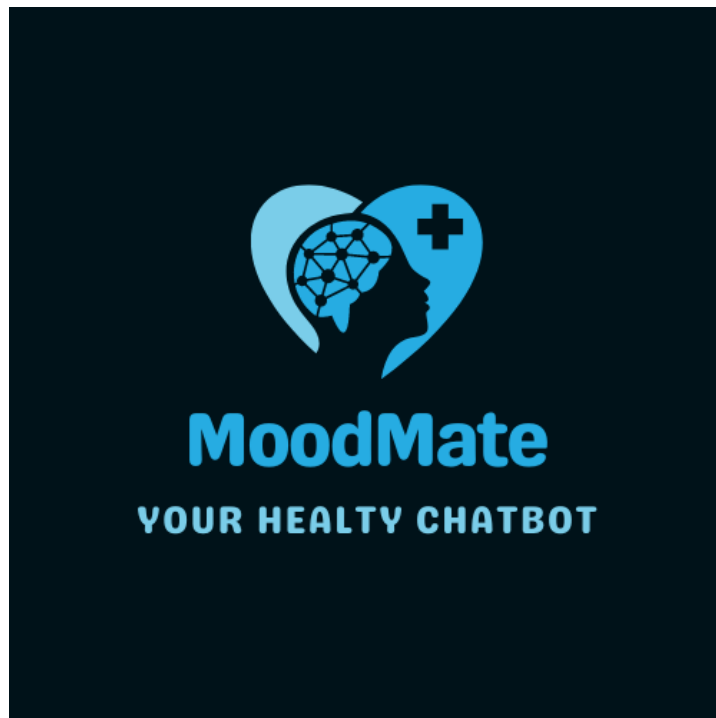


# MoodMate

Analisi del parlato e dello scritto per la rilevazione della depressione adolescenziale



## Link

GitHub – [MoodMate](#)

Notebook – [Colab](#)

## Gruppo di lavoro

**Giacomo Signorile**, 704897, g.signorile14@studenti.uniba.it

**Ester Molinari**, 716555, e.molinari3@studenti.uniba.it

AA 2022-23

<b>Obiettivi</b>	<b>3</b>
<b>Identificazione degli utenti</b>	<b>3</b>
<b>Personas</b>	<b>3</b>
Antonella	3
Luigi	4
<b>Happy Paths</b>	<b>4</b>
<b>Flowchart</b>	<b>4</b>
Inizio conversazione e analisi	4
<b>Apprendimento sul testo</b>	<b>7</b>
Introduzione	7
Strumenti utilizzati	7
Raccolta dei dati	7
Elaborazione dei dati	7
Addestramento del modello	10
Valutazione del modello	11
Integrazione del modello	11
<b>Apprendimento sulla voce</b>	<b>12</b>
Introduzione	12
Strumenti utilizzati	12
Raccolta dei dati	12
Elaborazione dei dati	12
Addestramento del modello	12
Valutazione del modello	12
Integrazione del modello	12
<b>Chatbot</b>	<b>13</b>
<b>Conclusioni</b>	<b>13</b>

# Obiettivi

Secondo l'Organizzazione Mondiale della Sanità (OMS), circa il 10-20% degli adolescenti in tutto il mondo soffre di disturbi mentali, tra cui la **depressione**. In uno studio condotto nel 2018, si stima che circa il 5-10% degli adolescenti **in Italia** soffra di depressione. Questa percentuale viene influenzata da molteplici fattori, tra cui la salute mentale, lo stile di vita, le circostanze familiari e la disponibilità di trattamenti adeguati.

Il nostro obiettivo è di fornire un chatbot in grado di **rilevare una possibile situazione di depressione** analizzando tre aspetti:

1. Costruzione delle **frasi** e utilizzo delle parole durante una banale conversazione non guidata affidata al bot
2. Analisi del tono della **voce** dato da un audio che comprende una breve descrizione della propria giornata
3. **Test** di 10 domande non cliniche per accertare la possibile situazione di depressione

## Identificazione degli utenti

Il nostro progetto è indirizzato per gli **adolescenti** (over 13) ed appare come una normale chat, analizzandone le informazioni prima di proporre un'analisi o un quiz per approfondire la condizione di tristezza dell'utente.

## Personas

Utilizziamo due **Personas** per descrivere degli utenti base del chatbot e analizzare le possibili situazioni che si possono verificare.

### Antonella

<b>Chi è l'utente?</b>	Antonella è una ragazza di 15 anni che frequenta le superiori. Le piace andare a scuola e studiare ma non le piace molto l'ambiente scolastico, quindi tende a chiudersi e non esprimersi con gli altri. Le piace fare sport e uscire con gli amici.
<b>Quali sono i suoi obiettivi?</b>	Vorrebbe avere qualcuno con cui parlare senza doversi preoccupare di come venga giudicata, dato che non si fida molto delle persone.
<b>Scenario</b>	Antonella ha passato una giornata stressante a scuola e sta pensando di non andare a lezione il giorno dopo. Prima di andare a dormire, decide di provare a parlare un po' con MoodMate.

## Luigi

<b>Chi è l'utente?</b>	Luigi è un ragazzo di 19 anni. Quest'anno deve superare l'esame di maturità ed è indeciso se continuare gli studi o andare a lavorare. Gli piace passare le giornate giocando online con gli amici perché non gli piace uscire.
<b>Quali sono i suoi obiettivi?</b>	Luigi vorrebbe capire se la depressione lo sta costringendo a rimanere in casa al posto di uscire. Si vergogna troppo di dire ciò ai propri genitori e vorrebbe che ci fosse un sistema che lo possa aiutare, senza dover uscire di casa.
<b>Scenario</b>	Luigi ha problemi nel addormentarsi, dato che resta fino a tarda notte a giocare al pc. Non riuscendo a prendere sonno, decide di provare MoodMate dal suo telefono.

## Happy Paths

Dato che il chatbot utilizza una API di OpenAI per la conversazione, le risposte non sono predefinite, ma si possono prevedere alcuni comportamenti:

1. L'utente ha inserito **meno di 3** input classificati come "tristi"
2. L'utente ha inserito **più di 3** input classificati come "tristi"

Il sistema riconoscerà un primo segnale di depressione legato alla **tristezza** ed approfondirà la situazione richiedendo l'analisi del tono della voce. Dopo aver fatto ciò, proporrà il **test** per confermare o smentire l'assunzione di depressione dell'utente rilevata.

## Flowchart

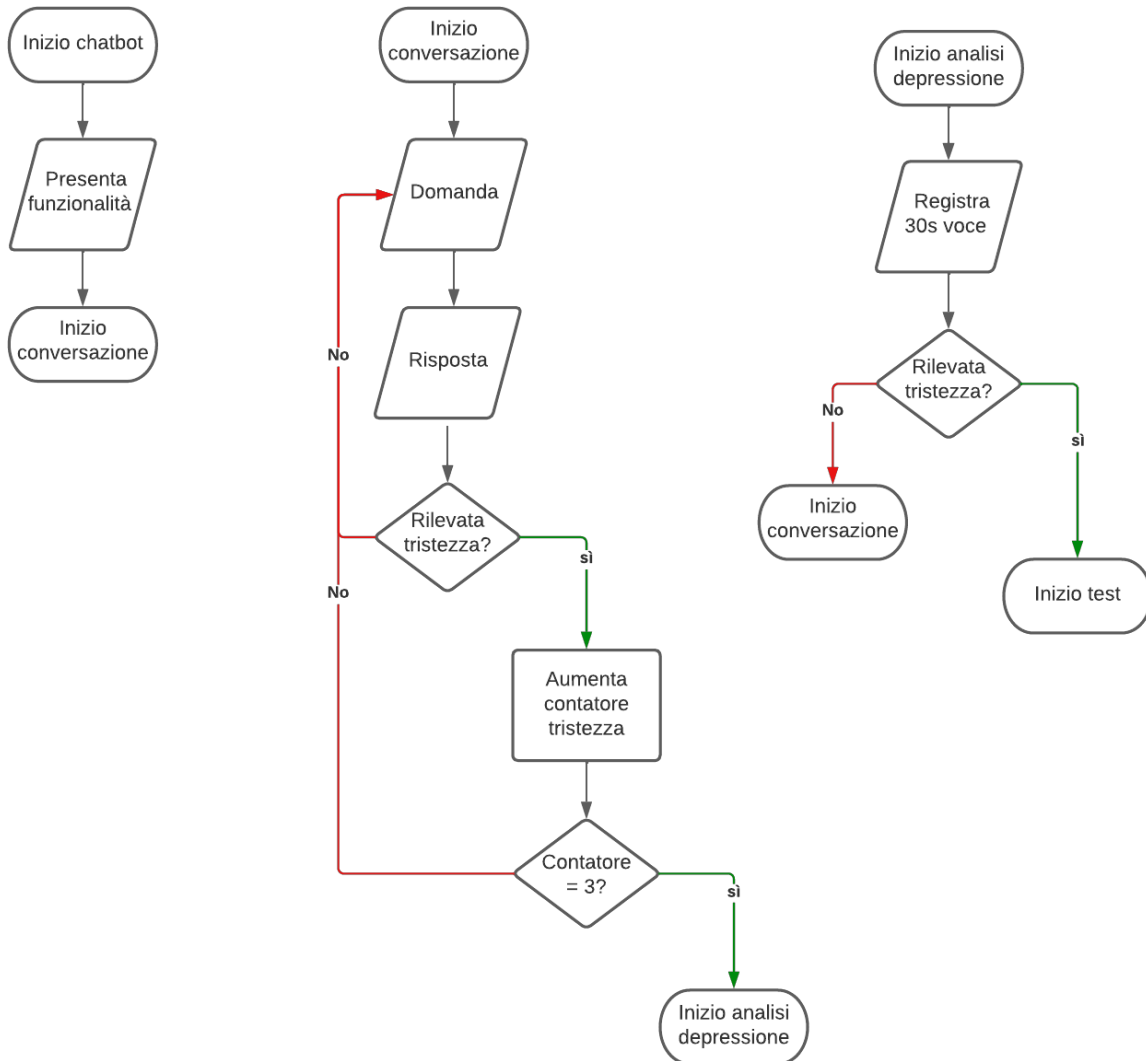
Analizziamo il funzionamento del bot mediante i seguenti flowchart che descrivono l'innescamento delle funzioni che sa applicare.

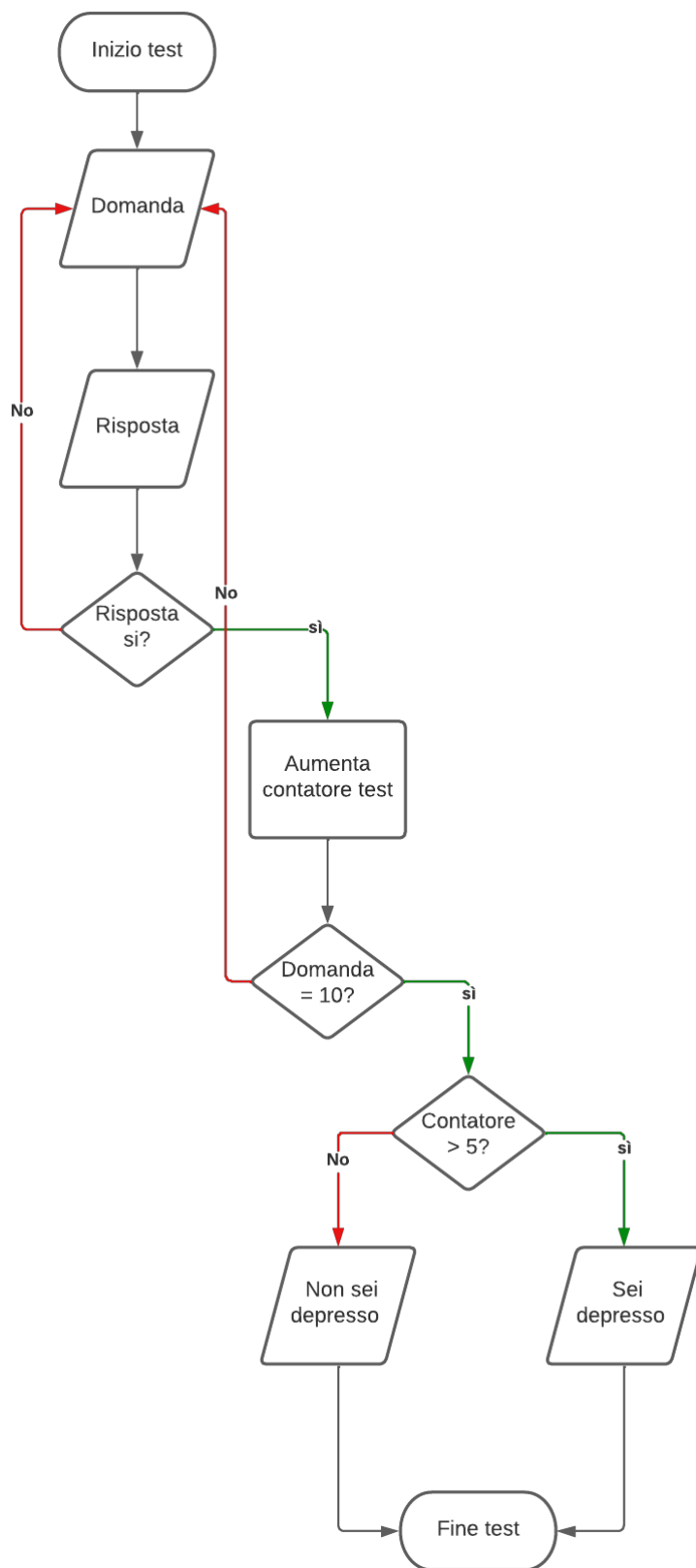
### Inizio conversazione e analisi

Il chatbot inizia con una **semplice conversazione** senza scopo composta di domande da parte dell'utente, che possono essere semplicemente frasi del tipo "Oggi non è stata una bella giornata" e risposte da parte del bot. L'API risponde, senza controllare lo stato d'animo dell'utente. Mentre va avanti la conversazione, **classifichiamo** le domande dell'utente in "tristi", "felici" o "neutre".

Ogni volta che viene classificata una domanda, viene incrementato un **contatore** relativo alla feature. Quando il contatore della tristezza supera una certa soglia, si attiva l'**analisi della depressione**.

Questa prima analisi è un semplice **controllo del tono della voce**. Viene chiesto all'utente di raccontare la giornata in 30 secondi e successivamente viene rilevato il tono. Se il tono è "triste", viene avviato il **test**, altrimenti continua la conversazione per rilevare la tristezza.





Se l'esito dell'analisi del tono della voce risulta "triste", inizia un **test di 10 domande** non cliniche per il riconoscimento della depressione. Alle domande si risponde solo con **"sì" o "no"** per rendere più facile l'interpretazione della risposta e l'inserimento da parte dell'utente.

Una volta finito il test, si ritorna alla conversazione del chatbot e si chiede se si vuole interrompere la conversazione, dopo aver mostrato i **risultati** del test.

Il chatbot è in grado di rispondere in ogni lingua, però predilige l'**italiano** durante la conversazione, ipotizzando l'utilizzo da parte di utenti italiani.

Per la parte di analisi del testo, l'input viene tradotto in **inglese**, avendo usato un training basato su frasi in inglese. Per la parte di analisi del parlato, viene richiesto l'utilizzo della lingua **inglese**, per lo stesso motivo.

# Apprendimento sul testo

**Nota:** per informazioni più dettagliate, consultare [questo link](#)

## Introduzione

Per riconoscere la tristezza dal parlato e dallo scritto abbiamo effettuato le seguenti operazioni:

1. **Raccolta dei dati:** abbiamo raccolto un insieme di dati audio o di testo (come tweet) di persone classificabili come “tristi”, “neutri” o “felici”
2. **Pre-elaborazione dei dati:** abbiamo elaborato i dati audio per poterli trascrivere e i dati testuali per poterli analizzare
3. **Addestramento del modello:** abbiamo applicato varie tecniche di apprendimento automatico per addestrare il modello sui dati elaborati (naive bayes, regressione logistica e reti neurali)
4. **Valutazione del modello:** abbiamo utilizzato un insieme di dati di test per determinare l'accuratezza del modello
5. **Integrazione del modello:** abbiamo applicato il modello al nostro chatbot per consentire la classificazione di dati di input diversi da quelli conosciuti

Riconoscere la tristezza in un testo può essere un compito complesso, poiché può manifestarsi in modo implicito o sottinteso. Inoltre, il significato delle parole può cambiare a seconda del contesto in cui vengono utilizzate. Per tanto, abbiamo selezionato dataset molto **grandi** per ottenere risultati migliori.

## Strumenti utilizzati

Abbiamo realizzato tutto in Python utilizzando le librerie pandas, numpy, seaborn, matplotlib, tensorflow, keras, nltk, sklearn e pickle. Abbiamo realizzato anche un Notebook su Colab.

## Raccolta dei dati

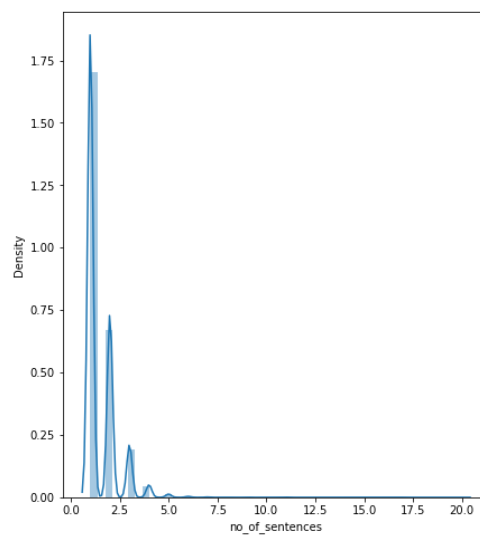
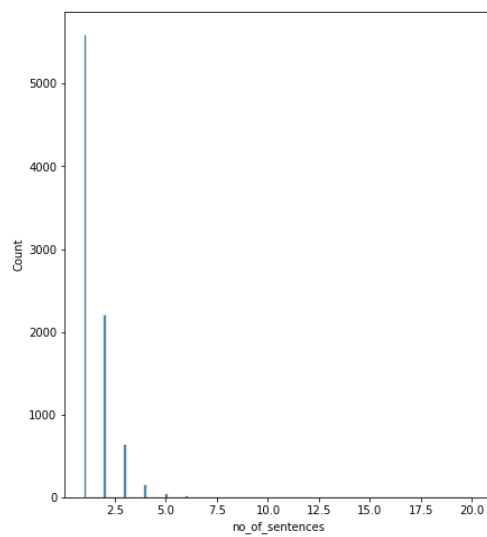
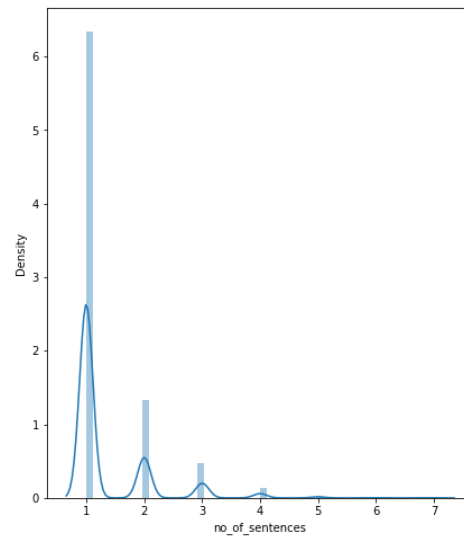
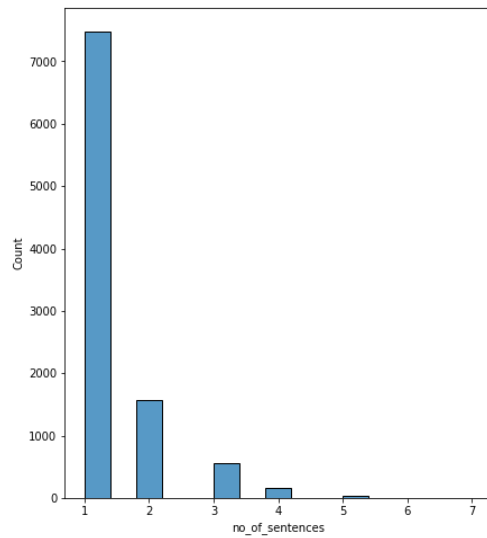
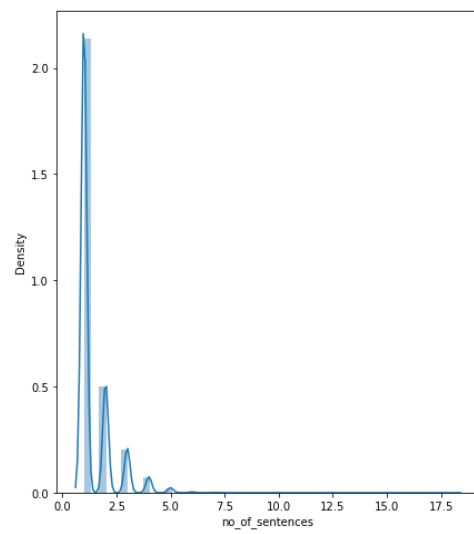
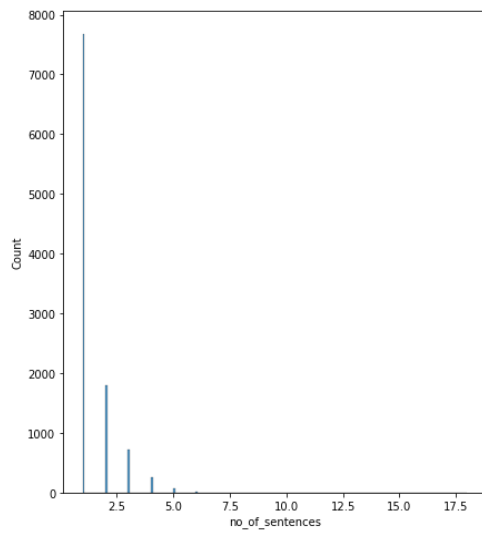
Abbiamo utilizzato un dataset composto da frasi in inglese prelevate da vari **tweet** classificate in base alle emozioni prevalenti. Da questo abbiamo estrapolato un insieme di training, un insieme di validazione e un insieme di test.

Per l'addestramento sulla voce, abbiamo utilizzato diversi file **audio** classificati in base alla “tristezza”, “felicità” e “neutralità”.

## Elaborazione dei dati

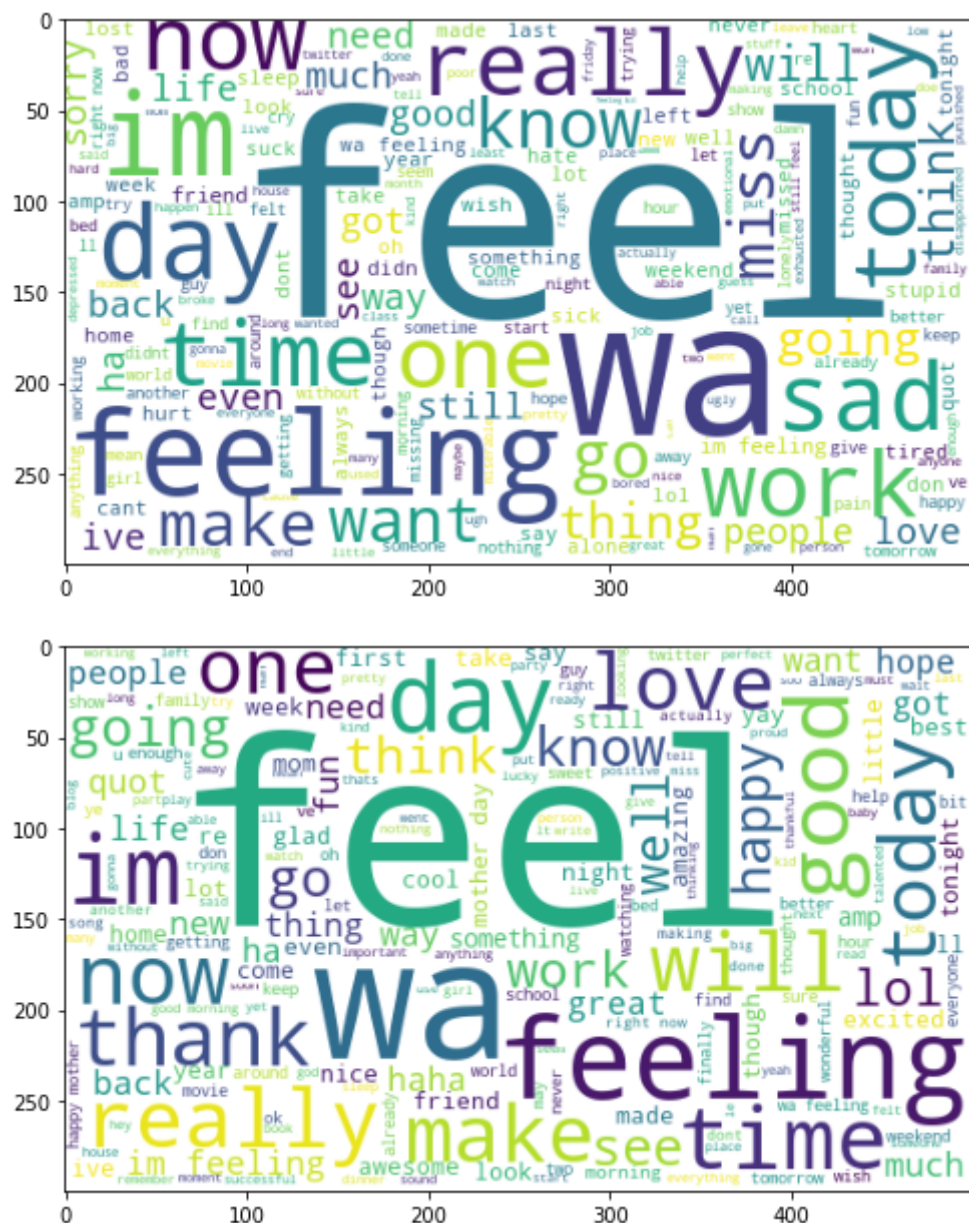
Abbiamo **rimosso** alcune feature che non erano utili ai fini del progetto e abbiamo verificato quali etichette erano presenti nei dataset. Abbiamo preso in considerazione solo i record etichettati come “**tristi**”, “**neutri**” e “**felici**”. A questo punto, abbiamo rappresentato

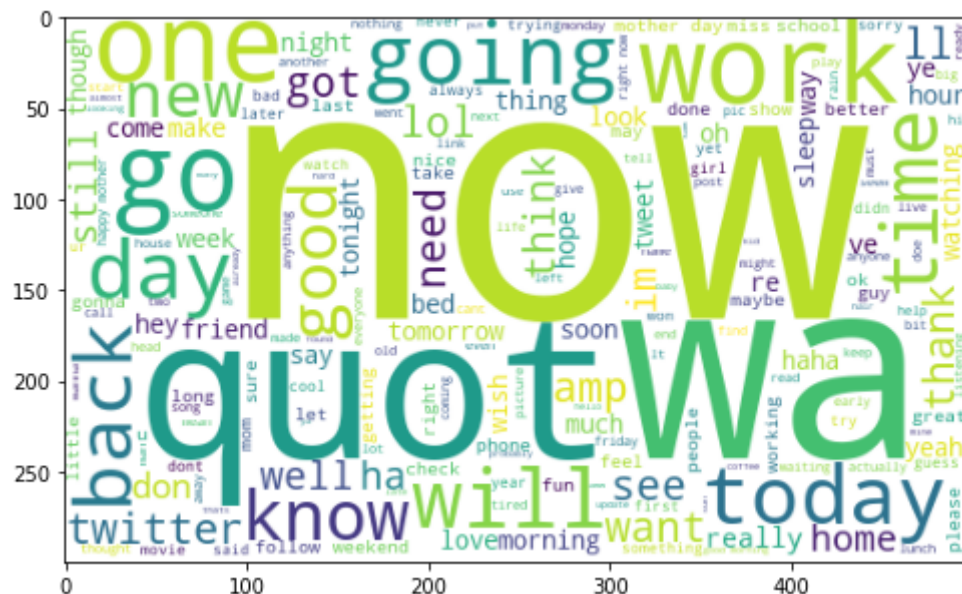
**graficamente** i dati selezionati. Rispettivamente, rappresentazione delle frasi classificate come “felici”, “tristi” e “neutre”.





Possiamo riassumere i nostri tre insiemi di frasi con le seguenti **word clouds**, rispettivamente per “tristezza”, “felicità” e “neutro”:





Dopo aver elaborato tutti i dati, abbiamo ottenuto il seguente risultato:

happiness	10425
sadness	9774
neutral	8282

Abbiamo voluto eliminare il **25%** dei record “neutrali” per evitare un **overfitting** in questa categoria, portando il valore **da 8282 a 6238**.

## Addestramento del modello

Iniziamo la fase di addestramento utilizzando un **LabelEncoder** sulle tre feature che abbiamo selezionato. Abbiamo utilizzato insiemi di frasi di dimensione costante 20000.

label[0]	sadness
label[1]	neutral
label[2]	happiness

A questo punto, abbiamo applicato un TF-IDF ricavando una matrice contenente valori numerici come classificazione iniziale. Applichiamo quindi tre modelli di **apprendimento bayesiano**:

1. **Gaussian Naive Bayes**, con una accuratezza del 0.52275
2. **Multinomial Naive Bayes**, con una accuratezza del 0.63125
3. **Bernoulli Naive Bayes**, con una accuratezza del 0.62575

Esportiamo il modello più accurato per poterlo usare dopo. Continuiamo l'applicazione dei modelli di apprendimento con la **Regressione Logistica**, che presenta un'accuratezza del 0.64975 mentre il **Random Forest** raggiunge un'accuratezza del 0.6155 quindi riassumiamo tutti i modelli nella sezione successiva.

## Valutazione del modello

Riassumiamo l'**accuratezza** di tutti i modelli utilizzati e la scelta del modello applicato per il chatbot:

Modello	Accuratezza
Gaussian Naive Bayes	0.52275
Multinomial Naive Bayes	0.63125
Bernoulli Naive Bayes	0.62575
Logistic Regression	0.64975
Random Forest	0.6155

## Integrazione del modello

Abbiamo scelto di utilizzare il **Multinomial Naive Bayes** e l'abbiamo testato con due semplici frasi in input:

- I didn't got the job
- I don't feel good

Entrambe le frasi sono state classificate correttamente in "sadness". Il modello è stato integrato nella **fase di conversazione** del chatbot per analizzare le risposte che l'utente fornisce al bot e rilevare quindi uno stato di "tristezza", indice generale di depressione.

# Apprendimento sulla voce

## Introduzione

Il procedimento applicato per l'apprendimento sulla voce è analogo a quello fatto sul testo, con file audio già suddivisi per feature di interesse.

## Strumenti utilizzati

Sono state utilizzate le librerie librosa, soundfile, glob, pickle, numpy ed sklearn. Anche questa parte di apprendimento è presente nel Notebook Colab.

## Raccolta dei dati

La raccolta dei dati è avvenuta durante la raccolta per i dati testuali, infatti tutti i dati sono presenti in una cartella unica presente sul Drive del collega Signorile.

## Elaborazione dei dati

Abbiamo estratto le features dai file audio ottenendo le seguenti emozioni: "neutral", "calm", "happiness", "sadness", "angry", "fearful", "disgust", "surprised". Anche in questo caso, le emozioni che vogliamo osservare sono solo **"happiness"**, **"sadness"** e **"neutral"**, come nell'apprendimento sul testo.

## Addestramento del modello

Abbiamo diviso quindi il dataset per avere un insieme di training e uno di test. Il modello è stato addestrato utilizzando un **Multi Layer Perceptron Classifier** con una accuratezza del 70.83% nella classificazione.

A questo punto, abbiamo realizzato un dataset per controllare la **correttezza** della classificazione verificando se l'etichetta effettiva era stata predetta correttamente. Abbiamo ulteriormente testato il modello con altri file audio in input per verificarne la classificazione.

## Valutazione del modello

Riassumiamo i dettagli tecnici della valutazione del modello utilizzato:

Modello	Accuratezza
Multi Layer Perceptron Classifier	70.83% (0.64 da sklearn)

## Integrazione del modello

Per integrare questo classificatore nel nostro chatbot, abbiamo utilizzato la libreria wave per **prendere in input l'audio** e salvarlo in un file .wav che verrà poi trascritto in testo dalla API Whisper di OpenAI ed effettuare anche la classificazione testuale della frase.

# Chatbot

La nostra idea è quella di non influenzare le risposte dell'utente sulla base del funzionamento del chatbot. Per questo motivo, abbiamo scelto di utilizzare l'API di OpenAI per il modello **Chat GPT-3**. Questa chat, precursore di Chat GPT-4, non raggiunge i livelli di precisione dell'ultima citata ma è in grado di comprendere il "tono" delle frasi, ingannando l'utente e spronando a comunicare con il bot.

Mentre l'utente utilizza il bot per esternare i propri pensieri, viene effettuato l'apprendimento sul testo che viene ricevuto in input dal bot. Se viene rilevata una grande quantità di messaggi tendenti alla "tristezza", il chatbot viene interrotto da una fase di accertamento delle condizioni dell'utente in due fasi:

1. **Analisi della voce**, per rilevare un possibile tono "triste" dell'utente applicando l'apprendimento sulla voce
2. **Somministrazione del test sulla depressione**, presentando una raccolta di 10 (o più) domande a cui rispondere con un "Sì" o un "No" per capire se c'è concordanza tra la "tristezza" rilevata nelle risposte con il chatbot e il risultato del test

## Conclusioni

Attualmente il progetto funziona solo su CLI e non è pronto per la distribuzione, però vorremmo implementare il chatbot come bot telegram.

Avendo dato più importanza allo sviluppo del progetto che alla sua distribuzione, l'implementazione in telegram avverrà successivamente.