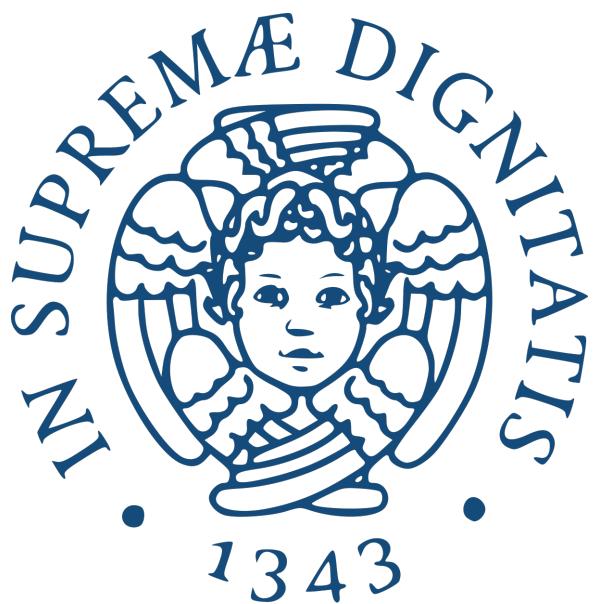


UNIVERSITY OF PISA

MASTER DEGREE IN
DATA SCIENCE AND BUSINESS
INFORMATICS



Course: Data Mining II

Project Report

Submitted by
Davide Ricci
Francesco Pio Capoccello
Alessandro Mastrorilli

a.y. 2022/2023

Contents

1	Introduzione	3
1.1	Data Understanding	3
1.2	Data Preparation	4
2	Imbalanced Learning & Dimensionality Reduction	5
2.1	Pre-processing	5
2.2	Emotion Analysis	5
2.3	Angry Emotion Analysis	6
2.4	Emotional Intensity Analisys	8
2.4.1	Feature Selection	9
3	Outlier Detection	10
3.1	Original Space	10
3.2	Projected Space	12
3.3	Conclusions	12
4	Advanced Classifiers	13
4.1	Logistic Regression	14
4.2	SVM	15
4.3	Neural Networks	16
4.3.1	Perceptron	16
4.3.2	Deep Neural Networks	16
4.4	Ensemble Models	17
4.4.1	Random Forest	18
4.4.2	Bagging	18
4.4.3	Boosting	19
4.4.4	Gradient Boosting Machine Models	19
4.5	Classification Conclusions	20
5	Advanced Regressors	21
6	Time Series	23
6.1	Data Understanding & Preparation	23
6.2	Clustering	24
6.2.1	K-Means	24
6.2.2	Gerarchico	25
6.3	Classification	26
6.4	Motifs & Discords Discovery	27

7 XAI Advanced Classifiers	29
7.1 Global Approach	29
7.1.1 Basic Decision Tree	29
7.1.2 Shap	30
7.2 Local Approach	31
7.2.1 Lime	31
7.2.2 Lore	31

Chapter 1

Introduzione

Il Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) è un set di dati in cui ogni record rappresenta la vocalizzazione di due frasi diverse pronunciate da attori diversi di entrambi i sessi, con una diversa emozione del linguaggio e un'intensità emotiva che può essere normale o forte. Il set di dati RAVDESS, in questa seconda parte di programma, è suddiviso in *training_set* e *test_set*.

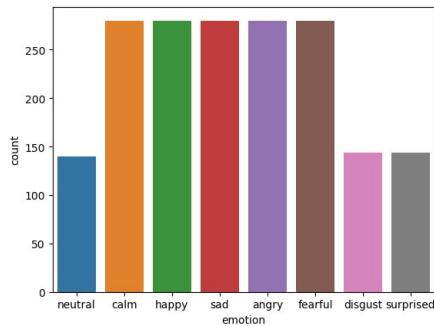
Il Training_set è composto da 1828 record (ogni record rappresenta una nota vocale) e 434 features (ogni feature rappresenta una caratteristica del relativo file audio) mentre il Test_set da 624 record e lo stesso numero di features del Training_set.

1.1 Data Understanding

Rispetto al primo modulo, nei due nuovi dataset sono presenti numerose features aggiuntive estratte dai file audio originali. In particolare sono state aggiunte una variabile categorica indicante il nome del file audio, diversa per ogni osservazione, e un ingente numero di variabili continue che possono essere divise in tre macro-categorie:

- **Statistiche di base:** Oltre alle consuete caratteristiche quali sum, mean, std, min, max, kur e skew (kurtosis e skewness) sono state inserite anche q01, q05, q25, q50, q75, q95, q99 che rappresentano i quantili della corrispondente variabile.
- **Trasformazioni:** Alle trasformazioni viste nel primo modulo(zc,mfcc,sc,stft) è stata aggiunta la caratteristica lag1 che rappresenta la differenza tra ogni osservazione e la precedente rispetto ad una determinata variabile: differenza(t) = osservazione(t) - osservazione(t-1). Inoltre per ciascuna di queste trasformazioni sono state estratte le consuete statistiche di base (somma, media, std, ...)
- **Caratteristiche temporali:** Queste caratteristiche rappresentano 4 finestre di tempo di uguali dimensioni in cui è stata suddivisa una determinata variabile. Le finestre sono indicate con la stringa w1, w2, w3 o w4. Ad esempio la variabile "stft_skew_w2" indica la skewness del cromogramma stft della seconda finestra temporale del segnale audio mentre "stft_skew" rappresenta semplicemente la skewness del cromogramma.

La principale variabile target dei successivi studi in questo secondo modulo è sempre il tipo emozionale che presenta la seguente frequenza di valori:



1.2 Data Preparation

Per affrontare i differenti task di data mining si è deciso di preparare i dati adottando la medesima metodologia conseguita nel modulo 1: pulizia errori, eliminazione di variabili con valore costante, eliminazione features ritenute non influenti per i nostri esperimenti. I dati sono stati preparati analizzando il dataset di train e in caso di modifica/cancellazione di colonne le stesse operazioni sono state conseguite anche nel dataset di test, evitando dunque qualsiasi problema in fase di testing(sul dataset di riferimento) dei modelli allenati sul dataset di train. Alla fine di questa fase sono stati infine creati due nuovi dataset in formato csv contenenti le informazioni pulite e preparate per i successivi task. Di seguito i passi principali conseguiti in questa fase:

- Riscontrata assenza valori nulli quindi nessuna alterazione dei records.
- Cancellazione variabili(52) contenenti valori costanti. In particolare in questa parte si è notato che nel dataset di test solamente 50 di queste erano costanti mentre le altre due(zc_q75_w1 , zc_q75_w2) differivano dal valore costante per 1 unico outlier. Come precisato precedentemente si è proceduto alla cancellazione di tutte e 52 le variabili anche nel dataset di test.
- Cancellazione variabili altamente correlate. In particolare è stata definita una matrice di correlazione col metodo di Pearson e sono state selezionate quelle coppie di variabili aventi una correlazione $\geq 95\%$ in valore assoluto, dopodichè si è deciso di eliminare il membro delle coppie ritenuto meno interessante in termini di quantità informativa.

Queste sono state le principali modifiche applicate ai dataset originali. Come ultima istanza si è notato che la variabile *filename* presenta valori differenti per ogni record del train e test, in quanto identifica in maniera univoca ogni osservazione e può essere considerata come una chiave primaria; si anticipa, infatti, che per i successivi esperimenti di classificazione, verrà eliminata dai dataframe generati dalla lettura dei dataset in quanto in fase di allenamento e testing dei modelli non avrebbe influenzato la capacità previsionale. Lo stesso ragionamento verrà conseguito anche per la variabile *actor* in quanto gli attori coinvolti nel dataset di train vanno da 0-18 mentre nel test da 19-24.

I nuovi dataset di train e test generati alla fine di questa fase di Data Preparation sono costituiti dunque da 261 colonne, si è dunque ridotta la complessità del caso di studio evitando potenzialmente anche eventuali problemi di *curse of dimensionality*.

Chapter 2

Imbalanced Learning & Dimensionality Reduction

In questo capitolo vengono analizzati tre esperimenti di classificazione sbilanciata utilizzando in particolare tecniche di riduzione della dimensionalità per allenare e testare i classificatori in un nuovo spazio di dati(selezionato,proiettato) oppure per visualizzare i valori bilanciati, con apposite tecniche di sampling, in uno scenario a 2 dimensioni.

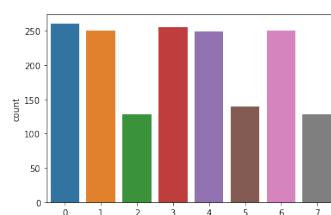
2.1 Pre-processing

Per conseguire i successivi esperimenti, i dati sono stati processati effettuando operazioni di label encoding e one-hot encoding sulle variabili categoriche; mentre per le features continue, in caso di applicazioni di metodi che operano sul concetto di distanza, i dati sono stati normalizzati utilizzando lo StandardScaler. I dataset di riferimento sono quelli preparati nel capitolo 1 e le operazioni di pre-processing sono state effettuate anche sul dataset di test. Si precisa che le tecniche di balancing sono state applicate solamente al dataset di train.

2.2 Emotion Analysis

In prima istanza si è deciso di classificare i dati in base al tipo di emozione. In particolare sono stati allenati, prima e dopo l'applicazione di tecniche di bilanciamento e riduzione della dimensionalità, due modelli basati su DT e KNN e sono state testate le performance sui dati di test. Per quanto riguarda il DT le performance migliori sono state ottenute allenando il modello sullo spazio originale ed applicando Random Undersampling come tecnica di bilanciamento. Invece per quanto riguarda il KNN, i risultati migliori sono stati ottenuti utilizzando il TomekLinks, come tecnica di undersampling, su uno spazio di 22 dimensioni generato tramite feature selection dal metodo RFE a cui è stato passato un albero di decisione come estimatore, parametrizzato opportunamente tramite grid-search. Nel seguito viene mostrato il paragone con i classificatori allenati senza tecniche di bilanciamento e riduzione dello spazio, ottenuti nel modulo 1 di Data Mining. A destra,invece, la nuova distribuzione della variabile target *Emotion*.

METODO	UNBALANCED, SPAZIO ORIGINALE	SAMPLING CON SPAZIO RIDOTTO
DT	35% Accuracy 35% F1_macro	40% Accuracy 39% F1_macro
KNN	41% Accuracy 39% F1_macro	50% Accuracy 49% F1_macro



Con le nuove tecniche le performance dei predittori sono migliorate, riuscendo a raggiungere anche il 50% di accuratezza per il modello basato sulle distanze. E' interessante notare come lo spazio di 22 dimensioni su cui ha formato meglio il modello sia stato ottenuto utilizzando come stimatore un albero di decisione anzichè lo stesso classificatore usato nel modello.

2.3 Angry Emotion Analysis

Come secondo esperimento, si è deciso di trasformare la variabile Emotion da multiclass a binaria dividendo dunque i dati in due classi: !Angry(class 0), Angry(class 1). Come si può vedere dal grafico di sinistra, la variabile risulta abbastanza sbilanciata; tale sbilanciamento è stato notevolmente accentuato facendo raggiungere alla classe minoritaria una distribuzione pari a 5.1%. In particolare il modello basato sul DT ha restituito 88% Acc. e 78% F1_macro nella configurazione originale e 95% Acc. e 70% F1_macro nella configurazione ulteriormente sbilanciata. Come ci si poteva aspettare l'accuratezza è aumentata mentre la metrica basata sulla media armonica tra precision e recall è diminuita. Stessa situazione per il KNN.

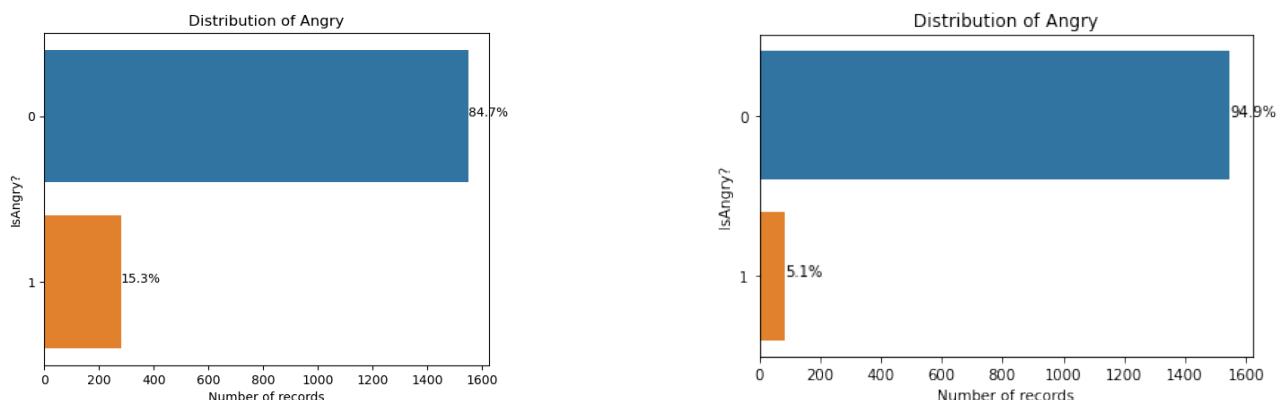
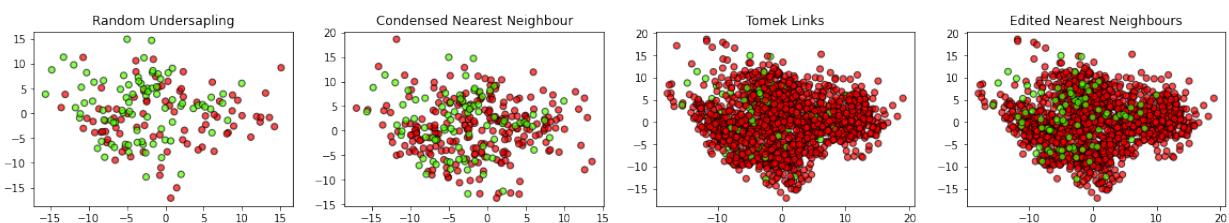
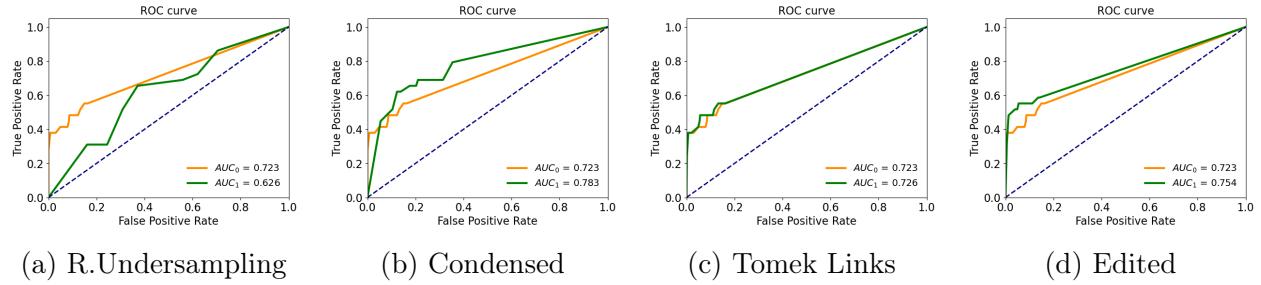


Figure 2.1: Confronto Distribuzioni

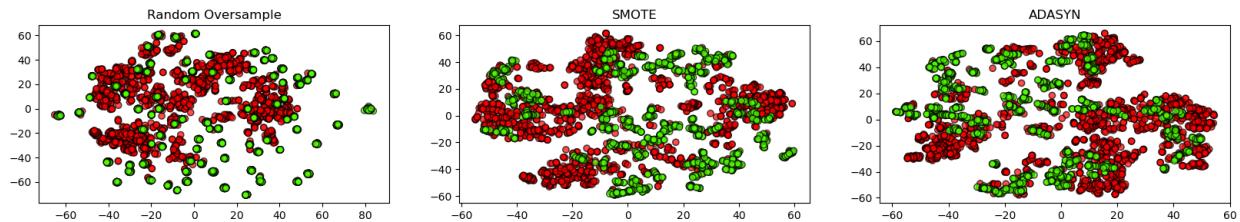
Successivamente sono state sperimentate le stesse tecniche di Undersampling(random,tomek links, condensed/edited nearest neighbours) e Oversampling(random, smote, adasyn) del primo esperimento per cercare di ribilanciare i dati in vista della creazione dei modelli di classificazione sui quali verranno allenati. Nel seguito viene mostrato come le differenti metodologie di Undersampling influenzano il numero e la distribuzione dei dati in uno spazio a 2 dimensioni ottenuto tramite PCA.



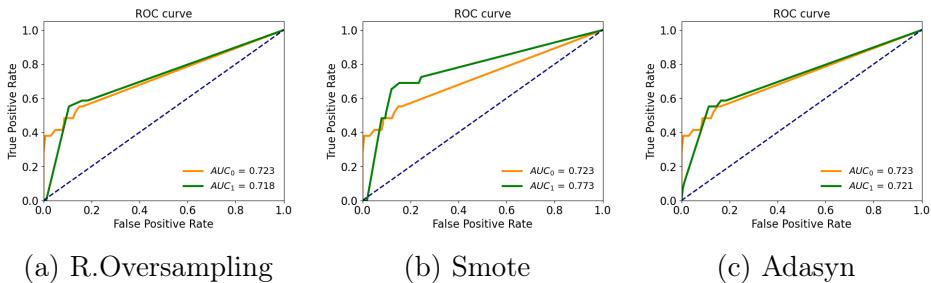
Questi metodi sono stati valutati ripetendo la classificazione con DT e KNN, analizzando in particolare come la ROC curve relativa alla classe minoritaria varia in ogni esperimento di undersampling. Nella figura sottostante sono visibili gli esperimenti conseguiti dove la curva di colore arancione rappresenta sempre la ROC della classe minoritaria relativa al DT pre-bilanciamento con un valore di AUC pari a 0.723.



Si nota dai grafici come le prestazioni, in fase di classificazione dei dati appartenenti alla classe Angry, variano per ogni tecnica. In particolare i risultati migliori si ottengono attraverso l'Edited Nearest Neighbours dove la curva riesce sempre ad essere maggiore rispetto a quella del classificatore pre-bilanciamento. Analizzando le percentuali raggiunge un valore di F1 score, della classe Angry, pari al 56% contro il 43% pre-bilanciamento. Vediamo adesso come si comportano le tecniche di Oversampling, seguendo la medesima metodologia appena vista per poi trarre una considerazione finale. In questo caso lo spazio per la visualizzazione dei dati è stato proiettato utilizzando il t-SNE.



Vediamo adesso come si comportano le ROC curve del modello basato sul DT in queste nuove configurazioni di dati bilanciati:



In questo caso, come si può vedere dai grafici sopra, il miglior risultato è stato ottenuto dal classificatore allenato sui dati bilanciati tramite Smote. Concludendo i confronti si può affermare che la tecnica di sampling che ha consentito al modello di DT di essere allenato meglio sui dati di train è quella di sotto-bilanciamento tramite Edited Nearest Neighbours.

In ultima istanza si è provato ad intervenire a livello algoritmico per il bilanciamento dei dati utilizzando inizialmente un meta-cost sensitive classifier utilizzando la seguente matrice dei costi $[1, 10, 0, 0] * \text{lunghezza}(\text{train set})$ le cui colonne corrispondono rispettivamente a $[FP, FN, TN, TP]$; e successivamente utilizzando un class weight classifier con i seguenti costi di misclassificazione: $\{0:1, 1: 7\}$. Nella tabella seguente vediamo i risultati ottenuti in fase di testing.

Intervenendo a livello algoritmico le performance sono ulteriormente incrementate, soprattutto per la classificazione della classe minoritaria.

METODI	ACCURACY	F1_MACRO	F1_MINORITARY CLASS
Meta-Cost- Sensitive	96%	73%	49%
Classic-Weight	93%	72%	49%
Classic DT	95%	70%	43%

Figure 2.4

2.4 Emotional Intensity Analisys

Il terzo esperimento prende in considerazione la feature relativa all'intensità emozionale. Come si può notare dai grafici sottostanti risulta essere abbastanza bilanciata; si è proceduto, anche in questo caso, ad incrementare lo sbilanciamento riducendo i valori della classe "strong" (classe minoritaria) fino al 5%. In questa sezione si è deciso di allenare un modello, pre e post bilanciamento, basato sul KNN per poi confrontare le relative performance. In particolare per quanto riguarda la configurazione successiva allo sbilanciamento si ottiene un'accuracy del 94% e una F1_macro del 57% contro i rispettivi 70% e 67% iniziali.

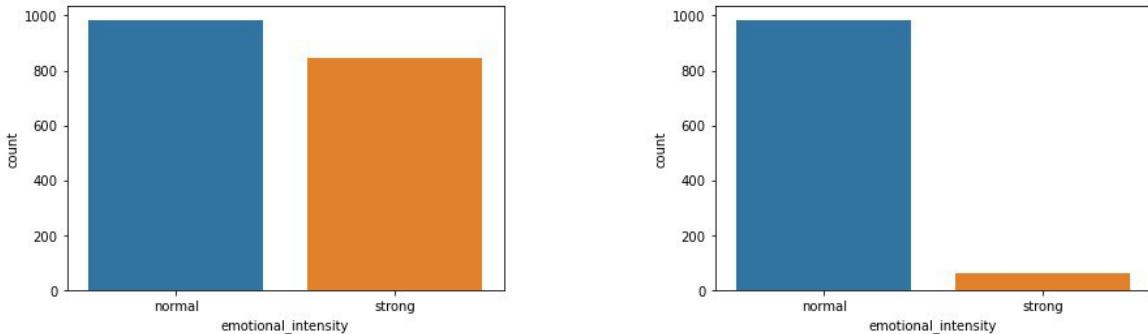


Figure 2.5: Confronto Distribuzioni

Successivamente sono state sperimentate diverse tecniche di Sampling per cercare di ribilanciare i dati in vista della creazione dei modelli di classificazione su cui allenare i dati.

Nella figura sottostante sono visibili gli esperimenti relativi alle tecniche di Undersampling, mostrando come varia la curva ROC della classe minoritaria (di colore arancione) relativa al classificatore pre-bilanciamento, con la stessa metodologia della sezione precedente.

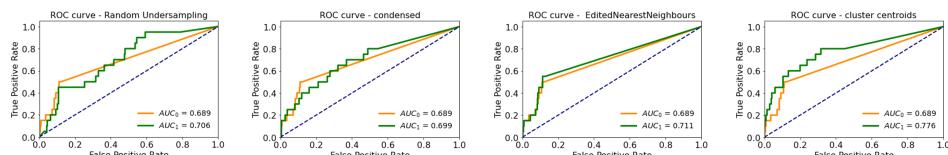
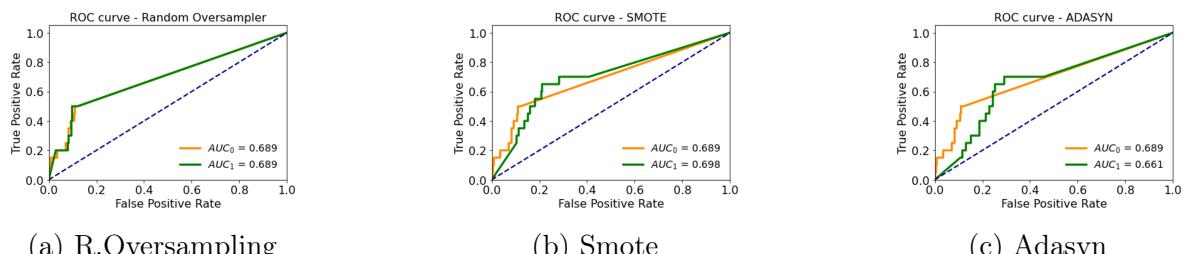


Figure 2.6: ROC Curve

I risultati migliori si ottengono con il metodo del Cluster Centroids, con un'Accuracy del 93%, una F1_macro del 67% e un F1 score della classe minoritaria del 38% (contro il 17% del modello non bilanciato). In particolare la curva ottenuta con questa metodologia riesce sempre ad essere maggiore rispetto a quella del classificatore pre-bilanciamento. Vediamo adesso come si comportano le ROC curve con le tecniche di Oversampling.



Dai grafici si può evincere che la tecnica che consente di restituire i migliori risultati sia il Random Oversampling, che presenta risultati pressoché simili rispetto alle performance ottenute senza bilanciamento.

In particolare, i risultati ottenuti sono caratterizzati da un' Accuracy del 89% , una F1_mmacro del 55% e una F1 score della classe minoritaria del 17%.

Concludendo i confronti si può affermare che la tecnica di sampling che ha consentito al modello di essere allenato meglio sui dati di train è quella di sotto-bilanciamiento tramite Cluster Centroids, restituendo anche un incremento nella classificazione della classe minoritaria.

2.4.1 Feature Selection

In quest'ultima sezione si è effettuato un cambio di dimensionalità testando le diverse tecniche di feature selection combinate con i vari metodi di Sampling, identificando lo spazio in cui il modello basato sul KNN performa meglio sui dati di test. Per la variance threshold sono state testate varie configurazioni parametriche:

$$[0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.15, 0.16, 0.25, 1]$$

Per i metodi SelectKBest e Recursive Feature Elimination si è scelta la migliore configurazione facendo variare il numero di features da 20 a 60. La tabella sottostante riassume i risultati ottenuti bilanciando il dataset con la migliore tecnica di sampling riscontrata nella sezione precedente, ovvero Cluster Centroids.

	VARIANCE THRESHOLD (0,02)	SELECT K BEST	RECURSIVE FEATURE ELIMINATION	SELECT FROM MODEL
DIMENSIONE	N.features 263	N.features 47	N.features 21	N.features 10
KNN	92% Accuracy 67% F1_macro	91% Accuracy 70% F1_macro	85% Accuracy 59% F1_macro	82% Accuracy 59% F1_macro

Figure 2.8

In questo caso, come si evince dalla tabella, i migliori risultati si ottengono con la tecnica del Select K best a 47 dimensioni, caratterizzata anche da un F1 score della classe minoritaria del 47%, andando di fatto ad aumentare notevolmente le performance(17% senza sampling e reduction).

La tabella sottostante riassume invece i risultati ottenuti bilanciando il dataset di train con Random Oversampling.

	VARIANCE THRESHOLD (1,00)	SELECT K BEST	RECURSIVE FEATURE ELIMINATION	SELECT FROM MODEL
DIMENSIONE	N.features 70	N.features 22	N.features 30	N.features 22
KNN	77% Accuracy 57% F1_macro	80% Accuracy 57% F1_macro	83% Accuracy 60% F1_macro	80% Accuracy 60% F1_macro

Figure 2.9

Da quest'ultima analisi, considerando anche l' F1 score della classe minoritaria pari al 31% , il metodo migliore risulta essere la Recursive Feature Elimination , andando anche in questo caso di fatto ad aumentare le performance. In conclusione si può affermare che attraverso la riduzione della dimensionalità, selezionando un sottoinsieme di feature, per il nostro caso di studio i modelli performano meglio.

Chapter 3

Outlier Detection

In questa sezione viene affrontato il task di Outlier Detection utilizzando i dataset preparati nel capitolo 1; in seguito viene fatto un esperimento sul dataset di train dimensionalmente ridotto per ulteriori confronti finali tra i risultati ottenuti.

3.1 Original Space

Nel primo esperimento di ricerca di valori anomali sono stati utilizzati metodi e approcci differenti:

- Distance-based (KNN)
- Density-based (LOF, DBSCAN)
- Angle-based(ABOD)
- Model-based(Isolation Forest)

Per alcuni metodi si è deciso di testare differenti valori parametrici come ad esempio k del KNN(5,10) e n_neighbours dell'ABOD(5,10); per il DBSCAN sono stati utilizzati invece eps=10 e min_samples=5. Nella tabella a sinistra (Fig 3.1), viene mostrato il numero di outliers rilevati per ogni tecnica. Come si può vedere dai risultati del DBSCAN, nonostante una fase di ricerca dei parametri migliori basata sulla distanza media dei punti rumorosi trovati e il numero di cluster generati, il metodo non ha formato come sperato, rilevando circa il 40% dei dati come punti anomali. Le altre tecniche hanno invece restituito risultati più accettabili. Per poter confrontare i risultati ottenuti dai vari metodi è stata effettuata prima un'analisi quantitativa attraverso l'intersezione tra i top 10 outliers trovati tra diverse coppie di algoritmi, notando che alcune osservazioni sono state classificate come valori anomali per ciascun incrocio. Nella tabella di destra, sono presenti anche le intersezioni tra i top 1%.

METODI	N. OUTLIER
KNN_5N	164
KNN_10N	176
DBSCAN	690
LOF	69
ABOD_5	219
ABOD_10	187
ISOLATION FOREST	54

METODI	INTERSEZIONE TOP 1%	INTERSEZIONE TOP 10
LOF_KNN_5N	10	7
LOF_KNN_10N	13	6
LOF_ABOD_5N	9	5
LOF_ABOD_10N	12	8
KNN_5N_ABOD_5N	9	4
KNN_10N_ABOD_10N	10	5

E' stata applicata la PCA a 2 dimensioni per visualizzare come vengono differenziati i valori anomali dagli inliers. Nella figura sottostante, l'analisi per il LOF e l' Isolation Forest che, rispetto agli altri metodi, riescono a distinguere meglio le osservazioni:

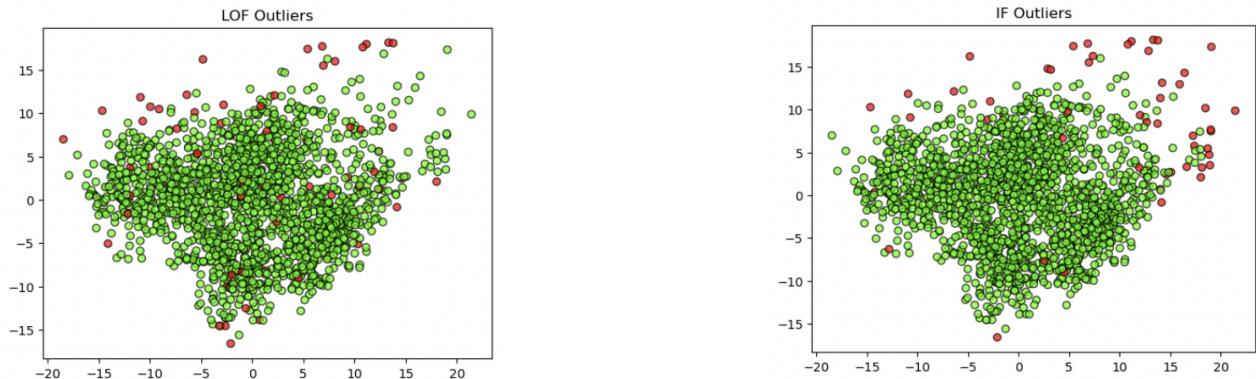


Figure 3.2: Lof vs Isolation Forest

Si è dunque effettuata un'analisi qualitativa andando ad esplorare l'insieme di osservazioni risultante dall'intersezione di tutte le tecniche testate per la rilevazione degli outliers. In particolare sono 21 records e condividono le seguenti proprietà:

- Prevalenza tipo emozionale "angry", "fearful" mentre assenza di emozioni di tipo "surprised", "happy", "neutral", "disgust".
- Prevalenza attori maschili (77%).
- Prevalenza intensità emozionale di tipo strong.

Si è ritenuto interessante utilizzare l'approccio model-based basato sull'Isolation Forest e allenato sui dati del dataset di training, applicandolo a dati non osservati e quindi sul dataset di test, restituendo come valori anomali un totale di 21 osservazioni, che corrispondono a circa il 3% dei dati totali. Nel seguito viene mostrato l' istogramma delle frequenze (Fig 3.3); in particolare il punteggio di outlierness più alto è di -0.036.

Come analisi finale si è deciso di selezionare le 21 osservazioni, rilevate come outliers dall' intersezione dei risultati di ogni tecnica di detection, e di cancellarle dal dataset di train. Sono stati successivamente allenati due modelli basati sul DT e KNN per la classificazione dei dati utilizzando come variabile target le emozioni, per confrontare le performance sul test set rispetto ai classificatori delle sezioni precedenti. Le performance dei classificatori sono state analizzate sul dataset di test e, come si può vedere dalla tabella, sono incrementate di alcuni punti percentuale per il modello basato sul DT; mentre il modello basato sulle distanze non è stato influenzato, in fase di training, dalla cancellazione dei valori anomali.

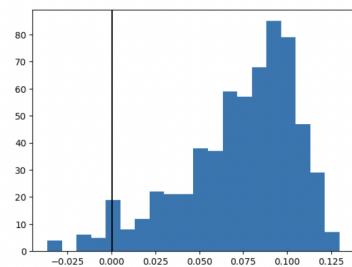


Figure 3.3

MODELLI	TRAINED WITH OUTLIERS	TRAINED WITHOUT OUTLIERS
DT	35% Accuracy 35% F1_macro	37% Accuracy 37% F1_macro
KNN	41% Accuracy 39% F1_macro	41% Accuracy 39% F1_macro

Figure 3.4

3.2 Projected Space

In questa sezione sono stati condotti alcuni esperimenti prendendo in considerazione uno spazio proiettato a 30 dimensioni delle variabili, attraverso l'MDS (multi-dimensional scaling). E' stato scelto tale numero in quanto si è riscontrato che la maggior varianza dei dati viene catturata dalle prime 30 componenti.

Per questo approccio sono stati utilizzati alcuni dei metodi precedenti; in particolare per le tecniche basate sulla distanza e sugli angoli il numero di vicini è stato impostato a 5, mentre per quanto riguarda le altre tecniche, i parametri utilizzati sono i medesimi del precedente esperimento.

Nella tabella viene mostrato il numero di outliers rilevati per ogni tecnica:

METODI	N. OUTLIER
KNN	168
LOF	27
ABOD	207
ISOLATION FOREST	272

(a) Outliers MDS

Dall'intersezione del risultato di tutte le tecniche utilizzate sono emersi 25 record con in comune principalmente la feature emotiva "calm" e l'assenza del valore "surprised".

3.3 Conclusions

Per concludere questo task si è effettuato un confronto tra i risultati ottenuti dall'intersezione dei metodi nello spazio originale e in quello proiettato, ottenendo 3 istanze che condividono le seguenti proprietà:

- "Kids are talking by the door" come valore della variabile Statement.
- Ripetizione=1 della stringa contenuta nello Statement.
- "calm, "fearful", "neutral" come valori emozionali.

In linea generale non sono state trovate molte proprietà in comune tra i risultati dei due differenti esperimenti. Il cambio di dimensionalità dello spazio dei dati, a un numero molto più piccolo, potrebbe aver condizionato molto la rilevazione dei valori anomali per le varie tecniche provate. E' interessante notare come valori prevalenti negli esperimenti condotti sullo spazio originale(emozione "angry" e intensità emozionale strong) siano pressochè assenti o presenti in minima parte.

Chapter 4

Advanced Classifiers

In questo capitolo viene affrontato nuovamente il task di classificazione dei dati in base al tipo emozionale, definendo modelli più avanzati e cercando di incrementare le performance dei classificatori base visti nel Modulo 1. I dataset di riferimento sono quelli preparati nel primo capitolo. Per ogni modello, come mostrato nella figura sottostante, è stata definita una griglia di iperparametri con cui fare tuning. Nell'analisi dei vari classificatori verrà specificata la configurazione di parametri migliore ottenuta in fase di validazione, ponendo particolare attenzione alla metrica F1_macro in quanto la variabile da predire è multiclasse e sbilanciata. Inoltre si è provato ad incrementare ulteriormente le performance dei modelli, allenandoli in uno spazio selezionato tramite RFE a 22 dimensioni, su un insieme di dati a cui è stata applicata la migliore tecnica di sampling riscontrata(Tomek Links), in quanto nella sezione 2.2 questi approcci hanno restituito i risultati migliori. Nelle seguenti sezioni ci limiteremo a riportare e commentare i risultati ottenuti per poi confrontarli nella sezione conclusiva.

SVC	LINEAR SVC
kernel : (linear, rbf, poly, sigmoid), C: (0.001, 0.05, 0.01, 0.1, 1.0, 10.0, 50, 100.0), tol: (1.0, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6), decision_function_shape:[ovo, ovr]	tol:(1.0, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6), C: (0.001, 0.05, 0.01, 0.1, 1.0, 10.0, 50, 100.0), loss : [hinge, squared_hinge], multi_class:[ovr,crammer_singer]
LOGISTIC REGRESSION	MULTILAYER PERCEPTION
tol:(1.0, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6), C: (0.001, 0.05, 0.01, 0.1, 1.0, 10.0, 50, 100.0), multi_class : [auto,ovr,multinomial], solver:[lbfgs, liblinear, newton-cg, newton-cholesky, sag, saga]	hidden layer sizes: (8),(16),(32),(64),(8-16),(16-32), (32-64),(32-16-8),(64-32-16),(128-64-32) activation: relu, tanh momentum: 0.1, 0.3, 0.5, 0.7, 0.9, 1.1 Learning rate: constant, adaptive alpha: 1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001 tol: 1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001
SINGLE PERCEPTRON	DEEP NEURAL NETWORK
penalty: l1,l2,elasticnet alpha: 1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001 tol: 1.0, 0.1, 0.01, 0.001, 0.0001, 0.00001	Loss: sparse categorical crossentropy, categorical crossentropy, mean squared error Optimizer: Sgd, Adam
BAGGING	ADABOOST
n_estimators: [50,800,900]	n_estimators: [50,800,900]
RANDOM FOREST	CATBOOST
n_estimators: [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]	n_estimators: 100, 150, 300 Max_depth: 3,4,5,10 Learning rate: 0.1, 0.2, 0.3 L2_leaf_reg: 1,3,5

Figure 4.1

4.1 Logistic Regression

Dall'esecuzione della *Grid Search* è stata ottenuta la seguente configurazione di iperparametri per eseguire un'efficiente classificazione mediante regressione logistica:

PARAMETRI	DESCRIZIONE	VALORI
C	Coefficiente di penalizzazione	0,1
Solver	Algoritmo da utilizzare per il problema di ottimizzazione	Newton-cg
Multi_class	Metodi per la multi-classificazione	Ovr
Tol	Tolleranza come criterio di stop	1.0

Figure 4.2

I parametri raffigurati in tabella sono stati scelti per i seguenti motivi:

- **C:** Per penalizzare gli esempi miss-classificati. Un corretto valore di C può, quindi, influire positivamente sulle prestazioni del classificatore
- **Solver:** Trattandosi di una classificazione multiclasse, il solver è utile per minimizzare e ottimizzare l'errore del classificatore.
- **Multi_class:** Parametro ottimale per scegliere la giusta metodologia da applicare ad un problema multiclasse.
- **Tol:** Necessario per trovare il giusto criterio di stop per l'algoritmo di ottimizzazione.

Emozioni	Precision	Recall	F1-Score
Angry	0,59	0,82	0,69
Calm	0,50	0,72	0,59
Disgust	0,49	0,42	0,45
Fearful	0,58	0,35	0,44
Happy	0,43	0,31	0,36
Neutral	0,44	0,65	0,52
Sad	0,46	0,28	0,35
Surprised	0,48	0,52	0,50

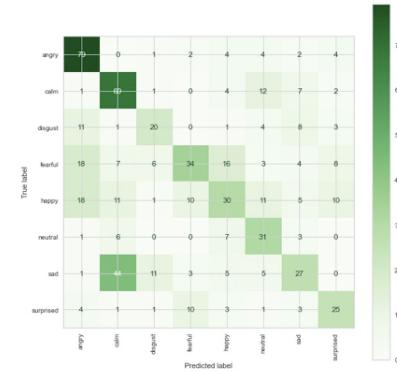


Figure 4.3

In figura sono rappresentate le performance, per ogni classe, ottenute dal modello in fase di testing. In particolare, la percentuale di accuracy è pari al 50% mentre la percentuale di F1_macro è pari al 49%.

Come si evince sia dalle metriche di classificazione che dalla confusion matrix le emozioni "angry" e "calm" sono quelle classificate in modo migliore rispetto alle altre (rispettivamente 79 e 69 istanze classificate correttamente), mentre, come si evince dalla recall, l'emozione "sad" risulta essere la peggiore in termini di percentuale relativa alla correttezza di classificazione (con 44 istanze classificate come "calm" e 11 classificate come "disgust").

4.2 SVM

Per questa tipologia di modelli sono state testate una tecnica *lineare* e una *non lineare*.

Seppur di poco, l'approccio non lineare, per i risultati ottenuti, è stato reputato il migliore tra i due.

Come si evince dalla tabella sottostante, che riporta la migliore configurazione di iperparametri trovata, il kernel ottenuto è lineare, a dimostrazione del fatto che i risultati dei due approcci per questo caso di studio risultano essere simili.

PARAMETRI	DESCRIZIONE	VALORI
C	Coefficiente di penalizzazione	0,1
Kernel	Tipo di kernel da utilizzare nell'algoritmo	Linear
Decision_function_shape	Metodi per la multi-classificazione	Ovo
Tol	Tolleranza come criterio di stop	1.0

Figure 4.4

I parametri raffigurati in tabella sono stati scelti per i seguenti motivi:

- **Kernel:** Utile per trovare la funzione kernel adatta per mappare lo spazio originale in uno spazio a più dimensioni, se il kernel è non lineare.
- **Decision _ function _ shape:** Parametro ottimale per scegliere la giusta metodologia da applicare ad un problema multiclass.
- **Tol e C** descritti nella sezione precedente

Emozioni	Precision	Recall	F1-Score
Angry	0,66	0,73	0,69
Calm	0,54	0,78	0,64
Disgust	0,48	0,50	0,49
Fearful	0,62	0,38	0,47
Happy	0,45	0,36	0,40
Neutral	0,38	0,69	0,49
Sad	0,40	0,26	0,32
Surprised	0,58	0,54	0,56

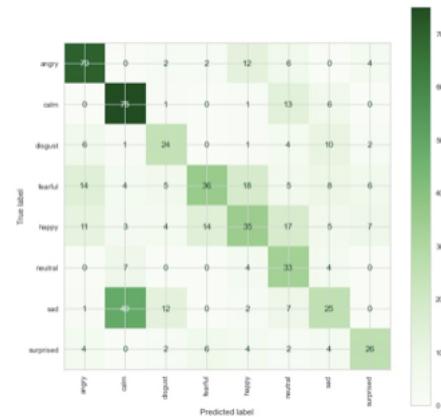


Figure 4.5

In figura vengono rappresentate le performance, sulle varie classi, utilizzando le medesime metriche degli altri esperimenti. Inoltre, la percentuale di accuracy ottenuta in questo caso è pari al 52% mentre la percentuale di F1_macro raggiunge il 51%.

Come si può notare le emozioni "angry" e "calm" sono quella classificate in modo migliore rispetto alle altre (rispettivamente 70 e 75 istanze classificate correttamente), mentre osservando i valori della recall, l'emozione "sad" risulta nuovamente essere la peggiore in termini di percentuale relativa alla correttezza di classificazione (con 49 istanze classificate come "calm" e 12 classificate come "disgust").

4.3 Neural Networks

In questo paragrafo si è cercato di costruire diversi modelli di reti neurali artificiali partendo da modelli semplici e senza layer nascosti come il Perceptron fino a definire strutture più complesse definite "profonde" che fanno uso di più livelli di layer intermedi, all'occorrenza regolarizzati mediante opportune tecniche per cercare di evitare situazioni di overfitting.

4.3.1 Perceptron

Come primo esperimento è stata definita una semplice rete neurale senza livelli nascosti e con la funzione di attivazione segno(Perceptron). Successivamente si è deciso di modificare tale struttura inserendo un numero variabile di livelli e unità intermedie(Multilayer Perceptron) per cercare di capire come le differenti versioni si comportano in fase di train e test. I parametri testati sono stati scelti principalmente per cercare di aggiustare correttamente i pesi dei layer intermedi e per gestire potenziali eventi di overfitting. Nella seguente tabella viene mostrata la migliore configurazione riscontrata in fase di validazione e le relative performance ottenute su dati non osservati.

Neural Network	Best Parameters	Hidden Layers Structures	Accuracy	F1-Score
Perceptron	Alpha = 0,0001 penalty = 'l1' Tol = 0,001	{}	46%	45%
MLPerceptron (1 hidden layer)	momentum = 0,9 learning_rate = adaptive alpha = 1,0 tol = 0,00001	{(64-tah)}	53%	52%
MLPerceptron (2 hidden layer)	momentum = 0,9 learning_rate = adaptive alpha = 1,0 tol = 0,0001	{(32-tah, 64-tah)}	55%	53%
MLPerceptron (3 hidden layer)	momentum = 0,7 learning_rate = constant alpha = 0,1 tol = 0,001	{(128-tah), (64-tah), (32-tah)}	52%	51%

Si evince dalla tabella che il miglior modello è quello strutturato su due livelli intermedi, il quale riesce a raggiungere anche il 55% di Accuracy e il 53% di F1_macro. Si è notato inoltre, durante l'allenamento dei vari modelli, che aumentare la complessità in termini di numero di layers e unità non porta miglioramenti e che il valore di Loss tende ad azzerarsi in media verso la 175esima epoca.

4.3.2 Deep Neural Networks

Sono stati definiti diversi modelli neurali di tipo sequenziale che differiscono in base al numero di livelli intermedi(tutti di tipologia Dense) e di unità dedicate a ciascuno di essi. In particolare sono state provate diverse combinazioni prima di definire le seguenti strutture che condividono il solito output layer a 8(numero classi emotion) unità:

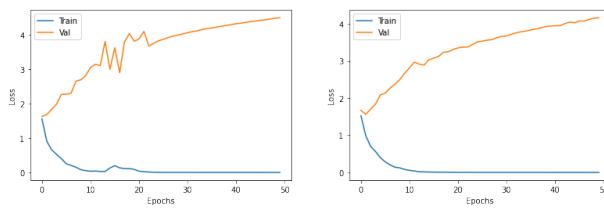
- **Model1:** Input layer(512-tanh) + 4*Hidden Layer(64-relu)
- **Model2:** Input layer(1024-tanh) + 8*Hidden layer(512-relu)
- **Model3:** Input layer(512-tanh) + 1*Hidden layer(512-tanh)
- **Model4:** Input layer(512-relu) + 1*Hidden layer(num.features-relu)

Come funzione di Loss è stata utilizzata "sparse_categorical_crossentropy" per il nostro caso multiclass e "adam" come ottimizzatore dei pesi della rete. Nel seguito sono visibili i risultati ottenuti sul test set dove i modelli 3 e 4 hanno performato meglio:

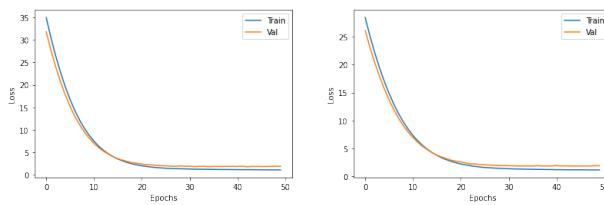
	Accuracy	F1_score
Model 1	50%	49%
Model 2	48%	49%
Model 3	52%	50%
Model 4	52%	51%

Figure 4.6

Si è notato che in linea generale, come per il caso del Perceptron, aumentare la complessità della rete non portava benefici ma anzi peggiorava le prestazioni sui dati non osservati e soprattutto la discrepanza già netta tra la curva loss del train e del validation per i vari modelli. Infatti, come si evince da grafici di seguito dei migliori modelli definiti(3-4), le reti neurali soffrono di overfitting.



Per limitare questo problema sono stati aggiunti all'interno delle due strutture un livello di Dropout=0.05 prima dell' output layer e sono stati inseriti i coefficienti=0.04 di regolarizzazione L2 per il layer di input e quello nascosto. Si è successivamente lavorato sul corretto numero di esempi di training,identificati dal parametro batch_size,su cui allenare la rete per ogni epoca, selezionando un valore pari a 400. Nel seguito i risultati dopo le modifiche apportate:



Con questa configurazione possiamo affermare che le due reti siano state allenate correttamente mediante un opportuno uso dei parametri messi a disposizione; purtroppo si è notato un decremento di circa 2 punti percentuali per l'accuracy e F1 macro una volta ritestati i modelli su dati non osservati.

4.4 Ensemble Models

In questa sezione sono stati sperimentati diversi modelli Ensemble: Random Forest, Bagging, AdaBoost e Gradient Boosting Machine. Si sono confrontati i risultati ottenuti dalle varie tecniche, tenendo conto del diverso numero di estimatori utilizzati. Dal momento che la computazione di questi algoritmi è abbastanza complessa, nella ricerca degli iperparametri sono stati testati diversi numeri di estimatori senza l'utilizzo della *Grid Search* (questo per quanto riguarda il *Bagging* e l'*AdaBoost*). Per quanto concerne i parametri degli estimatori base, sono stati utilizzati gli stessi individuati nelle sezioni precedenti. Si anticipa che, in linea generale, all'aumentare del numero di estimatori, le performance sui dati di test incrementavano fino a stabilizzarsi e all'occorrenza peggiorare.

4.4.1 Random Forest

Dalla *Grid Search* è risultato che il numero migliore di estimatori è pari a 1000. Per migliorare le performance del classificatore costruito si è deciso di utilizzare la RFE passando come modello base il random forest stesso, contrariamente rispetto al classico Decision Tree utilizzato nelle sezioni precedenti e successive. Solamente in questo caso, tale approccio è riuscito ad incrementare le performance del modello. Di seguito sono riportati i risultati ottenuti con questa tecnica:

Emozioni	Precision	Recall	F1-Score
Angry	0,58	0,84	0,69
Calm	0,47	0,74	0,58
Disgust	0,60	0,44	0,51
Fearful	0,49	0,23	0,31
Happy	0,41	0,48	0,44
Neutral	0,48	0,29	0,36
Sad	0,40	0,23	0,29
Surprised	0,47	0,58	0,52

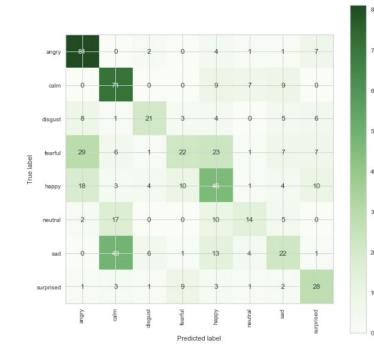


Figure 4.9

La percentuale di accuracy ottenuta in questo caso è pari al 49% mentre la percentuale di F1_macro raggiunge il 46%. Rispetto ai risultati precedenti l'emozione "happy" risulta essere classificata meglio (con 46 istanze classificate correttamente), a discapito dell'emozione "neutral" classificata peggio rispetto ai risultati precedenti, in cui solamente 14 istanze vengono classificate correttamente. Attraverso la figura 4.10 è possibile individuare le 15 features più importanti risultate nella costruzione del modello, che portano maggior guadagno in termini di information gain per i vari split e che quindi sono in grado di discriminare meglio la variabile target in considerazione. Le più importanti tra queste sono "zero_crossings_sum" e "lag1_skew" che ottengono un netto distacco da tutte le altre.

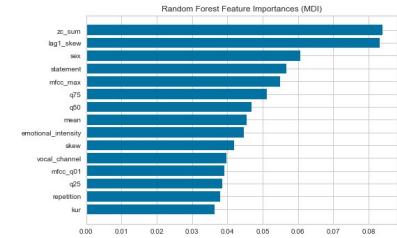


Figure 4.10

4.4.2 Bagging

Pe questa tipologia di classificatori sono stati provati come estimatori base l'svc, la logistic regression, il random forest classifier, il perceptron e il multi layer perceptron con due hidden layers. Quest'ultimo ha restituito le performance migliori con un numero di estimatori pari a 50. Di seguito i risultati:

Emozioni	Precision	Recall	F1-Score
Angry	0,65	0,83	0,73
Calm	0,54	0,72	0,62
Disgust	0,46	0,50	0,48
Fearful	0,57	0,33	0,42
Happy	0,43	0,32	0,37
Neutral	0,43	0,75	0,55
Sad	0,59	0,35	0,44
Surprised	0,55	0,58	0,57

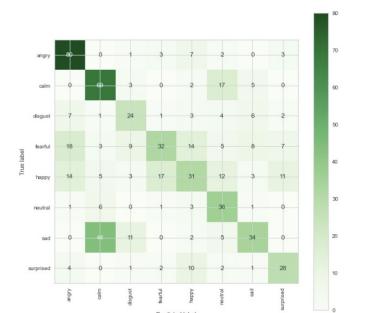


Figure 4.11

La percentuale di accuracy ottenuta in questo caso è pari al 54% mentre la percentuale di F1_macro raggiunge il 52%. Come per i risultati precedenti, l'emozione classificata meglio risulta essere "angry", ma si segnala un miglioramento generale di classificazione anche per quelle emozioni che precedentemente presentavano risultati mediocri.

4.4.3 Boosting

L'algoritmo utilizzato per questo esperimento è l'*AdaBoost*. Gli estimatori base sono gli stessi testati nel Bagging ma si è provata anche la versione di default con i decision tree stumps. Il migliore è risultato essere il modello che ha come estimatore base il random forest, con 50 estimatori. Nel seguito i risultati per questo approccio :



Figure 4.12

La percentuale di accuracy ottenuta in questo caso è pari al 52% mentre la percentuale di F1_macro raggiunge anch'essa il 52%. Anche in questo caso l'emozione "angry" risulta essere la migliore, ma rispetto ai modelli precedenti le emozioni "happy" e "sad" risultano essere classificate meglio dell'emozione "calm" (rispettivamente con 51 e 48 istanze classificate correttamente).

4.4.4 Gradient Boosting Machine Models

Come esperimento aggiuntivo sono stati provati i modelli definiti Gradient Boosting Machines che sfruttano il Gradiente per minimizzare la funzione di Loss, e alcune varianti. Dopo un'iniziale ricerca degli iperparametri per ogni tecnica provata, le performance migliori sui dati di test sono state ottenute dalla variante CatBoost. Un risultato inatteso e per niente positivo è stato ottenuto invece dal HistGradientBoosting che ha performato in maniera pessima in questo caso di studio non arrivando nemmeno al 30 % di f1_macro. Come si può notare dalla tabella riassuntiva di destra i restanti modelli hanno ottenuto risultati simili riscontrando però gli stessi difetti di misclassificazione per la classe "sad".

Parameter	Values
N.Estimators	500
Max_depth	5
Learning_rate	0,1
L2_leaf_reg	3

GBM Model	Accuracy	F1_score
GradientBoost	49%	47%
XGBoost	50%	48%
LGBM	49%	48%
CatBoost	53%	51%

Figure 4.13

Nella tabella di sinistra sono visibili invece per il modello CatBoost la configurazione migliore di parametri. In particolare si è utilizzato "*L2_leaf_reg*" come coefficiente di regolarizzazione

e si è notato che all'incrementare del numero degli estimatori le performance sui dati di test iniziavano a decrementare e soprattutto che allenando il modello su dati di training processati tramite Tomek Links le performance peggioravano di qualche punto percentuale.

4.5 Classification Conclusions

In questa sezione conclusiva vengono messi a paragone i migliori modelli, per ogni approccio testato, che hanno riscontrato buone performance sui dati non osservati. Per quanto riguarda gli approcci Ensemble e i modelli di Rete Neurale Artificiale costruiti, i migliori classificatori opportunamente parametrizzati, sono stati la variante CatBoost e il Perceptron a più livelli(2). Si puntualizza però che nell' esperimento ensemble del bagging con estimatore base il MLP a due livelli sono state ottenute performance simili al CatBoost. Nel seguito il confronto sulle metriche principali:

	Accuracy	F1_macro	Macro-Average AUC
Logistic regression	50%	49%	0,87
SVC	52%	51%	0,88
CatBoost	53%	51%	0,89
MLPerceptron (2 hidden layers)	55%	53%	0,88

Figure 4.14

Si precisa che, per determinati modelli, se allenati su dati di training preprocessati tramite Undersampling(Tomek Links) e Feature selection(RFE a 22 dimensioni), le performance migliorano di circa il 2% in accuracy e F1_Macro; infatti nella tabella sopra le performance si riferiscono direttamente a questo esperimento. Per ogni classificatore è stata realizzata una "Macro averaged Roc curve" perchè più significativa per il nostro caso di studio a 8 classi e per migliori confronti tra i vari esperimenti conseguiti:

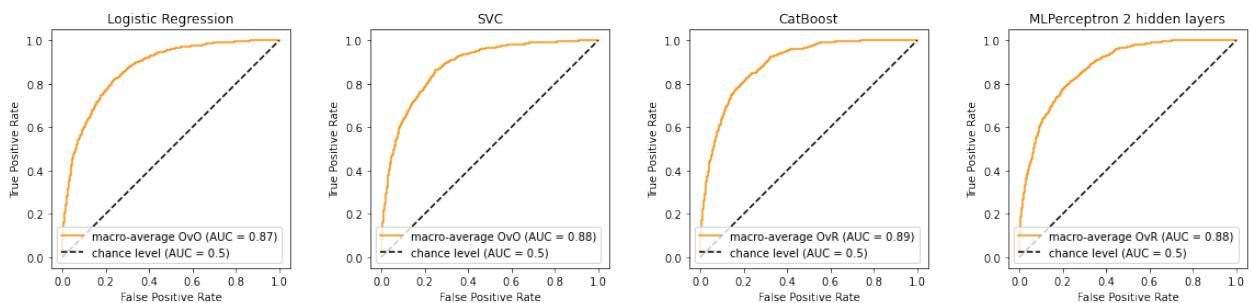


Figure 4.15: ROC Curve

In conclusione si può affermare che, se correttamente sperimentato, ogni approccio conseguito ha permesso di costruire un modello che riesca a classificare con buone performance, contando anche le problematiche relative al problema multiclass, i dati in base al tipo emozionale. Per il nostro caso di studio, un modello di rete neurale non troppo complesso ma correttamente parametrizzato e con 2 livelli nascosti, ha ottenuto i risultati migliori riuscendo ad arrivare al 55% di accuratezza sui dati di test; in linea generale però i diversi migliori modelli definiti riescono a comportarsi al solito modo sulla porzione dei dati di test restituendo all'incirca i soliti risultati. Risultati che, se paragonati ai classificatori "base" visti nel modulo 1, sono notevolmente migliorati.

Chapter 5

Advanced Regressors

In questo capitolo viene affrontato il Task di regressione multipla della variabile "frame_count" utilizzando modelli complessi ma anche molto efficienti. I dataset di riferimento, rispetto ai capitoli precedenti, sono quelli originali a cui è stato applicato un opportuno pre-processing per questi nuovi esperimenti. In particolare sono state cancellate quelle feature("length_w1/w2/w3/w4") risultate essere altamente correlate con la variabile che si è cercato di regredire e altre variabili altamente correlate tra di loro per evitare problemi di multicollinearità, seguendo la strategia vista nel capitolo 1. I primi esperimenti sono stati eseguiti utilizzando diversi sottoinsiemi di variabili indipendenti ma i risultati migliori sono stati ottenuti mediante l'impiego di tutte le variabili ad esclusione della target. Sono stati allenati e testati 4 modelli cercando di ottenere la configurazione migliore di parametri, in fase di validazione, attraverso una grid search:

- SVR(kernel=rbf, C=0.7)
- AdaBoost Regressor(n_estimators=100, loss=square)
- GradientBoosting Regressor(n_estimators=100, max_depth=10, learning_rate=1.0)
- Randomforest Regressor(n_estimators=50, criterion=square, min_samples_split=2, min_samples_leaf=1)

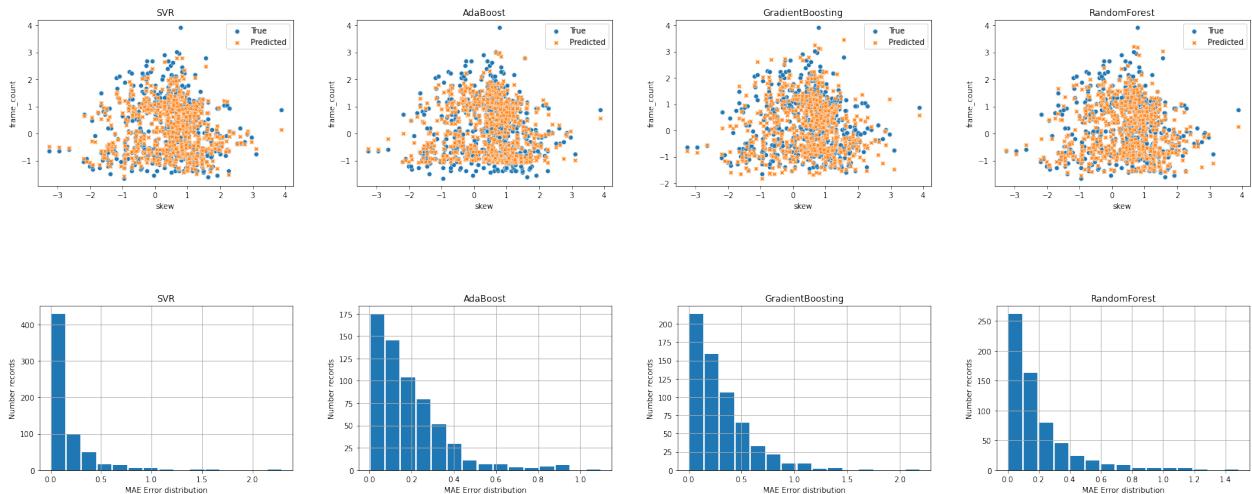
Nella tabella seguente vengono mostrati i risultati, in fase di testing, dei modelli mediante opportune metriche:

MODEL	R2	MSE	MAE
SVR	0,925	0,076	0,164
AdaBoost	0,938	0,064	0,938
Gradient Boosting	0,840	0,163	0,300
Random Forest	0,921	0,081	0,192

Figure 5.1

Rispetto al precedente capitolo di classificazione dei dati per tipo emozionale, in questo esperimento di regressione le performance dei modelli sono molto buone. In particolare l'AdaBoost ottiene il più alto valore per la metrica R2; invece concentrandosi sul calcolo della media dei residui in valore assoluto il punteggio migliore lo ottiene l' SVR. Nel seguito vengono mostrate le capacità predittive dei 4 modelli e la relativa distribuzione degli errori.

Si può notare dagli histogrammi che, in linea generale, la previsione dei valori è molto buona, benchè non ottima, poichè vi è un'elevata presenza di errori principalmente minori di 0.4



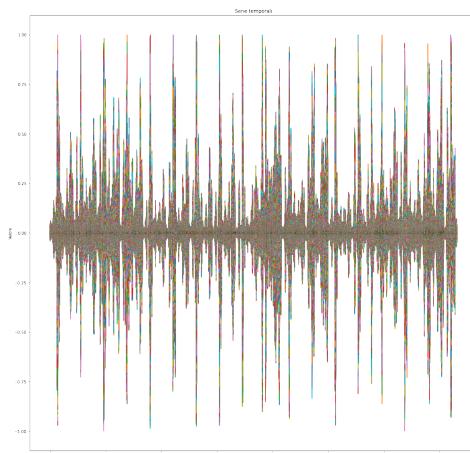
Chapter 6

Time Series

6.1 Data Understanding & Preparation

Per definire il nuovo dataset di serie temporali, sono stati importati e memorizzati tutti i file audio in formato .wav all'interno di una lista e successivamente, attraverso un'opportuna routine, sono stati convertiti in array di valori continui che rappresentano l'evolversi del segnale audio rispetto al tempo. Per evitare eventuali problemi durante il conseguimento dei successivi task, le serie temporali sono state rese di uguale lunghezza attraverso un'operazione di padding. Utilizzando il numero dell'attore che interpreta il corrispondente audio è stata poi conseguita la divisione delle serie temporali in due array di train e test. Per quanto riguarda le features categoriche , esse sono state trattate diversamente: sono state lette dai corrispondenti file .csv forniti ed inserite all'interno di due ulteriori array, sempre per il train e test. Dopo che i dati sono stati importati correttamente, si è proceduto con il controllo dei valori nulli riscontrando 0 valori nulli all'interno degli array rappresentanti le feature categoriche sia per il train che per il test. Analizzando invece i valori continui delle serie temporali, si è riscontrato un numero ingente(1827) di record contenenti almeno un valore nullo. Siccome ogni record è caratterizzato da 304.304 time steps si è deciso di ridurre la dimensionalità cancellando quelle colonne temporali contenenti i valori nulli, ottenendo due nuovi dataframe di train e test di dimensioni (1828,140.941) e (634,140.941) rispettivamente.

Nel seguito vediamo un'anteprima della rappresentazione delle time series in uno spazio a due dimensioni.



Data l'elevata lunghezza temporale per ogni serie, nelle successive sezioni sono state effettuate approssimazioni opportune a seconda del task da conseguire.

6.2 Clustering

Per il task di Clustering sono stati utilizzati due approcci differenti:

- *Partizionale*: Algoritmo K-Means che sfrutta il dynamic time warping come metrica di distanza.
- *Gerarchico*: Algoritmo agglomerativo con diversi criteri di linkage.

Entrambi gli algoritmi sono stati testati su 3 approssimazioni differenti delle serie temporali per poi mettere a paragone le differenti istanze di clusters. In particolare le approssimazioni applicate sono:

- *DFT* -> 32 coefficienti
- *PAA* -> 10 segmenti
- *SAX* -> 10 segmenti e alphabet-size=8

La scelta dei valori di approssimazione è stata fatta in relazione al tempo di elaborazione e al risultato dei clusters restituiti dai due differenti approcci. Inoltre relativamente al K-means si è provato ad applicare l'algoritmo sui dati aventi come attributi le seguenti features estratte: media, deviazione standard, skew, curtosi e varianza.

6.2.1 K-Means

Per la ricerca dell'opportuno valore del parametro K è stato effettuato un trade-off tra SSE e Silhouette relative alle istanze ottenute facendo variare il parametro tra [2,19] e selezionando il valore a seconda dell'esperimento in esame.

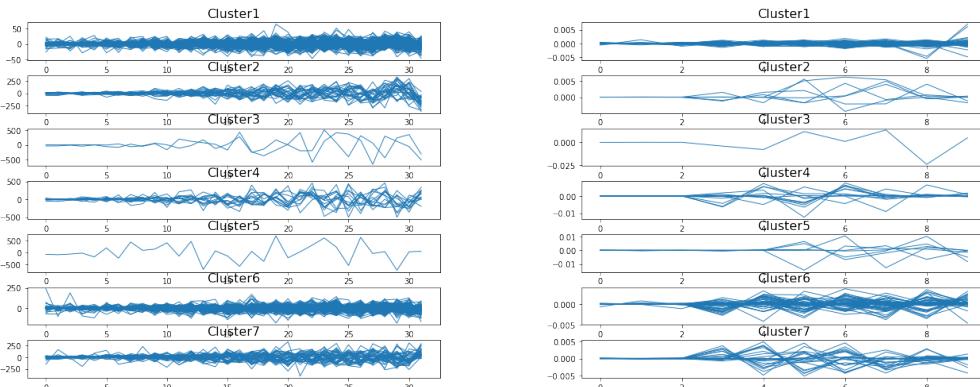


Figure 6.1: DFT vs PAA clusters

emotion	angry	calm	disgust	fearful	happy	neutral	sad	surprised	
row_0	0	212	276	131	221	244	140	260	129
1	10	0	0	8	6	0	1	2	
2	0	0	0	1	1	0	0	0	
3	3	0	1	2	5	0	2	1	
4	0	0	0	1	0	0	0	0	
5	42	4	8	35	20	0	12	11	
6	13	0	4	12	4	0	5	1	

emotion	angry	calm	disgust	fearful	happy	neutral	sad	surprised	
row_0	0	245	279	136	248	260	140	271	136
1	4	0	1	6	5	0	1	1	
2	0	0	0	1	0	0	0	0	
3	5	0	1	3	2	0	0	1	
4	3	0	1	4	3	0	1	0	
5	23	1	5	18	10	0	7	6	

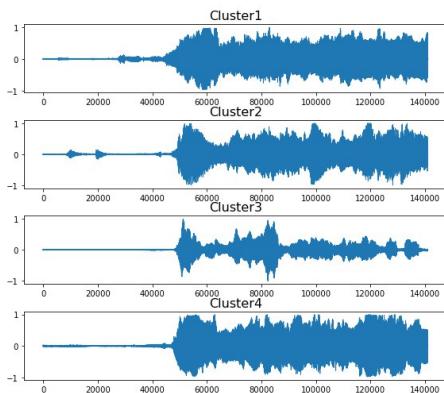
Figure 6.2: DFT vs PAA cross tab

Si può notare dalla figura che le approssimazioni tramite DFT e PAA hanno restituito clusters relativamente simili nella distribuzione dei dati come si può notare anche dalle relative cross-tab

che evidenziano le emozioni appartenenti a ciascun oggetto. In particolare tra le due istanze di clusters quello relativo alla PAA presenta uno score migliore di silhouette(0.80) che suggerisce una buona separazione tra i vari gruppi. Si evince inoltre che la maggior parte delle serie vengono raccolte all'interno del primo cluster e una scarsa capacità di distribuzione dei dati comune ad entrambi gli esperimenti.

Riguardo all'istanza di clusters con approssimazione SAX, non sono stati ottenuti buoni risultati con l'approccio partizionale, ottenendo un coefficiente di Silhouette pari a 0.09.

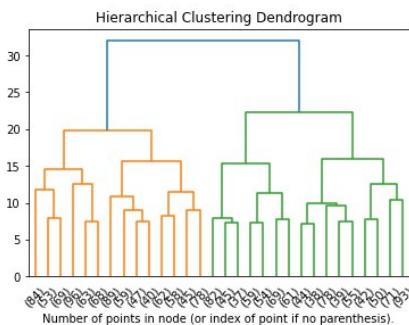
Nell' ultimo esperimento di feature-based clustering mediante K-means, i 4 clusters ottenuti sono caratterizzati da una silhouette pari a 0.60. Dalla figura sottostante si può evincere una distribuzione abbastanza regolare tra i vari raggruppamenti, in particolare il cluster 2 riesce ad isolare un discreto quantitativo di file audio solamente parlati.



vocal_channel	song	speech
row_0		
0	560	312
1	20	259
2	0	55
3	168	454

6.2.2 Gerarchico

Come anticipato precedentemente, per questo approccio, è stato utilizzato un algoritmo agglomerativo testando differenti tipologie di linkage e approssimazioni dei dati. In particolare l' esperimento che ha portato i risultati migliori è stato con SAX e complete linkage. Nella figura di sinistra viene mostrato il dendogramma generale senza utilizzo del parametro di cluster e sulla destra la tabella che descrive la distribuzione dei dati dopo aver reinstantziato l'algoritmo con un numero uguale a 4 di cluster in input. Nonostante la buona distribuzione nei vari gruppi, il coefficiente di silhouette pari a 0.11 suggerisce una scarsa separazione dei clusters in esame.



CLUSTER	TIME SERIES
0	510
1	478
2	433
3	407

In conclusione, si può affermare che il metodo che ha restituito i risultati migliori è quello partizionale con il K-Means feature-based, garantendo una discreta separazione dei dati e una sufficiente separazione tra i cluster.

6.3 Classification

Come ultimo task sulle serie temporali si è deciso di testare diversi modelli di classificazione, più o meno complessi, per determinare l'etichetta di classe dei dati in esame, confrontandole poi con le vere etichette determinando dunque le performance dei vari classificatori. Per questo Task si è deciso di prendere in esame come variabile target la feature "vocal_channel" che può assumere valori song(etichetta 0) e speech(etichetta 1); in quanto si è ritenuto interessante stabilire con quale accuratezza i vari modelli definiti riescano a classificare le serie temporali come file audio di tipo parlato oppure cantato. Per problemi computazionali è stato necessario approssimare i record in esame: sono state provate diverse tecniche di approssimazione e dopo un trade-off tra tempo computazionale e performance dei modelli ottenuti si è deciso di concentrarci sui dati approssimati tramite "Discrete Fourier Transform" utilizzando 1000 coefficienti.

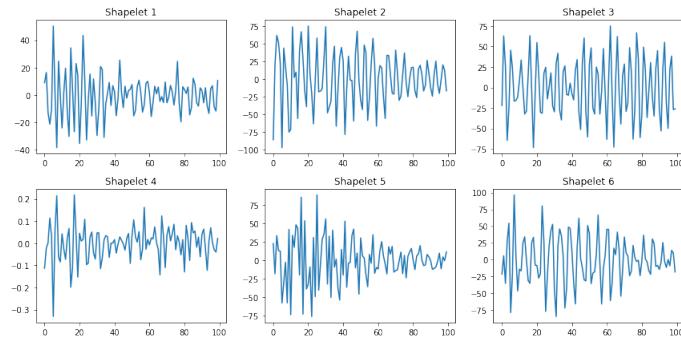
Nel seguito vengono elencati i diversi modelli definiti e valutati tramite una apposita tabella che mette a confronto l'accuracy e l'F1_macro score dei vari esperimenti.

- Shapelet Classifier
- Shapelet distance-based Classifier with KNN
- Shapelet distance-based Classifier with DT
- KNeighborsClassifier with Euclidean Distance
- KNeighborsClassifier with Manhattan Distance
- Convolutional NN
- CanonicalIntervalForest

In particolare il secondo e terzo classificatore sono stati eseguiti dopo aver calcolato la distanza tra le 6 shapelets (individuate dal primo modello) e ogni record di train ed utilizzando proprio queste distanze come dati di allenamento. Si precisa che la lunghezza e il numero di shapelets sono state definite in base al risultato di una funzione apposita che analizza le serie temporali in esame e restituisce le relative informazioni suggerite. Non sono state effettuate ricerche esaustive degli iperparametri tramite grid search come nei capitoli precedenti, a causa dell'elevata computazione in tempo nella validazione dei modelli. Si è provato ad utilizzare il KNN utilizzando il dynamic time warping per trovare il miglior allineamento tra le coppie di serie temporali ma purtroppo la computazione anche in questo caso richiedeva tempi ingenti. Concentrandoci sulla rete neurale convoluzionale, è stata strutturata su 4 blocchi sequenziali identici(separati ciascuno da un layer di Dropout per controllare l'overfitting) costituiti da 3 layer:Conv1D, BatchNormalization, Activation Relu; l'architettura culmina poi con un layer di pooling GlobalAverage1d e un nodo Dense che utilizza la funzione di attivazione sigmoidea per restituire la previsione di classe.

MODELLO	Accuracy	F1_macro
Shapelet Classifier	53%	53%
Shap dist-based with KNN	65%	64%
Shap dist-based with DT	73%	72%
KNN with Euclidean Distance	74%	74%
KNN with Manhattan Distance	78%	78%
Convolutional NN	82%	82%
Canonical Interval Forest	90%	89%

Come si evince dalla tabella i risultati migliori, sui dati di test, sono stati ottenuti con modelli più complessi come la rete convoluzionale che si distacca nettamente dalle performance degli esperimenti precedenti. Il modello che ha performato meglio, per questo caso di studio, è stato il Canonical Interval Forest che riesce a raggiungere il 90% di accuratezza e 89% di F1_macro. Infine si è deciso di focalizzare l'attenzione sullo Shapelet Classifier e di analizzare le sei shapelets utilizzate per la classificazione.



Nel grafico sono visibili le sei shapelets estratte dal modello ed utilizzate per discriminare le serie temporali nelle due classi Song e Speech. In particolare si può notare come le numero 4 e 6 differiscano dalle altre in termini sia di fase che di amplitudine in diversi intervalli; analizzandole più nello specifico si è scoperto che per i modelli di DT e KNN basati sulla distanza dalle shapelets, i valori di feature (che corrispondono proprio alle shapelets) importance più alti vengono ottenuti proprio dalla numero 4 e 6. Nel seguito i corrispondenti valori di importanza ottenuti per l'albero decisionale

—> (0.09695543, 0.11844816, 0.11227855, 0.3105481 , 0.14088996, 0.22087981)

6.4 Motifs & Discords Discovery

Per questo task sono state prese in esame due diverse serie temporali approssimate opportunamente mediante DFT (Discrete Fourier Transformation) con 1000 coefficienti:

- 1) audio con emozione "calm" ed intensità emozionale "normale"
- 2) audio con emozione "angry" ed intensità emozionale "strong"

Sono stati scelti questi due record per mettere a confronto le caratteristiche riscontrate tra due file audio di emozione e intensità emozionale diverse. Per analizzare i due differenti audio è stata definita la matrix profile provando diversi valori della finestra temporale e selezionando poi un window_size=50 per ogni serie.

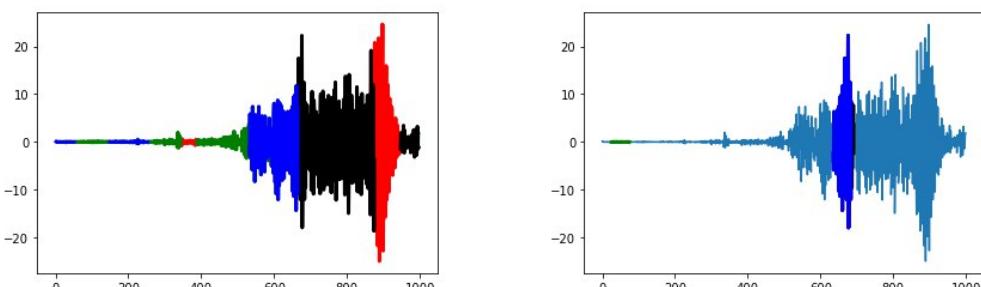


Figure 6.3: Motifs vs Discords primo file audio

Dalla rappresentazione dei motifs relativi al primo audio è possibile identificare 4 pattern, evidenziati da altrettanti colori, che si ripetono prevalentemente in sequenza come si vede anche dal corrispondente array di indici:

- *red*: [337,863,894,920]
- *green*: [28, 56, 89, 117, 177, 231, 265, 295, 388, 421, 449, 477, 505, 559]
- *black*: [649, 679, 713, 747, 791, 826, 947]
- *blue*: [2, 148, 204, 532, 586, 616]

Nella ricerca delle anomalie, ne sono state riscontrate invece essenzialmente 2, come si può vedere dal grafico a destra, che si verificano proprio all'inizio del file audio e quando l'intensità del segnale inizia ad aumentare fino ad un valore pari a 20.

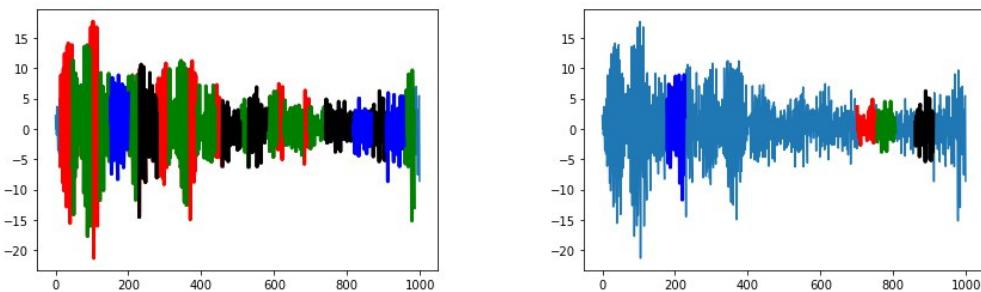


Figure 6.4: Motifs vs Discords secondo file audio

Per quanto riguarda i motifs del secondo caso in esame è possibile notare la presenza sempre di 4 pattern però distanziati e ripetuti più frequentemente rispetto al file audio precedente, probabilmente perché l'audio è vocalizzato da un'intensità emozionale strong, in cui sono prodotte onde sonore in maniera più marcata e regolare. Nel seguito i corrispondenti indici in cui si ripetono i pattern identificati:

- *red*: [13, 78, 268, 343, 421, 588, 656]
- *green*: [46, 122, 177, 311, 389, 494, 559, 696, 935]
- *black*: [230, 458, 530, 741, 784, 856]
- *blue*: [151, 818, 900]

I Discords riscontrati invece in questo caso sono 4 e come si può vedere dal grafico a destra quello più isolato corrisponde alla sequenza di colore blu; mentre i restanti tre risultano molto ravvicinati e in prossimità della fine dell'audio.

Chapter 7

XAI Advanced Classifiers

Per questo task si è deciso di prendere in esame il classificatore Multi-layer Perceptron con 2 hidden layers, dato che questo modello è quello che restituisce le performance migliori rispetto a tutti quelli testati nel terzo capitolo(55% accuracy e 53% F1_macro) per la classificazione emozionale dei dati. Un aspetto particolare riscontrato sia in questo classificatore che in tutti gli altri modelli è che una considerevole quantità di record di test aventi come etichetta l'emozione "Sad", vengono misclassificati con l'emozione "Calm"(circa il 60%). In questo capitolo cerchiamo dunque, attraverso metodologie differenti, di aprire la "Black-Box" della rete neurale in esame per cercare di interpretare e comprendere i motivi che spingono il modello a prendere determinate decisioni; soprattutto quella appena introdotta.

Si è deciso, per i nostri esperimenti, di adottare due approcci differenti:

- **Global Approach:** *Basic Decision Tree, Shap*
- **Local Approach:** *Lime,Lore*

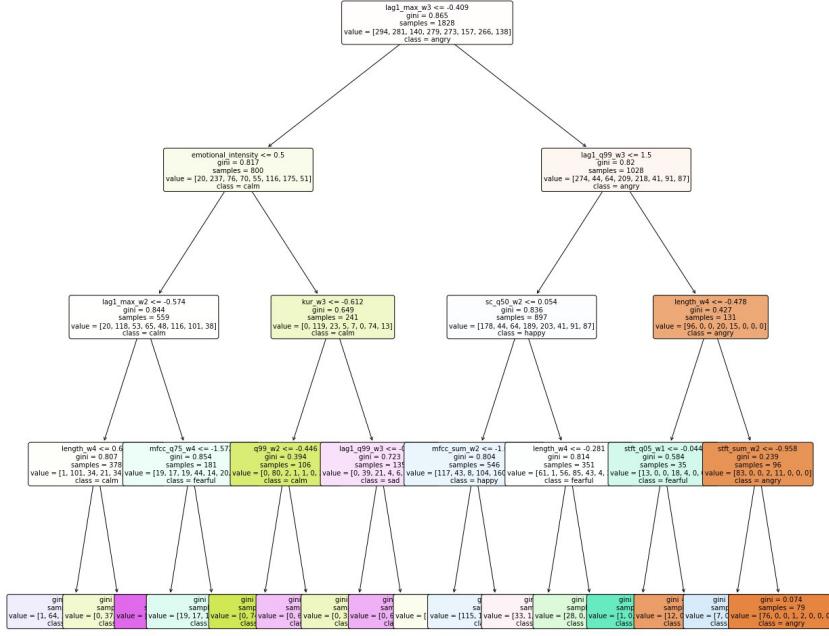
In particolare, per quanto riguarda l'approccio locale, sono state analizzate quali sono le features e i rispettivi valori che consentono alla black box di classificare correttamente un record etichettato con l'emozione "Sad" e di misclassificare un'ingente quantità di dati con etichetta "Sad" come "Calm".

7.1 Global Approach

7.1.1 Basic Decision Tree

Con questo primo metodo si è cercato di capire la logica e il processo di classificazione attraverso un semplice albero di decisione di profondità massima pari a 4. Per fare ciò sono stati seguiti due passaggi:

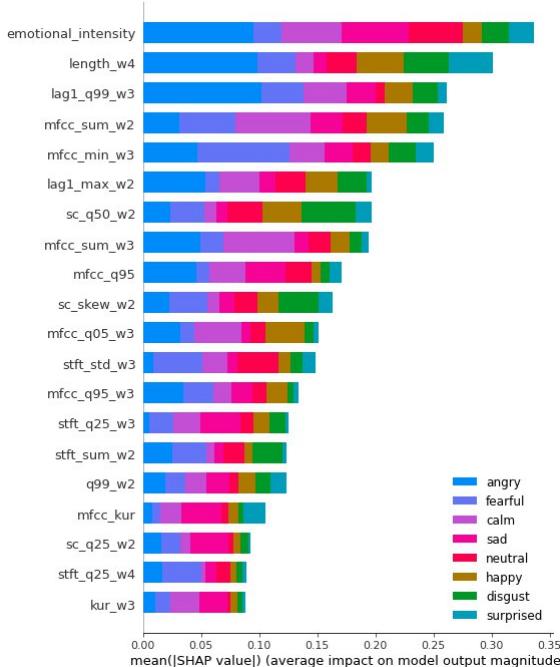
- **1)** Allenare il modello di rete neurale con i dati di train e testare la sua capacità predittiva sugli stessi dati.
- **2)** Istanziare il Decion Tree addestrandolo sui consueti dati di train aventi come etichette quelle predette al passo precedente.



Possiamo vedere dalla figura come l’ albero tenta di approssimare la logica del MLP; questa metodologia non è stata di grande aiuto in quanto andrebbe svolta un’analisi con un albero con profondità maggiore. Si è deciso infatti di provare un’altra metodologia sempre con approccio globale.

7.1.2 Shap

Per un analisi più attenta è stata effettuata una dimensionality reduction tramite RFE utilizzando 22 componenti e, dopo aver istanziato l’explainer, sono stati analizzati gli shap-values per cercare di capire il contributo di ciascuna feature nella classificazione delle emozioni (per ridurre il tempo di computazione sono state prese in considerazione solo 1000 coalizioni).



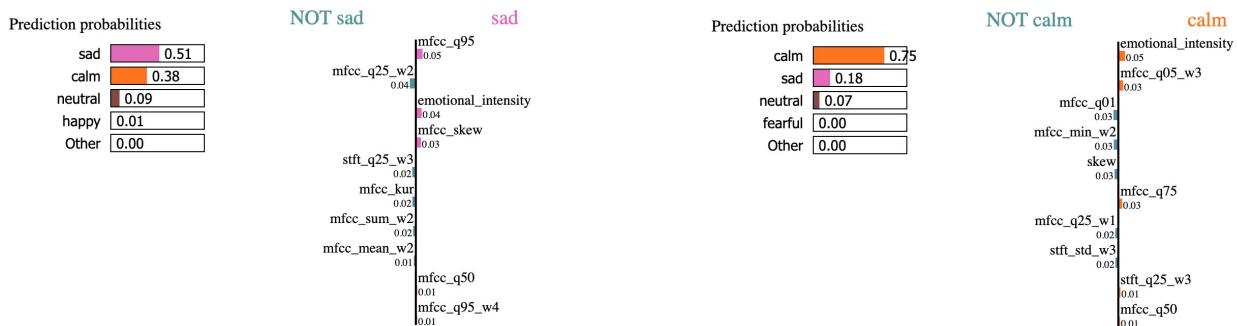
Come si nota nel grafico le features più rilevanti sono state: "emotional_intensity" e "length_w4" (il cui potere discriminativo, anche se meno chiaro, si poteva già evincere nell'albero rappresentato precedentemente). Ogni barra del summary plot è rappresentata da più colori e ciascuno di essi rappresenta un'emozione. Più ampia è una barra, più la corrispondente feature riesce a discriminare meglio la relativa emozione.

Fatta questa premessa, si può intuire, così come anche esplicitato nel capitolo 3, che i record aventi emozione "Angry" sono quelli con più possibilità di essere classificati correttamente dato che più features riescono a discriminari meglio. Concentrandoci invece sulle emozioni Sad e Calm possiamo notare come le barre corrispondenti alle due emozioni, per ciascuna feature siano tendenzialmente di solita ampiezza; ciò suggerisce di fatti un'alta probabilità di misclassificazione per i record aventi queste due etichette. Nella seguente sezione sono stati adottati approcci di tipo locale per andare ancora più nel dettaglio nell'analisi di questa situazione.

7.2 Local Approach

7.2.1 Lime

In questa metodologia, così come per il Lore, sono stati presi in esame due record aventi emozione Sad; il primo classificato correttamente dalla "black-box" della rete neurale e il secondo etichettato erroneamente come "Calm" (errore ricorrente nel nostro caso di studio).



Come si può notare dal grafico a sinistra, per il primo caso (correttamente classificato) la probabilità è poco sopra al 50% e alcuni attributi relativi all' mfcc di differenti quartili e finestre temporali contribuiscono negativamente. Infatti osservando il grafico di destra (misclassificato come Calm) si vede che oltre all'ingente percentuale restituita e alla classica feature emotional intensity già riscontrata, intervengono positivamente nella predizione errata sempre feature relative all'mfcc.

7.2.2 Lore

Con quest'ultimo metodo sono stati estratti una regola e diversi counterfactual per i due record sotto esame. Come si può notare dalla seconda colonna della tabella, per il primo esperimento si evince che determinati valori delle feature sc ≤ 0.23 , stft > -0.95 consentirebbero alla black-box di restituire una predizione errata; notare che nella corrispondente regola le feature assumono valori opposti. Per il secondo esperimento si nota nuovamente la presenza di 5 attributi relativi all'mfcc che condizionano la regola relativa al record classificato erroneamente.

	Rule	Counterfactual
Record classificato correttamente	<pre>{ lag1_q99_w3 <= 0.31, sc_q75 > 0.23, mfcc_q95 <= -0.87, kur_w1 > -0.32, stft_q25_w3 <= -0.95, lag1_max_w2 <= 0.02, lag1_max_w3 <= -0.09, lag1_q99_w2 > -0.60, zc_kur_w4 > -0.18 } --> { emotion: sad }</pre>	<pre>{ sc_q75 <= 0.23, emotional_intensity > 0.78 } { stft_q25_w3 > -0.95, zc_q95_w1 <= -0.63 }</pre>
Record classificato erroneamente	<pre>rule = { mfcc_q75 > 1.31, mfcc_q01 <= -0.17, stft_std_w1 <= 0.74, sc_std_w3 <= 1.59, mfcc_skew_w2 > -1.04, zc_std_w1 <= 0.42, mfcc_q99_w1 > -1.03, lag1_skew_w2 > -1.14 } --> { emotion: calm }</pre>	<pre>{ sc_std_w3 > 1.59, stft_q05_w3 <= -0.28, sc_q05 <= -0.36 }</pre>

In conclusione, in questa sezione si è cercato di motivare i risultati restituiti dal miglior classificatore(per accuracy e f1_macro) ma in particolare perchè viene riscontrato un alto tasso di errore di classificazione per i record con emozione Sad ma classificati con l' etichetta Calm(problematica riscontrata in tutti i modelli definiti nei capitoli precedenti). In particolare, dopo gli esperimenti di XAI visti in questa sezione, possiamo affermare che il problema maggiore è relativo alle diverse feature dell' mfcc e all' attributo emotional intensity.

This project was written in L^AT_EX