
# Advanced Analysis and Model Development on Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)*

Artesi S.† Fidone G.‡ Lleshi B.§

June 2023

**Abstract**

This report presents a comprehensive analysis of the RAVDESS dataset, covering a wide range of tasks including unsupervised anomaly detection, imbalanced binary classification, binary and multi-label classification, multiple regression, time series analysis (motif and discord detection, clustering, classification) and explainable AI.

*Project for Data Mining Course (Advanced Applications), A.Y. 2022/23, University of Pisa.

†Master in Data Science & Business Informatics – s.artesi@studenti.unipi.it

‡Master in Digital Humanities (Language Technologies) – g.fidone1@studenti.unipi.it

§Master in Data Science & Business Informatics - b.lleshi2@studenti.unipi.it

# Contents

# 1 Data Semantics

«Ryerson Audio-Visual Database of Emotional Speech and Song» (RAVDESS) is a validated multimodal dataset developed by Livingston & Russo (2018) which consists of audio-visual recordings of 24 professional actors vocalizing two lexically-matched statements in a neutral North American accent. The data employed in the first part of this study is a modified version of the original RAVDESS where alongside the original categorical attributes (Table 1) numerical attributes are created by extracting quantitative statistics from the raw audio signals (Table 2).

**Table 1:** Categorical Attributes

| **Name** | **Type** | **Description** |
|---|---|---|
| *modality* | Nominal | Recording mode |
| *vocal_ channel* | Nominal | Type of vocal communication |
| *emotion* | Nominal | Emotion expressed |
| *emotional_ intensity* | Ordinal | Degree of emotional involvement |
| *statement* | Nominal | Statement uttered |
| *repetition* | Ordinal | Repetition of the statement |
| *actor* | Nominal | Actor's ID |
| *sex* | Nominal | Actor's sex |
| *filename* | Nominal | Record's ID |

**Table 2:** Global-level Numerical Attributes

| Name(s) | Type | Description |
|---|---|---|
| *frame_ count* | Interval | Number of frames per sample |
| *mean*, *std*, *min*, *max*, *skew*, *kur*, *q_ 01*, *q_ 05*, *q_ 25*, *q_ 50*, *q_ 75*, *q_ 95*, *q_ 99* | Ratio | Statistics of original audio signal |
| *lag1_ sum*, *lag1_ mean*, *lag1_ std*, *lag1_ min*, *lag1_ max*, *lag1_ kur*, *lag1_ skew*, *lag1_ q01*, *lag1_ q05*, *lag1_ q25*, *lag1_ q50*, *lag1_ q75*, *lag1_ q95*, *lag1_ q99* | Ratio | Statistics of Lag (difference between each observation and the antecedent) |
| *zc_ sum*, *zc_ mean*, *zc_ std*, *zc_ min*, *zc_ max*,, *zc_ kur*, *zc_ skew*, *zc_ q01*, *zc_ q05*, *zc_ q25*, *zc_ q50*, *zc_ q75*, *zc_ q95*, *zc_ q99* | Ratio | Statistics of Zero Crossing Rate |
| *mfcc_ sum*, *mfcc_ mean*, *mfcc_ std*, *mfcc_ min*, *mfcc_ max*, *mfcc_ q01*, *mfcc_ q05*, *mfcc_ q25*, *lag1_ q50*, *mfcc_ q75*, *mfcc_ q95*, *mfcc_ q99*, *mfcc_ kur* | Ratio | Statistics of Mel-Frequency Cepstral Coefficients |
| *sc_ sum*, *sc_ mean*, *sc_ std*, *sc_ min*, *sc_ max*, *sc_ kur*, *sc_ skew*, *c_ q01*, *sc_ q05*, *sc_ q25*, *sc_ q50*, *sc_ q75*, *sc_ q95*, *sc_ q99* | Ratio | Statistics of Spectral Centroid |

continue on the next page

| Name(s) | Type | Description |
|---|---|---|
| *stft_ sum*, *stft_ mean*, *stft_ std*, *stft_ min*, *stft_ max*, *stft_ kur*, *stft_ skew*, *stft_ q01*, *stft_ q05*, *stft_ q25*, *stft_ q50*, *stft_ q75*, *stft_ q95*, *stft_ q99* | Ratio | Statistics of Short-Time Fourier Transform |

Further attributes have been created by dividing each time series into 4 non overlapping windows and computing all the quantitative statistics described in Table 2 at a local level. The names referring to such features can be easily derived from the expression:

$$\text{NAME\_}w\text{N}$$

and by replacing NAME with the name of the numerical attribute and N with the index (1, 2, 3 or 4) of the window considered. The only exception is represented by the global-level feature *frame_ count*, which is replaced locally with the denomination *length_ w*N.

## 2 Data Understanding and Preparation

The data is already provided with a partitioning into training (TR) and test (TS), with a fraction of 34.1% of all data reserved as TS.

Given the relatively high dimensionality of data, a global understanding of numerical attributes can be provided by looking at the distributions of their skewness and kurtosis (Figure 1).

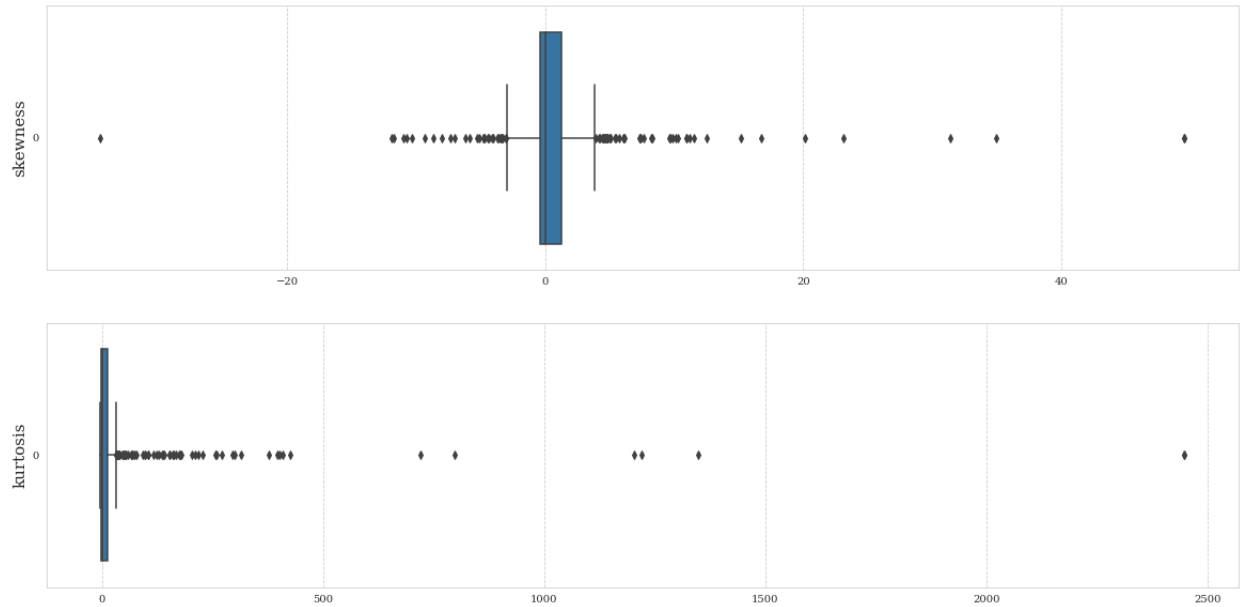



**Figure 1:** Distributions of skewness and kurtosis of numerical attributes. Data is restricted to TR as it is assumed that TS data are drawn from the same probability distributions.

As evidenced, only a small amount of distributions (4.5%) are nearly mesokurtic (-0.5 $<$ *kur* $<$ 0.5) and a slightly larger amount (8.5%) can be considered sufficiently symmetrical (-0.5 $<$ *skew* $<$ 0.5).

Data balancing for categorical attributes has been assessed by supporting visual inspection with the computation of relative frequencies. A modest imbalance has been observed in *vocal_ channel* (+18.2% for *speech* on TR) and *emotional_ intensity* (+8% for *normal* on TR). A more significant imbalance concerns *emotion*, where *neutral*, *disgust* and *surprised* cover about half of the data (8% on TR) than the other values.

This preliminary understanding has not revealed the presence of inconsistencies (syntactical or semantic), missing values or duplicates.[1] Data preparation has therefore concerned only the elimination of continuous

[1] Outlier detection will be addressed in the following section.

attributes with null variance as well as categorical attributes with a unique value, which has allowed to decrease the dimensionality of data from 434 to 383. Further pre-processing operations (standardization, one-hot encoding, feature reduction) will be considered from time to time where required.

# 3 Anomaly Detection

For the anomaly detection task we have compared the results from a broad selection of methods belonging to different classes of unsupervised algorithms: Histogram-Based Outlier Score (HBOS), belonging to visual-based methods; Deviation-Based (DB); Elliptical Envelope (EE), belonging to depth-based methods; K-Nearest Neighbors (KNN), belonging to distance-based methods; Local Outlier Factor (LOF), belonging to density-based methods; DBSCAN, belonging to clustering-based methods; Angle-Based Outlier Degree (ABOD); Lightweight Online Detector of Anomalies (LODA), belonging to ensemble-based methods; Isolation Forest (IF), belonging to model-based methods.

Pre-processing operations have concerned, in the order:

- Standardization (min-max) of numerical attributes;
- One-hot encoding of categorical attributes;
- Limitedly to the application of EE, KNN and DBSCAN (which are more sensitive to the curse of dimensionality), feature reduction with PCA (3 latent dimensions).

For each method the contamination rate has been set to 1%, which corresponds to the top 19 data points related to the largest «outlierness». This also applies to a labeling method as DBSCAN, where the tuning of the hyper-parameters (*Eps* and *MinPts*) has relied both on the «elbow» approach[2] and on the resulting number of noisy data. It must also be noticed that the top 1% scores of the DB algorithm have been considered independently from the attribute with respect to they have been computed.

Evaluation of unsupervised outlier detection is notoriously difficult. As each method assumes a different definition of «outlierness», a common practice consists in considering a combined response. Figure 2 shows the percentage of outliers shared by detection methods – since the number of possible intersections is exponential ($2^9$), visualization is restricted to those having a percentage of shared outliers not less than 20%.

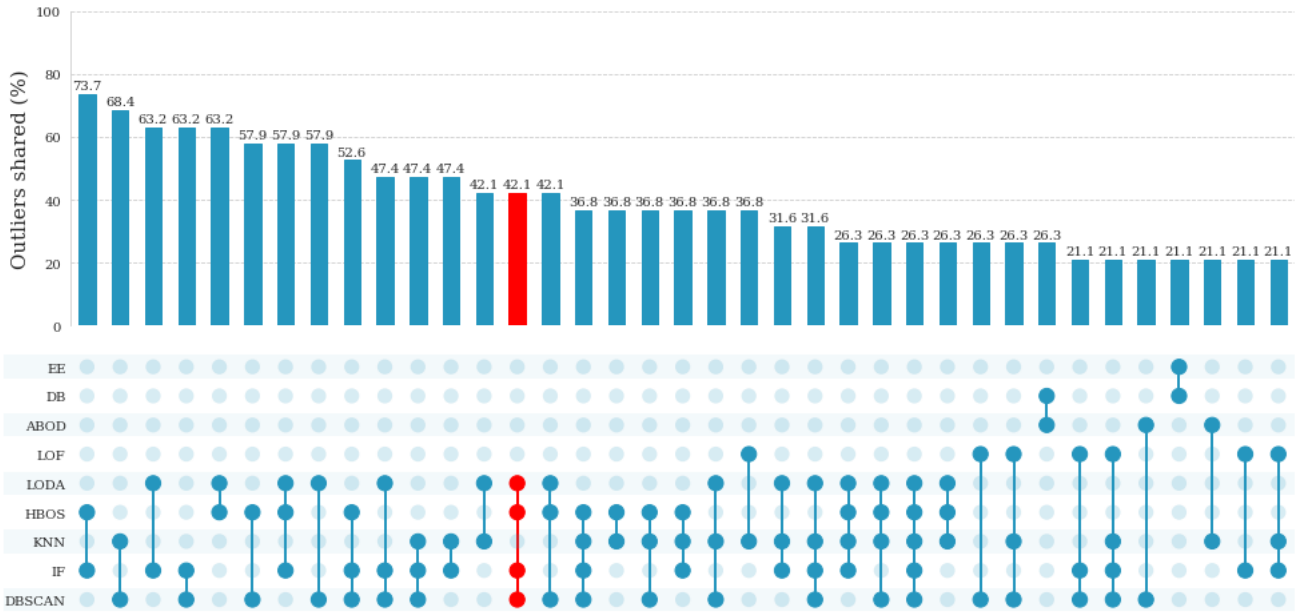



**Figure 2:** Outliers shared by detection methods (TR). Intersections with percentage of shared outliers below 20% are not displayed. Percentages (above) are sorted by decreasing order, whereas detection methods (below) are sorted w.r.t. increasing number of intersections. The intersection highlighted in red is the set of selected outliers.

[2]By analyzing the plot displaying all the data points (x-axis) sorted by their $k$-distances w.r.t. the value of the $k$-distance (y-axis), a sharp change in the slope of the curve is expected to be observed at the x-value which likely separates noisy points from other points, so that the correspondent $k$-distance is a suitable choice for *Eps* and $k$ itself is a suitable choice for *MinPts*.

As the number of methods involved increases, the degree of «outlierness» of a point is assumed to increase, but it is also more likely that the size of the intersection decreases. A suitable choice must be therefore a trade-off between the number of expected outliers and the degree of «outlierness». Our approach consists in setting a minimum threshold of 30% of shared outliers and considering the intersection with maximum number of methods which also maximizes the number of outliers. This corresponds to the outliers shared by LODA, HBOS, IF and DBSCAN (Figure 2 in red).

Once outliers have been identified, a straightforward way to manage them consists in replacing the values of each continuous attribute with a measure of central tendency. Given the asymmetry of most numerical attributes (Figure 1), the median should be a more robust indicator of the center of each distribution. After replacement, a final assessment consists of a further joint application of LODA, HBOS, IF and DBSCAN for ensuring that the new top 1% outliers do not overlap with the previous outliers (Figure 3).

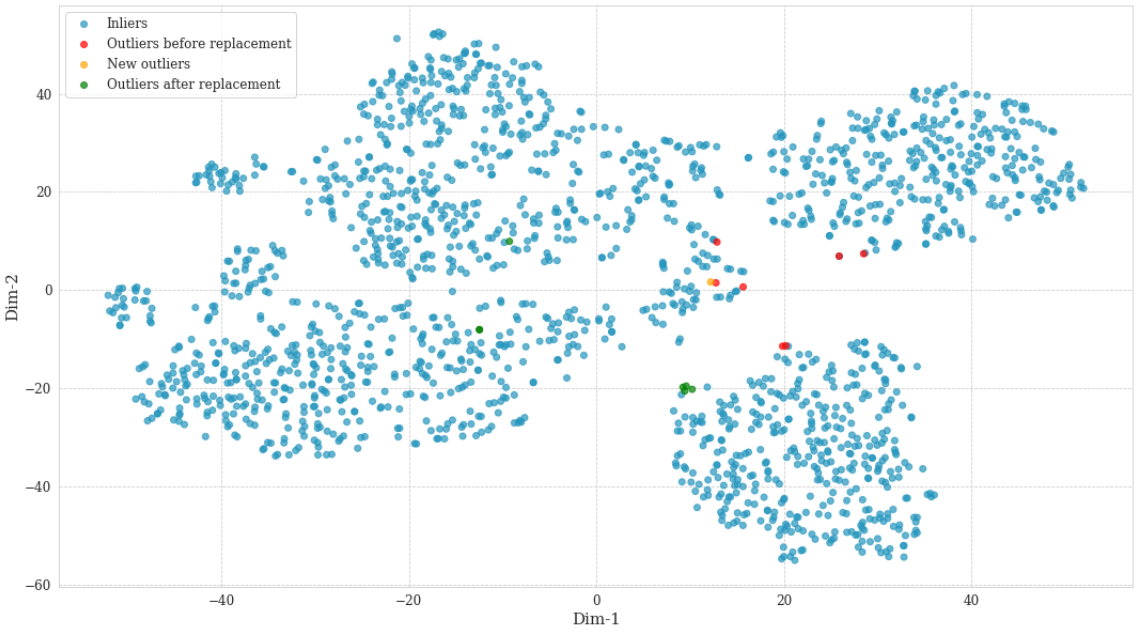



**Figure 3:** Inliers, outliers (before and after replacement) and new detected outliers. Dimensionality reduction has been performed with t-SNE.

# 4 Imbalanced Learning

This section aims to test the performance of three different techniques for handling a user-created imbalanced setting: random undersampling, Syntetic Minority Oversampling Technique (SMOTE) and class weight adjustment. We measure the performance of each technique by looking at the improvements in the F1-scores of two simple classifiers – Decision Tree (DT) and K-Nearest Neighbors (K-NN)[3] – before and after the re-balancing of data. The analysis of the performance of each classifier has been restricted to the binary classification of *vocal_channel*, *sex* and *emotional_intensity*.

For the accomplishment of this task we do not use the already provided TS data. For each target, pre-processing operations on TR have concerned, in the order:

- Standardization (min-max) of numerical attributes;
- One-hot encoding of categorical attributes (target excluded);
- Label encoding of the target attribute;
- User-created imbalance, obtained by removing random values from the majority class of the binary target in order to obtain a 96%-4% proportion;[4]
- TS data generation by separating 30% of the employed data.

[3]The latter is limited only to re-balanced data by random undersampling and SMOTE.

[4]We remove values from the majority class to maximize the number of data before applying the re-balancing techniques. The distributions of the target attributes before the imbalancing are reported in Figure 4.

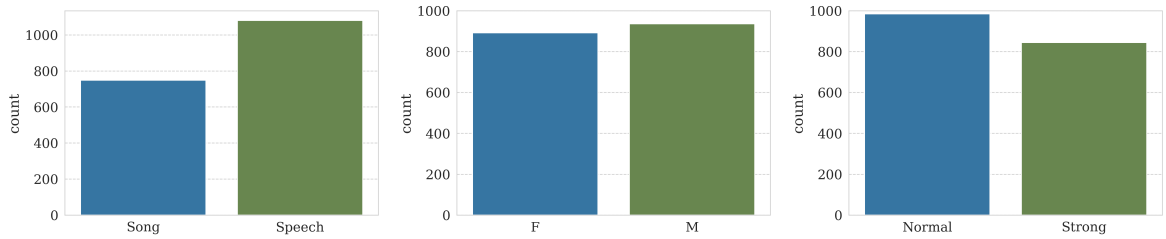



**Figure 4:** From left to right: distributions of *vocal_ channel*, *sex* and *emotional_ intensity* before imbalancing.

Differently from DT, K-NN can be sensitive to feature selection. For this reason we have also reduced the training features for the application of K-NN by employing a filter strategy. Since most numerical attributes display a non-Gaussian distribution (Figure 1), correlations between the target variable and each continuous attribute can be computed with the non-parametric Spearman coefficient ($\rho$), such that for *vocal_ channel* and *sex* we retain training attributes with correlation $\rho > 0.7$ or $\rho < -0.7$; while for *emotional_ intensity* we retain training attributes with correlation $\rho > 0.4$ or $\rho < -0.4$.

Model selection has relied on a randomized search with repeated stratified 5-fold CV. Tested hyper-parameters and correspondent values for DT and K-NN can be respectively found in Tables 3 and 4.

**Table 3:** Tested hyper-parameters for Decision Trees

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| Criterion | Measure used to select the best split | Gini, Entropy, Log- Loss |
| Max Depth | Maximum depth of the tree | Discrete interval [2, 200] |
| Min Samples Split | Minimum number of samples to split an internal node | Log-uniform distribution in the interval [0.01, 1] |
| Min Samples Leaf | Minimum number of samples for a leaf node | Uniform distribution in the interval [0.001, 0.2] |

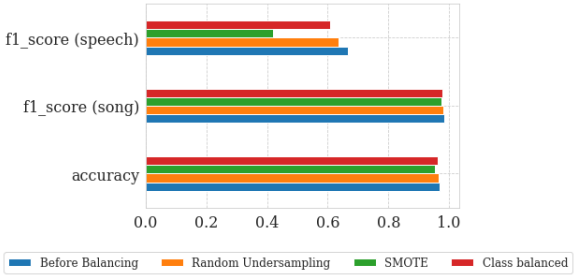
**Table 4:** Tested hyper-parameters for K-NN

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| K | Number of Neighbors | Discrete interval [2, N/2] |
| Weights | Weight function used in prediction | Uniform, Distance |
| Metric | Distance measure | City-Block, Euclidean, Cosine, Chebyshev |

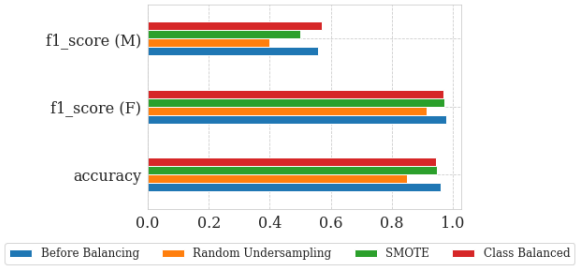
Model evaluation has been performed with a simple hold-out on the previously separated TS data.

Since random undersampling involves discarding potentially valuable data from the majority class, the performance of each classifier may be affected by the randomicity of the sampling process. For this reason we have performed random undersampling for 10 different iterations and considered as performance indicator of DT and K-NN the mean of the respective F1-scores. This further precaution is dispensable when applying SMOTE or class weight adjustment.

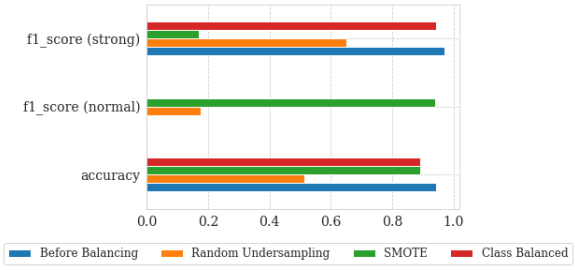
Figures 5 and 6 report the results obtained by Decision Trees and KNN on imbalanced data and after the application of each balancing method. For Decision Trees, balancing methods seem to have a negligible effect on the classification of *vocal_channel* and *sex*, but they play a significant role in the classification of *emotional_intensity*, where they allow for an increase in the model capability in detecting the minority class *normal* (whose F1-score is null in the unbalanced setting), even if at the cost of a decrease in the F1-score of the majority class *strong*. As for K-NN, the F1-score of minority classes (*speech*, *M*, *normal*) in each imbalanced learning is null. In this case, both SMOTE and random undersampling seem to have a major role in improving the capability of the model in recognizing the minority class. Overall SMOTE seems to perform better than random undersampling, probably because of the relatively small size of the data; and also better than class weight adjustment, which fails in improving the Decision Tree capability in recognizing the minority class instances of *emotional_ intensity*.




(a) *vocal_ channel*

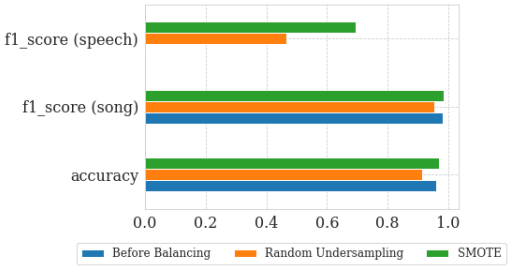



(b) *sex*

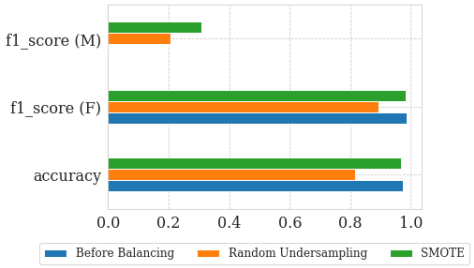



(c) *emotional_ intensity*

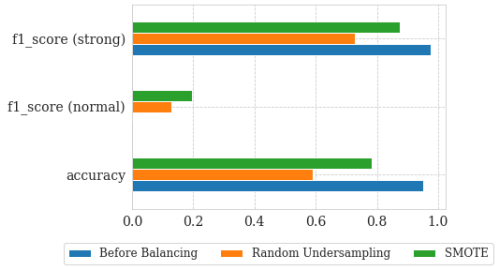
**Figure 5:** Results of Decision Tree for each target before balancing and after the application of balancing methods.




(a) *vocal_ channel*




(b) *sex*




(c) *emotional_ intensity*

**Figure 6:** Results of K-NN for each target before balancing and after the application of balancing methods.

# 5 Classification

In this section we address three binary classification tasks (*vocal_ channel*, *sex* and *emotional_ intensity*) and a multi-class classification task (*emotion*). The models that have been considered for solving the classification tasks are: Logistic Regression (LG), Support Vector Machines (SVM), Neural Networks (NN), Decision Tree Bagging (DTB), Random Forests (RF), AdaBoost (AB) and Gradient Boosting (GB). LG has also been regarded as a baseline model for evaluation.

For each model and task, pre-processing operation have concerned, in the order: the standardization of numerical attributes with min-max scaler; the one-hot encoding of categorical attributes (target excluded); and the label encoding of the target attribute. Final model evaluation is performed with a hold-out on the already

provided TS data.

## 5.1 Logistic Regression

Model selection for LG has consisted of a grid search with 5-fold CV. Tested hyper-parameters and correspondent values can be found in Table 5. Since some penalties are not compatible with some solvers, we set L2 for all the candidates. Furthermore, we set to 800 the maximum number of iterations required for the solver to converge.

**Table 5:** Tested hyper-parameters for Logistic Regression

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| C | Inverse of regularization strength | Log-uniform distribution in the interval $[10^{-4}, 10^{3}]$ |
| Solver | Algorithm used in optimization problem | L-BFGS, LIBLINEAR, Newton-CG, Newton-Cholesky, SAG, SAGA |

Table 6 reports the best result.

**Table 6:** Best results of Logistic Regression

| Target | C | Solver | Wt. Avg. F1 | Accuracy |
|---|---|---|---|---|
| *vocal_channel* | 1 | L-BFGS | 0.98 | 0.98 |
| *sex* | 1 | L-BFGS | 0.85 | 0.85 |
| *emotional_intensity* | 1 | L-BFGS | 0.77 | 0.77 |
| *emotion* | 1 | Newton-Cholesky | 0.43 | 0.49 |

## 5.2 Support Vector Machines

The assessment of SVM has first concerned linear SVM classifiers. Model selection has been performed with a randomized search with 5-fold CV searching for optimal values of the $C$ hyper-parameter (Table 7).

**Table 7:** Tested hyper-parameters for linear SVM

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| C | Penalty parameter of the error term | Log-uniform distribution in the interval $[10^{-4}, 10^{4}]$ |

Since the best linear SVM classifiers did not reach competitive results w.r.t. the baseline (LG), we have extended the same model selection procedure to non-linear SVM classifiers (Table 8), followed by a further (exhaustive) search over a more fine-grained grid – i.e. a grid where the tested ranges of values for a given hyper-parameter $\theta$ is the neighborhood of the best value of $\theta$ according to the first randomized search.

**Table 8:** Tested hyper-parameters for SVM

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| C | Penalty parameter of the error term | Log-uniform distribution in the interval $[10^{-4}, 10^{4}]$ |
| $\gamma$ | Kernel coefficient | Log-uniform distribution in the interval $[10^{-4}, 10^{4}]$ |
| Kernel | Kernel function | Linear, Polynomial, RBF |

For the targets *vocal_channel* and *emotional_intensity*, which display a slight imbalance[5], a second model selection has been performed with class weight adjustment. The best resulting models, however, have proved to reach lower accuracy scores than the models trained on the original imbalanced data.

The best results are reported in Table 9.

[5] See Section 2.

**Table 9:** Best results of SVM

| Target | C | $\gamma$ | Kernel | Wt. Avg. F1 | Accuracy |
|---|---|---|---|---|---|
| *vocal_ channel* | 95.454 | 0.0006 | RBF | 0.98 | 0.98 |
| *sex* | 0.7 | 0.1 | RBF | 0.90 | 0.90 |
| *emotional_ intensity* | 1072.26 | 0.0003 | RBF | 0.76 | 0.76 |
| *emotion* | 0.018 | 0.097 | Polynomial | 0.47 | 0.50 |

## 5.3 Neural Networks

Since most learning problems can be solved by employing shallow NNs, we first consider a simple FFNN having only one hidden layer with logistic as activation function. The model weights are initialized with the Glorot normal distribution and optimization is performed with mini-batch (32) gradient descent regularized with L2. For the binary classification tasks (*vocal_ channel, sex, emotional_ intensity*) we use logistic as output activation function and binary crossentropy as loss function. For the multi-label classification task (*emotion*) we use softmax as output activation function and sparse categorical crossentropy as loss function.

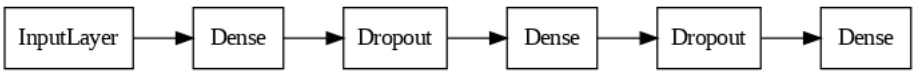
Model selection has been performed with a randomized search (3-fold CV) over a relatively large hyper-parameter space (Table 10). A fraction of 20% of all training data have been previously separated as validation data (VL).

**Table 10:** Tested hyper-parameters for NN (I)

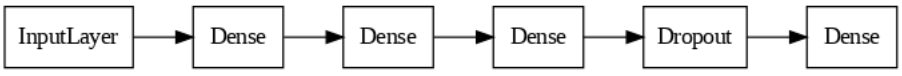
| Hyper-parameter | Description | Tested Values |
|---|---|---|
| Size | Number of units of hidden layer | Powers of 2 in the interval $[2, 2^8]$ |
| Epochs | Number of epochs for convergence | 10, 20, 50, 100, 200 |
| $\eta$ | Learning rate | Powers of 10 in the interval $[10^{-3}, 1]$ |
| $\alpha$ | Momentum coefficient | Powers of 10 in the interval $[10^{-3}, 1]$ |
| $\lambda$ | L2 regularization coefficient | Powers of 10 in the interval $[10^{-3}, 1]$ |

The best resulting model has been trained on TR in order to estimate the generalization error on VL and visualize the learning curves (Figure 8). We have also implemented early stopping in order to set a suitable number of epochs for convergence. The final model is therefore retrained on all data (TR + VL) and tested on TS.

For the *emotional_ intensity* and *emotion* targets the model proposed in Table 10 did not outperform the baseline (LG). For this reason, we have tested two deeper architectures with Adam and drop-out regularization (Figure 7).




(a)




(b)

**Figure 7:** Architectures of the deeper NNs tested for the targets *emotional_ intensity* (a) and *emotion* (b).

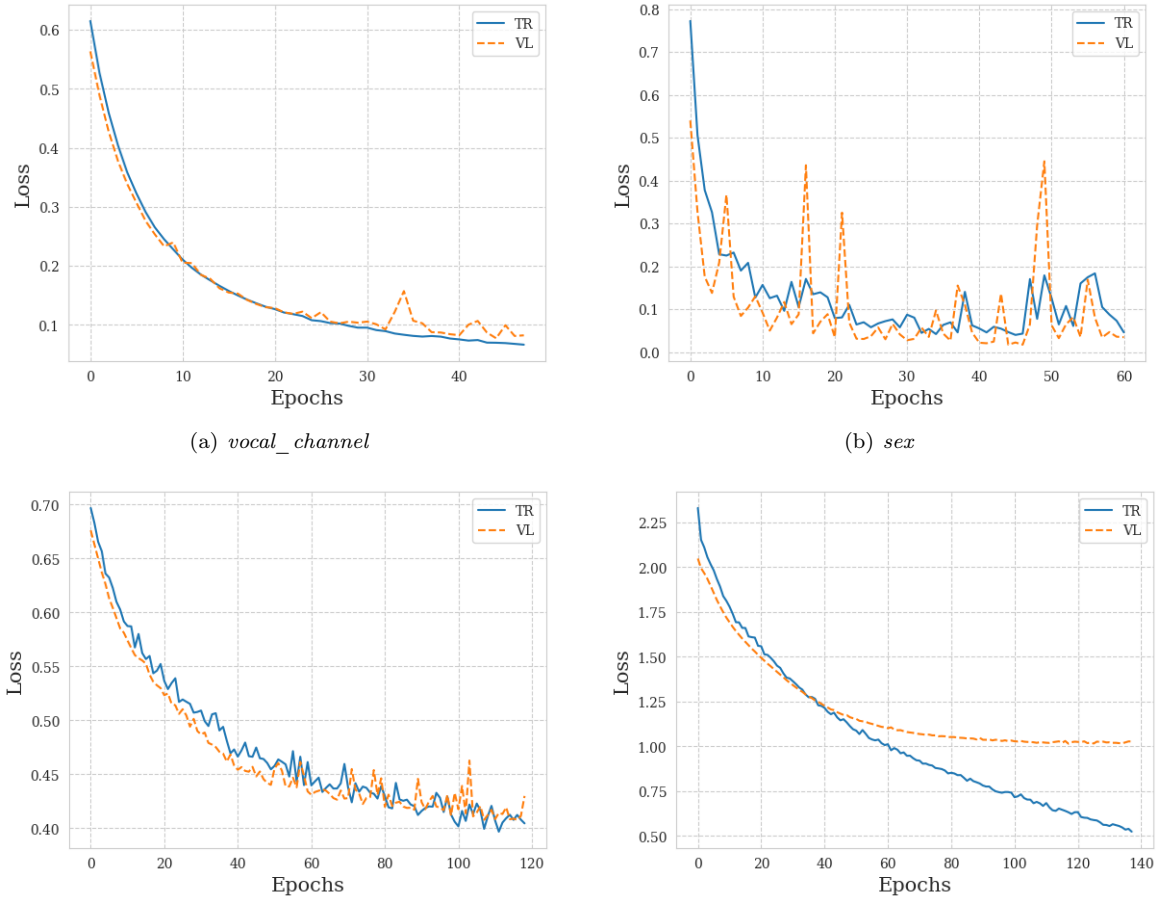
Tested hyper-parameters are reported in Table 11.

**Table 11:** Tested hyper-parameters for NN (II)

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| Size[6] | Number of units of hidden layer | Powers of 2 in the interval $[2, 2^8]$ |
| Epochs | Number of epochs for convergence | 10, 20, 50, 100, 200 |
| $\eta$ | Learning rate | Powers of 10 in the interval $[10^{-3}, 1]$ |
| $p$ | Drop-out rate | 0.2, 0.4, 0.6 |

[6]The same values are tested for each hidden layer in the architecture.

Table 12 reports the best results.




(a) *vocal_ channel* (b) *sex*

(c) *emotional_ intensity* (d) *emotion*

**Figure 8:** Learning curves of best models for each target

**Table 12:** Best results of NN

| Target | Size(s) | Epochs | $\eta$ | $\alpha$ | $\lambda$ | $p$ | Wt. Avg. F1 | Accuracy |
|---|---|---|---|---|---|---|---|---|
| *vocal_ channel* | 2 | 35 | 0.1 | 0.01 | 0 | - | 0.98 | 0.98 |
| *sex* | 128 | 61 | 1 | 0.001 | 0 | - | 0.95 | 0.95 |
| *emotional_ intensity* | 16, 8 | 118 | 0.001 | - | - | 0.2 | 0.77 | 0.78 |
| *emotion* | 256, 256, 256 | 290 | 0.0001 | - | - | 0.4 | 0.52 | 0.53 |

In order to better appreciate their robustness, each of the models in Table 12 has also been re-trained with 10 different random weights configurations. Table 13 reports the average and the standard deviation of the performance metrics over the 10 random initializations of the weights.

**Table 13:** Average and Standard Deviation of performance metrics

| Target | Wt. Avg. F1 | Accuracy |
|---|---|---|
| *vocal_ channel* | $0.96 \pm 0.02$ | $0.96 \pm 0.02$ |
| *sex* | $0.95 \pm 0.00$ | $0.95 \pm 0.00$ |
| *emotional_ intensity* | $0.77 \pm 0.01$ | $0.77 \pm 0.01$ |
| *emotion* | $0.51 \pm 0.01$ | $0.52 \pm 0.01$ |

## 5.4 Ensemble Models

We have analyzed the performance of four ensemble classifiers: Bagging with Decision Tree (DTB), Random Forest (RF), AdaBoost (AB) and Light Gradient Boosting Machine (GB).

For DTB, RF and AB model selection has been carried out with a randomized search with 3-fold CV. Tested hyper-parameters and correspondent values can be found in Tables 14, 15 and 16. For both models the hyper-parameter space includes the hyper-parameters of the base estimator (Table 3). Differently, for AB we assume that the base estimator is a decision stump.

**Table 14:** Tested hyper-parameters for Decision Tree Bagging

| Hyper-Parameter | Description | Tested Values |
|---|---|---|
| Max Samples | Maximum number of samples to be used for training base estimator | 0.5, 0.6, 0.7, 0.8 |
| Max Features | Maximum number of features to be used for training base estimator | Discrete interval [2, $N$] [7] |

**Table 15:** Tested hyper-parameters for Random Forest

| Hyper-Parameter | Description | Tested Values |
|---|---|---|
| Max Features | Maximum number of features for selecting the best split | $\sqrt{N}$, $\log_2(N)$, $N$ [8] |

**Table 16:** Tested hyper-parameters for AdaBoost

| Hyper-Parameter | Description | Tested Values |
|---|---|---|
| Learning Rate | Weight applied to each classifier at each boosting iteration | Powers of 10 in the interval [$10^{-4}$, 1] |

In Tables 17, 18, 19 we report respectively the best results obtained with DTB, RF and AB.

**Table 17:** Best results of Decision Tree Bagging

| Target | Criterion | Max Depth | Min Split | Min Leaf | Max Samples | Max Features | Wt. Avg F1 | Accuracy |
|---|---|---|---|---|---|---|---|---|
| *vocal_channel* | Entropy | 58 | 0.022 | 0.007 | 0.7 | 330 | 0.95 | 0.95 |
| *sex* | Log-Loss | 51 | 0.010 | 0.0028 | 0.7 | 78 | 0.88 | 0.88 |
| *emotional_intensity* | Gini | 16 | 0.013 | 0.016 | 0.6 | 381 | 0.75 | 0.75 |
| *emotion* | Entropy | 23 | 0.027 | 0.0017 | 0.7 | 126 | 0.39 | 0.42 |

**Table 18:** Best results of Random Forest

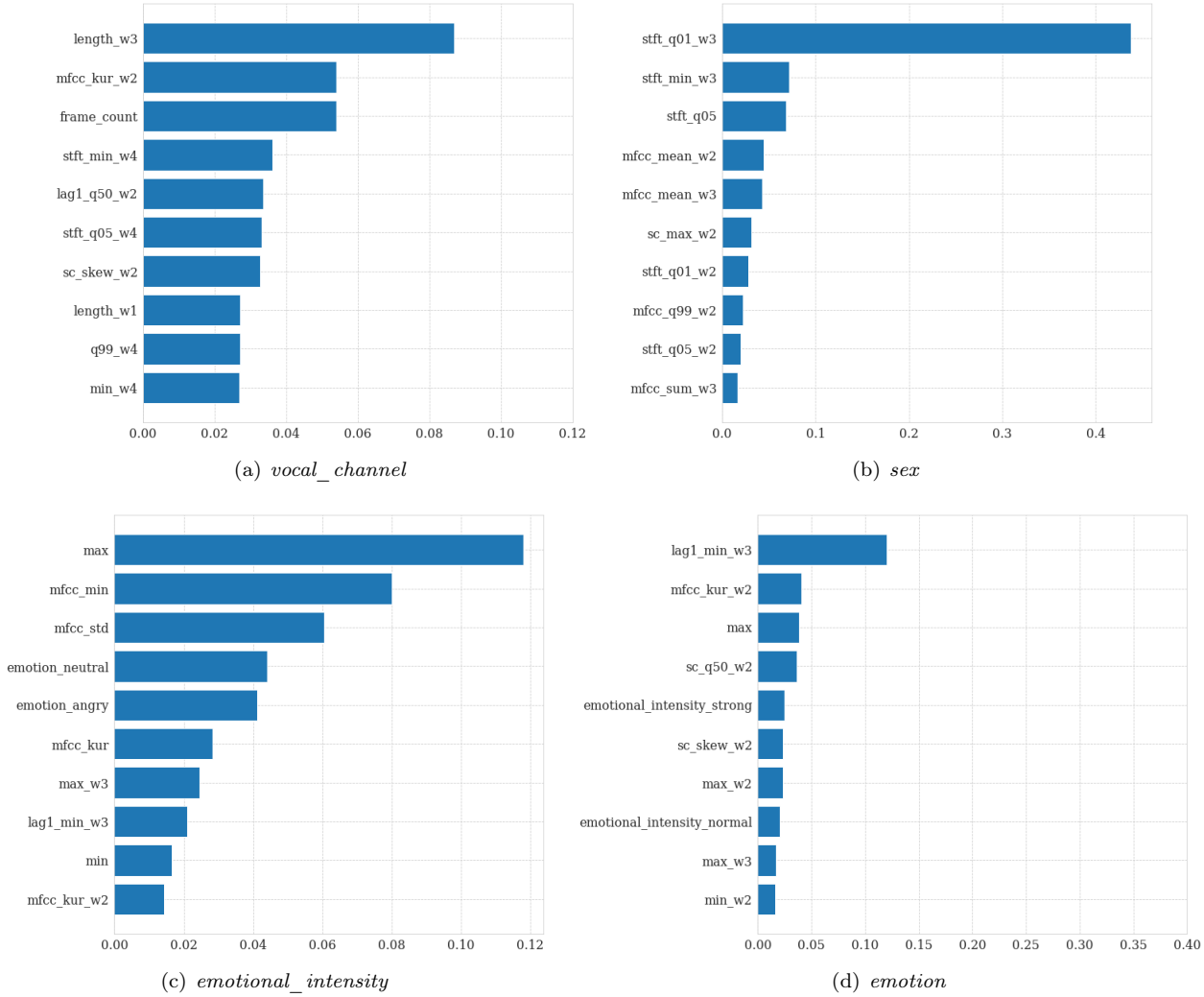
| Target | Criterion | Max Depth | Min Split | Min Leaf | Max Features | Wt. Avg F1 | Accuracy |
|---|---|---|---|---|---|---|---|
| *vocal_channel* | Gini | 75 | 0.043 | 0.0044 | $\sqrt{N}$ | 0.96 | 0.96 |
| *sex* | Entropy | 78 | 0.011 | 0.005 | $N$ | 0.85 | 0.85 |
| *emotional_intensity* | Gini | 38 | 0.018 | 0.0040 | $N$ | 0.77 | 0.77 |
| *emotion* | Log-Loss | 81 | 0.021 | 0.010 | $N$ | 0.41 | 0.43 |

**Table 19:** Best result of AdaBoost

| Target | Learning Rate | Wt. Avg. F1 | Accuracy |
|---|---|---|---|
| *vocal_channel* | 0.0001 | 0.89 | 0.89 |
| *sex* | 0.0001 | 0.83 | 0.83 |
| *emotional_intensity* | 0.0001 | 0.64 | 0.64 |
| *emotion* | 0.0001 | 0.29 | 0.30 |

[7] Where $N$ is the number of records in TR.
[8] Where $N$ is the original number of features.

A further analysis has concerned the computation of the importance of each input feature in RF's predictive performance (Figure 9), which allows to gain a more in-depth understanding of the information used by the model to discriminate between classes of the target variable.




(a) *vocal_channel* (b) *sex*

(c) *emotional_intensity* (d) *emotion*

**Figure 9:** Top 10 most important input features for the classification of each target with Random Forest

As evidenced, *emotion* and *emotional_intensity* share a significant number of their top 10 most important features – *max*, *max_w3*, *lag1_min_w3*. Furthermore, some emotions – *neutral* and *angry* – seem to be significant in the classification of *emotional_intensity*, and both of *emotional_intensity*'s values – *strong* and *normal* – seem to play a significant role in the recognition of emotions. On the other hand, *vocal_channel* shares only one attribute with *emotional_intensity* and *emotion* (*mfcc_kur_w2*), whereas none of the most important features for the recognition of *sex* are present in the other targets.

Model selection for GB has consisted of a randomized search with 5-fold CV (Table 20). Additionally, a model selection has been performed by exploiting automatic model transformation of categorical attributes in place of one-hot-encoding, but nonetheless the former has yielded a better performance. The best results are reported in Table 21.

**Table 20:** Tested hyper-parameters for Light Gradient Boosting

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| Boosting type | Gradient boosting algorithm | GBDT, GOSS, DART |
| Estimators | Number of boosted trees | Uniform distribution in the interval $[50, 500]$ |
| $\eta$ | Boosting learning rate | Powers of 10 in the interval $[10^{-4}, 1]$ |

continue on the next page

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| Leaves | Maximum tree leaves for base learners | Uniform distribution in the interval $[5, 50]$ |

**Table 21:** Best results of Light Gradient Boosting

| Target | Boosting type | Estimators | $\eta$ | Leaves | Max depth | Wt. Avg. F1 | Accuracy |
|---|---|---|---|---|---|---|---|
| *vocal_ channel* | GOSS | 395 | 0.1 | 20 | 191 | 0.98 | 0.98 |
| *sex* | GOSS | 422 | 0.3 | 21 | 45 | 0.88 | 0.88 |
| *emotional_ intensity* | GBDT | 467 | 0.2 | 16 | 64 | 0.74 | 0.74 |
| *emotion* | GOSS | 432 | 0.1 | 49 | 35 | 0.46 | 0.48 |

## 5.5 Comparative Evaluation

In this concluding section we make a comparative evaluation of the classification models considered for the binary tasks (*vocal_ channel*, *sex*, *emotional_ intensity*) and the multi-class task (*emotion*). In order to compare the relative performance of each classifier, we consider as a baseline (alongside LG) the performance of a random classifier with accuracy and weighted average F1 equal to the inverse of the number of target classes – 0.5 for the binary tasks and 0.125 for the multi-class task.

Figure 10 displays the accuracy of each model for each task.

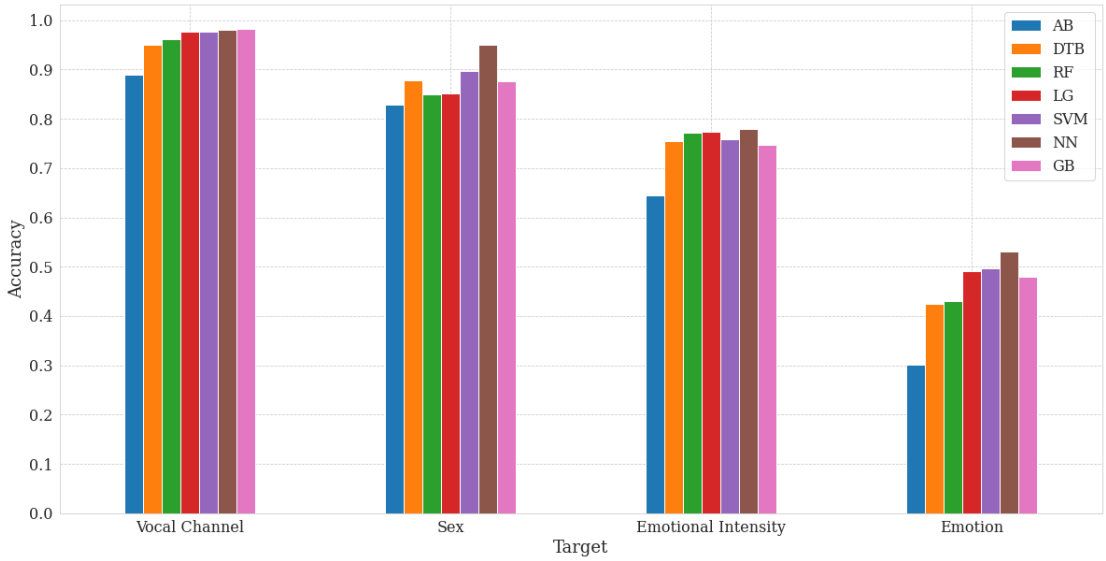
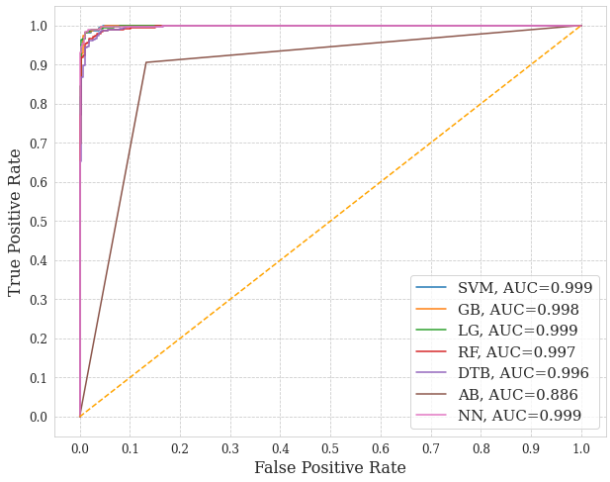


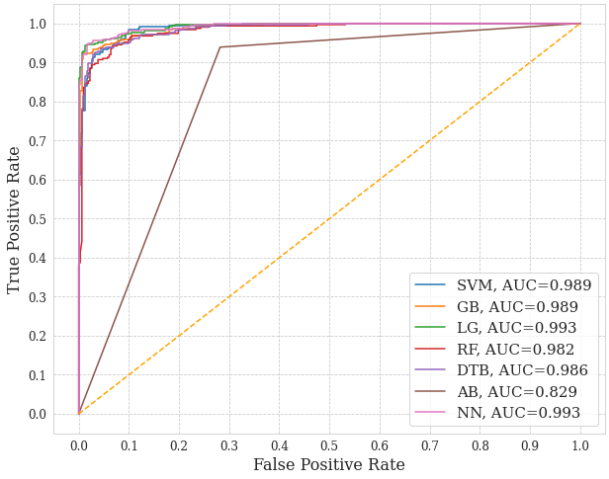

**Figure 10:** Accuracy of classification models for each target.

All the models reach optimal accuracy results in the classification of *vocal_ channel*, with slight better values for GB and NN. For the classification of *sex*, most of the models reach competitive results w.r.t. the baseline (LG), but a significantly higher accuracy is reached by NN. Differently, for the classification of *emotional_ intensity* only NN is able to outperform LG, followed by RF. These results are confirmed by the visual inspection of ROC curves and the correspondent AUC (Figure 11).

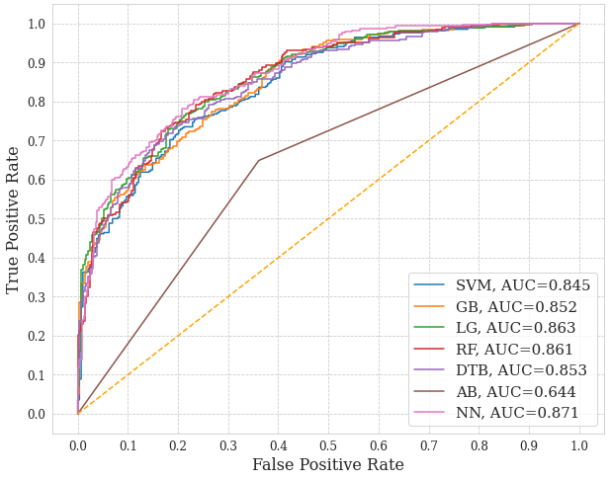
As evidenced in Figure 10, the multi-class problem is more challenging for all classifiers: even if all of them outperform the random prediction, only NN provides an accuracy above 0.5. A further comparison of the relative F1 scores can be useful to better appreciate the capability of each model to discriminate between emotions (Figure 12).




(a) *vocal_channel*




(b) *sex*




(c) *emotional_intensity*

**Figure 11:** ROC curves of classification models for each binary target.

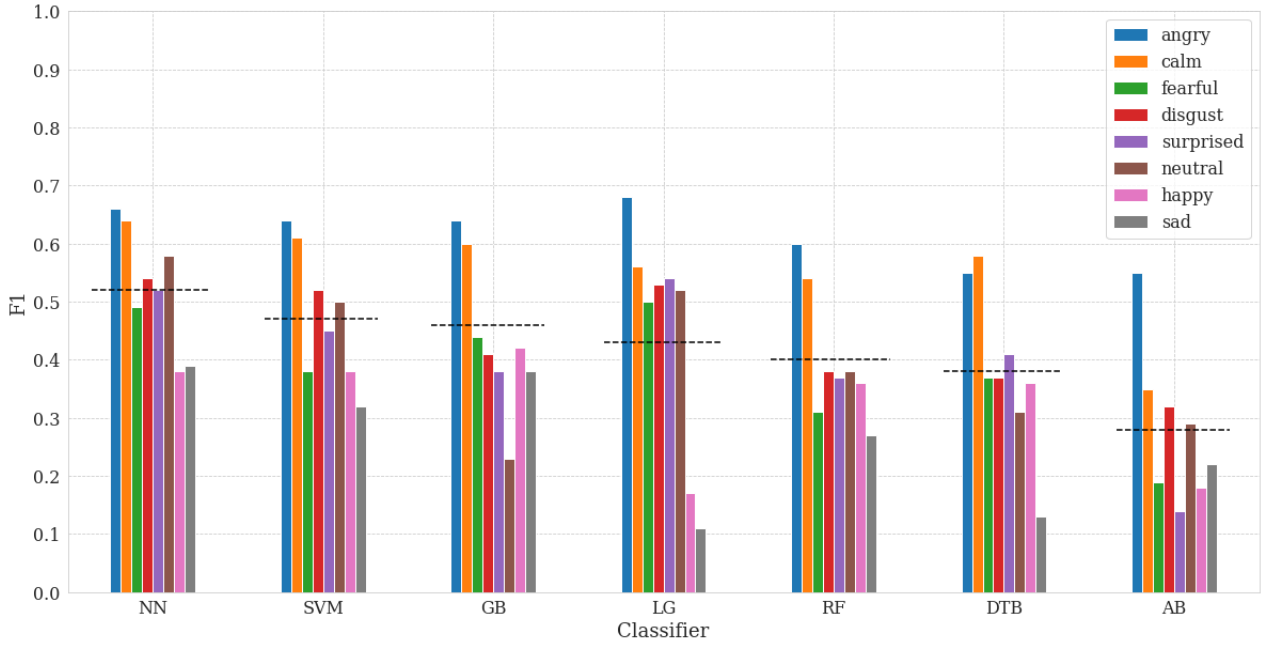



**Figure 12:** F1 scores computed by each classifier for each value of the *emotion* target. Models are sorted w.r.t. their weighted averages F1, which are indicated by the black dashed lines.

These results should be also compared with the capability of human raters in recognizing emotions from audio-only sources as reported by Livingstone & Russo (2018, p. 17), where the accuracy is 0.62 for emotions expressed through speech and 0.57 for emotions expressed through song.[9] A more in-depth confrontation reveals some remarkable analogies. Taking into account our best performing model (NN), the recalls of human classification and neural classification tend to follow a similar trend, i.e. emotions which are harder to be recognized by humans tend to be harder to be recognized by the NN (Figure 13).[10]

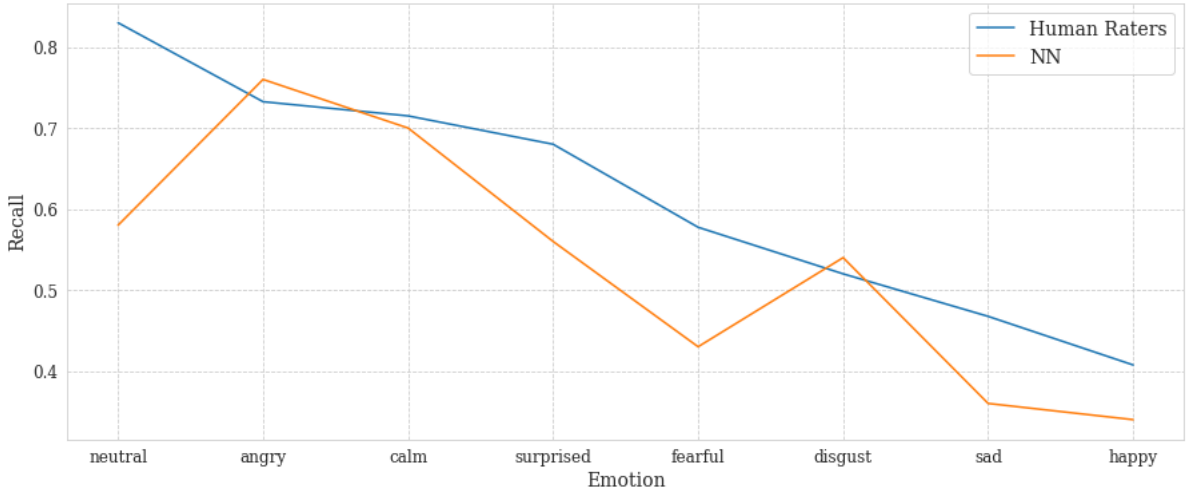



**Figure 13:** Recall of Human Raters and NN in recognizing emotions

On the other hand, a visual inspection of the confusion matrix of the neural classification (Figure 14) highlights similar mistakes committed by human classification (Livingstone & Russo 2018, p.18): *sad* is frequently confused with *calm*; *fearful* is often confused with *sad* and *angry*; *disgust* is mainly confused with *angry*; and *surprised* is often confused with *happy*.

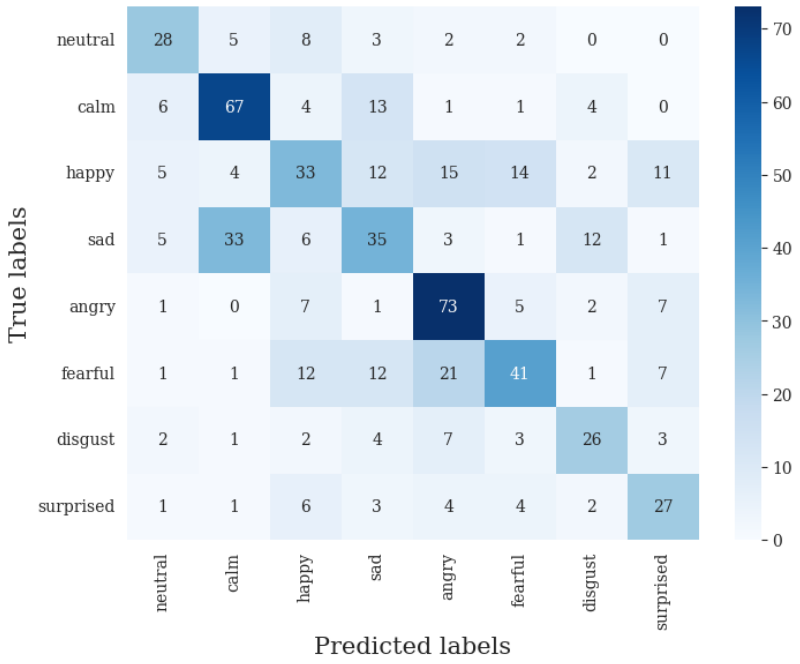



**Figure 14:** Confusion matrix of NN classification

[9] Accuracy is expressed by Livingston & Russo in terms of the mean proportion correct scores w.r.t. the total number of predictions made.

[10] We assume that the recall of human classification for a given emotion can be estimated as the average of the mean proportion correct scores for that emotion as reported by Livingstone & Russo (2018, p. 17) limitedly to audio-only (AO) predictions.

# 6 Regression

For the multiple regression task, the attribute *sc_ min* has been selected as dependent variable for its non-linear relationship with the TR data, as suggested by the results of Ordinary Linear Regression (OLR – Table 24). Pre-processing operations have consisted of the standardization (min-max) of numerical attributes and the one-hot encoding of categorical attributes.

We have tested the performance of several models: Support Vector Machines (SVM), Neural Networks (NN), Random Forest (RF), Decision Tree Bagging (DTB) and AdaBoost (AB). OLR and Decision Tree (DT) have also been considered as baseline models to assess the relative performance of each regressor.

For all the models, hyper-parameter selection has been performed *via* a randomized search with 3-fold CV and coefficient of determination ($R^2$) as scoring criterion. Tables 22 and 23 report the hyper-parameter spaces tested for DT and SVM.

**Table 22:** Tested hyper-parameters for Decision Trees

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| Criterion | Measure used to select the best split | Squared Error, Absolute Error, Friedman MSE, Poisson |
| Max Depth | Maximum depth of the tree | Discrete interval [2, 200] |
| Min Samples Split | Minimum number of samples to split an internal node | Log-uniform distribution in the interval [0.01, 1] |
| Min Samples Leaf | Minimum number of samples for a leaf node | Uniform distribution in the interval [0.001, 0.2] |

**Table 23:** Tested hyper-parameters for SVM

| Hyper-parameter | Description | Tested Values |
|---|---|---|
| C | Regularization parameter | Log-uniform distribution in the interval [$10^{-4}$ to $10^{4}$] |
| $\epsilon$ | Width of the $\epsilon$-insensitive loss | Powers of 10 in the interval [$10^{-4}$, $10^{4}$] |
| Kernel | Kernel function | Linear, Polynomial, RBF |

For RF and DTB the reader can refer to Tables 15 and 14, assuming that the hyper-parameters tested for the weak learners are the same reported in Table 22.

As for NNs, the training data used in the randomized search have been previously splitted from validation data (VL) with a 80%-20% proportion. The latter have been employed to visualize the learning curves of the best resulting model and control the convergence time. We have tested the performance of two architectures, sharing the same activation functions (logistic for hidden layers and linear for output layer), initialization function (Glorot normal) and batch-size (32): a network with a single hidden layer, optimized with gradient descent and regularized with L2; and a deeper network with two hidden layers, optimized with Adam and regularized with drop-out (Figure 15). Corresponding hyper-parameters tested in the randomized search are those reported, respectively, in Tables 10 and 11.

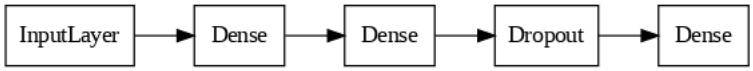



**Figure 15:** Architecture of the deeper network tested for *sc_ min* regression.

Table 24 shows the best models and their performance (TS) in terms of Mean Squared Error (MSE), Mean Absolute Error (MAE), Spearman correlation ($\rho$) and $R^2$.

**Table 24:** *sc_min* regression results

| Model | Selected hyper-parameters | MSE | MAE | $\rho$ | $R^2$ |
|---|---|---|---|---|---|
| OLR | | 5714.61 | 69.46 | 0.19 | -107887.08 |
| DT | *Criterion*: Friedman MSE<br>*Max Depth*: 171<br>*Min Samples Split*: 0.01<br>*Min Samples Leaf*: 0.02 | 0.003 | 0.03 | 0.96 | 0.94 |
| SVM | *Kernel*: RBF<br>C: 10<br>$\epsilon$: 0.001 | 0.009 | 0.06 | 0.89 | 0.84 |
| NN | *Sizes*: 128, 64<br>*Dropout rate*: 0.4<br>$\eta$: 0.001<br>*Epochs*: 200 | 0.007 | 0.05 | 0.90 | 0.87 |
| RF | *Criterion*: Poisson<br>*Max Depth*: 16<br>*Min Samples Split*: 0.03<br>*Min Samples Leaf*: 0.02<br>*Max Features*: None | 0.003 | 0.03 | 0.97 | 0.95 |
| DTB | *Criterion*: Friedman MSE<br>*Max Depth*: 59<br>*Min Samples Split*: 0.01<br>*Min Samples Leaf*: 0.02<br>*Max Samples*: 0.7 | 0.003 | 0.03 | 0.97 | 0.95 |
| AB | *Criterion*: Squared Error<br>$\eta$: 0.1<br>*Loss*: Square | 0.007 | 0.07 | 0.95 | 0.87 |

NN and AB perform slightly better than SVM, but none of them reaches competitive results w.r.t. to DT. The best results come from RF and DTB – with no visible differences, suggesting a negligible role of feature randomization – probably for a higher ease in controlling the variance of the model.

# 7 Time Series Analysis

In this section we address three kind of tasks – clustering, motif and discord discovery, classification – on time series which have been extracted from the raw audio signals of the original RAVDESS dataset with a sampling rate $f_s = 8$ kHz, resulting in 50718 timestamps. As in Section 1, time series data has been preliminarily partitioned into training data (TR) and test data (TS) with a fraction of 34.1% of all data reserved as TS.

Given the large amount of missing values in the terminal timestamps, time series have been reduced by computing their average length $\bar{n}$ without missing values and by removing all but the first $\bar{n}$ values from each time series, resulting in 32430 timestamps. Any remaining missing value has been replaced with the average value of the corresponding time series. The only further pre-processing operation undertaken on raw time series has concerned noise smoothing adopting moving average with window size 3. Other possible transformations – including normalization and approximation – will be considered where required.

## 7.1 Clustering

For performing time series clustering we have opted for K-Means and Hierarchical Agglomerative Clustering and limited the application of each algorithm to TR data. In order to improve the computational efficiency of each algorithm, time series have been approximated using Symbolic Aggregate Approximation (SAX) with 600

segments and 14 symbols.

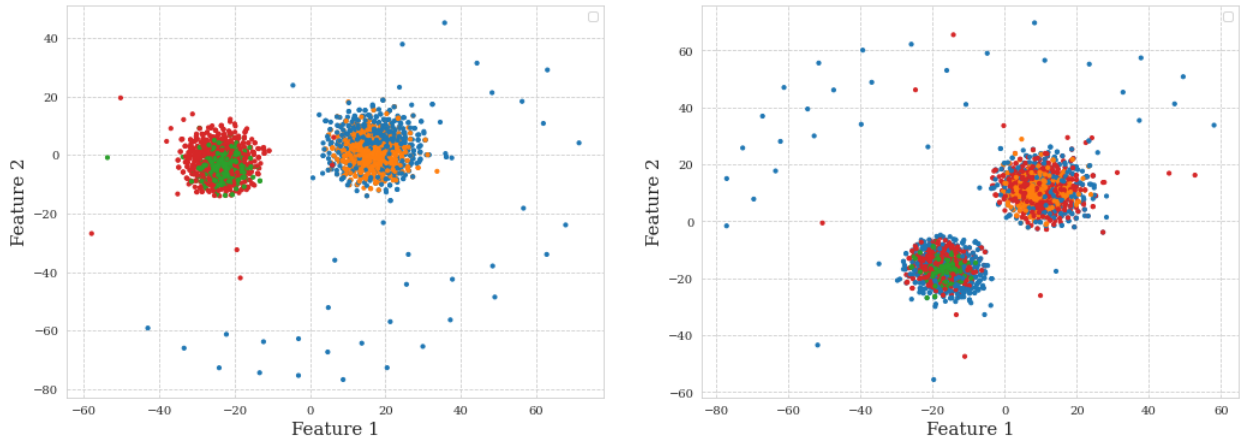
K-Means initialization has relied on the «elbow method», based on the selection of the value of $k$ which minimizes the Sum of Squared Error (SSE). We have tested four versions of K-Means, respectively with Euclidean distance, with Dynamic Time Warping (DTW), with DTW constrained by Sakoe-Chiba band and with DTW constrained by Itakura parallelogram (Table 25).

**Table 25:** K-Means Clusterings

| Metric | $k$ | SSE | Highest Purity |
|---|---|---|---|
| Euclidean | 4 | 84.86 | 0.75 (*vocal_ channel*) |
| DTW | 4 | 10.72 | 0.78 (*vocal_ channel*) |
| DTW (Sakoe-Chiba) | 4 | 55.84 | 0.78 (*vocal_ channel*) |
| DTW (Itakura) | 4 | 17.22 | 0.73 (*vocal_ channel*) |

The clustering obtained with DTW shows a significant decrease of the SSE and a modest increase in purity w.r.t. *vocal_ channel*. This proves that the capability of DTW in capturing the misaligned temporal structure of similar time series can be beneficial for the internal homogeneity of a cluster.

Figure 16 shows the clusterings obtained with Euclidean distance and DTW, where dimensionality reduction has been performed with t-SNE.

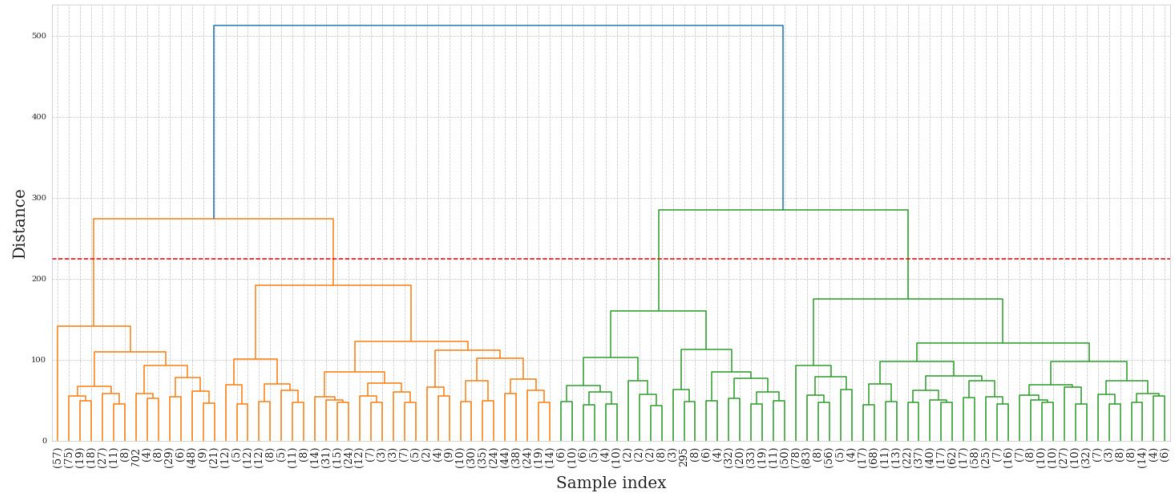



(a) *Euclidean Distance* (b) *DTW Distance*

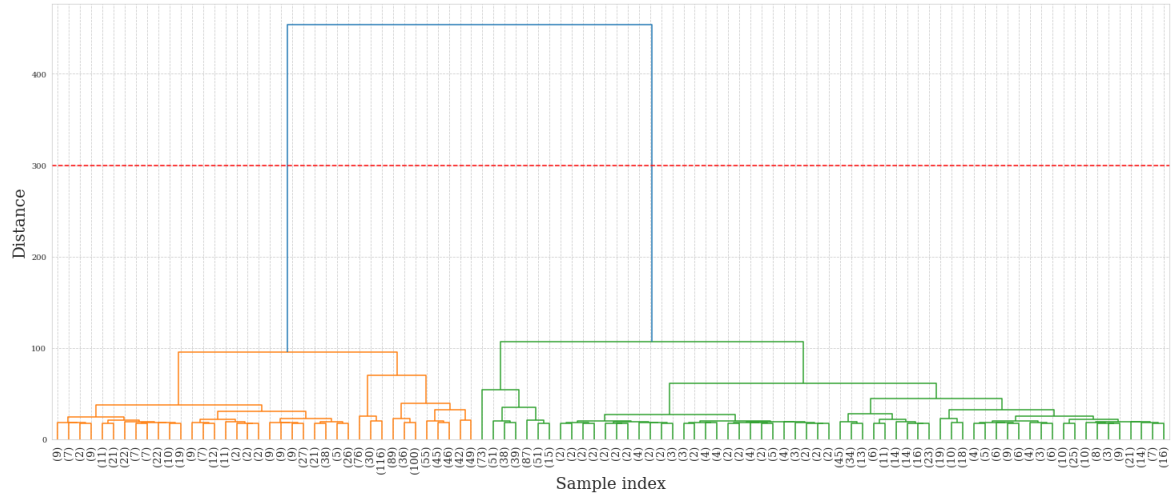
**Figure 16:** K-Means clusterings (t-SNE)

For the implementation of Hierarchical Agglomerative Clustering we have tested both Euclidean distance and DTW in combination with different proximity measures, i.e. Single Link, Complete Link, Group Average and Ward's method (the latter only for Euclidean distance). The assessment of possible candidate clusterings has been conducted through the visual inspection of resulting dendograms (Figure 17). A common trend has been noticed: Single Link and Group Average have given unsatisfactory results independently from the distance measure employed, while Complete Link and Ward have output more readable dendograms. Specifically, optimal results have been found by using Complete Link for DTW and Ward's method for Euclidean distance.

As displayed in Figure 17, the first clustering has been obtained by cutting dendogram (a) at a distance equal to 225, resulting in 4 clusters; whereas the second has been obtained by cutting dendogram (b) at a distance equal to 300, resulting in 2 clusters. Figure 18 shows the two clusterings by exploiting PCA dimensionality reduction.

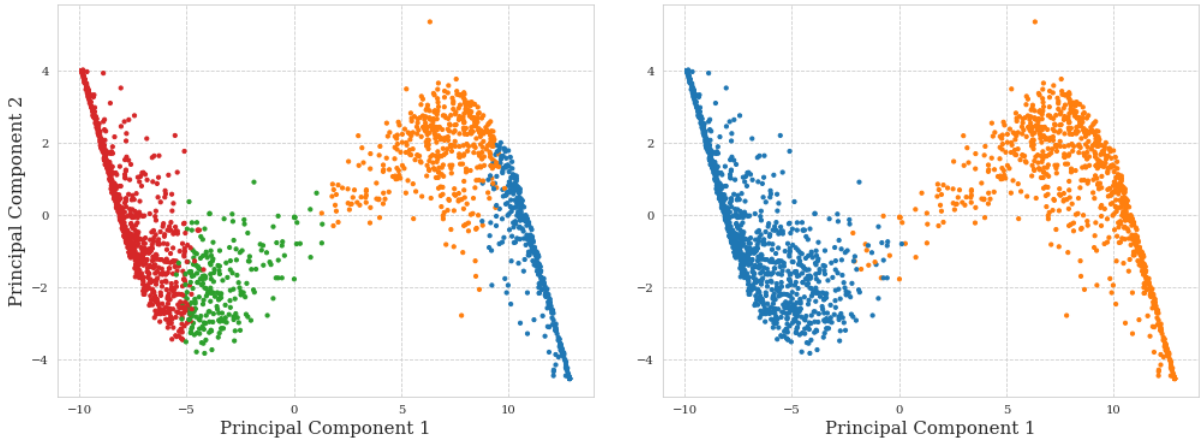



(a) *Complete Link and DTW Distance*




(b) *Ward's method and Euclidean Distance*

**Figure 17:** Dendrograms using Complete Link and DTW (a) and Ward's method and Euclidean distance (b). Red dashed horizontal lines indicate the cut locations.




(a) *Complete Link and DTW Distance*

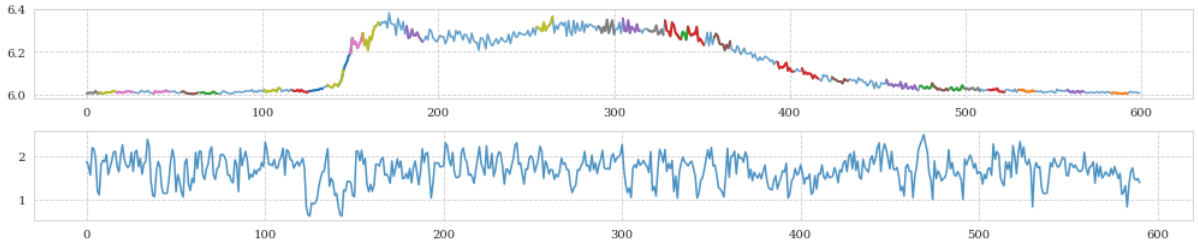
(b) *Ward's Method and Euclidean Distance*

**Figure 18:** Hierarchical clusterings (PCA)

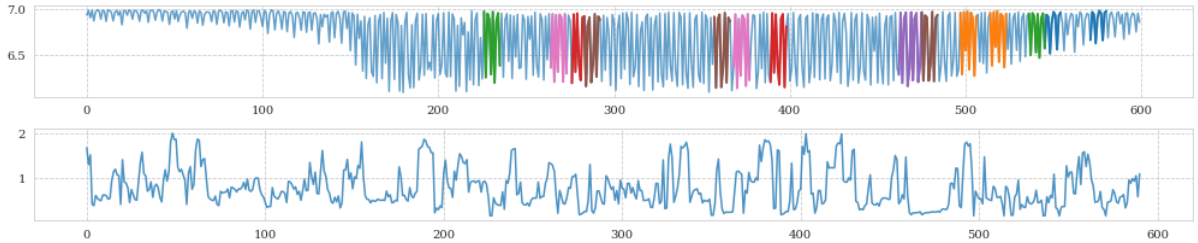
## 7.2 Motif and Anomaly Discovery

In this section we present the extraction of the top-10 motifs and the top-5 anomalies from two time series: a centroid from the K-Means clustering with Euclidean distance; and a centroid from the K-Means clustering with DTW constrained by Sakoe-Chiba band. For handling both tasks we use the approach based on the computation of a Matrix Profile (MP) with time window set to 10.

Figure 19 displays the motifs detected in each centroid and the correspondent MP.



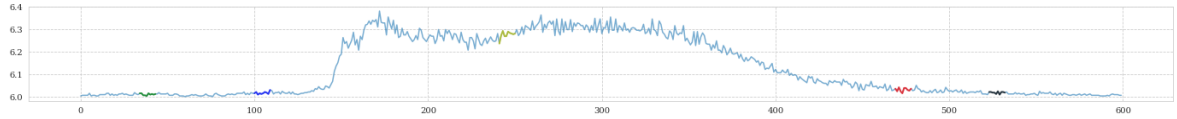
(a) *Centroid from K-Means with Euclidean distance*



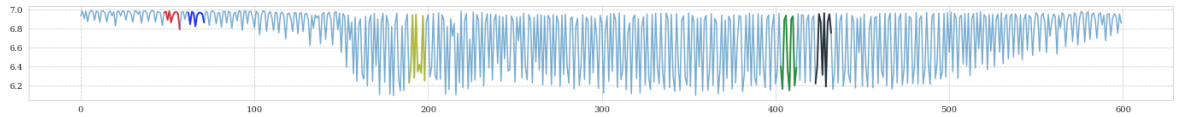
(b) *Centroid from K-Means and DTW with Sakoe-Chiba*

**Figure 19:** Motifs and MP of centroids

For handling the anomaly discovery task the exclusion zone has been set to half of time window used in the MP (5). Figure 20 displays the anomalies detected in each centroid.



(a) *Centroid from K-Means with Euclidean distance*



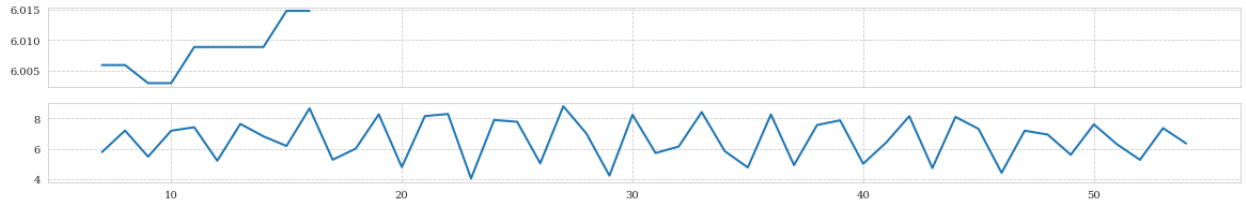
(b) *Centroid from K-Means and DTW with Sakoe-Chiba*

**Figure 20:** Anomalies of centroids

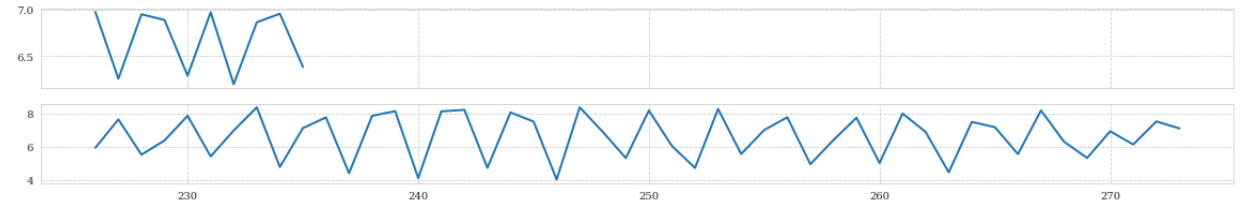
A futher analysis has concerned the comparison between motifs and shapelets. Since motifs have been detected in the centroids of K-Means clusterings, we compute shapelets w.r.t. the class which has proved to have the maximum purity with each clustering (*vocal_channel*). Shapelets extraction has been performed with the algorithm proposed by Grabocka et al. (2014) with Adam as loss function and L2 as regularizer with coefficient $\lambda = 0.01$. We use only one shapelet length, namely a fraction of 0.1 of the total length of the time series.

The 5 shapelets found have been compared with each centroid for looking at the best matching location w.r.t. Euclidean distance. Therefore for each shapelet and centroid we look whether its best matching location corresponds to the index of a motif in the MP. Only two shapelets where found in the same location of a motif, i.e. one shapelet in the location of a motif in the centroid of K-Means clustering with Euclidean distance; and one shapelet in the location of a motif in the centroid of K-Means clustering with DTW constrained by Sakoe-Chiba.

Figure 21 shows the comparison between each each motif and corresponding shapelet.




(a) *Motif (above) of centroid from K-Means clustering with Euclidean distance and corresponding shapelet (below)*




(b) *Motif (above) of centroid from K-Means clustering with DTW distance constrained by Sakoe-Chiba and corresponding shapelet (below)*

**Figure 21:** Comparison between Shapelets and Motifs

## 7.3 Classification

Time series classification has been performed on the same targets employed in Section 5, namely *vocal_ channel*, *sex*, *emotional_ intensity* and *emotion*. The models we have considered for solving these tasks are: K-Nearest Neighbors (KNN) with both Minkowski distance and Dynamic Time Warping (DTW), K-Nearest Neighbors (Minkowski) with shapelets (KNN-Shap), Canonical Interval Forest (CIF) and Random Convolutional Kernel Transform (ROCKET) with a ridge classifier.

Time series for KNN, KNN-Shap, and CIF have been preliminarly approximated with Piecewise Aggregate Approximation (PAA) set with 300 segments.

Model selection for KNN with Minkowski distance has been performed with a randomized search on the same hyper-parameter space described in Table 4, where the tested values for the metric have been restricted to Euclidean ($p = 2$) and City-Block ($p = 1$). Given the larger computational cost of DTW, we have adopted Sakoe-Chiba band as a global constraint and we have initialized K-NN with DTW with the same hyper-parameter configuration of the best resulting KNN with Minkowski distance. Results are reported in Table 26.

**Table 26:** Results of K-NN with Minkowski and DTW distances

| Target | $K$ | Weights | Metric | Wt. Avg. F1 | Accuracy |
|---|---|---|---|---|---|
| *vocal_ channel* | 20 | Distance | Euclidean | 0.47 | 0.59 |
| | | | DTW | 0.71 | 0.73 |
| *sex* | 34 | Uniform | City-Block | 0.56 | 0.59 |
| | | | DTW | 0.69 | 0.71 |
| *emotional_ intensity* | 105 | Distance | Euclidean | 0.38 | 0.54 |
| | | | DTW | 0.59 | 0.64 |
| *emotion* | 73 | Distance | Euclidean | 0.10 | 0.18 |
| | | | DTW | 0.34 | 0.37 |

For each target, shapelets extraction has been performed with the algorithm proposed by Grabocka *et al.* (2014) with Stochastic Gradient Descent as loss function and L2 as regularizer with coefficient $\lambda = 0.1$. As before, model selection has been performed randomized search on the same hyper-parameter space described in Table 4 with Euclidean and City-Block as possible metrics. Best results are reported in Table 27.

**Table 27:** Results of KNN-Shap

| Target | $K$ | Weights | Metric | Wt. Avg. F1 | Accuracy |
|---|---|---|---|---|---|
| *vocal_channel* | 79 | Distance | City-Block | 0.62 | 0.63 |
| *sex* | 43 | Distance | City-Block | 0.65 | 0.65 |
| *emotional_intensity* | 303 | Distance | City-Block | 0.69 | 0.70 |
| *emotion* | 166 | Distance | Euclidean | 0.24 | 0.30 |

As for CIF and ROCKET, the former has been initialized with 200 estimators and Decision Tree as base model; and the latter has been initialized with 10000 convolutional kernels, and the resulting representations have been employed for a simple linear classification with L2 regularization ($\lambda = 1$). The results are reported respectively in Tables 28 and 29.

**Table 28:** Results of CIF

| Target | Wt. Avg. F1 | Accuracy |
|---|---|---|
| *vocal_channel* | 0.90 | 0.90 |
| *sex* | 0.64 | 0.64 |
| *emotional_intensity* | 0.66 | 0.66 |
| *emotion* | 0.39 | 0.41 |

**Table 29:** Results of ROCKET

| Target | Wt. Avg. F1 | Accuracy |
|---|---|---|
| *vocal_channel* | 0.93 | 0.93 |
| *sex* | 0.70 | 0.69 |
| *emotional_intensity* | 0.67 | 0.66 |
| *emotion* | 0.34 | 0.28 |

Figure 22 displays the accuracy of each model for each task. Given that KNN with Minkowski and KNN with DTW are generally regarded as a baseline for more advanced models, we can see that only CIF outperforms them in all the classification tasks.

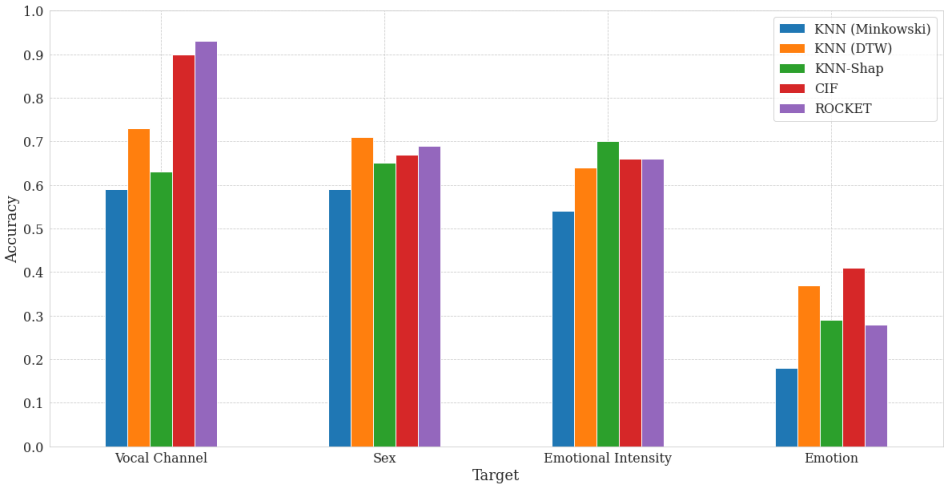



**Figure 22:** Accuracy of time series classification models for each target

# 8 Explainable AI

In this final section we present briefly two local, model-agnostic post-hoc explainability techniques, namely Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP), for providing an explanation of the decision logic of the non-linear SVM (Table 9) for the classification of a TS record $x$ into one of the *emotion* classes.

The output of LIME are continuous values which reflect the contribution of each feature for predicting the target class of $x$. This provides local interpretability of the behaviour of the black box model, and specifically

its response to random permutations of input features. Figure 23 shows feature importances computed by LIME w.r.t. the target pattern $x$, whose ground truth is *neutral*.

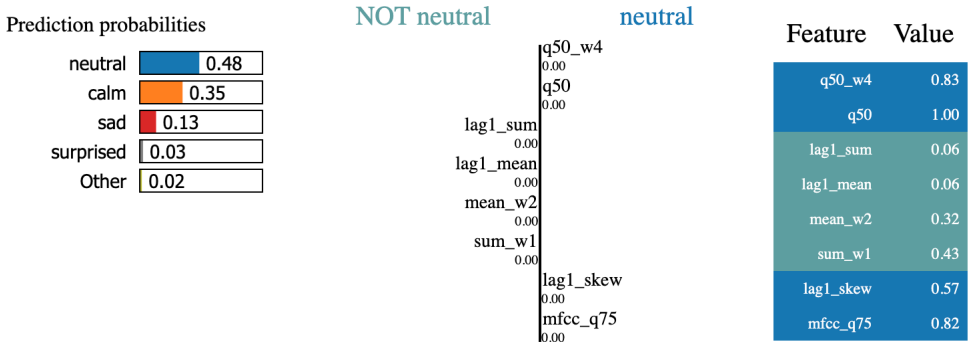



**Figure 23:** LIME explanation for pattern $x$. Input features on the left side contribute to increase prediction probabilities for classes «not neutral»(*calm*, *sad*), whereas input features on the right side contribute to increase prediction probability for *neutral*.

Similarly to LIME, SHAP locally explains the behaviour of the black box model by providing measures for the impact of each input feature to the final prediction. Figure 24 displays the results obtained with SHAP.


**Figure 24:** SHAP explanation for pattern $x$

In this plot, each feature corresponds to a horizontal bar, whose length is proportional with the magnitude of its impact on the black box prediction. The bars are color-coded w.r.t. the direction of input features' effect on the prediction. Positive impacts are shown in shades of red, indicating that higher values of features located in that position contribute positively to the prediction; whereas negative impacts are displayed in shades of blue, indicating that lower values of the input features' contribute negatively.

# References

Grabocka, J., Schilling, N., Wistuba, M., & Schmidt-Thieme, L. (2014). Learning Time-Series Shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 392-401.

Livingstone S.R. & Russo F.A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391.