**Medical Symptoms Audio Classification Report**

**Introduction**

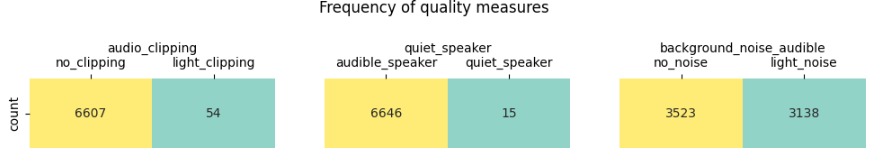
This project focuses on using machine learning techniques in order to classify medical symptoms based on self-reported audio recordings, with the goal of improving conversational agents in the medical field. The dataset comprises thousands of audio snippets, totaling over 8 hours of recording time, these recordings were created through a multi-step process where contributors first provided textual descriptions of symptoms, followed by recording corresponding audio. However, challenges such as incorrect labels and poor audio quality necessitate thorough data cleaning before model training. By addressing these issues, the project aims to create robust models that can accurately classify medical symptoms, thus enhancing the efficacy of conversational agents and improving healthcare diagnosis.

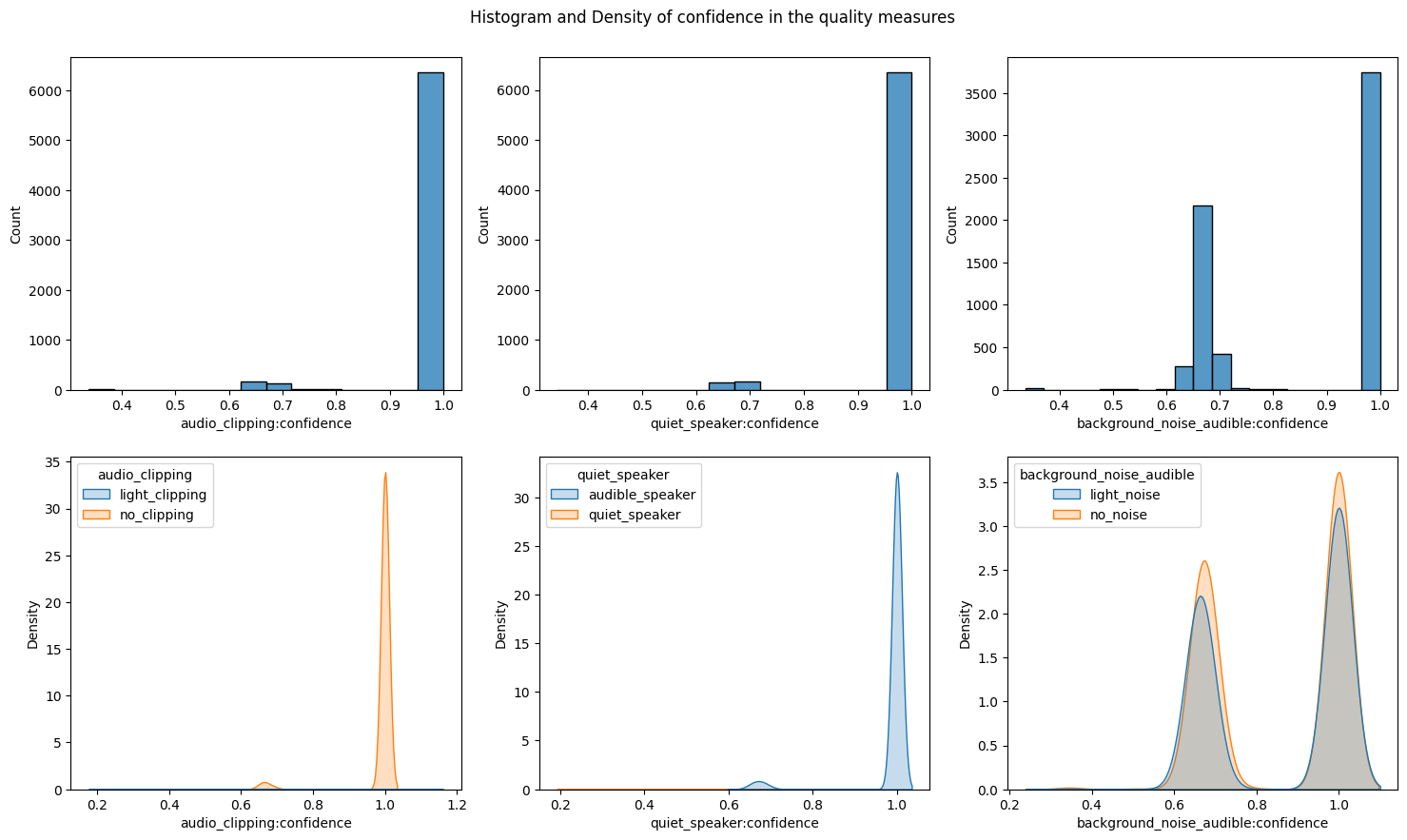
The research questions are:

* What is the symptom of the patient according to the audio file?
* Are there groups of patients that could be clustered together from the recording and what are their main attributes?
* Which model classify better, audio file classification model or NLP model of the audio files transcriptions?

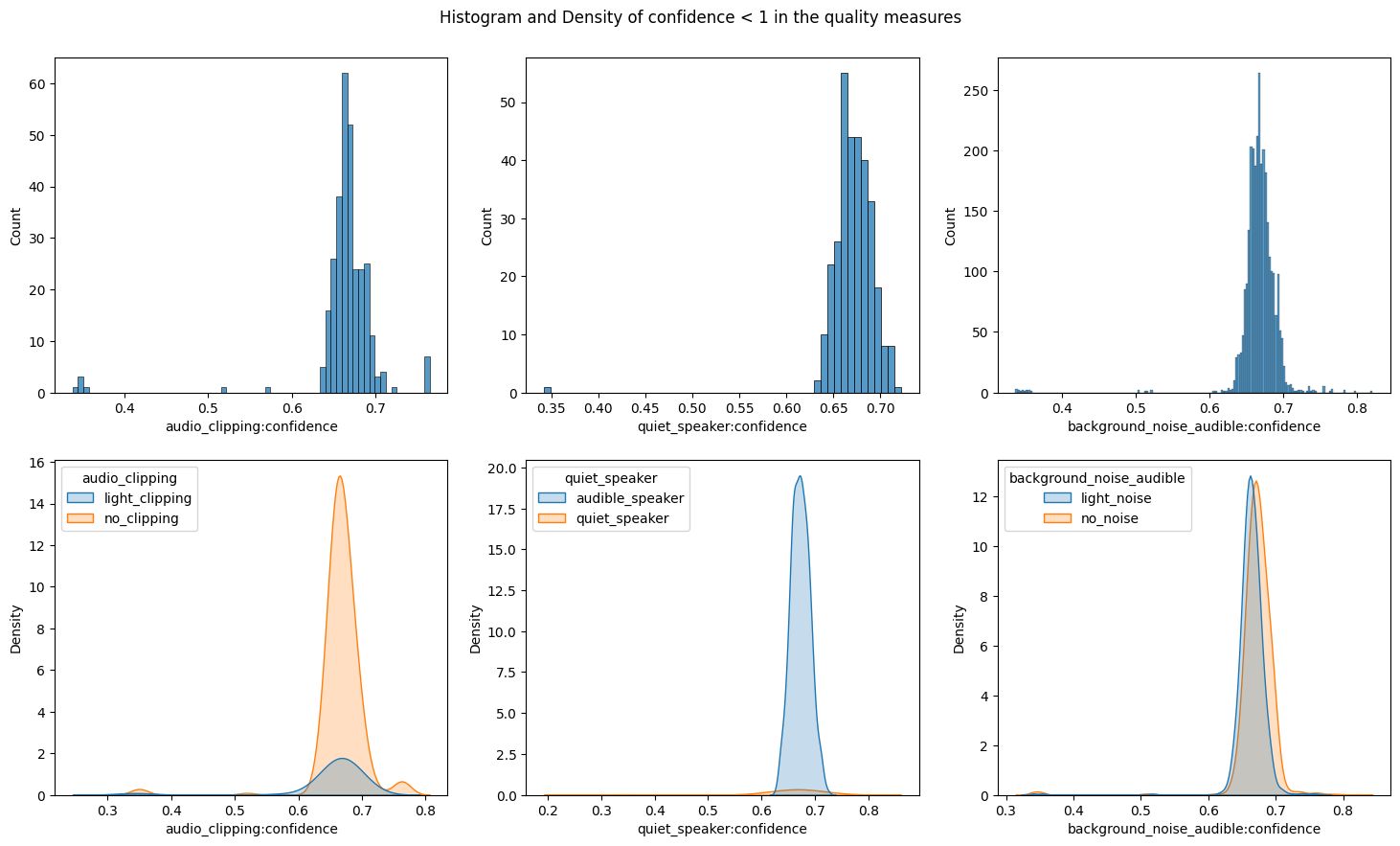
The dataset contains 6661 audio recordings and a corresponding CSV file, where each recording has 3 indicators of audio quality measures (audio clipping, background noise audible and quiet speaker) with confidence score for those indications (between 0 and 1), an overall numeric score between 3 and 5 of the quality of the audio which is uncorrelated with the other audio quality measures. Furthermore, for each audio recording the CSV file contains the transcription of the audio, the writer ID, the speaker ID and the label which is one of 25 medical symptoms. In order to develop the NLP model, I used the English language dictionary.

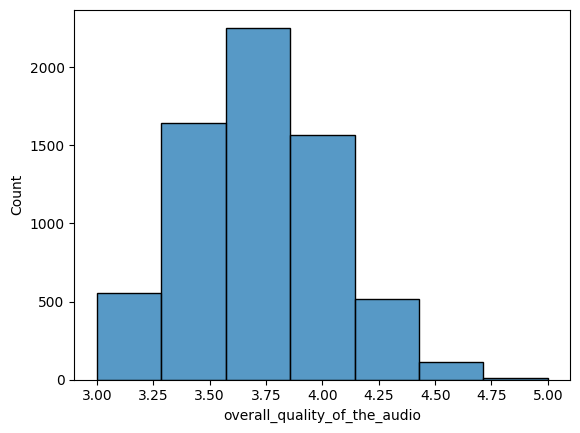
**Exploratory Data Analysis**

First I explored the audio quality meausre:

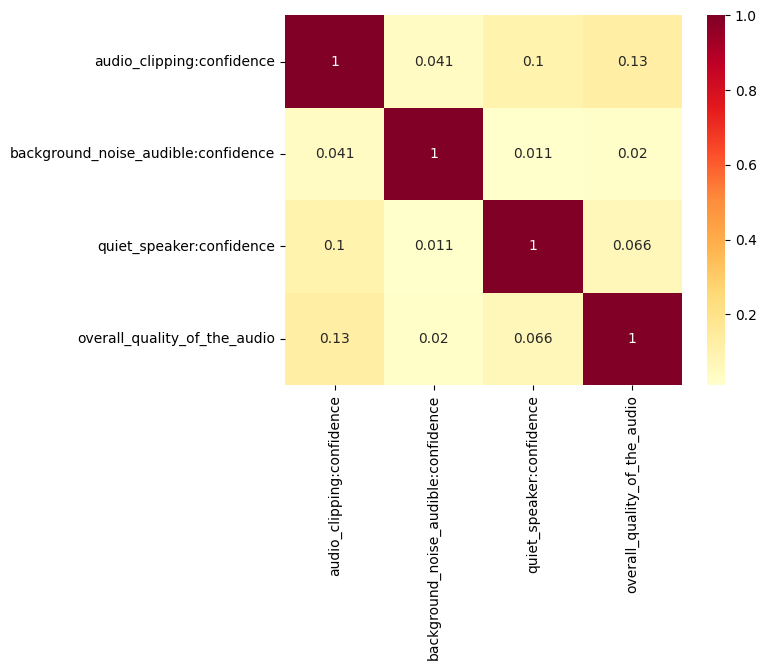
The plot above shows the frequency of each audio quality measure, we can see that most of the recordings have no clipping, the speaker is audiable and not quiet and half of the recordings have light noise in the background and half have no noise.

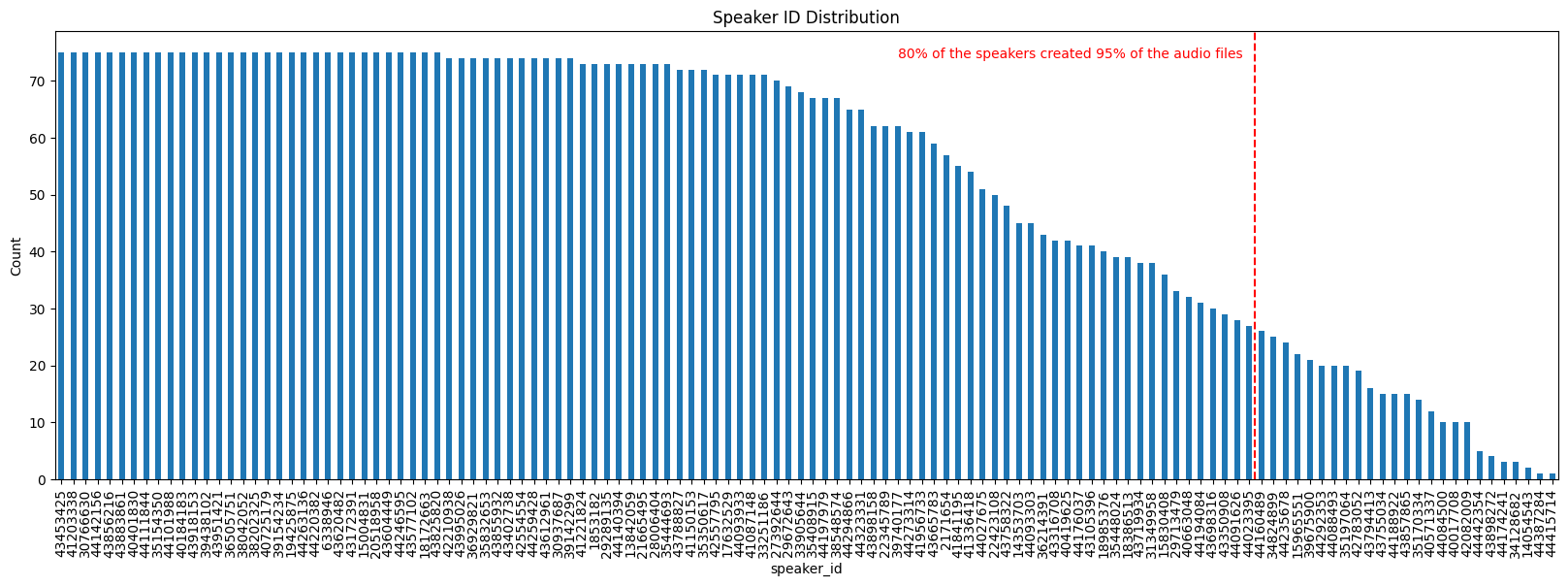
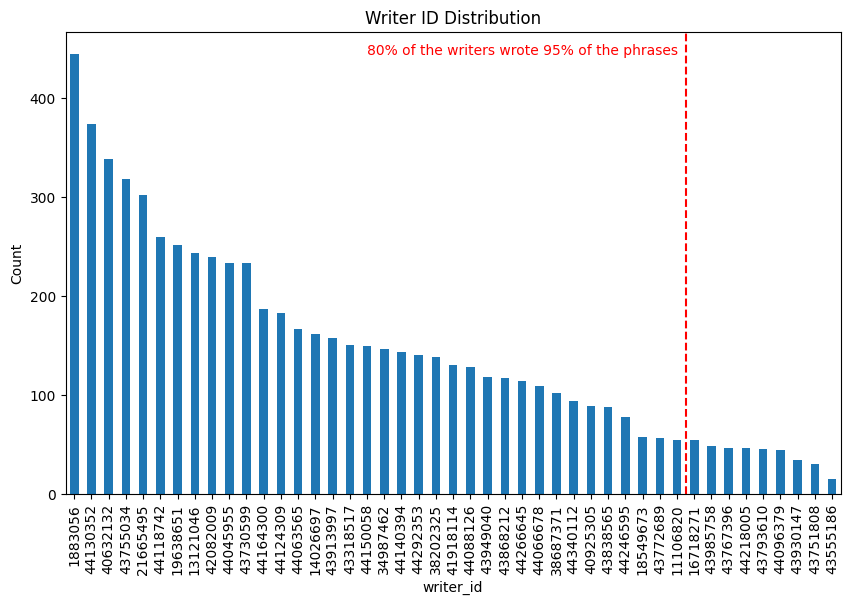
The plot above shows the histogram of the confidence score of each quality measures in total and by each level. We can see that for most of the recordings the confidence scores in the audio clipping and quiet speaker measures are 1, and in the background noise there is a noticable difference, but it is hard to determine if the different levels behave the same. So, I will look at all the confidence scores less than 1.

we can see that for confidence score less than 1, the recording audio quality confidence scores are around 0.667. We can see that the confidence score for the quiet speaker feature doesn't give any interesting information as most of the recordings with quiet speaker has confidence 1.

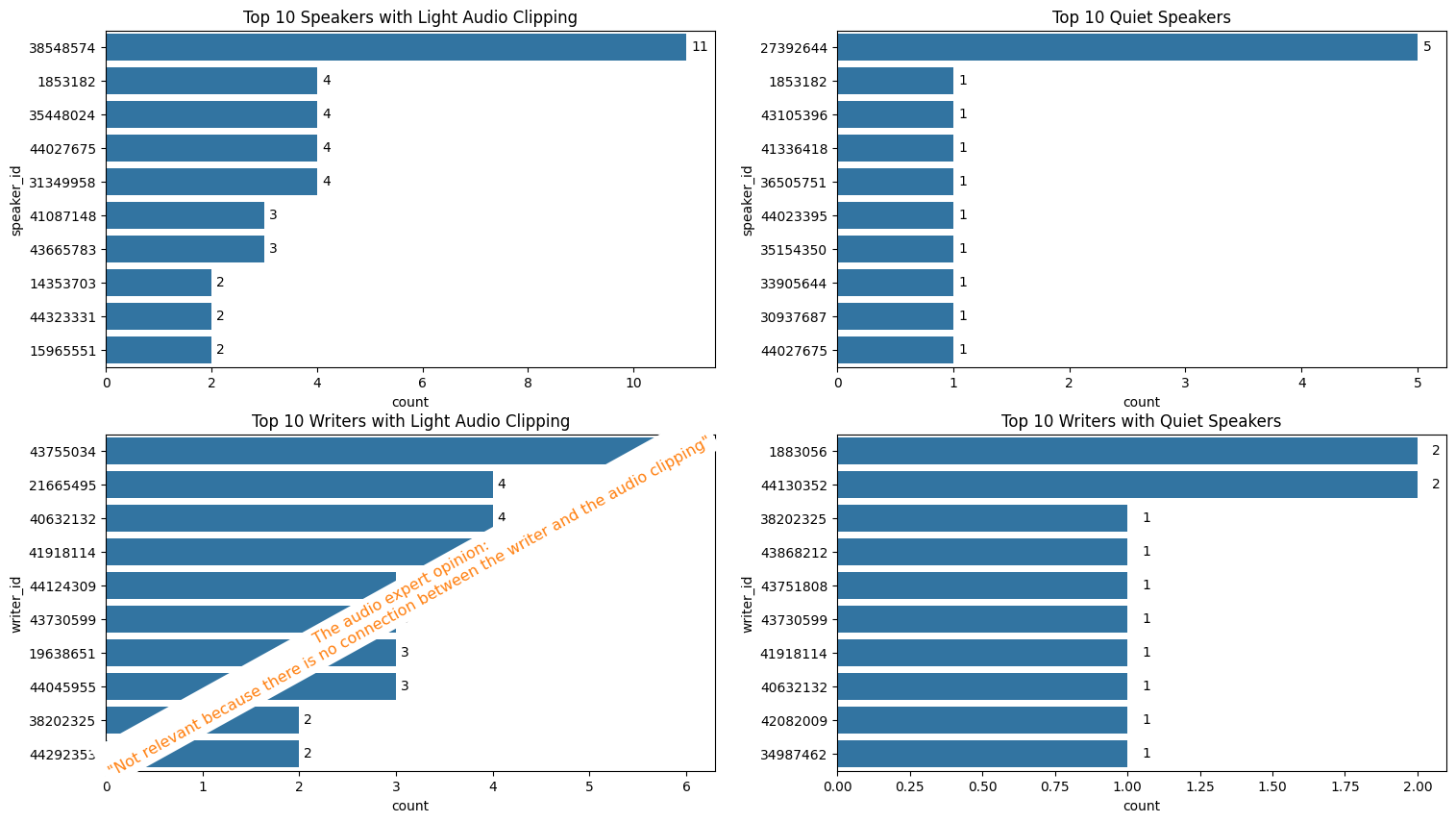


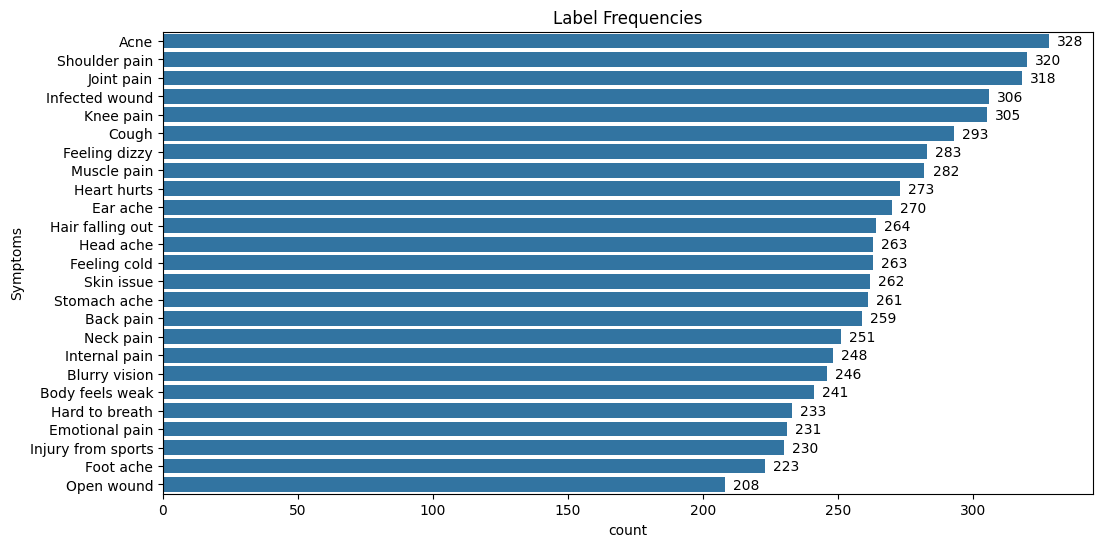
In the plot above we can see the histogram of the overall quality score of the recordings, the avereage score is 3.678 with standard deviation of 0.379, we can see somewhat of a normality in the distribution of the feature.



Secondly, I looked at the speaker ID and writer ID features

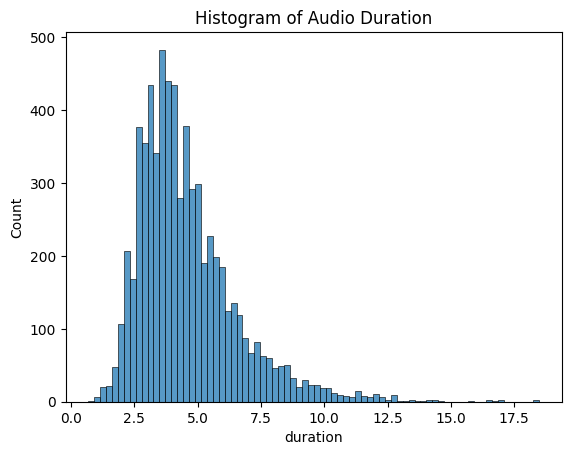
We can see that 80% of all the speakers created 95% of the audio files and 80% of all the writers wrote 95% of the phrases. Then I tried to see if there is any correlation between the audio quality and the speaker or the writer.

We can see that speaker 38548574 had the most light audio clipping while he was speaking, and speaker 27392644 had the most audio recording with him speaking quietly. In regards to the writers, the audio expert could say to us that there is not relationship between them and the quality of the audio recordings.

Third, I looked at the label frequencies

We can see that the data is balanced and doesn't need SMOTE (Synthetic Minority Oversampling Technique).

Lastly, I looked at the audio files themselves



In the plot above, we can see the histogram of the lengthes of the audio recordings. We can see that the audio recordings are differ in their length and we would need to make them in the same dimensions.

**Feature Engineering**

The first features I modified were the audio quality measures and their confidence score. I took each measure and assigned to the levels of the indicator and multiplied it by the confidence score, by that the quality measure is now between where values around means bad confidence in the quality measure and around the edges means good confidence in the quality measure.

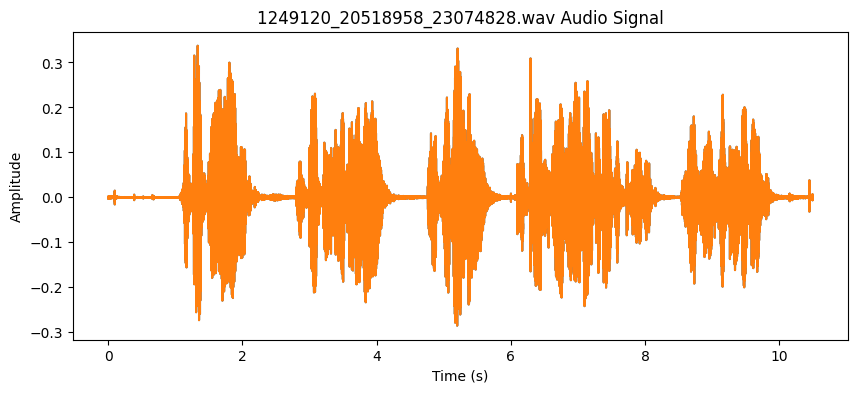
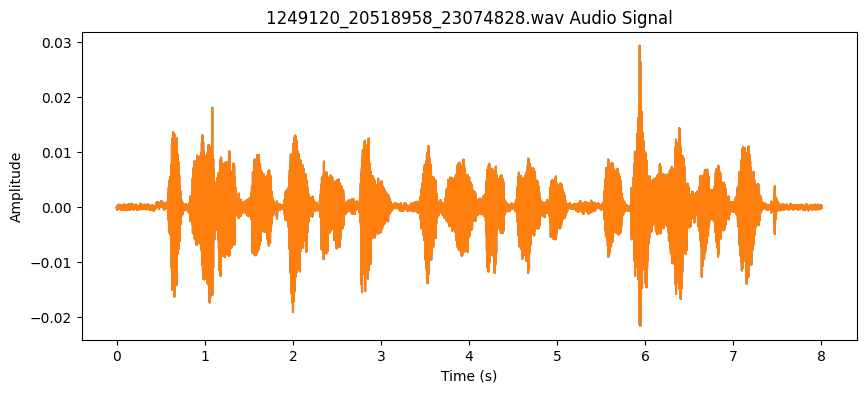
תמונה שמכילה טקסט, צילום מסך, תרשים, מקביל

התיאור נוצר באופן אוטומטי

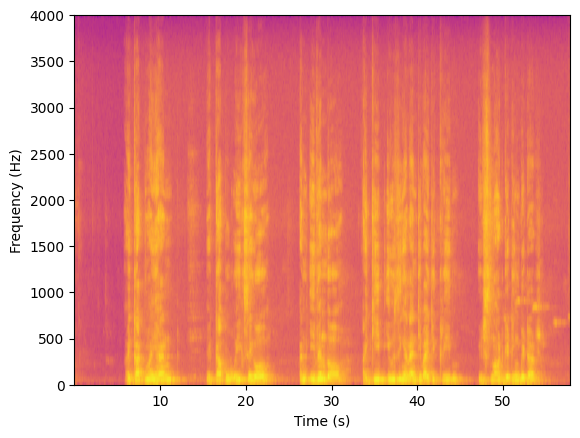
Now, looking at the correlation matrix we can see higher correlations between the new quality measure features to the label but still non significent one.

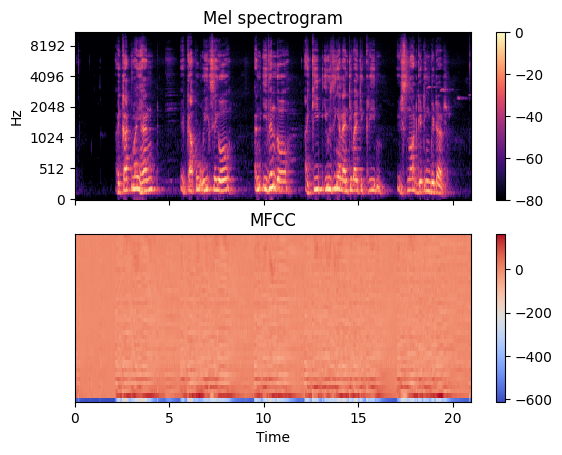
The second modifications I did is to the audio recordings:

* initally I looked at the time series and the amplitude of the wave but since the audio recordings are not in the same length I needed to use more universal methods



* Then I transform it to spectogram



* Then I transform it to Mel-Spectogram and visualize the Mel coefficients
* The final data contains the new quality measures columns, the overall quality score, the audio recording duration and the label.
* I split the data to train and test set (80%-20%).
* I scaled the data by fitting the train set and transforming both train and test.

**Training Models**

The models I chose to predict the symptom are:

* Logistic Regression with penalty (LASSO and Ridge)
* K-NN
* SVM (linear, polynomial and radial kernals)

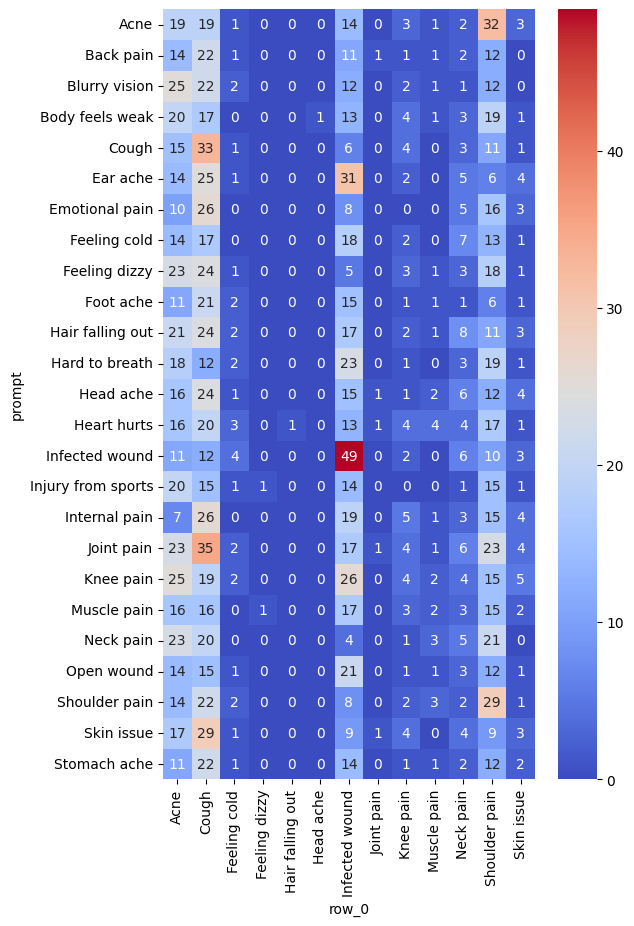
I used GridSearch Cross-Validation with 5-crosses in order to find the best hyperparameters.

### **Logistic Regression**

The Tuning parameters are:

* Panelty ("L1" or "L2")
* C (range from 1 to , by 0.2 jumps)

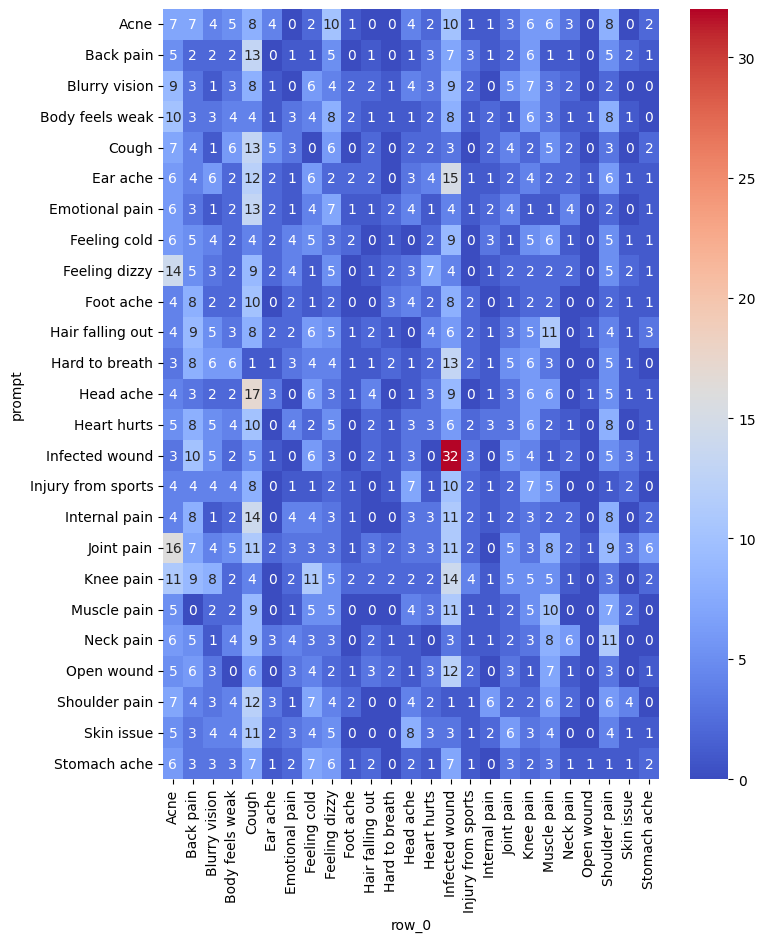
The running time is minutes, the best params are {'C': 3.32, 'penalty': 'l1'}, the train accuracy is 7.2% and the test accuracy is 7.3% (really really bad) and the confusion matrix is



### **KNN**

The Tuning parameters is K (range from 3 to 150, only odd numbers)

The running time is minutes, the best params are {'n\_neighbors': 85}, the train accuracy is 10.3% and the test accuracy is 5.9% (really really bad) and the confusion matrix is



**Future Steps**

* Finish engineering the audio recording features
* EDA on the phrases transcriptions
* Train the models more time with greater range
* CNN – must
* Develop the NLP models
* Use clustring methods to identify the optimal number of groups in the data, and their attributes