UNIVERSITÀ DI PISA

Department of Informatics

Master’s Degree in Data Science and Business Informatics

# Data Mining - 1 Final Report

Authors:

Mohamed Arafaath Sathik Basha - 659588

Vincenzo Rocchi - 664957

Cristian Ferrara - 657496

Professor:
Riccardo Guidotti

Academic year 22/23

# Table of Contents

# 1.Data Understanding

"The RAVDESS is a validated multimodal database of emotional speech and song. The database is gender balanced consisting of 24 professional actors, vocalizing lexically matched 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, with an additional neutral expression. All conditions are available in face-and-voice, face-only, and voice-only formats." We are looking at part of the RAVDESS original dataset, more precisely at the voice-only one. The idea that lays behind the construction of this dataset is due to the changes occurred in the emotion studies during the last decade, a lot has changed in how we see and treat emotions and so are the studies associated with the topic. The importance of a reliable and validated record of expression of emotions is crucial to the integrity of the studies focused on vocal recognition or more in general sound classification and recognition.

## 1.1.Data semantics

Analyzing it from up close we can clearly distinguish two parts, one containing the details of the recordings, the details of the actor that recorded them and the technical audio data, the other containing the extracted statistical data used for: zero-crossing rate, Mel-Frequency Cepstral Coefficients, spectral centroid, and the stft chromagram. The first half represents the 40% of the total data and the other the remaining 60%. Starting the analysis from the non-statistical part the "*modality*" represents the type of file and in this case there are audio only ones, that can be further distinguished by the "*vocal_channel*" attribute that specifies if the recording was spoken or singed. The phrases utilized by the 24 actors (identified by being assigned a number from 1-24 under the "*actor*" attribute and further divided by "*sex*" in F or M) are "Kids are talking by the door" and "Dogs are sitting by the door" the one used is specified under the "*statement*" attribute. The phrases are repeated 2 times for each actor the order can be seen under the "repetition" attribute. The most interesting part comes when looking at the "emotion" and "emotional_intenisty" attributes that indicate respectively the emotion that is interpreted by the actor and the intensity at which that emotion is represented. Those attributes alone though are not useful enough, but combining them with the statistical data describing the audio waves, make us seek for a correlation path between the emotion and the small changes in the spectrum of the audio recording. Such features are already used for various audio self-recognition techniques, the "MFCC" or Mel-Frequency Cepstral Coefficients 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 and is used in music information retrieval applications such as genre classification, audio similarity measures, etc. The "SC" or spectral centroid has many applications in audio classification and music classification and it's used in processing of audio signals, especially music genre classification. The "STFT" or Chroma STFT value of an audio basically represent the intensity of the twelve distinctive pitch classes that are used to study music. They can be employed in the differentiation of the pitch class profiles between audio signals.

A quick recap of all the attributes and their definitions:

**Descriptive part**:

- **modality** (audio-only)
- **vocal_channel** (speech, song)
- **emotion** (neutral, calm, happy, sad, angry, fearful, disgust, surprised)
- **emotional_intensity** (normal, strong). NOTE: no strong intensity for the 'neutral' emotion
- **statement** ("Kids are talking by the door", "Dogs are sitting by the door")
- **repetition** (1st repetition, 2nd repetition)
- **actor** (01 to 24)
- **sex** (M, F)

**Technical part:**

- **channels** (number of channels; 1 for mono, 2 for stereo audio)
- **sample_width** (number of bytes per sample; 1 means 8-bit, 2 means 16-bit)
- **frame_rate** (frequency of samples used (in Hertz))
- **frame_width** (Number of bytes for each frame. One frame contains a sample for each channel.)
- **length_ms** (audio file length (in milliseconds))
- **frame_count** (the number of frames from the sample)

- **intensity** (loudness in dBFS (dB relative to the maximum possible loudness))

**Statistical part:**

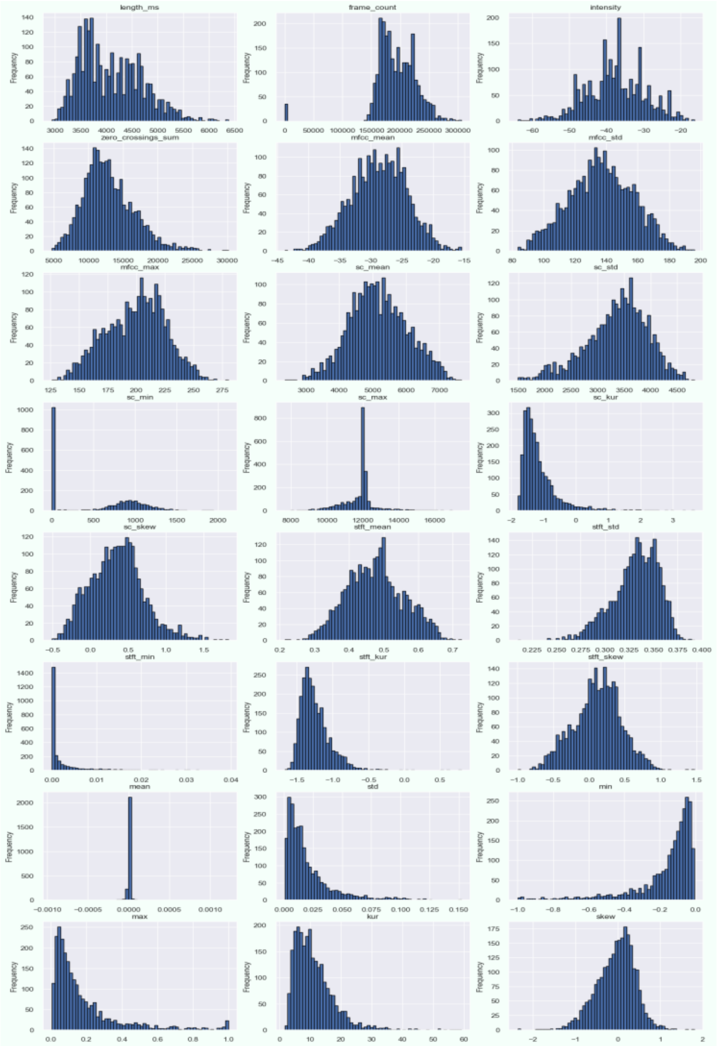
- **zero_crossings_sum** (sum of the zero-crossing rate)
- **'mean', 'std', 'min', 'max', 'kur', 'skew'** (statistics of the original audio signal)
- **mfcc_ 'mean', 'std', 'min', 'max'** (statistics of the Mel-Frequency Cepstral Coefficients)
- **sc_ 'mean', 'std', 'min', 'max', 'kur', 'skew'** (statistics of the spectral centroid)
- **stft_ 'mean', 'std', 'min', 'max', 'kur', 'skew'** (statistics of the stft chromagram)#my initial idea of correlating the stat part is probably wrong or I did it too quickly im gonna work on it better tomorrow morning already sent you a message.

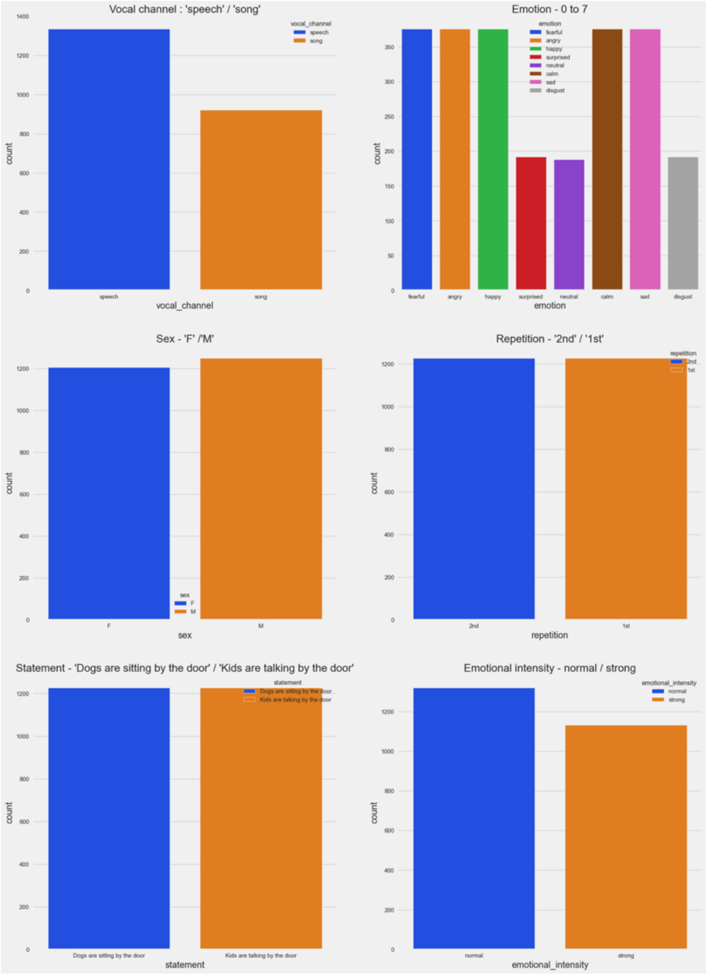
## 1.2.Distribution of variables and statistics

To better understand the dataset, we decided to visualize the distributions of all variables through histograms and boxplots, that are presented below, no clear pattern emerges from this first visualization, only some possible errors or deviations in the numerical attributes (addressed later on),  except for the vocal channel and the emotion attribute that show less presence of different characteristics throughout the test.

*Fig.1 - numerical values*

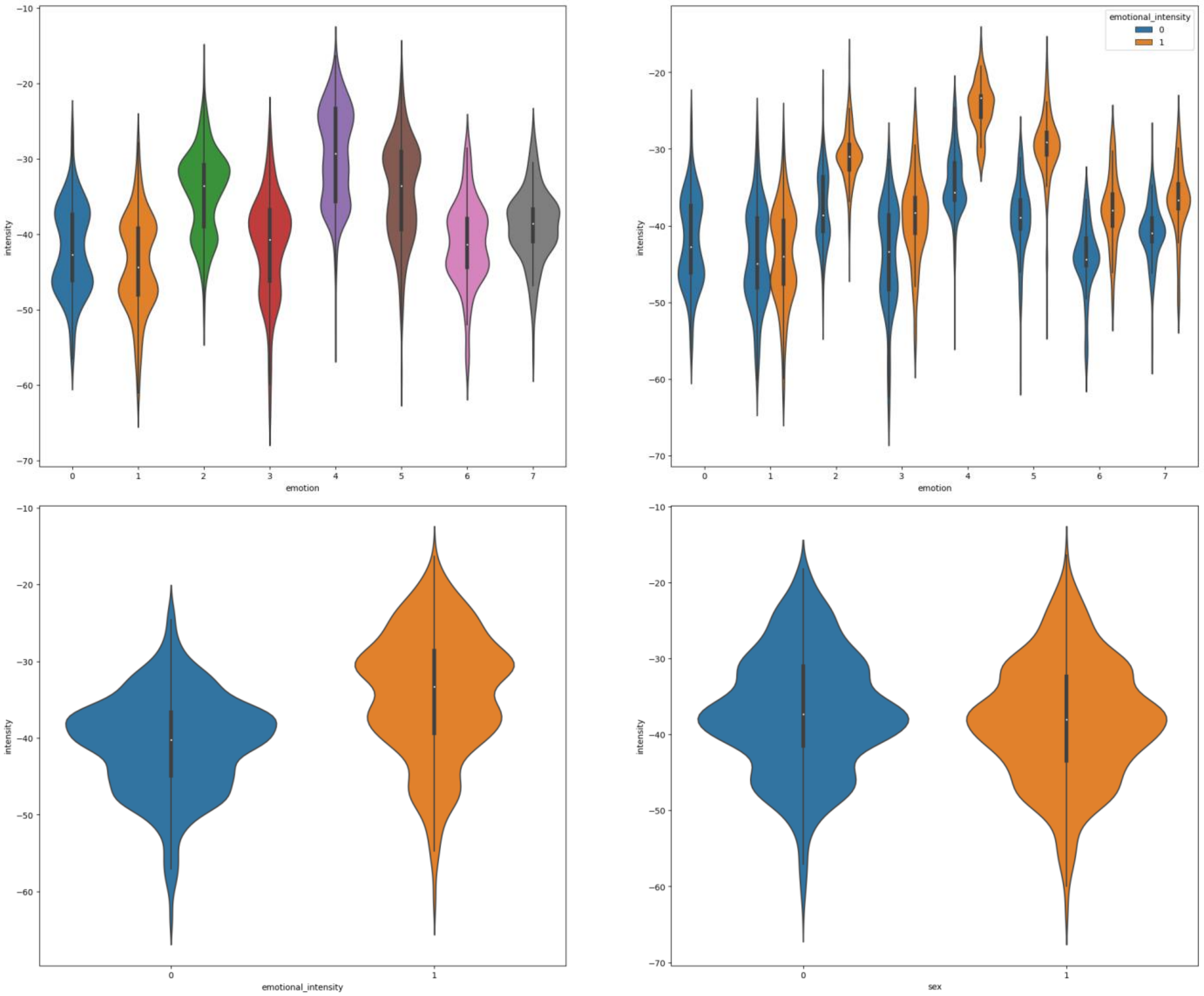
*Fig.2  - categorical values*





since we concluded that the most interesting attributes to plot mainly against are the intensity and the emotion one, we created some violin plot that helps us visualize rapidly the trends between those 2 attributes and the other chosen: for the first one (intensity) we decided to plot against emotion, emotional intensity and sex. The emotion is mapped as follows (0 = neutral, 1 = calm, 2 = happy, 3 = sad, 4 = angry, 5 = fearful, 6 = disgust, 7 = surprised), the intensity is represented negatively relative to the maximum possible level of loudness in negative dB, it works in exactly the same way as the positive representation, but with every negative 10 dB simply indicating a factor of 10 LESS. So, -10 dB is 1/10th that of 0 dB, -20 dB is 1/100th of 0 dB, -30 dB is 1/1000th of 0 dB and so on again. We can conclude that when people are angry, fearful and happy the intensity can be much higher than the other feelings and the probability of a more intense sound is much higher; in the disgusted and surprised state the intensity is generally on pair with the neutral and calm feelings although the mean is higher, it presents a shallower 1.5x interquartile range and for so a higher probability of it being at a higher level than the neutral and calm state, making it more distinguishable and less situational; when they present a sad expression the intensity can be much lower than the other emotions but the probability of it being at a normal level is still higher than it being less intense and this makes the sadness the less distinguishable emotion of the bunch. Comparing the intensity and the emotional intensity (0 = normal, 1 = strong) we can see that the normal intensity has a broader range, a lower mean and a shallower inter quartile range, the probability of it being not so loud is much higher than the strong emotional intensity and so is less variable and more distinguishable. Next to the first plot is the one that merges the first two and let us comprehend how the emotional intensity also plays a big role in the definition of the loudness and the ability to recognize a felling by the intensity of the voice. The last plot is the sex against the intensity and we can clearly see that there are only small variations between male and female and the correlation with the intensity of the statement.

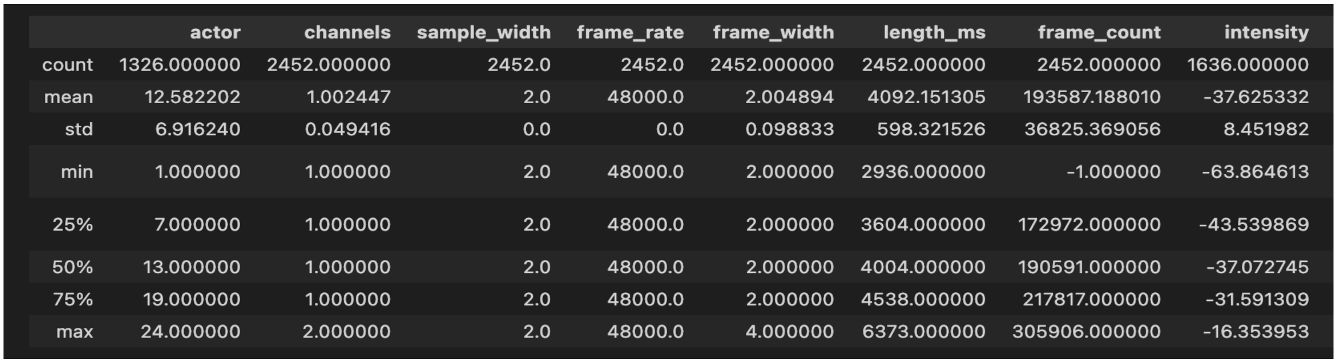
*Fig.3 - emotion and intensity (h-left) - emotional intensity hue (h-right) – emot-int and intensity (l-left) – sex and intensity (l-right)*



## 1.3.Dataset preliminary analysis

The “RAVDESS” data set we are working on is composed by 2452 rows and 38 columns, the first 14, as said, are the one that do not make an in depth statistical analysis on the more technical audio part, those are the ones that we’ll be analyzing as in first and these are also the ones that require an in depth data quality assessment. Doing so, we’ll address the data quality part of the essay, making our lives much easier finding the purpose of the analysis itself. Starting with a quick look on the statistical means of the numerical data:

*Table 1 - statistical means of the dataset*



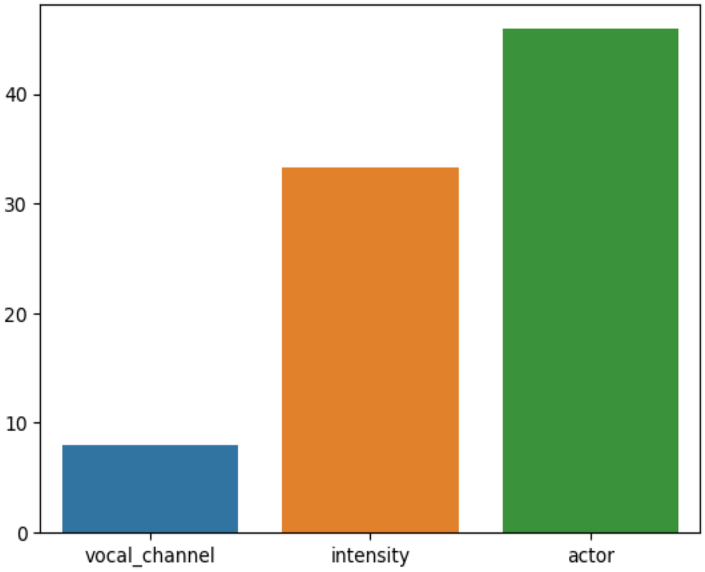
| | actor | channels | sample_width | frame_rate | frame_width | length_ms | frame_count | intensity |
|---|---|---|---|---|---|---|---|---|
| count | 1326.000000 | 2452.000000 | 2452.0 | 2452.0 | 2452.000000 | 2452.000000 | 2452.000000 | 1636.000000 |
| mean | 12.582202 | 1.002447 | 2.0 | 48000.0 | 2.004894 | 4092.151305 | 193587.188010 | -37.625332 |
| std | 6.916240 | 0.049416 | 0.0 | 0.0 | 0.098833 | 598.321526 | 36825.369056 | 8.451982 |
| min | 1.000000 | 1.000000 | 2.0 | 48000.0 | 2.000000 | 2936.000000 | -1.000000 | -63.864613 |
| 25% | 7.000000 | 1.000000 | 2.0 | 48000.0 | 2.000000 | 3604.000000 | 172972.000000 | -43.539869 |
| 50% | 13.000000 | 1.000000 | 2.0 | 48000.0 | 2.000000 | 4004.000000 | 190591.000000 | -37.072745 |
| 75% | 19.000000 | 1.000000 | 2.0 | 48000.0 | 2.000000 | 4538.000000 | 217817.000000 | -31.591309 |
| max | 24.000000 | 2.000000 | 2.0 | 48000.0 | 4.000000 | 6373.000000 | 305906.000000 | -16.353953 |

The values extracted, as at this stage, are: firstly the count of the values itself and this first data already tells us that some of the attributes lack a lot of records and that we’ll be dealing with a massive amount of missing values later on, not all the attributes are full of data and some are missing as much as 50% of all the records done. The min/max let us take a quick snap on the length and the extremities of the values in the records itself. The std deviation and the mean help us further identify mismatches in the distribution of our data, by comparing the std value with the mean we can see that most of the data does not vary by a great amount and that the distribution rests accordingly. Utilizing exclusively the std deviation is not quite enough to take a full picture of the data deviation in the dataset though. Lastly, accounting for the interquartile range, looking at the difference between the first and the third quartile (50% on the table), we can say that our dataset does not sit in a broader than normal deviation range. The analysis helped us also identifying some of the problems with the data quality in our dataset and so we can conclude that it clearly presents some errors probably due to the registration of the records themselves.

# 2.Data quality

## 2.1.Missing values:

*Table 2 - minssing values (%)*



1)We can currently see that all the values indicating something that is logically obvious are in fact useless for the analysis and so do the attributes that account only for one value. The first obvious column we account for is the modality one, as we already said, the modality in this dataset will be voice-only and not include the facial video recording part. We are done for now and we’ll be looking again at unique values later on to get the job computationally easier. Assessing the missing values is the priority now and getting a better understanding of the distribution and the correlation of all the variables is the puprose. The picture is much more complete and we can confirm the exact percentages of the missing values already discovered in the first preliminary analysis. In fact the vocal channel column has 7,99%, the intensity one has 33,28% and the actor has 45,92% of the total values missing. The first one to asses is the vocal channel and we are filling it with the mode of the values recorded, due to the relatively small percentage of missing values. Regarding the intensity we grouped the data by emotional intensity, vocal channel and emotion (which are the most correlated ones) and then filled with the mean of the resulted grouped dataset. This procedure assures us, by comparing a distribution graph of the before and after, that filling the missing values hasn't moved the weight of the distribution, biasing the results.

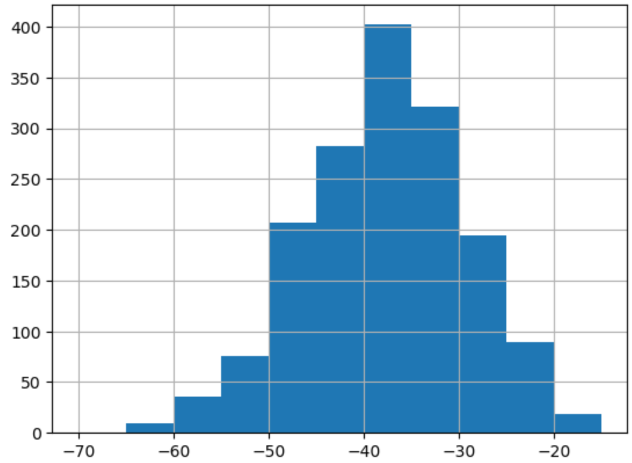


*Fig.4 - before*

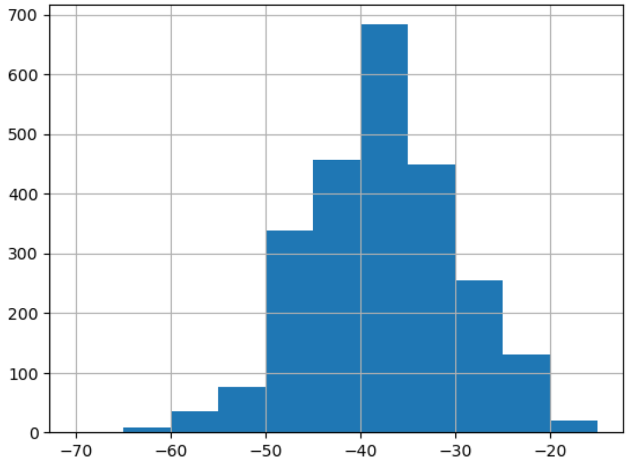


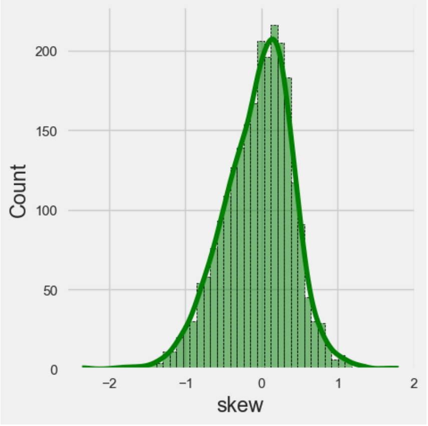
*Fig.5 - after*

The last attribute that has missing values is the actor one, they account for as much as 50% of the total data, and no visible correlation appears between this and other attributes, not having enough information to fill the data resulted in us deciding that is better to entirely drop the column.

## 2.2.Variable transformation:

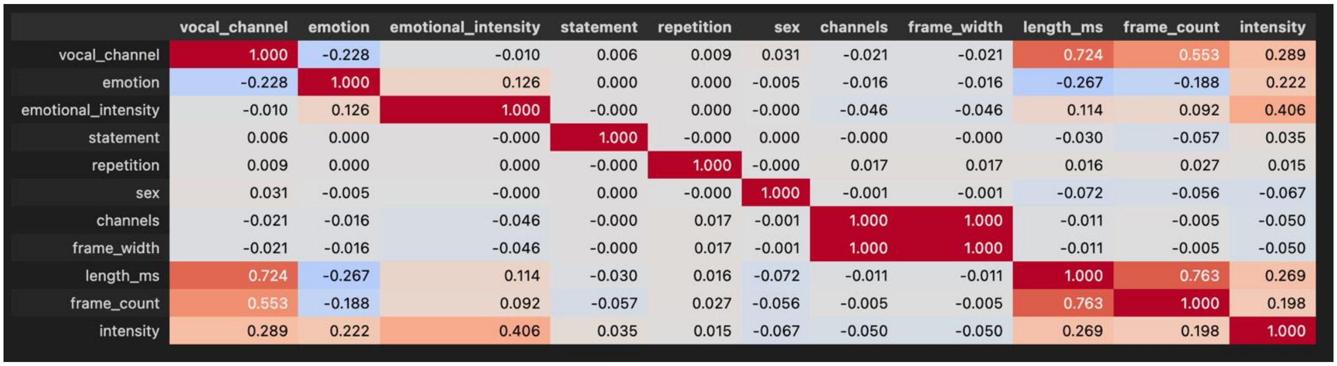
Looking at the categorical values we have in our dataset we can see how most of them are composed by only 2 different values, so we used the binary encoding to transform them over one hot encoding. This transformation accounts for the lack of the computational possibility to process text values and get information as the correlation between the different attributes. All the categorical attributes except the emotion one are expressed this way, for this last one instead of using a binary encoding technique (0,1), as the values can be ordered in a range that goes from the less intense emotion, it being the “neutral” one, to the highest intense emotion, it being “surprised”, we have encoded the values with an ordinal range of 0 – 7 based on its intensity. The mapping schema will be useful as we go along in our analysis to make plotting and working with the data easier in a data manipulation perspective. Except the mapping part, regarding variable transformation, we did not see the need to go with log transformed data since, like you can see in the figure on the right, the skeweness of the dataset is only 0.5 (falling in the normal zone) therefore the dataset is marginally left-skewed but a log-transformation is not useful enough to justify the use of it. Furthermore, like already said above, the variability of the data (obtained by comparing the mean and the STD) is marginal and even in this possible case for log-transformation, we did not see the need to apply it.

*Fig.6 – skew vs count*



## 2.3.Correlation matrix and eliminated variables

After dealing with missing values and transforming the variables above in numerical ones we can create a correlation matrix, also related to the first part of the dataset, that gives us a quick look on the relations between the various records. Before proceeding further with it is appropriate to reduce the dataset by removing other not useful records. The two attributes that are representing only one unique value are: sample width and frame rate. Those 2 have the same value across all the records, as said before, are useless for our analysis and therefore will be dropped entirely. Managed those two we can go on with the correlation matrix.

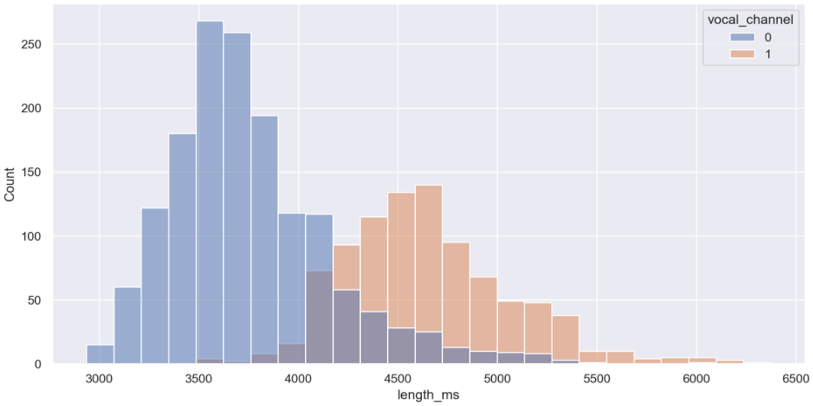


| | vocal_channel | emotion | emotional_intensity | statement | repetition | sex | channels | frame_width | length_ms | frame_count | intensity |
|---|---|---|---|---|---|---|---|---|---|---|---|
| vocal_channel | 1.000 | -0.228 | -0.010 | 0.006 | 0.009 | 0.031 | -0.021 | -0.021 | 0.724 | 0.553 | 0.289 |
| emotion | -0.228 | 1.000 | 0.126 | 0.000 | 0.000 | -0.005 | -0.016 | -0.016 | -0.267 | -0.188 | 0.222 |
| emotional_intensity | -0.010 | 0.126 | 1.000 | -0.000 | 0.000 | -0.000 | -0.046 | -0.046 | 0.114 | 0.092 | 0.406 |
| statement | 0.006 | 0.000 | -0.000 | 1.000 | -0.000 | 0.000 | -0.000 | -0.000 | -0.030 | -0.057 | 0.035 |
| repetition | 0.009 | 0.000 | 0.000 | -0.000 | 1.000 | -0.000 | 0.017 | 0.017 | 0.016 | 0.027 | 0.015 |
| sex | 0.031 | -0.005 | -0.000 | 0.000 | -0.000 | 1.000 | -0.001 | -0.001 | -0.072 | -0.056 | -0.067 |
| channels | -0.021 | -0.016 | -0.046 | -0.000 | 0.017 | -0.001 | 1.000 | 1.000 | -0.011 | -0.005 | -0.050 |
| frame_width | -0.021 | -0.016 | -0.046 | -0.000 | 0.017 | -0.001 | 1.000 | 1.000 | -0.011 | -0.005 | -0.050 |
| length_ms | 0.724 | -0.267 | 0.114 | -0.030 | 0.016 | -0.072 | -0.011 | -0.011 | 1.000 | 0.763 | 0.269 |
| frame_count | 0.553 | -0.188 | 0.092 | -0.057 | 0.027 | -0.056 | -0.005 | -0.005 | 0.763 | 1.000 | 0.198 |
| intensity | 0.289 | 0.222 | 0.406 | 0.035 | 0.015 | -0.067 | -0.050 | -0.050 | 0.269 | 0.198 | 1.000 |

*Fig.7- non statistical part correlation matrix (Pearson method)*

To create the correlation graph we used the Pearson corr. method together with the Kendall one to better understand the type of correlation between the variables. Briefly there are no highly correlated values except for the ones we already used to fill the missing ones and the length in ms and the frame count. Those last two are positively correlated (0.73P - 0.97K) as the length of the recording increases so does the frame count due to the recording been physically bigger and requiring more frames to be stored, the two methods used help us understand that there is a stronger monotonic relation (kendall) and so the variables move for 97% of the time in the same direction but not at a constant rate (only 73%). Another strong positive correlation is between the length and the vocal channel, as the length increases the vocal channel does to (0 = speech, 1 = song), so we can conclude here that the lengthier audio recordings are the one in which the actors sing (a visual representation of the output on the right)

*Fig.7 – count plot for length_ms with vocal channel as hue*



## 2.4.Outliers

Checking if there are outliers in the dataset is as important as accounting for missing values, it makes the representation of the data itself clearer and not biased by far data points (unusual values that vary from one attribute to another). To identify the outliers there is no strict mathematical rule but we can use boxplots to get a visual representation of the values outside the normal range and decide from there. In the case of the RAVDESS dataset there is no excessive variability in the data regarding the non-statistical part, however the statistical means part is highly variable and presents a lot of outliers and removing all of them resulted in a dataset more than halved. Therefore, we defined a method to get the outliers of all the categorical attributes compared to the most correlated continuous one. When we got all the outliers, following the method explained above, we intersected those that we found and removed the one that are common between the analyzed one. Using this approach guarantees us continuity and gives a less biased dataset from a logical-mathematical perspective (even though the statistical correlation between the records is nothing to be impressed with, this approach implies a more conscious understanding of the dependencies of the records).

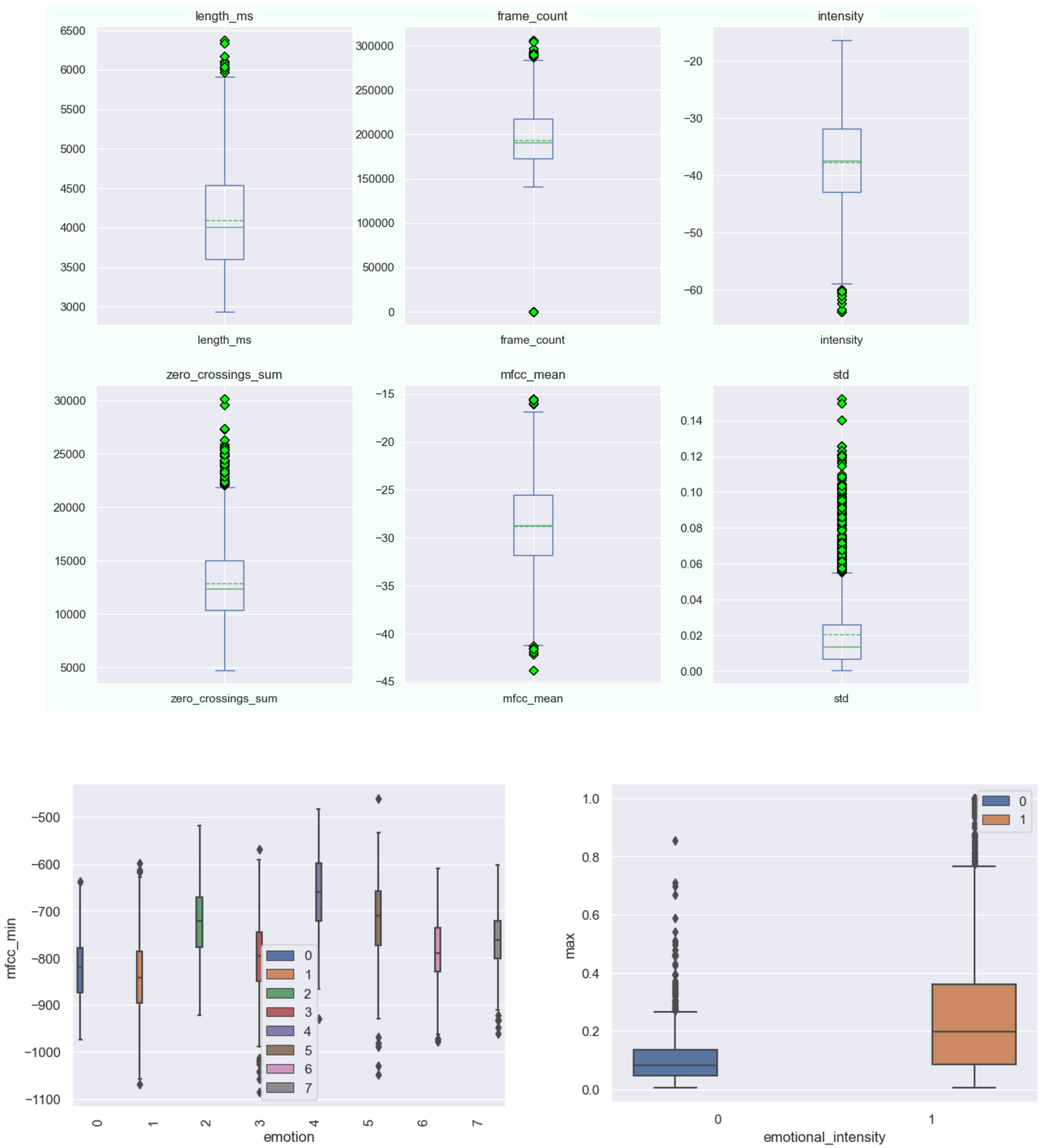
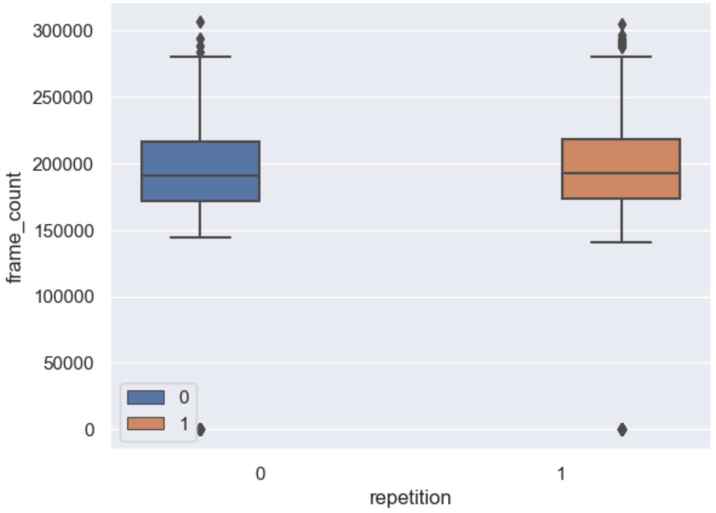
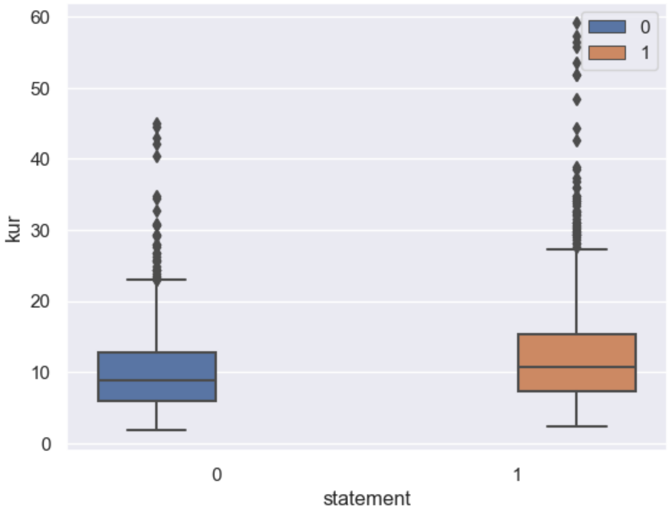
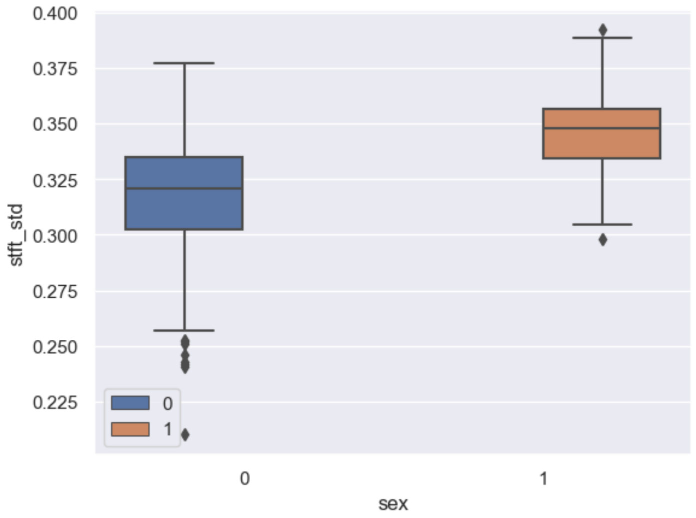


*Fig.8 - Outliers*



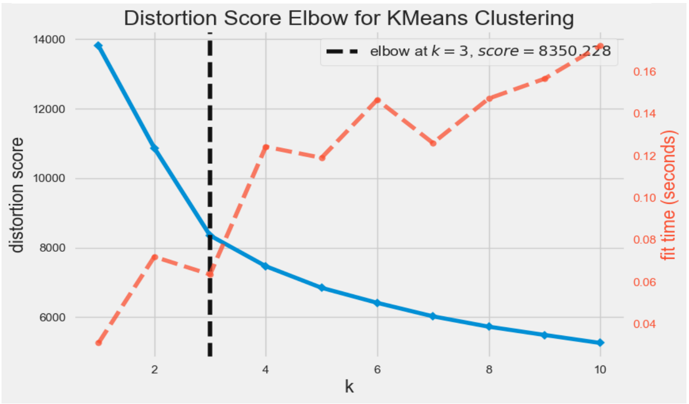




# 3.Data Clustering

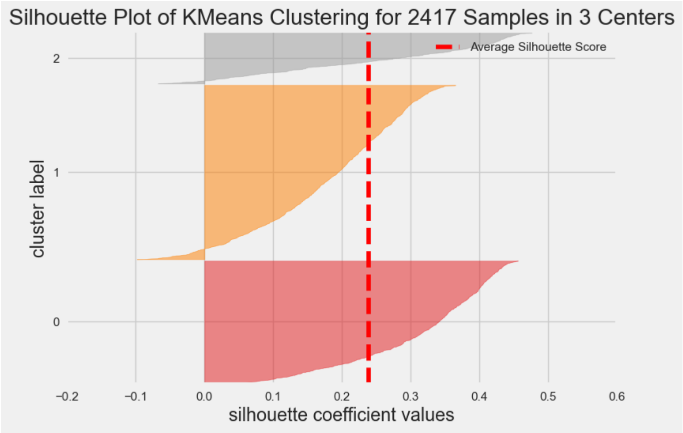
In this section we will deal with clustering the units present in the dataset, in order to obtain groups that have common characteristics. What is expected is to find any correlations that, in the previous phase of Data Understanding had not emerged. The chosen attributes to carry out the cluster analysis will all be quantitative and will be specified from time to time for each technique used. First, the K-means clustering technique will be applied. After that the results related to the DBSCAN and Hierarchical techniques will also be shown. In order to do the clustering, after our analysis, we chose the following attributes mentioned below for best results which are as follows: "vocal_channel", "length_ms", "emotion", "mfcc_min", "emotional_intensity", "max", "statement", "kur", "repetition", "frame_count", "sex", "stft_std". We have used Min_Max Scaler to scale the values in dataset to start with clustering. Our goal was to identify any correlations with the attributes considered.

*Fig.9 -Elbow Graph*



## 3.1.K-means

The **K- means** is a partitioned group analysis algorithm that allows to subdivide a set of objects into k groups based on their attributes. It is a variant of the expectation-maximization (EM) algorithm whose goal is to determine the k groups of data generated by Gaussian distributions. The initial value of the centroids was defined using the 'k- means++' method. This method initializes the



centroids so that they are far apart, thus allowing better results than the random initialization method. The latter, in fact, could lead to a non-optimal but local solution. However, with the chosen method, the centroids will be more precise and therefore fewer iterations will be required. To identify the optimal number of clusters (k), it was decided to calculate the value of the SSE (Sum of Squared Errors) for a number of clusters ranging from 2 to 11 and also plot them to find distortion score elbow, we found out that at k = 3 we had an elbow score of 8350.23 after which k-means algorithm tries to minimize distortion (SSE) between each observation vector and its dominating centroid. We also calculated the Silhouette Coefficient and plotted to view the average Silhouette Coefficient which in our case was 0.24. The value of the Silhouette coefficient in the best case is close to 1, therefore the value of k is sought such that this is the maximum and in our case k=3 gave us the best coefficient.

From the figure shown, it can be seen that the clusters are much more influenced by the variables mfcc_min, frame_count, length_ms and stft_std.. Indeed, the centroids of Clusters B and C have similar values for the emotion, statement, kur, repetition and sex attributes, while the centroids of Clusters A and B have similar values for the emotionl_intensity, statement, repetition and sex attributes. The clusters are well seperated at vocal_channel, length_ms, mfcc_min, frame_count and stft_std. By our analysis, we found that, the 3 different clusters are perfectly separated at mfcc_min and frame_count which is plotted below.We should also note that, inspite of having few outliers, our K Means algorithms has perfectly distinguished one cluster from another.

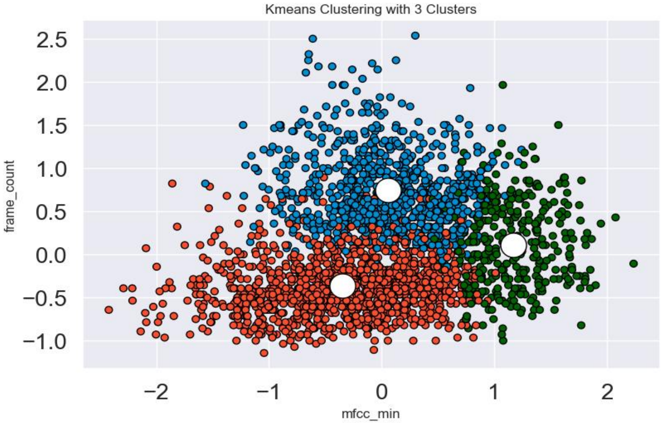
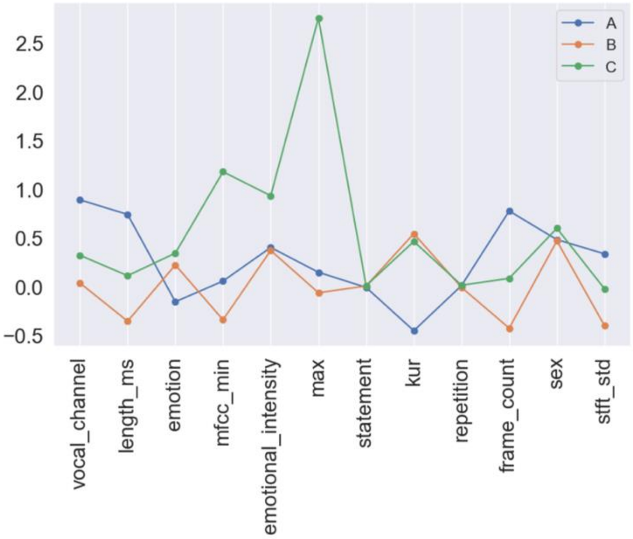


*Fig.10 – K-Means Clustering*



When we tried to compare the clusters with all the categorical attributes, the attribute “Statement” had the highest percentage (42 %) of values which were similar to the clusters formed by our K-Means Algorithm.

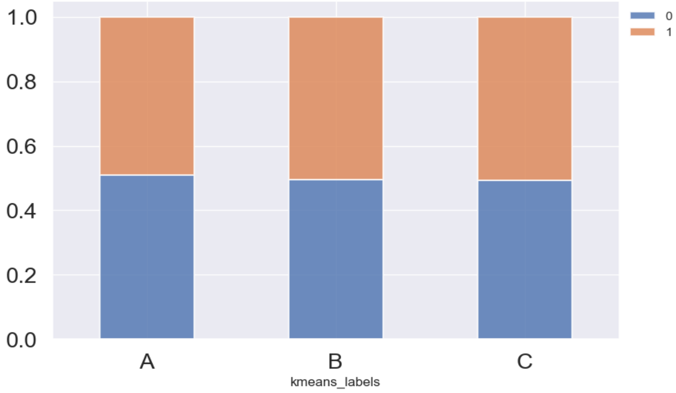


*Fig.11 – K-Means Labels*

## 3.2.DBSCAN

The dbscan is based on density because it connects regions of points with sufficiently high density. DBSCAN is one of the most used clustering algorithms and is also the most cited in the scientific literature. DBSCAN estimates the density around each point (item) by counting the number of points in a neighborhood (eps) specified by the user, and applies call thresholds minPts to identify the “core”, “border” and “noise” points. In a second step, the core points are gathered in a cluster, if they are "density-reachable" ("reachable by density", i.e. if there is a chain of core points in which each point falls within the eps-around the following). Finally the edge points are assigned to the clusters. The algorithm requires only the parameters eps and minPts

Shows the distribution of the cluster and noise points with respect to the target variable in the scatter plot below:

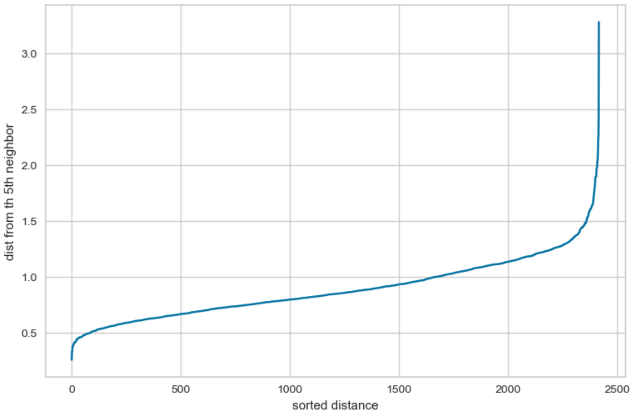


*Fig.12 – Distance plots*

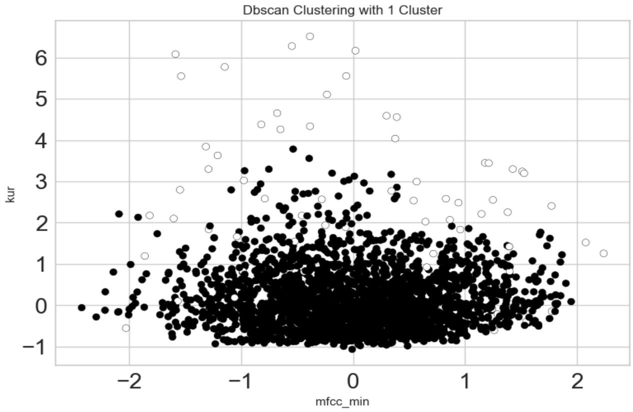
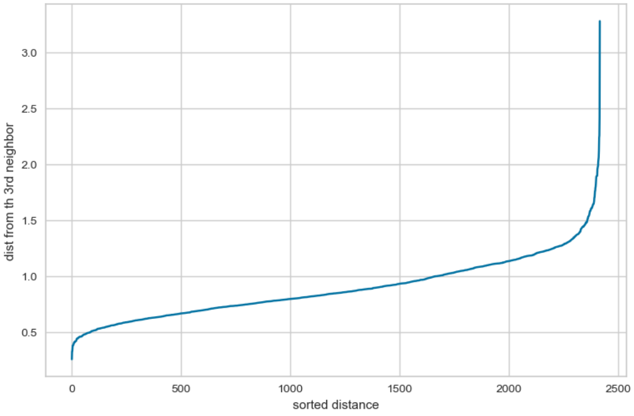
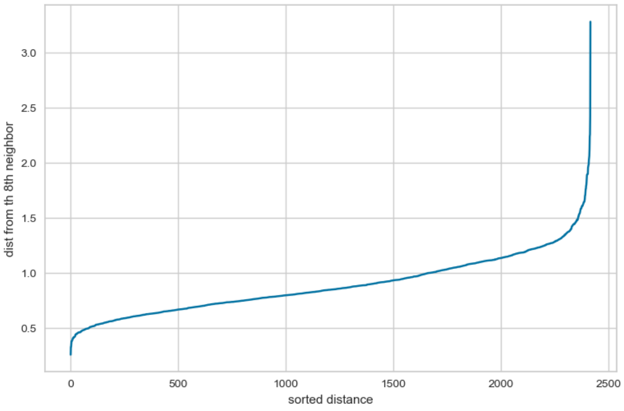


*Fig.11 – DBSCAN Clustering*





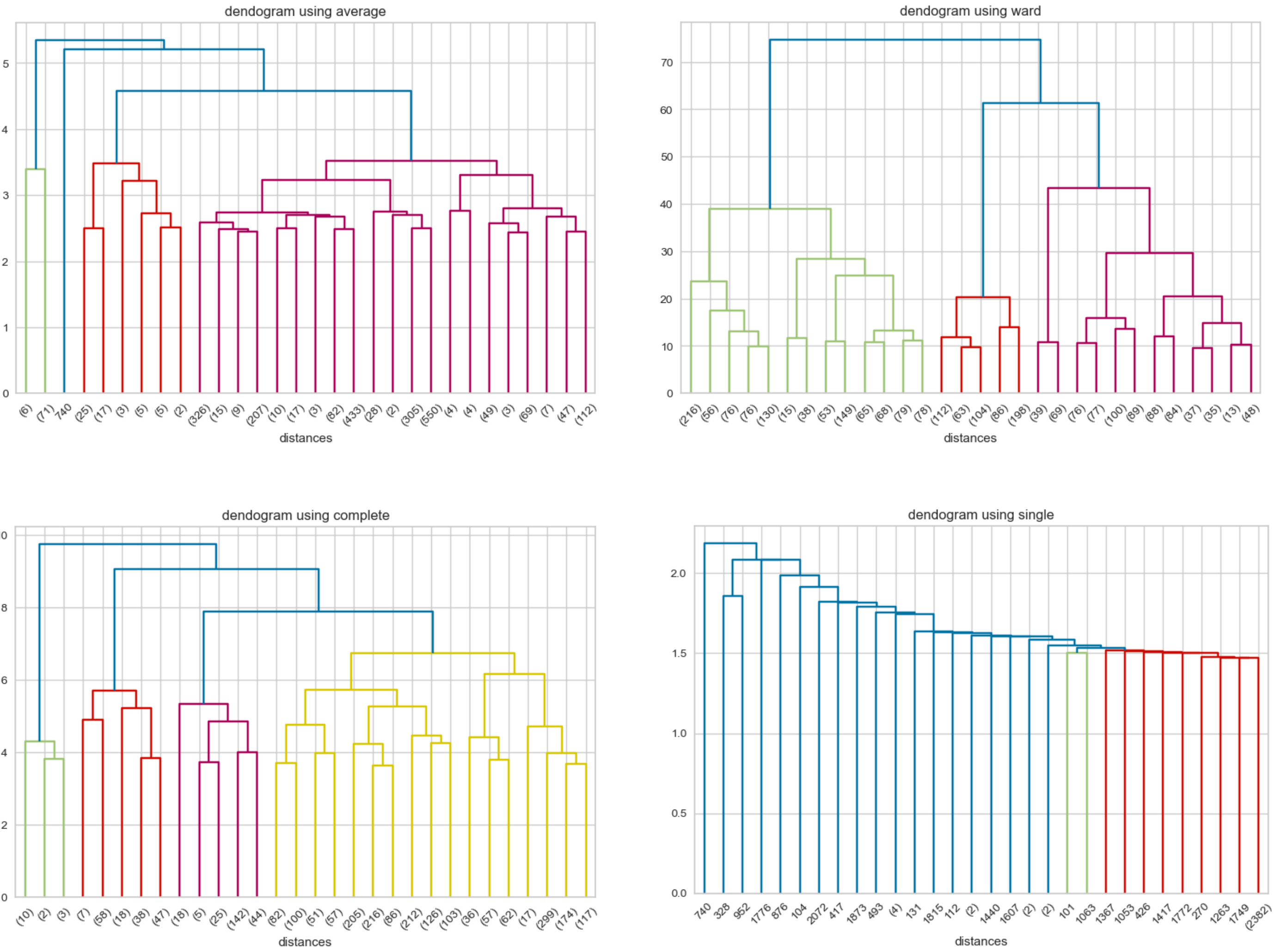
The best value for the Epsilon parameter is found near the 'elbow' of the function represented. This is because, above that part of the curve, increasing Epsilon does not further increase the maximum distance between points. Having determined the parameters, the DBscan technique was applied to the entire dataset. Good results are obtained considering MinPoints=3 and Epsilon=1.25. A single cluster made up of 58,349 vehicles and 37 noise points was identified. The calculated Si lhouette coefficient is equal to 0.301. It can be seen that the noise points are effectively distributed in the upper part of the graph, i.e., for high values of the variable relating to the high “kur” and can be considered outliers which are viewed by white color except the points which are black.

When we tried to compare the clusters with all the categorical attributes, the attribute “Statement” had the highest percentage (49 %) of values which were similar to the clusters formed by our Dbscan Algorithm.

## 3.3.Clustering with agglomerative hierarchical technique

For the application of the agglomerative hierarchical technique we considered the same 12 variables used for the K means and the DBscan. The distance function used will be the Euclidean distance. For problems related to computational complexity, a clusterization with the K Means technique was first carried out, 4 different aggregation methods were considered: full bond, single bond, average bond and Ward's method. The results are shown using the dendograms below. For each of the methods considered, the height at which the cut was chosen was mainly based on the possibility of obtaining 3 clusters. The table below the dendograms shows the results obtained. At first glance, it can be seen that using the single bond method, observing the merging distance, it is very low until the penultimate iteration.

*Fig.12 – Dendograms*



Dendograms for different aggregation methods

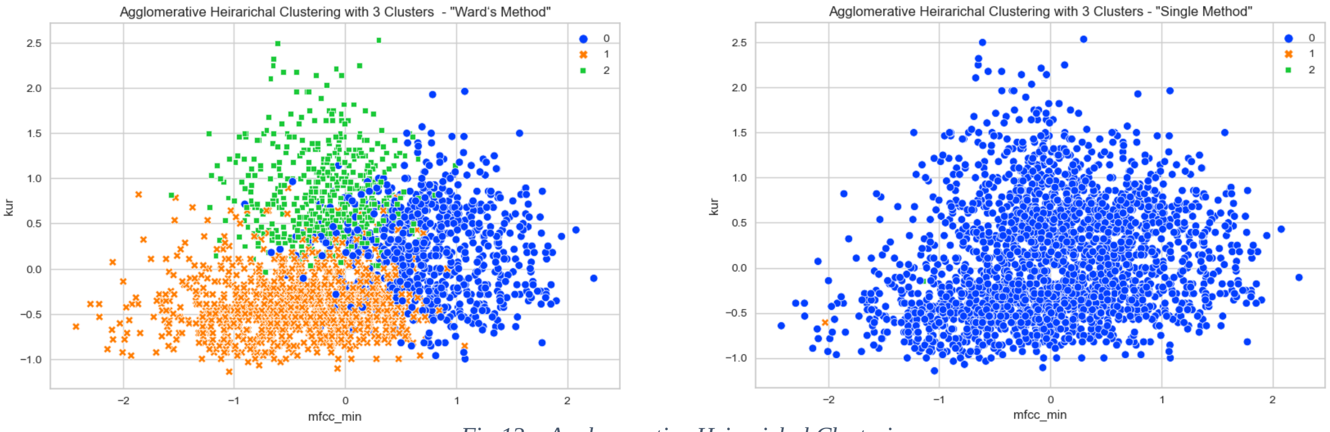
For each of the methods considered, the height at which the cut was chosen was mainly based on the possibility of obtaining 3 clusters. The table below shows the results obtained. At first glance, it can be seen that, using the single bond method, the merging distance is very low up to the penultimate iteration.

|  | Cutting height | Silhouette Coefficient | Predicted Labels |  |  |
|---|---|---|---|---|---|
|  |  |  | 0 | 1. | 2 |
| Complete | 9 | 0.3316340087583796 | 2234 | 15 | 168 |
| Single | 2.15 | 0.3535557732337175 | 2414 | 1 | 2 |
| Group Average | 5.2 | 0.32842667898157857 | 2339 | 77 | 1 |
| Ward‘s Method | 62 | 0.18732693703351677 | 755 | 1099 | 563 |

*Table.3 – Cluster Prediction*

The table shows that, using the single link method, all but two vehicles belong to a single cluster. This can be explained by the fact that the single bond joins the points based on the smallest distance between all pairs. The averaging method allows to obtain highly unbalanced clusters. In fact, clusters 1 and 3 have a very small size compared to cluster 2.

Although the Silhouette score is very high using “single” method, the choice is oriented towards a more balanced clustering albeit with a lower Silhouette score. For this reason, Ward's method was considered. Hence, we prefer the silhouette coefficient 0.18, with the predicted labels {0: 755, 1: 1099, 2: 563}. When we tried to compare the clusters with all the categorical attributes, the attribute “Statement” had the highest percentage (38 %) of values which were similar to the clusters formed by our Dbscan Algorithm


Agglomerative Heirarichal Clustering with 3 Clusters - "Ward's Method"

Agglomerative Heirarichal Clustering with 3 Clusters - "Single Method"

*Fig.13 – Agglomerative Heirarichal Clustering*

## 3.4.Final evaluation of the best clustering algorithm

The table below shows the Silhouette score for the considered techniques. The K-Means technique creates 3 clusters of balanced dimensions but with a low Silhouette score, as well as the by Heirarichal Clustering methods which created highly unbalanced clusters with a high silhouette score. Although the Silhouette score suggests that the best method is that of the single link, we believe that strongly unbalanced clusters as in our case are not a good result. For this reason, our choice was oriented towards the DBscan technique.

*Table.4 – Evaluating Best Clustering Method*

| **Clustering Algorithms** | **No. Of Clusters** | **Silhoutte Score** |
|---|---|---|
| **K-Means** | 3 | 0.24 |
| **DBscan** | 1 | 0.33 |
| **Heirarichal** | 3 | 0.18 |

Ward's method allows to obtain good results but less defined clusters than the K-Means technique. The Silhouette score for the two techniques considered confirms our hypothesis. In conclusion, the DBscan technique, as well as the K-Means

Clustering, are to be considered good techniques to identify clusters. But, by our analysis, the best clustering technique is DBscan because it has the highest Silhouette score, idenitifying balanced clusters and outliers as well.

# 4.Classification:

The purpose of the classification is to form a predictive analysis in order to define the various records of the dataset on the basis of predefined classes. In this section different classification algorithms (K Nearest Neighbors, Decision Tree, Random Forest and Naives Bayes) will be applied using different parameters in order to maximize the performance of the model. It was initially thought to use as a class the variable "**sex**" as intuitively fitting for the purpose of classification, but later it was preferred to opt for another attribute that was not so strongly unbalanced as “sex”. The choice, for the practical purposes of evaluating different algorithms, fell on "**vocal channel"**, transformed into binary values 0 and 1 to indicate “speech” and “song” class membership respectively.

Before testing different classification algorithms, the Features (all features of dataset except target variable) and label(“vocal channel”) was selected and later splitted into training and testing set in 70 : 30 ratio.

## 4.1. K Nearest Neighbours:

The main metrics to be chosen to alter the performance of KNN model are Nearest neighbours and distance metric which was taken as 1 and “Euclidean” respectively. Standard scaler has been used to scale the dataset.

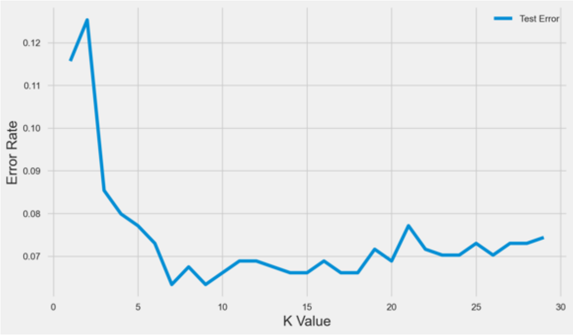
Note: All the evaluation metrics below are evaluated using average = “macro”.

*Table.5 – Evaluation Metrics (Primary Evaluation)*

| Accuracy | **Precision** | **Recall** | **F-1 Score** | **Cross Val Score** |
|---|---|---|---|---|
| 0.889 | 0.880 | 0.882 | 0.881 | 0.154 |

TABLE

*Fig.14 – K-Value vs Error Rate*



Both Accuracy and F-1 Score was good, but the false prediction rate was 11.16%. So,we used “Elbow Method” to plot “K Values” vs “Error Rates” (Root Mean Squared Error metric used) with test data to find the best k neighbour value with least error. We found that the error is least at k = [7, 9, 13]. But, to confirm it mathematically, we did K-Fold Cross Validation using Grid Search(cv = 5) with pipeline to choose the best parameters to get highest performance of the model possible. It was not necessary to manually split the training dataset into training sets and validation sets, as this division is done by the class Grid Search during model selection. From Grid Search, we got the best k value = 13.
After retraining the model with k=13 the results observed were as fo llows,

*Fig.15 – ROC Curve*

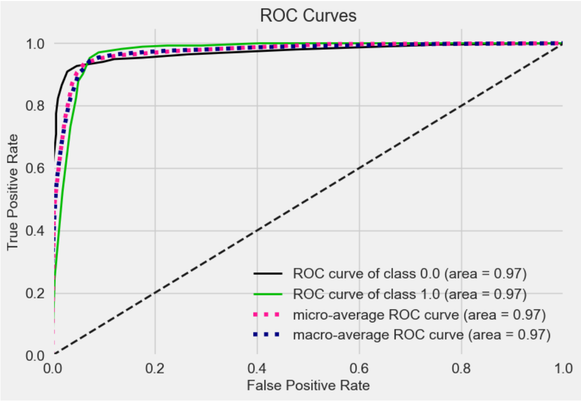
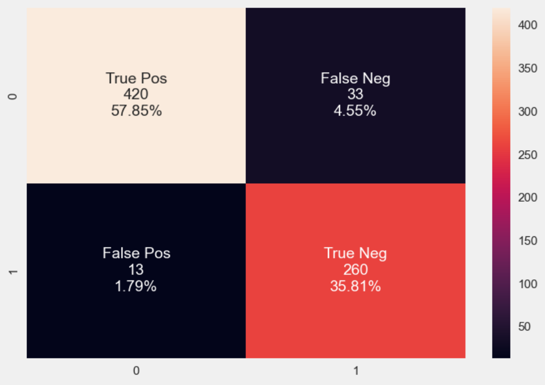


*Fig.16 – Confusion Matrix*



| Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|
| 0.936 | 0.929 | 0.940 | 0.933 | 0.075 |

*Table.6 – Evaluation Metrics (After Hyperparamter Tuning)*

All the evaluation metrics were increased especially F1 Score and Accuracy turned out to be very high indicating the best performance of the model. The best working of the model can also be visualized with the confusion matric wherein only 6.34% has been classified incorrectly which is far better compared to 11.16 % false prediction, which can be clearly visualized in the ROC Curves wherein our KNN model gives ROC curves closer to top left corner indicating a better performance with both high True Positive and False Positive rates.

Finally, thanks to the K Nearest Neighbors it was possible to extract useful information about the importance of the variables of the Data-Set. What emerged is that the frame_count, emotion and sc_mean variables contributed respectively 78%, 6.9% and 6.6% to the classification. All other variables, on the other hand, contributed at most to 2%.

## 4.2. Decision Tree:

The main metrics to be chosen to alter the performance of Decision tree model are maximum depth and split criterion ("gini" or "entropy"). Initially, we chose the default parameters. By primary evaluation we got,

*Table.7 – Evaluation Metrics (Primary Evaluation)*

| Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|
| 0.891 | 0.927 | 0.938 | 0.882 | 0.076 |

The confusion matrix showed false prediction rate to be 10.88%. So, in order to reduce false prediction rate, we did K-Fold Cross Validation using Grid Search(cv = 5) with pipeline to choose the best parameters to get highest performance of the model possible. From Grid Search, we got the optimal max depth value = 4 and best split criterion to be "gini".

After retraining the model with the best parameters found, the results observed were as follows,

*Table.8 – Evaluation Metrics (After Hyperparamter Tuning)*

| Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|
| 0.924 | 0.927 | 0.938 | 0.921 | 0.081 |

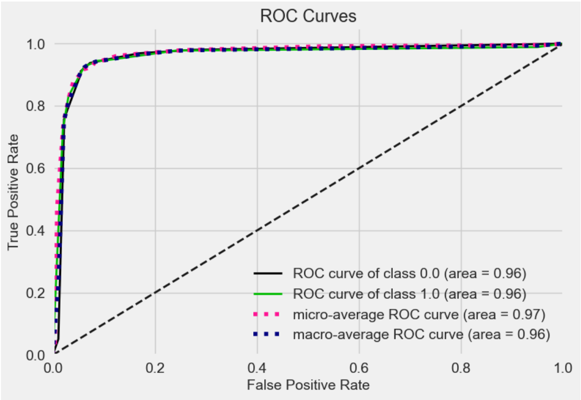


*Fig.17 – ROC Curve*

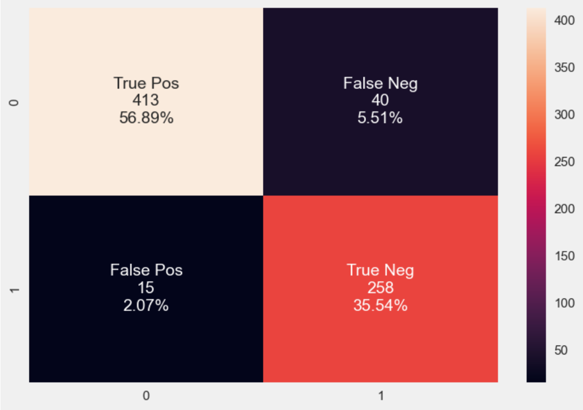


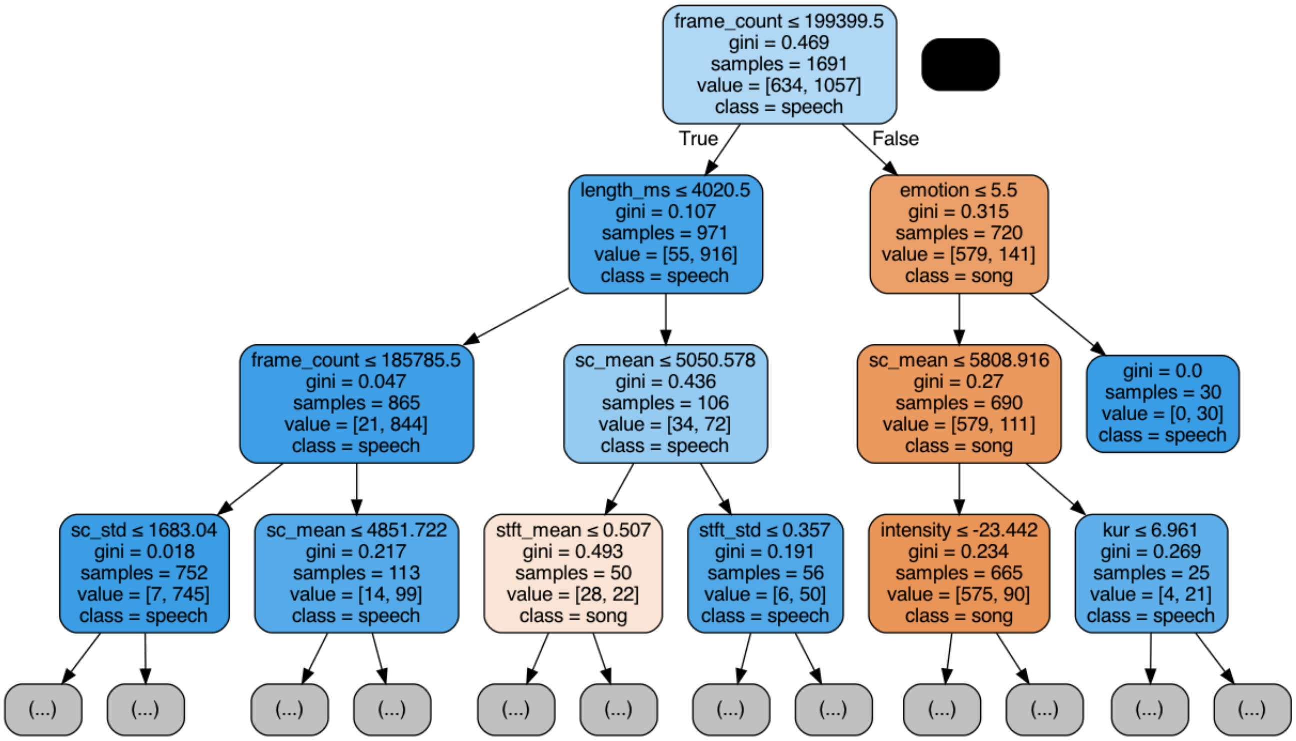
*Fig.18 – Confusion Matrix*

Both accuracy and F1 score was increased and even the false prediction rate was reduced from 10.88% to 7.58% , which can be clearly visualized in the ROC Curves wherein our Decision tree model gives ROC curves closer to top left corner indicating a better performance with both high True Positive and False Positive rates. It should be noted that F1-score represents the harmonic mean between the two previous values and can therefore be takeninto consideration to get a general picture of performance. It was also found from feature importance that only 9 features out of 31 were used to build the decision tree.

### 4.2.1. Interpretation of the tree

The result of the best Decision Tree is shown in Figure . The first node(highest gini=0.469) divides the unbalanced dataset with respect to the length_ms and emotion variables of lower gini than root node. In general, the nodes tending towards blue refer to the speech classification and the nodes tending towards orange to the classification of vocal channel by song. Looking at the Decision Tree it turns out that data with frame_count <= 199399.5(low gini) can be classified as a speech. In particular, when length_ms <= 4020.5(low gini than parent node) and sc_mean <= 0.5050.578(high gini than parent node) are classified as speech. Dataset with just frame_count <= 185785.5 or sc_mean <= 5050.578 arealso classified as speech without further splitting. Even at depth = 3, when frame_count is further splitted both the splits followed are speech which can be pruned if the attributes if the following split contains same value, but in our case its not same as sc_std splits further into [7, 745] where it has 7 values with class speech and 745 values with class song.

*Fig.19 – Decision Tree*



The root Node (frame_count > 199399.5) with emotion <= 5.5 along with and without further split into sc_mean <= 5808.916 are classifies as song. Conversely, the root Node (frame_count > 199399.5) with emotion > 5.5 is directly a terminal node which is classified as speech because it has 30 samples all belonging to speech class.

Finally, thanks to the Decision Tree it was possible to extract useful information about the importance of the variables of the Data-Set. What emerged is that the frame_count, emotion and sc_mean variables contributed respectively 78%, 6.9% and 6.6% to the classification. All other variables, on the other hand, contributed at most to 2%.

## 4.3. Naive Bayes:

We have used all the 3 "probabilistic classifiers" – (Gaussian, Bernoulli and Multinomial) Naive Bayes.

In our analysis among the 3 Bayes classifiers, Gaussian turned out to have high model performance with high F1-score, accuracy, and less false prediction rate. Hence, we will discuss it further and compare the evaluation results of other Naive classifiers at last.

The main metrics to be chosen to alter the performance of Gausian Naive Bayes model are prior probability and variance smoothing. Initially, we chose the default parameters. By primary evaluation we got,

*Table.9 – Evaluation Metrics (Primary Evaluation)*

| Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|
| 0.913 | 0.904 | 0.915 | 0.909 | 0.089 |

All the evaluation metrics was good, in order to further improve performance if possible, we did K-Fold Cross Validation using Grid Search(cv = 5) with pipeline to choose the best parameters to get highest performance of the model possible. From Grid Search, we got the optimal max depth value = 4 and best split criterion to be "gini".

*Table.10 – Evaluation Metrics (After Hyperparamter Tuning)*

| Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|
| 0.913 | 0.904 | 0.915 | 0.909 | 0.090 |

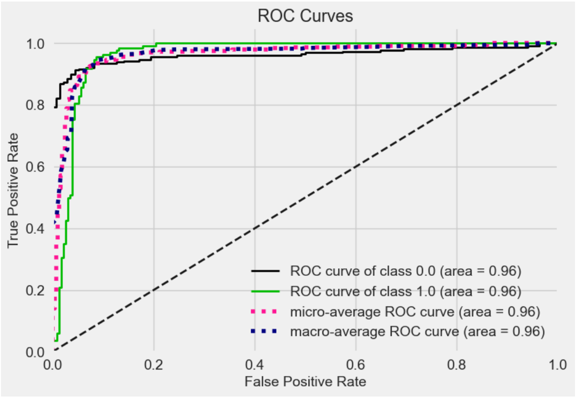


*Fig.20 – ROC Curve*

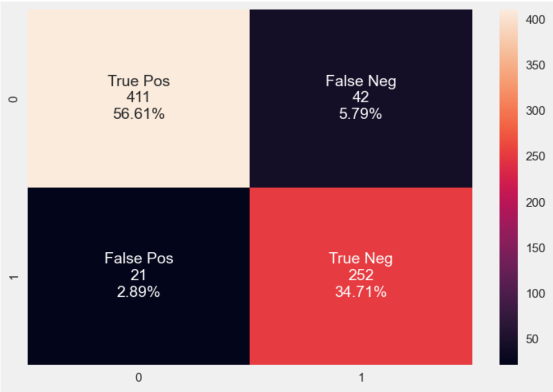


*Fig.21 – Confusion Matrix*

All the evaluation metrics had the same values implying that the default values work better for our model. The false prediction rate was also better which can be clearly visualized in the ROC Curves wherein our Gaussian NB model gives ROC curves closer to top left corner indicating a better performance with both high True Positive and False Positive rates.

Also the other Naive Bayes classifiers (Bernoulli and Multinomial) were performed in the same was as Gaussian NB. Model performance with those classifiers are mentioned below:

*Table.11 – Evaluation Metrics (For other NB Types)*

| NB Types | Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|---|
| **Bernoulli NB** | 0.909 | 0.910 | 0.895 | 0.901 | 0.104 |
| **Multinomial NB** | 0.879 | 0.890 | 0.852 | 0.865 | 0.132 |

## 4.4. Conclusion:

*Table.12 – Evaluating best classification method*

| Classifiers | Accuracy | Precision | Recall | F-1 Score | Cross Val Score |
|---|---|---|---|---|---|
| **KNN** | 0.936 | 0.929 | 0.940 | 0.933 | 0.075 |
| **Decision Tree** | 0.924 | 0.927 | 0.938 | 0.921 | 0.081 |
| **Gaussian NB** | 0.913 | 0.904 | 0.915 | 0.909 | 0.090 |

With our analysis, we found that K-Nearest Neighbor classifier turned out to be the best classification algorithm as it shows high performance in classifying our target variable "vocal channel" than all other algorithms by not only having high values for all evaluation metrics (F1 score, accuracy, precision and recall), but also it has the highest True Prediction rate (true positives and true negatives rates).

# 5. Pattern mining

For this section it was decided to discretize the variables through the pandas "**qcut"** function, thus creating frequency classes based on the quartiles of the distribution of each variable. We decided once again to use the following eight variables **'vocal_channel', 'emotional_intensity', 'max', 'statement', 'frame_count', 'sex', "kur"** and **'stft_std'** already used in the clustering section.

## 5.1 Extraction of frequent patterns and analysis of the number of patterns with respect to the MinSup parameter :

We tried to extract frequent patterns with different support values and different typologies (frequent, closed and maximal). Extracting the most frequent itemset it is immediately evident that using a low support (0.02) the first itemset of the list (a large list of 2683 itemset) all contain the variable *vocal_channel* with value "speech". By raising the support to 0.15, the list is reduced (11 itemset) and all the items has the value "Speech".

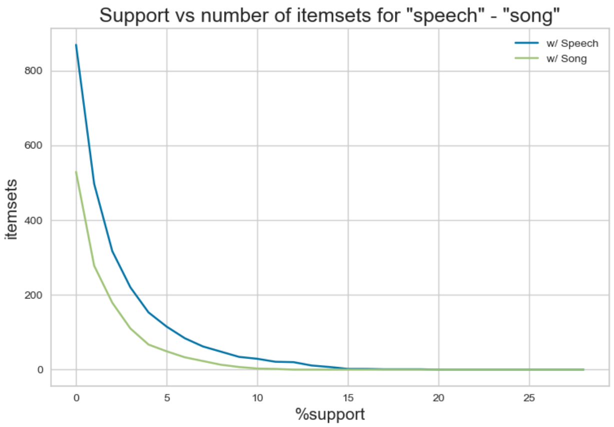


*FIGURE .22.-TREND OF TOTAL FREQUENT ITEMSETS AND CLOSED ITEMSETS.*

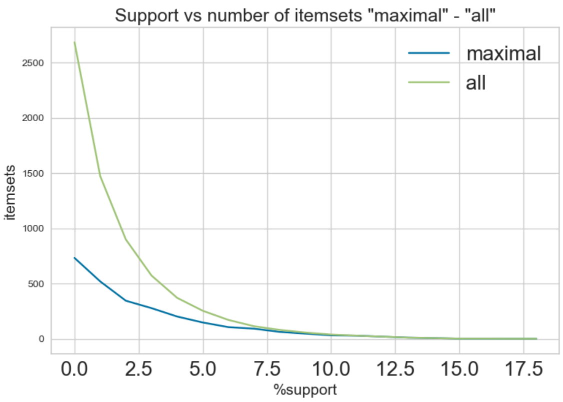


*FIGURE.22-TABLE FREQUENT ITEMSETS WITH SUPPORT*

*Table.10 – Table Frequent Itemset (Support vs Useless Itemset)*

| Support (Supp) | Number of Frequent Itemset | Value "speech" | Value "song" | Useless Itemset |
|---|---|---|---|---|
| **Supp <= 11** | 2626 | 869 | 529 | 1228 |
| **Supp > 11** | 57 | 34 | 7 | 16 |
| **Supp > 14** | 20 | 20 | 0 | 0 |

For support less than 0.11, only 53% of the total frequent patterns observed has the target values necessary, for support greater than 0.11, 72% of the total frequent patterns observed has the target values necessary. When the support is further increased to 0.14, we have only 20 frequent patterns observed out of which all the target value is only "speech" which indicates that all the patterns observed with support 0.14 will be "speech". This tendency, in fact, is also found by observing the number of **maximal itemset** and **closed itemset** of the two values of the variable as the support increases, as shown in figure 1.

Some of these itemset with support greater than 0.14 are given below to comment on them. (20 itemset)

*Table.12 – **FREQUENT ITEMSETS WITH SUPPORT***

| PATTERN | | | SUPPORT |
|---|---|---|---|
| (14.095, 59.086]_Kur | M | speech | 14.108399 |
| Dogs are sitting by the door | F | speech | 15.308233 |
| Dogs are sitting by the door | normal | speech | 16.425321 |
| (0.209, 0.318]_StftStd, | M | speech | 21.348779 |

Most patterns with support greater than 0.14 are within the range (0.14 - 0.16). All the patterns observed has rules for either "speech" or "song".

The first of the patterns reports that high kurtosis, sex being "Male" with the vocal channel being "speech". We could visualize negative linear correlation between kur, sex and vocal channel implying that if the sex was Female and kur was still high, there is high possibility of vocal channel to be "song".

The second pattern reports with statement "Dogs are sitting by the door", sex being "Female" with the vocal channel being "speech". There is almost 0 correlation with statement, sex and vocal channel. So, even if statement changes, vocal channl might still be speech, but sex has a small negative correlation almost close to 0 which still has the possibility of changing the vocal channel value.

The third and fourth patterns again confirm some of the evidence found in the data preparation phase. of non-polysemy. Vice versa, lowering the threshold of support, we find the itemset below, in which we find the same variables with opposite values. As the support is lower, the same variables in the 1st pattern observed earlier with opposite values (lowest kurtosis, sex being "Female", hence ther vocal channel with this patter is observed is "song" instead of "speech".

*Table.13 – Pattern vs Support*

| PATTERN | | | SUPPORT |
|---|---|---|---|
| 1.7570000000000001, 6.53]_Kur | F | song | 9.722797 |

*('(1.7570000000000001, 6.53] _Kur', 'song', 'F'), 9.722796855606124),* Most patterns with support lesser than 0.14 are within the range (0.02 - 0.06). Not all the patterns observed has rules for either "speech" or "song". It could be analyzed from the table Frequent Itemset (Support vs Useless Itemset). The first of the patterns reports that of "*kur*", sex being "Male" with the vocal channel being "speech".

## 5.2 Extraction of associative rules for different values of MinConf :

### 5.2.1.RULES ACCORDING TO THE PERCENTAGE OF CONFIDENCE AND SUPPORT

Regarding the extraction of association rules, it was decided to use 0.1 as the support threshold, 60 as the minimum confidence value and with a minimum number of 3 sets per set to avoid trivial rules by inserting too low thresholds.With these parameters we find 301 rules, moreover, as shown in the graph in Fig.1 Rules according to Confidence and Support, using a higher threshold of confidence the number of rules found would drop significantly. We also point out that wanting to search for rules with 4 item sets and the same other parameters, we find only more 5 rules. Observing the histograms of the lift (Fig.1 ), it can be observed that after the peak around 1.8 there are few rules with a high lift, as regards the confidence (Fig.2 ), instead, it is underlined how a low negative correlation characterizes many rules (lift values less than 1.6). Moreover, it is possible to notice some rules with a lift value greater than 1.6, indicating a positive correlation between the terms of the rule.

*FIG.23- LIFT HISTOGRAM*

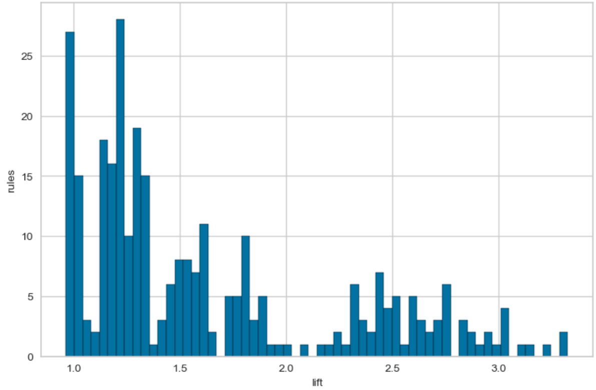
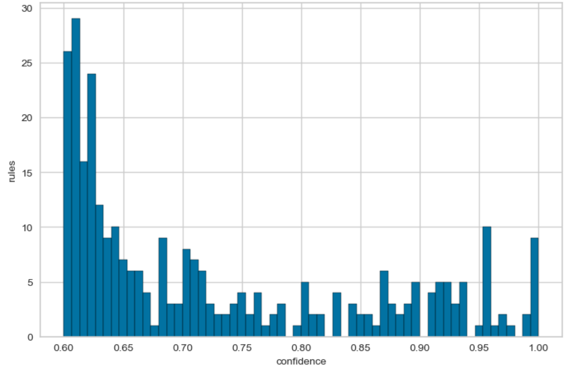


*FIG.24-CONFIDENCE HISTOGRAM*



Some rules are given below to comment on the semantics and any matches with other sections of the project.

*Table.14 – Rules vs Pattern Mining Metrics*

| RULES | ABS_SUPP | SUPP% | CONF | LIFT |
|---|---|---|---|---|
| 'speech',<br>'(172973.0, 192192.0]_FrameCount') | 533 | 22.052131 | 0.882450 | 1.796868 |
| 'speech',<br>('(140940.999, 172973.0]_FrameCount', 'Dogs are sitting by the door' | 341 | 14.108399 | 0.991279 | 1.586703 |
| 'speech',<br>('(0.209, 0.318]_StftStd', 'Kids are talking by the door', 'M') | 259 | 10.715763 | 0.935018 | 1.496648 |
| 'speech',<br>('(0.225, 0.999]_Max', 'strong', 'Kids are talking by the door') | 153 | 6.330161 | 0.604743 | 0.967989 |

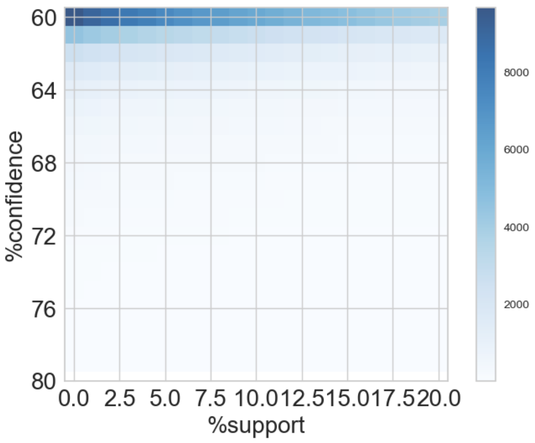


*Fig.25 – Support% vs Confidence%*

Regarding the extraction of association rules, it was decided to use 0.1 as a support threshold, 60 as a minimumconfidence value and with a minimum number of 3 sets per item set to avoid trivial rules by inserting thresholds that are too low.

The reported rules are to be read in this way Y » X and, for each of them, the values of absolute support (number of transitions), support in percentage, confidence and lift have been reported. Looking at the value of lift, which indicates the ratio between the support observed and that expected if X and Y were independent (confidence normalized with the support of Y), we could immediately discredit the rules in which this value is close to 1, because it indicates that the probability of the antecedent and the consequent are independent, so they could be associated randomly.

Regarding the first rule, in which a high lift is observed, with low frame count is present again (a trend already seen), all the trends in This pattern observed are positively correlated, hence if any of the trend increases, or changes, there is high possibility of vocal channel to be "speech". In the first rule as the antecedent trends both are positively correlated and has high confidence 0.88, high lift above 1.75 along with very high ABS support as well. Therefore, it implies that there is high chance for consequent ("vocal channel") to be "speech" rather than it being non-vocal channel value. According to our previous discussions, for support above 0.14, consequent can have only value mostly ”speech” or non-vocal channel value and not “song” at any case, which is hereby verified. Regarding the second rule, when the fame count is so high with the statement as “Dogs are sitting by the door”,high lift is observed with low support, altogether implying it to have high possibility to be a song.

The last rule is one of the most interesting, since it contains more trends already found previously, moreover, the confidence is very less with 0.6, the lift value less than one indicates that, if a “max” is high with “emotional intensity” being strong, “statement” being “Kids are talking by the door”, there is high possibility of vocal channel to be “speech”. In this rule as the antecedent trends(max) is positively correlated with all other trends, as the other trends are negatively correlated with a value very close to zero, so it considered to be positively correlated, and has lift less than 1, confidence less than 0.6 and support 6.3% and the ABS support is very low below 150. Therefore, it implies that there is low chance (14.28%) for consequent (“vocal channel”) to be “speech”, consequent has high probability (38%) of being “normal” (emotional intensity value) but has 0% chance for consequent to be in “song”. Hence, this rule is not a good to predict vocal channel target variable.

## 5.3. Predicting Vocal Channel (Classification):

The choice of the target variable that we decided to predict fell on the vocal channel variable, since it is the only binary variable in the dataset. Observing the association rules where this variable appears at the Y, however, we immediately notice that with the parameters used we found had rules with a negative value of the variable, a factor due to the unbalance of the dataset.

In addition, looking at the lift values of the rules where the variable appears at the Y with a positive value, there are predominantly linear correlations or independence conditions. Therefore, for the positive value, it was decided to carry out an interpretation workof the rules based on the lift value. For the positive value, on the other hand, a rule was first sought by loweringthe support, coming to no result. Lowering, instead, also the confidence value up to 0,4 % finally appears the following rule:

The choice of the target variable that we decided to predict fell on the vocal channel variable. Observing the association rules where this variable appears at the Y, however, we immediately notice that with the parameters used we cannot find any rule with a positive value of the variable, a factor due to the unbalance of the dataset. In addition, looking at the lift values of the rules where the variable appears at the Y with a negative value, there are predominantly negative correlations or independence conditions. Therefore, for the negative value, it was decided to carry out an interpretation workof the rules based on the lift value. For the positive value, on the other hand, a rule was first sought by increasingthe support and also the confidence value up to 0,8 % finally appears the following rule which is opposite in values with the second rule observed above which has low lift, high confidence, high support with lowest frame count and same statement as the rule mentioned below. With just changing frame count‘s value from high to low, the class can be easily distinguished as speech or song.

*Table.15 – Rule for further classification*

| RULES | ABS_SUPP | SUPP% | CONF | LIFT |
|---|---|---|---|---|
| 'song',<br>'(217818.0, 305906.0]_FrameCount', 'Dogs are sitting by the door' | 254 | 10.508895 | 0.888111 | 2.366666 |

In fact, all instances with vocal channel as “speech” report with lowest frame and highest kurtosis as per the analysis. Applying what has been described, let’s check it with the rule mentioned above wherein the frame count is high and the kurtosis is the lowest, opposite to “speech” predicting pattern implies these rules to be classified as “song”.

Thus, the results obtained are as follows:

*Table.16 – Confusion Matrix*

| Prediction | Speech | Song |
|---|---|---|
| Speech | 1435 | 73 |
| Song | 146 | 763 |

The accuracy of the model is 0.857.

The F-measure of the model for predicting **Speech** is 0.871 while the F-measure for predicting **Song** is 0.829. These results were obtained using the whole dataset, to compare them with the results of a classification algorithm we divided the dataset into a train part and a test part (through a random splittingas for the classification) and applied the rules on the test set. Here are the results:

The accuracy of the model is 0.873.

*Table.17 – Classification Evaluation with metrics*

| **Prediction** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|
| **Speech** | 0.86 | 0.96 | 0.91 |
| **Song** | 0.92 | 0.74 | 0.82 |

In the end, it is observed that the classification algorithm achieves better results than the association rules based model, except for the precision and recall values of the positive value in the test set. Our model achieves higher results in classifying vocal channel from the Decision Tree when tested on the sametest set. Nevertheless, the accuracy of Decision Tree is still higher than the other model.

# 6.Regression:

Regression analysis is a reliable method of identifying which variables have impact on a topic of our interest. The process of performing a regression allows us to confidently determine which factors matter most, which factors can be ignored, and how these factors influence each other.

Before testing different regression algorithms, all the Features was selected and later splitted into training and testing set in 70: 30 ratio.All the regression algorithms were fitted with the same training sets. And were predicted using the respective regressors. Further to improve the performance, we used K-Fold Cross Validation to get the best hyperparameters. And, we have used the evaluation metrics namely (Mean Absolute Error, Mean Squared Error, Root Mean Squared Error) to measure the model‘s performance.

Accuracy (e.g. classification accuracy) is a measure for classification, not regression. We cannot calculate accuracy for a regression model. Hence, the skill or performance of a regression model must be reported as an error in those predictions as mentioned below:

*Table.18 – Evaluating Best Regression Method*

| **Regressor** | **MAE** | **MSE** | **RMSE** |
|---|---|---|---|
| **Linear** | 1.944 | 7.491 | 2.737 |
| **Lasso** | 2.352 | 9.807 | 3.132 |
| **Ridge** | 1.950 | 7.378 | 2.716 |
| **KNN** | 2.307 | 6.790 | 2.604 |
| **Decision Tree** | 1.752 | 8.466 | 2.910 |

## 6.1Conclusion:

We have considered RMSE as the error metric to choose the best model because all the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable. According to our analysis, we found K-Nearest Neighbor Regressor to be the best model with lowest Root Mean Square Error.