UNIVERSITÀ DI PISA

Corso di Laurea in Informatica Umanistica

TESI DI LAUREA

Learn to Detox: un'applicazione web per rilevare l'hate speech

Relatore:
Prof. Claudio Gallicchio

Correlatore:
Prof. Alessandro Lenci

Candidato:
Agnese Marchionneschi

ANNO ACCADEMICO 2020/2021

Indice

Elenco delle figure

Introduzione

La possibilità di rilevare automaticamente la presenza di contenuti tossici all'interno di un testo si configura come una risposta all'esigenza di prevenire e contrastare le forme di aggressività sul Web. I comportamenti degli utenti sui Social Network continuano ad evolversi in un contesto estramemente mutevole che scaturisce problematiche complesse come il diffondersi dell'odio in rete e dei fenomeni di disinformazione. La pandemia di COVID-19 ad esempio, legandosi fin da subito ad una pericolosa diffusione di disinformazione, ha dato origine a nuove forme di odio e discriminazione online che si sono riversate su gruppi di persone considerate colpevoli di quanto avvenuto a livello globale. Di conseguenza le piattaforme digitali necessitano di soluzioni che ostacolino una comunicazione basata sull'utilizzo di stereotipi e manifestazioni di aggressività.

Il progetto realizzato per questa tesi "Learn to Detox", si propone di fornire un esempio concreto di un'applicazione Web il cui scopo principale è quello di rilevare automaticamente linguaggio offensivo all'interno di contenuti testuali, adattandosi alle esigenze di un contesto in cui i flussi comunicativi sono intensi e complessi. Il progetto sarà illustrato nel dettaglio in seguito ad un quadro generale delle condizioni preliminari.

In particolare il Capitolo 1 tratterà il contesto generale in cui è opportuno collocare il lavoro svolto per realizzare il progetto, analizzando le esigenze dalle quali deriva l'urgenza di impedire la diffusione di un'ideologia dell'odio. Il Capitolo 2 si soffermerà invece sulle tecnologie ed i linguaggi utilizzati per realizzare il progetto. Il Capitolo 3 tratterà le caratteristiche degli strumenti utilizzati per classificare il testo in seguito ad un'introduzione teorica. Il Capitolo 4 presenterà le funzionalità dell'applicazione Web ed esaminerà i risultati restituiti dall'applicazione con il supporto di alcuni esempi. Il Capitolo 5 descriverà la realizzazione e il funzionamento dell'estensione Chrome che completa il progetto Learn to Detox. Infine il Capitolo Conclusioni riassumerà i principali risultati della tesi ripercorrendo le tappe fondamentali della realizzazione del progetto. Il codice dell'applicazione Web e dell'estensione Chrome di Learn to Detox è disponibile e consultabile su *GitHub*. (*https://github.com/lasaghnes/LearnToDetox*)

1. Il contesto

All'interno di questo Capitolo verrà introdotto il contesto in cui è possibile collocare la genesi del progetto "Learn to Detox", partendo nella Sezione 1 dalla definizione dell'incitamento all'odio e analizzando le strategie per contrastarlo. We Are Social ha presentato nel 2021 il report annuale che analizza lo scenario social e digital a livello locale e globale, realizzato in collaborazione con Hootsuite. (*Digital 2021: i dati globali - We Are Social Italia* - wearesocial.com) In base alla loro analisi a gennaio 2021, su una popolazione globale di 7,83 miliardi di persone si contavano 4,2 miliardi di persone attive in almeno un social media, la penetrazione delle piattaforme social si attestava quindi al 53% della popolazione mondiale. (We Are Social Italia) I dati raccolti riportano inoltre che i social media più utilizzati sono: Facebook in prima posizione, YouTube e Whatsapp. Nonostante l'aggressività e la violenza rientrino nelle dinamiche intrinseche biologiche e psichiche degli esseri umani, queste manifestazioni assumono nuove caratteristiche sul Web e richiedono nuove forme di regolazione.

La libertà d'espressione e il diritto all'anonimato contribuiscono a rendere Internet una realtà libera da ogni condizionamento, rendendo altresì la versione online del discorso d'odio una questione complessa. Diviene perciò indispensabile individuare una risposta all'aggressività efficace senza contraddire le libertà fondamentali dell'individuo, che percorra canali diversi per non stimolare gli stessi meccanismi che cerca di combattere.

1.1 L'esigenza di definire e combattere l'hate speech

Ci sono numerosi problemi legati alla definizione dell'espressione "hate speech" che devono essere risolti al fine di sviluppare un modello di analisi coerente, tenendo presente che il confine tra libertà d'espressione e violenza può sovente rivelarsi sottile. Nockleby ha definito l'incitamento all'odio come "qualsiasi forma di comunicazione che denigri una persona o un gruppo sulla base di razza, colore, etnia, genere, orientamento sessuale, nazionalità, religione o altra caratteristica." (Warner e Hirschberg, 2012) Tuttavia se dispo-

niamo di metodi per identificare l'hate speech, ciò che manca è un definizione concordata a livello internazionale. Una delle iniziative più importanti a livello europeo è stata la campagna giovanile "Movimento contro il discorso dell'odio" (No Hate Speech Movement) condotta dal Consiglio d'Europa dal 2012 al 2017, che aveva proprio lo scopo di attirare l'attenzione sul discorso dell'odio online. (*No Hate Speech Movement* - www.coe.int)

Una definizione più recente e di ampie vedute arriva dall'Ecri (Commissione europea contro il razzismo e l'intolleranza, 2015): "fomentare, promuovere o incoraggiare, sotto qualsiasi forma, la denigrazione, l'odio o la diffamazione nei confronti di una persona o di un gruppo, nonché il fatto di sottoporre a soprusi, insulti, stereotipi negativi, stigmatizzazione o minacce una persona o un gruppo e la giustificazione di tutte queste forme o espressioni di odio testé citate, sulla base della "razza", del colore della pelle, dell'ascendenza, dell'origine nazionale o etnica, dell'età, dell'handicap, della lingua, della religione o delle convinzioni, del sesso, del genere, dell'identità di genere, dell'orientamento sessuale e di altre caratteristiche o stato personale". (*ECRI General Policy Recommendation N°15*)

Certamente la mancanza di una definizione universalmente condivisa si configura come il primo ostacolo alla realizzazione di un sistema uniforme, coerente ed efficace sia da un punto di vista legislativo che operativo.

1.1.1 Ambiti di diffusione dell'odio sul Web

Risulta opportuno premettere come l'hate speech e più in generale tutte le dinamiche aggressive, non siano ovviamente prodotti ma al più favoriti dai Social Network. Non sono solo le piattaforme più comuni come Facebook, Twitter, YouTube, a dover gestire politiche di prevenzione dei contenuti tossici, occore menzionare anche i forum, i quotidiani online e tutte le agenzie e i professionisti della comunicazione. In virtù delle caratteristiche intrinseche della comunicazione online, il Web tende a divenire un terreno particolarmente fertile per la comparsa di comportamenti tendenti all'incitamento all'odio. La prima caratteristica che marca le forme di dialogo in rete è la **permanenza**. Questa comporta infatti la possibilità che le manifestazioni di odio rimangano impresse per lunghi periodi di tempo e che vengano trasferite da una piattaforma all'altra. Ad esempio sul noto So-

cial Network Twitter sono i trending topics a dirigere l'attenzione degli utenti e qualsiasi questione rientri tra gli argomenti di tendenza, verrà diffusa, discussa e trasmessa in modo capillare.

Un'altra importante caratteristica è il **ritorno** imprevedibile dei contenuti. Rimuovendo un contenuto dal Web non si ha la certezza della sua scomparsa, esso può riapparire su un'altra piattaforma o all'interno della stessa, con una forma diversa. Inoltre, avendo la possibilità di agire senza essere identificati, sono molti gli utenti che si sentono incoraggiati a esprimere sentimenti negativi. Infine la **trasnazionalità**, ovvero l'assenza di confini che permette la diffusione capillare dei messaggi, responsabile di rendere sicuramente più complessa un'omogeneità legislativa abbastanza coesa da riuscire a contrastare il fenomeno della tossicità dei contenuti.

1.1.2 Differenti vedute sul fenomeno dell'incitamento all'odio

Coesistono due visioni opposte riguardo all'atteggiamento da tenere per contrastare le manifestazioni di odio online:

• L'approccio europeo, secondo cui la rete andrebbe regolamentata più rigidamente, per ostacolare la diffusione di opinioni discriminatorie e non rispettose del principio della dignità umana.

• L'approccio americano, che invece sostiene che regolamentare la libertà di espressione su Internet non servirebbe a tale scopo ma, al contrario, avrebbe come conseguenza quella di alterare non solo il sistema di protezione della libertà di manifestazione del pensiero, ma anche le strategie commerciali dei grandi player dell'economia digitale. (*Hate speech, la normativa in Europa e Usa sull'odio online - Agenda Digitale* - www.agendadigitale.eu)

Su un piano concreto la Commisione Europea, il 31 maggio 2016, ha presentato il "Codice di condotta sulle espressioni illegali di odio online" creando inoltre un Sottogruppo ad alto livello per la lotta contro l'hate speech online. (*Codice di condotta dell'UE per contrastare l'illecito incitamento all'odio* - ec.europa.eu)

Mentre per quanto riguarda l'America, oltre ad esserci un grande divario in materia tra i democratici e i repubblicani, le piattaforme Social non sono ritenute responsabili per i

contenuti che ospitano.

Nel 2016 alcune delle più note aziende IT (Microsoft, Facebook, Twitter, YouTube) si impegnavano quindi pubblicamente ad adottare alcune misure atte a realizzare l'obiettivo di rendere lo spazio Web privo di "hate speech", come ad esempio la predisposizione di procedure chiare per esaminare le segnalazioni riguardanti forme di incitamento all'odio e l'esame di tali segnalazioni in meno di 24 ore.

Da allora, con il contributo di organizzazioni operative negli stati membri, la Commissione europea ogni anno pubblica un report in cui raccoglie gli esiti di un lavoro di monitoraggio volto a verificare lo stato di attuazione del codice di condotta. Nel 2019 Amnesty International Italia è entrata a far parte delle organizzazioni che partecipano a tale attività di osservazione, segnalando contenuti che incitano all'odio e monitorando l'esito delle segnalazioni. Al termine dei quattordici mesi di attività, la Commissione elaborò una relazione finale che riconobbe l'esistenza di una piramide dell'odio alla cui base si pongono stereotipi, rappresentazioni false o fuorvianti, insulti, linguaggio ostile 'normalizzato' o banalizzato e, ai livelli superiori, le discriminazioni e quindi il linguaggio e i crimini di odio.

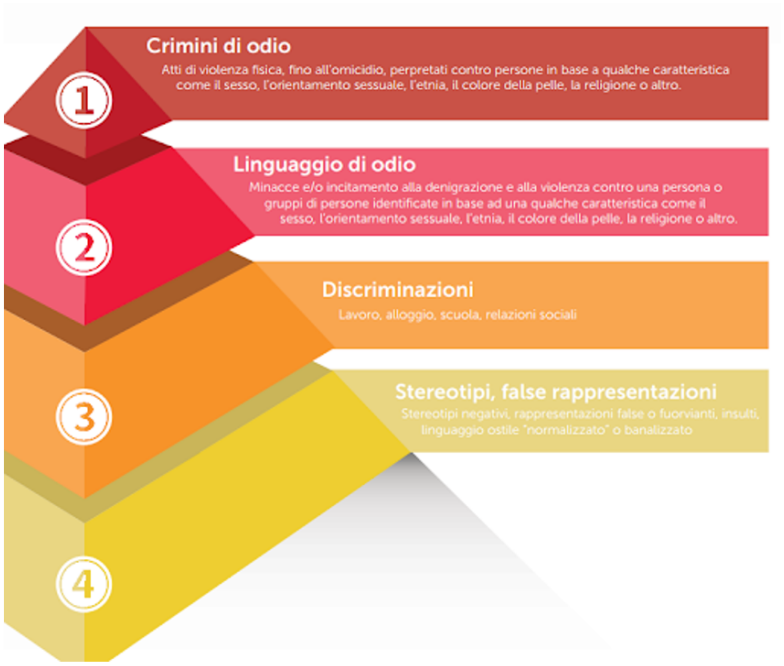

Figura 1.1: Camera dei Deputati, La piramide dell'odio in Italia, 2017

Fonte: Commissione "Jo Cox" sui fenomeni di odio, intolleranza, xenofobia e razzismo

Tuttavia nonostante i progressi fatti, sono ancora molti i punti critici e ci sono grandi differenze tra un social network e l'altro.

Infatti a parità di tipologia e gravità di contenuto che incita all'odio, la frequenza con cui la rimozione si verifica è estremamente variabile da una piattaforma all'altra e gli strumenti che ogni piattaforma mette a disposizione dell'utente sono diversi.

1.1.3 Strategie per contrastare la diffusione di contenuti tossici

In linea di principio le strategie per contrastare la diffusione di contenuti tossici si basano su una combinazione di intelligenza artificiale, segnalazioni degli utenti e personale noto come moderatori dei contenuti.

Un tattica considerata efficace e tenuta in considerazione da Facebook è il *Counter Speech*: una risposta diretta affidata agli utenti del Web che cerca di minare i contenuti che incitano all'odio. (*Counterspeech — Dangerous Speech Project* - dangerousspeech.org)

Esistono due tipi di questa tipologia di replica: campagne organizzate e risposte spontanee. Lo scopo comune è quello di avere un effetto positivo su chi parla ma anche sul pubblico, comunicando in modo esplicito le norme che rendono il discorso socialmente inaccettabile. Un altro importante strumento utilizzato per rilevare e combattere la tossicità dei testi sono le tecnologie nel campo dell'intelligenza artificiale.

Un noto esempio è "Perspective", la API (Application Programming Interface), lanciata da Google in collaborazione con Jigsaw. Perspective utilizza modelli di apprendimento automatico per identificare i commenti offensivi. I modelli classificano una frase in base all'impatto percepito che il testo può avere in una conversazione. (*GitHub - conversationai/perspectiveapi*) Il risultato può essere utilizzato per fornire feedback agli utenti, aiutare i moderatori a rivedere più facilmente i commenti o aiutare i lettori a filtrare i contenuti che vogliono vedere.

Anche Facebook ha compiuto dei progressi: le loro tecnologie sono addestrate a riconoscere i contenuti tossici degli utenti, vengono richiamate ogni volta che un nuovo post appare e individuano se il contenuto violi o meno le linee guida della community dell'azienda. Con tutta evidenza l'intento di mitigare l'incitamento all'odio che plausibilmente inquina il mondo reale come quello virtuale, può essere perseguito contando sul supporto dell'intelligenza artificiale. In primo luogo, puntando su algoritmi di apprendimento automatico è possibile ottenere la rimozione immediata di contenuti tossici, dall'altro lato essi consentontono di ridurre la viralità, contenere la diffusione e la portata dei fenomeni di incitamento all'odio.

OpenWeb, una piattaforma di social engagement che si occupa di monitorare e filtrare le sezioni dei commenti online per diversi editori, ha portato avanti uno studio riguardo ad una metodologia di prevenzione della tossicità online. (*Nudge Theory Examples In Online Discussions — OpenWeb* - www.openweb.com) La strategia sperimentata prevedeva l'utilizzo di modelli Perspective su milioni di commenti per rilevare abusi e tossicità in tempo reale. Combinando questo meccanismo di punteggio con la loro tecnologia proprietaria, fornivano una funzione di feedback in tempo reale, offrendo agli utenti l'opportunità di modificare il loro messaggio, in caso di avviso di violazione delle linee guida della community. I risultati hanno evidenziato che il comportamento dell'utente può essere

influenzato in tempo reale, scoraggiando la pubblicazione di contenuti tossici, ma che possono anche nascere comportamenti adattivi volti ad aggirare la rilevazione di linguaggio offensivo per sfuggire all'analisi dell'algoritmo.

1.1.4 Rilevare la tossicità come primo passo fondamentale

Appare chiaro come le strategie e le metodologie finalizzate a prevenire oppure eliminare le discriminazioni e l'odio sul Web si basino in primo luogo su una solida capacità di distinguere i contenuti. Senza dubbio la sfida più complessa relativa al perseguimento dell'obiettivo, è il miglioramento degli algoritmi che esaminano i contenuti, poiché una maggiore precisione comporta una selezione efficace e in linea con il principio cardine che governa il Web, la libertà d'espressione.

2. Tecnologie utilizzate nel progetto

Nel capitolo corrente verranno introdotte le tecnologie e i linguaggi utilizzati per realizzare il progetto "Learn to Detox". In particolare nella Sezione 2.1 verrà approfondito il linguaggio HTML, utilizzato per costruire le pagine Web, nella Sezione 2.2 sarà possibile trovare informazioni riguardo al linguaggio CSS, utilizzato nel progetto per la resa grafica delle pagine Web, la Sezione 2.3 introdurrà invece Bootstrap, il popolare framework per lo sviluppo di siti Web responsive. La Sezione 2.4 riguarderà JavaScript, utilizzato per l'interazione dinamica con le pagine Web e la Sezione 2.5 tratterà il runtime system Node.js, il cui utilizzo è finalizzato all'implementazione di JavaScript lato server. La Sezione 2.6 introdurrà brevemente le tecnologia utilizzata nell'applicazione Web per il riconoscimento vocale e la Sezione 2.7 infine offrirà un quadro sintetico dei passaggi seguiti per realizzare un'estensione per il browser Google Chrome.

2.1 HTML

L'HTML, acronimo di HyperText Markup Language (traduzione letterale: linguaggio a marcatori per ipertesti) è un linguaggio di markup creato alla fine degli anni '80 da Tim Berners-Lee, un ricercatore del Cern di Ginevra, per organizzare i documenti scientifici.

Il linguaggio di marcatura (o di markup) consiste in un insieme di regole che descrivono i meccanismi di rappresentazione (strutturali, semantici, presentazionali) o layout di un testo; facendo uso di convenzioni rese standard, i cosiddetti tag, che non sono altro che dei marcatori, tali regole sono utilizzabili su più supporti. L'HTML è un linguaggio di pubblico dominio, la cui sintassi è stabilita dal *World Wide Web Consortium (W3C)* (www.w3.org). L'HTML è un linguaggio di formattazione che descrive le modalità di impaginazione o visualizzazione grafica (layout) del contenuto, testuale e non, di una pagina Web attraverso tag di formattazione. Sebbene supporti l'inserimento di script e oggetti esterni, non è un linguaggio di programmazione: non prevede infatti alcuna definizione di variabili, strutture dati, funzioni o strutture di controllo che possano realizzare programmi,

il suo codice è in grado soltanto di strutturare e decorare dati testuali. Il componente principale della sintassi di questo linguaggio è l'elemento, inteso come struttura di base a cui è delegata la funzione di indicare al browser delle informazioni. Ogni elemento è racchiuso all'interno di marcature dette tag, costituite da una sequenza di caratteri racchiusa tra due parentesi angolari. Quando il tag deve essere applicato a una sezione di testo o di codice, l'ambito di applicazione deve essere delimitato fra un tag di apertura ed uno di chiusura (chiusura esplicita), che coincide col tag di apertura preceduto da una barra (/) dopo la parentesi angolare aperta. I documenti HTML vengono immagazzinati sui dischi rigidi di macchine elaboratrici (computer-server) costantemente collegate e connesse alla rete Internet. Su queste macchine è installato un software specifico (Web server) che si occupa di produrre e inviare i documenti ai browser degli utenti che ne fanno richiesta usando il protocollo HTTP per il trasferimento dati. Spesso il documento HTML viene generato del tutto o parzialmente tramite un codice eseguibile residente sul server Internet (elaborazione lato server) in grado di interagire con altre applicazioni presenti sul server stesso, come per esempio una base di dati, e inviare poi al browser il risultato finale, realizzando le cosiddette pagine Web dinamiche con cui un utente può compiere operazioni interattive avanzate. In altri casi invece alcuni tipi di elaborazione sono operati lato client con linguaggi come JavaScript. L'ultima versione della tecnologia HTML, è HTML5 pubblicato come W3C Recommendation da ottobre 2014. Relativamente al progetto il linguaggio HTML5 è il centro nevralgico delle pagine Web che sono state create.

2.2 CSS

Il CSS, acronimo di Cascading Style Sheets (fogli di stile a cascata), è un linguaggio usato per definire la formattazione di documenti HTML, XHTML e XML. È usato per programmare la resa grafica di documenti scritti in un linguaggio di markup. Le classi CSS sono usate per specificare attributi grafici come font, dimensione, colore, spaziatura, bordo e posizione degli elementi all'interno di una rappresentazione Web. I fogli di stile CSS si applicano a cascata: tutte le regole sono valide, ma prevale quella più specifica. I CSS nascono ufficialmente nel 1996, con il rilascio della versione di livello 1. Tuttavia la loro

storia e la genesi dell'idea che portò alla loro realizzazione risalgono al 1993. All'epoca infatti le pagine Web non avevano stili, e la loro formattazione era affidata unicamente alle preferenze del browser. Gli utenti cominciarono così a chiedere agli sviluppatori di permettere agli autori di influenzare la presentazione delle proprie pagine. (Frain, 2020)

2.3 Bootstrap

Bootstrap è una raccolta di strumenti grafici, stilistici e di impaginazione che permettono di avere a disposizione una gran quantità di funzionalità e di stili modificabili e adattabili a seconda delle esigenze. Ha un utilizzo molto versatile, è compatibile con tutte le ultime versioni dei principali browser. La sua funzione principale è il Design Responsivo, l'approccio per il quale la progettazione e lo sviluppo di un sito dovrebbero adattarsi al comportamento e all'ambiente dell'utente, in base a fattori come le dimensioni dello schermo, la piattaforma e l'orientamento del device. La pratica consiste in un mix di griglie, layout e immagini flessibili, più un uso accorto delle media queries CSS. (Marcotte, 2017) Bootstrap nasce nell'anno 2010 per opera degli sviluppatori Mark Otto e Jacob Thornton. Inizialmente si presentava come un progetto interno a Twitter, ma successivamente è diventato indipendente ed è perciò utilizzabile dagli sviluppatori di tutto il mondo come base per la realizzazione di interfacce Web. Al momento è considerato il più efficiente ed il più utilizzato framework per rendere i siti Web responsive. (Temere, 2017)

All'interno del progetto è stato usato Bootstrap per costruire e rendere responsive le pagine dell'applicazione Web, all'interno del Capitolo 4 verranno forniti degli esempi in cui è stato utilizzato il framework.

2.4 JavaScript

JavaScript è un linguaggio di programmazione orientato agli oggetti e agli eventi utilizzato nella programmazione Web lato client (esteso poi anche al lato server).

Originariamente sviluppato da Brendan Eich della Netscape Communications con il nome di Mochan e successivamente di LiveScript, in seguito è stato rinominato "Java-

Script". Standardizzato per la prima volta il 1997 dalla ECMA (European Computer Manufacturers Association) con il nome ECMAScript, l'ultimo standard, di giugno 2017, è ECMA-262 Edition 8 ed è anche uno standard ISO (International Organization for Standardization). Di seguito sono elencate le principali caratteristiche di JavaScript.

- È un linguaggio interpretato: il codice non viene compilato, ma eseguito direttamente; in JavaScript lato client, il codice viene eseguito dall'interprete contenuto nel browser dell'utente.
- Rispetta le funzionalità tipiche dei linguaggi di programmazione ad alto livello (strutture di controllo, cicli, ecc.) e consente l'utilizzo del paradigma object oriented.
- È un linguaggio debolmente tipizzato, ovvero le variabili possono riferirsi a valori di qualsiasi tipo, che possono cambiare dinamicamente in seguito a manipolazioni esterne. (Flanagan, 1998)

All'interno del progetto, il linguaggio JavaScript è stato utilizzato sia per realizzare l'applicazione Web, sia per realizzare l'estensione Chrome. In entrambi i casi la sua funzione è stata quella di implementare il modello Toxicity.js, allo scopo di rilevare elementi tossici all'interno del testo, ma anche di rendere dinamiche le pagine gestendo l'interazione dell'utente con i pulsanti e gli input presenti nella pagina.

2.5 Node.js

Javascript è nato come linguaggio utilizzabile esclusivamente lato client e quindi eseguibile solo all'interno di un browser. Grazie a Node.js, che è un runtime Javascript, è divenuto possibile utilizzare JavaScript per svolgere qualsiasi tipo di programma: da elaborazioni statistiche a interazioni con rete e database, fino all'utilizzo come server. Node.js è basato sul JavaScript Engine V8, che è il runtime di Google utilizzato anche da Chrome e disponibile sulle principali piattaforme. (Syed e Bean, 2014) L'utilizzo del modello di Tensorflow.js utilizzato nel progetto si basa su Node.js per la parte server side.

2.6 Il riconoscimento vocale con la Web Speech Api

Nell'applicazione Web realizzata per questa tesi è stata utilizzata una tecnologia che implemente il riconoscimento vocale attraverso l'uso del microfono del dispositivo, per classificare testo trascritto in seguito alla ricezione del parlato. Sarà possibile osservare nel dettaglio il funzionamento di questa feature nel Capitolo 4.

La Web Speech Api, introdotta alla fine del 2012, consente agli sviluppatori Web di fornire funzionalità di input vocale e output di sintesi vocale in un browser Web, tenendo in considerazione la privacy degli utenti. Prima di consentire all'applicazione o sito Web di accedere alla voce tramite microfono, l'utente deve infatti concedere esplicitamente l'autorizzazione. Se la pagina che esegue questa API utilizza il protocollo HTTPS, il browser richiede l'autorizzazione una sola volta, altrimenti lo fa ogni volta che inizia un nuovo processo. La Web Speech Api implementa un'interfaccia complessa, chiamata SpeechRecognition. Per creare un'istanza di un riconoscimento vocale, è necessario utilizzare la funzione speechRecognition() come mostrato di seguito (Codice 2.1):

```
1   var recognizer = new speechRecognition();
```

Code 2.1: Istanza di riconoscimento vocale

Questo oggetto implementa i seguenti metodi:

- onstart: implementa una callback che viene attivata quando il servizio di riconoscimento ha iniziato ad ascoltare l'audio con l'intenzione di riconoscere.
- onresult: implementa una callback che viene attivata quando il riconoscimento vocale restituisce un risultato. L'evento deve utilizzare l'interfaccia SpeechRecognitionEvent. Questo è il metodo che è stato scelto per essere implementato nel progetto "Learn to Detox".
- onerror: implementa una callback che viene attivata quando si verifica un errore di riconoscimento vocale. L'evento deve utilizzare l'interfaccia SpeechRecognitionError.

- onend: implementa una callback che viene attivata quando il servizio è stato disconnesso.

Oltre a questi metodi, è possibile configurare l'oggetto di riconoscimento vocale utilizzando le seguenti proprietà:

- continuous: imposta il tipo di riconoscimento. Se il suo valore è impostato su true abbiamo un riconoscimento continuo, altrimenti il processo termina non appena l'utente smette di parlare. Per impostazione predefinita è impostato su false. All'interno dell'applicazione Web realizzata è impostato su true, il riconoscimento termina non appena l'utente clicca nuovamente sull'icona del microfono.

- lang: specifica la lingua di riconoscimento. Per impostazione predefinita corrisponde alla lingua del browser. Nel caso dell'applicazione Web è stata impostata sulla lingua inglese per coerenza con il modello Toxicity.js utilizzato per classificare il testo.

- interimResults: specifica la restituzione di risultati intermedi. Se il suo valore è impostato su true, sarà possibile avere accesso a risultati intermedi che è possibile mostrare agli utenti per fornire feedback. Se il valore è false, sarà possibile ottenere i risultati solo al termine del riconoscimento. Per impostazione predefinita è impostato su false. Nel caso dell'applicazione Web questa feature non è stata utilizzata in quanto non utile ai fini della classificazione del testo.

Result restituisce un oggetto di tipo SpeechRecognitionEvent che contiene i seguenti dati:

- results[i]: un array contenente i risultati del riconoscimento. Ogni elemento dell'array corrisponde a una parola riconosciuta. Questo array è stato utilizzato nel progetto per trascrivere i risultati del riconoscimento del parlato allo scopo di rilevare la presenza di linguaggio offensivo con il modello Toxicity.js.

- resultIndex: l'indice del risultato del riconoscimento corrente.

- results[i].isFinal: un booleano che indica se il risultato è definitivo o provvisorio.

- results[i][j]: un array 2D contenente parole riconosciute alternative. Il primo elemento è la parola riconosciuta più probabile.

- results[i][j].transcript: la rappresentazione testuale delle parole riconosciute. results[i][j].confidence: la probabilità che il risultato sia corretto. Il valore va da 0 a 1.

I risultati del riconoscimento sono stati trascritti tramite la proprietà di sola lettura dell'interfaccia SpeechRecognitionResult: **transcript**. Essa restituisce una stringa contenente la trascrizione delle parole riconosciute, in caso di riconoscimento continuo, vengono inclusi spazi bianchi ove necessario, in modo tale che la concatenazione di messaggi consecutivi produca una trascrizione corretta della sessione.

2.7 Realizzare un'estensione Chrome

Le estensioni Chrome sono piccoli programmi che aggiungono varie funzionalità al browser Google Chrome, vengono implementate utilizzando tecnologie Web standard come HTML, CSS, JavaScript. Un'estensione è in realtà una cartella compressa, più precisamente un archivio zip con una firma che contiene diversi file. (*Creare e pubblicare estensioni e app di Chrome personalizzate - Guida di Google Chrome Enterprise* - developer.chrome.com)

Ogni estensione per Google Chrome necessita di un file di configurazione che fornisce i dettagli del programma come il nome, l'icona, la versione etc. Questo file si chiama manifest ed è in formato JSON (JavaScript Object Notation). Questo formato è adatto all'interscambio di dati fra applicazioni client/server, è basato sul linguaggio JavaScript e supporta i seguenti tipi di dati:

- booleani (true e false);

- interi, numeri in virgola mobile;

- stringhe racchiuse da doppi apici (");

- array (sequenze ordinate di valori, separati da virgole e

- racchiusi in parentesi quadre [ ]);
- array associativi (sequenze coppie chiave-valore separate da virgole racchiuse in parentesi graffe); null.

Il codice seguente (Code 2.2) mostra i campi manifest supportati per le estensioni e nello specifico quelli richiesti, quelli raccomandati e quelli opzionali.

```
[
    tools_page": "devtools.html",
  "differential_fingerprint": ...,
  "event_rules": [{...}],
  "externally_connectable": {
    "matches": ["*://*.example.com/*"]
  },
  "file_browser_handlers": [...],
  "file_system_provider_capabilities": {
    "configurable": true,
    "multiple_mounts": true,
    "source": "network"
  },
  "homepage_url": "https://path/to/homepage",
  "host_permissions": [...],
  "import": [{"id": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"}],
  "incognito": "spanning, split, or not_allowed",
  "input_components": ...,
  "key": "publicKey",
  "minimum_chrome_version": "versionString",
  "nacl_modules": [...],
  "natively_connectable": ...,
  "oauth2": ...,
  "offline_enabled": true,
  "omnibox": {
    "keyword": "aString"
  },
  "optional_permissions": ["tabs"],
  "options_page": "options.html",
```

```
30   "options_ui": {
31     "chrome_style": true,
32     "page": "options.html"
33   },
34   "permissions": ["tabs"],
35   "platforms": ...,
36   "replacement_web_app": ...,
37   "requirements": {...},
38   "sandbox": [...],
39   "short_name": "Short Name",
40   "storage": {
41     "managed_schema": "schema.json"
42   },
43   "system_indicator": ...,
44   "tts_engine": {...},
45   "update_url": "https://path/to/updateInfo.xml",
46   "version_name": "aString",
47   "web_accessible_resources": [...]
48 }
49 ]
```

Code 2.2: Esempio file Manifest.json

I file che costituiscono l'infrastruttura di un'estensione Chrome sono analoghi a quelli che compongono un'applicazione Web, i file HTML contengono la struttura della pagina, ai file CSS è demandata la resa grafica e ai file JavaScript la dinamicità. Una volta creata l'estensione, per vederla all'opera è necessario caricarla tra le estensioni non pacchettizzate (quelle non ufficiali), nella sezione "Estensioni" del menù del browser Google Chrome. Nel Capitolo 5, interamente dedicato all'estensione realizzata per il progetto, verranno offerti maggiori dettagli e verrà mostrato il funzionamento dell'estensione di "Learn to Detox".

3. Toxicity di Tensorflow.js: un modello di rilevazione della tossicità

Nel capitolo corrente verrà offerto un quadro semplificato sia a livello tecnico che teorico, degli strumenti utilizzati per realizzare l'infrastruttura del progetto "Learn to Detox". In particolare, nella Sezione 3.1 verrà introdotto il concetto di apprendimento automatico come premessa alla presentazione di Tensorflow e alla successiva comprensione di Tensorflow.js. Mentre nella Sezione 3.2 verrà spiegato nel dettaglio il funzionamento del modello Toxicity di Tensorflow.js in un contesto di reale utilizzo, prendendo in considerazione anche la questione della spiegabilità dei sistemi di intelligenza artificiale.

3.1 Introduzione a Tensorflow

Il progetto "Learn to Detox" include due strumenti: un'applicazione Web e un'estensione per il browser Google Chrome: entrambi realizzati con la libreria **Tensorflow**, essi costituiscono un esempio dell'utilizzo di un modello di apprendimento automatico per la rilevazione di linguaggio offensivo in un testo. L'obiettivo di creare dei meccanisimi basati sull'intelligenza artificiale che comprendano le sfumature del linguaggio naturale è notoriamente ostacolato dal fatto che esso non segue un insieme di regole rigide, oltre che dall'impossibilità di conoscere il contesto e il fine della comunicazione.

Prima di entrare nel dettaglio del funzionamento di Tensorflow è opportuno delinare a grandi linee il contesto all'interno del quale è operativo.

3.1.1 Machine Learning e hate speech

Nella sua accezione più semplice, l'intelligenza artificiale si riferisce a sistemi che imitano l'intelligenza umana per eseguire certe attività e che sono in grado di migliorarsi in base alle informazioni raccolte. Il Machine Learning è una branca dell'intelligenza artificiale

che si occupa di creare sistemi che apprendono in base ai dati che utilizzano. (Géron, 2019) (*Cos'è il machine learning? — Oracle Italia*)

Gli algoritmi alimentano il Machine Learning e quelli attualmente utilizzati possono essere principalmente di due tipologie:

- Algoritmi supervisionati: l'algoritmo apprende da un set di dati già etichettato e con un output predefinito.
- Algoritmi non supervisionati: l'algoritmo utilizza i dati non etichettati per identificare nuovi modelli.

La differenza tra questi due tipi viene definita dal modo in cui ciascun algoritmo apprende i dati per fare previsioni. In sostanza, l'apprendimento automatico o Machine Learning si riferisce a un processo in base al quale un algoritmo viene addestrato su un dataset, per identificare dei modelli all'interno dell'insieme di dati. Il modello risultante potrà quindi essere applicato a nuovi dati. Un ramo avanzato del Machine Learning è il Deep Learning, un insieme di tecniche basate su reti neurali artificiali organizzate in diversi strati. Con l'intento di fornire una definizione semplificata di rete neurale, è possibile affermare che essenzialmente si tratta di un processore distribuito, basato sul modello del sistema nervoso centrale umano, generalmente composto da unità computazionali elementari dette neuroni, concepibili come nodi di una rete con determinate capacità di elaborazione e tra loro interconnessi (o connessi a cascata). (Cai et al., 2020)

In relazione all'incitamento all'odio, un algoritmo potrebbe essere addestrato su un set di dati esistente dove il linguaggio offensivo è etichettato e quindi applicato a nuovi dataset per determinare se sono presenti istanze simili di incitamento all'odio. L'idea è che attraverso l'uso dei modelli di apprendimento automatico, come l'elaborazione del linguaggio naturale, l'hate speech possa essere rilevato automaticamente per essere contrastato.

3.1.2 Machine Learning con Javascript: Tensorflow.js

TensorFlow (www.tensorflow.org) è una libreria software open source per l'apprendimento automatico creata da Google Brain nel 2015, che consente di sviluppare e addestrare

modelli di Machine Learning, semplificando la creazione di reti neurali tramite un'API chiamata Keras (keras.io). (Chollet, 2017)

Nel 2018 gli sviluppatori hanno lanciato Tensorflow.js (www.tensorflow.org/js), una libreria open source utilizzabile per definire, addestrare ed eseguire modelli di Machine Learning interamente nel browser, utilizzando Javascript e un'API di alto livello. La progettazione di TensorFlow.js è basata sulla possibilità di arricchire le funzionalità di Tensorflow grazie alle caratteristiche specifiche dell'ambiente Javascript. (Smilkov et al., 2019) Una prima importante qualità riguarda la possibilità di utilizzo in ambienti diversi: lato client e lato server grazie al framework Node.js. Ad arricchire ulteriormente la funzionalità di Tensorflow.js, è il fatto di essere stato progettato per essere un linguaggio multipiattaforma supportato da tutti i principali browser, mediante API Web standardizzate. Questi attributi offrono l'opportunità di realizzare l'esecuzione di programmi di Machine Learning interamente lato client nel browser apportando considerevoli vantaggi di seguito elencati. (Abadi et al., 2016)

Condivisione: una delle principali motivazioni dietro la nascita di TensorFlow.js è la possibilità di eseguire modelli di Machine Learning nei browser standard, senza alcuna installazione aggiuntiva. Modelli e applicazioni scritte in TensorFlow.js sono facilmente condivisibili sul Web, abbassando le barriere all'ingresso per l'apprendimento automatico. Questo è particolarmente importante per i casi a scopo educativo e per aumentare l'eterogeneità dei contributori sul campo.

Interattività. Dal punto di vista dell'apprendimento automatico, la natura interattiva dei browser e le capacità versatili delle API Web offrono la possibilità di eseguire un'ampia gamma di nuove applicazioni incentrate sull'utente che possono servire finalità didattiche e di ricerca.

Utilizzo del proprio dispositivo. Infine, l'accesso standardizzato a vari componenti dell'hardware del dispositivo come la webcam, il microfono nel browser facilitano l'integrazione tra modelli di Machine Learning e dati dei sensori.

Un risultato importante di questa integrazione è che i dati dell'utente possono rimanere sul dispositivo e salvaguare la privacy, infatti abilitando le applicazioni e impostando l'accessibilità è possibile ottenere un modello personalizzato. Ad esempio, gli utenti con disturbi del linguaggio possono utilizzare i loro telefoni per raccogliere campioni audio per addestrare un modello personalizzato.
In sintesi, TensorFlow.js ha il potenziale per ampliare notevolmente la platea di utilizzo di tecnologie di Machine Learning.

3.1.3 Cosa è possibile fare con Tensorflow.js

Principalmente sono i tre i flussi di lavoro che possono essere presi in considerazione sfruttando Tensorflow.js. È possibile importare un modello pre-addestrato esistente per l'inferenza. Disponendo di un modello TensorFlow o Keras esistente, precedentemente addestrato, è possibile convertirlo nel formato TensorFlow.js e caricarlo nel browser per l'inferenza. È consentito inoltre addestrare nuovamente un modello importato, utilizzando il trasferimento dell'apprendimento per migliorare un modello esistente addestrato, necessitando solo di una ridotta quantità di dati.

Infine è possibile creare modelli direttamente nel browser, usufruendo di un'API di alto livello, come ad esempio Keras, una libreria open source per l'apprendimento automatico che fornisce una potente e intuitiva interfaccia per creare e allenare reti neurali profonde.(Abadi et al., 2016)

3.2 Come funziona Toxicity

Per la realizzazione del progetto "Learn to Detox" è stata utilizzato un modello pre-addestrato di Tensorflow.js. Il modello è stato addestrato su un dataset contenente due milioni di commenti etichettati a seconda del tipo di tossicità ed è stato costruito sulla base di un modello universale per la codifica della frasi, l'Universal Sentence Encoder (USE). (Cer et al., 2018) Il modello codifica il testo in un incorporamento a 512 dimensioni. Questi incorporamenti possono essere utilizzati come punti di partenza per attività di elaborazione del linguaggio come ad esempio la Sentiment Analysis. L'Universal Sen-

tence Encoder utilizza l'architettura Transformer, che rappresenta una classe di modelli di Machine Learning all'avanguardia nella modellazione del linguaggio, ed è basato sull'uso di reti neurali profonde. L'Universal Sentence Encoder può essere trovato su TensorFlow Hub (*TensorFlow Hub* - www.tensorflow.org/hub), un repository di modelli di Machine Learning addestrati pronti per l'ottimizzazione e distribuibili ovunque, ma è anche disponibile come singolo modulo TensorFlow.js. (Vaswani et al., 2017)

3.2.1 L'annotazione dei dati in base ad etichette di tossicità

Conversation AI, un'iniziativa di ricerca guidata da Jigsaw e dal Google Counter-Abuse Technology Team ha gestito le regole per l'annotazione del dataset, composto da milioni di commenti. Agli annotatori è stato chiesto dapprima di selezionare il livello di tossicità presente nel commento, utilizzando le seguenti definizioni come guida:

- **Molto tossico**: un commento colmo di odio, aggressivo, irrispettoso o che è probabile che induca un utente a lasciare una discussione o a rinunciare a condividere il proprio punto di vista.
- **Tossico**: un commento maleducato, irrispettoso, irragionevole o in qualche modo in grado di indurre un utente a lasciare una discussione o a rinunciare a condividere il proprio punto di vista.
- **Leggermente tossico**: un commento che ospita parole sgradevoli ma non completamente offensive.
- **Non tossico**: un commento totalmente privo di linguaggio offensivo.

In seguito è stato chiesto agli annotatori di determinare se un commento contenesse volgarità/oscenità, contenuti sessualmente espliciti, attacchi basati sull'identità, linguaggio minaccioso e insulti, utilizzando le seguenti definizioni come guida:

- **Oscenità**: in caso di parolacce o in generale linguaggio osceno o blasfemo.

- **Sessualmente esplicito**: in caso di riferimenti ad atti sessuali o a parti del corpo in modo sessuale o altri contenuti osceni.
- **Attacco basato sull'identità**: un commento negativo, discriminatorio o offensivo su una persona o un gruppo di persone, basato su criteri che includono (ma non solo) razza o etnia, religione, genere, nazionalità o cittadinanza, disabilità, età o orientamento sessuale.
- **Insulto**: commento offensivo, provocatorio o negativo nei confronti di una persona o di un gruppo di persone.
- **Minaccia**: descrive un desiderio o intenzione di provocare dolore, lesioni o violenza contro un individuo o un gruppo. (*conversationai.github.io/toxicity*)

3.2.2 Limitazioni del sistema in ambito reale

Risulta opportuno tenere presente che il modello Toxicity di Tensorflow.js è sperimentale e in quanto tale è soggetto ad una precisione ridotta e ad una distorsione non intenzionale. Ci sono inoltre delle limitazioni riguardanti la possibile restituzione di una classificazione errata, commenti erroneamente classificati come tossici o viceversa, in presenza di elementi che rendono più complessa l'analisi. In primo luogo la tossicità può essere espressa anche senza utilizzare un linguaggio esplicitamente offensivo.

In altre occasioni il significato tossico si rivela solo per mezzo della conoscenza del contesto e della comprensione dell'intera frase. Prendendo la frase: "lei somiglia ad un cavallo", è possibile offrire un esempio concreto di una potenziale classificazione errata. Essa consente di osservare che un modello per rilevare la tossicità della proposizione, necessiterebbe di comprendere che il pronome "lei" in "lei somiglia ad un cavallo" fa sì che l'espressione sia considerabile come un insulto perché si rivolge ad una persona, nonostante la parola "cavallo" non sia una parola comunemente considerabile come offensiva. D'altra parte possono verificarsi anche falsi positivi che contengono parole tossiche, sebbene nel contesto non siano tali. Ad esempio, un utente che pubblica un commento autoreferenziale come "Mi sento un tale idiota adesso." può innescanare una classificazione errata, a causa della parola "idiota", generalmente considerata come insulto. Prendere

in considerazione una frase completa e coglierne il significato resta ancora una sfida per gli approcci di Deep Learning, considerato che ci sono citazioni, riferimenti, metafore e confronti che possono indurre il modello in errore.

Sarcasmo, ironia e domande retoriche hanno in comune il fatto che il significato figurativo che assume la frase è diverso dal suo significato letterale. Questa situazione può causare errori nella classificazione a causa della mancanza di comprensione del contesto. Anche parole intenzionalmente offuscate, errori di battitura, slang, abbreviazioni e neologismi rappresentano una sfida particolare nei dataset di commenti tossici. Se non ci sono abbastanza campioni contenenti queste espressioni, infatti, le rappresentazioni apprese potrebbero non spiegare il vero significato di una parola o di una espressione.

3.2.3 Sviluppi futuri per migliorare la classificazione

Una campo di ricerca in crescita per migliorare la classificazione dei commenti contenenti linguaggio offensivo è quello che si occupa di creare modelli basati sulla rilevazione del loro esatto opposto: contenuti di alta qualità. Una possibile applicazione di questa soluzione consisterebbe nel permettere di scegliere e mantenere un determinato standard di qualità di conversazione alle piattaforme. Una direzione diversa è quella invece di migliorare la classificazione tenendo conto del contesto di un commento. Invece di utilizzare un singolo commento come input quindi, prendere in considerazione la possibilità di utilizzare la discussione completa e altri elementi che compongono il contesto, come l'articolo del quotidiano o la cronologia dell'utente. (Risch e Krestel, 2020)

3.2.4 Explainable AI

Una questione di grande importanza, nell'eventualità in cui alla rilevazione automatica di linguaggio offensivo segua l'immediata rimozione del commento, è la spiegabilità delle decisioni di classificazione. Questo vale ovviamente anche per altri modelli di Deep Learning, ma nel contesto della moderazione dei commenti assume un ruolo fondamentale nel processo. L'Explainable AI, intelligenza artificiale spiegabile, è una disciplina emersa recentemente dalla necessità di rendere meno opaco il funzionamento degli algoritmi, ed

è sinteticamente definibile come un insieme di strumenti e tecniche utilizzate per rendere sempre più trasparente e facile da capire il funzionamento dei sistemi di intelligenza artificiale, la loro logica interna. Oltre a fornire agli utenti informazioni sull'accuratezza della previsioni e su altri aspetti delle prestazioni, offrire una spiegazione efficace del comportamento del sistema di intelligenza artificiale può aumentare la fiducia dell'utente nei confronti del sistema. Per esempio se un algoritmo consiglia all'utente un prodotto da acquistare o una connessione da aggiungere alla propria rete professionale o social, è probabile che una spiegazione della raccomandazione, renda le informazioni più utili per l'utente e abbia una maggiore influenza sulle azioni dell'utente. Spiegare la cancellazione automatica di un commento assume importanza cruciale nel contesto della libertà di espressione. Trovare un equilibrio tra la censura e la protezione di individui e gruppi sul Web è l'obiettivo per una gestione ottimale dei flussi comunicativi online. (Liao et al., 2020)

4. Il progetto “Learn to Detox”

Nel capitolo corrente verrà illustrato nel dettaglio il progetto realizzato per questa tesi. Delineata la struttura e lo scopo, verranno approfonditi i dettagli sul codice. “Learn to Detox” comprende un’applicazione Web che analizzeremo in questo capitolo ed un’estensione per il browser Google Chrome, che verrà approfondita nel Capitolo 5. Nello specifico, nella Sezione 4.1 verrà illustrato il codice utile ad implementare il modello con l’intento di rendere più chiara anche la comprensione dei risultati della classificazione. La Sezione 4.2 ospiterà invece un focus su ogni pagina che compone l’applicazione Web “Learn To Detox”, analizzando alcuni frammenti di codice.

4.1 Il codice per implementare il modello

Per poter usufruire del modello Toxicity.js, illustrato nel Capitolo 3 è necessario installarlo tramite NPM (Node Package Manager), il gestore di pacchetti ufficiale che viene installato tramite Node.js (introdotto nel Capitolo 2), oppure includerlo in bundle tramite un tag di script nel file HTML. Nel caso di “Learn to Detox” è stata seguita la modalità di inclusione, con il seguente codice:

```
1 <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/
     dist/tf.min.js"></script>
2 <script src="https://cdn.jsdelivr.net/npm/@tensorflow-models/
     toxicity"></script>
```

Code 4.1: Script utilizzato per inculudere Toxicity

Per iniziare a classificare il testo, è necessario caricare prima il modello specificando alcuni parametri:

Threshold è il parametro che rappresenta il livello di confidenza minimo al di sopra del quale una previsione di testo tossico è considerata valida. L’impostazione del parametro threshold a un valore più alto aumenta le probabilità che le previsioni vengano restituite “null”, perché scendono al di sotto della soglia.

Il parametro **labelsToInclude** è invece un array di stringhe contenente le sette etichette di tossicità, che è possibile selezionare in base alle preferenze. Le etichette selezionabili all'interno di labelsToInclude sono: toxicity, severe toxicity, identity attack, insult, threat, sexual explicit, obscene, è possibile scegliere di mostrarle tutte o solo alcune. Per impostazione predefinita, il modello restituirà le previsioni per tutte e sette le etichette. L'API chiamata per prevedere la tossicità è model.classify() e restituisce una Promise. Nel frammento di codice sottostante (Code 4.2), viene riportato un esempio in cui viene scelto un valore da assegnare alla costante threshold, tre delle sette etichette a disposizione per la classificazione e il caricamento del modello seguito dalla classificazione della frase "you are an idiot".

```
1 // The minimum prediction confidence.
2 const threshold = 0.9;
3
4 // Which toxicity labels to return.
5 const labelsToInclude = ['identity_attack', 'insult , 'threat'];
6
7 toxicity.load(threshold, labelsToInclude).then(model => {
8     model.classify(['you are an idiot']).then(predictions => {...});
9 });
```

Code 4.2: Codice per caricare il modello

4.1.1 I risultati della classificazione

Le previsioni sono un array di oggetti ("results") e ne viene restituito uno per ogni etichetta di tossicità. Questi oggetti contengono a loro volta un array "probabilities" che contiene le probabilità a priori (la probabilità incondizionata che viene assegnata prima che venga presa in considerazione qualsiasi evidenza rilevante) e le probabilità a posteriori (la probabilità condizionata che è assegnata dopo che si è tenuto conto dell'informazione rilevante) per ogni frase data in input.

La previsione finale può risultare:

- **true** se la probabilità condizionata supera la soglia di confidenza;

- **false** se la probabilità condizionata non supera la soglia di confidenza;
- **null** se nessuna delle due probabilità supera la soglia.

È possibile osservare un esempio della forma in cui vengono restituiti i risultati nel frammento di codice sottostante (Code 4.3), il formato è JSON (JavaScript Object Notation).

```
[
    {
      "label": "identity_attack",
      "results": [{
        "probabilities": [0.965, 0.034],
        "match": false
      }]
    },
    {
      "label": "insult",
      "results": [{
        "probabilities": [0.081, 0.919],
        "match": true
      }]
    },
    {
      "label": "threat",
      "results": [{
        "probabilities": [0.364, 0.635],
        "match": null
      }]
    },
    // ...
]
```

Code 4.3: Risultati classificazione

4.2 Learn to Detox: una WebApp interattiva per rilevare la tossicità

La WebApp realizzata "Learn to Detox" è intesa come un esempio della possibilità di utilizzare un modello di analisi del testo interamente lato client, in tempo reale nel browser. Nell'immagine seguente (Figura 4.1) è possibile visualizzare il contenuto e la struttura della cartella dell'applicazione Web "Learn to Detox".

```
LearnToDetox/
├─ home.html
├─ speech.html
├─ toxicity.html
├─ assets/
│   ├─ microphone.svg
│   ├─ microphone-recording.svg
├─ css/
│   ├─ home.css
│   ├─ speech.css
│   ├─ toxicity.css
│   ├─ examples.css
├─ script/
│   ├─ speech.js
│   ├─ toxicity.js
│   ├─ examples.js
```

Figura 4.1: Struttura cartella LearnToDetox

L'applicazione ha lo scopo di rilevare linguaggio offensivo tramite il modello Toxicity di Tensorflow.js in due modalità. La prima classifica una parola o una frase digitata in un'area di testo dall'utente. La seconda mediante l'utilizzo del microfono del dispositivo in uso classifica l'input vocale dell'utente, trascritto in modalità Speech to Text, servendosi della Web Speech (introdotta nel Capitolo 2). (*Web Speech API - Web APIs — MDN* - developer.mozilla.org)

Allo scopo di predisporre uno spettro più ampio dei potenziali utilizzi della WebApp, è stata inclusa una pagina contenente una sintetica guida per chiarire l'utilizzo del modello all'interno del progetto e le funzionalità dell'applicazione. È stata inoltre predisposta la possibilità di interagire con il modello prima di classificare l'input digitato dall'utente. Nell'ultima pagina, "Examples" sarà possibile invece osservare alcuni esempi di com-

menti raccolti su Twitter ed analizzati dal modello, e quindi confrontare i risultati che mostreranno la percentuale di tossicità per ogni etichetta.

4.2.1 La pagina Home e la pagina Learn

La pagina Home della WebApp contiene oltre al titolo e sottotitolo, un menù creato seguendo il principio del Design Responsivo, la tecnica di Web design per la realizzazione di siti in grado di adattarsi graficamente in modo automatico al dispositivo coi quali vengono visualizzati, riducendo al minimo la necessità dell'utente di ridimensionare e scorrere i contenuti. Per seguire questo approccio è stato utilizzato il framework Bootstrap, introdotto nel Capitolo 2. Nell'immagine sottostante (Figura 4.2) è possibile osservare lo screenshot della pagina Home dell'applicazione.

Figura 4.2: Pagina Home

Il frammento di codice sottostante (Code 4.4) rappresenta il metodo con cui è stato incluso Bootstrap (sia il CSS che lo script) all'interno della pagina Home.

```
1  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/
     bootstrap.min.css" rel="stylesheet" integrity="sha384-
     F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU
     " crossorigin="anonymous">
2  //...
3  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/
     bootstrap.bundle.min.js" integrity="sha384-/bQdsTh/da6pkI1MST/
```

```
    rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2THRZ" crossorigin="
    anonymous"></script>
});
```

Code 4.4: Codice per includere Bootstrap nella pagina Home

Nell'immagine sottostante (Figura 4.3) è possibile notare il cambiamento della visualizzazione del menù nel caso di schermi di dimensioni ridotte rispetto a quelle di un computer, come un tablet per esempio. È possibile notare che il menù non è più contenuto in forma estesa nella parte alta della pagina, ma si apre attraverso l'interazione con l'icona in alto a destra.

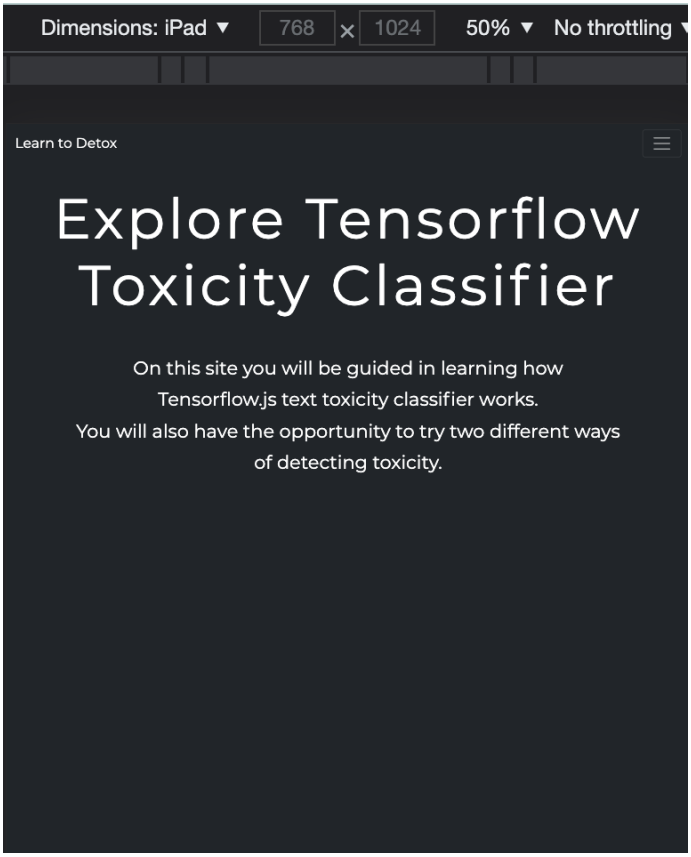

Figura 4.3: Pagina Home Responsive Menù

Nella pagina intitolata "Learn", il cui screenshot è visibile nell'immagine sottostante (Figura 4.4) è presente una guida alla comprensione del modello, al fine di preparare l'utente ad utilizzare l'applicazione Web.

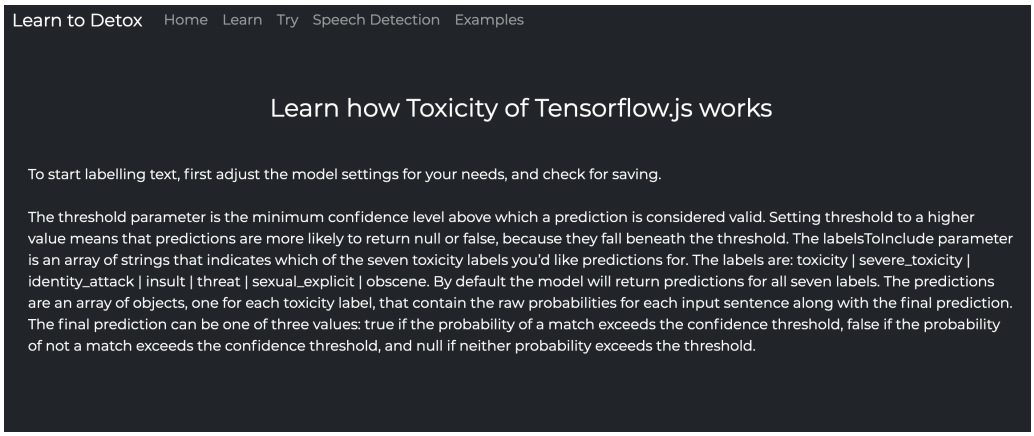

Figura 4.4: Pagina Learn

4.2.2 Hate Speech rilevato con il riconoscimento vocale

Per iniziare a classificare contenuti è necessario visitare la pagina speech, nella quale dopo aver concesso l'autorizzazione al browser per utilizzare il microfono, sarà possibile classificare una frase pronunciata. È infatti sufficiente cliccare sull'icona del microfono, pronunciare la frase e cliccare di nuovo per determinare la fine della frase da dare in input al modello. Nell'immagine seguente (Figura 4.5) è possibile osservare uno screenshot della schermata al momento della restituzione dei risultati della classificazione effettuata sulla trascrizione della frase"You are idiot".

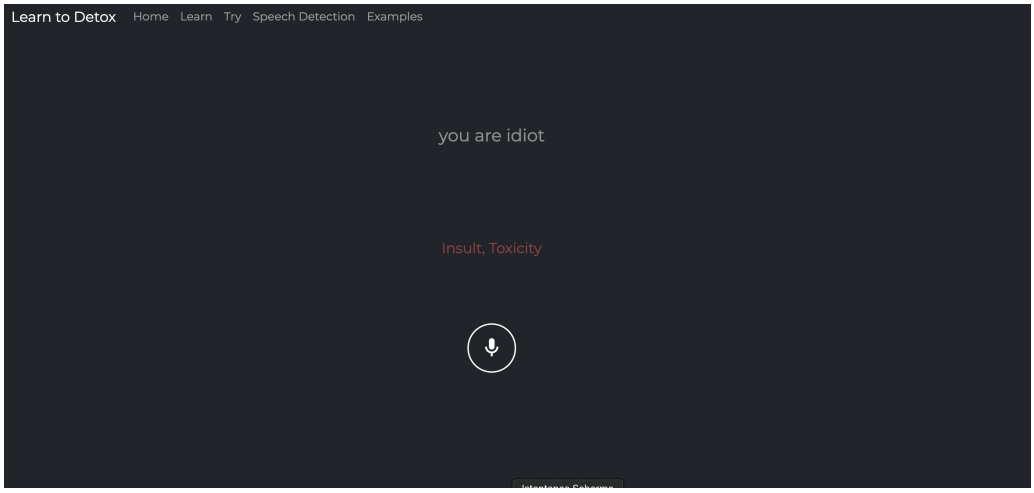

Figura 4.5: Screenshot pagina speech recognition

Una volta trascritto il testo pronunciato, viene caricato il modello e poi classificato, in seguito compariranno le etichette per cui è risultata positiva l'analisi di tossicità. Le modalità tecniche tramite le quali si svolgono queste operazioni sono quelle descritte nel Capitolo 2.

Nel frammento di codice sottostante (Code 4.5), è possibile osservare che è stata utilizzata una API per il riconoscimento vocale.

```
1  const recognition = new webkitSpeechRecognition()
2  recognition.lang = 'en-US'
3  recognition.onresult = function(event) {
4      let final_transcript = ''
5
6      for (let i = event.resultIndex; i < event.results.length; ++i) {
7          if (event.results[i].isFinal) {
8              final_transcript += event.results[i][0].transcript
9          }
10     }
```

Code 4.5: Frammento del codice che contiene la Web Speech API

La Web Speech Api utilizzata, come descritto nel Capitolo 2, implica la ricezione del parlato tramite il microfono del dispositivo, che viene quindi analizzato da un servizio di riconoscimento vocale. Quando una parola o una frase viene riconosciuta correttamente, viene restituita e visualizzata come stringa di testo. Risulta opportuno specificare che sul browser Chrome, l'utilizzo del riconoscimento vocale su una pagina Web comporta un meccanismo di riconoscimento basato su un server.

L'audio viene infatti inviato a un servizio Web per l'elaborazione del riconoscimento, pertanto non funziona offline.

4.2.3 Classificare il testo personalizzando i parametri

La pagina intitolata Try (nella cartella illustrata in Figura 4.1 è il file toxicity.html) offre la possibilità di rilevare linguaggio offensivo digitando testo in un'area dedicata e di per-

sonalizzare i parametri introdotti nel Capitolo 3 del modello Toxicity.js. Nell'immagine sottostante (Figura 4.6) è possibile visualizzare uno screenshot della pagina Try.

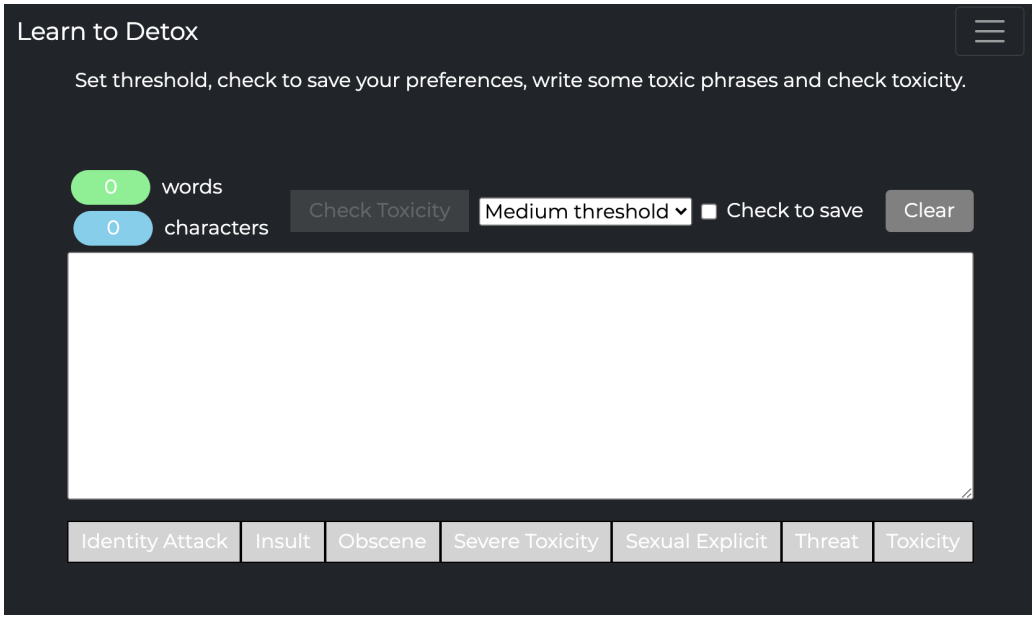

Figura 4.6: Interfaccia pagina Try

Viene mostrata inizialmente una sintetica guida all'utilizzo dell'interfaccia sottostante che ospita due contatori, uno per le parole ed uno per i caratteri, un menù di selezione per il valore threshold, una checkbox per salvare i dati impostati e caricare il modello e un pulsante "Clear" che aggiorna la pagina e permette di ricominciare a classificare con una nuova personalizzazione. Per iniziare a rilevare la tossicità è infatti necessario personalizzare il valore threshold (descritto nel Capitolo 3), ovvero il livello di confidenza minimo al di sopra del quale una previsione di testo tossico è considerata valida.

I valori selezionabili sono:

- **Low threshold**: il valore threshold viene impostato a 0.2.
- **Medium treshold**: il valore threshold viene impostato a 0.5 ed è il valore di default.
- **High treshold**: il valore threshold viene impostato a 0.8.

Nel frammento di codice sottostante (Code 4.6) è possibile notare il modo in cui è stata implementata nel file toxicity.html la possibilità di scegliere un valore threshold.

```
<select id="soglia">
    <option value="0.2">Low threshold</option>
    <option value="0.5" selected="selected">Medium threshold</option
   >
    <option value="0.8">High treshold</option>
</select>
```

Code 4.6: Codice nella pagina Try per impostare il valore treshold

Nel file toxicity.js allo scopo di caricare il modello con il valore threshold personalizzato dall'utente, nella funzione asincrona enableToxicityCheck(), viene rilevato il valore selezionato tramite il metodo Document.getElementById() e assegnato alla variabile "soglia", quindi il parametro threshold assume il valore di "soglia". Nel frammento di codice riportato (Code 4.7) sono visibili i passsaggi effettuati.

```
async function enableToxicityCheck(){

  toxicBtn.toggleAttribute("disabled");

  clearToxicityLabels();

  if ( isToxicLoaded ){
    isToxicEnabled = !isToxicEnabled;
    return;
  }



  const labelsToInclude = ['identity_attack', 'insult', 'threat', '
   toxicity', 'severe_toxicity', 'sexual_explicit', 'obscene' ];
  var soglia = document.getElementById("soglia");
  let threshold = soglia.value;
  loadingEl.style.opacity = 1;

  model = await toxicity.load( threshold, labelsToInclude);

  console.log("Toxicity Model Loaded...");
```

```
21    console.log(threshold);
22    loadingEl.style.opacity = 0;
23    isToxicEnabled = true;
24    isToxicLoaded = true;
25
26  }
```

Code 4.7: Codice nel file toxicity.js per caricare il modello con parametro threshold selezionata dall'utente

Naturalmente il valore threshold influenza i risultati e la rilevazione della tossicità. Nelle immagini seguenti (Figura 4.7 e Figura 4.8) è possibile osservare un confronto espresso nei due screenshot che riportano i risultati della classificazione effettuata sulla stessa frase, ma caricando il modello con diversi valori assegnati al parametro threshold.

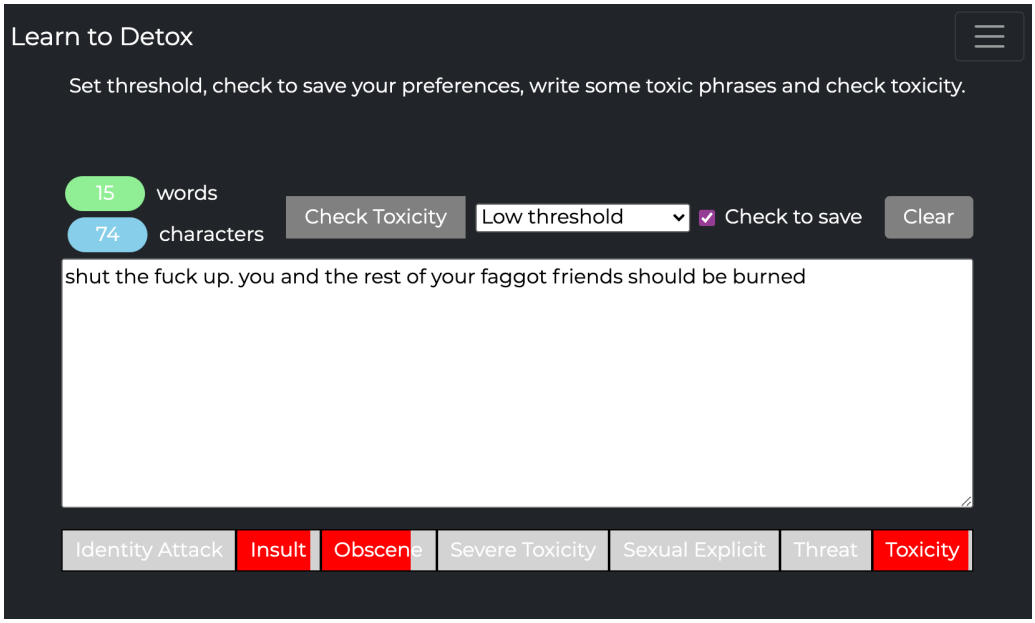

Figura 4.7: Risultati analisi tossicità con valore soglia a 0.2

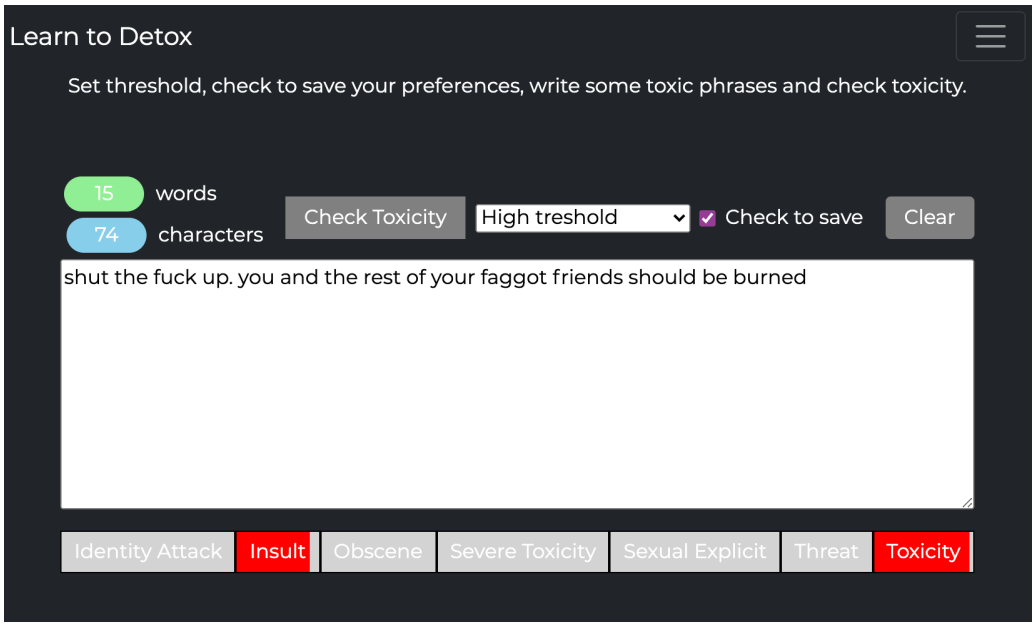

Figura 4.8: Risultati analisi tossicità con valore soglia a 0.8

Nel primo esempio riportato (Figura 4.7), la frase presa dal dataset di Kaggle (*Toxic Comment Multilabel Classification — Kaggle*), viene analizzata con un valore treshold impostato a 0.2 e restituisce come output un'altissima corrispondenza con l'etichetta "Insult", un'alta corrispondenza con l'etichetta "Obscenity", quindi un valore altissimo di tossicità, rappresentato dall'etichetta "Toxicity".

Per contro, nel secondo esempio riportato (Figura 4.8), la stessa frase analizzata con un valore treshold più alto, impostato a 0.8, restituisce come output una corrispondenza molto alta con l'etichetta "Insult" e un valore altissimo di tossicità, assente invece la corrispondenza con l'etichetta "Obscenity". Questo al momento che al crescere del valore che rappresenta il livello di confidenza minimo al di sopra del quale una previsione è considerata valida, diminuisce la probabilità che emergano corrispondenze con etichette di tossicità aventi una bassa probabilità condizionata. In una prospettiva di utilizzo in ambito reale la possibilità di impostare diversi valori come livello di confidenza minimo, potrebbe rappresentare un diverso grado di severità con la quale determinare la tossicità di un testo anche in base allo scopo della classificazione.

Come è possibile notare dagli screenshots (Figura 4.7 e Figura 4.8) il compartimento dedicato ad ogni etichetta diviene colorato di rosso in modo proporzionale al livello

di tossicità riconosciuto, allo scopo di evidenziare il risultato tramite la resa grafica. Il frammento di codice seguente (Code 4.8), rappresenta il modo in cui è stata implementata questa feature nel file toxicity.js, all'interno della funzione asincrona checkToxicity () che si occupa di lanciare la classificazione sull'input digitato dall'utente e di restituire i risultati.

```
1  async function checkToxicity(){
2    if ( !isToxicEnabled || !isToxicLoaded ){ return; }
3    toxicBtn.classList.add("checking");
4    const predictions = await model.classify([textarea.value]);
5    toxicBtn.classList.remove("checking");
6    predictions.forEach( prediction => {
7      if ( prediction.results[0].match ){
8        const probability = Math.round(prediction.results[0].
     probabilities[1] * 100 );
9        document.querySelector(`.${prediction.label}`).style.
     background = `linear-gradient(90deg, red ${probability}%,
     lightgray ${100-probability}%)`;
10     }
11   })
12 }
```

Code 4.8: Codice nel file toxicity.js per colorare in modo proporzionale i compartimenti delle etichette

Il metodo querySelector() restituisce il primo elemento all'interno del documento che corrisponde al selettore specificato o al gruppo di selettori. Se non vengono trovate corrispondenze, viene restituito null. All'interno del file toxicity.html è stato un creato un contenitore per ogni etichetta di tossicità, poi è stata assegnata una classe denominata con una stringa identica a quella delle label (l'array che contiene le sette etichette) ai contenitori dei compartimenti che mostrano i risultati, nello script viene creata una regola di stile specifica che colora il commpartimento in base alla percentuale di tossicità della label corrispondente.

4.2.4 Alcuni esempi della classificazione

Nell'ultima pagina dell'applicazione Web, "Examples", vengono mostrate le classificazioni di alcune frasi significative prese da Twitter, in questo contesto le corrispondenze sono espresse in valori percentuali per fornire un ulteriore esempio di restituzione dei risultati. Il parametro threshold è stato impostato 0.5 costante per agevolare la fluidità nella classificazione di frasi diverse nello stesso momento. Per quanto riguarda la rappresentazione grafica è stata scelta la visualizzazione delle percentuali di tossicità arrotondate tramite la funzione Math.round(), che restituisce il valore di un numero arrotondato all'intero più vicino. Sono state scelte per fornire un'esemplificazione quattro frasi di cui una priva di linguaggio offensivo e le restanti tre aventi un diverso grado di tossicità. L'immagine seguente (Figura 4.9) rappresenta lo screenshot della pagina Examples.

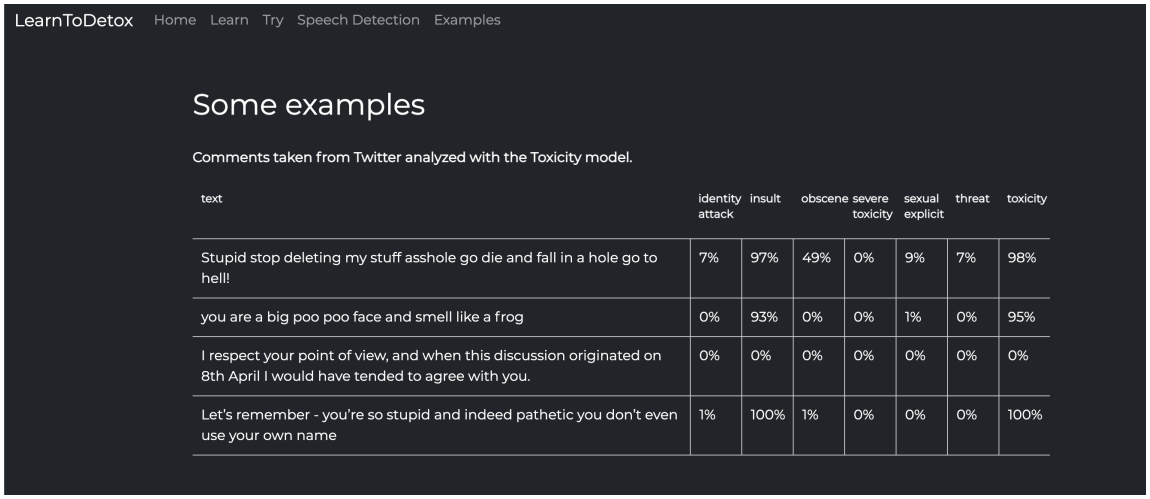

Figura 4.9: Screenshot della pagina "Examples" con classificazione

Com'è possibile notare nello screenshot (Figura 4.9), nella frase "I respect your point of view, and when this discussion originated on 8th April I would have tended to agree with you." (tradotta: "Rispetto il tuo punto di vista e quando questa discussione è iniziata, l'8 aprile, tendevo a essere d'accordo con te".) non è stata rilevata nessuna percentuale di tossicità. Mentre nelle altre, in diverse percentuali in base all'etichetta sono state rilevate forme di linguaggio offensivo.

4.2.5 Scopi ed eventuali sviluppi

Lo scopo di questa applicazione è quello offrire modalità diverse per rilevare varie forme linguaggio offensivo all'interno di contenuti testuali, per favorire una classificazione precisa e idonea a porsi come base sulla quale effettuare delle scelte riguardanti politiche di selezione dei contenuti sulle piattaforme. In linea con l'ampia accessibilità di Tensorflow, che propone modelli Open Source, l'approccio è stato rivolto anche a rendere chiare e comprensibili le modalità in cui il modello è stato usato, allo scopo di favorirne e incoraggiarne la manipolazione e il futuro miglioramento.

Con il riconoscimento vocale è stato preso in considerazione anche l'aspetto relativo al recente ingresso delle interfacce vocali per i sistemi di dettatura, attivi nella ricerca vocale ma anche nella messaggistica, settore che ha compiuto importanti passi negli ultimi anni. (De Marsico e Mattei, 2021)

5. L'estensione Chrome del progetto "Learn to Detox"

A completare l'intero progetto realizzato per questa tesi si unisce all'applicazione Web un'estensione Chrome che si basa sulla stessa tecnologia fondamentale: il modello Toxicity di Tensorflow.js. Il Capitolo corrente nella Sezione 1 percorrerà sinteticamente le fasi di realizzazione dell'estensione per il browser Google Chrome, mentre la Sezione 2 si focalizzerà sulle funzionalità nello specifico dell'estensione "Learn to Detox" analizzondone gli scopi.

Nell'immagine che segue (Figura 5.1) viene fornito il dettaglio del contenuto della cartella dell'estensione Chrome del progetto.

```
detoxExtension/
├─ index.html/
├─ index.js/
├─ manifest.json
├─ toxicity.js
├─ style.css
├─ tfjs.js
├─ 128.png
```

Figura 5.1: Struttura cartella estensione Chrome

Per realizzare l'estensione è stata seguita la guida della documentazione ufficiale di Google che offre un percorso guidato in particolare nella costruzione del file manifest e nella pubblicazione dell'estensione. (*Creare e pubblicare estensioni e app di Chrome personalizzate - Guida di Google Chrome Enterprise* - developer.chrome.com)

Il codice seguente (Code 5.1) rappresenta il file manifest.json dell'estensione Chrome del progetto "Learn to Detox", il file manifest è stato introdotto nel Capitolo 2 ed è un file di configurazione che fornisce i dettagli del programmma.

```
{
  "name": "TensorFlow Toxity Recognition Chrome Extension",
  "version": "0.1",
  "icons": {"128": "128.png" },
  "description": "TensorFlow chrome extension for analyzing text
   toxicity",
  "author": "Agnese Marchionneschi",
  "browser_action": {
    "default_popup": "index.html"
  },
  "permissions": [
    "tabs","<all_urls>",
    "activeTab"
  ],
  "options_page": "index.html",
  "content_security_policy": "script-src 'self' 'unsafe-eval';
   object-src 'self'",
  "content_scripts": [
    {
      "matches": [
        "http://*/*",
        "https://*/*"
      ],
      "js": ["tfjs.js", "toxicity.js", "index.js"],
      "run_at": "document_end"
    }
  ],
  "manifest_version": 2
}
```

Code 5.1: File di configurazione manifest.json dell'estensione Learn to Detox

In modo analogo alla realizzazione dell'applicazione Web del progetto, l'estensione comprende oltre al file manifest, un file index.html nel quale risiede la struttura della pagina Web, il file per la resa grafica è style.css, mentre il file che contiene lo script

necessario al funzionamento dinamico per la classificazione della tossicità è index.js, i file tfjs.js e toxicity.js contengono gli script di Toxicity di Tensorflow.

Di seguito è riportato il codice del file index.html (Code 5.2)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <link href="style.css" rel="stylesheet">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com"
   crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Roboto&
   display=swap" rel="stylesheet">
    <meta charset="UTF-8">
    <title>toxicity</title>
    <script src="tfjs.js"></script>
    <script src="toxicity.js"></script>
    <script src="index.js"></script>
</head>
<body>
<h1>Toxicity Analysis</h1>
<div id="text"></div>
</body>
</html>
```

Code 5.2: Codice index.html

Il file index.html dell'estensione di "Learn to Detox" contiente principalmente soltanto un contenitore, che viene poi riempito grazie allo script contenuto in index.js con i risultati della classificazione.

Per quanto riguarda il codice JavaScript (file index.js), esso è stato utilizzato in questo contesto per interagire con il modello Toxicity.js e restuire i risultati nel popup che si apre selezionando testo e cliccando sull'icona dell'estensione (nella Sezione 2 sarà presente uno screenshot che riporta un esempio concreto di utilizzo). Di seguito è riportato il codice contenuto in index.js (Code 5.3).

```
document.addEventListener('DOMContentLoaded', (event) => {
    const threshold = 0.3;
    const text = document.getElementById('text');
    let string = "";
    chrome.tabs.executeScript( {
        code: "window.getSelection().toString();"
    }, (selection) => {
        toxicity.load(threshold).then(model => {
            model.classify(selection).then(predictions => {
                predictions.forEach(prediction =>{
                    var res = '';
                    var strin = '';
                    var a = '&#129324;';
                    var b = '&#128513;';
                    strin += prediction.label.replace('_',' ') + "
   &#8594;";
                    string += strin.charAt(0).toUpperCase() + strin.
   slice(1);
                        if(prediction.results[0].match == true ||
   null){
                            res = a;
                        } else{
                            res = b;
                        }
                    string += (Math.round(prediction.results[0].
   probabilities[1]* 100))+ "% " + res + "<br>";
                    document.getElementById("text").innerHTML =
   string;
                })
            });
        });
    });
})
```

Code 5.3: Codice index.js

L'input testuale da classificare viene estrapolato tramite la selezione di una qualunque porzione di testo all'interno di una pagina Web aperta nel browser Google Chrome. La selezione viene catturata tramite il metodo Window.getSelection(), che restituisce un oggetto che rappresenta l'intervallo di testo selezionato dall'utente. Quindi viene trasformata in stringa grazie al metodo toString() che restituisce una stringa che rappresenta l'oggetto. Una volta ottenuta la stringa essa viene classificata dal modello Toxicity.js che restituisce i risultati nella forma di una lista di etichette di tossicità, seguite dalla percentuale rilevata nella porzione di testo.

5.1 Rilevare la tossicità nelle pagine Web

L'estensione Chrome del progetto "Learn to Detox" ha lo scopo di analizzare la tossicità su tutte le pagine Web. A differenza dell'applicazione Web descritta nel capitolo 4, l'estensione consente di analizzare contenuto testuale senza doverlo scrivere in un'area di testo ma semplicemente selezionandolo.

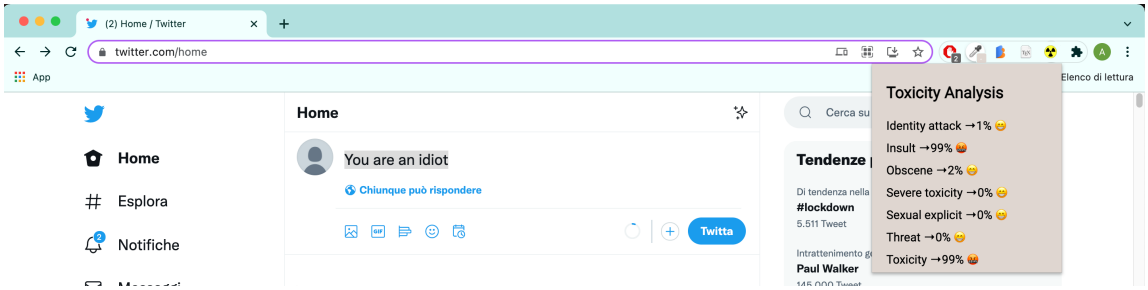

Figura 5.2: Esempio classificazione all'interno di Twitter

Nell'esempio riportato (Figura 5.2) l'estensione è stata utilizzata sul Social Network Twitter, per lanciare la classificazione è sufficiente selezionare la porzione di testo d'interesse e cliccare sull'icona dell'estensione. Compariranno le etichette della tossicità affiancate dalla percentuale di linguaggio offensivo rilevata e da un emoticon sorridente se la probabilità condizionata non supera la soglia di confidenza e un'emoticon arrabbiata se al contrario la probabilità condizionata supera la soglia.

```
1 string += (Math.round(prediction.results[0].probabilities[1]* 100))
```

Code 5.4: Frammento di codice con calcolo della probabilità in percentuale

Allo scopo di trasformare il valore della probabilità condizionata in percentuale è stata utilizzata la funzione Math.round() che arrotonda un valore all'intero più vicino o al numero specificato (Code 5.4).

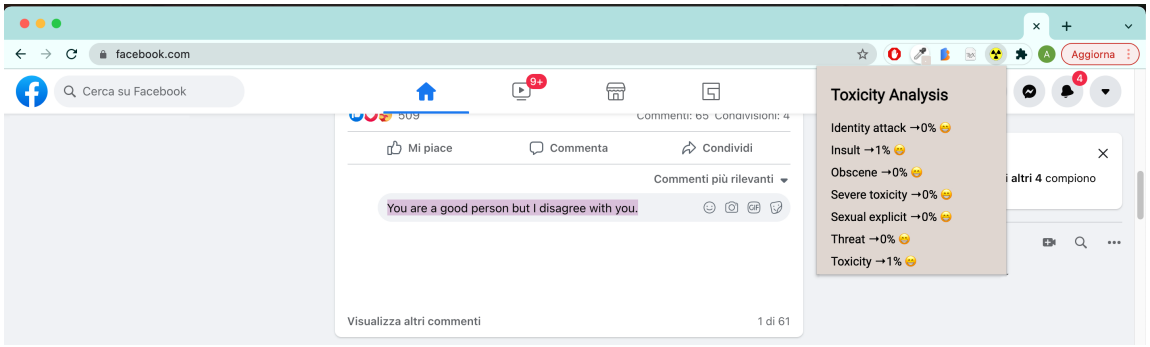

Figura 5.3: Esempio classificazione all'interno di Facebook

Nell'esempio riportato in figura 5.3, all'interno di un commento sul Social Network Facebook l'estensione ha rilevato correttamente l'assenza pressochè totale di linguaggio offensivo all'interno della frase "You are a good person but I disagree with you" ("Sei un brava persona, ma non sono d'accordo con te").

5.2 Possibili sviluppi dell'estensione

In una prospettiva di reale utilizzo dell'estensione, la rilevazione di linguaggio offensivo su qualsiasi pagina Web potrebbe unirsi agli strumenti già esistenti facenti parte del parental control. Sono già state implementate delle forme di controllo sui tempi di utilizzo di internet nel browser Chrome e forme di selezione dei risultati di ricerca in base ai contenuti.

Tuttavia il problema potrebbe porsi anche su siti e applicazioni Web che apparentemente non risultano ospitare contenuti sensibili che li rendono inadatti ai minori. Le forme di odio online possono nascere e propagarsi ovunque vi sia una forma di comunicazione,

perciò fare uso di Social Network e piattaforme simili, espone automaticamente l'utente all'eventualità di leggere, subire o perfino perpetrare hate speech.

L'estensione Chrome per la classificazione di linguaggio offensivo, potrebbe facilmente adattarsi in futuro allo scopo di prevenire su qualunque pagina Web l'esposizione all'eventuale presenza di hate speech, ma potrebbe anche scoraggiare l'utente che si trova sul punto di pubblicare contenuti tossici, avvertendolo in tempo reale della violazione delle linea guida della community.

Conclusioni

Nel corso della realizzazione del progetto sono state individuate le problematiche relative alla diffusione di hate speech online, analizzando lo stato attuale delle politiche esistenti e rilevando la mancanza di un piano strategico condiviso per contrastare la diffusione di incitamento all'odio (Capitolo 1).

Learn to Detox è un'applicazione Web corredata da un'estensione Chrome basata su una tecnologia di rilevamento automatico della tossicità, che si propone come soluzione sperimentale alla problematica principale dalla quale partire per sviluppare un piano strategico per contrastare l'hate speech: rilevare la tossicità in flussi comunicativi intensi come quelli dei Social Network.

Da un punto di vista tecnico, sono stati utilizzati diversi linguaggi per realizzare il progetto: HTML per costruire le pagine Web, CSS per la resa grafica delle pagine, JavaScript per la diminacità e Node.js per usufruire del modello di Tensorflow.js, il cui utilizzo è stato descritto nel dettaglio nel Capitolo 2. Il modello Toxicity è un modello di Machine Learning che classifica contenuto testuale in base a diverse etichette di tossicità (Capitolo 3), è stato usato per costruire un'applicazione Web che ha come obiettivo principale la classificazione di testo tossico, rilevando diverse tipologie di tossicità. Per raggiungere questo risultato è stato seguito un approccio che tenesse in considerazione anche l'importanza della comprensione del funzionamento del modello, cercando di rendere più chiara e trasparente la restituzione dei risultati. L'importanza della chiarezza in una prospettiva di utilizzo del modello è dovuta senza dubbio anche alla necessità di poter motivare l'eventuale rimozione di un commento per la violazione delle linee guida, allo scopo di non incorrere in limitazioni della libertà d'espressione.

Come descritto nel capitolo 5, l'estensione per il browser Chrome si propone invece come uno strumento per classificare testo nelle pagine Web, reso più rapido ed efficiente dalla possibilità di selezionare il testo da analizzare. Ciò rende possibile proiettare questa estensione in un percorso evolutivo che potrebbe eventualmente culminare con l'analisi automatica dei contenuti della pagina come supporto a gestioni come il parental control.

Bibliografia

Abadi, Martın, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu e Xiaoqiang Zheng (nov. 2016). ≪TensorFlow: A System for Large-Scale Machine Learning≫. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, pp. 265–283. ISBN: 978-1-931971-33-1. URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

Cai, S., S. Bileschi e E. Nielsen (2020). *Deep Learning with JavaScript: Neural networks in TensorFlow.js*. Manning. ISBN: 9781617296178. URL: https://books.google.it/books?id=N2dswgEACAAJ.

Cer, Daniel, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope e Ray Kurzweil (2018). ≪Universal Sentence Encoder≫. In: *CoRR* abs/1803.11175. arXiv: 1803.11175. URL: http://arxiv.org/abs/1803.11175.

Chollet, François (nov. 2017). *Deep Learning with Python*. Manning. ISBN: 9781617294433.

Codice di condotta dell'UE per contrastare l'illecito incitamento all'odio. URL: https://ec.europa.eu/commission/presscorner/detail/it/ip_20_1134.

conversationai.github.io/toxicity. URL: https://github.com/conversationai/conversationai.github.io/blob/main/crowdsourcing_annotation_schemes/toxicity_with_subattributes.md.

Cos'è il machine learning? — Oracle Italia. URL: https://www.oracle.com/it/data-science/machine-learning/what-is-machine-learning/.

Counterspeech — Dangerous Speech Project. URL: https://dangerousspeech.org/counterspeech/.

Creare e pubblicare estensioni e app di Chrome personalizzate - Guida di Google Chrome Enterprise. URL: https://support.google.com/chrome/a/answer/2714278?hl=it.

De Marsico, Maria e Francesca Romana Mattei (2021). «VoiceWriting: a completely speech-based text editor». In: *CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter*, pp. 1–5.

Digital 2021: i dati globali - We Are Social Italia. URL: https://wearesocial.com/it/blog/2021/01/digital-2021-i-dati-globali/.

ECRI General Policy Recommendation N°15. URL: https://www.coe.int/en/web/european-commission-against-racism-and-intolerance/recommendation-no.15.

Flanagan, David (1998). *Javascript: The Definitive Guide*. Third. Sebastopol, California: O'Reilly.

Frain, B. (2020). *Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition*. Packt Publishing. ISBN: 9781839219795. URL: https://books.google.it/books?id=P03iDwAAQBAJ.

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated. ISBN: 9781492032649. URL: https://books.google.it/books?id=OCS1twEACAAJ.

GitHub. URL: https://github.com/.

GitHub - conversationai/perspectiveapi. URL: https://github.com/conversationai/perspectiveapi.

Hate speech, la normativa in Europa e Usa sull'odio online - Agenda Digitale. URL: https://www.agendadigitale.eu/cultura-digitale/le-strategie-di-contrasto-allodio-online-nellunione-europea-46113/.

https://github.com/lasaghnes/LearnToDetox. URL: https://github.com/lasaghnes/LearnToDetox.

Liao, Q Vera, Daniel Gruen e Sarah Miller (2020). «Questioning the AI: informing design practices for explainable AI user experiences». In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–15.

Marcotte, Ethan (2017). *Responsive web design: A book apart n 4*. Editions Eyrolles.

No Hate Speech Movement. URL: https://www.coe.int/en/web/no-hate-campaign/no-hate-speech-movement.

Nudge Theory Examples In Online Discussions — OpenWeb. URL: https://www.openweb.com/blog/openweb-improves-community-health-with-real-time-feedback-powered-by-jigsaws-perspective-api.

Risch, Julian e Ralf Krestel (gen. 2020). «Toxic Comment Detection in Online Discussions». In: pp. 85–109. ISBN: 978-981-15-1215-5. DOI: 10.1007/978-981-15-1216-2_4.

Smilkov, Daniel, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, Stan Bileschi, Michael Terry, Charles Nicholson, Sandeep N. Gupta, Sarah Sirajuddin, D. Sculley, Rajat Monga, Greg Corrado, Fernanda B. Viégas e Martin Wattenberg (2019). «TensorFlow.js: Machine Learning for the Web and Beyond». In: *CoRR* abs/1901.05350. arXiv: 1901.05350. URL: http://arxiv.org/abs/1901.05350.

Syed, Basarat Ali e Martin Bean (2014). *Beginning Node. js*. Springer.

Temere, Befekadu (2017). «Responsive web application using Bootstrap and Foundation: Comparing Bootstrap and Foundation Frontend Frameworks». In:

TensorFlow. URL: https://www.tensorflow.org/.

TensorFlow Hub. URL: https://www.tensorflow.org/hub.

Toxic Comment Multilabel Classification — Kaggle. URL: https://www.kaggle.com/geochatz/toxic-comment-multilabel-classification/data.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser e Illia Polosukhin (2017). «Attention is all you need». In: *Advances in neural information processing systems*, pp. 5998–6008.

Warner, William e Julia Hirschberg (giu. 2012). «Detecting Hate Speech on the World Wide Web». In: *Proceedings of the Second Workshop on Language in Social Media*. Montréal, Canada: Association for Computational Linguistics, pp. 19–26. URL: https://aclanthology.org/W12-2103.

Web Speech API - Web APIs — MDN. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API.

World Wide Web Consortium (W3C). URL: https://www.w3.org/.

Ringraziamenti

Desidero ringraziare il professor Claudio Gallicchio che ha seguito pazientemente lo sviluppo di questa tesi fornendomi sempre preziosi suggerimenti.

Un doveroso ringraziamento alla mia famiglia e in particolare babbo, per aver incosapevolmente lasciato che io imparassi ad utilizzare il computer per la prima volta all'età di cinque anni, mamma per la sua forza e la sua capacità di prendersi cura di tutti e Costanza per la compagnia.

Ringrazio Vale per l'amorevole pazienza e l'importanza che dà a tutto quello che faccio. Un grazie anche a Lisa, amica insostituibile e sempre presente e alle persone straordinarie che ho conosciuto durante questi anni all'Università, in particolare Alice, Michele, Sofia, Elena, Lorenzo e Marco, impossibile dimenticare la bellezza di aver conosciuto delle personalità così brillanti, simpatiche e interessanti in un contesto così stimolante come quello universitario.