dati sanitari, è la protezione dei dati e della privacy dei pazienti.

“AICare” è stato progettato con l'obiettivo di garantire una gestione sicura dei dati, anche se attualmente non sono state implementate misure avanzate di filtraggio delle informazioni. Attualmente, il sistema permette al medico di visualizzare direttamente i dati estratti dal database, ma non sono ancora state introdotte restrizioni automatiche per evitare l'accesso a informazioni sensibili non elaborate. Questo implica che l'LLM potrebbe generare risposte contenenti dettagli direttamente estratti dai dati disponibili nel database.

Tuttavia, per garantire un utilizzo corretto della piattaforma, è fondamentale considerare possibili miglioramenti futuri nella gestione della sicurezza, tra cui anche livelli di accesso personalizzati che limitino la visualizzazione dei dati in base ai ruoli degli utenti. Ad esempio, in un contesto ospedaliero, solo determinati professionisti potrebbero accedere a informazioni più dettagliate.

Capitolo 5

Analisi dei Risultati

L'uso di Large Language Models nel settore medico rappresenta una delle sfide più complesse dell'intelligenza artificiale, poiché richiede un elevato grado di affidabilità e precisione. Errori nell'interpretazione dei dati o nella generazione delle risposte potrebbero portare a implicazioni cliniche significative. Risulta quindi fondamentale valutare le capacità di un LLM quando viene integrato in un sistema di supporto medico [6].

L'obiettivo principale di questo capitolo è analizzare i risultati della valutazione del modello *Mistral 7B OpenOrca Q8* da parte di cinque medici, al fine di comprendere le sue prestazioni nel contesto dell'HealthCare digitale. Durante la valutazione, i medici hanno analizzato la coerenza, la completezza e l'affidabilità delle risposte generate dal modello, evidenziando sia i punti di forza che le aree di miglioramento.

Nei prossimi paragrafi verranno presentati i risultati ottenuti, evidenziando le tendenze emerse e verranno discusse le possibili strategie di miglioramento, mettendo in luce possibili sviluppi futuri.

5.1 Strategie Adottate per la Valutazione

La valutazione dell'LLM è stata effettuata attraverso l'analisi di un documento esplicativo, intitolato *"Valutazione del modello"*, seguito dalla compilazione di un questionario strutturato su *Google Forms*. Questo processo ha permesso di raccogliere il feedback dei cinque medici riguardo l'affidabilità, l'utilità e le limitazioni di Mistral nel contesto clinico.

L'approccio adottato nella costruzione del questionario si basa su standard consolidati per la valutazione dell'AI in ambito sanitario e su modelli di analisi validati per l'usabilità e la trasparenza dell'intelligenza artificiale nei contesti medici.

5.1.1 Struttura del documento "Valutazione del modello"

A causa di problemi tecnici legati al server fuori uso, i medici non hanno potuto interagire direttamente con la piattaforma AICare. Per ovviare a questa limitazione,

è stato predisposto un documento per guidare la valutazione del modello Mistral 7B OpenOrca, includendo informazioni dettagliate e casi di studio simulati.

Il documento presenta una serie di otto casi di studio, ciascuno relativo a un paziente fittizio generato tramite *Synthea*. Per ogni caso:

1. Vengono forniti i dati demografici del paziente e le informazioni strutturate che sono state passate al modello nelle interrogazioni, come patologie, terapie in corso, piani di cura, farmaci assunti e risultati di esami diagnostici.
   I dati sono stati estratti in modo da rappresentare situazioni cliniche eterogenee, includendo pazienti con ipertensione, diabete, disturbi dell'umore, condizioni cardiovascolari, malattie infettive e oncologiche.

2. Dopo la presentazione dei dati clinci, vengono riportate le domande, relative al paziente, che sono state poste al modello. Queste domande o sono la spiegazione delle funzionalità preimpostate dal sistema, come *"Comprehensive Analysis"*, oppure rappresentano interrogazioni che un medico potrebbe formulare in un contesto clinico reale.

3. Ad ogni domanda segue la risposta del modello, riportata integralmente nel documento.

Al termine della lettura del documento, è stato sottoposto ai medici un questionario per raccoglierne i feedback.

5.1.2 Struttura del Questionario

Il questionario di valutazione di Mistral 7B OpenOrca è stato progettato basandosi su principi consolidati nella valutazione di strumenti di intelligenza artificiale in ambito sanitario. Esso è stato strutturato seguendo un approccio metodologico che integra linee guida internazionali per la valutazione dell'AI in ambito clinico, modelli di accettazione della tecnologia sanitaria e strumenti per la misurazione dell'usabilità e dell'affidabilità dei modelli di intelligenza artificiale.

Standard di riferimento. Per la valutazione dell'efficacia di Mistral nell'analisi dei dati clinici si è fatto riferimento a diverse linee guida scientificamente validate, tra cui:

- CONSORT-AI (*Liu et al., 2020*) ovvero uno standard per la valutazione di modelli AI in studi clinici [21].

- SPIRIT-AI (*Riviera et al., 2020*), ovvero dei criteri per la trasparenza e riproducibilità dell'AI in medicina [32].

Per misurare l'esperienza dei medici nell'interazione con l'LLM e l'accettabilità e la fiducia dell'AI in medicina , il questionario prende come riferimento:

- mHealth App Usability Questionnaire (MAUQ) utilizzato per valutare l'esperienza d'uso di applicazione AI nel settore sanitario [40].
- Technology Acceptance Model (TAM), che valuta il grado di accettazione di una nuova tecnologia da parte degli utenti [13].

Struttura delle domande. Anche le domande sono state progettate sulla base di studi scientifici riguardanti l'uso dell'intelligenza artificiale nel supporto clinico. In particolare, le ricerche hanno ispirato le seguenti aree di valutazione del questionario:

- *Analisi Globale.* Una valutazione della capacità del modello di estrarre informazioni utili [25].
- *Interpretazione degli esami di laboratorio.* Sezione per valutare la precisione delle risposte relative ai risultati diagnostici [5].
- *Piani di trattamento e follow-up.* Valutazione dell'affidabilità del modello nell'elaborazione di suggerimenti terapeutici [27].
- *Informazioni sui farmaci.* Sezione per valutare la correttezza delle informazioni sui medicinali [36].
- *Utilità dell'LLM nel supporto clinico.* Per valutare la capacità del modello di affiancare il medico nelle decisioni [18].

Scala di Valutazione. Per garantire una misurazione quantitativa del grado di accordo dei medici con le domande proposte, è stata utilizzata la *Likert Scale*. Secondo questo tipo di scala, ad ogni risposta sono associate cinque possibili risposte, di cui la terza rappresenta il punto neutrale. L'uso di questa scala consente di ottenere dati oggettivi e confrontabili, facilitando l'analisi delle percezioni dei medici riguardo l'accuratezza e l'affidabilità dell'LLM. Infatti, ad ogni risposta predefinita è possibile associare un numero da 1 a 5 per esprimere il grado di accordo con la domanda.

5.2 Risultati Ottenuti

I medici che hanno partecipato alla valutazione di Mistral sono stati cinque e tutti sono stati piacevolmente sorpresi dalla qualità delle risposte dell'LLM.

Per interpretare in modo efficace le risposte fornite dai partecipanti nel questionario, è stata adottata una strategia di analisi su due livelli.

5.2.1 Analisi per Sezioni

In un primo momento le domande sono state analizzate facendo riferimento alle aree di valutazione previste nel questionario e riportate sopra. I risultati sono stati elaborati in due istogrammi:

1. Il primo rappresenta la valutazione media espressa da ciascun medico per ogni area. Il massimo punteggio ottenibile è di 5.

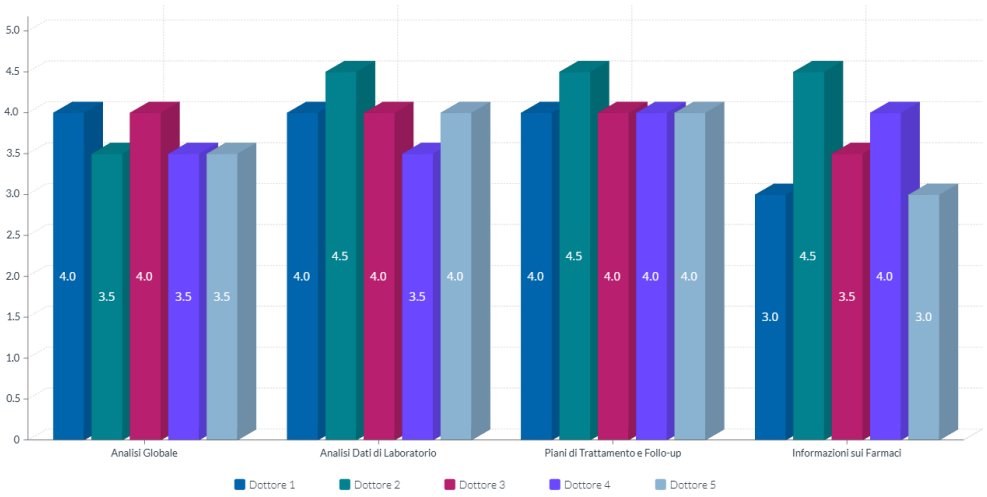

Figura 5.1: Istogramma 1 per l'analisi delle sezioni.

2. Il secondo istogramma serve a fornire una migliore rappresentazione visiva dell'andamento delle aree di valutazione. Il punteggio massimo è di 25.

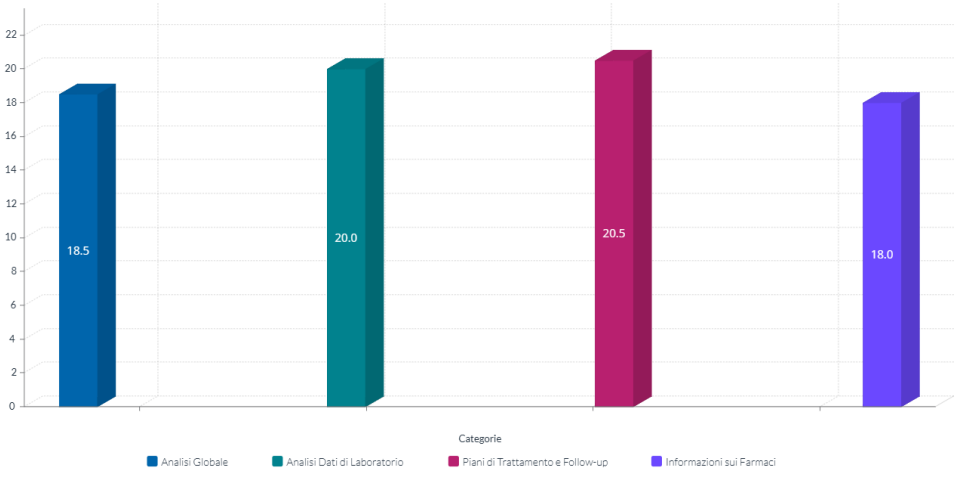

Figura 5.2: Istogramma 2 per l'analisi delle sezioni.

Da questi grafici si evince come il modello ha ottenuto valutazioni mediamente positive in tutte le categorie. L'area relativa alle informazioni sui farmaci ha mostrato la maggiore dispersione nei punteggi, suggerendo che è un punto critico da migliorare. Invece, le valutazioni più alte sono state ottenute nei piani di trattamento e follow-up, ciò indica che le raccomandazioni terapeutiche fornite sono state generalmente utili e affidabili.

5.2.2 Analisi per Macrocategorie

. Successivamente, le domande del questionario sono state raggruppate in tre macrocategorie principali:

1. *Accuratezza e Coerenza.* Include cinque domande relative alla capacità del modello di fornire risposte precise e affidabili rispetto all'analisi clinica del paziente.

2. *Utilità.* Raccoglie due domande riguardanti il valore del supporto fornito da Mistral nella valutazione diagnostica e nelle raccomandazioni terapeutiche.

3. *Affidabilità.* Include una domande relativa alla conformità delle risposte dell'LLM rispetto agli standard clinici.

Anche in questa analisi sono stati elaborati due istogrammi simili ai precedenti, la necessità del secondo in questo caso è stata dovuta allo squilibrio nel numero di domande per ogni macrocategoria. Per normalizzare i punteggi è stata quindi effettuata una media.

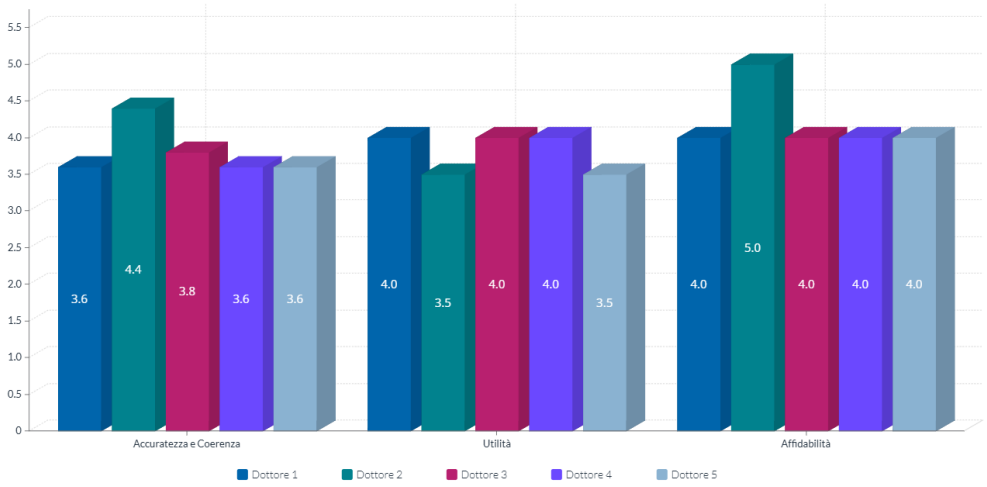

Figura 5.3: Istogramma 2 per l'analisi delle macrocategorie.

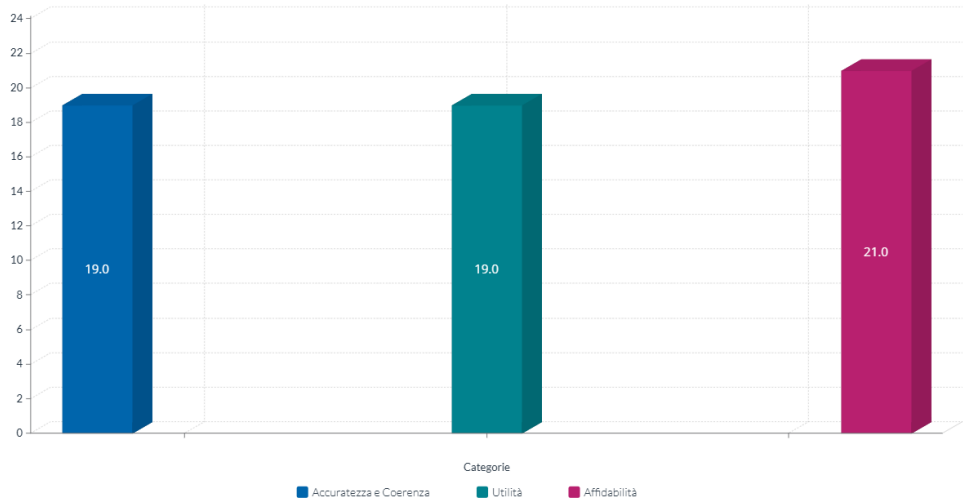

Figura 5.4: Istogramma 2 per l'analisi delle macrocategorie.

L'affidabilità ha ricevuto il punteggio complessivo più alto, mentre le altre due macrocategorie hanno una media leggermente più bassa, indicando che ci sono margini di miglioramento.

Capitolo 6

Conclusioni

L'obiettivo del lavoro di tesi è stato quello di studiare in primis le potenzialità dell'utilizzo degli LLM in ambito HealthCare, e poi quello di realizzare una piattaforma di supporto ai medici nell'analisi e nella gestione dei pazienti.

I risultati ottenuti dal questionario confermano che *Mistral 7B OpenOrca* possiede un elevato potenziale di applicabilità in ambito clinico, soprattutto per quanto riguarda l'analisi diagnostica e la formulazione dei piani terapeutici. Le valutazioni espresse dai medici mostrano una buona coerenza tra le diverse categorie, con un livello di accuratezza e utilità ritenuto soddisfacente per la maggior parte dei compiti clinici valutati. Tuttavia, emergono alcune aree di miglioramento che potrebbero essere affrontate in sviluppi futuri della piattaforma.

Un'ulteriore conferma dell'utilità del modello è fornita dal grafico a torta riportato in figura 6.1, che illustra la percezione dei medici riguardo al possibile impiego di un LLM come Mistral nella pratica clinica. Nessuno dei partecipanti ha infatti escluso l'utilità di un modello AI in sanità, ma la maggior parte ha sottolineato il suo ruolo come strumento di supporto. Il 40% dei medici ha ritenuto che il modello sia utile per la consultazione dei dati clinici, mentre il restante 60% lo ha visto come un potenziale ausilio nella diagnosi, nei suggerimenti terapeutici e nella gestione farmacologica. Questo evidenzia la necessità di una maggiore integrazione tra l'LLM e i flussi di lavoro clinici per sfruttare appieno il suo potenziale.

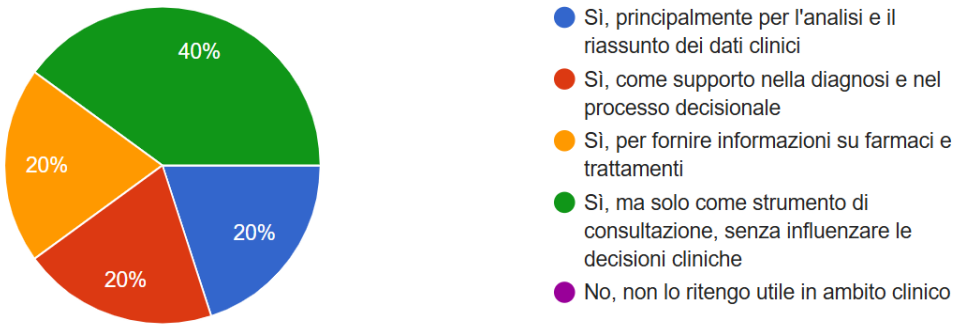

Figura 6.1: Percezione dell'utilità di un modello AI come Mistral nella gestione clinica.

Nel complesso, i risultati ottenuti suggeriscono che *Mistral 7B OpenOrca* può rappresentare un valido supporto per i medici, evidenziando il valore aggiunto dei Large Language Models nel supportare la pratica clinica. Tuttavia, permangono alcune limitazioni che dovranno essere mitigate in sviluppi futuri e ulteriori miglioramenti e validazioni su larga scala saranno necessari per garantire un impiego sicuro ed efficace nel contesto medico reale.

6.1 Sviluppi futuri

L'analisi dei risultati ottenuti ha evidenziato il potenziale di *Mistral 7B OpenOrca* come supporto decisionale per i medici. Per potenziare l'efficacia del sistema *"AI-Care"*, si delineano tre principali strategie di sviluppo futuro.

Un primo miglioramento riguarda l'**ottimizzazione dei prompt**, con l'obiettivo di rendere le risposte dell'LLM più coerenti e aderenti alle necessità cliniche. Infatti, come emerso dalla valutazione, alcune risposte potrebbero risultare più precise se il modello venisse guidato con istruzioni più dettagliate e contestualizzate. L'introduzione di tecniche avanzate di Prompt Engineering, come il *Chain-of-Thought Prompting*, che spinge il modello a suddividere le risposte in passaggi logici, o l'uso di *Few-Shot Prompting* con esempi mirati, potrebbe migliorare la coerenza e l'accuratezza delle informazioni fornite. Inoltre, un'ulteriore personalizzazione del prompt potrebbe permettere al modello di adattarsi meglio a specifici domini clinici, garantendo risposte più pertinenti e affidabili.

Un'altra possibile evoluzione naturale è l'integrazione della **Retrieval - Augmented Generation** (RAG). Attualmente, uno dei limiti principali di Mistral, così come di molti altri LLM, è la dipendenza dai dati del proprio training set, che potrebbero essere non aggiornati o sufficientemente dettagliati per applicazioni cliniche complesse. Implementare un sistema di RAG permetterebbe al modello di accedere a fonti di conoscenza esterne durante la generazione delle risposte. Implementando un database di lettura medica, linee guida cliniche aggiornate e dati farmaceutici ufficiali, il sistema potrebbe migliorare la qualità delle informazioni fornite, riducendo il rischio di errori e garantendo un supporto più affidabile ai medici. Questa soluzione, inoltre, permetterebbe al modello di rispondere a domande complesse basandosi su fonti cliniche recenti, anziché limitarsi alle informazioni statiche dell'addestramento.

Infine, un'estensione particolarmente promettente riguarda l'integrazione con un **Patient Digital Twin** (PDT), ovvero una rappresentazione digitale dinamica del paziente, basata su dati clinici storici e in tempo reale. l'integrazione di

un PDT consentirebbe al modello di adattare le risposte sulla base di una rappresentazione più dettagliata e dinamica dello stato clinico, piuttosto che affidarsi solamente ai dati statici passati nei prompt. Questo approccio potrebbe migliorare la capacità predittiva del sistema rendendolo un supporto ancora più efficace per la personalizzazione dei trattamenti e dei piani di cura. Inoltre, sarebbe possibile simulare l'evoluzione delle condizioni del paziente, fornendo così un ausilio concreto per la presa di decisioni mediche.

In prospettiva, queste strategie potrebbero rendere *"AICare"* non solo un assistente virtuale avanzato, ma anche uno strumento proattivo capace di contribuire in modo più significativo alla personalizzazione delle cure e al miglioramento della sanità.

Bibliografia

[1] DeepSeek AI. *DeepSeek-V2: Advancing Open-Source LLMs with Efficient Architecture*. Accessed: 2025-01-27. 2024. URL: https://github.com/deepseek-ai/DeepSeek-LLM.

[2] Mistral AI. *Announcing Mistral 7B*. 2023. URL: https://mistral.ai/news/announcing-mistral-7b/.

[3] Joshua Ainslie et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*. 2023. arXiv: 2305.13245 [cs.CL]. URL: https://arxiv.org/abs/2305.13245.

[4] Jay Alammar. *The Illustrated Word2vec*. Accessed: 2025-01-27. 2018. URL: https://jalammar.github.io/illustrated-word2vec/.

[5] Andrew L. Beam e Isaac S. Kohane. "Big Data and Machine Learning in Health Care". In: *JAMA* 319.13 (2018), pp. 1317–1318. DOI: 10.1001/jama.2017.18391.

[6] Emily M. Bender et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (2021), pp. 610–623. DOI: 10.1145/3442188.3445922.

[7] B. Björnsson et al. "Digital twins to personalize medicine". In: *Genome Medicine* 12.1 (2020), p. 4. DOI: 10.1186/s13073-019-0701-3.

[8] The Bloke. *Mistral 7B OpenOrca GGUF - Quantized*. 2023. URL: https://huggingface.co/TheBloke/Mistral-7B-OpenOrca-GGUF.

[9] Tom Brown et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1877–1901. DOI: 10.48550/arXiv.2005.14165.

[10] Koen Bruynseels, Filippo Santoni de Sio e Jeroen van den Hoven. "Digital twins in health care: Ethical implications of an emerging engineering paradigm". In: *Frontiers in Genetics* 9 (2018), p. 31. DOI: 10.3389/fgene.2018.00031.

[11] Google Cloud. *Introducing MedLM for the healthcare industry*. Accessed: 2025-01-28. 2024. URL: https://cloud.google.com/blog/topics/healthcare-life-sciences/introducing-medlm-for-the-healthcare-industry.

[12] The MITRE Corporation. *Synthea Wiki - Synthetic Patient Population Simulator*. Accessed: 2025-01-29. 2024. URL: https://github.com/synthetichealth/synthea/wiki.

[13] Fred D. Davis. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Quarterly* 13.3 (1989), pp. 319–340. DOI: 10.2307/249008.

[14] Jacob Devlin et al. "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *NAACL-HLT* (2019). URL: https://arxiv.org/abs/1810.04805.

[15] Agenda Digitale. *AI e Data Privacy: La necessaria sinergia tra normativa e ricerca*. Accessed: 2025-01-27. 2024. URL: https://www.agendadigitale.eu/sicurezza/privacy/ai-e-data-privacy-la-necessaria-sinergia-tra-normativa-e-ricerca/.

[16] Jeffrey L. Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211.

[17] Michael Grieves e John Vickers. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems". In: *Transdisciplinary Perspectives on Complex Systems*. 2017, pp. 85–113. DOI: 10.1007/978-3-319-38756-7_4.

[18] Jin He et al. "The practical implementation of artificial intelligence technologies in medicine". In: *Nature Medicine* 25.1 (2019), pp. 30–36. DOI: 10.1038/s41591-018-0307-0. URL: https://doi.org/10.1038/s41591-018-0307-0.

[19] Leroy Hood e Stephen H. Friend. "Predictive, personalized, preventive, participatory (P4) cancer medicine". In: *Nature Reviews Clinical Oncology* 8.3 (2011), pp. 184–187. DOI: 10.1038/nrclinonc.2010.227.

[20] Patrick Lewis et al. "Retrieval-augmented generation for knowledge-intensive NLP tasks". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 9459–9474. DOI: 10.5555/3495724.3496517.

[21] Xiaoxuan Liu et al. "CONSORT-AI extension: Reporting guidelines for clinical trials evaluating artificial intelligence interventions". In: *Nature Medicine* 26.9 (2020), pp. 1364–1374. DOI: 10.1038/s41591-020-1034-x.

[22] Shayne Longpre et al. *The Flan Collection: Designing Data and Methods for Effective Instruction Tuning*. 2023. arXiv: 2301.13688 [cs.AI]. URL: https://arxiv.org/abs/2301.13688.

[23] Miehleketo Mathebula, Abiodun Modupe e Vukosi Marivate. "Fine-Tuning Retrieval-Augmented Generation with an Auto-Regressive Language Model for Sentiment Analysis in Financial Reviews". In: *Applied Sciences* 14.23 (2024), p. 10782. DOI: 10.3390/app142310782.

[24] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[25] Riccardo Miotto et al. "Deep learning for healthcare: Review, opportunities and challenges". In: *Briefings in Bioinformatics* 19.6 (2017), pp. 1236–1246. DOI: 10.1093/bib/bbx044.

[26] Subhabrata Mukherjee et al. *Orca: Progressive Learning from Complex Explanation Traces of GPT-4*. 2023. arXiv: 2306.02707 [cs.CL]. URL: https://arxiv.org/abs/2306.02707.

[27] Ziad Obermeyer e Ezekiel J. Emanuel. "Predicting the Future — Big Data, Machine Learning, and Clinical Medicine". In: *New England Journal of Medicine* 375.13 (2016), pp. 1216–1219. DOI: 10.1056/NEJMp1606181.

[28] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI Blog* 1.8 (2019). URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[29] Alec et al. Radford. "Learning Transferable Visual Models From Natural Language Supervision". In: *arXiv preprint arXiv:2103.00020* (2021). URL: https://arxiv.org/abs/2103.00020.

[30] Avijit Ram, Aravind Chandrasekaran, Ankit Gupta et al. "Combining Retrieval-Augmented Generation and Fine-Tuning for Domain-Specific Question Answering". In: *arXiv preprint arXiv:2312.05934* (2023). DOI: 10.48550/arXiv.2312.05934.

[31] Google Research. *Med-PaLM 2: Advancing AI for Healthcare*. Accessed: 2025-01-27. 2023. URL: https://ai.googleblog.com/2023/07/introducing-medpalm-2.html.

[32] Samuel C. Rivera et al. "SPIRIT-AI and CONSORT-AI: Guidelines for reporting artificial intelligence interventions in trials". In: *BMJ* 370 (2020), p. m3164. DOI: 10.1136/bmj.m3164.

[33] Daria Simonetti. *AICare*. Accesso il: 8 Febbraio 2025. 2024. URL: https://github.com/dariasimonetti/AICare.

[34] Yuhan Sun et al. “To be or not to be? An exploration of continuously controllable prompt engineering”. In: *arXiv preprint arXiv:2311.09773* (2023). DOI: 10.48550/arXiv.2311.09773.

[35] Sandesh Swamy et al. *Contextual Dynamic Prompting for Response Generation in Task-oriented Dialog Systems*. 2023. arXiv: 2301.13268 [cs.CL]. URL: https://arxiv.org/abs/2301.13268.

[36] Nicholas P. Tatonetti et al. “Data-driven prediction of drug effects and interactions”. In: *Science Translational Medicine* 4.125 (2012), 125ra31. DOI: 10.1126/scitranslmed.3003377. URL: https://www.science.org/doi/10.1126/scitranslmed.3003377.

[37] Hugo Touvron et al. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint arXiv:2302.13971* (2023). URL: https://arxiv.org/abs/2302.13971.

[38] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30 (2017), pp. 5998–6008. URL: https://dl.acm.org/doi/10.5555/3295222.3295349.

[39] Jason Wei et al. “Chain of thought prompting elicits reasoning in large language models”. In: *arXiv preprint arXiv:2201.11903* (2022). DOI: 10.48550/arXiv.2201.11903.

[40] Liang Zhou et al. “The mHealth App Usability Questionnaire (MAUQ): Development and Validation Study”. In: *JMIR Mhealth Uhealth* 7.4 (2019), e11500. DOI: 10.2196/11500. URL: https://doi.org/10.2196/11500.

Ringraziamenti

Desidero esprimere la mia più profonda gratitudine a tutti coloro che mi hanno sostenuto lungo il percorso che ha portato alla conclusione di questo cammino accademico.

Ringrazio mamma e papà, per avermi permesso di coltivare i miei studi e avermi supportato continuamente. Grazie per aver messo dei sogni nelle mie mani con i vostri costanti sacrifici e per avermi salvata quando mi stavo perdendo. Grazie per avermi educata come una donna indipendente, con valori e principi di cui sono fiera. Grazie per il vostro coraggio e per il vostro impegno costante che mi hanno spronata a non arrendermi mai e a dare sempre il massimo. Abbiamo litigato spesso durante questo percorso ma è soprattutto grazie a voi che sono riuscita a raggiungere il traguardo. Vi sarò eternamente riconoscente per tutto ciò che avete fatto e continuerete a fare per me.

Grazie ai miei fratelli, Chiara e Davide, per avermi supportata, incoraggiata ed essere stati degli esempi da seguire mentre affrontavamo lo stesso viaggio. Davide, sei una delle persone più intelligenti che io conosca, la tua voglia di studiare e conoscere è stata di ispirazione, grazie. Chiara, io e te abbiamo avuti alti e bassi ma non c'è mai stato un momento in cui ho pensato davvero che non ci saresti stata per me. Tre anni fa, quando passai due giorni fuori casa, mi scrivesti facendomi un discorso sulle priorità della vita; sappi che senza quei messaggi, che mi sono tornati alla mente nei momenti più bui di questo percorso, probabilmente oggi non sarei ancora arrivata dove sono, grazie.

Ringrazio i miei nonni: nonna Titi, nonna Ester, nonno Enrico e nonno Enzo. Le vostre qualità mi hanno insegnato più di quanto possiate immaginare e, anche se non tutti potete condividere questo giorno con me, so che sareste orgogliosi del traguardo raggiunto. Grazie.

Ringrazio "zietta", zia Claudia, che nonostante la distanza mi è sempre stata vicina, sempre pronta a supportarmi, incoraggiarmi e a festeggiare con me il superamento di ogni esame. La tua presenza e il tuo sostegno sono fondamentali per me.

A tutta la mia famiglia, ai miei zii e ai miei cugini, un grazie di cuore per avermi sempre incoraggiata e sostenuta.

Quando ho cominciato l'università quattro anni fa, alla Federico II, mi ritrovai davanti un percorso che non mi piaceva e non mi ispirava, che non sentivo mio. Quando scelsi di cambiare facoltà, tu Fabio, sei stato il primo a supportarmi in questa scelta; abbiamo passato pomeriggi interi a parlare del mio futuro e ad esplorare le possibilità che mi si presentavano avanti. Grazie per avermi dato il coraggio di intraprendere una decisione che allora mi spaventava e mi risultava difficile. Grazie per esserci sempre stato per me, per avermi sempre supportato e per avermi anche fatto notare quando sbagliavo. Sei sempre stato un vero amico e ti sono immensamente grata.

Una volta cominciato il nuovo e attuale percorso ho avuto la fortuna di non intraprenderlo da sola, ma con Antonio, Carmine, Angelo e Nicolò. Carmine, mi hai accompagnato sia negli ultimi anni di liceo che in tutto il percorso universitario, e insieme ad Angelo e Nico siete stati i migliori colleghi universitari, nonché amici, che potessero capitarmi. Abbiamo condiviso risate, ansie pre esame e lunghi ed estenuanti periodi studio alternati ad altrettanto lunghi e felici periodi di nullafacenza; vi ringrazio per il supporto, la sopportazione e lo svago che mi avete offerto in questi anni. Senza di voi sarebbe stato sicuramente tutto molto più triste. Vi ringrazio di cuore ed auguro a tutti e tre il meglio che possa capitarvi nella vita.

Prima di iniziare l'università, ho cominciato anche a lavorare da "Listo!", qui ho conosciuto Erasmo e Silvia, Vi ringrazio immensamente per avermi aiutata ad affrontare le mie insicurezze, per avermi sempre offerto una spalla su cui piangere ma anche una mano per rialzarmi nelle mie cadute. Grazie per avermi supportata e incoraggiata ed essere sempre venuti incontro alle mie esigenze. Un grazie va a anche a tuttta la "Listo Gang", in particolare a Martina, Sabina e Maria. Oltre che mie colleghe siete state in primis mie amiche e confidenti, vi ringrazio di cuore.

L'anno scorso, giunti all'ultimo anno della triennale, ho avuto la fortuna di conoscere Luca Wood, Alice e tutti gli altri ragazzi del server. Ringrazio tutti voi per tutto il supporto che mi avete dato, avete riempito tutti le giornate dell'ultimo anno fornendomi il giusto quantitativo di distrazione che mi ha permesso di non arrivare all'esaurimento nervoso. Grazie di cuore siete stati "un pezzo" importante di questo cammino.

Infine voglio ringraziare Antonio, il mio ragazzo, che merita un ringraziamento speciale. Bimbo, non so esprimere tutta la gratitudine che provo nei tuoi confronti, non sei solo il mio compagno ma sei anche il mio migliore amico e siamo cresciuti insieme nel corso di questi anni. Abbiamo condiviso insieme tutti i momenti felici e tristi e sei sempre stato pronto a farti una risata con me, a festeggiare i miei risultati come se fossero i tuoi e ad asciugare le mie lacrime con le tue parole. Tu, più di tutti, mi hai sopportata nei miei nervosismi causati dallo studio, mostrandomi un pazienza infinita. Mi hai sostenuta aiutandomi a rialzarmi e ad affrontare gli ostacoli che mi si presentavano avanti. Hai calamato le mie ansie e preoccupazioni con parole dolci e piene di saggezza. Mi hai aiutato a concentrami sulle cose giuste nel momento giusto, e mi hai offerto la leggerezza e lo svago necessari a ricarmi le energie. Non è stato semplice arrivare qui oggi, ma mi hai insegnato che guardando solo il passo successivo e concentrandomi su di esso sarebbe stato più facile finire la strada, evitando di prendere una storta pensando a tutto il percorso. Averti al mio fianco mi ha fatta sentire tranquilla e protetta, fornendomi il coraggio e la forza necessari. È grazie al tuo affetto e alla tua gentilezza se oggi taglio un traguardo del cammino che conduce ai miei sogni. Hai creduto in me più di chiunque altro, più di me stessa. Ti sono immensamente grata per tutto, ti amo.