Sistema per la Classificazione Automatica di Contenuti Offensivi su Piattaforme Social

Università degli Studi di Salerno
Dipartimento di Informatica
a.a. 2024–2025

Prof. Fabio Palomba
Michele Spinelli 0512110343

Indice

1 Introduzione

Negli ultimi anni, piattaforme di social media come TikTok e Threads hanno registrato una crescita significativa in termini di utenti attivi e volume di contenuti multimediali condivisi quotidianamente. Questi contenuti, frequentemente costituiti da brevi video accompagnati da tracce audio, rappresentano una fonte ricca di informazioni, ma possono anche fungere da veicolo per la diffusione di messaggi offensivi, discriminatori o socialmente dannosi.

In questo contesto, il presente progetto ha come obiettivo lo sviluppo di un sistema automatizzato in grado di acquisire, elaborare e classificare contenuti audio e testuali estratti da tali piattaforme, a partire da un semplice link fornito dall'utente. Il flusso audio viene trasformato in testo tramite tecniche di riconoscimento vocale (speech-to-text), per poi essere analizzato attraverso algoritmi di apprendimento automatico e metodi avanzati di elaborazione del linguaggio naturale (NLP). Il sistema integra modelli supervisionati classici, come la regressione logistica e le Support Vector Machines (SVM), insieme a tecniche di deep learning, tra cui il fine-tuning di modelli pre-addestrati come BERT. Inoltre, si esplora l'impiego di Large Language Models (LLM), come ChatGPT e Mistral, per un'analisi semantica più profonda del contenuto.

La capacità del sistema di rilevare e classificare contenuti potenzialmente offensivi rappresenta un contributo rilevante nel contrasto alla diffusione di discorsi d'odio, discriminazione e linguaggio ostile online. Oltre alla semplice individuazione, l'approccio adottato permette una categorizzazione fine delle offese, distinguendo tra tipologie specifiche quali omofobia, sessismo e razzismo, con l'obiettivo di fornire uno strumento efficace e scalabile per la moderazione automatica dei contenuti.

2 Metodologia

2.1 Dataset

Per il nostro studio, sono stati utilizzati quattro dataset principali, ognuno dei quali contiene esempi di contenuti testuali classificati in base alla presenza di contenuti offensivi o discriminatori. I dataset provengono da diverse fonti e trattano argomenti quali omofobia, sessismo, razzismo e contenuti neutrali. Di seguito vengono descritti i passaggi di preprocessing e la preparazione dei dati per il successivo utilizzo nei modelli di machine learning.

Raccolta dei Dati I dati provengono da due principali fonti pubbliche e sono stati utilizzati per ciascuna delle categorie di offesa analizzate:

- **Omofobia :** Il dataset relativo a contenuti omofobi 'e stato scaricato da `Hugging Face` e contiene un insieme di tweet etichettati come contenenti o meno omofobi. La versione utilizzata è disponibile al repository `JoshMcGiff/HomophobiaDetectionTwitterX`
- **Sessismo :** Il dataset relativo a contenuti sessisti è stato scaricato da `Kaggle` e include testi etichettati come sessisti o non sessisti. La versione utilizzata è `Sexism Detection in English Texts di Aadyasingh55`
- **Razzismo :** Il dataset relativo a contenuti razzisti proviene da `Kaggle` e contiene tweet annotati come razzisti o non razzisti. La versione utilizzata è `Racism Tweets di Raghad Abdullah`

- **Contenuti Neutrali :** Il dataset dei contenuti neutrali è stato generato artificialmente tramite un modello di linguaggio di grandi dimensioni (LLM), in particolare `ChatGPT di OpenAI`, al fine di creare esempi di testo non offensivi e bilanciare le classi presenti nel dataset.

Preprocessing dei Dati I dati sono stati sottoposti a una serie di operazioni di pulizia per rimuovere caratteri non rilevanti e preparare i testi per la successiva fase di tokenizzazione e vettorizzazione. Le seguenti operazioni sono state applicate a ciascun dataset:

- Rimozione di URL e menzioni di utenti (@username).
- Rimozione di hashtag.
- Rimozione di entità HTML e caratteri non ASCII.
- Normalizzazione del testo, inclusa la rimozione di ripetizioni di lettere.
- Conversione del testo in minuscolo.

Dopo queste operazioni di pulizia, il testo è stato pronto per essere utilizzato come input nei modelli di machine learning.

Bilanciamento e Unione dei Dataset Per affrontare eventuali squilibri nel numero di campioni tra le categorie, è stato eseguito un `undersampling` delle classi maggioritarie, riducendo il numero di esempi delle classi più rappresentate fino a ottenere un dataset bilanciato. Successivamente, i dataset di omofobia, sessismo, razzismo e contenuti neutrali sono stati concatenati in un unico dataset, che è stato ulteriormente bilanciato. Il dataset finale è stato quindi suddiviso in due set: uno di `addestramento` e uno di `test`, utilizzando una proporzione dell'80% per l'addestramento e del 20% per il test. Il dataset finale bilanciato è stato poi salvato in formato .csv per un facile utilizzo durante le fasi di addestramento e valutazione dei modelli.

Il dataset bilanciato contiene `621` campioni per ciascuna classe. Tutti i campioni sono stati etichettati correttamente con `text` e `category` e sono pronti per essere utilizzati nei modelli di classificazione.

2.2 Classificazione

Il processo di classificazione del contenuto testuale si articola in due fasi distinte, seguendo un approccio gerarchico e modulare volto a massimizzare l'efficienza e la precisione.

Fase 1 : Rilevamento del contenuto offensivo In un primo momento, ogni testo in input viene tradotto in lingua inglese e sottoposto a una pulizia e pre-processing. Successivamente, il contenuto viene valutato attraverso un classificatore preaddestrato di tipo `toxic-BERT`, specializzato nell'individuazione di contenuti tossici. Il modello restituisce una serie di etichette binarie associate a punteggi di confidenza per varie categorie di tossicità, tra cui: `toxicity, severe toxicity, obscene, identity attack, insult, threat, sexual explicit`. Se almeno una di queste etichette supera una soglia predefinita, il testo viene considerato potenzialmente offensivo e viene inoltrato alla seconda fase per una classificazione più specifica. In caso contrario il contenuto viene ignorato, evitando inutili computazioni.

Fase 2 : Classificazione specifica della tipologia di offesa Una volta identificato come offensivo, il testo viene eventualmente vettorializzato tramite `TF-IDF` e classificato in una delle seguenti categorie:

- `Omofobia` (homophobia)
- `Sessismo` (sexism)
- `Razzismo` (Racism)
- `Altro contenuto offensivo` (offensive content)

La classificazione avviene tramite più modelli supervisionati addestrati sul dataset bilanciato. I modelli impiegati includono: `regressione logistica` e `SVM lineare` Questo approccio consente di separare la componente generica di rilevamento da quella specifica di classificazione, migliorando la scalabilità del sistema e facilitando la manutenzione modulare. Inoltre, riduce il numero di falsi positivi e permette di utilizzare modelli più complessi solo quando necessario.

Pseudocodice del Processo

Input: Testo

1. Traduci il testo (se necessario)
2. Pulisci il testo (es. rimozione punteggiatura, stopword, emoji, ecc.)
3. Verifica la tossicità con `toxic-BERT`

Se il testo non è tossico:

4. Termina il processo

Altrimenti (testo tossico):

5. modello di classificazione:
   `Logistic Regression` o `SVM`:
   i. Vettorializza il testo con `TF-IDF`
   ii. Classifica il testo

3 Logistic Regression

Per l'addestramento di un classificatore lineare, si è fatto uso della **regressione logistica**, una tecnica largamente impiegata per problemi di classificazione binaria e multiclasse. La regressione logistica è un modello di classificazione lineare che stima la probabilità che un'osservazione appartenga a una determinata classe. Applica una **funzione sigmoide** alla combinazione lineare delle feature in input, producendo un output compreso tra 0 e 1. L'addestramento avviene minimizzando la *log-loss*, penalizzando le previsioni errate in base alla loro incertezza.

3.1 Preprocessing, Bilanciamento del Dataset e Rappresentazione dei Dati

Il dataset utilizzato è stato sottoposto a una fase di **pulizia e bilanciamento**, includendo unicamente le istanze appartenenti alle tre categorie di offesa: *omofobia*, *sessismo* e

razzismo. Le istanze appartenenti alla categoria generica "altro" sono state escluse per focalizzare l'analisi su classi ben definite. Dopo questa operazione, la distribuzione delle classi è stata controllata e visualizzata per garantire un dataset bilanciato e adatto all'addestramento supervisionato.

I testi trascritti sono stati suddivisi in **training set** (80%) e **test set** (20%), mantenendo la proporzione originale delle classi tramite stratificazione. Per trasformare i testi in una rappresentazione numerica adatta ai modelli di apprendimento automatico, è stato utilizzato il metodo **TF-IDF (Term Frequency - Inverse Document Frequency)**, con un massimo di 1000 feature, includendo uni-grammi e bi-grammi e rimuovendo le *stop words* in lingua inglese.

3.2 Modello di Regressione Logistica

Il modello utilizzato per questa fase iniziale di classificazione è la **regressione logistica multinomiale**, un metodo lineare interpretabile che permette di stimare la probabilità che un dato testo appartenga a ciascuna classe. L'algoritmo è stato addestrato utilizzando i dati vettorializzati e ottimizzato tramite la **discesa del gradiente (Gradient Descent)**, con un numero massimo di iterazioni pari a 1000 per garantire la convergenza.

3.3 Valutazione del Modello

Le performance del modello sono state valutate sull'insieme di test tramite diverse metriche: **accuracy**, **precision**, **recall** e **F1-score** per ciascuna classe. Inoltre, è stata prodotta una **matrice di confusione** per visualizzare le predizioni corrette e gli errori di classificazione per ciascuna categoria.

3.4 Analisi delle Feature

Per approfondire l'analisi e migliorare l'interpretabilità del modello, sono stati visualizzati i vocaboli con i **pesi più alti** nel vettore dei coefficienti della regressione logistica per ciascuna classe. Questi pesi rappresentano l'importanza relativa dei termini nella classificazione di un testo come offensivo in una specifica categoria. Tali visualizzazioni offrono un'utile panoramica sui segnali linguistici più informativi per il modello.

Il modello addestrato, insieme al vettorizzatore TF-IDF, è stato salvato per consentire la futura applicazione su nuovi dati e garantire la riproducibilità dell'esperimento. I risultati della valutazione sono stati inoltre esportati in formato `JSON`, rendendo possibile l'integrazione con altri moduli del sistema.

4 Linear SVM

Successivamente, è stato adottato un approccio supervisionato basato su **Support Vector Machine lineare (Linear SVM)**, implementato tramite la libreria `scikit-learn`. È una tecnica di apprendimento supervisionato utilizzata per compiti di classificazione e regressione. Nella variante usata in questo lavoro, la **Support Vector Classification (SVC)**, il modello cerca di trovare il **margine massimo** che separa le classi nel dataset. Solo un sottoinsieme di esempi (vettori di supporto) contribuisce a determinare il confine decisionale, rendendo il modello particolarmente efficace anche in presenza di dati ad alta dimensionalità.

Il dataset utilizzato è stato caricato da file `.csv` e filtrato per rimuovere le istanze con categoria etichettata come 3 (contenuto neutrale).

4.1 Preprocessing e Rappresentazione dei Dati

Anche in questo caso i testi sono stati rappresentati utilizzando la vettorizzazione **TF-IDF**, limitando il numero massimo di feature a 1000 e considerando *n-grammi* di ordine 1 e 2 (unigrammi e bigrammi). Sono state rimosse le *stop words* in lingua inglese. Il dataset è stato suddiviso in training set (80%) e test set (20%), mantenendo la proporzione delle classi mediante stratified sampling.

4.2 Modello di Classificazione

Il modello adottato è una **Linear SVM** con parametro `max_iter=1000` e `random_state=42` per garantire ripetibilità. Il classificatore è stato addestrato sui vettori TF-IDF e ha appreso un iperpiano separatore per ogni classe, massimizzando il margine tra le classi.

4.3 Analisi delle Feature

Per ciascuna classe, sono stati identificati i 10 *n-grammi* con peso maggiore nei vettori di coefficiente della SVM. Questi termini rappresentano quelli più fortemente associati alla rispettiva categoria di hate speech. I pesi sono stati visualizzati mediante **grafici a barre**. Il modello addestrato e il vettorizzatore sono stati serializzati tramite `joblib` e salvati in formato `.pkl`. I risultati della valutazione sono stati salvati in formato `JSON` per un facile riutilizzo.

5 Acquisizione e Preprocessing dei Dati Multimediali

Per supportare l'utente nell'analisi di nuovi contenuti multimediali provenienti da piattaforme social, è stata implementata una procedura di estrazione e trascrizione di dati audio e testuali. È importante sottolineare che i dati acquisiti tramite questa procedura non sono utilizzati per l'addestramento dei modelli, che si basano esclusivamente sui dataset precedentemente raccolti e pre-elaborati.

5.1 Download e Estrazione Audio da TikTok

È stato sviluppato uno script basato sulla libreria `yt-dlp` per scaricare l'audio dai video di TikTok. Il sistema:

- Verifica la validità dell'URL come link TikTok.
- Effettua tentativi multipli di download utilizzando cookie di diversi browser per superare eventuali limitazioni.
- Estrae e salva temporaneamente l'audio in formato MP3 per la successiva trascrizione.
- Elimina il file temporaneo dopo aver acquisito il contenuto audio.

5.2 Estrazione del Testo da Threads tramite Web Scraping

Per i post di Threads, si è utilizzato un approccio di web scraping con `Selenium` e `BeautifulSoup` che:

- Carica la pagina in modalità headless per automatizzare la navigazione.
- Identifica e analizza gli elementi HTML contenenti autore, data e testo del post.
- Gestisce eventuali errori dovuti a modifiche nella struttura della pagina o assenza di dati.

5.3 Trascrizione Audio con Whisper

I file audio estratti vengono trascritti in testo tramite il modello **Whisper** di OpenAI, un sistema di riconoscimento vocale automatico avanzato, per consentire un'analisi testuale immediata da parte dell'utente.

5.4 Considerazioni e Criticità Tecniche

Durante la progettazione di questa pipeline sono emerse alcune difficoltà operative, legate soprattutto alla natura delle piattaforme da cui si estraggono i contenuti. L'utilizzo di cookie di browser diversi si è reso necessario per superare restrizioni dinamiche applicate da TikTok, ma può risultare instabile e richiedere aggiornamenti frequenti. Allo stesso modo, l'approccio tramite web scraping con Selenium e BeautifulSoup, sebbene efficace, è soggetto a rottura nel momento in cui la struttura HTML dei siti cambia, rendendo necessaria una manutenzione continua dello script.

Infine, anche la gestione dei file temporanei e delle trascrizioni comporta attenzione nella pulizia e ottimizzazione delle risorse locali, specialmente in scenari di uso ripetuto. Nonostante queste criticità, la pipeline si è dimostrata funzionale e adattabile per il supporto all'analisi manuale di contenuti, mantenendo la flessibilità necessaria a un contesto sperimentale.

6 Risultati

6.1 Logistic Regression

Abbiamo valutato il modello di regressione logistica sul nostro dataset bilanciato, ottenendo un'accuratezza complessiva (*accuracy*) del **92.76%**. Di seguito riportiamo una tabella riassuntiva con le principali metriche di performance suddivise per classe.

Tabella 1: Metriche di classificazione per il modello di Regressione Logistica

Classe	**Precision**	**Recall**	**F1-score**	**Support**
0 (Omofobia)	0.960	0.968	0.964	124
1 (Sessismo)	0.897	0.911	0.904	124
2 (Razzismo)	0.926	0.904	0.915	125
Macro Avg	0.928	0.928	0.928	373
Weighted Avg	0.928	0.928	0.928	373

6.2 Linear SVM

Il modello SVM lineare ha ottenuto risultati superiori rispetto alla regressione logistica, raggiungendo un'*accuracy* del **94.91%**. Come si evince dalla Tabella 2, tutte e tre le classi vengono identificate con *precision* e *recall* molto elevate, in particolare la classe 0 (Omofobia), per cui il modello mostra una quasi perfetta capacità di classificazione.

Tabella 2: Metriche di classificazione per il modello SVM Lineare

Classe	**Precision**	**Recall**	**F1-score**	**Support**
0 (Omofobia)	0.968	0.984	0.976	124
1 (Sessismo)	0.914	0.944	0.929	124
2 (Razzismo)	0.966	0.920	0.943	125
Macro Avg	0.950	0.949	0.949	373
Weighted Avg	0.950	0.949	0.949	373

7 Conclusioni e Sviluppi Futuri

Il lavoro svolto ha portato allo sviluppo di un sistema modulare per la classificazione automatica di contenuti offensivi provenienti da piattaforme social, integrando tecniche di preprocessing, machine learning classico e deep learning. L'architettura proposta, articolata in due fasi, ha dimostrato una buona capacità di filtrare contenuti potenzialmente tossici e categorizzarli in modo accurato all'interno delle classi considerate (omofobia, sessismo, razzismo).

In particolare, i modelli supervisionati addestrati sui dati trascritti hanno ottenuto prestazioni significative, con la **Support Vector Machine** che ha raggiunto un'accuracy del **94.91%**, superando la regressione logistica, che si è comunque attestata su ottimi livelli con un'accuracy del **92.76%**. L'utilizzo del modello `toxic-BERT` nella fase iniziale ha consentito di filtrare efficacemente contenuti neutri, riducendo la computazione necessaria nella fase di classificazione specifica e migliorando l'efficienza complessiva del sistema.

Il sistema di acquisizione automatica dei contenuti audio e testuali, basato su tecniche di scraping e riconoscimento vocale (tramite Whisper), si è dimostrato funzionale, sebbene siano emerse criticità legate all'instabilità delle piattaforme di origine e alla necessità di aggiornamenti frequenti negli script di estrazione.

Sviluppi Futuri

Sulla base dei risultati ottenuti, sono state identificate diverse direzioni per il miglioramento e l'estensione del progetto:

- **Espansione delle categorie offensive**: Integrare nuove classi di contenuti dannosi, come islamofobia, abilismo, body shaming o xenofobia, al fine di aumentare la granularità dell'analisi.
- **Incremento della dimensione e varietà del dataset**: Raccogliere ulteriori dati da piattaforme social differenti (es. X, YouTube, Instagram) per migliorare la robustezza dei modelli e garantire una maggiore generalizzazione.

- **Utilizzo di modelli transformer avanzati**: Integrare modelli come `RoBERTa`, `DistilBERT` o `DeBERTa` per migliorare ulteriormente le performance di classificazione, oppure valutare l'impiego di tecniche ensemble.
- **Classificazione multimodale**: Estendere l'analisi al contenuto visivo (frame video o immagini), combinando le informazioni testuali e visive per una classificazione multimodale completa.
- **Interfaccia utente e deployment**: Realizzare un'interfaccia web user-friendly per permettere a moderatori o ricercatori di caricare contenuti e ottenere una valutazione automatica. Inoltre, si prevede il deployment del sistema come servizio API accessibile in remoto.
- **Valutazione dell'impatto etico e sociale**: Considerare gli aspetti legati alla trasparenza dei modelli e alla possibilità di falsi positivi, sviluppando metodi per spiegare le decisioni dei modelli e gestire l'eventuale contestazione da parte degli utenti.

In conclusione, il sistema proposto rappresenta un passo significativo verso lo sviluppo di strumenti intelligenti per la moderazione automatica dei contenuti online, con potenzialità di applicazione in contesti reali e un ampio margine di miglioramento tramite futuri sviluppi.