University of Pisa
Department of Computer Science
M.Sc. in Data Science and Business Informatics

Data Mining I project report

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

Authors

Pierfrancesco Benincasa

Francesco Giacomo Curcio

Niccolò Seghieri

Tutors

Prof. Dino Pedreschi

Prof. Riccardo Guidotti

Dr. Francesco Spinnato

December 30, 2022

Contents

List of Figures

Introduction

The RAVDESS dataset contains audio of 24 professional actors (12 female, 12 male), vocalizing two statements in a neutral North American accent. Speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

The initial one contains 2452 rows and 38 columns, in which data are divided in two different data types: **numerical** (integers or floats) and **categorical** (describing objects with different attributes). In our analysis the first step is represented by a part of describing each variable and then of an epuration and management of the incomplete/null values that can affect the applications of methodologies and tools. The reason is simple: to further a justification for each choice, is better to know what variables singularly are with their definition.

1 Data understanding & preparation

1.1 Data semantics

In the following sentences there's a brief description of each column in the dataset:

- **modality** *(object)*: represents in our case only vocal audio;
- **vocal channel** *(object)*: represents binary speech or song;
- **emotion** *(object)* explains of what kind of tone is the data (speech includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, and song contains calm, happy, sad, angry, and fearful emotions);
- **emotional intensity** *(object)* is the same for both of vocal channel and also in this case the description is above;
- **statement** *(object)* showed the two in the dataset;
- **repetition** *(object)* for each statement stands for statement times reproduced;
- **actor** *(float)* indicates the number referred to the specific actor (each one if enumerated with a specific);
- **sex** *(object)* refers to male/female binary;
- **channels** *(object)*: 1 for mono and 2 for stereo audio;
- **sample width** *(int)* as number of bytes per sample (1 means 8-bit, 2 means 16-bit);

- **frame rate** *(int)* frequency of samples used (in Hertz);
- **frame width** *(int)* is the number of bytes for each frame (one frame contains a sample for each channel);
- **length ms** *(int)* audio file length (in milliseconds);
- **frame count** *(float)* the number of frames from the sample;
- **intensity** *(float)* loudness in dBFS (dB relative to the maximum possible loudness);
- **zero crossings sum** *(int)* like the sum of the zero-crossing rate (generally the zero-crossing rate is the rate at which a signal changes from positive to zero to negative or vice versa and its value has been widely used in both speech recognition and music information retrieval, being a key feature to classify percussive sounds).

There are also other attributes (and for each one of the following are included columns with mean, standard deviation, minimum, maximum and, if necessary, skewness and Kurtosis) which refer to the signal processing or represent physical attributes: the **Mel-Frequency Cepstrum (MFC)**, **the short-term Fourier transformation** and **the spectrum centroid**. All of these variables are float. In particular and in a more general sense:

- **The Mel-Frequency Cepstrum (MFC)** is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency [1].
- **The Spectral Centroid** is a measure used in digital signal processing to characterise a spectrum. It indicates where the center of mass of the spectrum is located. Perceptually, it has a robust connection with the impression of brightness of a sound. It is sometimes called center of spectral mass [2].
- The **Short-Time Fourier transform (STFT)** is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment [3].

[1] https://en.wikipedia.org/wiki/Mel-frequency$_c epstrum$
[2] https://en.wikipedia.org/wiki/Spectral$_c entroid$
[3] https://en.wikipedia.org/wiki/Short-time$_F ourier_t ransform$

1.2 Statistical analysis

In this section will be presented many graphical aspect of the most important and comprehensible features. For example, in the following figure there's a pairplot of some statistical sound characteristics qualified by vocal channel.

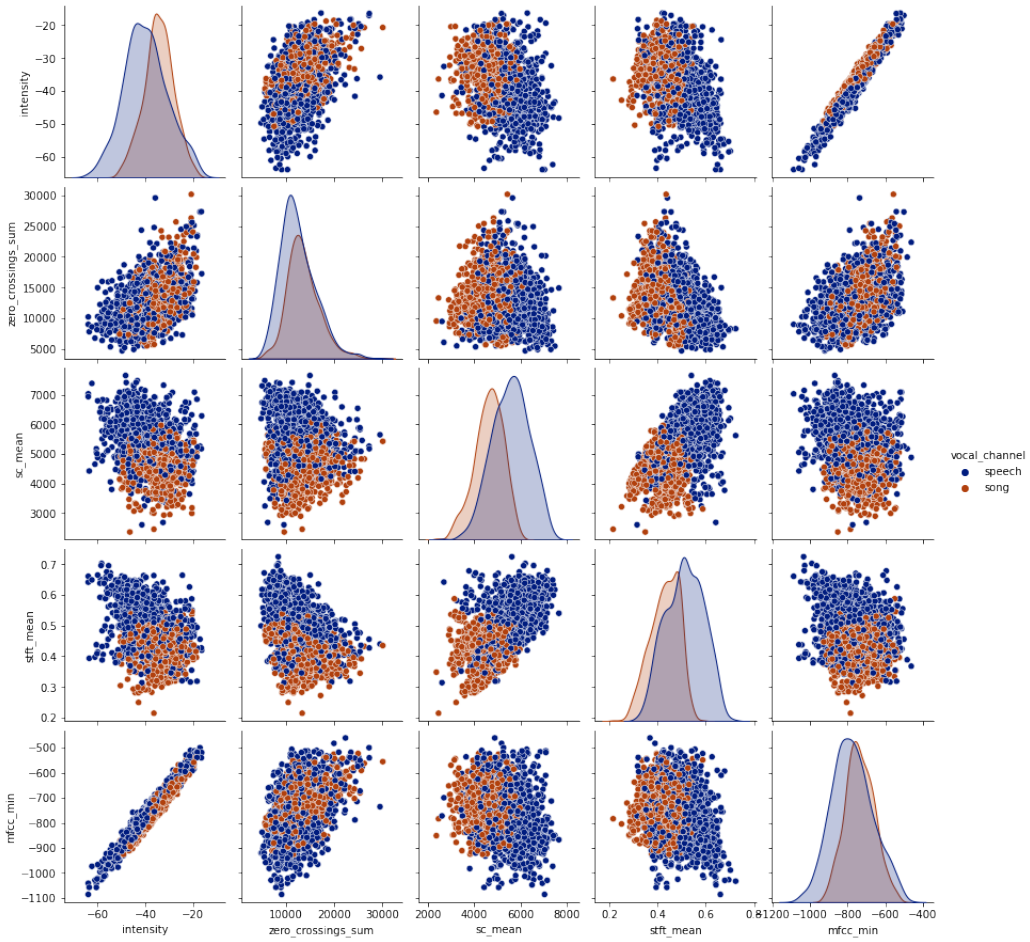


Figure 1: Pairplot conditioned using vocal channel.

Starting from the main diagonal it's possible to observe some overlapped distributions and it's noticeable that there are no differences related to the two vocal channels. For the other pictures two patterns are in evidence and each one refers to a different channel: that's the reason why this representation is helpful to understand how the dataset is composed. The only fragmentation is represented in the pair made up by mfcc-min and intensity because data are not well distributed in the space (sparse) as in other cases; this is an evidence of a strong positive correlation.

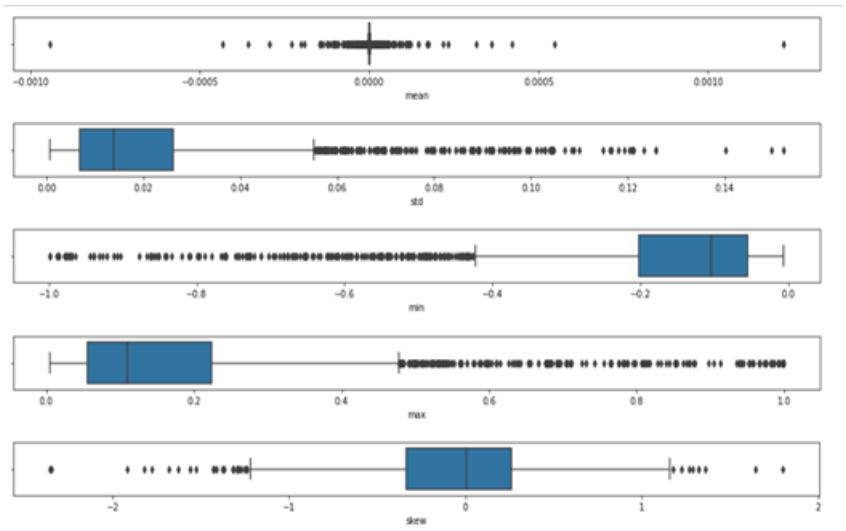


Figure 2: Boxplot of statistical indexes about audio signal.

This picture is showing audio signals' statistics: for example in the last box the information is that all the values out of it show a different kind of skewness (strong/moderate and positive/negative). In the case of strong skewness all the values are over the range [-1;1]. Moderately skewed values are instead inside [-1;-0,5] or [0,5;1]. Mean has 'no box' due to the fact that the difference for each values is really close to the zero.

In the following picture from top left to bottom right are presented the distribution of emotions: fearful, neutral, happy, angry, sad, disgusts, surprised, calm; all conditioned by sex (female groups in orange, male in the other one).

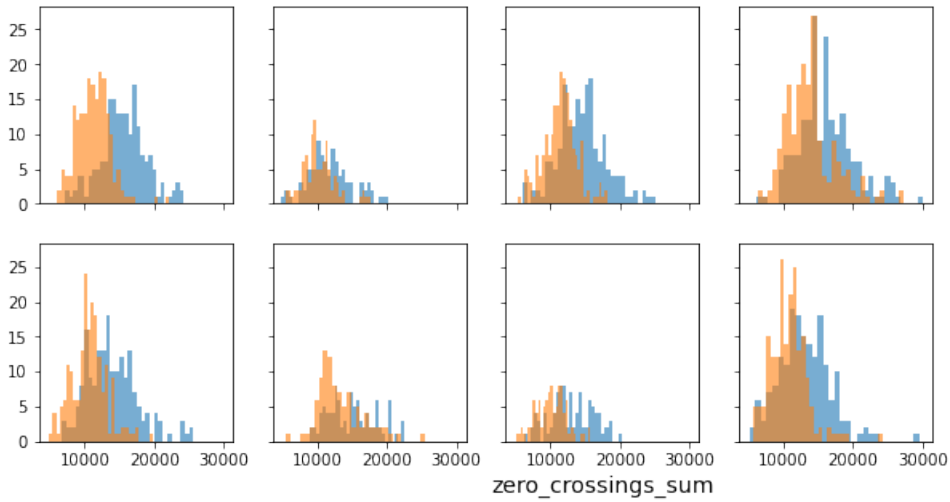


Figure 3: Zero crossings related to emotion and divided by sex.

Another interesting visualization of data involves, in fact, *zero crossing sum* for each emotion and divided per sex. The reason of this choice is related to the signal and its evolution: for different emotion it's reasonable to assume that the number of zero crossing change. To different emotion it's possible to think different peaks and structure in the signal. Results could be also affected by the tone of the voice, and this is an intrinsic characteristic of the natural difference between the two sex's voice.

Something interesting was also found in the discrimination per vocal channel of length ms divided by sex. The evidence is that audio tracks aren't affected by differences between the two genders but from the nature of the vocal channel: song has clearly a mean length above the one of speech (4650ms against 3700ms).

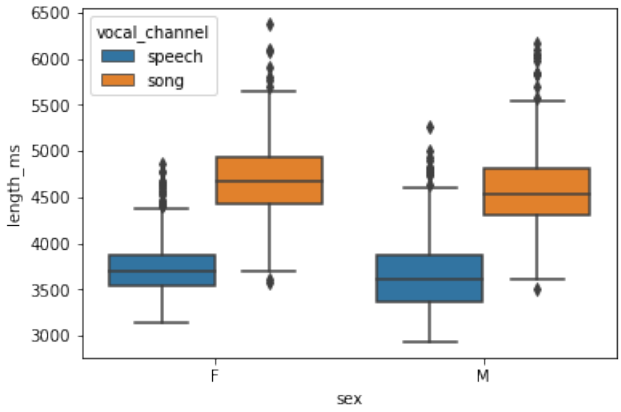
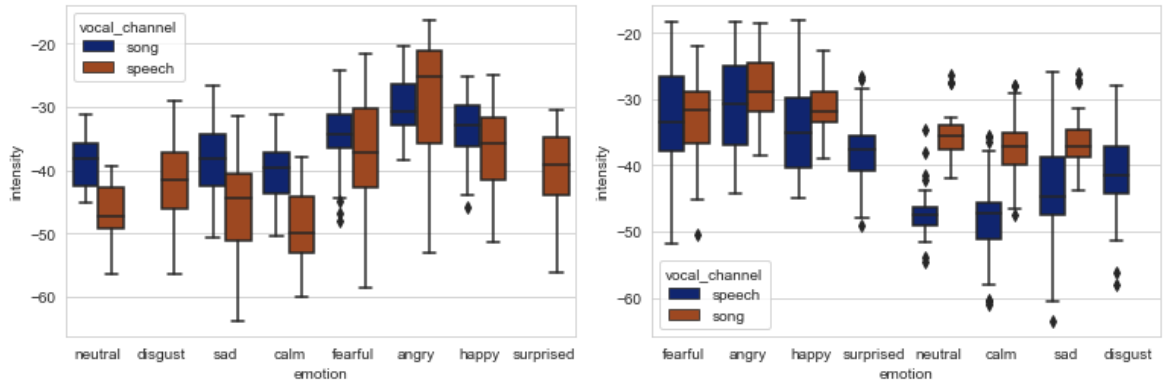


Figure 4: Length ms for speech and song divided by sex.

Using always the vocal channel and sex, it's interesting to see the differences in values using emotions in order to have a perception about outliers. Next figures will express the concept.

1.3 Data preparation

In our dataset there are some columns in which data are incomplete/with null values. Not maintaining these fields could affect the obtainable result or, worst, making heavier to read and understand its content. Fist involved column is actor. The decision of the team about them is: deleting actor from the analysis because it's a field that doesn't affect other parameters (about $\approx 50\%$ is missing). At this point, modality and repetition (both object data type) are not considered because modality is related to the same kind of tracks (audio-only in this case) and repetition is a parameter that don't involve changes from the point of view of signal processing (once the audio presents some characteristics, it doesn't matter the repetition, otherwise the pattern would have been too complicated to explore and extract some appreciable inferences). Now,



(a) (b)

Figure 5: Females (a) and Males (b) boxplot of emotions

assuming that from an architectural point of view, sample width (1 or 2 in the binary scale of values) can be simplified because there's only a value set as 2 (corresponding to 16-bit) in this case. For a conceptual schema visualization of the problem, in the same way it's possible to evitate the use of stft-max because also in this case there's only a unique value (1), and the use of frame rate as unique value of 48000. framecount, instead, contains the same information of length ms with a conversion factor of 48x. For this reason, it is possible not to considerate it. frame width and channel carry the same information, the second one is dropped for this reason.

From these considerations, subset obtained contain: *categorical* attributes which will be used to categorize and differenciate records and *numerical* attributes that can quantify the existing values without the elimination from the information content. Before any grouping using categorical ones, it's shown the correlation matrix:

To clarify, some attributes are not used because of a low correlations factor with all the others. These are: 'kur','mean','skew','stft-kur','sc-max'.

Anyways, information aboout missing values in this dataset are related only to the following columns: actor (1126 NaN), vocal channel (196 NaN) and intensity (816 NaN). Some of them, as said before, are not considered in the subset. Using a replacement for intensity (using mean conditioned by sex, emotion and statement to be specific), instead, make a change on distribution from uni-modal to bi-modal. For this reason, seen also a high correlation factor with mfcc-min, it's preferrable to use it to make some inferences, without influencing the structure of the possible implications.

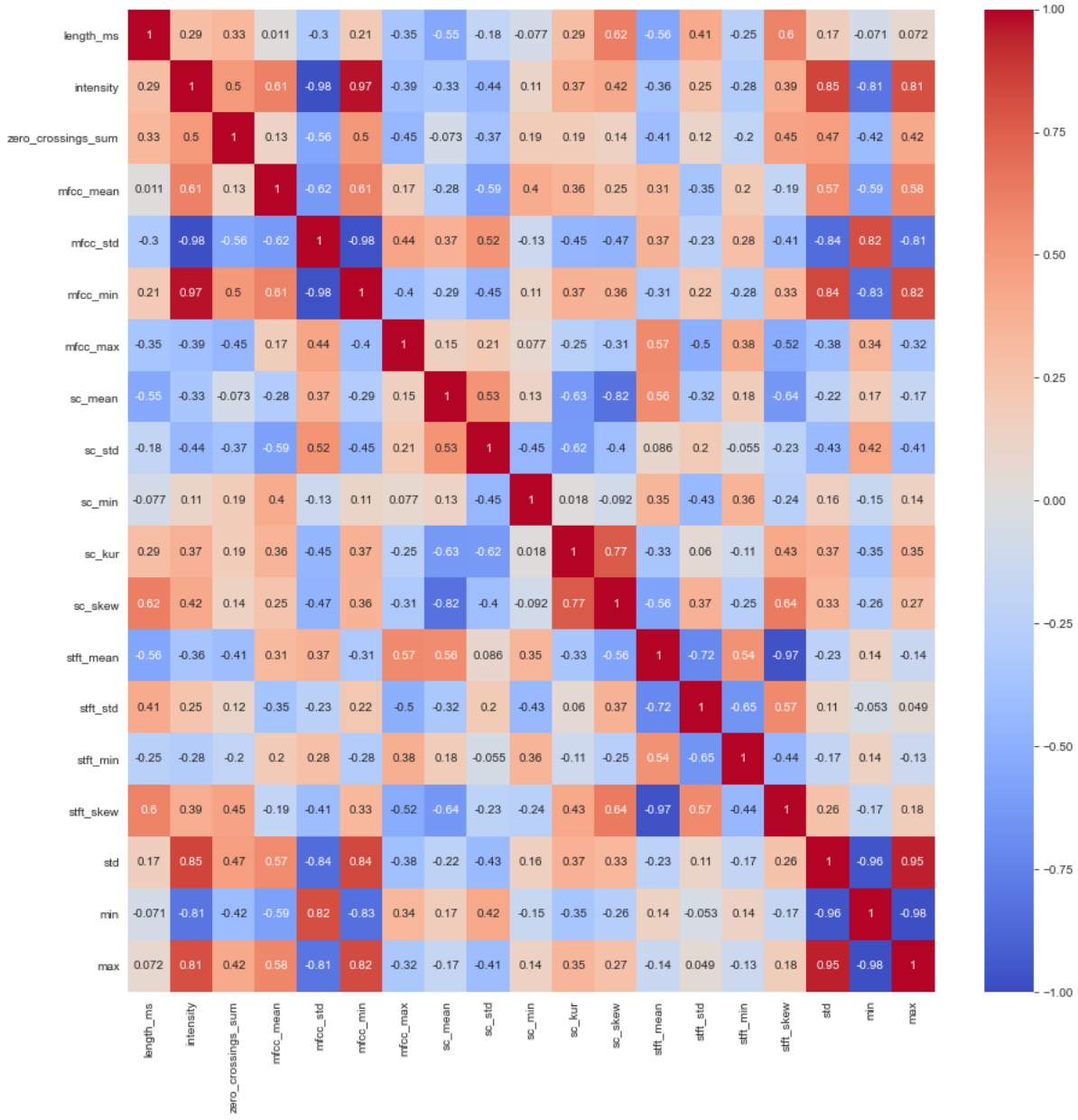
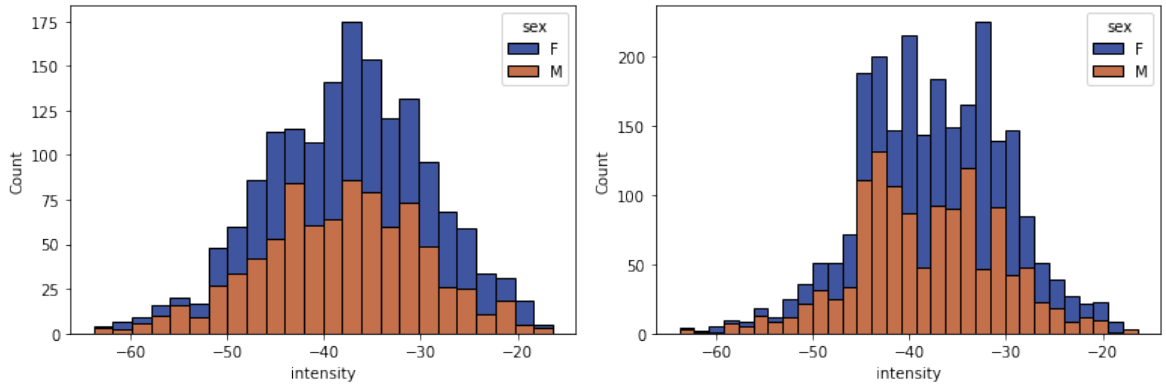


Figure 6: Heatmap of the dataset.

2 Clustering

Continuing with the exploratory analyses, in this section our aim is to identify data aggregates that are as homogeneous as possible among themselves and, at the same time, distant from each other. This requires the use of clustering algorithms, i.e. unsupervised methods that will lead us to a deeper knowledge of the structures present in our dataset.



(a) (b)

Figure 7: Intensity value count divided by sex with NaN (a) and with filled NaN (b).

In particular, the algorithms used to carry out this analysis are: K-means, DBScan and hierarchical. An important premise to mention is: for a correct implementation, it is necessary to standardise the variables as they are based on the concept of distance, and this was done.After a first attempt with all numeric variables left over from the date preparation, we opted to use and report in the document the results obtained with the following attributes: length-ms, zero-crossings-sum, mfcc-min, sc-min, stft-kur, mean, skew, std, kur.

Our choice was made by choosing only one feature for each of the physical statistics of sound taking into consideration the correlation matrix by choosing the least correlated attributes.

2.1 K-means

K-means is an algorithm that divides into k clusters associated with a midpoint called centroid.

One of the interesting and fundamentally important aspects is therefore the optimal choice of the k number of clusters. To make this estimatation, we relied on the representations provided by the following pictures.

They respectively represent the Sum Square Error (SSE), our goal is to minimise it, and the Silohuette, our goal is to maximise it and the values for these indices was computed in the range k [2, 50] There is therefore a certain trade off between the two and, so, our choice was to opt for a number of k clusters equal to 6, following the elbow rule and having a not excessively low silhouette. So values are 190 for SSE and 0,24 for

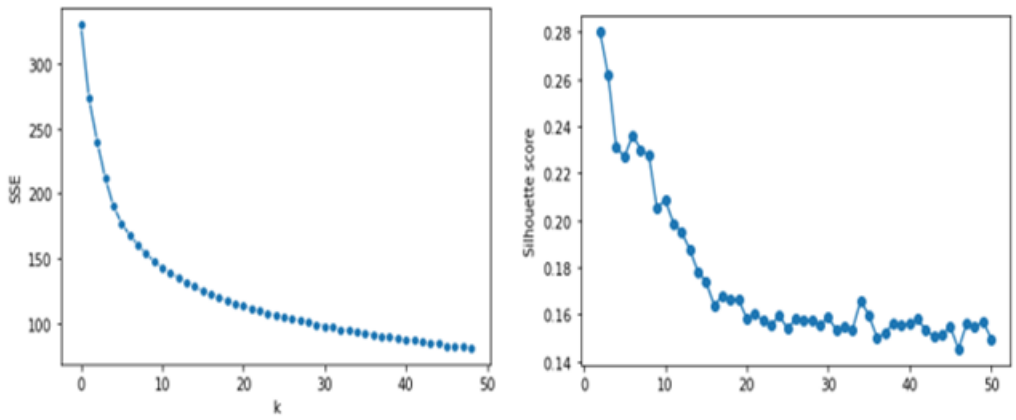


Figure 8: SSE ans Silhouette represented.

Silohuette score. The next image illustrates how clusters appear graphically in relation to the variables mean, std and length ms.

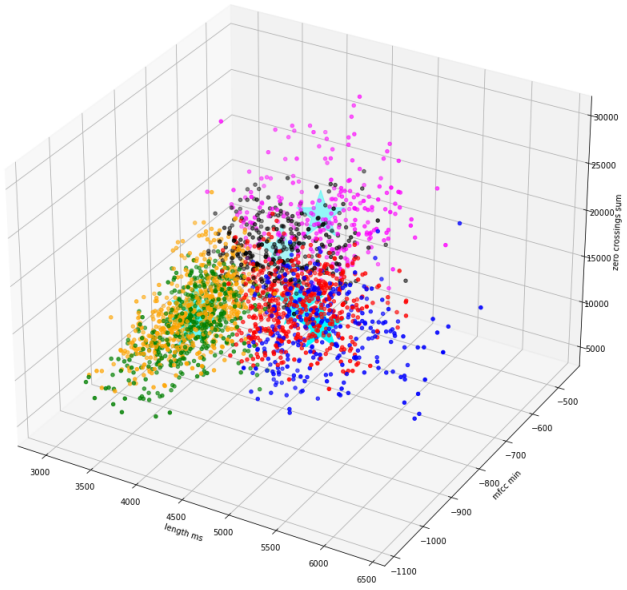


Figure 9: 3d-plot on variables used in K-meas.

Unfortunately, many points cannot be clearly distinguished despite the 3D image. This is probably due to the choice of the numerosity of the clusters and, mainly, the relatively low silhouette score, which suggests a not too high quality of the clusters obtained as they are not well separated. It may therefore be important to visualise

the desired information via countplots in order to understand what divisions have been made regarding emotions and voice channels for each clusters.

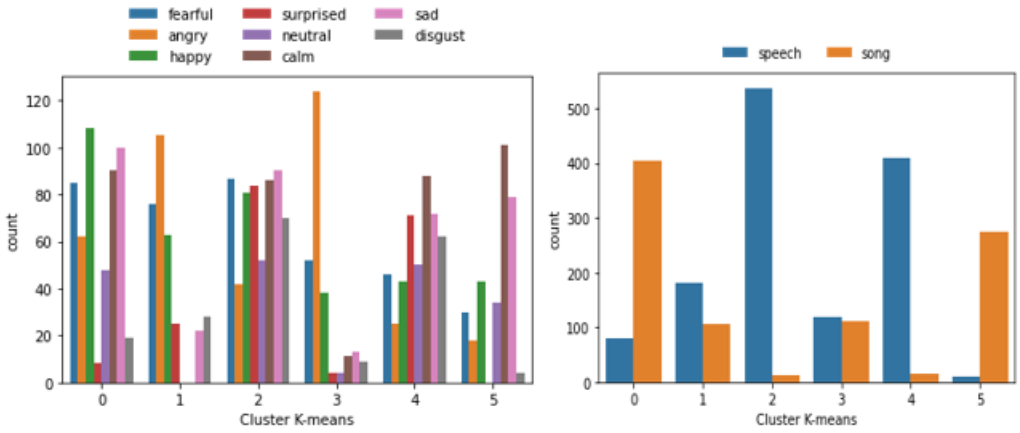


Figure 10: Countplots from K-means.

In the first countplot concerning emotions, on a visual level, clusters 1, 3 and 5 are undoubtedly remarkable, in which we can appreciate peaks for the emotion angry and calm, followed by sad. In the second, however, with the exception of clusters 1 and 3, we can see how there is always one vocal channel dominating over the other.

To offer a numerical impact, a table is presented showing the averages for each of the variables used for each cluster.

Cluster K-means	length_ms	mfcc_min	sc_min	stft_kur	zero_crossings_sum	mean	skew	std	kur
0	4526.873077	-749.730128	892.041803	-1.353129	12981.082692	1.592069e-06	-0.145659	0.018801	7.478323
1	4045.492163	-659.402852	47.351915	-1.289329	14977.749216	2.096589e-06	0.214222	0.037010	12.978748
2	3581.081081	-811.490796	998.401841	-1.165461	11445.060811	-5.562877e-07	-0.243820	0.010003	15.192985
3	4332.039216	-613.605936	1084.364354	-1.153325	18330.035294	-1.559027e-06	0.209124	0.059179	10.407602
4	3631.461707	-834.204398	11.583213	-1.285854	10408.306346	-1.282327e-06	-0.038482	0.007704	13.090036
5	4871.265372	-784.896930	4.109851	-1.208134	12493.508091	-4.842010e-07	-0.007414	0.013463	5.859158

Figure 11: Means related to K-means clustering application.

We can see how the average of length-ms is higher for clusters 0 and 3, which not surprisingly have peaks per song in the countplot, and also how higher values of mfcc-min (proxy for intensity removed from the dataset as explained in section 1.3) and zero crossings sum are found in correspondence with a 'strong' emotion such as angry.

2.2 DBScan

DBScan is an algorithm based on the concept of density (intended as the number of point inside a certain radius) and it requires the setting of two parameters: radius and min-samples. Min samples is the minimum number of points that belong to the region of space described by the radius. The radius itself describe the region of interest. In the start phase of the use of DBScan all the variables were used (transforming categorical ones into numerical).

The following figure is showing a combination of different set obtained with some min-samples useful for fitting DBScan with the use of a for-loop iteration. This way, from an initial distance quadratic matrix, it's possible to obtain the graphical representation below:

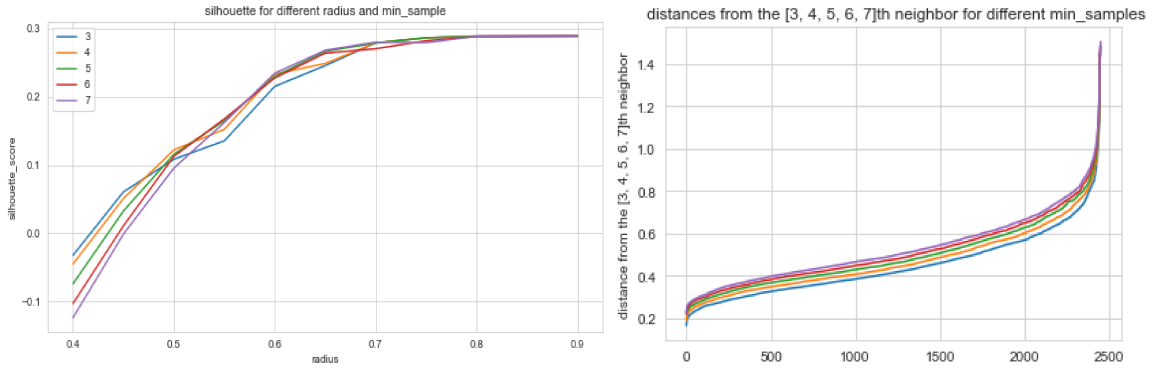


Figure 12: "Silhouette scores for different radius and min-samples" and "k-th nearest neighbor distance".

The left picture represents different silhouette scores.The right picture represents the distance from the k-nearest point to the one selected, the so called *k-th*. The optimal point is corresponds to the elbow of the curve, and every colour represent a different selected k. In this case the parameters are min-samples = 4 and radius = 0,75 (in which most of the curves in the left converges, with the highest degree of silhouette's score).

Cluster results are: {-1:41, 0:162, 1:160, 2:137, 3:160, 4:161, 5:135, 6:136, 7:135, 8:164, 9:139, 10:139, 11:168, 12:144, 13:165, 14:166, 15:140}, outlier cluster is indicated with -1. Silhouette score is calculated as 0,286.

Same procedure was applied to a subset of variables, same ones used for k-means. In this case, results shown:

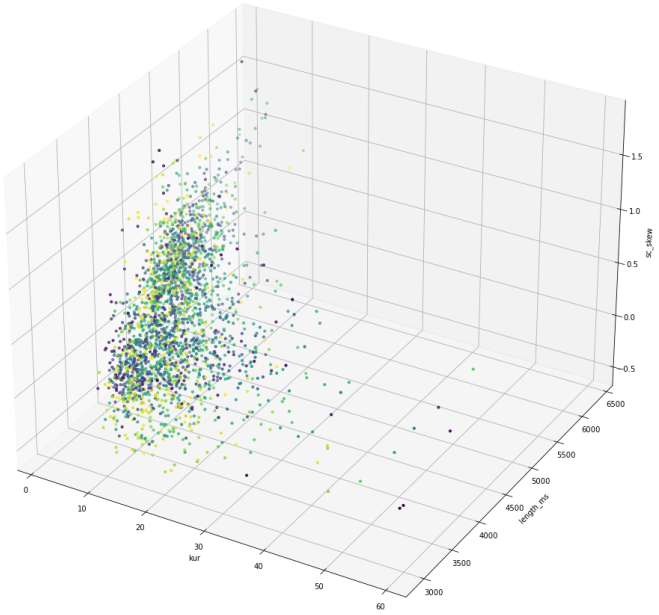
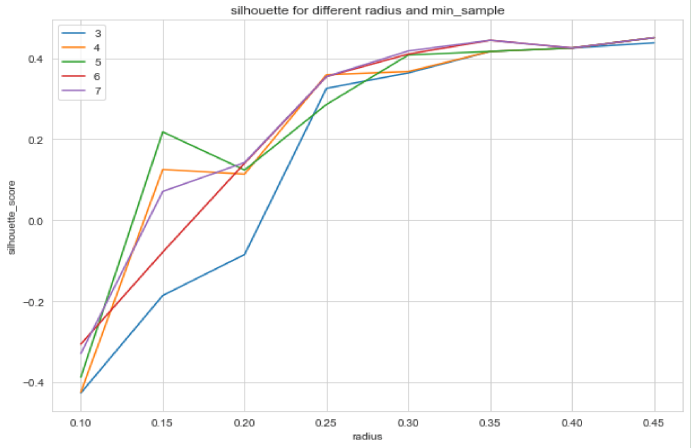


Figure 13: Resulting plot of DBScan.



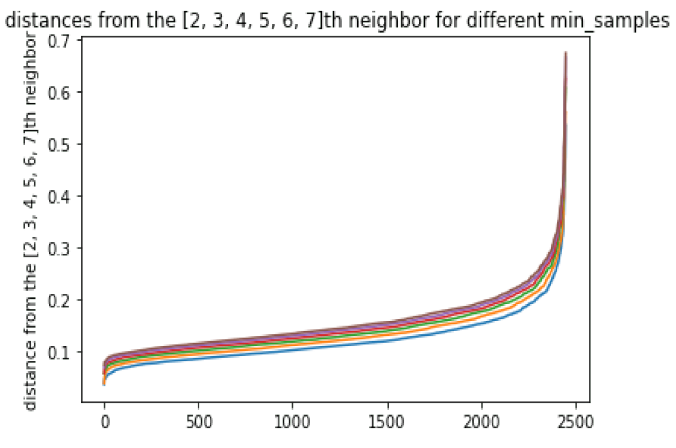


Figure 14: "Silhouette scores for different radius and min-samples" and "k-th nearest neighbor distance" for the subset of variables.

This way, the value of the radius drastically is reduced of 0,5 and the silhouette score has increased to 0,334. min-samples remained the same. Cluster results are: {-1:96, 0:2356}, outliers indicated with -1. Globally, the number of outliers augmented. Everything was aggregated to a unique cluster.

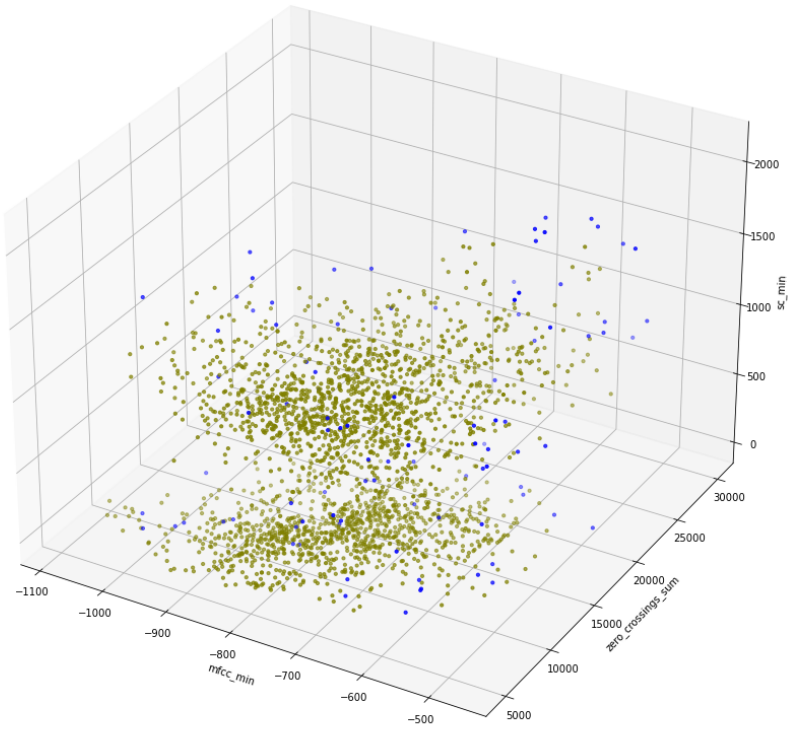


Figure 15: Resulting plot of DBScan applied to the subset of variables.

2.3 Hierachical clustering

The strength of the hierarchical cluster is that it has a strong visual impact using dendograms which, at a certain level of dissimilarity, from a cutting return a certain number of clusters. There are different agglomerates methods based on different types of linkages and ones used are: *single linkage*, *complete linkage*, *average linkage* and *ward linkage*. Variables chosen are always the same and, as a measure of dissimilarity, Euclidean distance is always used. The following figures show typical dendograms:

These are the main results:

- dendogram obtained from *single linkage* is skew to the left, so if we decide to "cut" in such a way as to obtain k =2, as can be seen from the figure, a cluster will consist of a single element. Increasing the number of clusters to three or four gives as many clusters with a number equal to two elements and another with only a element.On the other hand *average linkage's dendogram* is skew to the right but certainly appear a little more balanced than single linkage. In particular, slicing to obtain k = 2 as output will conduct to have one large cluster and another with

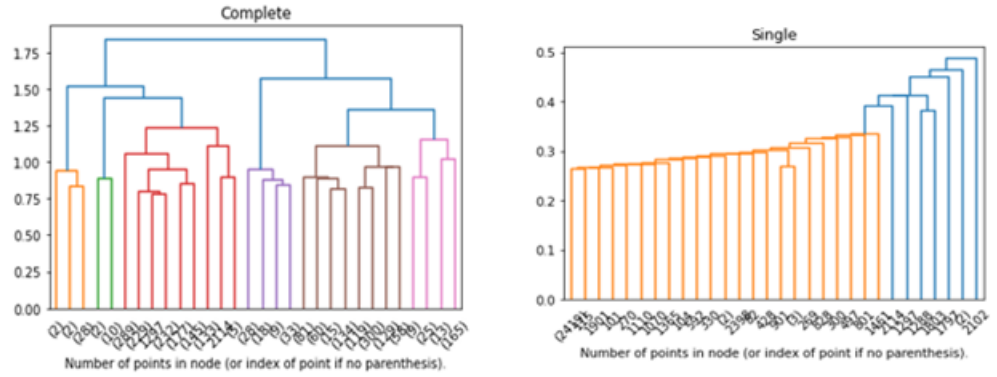


Figure 16: Hierarchical plots of complete and single methods.

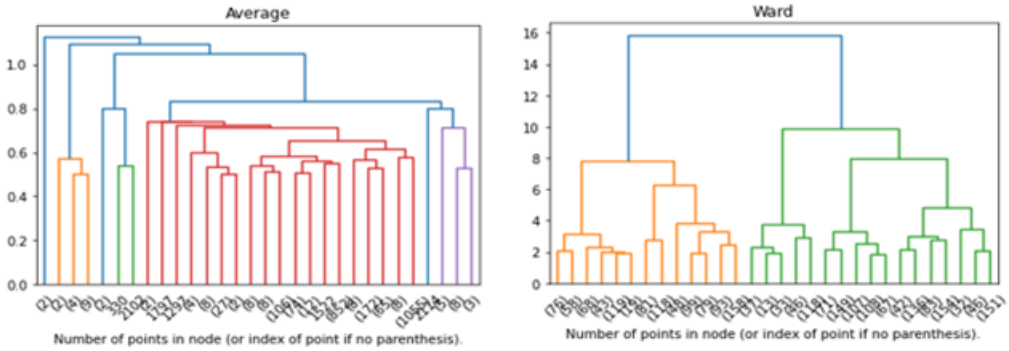


Figure 17: Hierarchical plots of average and ward methods.

two values. Increasing the number of clusters get groups with a few elements flanked by one that contains almost all the records in the dataset.

- dendogram obtained from *complete linkage* is balanced in terms of skewness but the number of cluster seem to show a bigger fragmentation compared to the average (also if it's clearly less fragmented than single linkage).
- *ward linkage dendogram* is the most balanced one. It's possible to have with a single two clusters similar in terms of distribution of data. Silhouette score for this dendogram is the biggest obtainable and it is of 0,28. This is clearly the perfect hierarchical kind of clustering for our dataset.

2.4 Conclusions about clustering

Between the three kind of algorithms applied to our dataset, the worst one in terms of fitting with the nature of the data we have is DBScan and the demonstration of this comes from the over-agglomeration and by the fact that dimensionality in this case is probably too high for best-practice application. K-means, instead, gives information

about the distribution of categorical variables. Hierarchical methods, finally, can just give an hypothetical division about data in natural clusters. Globally, is possible to say that the best compromise in this case is represented by k-means.

3 Classification

In this section, we manage the classification problem using three types of algorithms: decision-tree, KNN and Naive-Bayes classifier. We used vocal_channel and emotions as target variables. The former was chosen because there were 196 missing values in the data set and we considered it appropriate to build a model that could predict the values; therefore, the 196 records were isolated during the construction of the various models so as to implement replacement once results were obtained (in order to select the most satisfactory classifier).

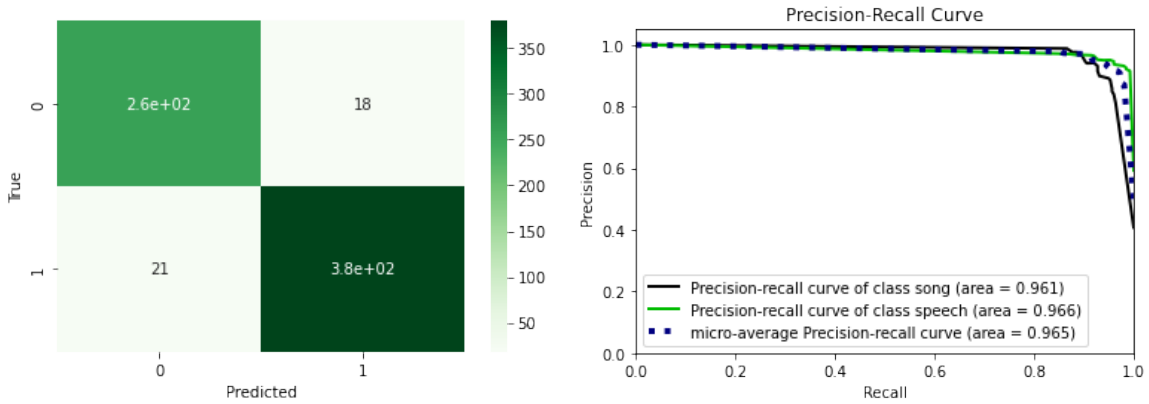
3.1 Decision Tree (vocal_channel)

Before the application of the model, some data pre-processing was done with the get _dummies functionality in order to transform the categorical variables into dummies. Next, the division was made between train and test with 70% of the instances in the train and the remaining in test. In addition, it was opted for stratification (also used in the following algorithms) so as to have the two classes, speech and song, equally represented, despite the fact that there is no excessive imbalance in the set (1335 speech, 921 song). After running the model with the default parameters, it's used for an optimal search RandomizeSearchCV obtaining: split criterion = gini, max depth = None, min samples leaf = 0.003324, min samples split = 0.01555 and the mean of cross-validated score about 93%. Applying the model obtained to the test, the result in terms of accuracy is very similar to that obtained by cross-validation which is about 94%. Below are the results of model's performance on the test, which shows an AUC of 0.968:

It is therefore possible to state that model obtained from the decision tree classifier performs well in general. To conclude, it's observed that the obtained tree as a result of parameter adjustment is less complex, being less deep, and therefore the inherent problem of overfitting, which is very typical of the algorithm used, has been slightly reduced. Below a representation of it:

3.2 Decision Tree (emotions)

Before the application of the model, some data pre-processing was made with the get_dummies functionality in order to transform the categorical variables into dummies. Next, the division between train and test was made with 75% of the instances in



(a) (b)

	precision	recall	f1-score	support
song	0.92	0.93	0.92	276
speech	0.95	0.95	0.95	401
accuracy			0.94	677
macro avg	0.94	0.94	0.94	677
weighted avg	0.94	0.94	0.94	677

(c)

Figure 18: Confusion Matrix, Recall Curve and Prediction results decision tree applied to vocal channel

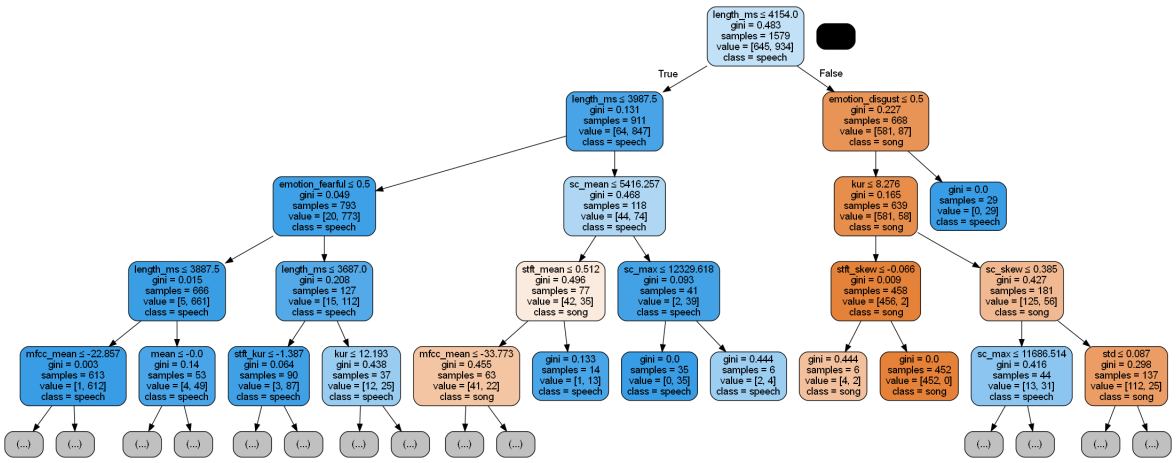
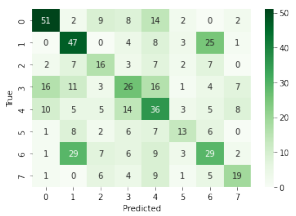


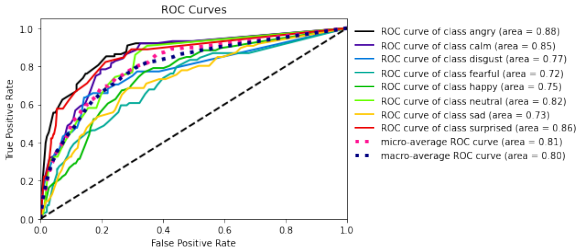
Figure 19: Decision tree for vocal_channel target variable.

the train and the remaining in the test. Running the model with the default parameters, we immediately observe that there is a serious overfitting problem with a tree being extremely branched and difficult to read. This is why adjusting parameters was

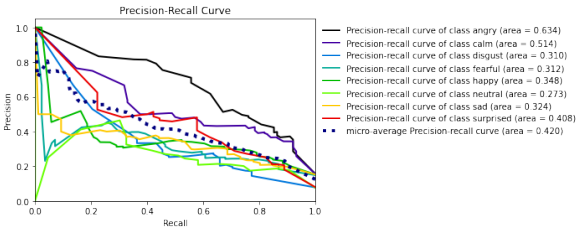
necessary and, again, it was done using RandomizeSearchCV with a cross validation of 5 folds repeated 15 times. It was therefore obtaine the following parameters as the best result: split criterion = gini, max depth = 19, min samples leaf = 0.0025852, min samples split = 0.012793 and the mean of cross-validated score about 39%. Looking again at the obtained model it's possible to observe again that there is probably an overfitting problem. For this reason it was opted for a post-pruning strategy in order to reduce the complexity. Once adjusted the ccp alpha parameter, which introduces a cost for the complexity of the model, with a GridSearchCV with the model parameters obtained previously, the resulting ccp alpha value is of 0.002091. Now, drawing the tree again and observing that it is much leaner than before and the following results in terms of model performance demonstrate it.



(a)



(b)



(c)

	precision	recall	f1-score	support
angry	0.62	0.58	0.60	88
calm	0.43	0.53	0.48	88
disgust	0.33	0.36	0.35	44
fearful	0.37	0.31	0.34	84
happy	0.34	0.42	0.37	86
neutral	0.46	0.30	0.37	43
sad	0.36	0.34	0.35	86
surprised	0.49	0.42	0.45	45
accuracy			0.42	564
macro avg	0.43	0.41	0.41	564
weighted avg	0.43	0.42	0.42	564

(d)

Figure 20: Confusion Matrix, Recall Curve and Prediction results decision tree applied to emotion

Results obtained are not completely satisfactory. In particular, the accuracy was 42%, which may be due to the fact that many emotions have similar traits. Specifically, it is observable that sad is equally classified as calm, while angry is the emotion classified in the most correct manner, as observed in the report.

3.3 KNN

The KNN classifier is an algorithm that relies on the concept of distance to make its predictions. For this reason, it is fundamental to standardise the data so that they all

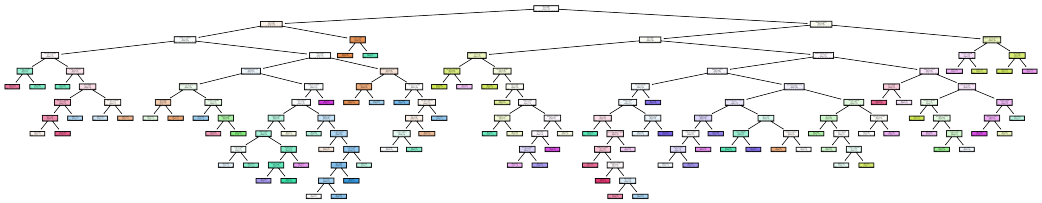
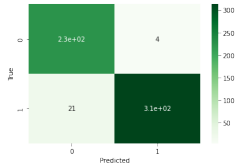


Figure 21: Decision tree for emotions target variable, after solving issues with post-pruning and using RandomizeSearchCV.

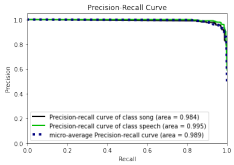
have the same scale.

3.3.1 KNN (vocal_channel)

Before proceeding, most important features were selected in order to make the model more efficient and solve problems related to high dimensionality. The selection was done by looking at the importance of the features in the decision tree and this led us to exclude: statement, stft_min, mfcc_std, mfcc_min, mfcc_max, min, max, emotional_intensity and sex. For implementation, division between train and test was made with 75% of the instances in the train and the remaining in the test with stratification. After this, it was on to search for the best parameter setting with GridSearchCV by comparing: n_neighbors: from 1 to 100; weights: uniform or distance, i.e. whether to give the instances the same weight or decreasing according to distance; metric: Euclidean or cityblock. There was as best result: metric = cityblock, n - neighbours = 4 and weights = distance; the mean of cross-validated score was about 95% (5 folds and 10 repetitions). Below are the results of the model's performance on the test, which has an AUC of 0.99.



(a)



(b)

	precision	recall	f1-score	support
song	0.91	0.98	0.95	230
speech	0.99	0.94	0.96	334
accuracy			0.96	564
macro avg	0.95	0.96	0.95	564
weighted avg	0.96	0.96	0.96	564

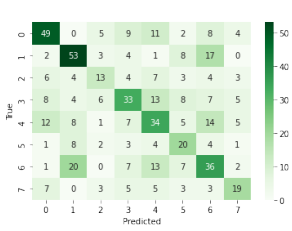
(c)

Figure 22: Confusion Matrix, Recall Curve and Prediction results of KNN applied to vocal channel

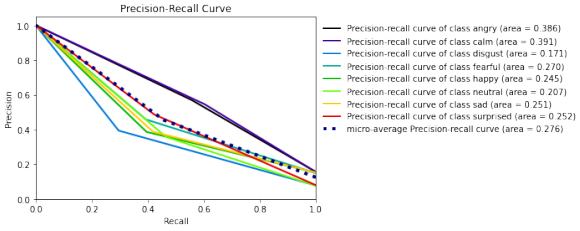
Results obtained can be described as more than satisfactory given the 96% accuracy.

3.3.2 KNN (emotion)

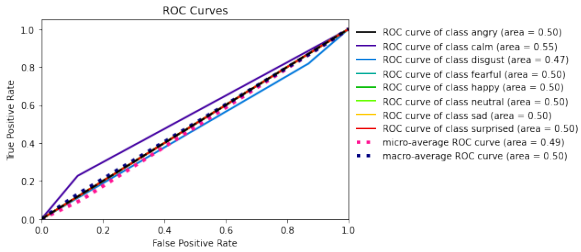
As seen for KNN vocal_channel, here too a feature selection was made for what was seen for decision_tree. In particular, emotional_intensity, stft_mean, sc_min, vocal_channel, statement and sex were excluded. For implementation, division between train and test was made with 75% of the instances in the train and the remaining in the test with stratification. Looking for the best parameter setting was done with GridSearchCV by comparing: n_neighbors: from 1 to 60; weights: uniform or distance, i.e. whether to give the instances the same weight or decreasing according to distance; metric: Euclidean or cityblock. It was obtained as best result: metric = cityblock, n - neighbours = 1 and weights = uniform and the mean of cross-validated score about 46% (5 folds and 10 repetitions). Below are the results of the model's performance on the test.



(a)



(b)



(c)

	precision	recall	f1-score	support
angry	0.57	0.56	0.56	88
calm	0.55	0.60	0.57	88
disgust	0.39	0.30	0.34	44
fearful	0.46	0.39	0.42	84
happy	0.39	0.40	0.39	86
neutral	0.36	0.47	0.40	43
sad	0.39	0.42	0.40	86
surprised	0.49	0.42	0.45	45
accuracy			0.46	564
macro avg	0.45	0.44	0.44	564
weighted avg	0.46	0.46	0.45	564

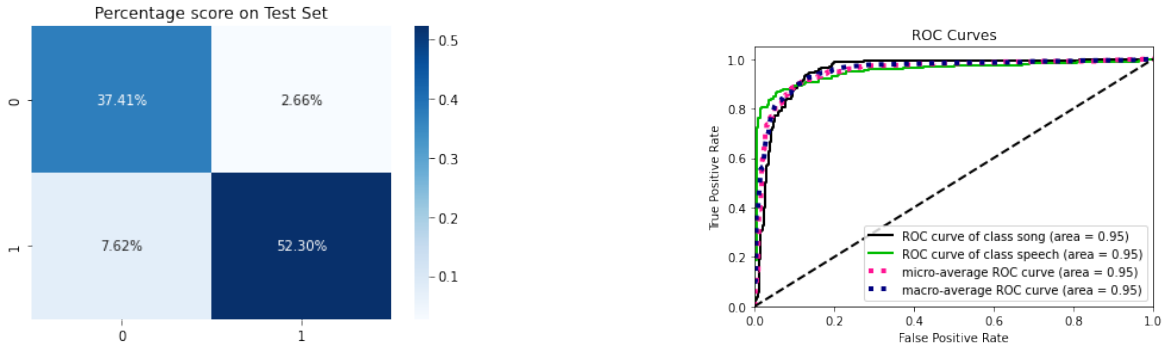
(d)

Figure 23: Confusion Matrix, Recall Curve, False positives and Prediction results KNN applied to emotion

3.4 Naive-Bayes

Naive Bayes was for predicting and inferencing two different target variable, 'vocal_channel' and 'emotion' (as in previous cases). This model works on a-priori probabilities and predict the possible variable target that a record with the highest probability must assume. Very useful will be the model fitted for predicting the missing

values for 'vocal_channel' target variable, and there were used this data like a validation set. Fitting the model on the train set score is good with only the 8.68% of mis-classification error, predicting well the rest; divided in 38.77% song and 52.54% speech (performance on the test are quite similar). This model have a better precision on the speech recognition respect to the song with a percentage score, of 83% against 95%. Area Under the ROC Curve has a score of 0.95.



(a) (b)

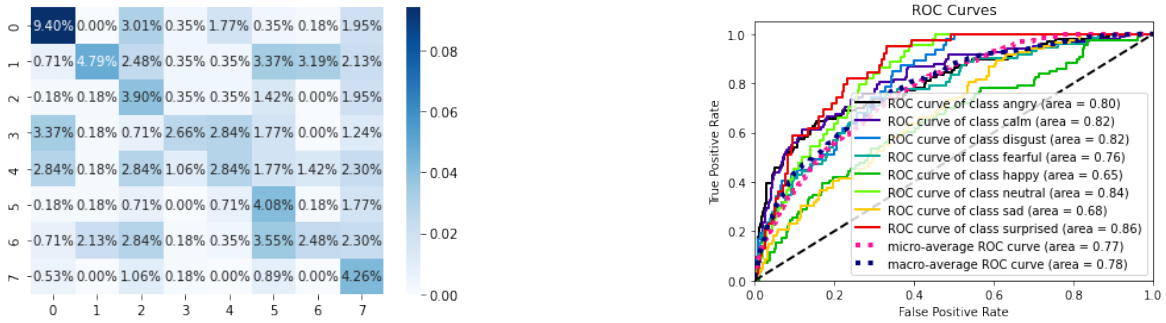
Figure 24: Confusion matrix and ROC curve for Naive Bayes application on test set.

For assumptions of the model (independence of the variable) and also for ensuring better performances (like avoiding overfitting problem and reducing model complexity and computational cost) it was deleted the most correlated variable (r = 0.75). After deleting 7 variable ('max','std','intensity','sc_skew','channels', 'mfcc_std','repetition') over others done during data preparation, model was re-trained obtaining same performances with benefits listed above:

With the prediction of the variable 'emotion' it was seen a different scenario because the number of labels is higher than 'vocal_channel' and the average accuracy is around 0.40. After a split trial of the dataset for vocal_channel, for each split there were a different training model obtaining a 0.05 precision increase for song split and a 0.02 precision decrease in speech split; so this couldn't be a solution for this task.

3.5 Conclusions about classification

In general, we can say that the best score in terms of accuracy is represented by KNN applied to vocal_channel. Other excellent scores to the same field are obtained applying decision trees and Naive-Bayes (94% and 90% respectively). For what concerns emotions, performances score are not comparable with previuos case, best score is obtained applying KNN with a 46% of accuracy. Clearly, it's all related to the variability of the target feature, considering also similarity between some of the emotions in the dataset.



(a) (b)

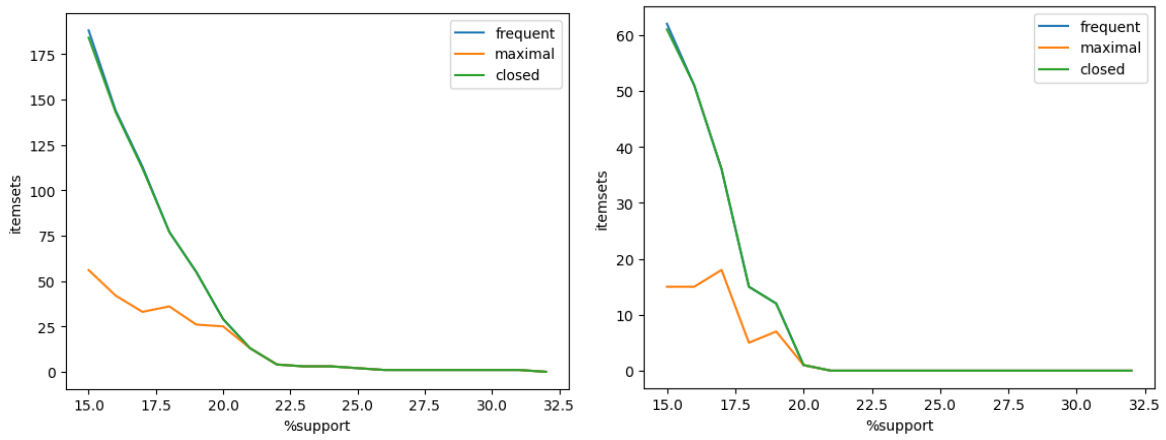
Figure 25: Confusion matrix and ROC curve for Naive Bayes application on emotion prediction.

4 Pattern Mining

In order to implement the Apriori algorithm, it was necessary to do some data preprocessing for the numerical variables. In particular, it was opted for quartile division using all the attributes in the dataset (excluding the variables already eliminated during data preparation).

4.1 Frequent Pattern Extraction

In order to extract frequent itemsets, plot of the two graphs were made with the curves of frequent, closed and maximal item-sets for a support in the range of [15,33] and z_min (the minimum number of items per itemset) of 3 and 4.



(a) (b)

Figure 26: Graphs of frequent itemsets for z_min 3 and 4, respectively.

We immediately notice that for a z_min of 4 the generated itemsets are already very few for a support of 15%, so it was decide not to lose too much information and be less selective by choosing a z_min of 3, even though it can create rather basic patterns. So, sorting the itemsets in descending order by their level of support, it was obtained that the first 4 turn out to be very similar to each other:

- (F, (-0.001, 709.51]sc_min, (-0.001, 0.000197]stft_min) Support: 31.20

- ((-0.001, 709.518]sc_min, (-0.001, 0.000197]stft_min, speech) Support: 25.08

- ((-0.001, 709.518]sc_min, (-0.001, 0.000197]stft_min, normal) Support: 24.69

- ((-0.815, -0.0961]stft_skew, (0.531, 0.724]stft_mean, speech) Support: 22.07

it's possible to obesrve also the presence of sc_min and stft_min in the first 3. Moreover, in terms of significance, they do not give much information and the fact that these items are together could be mere randomness.

4.2 Association Rules extraction and Classification

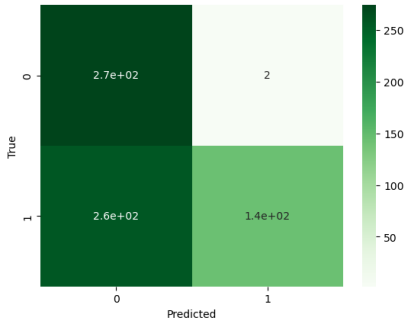
At this point our goal is to extract the most interesting association rules. To do this we chose a confidence level of 70%, in addition a threshold for support of 20% and z_min of 3. Once extracted, we filtered the association rules for speech as consequent, and here there are the results:

-((0.531, 0.724]stft_mean, (-0.815, -0.0961]stft_skew) Support: 22.07, Confidence: 0.98, Lift: 1.65.

- ((0.209, 0.318]stft_std, M) Support: 21.01, Confidence: 0.92, Lift: 1.55.

- ((0.531, 0.724]stft_mean, M) Support: 20.79, Confidence: 0.97, Lift: 1.64.

All three extracted rules present high confidence levels and at the same time a lift greater than one; therefore, the variables present a positive relationship and are not statistically independent. We decide to use the first rule, as it presents the highest support, confidence and lift values, to make predictions on the same test set obtained during the classification by setting vocal_channel as the target, in order to make a comparison between our classifiers and the extracted rules. Below are the results obtained.



(a)

	precision	recall	f1-score	support
song	0.52	0.99	0.68	276
speech	0.99	0.36	0.53	401
accuracy			0.62	677
macro avg	0.75	0.68	0.60	677
weighted avg	0.79	0.62	0.59	677

(b)

Figure 27: Graphs for association rules from highest support.

5 Regression

Building a multiple linear regression model allows you to quantify the relationship between the dependent variable (the y) and a set of explanatory variables (the x). It also helps you predict what the value of y will be for given values of x. For this model the target variable is 'intensity' because we have the opportunity to predict the big amount of missing value (33.28%). The model is built with the same procedure of the Naive Bayes, deleting the most correlated variable, passing to 38 features to 26 gaining reducing overfitting and model complexity. Before fitting the mdoel, data were standardized and then back transformed to normal scale.

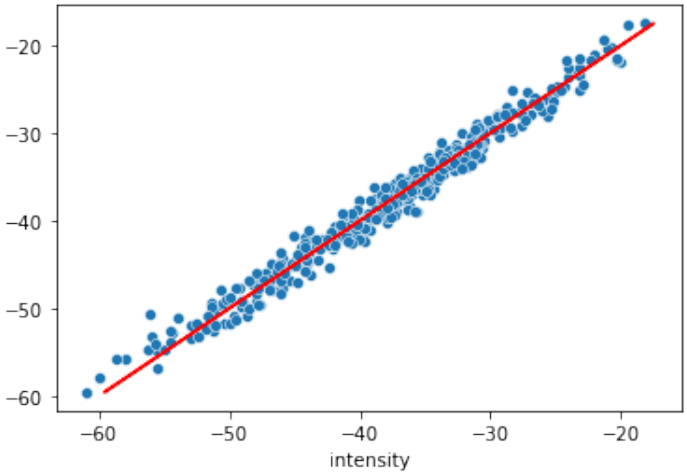


Figure 28: Scatter plot of test prediction vs test set.

The R2 on the test set is very high, 0,976 and the Mean Square Error is 1.605, so the model fit and predict very well the data. It's possible to notice by the plot that model has high performance due to association of points with the red line.

Below we compare the replacement of missing values with multiple linear regression and mean conditioned by sex, emotion and statemen, as shown in the Data Preparation.

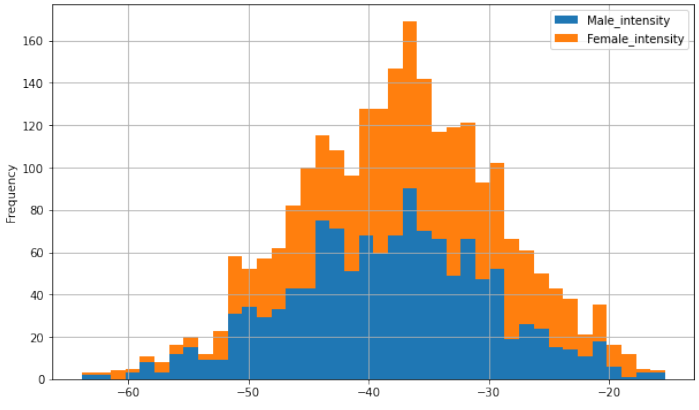


Figure 29: Distribution of filled data of intensity conditioned by sex.

In conclusion, it's possible to say that the best way to fill missing value about intensity is using regression (of course) because distribution is more similar to a Gaussian than the mean conditioned by sex, emotion and statement, as shown in the Data Preparation (Fig.7).