



# UNIVERSITÀ DEGLI STUDI DI SALERNO



## Progetto di Intelligenza Artificiale

Studente	Matricola
Scaparra Daniele Pio	0512116260
Fasolino Pietro	0512116473
Vitulano Antonio	0512116776

# Documento di Deployment

## Indice

<b>1</b>	<b>Obiettivo del Tool</b>	<b>3</b>
<b>2</b>	<b>Ambiente di Deployment</b>	<b>3</b>
2.1	Infrastruttura . . . . .	3
2.2	Requisiti Software . . . . .	3
<b>3</b>	<b>Installazione del Tool</b>	<b>4</b>
3.1	Preparazione dell'Ambiente . . . . .	4
3.2	Avvio del Tool . . . . .	4
<b>4</b>	<b>Funzionamento del Tool</b>	<b>4</b>
4.1	Caricamento dei Dati . . . . .	4
4.2	Analisi della Conversazione . . . . .	4
4.3	Generazione del Report . . . . .	4
<b>5</b>	<b>Testing e Validazione</b>	<b>5</b>
5.1	Test Tecnico . . . . .	5
5.2	Feedback Utente . . . . .	5
<b>6</b>	<b>Monitoraggio e Manutenzione</b>	<b>5</b>
6.1	Monitoraggio . . . . .	5
6.2	Manutenzione . . . . .	5
<b>7</b>	<b>Scalabilità</b>	<b>5</b>
<b>8</b>	<b>Limitazioni e Piano di Rollback</b>	<b>6</b>
8.1	Limitazioni . . . . .	6
8.2	Piano di Rollback . . . . .	6

# 1 Obiettivo del Tool

Il tool di intelligenza artificiale si propone di analizzare una conversazione tra due interlocutori e generare un report in formato PDF che evidenzi l'emozione predominante di ciascun intervento. L'algoritmo si basa sul dataset **GoEmotions** (opportunamente pulito e bilanciato) ed è stato sviluppato in **Python** utilizzando la libreria **Streamlit**<sup>1</sup>. Il tool opera localmente e consente il caricamento di file di testo contenenti le conversazioni.

## 2 Ambiente di Deployment

### 2.1 Infrastruttura

- **Ambiente di esecuzione:** Locale (PC o workstation).
- **Requisiti hardware:**
  - CPU: Dual-core o superiore
  - RAM: Minimo 4 GB
  - Storage: Minimo 500 MB di spazio disponibile

### 2.2 Requisiti Software

- Sistema operativo: Windows, macOS o Linux
- Python  $\geq 3.8$
- Librerie necessarie:
  - Streamlit
  - Transformers<sup>2</sup>
  - PyPDF2<sup>3</sup>
  - Matplotlib/Seaborn<sup>4</sup>
  - Pandas
  - NumPy
- Strumenti opzionali:
  - Ambiente virtuale (es. **venv**<sup>5</sup> o **conda**<sup>6</sup>) per gestire le dipendenze

---

<sup>1</sup>Streamlit è una libreria Python che permette di creare applicazioni web interattive per il machine learning e la data science.

<sup>2</sup>Transformers è una libreria open-source di Hugging Face che fornisce modelli pre-addestrati per il Natural Language Processing (NLP).

<sup>3</sup>PyPDF2 è una libreria Python per manipolare file PDF, che consente di leggere, scrivere e modificare documenti PDF.

<sup>4</sup>Matplotlib è una libreria di visualizzazione dati per Python, mentre Seaborn è una libreria basata su Matplotlib che fornisce visualizzazioni statistiche più avanzate.

<sup>5</sup>venv è uno strumento integrato in Python per creare ambienti virtuali, isolando le dipendenze di progetto.

<sup>6</sup>Conda è un gestore di pacchetti e ambiente che permette di installare e gestire pacchetti Python e non, e di creare ambienti isolati.

## 3 Installazione del Tool

### 3.1 Preparazione dell'Ambiente

1. **Installare Python:** Verificare che Python sia installato e configurato correttamente (versione  $\geq 3.8$ ). **Creare un ambiente virtuale:**

### 2. 3.2 Avvio del Tool

- (a) **Eseguiare Streamlit:**

```
streamlit run demo.py
```

- (b) **Interfaccia locale:** Una volta avviato, il tool sarà accessibile tramite un browser all'indirizzo `http://localhost:8501`.

## 4 Funzionamento del Tool

### 4.1 Caricamento dei Dati

Il tool accetta file di testo (.txt) contenenti conversazioni formattate come segue:

```
Interlocutore1: Buongiorno, come stai?  
Interlocutore2: Bene, grazie. E tu?  
Interlocutore1: Non c'è male, grazie per aver chiesto.
```

### 4.2 Analisi della Conversazione

- Il file viene processato, e per ogni intervento:
  - Viene identificata l'emozione predominante (ad es. gioia, tristezza, rabbia).
  - Viene calcolata una probabilità associata all'emozione.

### 4.3 Generazione del Report

Il report PDF include:

- Una panoramica generale delle emozioni presenti nella conversazione.
- Un'analisi dettagliata per ciascun intervento.
- Grafici e visualizzazioni (es. grafico a barre delle emozioni).

## 5 Testing e Validazione

### 5.1 Test Tecnico

- **Performance:** Verifica dei tempi di risposta del modello per conversazioni di lunghezza variabile.
- **Accuratezza:** Confronto tra le emozioni previste dal modello e una classificazione manuale.

### 5.2 Feedback Utente

- Raccolta di opinioni dagli utenti finali riguardo l'usabilità dell'interfaccia e l'accuratezza del report.

## 6 Monitoraggio e Manutenzione

### 6.1 Monitoraggio

- Controllo dei log generati dal tool per rilevare eventuali errori.
- Monitoraggio dell'accuratezza tramite verifiche periodiche con nuovi dati.

### 6.2 Manutenzione

- Aggiornamento delle dipendenze Python ogni 6 mesi.
- Retraining del modello con dati aggiornati (se disponibili).

## 7 Scalabilità

Sebbene il tool funzioni localmente, è possibile pianificarne una migrazione su cloud per supportare più utenti. Le opzioni includono:

- Deploy su **AWS** o **Google Cloud**.
- Containerizzazione con **Docker**<sup>7</sup> e orchestrazione tramite **Kubernetes**<sup>8</sup>.

---

<sup>7</sup>Docker è una piattaforma che consente di automatizzare la distribuzione di applicazioni all'interno di contenitori leggeri e portabili.

<sup>8</sup>Kubernetes è un sistema open-source per l'automazione del deployment, scaling e gestione di container applicativi.

## 8 Limitazioni e Piano di Rollback

### 8.1 Limitazioni

- Il tool non supporta file di input non strutturati o formati diversi da .txt.
- Funziona solo in ambienti locali, il che limita l'accessibilità.

### 8.2 Piano di Rollback

In caso di malfunzionamenti, ripristinare l'ambiente locale eliminando e ricreando l'ambiente virtuale:

```
rm -rf venv
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```