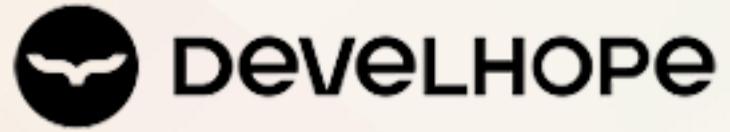


# Il valore delle opinioni

**Sentiment Analysis**  
per e-commerce

**Team:**

Francesca Amarena  
Matteo Gagliardi



# Progetto: Il Valore delle opinioni



**Scegliere un prodotto  
può essere  
complesso.**



**Interpretare le  
opinioni lo rende più  
semplice.**

**Team:**

Francesca Amarena  
Matteo Gagliardi

**Tutor:**

Pier Giorgio Poloni



# Obiettivi del progetto

## Obiettivo principale:

Costruire un modello di Sentiment Analysis in grado di analizzare una frase in Linguaggio Naturale e classificare automaticamente le recensioni in tre classi:

- Negativo
- Neutro
- Positivo

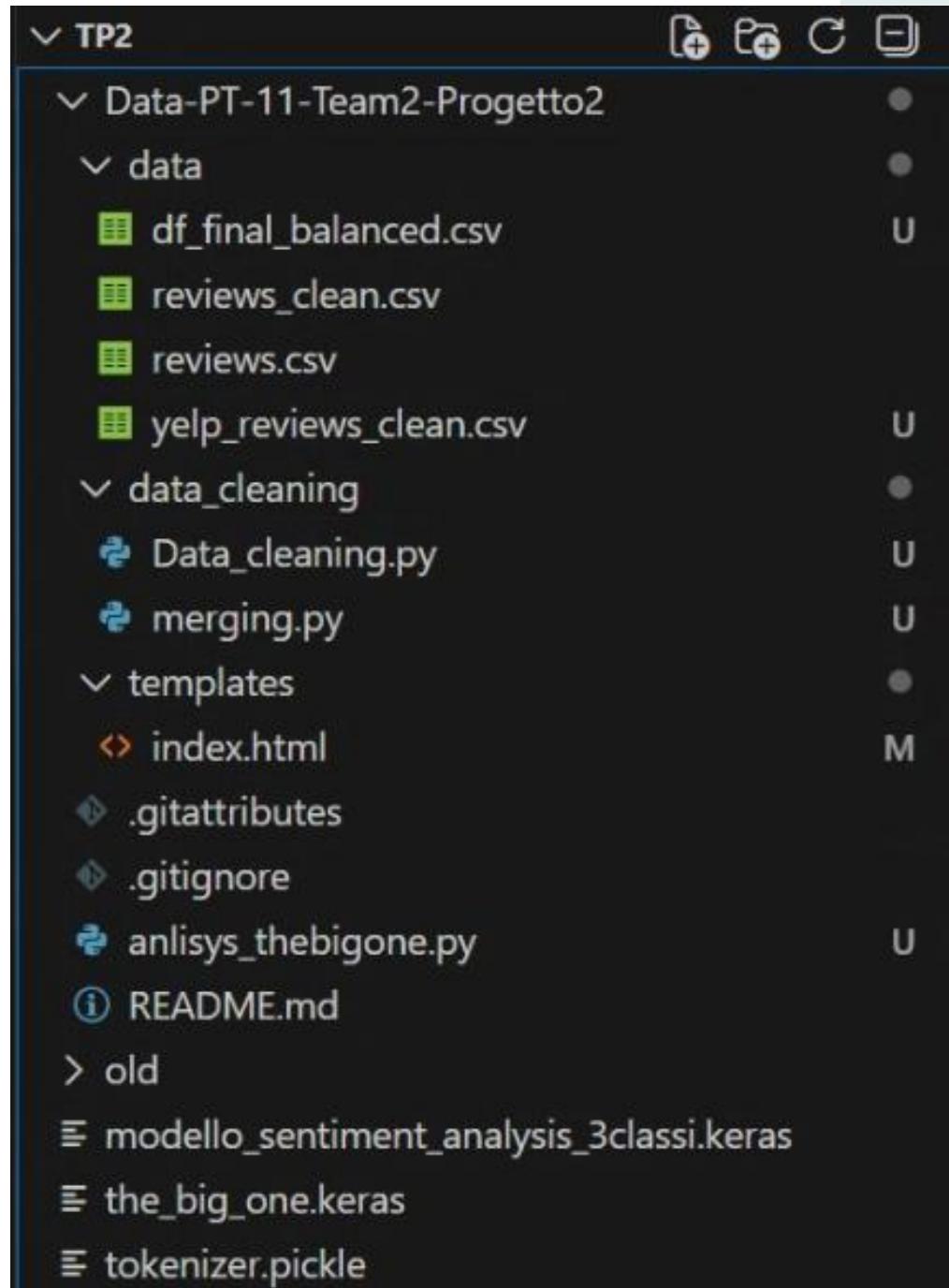
## Obiettivo secondario:

Testare il modello su dati mai visti

Rendere il modello utilizzabile tramite una semplice interfaccia web

# Dataset utilizzati

<b>Dataset</b>	<b>Fonte</b>	<b>Ruolo</b>	<b>Motivazione della scelta</b>
Amazon Reviews	Kaggle	Training	<ul style="list-style-type: none"><li>• Recensioni reali</li><li>• Rating 1-5</li><li>• Colonna testo</li></ul>
Yelp Reviews	Yelp Open Data	Generalizzazione	<ul style="list-style-type: none"><li>• Rating 1-5</li><li>• Colonna testo</li></ul>



# Struttura del Progetto

Abbiamo suddiviso il progetto come segue:

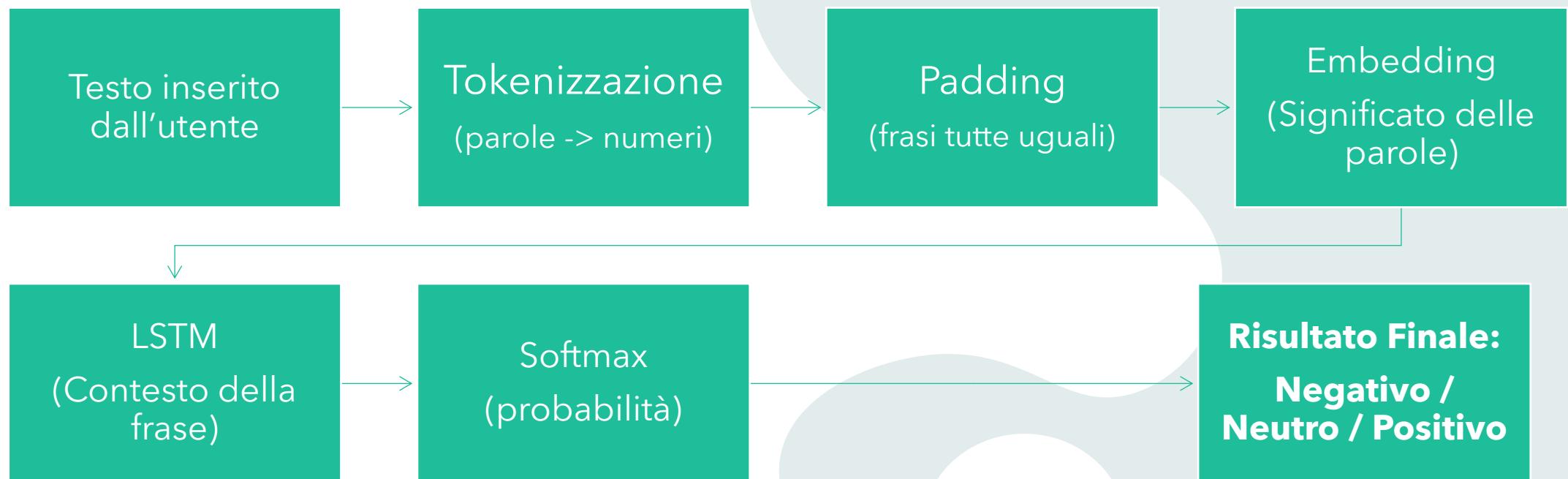
- La cartella **Data** contiene i file csv
- La cartella **Cleaning** contiene i file di pulizia dei dataset ed il file di merging
- La cartella **Templates** contiene l'HTML chiamato dall'app.py
- La macro-cartella **DATA-PT** contiene all'interno, oltre le suddette, anche i file di funzionamento del modello

# Data Cleaning & Preprocessing

Pulizia del testo	Sentiment Analysis	Unione dataset	Interfaccia web
Rimozione valori nulli e duplicati	<b>Tokenizzazione</b> -> spezza il testo e lo trasforma in numeri	Concatenazione tra i due dataset	HTML per il frontend
Normalizzazione del testo	<b>Embedding</b> -> da un valore numerico alle parole	Riduzione testo per addestrare modello	Flask per il Backend
Rimozione stopwords	<b>LSTM</b> -> capisce il senso della frase leggendo le parole in sequenza		Chiamata al modello

# Il modello

Il testo viene trasformato in numeri, analizzato nel suo contesto e classificato automaticamente in base al sentimento.



# MODELLO

---

## **Addettoamento del modello:**

---

- Dataset di training pulito
  - Classificazione multiclasse (negativo / neutro / positivo)
- 

## **Testing**

---

- Valutazione delle performance sui dati di training
- Test del modello su una lista all'interno del codice
- Verificare generalizzazione e affidabilità

# Difficoltà durante l'addestramento

## 1. Addestramento con grandi quantità di dati

Necessità di :

- Pulizia dei dati
- Bilanciamento delle classi
- Gestione del rumore

(Non è sufficiente avere molti dati: devono essere **puliti, coerenti e rappresentativi.**)

## 2. Difficoltà nel riuscire a «scovare» il valore Neutrale, perché le parole usate nel neutrale sono facilmente confondibili

Esempio:

- The product is okay -> neutro
- The product is not bad -> positivo, ma linguisticamente ambiguo

# Sentiment Classification

• Online

Enter a sentence to analyze its sentiment.

Sentence

- ◆ Example: This course was very useful

Analyze

Tip: longer sentences usually give more accurate results.

# Interfaccia Web

# Interfaccia WEB

## Obiettivo:

Realizzare una **web application** che consenta all'utente di inserire una frase in linguaggio naturale e ottenere una **classificazione del sentimento** in tre classi

## Architettura dell'applicazione:

**Frontend:** pagina web con form di input testuale

**Backend:** Flask (Python)

**Modello AI:** TensorFlow / Keras

## Tecniche NLP:

Tokenizzazione del testo

Padding delle sequenze

Classificazione multi-classe

```
{% if classificazione %}  
<div class="result">  
  <p class="k">Input</p>  
  <p class="v">"{{ frase }}"  
  <p class="k" style="margin-top:14px;">Result</p>  
  <div class="badge"><i>{{ classificazione }}</i></div>  
</div>  
{% endif %}
```

```
if request.method == 'POST':  
    frase = request.form['frase']  
  
    sequenze = tokenizer.texts_to_sequences([frase])           # Trasformiamo la frase in sequenze  
    sequenze_padded = pad_sequences(  
        sequenze,  
        maxlen=lungh_sequenze,  
        padding='post',  
        truncating='post'  
    )  
  
    pred = model.predict(sequenze_padded)    # shape (1,3)  
    proba = pred[0]                         # es: [0.2, 0.3, 0.5]  
    classe = int(np.argmax(proba))          # 0/1/2 (nello specifico prende l'indice della probabilità più alta)  
    # Mappa numero → etichetta testuale  
    mappa = {0: "Negativa", 1: "Neutrale", 2: "Positiva"}  
    # Recuperiamo la classificazione finale  
    classificazione = mappa[classe]  
    questo rimanda i dati alla pagina html  
    return render_template('index.html', frase=frase, classificazione=classificazione)
```

# Dimostrazione Recensione Positiva

Frase analizzata: "This product is amazing and exceeded my expectations!"

Risultato: "Positiva"

The screenshot shows a web-based sentiment classification tool. At the top right, there is a teal button labeled "Online". The main title is "Sentiment Classification". Below it, a placeholder text says "Enter a sentence to analyze its sentiment.". A "Sentence" input field contains the text "This product is amazing and exceeded my expectations!". To the left of the input field is an example sentence: "Example: This course was very useful". Below the input field is a large teal "Analyze" button. To the right of the input field, a tip in a light gray box reads "Tip: longer sentences usually give more accurate results.". At the bottom, under the heading "Input", the analyzed sentence is shown again. Under the heading "Result", there is a teal button with a white dot and the word "Positiva".

# Dimostrazione Recensione Negativa

Frase analizzata: "It looks cheap and does not do what it promise"

Risultato: "Negativa"

**Sentiment Classification**

Enter a sentence to analyze its sentiment.

Sentence

Example: This course was very useful

Analyze

Tip: longer sentences usually give more accurate results.

Input  
"It looks cheap and does not do what it promises."

Result

Negativa



# Dimostrazione Recensione Neutrale

Frase analizzata: "It works fine,  
but I wouldn't buy it again"

Risultato: Neutrale

The screenshot shows a web-based sentiment analysis tool. At the top, there's a header with the title "Sentiment Classification" and an "Online" status indicator. Below the header is a text input field labeled "Sentence" with the placeholder "Enter a sentence to analyze its sentiment." An example sentence, "This course was very useful", is shown with a blue info icon. A large green button labeled "Analyze" is centered below the input field. To the right of the button is a tip: "Tip: longer sentences usually give more accurate results." The main result area has a light gray background and displays the input sentence "It works fine, but I wouldn't buy it again." followed by the word "Result" and a green button with a white dot and the word "Neutrale".

# Conclusioni

Il progetto dimostra come la Sentiment Analysis possa trasformare feedback testuale non strutturato in **insight utili e azionabili**.

Il modello sviluppato è in grado di analizzare nuove recensioni in modo affidabile e rappresenta una soluzione valida e veloce per interpretare opinioni e/o recensioni nel campo dell' e-commerce.

# SVILUPPI FUTURI

Miglioramento grafico dell'interfaccia web con inserimento dell'accuratezza del modello  
(matplotlib)

Riconoscimento di più lingue

Aumentare il riconoscimento delle sfumature delle recensioni

# **GRAZIE PER L'ATTENZIONE**

**Corso: Data & AI Analyst**



**DevelHOPE**