2)

Machine learning can help us with diagnosis of human illnesses. Machine learning uses data and produces a program to perform a task. To do this first we should 1- Acquire data 2- Preprocess data 3- Develop predictive models 4- Integrate analytics with systems, we will explain these further. The project is about classifying heart sounds as normal or abnormal using machine learning.

Access and Explore Data

first section of the code is about downloading the training and validation data from physionet.org, then examining two samples where one is normal heart sound and the other is abnormal.
Next part is about loading the recordings into memory, then making a reference table from given "REFERENCE.csv" files to represent the true category of each sample, its crucial for "supervised" machine learning. The "1" label means normal and "-1" mean abnormal.

In this step we prepared the data to do the preprocessing next.

Preprocess Data

This part is about extracting features from the raw recordings. statistical and signal processing functions available in MATLAB are used to process and extract features from the raw heart sound signal. This section extracts 26 features for each recording, and splits each recording into windows consisting of 5 seconds of heart sound. Some of the features are: mean, median, standard deviation, …

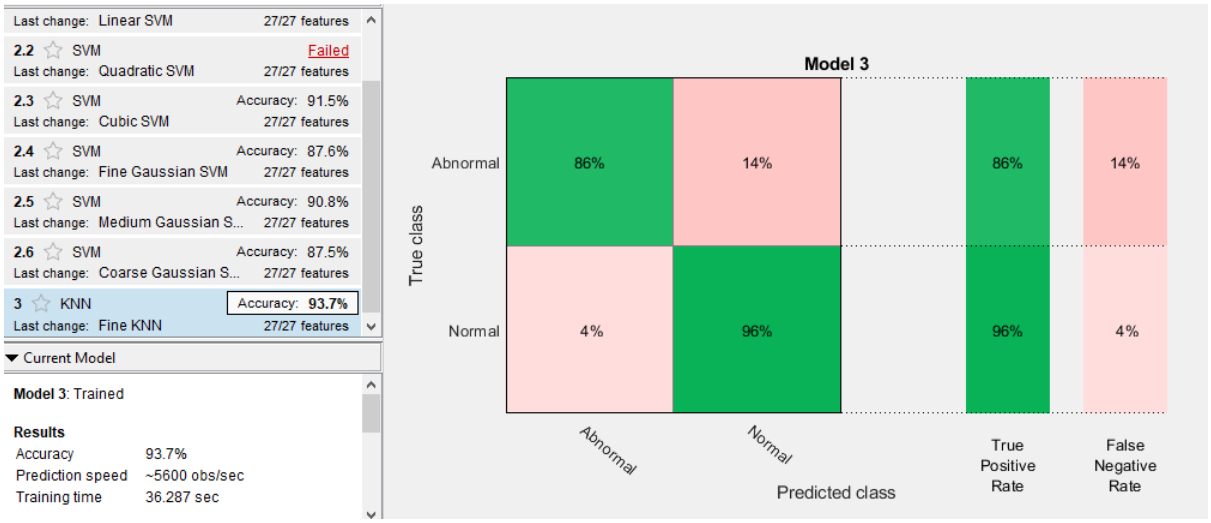
So features table is ready. Now, it's easy to develop predictive models.

Develop Predictive Models

Having an initial set of features, we can proceed to the next phase of the machine learning workflow, the training of various predictive models. Using the Classification Learner App to interactively train, compare and select classifiers.

After opening Classification Learner App, we should select features table and specify the predictors and the response. Then we can choose what percentage of the data be held for validation. Then train a bunch of different classifiers, including linear regression, SVM, and

decision trees. We can further note which ones deliver higher accuracy on the held out validation data, also explore the confusion matrices.



In above graph Confusion Matrix of "Fine KNN" model is shown, we can see that 14% of abnormal cases are classified as normal which is undesirable in medical practice. The error is because the number of normal class is significantly higher than abnormal class. To improve model performance, additional optimization steps are needed.

Method1: using a cost matrix that assigns higher misclassification cost to the 'Abnormal' class. At the same time, performing hyperparameter tuning by using Bayesian Optimization help us find optimal values for model parameters.

Above method consumes a lot of time and too large for deployment on a small embedded device.

Method2: preprocessing data by using NCA. Neighborhood Component Analysis (NCA) is an automated approach for selecting a small subset of features that carry information most relevant to the classification task while minimizing redundancy among selected features. Then we can train the model with these 15 features to reduce time and increase accuracy.

Integrate Analytics with Systems

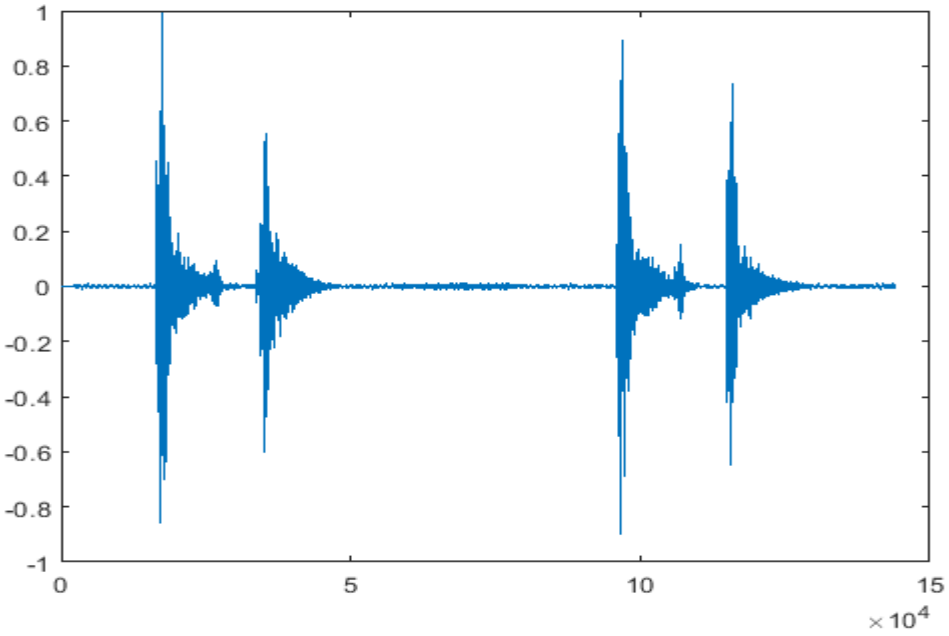
We can use coder of Matlab to generate C code so we can implement it on a device.

Challenges in Machine Learning: 1-data diversity 2-lack of domain tools 3-time consuming 4-platform diversity

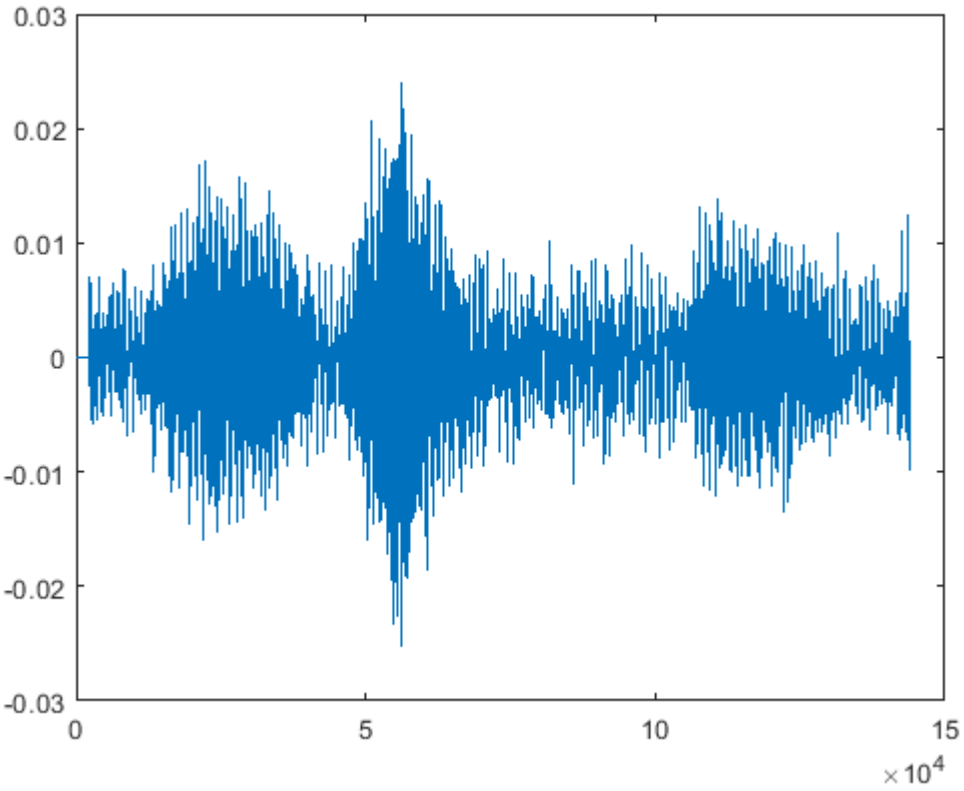
But Matlab has tools to deal with above challenges.

3)

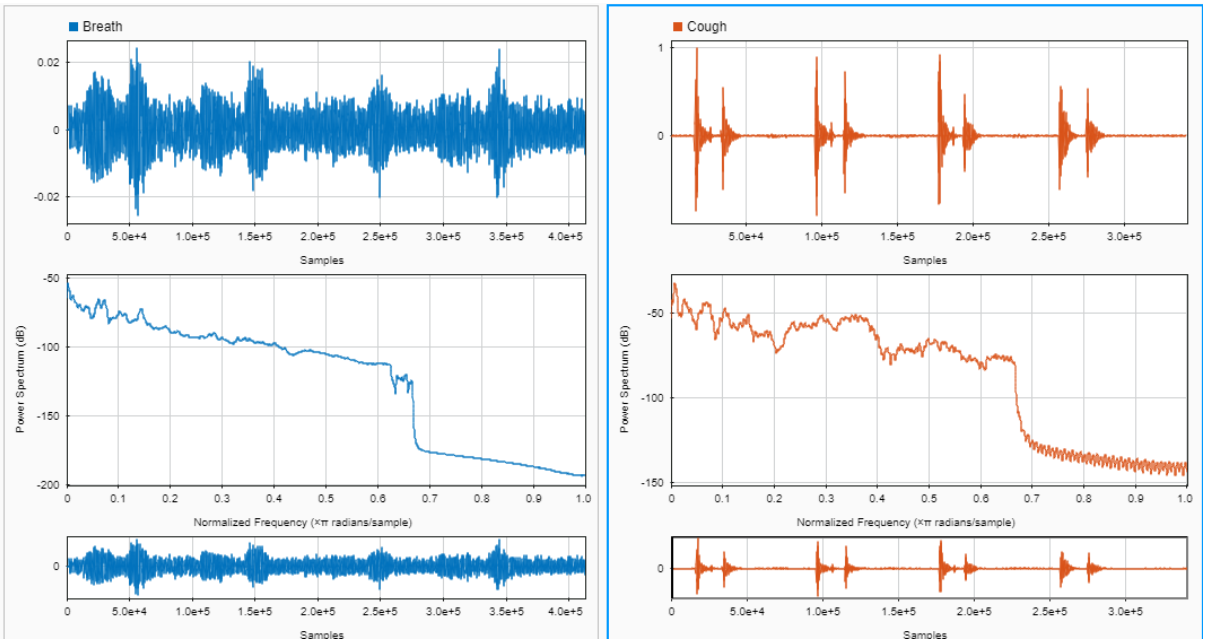
We recorded 26 voices of normal breathing and 26 voices of coughing, 5 of each are for validation and the rest is for training.



Coughing



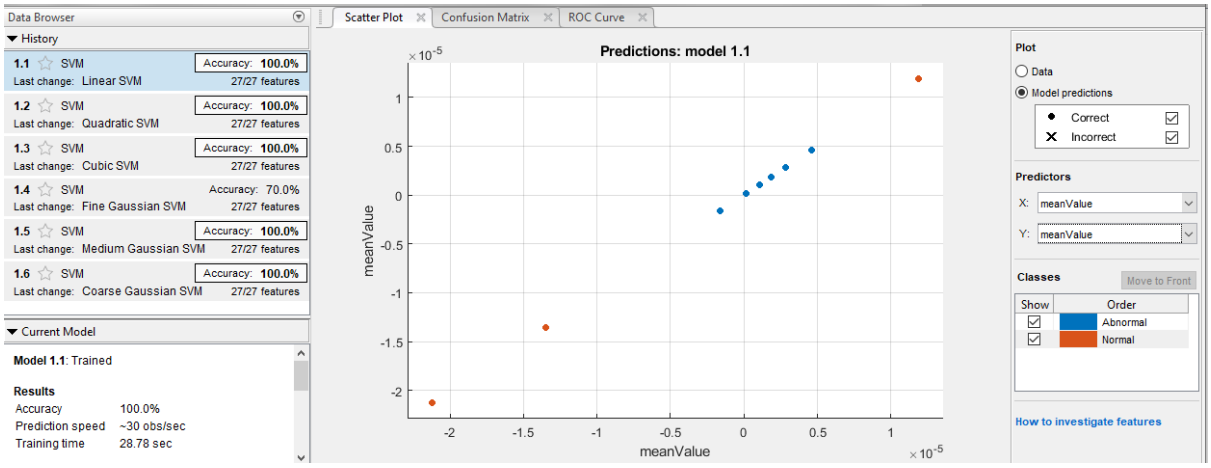
Breathing



feature_table

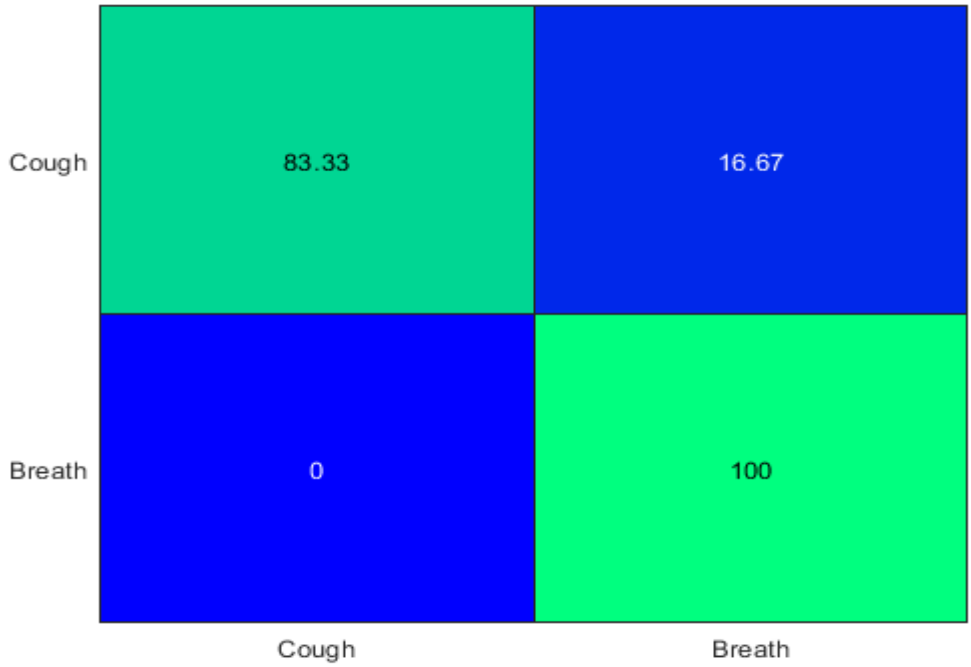
41x28 table

	1 meanValue	2 medianValue	3 standardDeviation	4 meanAbsoluteDeviation	5 quantile25	6 quantile75	7 signalIQR	sampl
1	4.6346e-06	-9.2514e-05	0.0037	0.0029	-0.0024	0.0023	0.0047	
2	-5.9451e-06	-8.4960e-05	0.0039	0.0029	-0.0023	0.0022	0.0045	
3	-5.6685e-07	-1.0596e-04	0.0039	0.0029	-0.0023	0.0022	0.0046	
4	3.1552e-06	-6.0093e-05	0.0039	0.0029	-0.0023	0.0023	0.0046	
5	6.1964e-06	-7.8153e-05	0.0037	0.0028	-0.0023	0.0022	0.0045	
6	-4.8799e-06	-8.4811e-05	0.0044	0.0032	-0.0024	0.0024	0.0048	
7	-1.3636e-06	-1.4446e-04	0.0038	0.0028	-0.0022	0.0022	0.0044	
8	2.3340e-06	-7.5461e-05	0.0046	0.0034	-0.0025	0.0025	0.0050	
9	-5.7927e-06	-8.7753e-05	0.0045	0.0032	-0.0025	0.0024	0.0048	
10	-2.1302e-06	-7.9896e-05	0.0044	0.0031	-0.0023	0.0023	0.0046	
11	1.8411e-06	-8.0255e-05	0.0057	0.0040	-0.0028	0.0028	0.0056	
12	-1.6073e-06	-7.1858e-05	0.0051	0.0036	-0.0026	0.0026	0.0052	
13	8.2111e-06	0	0.0053	0.0040	-0.0031	0.0031	0.0062	
14	1.5313e-07	-1.0334e-04	0.0050	0.0035	-0.0025	0.0025	0.0049	
15	-7.5900e-06	-1.1165e-04	0.0046	0.0033	-0.0025	0.0024	0.0048	



Normal means normal breathing and Abnormal means coughing.

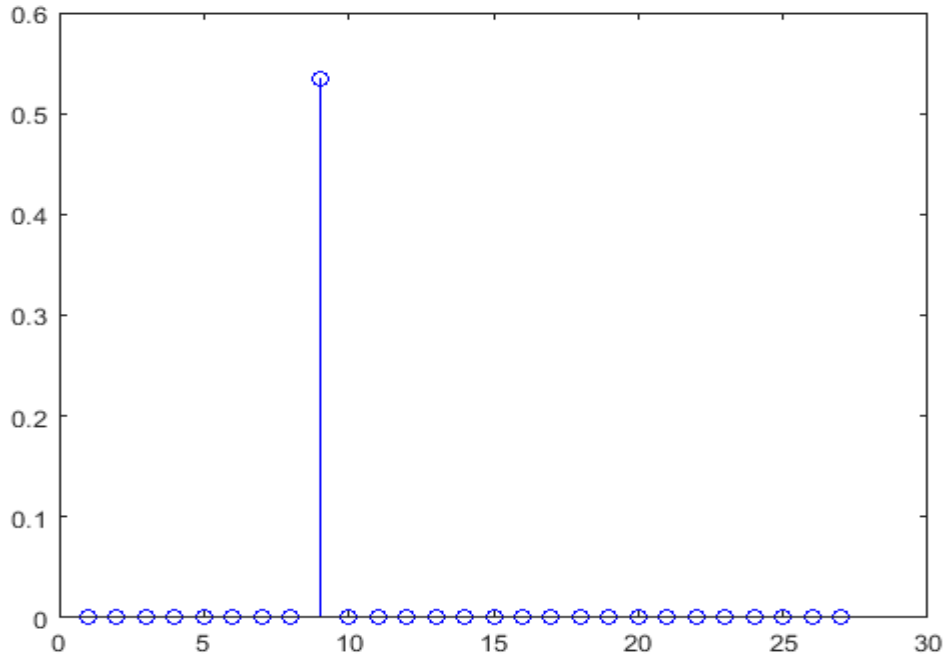
We used Classification Learner App and gave it training table, we selected 10 data out of 42 for testing and all predictions were successful. Next, we are going to train a model and test it on validation data set.



Now we gave the validation data (6 coughing and 4 breathing voices, in fact it is 5 of each but we labeled 1 of them wrong to test the model), to the model. Here is the result

TN	4
TP	5
FP	0
FN	1
Sensitivity	83.33%
Specificity	100%

In the next step we will try to optimize the predicting model by using NCA.



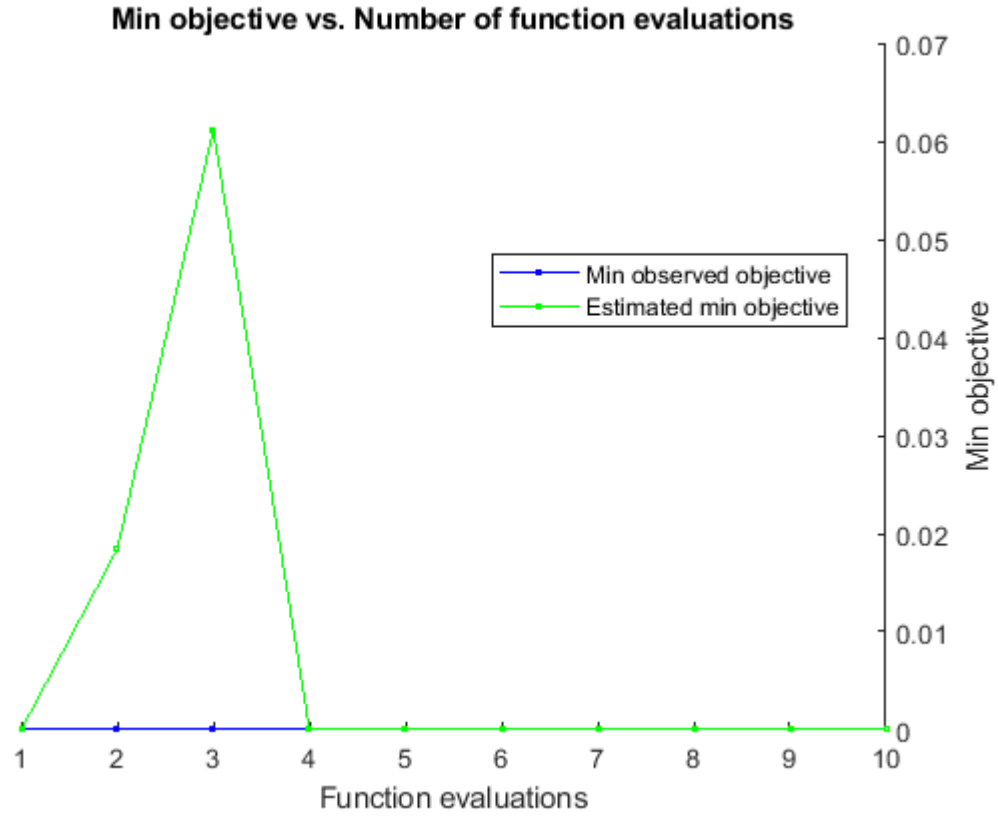
This graph tells us after applying NCA, only 1 feature is needed to train the model, others may be redundant or cause overfitting.

```
% Display list of selected features
disp(feature_table.Properties.VariableNames(selected_feature_indx))
```

```
    'sampleKurtosis'
```

The selected feature in this case is 'sampleKurtosis'

We also assigned higher cost for misclassification of coughing sounds.



We can see that after 3 iterations, model is ready.

Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	Method	NumLearnicycles
1	Best	0	1.2373	0	0	GentleBoost	
2	Accept	0.46341	0.18267	0	0.018426	RUSBoost	
3	Accept	0.46341	8.5202	0	0.06115	Bag	
4	Accept	0.46341	0.19461	0	3.4756e-05	RUSBoost	
5	Accept	0	1.0829	0	2.7794e-05	LogitBoost	
6	Accept	0.46341	0.18306	0	3.0895e-05	AdaBoostM1	
7	Accept	0	21.232	0	1.6551e-05	LogitBoost	
8	Accept	0	0.60228	0	1.655e-05	GentleBoost	
9	Accept	0	20.352	0	1.0727e-05	GentleBoost	
10	Accept	0	0.51163	0	1.0298e-05	LogitBoost	

Optimization completed.

Best observed feasible point:

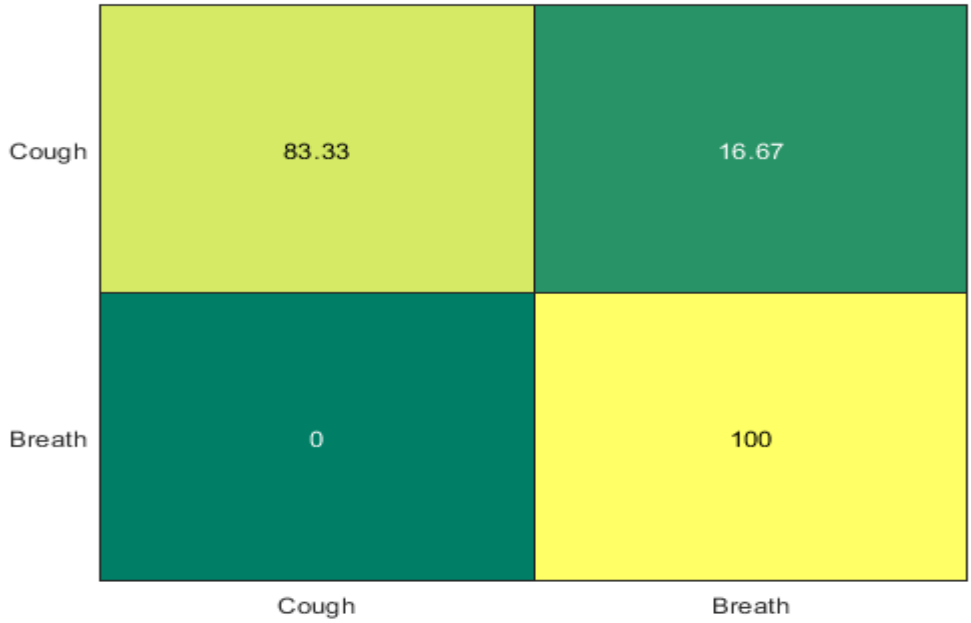
Method	NumLearningCycles	LearnRate
GentleBoost	25	0.0024335

Observed objective function value = 0
Estimated objective function value = 1.0298e-05
Function evaluation time = 1.2373

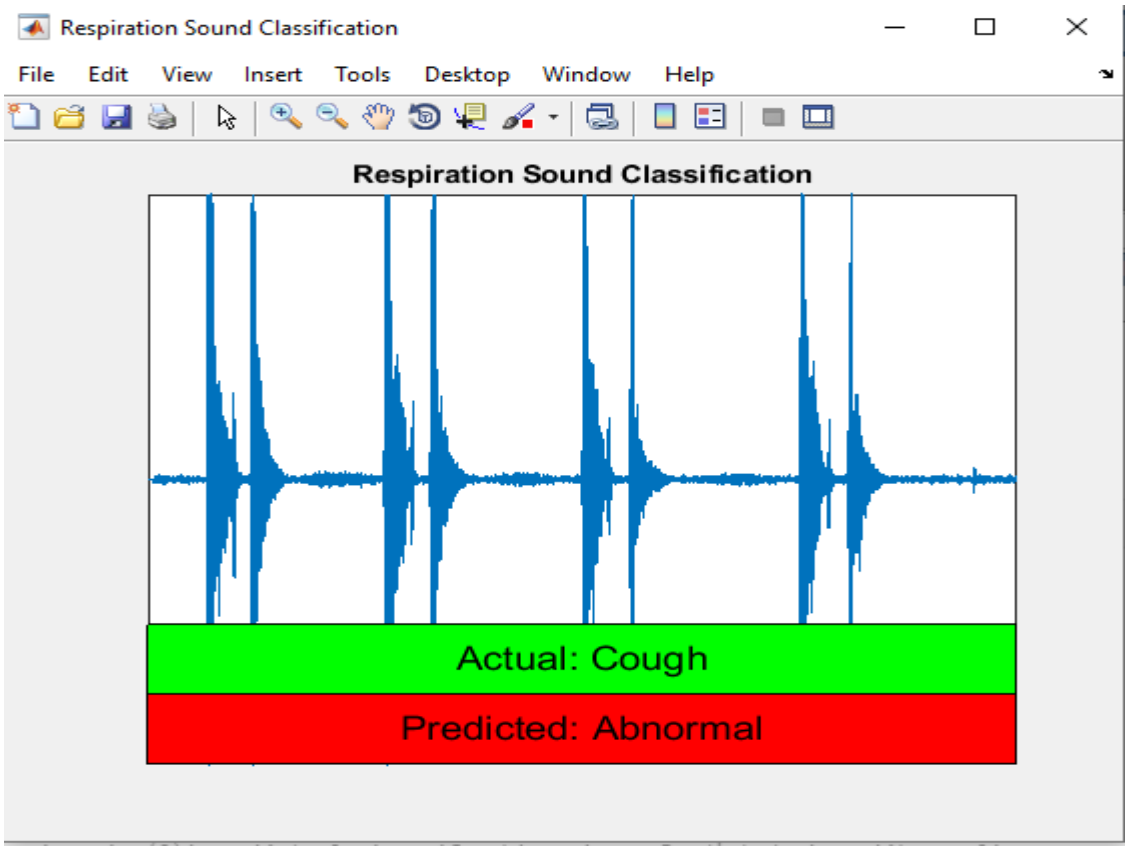
Best estimated feasible point (according to models):

Method	NumLearningCycles	LearnRate
LogitBoost	20	0.003726

Estimated objective function value = 1.0298e-05
Estimated function evaluation time = 1.0822



We can see that our model didn't improve in predicting validation data set, because one of entries in that data set is labeled wrong and improving the model doesn't help.



Abnormal means cough