



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



“POC 2014/2020”

Azione 1.1.1

“Investimenti qualificati per lo
sviluppo e il potenziamento della capacità di ricerca e innovazione”

“UDMA- Universal Document Management Application”

D.D.G. n. 1767.5s del 28/10/2022

Progetto n. **086311190415** - CUP **G69J18001040007**

FASE 02 - “ANALISI ED IDENTIFICAZIONE DEI REQUISITI”

Deliverable D.13.3

Proof Of Concepts – Manuale di Installazione

Responsabile del documento: Vincenza Carchiolo

Versione del documento: **0.02**

Data Rilascio: **12/05/2025**

PARTENARIATO Progetto “UDMA” n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		 Università di Catania



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



REVISIONI

Versione	Data	Note	Autori
1.00	22/04/2025	Creazione del documento	Alfameg
1.01	23/05/2025	Cambio di formato e layout per renderlo aderente allo standard di progetto	M. Malgeri

DOCUMENTI RELAZIONATI

Documento	Descrizione
D13.01	Il documento descrive le problematiche della interazione uomo macchina con un focus specifico sulle tecniche di NLP e sulle interfacce conversazionali
D13.03	Descrizione del progetto e delle scelte tecniche per la creazione del POC

ACRONIMI

Acronimo	Descrizione
LM	Language Model
POC	Proof Of Concepts
OCR	Optical Character Recognition
GPT	Generative Pre-training Transformer
SPID	Sistema Pubblico di Identità Digitale

CONTENUTI

Revisioni.....	3
Documenti relazionati	3

PARTENARIATO Progetto “UDMA” n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



Acronimi	3
Contenuti	3
1 Introduzione	5
2 Flusso di lavoro	Errore. Il segnalibro non è definito.
3 Descrizione scelte tecniche	Errore. Il segnalibro non è definito.
3.1 Modelli di linguaggio per l'interazione uomo-macchina.	Errore. Il segnalibro non è definito.
3.1.1 Padronanza della lingua italiana	Errore. Il segnalibro non è definito.
3.1.2 Riconoscimento del contesto medico	Errore. Il segnalibro non è definito.
3.1.3 Adattabilità a errori	Errore. Il segnalibro non è definito.
3.2 Algoritmi per la discriminazione degli alberi decisionali.....	Errore. Il segnalibro non è definito.
3.2.1 Gini Index	Errore. Il segnalibro non è definito.
3.3 Algoritmi di classificazione	Errore. Il segnalibro non è definito.
3.3.1 Algoritmo KNN	Errore. Il segnalibro non è definito.
3.3.2 Metriche della distanza	Errore. Il segnalibro non è definito.
3.3.3 Applicazione dell'algoritmo nel progetto	Errore. Il segnalibro non è definito.
3.4 Strumenti per l'OCR	Errore. Il segnalibro non è definito.
4 Considerazioni finali	Errore. Il segnalibro non è definito.

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



1 INTRODUZIONE

Il progetto prevede la progettazione e l'implementazione di una POC che possa essere interrogata dagli utenti per recuperare record medici salvati su una base dati interna, contenente i documenti forniti dalle strutture sanitarie aderenti nel territorio della Regione Sicilia. L'applicazione deve essere in grado di interagire con l'utente ponendo domande al fine di identificare il documento di interesse e di adattarsi ai comportamenti e alle modalità di interazione dell'utente stesso.

Il progetto è stato sviluppato in Python, integrando le SDK di LM Studio, un software gratuito che permette di scaricare in locale un modello di linguaggio e interagire con lo stesso sia tramite console integrata nell'app desktop e sia tramite Python SDK tramite un server locale avviabile sempre dall'app desktop.

Per lo storage dei record e dei dati relativi agli utenti, sono utilizzate due basi di dati distinte. Una MySQL per le informazioni riguardanti gli utenti e una NoSQL, specificamente MongoDB per i record medici.

2 PREREQUISITI DI SISTEMA

Per eseguire il codice è necessario soddisfare i seguenti requisiti minimi: - - - -

Windows 11 (sia x64 che ARM)

Python 3.12+

Sono fortemente consigliati almeno 16 GB di RAM

Non è necessaria una GPU dedicata

La POC è stata progettata per poter essere eseguita su diversi tipi di macchine e, in base alle stesse, l'applicazione può garantire prestazioni differenti. Le modalità per modificare le impostazioni in base alle specifiche della macchina, vengono mostrate nel capitolo N.

Per avviare correttamente il codice è necessario avere installato LM Studio, versione 0.3.14+ [link al download], Docker [link al download] e un IDE a piacimento per l'esecuzione del codice come VS Code o PyCharm.

3 STRUTTURA POC

Il codice è strutturato per avere un core centrale e una serie di servizi accessori. Il design della struttura è tale da permettere facilmente la modifica di una o più parti di questi servizi senza intaccare l'intero codice. A tal proposito, ogni servizio può essere visto come una black-box con determinati input fissi e uno o più output anch'essi fissi. La modifica del contenuto della scatola non rende necessarie ulteriori operazioni in altre parti del progetto.

In questa sezione vengono analizzate tre componenti della POC:

1. Batch

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



2. Interazione con l'utente
3. Organizzazione dei db

3.1 Batch

Al fine di estrarre informazioni dai file in formato pdf, è stato programmato un batch notturno che si occupa di recuperare inizialmente i dati dall'archivio Sikelia per inserirli nello storage MinIO e, successivamente, effettuare l'ocr e recuperare i dati in maniera strutturata. Questo batch è controllato da uno scheduler, dalla libreria schedule, che ne scandisce l'esecuzione. Il batch viene dunque attivato ogni giorno ad un orario specificato all'interno del file config.py (vedi sezione dedicata in seguito), che scatena un flusso che porta all'analisi del singolo file e al salvataggio delle informazioni all'interno del database MongoDB.

3.1.1 OCR

L'ocr viene gestito interamente dal servizio Document Intelligence di Azure AI (Microsoft). L'utilizzo del servizio, per rispettare le direttive per la licenza gratuita, viene gestito attraverso una funzione che ne scandisce l'avvio, limitando le chiamate per minuto ed il numero di utilizzi totali durante il giorno. Questo parametro è inserito all'interno del file config.py (vedi sezione dedicata in seguito).

3.2 Interazione con l'utente

La parte centrale della POC consiste nell'interazione tra l'utente ed il sistema, tramite una chat interattiva capace di adattarsi alla stima delle caratteristiche dell'utente e di comunicare in maniera diretta con lo stesso.

3.2.1 Stima delle caratteristiche dell'utente

Idealmente, al momento del login tramite SPID, vengono acquisite dal sistema le informazioni relative a età, comune di residenza e codice fiscale. Queste informazioni sono necessarie per due motivi: il primo è quello di identificare i file medici tramite codice fiscale e il secondo è quello di stimare il livello di scolarizzazione dell'individuo, utilizzando le metriche di età e popolazione nel comune di residenza. Per simulare il recupero di queste informazioni, all'avvio del programma vengono chiesti all'utente questi tre dati. Dal comune di residenza poi, il sistema va a recuperare autonomamente il numero di abitanti, per ricercare poi gli utenti con caratteristiche più simili, tramite metriche euclidee basate sull'algoritmo KNN, e stimare per quali label è più probabile che l'utente possa sapere la risposta. Il nuovo vettore con i pesi delle label viene quindi associato all'utente e aggiornato durante l'interazione dello stesso, per poi essere inserito a db una volta conclusa la ricerca del file.

3.2.2 Valutazione della domanda da porre

Durante l'interazione con l'utente, vengono poste delle domande per identificare il record ricercato dall'utente stesso. La domanda da porre, ad ogni iterazione, viene calcolata considerando:

- Possibilità che l'utente sappia rispondere
- Quanto una risposta restringe il campo dei record

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		 Università di Catania



UNIONE EUROPEA



REPUBBLICA ITALIANA



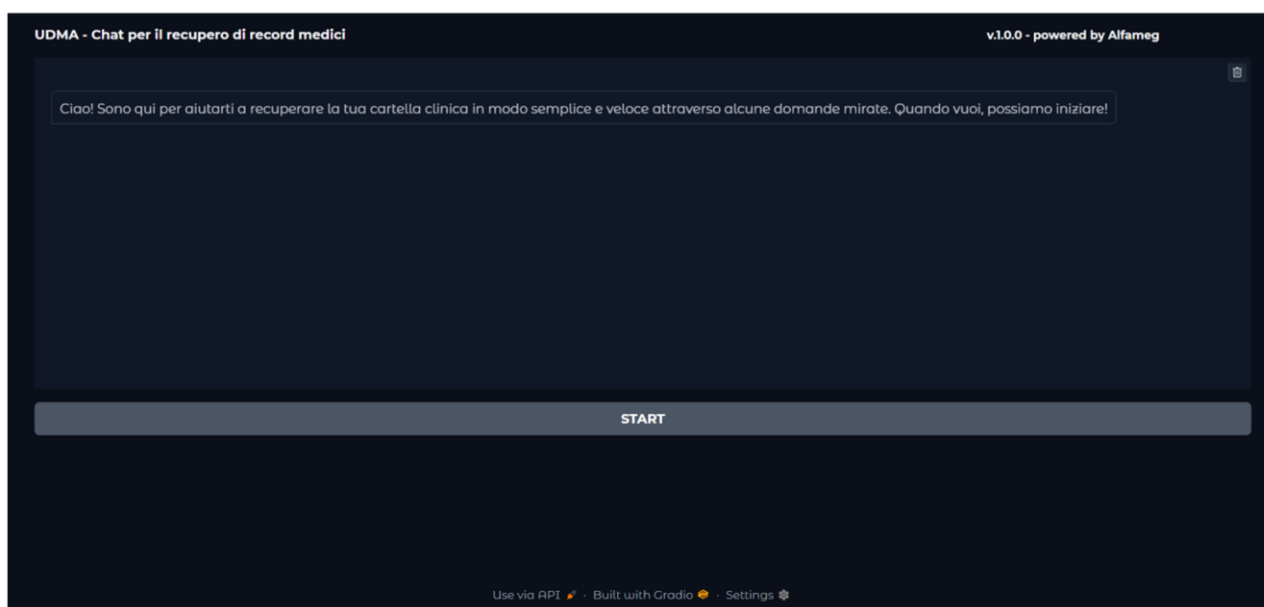
REGIONE SICILIANA



Il primo punto viene estratto facilmente dal vettore dei pesi dell'utente, estratto al momento del login (vedi sezione dedicata). Per il secondo invece viene calcolato attraverso il Gini Index, per identificare la label maggiormente discriminativa. Il valore del Gini Index della label, moltiplicato per il peso della stessa, produce un valore di score per la singola iterazione. Tra tutti i valori ottenuti, viene scelto il valore più alto, recuperata la domanda dal database e posta all'utente. In base alla risposta ottenuta, il vettore dei pesi viene modificato per tenere traccia delle capacità dell'utente a rispondere alle varie domande. In ogni caso, una volta fatta una domanda, questa non potrà essere più effettuata.

3.2.3 Chat

Lo sviluppo grafico della chat è stato fatto utilizzando Gradio e prevede una finestra comprensiva di tutti i messaggi inviati, seguita inferiormente da una sezione dove l'utente può rispondere alle domande poste.



All'avvio, la sezione dedicata alla risposta è nascosta per lasciare spazio a un bottone di START. Una volta iniziata l'interazione, questa viene divisa orizzontalmente in due parti, con un rapporto 1 a 5, dove a sinistra è presente un tasto per indicare che non si sa rispondere alla domanda ("No, non ricordo"), mentre a destra è presente una barra di input nella quale l'utente può scrivere liberamente la risposta.



3.3 Organizzazione dei db

I dati vengono salvati all'interno di due basi di dati, il primo MySQL, per le informazioni legate a utenti e pesi, e il secondo MongoDB per i record medici.

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



3.3.1 MySQL

Il database MySQL contiene informazioni riguardo gli utenti, i pesi, le label e le parole chiave per l'identificazione della diagnosi. È composto da cinque tabelle:

- **user_info**: contiene informazioni, per ogni utente, riguardo età e numero di abitanti nel comune di residenza.
- **user_weights**: contiene informazioni, per ogni utente, riguardo i pesi di ogni label. Ovvero se l'utente è stato in grado o meno di rispondere a determinate domande.
- **labels**: contiene informazioni riguardo le label, ovvero i campi dei record medici. Per ogni label viene indicato se viene utilizzata per identificare il record medico, se viene fornita tra i metadati di Sikelia, se viene valorizzata a seguito dell'OCR e se contiene informazioni sui dati anagrafici. Inoltre vengono indicate le domande da effettuare per estrarre informazioni e strutturare i dati a seguito dall'OCR e quelle da sottoporre all'utente durante l'interazione.
- **keywords**: contiene le parole chiave che verranno utilizzate durante il calcolo della gini_impurity per equiparare le diagnosi.
- **census**: contiene le informazioni riguardanti il numero di abitanti per ogni comune della regione Sicilia.
- **mr_archive**: contiene metadati e riferimenti ai file contenuti nello storage esterno.

Nel dettaglio, queste sono le proprietà delle tabelle considerate.

user_info		
uid	int (auto-increment)	Identificativo univoco dell'utente
age	int	Età
population	int	Numero di abitanti nel comune di residenza

user_weights		
uid	int	Identificativo dell'utente, legato alla tabella user_info
label	varchar(255)	Denominazione della label
value	float	Peso della label per l'utente

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		 Università di Catania



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



labels		
label_name	varchar(255)	Denominazione della label
is_active	int	Flag che indica se la label riguarda un campo utilizzabile durante l'interazione con l'utente
is_primary	int	Flag che indica se la label è contenuta nei metadati ottenuti da Sikelia
is_late	int	Flag che indica se la label riguarda campi che verranno valorizzati in seguito
is_anag	int	Flag che indica se la label riguarda dati anagrafici
question	varchar(255)	Domanda da porre al chat-bot per estrarre informazioni dal testo
user_question	varchar(255)	Domanda da porre all'utente durante l'interazione

keywords		
keyword_name	varchar(255)	Denominazione della keyword

census		
city	varchar(255)	Denominazione del comune
population	int	Numero di abitanti nel comune

mr_archive		
metadati*	varchar(255)	Serie di metadati
ocr	int	Flag che indica se il file è già stato analizzato
filename	varchar(255)	Nome del file
file_content	text	Link allo storage esterno

*I metadati sono una serie di colonne contenenti diverse informazioni interne al file. Non sono stati esplicitati tutti per motivi di compattezza.

3.3.2 MongoDB

Il database MongoDB contiene i record medici tra cui cercare per identificare il record richiesto dall'utente. I record vengono salvati con una struttura JSON, quindi chiave-valore, e sono identificati da un id univoco costruito dall'insieme del codice fiscale e dal numero di ricovero del record medico. Inoltre, ogni record a db contiene un indirizzo uri che rimanda al file pdf originale, ottenuto da Sikelia.

4 AZIONI PRELIMINARI

Per eseguire il programma sono necessarie alcune operazioni preliminari. Queste coinvolgono l'avvio e l'eventuale inizializzazione dei database, l'avvio del server di LM Studio e il caricamento delle cartelle di test su MinIO.

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		 Università di Catania



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



4.1 Avvio container docker

I dati vengono salvati all'interno di due database, uno MySQL e uno MongoDB. Entrambi sono salvati in locale all'interno di un container Docker, che è necessario attivare tramite il docker-compose all'interno del codice. Questo è un codice con estensione yaml che si occupa di avviare e inizializzare 4 servizi in locale, con le rispettive porte:

- mongoddb: 27017
- mongoexpress: 8085
- minio: 9000
- mysql: 3307

Durante il primo avvio, vengono inoltre eseguiti degli script di inizializzazione. Nel caso del database MySQL, in questa fase vengono create le tabelle e inizializzate la label (i campi dei record medici) e le parole chiave, necessarie per la distinzione della diagnosi.

Per avviare il docker compose, si può procedere tramite l'IDE o tramite riga di comando da terminale. Nel primo caso dipende dall'IDE utilizzato, mentre da terminale bisogna spostarsi nella cartella root medical-record-extractor ed eseguire

```
> docker compose up -d
```

Al primo avvio del docker compose, vengono eseguiti due file di init, uno per ogni database, che vanno a creare e popolare le tabelle in locale. Per quanto riguarda gli utenti, vengono creati una serie di utenti con dati e pesi randomici, per permettere di effettuare alcune interazioni di test.

4.2 LM Studio

Per eseguire correttamente il codice è necessario prima avviare LM Studio. Questo può essere fatto semplicemente lanciando l'eseguibile dell'applicazione da Windows.

4.2.1 Download del modello

Una volta avviata l'applicazione, è necessario, per il primo utilizzo, effettuare il download del language model. Questo può essere fatto direttamente all'interno dell'applicazione in pochi passaggi.

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



UNIONE EUROPEA



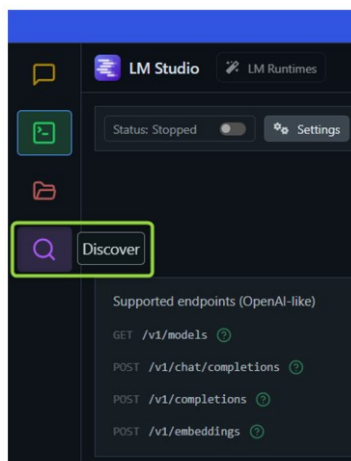
REPUBBLICA ITALIANA



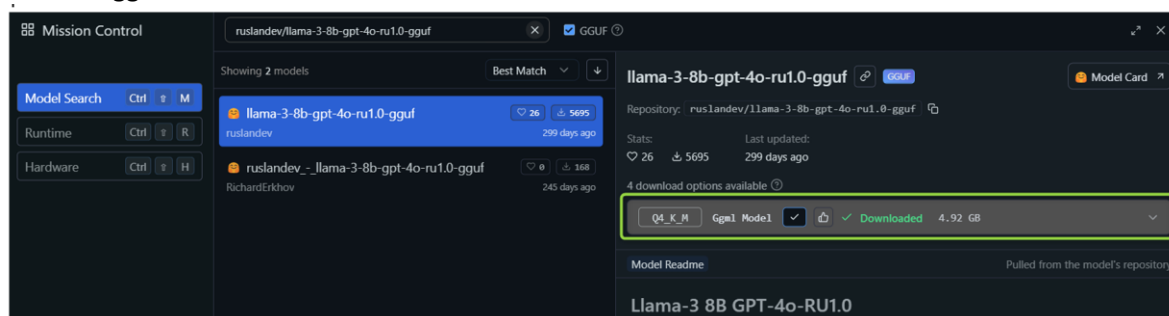
REGIONE SICILIANA



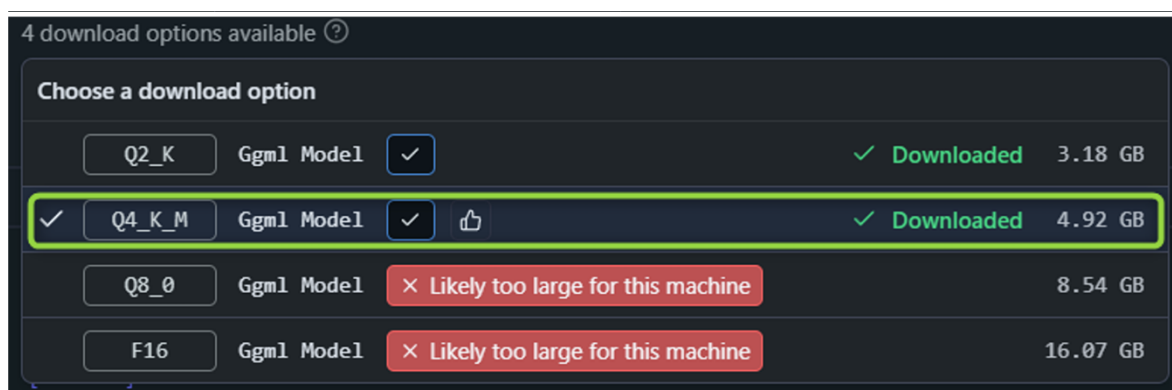
1. Nella barra verticale di sinistra, selezionare l'icona viola della lente di ingrandimento (Discover)



2. Nella barra di ricerca in alto, inserire il nome del modello, in questo caso: ruslandev/llama-3-8b-gpt-4o-ru1.0-gguf



3. Nella sezione a destra, selezionare l'opzione del modello in base alle caratteristiche della propria macchina. Modelli con dimensioni maggiori hanno risultati migliori, riducendo però le prestazioni.



PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



UNIONE EUROPEA



REPUBBLICA ITALIANA

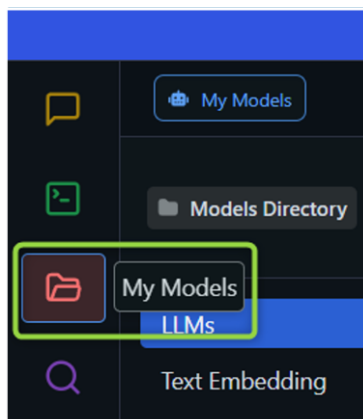


REGIONE SICILIANA





Non scaricare i modelli con la dicitura “Likely too large for this machine” perché si rischia di bloccare la macchina.

4. Attendere il completamento del download e verificare il corretto download nella cartella My Models, riconoscibile dalla folder rossa



5. Opzionale: è possibile modificare le impostazioni standard della macchina riguardo il bilanciamento della potenza o, più specificamente, le impostazioni relative al modello, cliccando sull’ingranaggio a destra del modello scelto. Una volta effettuata una modifica, comparirà un puntino rosso, a segnalare che le impostazioni di base sono state modificate

Arch	Params	Publisher	Model	Quant	Size	Date Modified	Actions
llama	8B	ruslandev	llama-3-8b-gpt-4o-ru1.0@q2_k	Q2_K	3.18 GB	9 days ago	 ...
llama	8B	ruslandev	llama-3-8b-gpt-4o-ru1.0@q4_k_m	Q4_K_M	4.92 GB	9 days ago	 ...

L’applicazione comunque notificherà se le impostazioni scelte potrebbero compromettere eccessivamente le prestazioni della macchina.

4.2.2 Caricamento del modello

Una volta scaricato il modello, questo può essere selezionato e caricato come modello in uso.

1. Nella barra verticale di sinistra, selezionare l’icona verde della shell (Developer).

PARTENARIATO Progetto “UDMA” n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



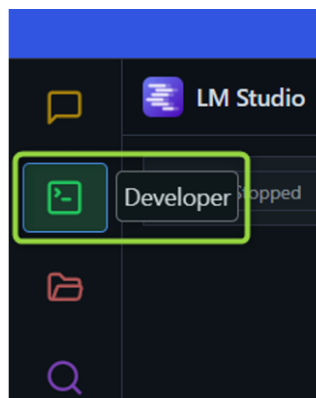
UNIONE EUROPEA



REPUBBLICA ITALIANA



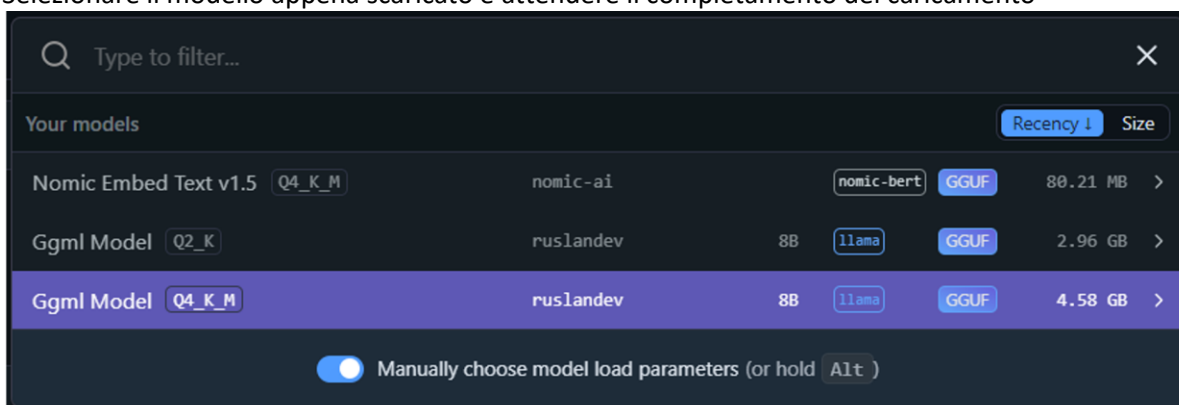
REGIONE SICILIANA



2. Cliccare quindi su “Select a model to load”, nella parte alta e centrale dello schermo, per visualizzare menu a tendina con i modelli scaricati disponibili



3. Selezionare il modello appena scaricato e attendere il completamento del caricamento



4.2.3 Avvio server LM Studio

Per utilizzare le API di LM Studio, è necessario avviare il server locale. Anche questo passaggio può essere fatto semplicemente dall'applicazione ed è necessario per il corretto funzionamento del chat bot.

1. Nella barra verticale di sinistra, selezionare l'icona verde della shell (Developer).

PARTENARIATO Progetto “UDMA” n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



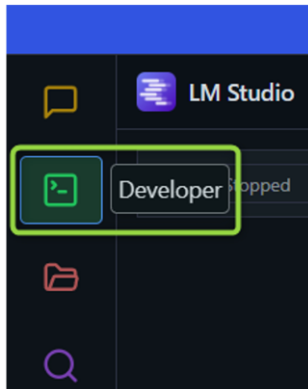
UNIONE EUROPEA



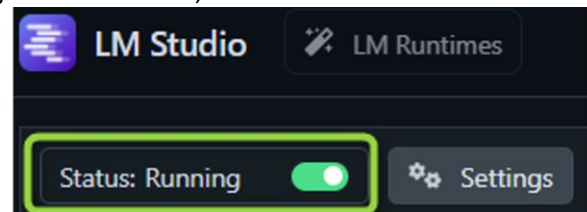
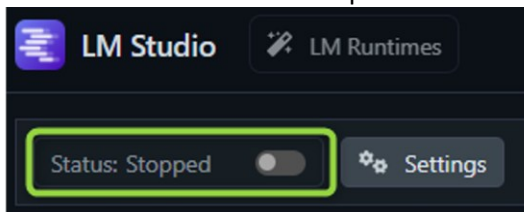
REPUBBLICA ITALIANA



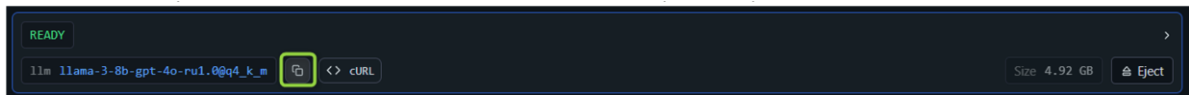
REGIONE SICILIANA



2. Nella parte alta a sinistra della finestra, si trova uno switch che controlla lo stato di attivazione del server. Fare click sullo switch per accenderlo/spegnere. Se acceso, lo switch sarà verde.



3. Una volta caricato correttamente il modello, verrà visualizzata una sezione con la scritta 'READY' e il nome specifico del modello, con un tasto per copiare direttamente il nome



4.3 Caricamento file su Minio

Per eseguire il codice in locale, è necessario simulare il recupero dei file da una fonte esterna. All'interno della POC abbiamo adottato Minio come storage e predisposto il recupero dei file da codice. È quindi necessario, prima di eseguire il progetto, caricare su MinIO alcuni file di test. Questo può essere fatto semplicemente da interfaccia grafica del browser, collegandosi a questo indirizzo (dopo aver avviato il docker-compose) <http://localhost:9001/login> e, dopo aver effettuato il login con le seguenti credenziali:

- Username: root
- Password: password

Cliccare su "Browse"

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		



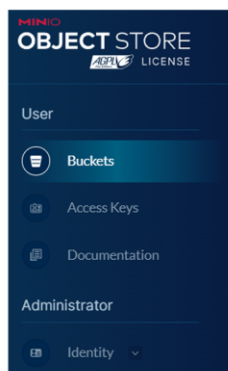
UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



Buckets

Search Buckets

Create Bucket +

mdfiles

Created: Mon May 05 2025 11:33:13 GMT+0200 (Ora legale dell'Europa centrale) Access: R/W

Manage

Browse →

Usage

28.8 KiB

Objects

20

E quindi su “Upload”

← Buckets

Start typing to filter objects in the bucket

mdfiles

Created: Mon May 05 2025 11:33:13 GMT+0200 (Ora legale dell'Europa centrale) Access: PRIVATE 28.8 KiB - 20 Objects

Rewind ↶

Refresh ↻

Upload ↗

Si aprirà una finestra di navigazione file di Windows, dalla quale sarà possibile selezionare i file da caricare.

4.4 Parametri di configurazione

Il codice è predisposto per eventuali modifiche che possono riguardare diversi campi. In tal senso, all'interno del file config.py (src/server/config/config.py) vi sono una serie di parametri di configurazione interni che permettono di personalizzare il flusso in base alle specifiche esigenze. I campi editabili sono:

- Configurazioni dei db (username, password, host, porte e nomi)
- Configurazioni sui log e sul debug
- Chiave di accesso a Document Intelligence di Azure AI
- Configurazioni di MinIO
- model_name: Nome del modello da utilizzare (vedi capitoli precedenti)
- knn_threshold: Numero di utenti da considerare durante la valutazione dei pesi
- n_threshold: Numero di record per il quale la ricerca può terminare
- batch_time: Orario in cui si avvia il batch
- ocr_calls_per_minute: Numero massimo di chiamate per minuto che il batch può effettuare

4.4.1 Modifica del modello di linguaggio

È possibile modificare il language model liberamente, in base alle esigenze e alle disponibilità della macchina. Per farlo, basta scaricare il modello desiderato seguendo le istruzioni nella sezione dedicata e quindi fare un

PARTENARIATO Progetto “UDMA” n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca



UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



restart del server. Inoltre, è necessario modificare il nome del modello nel file config.py. Il nome corretto può essere recuperato seguendo le indicazioni del punto 3 di questa sezione.

4.5 Installazione dei requirements

Aperto il codice tramite l'IDE, provvedere all'installazione dei requirements. Questi sono contenuti all'interno del file requirements.txt e, solitamente, viene individuato automaticamente dall'IDE che propone l'installazione delle librerie necessarie.

5 ESECUZIONE DEL CODICE IN LOCALE

Per eseguire il codice, una volta avviato il server LM Studio (vedi sezione dedicata) e avviato il container docker contenente i database (vedi sezione dedicata), bisogna avviare il file main.py. Questo si occuperà di avviare il server locale e gestire i vari flussi su thread separati. Una volta salito, sarà possibile avviare un'interazione con il sistema interrogando un endpoint esposto dal server stesso. Per simulare la chiamata si possono usare diverse metodologie, per semplicità nei test è stato usato il software Postman. L'endpoint esposto si trova a questo URL e può essere interrogato tramite una chiamata POST, inserendo nel body un oggetto JSON contenente età, comune di residenza e codice fiscale dell'utente che per cui si vuole effettuare l'interazione. Questa operazione è necessaria per simulare i dati in arrivo a seguito del login tramite SPID. Una volta inseriti i dati ed effettuata la chiamata, in caso di successo, la risposta conterrà un campo "link", contenente un URL locale (http://127.0.0.1:7860/) dove poter interagire con il sistema. Cliccare dunque su "START" e iniziare l'interazione.

Un esempio di corpo per la chiamata POST è:

```
{
  "age": 40,
  "city": "Catania",
  "codice_fiscale": "RSSMRC70C15H501U"
}
```

E la schermata completa risulta simile a:

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca



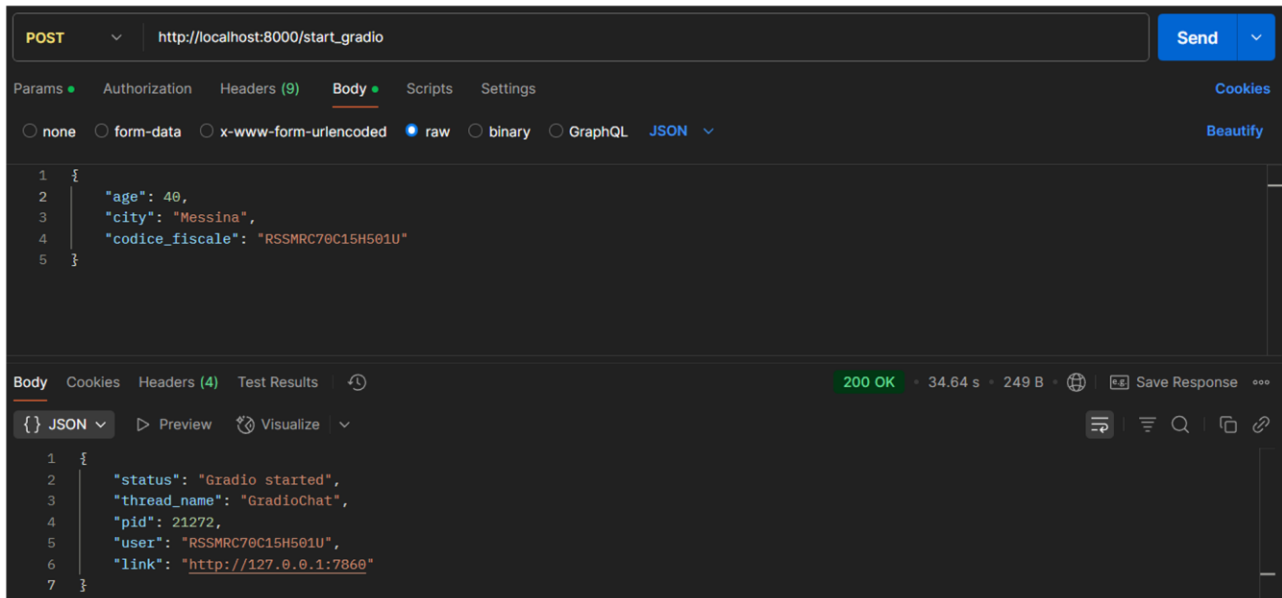
UNIONE EUROPEA



REPUBBLICA ITALIANA



REGIONE SICILIANA



5.1 Note di esecuzione

Vengono elencati alcuni punti di attenzione durante l'esecuzione in locale.

- Dopo un tempo di 120 secondi (modificabile nel file config.py), il sistema considera l'utente inattivo e chiude l'interazione, al fine di liberare risorse.
- Il nome del comune deve essere corretto, indipendentemente dall'iniziale maiuscola o minuscola. Questo perché si intende simulare il recupero delle informazioni dal login SPID. In caso di errore, il sistema risponderà con un errore "Comune non valido (es. 'Catania')".
- I test sono stati effettuati su 20 cartelle cliniche generate dall'intelligenza artificiale, per cui potrebbero esserci incongruenze a livello medico.
- I record medici di test, sono associati a 4 utenze di test con la seguente ripartizione.
 - o RSSMRC70C15H501U: 15 record
 - o BNCLRA82M62H501X: 3 record
 - o NRILCU90H03H501V: 1 record
 - o VRDRRT75T50H501F: 1 record

PARTENARIATO Progetto "UDMA" n. 086311190415 - CUP G69J18001040007		
Aziende (PMI)		Organismo di Ricerca
		