Samay - Machine Learning Test

By Jose Daniel Bolaños - ML Engineer

Problem Statement

The primary goal of this technical test is to design, develop, and optimize a machine-learning solution for detecting and classifying respiratory diseases. This will be achieved using a dataset of lung sounds recorded with an electronic stethoscope.

Summary Results

- ➢ After analyzing database annotations it was possible to note that the only clear correlation with "Disease" is the "Sound Type" value, which is why the proposed model is based only on the recorded sounds.
- ➢ The database does not include enough information to predict all diseases. Database data was filtered out taking just information of "asthma", "copd", "'heart failure" and "normal".
- ➢ After waveform, FFT spectrum and MFCC analyzes, it was possible to note that each disease could be described using MFCC features. Besides, the signals with Bell filtering mode (B) does not offer clearly differentiable information between the diseases. Taking into account the above, the model was based on MFCC features from signal with D and E filtering mode. B signals were excluded from the data.
- ➢ Data augmentation techniques were used in order to balance the data between classes, in this case audio transformations like "add_noise", "shift", "pitch_shit" and "stretch" were used to create new signals.
- ➢ After testing several models like SVM, RNN and various CNN, one CNN architecture, originally used for speech emotion classification, which in this case was modified including a couple of layers, reached the best performance with a 98% of accuracy classifying "asthma", "copd", "'heart failure" and "normal" audio signals.

Task Description

Task 1 - Load the dataset and preliminary analysis

The first step in this task is to load the dataset, to do this you can follow the instructions in [1-2] or just follow each section of the attached jupyter file which includes each step description.

Additional to the sound signal, the database includes the file “data annotation.xlsx”, which contains anonymous demographic patient information like age and gender, as well as information about the specific location on the human chest, from where the recording was captured, and also sound_type and disease.

In this case, all data is taken as string and “Location” and “Sound type” values were stripped and spaces were removed. Besides, “Disease” field is converted to lowercase in order to unify all names. Finally, the patient number was included as a new column.

Also it is important to mention that there are null values in the “data annotation.xlsx” file due to some 0 values in column F and G. All null values were removed.

After that a new column is generated “age_range”, in order to see if age range is related with the diseases. The values of this new column are: Age-1: [0 - 20), Age-2: [20 - 40), Age-3: [40 - 60), Age-4: [60 - 80), Age-5: >=80. A snapshot of the first 3 columns of Database annotations is shown in Fig. 1.

	age	sex	location	sound_type	disease	patient	age_range
0	70	M	PLL	IEW	asthma	1	Age-4
1	52	F	PLL	EW	asthma	2	Age-3
2	50	F	PLL	IEW	asthma	3	Age-3
3	72	F	PRL	IC	heart failure + lung fibrosis	4	Age-4

Fig 1. Database Annotations

Several distributions and analyses were made with this data, you can see some of them in the attached jupyter file. The most important one is the “disease” distribution which is presented in Fig.2, where it's possible to note that the database does not include enough information to predict all diseases, that is why this proposed solution is focused on classifying just “asthma”, “copd”, “'heart failure” and “n” (normal) signals.

Also, different visualizations about the relationship between data were made up. One of the most interesting is the relation between “Disease” and “Sound type” which is presented in Fig.3, where It is clearly noted that there is a direct relation between these two values.

In order to validate the last and to find relationships between “Disease” and other values, a correlation process between all values was performed. Before this all string values were represented with integer numbers. The results of correlation are presented in Fig.4.

disease	
n	35
asthma	32
heart failure	18
copd	9
pneumonia	5
lung fibrosis	4
bron	3
heart failure + copd	2
plueral effusion	2
heart failure + lung fibrosis	1
asthma and lung fibrosis	1

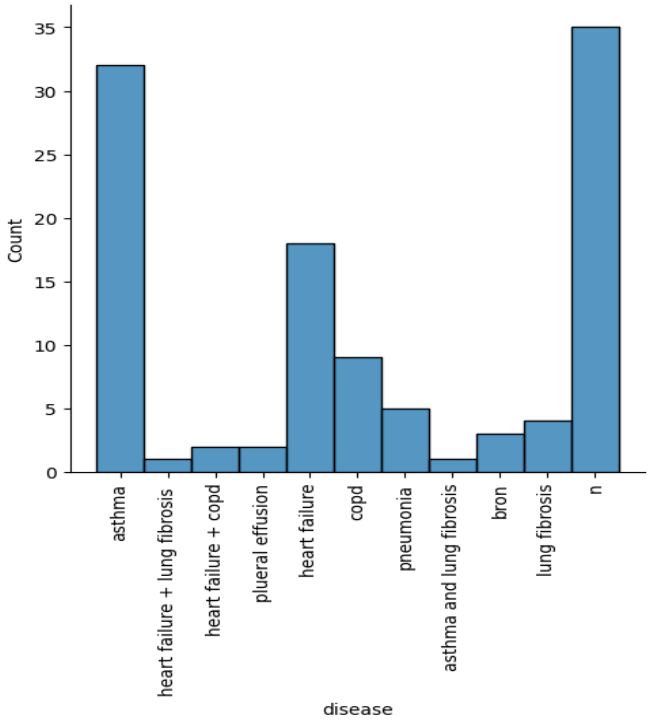


Fig 2. Disease distribution

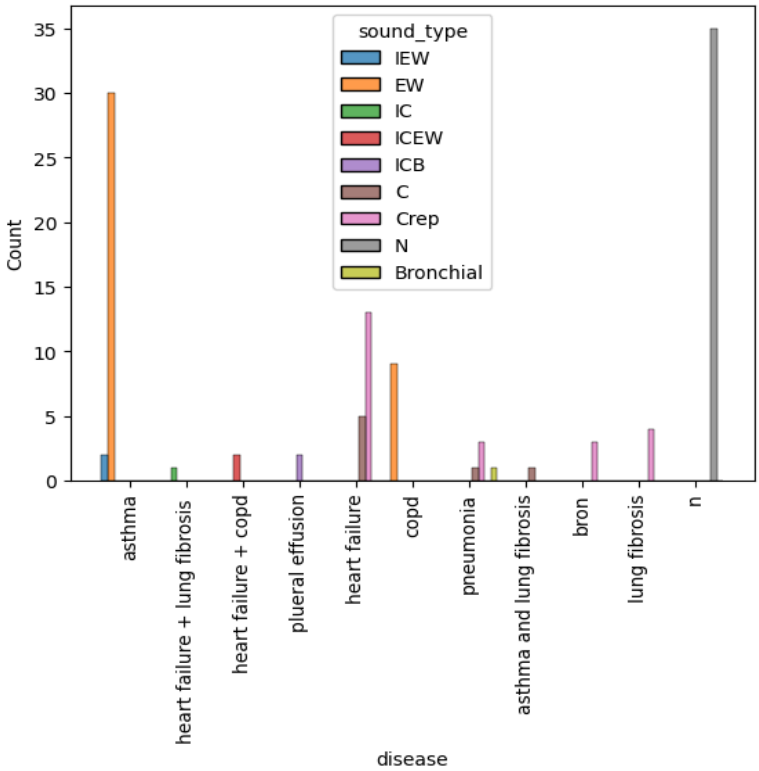


Fig 3. Disease and Sound Type relationship

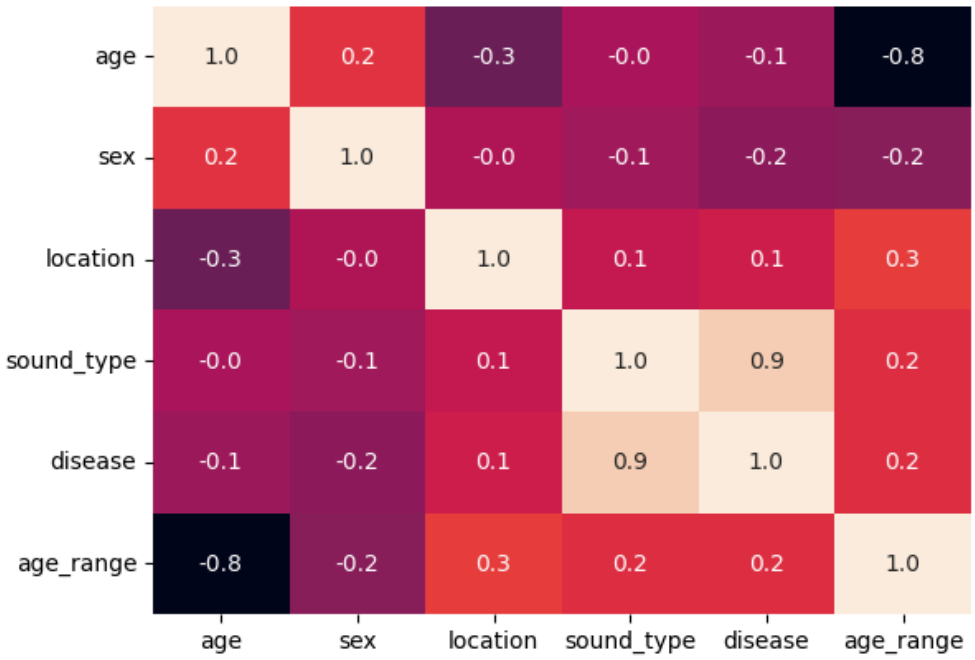


Fig 4. Data Correlation

From Fig 4 it is possible to conclude that the only high correlation is presented between disease and sound type. According to this, the values “age”, “sex”, and “location” are not included in the train data. The proposed model uses only sound signals to perform the classification process.

After this first stage of the process we can see that we have imbalance data. This problem will be handled in the next stage. Also, we can see there are common sound types for each disease, for example, asthma is EW, for heart failure are C and Crep, and for COPD is EW too. This last could be a problem, because it is the same sound type for asthma, but in the next stage it is validated that those two sound types can be perfectly classified.

Task 2 - Signal Processing and Features Engineering

About the sound signals dataset, it includes respiratory sounds from 112 subjects (35 healthy and 77 unhealthy) . The name of each data file starts with the type of filter encoded as letter B, D, or E. This is followed by the letter P, a unique sequential patient number starting from 1, and an underscore. After that, the file name includes the diagnosis, type of sound, location of measurement on chest, subject’s age, and subject’s gender.

In order to group and analyze the sound dataset, filenames are joined to the database annotation using patient number. Three new columns were created “B”, “D”, and “E” to store the

specific filename of each patient according to the filtering mode letter. The results are shown in Fig. 5. Please note that data is also filtered taking just the disease values “asthma”, “copd”, “'heart failure” and “n”

	age	sex	location	sound_type	disease	patient	age_range	B	D	E
0	70	M	PLL	lEW	asthma	1	Age-4	kaggle_data/Audio Files/BP1_Asthma,l E W,P L L...	kaggle_data/Audio Files/DP1_Asthma,l E W,P L L...	kaggle_data/Audio Files/EP1_Asthma,l E W,P L L...
1	52	F	PLL	EW	asthma	2	Age-3	kaggle_data/Audio Files/BP2_Asthma,E W,P L L R...	kaggle_data/Audio Files/DP2_Asthma,E W,P L L R...	kaggle_data/Audio Files/EP2_Asthma,E W,P L L R...
2	50	F	PLL	lEW	asthma	3	Age-3	kaggle_data/Audio Files/BP3_Asthma,l E W,P L L...	kaggle_data/Audio Files/DP3_Asthma,l E W,P L L...	kaggle_data/Audio Files/EP3_Asthma,l E W,P L L...
8	59	M	PRL	EW	asthma	9	Age-3	kaggle_data/Audio Files/BP9_Asthma,E W,P R L ,...	kaggle_data/Audio Files/DP9_Asthma,E W,P R L ,...	kaggle_data/Audio Files/EP9_Asthma,E W,P R L ,...
9	59	M	PRU	EW	asthma	10	Age-3	kaggle_data/Audio Files/BP10_Asthma,E W,P R U,...	kaggle_data/Audio Files/DP10_Asthma,E W,P R U,...	kaggle_data/Audio Files/EP10_Asthma,E W,P R U,...
...	...	...	...	...	...	...	...	...	...	...
107	63	M	PRL	EW	copd	108	Age-4	kaggle_data/Audio Files/BP108_COPD,E W,P R L ,...	kaggle_data/Audio Files/DP108_COPD,E W,P R L ,...	kaggle_data/Audio Files/EP108_COPD,E W,P R L ,...
108	26	M	PLM	N	n	109	Age-2	kaggle_data/Audio Files/BP109_N,N,P L M,26,M.wav	kaggle_data/Audio Files/DP109_N,N,P L M,26,M.wav	kaggle_data/Audio Files/EP109_N,N,P L M,26,M.wav
109	62	M	PLL	EW	copd	110	Age-4	kaggle_data/Audio Files/BP110_COPD,E W,P L L,6...	kaggle_data/Audio Files/DP110_COPD,E W,P L L,6...	kaggle_data/Audio Files/EP110_COPD,E W,P L L,6...
110	51	M	PRL	EW	copd	111	Age-3	kaggle_data/Audio Files/BP111_COPD,E W,P R L ,...	kaggle_data/Audio Files/DP111_COPD,E W,P R L ,...	kaggle_data/Audio Files/EP111_COPD,E W,P R L ,...
111	30	M	PLM	N	n	112	Age-2	kaggle_data/Audio Files/BP112_N,N,P L M,30,M.wav	kaggle_data/Audio Files/DP112_N,N,P L M,30,M.wav	kaggle_data/Audio Files/EP112_N,N,P L M,30,M.wav

Fig 5. Filtered Database Annotations with Filenames

Next, one filename by disease and filtering mode is taken in order to analyze the audio waveform, FFT spectrum and MFCC features. An audio waveform is a graph that displays amplitude or level changes over time, while FFT is a mathematical function that transforms an audio signal to a representation in the frequency domain. Finally, Mel Frequency Cepstral Coefficients (MFCCs) is a way of extracting features from an audio. The MFCC uses the MEL scale to divide the frequency band to sub-bands and then extracts the Cepstral Coefficients using Discrete Cosine Transform (DCT). MFCC features describe the overall shape of the spectral envelope and are frequently used for voice recognition. [3][4]. The results of this analysis are shown in Fig 6, 7, 8, 9. Visualization python algorithm was created taking some tips from [5][6].

From the waveform it is possible to see that data sound is already normalized between 1 and -1 values and there are notable differences between diseases but not in filtering mode.

From FFT spectrum, it is possible to note that FFT information is quite similar between asthma and heart failure, and could not be enough to effectively differentiate the diseases.

Finally from MFCC-mean and MFCC-raw, it is possible to note clear differences in features from different diseases in D and E filtering modes, but not for B filtering mode.

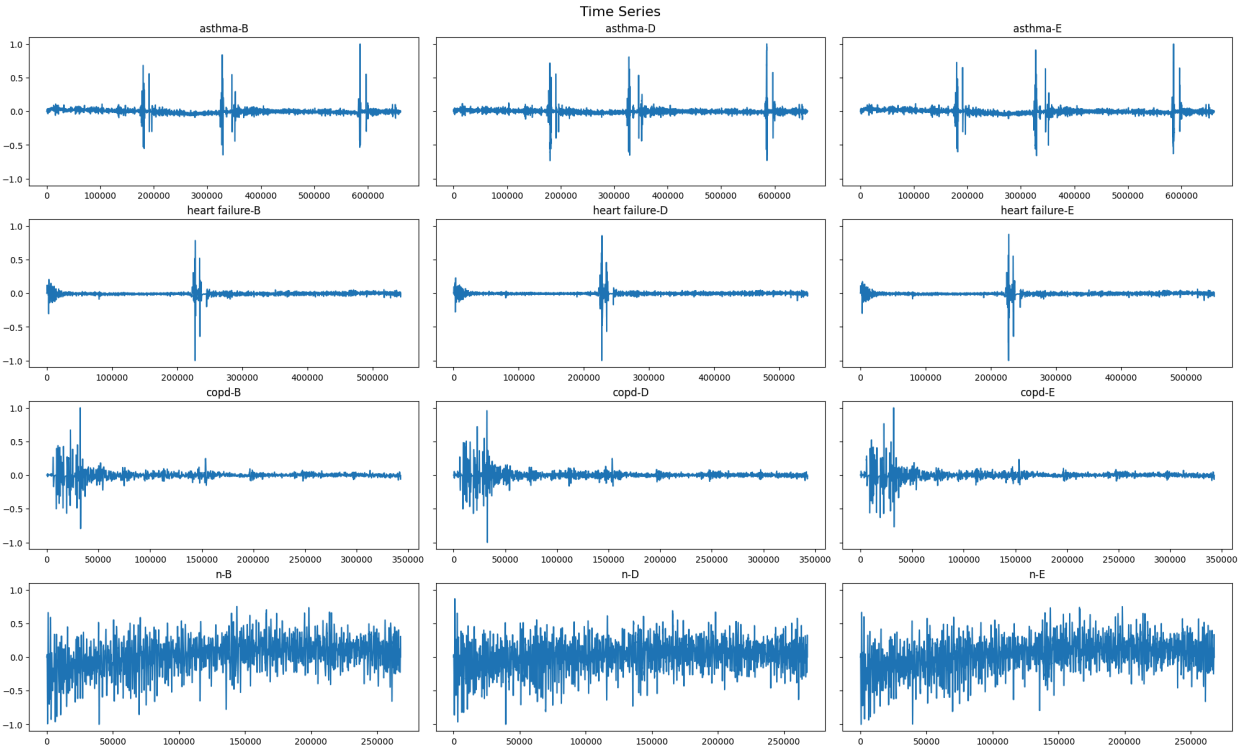


Fig. 6 Audio Waveform for Diseases and Filtering mode.

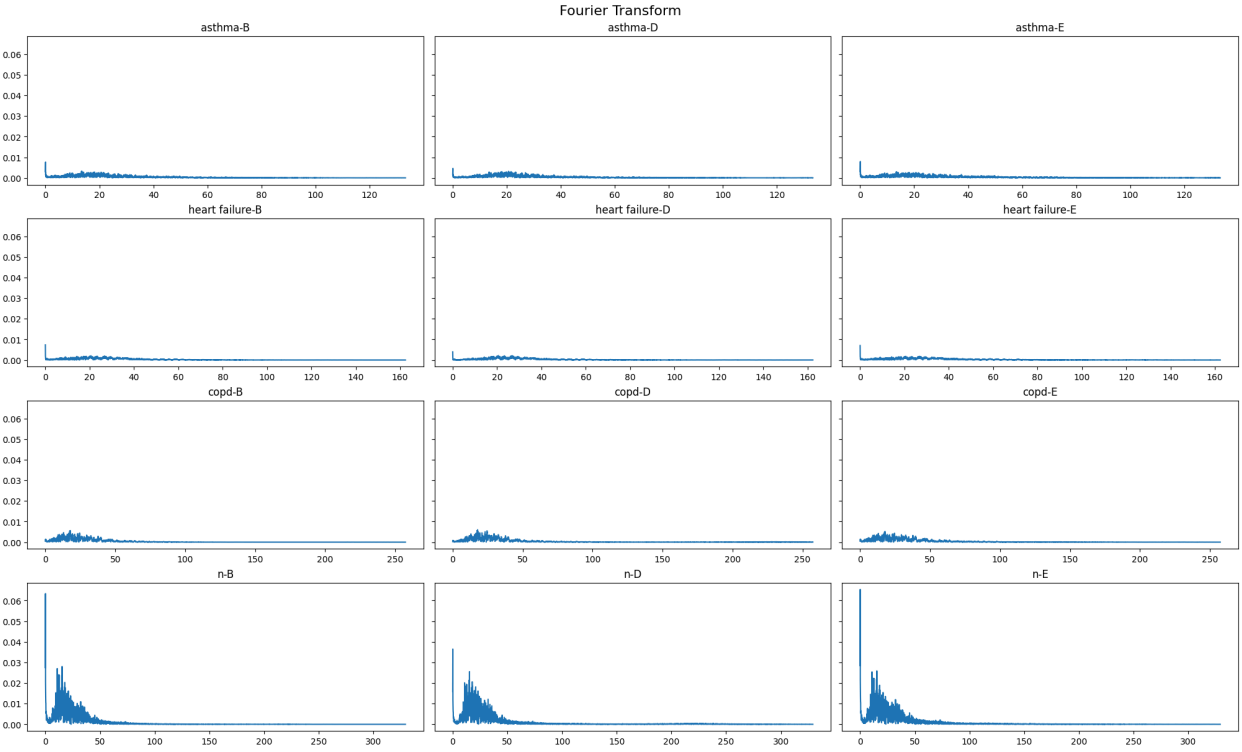


Fig. 7 FFT spectrum for Diseases and Filtering mode.

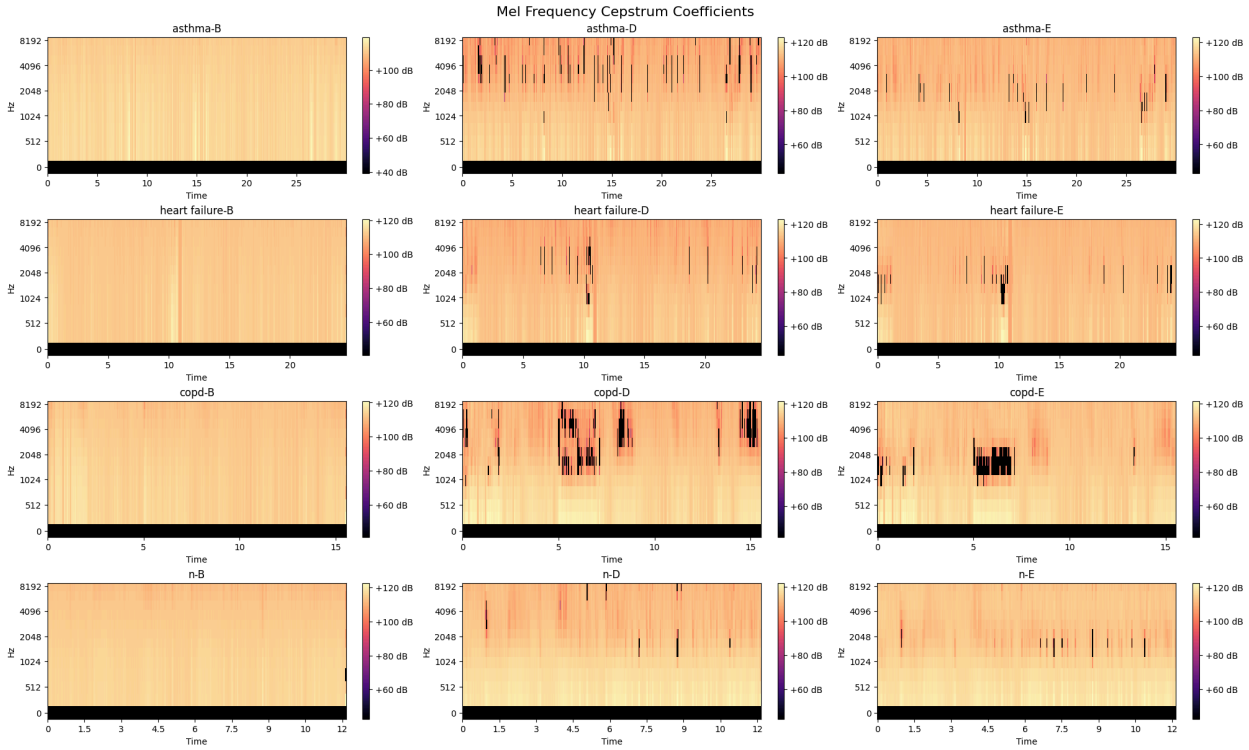


Fig. 8 MFCC-mean for Disease and Filtering mode.

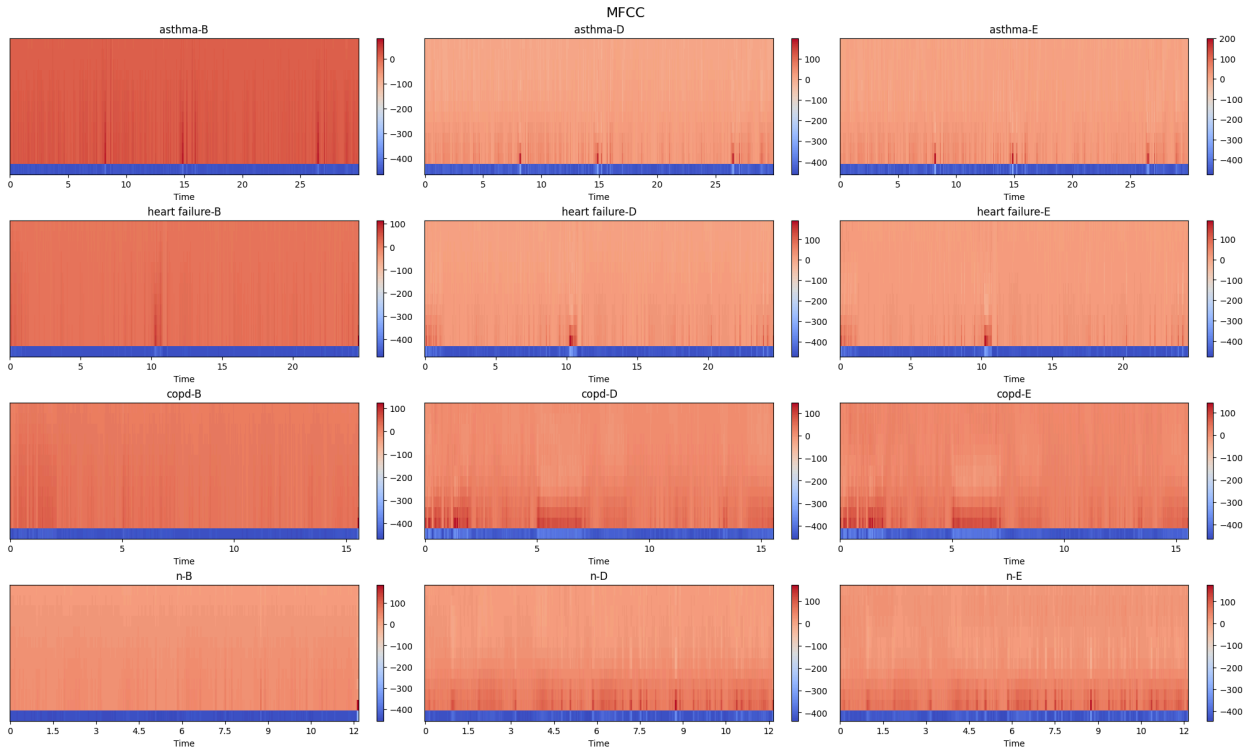


Fig. 9 MFCC-raw for Disease and Filtering mode.

Taking into account the above results, the proposed solution uses MFCCs features to classify the Disease using sound data for “asthma”, “copd”, “'heart failure” and “n” with just filtering modes D and E.

Now, in order to handle imbalance data, Data augmentation is performed using audio transformations like “add_noise”, “shift”, “pitch_shit” and “stretch”. Also D, and E filtering mode files are taken separately (taking them as if they were data from different patients), this duplicates the input sound data. The final distribution after this process is {'asthma': 128, 'copd': 126, 'heart failure': 144, 'n': 140}

Sound transformation results are presented in Fig. 10, and the final data distribution after data augmentations is shown in Fig.11.

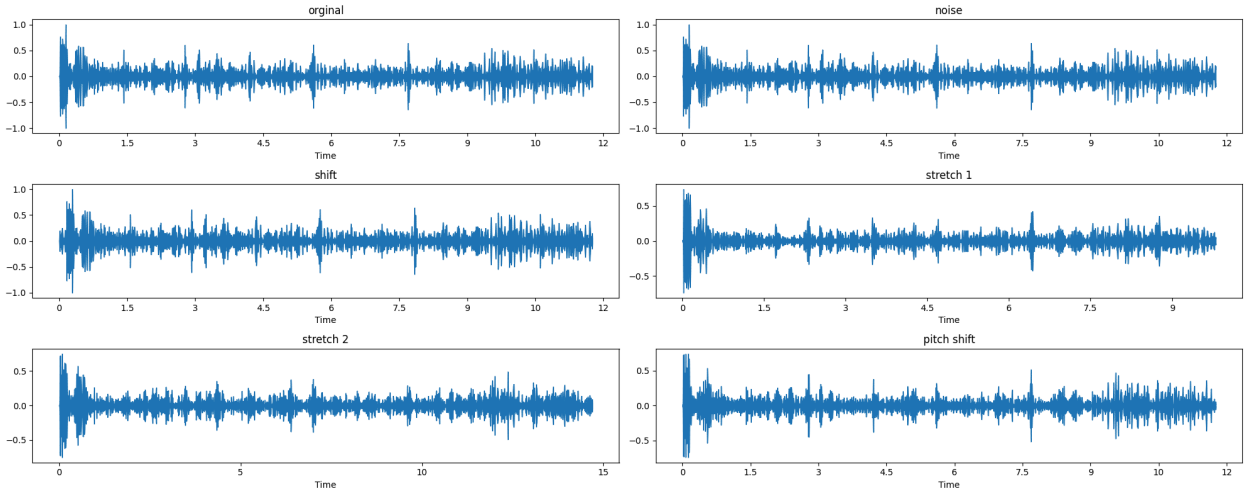


Fig. 10 Sound Transformations

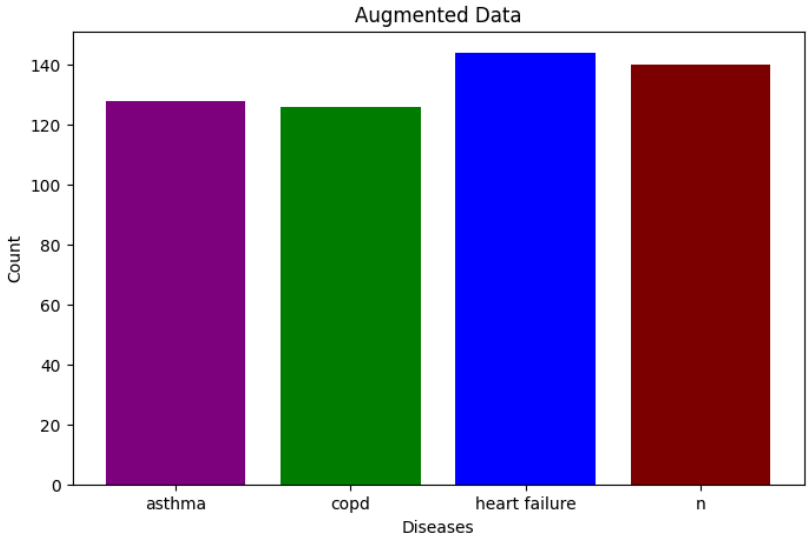


Fig. 11 Final Data Distribution after Data Augmentation.

Task 3 - Model

After testing several approaches using RNN [7], SVM [8] and some CNN [9]. The best result was reached using a CNN architecture from [10], which is originally used for speech emotion classification. In this case this architecture was modified including a couple of Convolution1D layers.

The first stage of model design involves feature preprocessing following the next steps:

- Generate 40 MFCCs features for each sound signal using librosa library. After this each one of the 538 samples in total will have a 2 dimensional MFCCs features with different shape in X axis.
- In order to unify the shape of features a zero padding technique is used. This process includes zero values to unify the shape of features [11][12].
- After the zero padding, the shape of features for each 538 sample is (1615, 40).
- Next, features are flattened in order to apply StandardScaler from scikit-learn library. This function standardizes features by removing the mean and scaling to unit variance [13].
- After, features are reshaped back to (538, 1615, 40)
- Data is splitted into train and test data sets with 70% for train and 30% for test.
- Finally, labels from train and test are one hot encoded

The CNN model is composed mainly of consecutive blocks of Convolution1D (size 128 and Relu activation) - Dropout(rate=0.4) - MaxPool1D(2), with a Flatten layer and Dense at the end. The CNN architecture is shown in Fig.12.

The compile configuration for training the model used a learning rate of 0.0005, batch size =16, epoch=80 and a RMSprop optimizer.

After the training process the model reached a loss of 0.0879 and an accuracy of 0.9815 over a prediction using the test dataset.

The results of the training process are shown in Fig. 13. The confusion matrix obtained comparing the test labels and predicted ones is shown in Fig.14.

Results showed excellent performance in signal classification into “asthma”, “copd”, “'heart failure” and “n” improving several results presented in literature.

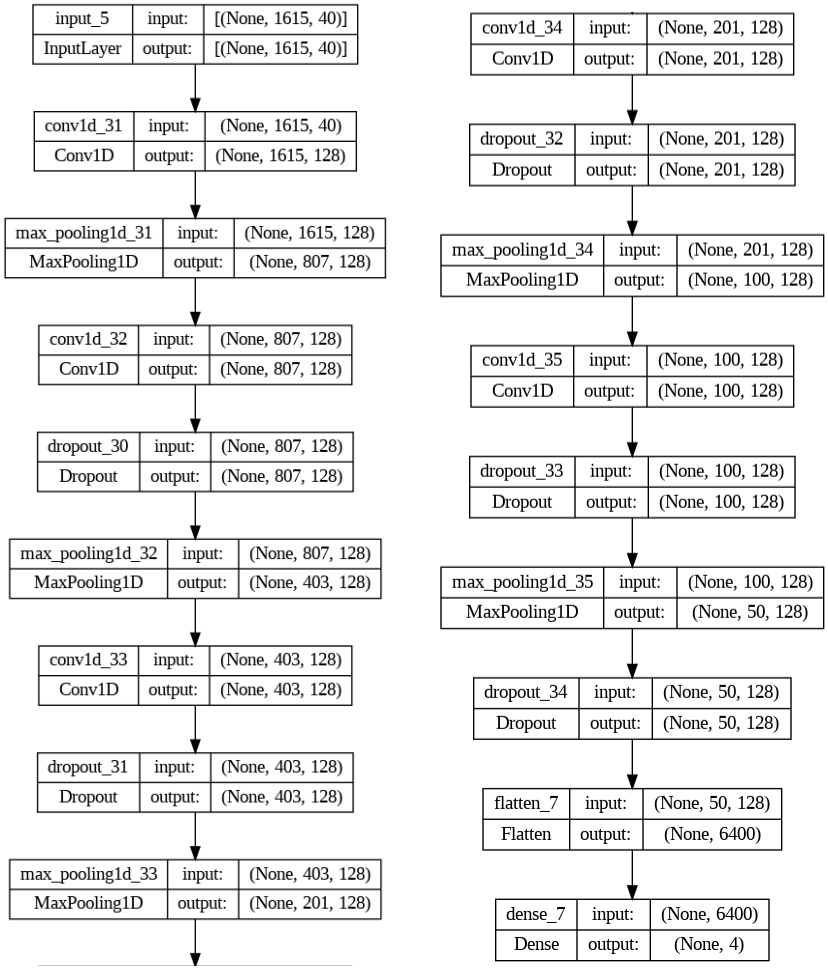


Fig. 12 CNN Model

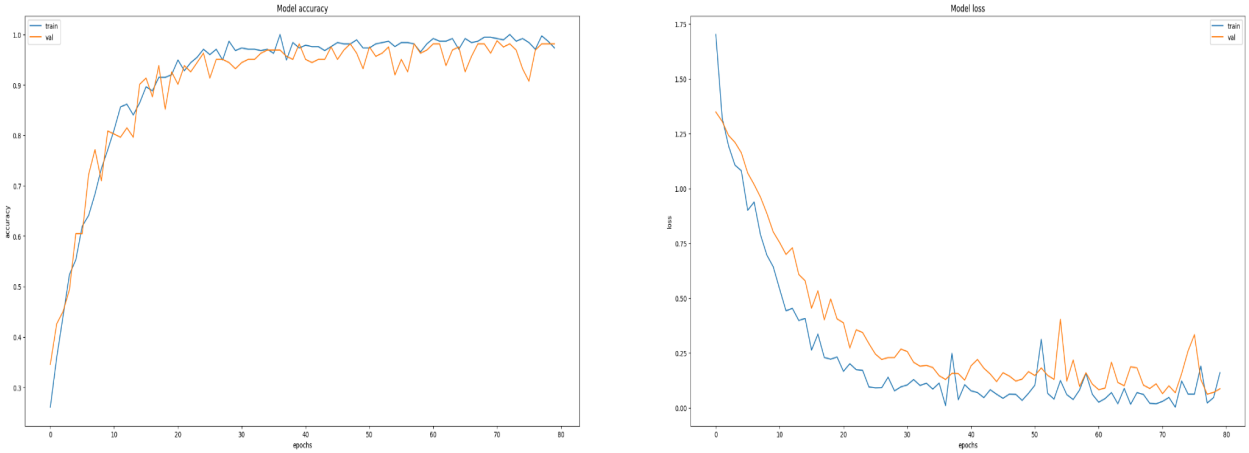
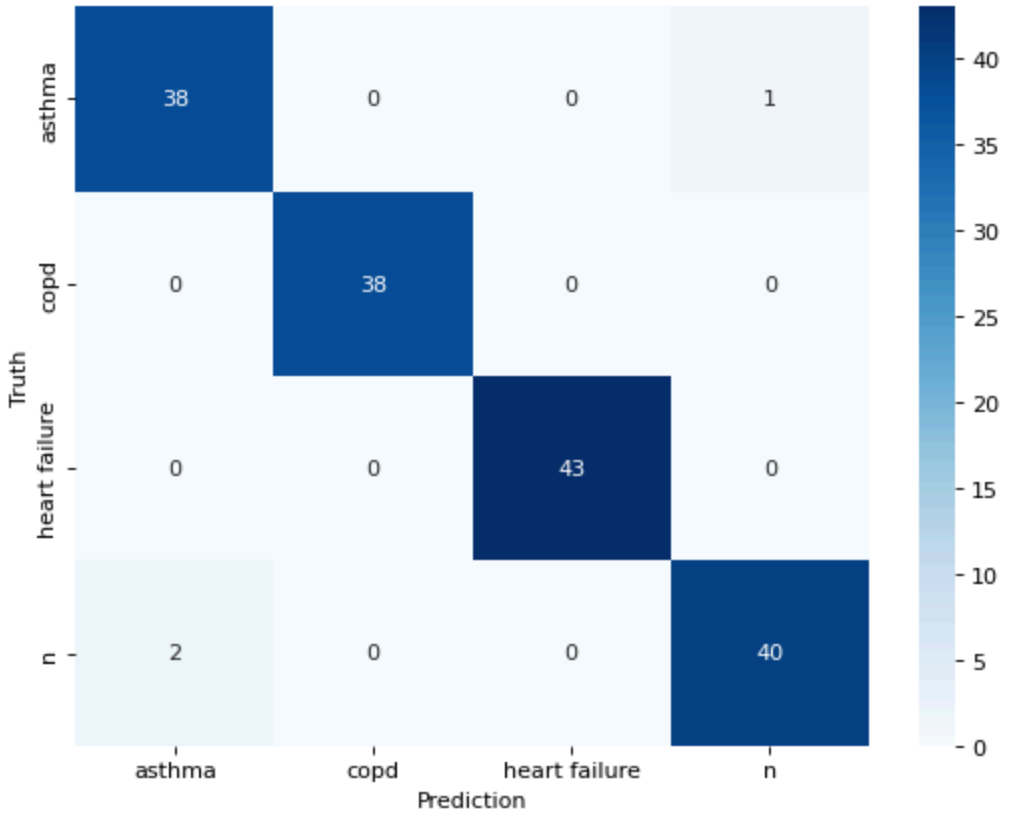


Fig.13 Accuracy and Loss results in training process



	precision	recall	f1-score	support
asthma	0.95	0.97	0.96	39
copd	1.00	1.00	1.00	38
heart failure	1.00	1.00	1.00	43
n	0.98	0.95	0.96	42
accuracy			0.98	162
macro avg	0.98	0.98	0.98	162
weighted avg	0.98	0.98	0.98	162

Fig.14 Confusion Matrix

Conclusions

- This document presents a machine learning model for detecting and classifying respiratory diseases using MFCCs features with a CNN architecture.
- Data augmentation techniques were used in order to balance the data between classes, in this case audio transformations like “add_noise”, “shift”, “pitch_shit” and “stretch” were used to create new signals.
- B filtering mode signals were excluded due to their information being really ambiguous between diseases.
- Results showed excellent performance in signal classification into “asthma”, “copd”, “'heart failure” and “n” with a 98% of accuracy which overpass several results presented in literature.

Future Work

- ➔ Increase the number of samples for each disease in order to have balanced data and improve final results
- ➔ Filter data using envelope filters.
- ➔ Test more audio features ike spectral entropy, Zero-crossing rate, Energy, Spectral-roll off, Spectral flux [14]
- ➔ Test more ML models and new CNN architectures [15]

References

[1] https://www.kaggle.com/discussions/general/74235
[2] https://www.freecodecamp.org/news/how-to-download-kaggle-dataset-to-google-colab/
[3] https://www.kaggle.com/code/ilyamich/mfcc-implementation-and-tutorial
[4] https://medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779
[5] https://librosa.org/doc/main/generated/librosa.feature.mfcc.html
[6] https://www.youtube.com/playlist?list=PLhA3b2k8R3t2Ng1WW_7MiXeh1pfQJQi_P
[7] https://www.kaggle.com/code/ytpsgamer/final-year-project
[8] https://github.com/Jason-Oleana/speech-emotion-classification/blob/main/RAVDESS-SVM-MFCC.ipynb
[9] https://github.com/fereshtehshah/Respiratory_Disorders/blob/master/cnn.py
[10] https://github.com/Jason-Oleana/speech-emotion-classification/blob/main/RAVDESS-CNN-MFCC.ipynb
[11] https://www.ni.com/docs/en-US/bundle/labwindows-cvi/page/advancedanalysisconcepts/lvac_zero_padding.html
[12] https://deeplizard.com/learn/video/qSTv_m-KFk0
[13] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[14] https://towardsdatascience.com/how-i-understood-what-features-to-consider-while-training-audio-files-eedfb6e9002b
[15] https://github.com/harmanpreet93/audio_classification/blob/master/audio_classification_conv1d.ipynb