UNIVERSITÀ DI PISA

MASTER'S DEGREE IN

DATA SCIENCE & BUSINESS INFORMATICS

DATA MINING 1

RAVDESS

CATAUDELLA SERENA [664635]

COLOSIMO DOMENICO ANTONIO [609180]

INCERTI ALESSANDRO [648318]

January 2023

1. Introduzione

Il dataset preso in esame è il RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song). Contiene file audio di 24 attori che pronunciano due diverse affermazioni con accento nordamericano neutro.
Le espressioni sono prodotte dagli attori a due livelli di emotività: Normal e Strong.

Lo scopo della seguente indagine è quello di analizzare il dataset per riconoscere gli aspetti emotivi del discorso indipendentemente dai contenuti semantici.

2. Data Understanding

Il dataset è composto da 2452 righe e 38 colonne.

Gli attori, equamente distribuiti per genere, pronunciano due diverse affermazioni sia sotto forma discorsiva che cantata. Ogni riga del dataset rappresenta quindi un attore che esegue una delle seguenti otto emozioni: calm, neutral, happy, sad, angry, fearful, surprised, e disgust.

2.1 Data Semantics

Ai fini della comprensione dell'analisi, riportiamo di seguito le features trovate nel nostro dataset:

1. **Modality**: formato del file
2. **Vocal_channel:** modalità di registrazione (*song* o *speech*)
3. **Emotion:** emozione riprodotta nel file audio
4. **Emotional_intensity**: grado di intensità dell'emozione riprodotta (*normal*, *strong*)
5. **Statement**: frase pronunciata nel file audio
6. **Repetition**: numero di audio per statement
7. **Actor**: identificativo dell'attore
8. **Sex**: genere dell'attore
9. **Channels**: numero del canale della registrazione audio (1=mono, 2=stereo)
10. **Sample_width:** numero di bytes della registrazione audio (1=8-bit, 2=16-bit)
11. **Frame_rate**: frequenza della registrazione dei campioni vocali (misurata in Hertz)
12. **Frame_width:** numero di bytes per ogni frame
13. **Lenght_ms**: lunghezza del file audio (in millisecondi)
14. **Frame_count**: numero di frame che compongono ciascun campione vocale
15. **Intensity**: decibel del segnale audio relativo al suo picco massimo
16. **Zero_crossings_sum**: somma del Zero Crossing Rate (ZCR): velocità di passaggio per lo zero

Tipologia	Specificità	Nome attributo
Categoriche	Nominali	Modality, vocal_channel, emotion, emotional_intensity, statement, repetition, Sex , Actor
Numeriche	Discreti	channels, sample_width, frame_rate, frame_width
	Continui	length_ms, zero_crossings_sum, frame_count, intensity, mfcc_mean, mfcc_std, mfcc_min, mfcc_max, sc_mean, sc_std, sc_min, sc_max, sc_kur, sc_skew, stft_mean, stft_std, stft_min, stft_max, stft_kur, stft_skew, mean, std, min, max, kur, skew

Infine troviamo una serie di rilevazioni statistiche relative ad informazioni tecniche sulle note audio:

1. **Original audio signal**: mean, std, min, max, kur, skew
2. **Mel-Frequency Cepstral Coefficients**: mean, std, min, max
3. **Spectral centroid**: mean, std, min, max, kur, skew
4. **Stft chromagram**: mean, std, min, max, kur, skew

2.2 Distribuzione statistica delle variabili

Abbiamo analizzato graficamente le distribuzioni statistiche degli attributi, utilizzando gli istogrammi per le variabili continue, i bar chart per quelle discrete e gli scatterplot per studiare le relazioni di coppie di variabili. Tutte le variabili sono state studiate in relazione ad emotional_intensity, che può assumere come valori Normal (visualizzato in verde) e Strong (in rosso). Riportiamo di seguito quelle che abbiamo ritenuto più rilevanti ai fini della nostra analisi.

Fig 2.1

Emotion

Notiamo che, per le varie emotion, abbiamo la stessa proporzione di intensità emotiva tra Normal e Strong. Nel dataset sono presenti meno registrazioni relative alle emotion disgust, surprised e neutral. In particolare quest'ultima registra esclusivamente emotional_intensity Normal.

Intensity

Questa variabile segue una distribuzione tendenzialmente Gaussiana.
A valori più bassi di intensity corrisponde un numero maggiore di osservazioni con intensità emotiva Normal, e viceversa.

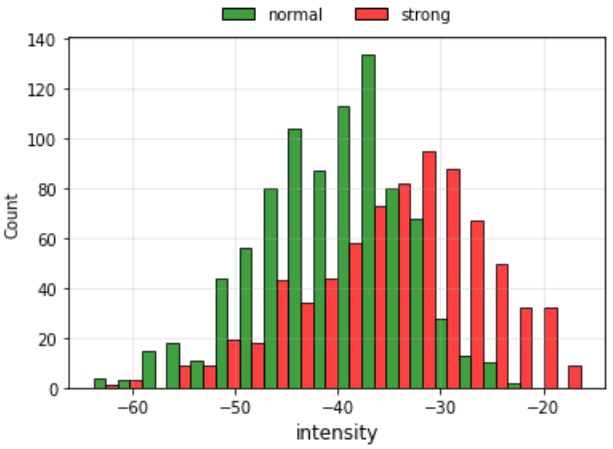


Fig 2.2

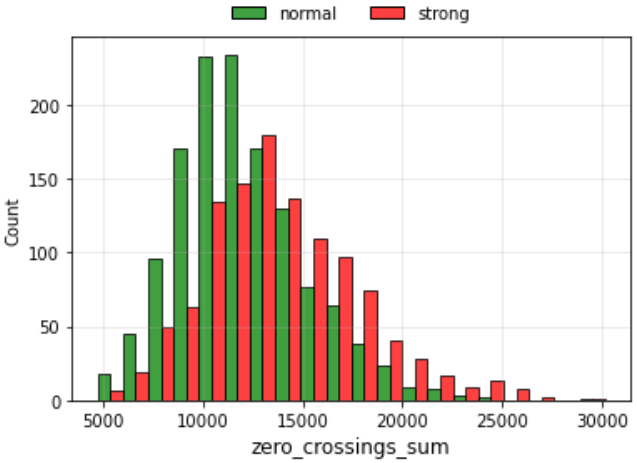


Fig 2.3

Zero Crossings Sum

La maggior parte delle osservazioni che hanno un valore di zero_crossings_sum inferiore a 12500, presentano intensità emotiva Normal. Superata tale soglia, la situazione si inverte.

Frame Count

Notiamo subito dei valori "esterni" rispetto alla distribuzione, che potrebbero essere errori di rilevazione e che saranno dunque approfonditi nel paragrafo successivo.

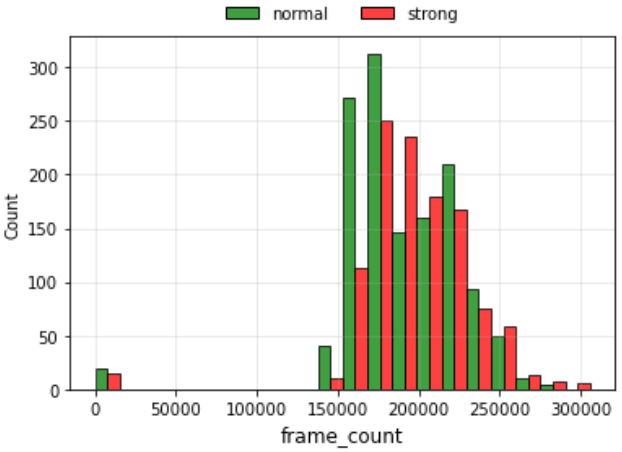


Fig 2.4

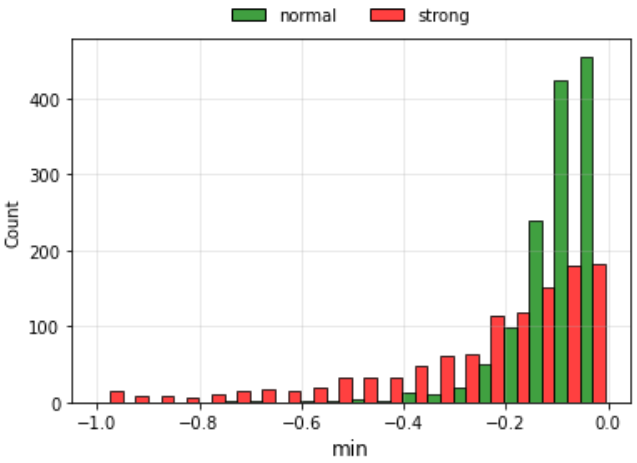


Fig 2.5

Min

Questa variabile mostra una forte asimmetria negativa, come confermato dai valori di mediana e media, rispettivamente -0.10 e -0.16.
La lunga coda della distribuzione coincide con i campionamenti audio relativi a intensità emotive Strong.

Max

In questo caso notiamo una forte asimmetria positiva, con una lunga coda che coincide nuovamente con i campionamenti audio relativi a intensità emotive Strong.
Le due variabili Max e Min sembrano avere un comportamento simile ma opposto. Approfondiremo nei capitoli successivi.

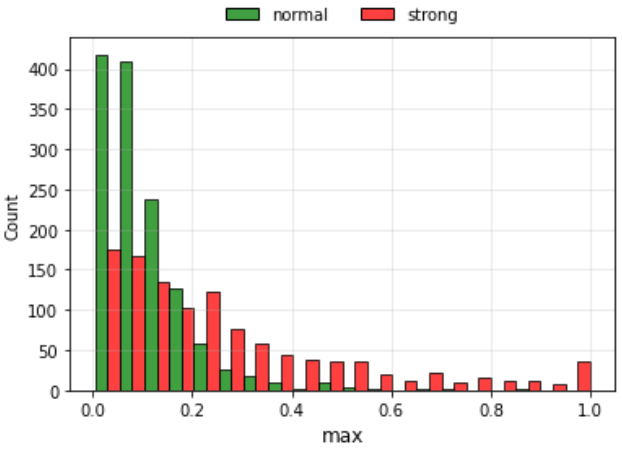


Fig 2.6

I due grafici seguenti mostrano la relazione tra emotion e intensity (Fig 2.7), ed emotion e length_ms (Fig 2.8). E' interessante osservare che emozioni che generano reazioni "forti" come la paura, la rabbia e la felicità abbiano intensità emotive Strong per valori più alti di intensity, e che vengano espresse in registrazioni audio un po' più brevi (tra i 3 e i 5,5 secondi circa).
Emozioni che generano reazioni più pacate, come la calma e la tristezza, invece, hanno dei valori di intensity più bassi, e vengono espresse in registrazioni audio che coprono tutto il range di length_ms. In queste due reazioni, inoltre, la distribuzione tra intensità emotive Normal e Strong è mista.

Fig 2.7

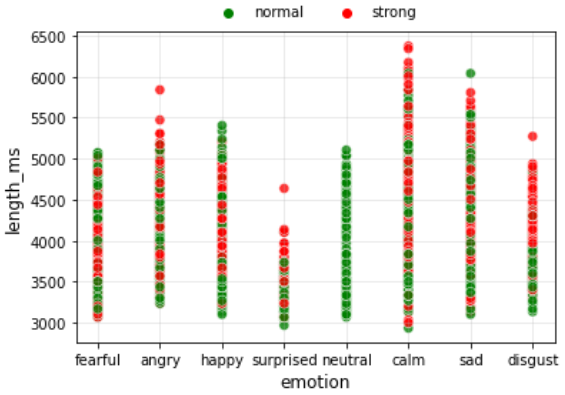


Fig 2.8

Riportiamo infine lo scatterplot tra mfcc_min ed intensity, che mostra una forte correlazione lineare positiva tra le due variabili.
Questa relazione sarà particolarmente utile nel prossimo paragrafo, quando dovremo trattare i missing values della variabile intensity.

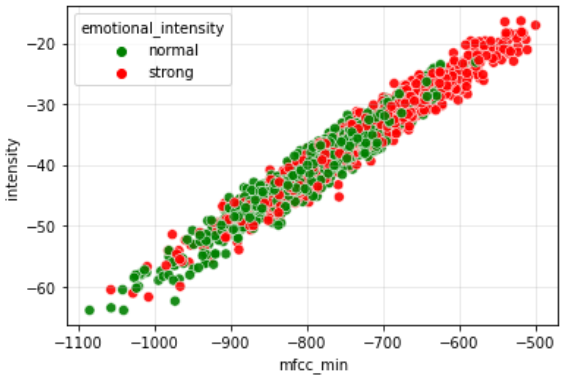


Fig 2.9

2.3 Data Quality

La qualità del dataset utilizzato incide profondamente sull'efficienza e le performance dei modelli, e più in generale sull'analisi effettuata. In questa sezione ci siamo dunque occupati di migliorare la qualità del dato, cercando inconsistenza semantica, missing values, outliers e valori duplicati.
Prima di tutto abbiamo rimosso dal dataset alcuni attributi che assumevano un solo valore e che quindi non erano utili ai fini dell'analisi: modality (*audio-only*), sample_width (*2*) , frame_rate (*48000 Hz*) e stft_max (*1*).

Inconsistenza semantica

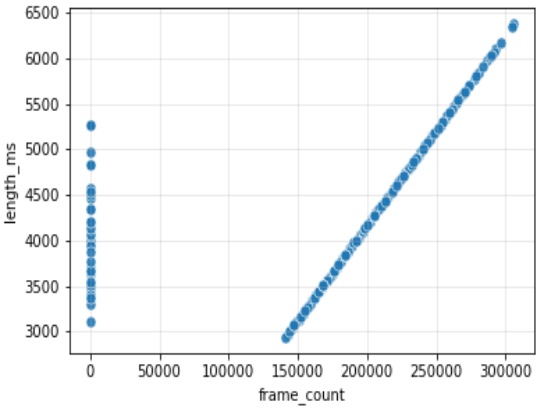


Fig 2.10

Dallo studio delle distribuzioni delle variabili, avevamo già notato delle osservazioni probabilmente associate a errori nella rilevazione dell'attributo frame_count. Si passa infatti da -1 a 140941. Tuttavia i valori coincidenti con -1 corrispondono solo a 35 righe, e quindi, prima di decidere se valesse la pena rimuoverle o no, abbiamo preferito svolgere uno studio più approfondito.
In particolare, studiare la sua correlazione con la variabile length_ms, ha evidenziato un'inconsistenza rispetto a quella che altrimenti sembrerebbe una stretta correlazione lineare positiva.

Abbiamo dunque eliminato questi valori, e ottenuto uno scatterplot che mostra la forte correlazione positiva tra le due variabili.

Il nostro Dataset ora è composto da 2417 records.

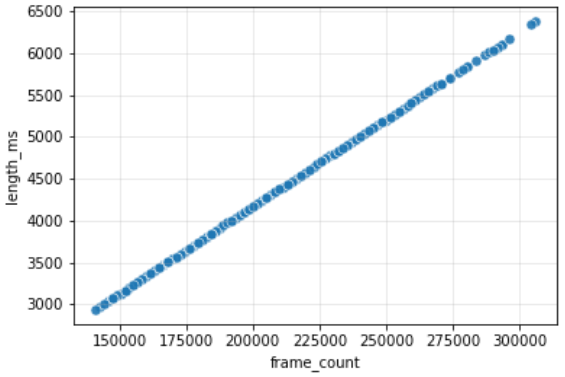


Fig 2.11

Missing values

Analizzando il dataset è possibile osservare che 3 variabili (actor, vocal_channel ed intensity) contengono missing values.
Poichè non c'è un metodo generale per trattarli, siamo andati ad analizzare caso per caso per capire come comportarci.

actor	1107
vocal_channel	193
intensity	805

Actor
Il 46% delle righe presenta valore NaN e, poichè può essere considerato come un

identificativo (quindi un ID poco utile ai fini dell'analisi), abbiamo deciso di eliminare la feature dal dataset.

Vocal_channel

Essendo una variabile categorica, abbiamo deciso di rimpiazzare i valori mancanti con la moda (che coincide con speech)

Intensity

Anche se questa variabile presenta una grande quantità di missing values, risulta essere significativa ai fini della nostra analisi, di conseguenza non possiamo eliminare la feature ma dobbiamo rimpiazzare i missing values.
Abbiamo prima di tutto provato a sostituirli tramite la mediana, e poi utilizzando il metodo del raggruppamento (con le variabili emotion ed emotional_intensity). Tuttavia abbiamo notato che questi metodi modificavano in modo significativo la distribuzione della variabile.
Abbiamo quindi sfruttato l'alta correlazione con la feature **mfcc_min** (mostrata nel paragrafo precedente) e abbiamo applicato una regressione lineare. Abbiamo diviso il dataset in test set (righe contenenti i missing values, il 33% circa) e training set (tutto il resto), garantendo quindi un'ottima proporzione di holdout. La regressione ha restituito un valore di R^2 pari a 0.95.
Questa metodologia di sostituzione dei missing values, ha addirittura migliorato la distribuzione della variabile, rendendola perfettamente Gaussiana.

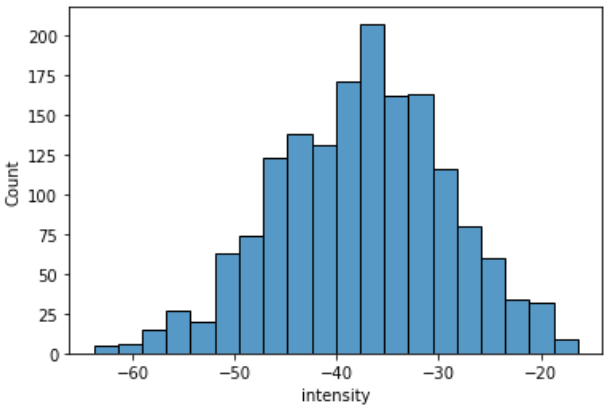


Fig 2.11: Distribuzione iniziale intensity

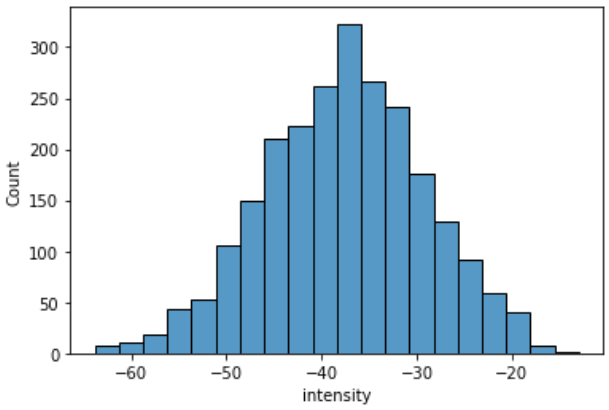


Fig 2.12: Distribuzione finale di intensity

Outliers

Per la ricerca di eventuali outliers abbiamo provveduto alla visualizzazione grafica dei dati tramite boxplots. Ne riportiamo, a titolo esemplificativo, due tra i più significativi che abbiamo ottenuto.

Il boxplot della variabile *max* ci segnala un'inconsueta quantità di outliers, quando, in realtà, non sono altro che i valori della lunga coda della distribuzione (riportata in figura 2.6 del paragrafo precedente).

Per quanto riguarda il boxplot della variabile *mean*, invece, notiamo come il range interquartile sia prossimo allo 0. Questo ci fa intuire che la distribuzione della variabile è interamente raccolta intorno alla mediana. Anche in questo caso il boxplot segnala la presenza di una grande quantità di outliers, ma, come prima, questo è dovuto alla distribuzione della variabile (riportata in Fig 2.14). Il boxplot mostra due valori agli estremi che potrebbero effettivamente essere anomali. Tuttavia, essendo solo 2, abbiamo deciso di non rimuoverli.

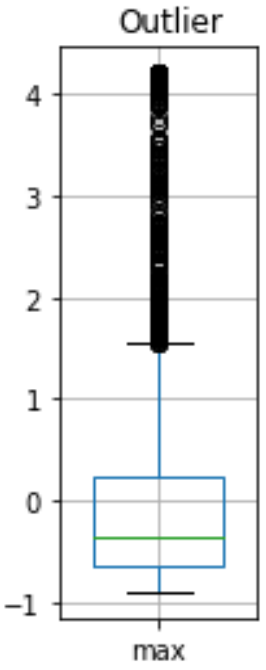


Fig 2.12

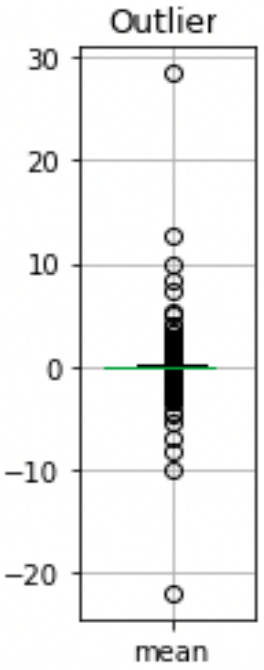


Fig 2.13

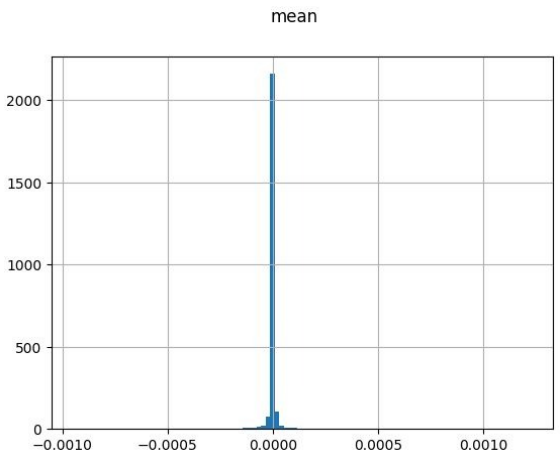


Fig 2.14

2.4 Variables transformations

Ai fini dell'analisi e quindi anche per una successiva utilità, abbiamo binarizzato la variabile categorica **vocal_channel.**

Song	0
Speech	1

2.5 Correlations and redundant variables

L'analisi di correlazione tra le variabili ha permesso l'eliminazione di alcune valori ridondanti.
In particolare abbiamo osservato che:

- length_ms e frame_count hanno una correlazione positiva massima (pari a 1). Tale correlazione è spiegabile dal fatto che le due features descrivono la stessa caratteristica (durata) della nota audio ma con due unità di misura differenti. Eliminiamo dunque la variabile frame_count
- frame_width e channels hanno correlazione pari a 1. Frame_width è infatti calcolata moltiplicando la dimensione del campione in byte per il numero di canali. Escludiamo quindi la variabile frame_width. Tuttavia anche la variabile channels risulta essere poco utile ai fini dell'analisi, in quanto assume sempre valore 1, tranne in 6 righe in cui assume valore 2. Quindi escludiamo anche questa
- come avevamo già intuito nel paragrafo precedente (Fig 2.9), intensity e mfcc_min sono strettamente correlate positivamente (0.98). Inoltre intensity e mfcc_std sono strettamente correlate negativamente (-0.98). Per questi motivi decidiamo di escludere dal dataset le variabili mfcc_min e mfcc_std
- Stft_mean e stft_skew sono strettamente correlate negativamente (-0.97), pertanto rimuoviamo la seconda dal dataset
- la variabile std è strettamente correlata con min e max (-0.96 e 0.95 rispettivamente). Decidiamo di eliminare std e min.

In seguito a questa pulizia, visualizziamo nuovamente la matrice di correlazione e la riportatiamo in Fig. 2.15

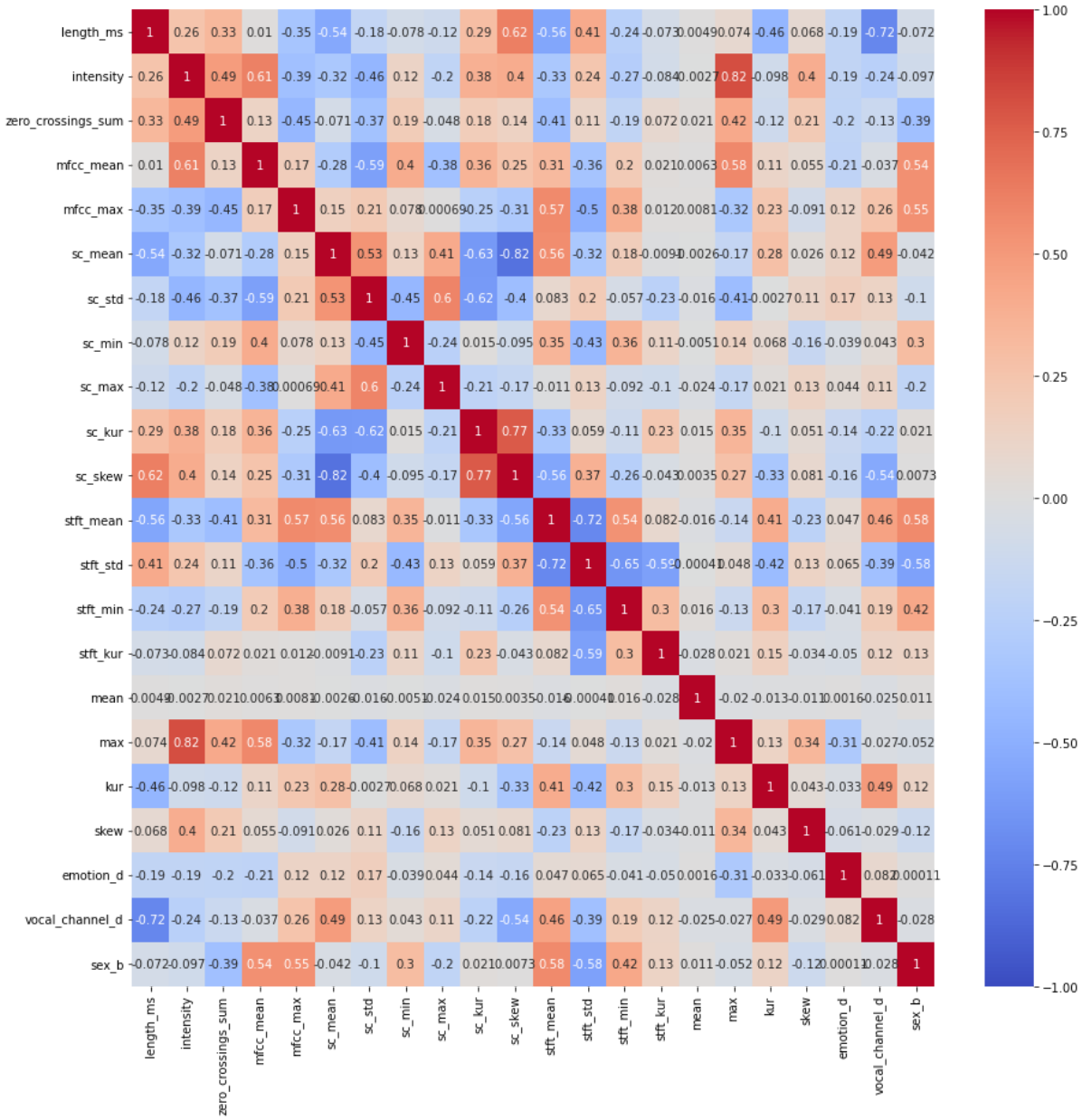


Fig 2.15

3. Clustering

Abbiamo proseguito la nostra analisi con lo scopo di dividere il dataset, ripulito secondo le indicazioni del capitolo precedente, in gruppi. Per individuare tali raggruppamenti abbiamo utilizzato tre tecniche: **K-means, Hierarchical clustering e DBSCAN.**

3.1 K-means

Prima di procedere, però, abbiamo scelto le variabili da utilizzare. Alcuni metodi di clustering, infatti, come il K-means, hanno dei problemi con gli attributi categorici e quindi abbiamo escluso emotion, vocal_channel e sex.
In seguito, dopo aver comparato i risultati ottenuti dalle differenti configurazioni di attributi rimasti, abbiamo deciso di considerare solo un subset di attributi continui, in quanto distribuzioni asimmetriche (come ad esempio stft_min) avrebbero influenzato negativamente la ricerca dei clusters. Con la progressiva eliminazione delle variabili più asimmetriche, abbiamo notato un significativo miglioramento dei valori di SSE e Silhouette,

fino a raggiungere una situazione che abbiamo ritenuto ottimale.
Proseguiamo quindi, per i modelli di clustering, con gli attributi: **intensity**, **lenght_ms** e **zero_crossings_sum.**
Dopo aver normalizzato i dati e settato la distanza euclidea, abbiamo utilizzato il metodo della curva a gomito per scegliere il miglior numero di cluster k (in relazione ai nostri dati).

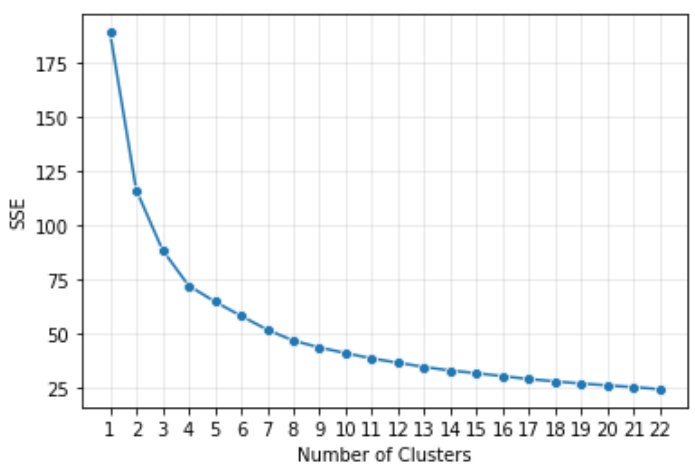


Fig 3.1

Per definizione matematica della SSE, essa diminuisce all'aumentare del numero di cluster.
Il gomito della curva rappresenta quel valore per cui un aumento del numero di cluster non porta ad una notevole diminuzione dell'errore. E viceversa, se decidessimo di diminuire il numero di cluster, avremmo un errore considerevolmente più alto (rispetto alle crescite precedenti).

k	Silhouette	SSE
3	0.323	88.64
4	0.299	72.13
5	0.276	64.79

Nel nostro caso k=4 rappresenta un giusto trade-off tra numero di cluster e SSE bassa.

I cluster individuati hanno una distribuzione abbastanza equilibrata

Cluster 1	29.7%
Cluster 2	28.1%
Cluster 3	26.4%
Cluster 4	15.9%

Per valutare la bontà dei nostri cluster, abbiamo visualizzato, tramite il metodo delle coordinate parallele, i 4 centroidi.

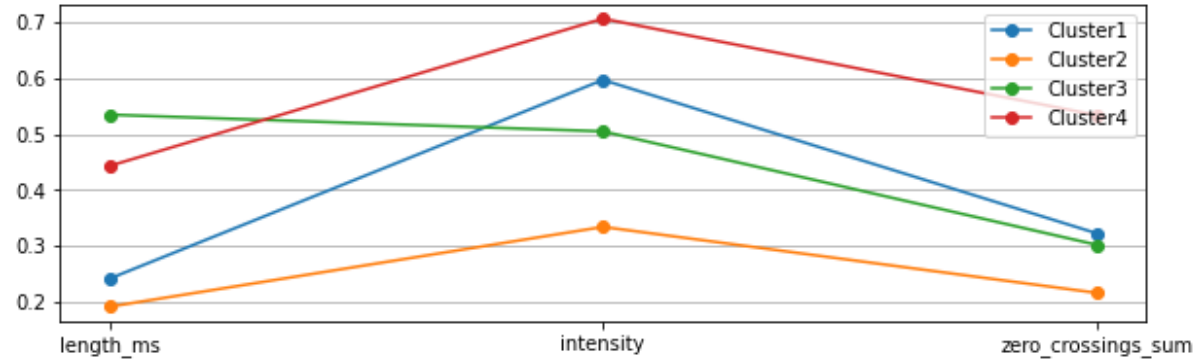


Fig 3.2

Nel caso ideale, le spezzate rappresentanti i clusters si troverebbero in posizione parallela le une rispetto alle altre, in modo da massimizzare la distanza. Nel nostro caso, questo non si verifica (possiamo però osservare che, il centroide del cluster 3, nel primo tratto che va da length_ms ad intensity, è quasi parallelo all'asse delle ascisse).

Tuttavia, possiamo notare che i centroidi degli attributi lenght_ms ed intensity risultano abbastanza separati. Per quanto riguarda zero_crossings_sum, invece, i centroidi dei cluster 1 e 3 sono molto vicini, mostrando nel complesso una minore differenziazione dei 4 gruppi individuati.

In Fig 3.3 è riportato lo scatterplot tra le variabili length_ms e intensity. Abbiamo aggiunto una terza dimensione tramite il colore, per differenziare i cluster, plottandone anche i centroidi. Possiamo visualizzare i 4 cluster ma risulta difficile individuare una netta separazione tra gli stessi. La nuvola di punti che osserviamo infatti è molto densa, e la conseguenza è che punti appartenenti a cluster diversi sono molto vicini.

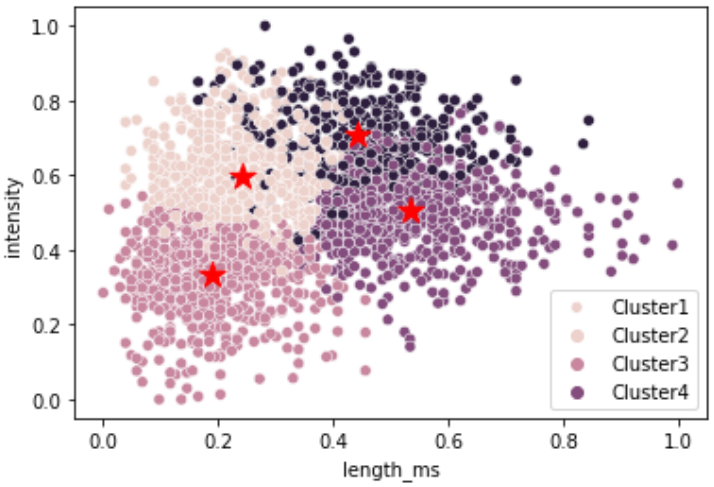


Fig 3.3

Particolarmente interessante è stato lo studio delle distribuzioni dei 4 cluster nelle variabili categoriche che non abbiamo potuto utilizzare nell'algoritmo.

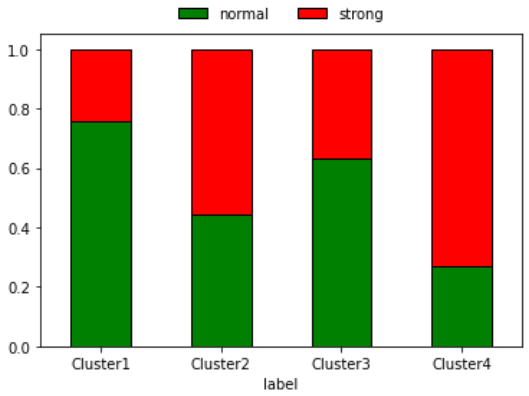


Fig 3.4

Emotional_intensity

La maggior parte delle osservazioni che rientrano nel cluster 4, sono relative ad intensità emotive Strong. Anche il cluster 2 raggruppa più valori Strong (seppur in minor misura).
I cluster 1 e 3, al contrario, raccolgono la maggior parte delle registrazioni relative ad intensità emotive Normal.

Vocal_channel

La quasi totalità delle registrazioni in modalità discorsiva, ricade nei cluster 1 e 2. Invece le canzoni sono concentrate nei cluster 3 e 4.

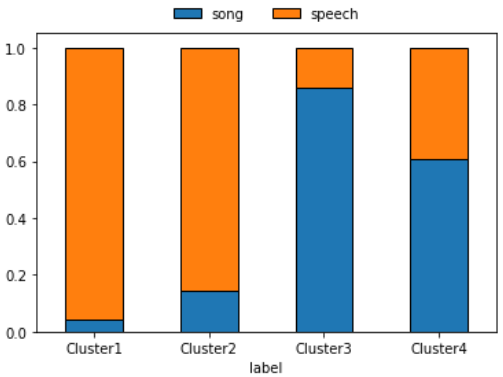


Fig 3.5

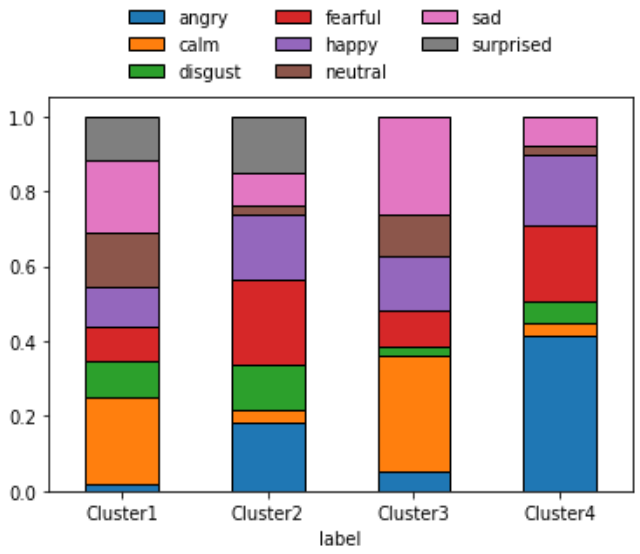


Fig 3.6

Emotion

Nei clusters 2 e 4 troviamo una maggiore concentrazione di emozioni angry, disgust, fearful, happy e surprised, caratterizzate in genere da un'espressività emotiva molto intensa e impulsiva e da reazioni che possiamo definire appunto più "forti".
Viceversa, i clusters 1 e 3 sono accomunati dalla maggiore presenza di stati emotivi che sono soliti essere espressi attraverso reazioni emotive più "neutre" e "posate", quali sad, neutral e calm.
In particolare questa distinzione tra emozioni che generano reazioni più forti o più pacate, verrà tenuta in considerazione nell'ultimo capitolo in cui studieremo i pattern più frequenti all'interno del nostro dataset.

Abbiamo infine controllato la distribuzione della variabile sex all'interno dei cluster, ma, dato che non ha fornito dettagli utili ai fini della descrizione dei cluster, abbiamo deciso di non riportare il bar chart nel report.

3.2 Hierarchical algorithm

Il secondo metodo di clustering che abbiamo utilizzato è stato quello gerarchico, in particolare agglomerativo, sulla base della definizione di distanza Euclidea e di clustering proximity (Single, Average, Complete e Ward). È stato possibile sfruttare tale metodo di clusterizzazione in quanto il nostro dataset conta 2417 records, una quantità non eccessivamente elevata per il costo computazionale dell'algoritmo. Per la nostra analisi abbiamo adoperato lo stesso dataset utilizzato per il K-Means.

Riportiamo di seguito i risultati ottenuti:

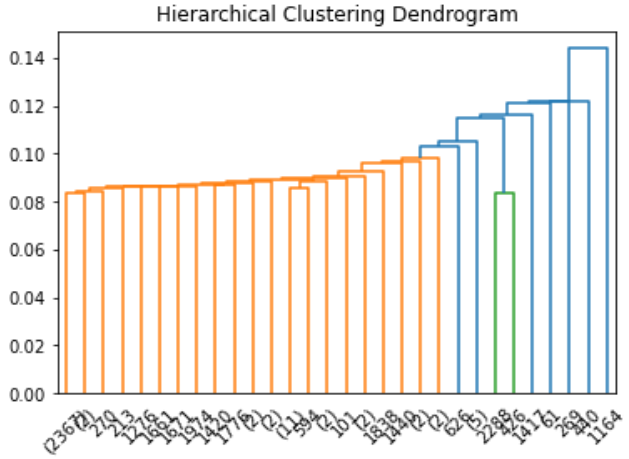


Fig 3.7

Single Linkage

La distanza ridotta tra i punti non ha permesso una buona caratterizzazione dei clusters: presenta un cluster molto grande e un cluster di dimensioni molto piccole, conducendo a un esito poco significativo da un punto di vista intuitivo.
Inoltre non è possibile tracciare una threshold orizzontale che sia in grado di separare contemporaneamente tutti e 2 i clusters.

Average

Anche in questo caso, la clusterizzazione non è stata molto efficace, in quanto, pur producendo dei cluster facilmente distinguibili (distanza tra nodi e biforcazioni considerevole), la distribuzione dei data point all'interno dei cluster è molto sbilanciata, producendo un cluster notevolmente più popoloso rispetto agli altri 2.

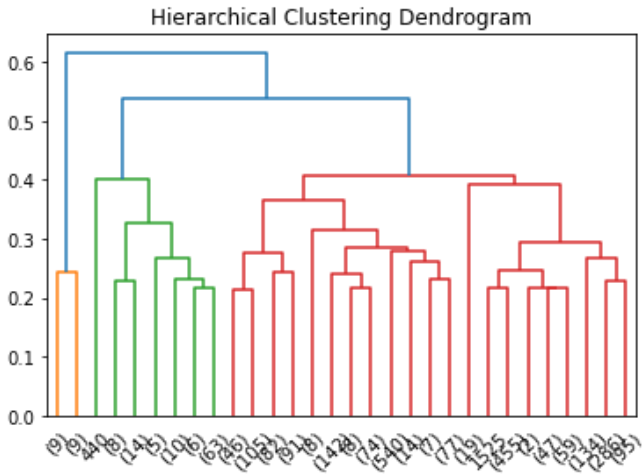


Fig 3.8

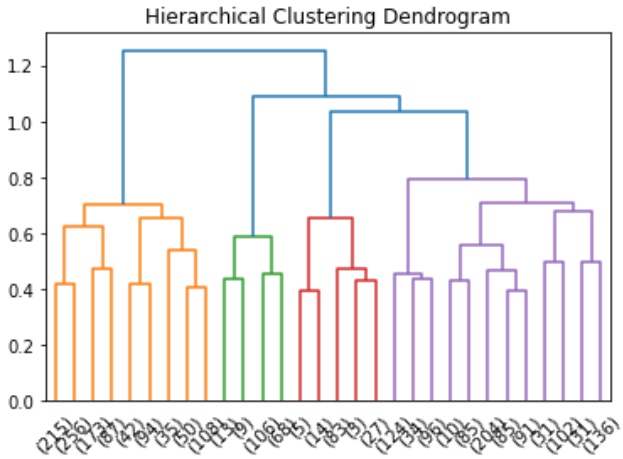


Fig 3.9

Complete

Questo metodo fornisce un risultato più soddisfacente, in quanto i clusters sono ben distinti e più bilanciati rispetto alle due situazioni precedenti.

Ward

Come osservato nel paragrafo precedente, i punti presentano un'elevata densità e i cluster individuati con il K-Means non hanno confini ben definiti. Dato che il metodo di Ward è più efficiente nel trattare outliers e rumore, si adatta molto meglio al nostro dataset, come confermato dal dendrogramma di fianco.

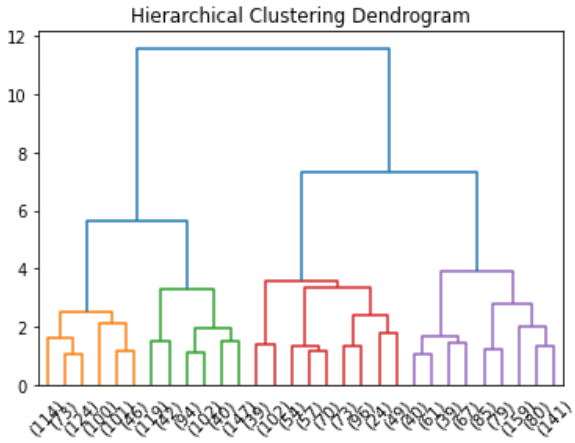


Fig 3.10

Per determinare la migliore configurazione di cluster, teniamo conto dei dendrogrammi appena mostrati, dei coefficienti di Silhouette ottenuti, e delle distribuzione dei data points nei cluster.

I coefficienti di Silhouette sono molto simili tra di loro. Tuttavia, per quanto detto fino ad ora i metodi migliori sembrano essere Complete e Ward.

Metodo	Silhouette
Single	0.258
Average	0.251
Complete	0.231
Ward	0.242

Cluster 1	31.1%
Cluster 2	22.5%
Cluster 3	23.3%
Cluster 4	23.1%

Con quest'ultimo metodo, però, otteniamo dei cluster più bilanciati in termini di numeriche, e per questo motivo la scelta finale ricade sul metodo di Ward.

3.3 DBSCAN

Il DBSCAN è l'ultimo algoritmo di clusterizzazione implementato nella nostra analisi. Si tratta di un metodo density-based che si presta bene all'analisi di dataset con forme irregolari, affrontando inoltre in maniera particolarmente efficace lo scoglio rappresentato dal rumore.
Anche in questo caso, analogamente ai precedenti, abbiamo utilizzato la distanza euclidea.
Il metodo DBSCAN richiede la scelta di due parametri: **epsilon** e **min_sample.** Per poter determinare il miglior valore di epsilon abbiamo provato a plottare le distanze ordinate dal k-esimo vicino utilizzando come k valori compresi tra 3 e 30.

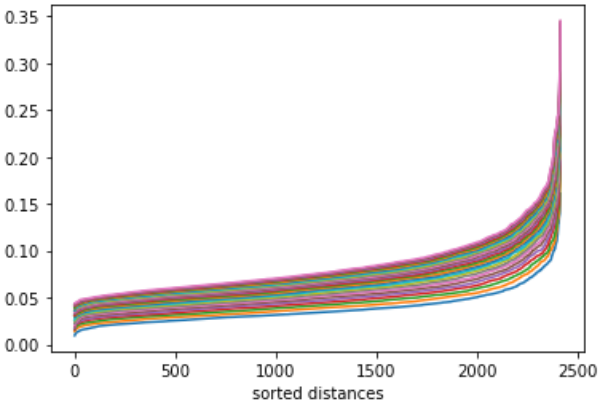


Fig 3.11

Dopo aver eseguito l'algoritmo per tutti i valori individuati, abbiamo deciso di tenere in considerazione solamente i risultati con un valore di silhouette > 0.2.

eps	min_sample	Silhouette
0.1	4	0.222
0.1	5	0.396
0.1	6	0.396

Come si evince dalla tabella sopra, la miglior configurazione dei parametri si ha con eps=0.1 e min_sample=5 che però producono un risultato insoddisfacente in quanto individuano un cluster di outliers contenente 12 data points e un altro cluster contenente 2405 points.

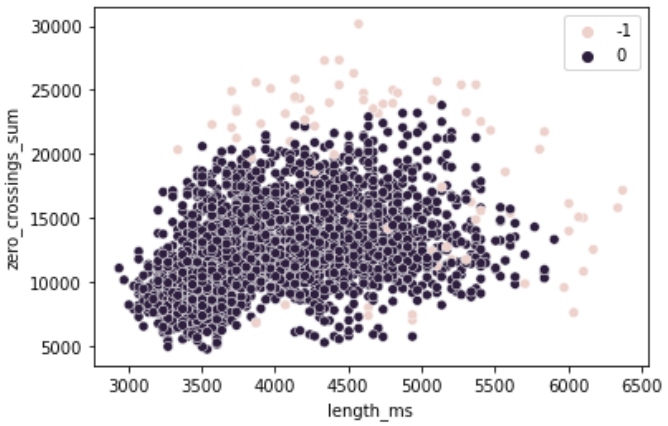


Fig 3.12

Pur provando a scegliere diverse configurazioni dei parametri **min_sample** ed **eps** il risultato del clustering risulta essere il medesimo.
Possiamo quindi concludere che l'algoritmo DBSCAN non è efficiente per un dataset che presenta punti molto densi tra loro come il nostro.

3.4 Conclusioni

Tutte le osservazioni fatte fino ad ora, ci portano alla conclusione che, tralasciando l'algoritmo DBSCAN, nel nostro dataset sono facilmente individuabili 4 cluster. In particolare, i due metodi più efficaci sono il K-Means e il clustering agglomerativo (con il metodo di Ward). Tuttavia, studiando gli scatterplot tra le coppie di variabili del dataset con i cluster visualizzati, abbiamo dedotto che l'algoritmo che riesce a separare meglio i nostri data points è il K-Means.

4. Classification

In questo capitolo ci siamo dedicati alla costruzione di un modello di classificazione per predire la varabile target, rispetto ad un set di features conosciute. Abbiamo utilizzato 3 algoritmi differenti (Decision Tree, KNN e Naïve Bayes).

Prima di tutto abbiamo provato ad utilizzare come variabile target emotion, ottenendo però delle performance basse. Tra vocal channel ed emotional intensity, abbiamo poi preferito proseguire con l'ultima, in quanto vocal channel aveva l'8% di missing values, che sono stati da noi rimpiazzati con la moda.

Dopo aver effettuato diversi tentativi per individuare lo spazio delle features che generasse il miglior modello di predizione, abbiamo deciso di proseguire con le seguenti variabili: intensity, lenght_ms, zero_crossings_sum ed emotion. Quest'ultima, essendo categorica, è stata trasformata tramite il metodo one-hot encoding.
Il 70% del dataset è stato usato per il training set, e il restante 30% per il test set.

4.1 Decision Tree

Al fine di ottenere il modello più efficace, abbiamo utilizzato la tecnica del Grid Search con il K-Fold Cross Validation (con K=*5*), fissando il metodo di Gini come misura di impurità dei nodi. Abbiamo ottenuto i seguenti risultati:
max_depth=12; *min_samples_leaf*=0.01; *min_samples_split*=0.05.

L'algoritmo è stato quindi trainato nuovamente con i parametri di cui sopra, e testato. Di seguito la valutazione del modello tramite le metriche:
- Accuracy: 0.75. Questo valore spiega che il 75% delle predizioni che fa il modello, risultano essere corrette.

- Precision: 0.76. Su tutte le righe che il modello classifica come strong, il 76% sono realmente strong.
- Recall: 0.75. Rispetto a tutte le osservazioni che sono veramente strong, il 75% sono classificate come strong.
- F1-Score: 0.75. E' la media armonica tra precision e recall.

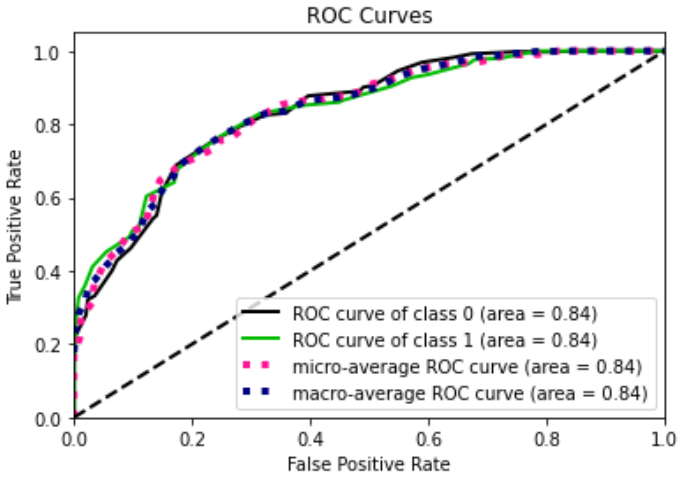


Fig 4.2

Per plottare la ROC Curve analizziamo come varia il numero di falsi positivi e veri positivi al variare della soglia che scegliamo. Facendo una predizione randomica (quindi senza l'utilizzo di un modello) si ottiene la diagonale (l'area al di sotto è pari a 0.5). Con il nostro modello di Decision Tree otteniamo un valore considerevolmente più alto, in quanto l'area sottesa dalla curva coincide con 0.84.

Le prime due variabili più utili negli split sono intensity e length_ms.
Tramite la costruzione dell'albero decisionale, notiamo infatti che il primo split viene fatto sulla variabile intensity, l'impurità del nodo corrisponde a 0.496, e sono presenti 917 righe in cui emotional intensity ha valore 0 (quindi corrispondente a normal), e 774 in cui ha valore 1 (strong).

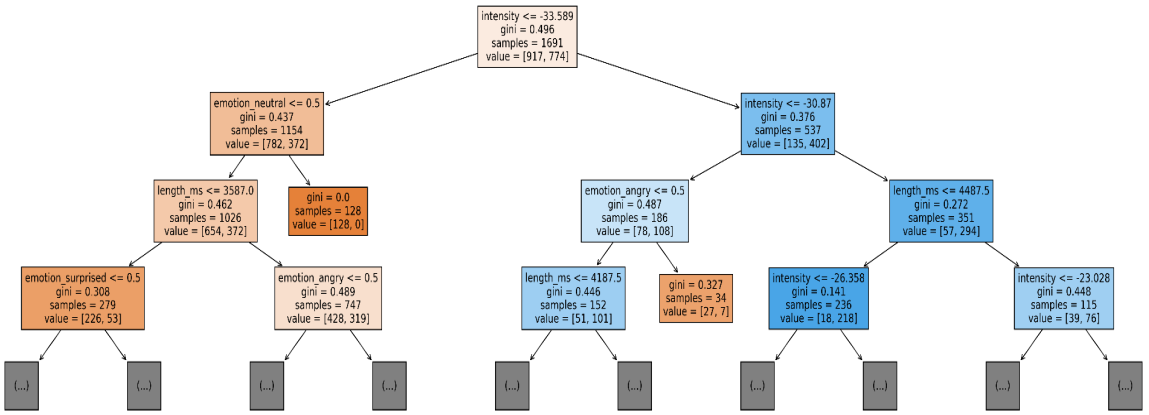


Fig 4.3

4.2 KNN e Naive Bayes

Infine abbiamo applicato al nostro dataset gli algoritmi del KNN e di Naive Bayes.
In particolare per quanto riguarda il KNN, abbiamo standardizzato il dataset ed utilizzato il GridSearchCV per trovare i parametri migliori, che sono risultati essere*:*
metric: cityblock*, n_ neighbors' =22, weights:* distance.

KNN

Accuracy	Precision	Recall	F1-Score
0.77	0.77	0.77	0.77

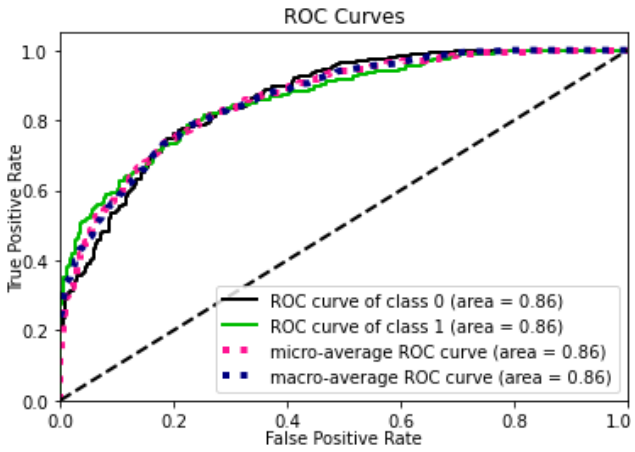


Fig 4.4

Naive Bayes

Accuracy	Precision	Recall	F1-Score
0.64	0.70	0.65	0.62

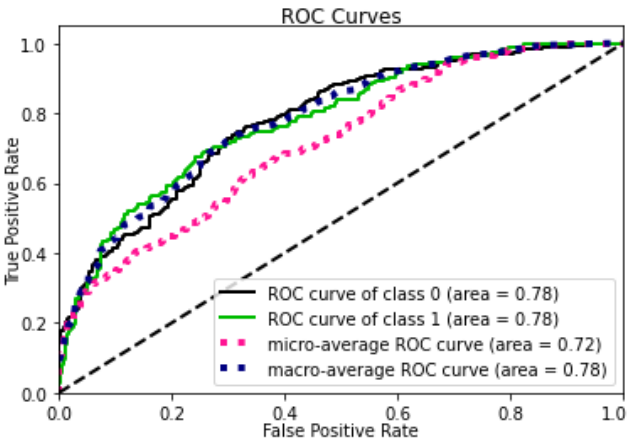


Fig 4.5

4.3 Conclusioni

Confrontando le performance dei vari algoritmi, possiamo concludere che il migliore per il nostro set di dati sembra essere il KNN, in quanto abbiamo un AUC pari a 86%, accuracy, precision e recall pari al 77%.
Notiamo anche che il Decision Tree non performa così male rispetto al KNN: infatti ha un AUC pari a 84%, e anche i valori di accuracy, precision,e recall non discostano tanto da quelli del KNN.
Infine con il Naive Bayes non otteniamo delle performance soddisfacenti, in quanto abbiamo un AUC pari al 78%, accuracy pari al 64% e F1-Score pari al 62%.

5. Pattern Mining

L'ultimo task consiste nell'estrarre i frequent itemset e di trovare association rules interessanti nel dataset.
Il pattern mining ha bisogno di un pre-processing dedicato, in quanto se utilizzassimo le variabili continue, risulterebbe alquanto difficile estrarre dei pattern frequenti. Quindi il

primo step è stato quello di discretizzare le variabili intensity, length_ms e zero_crossings_sum in 4 intervalli. Per lo stesso motivo, abbiamo deciso di raggruppare le emotion tra di loro, sulla base delle sensazioni che suscitano:

- calm, neutral e sad sono state raggruppate nella nuova variabile binaria emozioni_controllate
- fearful, disgust, surprised e happy sono state raggruppate nella nuova variabile binaria emozioni_forti.

Queste due variabili sono complementari, quindi è stata utilizzata solo la seconda.
Abbiamo inoltre inserito emotional_intensity e vocal_channel.
Su questo set di variabili abbiamo applicato l'algoritmo Apriori.

5.1 Frequent Itemsets

Abbiamo fissato a 2 il valore minimo di item per itemsets e abbiamo studiato come varia il numero di itemsets al variare del supporto, considerando i frequent itemsets, i maximal frequent itemsets e i closed frequent itemsets, come mostrato dal grafico 5.1. Possiamo notare che i maximal frequent itemsets sono un sottoinsieme degli altri due e che per supporti inferiori al 15%, abbiamo molto più frequent itemsets e closed frequent itemsets. Superata quella soglia, le tre curve si vanno a sovrapporre.

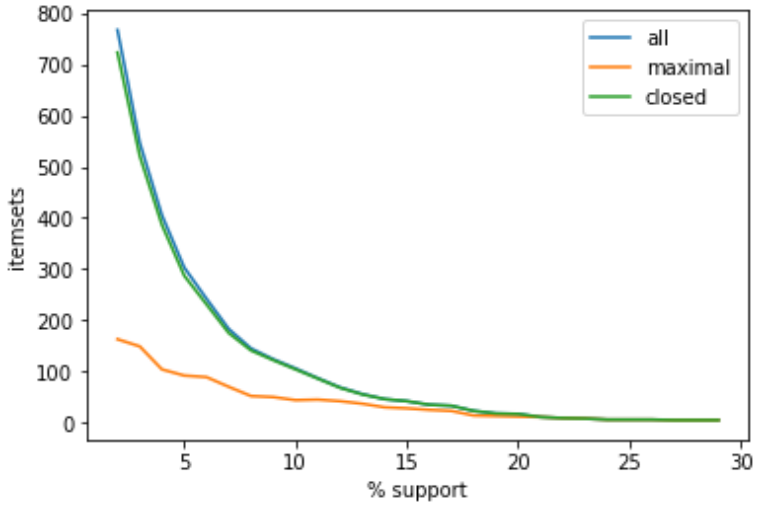


Fig 5.1

Estraendo i frequent itemsets con supporto pari al 20%, ne otteniamo 17. Di seguito riportiamo quelli che abbiamo ritenuto più interessanti:

Frequent Itemsets	Support
Vocal_channel=speech e **Emozioni forti**=1	41,4%
Emotional_intensity=strong e **Emozioni forti**=1	30,9%
Intensity in (-31.84, -12,98] e **Emotional_intensity**=strong	20,2%

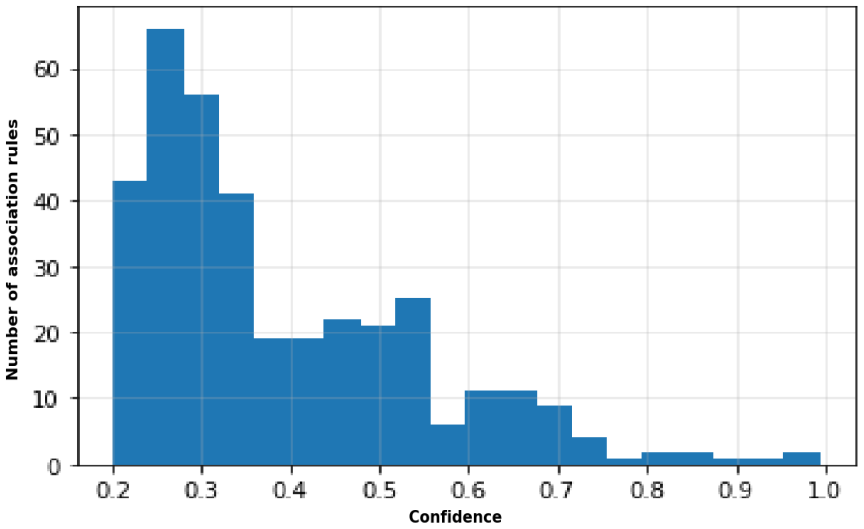


Fig 5.2

Nella figura 5.2 sono stati riportati i valori delle Association Rules per delle misure di confidence che vanno da 20 a 100 così da individuarne l'andamento.

Attraverso un istogramma è stato studiato l'andamento del numero di regole di association al variare del valore di Lift.

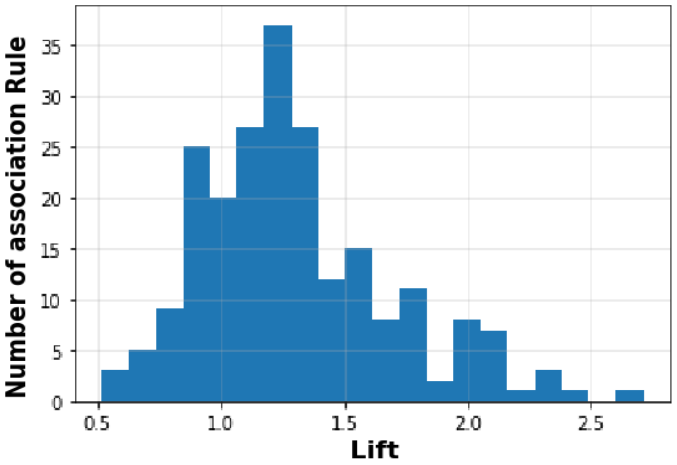
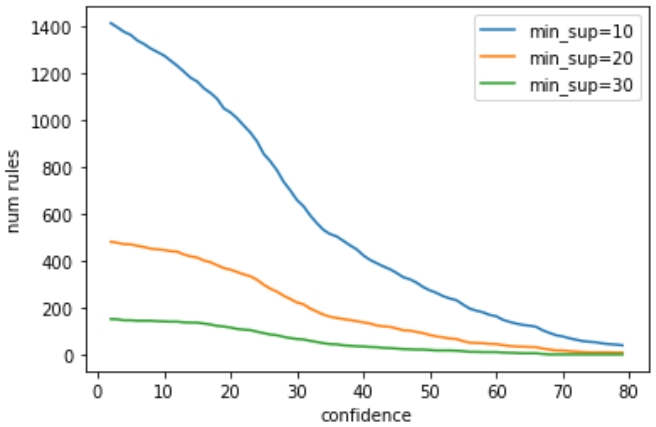


Fig 5.3

5.2 Association Rules

Abbiamo poi studiato come varia il numero di regole al variare della confidenza, con diversi valori di supporto. Anche in questo caso notiamo che per valori piccoli di confidenza, abbiamo molte regole, mentre per valori grandi ne abbiamo molte meno.



Abbiamo deciso di proseguire con supporto pari al 20% e confidenza pari al 70%. Mostriamo le prime 3 regole con lift più alto.

Antecedent	Consequent	Confidence	Lift
length_ms in (4538,6373]	vocal_channel=song	86.8%	2.31
Intensity in (-31.84, -12,98] e Emozioni forti=1	emotional_intensity=strong	81.8%	1.77
Intensity in (-31.84, -12,98]	emotional_intensity=strong	80.9%	1.75

L'association rules non è una predizione, ma un'implicazione.
In particolare la seconda regola sembra essere molto interessante, in quanto, quando intensity ha un valore alto (range (-31.84, -12,98]) e l'emozione è forte, implica che emotional_intensity=strong l'81,8% delle volte. Inoltre l'antecedent e il consequent sono correlati in quanto abbiamo un lift pari a 1.77.

Di conseguenza possiamo provare ad utilizzare questa regola per predirre la variabile target. Quindi modificando il dataset in modo che emotional intensity sia 1 (strong) quando intensity è nel range (-31.84, -12,98] e emozione forte=1, e sia 0 (normal) per tutti gli altri valori otteniamo una valutazione di questo tipo:

Accuracy	**Precision**	**Recall**	**F1-Score**
0.70	0.75	0.68	0.67

I modelli di classificazione (in particolare il Decision Tree e il KNN) creati nell'apposita sezione mostrano dei risultati migliori, tuttavia la predizione fatta tramite estrazione delle regole, sembra essere più qualitativa di quella fatta tramite l'algoritmo del Naive Bayes.