Analysis Report of Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)*

Simone Artesi† Giacomo Fidone‡

November 2022

Abstract

This report describes an analysis of «Ryerson Audio-Visual Database of Emotional Speech and Song» (RAVDESS) intended to gain useful knowledge from data through the implementation of unsupervised methods (K-Means, DBSCAN, Hierarchical Clustering), the development of supervised models (Decision Trees, K-Nearest Neighbors, Naïve Bayes, Regression) and the extraction of patterns and association rules.

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

†Master in Data Science & Business Informatics – s.artesi[at]studenti.unipi.it

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

Contents

1 Data Semantics

«Ryerson Audio-Visual Database of Emotional Speech and Song» (RAVDESS)[1] is a validated multimodal dataset consisting of 24 professional actors vocalizing two lexically-matched statements in a neutral North American accent. In this analysis a modified version of RAVDESS will be considered, where the original records have been replaced with different quantitative measures of the audio signal. A complete description of the attributes can be found in Table 1.

Table 1: Attributes' description

Name(s)	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
channels	Numerical	Number of channels
sample_width	Numerical	Number of bytes per sample
frame_rate	Numerical	Frequency of samples used in Hertz
frame_width	Numerical	Number of bytes for each frame
length_ms	Numerical	Length of the record in ms
frame_count	Numerical	Number of frames per sample
intensity	Numerical	Intensity of sound in Db
zero_crossings_sum	Numerical	Sum of Zero Crossings Rates
mfcc_mean, *mfcc_std*, *mfcc_min*, *mfcc_max*	Numerical	Statistics of Mel-Frequency Cepstral Coefficients
sc_mean, *sc_std*, *sc_min*, *sc_max*, *sc_skew*, *sc_kur*	Numerical	Statistics of Spectral Centroid
stft_mean, *stft_std*, *stft_min*, *stft_max*, *stft_skew*, *stft_kur*	Numerical	Statistics of Short-Time Fourier Transform
mean, *std*, *min*, *max*, *skew*, *kur*	Numerical	Statistics of original audio signal

2 Data Understanding

In this section we seek for useful information about data using both visualization tools and statistical measures. In particular, this preliminary exploration will address univariate analysis, bivariate analysis and data quality assessment.

2.1 Univariate Analysis

Figure 1 catches the main properties of univariate distributions of numerical attributes, namely the values of skewness, kurtosis and number of outliers.[2]

[1] Livingstone S.R., Russo F.A. (2018)

[2] The kurtosis coefficient is computed with Fisher formula and outliers are defined as values above the upper-fence or below the lower-fence. The attribute *mean* is not displayed for its exceptional skewness (6.05) and kurtosis (398.3).

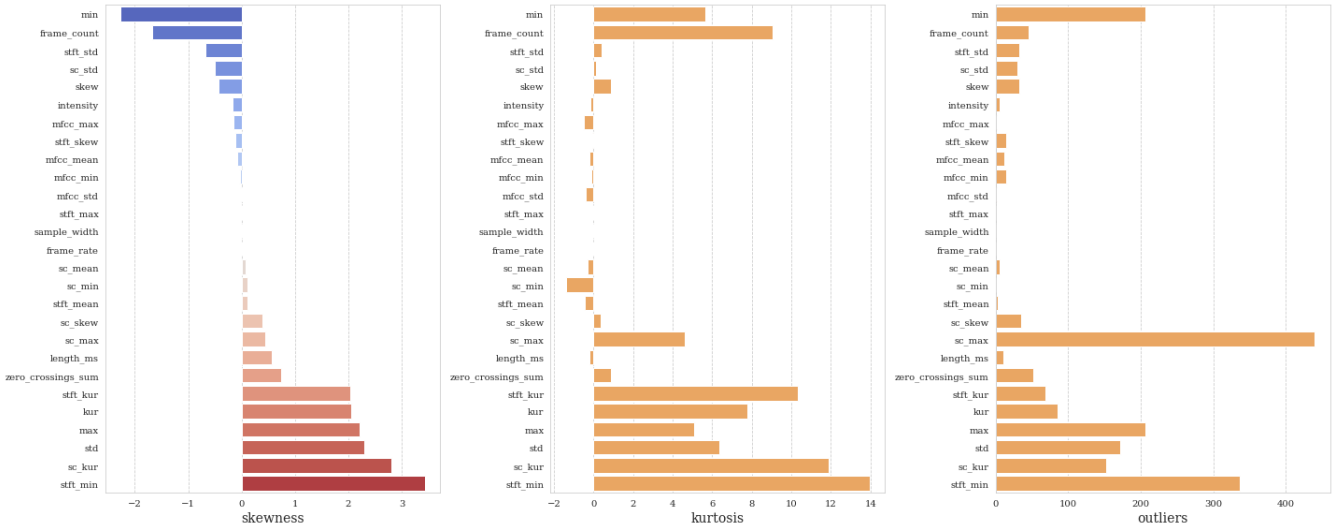


Figure 1: Skewness, kurtosis and number of outliers of numerical attributes (ordered by ascending skewness)

About 14% of numerical attributes are moderately skewed ($0.5 < skew < 1$ or $-1 < skew < -0.5$) and about 28% of them are highly skewed ($skew > 1$ or $skew < -1$). As evidenced, many skewed distributions are also leptokurtic and related to a larger number of outliers.

Among numerical attributes, *length_ ms*, *frame_ count* and *sc_ min* display a bi-modal distribution. As shown in Figure 2, the two peaks in *length_ ms* and *frame_ count* depend on the binary values of *vocal_ channel*:

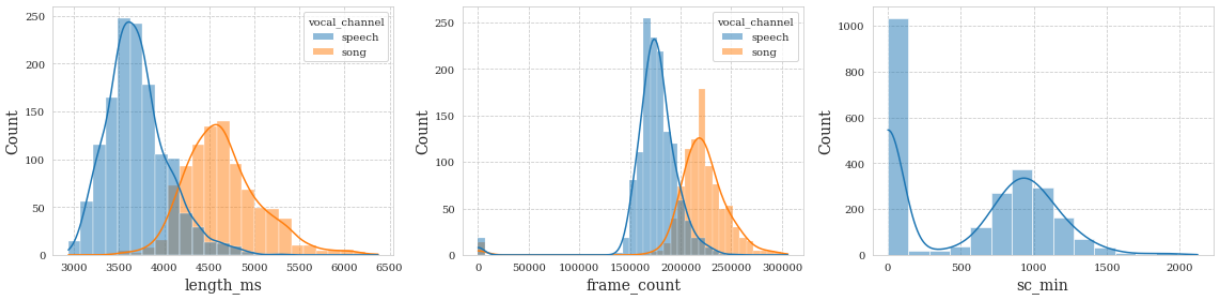


Figure 2: Bi-modal distributions of *length_ ms*, *frame_ count*, *sc_ min*

As to qualitative attributes, most of them have sufficiently balanced distributions due to convenient experimental-design choices. The only relevant exceptions can be found in *vocal_ channel* and *emotion* (Figure 3).

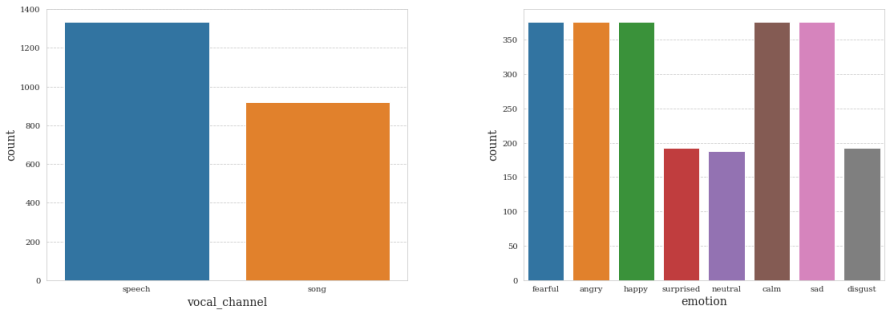


Figure 3: Distributions of *vocal_ channel* and *emotion*. As evidenced, in *vocal_ channel* the value *speech* occurs about 10% more than song, whereas in *emotion* the values *surprised*, *neutral* and *disgust* cover each about 7.7-7.9% of data, which is the half if compared to the other classes.

2.2 Bivariate Analysis

Since most of numerical attributes display a non-Gaussian distribution, linear correlations in the following heatmap (Figure 4) are computed according to the non-parametric Spearman's coefficient.

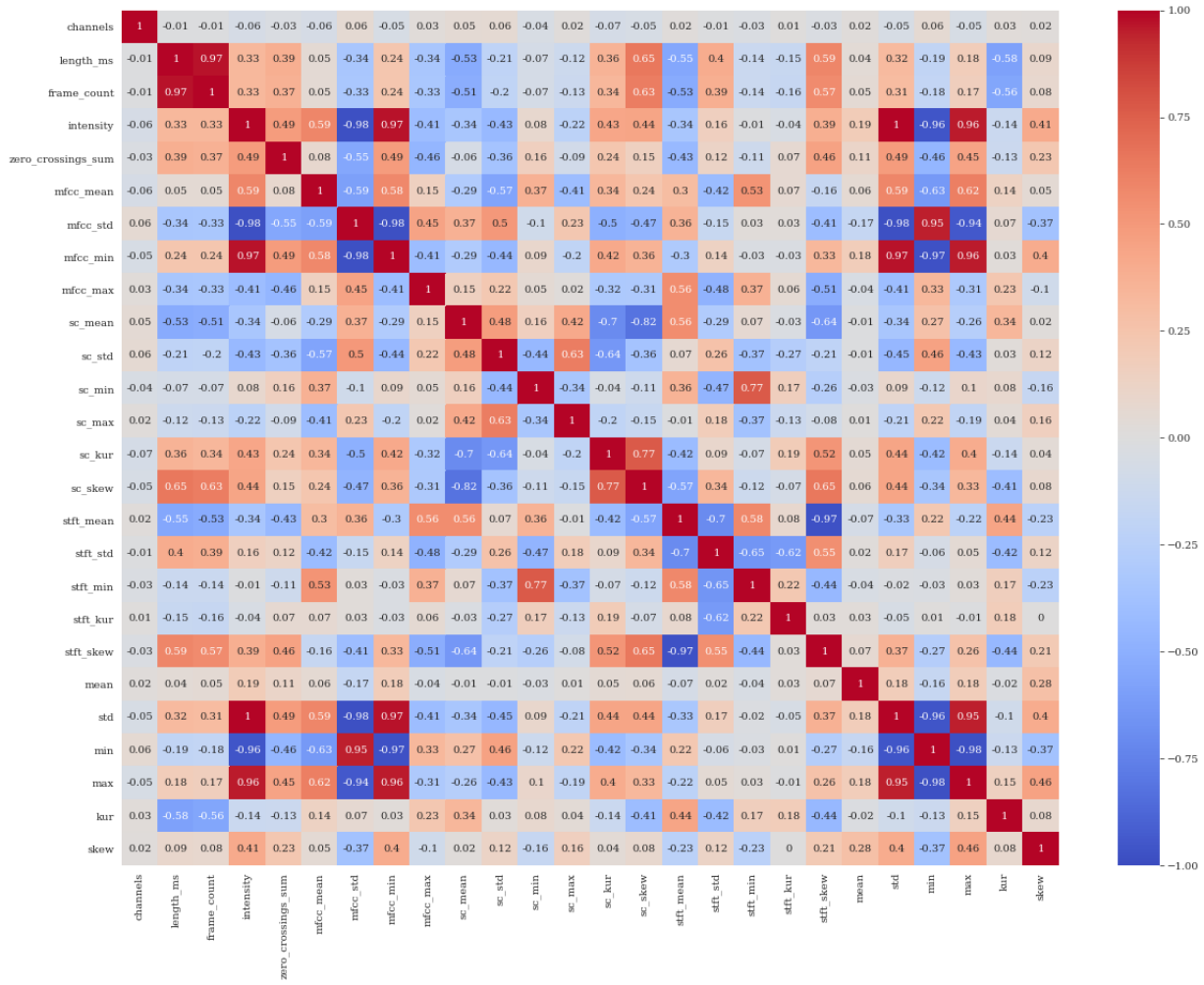


Figure 4: Correlations (Spearman) between quantitative attributes. Distributions with variance $\sigma^2 = 0$ have been omitted.

Association between nominal attributes has been evaluated analyzing the correspondent contingency tables. As before, categorical data seem to be sufficiently balanced: the only exception concerns *emotion* w.r.t. *vocal_ channel*, as shown in Figure 5.

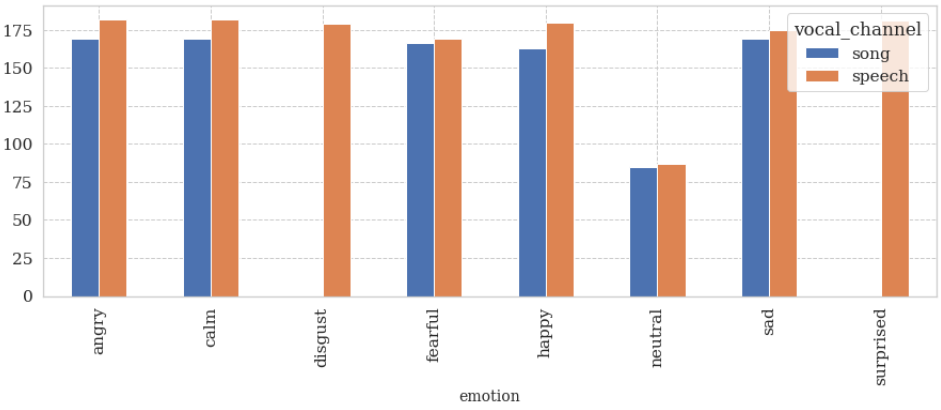


Figure 5: Distribution of *emotion* w.r.t. *vocal_ channel*.

Associations between qualitative attributes and numerical attributes have been analyzed supporting visual inspection with the computation of synthetic measures:

- For binary attributes, we have mapped the original nominal values onto Boolean ones (0, 1) and exploited Spearman Coefficient to understand how the binary values are distributed in the space of each continous attribute (Figure 6);

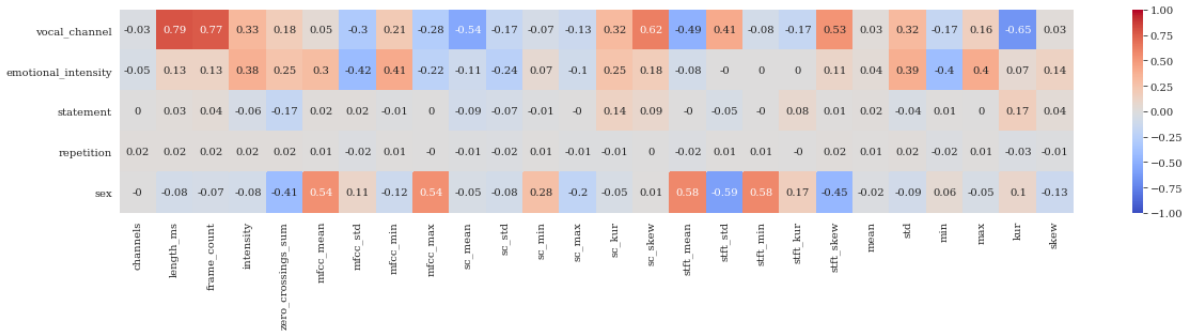


Figure 6: Correlations (Spearman) between binary attributes and numerical (continous) attributes.

- As *emotion* cannot mapped onto numerical values, we have opted for a different strategy: if there is a significant relationship between a nominal attribute and a numerical attribute, we should be able to build a classifier reaching a equally significant accuracy. We have therefore exploited Logistic Regression to estimate the association of *emotion* with numerical attributes (Figure 7).

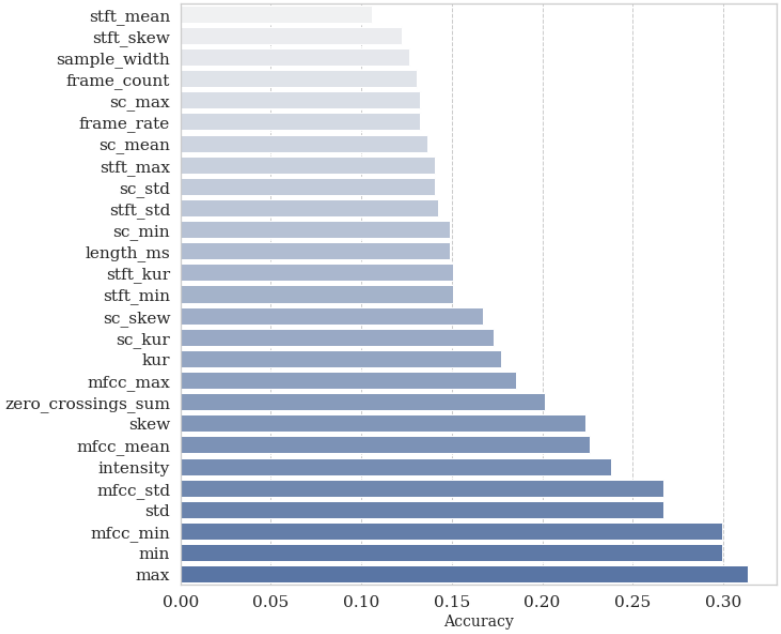


Figure 7: Accuracy of Logistic Regression in classifying *emotion* with each numerical attribute.

2.3 Data Quality Assessment

Evaluation of data quality has focused on:

- Missing values detection. Missing values have been found in *vocal_channel* (196), *actor* (1126) and *intensity* (816).
- Outliers detection. As shown above (Figure 1), outliers can be detected in many skewed quantitative distributions. However, only in $frame\_count$ they can be confidently interpreted as erroneous measurements as they consist of negative values (-1) discordant with a positive $length\_ms$. In other cases, no relevant inconsistencies have been found. Among categorical attributes, only *channels* has an evidently biased distribution: only few records (6) are associated with value 2.
- Data balancing. As mentioned above, data is sufficiently balanced with few exceptions.

Further checks on syntactic accuracy, semantic inconsistencies and duplicates have proved negative.

3 Data Preparation

In this section we discuss about pre-processing operations on data which have been considered convenient before the implementation of any supervised or unsupervised algorithm: feature selection, missing values replacement,

outliers' management and data transformation. Further data preparation issues may be addressed later where required.

3.1 Feature Selection

In order to decrease dimensionality, non-discriminative features have been excluded from the original dataset:

- Numerical attributes with variance $\sigma^2 = 0$: *sample_ width*, *frame_ rate*, *frame_ width*, *stft_ max*;
- Qualitative attributes with a single unique value: *modality*;

3.2 Missing Values Replacement

Given the modest size of the dataset, the management of missing values has aimed (where possible) at preserving observations with proper replacement strategies:

- Missing values in *vocal_ channel* have been estimated using a Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel. The SVM has been trained on *length_ ms*, given the already seen bimodal distribution (Figure 2), and selection of hyper-parameters C and γ has been performed with a simple hold-out validation, reaching a final accuracy of 0.90 on test set.[3]
- Missing values in *intensity* have been estimated using a simple linear regression model trained on *mfcc_ min*, given the large Spearman coefficient (Figure 3), with a final accuracy (R^2) of 0.94 on test set.

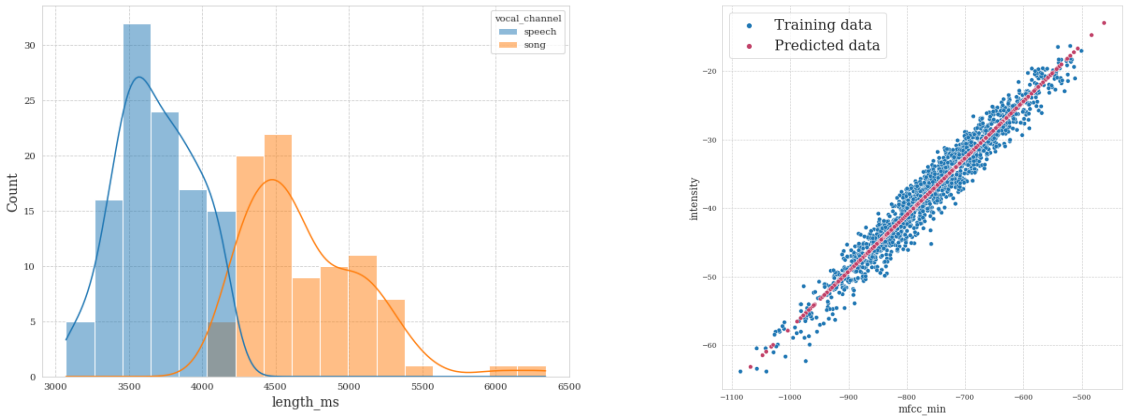


Figure 8: Predicted distribution of *vocal_ channel* w.r.t. *length_ ms* distribution (left) and predicted distribution of *intensity* w.r.t. *mfcc_ min* (right)

- A trickier problem has been posed by missing values in *actor*: they cannot be reasonably dropped for their large number; they cannot be replaced with a modal value without strongly unbalancing the data; they cannot be accurately estimated for the small size of training data. We have therefore decided to keep the *NaN* values and discard them only where required.

3.3 Outliers Management

As evidenced in section 2.3, outliers management cannot be separated from the access we have to the semantic interpretation of the attribute. For this reason, only the detected outliers in *frame_ count* have been replaced with the same approach proposed above, i.e. with a simple linear regression model trained on *length_ ms* which has reached a 0.99 accuracy on test set. As for all the other attributes, outliers will be managed only where required.

3.4 Data Transformation

Given the large number of skewed distributions (Figure 1), their normalization may be a necessary pre-processing operation for the successful application of both unsupervised and supervised methods. However, the kind of normalization chosen cannot again be separated from the semantic knowledge of data. Hence, data transformation has not aimed to replace the original dataset, but rather to create new ones which may be tested alongside the original one:

[3]The choice of SVM with RBF kernel was motivated by the non-linearly separability of data, assessed through visual inspection.

1. A partially-transformed dataset where only attributes with strictly positive values have been normalized;
2. A full-transformed dataset where all attributes have been normalized.[4]

Normalization functions – which have included logarithmic transformations, quadratic transformations and n-root transformations – have been specifically tailored to each attribute in order to approximate at best a Gaussian distribution.

4 Cluster Analysis

This section aims to exploit different unsupervised clustering techniques (K-means, DBSCAN, Hierarchical) in order to discover natural groupings in the data. Each cluster analysis has followed shared guidelines concerning:

- Feature scaling. Data have been standardized either with Z-scores or Min-Max;[5]
- Subset selection. We have considered two kinds of subsets: subsets comprising the statistics of single audio measurements (Mel-Frequency Cepstral Coefficients, Spectral Centroid, Short-Time Fourier Transform, audio signal); and subsets comprising features which have proved to have a significant association with a certain categorical attribute.[6]
- Validation. Cluster validity has been assessed on the basis of two kinds of criteria: internal criteria, i.e. degree of intra-cluster cohesion and inter-cluster separation (measured with Silhouette Coefficient); and/or external criteria, i.e. match with ground truth (measured with Purity).

4.1 K-means

Initialization of K-means has been performed with 100 preliminary iterations with random centroids selection in order to find the starting centroids configuration associated with the lowest Sum of Squared Error (SSE). Hyper-parameter k has been selected w.r.t. the two different validation criteria introduced above, i.e. by comparing different values of k and selecting the one which minimizes SSE (through the so-called «elbow method») and maximizes the Silhouette Coefficient; or simply by setting it to be equal to the number of classes of known nominal attributes.

As expected, K-means has globally worked better on full-normalized data. To further reduce noise, we have tested from time to time the efficacy of two additional strategies, i.e. outliers' elimination and Principal Component Analysis (PCA), which have regularly improved the internal cohesion and external separation of clusters. No relevant results have been found in high-dimensional data, neither by reducing it into a lower-dimensional space with PCA or with a so-called «filter approach» – consisting in selecting one attribute for each group of correlated attributes.

Table 1 shows the most relevant results.

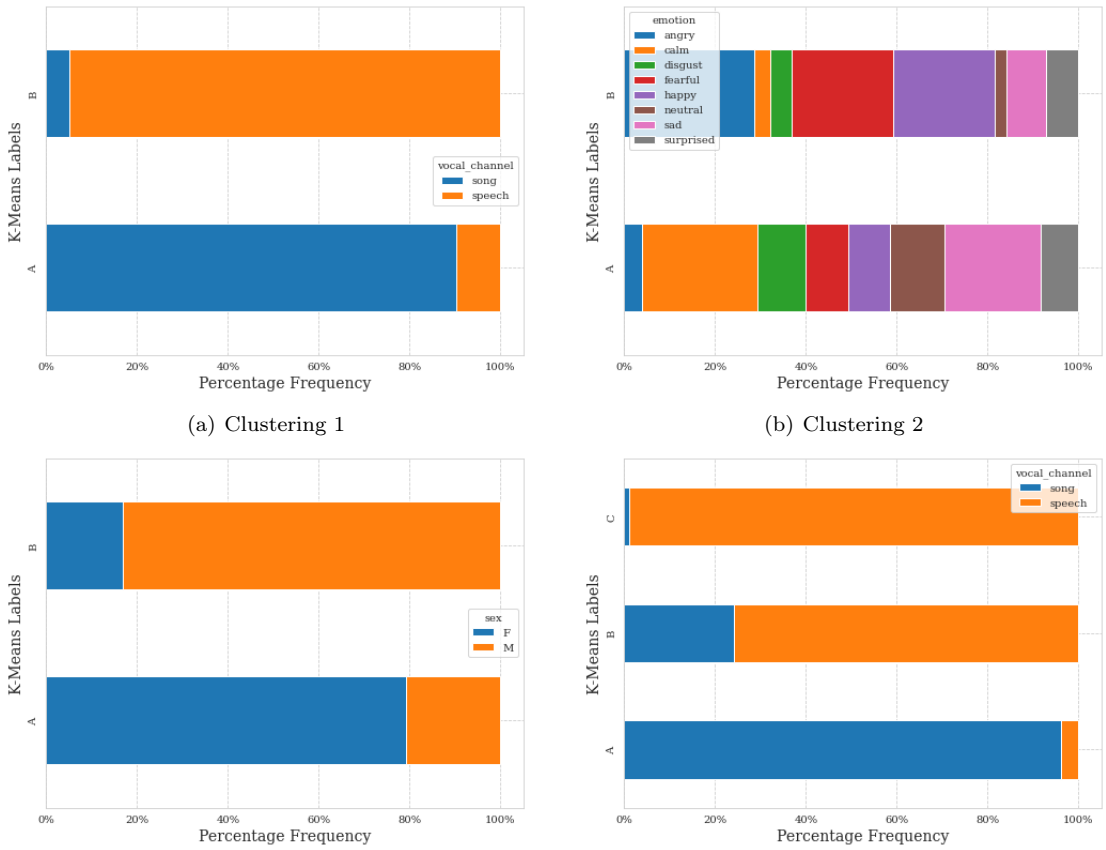
Table 2: Best clusterings found with K-Means

N	Subset	k	SSE	Silhouette	Best Purity
1	*length_ ms*, *frame_ count*, *sc_ skew*, *kur*	2	118.9	0.53	0.93 (*vocal_ channel*)
2	*max*, *min*, *std*, *mfcc_ std*, *mfcc_ min*, *intensity*, *mfcc_ mean*	2	221.6	0.49	0.76 (*emotion*)
3	*stft_ min*, *stft_ mean*, *stft_ std*, *stft_ skew*, *mfcc_ max*, *zero_ crossings_ sum*	2	273.3	0.50	0.81 (*sex*)
4	*length_ ms*, *frame_ count*, *sc_ skew*, *kur*, *sc_ mean*, *stft_ mean*, *stft_ skew*	3	102	0.4	0.63 (*vocal_ channel*)

[4]Since the sign of a value may be semantically relevant, this distinction is motivated by the lack of advantageous transformations able to preserve the negative sign.

[5]The choice of the standardization has proved not to be influent on the results.

[6]See section 2.2.



(a) Clustering 1

(b) Clustering 2

(c) Clustering 3

(d) Clustering 4

Figure 9: Percentage Frequencies of ground truths w.r.t. found clusterings.

4.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Hyper-parameters *Eps* and *MinPts* have been selected analyzing the plot displaying all the data points (x-axis) sorted by their k-distances – the distance from their k-nearest neighbor – w.r.t. the value of the k-distance (y-axis): since a sharp change in the derivative of the curve is expected to be observed at the x-value which likely separates noisy points from other points, the correspondent k-distance is a suitable choice for *Eps* (Figure 10). To set the value of *MinPts* we have followed the general rule:

$$MinPts = k = 2D$$

where D is the dimensionality of data.

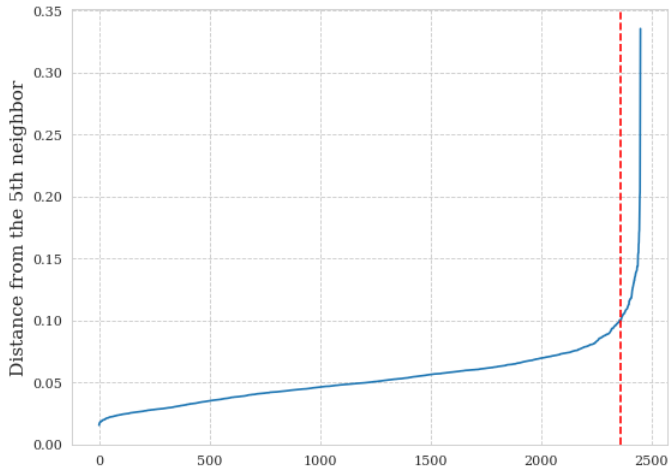
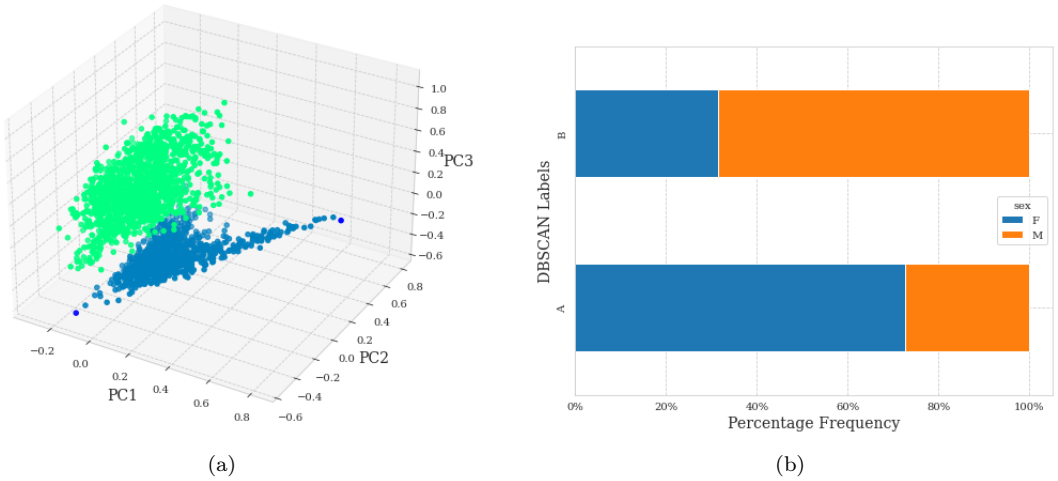


Figure 10: Example of 5-distance plot w.r.t. data points ordered by ascending 5-distance. The red dashed line in the elbow indicates a suitable value for *Eps*.

Globally DBSCAN has worked poorly in high dimensional data, even employing a more suitable distance measure (e.g. max norm). No relevant results have been found by reducing the input space through PCA or a filter approach as the one already used for K-Means. Differently from K-means, outliers' elimination has proved (as expected) not to be influent, and we have also noticed significant differences in the partitioning of data by using full-normalized data, partially-normalized data or non-normalized data. However, only few relevant clusterings have been found (as evidenced in Table 3).

Table 3: Best clusterings found with DBSCAN

N	Subset	*Eps*	*MinPts*	Silhouette	Best Purity
1	*stft_mean*, *stft_min*, *stft_std*, *stft_skew*, *stft_kur*	0.14	10	0.41	0.70 (*sex*)
2	*sc_mean*, *sc_std*, *sc_min*, *sc_max*, *sc_skew*, *sc_kur*	0.15	12	0.32	0.70 (*sex*)



(a) (b)

Figure 11: Clusters of Clustering 1 visualized with PCA (a); and Percentage Frequency of *sex* w.r.t. found clusterings (b).

4.3 Hierarchical Clustering

In the implementation of this clustering methodology, several methods have been used in the analyzed dataset to calculate the proximity of two clusters (affinity) and the metric, specifically:

- Affinity: Single Link, Complete Link, Group Average and Ward's Method.
- Metrics: Euclidean, Chebyshev, Manhattan and Cosine.

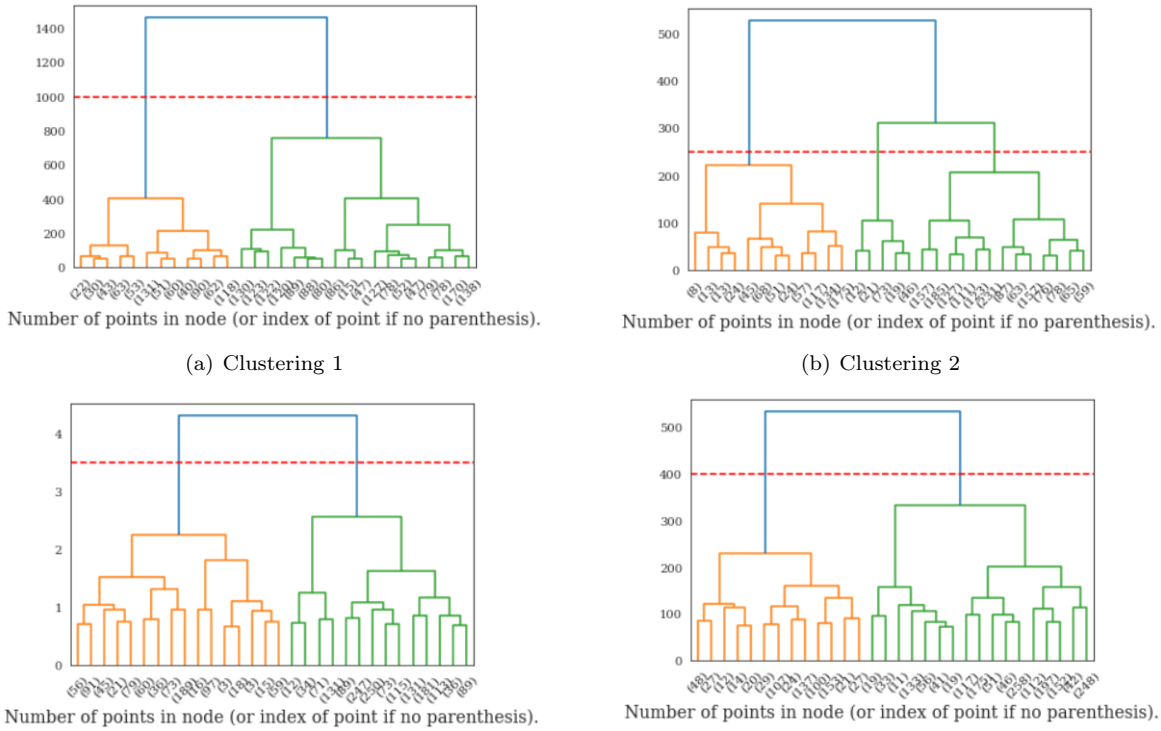
Each affinity has been tested on each metric listed above. To decide on the best clustering, the dendrograms of each individual combination have been plotted to understand the quality of the clustering. Firstly we have considered the entire dataset, finding particularly low Silhouette values. Secondly, we have considered lower-dimensional subsets, finding higher Silhouette values and graphs with more identifiable clusters. A common trend has been noticed: Single Link and Group of Average have given unsatisfactory results independently from the metric and the subset considered, while Complete Link and Ward (compatible only with Euclidean) have always output a more readable dendrogram. As in K-Means, PCA and outliers elimination have proved to improve the silhouette of clusters. The best found results (which share the Euclidean metric) are shown in Table 4.

Table 4: Best clusterings found with Hierarchical

N	Subset	N° of clusters	Affinity	Silhouette	Best Purity
1	*stft_mean*, *stft_min*, *stft_std*, *mfcc_max*, *mfcc_mean*	2	Ward	0.57	0,75 (*sex*)

continue on the next page

N	Subset	N° of clusters	Affinity	Silhouette	Best Purity
2	*max*, *min*, *mfcc_ min*, *std*, *mfcc_ std*	3	Ward	0.51	0,74 (*emotion*)
3	*stft_ mean*, *stft_ std*, *stft_ min*, *stft_ kur*, *stft_ skew*	2	Complete	0.49	0,57 (*sex*)
4	*mfcc_ mean*, *mfcc_ std*, *mfcc_ min*, *mfcc_ max*	2	Complete	0.47	0,70 (*sex*)



(a) Clustering 1

(b) Clustering 2

(c) Clustering 3

(d) Clustering 4

Figure 12: Best results with hierarchical clustering.The number of clusters that returns better values is indicated with the red line.

4.4 Concluding Remarks

Between the three methods considered, DBSCAN has undoubtedly been the worst-performing one, both in terms of Silhouette score and correspondence with known classes. As DBSCAN assumes that a cluster is a dense region of points separated by other dense regions of points by low-density regions, its poor results may be attributed to the structure of data itself, i.e. to a overall homogeneous density. K-Means and Hierarchical Clustering have proved better perfomances – suggesting a more globular and center-based structure of data - with slight but interesting differences: if K-Means' clusterings seem to fit more to the ground truth, Hierarchical clusterings seem to improve better the internal cohesion and external separation of clusters.

5 Classification

This section aims to exploit different supervised classification algorithms (Decision Trees, k-NN, Naïve Bayes) to build well-performing predictive models for the available target classes. In particular, the analysis of the performance of each classifier has been restricted to the binary classification of *vocal_ channel*, *sex* and *emotional_ intensity* as for the other possible target variables (*emotion*, *repetition*, *actor*, *statement*) accuracy values have proved to be systematically unsatisfactory.

5.1 Decision Trees

In order to fit the data avoiding high variance (overfitting) and possible biases due to data partitioning (e.g. with a hold-out), model selection has relied on a double search with repeated stratified 10-Fold cross validation,

consisting of:

1. A first random search (with 200 iterations) over a relatively large hyper-parameter space;
2. A second (exhaustive) search over a more fine-grained grid, i.e. a grid where the tested ranges of values for a given hyper-parameter θ is the neighborhood of the best value of θ according to the first search.

Tested hyper-parameters (which allow for a pre-pruning regularization) and correspondent values for the first search can be found in Table 5.

Table 5: 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[7]	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]

Selected ranges of values in Table 5 are based on empirical evidence: theoretically, Max Depth and Min Samples Split could range in the interval $[2, N]$ and Min Samples Leaf in the interval $[1, N]$ (where N is the number of samples), but it is unlikely that extreme values will perform well on validation data.

As for feature selection, since Decision Trees are able to handle irrelevant attributes – feature selection can be considered as part of the learning process itself – all of the attributes in the dataset have been considered for training the model, included the one-hot-transformed categorical attributes. However, redundant attributes could be accidentally selected (generating unnecessary complexity), so they have been beforehand removed looking at the known correlations (Figure 4).

Model evaluation has been performed with a preliminary hold-out where 30% of data has been separated as test data (proportions of target classes have been observed in the partitioning).

Table 6 shows the most relevant results.

Table 6: Best results of Decision Trees

Target	Criterion	Max Depth	Min Samples Split	Min Samples Leaf	Accuracy
vocal_channel	Entropy	16	0.025	0.01	0.94
sex	Entropy	43	0.012	0.002	0.90
emotional_intensity	Gini	163	0.16	0.009	0.73

Analyzing the trees, binary splits selected by CART are (as expected) consistent with correlations observed in Figure 6. Figure 13 provides an example showing the first three levels of data partitioning performed for the *sex* target.

[7] For Min Samples Split and Min Samples Leaf, the float value x is converted into an integer value with $ceil(x \times N)$, where N is the number of samples.

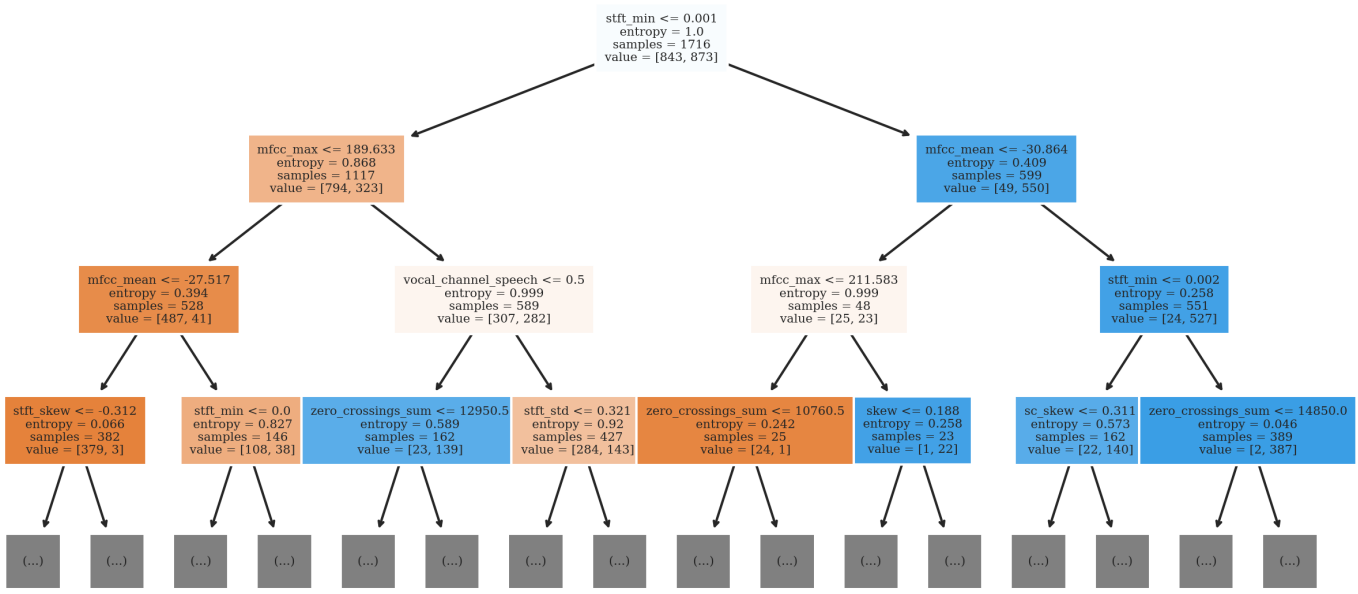


Figure 13: Graphical representation of decision tree for the *sex* target (only the first three levels are displayed).

5.2 K-NN

As in Decision Trees, to avoid overfitting and possible biases due to data partitioning, model selection has relied on the same double search described above. Tested hyper-parameters and correspondent values are shown in Table 7.

Table 7: Tested hyper-parameters for K-NN

Hyper-parameter	Description	Tested Values
Number of neighbors	Value of K	Discrete interval [2, N / 2][8]
Weights	Weight function used in prediction	Uniform, Distance
Metric	Distance measure	City-Block, Euclidean, Cosine, Chebyshev

Also for K-NN, the selected range of values for the number of neighbors is based on empirical evidence – theoretically the number of neighbors can range in the interval [1, N - 1].

Differently from Decision Trees and Naïve Bayes, K-NN is sensible to feature selection. Best results have been found by restricting the attribute set to the attributes which display a significant correlation value with the target variable (Figure 6). We have also found an improvement in the accuracy of the classifier by the exclusion of the one-hot-transformed categorical attributes.

Model evaluation has followed the same criteria described in the previous section, i.e. a preliminary hold-out with 30% of data as test data.

Table 8 shows the best results.

Table 8: Best results of K-NN

Target	Subset	N° of neighbors	Weights	Metrics	Accuracy
vocal_channel	*length_ms*, *frame_count*, *kur*, *sc_skew*, *sc_mean*	45	Distance	City-Block	0.94
sex	*stft_std*, *stft_min*, *stft_mean*, *mfcc_max*, *mfcc_mean*	12	Distance	City-Block	0.91

continue on the next page

[8]Where N indicates again the number of samples in training data.

Target	Subset	N° of neighbors	Weights	Metric	Accuracy
emotional_intensity	*mfcc_min*, *min*, *mfcc_std*, *max*, *std*	456	Distance	Euclidean	0.70

5.3 Naïve Bayes

As Decision Trees, Naïve Bayes Classifiers can handle irrelevant attributes and don't require a previous feature selection. However, the presence of many continuous attributes requires a strategy to estimate the class-conditional probabilities. The available approaches are:

1. Assuming that each continuous attribute is normally distributed (Gaussian Naïve Bayes);
2. Discretizing each continuous attribute (mapping it onto a ordinal one) and extending the standard frequentist estimation of probabilities (Categorical Naïve Bayes).

Since the Gaussian Naïve Bayes classifier makes an explicit parametric assumption (2), we should expect adequate results by using full-normalized data. Another possibility is offered by tuning the hyper-parameter shown in Table 9, e.g. with a repeated stratified 10-Fold cross validation.

Table 9: Tested hyper-parameters for Gaussian Naïve Bayes

Hyper-parameter	Description	Tested Values
Variance Smoothing	Portion of the largest variance of all features that is added to variances	Powers of 10 from 10^{0} to 10^{-9}

The Variance Smoothing hyper-parameter determines the amount of smoothing to apply when estimating the variance of each feature: by finding an optimal value we can employ directly the original non-transformed features without the risk of obtaining poor performances.

Table 10 shows the most relevant results.

Table 10: Best results of Gaussian Naïve Bayes

Target	Variance Smoothing	Accuracy
vocal_channel	10^{-8}	0.93
sex	10^{-9}	0.88
emotional_intensity	10^{-9}	0.71

For training the Categorical Naïve Bayes Classifier, continuous attributes have binned with a quartile-based discretization. To improve its performance, a different hyper-parameter has been tuned with a 10-Fold cross validation (Table 11).

Table 11: Tested hyper-parameters for Categorical Naïve Bayes

Hyper-parameter	Description	Tested Values
Alpha	Laplace smoothing parameter	Powers of 10 from 10^{-3} to 10^{1}

The Alpha hyper-parameter is used to implement the Laplacian m-estimation of class-conditional probabilities and prevent the vanishing of posterior probability for the possible presence of null class probabilities.

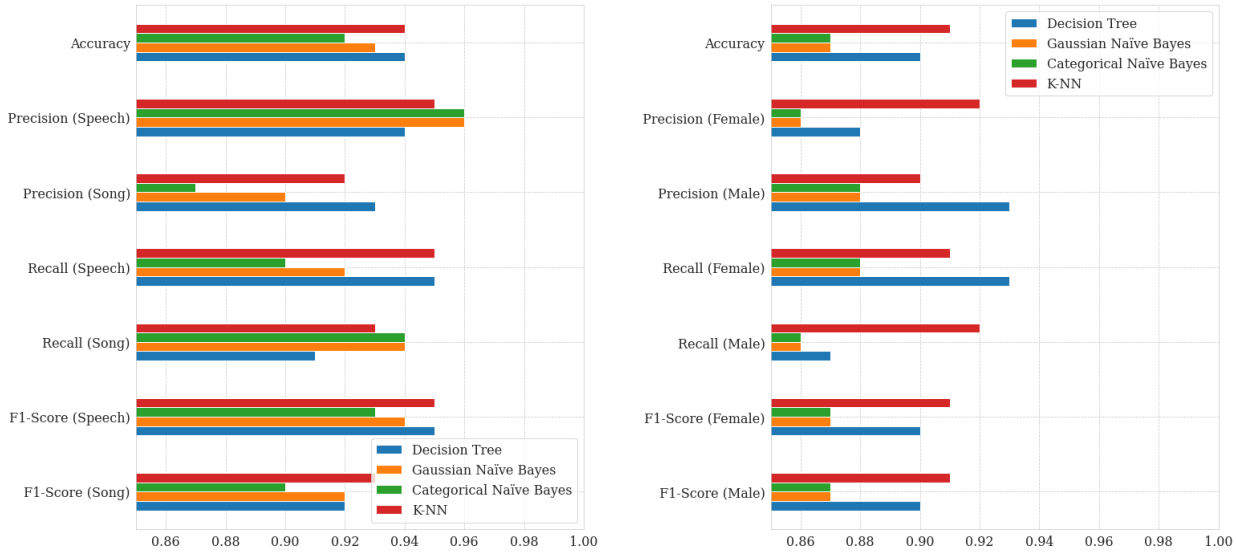
Best results of Categorical Naïve Bayes are displayed in Table 12.

Table 12: Best results of Categorical Naïve Bayes

Target	Alpha	Accuracy
vocal_channel	10^{-3}	0.92
sex	10^{-3}	0.87
emotional_intensity	10^{-3}	0.71

5.4 Concluding Evaluations

The following concluding evaluations about the relative performance of each classifier will be restricted to the targets *vocal_channel* and *sex* for their reasonably large accuracy scores. Figure 14 summarizes the main performance metrics for each target and classifier.



(a) *Vocal Channel* Classification Report

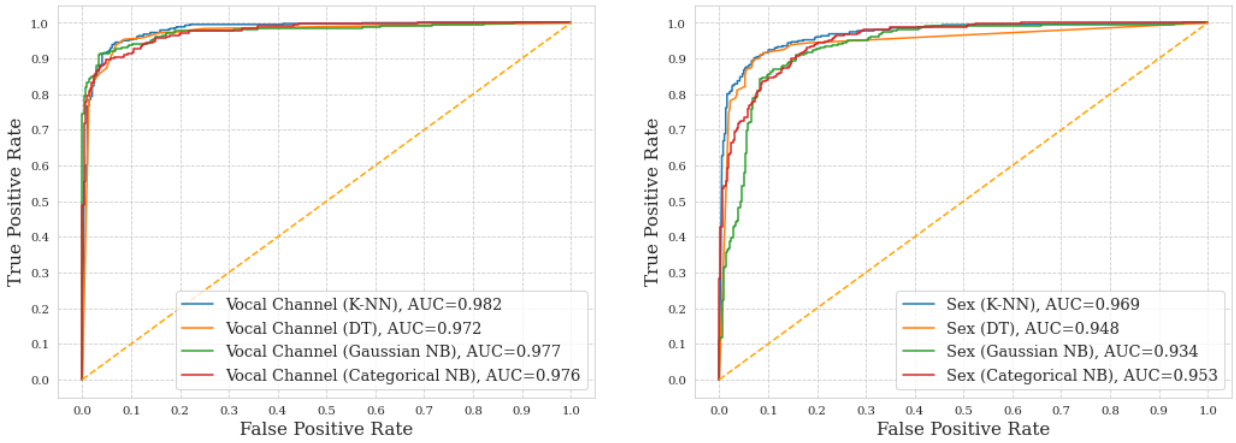
(b) *Sex* Classification Report

Figure 14: Classification Reports

As evident from the plot, all the classifiers have reached good performances in the classification of *vocal_channel* with little differences. Naïve Bayes classifiers have behaved similarly – slightly worse results can be observed in the Categorical Naïve Bayes. Despite better precision on the *speech* class and better recall on the *song* class, the overall predictive capability of Naïve Bayes Classifiers seems worse if compared to the other models. Between Decision Tree and K-NN, the latter seems to be preferable for better F-1 score and sensitivity, despite a lower precision on the *song* class.

Looking at the classification of *sex* we observe a similar scenario: there are no visible differences between Gaussian and Categorical Naïve Bayes; and the best models are again Decision Tree and K-NN, with K-NN performing better on most of the metrics.

The advantage of K-NN over Decision Tree is further confirmed by the AUC values displayed in Figure 15 with the relative ROC curves.



(a) *Vocal Channel* ROC Curves

(b) *Sex* ROC Curves

Figure 15: ROC Curves

6 Pattern Mining

This section aims to study frequent patterns and association rules with the purpose of using one of the most interesting extracted rules for predicting a target variable. Before the pattern mining analysis, data has been subjected to a pre-processing phase that included: a quantile-based discretization of continuous attributes; conversion into transaction data.

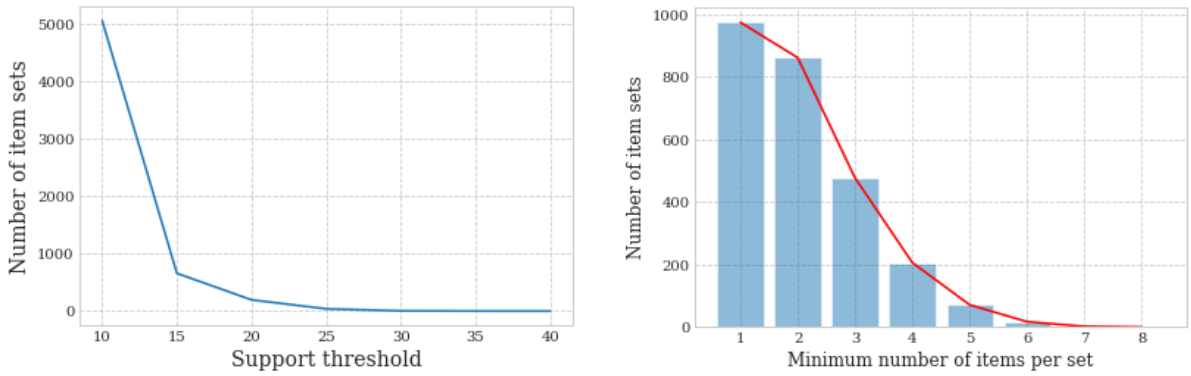
For both frequent pattern extraction and association rule mining, both Apriori and FP-Growth have been used – although no particular efficiency gains have been observed by exploiting the latter for the relatively small size of data.

6.1 Frequent Pattern extraction

For frequent pattern extraction, we have examined either frequent itemsets, closed itemsets and maximal itemsets. For each of these type of itemsets, we have analyzed the number of itemsets with different values of support and *zmin* – i.e. the minimum number of items that should be in a itemset for it to be considered frequent.

Regarding the support values, they have been tested in the range [5, 40] since the maximum support value found is 42.53. On the other hand, for zmin the range tested is [2, 8] as 8 is the maximum width of any itemset.

The results of frequent itemset extraction are shown in Figure 16.



(a) Variation of number of itemsets w.r.t. support

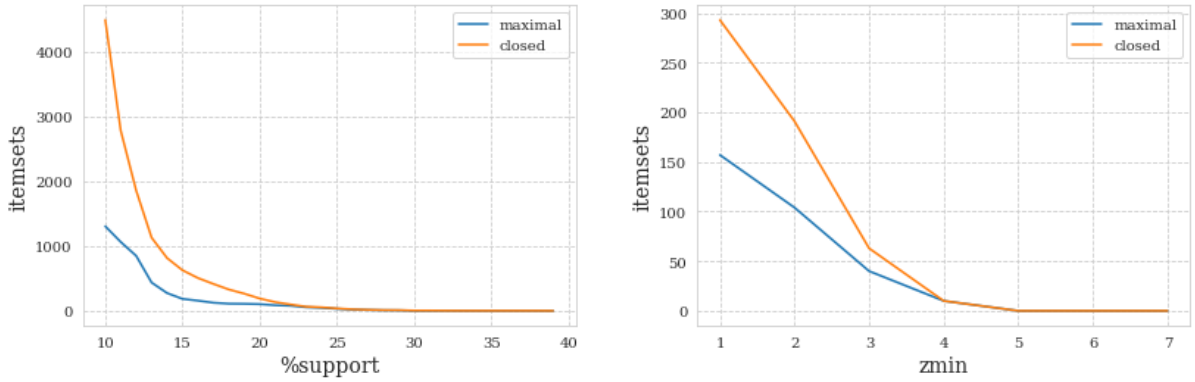
(b) Variation of number of itemsets w.r.t zmin value

Figure 16: Results of frequent itemsets extraction

Analyzing the graphs in Figure 16, considering the different values of support and zmins, we can observe:

- In the first graph there is a clear «elbow» near the support value of 15. The elbow point represents a balance between a large number of sets with low support and a small number of sets with high support. This just described may be a reason for using this support threshold.
- In the second graph it can be seen that moving from zmin = 2 to zmin = 3, a number of itemsets of about 40% are discarded by the algorithm, so a very large number of records would be lost if 3 was chosen as the value of zmin. This may be a reason for selecting 2 as the value of zmin.

Figure 17 shows the results of maximal and closed itemsets extraction.



(a) Variation of number of itemset w.r.t. support

(b) Variation of number of itemset w.r.t. zmin value

Figure 17: Results of maximal and closed itemsets extraction

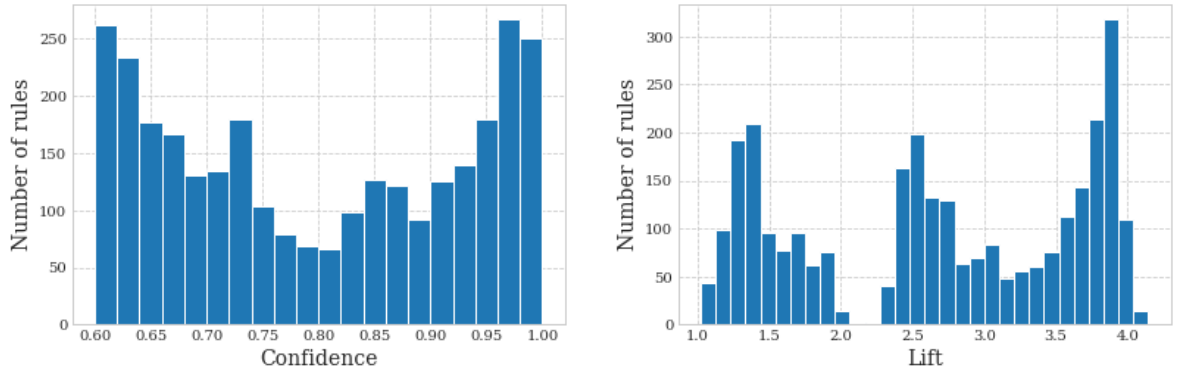
It can therefore be deduced that:

- As for the level of support, in this case we do not observe a clear elbow in the graph, but still there is a downturn in the trend in the vicinity of 15.
- For the value of zmin, on the other hand, a greater than 50 % decrease between zmin values is noted in the interval [2,3]. Also here, therefore, 2 is a reasonable value for zmin.

6.2 Association Rules extraction

Regarding association rules extraction, the parameter values used in the algorithm are respectively those previously found in frequent pattern extraction (15 for support and 2 for zmin), and the range [60, 100] for the confidence value. A lower confidence value has not been selected in order to maximize the quality of the rules to be found. Alongside confidence values, also lift has been assessed to evaluate the interestingness of each highly-confident rule.

Figure 18 shows the distributions of extracted rules w.r.t. confidence and w.r.t. lift.



(a) Variation of number of rules w.r.t. confidence

(b) Variation of number of rules w.r.t. Lift

Figure 18: Result of the association rules study

Observing the graphs, the following conclusions can be drawn:

- The first graph shows bimodal distribution with two peaks at 0.60 and 0.95.
- The second graph shows a trimodal distribution. Interestingly, there are no rules with lift < 1, so it is possible to say that consequent and antecedent appear in the same transaction not by chance, but by correlation.

6.3 Classification with Association Rules

In this final section, we will use the previously evaluations to identify and apply one of the best rules for predicting a target variable.

The selected rule and its values are shown in Table 13.

Table 13: Selected rule

Consequent	Antecedent	%_Support	Confidence	Lift
intensity: (-63.86;-43.41)	*mfcc_ min*: (-1085.48;-826.46) *std*: (-0.0003;0.0067)	21.445221	0.994595	4.102703

As usual, model assessment has been done leaving 30% of the data as test data. True labels have been detected in test data by collecting the target values associated with the antecedent of the rule. It should be noticed that predictions – the consequent of the rule per each true label - refer only to one value of the target attribute. For this reason the confusion matrix shown in Figure 19 does not compute any False Positives or True Negatives. Also, since this is a multi-class evaluation, all the values different from the positive class (i.e. the consequent of the rule) have been merged together to form the negative class.

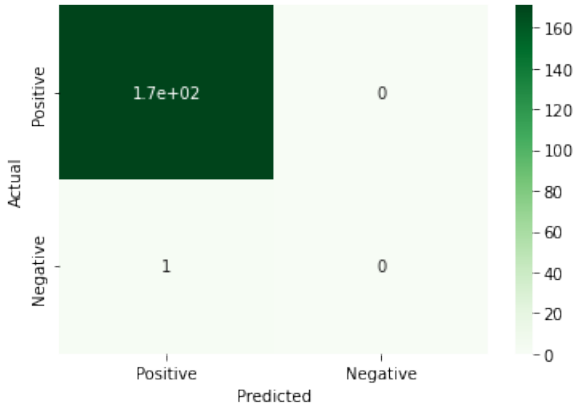


Figure 19: Confusion Matrix

As expected, the rule has reached excellent performance values: 0.99 of Accuracy, 0.99 of Precision, 1 of both Recall and F-1 Score.

7 Regression

In this final section we analyze the performance of different regression models for predicting the output of a continuous target attribute from a selected set of explanatory features. Specifically, the regression models that will be taken into consideration are Ordinary Linear Regression (OLR), Ridge, Lasso, Decision Tree and K-NN.

In order to better appreciate the different behaviour of each regressor, we will consider two multivariate regressions:

1. A linear regression task with target *intensity* and explanatory attributes *mfcc_ min* and *mfcc_ std*;
2. A non-linear regression task with target *std* and explanatory attributes *intensity*, *mfcc_ std*, *mfcc_ min*.

Since the objective is finding optimal performing models, explanatory attributes selection for each regression task has been based on the Spearman Correlation values seen in Figure 4. Figure 20 and 21 display the correlations between each target and its explanatory attributes.

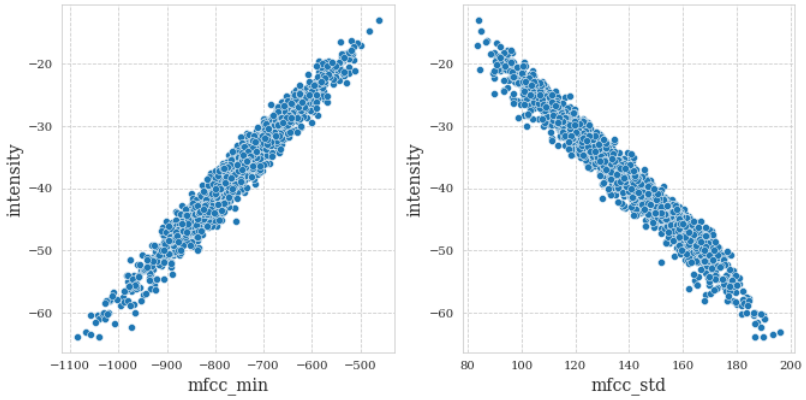


Figure 20: Target *intensity* w.r.t. selected explanatory attributes

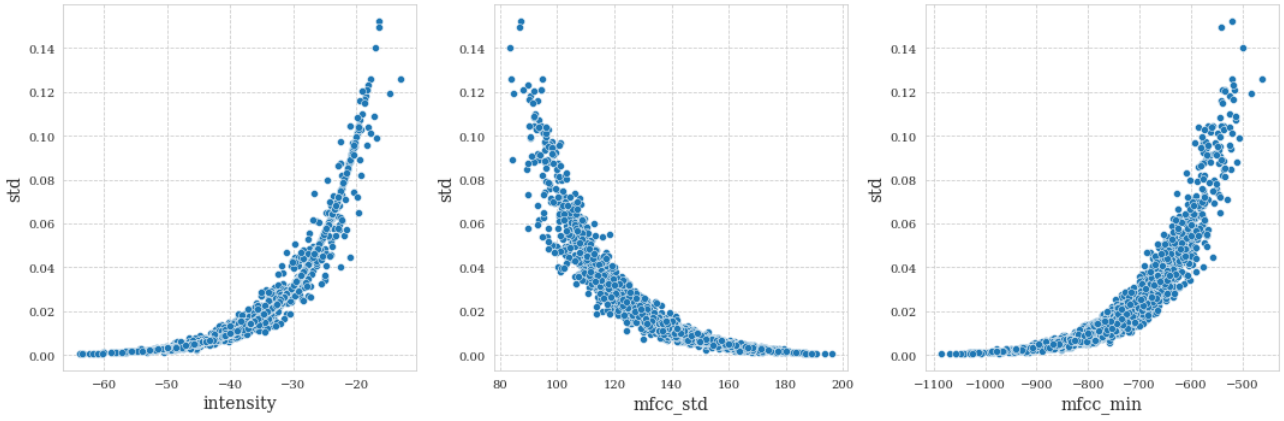


Figure 21: Target *std* w.r.t. selected explanatory attributes

Model evaluation has been performed as usual with a hold-out with 30% of data as test data. Hyper-parameters of each regressor (where present) have been selected with a proper model selection procedure. The *Alpha* regularization parameter of Ridge and Lasso has been selected with a 10-Fold cross validation (tested values are powers of 10 from 10^{-4} to 10^{-1}). For the larger hyper-parameter spaces of Decision Tree and K-NN we have exploited again a randomized search with 10-Fold cross validation. For more details, the reader can refer to Tables 5 and 7 – the only difference concerns Decision Tree, where the values for *Criterion* are replaced with Squared Error, Friedman MSE, Absolute Error and Poisson.

Tables 14 and 15 summarize the configuration of each model and relative performance metrics for each regression task.

Table 14: Regression results for *intensity* target

Model	Selected hyper-parameters	MSE	MAE	R^2
OLR		1746	1010	0.974
Ridge	*Alpha*: 10^{-4}	1746	1010	0.974
Lasso	*Alpha*: 10^{-4}	1746	1010	0.974
Decision Tree	*Criterion*: Absolute Error *Max Depth*: 66 *Min Samples Split*: 0.026 *Min Samples Leaf*: 0.013	2319	1145	0.966
K-NN	*Number of Neighbors*: 22 *Weights*: Uniform *Metric*: City-Block	1968	1077	0.971

Table 15: Regression results for *std* target

Model	Selected hyper-parameters	MSE	MAE	R^2
OLR		19388	3452	0.716
Ridge	*Alpha*: 10^{-4}	19388	3452	0.716
Lasso	*Alpha*: 10^{-4}	19388	3452	0.716
Decision Tree	*Criterion*: Squared Error *Max Depth*: 126 *Min Samples Split*: 0.023 *Min Samples Leaf*: 0.012	1.485	0.748	0.978
K-NN	*Number of Neighbors*: 29 *Weights*: Distance *Metric*: Euclidean	0.003	0.030	0.902

We can first notice that in both cases the *Alpha* parameter of Ridge and Lasso tends to zero, thus minimizing the contribution of the penalty term and making the two models approximately equivalent to the non-regularized OLR. As expected, OLR is not only sufficient, but also optimal for solving the linear problem – despite non-linear models perform equally well. A wider gap can be observed in the non-linear task, where Decision Tree and K-NN reach definitely better performances.

References

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.