

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea triennale in Informatica

Relatore: **Federico Cabitza**

Correlatore: **Andrea Campagner**

A.A. 2022-2023



Sviluppo e implementazione di un Sistema Decisionale in ambito medico

Relazione della prova finale di

Matilde Ghidini

Matricola n. 852256

Contenuti

01 Introduzione al progetto di stage

02 Clinical Decision Support System

03 Il progetto Epimetheus

04 Sviluppi futuri

Introduzione

Il progetto

Sviluppo e deployment di una web app, che implementa un sistema di supporto alle decisioni in ambito clinico.

Obiettivi

- Sviluppo back-end e di flusso logico
- Implementazione tramite l'utilizzo del linguaggio Python e del framework Flask
- Integrazione di componenti di Machine Learning preesistenti nel sistema
- Deployment dell'applicazione in ambiente di produzione

MUDI è un laboratorio di ricerca del dipartimento di Informatica, Sistemistica e Comunicazione (DISCo) dell'Università degli Studi Milano Bicocca. Si focalizza su interazione uomo-macchina, sistemi di supporto decisionali, Machine Learning e incertezza.



Clinical Decision Support System

Cosa sono?

Software progettati per essere un aiuto diretto per la presa di decisioni di tipo clinico.

Le caratteristiche del singolo paziente sono confrontate con una base di conoscenza clinica computerizzata e la situazione dei pazienti o le raccomandazioni sono presentate al medico per una decisione.

Obbiettivo

Miglioramento dei sistemi sanitari e delle decisioni mediche tramite conoscenze cliniche specifiche, informazioni sul paziente e altre informazioni.

Come si presentano?

Web app, integrazioni con cartelle cliniche elettroniche, CPOE, dispositivi biometrici e tecnologia wearable.

Il progetto Epimetheus

Clinical Decision Support System

Supporto digitale alle prognosi di interventi chirurgici complessi in ambito ortopedico.

Modelli di Machine Learning

Calcolano predizioni sulla base di dati inseriti dall'utente in un form.

EPIMETHEUS

```
graph TD; E[EPIMETHEUS] --> C[Clinical Decision Support System]; E --> M[Modelli di Machine Learning]; E --> W[Web App]; E --> V[Visualizzazione dell'incertezza];
```

Visualizzazione dell'incertezza

I risultati delle predizioni vengono mostrate attraverso l'uso di diagrammi chiari e intuitivi.

Web App

Offre agli utenti un accesso semplice e intuitivo tramite browser.

1

2

3

4

REQUIREMENT SELECTION

Select the procedure for which you want to make the comparison

Procedure

Hip

selezione della procedura

PREFERENCE SELECTION

☐ False positive minimization☒ Balanced prediction☐ False negative minimization

form

INPUT DATA

Select the score and data source

Score

Physical

Data source

☐ Insert batch data☒ Insert data manually

Age patient

Gender

☐ Male☐ Female

ASA Class

select ASA class

Height Pre operation (cm)

Weight Pre operation (kg)

BMI Total

VAS Total Pre operation

SF12 PhysicalScore Pre operation

SF12 MentalScore Pre operation

SF12 data of initial response

SF12 health self-assessment

SF12 response scale

SF12 yield of last month

SF12 limit of last month

SF12 Emo of last month

SF12 hurdle of last month

SF12 clear of last month

SF12 energy of last month

SF12 sadness of last month

SF12 socialization of last month

Reset

Go to results

1

2

3

4

< Go back to compilation

Results

Editable parameters:

ETA

45

SF12 PHYSICALSCORE_PREOP

70.00

Patient status 6 months post operation

diagramma
cricolare

Evaluate prediction

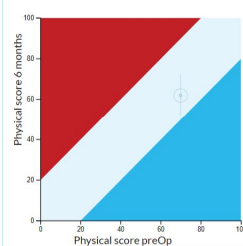
Patient status 6 months post operation



stick bar

Evaluate prediction

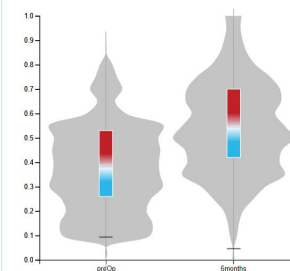
Patient status 6 months post operation



scatter plot

Evaluate prediction

Comparison of pre-operation and 6-month post-operation situation



Legend:

- Improved
- Uncertain
- Worsened
- Density of value
- Prediction
- Median

Evaluate prediction

violin plot

Punti critici

- Architettura monolitica
- Logica applicativa gestita tramite Javascript (lato front-end)
- Elementi del form codificati staticamente in Javascript
- Nessuna validazione dei dati inseriti nel form



SVILUPPO DI UNA NUOVA ARCHITETTURA SU DUE LIVELLI

Separazione tra
front-end e back-end

- Possibilità di sviluppare e testare UI e back-end come entità separate
- Maggiore scalabilità
- Maggiore modularità
- Maggiore flessibilità nello sviluppo in vista di modifiche future

Separazione tra back-end e front-end

- FE: applicazione React
- BE: server Flask che implementa una API RESTful

API endpoints

Per lo scambio di dati con il front-end React.

```
/api/get_form  
/api/predict_spine  
/api/predict_hip_knee
```

Classe forms.py

Per mantenere le strutture dei form delle diverse procedure.

Validazioni sui campi del form

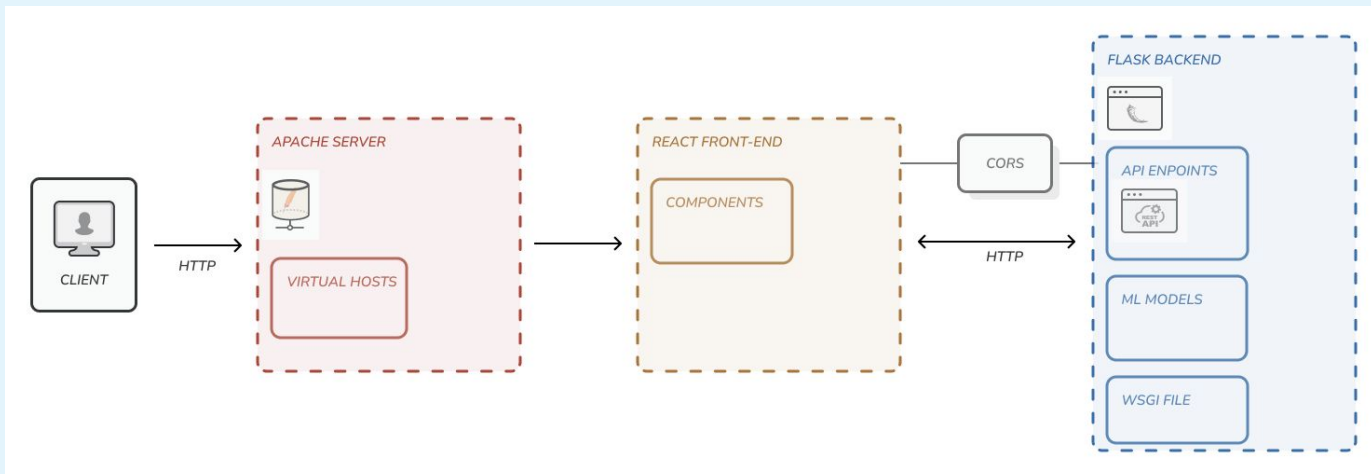
Funzioni per il controllo dei dati inseriti nei campi del form e validazione rispetto al range di valori che essi possono assumere.



Deployment dell'applicazione

Server Apache configurato su una macchina ad indirizzo IP pubblico

- Nessun limite in termini di spazio di archiviazione
- Controllo completo sull'ambiente di hosting e sulla configurazione del server
- Possibilità di ottimizzare il server per le esigenze specifiche dell'applicazione
- Capacità di gestire un grande volume di traffico e di garantire prestazioni elevate
- Integrazione di Flask con Apache tramite mod_wsgi



Sviluppi futuri

- Integrazione di nuovi modelli di Machine Learning per procedure diverse
- Creazione di un database per mantenere le strutture dei form delle procedure
- Creazione di una base di dati per mantenere dati riguardanti gli utenti e gli utilizzi del sistema, al fine di poter creare statistiche.
- Completare il deployment, integrando React con il server Apache
- Testare l'applicazione e raccogliere feedback

Strumenti utilizzati

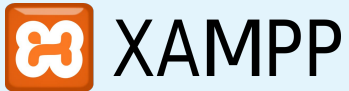
Linguaggi di programmazione e Framework



Strumenti di sviluppo



Tecnologie per il deployment



Strumenti per la documentazione



Grazie