University of Canterbury

Dysphagia detection with Machine learning and Esophageal Manometry

05 Feb 2020

Shankey Raheja
Student ID: 28153489

Albee Philip Neeramplackel
Student ID: 88572714

Table of Contents

Introduction:

Dysphagia is the medical term for swallowing related disorders. The study published by the American Speech-Language-Hearing association reported that, nearly 22% of adults and 30% of the elderly population in the United States experience some form of swallowing disorders in their lifetime [1]. One in 25 adults every year reports some kind of difficulty in swallowing food. Dysphagia or swallowing disorders in general, are usually triggered by other health conditions that patients might have experienced in the past, such as Stroke, head injury, acid reflux, dementia, or Parkinson's disease. The most common symptoms reported by patients with dysphagia include coughing or choking during swallowing, food coming back up through the nose, a feeling of food being stuck in the throat difficulty chewing food, or an unusual sound when eating or drinking. In some severe cases, dysphagia symptoms can develop into life-threatening conditions such as chest infections or pneumonia. Nearly one-third of the Patients with dysphagia, end up suffering from pneumonia, and close to 60,000 individuals die each year from such complications.

Dysphagia can be diagnosed by various methods. Some of the most commonly used methods are illustrated below:

X-Ray:
A patient is instructed to drink a barium solution that coats the esophagus and improves the visibility in the X-rays. An authorized medical practitioner can visually inspect the movement of the food through the esophagus while the patient swallows the food, using X-ray videos and detect any blockages.

A visual examination of your esophagus (endoscopy):
In this method, a thin instrument with a camera is inserted through the throat of the patient, so that medical expert can have a look at the esophagus directly and detect any possibility of inflammation or a tumor, which might have led to a swallowing disorder.

Esophageal muscle test (manometry):
In esophageal manometry, a small tube is inserted into the esophagus. This tube is fitted with three sensors connected to pressure recorders to record the changes in the pressure at three specific locations within the throat, while the patient swallows the food. The changes in the pressure levels at these three locations and the sequence of these changes are used to determine whether the patient is able to swallow the food, without any blockages.

Although all these techniques are highly effective in the detection of swallowing related disorders, they are extremely complex, time-consuming, expensive evaluations that require a lot of resources. It is not practical and cost-effective to perform these procedures, for each and every patient. Hence, in regular practice, clinicians use some initial screening to filter out the patients that might be at significantly high risk of these disorders such as patients with stroke or some neurological

disorders. These highly sophisticated and complex evaluations are performed only for those individuals. In the screening process, several patients with mild symptoms remain undetected until the problems become more severe and symptoms become explicitly evident.

In order to improve the efficacy in the diagnosis of swallowing disorders, by an amalgamation of existing methods, that is, "Esophageal Manometry" and the potential of machine learning, this study has been undertaken in conjunction with 'University of Canterbury Rose Centre for Stroke Recovery and Research'. Esophageal Manometry has been chosen for this project, as this technique totally relies on data of pressure recordings to identify the swallowing disorders, whereas other methods like X-ray and endoscopy rely heavily on visual examination by an authorized medical practitioner. UC rose center was established in 2014 by virtue of Mrs. Shirley Rose who had spent several years intending to the disability created by a stroke in her husband. Since 2014, the organization has achieved success in establishing a swallowing rehabilitation laboratory and is also planning to extend its research in building rehabilitation centers for various other diseases across the country. The organization provides rehabilitation facilities for stroke patients suffering from swallowing disorders, speech, and physical impairments. The research conducted by the organization reports that stroke is the second most common cause of death across the world and a common cause of disability in adults in developed countries. An article published by "The New Zealand Medical Journal" in 2018 reported that stroke incidents in New Zealand are relatively higher as compared to other high-income countries [9]. Stroke is also one of the major causes that lead to dysphagia. The effects of stroke can be for a long period of time and hence need to be observed regularly. Studies also suggest that more than 50% of stroke patients, eventually suffer from difficulties in swallowing food [10].

The main objective of this study is to improve the process of detecting swallowing disorders and make it more efficient and cost-effective, which will facilitate the detection in the early stages of the disease. This study is focused on using the potential of esophageal manometry and machine learning algorithms to train a model and automate the detection of Dysphagia, in patients.

Background:

In order to make any significant contribution to improving the swallowing disorders detection, it is imperative to have an understanding of the dynamics of the swallowing process and the series of events that happen around the esophagus, when a person swallows food or a liquid. The various phases of the swallowing process are illustrated as follows:

Oral preparatory phase:

The oral preparatory phase includes suckling, chewing, and masticating, that is, mixing of food with saliva. It then forms a bolus, that is, a ball-like substance of suitable size and consistency, and is pushed down through the throat with the help of the tongue.

Pharyngeal phase:

In the pharyngeal phase, the vocal closes to prevent food or liquid to enter the windpipe. The epiglottis moves to cover the airway and also the larynx pulls upwards along with it to provide more protection.

Esophageal phase:

In the esophageal phase, the bolus formed in the mouth is pushed downwards in a peristaltic movement, that is, symmetrical contraction and expansion movement. The lower muscle of the esophagus also called the sphincter relaxes at the start of the swallow and contracts only after the food substance enters the stomach.

Esophageal Manometry is a technique in which, pressure movements are recorded at 3 different locations, that is, Tongue base, Hypopharynx, and upper esophageal sphincter (UES), during the entire swallow duration (i.e., all 3 phases of swallow described above), which is usually 1.5 – 2 seconds. The recordings are usually taken over an interval of 1 millisecond or 4 milliseconds in some cases. The typical example of pressure recordings from a swallow sample is shown in Figure.01.

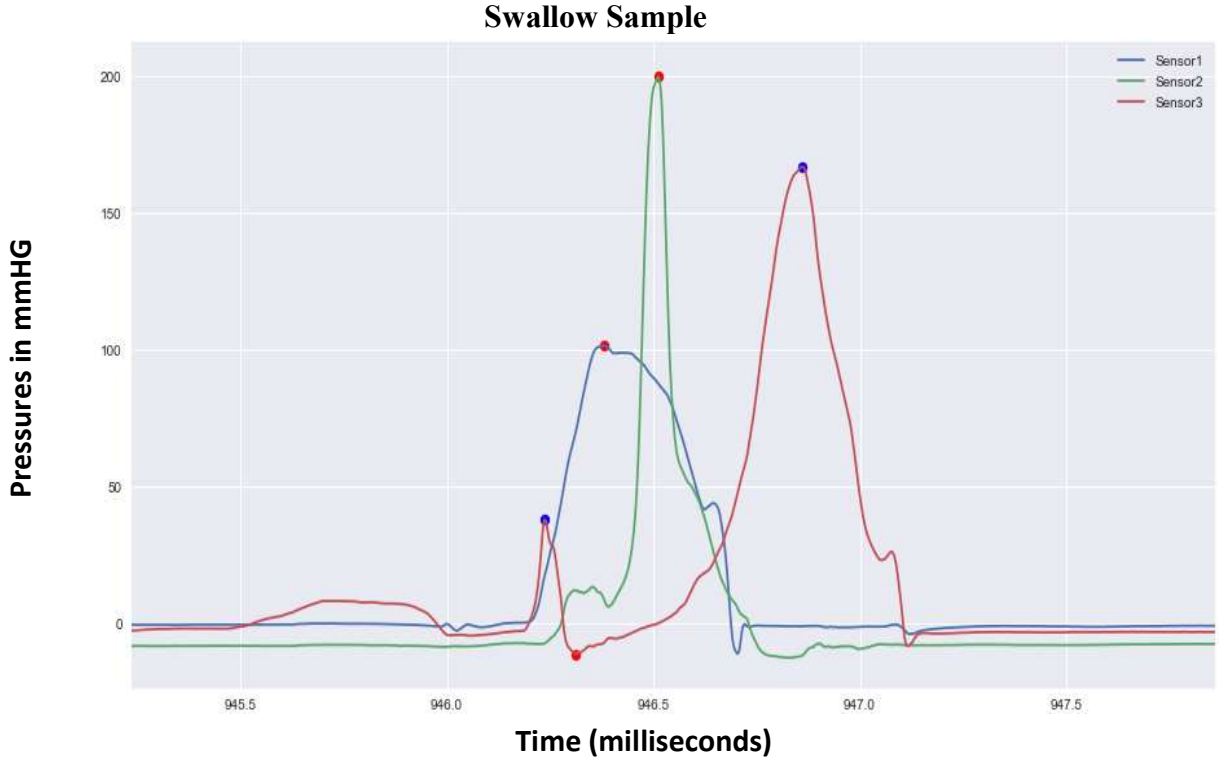

Figure.01 - Swallow sample

Sensors 1, 2 & 3 record the pressures at the tongue base, hypopharynx & UES respectively. As it is evident from the plot, that pressure goes to peak at the tongue base first and then hypopharynx. In the case of UES, pressure rise at the start and end of the swallow. The evaluation of the patient is conducted by using relevant parameters from the swallowing samples such as Pressure amplitude at the tongue base and hypopharynx, swallowing duration, and minimum pressure at UES.

In a study published in 2014, Kristin Lamvik, Phoebe Macrae, Sebastian Doeltgen, Amy Collings & Maggie Lee Huckabee conducted joint research on a group of healthy patients from different gender and age groups, with the objective to derive the 95% confidence interval for a range of parameters that play a significant role in the classification of the swallow as normal or abnormal [2]. In the current research, we have treated the results of this study as a baseline to choose meaningful swallow samples from raw data and to avoid samples with the presence of noise and outliers.

Methodology

Data Ethics:

Data ethics is an aspect that has been managed judiciously while handling the data that is used in the implementation of this project. The usage of data has been purely for research purposes. Any form of data that might not be useful in this project, has not been collected. In this research, we are using the patient's manometry test records which includes the pressures readings from the patient's throat, collected while the patient swallows a food. However, this data does not contain any form of personal information on Patients such as name, age, or gender. So, the primary step of data ethics by not displaying the patient's information is taken care of. By checking the data, it is not possible to know that the data belongs to which patient. It is data of several patients from experiments conducted on them, for manometry research. Therefore, the patient and associated organization has complete rights to this data and is unethical to share this data for any other purpose other than research. The next important step is to avoid any data leaking. This shows disrespect and mishandling of personal data which is unethical by all means. The data collected is utilized solely for knowledge, truth, and other predictive analysis depending on the field of research. Apart from the organization that owns this data, it requires consent for anyone else to use it and responsible authority has issued a consent only for this particular research. Any form of data collected during this research has been handled with complete integrity.

Constraints

In this section, the main objective is to highlight the constraints and the roadblocks that were encountered along the way and limited us to address the entire scope of a problem that we had undertaken in the initial phase of the project. The intent of this research originally was not only limited to the detection of unhealthy patients experiencing swallowing disorders but also to identify the underlying disease that led to disorders such as Parkinson's disease or Stroke. The research project was originally undertaken as a multiclass classification problem, however, due to

insufficient data available on different classes of the patients, the scope of the problem was reduced to the binary classification which would facilitate to detect of the swallowing disorder, without taking into account the underlying cause of disease.

In addition to the above limitations, there are some important features or variables that hold a reasonable level of significance in classifying patients as healthy or unhealthy with respect to swallowing, such as the patient's age, gender, height or weight. However, this data was not accessible for the patients due to limited availability as well as restriction to access any form of personal information on the patient, even if available.

Moreover, during the collection of the data, the storage device of high-resolution manometry equipment was collapsed and the delay was encountered in accessing the manometry record of patients, which further limited the magnitude of our analysis, in this research.

Data Collection:

The data used in this study is composed of 40 healthy and 40 Unhealthy patients. This data was retrieved from the directory of records maintained by Ph.D. scholars on various studies conducted previously, to investigate the relationship between swallowing disorders and underlying health issues such as Stroke, Parkinson's, or Huntington disease. The recordings were available in the form of text files. The basic structure of the data is shown in Table.01 (first 7 rows):

S. No	*Time Interval*	*Sensor1*	*Sensor2*	*Sensor3*
0	*0.000000*	*0.666056*	*2.428473*	*3.884184*
1	*0.004000*	*0.579080*	*2.506294*	*3.971160*
2	*0.008000*	*0.473793*	*2.570382*	*3.916228*
3	*0.012000*	*0.377661*	*2.616159*	*4.007782*
4	*0.016000*	*0.341039*	*2.639048*	*3.929961*
5	*0.020000*	*0.331884*	*2.698558*	*3.934539*
6	*0.024000*	*0.304417*	*2.808423*	*3.929961*
7	*0.028000*	*0.281529*	*2.863355*	*3.962005*

Table.01 Structure of Raw Data

There are 5 columns in the data which are "Serial No", "Time Interval", "Sensor1 recording", "Sensor2 recording" and "Sensor3 recording". The number of recordings/rows per patient ranged from 115000 to 786000, for healthy patients and 600,000 to 1,800,000 for unhealthy patients, depending on the length of the manometry session.

Data Quality:

The recordings in the data contain a lot of noise which usually arises during the manometry session because of the following reasons. Pressures spikes of changes that get recorded during sensors calibration is usually noise, which has no meaningful information. Patients can also experience a cough or sneeze during manometry sessions and pressures spikes recorded during this will usually appear in the data as noise. In some cases, patients fail to comply with the guidelines of the medical

practitioners and this can result in imperfect recordings which may appear as noise in the data, in the form of unusual spikes or outliers. The typical examples of noisy data are shown in Figure.02 and Figure.03.

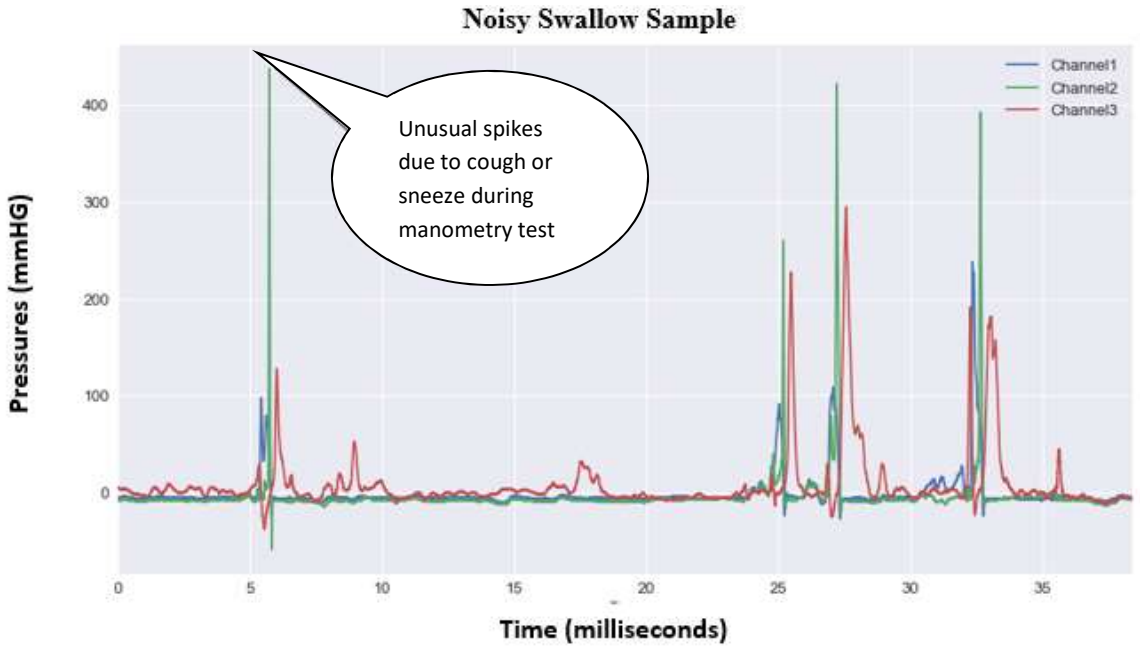

Figure.02 – Example of Noise in Swallow sample

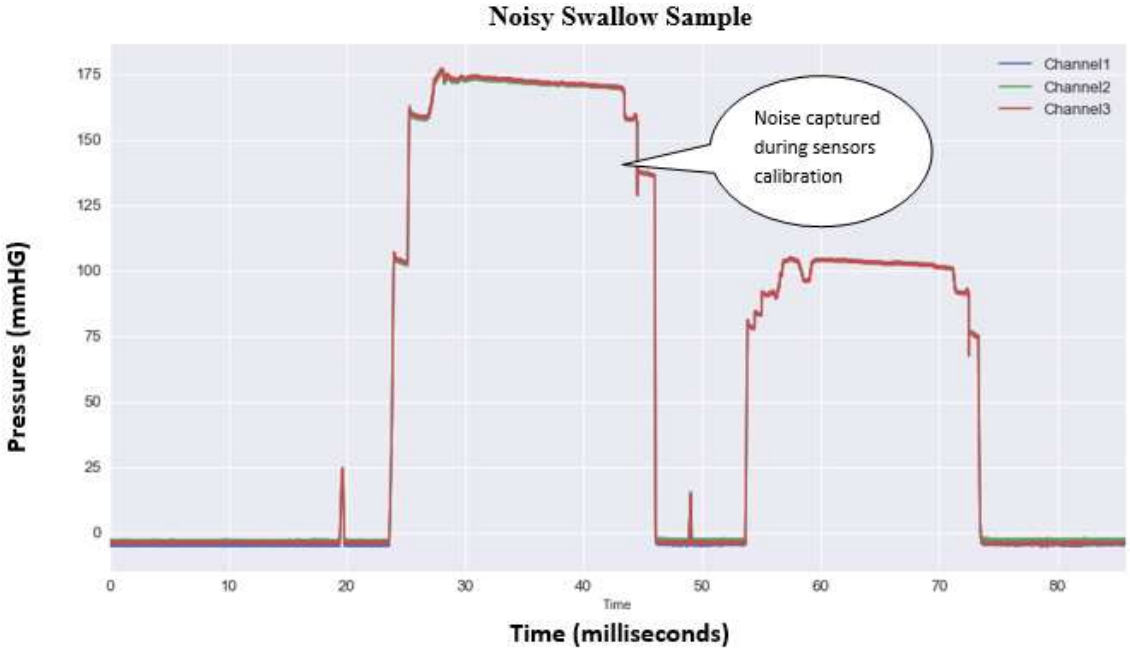

Figure .03 – Example of Noise in Swallow sample

Missing values appear in the data at few places. However, during the sampling process, it has been ensured that decent swallow samples with no missing values and no underlying noise are collected. The process of sampling is described in the section below.

Data Preprocessing & Sampling:

The steps involved in the procedure of data preprocessing and sampling are elaborated as below. The data cleaning & processing was mainly completed using "Pandas" & "matplotlib" libraries in python. The Python script used in the process has been included in the Appendix section:

Step.01 Reading the Raw data and Transforming into suitable format for plotting & Visualization

In this step, the raw data which is in the form of text files is loaded into python, using the Pandas library. The readings per line, in the text file, were in the form of tab-separated string values which were split and saved in separate columns, that is, "Time", "Sensor1", "Sensor2", "Sensor3". The datatype of readings stored in each column was converted from string to floating numbers. The "Time" Column was set as an index column in the data frame, to facilitate the plotting and visualization with "Time" on the x-axis.

Step.02 Extracting swallow samples from the raw data and exporting in the form of excel files

After the data preprocessing & visualization in the form of Time series plots, it was observed that data for every patient contained recordings of hundreds of swallow samples. As a next step in the process, the challenge encountered was to extract the relevant and meaningful swallow samples by filtering out all the noise, outliers, and samples with missing values, for each and every patient. Since the patterns in the data are quite inconsistent, it's is challenging to put together a logic that can automatically detect the start and end of meaningful swallow samples and extract those specific rows from the data. However, an attempt was made to automate this process, in which 200 Healthy and 200 Unhealthy random samples were collected using a python script.

Ideally, all the swallow samples belonging to a particular patient should be identical and have same features. In such case, one sample per patient would have been sufficient for modelling process. However, due to some limitations of manometry such as slight displacement of sensors from required position in the esophagus which can happen during swallowing, variation usually appears in the pressure recordings of different swallow samples from the same patient. In order to account for these slight variations, 5 samples per patient were collected for modeling process.

After plotting & analyzing the samples, it was noticed that there were instances where the script captured noisy samples and in some instances, the script was unable to detect the start time and the end time for a swallow perfectly due to inconsistent patterns in the data. In such cases, it would have been difficult to evaluate the independent variables or parameters from a swallow sample. Hence, the time series plots of these 400 samples were scanned to discard the noisy samples or samples where the start and end of the swallowing process were not captured perfectly. After this

filtration process, we were able to have 124 Healthy and 120 Unhealthy swallow samples perfect for modeling, in the form of excel files. The average swallow duration for a person is usually 2 seconds and if recordings are at an interval of 1 millisecond, one swallow sample usually consists of 2000 rows of data.

Step.03 Evaluating features of a Sample:

The features were evaluated for 124 Healthy and 120 Unhealthy Swallow samples are shown in Table.02.

S. No	*Features*	*Unit of Measurement*
1	*Sensor 1 Peak value*	*mmHG*
2	*Sensor 1 Peak value*	*mmHG*
3	*Sensor 3 minimum value*	*mmHG*
4	*Latency (Time distance between sensor 1 peak and sensor 2 peak)*	*milliseconds*
5	*Swallow start time*	*milliseconds*
6	*Swallow end time*	*milliseconds*
7	*Swallow duration*	*milliseconds*

Table.02. Independent Variables

These parameters play a crucial role in identifying the swallow as healthy or unhealthy and have been used as independent variables in the data modeling process. The findings of exploratory data analysis on these parameters are recorded in the results section.

Data Modelling:

In the previous section, we illustrated the process of data preparation, sampling, and evaluation of required parameters from each and every swallow sample such as Sensor1 Amplitude, Sensor2 Amplitude, Sensor3 minimum value, Latency & UES/Swallow duration. In this section, we will train classification algorithms by using these parameters from 124 healthy and 120 unhealthy swallow samples. Sensor1_max, Sensor2_max, Sensor3_min, Latency, and UES_Duration are independent variables that have been used to train a model that can classify the swallowing type as “Healthy” or “Unhealthy”. The dataset composed of 124 Healthy and 120 Unhealthy samples was split into training and test dataset in the ratio of 70:30. The class balance for both the classes was maintained during the splitting process, in both the training and test dataset. The training dataset is composed of 70% of Healthy and 70% of Unhealthy samples. Similarly, the test dataset is composed of 30% healthy and 30% unhealthy samples. The classification algorithms that were implemented to model the data are described below:

Logistic Regression:

Logistic regression is an effective machine learning algorithm for classification problems. Logistic regression provides a probability distribution of a dependent variable over the given classes. It can be used for both binary and Multiclass classification problems. However, in this project, we are dealing with Binary classification. The implementation and interpretation of this algorithm are

very straight forward. The decision of a dependent variable belonging to any particular class is based on probability. We can use the probability threshold for hyper parameter tuning, which can help to improve the accuracy of the model in case of class imbalance issues. It is efficient to train and provides good accuracy. However, it does not support collinearity, which means that all the independent variables used to train the model must not be significantly correlated.

Support Vector classifier:

The support vector classifier is another algorithm that is widely used for classification problems. In the Support vector classifier, the main idea is to find a hyperplane in N-dimensional space that can act as a decision boundary for various classes. The idea is to choose a hyperplane that has the maximum distance between data points of different classes. By maximizing this distance, the data points can be classified with more accuracy. In the case of binary classification problems, Hyperplane is just a line. This algorithm is easy to interpret up to 2 or 3 classes, however, when there are more than 3 classes, it becomes less intuitive. It is suitable for large datasets and provides good accuracy. However, when classes have a lot of overlap, this algorithm may not provide accurate results. In this algorithm, we use the C value for hyperparameter tuning which basically tries to optimize the maximum distance from the classes while choosing the hyperplane as well as the number of data points that the model classify correctly for training data. This is done to avoid the problem of overfitting.

Random Forest:

Random forest is another machine learning algorithm used for classification problems. Random forest is basically a combination of decision trees. Each tree is trained with different samples and features from the dataset and outcomes from the several random trees are obtained using the test data. The final prediction is evaluated by averaging the outputs from several decision tress. The model can handle a lot of variables and can work well with datasets with missing values. It provides good accuracy. The main hyperparameters that are used for tuning are a number of decision trees and the number of features that are used by each decision tree to make a classification which is basically a depth of a tree. It is easy to interpret and can be implemented effectively. However, when the number of trees is larger, it can become slow and inefficient.

XGBoost:

The XG boost algorithm is a power machine learning algorithm that is implemented using gradient boosted decision trees and is extremely popular for its speed and performance. Gradient boosting is a technique that contains several decision trees and is used for regression and classification problems. The normal gradient boosting uses the loss function of the base model, that is, a decision tree to minimize the overall error in the model, however, the XGBoost algorithm uses the second derivative as a proxy to minimize the overall error in the model. The procedure and principles of both gradient boosting and XG boosting are similar in terms of the modeling process. However, the XGBoost algorithm produces a more regularised model that handles over-fitting better than the gradient boost algorithm. The XGBoost algorithm is highly effective because it produces decision trees iteratively and tries to minimize the loss in every next iteration. One tree predicts the target

output, the second tree predicts any residuals of the first tree. These residuals are formed as a result of the loss function used in the model. This procedure is continued as the next tree predicts the residuals of the second tree and so on. The XG boosting has proved to be better than gradient boosting and random forest as it leverages pattern in residuals and iteratively improves the model. The XGBoost algorithm is a black box algorithm along with other tree ensemble models such as random forest and LightGBM. XGBoost performs really well, with the data that has more records compared to the number of features or independent variables. This algorithm can be used when the data contains both categorical and numerical features, and sometimes just the numerical features. It also facilitates data cleansing as it can handle missing values without any imputation pre-processing.

LightGBM:

The light gradient boosting machine is another powerful classification algorithm that follows a gradient boosting framework that works on decision trees to improve the efficiency of the model. This algorithm is fast, consumes less memory, distributed, and has a high performance based on decision trees which also allows ranking and classification in machine learning. It is capable of handling large scale data and also supports parallel and graphics processing unit (GPU) learning. The Light GBM contains histogram-based splitting, gradient-based one-side sampling (GOSS), and an exclusive feature bundling (EFB) which makes it relatively faster. One of the major differences between LightGBM and XGBoost is that LightGBM uses the gradient-based one-side sampling technique to filter the observations or samples whereas the XGBoost algorithm uses a pre-sorted algorithm and histogram-based algorithm to filter the samples. The other difference between LightGBM and other algorithms is that LightGBM grows vertically, that is, it grows leaf-wise whereas other decision tree-based algorithms grow horizontally, that is, level-wise. The leaf with the max delta loss will allow the decision tree to grow further. Overall, LightGBM is popular for its high efficiency and faster training speed.

Modelling Process:

Step.01:

All the mentioned classification algorithms were trained using training data and default settings for hyperparameters. Evaluation metrics of these models are estimated on the testing data. The precision, recall, and accuracy values were recorded for each of these models.

Step.02:

As a next step in the process, hyper-parameter tuning was performed for all the classification models to further improve the performance of these models, and metrics were re-evaluated to record any improvements. The objective of this project is to detect unhealthy patients in such a way that no Unhealthy patient is left undetected. Hence, hyperparameter tuning was solely focused on improving the recall value of the models for the “Unhealthy” class, without compromising much on overall accuracy as well as precision to recall ratio.

Step.03:

In step.03, the process of hyperparameter tuning was repeated to achieve the recall value of 100% for the "Unhealthy" class and precision value of 100% for the "Healthy" class. The objective is to train a model that can classify 100% of "Unhealthy" patients correctly and at the same time, identify some percentage of patients that can be classified as healthy with high confidence. Such a model can be used in the initial screening of patients and help to save the cost and resources for achieving the goal that no unhealthy patient is left undetected.

Results

In the previous sections, the process of Data Cleaning, sampling, pre-prepossessing, and modeling has been outlined in detail. In this section, samples of results obtained from every step of the data cleaning process have been recorded. In addition to this, we have recorded the performance of all the Classification models and their evaluation metrics.

Data Preprocessing & Sampling:

Step.01 Reading the Raw data and Transforming into suitable format for plotting & Visualization

The sample format of the data, after cleaning & pre-processing is shown in Table.03. The clean data contains 4 columns i.e. Time in milliseconds and pressure recordings from Sensor1, 2 & 3 in mmHg.

Time	S.No	Sensor1	Sensor2	Sensor3
0.000	0	-0.288396	-0.178531	0.297551
0.004	1	-0.288396	-0.123598	0.059510
0.008	2	-0.178531	-0.178531	0.169375
0.012	3	-0.251774	-0.233463	0.325017
0.016	4	-0.233463	-0.361639	0.160220
...	...	...	...	...
1019.652	254812	-3.099107	-1.121538	131.137560
1019.656	254813	-3.117418	-1.277180	131.613642
1019.660	254814	-3.089952	-1.396201	132.272831
1019.664	254815	-3.126574	-1.542687	132.977798
1019.668	254816	-3.126574	-1.670863	133.563745

Table.03 Format of Pre-processed Data

The time series plot for this sample is shown in Figure.04

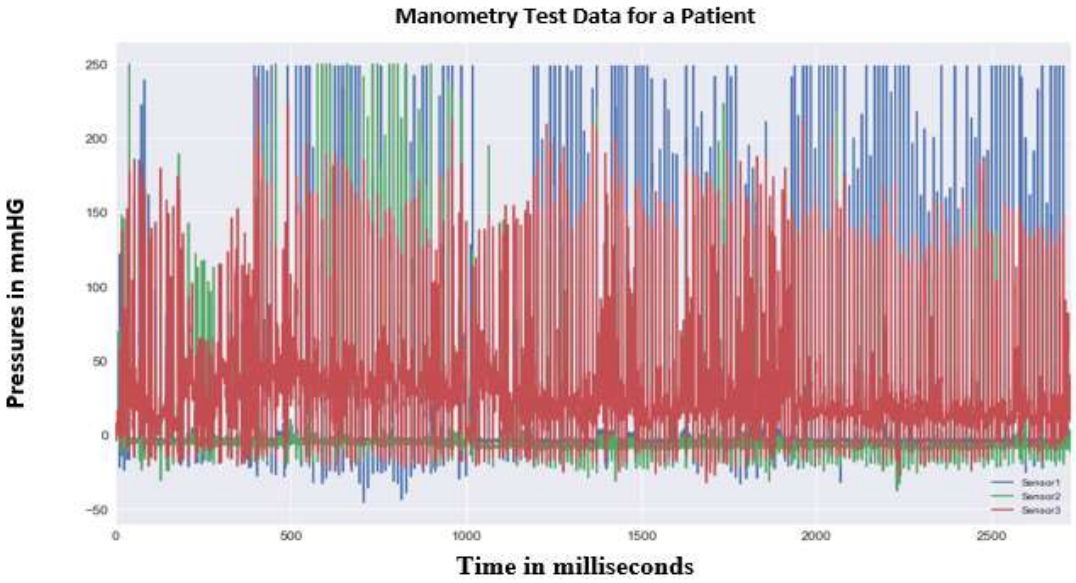

Figure.04 – Manometry Data for a Patient

Step.02 Extracting swallow samples from the raw data and exporting in the form of excel files

The time series plot of manometry data for one patient shows hundreds of pressure spikes and each spike indicates that a person has swallowed food or a liquid. However, in the data sampling process, only 5 samples per patient were extracted. The Time series plot of one swallow sample extracted from the clean data in Figure.05.

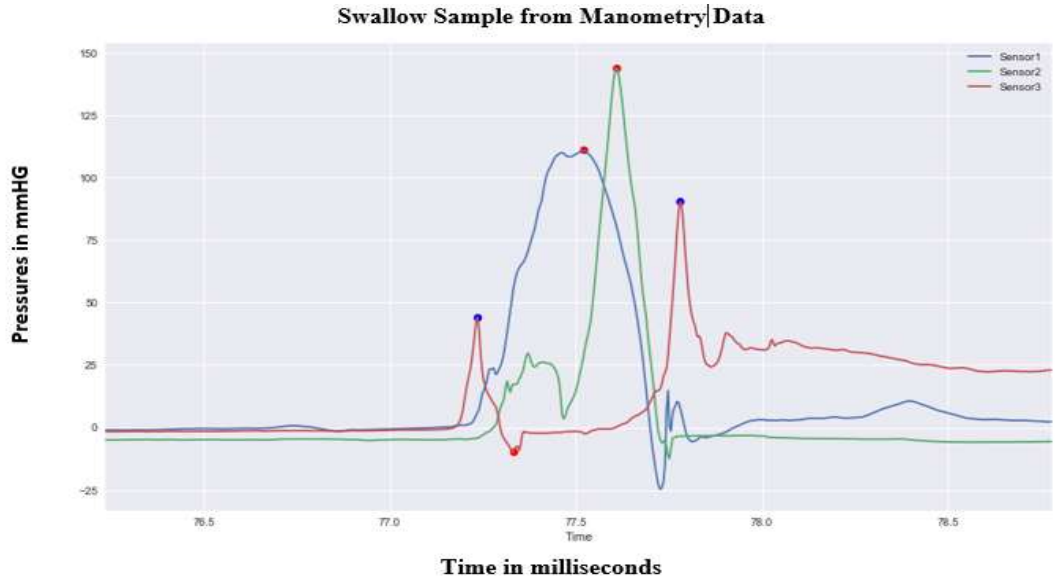

Figure.05 – Swallow sample

As we can observe from the plot in Figure.05, pressure changes recorded by Sensor1, Sensor2 & Sensor3 follow a general pattern during the swallowing. Pressure spike is observed in the Sensor3,

at the start and the end of the swallow. During the swallow, pressure in the sensor3 drops to a minimum value at some point. At the same time, Pressure peaks can be observed in the Sensor1 and Sensor2. This pattern is generally observed in all the samples and the presence of this pattern has been used to implement the algorithm that can detect the start and endpoint of the swallow and extract all the recordings within this duration.

Step.03 Evaluating features of a Sample:

The features of a Swallowing sample, that is, Sensor1_max, Sensor3_max, Sensor3_max, Latency, UES_start, UES_end & UES_duration were evaluated for each and every sample and recordings from few samples are shown in the Table.04.

	Patient	Sensor1_max	Sensor2_max	Sensor3_min	Latency	UES_start	UES_end	UES_Duration
2	P21_F_25.txt	95.925841	107.818723	-8.940261	0.056	802.164	802.988	0.824
3	P21_F_25.txt	95.934997	119.299611	-9.709316	0.044	663.288	664.464	1.176
9	P21_F_25.txt	87.475395	100.860609	1.075761	0.156	845.412	846.276	0.864
10	P21_F_25.txt	81.954681	116.168460	-8.418402	0.048	564.732	565.860	1.128
14	P21_F_25.txt	81.560998	116.955825	-7.264820	0.012	164.852	165.708	0.856
16	P22_F_24.txt	137.051957	124.838636	-1.771572	0.168	286.720	287.304	0.584
18	P22_F_24.txt	107.626459	81.863127	-2.229343	0.104	92.756	93.416	0.660
19	P22_F_24.txt	145.136187	120.434882	-6.871138	0.060	277.976	278.604	0.628
22	P22_F_24.txt	102.627604	119.922179	-11.705196	0.100	264.536	265.192	0.656

Table.04. Features of Swallowing Sample

Data Modelling:

The structure of the data that has been used in the modelling process is shown in Table.05.

	Patient	Sensor1_max	Sensor2_max	Sensor3_min	Latency	UES_Duration	Swallow type
0	Control 2.txt	101.393912	199.782559	-11.066606	0.132	0.624	Healthy
1	Control 2.txt	110.988785	143.783474	-9.734493	0.088	0.544	Healthy
2	Control 3.txt	164.236667	100.936141	-13.469902	0.268	0.812	Healthy
3	Control 3.txt	104.612039	186.672007	-17.383841	0.248	0.988	Healthy
4	Control 5.txt	113.895628	158.551156	-14.614328	0.088	0.644	Healthy
5	HF1-E.txt	137.317464	119.958801	-11.723507	0.309	0.792	Healthy
6	HF10-HL.txt	108.303960	45.131609	-13.014420	0.079	0.716	Healthy

Table.05.Structure of Data

Exploratory Data Analysis:

The descriptive statistics for 2 different classes in the data, that is, Healthy and Unhealthy are listed in Table.06 and Table.07.

	Sensor1_max	Sensor2_max	Sensor3_min	Latency	UES_Duration
count	124.000000	124.000000	124.000000	124.000000	124.000000
mean	124.158939	129.857740	-9.502119	0.145363	0.845387
std	44.465992	53.531787	7.791889	0.128389	0.222639
min	80.599680	44.151980	-33.376059	-0.172000	0.436000
25%	92.439345	89.732204	-13.897917	0.056000	0.664000
50%	110.516137	119.357977	-9.127947	0.150000	0.828000
75%	137.118334	161.465438	-3.650721	0.239250	0.981000
max	274.703593	250.000000	8.228428	0.532000	1.892000

Table.06. Descriptive statistics of Healthy Samples

	Sensor1_max	Sensor2_max	Sensor3_min	Latency	UES_Duration
count	120.000000	120.000000	120.000000	120.000000	120.000000
mean	144.082960	145.538535	-13.641965	0.159658	0.814517
std	51.947881	79.275492	7.535588	0.103103	0.199220
min	80.756810	47.114770	-48.677570	-0.142000	0.420000
25%	98.738570	88.117125	-17.293390	0.104500	0.671000
50%	134.162300	127.453150	-12.117235	0.166500	0.786500
75%	170.505650	174.013475	-8.784615	0.217500	0.955000
max	309.871200	435.149900	1.642600	0.573000	1.601000

Table.07. Descriptive statistics of Unhealthy Samples

In addition, the correlation matrix for key independent features in the data is shown in Figure.06

	Sensor1_max	Sensor2_max	Sensor3_min	Latency	UES_Duration
Sensor1_max	1	-0.0346022	-0.038871	0.149575	0.0296155
Sensor2_max	-0.0346022	1	-0.124292	0.0283056	0.056447
Sensor3_min	-0.038871	-0.124292	1	-0.0584012	0.063469
Latency	0.149575	0.0283056	-0.0584012	1	0.316552
UES_Duration	0.0296155	0.056447	0.063469	0.316552	1

Figure.06 – Correlation matrix

It is evident from the correlation matrix that independent variables are not significantly correlated with each other. The density plots for each of the independent features shown in Figure.07 gives a general idea of the probability distribution of the data.

Density Plots

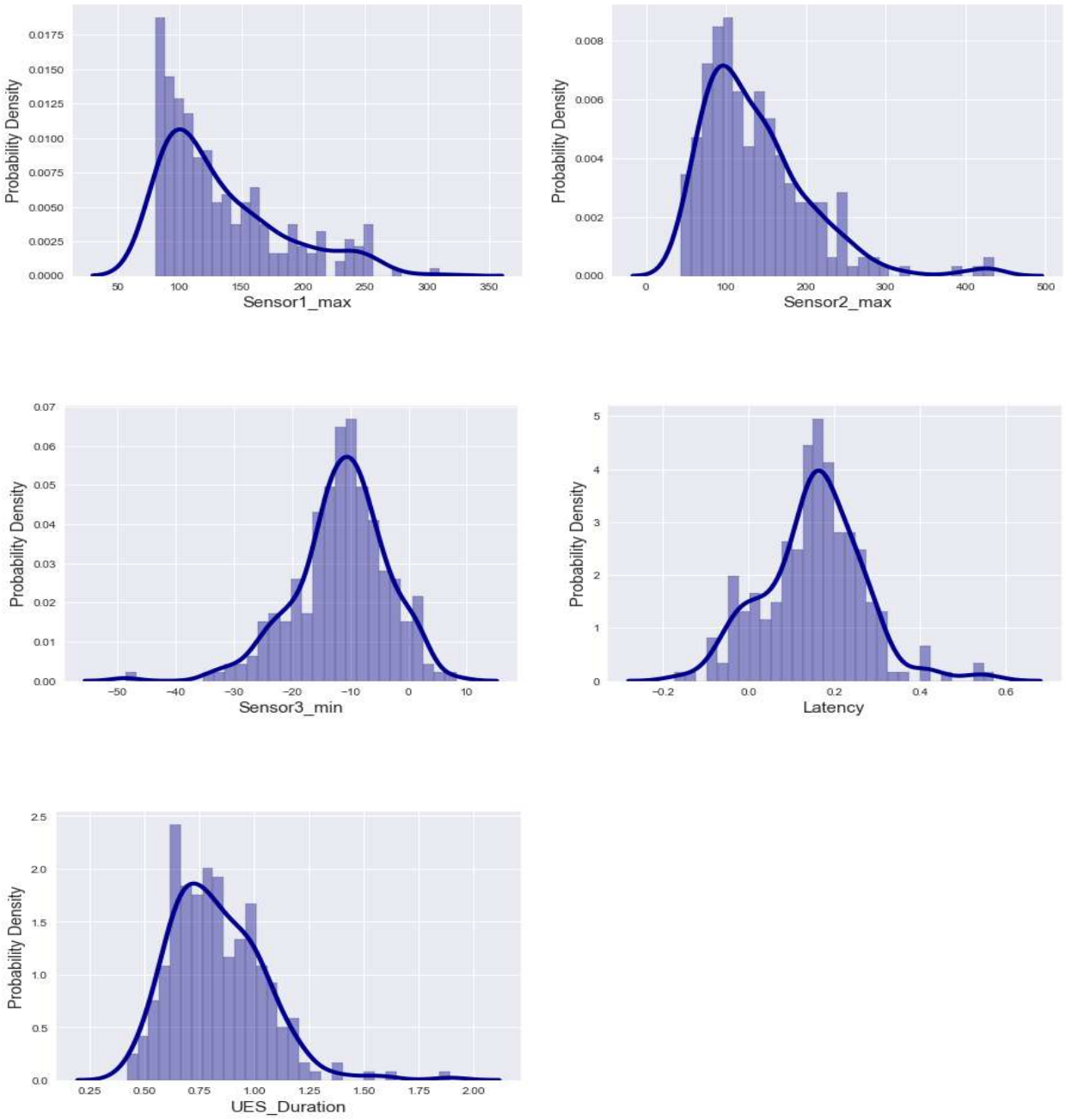

Figure.07 – Density plots for Independent features

Training and Test data:

The composition of training and testing dataset is shown in Table.08.

Composition	***Training data***	***Testing data***
No of Healthy swallow samples	*86*	*38*
No of Unhealthy swallow samples	*84*	*36*
Total samples	***170***	***74***

Table.08. Composition of Training and Test Data

The performance of classification algorithms trained using training dataset and default hyperparameters and evaluated using test dataset is recorded as below:

Logistic Regression:
The confusion matrix for Logistic Regression model is shown in Table.09.

Confusion Matrix		*Predicted*	
		Healthy	*Unhealthy*
Actual	*Healthy*	*24*	*14*
	Unhealthy	*10*	*26*

Table.09. Confusion Matrix for Logistic Regression model

Table.10 describes the evaluation metrics of logistic regression model:

	Precision	***Recall***	***F1-score***	***Support***
Healthy	*0.71*	*0.63*	*0.67*	*38*
Unhealthy	*0.65*	*0.72*	*0.68*	*36*
Accuracy			*0.68*	*74*
Macro avg	*0.68*	*0.68*	*0.68*	*74*
Weighted avg	*0.68*	*0.68*	*0.68*	*74*

Table.10. Evaluation metrics for Logistic Regression model

The overall accuracy of the model is 68%. However, the focus of this project is to detect an unhealthy patient so that no Unhealthy patient is left undetected in the process. Hence, the recall value of the model for the "Unhealthy" class is much more significant than Overall accuracy. The recall value of 72% for the "Unhealthy" class suggests that the trained model is capable of detecting an unhealthy patient 72% of the time and 28% of patients are still left undetected by the model. It is also evident from the confusion matrix that the model has detected 26 Unhealthy patients correctly from a total of 36 Unhealthy patients i.e. 72% detection rate.

Support Vector Classifier:

The confusion matrix for Support Vector Classifier is shown in Table.11.

Confusion Matrix		*Predicted*	
		Healthy	*Unhealthy*
Actual	*Healthy*	*36*	*2*
	Unhealthy	*32*	*4*

Table.11. Confusion Matrix for Support Vector Classifier

Table.12 describes the evaluation metrics of Support Vector Classifier:

	Precision	***Recall***	***F1-score***	***Support***
Healthy	*0.53*	*0.95*	*0.68*	*38*
Unhealthy	*0.67*	*0.11*	*0.19*	*36*
Accuracy			*0.54*	*74*
Macro avg	*0.60*	*0.53*	*0.43*	*74*
Weighted avg	*0.60*	*0.54*	*0.44*	*74*

Table.12. Evaluation metrics for Support Vector Classifier

The overall accuracy of the model is 54%. However, it can be noted that the recall value of the model for the "Unhealthy class" is extremely poor, suggesting that the model has detected only 11% of the Unhealthy patients, and 89% of unhealthy patients are classified incorrectly. It is also evident from the Confusion matric that the model has detected 4 Unhealthy patients correctly from a total of 36 Unhealthy patients i.e. 11% detection rate. Overall, the performance of the Support Vector Classifier trained with default hyperparameters is unsatisfactory. Hyperparameter tuning was attempted to improve the performance and results after hyperparameter tuning have been recorded in the separate section on hyperparameters.

Random Forest Classifier:

The confusion matrix for random forest classification model is shown in Table.13.

Confusion Matrix		*Predicted*	
		Healthy	*Unhealthy*
Actual	*Healthy*	*26*	*12*
	Unhealthy	*11*	*25*

Table.13. Confusion Matrix for Random Forest Classifier

The Table.14 describes the evaluation metrics of Random forest Classifier:

	Precision	***Recall***	***F1-score***	***Support***
Healthy	*0.70*	*0.68*	*0.69*	*38*
Unhealthy	*0.68*	*0.69*	*0.68*	*36*
Accuracy			*0.69*	*74*
Macro avg	*0.69*	*0.69*	*0.69*	*74*
Weighted avg	*0.69*	*0.69*	*0.69*	*74*

Table.14. Evaluation metrics for Random Forest Classifier

The overall accuracy of the model is 69%. A recall value of 69% for the "Unhealthy" class suggests that the model has detected 69% of the Unhealthy patients and 31% of unhealthy patients remain undetected. It is also evident from the confusion matrix that the random forest classifier has classified 25 unhealthy patients correctly from the total of 36 unhealthy patients i.e. 69% detection rate.

XGBoost:

The confusion matrix for XGBoost classification model is shown in Table.15

Confusion Matrix		*Predicted*	
		Healthy	*Unhealthy*
Actual	*Healthy*	*21*	*17*
	Unhealthy	*8*	*28*

Table.15. Confusion Matrix for XGBoost Classifier

Table.16 describes the evaluation metrics of XGBoost Classifier:

	Precision	***Recall***	***F1-score***	***Support***
Healthy	*0.72*	*0.55*	*0.63*	*38*
Unhealthy	*0.62*	*0.78*	*0.69*	*36*
Accuracy			*0.66*	*74*
Macro avg	*0.67*	*0.67*	*0.66*	*74*
Weighted avg	*0.67*	*0.66*	*0.66*	*74*

Table.16. Evaluation metrics for Random Forest Classifier

The overall accuracy of the model is 66%. Recall value of 78% for the "Unhealthy" class suggests that the model has detected 78% of the Unhealthy patients and 22% of unhealthy patients remain undetected. It is also evident from the confusion matric that the XGBoost classifier has classified 28 unhealthy patients correctly from a total of 36 unhealthy patients i.e. 78% detection rate.

LightGBM:

The confusion matrix for LightGBM Classifier is shown in Table.17

Confusion Matrix		*Predicted*	
		Healthy	*Unhealthy*
Actual	*Healthy*	*19*	*19*
	Unhealthy	*12*	*24*

Table.17. Evaluation Metrics for LightGBM Classifier

Table.18 describes the evaluation metrics of XGBoost Classifier:

	Precision	***Recall***	***F1-score***	***Support***
Healthy	*0.61*	*0.50*	*0.55*	*38*
Unhealthy	*0.56*	*0.67*	*0.61*	*36*
Accuracy			*0.58*	*74*
Macro avg	*0.59*	*0.58*	*0.58*	*74*
Weighted avg	*0.59*	*0.58*	*0.58*	*74*

Table.18. Confusion Matrix for LightGBM Classifier

The overall accuracy of the model is 58%. The recall value of 67% for the "Unhealthy" class suggests that the model has detected 67% of the Unhealthy patients and 33% of unhealthy patients remain undetected. It is also evident from the confusion matric that the LightGBM classifier has classified 24 unhealthy patients correctly from a total of 36 unhealthy patients i.e. 67% detection rate.

Performance Summary:

The Table.19 summarizes the performance of the models trained with training data and default hyperparameter values, which were then tested using test dataset.

S. No	***Classification Model***	***Accuracy***	***Precision***	***Recall***
1	*Logistic Regression*	*68%*	*65%*	*72%*
2	*Support Vector Classifier*	*54%*	*67%*	*11%*
3	*Random Forest*	*69%*	*68%*	*69%*
4	*XGBoost*	*66%*	*62%*	*78%*
5	*LightGBM*	*58%*	*66%*	*67%*

Note: Precision and Recall values are specified for Class 1 i.e. unhealthy patients

Table.19. Performance summary of Classification models

It is evident from the results that Overall accuracy is comparable in Logistic regression, Random forest, and XG boost classifiers, whereas it is relatively low in the Support Vector classifier. Precision values are comparable in all the models. However, our main focus is on recall values, which is indicative of the fact that how effective the model has been able to detect Unhealthy Patients. In terms of recall value, the XGBoost classifier has performed better than other models. Recall values are comparable in the Random forest and LightGBM Classifier. However, the Performance of the Support Vector classifier in terms of recall value is unsatisfactory.

Hyperparameter tuning:

In the previous section, we have trained and tested different machine learning algorithms for the classification of swallow samples as "Healthy" or "Unhealthy". It was observed that none of the models performed exceptionally well, in terms of detecting Unhealthy patients. However, the recall value for the "Unhealthy" class in Logistic regression, Random Forest, and XGBoost Classifier was impressive. In real-life scenarios, we would hesitate to implement an algorithm, which classifies even a single Unhealthy patient as healthy, as we are dealing with an illness that can end up in a life-threatening situation for some patients if left undetected. However, these algorithms can still be extremely powerful and can have a significant contribution in reducing the load on the medical system of any country or organization, by substantially reducing the cost and resources for achieving the underlying goal. In this section, we have performed hyperparameter tuning to improve the performance of these models in two stages:

Stage 1:

In Stage 1, the main objective in this section is to evaluate optimum parameters that can improve the recall value of these models while maintaining an optimum level of precision to recall trade-off. The hyperparameter tuning is implemented using a "RandomizedSearchCV" class in the "scikit-learn" library in python. The range of values for different parameters is given as input and scoring criteria are defined in such a way that "RandomizedSearchCV" can find the optimum value of these parameters, thereby maximizing the AUC (Area under the ROC Curve) score of a model. The hyperparameter tuning is implemented in combination with 6 fold cross-validation. The AUC score is evaluated using 6 fold cross-validation for every possible combination of parameters and a set of parameters that deliver a maximum AUC score for a model have been identified.

The hyperparameters that have been tuned for each of these models are listed in Table.20. The table displays the default values of these parameters as well as best performing parameter values identified from the list of values passed to the model.

S. No	Hyper-parameters	Default values	List of values passed to Model	Best Performing Parameters
Logistic Regression				
1	C	1	[0.001, 0.01, 0.1, 1, 10, 100, 1000]	1
2	penalty	l2	['l1','l2']	l1
3	max_iter	100	[100,200,300,400,500,600, 700,800]	200
4	solver	warn	['liblinear', 'saga']	liblinear
Support Vector Classifier				
1	C	1	[0.1, 1, 10, 100, 1000]	1
2	gamma	"auto_deprecated"	[1, 0.1, 0.01, 0.001, 0.0001, 0.00001]	0.0001
3	max_iter	-1	[100,200,300,400,500,600, 700,800,900]	800
Random Forest Classifier				
1	bootstrap	True	[True, False]	True
2	max_depth	None	[2, 3, 4, 5,10, 20, 30, 40, 50, 60, 70, 80, 90, 100, None]	2
3	max_features	sqrt	['auto', 'sqrt']	sqrt
4	min_samples_leaf	1	[1, 2, 4]	2
5	min_samples_split	2	[2, 5, 10]	2
6	n_estimators	100	[200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]	1800
XGBoost Classifier				
1	learning_rate	0.30	[0.05, 0.10, 0.15, 0.20, 0.25, 0.30]	0.05
2	max_depth	6	[3,4,5,6,7,8,10,12]	6
3	min_child_weight	1	[1,3,5,7]	7
4	gamma	0	[0.0,0.1,0.2,0.3,0.4]	0.3
5	colsample_bytree	1	[0.3,0.4,0.5,0.7]	0.5
6	n_estimators	100	[100,200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]	600
LightGBM Classifier				
1	max_depth	-1	[4, 6, 10]	10
2	min_child_weight	0.001	[1,3,5,7]	3
3	learning_rate	0.1	[0.1, 0.01, 0.001,]	0.1
4	colsample_bytree	1	[0.3,0.4,0.5,0.7]	0.4
5	n_estimators	100	[200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]	1000
6	num_iteration	NA	[3,5,7,9,12]	12

Table.20. List of Hyperparameters

The significance of hyperparameters tuned for trained models is explained below:

"C" Parameter:
"C" is inverse regularization parameter ($C = 1/\lambda$) which is used to regulate the influence that independent features have on the output of the model. It is regulated to avoid the problem of

overfitting and value chosen are intended to minimize the loss function. Coefficients of the independent features are penalized with a regularization parameter.

Penalty:
Penalty defines the form of regularization that will be used to penalize the effect of the features. In the L1 penalty, the regularization term of the cost function is evaluated using the absolute value of weights whereas, in the L2 penalty, the squared value of weights is used.

max-iter/n-iter:
It defines the number of iterations allowed for a model.

Solver:
It defines the type of the loss function that has been used in the model, to minimize the error.

gamma:
gamma parameter defines the smoothness of the decision boundary that the model will generate between the two classes. A large value of gamma results in a smoother boundary whereas low gamma can result in pointed bumps in the decision boundary. This parameter holds significance in the problems where different classes in the data are not linearly separable.

Bootstrap:
In random forest, decision trees are trained by selecting samples from data. The bootstrap parameter defines a way of sampling. If bootstrap is true, samples are drawn from the overall dataset with replacement. This means that sample is returned to the overall data before the next sample is chosen. However, if bootstrap is False, once the sample is drawn, It is removed from the data and the next sample cannot have any repeat observations.

max_depth:
This parameter is used to limit the number of nodes allowed from the root to the farthest leaf of a decision tree in Random forest or XGBoost classifier

max_features:
This parameter defines the maximum number of features that can be considered for splitting each node in a decision tree.

min_samples_leaf:
This parameter defines the minimum number of samples that should be at leaf node in a decision tree.

min_samples_split:
This parameter defines the minimum number of samples required to split an internal node in a decision tree.

n_estimators:
This parameter helps to define the number of decision trees the model should generate, before arriving at an output.

learning_rate
In the XGBoost classifier, the model is improved iteratively. At every iteration, the model improves itself by trying to minimize the loss function and the learning rate parameter can help to define step size at every iteration while moving towards the lowest possible value of the loss function.

min_child_weight
It defines the minimum weight (which is basically the number of samples if all samples have a weight of 1) required to split a node in a decision tree. If weight is less than minimum child weight, the algorithm will stop further splitting.

colsample_bytree
This parameter helps to limit the percentage of features that can be used to train a decision tree in a model.

Table.21 illustrates the performance of the classification algorithms trained with best-performing hyperparameter values and 6 fold cross-validation:

Logistic Regression:	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*0.70*	*0.61*	*0.65*	*38*
Unhealthy	*0.63*	*0.72*	*0.68*	*36*
Accuracy			*0.66*	*74*
Macro avg	*0.67*	*0.66*	*0.66*	*74*
Weighted avg	*0.67*	*0.66*	*0.66*	*74*
Support Vector Classifier:	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*0.68*	*0.74*	*0.71*	*38*
Unhealthy	*0.70*	*0.64*	*0.67*	*36*
Accuracy			*0.69*	*74*
Macro avg	*0.69*	*0.69*	*0.69*	*74*
Weighted avg	*0.69*	*0.69*	*0.69*	*74*

Random Forest Classifier:	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*0.65*	*0.53*	*0.58*	*38*
Unhealthy	*0.58*	*0.69*	*0.63*	*36*
Accuracy			*0.61*	*74*
Macro avg	*0.61*	*0.61*	*0.61*	*74*
Weighted avg	*0.61*	*0.61*	*0.61*	*74*
XGBoost Classifier:	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*0.66*	*0.55*	*0.60*	*38*
Unhealthy	*0.60*	*0.69*	*0.64*	*36*
Accuracy			*0.62*	*74*
Macro avg	*0.63*	*0.62*	*0.62*	*74*
Weighted avg	*0.63*	*0.62*	*0.62*	*74*
LightGBM Classifier:	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*0.71*	*0.63*	*0.67*	*38*
Unhealthy	*0.65*	*0.72*	*0.68*	*36*
Accuracy			*0.68*	*74*
Macro avg	*0.68*	*0.68*	*0.68*	*74*
Weighted avg	*0.68*	*0.68*	*0.68*	*74*

Table.21. Performance of Classification algorithms after hyperparameter tuning

Table.22 summarize the performance of classification algorithms trained with default hyperparameters as well as best performing hyperparameters

S. No	***Classification Model***	***Default hyperparameters***			***Best performing hyperparameters***		
		Accuracy	***Precision***	***Recall***	***Accuracy***	***Precision***	***Recall***
1	*Logistic Regression*	*68%*	*65%*	*72%*	*66%*	*63%*	*72%*
2	*Support Vector Classifier*	*54%*	*67%*	*11%*	*69%*	*70%*	*64%*
3	*Random Forest*	*69%*	*68%*	*69%*	*61%*	*58%*	*69%*
4	*XGBoost*	*66%*	*62%*	*78%*	*62%*	*60%*	*69%*
5	*LightGBM*	*58%*	*66%*	*67%*	*68%*	*65%*	*72%*

Note: Precision and Recall values are specified for Class 1 i.e. unhealthy patients

Table.22. Performance summary of Classification algorithms

It is evident from the results that the detection rate i.e. recall value is improved in Support Vector and LightGBM classifier whereas no significant improvement is noticed in the case of logistic regression and random forest. In the case of XGBoost, the detection rate has declined from 78% to 69%. The performance metrics after hyperparameter tuning are evaluated based on 6 fold cross-validation method, in order to avoid the problem of overfitting. Hence, these figures portray the realistic performance of these models. The performance of all the models is comparable, however,

Logistic regression and LightGBM Classifier have demonstrated a relatively better detection rate (72%).

Stage 2:

In stage 2, we have made an effort to improve the recall value of all the classification models by using hyperparameter tuning, to such an extent that, no unhealthy patient is wrongly classified as healthy. This will lead to a decline in the recall value of the models for the "Healthy" class, which means that a lot of healthy patients would get classified as Unhealthy. The objective of this form of the model is to detect 100% of Unhealthy patients and at the same time identify some percentage of the healthy patients, which have close to zero probability of having a swallowing disorder. It can be implemented in real-life scenarios for the initial screening process and to save the required resources and thereby cost of the associated organization.

The classification algorithms used in the modeling process are probability-based which means that the sample is classified as healthy or unhealthy according to the probability of a sample falling in either of these two classes. The default probability threshold used by algorithms to classify a particular sample is 0.5 which gives an equal possibility for the samples to belong to either class 0 or class 1. However, we are mainly concerned about the detection of unhealthy patients in this case as it would be objectionable to classify them as healthy. Therefore, the probability threshold is regulated in order to achieve a 100% recall for "Unhealthy. The probability thresholds used for classification models is shown in Table.23.

S. No	***Model***	***Default Probability threshold***	***New Probability threshold***
1	*Logistic Regression*	*0.5*	*0.37*
2	*Support Vector classifier*	*0.5*	*0.41*
3	*Random forest classifier*	*0.5*	*0.39*
4	*XGBoost Classifier*	*0.5*	*0.22*
5	*LightGBM Classifier*	*0.5*	*0.30*

Table.23. Performance summary of Classification algorithms

The performance of the models evaluated with revised probability thresholds is recorded in Table.24.

Logistic Regression	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*1.00*	*0.21*	*0.35*	*38*
Unhealthy	*0.55*	*1.00*	*0.71*	*36*
Accuracy			*0.59*	*74*
Macro avg	*0.77*	*0.61*	*0.53*	*74*
Weighted avg	*0.78*	*0.59*	*0.52*	*74*
Support Vector Classifier	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*1.00*	*0.03*	*0.05*	*38*

Unhealthy	*0.49*	*1.00*	*0.66*	*36*
Accuracy			*0.50*	*74*
Macro avg	*0.75*	*0.51*	*0.36*	*74*
Weighted avg	*0.75*	*0.50*	*0.35*	*74*
Random Forest Classifier	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*1.00*	*0.21*	*0.35*	*38*
Unhealthy	*0.55*	*1.00*	*0.71*	*36*
Accuracy			*0.59*	*74*
Macro avg	*0.77*	*0.61*	*0.53*	*74*
Weighted avg	*0.78*	*0.59*	*0.52*	*74*
XGBoost Classifier	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*1.00*	*0.24*	*0.38*	*38*
Unhealthy	*0.55*	*1.00*	*0.71*	*36*
Accuracy			*0.61*	*74*
Macro avg	*0.78*	*0.62*	*0.55*	*74*
Weighted avg	*0.78*	*0.61*	*0.54*	*74*
LightGBM Classifier	***Precision***	***Recall***	***F1-score***	***Support***
Healthy	*1.00*	*0.13*	*0.23*	*38*
Unhealthy	*0.52*	*1.00*	*0.69*	*36*
Accuracy			*0.55*	*74*
Macro avg	*0.76*	*0.57*	*0.46*	*74*
Weighted avg	*0.77*	*0.55*	*0.45*	*74*

Table.24. Performance summary of Models with revised threshold

As we can observe from the evaluation metrics of these classification models, after hyperparameter tuning, the Recall value of all the models for the "Unhealthy" class is 100% which is indicative of the fact that models have classified 100% of Unhealthy patients correctly. However, the recall value of Healthy class in these models suggests the percentage of Healthy patients who have almost no probability of having any swallowing disorder and have been ruled out by the model with high confidence. Recall value of Healthy class in Logistic regression (21%), Random forest (21%), XGboost (24%) determines how efficient the model has performed in terms of the screening process. These figures suggest that models can rule out 21-24% of patients who have no probability of swallowing disorder and requires no further evaluation. Hence, they have the potential to reduce the resources and cost by 21-24%. Similarly, in the case of LightGBM and Support Vector classifier, this percentage stands at 13% and 3% respectively.

Conclusion:

The detection of Swallowing disorders demands exhaustive evaluation and Analysis by medical practitioners thereby making it an expensive and time-consuming process that requires a lot of resources. In spite of this, swallowing disorders remain undetected in a sizeable number of cases,

which might be due to improper screening or human error in evaluation. In this research, an attempt was made to train machine learning algorithms that can help to improve the conventional way of

dealing with patients suffering from swallowing disorders. The manometry test records consisting of 124 Healthy and 120 unhealthy samples were collected and machine learning algorithms i.e. Logistic Regression, Support Vector classifier, Random forest classifier, XGBoost & LightGBM classifier were trained using these records. Models were able to deliver a reasonable level of accuracy. The best performing models i.e. Logistic regression & LightGBM are capable of detecting 72% of unhealthy patients. However, considering the sensitive nature of the problem that is being dealt with here, non-detection of the remaining 28% of patients in the real-life scenario, can be a serious issue. In the second scenario, models were retrained with hyperparameter tuning and regulating the probability threshold in such a way that, all the Unhealthy Patients are classified correctly and at the same time, the model can rule out some healthy patients, which have no probability of swallowing disorder. For instance, if there are 100 patients to be tested, and the model can rule out 20% of healthy patients with high confidence, only 80% of patients would require detailed evaluation. This will significantly reduce the resources required for the entire process. If the performance of the model is evaluated in terms of resource benefits, Classification models such as Logistic regression (21%), Random forest (21%), XGboost (24%) hold the potential to reduce the overall resources by 21-24%.

Future Work:

In this research, we have successfully trained models that can classify the swallow sample as “Healthy” or “Unhealthy” based on the pressure readings of a group of healthy and Unhealthy patients. However, the effectiveness of the model can further be enhanced by making improvisations in some areas which are illustrated as follows:

This research utilizes data from high-resolution manometry, in which sensors are placed linearly at a distance of 1cm, inside the tube. This tube is inserted into the esophagus and pressure recordings are collected from the patient’s throat at an interval of 1 ms. In order to make the detection much more efficient and precise, data could be collected using advanced techniques such as high definition manometry. In high definition manometry, sensors are placed even closer than the 1cm standard of high-resolution manometry. In addition to this, sensors are placed in the form of rings around the esophagus, which helps to record radial characteristics at each axial position. Overall, high definition manometry has an enhanced level of spatial resolution compared to high-resolution manometry.

In addition to the above, the potential exists to advance the modeling, by improving the process of selecting samples from raw data, which are used to train the models. In this research, we have opted for a path of random sampling for this process. However, during a manometry test, clinicians usually collect three different types of samples. The patient is instructed to swallow a “saliva”, “10ml bolus swallowing”, and in some cases, the patient is also instructed to perform “Effortful swallowing”. The pressure changes/spikes get recorded for each type of swallow in the overall

data. However, the process of tagging a swallow as “Saliva swallow”, “10ml bolus swallow” and “Effortful swallow” is something that is done manually by the clinicians and it doesn’t get recorded in the manometry readings. Hence, for this research, there was no means to identify the nature of the swallow, when samples are selected for modeling. In future research, this distinction can be made for different kinds of swallows, during sampling, in order to keep an equal proportion of these different kinds of swallows in the training data.

References:

[1] American Speech-Language-Hearing Association. (n.d.). *Adult Dysphagia*. (Practice Portal). Retrieved month, day, year, from www.asha.org/Practice-Portal/Clinical-Topics/Adult-Dysphagia/.

[2] Lamvik, Kristin, et al. "Normative Data for Pharyngeal Pressure Generation during Saliva, Bolus, and Effortful Saliva Swallowing Across Age and Gender." Speech, Language and Hearing, vol. 17, no. 4, 2014, pp. 210-215.

[3] Molfenter, Sonja M., et al. "Physiological Variability in the Deglutition Literature: Hyoid and Laryngeal Kinematics." Dysphagia, vol. 26, no. 1, 2011, pp. 67-74.

[4] Molfenter, Sonja M., and Catriona M. Steele. "Kinematic and Temporal Factors Associated with Penetration–Aspiration in Swallowing Liquids." Dysphagia, vol. 29, no. 2, 2014, pp. 269-276.

[5] Macrae, Phoebe R., et al. "Pharyngeal Pressures during Swallowing within and Across Three Sessions: Within-Subject Variance and Order Effects." Dysphagia, vol. 26, no. 4, 2011, pp. 385-391.

[6] van Herwaarden, Margot A., et al. "Are Manometric Parameters of the Upper Esophageal Sphincter and Pharynx Affected by Age and Gender?" Dysphagia, vol. 18, no. 3, 2003, pp. 211-217.

[7] Mielens, Jason D., et al. "Application of Classification Models to Pharyngeal High-Resolution Manometry." Journal of Speech, Language, and Hearing Research, vol. 55, no. 3, 2012, pp. 892-902.

[8] Jones, C. A., et al. "Identification of Swallowing Disorders in Early and mid-stage Parkinson's Disease using Pattern Recognition of Pharyngeal high-resolution Manometry Data." Neurogastroenterology and Motility, vol. 30, no. 4, 2018, pp. e13236-n/a.

[9] Ranta, Annemarei. “Projected Stroke Volumes to Provide a 10-Year Direction for New Zealand Stroke Services.” The New Zealand Medical Journal, 22 June 2018, www.nzma.org.nz/journal-articles/projected-stroke-volumes-to-provide-a-10-year-direction-for-new-zealand-stroke-services#:%7E:text=When%20comparing%20New%20Zealand%20to,76%20per%20100%2C00

0%20in%20Adelaide.&text=Mortality%20was%20also%20higher%20in,other%20countries%20such%20as%20Australia.

[10] González-Fernández, M., Ottenstein, L., Atanelov, L. et al. Dysphagia after stroke: an overview. Curr Phys Med Rehabil Rep 1, 187–196 (2013). https://doi.org/10.1007/s40141-013-0017-y

[11] Martino R, Foley N, Bhogal S, Diamant N, Speechley M, Teasell R. Dysphagia after stroke: Incidence, diagnosis, and pulmonary complications. Stroke. 2005;36(12):2756–2763. doi: 10.1161/01.STR.0000190056.76543.eb.

[12] Kahrilas, Peter J et al. "Challenging the limits of esophageal manometry." Gastroenterology vol. 134,1 (2008): 16-8. doi:10.1053/j.gastro.2007.11.031