Università degli Studi di Milano - Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione
Corso di Laurea in Informatica

Verso Sistemi SER Affidabili: Analisi critica delle prestazioni e Sfide Multilingua

Relatore: Prof. Francesca Gasparini

Correlatore: Dott. Alessandra Grossi

Tesi di Laurea di:
Riccardo Mattia
Matricola 885964

Anno Accademico 2023-2024

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

I sistemi di **Speech Emotion Recognition (SER)** hanno l'obiettivo di identificare automaticamente le emozioni espresse attraverso il parlato, essi rappresentano un'area di crescente interesse sia accademico che industriale. La loro applicazione infatti, risulterebbe particolarmente promettente in svariati settori [15]: nel marketing, la capacità di rilevare l'emozione di un consumatore in tempo reale potrebbe consentire la creazione di esperienze di vendita più personalizzate e di pubblicità mirate, aumentando il coinvolgimento e la soddisfazione del cliente. Allo stesso modo, nel settore della sicurezza stradale, un sistema SER integrato potrebbe rilevare stati emotivi come rabbia o stress nel conducente [29] [33], attivando misure preventive, ad esempio limitando temporaneamente la velocità del veicolo o suggerendo una pausa, riducendo così il rischio di incidenti.

Tuttavia, nonostante i progressi tecnologici, allo stato attuale i sistemi SER non sono ancora maturi per una diffusione su larga scala. Uno dei principali ostacoli è rappresentato dalla bassa accuratezza nel riconoscimento delle emozioni, specialmente in contesti reali e non controllati [30], dove il rumore ambientale e la variabilità del parlato umano complicano ulteriormente il processo di riconoscimento. Questo fenomeno evidenzia come il riconoscimento delle emozioni in ambienti naturali rimanga una sfida aperta nella letteratura scientifica, con molti modelli che faticano a generalizzare in scenari diversi da quelli in cui sono stati addestrati.

Il lavoro presentato in questa tesi si propone di migliorare le pratiche di valutazione dei modelli SER, ponendo l'accento su un approccio metodologico più *fair*. Questo approccio intende evitare uno dei bias più comuni che affliggono molti studi precedenti [27] [8] , ovvero l'uso dello stesso soggetto sia nelle fasi di training che di testing, che può gonfiare artificialmente le performance del sistema. In questo modo si mira ad ottenere una valutazione più accurata delle capacità **effettive** degli algoritmi di riconoscimento.

Inoltre, questo lavoro esplora le difficoltà legate allo sviluppo di un sistema SER multilingua, capace di riconoscere emozioni dal parlato in diverse lingue. Le differenze culturali, linguistiche e fonetiche rappresentano [14] una sfida significativa per la creazione di un modello universale che possa adattarsi a più lingue senza perdere efficacia. Si discute come le metodologie esaminate evidenzino tali difficoltà, rendendo complesso lo sviluppo di un sistema SER che possa funzionare efficacemente in contesti multilingua, sottolineando la necessità di ulteriori ricerche in questo ambito.

Capitolo 2

Modelli Emozionali

Per costruire dei modelli di riconoscimento delle emozioni attraverso il parlato dobbiamo prima porci il problema di come modellare e definire efficacemente le emozioni.

Ad oggi in psicologia non esiste una definizione comune di emozione. In generale un'emozione può essere uno stato psicologico complesso composto da diverse componenti come esperienze personali, reazioni fisiologiche, comportamentali e comunicative [25].

Sulla base di queste definizioni due modelli sono diventati comuni nell'ambito del riconoscimento delle emozioni dal parlato: **il modello emozionale discreto** e **il modello emozionale dimensionale**.

Grazie ad essi è possibile formulare il problema di riconoscimento delle emozioni dal parlato come problema di classificazione, nel caso stessimo utilizzando il modello discreto, o di regressione nel caso stessimo utilizzando il modello dimensionale [30].

2.1 Il modello discreto

I modelli emozionali discreti si basano su categorie finite di emozioni. Uno dei più famosi modelli di questo tipo è quello proposto da Ekman [5] in cui esistono sei categorie di emozioni di base: **tristezza, felicità, paura, rabbia, disgusto e sorpresa**, da questo modello base sono nate altre varianti in cui tutte le altre emozioni sono espresse come una combinazione di queste sei principali. Il vantaggio principale di questo modello è che nella vita quotidiana, le persone utilizzano questo modello per definire le loro emozioni; quindi, lo schema di etichettatura basato su categorie emotive è intuitivo. Uno svantaggio di questo modello è che non riesce a cogliere alcuni stati emotivi complessi osservati nella comunicazione quotidiana.

2.2 Il modello dimensionale

I modelli emozionali dimensionali caratterizzano le emozioni attraverso un certo numero di **dimensioni**.

Uno dei modelli dimensionali più famosi è il modello circomplesso di Russel [3] dove le tre dimensioni principali sono valenza, attivazione e/o dominanza, vedi fig 2.1. La dimensione della valenza descrive se un'emozione è positiva o negativa e varia da spiacevole a piacevole. La dimensione dell'arousal definisce l'intensità dell'emozione percepita, in questo caso bassi valori di arousal corrispondono a stati di quiete come la noia o la tristezza, mentre elevati valori di arousal sono associati a emozioni di forte eccitazione come la felicità o la rabbia.

Con una dimensione in più si possono meglio definire alcune emozioni al costo però di complicare il modello, per esempio il modello tridimensionale solitamente utilizza come terza dimensione la dominanza o potere questa Si riferisce al livello di controllo o potere percepito da una persona in una situazione o emozione. Un'emozione con alta dominanza implica che la persona si sente in controllo, forte o autorevole, mentre un'emozione con bassa dominanza implica un senso di vulnerabilità o sottomissione. Ad esempio, con la terza dimensione riusciamo a differenziare la rabbia dalla paura, Entrambe possono avere un alto livello di attivazione (sono emozioni intense) ed entrambe possono avere un valore negativo di valenza (non sono emozioni positive), ma la paura è associata a bassa dominanza (sensazione di debolezza), mentre la rabbia è associata a alta dominanza (sensazione di potere).

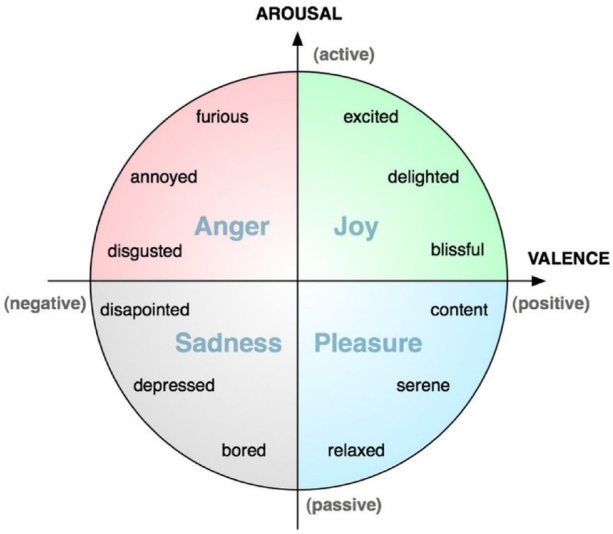


Figura 2.1: Modello emotivo bidimensionale [17]

Il punto di forza del modello dimensionale è quello appunto di riuscire a definire meglio gli stati emotivi più complessi. Tuttavia, ci sono diversi svantaggi per la rappresentazione dimensionale. Non è sufficientemente intuitiva e può essere necessaria una formazione speciale per etichettare ogni emozione. Inoltre, alcune emozioni diventano identiche, come paura e rabbia nel caso ci stessimo riferendo a un modello valenza-attivazione. infine, la mappatura delle emozioni discrete su uno spazio dimensionale continuo può risultare difficile. Ad esempio, emozioni come la sorpresa possono avere una valenza positiva o negativa a seconda del contesto, rendendo complessa la loro categorizzazione precisa. Tuttavia, questo non rappresenta un limite dello spazio continuo, ma piuttosto della sua interpretazione in relazione alle emozioni discrete.

Capitolo 3

Datasets

3.1 Introduzione

Nel contesto del riconoscimento delle emozioni nel discorso (SER), una vasta gamma di dataset è stata sviluppata per sostenere la ricerca e produrre modelli predittivi sempre più accurati. Questi dataset comprendono registrazioni di discorsi umani in una varietà di contesti e con una vasta gamma di espressioni emotive. Possiamo suddividere i dataset in tre grandi categorie:

- **Recitati:** le tracce audio sono recitate da attori professionisti o non professionisti e sono solitamente registrate con apparecchiatura professionale in ambienti privi di rumore.
- **Indotti:** in questi dataset le tracce sono registrate da persone, non più attori, a cui una certa emozione è stata indotta. Un'emozione può essere indotta attraverso una particolare domanda o facendo ascoltare al soggetto una canzone. In questo modo le emozioni sono più vicine a quelle reali ma rimane comunque la consapevolezza del soggetto di essere registrato con la possibilità quindi di non esprimere una certa emozione con la massima naturalezza.
- **Naturali:** questi dataset sono composti da registrazioni acquisite in contesti reali come conversazioni telefoniche, interviste, discorsi pubblici o situazioni sociali quotidiane. Questi dataset sono preziosi perché riflettono situazioni autentiche in cui le persone esprimono emozioni in modo spontaneo e naturale.

Il mio focus sarà incentrato su due dataset RAVDESS ed EMOVO

3.2 EMOVO

EMOVO è un dataset contente registrazioni in lingua italiana, **recitato** da attori professionisti e giovani, infatti hanno una media di 27.2 anni.

I 6 attori (di cui 3 maschi e 3 femmine) hanno ripetuto 14 frasi per ognuna delle 7 emozioni principali del modello di Ekman a cui è stato aggiunto lo stato emotivo neutro (assenza di emozione), in totale i file audio sono 6*14*7=588.

Ogni file nel dataset ha un nome composto da tre parti, ad esempio:

neu-m1-b1.wav

Nella prima parte del nome è indicato lo stato emotivo (**neu**tro, **dis**gusto, **gio**ia, **pau**ra, **rab**bia, **sor**presa, **tri**stezza).

Nella seconda parte del nome è identificato l'attore/attrice (**m1, m2, m3, f1, f2, f3**) che recita la frase.

Nella terza parte del nome è indicato il tipo di frase:

- le frasi 1-3 sono state classificate rispettivamente. b1, b2, b3 (**b**revi)
- le frasi 4-7 sono state classificate risp. l1, l2, l3, l4 (**l**unghe)
- le frasi 8-12 sono state classificate risp. n1, n2, n3, n4, n5 (**n**onsense)
- le frasi 13,14 sono state classificate risp. d1, d2 (**d**omande)

tutte le registrazioni nel dataset sono state acquisite con due microfoni professionali SHURE SM58LC nei laboratori della fondazione Ugo Bordoni con una frequenza di campionamento di 48 kHz, 16 bit stereo e in formato wav.

La durata totale delle registrazioni è di circa 60 minuti di audio.

3.3 RAVDESS

Il Ryerson Audio-Visual Database of Emotional Speech and Song [18] è un dataset contente registrazioni in lingua inglese precisamente con un accento nordamericano. anche questo dataset è **recitato** da attori professionisti delle quali viene anche fornito il range di età:

sono 24 attori (12 maschi e 12 femmine) con età media: 26 anni, SD: 3.75, range di età: 21-33 anni. Gli stati emotivi presenti nelle registrazioni sono otto: calma, neutro, gioia, tristezza, paura, disgusto, sorpresa, rabbia.

Due caratteristiche importanti di RAVDESS sono:

1. ogni emozione è espressa a **due livelli di intensità**: normale e forte. L'intensità è un aspetto fondamentale per determinare una certa emozione
2. include **due emozioni di base** neutra e calma. Questo è stato fatto perché le espressioni neutre hanno generato risultati percettivi contrastanti, talvolta trasmettendo una valenza emotiva negativa. Gli studiosi ipotizzano che ciò possa derivare dall'incertezza dell'interprete su come trasmettere la neutralità. Per ovviare a ciò, è stata introdotta una condizione di base calma, simile alla neutralità ma con una lieve valenza emotiva positiva

Il corpus audio contiene un totale di 1440 tracce audio che sono il risultato di 60 tracce per ognuno dei 24 attori. Ogni file nel dataset ha un nome composto da 7 identificatori numerici a due cifre separati da un trattino, ad esempio:

02-01-06-01-02-01-12.mp4

Gli identificatori sono ordinati:
Modalità–Canale–Emozione–Intensità–Frase–Ripetizione–Attore.mp4 o .wav.

Riporto la tabella che spiega i vari valori numerici per i possibili identificatori nei nomi (vedi figura 3.1)

Identifier	Coding description of factor levels
Modality	01 = Audio-video, 02 = Video-only, 03 = Audio-only
Channel	01 = Speech, 02 = Song
Emotion	01 = Neutral, 02 = Calm, 03 = Happy, 04 = Sad, 05 = Angry, 06 = Fearful, 07 = Disgust, 08 = Surprised
Intensity	01 = Normal, 02 = Strong
Statement	01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door"
Repetition	01 = First repetition, 02 = Second repetition
Actor	01 = First actor, . . ., 24 = Twenty-fourth actor

Figura 3.1: RAVDESS naming [18]

Le registrazioni vocali sono state catturate da un microfono a condensatore tubolare Rode NTK, dotato di un filtro pop Stedman proscreen XL, posizionato a 20 cm dall'attore. L'uscita del microfono è stata registrata utilizzando Pro Tools 8 e una workstation di missaggio Digidesign 003, a una frequenza di campionamento di 48 kHz, 16 bit, con file salvati in formato wave non compresso.

3.4 Validazione

3.4.1 Test di validazione di EMOVO

Per la validazione di EMOVO sono stati organizzati 2 gruppi da 12 persone in due diversi laboratori, ad ognuno è stato fatto ascoltare un "pre-testing" che consiste in 6 frasi una per ognuno dei 6 stati emozionali. Successivamente sono state scelte due frasi nosense ovvero:

- *la casa forte vuole col pane*
- *il gatto sta scorrendo nella pera*

per ogni attore e fatte ascoltare ai soggetti che hanno dovuto scgliere tra due possibili emozioni. Le frasi nosense sono state scelte per evitare che il significato semantico della frase potesse influenzare la scelta dell'emozione. I risultati di questo test di validazione sono riportati nella figura 3.2

		RECOGNIZED EMOTION						
		NEUTRAL	**DISGUST**	**JOY**	**FEAR**	**ANGER**	**SURPRISE**	**SADNESS**
ELICITED EMOTION	**NEUTRAL**	**93%**	**1%**	**0%**	**0%**	**4%**	**0%**	**2%**
	DISGUST	**3%**	**67%**	**2%**	**6%**	**10%**	**6%**	**6%**
	JOY	**2%**	**4%**	**65%**	**7%**	**7%**	**10%**	**4%**
	FEAR	**2%**	**7%**	**2%**	**74%**	**3%**	**3%**	**9%**
	ANGER	**1%**	**1%**	**1%**	**3%**	**92%**	**1%**	**1%**
	SURPRISE	**1%**	**3%**	**4%**	**1%**	**1%**	**81%**	**9%**
	SADNESS	**2%**	**2%**	**1%**	**3%**	**0%**	**0%**	**92%**

Figura 3.2: EMOVO validation test [13]

3.4.2 Test di validazione di RAVDESS

Premetto che RAVDESS è composto da quattro subset, due di questi sono audio-only, tra questi due uno è composto da audio cantati, mentre l'altro da audio solo parlati. Per questa parte di test di validazione descriverò solo quel che riguarda il subset solo audio e solo parlato.

Per valutare la validità del corpus, hanno preso parte all'attività di valutazione duecentoquarantasette valutatori. Questi sono stati esposti a 174 file audio parlati; i valutatori sono stati posizionati a circa 60 cm dal display del computer. Oltre alle istruzioni verbali, sono state presentate istruzioni visive sullo schermo che chiedevano ai valutatori di identificare la categoria, l'intensità e la genuinità dell'emozione espressa nelle registrazioni. Le frasi che sono state valutate sono:

- *Kids are talking by the door*
- *Dogs are sitting by the door*

I valutatori dovevano identificare la categoria di emozione utilizzando un formato di risposta a scelta forzata. Le opzioni per i discorsi erano: neutro, calmo, felice, triste, arrabbiato, spaventato, disgustato e sorpreso. Inoltre, è stata fornita l'opzione di fuga "Nessuna di queste è corretta". I valutatori hanno poi valutato l'intensità dell'emozione utilizzando una scala Likert a 5 punti che va da molto debole (1) a molto forte (5),allo stesso modo i valutatori hanno poi valutato la genuinità della presentazione utilizzando una scala Likert a 5 punti che va da non genuina (1) a molto genuina (5).

Le risposte dei valutatori potevano essere fornite solo dopo la visualizzazione dello schermo di feedback, garantendo che i partecipanti avessero ascoltato l'intero file e impedendo loro di avanzare rapidamente attraverso il compito. I risultati di questo test di valutazione sono riportati nella figura 3.3

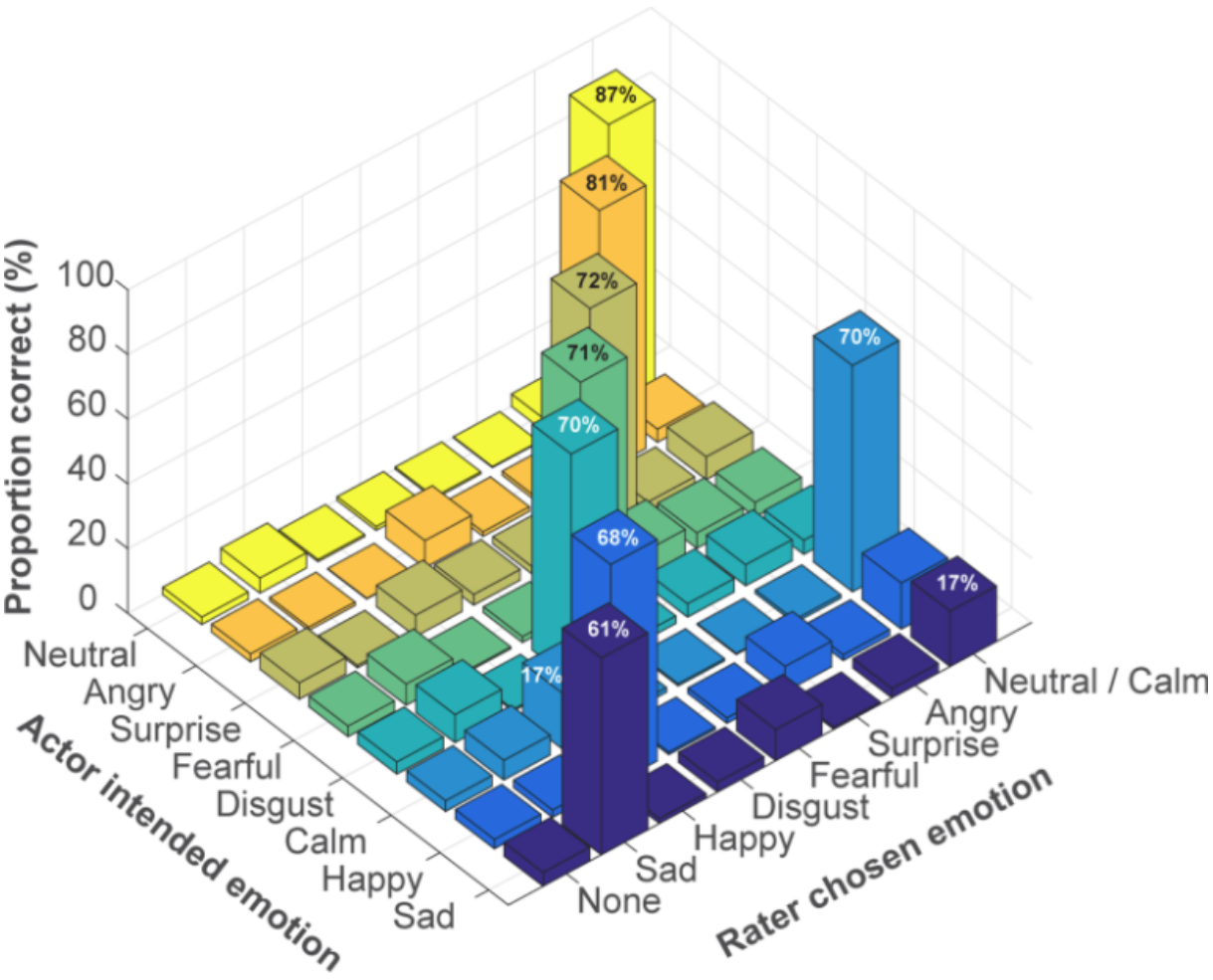


Figura 3.3: La matrice di confusione costruita con i dati del validation test [18]

3.4.3 Considerazioni

Come è possibile osservare dalla matrice di confusione di EMOVO nella figura 3.2 tutte le classi di emozioni sono state ben interpretate e distinte con precisioni che non scendono oltre al 65%. Alcune classi sono state più problematiche da individuare ad esempio la gioia scambiata con la sorpresa, la paura scambiata spesso con la tristezza e il disgusto scambiato con la rabbia. Mentre altre classi come l'emozione neutra la rabbia, la tristezza e la sorpresa sono state individuate con percentuali alte. È possibile osservare che questi risultati, a meno di qualche classe, sono simili a quelli ottenuti per il validation test di RAVDESS, nonostante le modalità di validazione siano state leggermente diverse.

La tristezza, nel caso di EMOVO viene riconosciuta con una percentuale tra le più alte mentre nel caso di RAVDESS viene riconosciuta con la percentuale più bassa e viene spesso scambiata per l'emozione neutra. Per quello che riguarda le altre classi di emozioni, in generale, vengono distinte con percentuali leggermente superiori nel dataset EMOVO ma comunque con distribuzioni simili. Le classi con percentuali più simili sono le classi paura e del disgusto.

Questi risultati potranno esserci utili da *"benchmark"* per i modelli sviluppati trattati nelle sezioni successive: se i valutatori umani riconoscono con questa accuratezza le classi di emozioni, un modello di machine learning riesce a "battere" la valutazione umana, e in che misura?

Capitolo 4

Preprocessing

La fase di preprocessing segue immediatamente la raccolta dei dati ed è cruciale per preparare i segnali audio per la classificazione. Durante questa fase, i segnali audio vengono depurati da informazioni non pertinenti quali il rumore di fondo e i periodi di silenzio, che potrebbero compromettere l'accuratezza del modello. Inoltre, i segnali vengono normalizzati per assicurare che le variazioni nelle registrazioni dei diversi oratori non influenzino il processo di riconoscimento delle emozioni. Questo step di normalizzazione è fondamentale per garantire che il modello sia robusto e in grado di generalizzare efficacemente a nuovi dati. Il preprocessing può comprendere gli step indicati nelle sezioni seguenti.

4.1 Framing

Questo processo, noto anche come segmentazione del discorso, consiste nel suddividere i segnali vocali continui in sezioni di lunghezza fissa, tipicamente da 20 a 30 millisecondi (questo è dato dal fatto che il discorso in questi piccoli archi di tempo rimane invariato). Questa suddivisione permette di trattare ogni frame come un'unità quasi stazionaria, facilitando l'analisi delle caratteristiche locali del segnale. Durante il framing, è comune sovrapporre i segmenti tra loro di un 40% o 50%, per mantenere la continuità e la coerenza delle informazioni tra i frame successivi. Le feature estratte dai frame costituiscono le caratteristiche **locali** del segnale, mentre le caratteristiche **globali** vengono solitamente calcolate come statistiche delle locali e sono ad esempio: media, varianza e deviazione standard. Una visualizzazione di framing può essere vista nella figura 4.1.

4.2 Windowing

La fase successiva al framing è generalmente il windowing. Questa fase implica la suddivisione del segnale audio in frame sovrapposti o non sovrapposti e la moltiplicazione di ciascun frame con una funzione finestra. Questo processo aiuta a **minimizzare gli effetti indesiderati** dovuti alla discontinuità dei

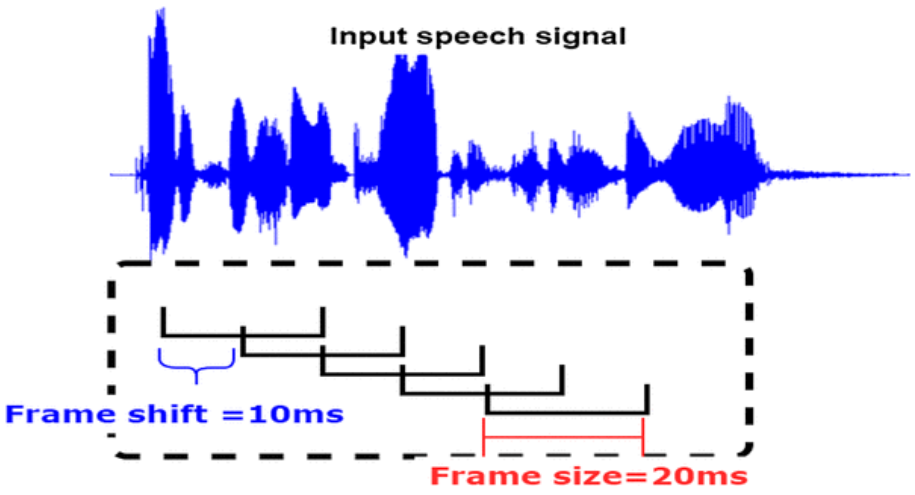


Figura 4.1: visualizzazione di come avviene il framing su un segnale [19]

bordi del segnale come il fenomeno dello spectral leakage che occorre quando si cerca di applicare la Fast Fourier Transform (FTT) ad un segnale. Questa fase è importante per migliorare la qualità dell'analisi spettrale. Una delle funzioni finestra più comune è quella di **Hamming**[25], definita come:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \leq n \leq M-1$$

dove la dimensione della finestra è M per il frame $w(n)$.

4.3 Normalizzazione

Consiste nell'aggiustare l'ampiezza del segnale per garantire che i suoi valori rientrino in un intervallo specifico. Il segnale viene normalizzato per vari motivi, ad esempio rendere i dati più uniformi per l'analisi e garantire che le variazioni nella registrazione o le variazioni dell'oratore non influenzino l'output di algoritmi di analisi o classificazione. **La z-normalization** è generalmente utilizzata per la normalizzazione e si calcola come:

$$z = \frac{x - \mu}{\sigma}$$

dove μ è la media e σ è la deviazione standard del segnale vocale considerato.

4.4 Riduzione del rumore

L'ambiente è ricco di rumori, i quali si insinuano inevitabilmente in ogni segnale vocale. La presenza di rumore nel segnale può influire negativamente sulla precisione del riconoscimento delle emozioni nel parlato. Per mitigare questo problema, è fondamentale applicare tecniche di **riduzione del rumore** che possano migliorare la qualità del segnale e, di conseguenza, le prestazioni del sistema di riconoscimento.

Per ridurre il rumore in un segnale audio, è possibile utilizzare algoritmi di noise reduction, come il Minimum Mean Square Error (MMSE), oppure impiegare filtri che rimuovono specifiche frequenze indesiderate dal segnale. Questi filtri, noti come passa-banda, sono particolarmente efficaci in queste applicazioni.

In questo lavoro, per la fase di riduzione del rumore, è stato applicato un **filtro di Chebyshev** [22] con una frequenza di **taglio bassa (lowcut)** a 100 Hz e una **frequenza di taglio alta (highcut)** a 8000 Hz, poiché la voce umana si colloca principalmente all'interno di questo intervallo di frequenze, le altre frequenze indesiderate vengono eliminate dal filtro.

Capitolo 5

Features del parlato

Per descrivere una frase in formato audio attraverso dei numeri vengono utilizzate le feature del parlato. Una selezione accurata di feature può migliorare il tasso di riconoscimento delle emozioni in un sistema SER; tuttavia non esiste un insieme di feature universalmente accettate per una classificazione precisa e specifica [21].

Il parlato è un segnale continuo ma di lunghezza variabile, che trasporta sia informazioni che emozioni. A seconda delle esigenze, possiamo estrarre feature globali, locali o entrambe[31].

- Le **feature globali**, rappresentano statistiche complessive come valori di massimo, minimo, deviazione standard e media calcolate sull'intero segnale.
- Le **feature locali**, catturano le dinamiche temporali per rappresentare uno stato specifico del segnale, sono in genere calcolate su segmenti del segnale chiamati anche finestre (vedi paragrafo 4.2). L'importanza di queste feature deriva dal fatto che le emozioni non sono distribuite uniformemente lungo tutto il segnale vocale ad esempio emozioni come la rabbia sono predominanti all'inizio delle frasi [25].

Le feature globali e locali sono ulteriormente suddivise nelle due seguenti categorie:

- Feature **prosodiche**
- Feature **spettrali**

esistono altre categorie di feature che qui non vengono trattate, riporto uno studio nel caso si volesse approfondire questo argomento [30]

5.1 Feature Prosodiche

Le feature prosodiche sono quelle che può percepire l'essere umano, descrivono aspetti come l'intonazione, il ritmo e l'intensità del parlato. Queste feature sono caratteristiche di grandi unità del parlato come parole o frasi, sono per questo feature globali. Fanno parte di queste feature ad esempio il pitch, energia, ritmo.

5.1.1 Pitch

Il pitch, o frequenza fondamentale F_0, è una caratteristica prosodica che rappresenta la percezione umana della "altezza" del suono. È strettamente legato alla frequenza delle vibrazioni delle corde vocali durante la fonazione. Nella voce umana, il pitch varia per esprimere emozioni, intonazioni e significati. Inoltre la gamma media di pitch di una donna adulta va da 165 a 255 Hz, mentre quella di un uomo va da 85 a 155 Hz [23], questo può essere utile per identificare il sesso della persona che sta esprimendo una frase. Per il calcolo del pitch si può utilizzare il metodo funzione di correlazione normalizzata [2].

5.1.2 Energia

L'energia è la forza del suono percepito dalle orecchie umane. L'energia di un segnale dipende dall'ampiezza dell'onda. Se l'ampiezza del segnale è elevata, il suono risulterà forte. Nel contesto del riconoscimento delle emozioni nel parlato (SER), l'energia è una caratteristica importante perché riflette la variazione nell'intensità della voce, che può essere influenzata dallo stato emotivo. Una misura dell'energia in

una frase può essere ottenuta calcolando il valore Root Mean Square (RMS) che rappresenta l'energia complessiva in una frase e può essere calcolato come:

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}$$

Dove:

- $x[n]$ rappresenta i campioni del segnale
- N è il numero totale di campioni.

5.2 Feature Spettrali

Descrivono come l'energia del segnale vocale è distribuita tra le varie frequenze. Esse catturano informazioni sul timbro e sulla struttura acustica della voce tramite il contenuto in frequenza del segnale. Utilizzando la trasformata di Fourier, il segnale viene convertito dal dominio del tempo a quello delle frequenze, consentendo di estrarre le feature spettrali come: Mel Frequency Cepstral Coefficents (MFCC), Zero Crossing Rate (ZCR), Skewness e Kurtosi spettrale.

5.2.1 Mel Frequency Cepstral Coefficents

Questi particolari coefficienti descrivono la forma del tratto vocale, si tratta di una delle features spettrali più comunemente utilizzata nel riconoscimento automatico del parlato, utile anche nei sistemi SER. Per essere calcolati le enunciazioni vengono suddivise in vari segmenti per poi essere convertite nel dominio delle frequenze utilizzando la trasformata discreta di Fourier. Si utilizza poi un banco di filtri Mel per calcolare le energie delle sotto-bande. Successivamente, si calcola il logaritmo delle rispettive sotto-bande. Infine, gli MFCC vengono determinati applicando la trasformata inversa di Fourier

5.2.2 Zero Crossing Rate

Lo Zero-Crossing Rate (ZCR) di un segnale è definito come il tasso con cui il segnale passa da positivo a negativo, o viceversa. È una caratteristica importante per determinare la rumorosità di un segnale (ZCR alto) o individuare le parti di parlato (ZCR basso). Per calcolare questa feature si utilizza la seguente formula:

$$\text{ZCR} = \frac{1}{2M}\sum_{k=1}^{M} |\text{sign}(a[k]) - \text{sign}(a[k-1])|$$

dove M è la dimensione e $\text{sign}(a[k]) = \begin{cases} 1 & \text{se } a[k] > 0 \\ 0 & \text{se } a[k] = 0 \\ -1 & \text{se } a[k] < 0 \end{cases}$

lo ZCR trova applicazione in tutte le aree del trattamento del parlato, come sintesi vocale, miglioramento del parlato e riconoscimento vocale.

5.2.3 Skewness spettrale

La skewness spettrale è una caratteristica spettrale che misura l'asimmetria della distribuzione dello spettro di potenza di un segnale vocale rispetto a una distribuzione normale. Una skewness positiva indica che lo spettro è asimmetrico con una coda più lunga a destra, mentre una skewness negativa indica una coda più lunga a sinistra.

5.2.4 Kurtosi spettrale

La kurtosi spettrale è una misura statistica che descrive quanto sia appuntita o piatta la distribuzione dello spettro di potenza di un segnale rispetto a una distribuzione normale. Essa fornisce informazioni sulla concentrazione dell'energia spettrale del segnale, indicando se questa è distribuita in modo uniforme o se si concentra in alcune frequenze specifiche con picchi elevati.

Capitolo 6

I classificatori

6.1 Introduzione

In questo capitolo, esamineremo i classificatori utilizzati nell'analisi delle emozioni. Inizialmente si cercherà di far luce su cosa voglia dire fare **classificazione**, seguirà una presentazione degli **algoritmi** principali applicati al riconoscimento delle emozioni dal parlato e infine si parlerà delle **metriche** utilizzate per valutare la bontà dell'attività di classificazione.

6.2 Classificazione

La classificazione è un processo che consiste nell'assegnare una classe a una o un'insieme di osservazioni in base alle loro caratteristiche anche dette *features*. Questo processo nei sistemi SER viene utilizzato, dato un'input che può essere una registrazione di una frase, per produrre un output che sarà una classe emotiva (gioia, rabbia, tristezza, ecc..) a cui appartiene l'input.

Per effettuare classificazione possiamo utilizzare due approcci:

- utilizzare **classificatori tradizionali**: prima di utilizzare questi algoritmi bisogna effettuare una fase di preprocessing e feature extraction. Esempi di questi sono Support Vector Machines (SVM), K-Nearest Neighbors (KNN) e Decision Trees.
- utilizzare classificatori **deep-learning**: questi non hanno bisogno di preprocessing o feature extraction in quanto questa fase viene compiuta dagli stessi algoritmi nella fase di *training*. Esempi di questi sono: Deep Neural Network (DNN) e Convolutional Neural Network (CNN)

In questa sezione, ci focalizzeremo sui primi. Una nota da fare sugli algoritmi di deep learning però è che nonostante abbiano bisogno di più dati e di processori più potenti, negli ultimi anni la ricerca si è progressivamente orientata verso l'uso di questi algoritmi, grazie ai risultati significativi che stanno dimostrando nelle applicazioni pratiche [25].

Nonostante esistano diverse tecniche di Machine Learning [20], ci concentreremo sulle **tecniche supervisionate**, che sfruttano la conoscenza acquisita da dati etichettati per prevedere la classe di dati non ancora visti. Questo approccio inizia con la fase di training, durante la quale l'algoritmo sviluppa una funzione inferita per predire i valori di output. Durante l'addestramento, l'algoritmo confronta i risultati ottenuti con quelli reali per identificare eventuali errori e aggiornare la funzione in base ai risultati. Al termine del processo di addestramento, si ottiene il modello finale. In generale per produrre un modello di machine learning che faccia classificazione bisogna:

1. fare preprocessing dei dati
2. dividere in training set e testing set il dataset originale. Circa l'80% dei dati comporrà il training set, il restante 20% il testing set.
3. allenare il modello sul training set
4. testare il modello sul test set
5. se si è soddisfatti delle performance del modello:
   (a) rieffettuare il training su tutto il dataset

(b) distribuire il modello

altrimenti

(a) cambiare gli iperparametri del modello o cambiare algoritmo

6.2.1 Parametri e Iperparametri

I parametri e iperparametri nel Machine Learning (ML) hanno due significati diversi. I primi sono delle variabili **interne** di un modello di machine learning che vengono apprese durante la fase di training. Questi parametri definiscono il comportamento del modello e sono ottimizzati attraverso algoritmi di ottimizzazione come la discesa del gradiente. Ad esempio, nei casi di reti neurali artificiali, i parametri includono i pesi delle connessioni tra i neuroni e i bias. Gli **iperparametri** invece sono variabili esterne al modello che influenzano il processo di training e la complessità del modello stesso. Questi iperparametri non sono appresi direttamente dal modello durante il training, ma devono essere fissati prima dell'avvio del processo di training dall'utente. L'ottimizzazione degli iperparametri è cruciale per ottenere modelli di machine learning che generalizzino bene su dati non visti. Un esempio di iperparametro è quale funzione kernel utilizzare per l'algoritmo Support Vector Machine, che verrà trattato nella sezione 6.3.3.

6.2.2 Training, Validation e Test set

Come accennato nella sezione precedente durante la creazione di un modello di machine learning abbiamo bisogno di diversi subset di dati. Solitamente il dataset originale viene suddiviso in training validation e test set ma nulla impedisce di usare dataset diversi per coprire il ruolo dei diversi subset di dati.

- Il **training set** è l'insieme di dati utilizzato per addestrare il modello di machine learning. Questo set contiene esempi di input e le rispettive etichette di output corrette. Durante il processo di training, il modello utilizza il training set per apprendere le relazioni tra gli input e gli output e per ottimizzare i suoi parametri in modo da minimizzare l'errore sul training set.
- Il **test set** è un insieme separato di dati utilizzato per valutare le prestazioni del modello dopo che è stato addestrato. Questo set contiene esempi di input e le rispettive etichette di output, ma il modello non ha accesso alle etichette di output durante la fase di test. Il test set viene utilizzato per valutare quanto bene il modello generalizza su dati non visti e per stimare le sue prestazioni in situazioni del mondo reale.
- Il **validation set** è un ulteriore insieme di dati utilizzato durante il processo di sviluppo del modello per stimare le sue prestazioni e ottimizzare gli iperparametri.

Un fattore importante è che questi sottoinsiemi di dati **devono** essere disgiunti, altrimenti nel modello potrebbe verificarsi il fenomeno del **sovradattamento** questo accade quando il modello anzichè "imparare" le relazioni tra i dati "impara" le etichette a memoria e dunque ottiene percentuali di precisione altissime quando classifica istanze del set su cui è stato allenato ma quando deve valutare nuovi dati le prestazioni degradano drasticamente [24].

6.3 Algoritmi

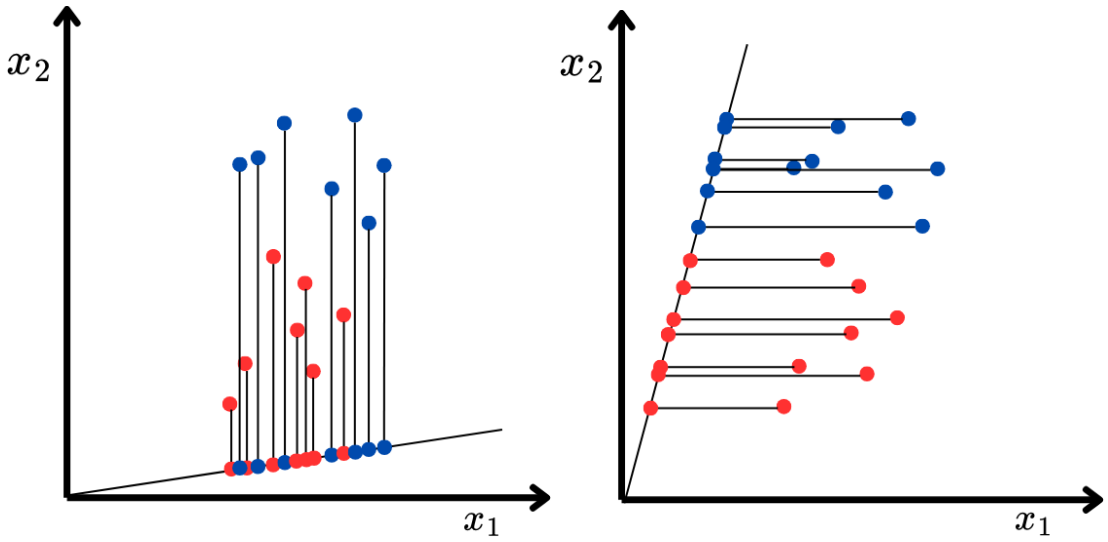
In questa sezione verranno descritti tre algoritmi di machine learning che trovano applicazione nei sistemi SER, e che successivamente verranno testati sui dataset di cui abbiamo parlato nel capitolo 3

6.3.1 Linear Discriminant Analysis (LDA)

La Linear Discriminant Analysis (LDA) è un algoritmo di machine learning utilizzato principalmente per problemi di classificazione. LDA è un algoritmo **parametrico** ovvero assume che i dati seguano una certa distribuzione, in particolare questo algoritmo assume che i dati di ogni classe seguano una distribuzione gaussiana [10]. L'obiettivo di questo algoritmo è ridurre la dimensionalità del dataset, mantenendo al contempo la separazione tra le classi tramite una particolare funzione di mappatura.

Idea di funzionamento dell'algoritmo

In questa sottosezione verrà data un'idea di funzionamento dell'algoritmo per problemi di classificazione binaria poichè la spiegazione risulterà semplice e comprensibile, si tenga a mente che il funzionamento si può estendere anche a problemi multiclasse.



(a) proiezione dove i dati si sovrappongono (b) proiezione dove i dati sono separati

Figura 6.1: differenza tra diverse proiezioni in dimensioni minori per la separazione dei dati

Come già accennato prima LDA cerca di proiettare i dati in una dimensione minore per effettuare la classificazione, tuttavia ci sono diversi modi di mappare i dati di partenza in uno spazio di dimensione minore, in fig. 6.1 viene evidenziata la differenza tra tra due funzioni di mappatura, una efficace per la classificazione (fig. 6.1b) e una inefficace (fig. 6.1a) dove i dati si sovrappongono rendendo difficile il compito di classificazione. Per trovare una mappatura efficace LDA tiene conto di due principi fondamentali:

1. **massimizzare la distanza tra le medie** delle classi che si vogliono separare
2. **minimizzare la variazione** (in LDA chiamata anche scatter e rappresentata da s^2) all'interno di ciascuna classe

Funzionamento Dell'algoritmo

L'algoritmo LDA segue questi passaggi fondamentali:

1. Calcolo della media per ogni classe: viene calcolata la media delle features μ_j per ciascuna classe nel j dataset.

$$\mu_j = \frac{1}{n_j} \sum_{x_i \in \ \omega j} x_i$$

   Dove:

   - n_j rappresenta il **numero di campioni** appartenenti alla classe j.
   - x_i è il **vettore delle feature** per il campione i.
   - ωj è l'**insieme** dei campioni appartenenti alla classe j.

2. Calcolo della varianza inter-classe: misura la distanza tra le medie delle diverse classi.

$$\left|\mu_1 - \mu_2\right|^2$$

3. Calcolo della varianza intra-classe: misura la dispersione dei dati all'interno di ciascuna classe, ovvero quanto i dati di una classe sono vicini alla loro media.

$$S_i^2 = \sum_{y \in \omega_i} (y - \widetilde{\mu}_i)^2$$

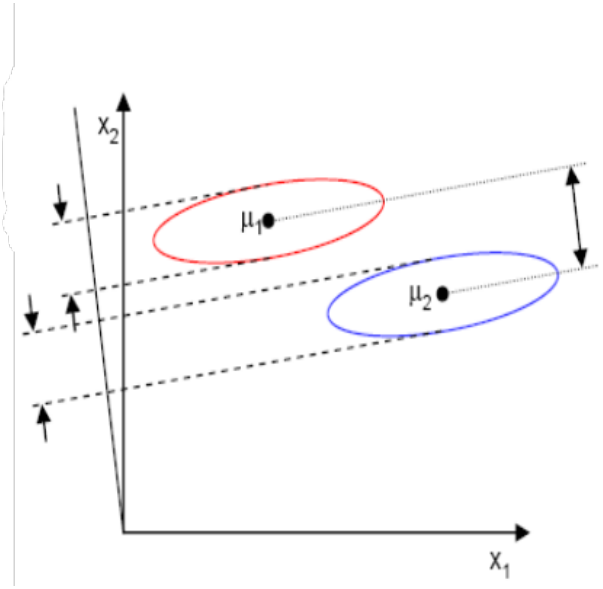


Figura 6.2: Visualizzazione del criterio per trovare l'iperpiano di dimensione minore che meglio separa le due classi

4. Trovare una funzione $w^T\mathbf{x}$ che massimizzi la funzione obiettivo:

$$J(w) = \frac{|\mu_1 - \mu_2|^2}{S_1^2 + S_2^2}$$

pertanto, cercheremo una proiezione in cui gli esempi appartenenti alla stessa classe siano proiettati molto vicini tra loro e, allo stesso tempo, le medie proiettate siano il più distanti possibile [6], vedi figura 6.2.

Quando utilizzare LDA

LDA è particolarmente utile quando **le classi sono separabili linearmente**, infatti questo algoritmo relativamente semplice è computazionalmente poco costoso ma potente nel fare separazioni lineari. Inoltre si può considerare di utilizzare LDA quando è necessaria una riduzione della dimensionalità per visualizzare meglio i dati o per migliorare l'efficienza di altri algoritmi di classificazione.

Limiti di LDA

Nonostante la sua efficacia in contesti ben definiti, LDA ha alcuni limiti:

- **Non funziona bene se le classi non seguono una distribuzione gaussiana**: se le distribuzioni sono significativamente non gaussiane, le proiezioni LDA non saranno in grado di preservare alcuna struttura complessa dei dati, che potrebbe essere necessaria per la classificazione.
- **Può risultare inefficace se le classi non sono separabili linearmente** questo perchè LDA viene utilizzato per trovare una trasformazione lineare che discrimini tra diverse classi. Tuttavia, se le classi non sono separabili in modo lineare, LDA non riesce a trovare uno spazio dimensionale inferiore che le separi [10].

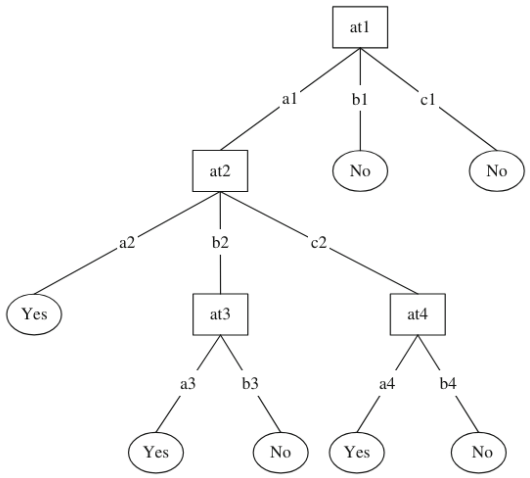
6.3.2 Decision Trees (DT)

Gli alberi decisionali sono modelli di apprendimento supervisionato utilizzati per la classificazione e la regressione. Questo algoritmo, per fare classificazione, costruisce una struttura ad albero dove sono presenti due tipi di nodi:

- **i nodi di decisione**: sono nodi nell'albero in cui si esegue un test semplice su una caratteristica del dato
- **i nodi foglia**: sono nodi dell'albero a cui si giunge dopo essere passati per uno o più nodi di decisione, questi nodi determinano la classe dell'osservazione da classificare.

at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

(a) training set d'esempio



(b) decision tree d'esempio

Figura 6.3: Training set e il suo albero di decisione

Un esempio di albero di decisione con il suo rispettivo training set è osservabile nella figura 6.3 in cui ogni nodo rettangolare è un nodo di decisione che rappresenta la feature che stiamo testando mentre i nodi ovali sono i nodi foglia che contengono le classi. Per classificare un nuovo dato basta seguire i test partendo dal nodo radice fino ad arrivare ad un nodo foglia, come si può intuire questo algoritmo presenta un vantaggio rispetto ai modelli "black-box", come le reti neurali, in termini di **comprensibilità**. Le regole logiche seguite da un albero decisionale sono molto più facili da interpretare rispetto ai pesi numerici delle connessioni tra i nodi in una rete neurale [11].

Per costruire un albero di decisione di solito si parte da un training set e un albero vuoto e per ogni nodo di decisione si seleziona la caratteristica "appropriata" da testare utilizzando una **misura di selezione.** Il principio è quello di selezionare la caratteristica che minimizza il mix di classi del sottoinsieme di dati creato dal test, rendendo quindi man mano più semplice l'identificazione della classe a cui appartengono i dati. Questo processo continua per ogni sotto albero di decisione fino a raggiungere le foglie e assegnare loro la corrispondente classe.

L'Entropia

L'entropia **è una misura del disordine o dell'impurità nel set di dati fornito**. Come detto in precedenza ad ogni nodo di decisione si cerca di suddividere i dati in modo da renderli più omogenei (appartenenti ad una stessa classe), si punta quindi a diminuire **l'entropia** . Per un dataset che ha C classi e ha probabilità Pi di scegliere casualmente i dati dalla classe i, l'entropia si calcola come:

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

le misure di selezione sfruttano l'entropia e sono le seguenti:

Information Gain

$$\text{Gain } (S,\ A) = \text{Entropy } (\mathbf{S}) - \sum(|\text{S}_\text{v}|/|\text{ s}|) \text{ Entropy } (\text{S}_\text{v})$$

Gain Ratio

$$Gain\ \ Ratio\ (S, A) = \text{Gain(S, A)/SplitInformation (S, A)}$$

$$SplitInformation\ (\text{S, A}) = \textstyle\sum_{i=1}^{c}(|\text{S}_i|/|\text{S}|) \log_2 (|\text{S}_i|/|\text{S}|)$$

Gini value:

$$Gini\,(D) = 1 - \sum_{j=1}^{n} p_j^2$$

dove p_j è la frequenza relativa della classe j in D.

Uno studio più approfondito delle misure di selezione può essere trovato qui [4]

6.3.3 Support Vector Machine (SVM)

Le macchine a vettori di supporto (SVM, dall'inglese Support Vector Machines) rappresentano un metodo di apprendimento supervisionato efficace per la classificazione binaria e la regressione. Le SVM si basano sul principio di separazione ottimale: il loro obiettivo è trovare l'iperpiano lineare che meglio separa le classi di dati nel feature space. Questo iperpiano è definito in modo da rispettare due condizioni:

1. l'errore di classificazione deve essere minimizzato
2. la distanza dell'iperpiano dall'istanza più vicina di ciascuna classe (detti supporting vectors) dovrebbe essere massima

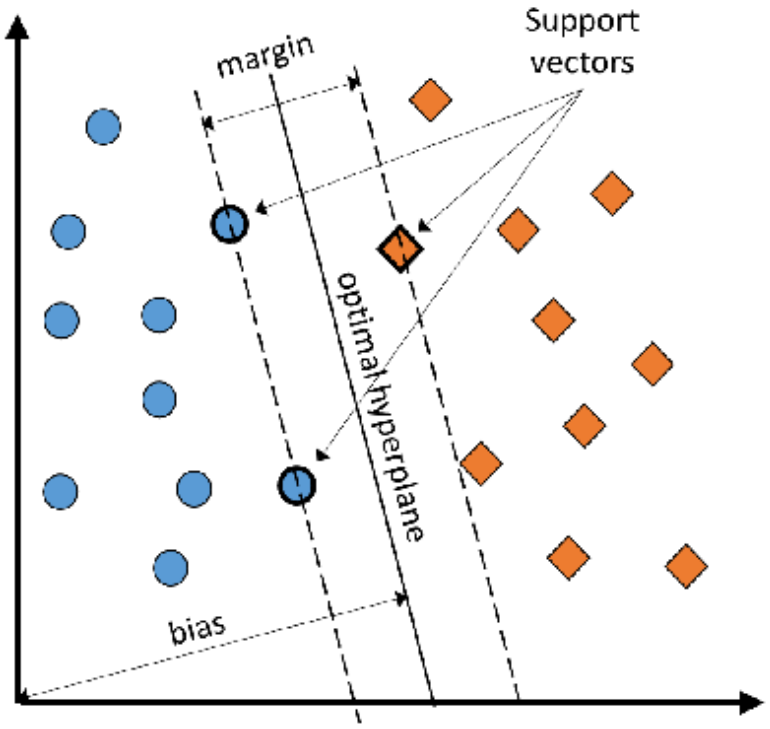


Figura 6.4: l'iperpiano ottimale di separazione tra due classi

se i dati sono linearmente separabili e appartengono a due sole classi è facile trovare una separazione netta, ma nella maggior parte dei casi i dati non saranno separabili linearmente e vorremmo poter fare classificazione multiclasse, vedremo ora delle tecniche per risolvere questi problemi.

La funzione Kernel

La funzione kernel permette di ottenere una separazione anche tra dati che NON sono linearmente separabili, questa tecnica consiste nel mappare i vettori delle feature in uno spazio di dimensione maggiore dove i dati sono linearmente separabili, a questo punto resta solo da trovare l'iperpiano che separa le due classi. In figura 6.5 abbiamo una visualizzazione di come una funzione kernel potrebbe mappare i dati di input $z_i \rightarrow \varphi(z_i)$

esistono varie funzioni kernel che non vengono trattate, uno studio più approfondito si può trovare qui [12]

Approcci alla classificazione multiclasse usando SVM

SVM non supporta la classificazione multiclasse in modo nativo. Supporta la classificazione binaria e la separazione dei punti dati in due classi. Per la classificazione multiclasse,viene utilizzato lo stesso principio nativo dopo aver scomposto il problema della multiclassificazione in più problemi di classificazione binaria.

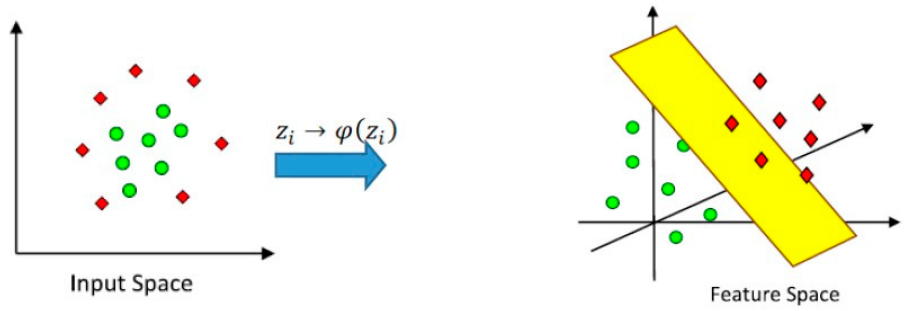
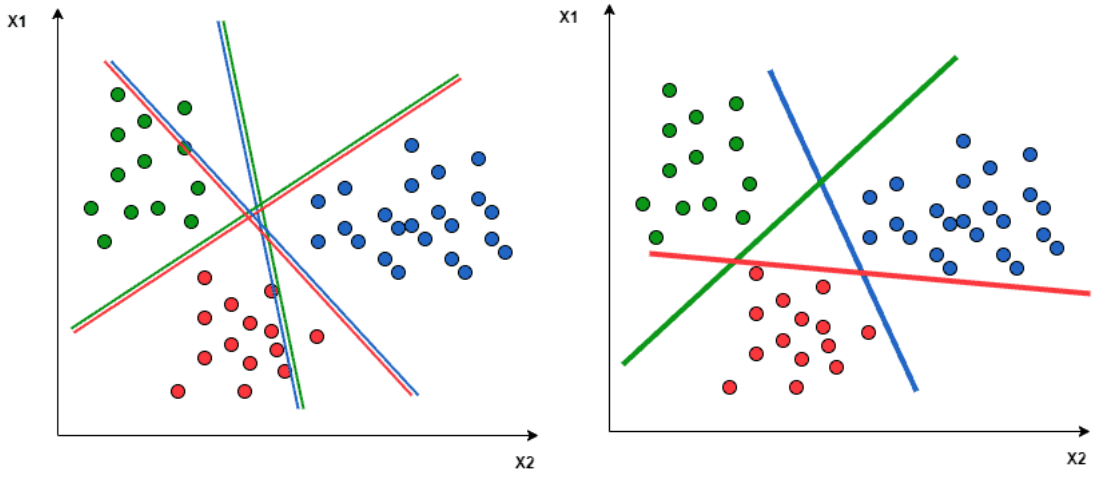


Figura 6.5: Trasformazione delle caratteristiche di input da una dimensione inferiore a una dimensione superiore

L'approccio *One-to-One*, o uno-a-uno, scompone il problema multiclasse in più problemi di classificazione binaria, in questo caso si vuole trovare un iperpiano che separi due classi, trascurando i punti delle altre classi che non stiamo prendendo in considerazione. Ciò significa che la separazione tiene conto solo dei punti delle due classi nel frazionamento attuale, viene utilizzato un classificatore SVM per ogni frazionamento.

Un altro approccio che è possibile utilizzare è *One-to-Rest* o uno-a-tutti. In questo approccio abbiamo bisogno di un iperpiano per separare una classe da tutte le altre contemporaneamente. Ciò significa che la separazione tiene conto di tutti i punti, dividendoli in due gruppi; un gruppo per i punti della classe che stiamo considerando e un gruppo per tutti gli altri punti, viene utilizzato un classificatore per ogni classe contro tutte le altre.



(a) approccio one-to-one

(b) approccio one-to-rest

Figura 6.6: rappresentazione intuitiva degli approcci presentati per la classificazione multiclasse usando delle SVM

6.4 Metriche di Valutazione delle performance

In questa sezione verranno discusse le metriche più importanti per valutare le performance dei modelli per quanto riguarda problemi di classificazione; inizialmente verranno trattate le metriche per la classificazione binaria che verranno poi generalizzate per valutare modelli di classificazione multiclasse.

6.4.1 La matrice di confusione

La matrice di confusione è una **tabella incrociata** che registra il numero di occorrenze tra due valutatori, la classificazione vera/effettiva e la classificazione prevista. In questo documento vengono rappresentate sulle righe le classificazioni reali, mentre sulle colonne sono rappresentate le classi che il modello ha predetto, un esempio di matrice di confusione è osservabile nella figura 6.8.

		PREDICTED		
	Classes	Positive (1)	Negative (0)	Total
ACTUAL	Positive (1)	TP = 20	FN = 5	25
	Negative (0)	FP = 10	TN = 15	25
	Total	30	20	**50**

Figura 6.7: matrice di confusione con 2 classi

		PREDICTED classification				
	Classes	a	b	c	d	Total
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
	Total	11	11	13	17	**52**

Figura 6.8: matrice di confusione multiclasse

Le classi sono elencate nelle righe e nelle colonne nello stesso ordine, quindi gli elementi classificati correttamente sono situati sulla diagonale principale, qui si trovano i valori dove la classe predetta e la classe effettiva corrispondono.

6.4.2 Precision (precisione)

Partendo dalla matrice di confusione con due classi in figura 6.7 la precisione è definita come:

$$Precision = \frac{TP}{TP + FP}$$

La precisione esprime la proporzione di unità che secondo il nostro modello sono positive e che in realtà sono positive. In altre parole, la precisione ci dice quanto possiamo fidarci del modello quando prevede che un individuo sia positivo [26].

6.4.3 Recall (richiamo)

Il richiamo è definito come:

$$Recall = \frac{TP}{TP + FN}$$

Il Recall misura l'accuratezza predittiva del modello per la classe positiva: intuitivamente, misura la capacità della classe positiva modello per trovare tutte le unità positive nel set di dati.

6.4.4 Accuracy (accuratezza)

L'accuratezza è una delle metriche più popolari nella classificazione multiclasse ed è calcolata direttamente dalla matrice di confusione. In riferimento alla figura 6.7 l'accuratezza è calcolata come:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

L'accuratezza ci dice la percentuale di volte in cui la previsione del modello è corretta se scegliessimo di prendere un dato reale a caso e di prevederne la sua classe.

Questa metrica, sebbene di facile interpretazione, non tiene conto dell'equilibrio tra le classi. In alcuni casi, alcune classi potrebbero essere più numerose di altre e potrebbero essere facilmente identificate dal modello predittivo; ciò potrebbe portare a un'alta percentuale di accuratezza complessiva, anche se il modello si comporta bene solo su alcune classi. Pertanto, se desideriamo che il nostro modello sia efficace su tutte le classi, è necessario valutare questa metrica insieme ad altre già menzionate, al fine di ottenere una valutazione completa delle prestazioni del modello.

6.4.5 F1-Score

Anche F1-Score valuta le prestazioni del modello di classificazione partendo dalla matrice di confusione, relaziona la precisione e il richiamo secondo il concetto di media armonica.

$$\text{F1} - \text{Score} = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall} \right)$$

La formula del f1-score può essere interpretata come una media ponderata tra Precisione e Richiamo, dove l' f1-score raggiunge il suo miglior valore in 1 e il suo peggiore in 0. Il contributo relativo di Precisione e Richiamo è uguale e la media armonica è utile per trovare il miglior compromesso tra le due quantità.

6.4.6 F1-score per problemi multiclasse

Quando si tratta di casi multiclasse, l'f1-score dovrebbe coinvolgere tutte le classi. Per fare ciò, abbiamo bisogno di una misura multiclasse di Precisione e Richiamo da inserire nella media armonica. Queste due metriche possono essere definite in modo differente nel caso multiclasse, dando così origine a due f1-score diverse.

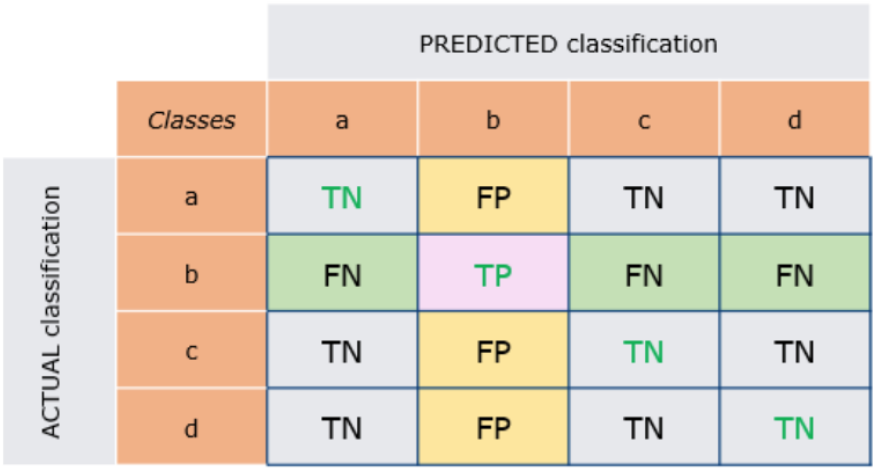


Figura 6.9: Precisione e Recall per problemi multiclasse con riferimento alla classe *b*

Per i calcoli richiesti, utilizzeremo la Matrice di Confusione concentrandoci su una classe alla volta ed etichettando le celle di conseguenza. In particolare, consideriamo True Positive (TP) le uniche celle correttamente classificate per la nostra classe, mentre Falsi Positivi (FP) e Falsi Negativi (FN) sono gli elementi erroneamente classificati sulla colonna e sulla riga della classe rispettivamente. Veri Negativi (TN) sono tutte le altre celle finora non considerate, come mostrato nella Figura 6.9 dove consideriamo la classe "b" come focus di riferimento.

Macro F1-score

Precisione e Richiamo per ciascuna classe vengono calcolati utilizzando le stesse formule del caso binario dopo aver ottenuto la CF rappresentata nella figura 6.9. Le seguenti formule rappresentano le due quantità per una generica classe k.

$$Precision_k = \frac{TP_k}{TP_k + FP_k}$$

$$Recall_k = \frac{TP_k}{TP_k + FN_k}$$

La macro precisione media e il richiamo vengono calcolati come media aritmetica di precisione e richiamo per le singole classi.

$$MacroAveragePrecision = \frac{\sum_{k=1}^{K} Precision_k}{K}.$$

$$MacroAwerageRecall = \frac{\sum_{k=1}^{K} Necall_k}{K}$$

Infine otteniamo la macro F1-score con:

$$MacroF1 - Score = 2 * (\frac{MacroAveragePrecision * MacrodwerageRecall}{MacroAveragePrecision^{-1} + MacrodverageReecall^{-1}})$$

I metodi di Macro-Media tendono a calcolare una media complessiva di diverse misure, poiché i numeratori della Macro Precisione Media e del Richiamo Medio sono composti da valori nell'intervallo [0, 1] allora non c'è alcun collegamento con le dimensioni delle classi. Perché le classi di dimensioni diverse sono ponderate allo stesso modo al numeratore. Questo significa che l'effetto sulla misura delle classi più grandi ha la stessa importanza delle più piccole. La metrica ottenuta valuta l'algoritmo da un punto di vista delle classi: valori elevati di Macro-F1 indicano che l'algoritmo ha buone prestazioni su tutte le classi, mentre valori bassi di Macro-F1 si riferiscono a classi previste in modo insoddisfacente.

Micro F1-score

Per ottenere il Micro F1-Score, dobbiamo calcolare prima la Micro-Precision e la Micro-Recall. L'idea della micro-media è quella di considerare tutte le unità insieme, senza prendere in considerazione eventuali differenze tra classi. Pertanto, la Micro Precisione Media e il Micro Richiamo Medio vengono calcolati nel seguente modo:

$$Micro\ Average\ Precision = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} Total\ Column_k} = \frac{\sum_{k=1}^{K} TP_k}{Grand Total}$$

$$Micro\ Average\ Recall = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} \mathcal{T}otal\ Row_k} = \frac{\sum_{k=1}^{K} TP_k}{Grand Total}$$

come si può osservare, effettuando i calcoli le misure sono identiche, se prima nel calcolo delle macro-misure precisione e richiamo differivano solo per il denominatore ora condividono lo stesso. la media armonica di due valori uguali è il valore segue che la micro F1-score è identica alla Micro-precisione e al Micro-richiamo

$$Micro\ Average\ F1 = \frac{\sum_{k=1}^{K} TP_k}{Grcand\ Total}$$

Dando un'occhiata alla formula, possiamo osservare che il Micro F1-Score è proprio uguale alla precisione. Quindi, pro e contro sono condivisi tra le due misure. Entrambi danno più importanza alle classi grandi, perché considerano tutte le unità insieme.

6.4.7 Coefficiente di correlazione di Matthews (MCC)

Questa metrica di valutazione è concepita per valutare correttamente anche modelli addestrati su dataset non bilanciati. Questa formula si comporta come un coefficiente di correlazione. Essa va quindi da +1 a -1 dove +1 indica una perfetta previsione, 0 indica una previsione casuale e -1 indica una previsione totalmente sbagliata. Nel caso binario si calcola come:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}}$$

Capitolo 7

Proposta di un Sistema SER: Valutazione Fair e Approccio Multilingua

7.1 Introduzione al progetto

In questo capitolo viene presentato il progetto di sviluppo di un sistema di Speech Emotion Recognition (SER), con un duplice obiettivo: promuovere l'adozione di **valutazioni eque (fair)** degli algoritmi, e affrontare le sfide tecniche legate alla creazione di un sistema SER **multilingua**. Il riconoscimento delle emozioni dal parlato rappresenta una delle frontiere più affascinanti e complesse nell'ambito dell'intelligenza artificiale e delle interfacce uomo-macchina,**con potenziali applicazioni** in settori quali il marketing, la sanità, e la sicurezza stradale. Tuttavia, la letteratura scientifica evidenzia ancora numerosi **ostacoli**, soprattutto per quanto riguarda l'accuratezza in contesti reali e multilingua.

Il progetto qui proposto si inserisce in questo contesto, cercando di colmare due lacune rilevanti. In primo luogo, si propone un approccio "fair" per la valutazione degli algoritmi di riconoscimento delle emozioni, evitando **bias** comuni come l'inclusione dello stesso soggetto sia nella fase di training che di testing, una pratica che può portare a risultati non rappresentativi delle reali capacità del modello. In secondo luogo, il lavoro esplora la sfida del multilinguismo, progettando e implementando un sistema SER capace di riconoscere emozioni sia in contesti monolingua che multilingua, con l'obiettivo di analizzare come la variabilità linguistica influisca sulla capacità del modello di **generalizzare**.

Il capitolo è strutturato in diverse sezioni che coprono in modo approfondito la progettazione di sistemi SER, verranno viste nel dettaglio le **metodologie** e le **scelte implementative** che hanno portato alla creazione di modelli monolingue e multilingue. Inoltre, viene discusso in dettaglio il metodo di valutazione adottato, evidenziando come una corretta valutazione possa incidere significativamente sull'efficacia del sistema. I risultati ottenuti saranno prima discussi in quanto a metriche quali **accuratezza** e **precisione** esaminando se i modelli creati producono risultati soddisfacenti o non accettabili. In secondo luogo i risultati verranno messi a confronto con le valutazioni umane, queste faranno da **benchmark** delle valutazioni per i modelli prodotti partendo dall'ipotesi che una macchina non riesca a superare il tasso di riconoscimento delle emozioni delle persone umane. Infine verranno fornite **indicazioni** su come questi risultati possano contribuire a migliorare le future implementazioni di sistemi SER.

7.2 Software utilizzato e Bibliografia correlata

Per lo sviluppo dei modelli, è stato scelto di utilizzare il linguaggio Python [7], insieme a librerie come Scikit-learn [9] per la costruzione degli algoritmi di machine learning e Pandas [28] per la gestione e l'elaborazione dei dati. La fase di estrazione delle feature è stata invece condotta utilizzando MATLAB [32].

Il codice sviluppato è stato gestito tramite il sistema di versionamento Git ed è stato pubblicato su GitHub, nella repository disponibile al seguente link.

Tutte le analisi effettuate su questa tesi riguardano i due dataset descritti nella sezione 3 (Dataset) EMOVO e RAVDESS.

Per la parte bibliografica, il lavoro ha seguito il seguente riferimento per quanto riguarda alcune delle feature e degli algoritmi impiegati nello sviluppo dei modelli:

- Feature extraction algorithms to improve the speech emotion recognition rate [27].

7.3 Preprocessing dei segnali audio

La prima fase del progetto è stata dedicata al preprocessing dei segnali audio per garantire la coerenza e la qualità dei dati su cui basare le successive analisi. Un passo fondamentale è stato l'uniformazione del sampling rate di tutti i segnali a **48.000 Hz**. Questa scelta è motivata dal fatto che 48.000 Hz è uno standard utilizzato in molte applicazioni audio professionali, fornendo una buona risoluzione temporale e frequenziale senza introdurre eccessiva ridondanza nei dati.

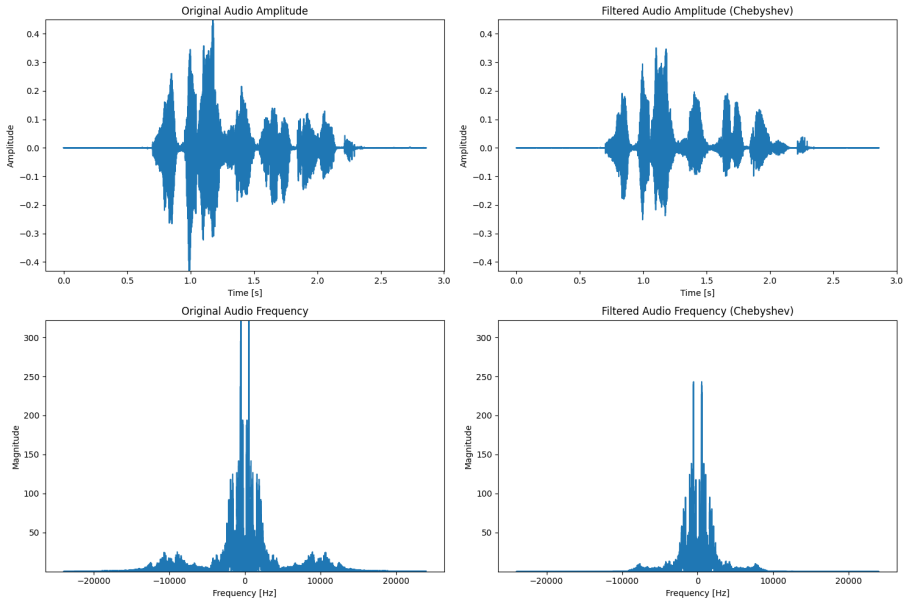


Figura 7.1: preprocessing del segnale audio *dis-f1-b1.wav* di emovo

Successivamente, è stato applicato un filtro di Chebyshev di tipo I per eliminare rumori indesiderati e limitare le frequenze rilevanti. Il filtro è stato configurato con una banda passante compresa tra **100 Hz** e **8000 Hz**, valori scelti in modo da catturare la gamma di frequenze più rilevante per il parlato umano. nella fig. 7.1 viene mostrato un segnale audio prima e dopo la fase di preprocessing.

7.4 Estrazione delle caratteristiche

Dopo la fase di preprocessing, si è proceduto con l'estrazione delle feature dai segnali audio. Le feature acustiche estratte includono:

- **Pitch**
- **Energia**
- **Zero-Crossing Rate (ZCR)**
- **Spectral Skewness e Spectral Kurtosis**

Le feature sopra citate sono state estratte lungo tutto il segnale audio, quindi non è stata applicata nessuna finestratura ne framing. Oltre a queste, sono stati estratti i primi 13 **Mel-Frequency Cepstral Coefficients (MFCCs)**, essi rappresentano una descrizione compatta delle caratteristiche spettro-temporali del segnale. Per ciascuno dei 13 coefficienti MFCC, sono stati calcolati i seguenti parametri statistici:

- **Minimo**
- **Massimo**
- **Media**
- **Mediana**

- **Deviazione standard**

Questi parametri statistici permettono di rappresentare efficacemente la variabilità dei coefficienti MFCC senza dover applicare ulteriori selezioni. Per l'estrazione dei coefficienti MFCC, il segnale è stato suddiviso in frame attraverso due fasi: framing e windowing.

Durante la fase di **framing**, il segnale è stato diviso in finestre temporali di 25 ms, con una sovrapposizione del 50%, per garantire uno stato quasi stazionario del segnale [27].

Successivamente, nella fase di **windowing**, è stata applicata una finestra a ciascun frame per ridurre le discontinuità ai bordi del segnale e minimizzare la distorsione spettrale. È stata utilizzata la **finestra di Hamming** [1] , poiché riduce le distorsioni e migliora la precisione dello spettro in frequenza. La finestra di Hamming è data da:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \leq n \leq M-1$$

dove M rappresenta il numero di campioni in un singolo frame.

7.5 Creazione dei file CSV

Dopo la fase di feature extraction, i dati sono stati organizzati in file CSV numerici, in modo da poter essere usati con algoritmi di Machine Learning. Ogni file CSV include, come prima colonna, l'identificativo dell'attore che ha pronunciato la frase, mentre l'ultima colonna contiene l'etichetta corrispondente all'emozione rilevata. La mappa utilizzata per associare i numeri alle emozioni è riportata nella tabella 7.1:

Numero	**Emozione**
1	Neutro (neu)
2	Calmo (calm)
3	Felice (happy)
4	Triste (sad)
5	Arrabbiato (ang)
6	Spaventato (fear)
7	Disgusto (disgust)
8	Sorpreso (surprise)

Tabella 7.1: Mappa delle etichette per le emozioni.

7.6 Creazione dei modelli di classificazione monolingua

In questa fase dello sviluppo, sono stati costruiti modelli di classificazione monolingua, ossia modelli addestrati e testati su un dataset in cui le frasi pronunciate appartengono alla stessa lingua. I modelli sono stati creati e ne sono state valutate le performance utilizzando due approcci di validazione distinti. Inizialmente, gli algoritmi di Machine Learning citati in precedenza (SVM, Decision Trees, LDA) sono stati addestrati, con un'ottimizzazione degli iperparametri sui rispettivi dataset, impiegando una semplice cross-validation (CV). Va sottolineato che questo tipo di validazione consente che lo stesso soggetto possa apparire sia nel training set.

Successivamente, i modelli sono stati costruiti nello stesso modo, ma utilizzando un approccio di validazione Leave-One-Subject-Out (LOSO), che garantisce che lo stesso soggetto non sia mai presente sia nel set di addestramento che in quello di test. Questi modelli sono ottenuti addestrando l'algoritmo su tutto il dataset escludendo un attore, e testandolo sull'attore lasciato fuori, ripetendo questo procedimento per tutti gli attori le prestazioni sono calcolate come media delle prestazioni su ogni attore.

Per questo lavoro di classificazione sono state selezionate quattro emozioni principali: neutra, felicità, tristezza e rabbia. Tuttavia, i dataset utilizzati includono anche altri stati emotivi, come sorpresa, paura e disgusto. Poiché l'obiettivo era concentrarsi esclusivamente sulle quattro emozioni selezionate, le altre sono state escluse dall'analisi. Di conseguenza, il numero di campioni disponibili è stato ridotto a 336 per il dataset EMOVO e a 672 per il dataset RAVDESS.

Passiamo ora all'analisi delle prestazioni ottenute con ciascun metodo di validazione.

7.6.1 Risultati dei modelli

Di seguito riporto i risultati dei modelli su entrambi gli approcci proposti per tutti e tre gli algoritmi

Approccio con Cross-Validation

- **I risultati ottenuti su EMOVO** con questo approccio si possono osservare attraverso le metriche di valutazione in tabella 7.2 e le matrici di confusione in figura A.1
- **I risultati ottenuti su RAVDESS** si possono osservare attraverso le metriche di valutazione in tabella 7.3 e le matrici di confusione in figura A.2

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.91	0.91	0.91	0.91
Decision Tree	0.56	0.56	0.55	0.55
LDA	0.71	0.71	0.72	0.71

Tabella 7.2: metriche di valutazione per la classificazione su EMOVO con CV

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.83	0.81	0.80	0.80
Decision Tree	0.54	0.53	0.52	0.52
LDA	0.71	0.75	0.71	0.72

Tabella 7.3: metriche di valutazione per la classificazione su RAVDESS con CV

Approccio con Validazione LOSO

- **I risultati ottenuti su EMOVO** con questo approccio si possono osservare attraverso le metriche di valutazione in tabella 7.4 e le matrici di confusione in figura A.1
- **I risultati ottenuti su RAVDESS** con questo approccio si possono osservare attraverso le metriche di valutazione in tabella 7.5 e le matrici di confusione in figura A.2

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.40	0.44	0.40	0.34
Decision Tree	0.38	0.37	0.38	0.35
LDA	0.42	0.46	0.42	0.37

Tabella 7.4: metriche di valutazione per la classificazione su EMOVO con validazione LOSO

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.67	0.71	0.67	0.65
Decision Tree	0.56	0.57	0.56	0.53
LDA	0.65	0.67	0.65	0.63

Tabella 7.5: metriche di valutazione per la classificazione su RAVDESS con validazione LOSO

Discussione dei Risultati

Come si può osservare dai risultati, le prestazioni dei modelli costruiti con validazione *Leave-One-Subject-Out* (LOSO) sono sensibilmente **più basse** rispetto a quelle ottenute con la cross-validation. Questo fenomeno può essere attribuito al fatto che, nella cross-validation, lo stesso soggetto può apparire sia nel training set che nel test set. Ciò introduce un bias che influisce negativamente sulla valutazione delle capacità di generalizzazione del modello, portando a prestazioni apparentemente migliori, ma poco realistiche.

D'altra parte, con la validazione LOSO, il modello si trova ogni volta ad affrontare un attore che non ha mai visto durante l'addestramento. In questo scenario, le prestazioni tendono a diminuire, ma riflettono con maggiore accuratezza la capacità del modello di **generalizzare** su casi nuovi e mai visti.

Un'osservazione interessante riguarda le prestazioni ottenute con il dataset EMOVO, dove si nota un significativo **crollo** delle performance nel caso di validazione LOSO. Questo potrebbe essere dovuto al fenomeno dell'**overfitting**, che si manifesta nei modelli costruiti tramite cross-validation, i quali tendono a classificare correttamente solo i dati che hanno già visto, fallendo quando si tratta di generalizzare su nuovi esempi.

Un altro fattore da considerare è la dimensione del dataset EMOVO, che è relativamente piccola. Questo potrebbe ulteriormente spiegare le scarse prestazioni osservate. A sostegno di questa ipotesi, possiamo notare come invece le prestazioni su RAVDESS, pur calando nel caso della validazione LOSO, non siano crollate drasticamente, mantenendosi comunque su livelli accettabili con accuracies intorno al **65-67%** per gli algoritmi SVM e LDA.

In definitiva, questo risultato rappresenta un'importante indicazione: nonostante il calo nelle prestazioni, stiamo valutando il modello in maniera *fair*, ottenendo quindi una stima più **realistica** delle sue reali capacità di generalizzazione.

7.7 Creazione dei modelli di classificazione multilingua

In questa sezione viene presentato l'approccio adottato per la creazione e valutazione di un sistema di riconoscimento delle emozioni in ambito multilingua. I dataset utilizzati, EMOVO per l'italiano e RAVDESS per l'inglese, e gli algoritmi di classificazione (SVM, Decision Trees, LDA) rimangono invariati rispetto a quanto descritto nella sezione precedente.

Il primo obiettivo è stato valutare la capacità dei modelli addestrati su una lingua di generalizzare a un'altra lingua, al fine di comprendere quanto i pattern acustici delle emozioni siano universali o, al contrario, dipendenti dalla lingua specifica. In particolare, si è esaminata la performance dei modelli addestrati su una lingua (ad esempio, l'italiano) e testati sull'altra (l'inglese) e viceversa. Questo esperimento ha fornito una prima indicazione sulla robustezza dei modelli alla lingua e ha evidenziato le eventuali limitazioni nell'adattamento delle caratteristiche emotive da una lingua all'altra.

Successivamente, è stato creato un dataset combinato ad hoc per il riconoscimento multilingua. Questo dataset a cui ci riferiremo come COMBINED è frutto dell'unione dei due dataset (EMOVO e RAVDESS), si è proseguito in questa direzione per simulare un contesto più realistico, in cui il sistema deve essere in grado di riconoscere le emozioni in più lingue contemporaneamente. Questo nuovo dataset multilingua è stato utilizzato per addestrare e testare i modelli, anche in questo caso per la fase di valutazione si sono voluti applicare i due diversi approcci di validazione: la cross-validation (CV) e la Leave-One-Subject-Out (LOSO) per gli stessi motivi esaminati nella costruzione di sistemi monolingua 7.6.

7.7.1 Risultati dei modelli addestrati e testati con dataset di lingua diversa

- **I risultati ottenuti addestrando gli algoritmi con EMOVO e testando su RAVDESS** sono visibili in tabella 7.6 mentre le matrici di confusione sono riportate nella figura A.3
- **I risultati ottenuti addestrando gli algoritmi con RAVDESS e testando su EMOVO** sono visibili in tabella 7.7 mentre le matrici di confusione sono riportate nella figura A.4

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.39	0.32	0.34	0.27
Decision Tree	0.27	0.22	0.24	0.21
LDA	0.28	0.30	0.26	0.24

Tabella 7.6: Metriche di valutazione dei modelli addestrati con EMOVO (ITA) e testati su RAVDESS (ENG)

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.30	0.25	0.30	0.24
Decision Tree	0.31	0.30	0.31	0.27
LDA	0.28	0.29	0.28	0.25

Tabella 7.7: Metriche di valutazione dei modelli addestrati con RAVDESS (ENG) e testati su EMOVO (ITA)

Discussione dei risultati per i modelli addestrati e testati su dataset di lingua diversa

Dai risultati riportati nelle tabelle 7.6 e 7.7, si può osservare che le prestazioni dei modelli addestrati su un dataset in una lingua e testati su un dataset in un'altra lingua risultano significativamente inferiori rispetto ai modelli monolingua. Ad esempio, i modelli addestrati su EMOVO (italiano) e testati su RAVDESS (inglese) mostrano un'accuratezza massima del 39% con SVM, mentre con Decision Tree e LDA le prestazioni calano ulteriormente, raggiungendo valori rispettivamente del 27% e del 28%. In modo simile, quando i modelli sono stati addestrati su RAVDESS e testati su EMOVO, si osservano performance altrettanto limitate, con un'accuratezza massima del 31% ottenuta con Decision Tree.

Questi risultati evidenziano le difficoltà nel generalizzare da una lingua all'altra, suggerendo che i pattern acustici delle emozioni, sebbene possano presentare alcune caratteristiche universali, sono fortemente influenzati dalla lingua in cui vengono espressi, inoltre le features scelte per questo lavoro potrebbero essere non totalmente appropriate in quanto non mostrano una robustezza alla lingua. Le differenze linguistiche possono influenzare aspetti come l'intonazione, il ritmo e il timbro vocale, tutti elementi che come per gli umani contribuiscono alla percezione delle emozioni e che i modelli di machine learning, addestrati su una singola lingua, faticano a riconoscere correttamente in un contesto diverso.

Questa analisi suggerisce che, per migliorare le prestazioni in un contesto multilingua, è necessario sviluppare tecniche specifiche per affrontare le differenze linguistiche, in particolare cercheremo di sviluppare un modello multilingua includendo più lingue nella fase di training dei modelli.

7.7.2 Risultati dei modelli sul dataset multisource

Di seguito riporto i risultati ottenuti su COMBINED con i due approcci di validazione già citati

Approccio con Cross-Validation

- **I risultati ottenuti sul dataset COMBINED** con questo approccio sono visibili in tabella 7.8 mentre le matrici di confusione sono riportate nella figura A.5

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.85	0.84	0.85	0.85
Decision Tree	0.61	0.61	0.61	0.61
LDA	0.64	0.64	0.64	0.63

Tabella 7.8: Metriche di valutazione per la classificazione su COMBINED con CV

Approccio con validazione LOSO

- **I risultati ottenuti sul dataset COMBINED** con questo approccio sono visibili in tabella 7.9 mentre le matrici di confusione sono riportate nella figura A.5

Algoritmo	Accuracy	Precision	Recall	F1-score
SVM	0.59	0.58	0.57	0.53
Decision Tree	0.44	0.43	0.42	0.40
LDA	0.58	0.57	0.56	0.52

Tabella 7.9: Metriche di valutazione per la classificazione su COMBINED con CV

Discussione dei risultati ottenuti su COMBINED

I risultati ottenuti per i modelli multilingua evidenziano una significativa differenza nelle prestazioni dei modelli monolingua. Questo calo era atteso, poiché il linguaggio influenza il modo in cui le emozioni sono espresse, questo risultato è in accordo con altri studi quale ad esempio [16].

Per quanto riguarda questi modelli, risulta che: con la cross-validation, l'SVM ha raggiunto un'accuracy dell'85%, la più alta, mentre per Decision Tree siamo al 61% e LDA al 64%. Con la validazione LOSO invece abbiamo risultati più scarsi siamo ad un accuracy sotto al 60% per tutti gli algoritmi.

I risultati ottenuti dai sistemi multilingua evidenziano un divario di prestazioni più ampio tra le due validazioni rispetto a quanto osservato nei sistemi monolingua. Questo divario si può spiegare con il fatto che i modelli addestrati su una sola lingua erano in grado di generalizzare meglio su dati non visti,

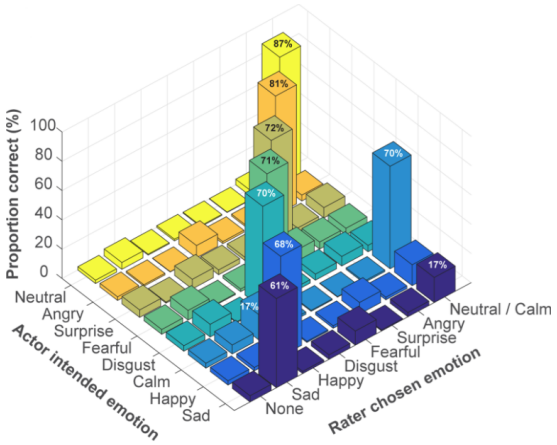
rispetto ai modelli multilingua, che faticano a gestire la diversità linguistica. In conclusione, sebbene i modelli monolingua abbiano mostrato prestazioni accettabili con la metodologia utilizzata, lo stesso non si può dire per i sistemi multilingua. Per costruire modelli multilingua affidabili sarebbe necessario esplorare algoritmi più potenti o considerare un set di feature più complesso. Tuttavia, metodologie relativamente semplici, come quella descritta in questa tesi, non risultano sufficientemente efficaci in questo contesto. Questo evidenzia come, allo stato dell'arte, lo sviluppo di modelli di riconoscimento delle emozioni multilingua rappresenti una sfida ancora più complessa rispetto alla costruzione di modelli monolingua.

7.7.3 Confronto con i risultati di validazione umani

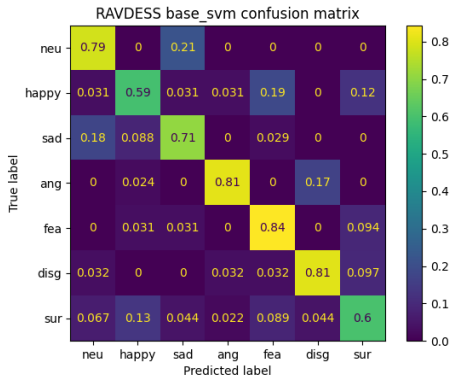
Per realizzare questo confronto ho utilizzato le matrici di confusione di SVM su RAVDESS (costruendo modelli che potessero riconoscere 7 emozioni) e confrontate con i risultati di validazione ottenuti dagli umani sempre su RAVDESS, per semplicità ho riportato tutti in figura 7.2. nelle matrici degli algoritmi l'etichetta "neu" corrisponde alla calma su RAVDNESS.

Confrontando i risultati degli algoritmi con la validazione umana, si può osservare che gli algoritmi testati con la cross validation riportano performance elevate, più elevate di quanto siano le persone a individuarle. In particolare, l'algoritmo mostra una buona accuratezza per emozioni come "fear" e "disgust", mentre tende a classificare in modo più deciso anche emozioni difficili da distinguere come "neutral" (che nella validazione umana corrisponde allo stato "calm"). Su qualche emozione come "happy" e "surprise" tuttavia l'algoritmo risulta poco accurato in questo caso i risultati sono più compatibili tra le due matrici.

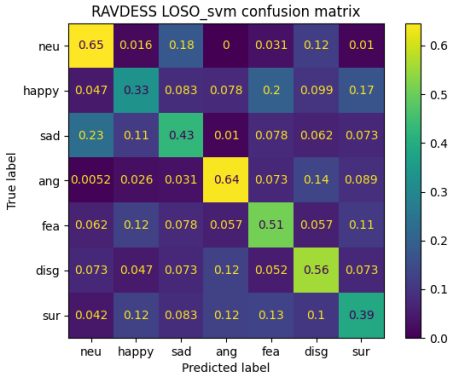
Al contrario, i risultati ottenuti con la validazione LOSO risultano nella totalità delle classi inferiori alla validazione umana, questi risultati riflettono meglio la difficoltà del compito di riconoscere le emozioni dal parlato e riflettono meglio la reale capacità dell'algoritmo di individuare l'emozione corretta.



(a) Confusione umana



(b) Confusione SVM con CV



(c) Confusione SVM con LOSO

Figura 7.2: Confronto tra le matrici di confusione: (a) Validazione umana, (b) Modello SVM con CV, (c) Modello SVM con LOSO

7.8 Conclusione

In conclusione in questo lavoro ho mostrato come, quando si tenta di sviluppare un sistema SER sia una buona pratica utilizzare una validazione di tipo Leave One Subject Out in quanto riflette maggiormente le reali capacità del modello di generalizzare su dati mai visti rispetto a una classica validazione di tipo cross-validation. Inoltre ho mostrato come, con metodologie semplici come quelle illustrate in questo lavoro risulti difficile costruire un modello che riesca a classificare accuratamente le emozioni dal parlato su più linguaggi, questo risultato rimane coerente con l'attuale letteratura scientifica. Conseguentemente questo lavoro vuole suggerire la sperimentazione di metodologie più complesse (deep learning, reti neurali e convoluzionali) per svolgere questo compito.

Appendice A

Matrici di Confusione

A.1 Sistema Monolingua

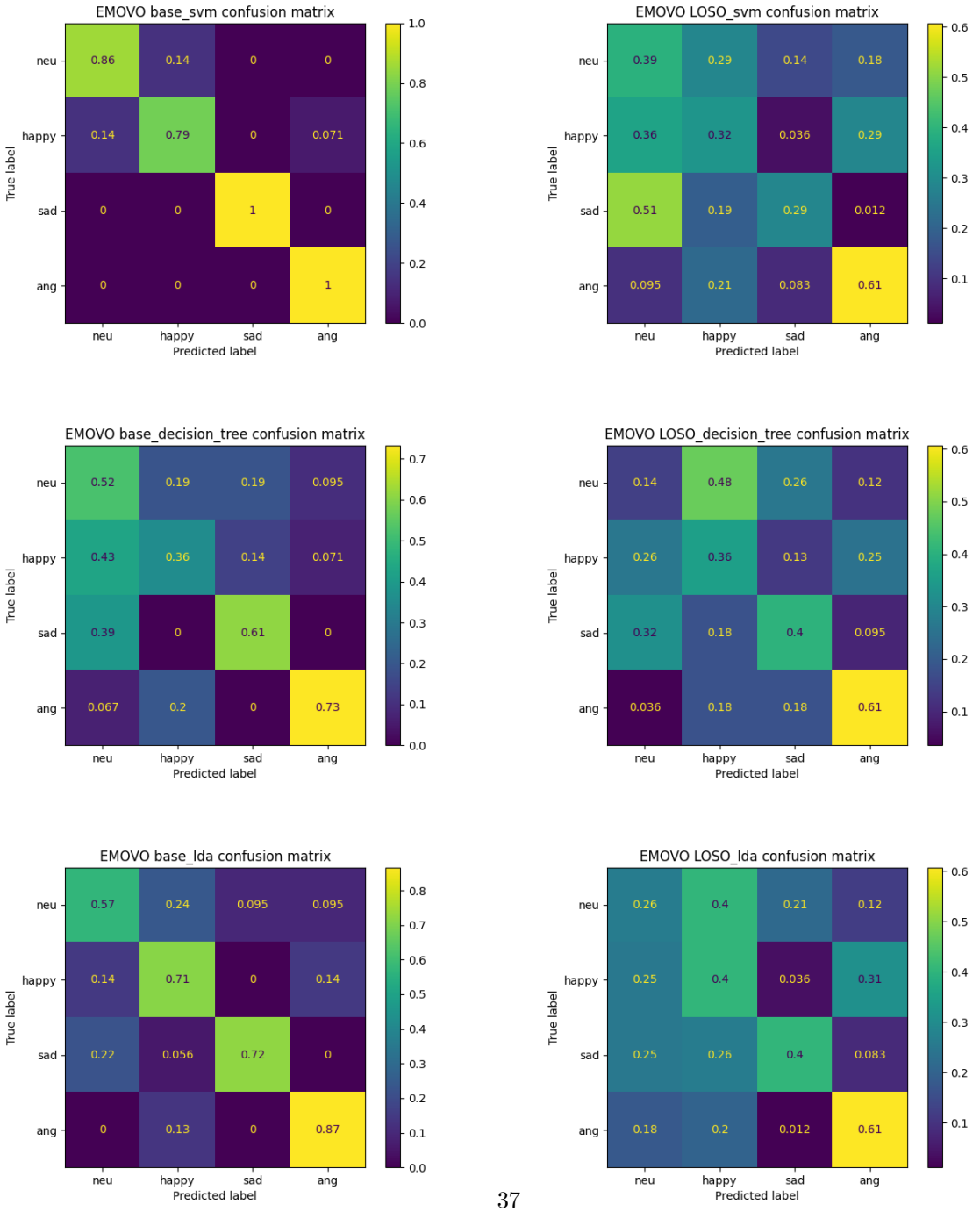


Figura A.1: Matrici di confusione su EMOVO per i tre algoritmi con i due metodi di validazione discussi

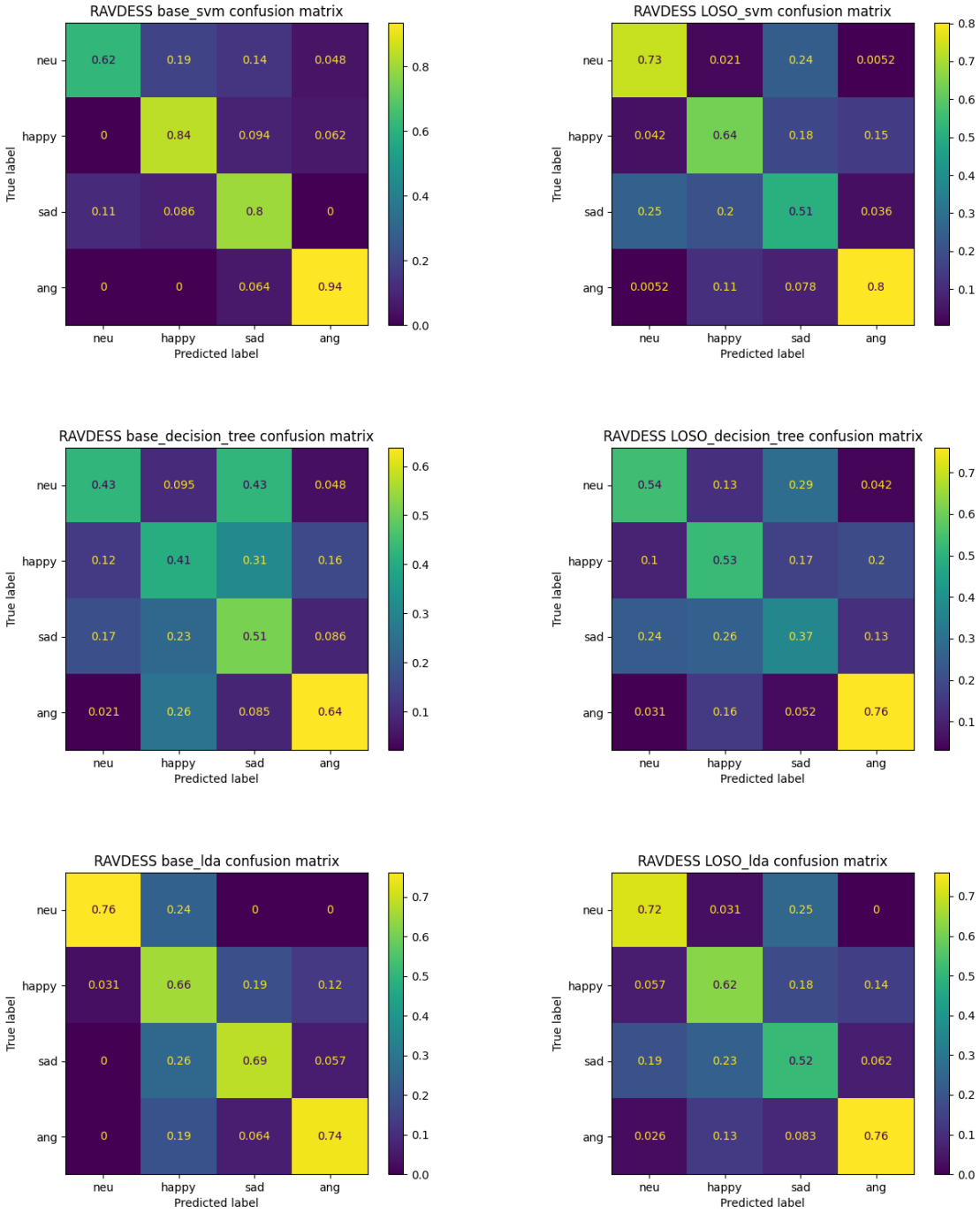


Figura A.2: Matrici di confusione su RAVDESS per i tre algoritmi con i due metodi di validazione discussi

A.2 Modelli addestrati e testati con dataset diversi

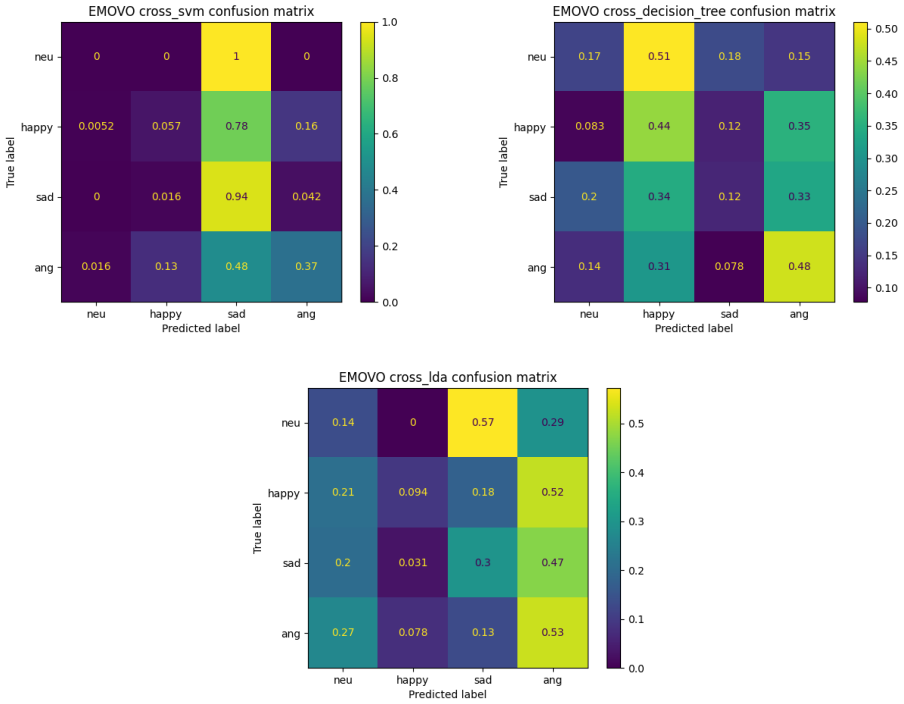


Figura A.3: Matrici di confusione per gli algoritmi SVM, Decision Tree e LDA dei modelli addestrati su EMOVO e testati su RAVDESS

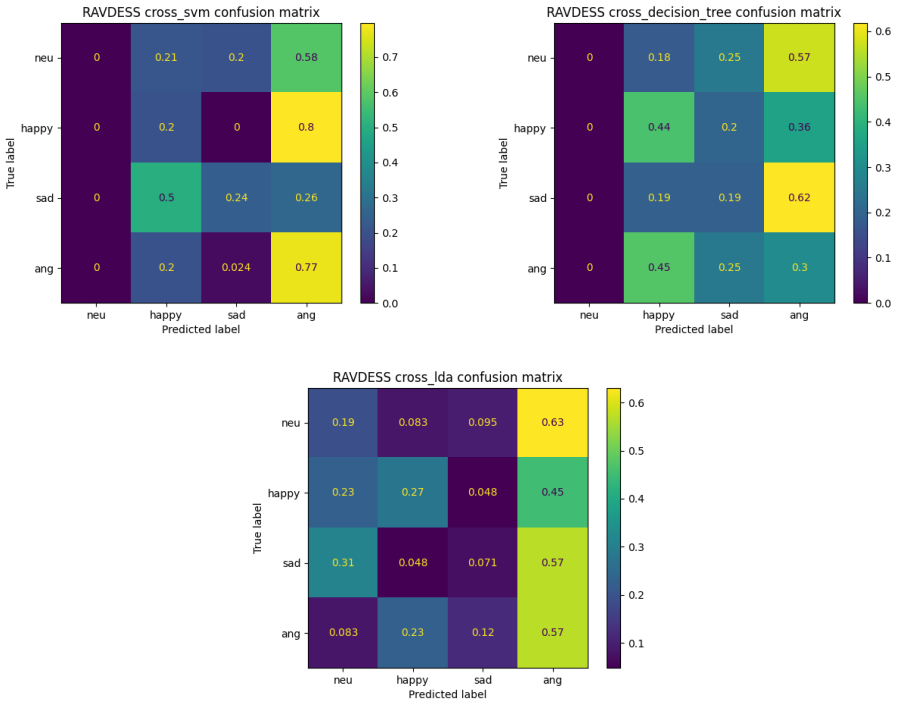


Figura A.4: Matrici di confusione per gli algoritmi SVM, Decision Tree e LDA dei modelli addestrati su RAVDESS e testati su EMOVO

A.3 Sistema Multilingua

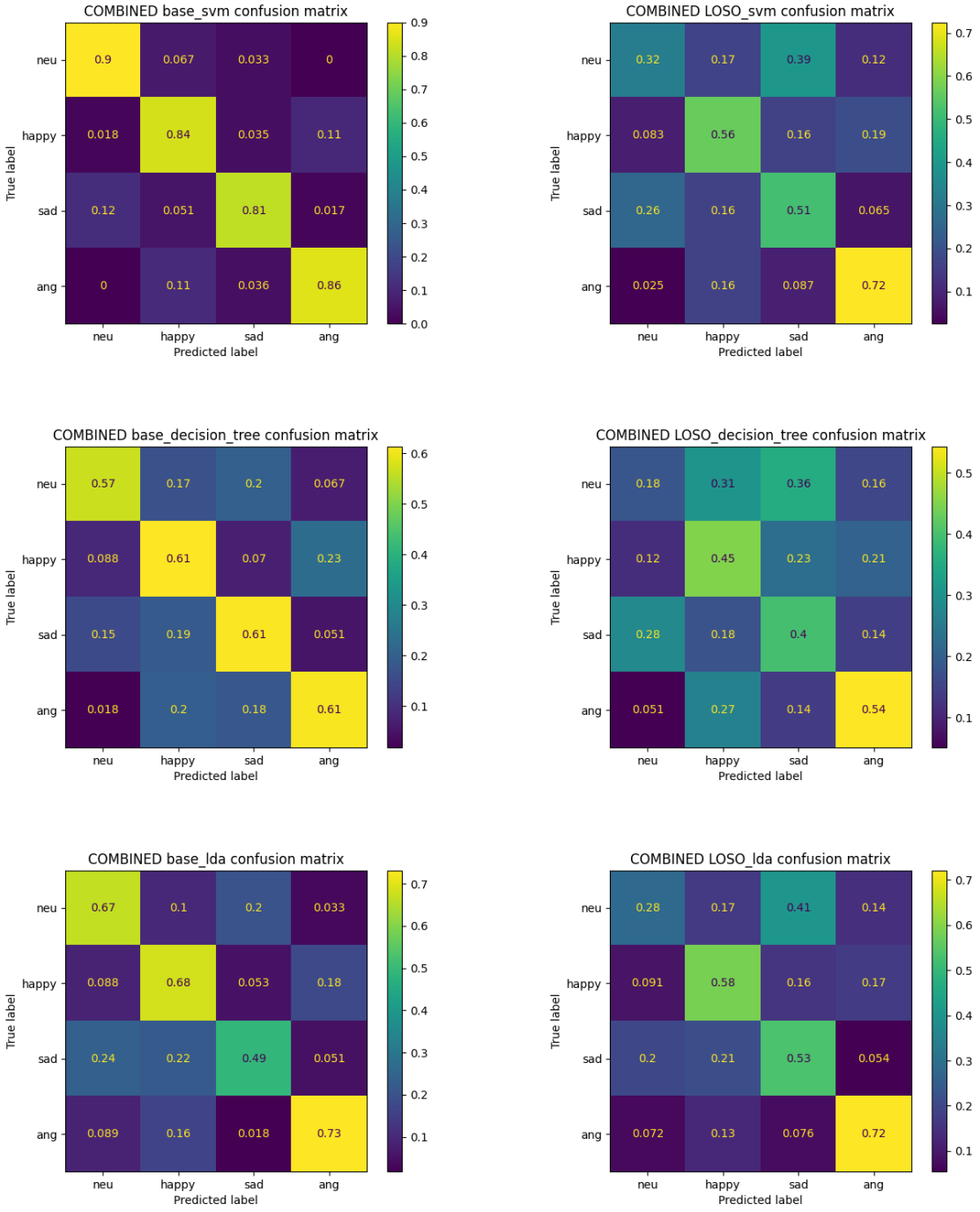


Figura A.5: Matrici di confusione su COMBINED per i tre algoritmi con i due metodi di validazione discussi

Bibliografia

[1] R. B. Blackman e J. W. Tukey, «The measurement of power spectra from the point of view of communications engineering — Part I», *The Bell System Technical Journal*, vol. 37, n. 1, pp. 185–282, 1958. DOI: 10.1002/j.1538-7305.1958.tb03874.x.

[2] B. S. Atal, «Automatic speaker recognition based on pitch contours», *The Journal of the Acoustical Society of America*, vol. 52, n. 6B, pp. 1687–1697, 1972.

[3] J. A. Russell, «A circumplex model of affect.», *Journal of personality and social psychology*, vol. 39, n. 6, p. 1161, 1980.

[4] S. R. Safavian e D. Landgrebe, «A survey of decision tree classifier methodology», *IEEE transactions on systems, man, and cybernetics*, vol. 21, n. 3, pp. 660–674, 1991.

[5] P. Ekman, «An argument for basic emotions», *Cognition & emotion*, vol. 6, n. 3-4, pp. 169–200, 1992.

[6] A. A. Farag e S. Elhabian, «A tutorial on data reduction linear discriminant analysis (LDA)», *University of Louisville, Tech. Rep*, 2008.

[7] G. Van Rossum e F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[8] C.-C. Lee, E. Mower, C. Busso, S. Lee e S. Narayanan, «Emotion recognition using a hierarchical binary decision tree approach», *Speech Communication*, vol. 53, n. 9, pp. 1162–1171, 2011, Sensing Emotion and Affect - Facing Realism in Speech Processing, ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2011.06.004. indirizzo: https://www.sciencedirect.com/science/article/pii/S0167639311000884.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort et al., «Scikit-learn: Machine Learning in Python», *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10] A. Iosifidis, A. Tefas e I. Pitas, «On the Optimal Class Representation in Linear Discriminant Analysis», *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, n. 9, pp. 1491–1497, 2013. DOI: 10.1109/TNNLS.2013.2258937.

[11] S. B. Kotsiantis, «Decision trees: a recent overview», *Artificial Intelligence Review*, vol. 39, pp. 261–283, 2013.

[12] A. Patle e D. S. Chouhan, «SVM kernel functions for classification», in *2013 International Conference on Advances in Technology and Engineering (ICATE)*, 2013, pp. 1–9. DOI: 10.1109/ICAdTE.2013.6524743.

[13] G. Costantini, I. Iaderola, A. Paoloni e M. Todisco, «EMOVO Corpus: an Italian Emotional Speech Database», in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, N. Calzolari, K. Choukri, T. Declerck et al., cur., Reykjavik, Iceland: European Language Resources Association (ELRA), mag. 2014, pp. 3501–3504. indirizzo: http://www.lrec-conf.org/proceedings/lrec2014/pdf/591_Paper.pdf.

[14] K. A. Lindquist, J. K. MacCormack e H. Shablack, «The role of language in emotion: Predictions from psychological constructionism», *Frontiers in psychology*, vol. 6, p. 444, 2015.

[15] S. Lugović, I. Dunđer e M. Horvat, «Techniques and applications of emotion recognition in speech», in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2016, pp. 1278–1283. DOI: 10.1109/MIPRO.2016.7522336.

[16] Z. Xiao, D. Wu, X. Zhang e Z. Tao, «Speech emotion recognition cross language families: Mandarin vs. western languages», in *2016 International Conference on Progress in Informatics and Computing (PIC)*, 2016, pp. 253–257. DOI: 10.1109/PIC.2016.7949505.

[17] S. Chakraborty e D. Das Chakladar, «EEG based emotion classification using "Correlation Based Subset Selection"», *Biologically Inspired Cognitive Architectures*, vol. 24, mag. 2018. DOI: 10.1016/j.bica.2018.04.012.

[18] S. R. Livingstone e F. A. Russo, «The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American

English», *PLOS ONE*, vol. 13, n. 5, pp. 1–35, mag. 2018. DOI: 10.1371/journal.pone.0196391. indirizzo: https://doi.org/10.1371/journal.pone.0196391.

[19] N. Londhe e G. Kshirsagar, «Chhattisgarhi speech corpus for research and development in automatic speech recognition», *International Journal of Speech Technology*, vol. 21, giu. 2018. DOI: 10.1007/s10772-018-9496-7.

[20] R. Saravanan e P. Sujatha, «A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification», in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 945–949. DOI: 10.1109/ICCONS.2018.8663155.

[21] M. Swain, A. Routray e P. Kabisatpathy, «Databases, features and classifiers for speech emotion recognition: a review», *International Journal of Speech Technology*, vol. 21, n. 1, pp. 93–120, mar. 2018, ISSN: 1572-8110. DOI: 10.1007/s10772-018-9491-z. indirizzo: https://doi.org/10.1007/s10772-018-9491-z.

[22] R. Thakur, M. K. Pandey e N. Gupta, «Filtering of Noise in Audio/Voice Signal», in *2018 3rd International Conference on Contemporary Computing and Informatics (IC3I)*, 2018, pp. 119–123. DOI: 10.1109/IC3I44769.2018.9007299.

[23] S. Watson, «The unheard female voice», *The ASHA Leader*, vol. 24, pp. 44–53, 2 2019. DOI: 10.1044/leader.ftr1.24022019.44.

[24] X. Ying, «An overview of overfitting and its solutions», in *Journal of physics: Conference series*, IOP Publishing, vol. 1168, 2019, p. 022 022.

[25] M. B. Akçay e K. Oğuz, «Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers», *Speech Communication*, vol. 116, pp. 56–76, 2020, ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2019.12.001. indirizzo: https://www.sciencedirect.com/science/article/pii/S0167639319302262.

[26] M. Grandini, E. Bagli e G. Visani, *Metrics for Multi-Class Classification: an Overview*, 2020. arXiv: 2008.05756 [stat.ML].

[27] A. Koduru, H. B. Valiveti e A. K. Budati, «Feature extraction algorithms to improve the speech emotion recognition rate», *International Journal of Speech Technology*, vol. 23, n. 1, pp. 45–55, 2020.

[28] T. pandas development team, *pandas-dev/pandas: Pandas*, ver. latest, feb. 2020. DOI: 10.5281/zenodo.3509134. indirizzo: https://doi.org/10.5281/zenodo.3509134.

[29] S. Zepf, J. Hernandez, A. Schmitt, W. Minker e R. W. Picard, «Driver emotion recognition for intelligent vehicles: A survey», *ACM Computing Surveys (CSUR)*, vol. 53, n. 3, pp. 1–30, 2020.

[30] M. S. Fahad, A. Ranjan, J. Yadav e A. Deepak, «A survey of speech emotion recognition in natural environment», *Digital signal processing*, vol. 110, p. 102 951, 2021.

[31] T. M. Wani, T. M. Wani, T. S. Gunawan et al., «A Comprehensive Review of Speech Emotion Recognition Systems», *IEEE Access*, 2021. DOI: 10.1109/access.2021.3068045.

[32] T. M. Inc., *MATLAB version: 9.13.0 (R2022b)*, Natick, Massachusetts, United States, 2022. indirizzo: https://www.mathworks.com.

[33] Y. Yang, Y. Zhang, Z. Zhong, W. Dai, Y. Chen e M. Chen, «Intelligent In-Car Emotion Regulation Interaction System Based on Speech Emotion Recognition», in *2024 4th International Conference on Computer, Control and Robotics (ICCCR)*, 2024, pp. 142–150. DOI: 10.1109/ICCCR61138.2024.10585371.

Ringraziamenti

Grazie mille ai miei genitori; grazie mamma e papà per essere stati il mio pilastro in tutto questo percorso. Il vostro supporto, la pazienza e l'incoraggiamento sono stati fondamentali, non potevo avere genitori migliori di voi, grazie. E ovviamente un grande grazie a mio fratello, alle mie nonne e ai miei zii: la vostra presenza non è scontata e vi ringrazio di cuore per essermi stati vicini.

Un ringraziamento speciale alla mia fidanzata Valentina, che non solo mi ha sostenuto in ogni fase, ma ha anche fatto da co-relatrice non ufficiale! Grazie per avermi sopportato e per la tua presenza qui, hai reso il mio percorso più semplice e sopportabile.

Un pensiero particolare va alla Prof.ssa Francesca Gasparini e alla Dott.ssa Alessandra Grossi. Non solo siete due docenti incredibilmente competenti, ma anche persone umane e comprensive, qualità rare da trovare. La vostra disponibilità e l'ascolto continuo mi hanno aiutato a crescere, sia a livello personale che accademico. Mi avete spronato a uscire dalla mia comfort zone con questa tesi impegnativa, e per questo non posso che ringraziarvi. È stata dura, ma mi ha fatto maturare tantissimo. Sono davvero fortunato ad aver lavorato con persone come voi.

E infine grazie anche anche a tutti i miei amici, con un ringraziamento particolare a PaoIo e Riccardo, i compagni di avventura di questa triennale in informatica. Abbiamo condiviso tutto,dalle giornate a programmare ai momenti di pura disperazione, anche se non mi piace essere sdolcinato ma devo ammettere che senza di voi sarebbe stato più difficile (e decisamente meno divertente!).