# DIAGNOSING COPD USING SUPERVISED MACHINE LEARNING

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

**M.Manoj (39110605)**

**S. Mohammed Afrid (39110631)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**
**(DEEMED TO BE UNIVERSITY)**
**Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE**
**JEPPIAAR NAGAR, RAJIV GANDHISALAI,**
**CHENNAI - 600119**

**APRIL - 2023**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **M.Manoj (39110605)** who carried out Project Phase-2 entitled **"DIAGNOSING COPD USING SUPERVISED MACHINE LEARNING"** under my supervision from January 2023 to April 2023.

RajaSree

**Internal Guide**

**Dr. RAJA SHREE S, M.E., Ph.D.**

**Head of the Department**

**Dr. L. LAKSHMANAN, M.E., Ph.D.**

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

**Submitted for Viva-voce Examination held on 20.04.2023**

**Internal Examiner** **External Examiner**

# DECLARATION

**I, M.Manoj (Reg.No - 39110605)** hereby declare that the Project Phase-2 Report entitled **“DIAGNOSING COPD USING SUPERVISED MACHINE LEARNING”** done by me under the guidance of **Dr. RAJA SHREE S, M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

M.Manoj

**DATE: 20.04.23**

**PLACE: CHENNAI** **SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D**, **Dean**, School of Computing, and **Dr. L. Lakshmanan M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Raja Shree S, M.E., Ph.D.,** for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

# ABSTRACT

Recent developments in computer operations and development of advanced artificial intelligence (AI) and machine learning (ML) technology and their applications in various fields such as medicine are increased rapidly. Medical data are difficult to capture, manage, and process using conventional tools in a timely manner because the datasets are huge, they are frequently updated, and the data come in diverse formats. Our project delves into the realm of clinical decision support systems within the healthcare industry with a specific focus on respiratory diseases such as Asthma and chronic obstructive pulmonary disease (COPD). The prevention, diagnosis, and treatment of these ailments are of utmost importance, particularly in regard to preventing exacerbation and determining the severity of the disease during hospitalization. The need for such measures is a global initiative, especially for COPDpatients and is only available during the stable-phase of the disease. This is where AI systems come into play, as traditional methods take too long for accurate prognosis. Machine-learning techniques have been proven effective in crucial healthcare applications and we have applied four supervised machine-learning algorithms - Random Forest Classifier, Naive Bayes, Decision Tree classifier and Logistic Regression - to aid respiratory physicians in estimating the severity of COPD patients in the early stages, thus guiding the cure strategy. It is crucial to detect and manage COPD in its early stages to greatly improve the quality of life for patients and reduce the burden on healthcare systems. By utilizing supervised machine-learning techniques, we can provide better care for those suffering from respiratory diseases and ultimately improve the overall health of the population.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Chronic Obstructive Pulmonary Disease (COPD), a respiratory condition, can cause considerable morbidity and death, if not identified and treated right once. Early detection of Chronic Obstructive Pulmonary Disease is crucial in avoiding the illness from advancing to a more severe state. Today, spirometry is the gold standard diagnostic test for Chronic Obstructive Pulmonary Disease. Spirometry is difficult to use in environments with limited resources because it needs specialized equipment and trained personnel. Hence, the creation of precise, non-invasive, and affordable diagnostic techniques is crucial.

For the diagnosis of Chronic Obstructive Pulmonary Disease, supervised machine learning algorithms have shown promise. Large data sets may be analyzed by machine learning algorithms to find patterns and generate precise predictions. The application of supervised machine learning algorithms for the diagnosis of COPD will be reviewed .

Ultrasound insonation of lungs that are dense with extravascular lung water (EVLW) produces characteristic reverberation artifacts termed B-lines. The number of B-lines present demonstrates reasonable correlation to the amount of EVLW. However, analysis of B-line artifacts generated by this modality is semi-quantitative relying on visual interpretation, and as a result, can be subject to inter-observer variability. The purpose of this study was to translate the use of a novel, quantitative lung ultrasound surface wave elastography technique (LUSWE) into the bedside assessment of pulmonary edema in patients admitted with acute congestive heart failure.

In the event of COPD, your wind current is diminished because of at least one issues. Lessening the seriousness of respiratory diseases and pneumonia. The divider between the numerous airbags imploded. The dividers of the street are thick and thick. The aviation routes produce a larger number of breaks than ordinary and are interfered. Emphysema influences the air sacs in the lungs and the dividers between them. They are harmed and won't change. Ongoing bronchitis is a sickness where the respiratory lot is continually consuming and consuming. She would balloon and conceive an offspring. The reason for COPD is the harm to your lungs and your breath and the drawn-out openness to it. Tobacco smoke is a significant reason in the United States. Smoke, tobacco, and different kinds of tobacco smoke can cause COPD, particularly when breathed in. Openness to other respiratory variables can add to COPD.

To prevent this problem in One of the most interesting (or perhaps most profitable) time series data using machine learning techniques. Hence, pulmonary disease prediction has become an important research area. The aim is to predict machine learning based techniques for pulmonary disease prediction results in best accuracy.

The analysis of dataset by supervised machine learning technique (SMLT) to capture several information’s like, variable identification, uni-variate analysis, bi-variate and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preparing and data visualization will be done on the entire given dataset.

To propose a machine learning-based method to accurately predict the pulmonary disease Index value by prediction results in the form of pulmonary disease classification best accuracy from comparing supervise classification machine learningalgorithms.

Additionally, to compare and discuss the performance of various machine learning algorithms from the given dataset with evaluation classification report, identify the confusion matrix and to categorizing data from priority and the result shows that the effectiveness of the proposed machine learning algorithm technique can be compared with best accuracy with precision, Recall and F1 Score.

# CHAPTER 2

# LITERATURE SURVEY

This problem statement has been extensively studied over the past 5 years by researchers and in medical field in bid to create a solution, and all their solutions vary from analyzing various patterns for the improvement of data.

E. YatishVenkata Chandra, K. Ravi Teja, M.Hari Chandra Siva Prasad, Mohammed.Ismail.B (2019) [4] says about The major cause for death in human beings is because of cancer .Lung cancer is one of the most common and serious types of cancer that severely harms the human body. In order to cure the cancer earlycancer detection is required. If lung cancer is diagnosed at early stages many lives will be saved. The other name for lung cancer is lung carcinoma, an uncontrolled malignant tumor distinguished by undisciplined cell growth in lung cells. There are many people suffering from this kind of cancer and confining to death. If this is left untreated, this may grow later than lung by metastasis into other parts of body. Many of the cancers starts from lungs, called as primary lung carcinoma.

The work of R. Moshavegh, K. L. Hansen, H. Moller-Sorensen, M. B. Nielsen, and J. A. Jensen, (2019) [5] says about the Lung Cancer could be a Disease of uncontrolled cell growth in tissues of the lung. Discovery of carcinoma in its initial stage is that the key of its cure. All in all, a measure for earlier than schedule stage lung disease determination essentially incorporates those using X-beam midsection movies, CT, MRI so forth.

Dr. Niranjan, S. K. (2018) [6] says about the major cause of deaths in human beings is Lung Cancer, Since the lung cancer symptoms appear in the advanced stages so it is hard to detect which leads to high mortality rate among other cancer types. Hence the early prediction of lung cancer is mandatory for the diagnosis process and it gives the higher chances for successful treatment. It is the most challenging way to enhance a patient’s chance for survival.

The work of M. Jozwiak, J.-L. Teboul, and X. Monnet (2015) [11] says about Cancer is a disease that is unregulated by cells in the body. Lung nodule is called lung cancer because the disease starts in the lungs. Cancer of the pulmonary system begins in the lungs and may travel to lymph nodes or other body species such as the brain.

The work of N. Sudhir Reddy, V Khanaa (2014) says about One of the major causes of cancer death is through Lung cancer. The overlapping of cancer cells acts as an impediment for its early detection. Identifying genetic and environmental factors plays a vital role in developing better techniques for its prevention. And in order to discover the anomalies in target images, time factor is paramount.

## 2.1 INFERENCES FROM LITERATURE SURVEY

The algorithms used to detect the pulmonary disease are Decision tree, k-Nearest neighbour, random forest, Logistic regression. In this paper by implementing 2 different datasets and various packages and libraries in python, it is compared and on implementation found suitable algorithms have more accuracy on certain data sets for optimum prediction rate of COPD.

The classification is performed and the results were evaluated with the performance comparison of various algorithms. This prediction system is useful for the doctors to take an appropriate decision based on patient’s condition. In this work, decision tree algorithm is used for prediction of COPD where in the important pattern with their corresponding weightage and score is studied.

## 2.2 OPEN PROBLEMS IN EXSISTING SYSTEM

- They are not classifying pulmonary disease on machine learning classification technique and not mention any accuracy results.
- There are not using any artificial intelligence technique
- It can’t thereby better determine the regularity of pulmonary disease prediction data and achieve more accurate prediction results.

## 2.3 OVERALL OBJECTIVE

The main goal of COPD (Chronic Obstructive Pulmonary Disease) diagnosis using machine learning is to create a precise and trustworthy model that may help with early detection and treatment of the condition. COPD is a progressive disease that affects the lungs and causes breathing difficulties, and it can lead to significant morbidity and mortality. To halt the disease's course and enhance patient outcomes, early identification and treatments are crucial.

Machine learning algorithms can be trained on large datasets of patient information, including demographic, clinical, and medical imaging data, to identify patterns and risk factors associated with COPD. The model can then be used to predict the likelihood of developing COPD in individuals with specific risk factors or symptoms.

Through earlier diagnosis, more precise risk assessment, and focused therapies, this mission ultimately aims to enhance patient outcomes. This may result in more individualized and efficient therapies, lessen the financial strain the condition places on healthcare systems, and ultimately enhance the quality of life for COPD sufferers.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## 3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

The project plan and objectives are well-defined, and the implementation has been successful up to the application level without encountering any issues. Moreover, the project risk is negligible, implying that the project can be carried out in the next phase with a wide scope. Currently, four classification algorithms have been utilized in the project, and their accuracy and performance have been used to predict the client-side data. There is also a possibility of improving the algorithms in the future. The built applications have low risk, and the project is feasible, meeting the deadline for submission. The feasibility study's objective is to determine if the application is technically, economically, operationally, and market feasible to ensure its longevity and generate a good return on investment. The findings from the feasibility study can be utilized to inform future decisions.

### *3.1.1 Feasibility Studies*

Feasibility study for diagnosing COPD using Supervised Machine Learning would help to determine whether the proposed project is viable and can be successfully implemented. It would help to identify any potential challenges and risks and provide insights into how these challenges can be addressed. A thorough feasibility study is essential to ensure the success of the project and to maximize the potential benefits and returns on investment.

- Technical feasibility: This involves evaluating whether the proposed project is technically feasible and whether the necessary technical expertise and resources are available. In the case of image forgery detection using ML, this would include assessing the availability of appropriate datasets, the required computing power and software, and the availability of trained personnel with the necessary skills to develop and implement this ML model.

- Economic feasibility: This involves assessing the financial viability of the project. This would include analyzing the costs involved in developing and implementing the ML model, such as hardware and software costs, personnel costs, and any other expenses that might be incurred.

- Operational feasibility: This involves assessing whether the proposed project can be successfully integrated into existing operational systems and processes. In the case of diagnosing COPD using Supervised Machine Learning, this would include assessing whether the proposed software can be integrated with existing image processing systems and whether the necessary protocols can be established to ensure smooth operation.

### *3.1.2 Risk Analysis*

Risk analysis for diagnosing COPD using Supervised Machine Learning involves identifying potential risks that could impact the success of the project and developing strategies to mitigate these risks. Some of the potential risks that may arise during diagnosing COPD using Supervised Machine Learning are as follows:

- Data quality risk: The quality of the data used to train the ML model may not be sufficient, leading to inaccurate predictions. To mitigate this risk, it is essential to ensure that the data used is high-quality and relevant to the problem at hand.

- Overfitting risk: Overfitting can occur when the ML model is trained on a small dataset or when it is trained for too long, resulting in poor generalization and inaccurate predictions. To mitigate this risk, it is essential to use a large dataset and to employ techniques such as early stopping to prevent overfitting.

- Hardware failure risk: Hardware failure, such as server or storage system failure, can result in data loss and system downtime. To mitigate this risk, it is essential to have backup systems in place and to regularly test and maintain the hardware.

- Cybersecurity risk: Cybersecurity risks such as hacking or data breaches can compromise the integrity of the data used to train this ML model or the predictions made by the model. To mitigate this risk, it is essential to implement robust cybersecurity measures such as firewalls, encryption, and access controls.

- False positive or false negative risk: The ML model may produce false positives or false negatives, resulting in inaccurate predictions. To mitigate this risk, it is essential to regularly test and refine the ML model and to use a combination of different algorithms and techniques to improve its accuracy.

- Model performance risk: The performance of the ML model may degrade over time due to changes in the input data or changes in the operating environment. To mitigate this risk, it is essential to regularly monitor and evaluate the performance of the ML model and to make necessary adjustments to ensure its continued accuracy.

## 3.2 REQUIREMENT PROCESS OF THE PROJECT

The requirement process for a project aimed at predicting COPD (Chronic Obstructive Pulmonary Disease) would typically involve the following steps:

Define the Problem Statement: The first step is to define the problem statement and determine the scope of the project. In this case, the problem statement would be to develop a predictive model that can accurately identify patients at risk of developing COPD.

Gather Data: The next step is to gather relevant data that can be used to train the predictive model. This would involve collecting patient data such as medical history, family history, lifestyle factors, environmental factors, and other relevant information.

Data Preprocessing: Once the data has been collected, it needs to be preprocessed and cleaned to remove any inconsistencies, missing values, or errors. This step is crucial as the accuracy of the predictive model depends heavily on the quality of the data used to train it.

Feature Selection: After the data has been preprocessed, the next step is to select the relevant features that will be used to train the predictive model. This step involves identifying the most significant factors that contribute to the risk of developing COPD.

Model Selection: Once the features have been selected, the next step is to choose an appropriate machine learning algorithm to train the predictive model. There are various algorithms available, such as logistic regression, decision trees, random forests, and neural networks, each with its strengths and weaknesses.

Model Training: The selected model is then trained on the preprocessed data using the chosen algorithm. The model's performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score.

Model Tuning: The model may be tuned by adjusting hyperparameters such as learning rate, regularization, and activation functions to improve its performance.

Model Testing: Once the model has been trained and tuned, it needs to be tested on new data to evaluate its real-world performance. This step involves using a hold-out dataset or cross-validation techniques to assess the model's accuracy, precision, recall, and F1-score.

Model Deployment: Finally, the model is deployed into a production environment, where it can be used to predict the risk of COPD in new patients. This step involves integrating the model into a software application or a web service that can be accessed by healthcare providers or patients.

In conclusion, the requirement process for a project aimed at predicting COPD involves a series of steps that include defining the problem statement, gathering and preprocessing data, selecting relevant features, choosing an appropriate machine learning algorithm, training and tuning the model, testing its performance, and deploying it into a production environment.

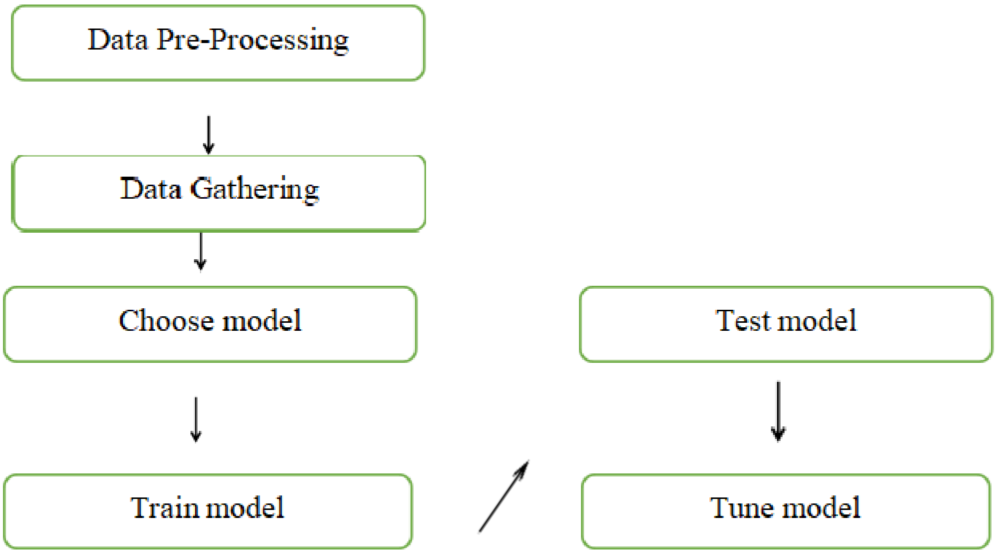
### *3.2.1. Building the classification model*

The prediction of COPD, A high accuracy prediction model is effective because of the following reasons: It provides better results in classification problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.
- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

### *3.2.2. Construction of a Predictive Model*

Machine learning needs data gathering have lot of past data's. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can't be used directly. It's used to pre-process then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.



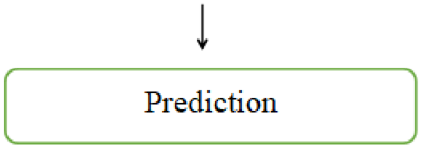


***Fig 3.1: Process of dataflow diagram***

### *3.2.3. Python Packages*

- Scikit-learn

  Scikit-learn is a machine learning library that provides tools for data pre- processing, model selection, and performance evaluation. It is widely used for developing classification and regression models, including those based on CNNs.

- NumPy

  NumPy is a package for numerical computing in Python, often used for handling and manipulating large arrays of data. It provides efficient tools for matrix and array operations, which are often used in deep learning applications.

- Pandas

  Pandas is a package for data analysis in Python, often used for data cleaning, exploration, and manipulation. It provides a powerful set of data structures and functions for working with structured data, which are often used in deep learning applications.

- Matplotlib

  Matplotlib is a plotting library for creating visualizations in Python, often used for displaying images and data visualizations. It provides a wide range of customizable plots, including scatter plots, bar charts, and histograms, which are often used in image processing tasks.

These packages, along with others that may be specific to the project requirements, can be used to develop and implement an Illegal Image Identification using CNN.

## 3.3 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

This document describes the Software Requirements Specification for a COPD prediction system using machine learning. Using a variety of medical data inputs, this system's objective is to anticipate the beginning of COPD (chronic obstructive pulmonary disease) in individuals. The system will analyze the input data using machine learning methods to forecast the start of COPD.

### *3.3.1 Purpose of this document*

The purpose of a paper on predicting COPD using machine learning is to investigate the possibilities of utilizing computer algorithms and statistical models to identify people who may be at risk of developing chronic obstructive pulmonary disease (COPD). Millions of individuals worldwide suffer from COPD, a progressive lung illness, and early identification and treatment have been shown to greatly improve patient outcomes.

Large datasets of patient data can be used to train machine learning algorithms to find patterns and risk factors related to COPD. These algorithms can forecast a person's chance of acquiring COPD in the future by looking at information such patient demographics, medical history, smoking status, and lung function tests. This can be used to identify people who are at risk and could benefit from specific therapies like early detection screens or smoking cessation programmes. In general, the objective of utilizing machine learning to predict COPD is to improve patient outcomes through early detection of at-risk patients and provision of suitable therapies to stop or delay the disease's progression.

### *3.3.2 Scope of this document*

The scope of the project is to integrate clinical decision support with computer-based patient records to reduce medical errors, enhance patient safety, decrease unwanted practice variation and to improve patient outcomes by identifying individuals who are at risk of developing COPD and providing appropriate interventions to prevent or delay the onset of the disease.

### *3.3.3 Functional Requirements*

- Data Collection:

  The system should be able to collect data related to a patient’s medical history, lifestyle habits, and demographic information.

- Data Preprocessing:

  The system should be able to preprocess the collected data to remove any missing or irrelevant data, and convert it into a format suitable for machine learning models.

- Machine Learning Model:

  The system should include a machine learning model that can learn from the preprocessed data and make predictions about the likelihood of the patient developing COPD in the future.

- Prediction:

  The system should be able to predict the likelihood of the patient developing COPD in the future based on the machine learning model.

### *3.3.4 Non-Functional Requirements*

Accuracy: The system should be able to predict the likelihood of the patient developing COPD with a high degree of accuracy.

Reliability: The system should be reliable and provide consistent results across multiple runs.

Security: The system should be secure and protect the privacy of patient data.

Scalability: The system should be scalable and able to handle large amounts of data.

Usability: The user interface should be easy to use and navigate.

### 3.4. Hardware Requirements

• Processor: Intel Core i5-11400F up to 4.5 GHz.

• Memory: 8 GB DDR4.

• Hard Drives: 512 GB NVMe2 SSD.

• GPU: NVIDIA GeForce GTX 1650 4 GB.

• Computing Power: 7.5

• Ports: 1x HDMI 2.0, 1x USB 3.1 Type-C, 2x USB 3.1, 1x USB 2.0.

• Connectivity: Wi-Fi 802.11ax, Gigabit LAN (Ethernet), Bluetooth.

### 3.5. Software Requirements

• OS: Windows 10/11 Home.

• Python 3.9 or Later.

• Pandas

• NumPy

- Anaconda with Jupyter Notebook

# CHAPTER 4
# DESCRIPTION OF THE PROPOSED SYSTEM

## 4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python's Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It’s important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it’s important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

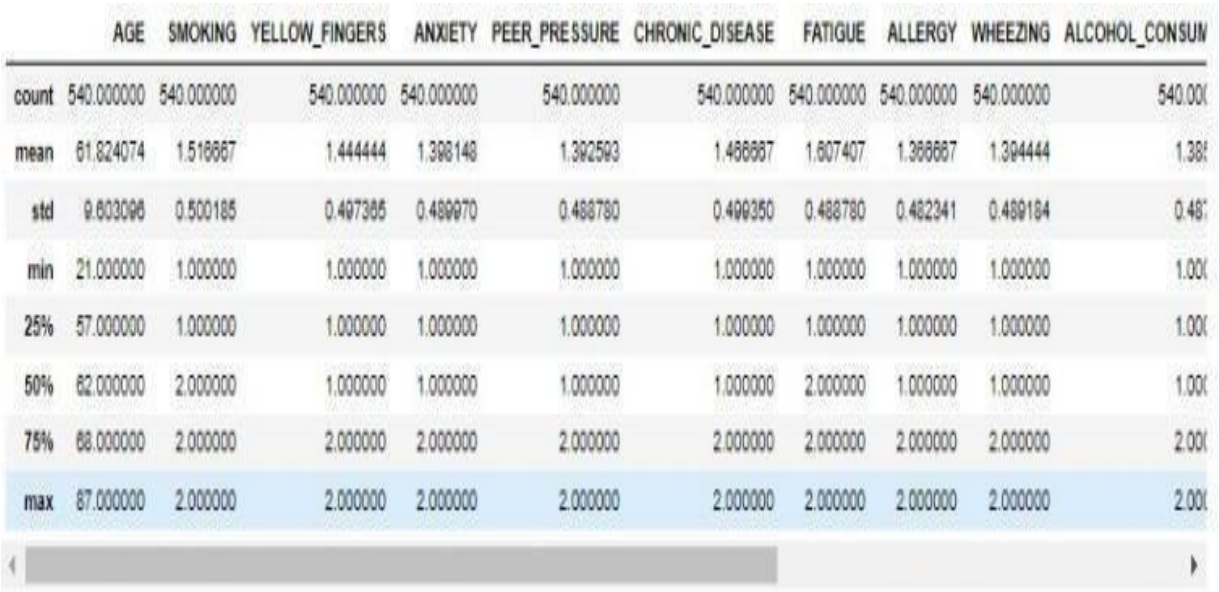
Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame
- Checking data type and information about dataset

- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

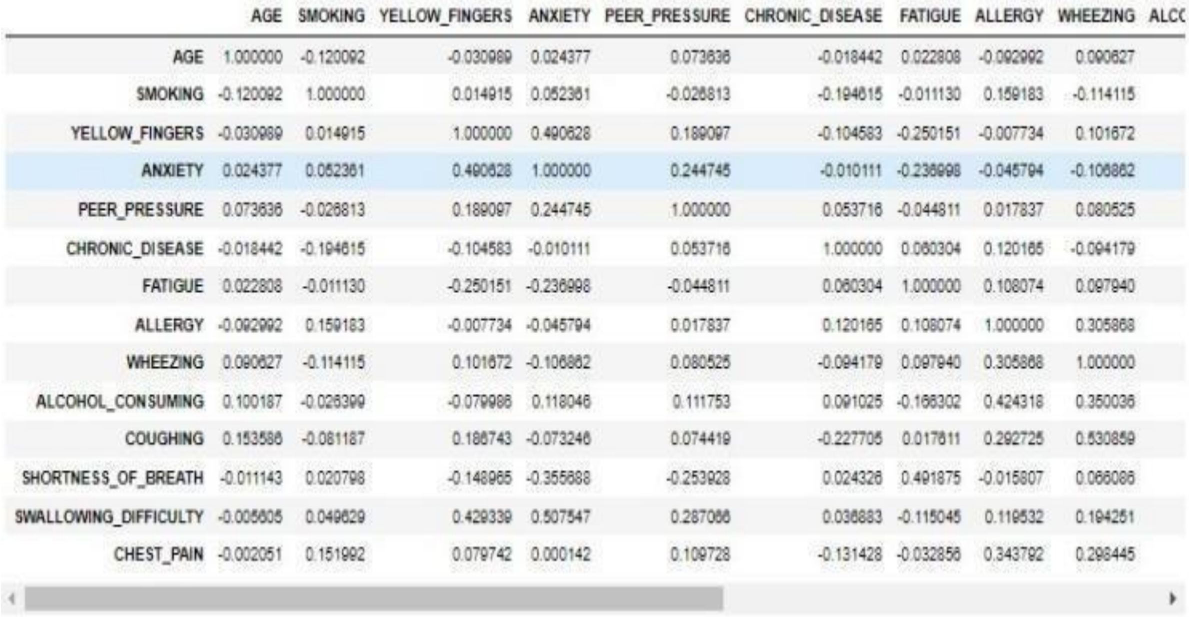
Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models.

Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.



| | AGE | SMOKING | YELLOW_FINGERS | ANXIETY | PEER_PRESSURE | CHRONIC_DISEASE | FATIGUE | ALLERGY | WHEEZING | ALCOHOL_CONSUM |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 540.000000 | 540.000000 | 540.000000 | 540.000000 | 540.000000 | 540.000000 | 540.000000 | 540.000000 | 540.000000 | 540.00 |
| mean | 61.824074 | 1.516667 | 1.444444 | 1.398148 | 1.392593 | 1.466667 | 1.607407 | 1.366667 | 1.394444 | 1.38 |
| std | 9.603096 | 0.500185 | 0.497365 | 0.489970 | 0.488780 | 0.499350 | 0.488780 | 0.482341 | 0.489184 | 0.48 |
| min | 21.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.00 |
| 25% | 57.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.00 |
| 50% | 62.000000 | 2.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 2.000000 | 1.000000 | 1.000000 | 1.00 |
| 75% | 68.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.00 |
| max | 87.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.000000 | 2.00 |

***Fig 4.1: data analysis***



| | AGE | SMOKING | YELLOW_FINGERS | ANXIETY | PEER_PRESSURE | CHRONIC_DISEASE | FATIGUE | ALLERGY | WHEEZING | ALCC |
|---|---|---|---|---|---|---|---|---|---|---|
| AGE | 1.000000 | -0.120092 | -0.030989 | 0.024377 | 0.073636 | -0.018442 | 0.022808 | -0.092992 | 0.090627 | |
| SMOKING | -0.120092 | 1.000000 | 0.014915 | 0.052361 | -0.026813 | -0.194615 | -0.011130 | 0.159183 | -0.114115 | |
| YELLOW_FINGERS | -0.030989 | 0.014915 | 1.000000 | 0.490628 | 0.189097 | -0.104583 | -0.250151 | -0.007734 | 0.101672 | |
| ANXIETY | 0.024377 | 0.052361 | 0.490628 | 1.000000 | 0.244745 | -0.010111 | -0.236998 | -0.045794 | -0.106862 | |
| PEER_PRESSURE | 0.073636 | -0.026813 | 0.189097 | 0.244745 | 1.000000 | 0.053716 | -0.044811 | 0.017837 | 0.080525 | |
| CHRONIC_DISEASE | -0.018442 | -0.194615 | -0.104583 | -0.010111 | 0.053716 | 1.000000 | 0.060304 | 0.120165 | -0.094179 | |
| FATIGUE | 0.022808 | -0.011130 | -0.250151 | -0.236998 | -0.044811 | 0.060304 | 1.000000 | 0.108074 | 0.097940 | |
| ALLERGY | -0.092992 | 0.159183 | -0.007734 | -0.045794 | 0.017837 | 0.120165 | 0.108074 | 1.000000 | 0.305868 | |
| WHEEZING | 0.090627 | -0.114115 | 0.101672 | -0.106862 | 0.080525 | -0.094179 | 0.097940 | 0.305868 | 1.000000 | |
| ALCOHOL_CONSUMING | 0.100187 | -0.026399 | -0.079986 | 0.118046 | 0.111753 | 0.091025 | -0.166302 | 0.424318 | 0.350036 | |
| COUGHING | 0.153586 | -0.081187 | 0.186743 | -0.073246 | 0.074419 | -0.227705 | 0.017611 | 0.292725 | 0.530859 | |
| SHORTNESS_OF_BREATH | -0.011143 | 0.020798 | -0.148965 | -0.355688 | -0.253928 | 0.024326 | 0.491875 | -0.015807 | 0.066086 | |
| SWALLOWING_DIFFICULTY | -0.005605 | 0.049629 | 0.429339 | 0.507547 | 0.287066 | 0.036883 | -0.115045 | 0.119532 | 0.194251 | |
| CHEST_PAIN | -0.002051 | 0.151992 | 0.079742 | 0.000142 | 0.109728 | -0.131428 | -0.032856 | 0.343792 | 0.298445 | |

***Fig:4.2 Data Validation***

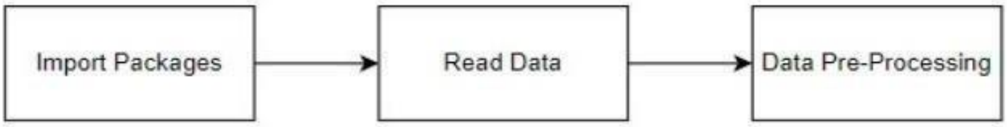


***Fig:4.3 Module Diagram for data pre-processsing***

input : data

output : removing noisy data

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

- How to chart time series data with line plots and categorical quantities with bar charts.
- How to summarize data distributions with histograms and box plots.

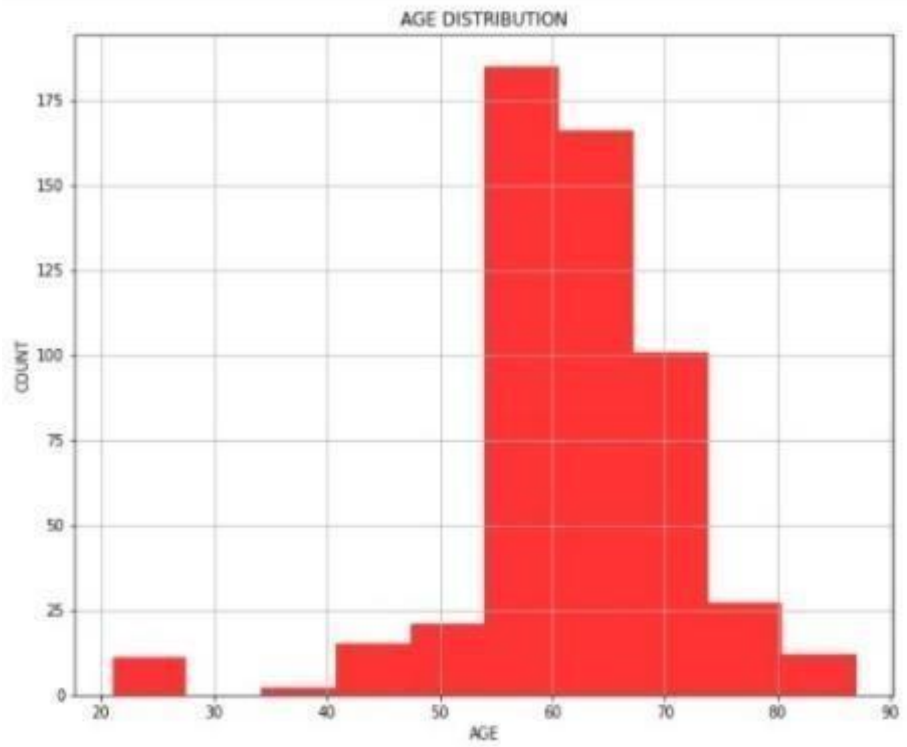


***Fig:4.4 Histogram Diagram***

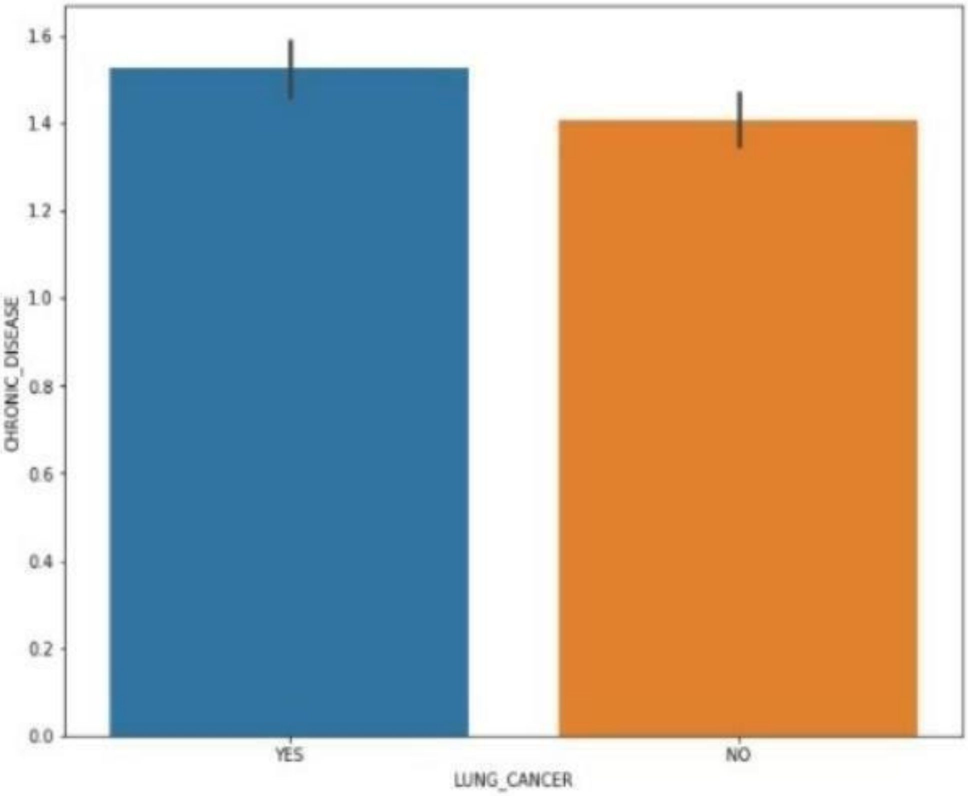


***Fig:4.5 Bar Plot Diagram***

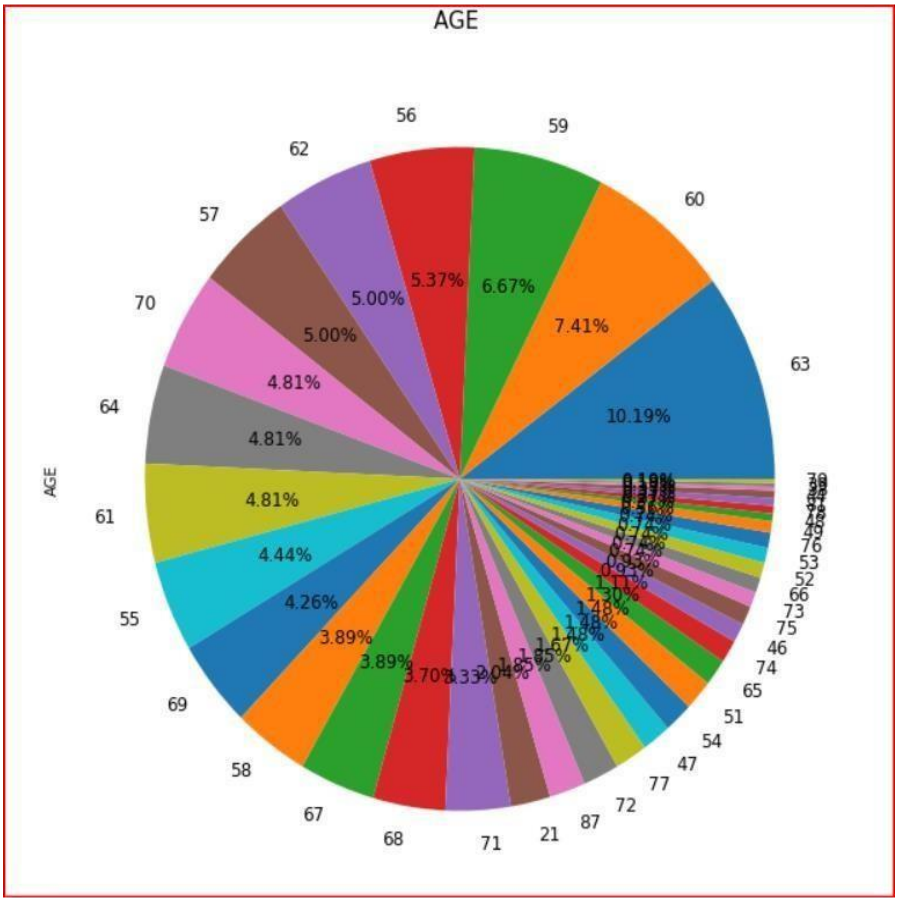


*Fig 4.6 Pie Plot Diagram*

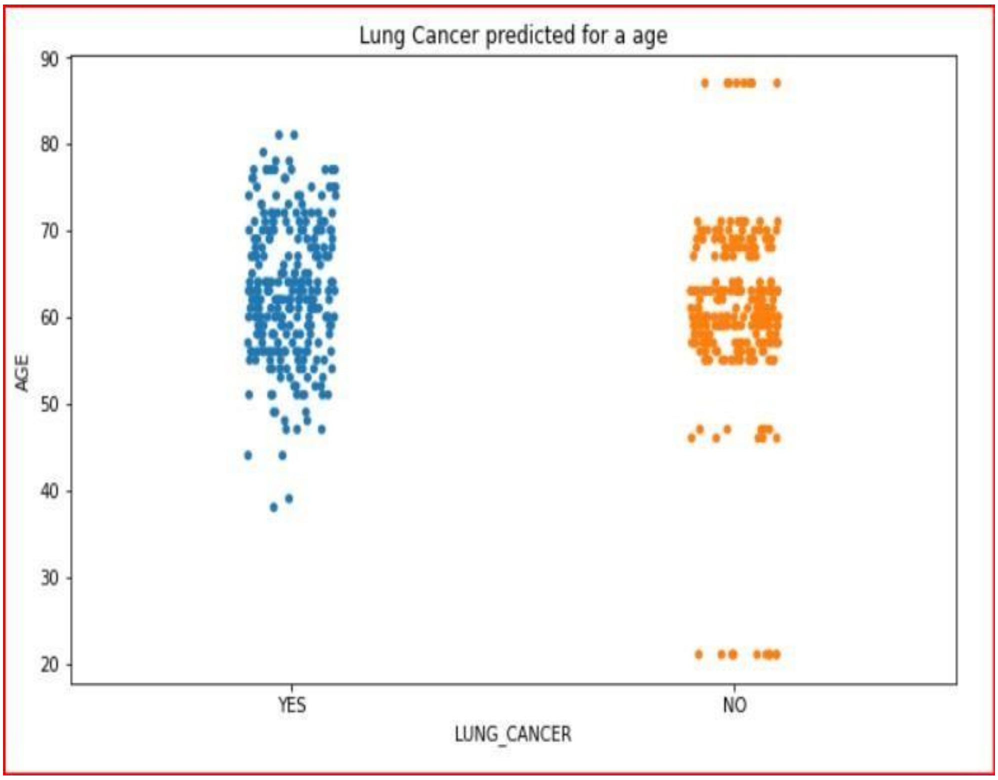


*Fig:4.7 Strip Plot Diagram*

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values.

Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

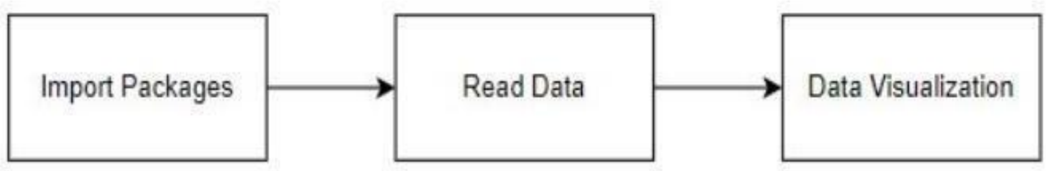


***Fig:4.8 Module Diagram for data visualisation***

input : data
output : visualized data

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data.

It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

- Logistic Regression
- Random Forest
- Decision Tree Classifier
- Naïve Bayes

### 4.2 ARCHITECTURE OF THE PROPOSED SYSTEM

**SYSTEM ARCHITECTURE:**

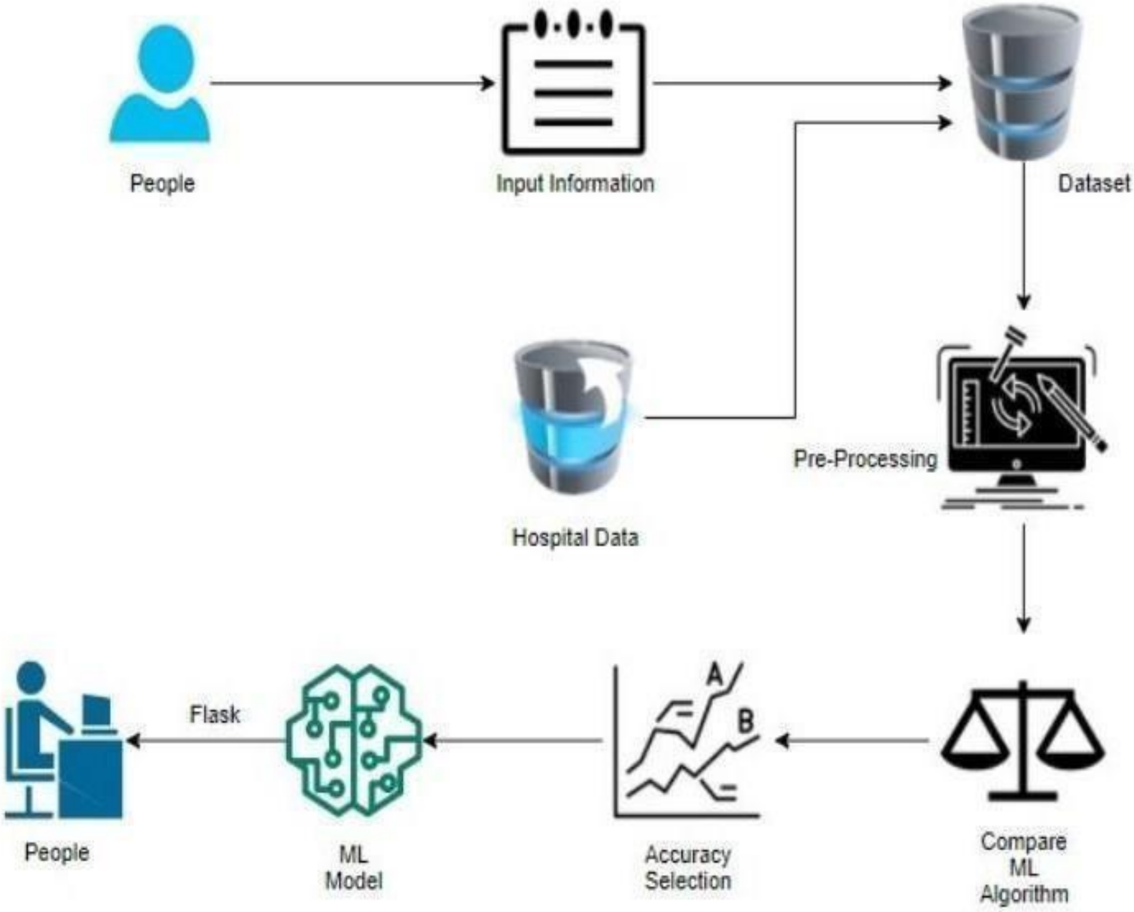


***Fig:4.9 system architecture***

Data Collection Component: This component will be responsible for collecting data related to a patient’s medical history, lifestyle habits, and demographic information.

Data Preprocessing Component: This component will preprocess the collected data to remove any missing or irrelevant data, and convert it into a format suitable for machine learning models.

Machine Learning Component: This component will include the machine learning model that can learn from the preprocessed data and make predictions about the likelihood of the patient developing COPD in the future.

Prediction Component: This component will be responsible for predicting the likelihood of the patient developing COPD in the future based on the machine learning model.

## 4.3 USE CASE DIAGRAM:

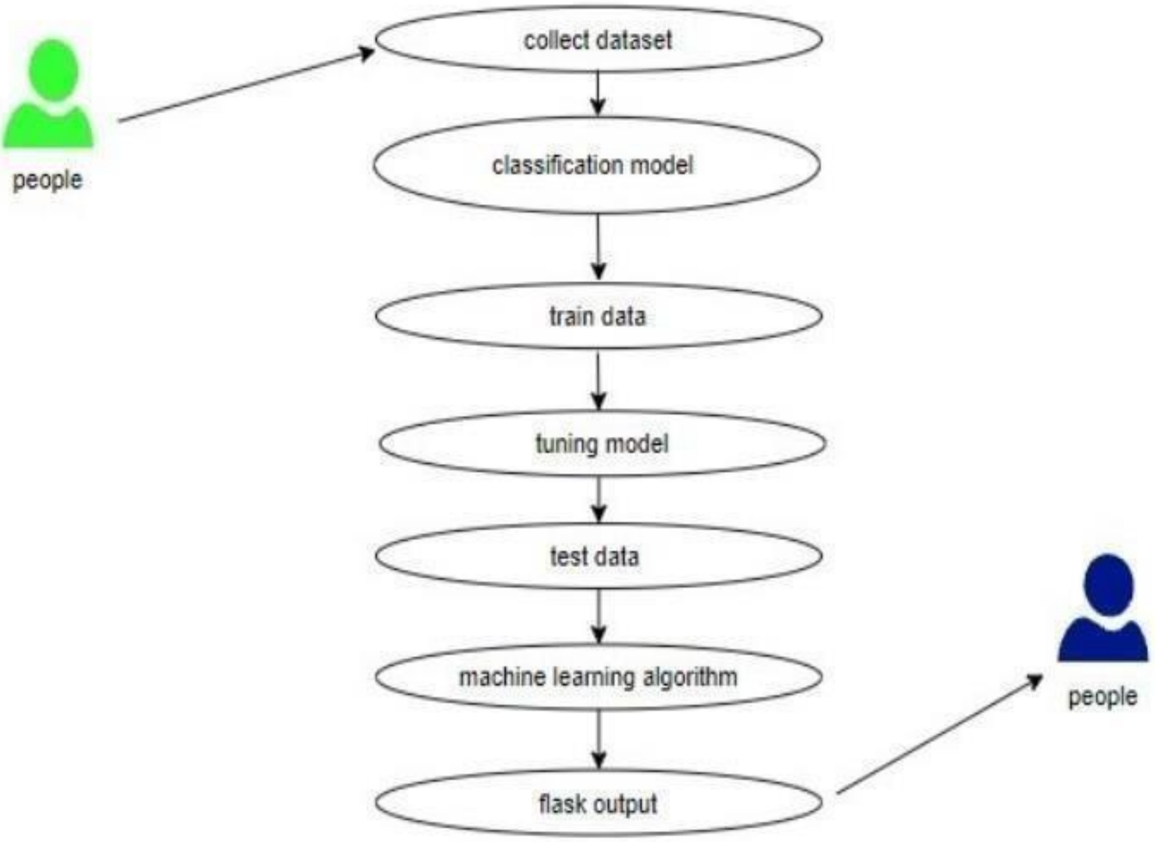


***Fig:4.10 Usecase diagram***

The Doctor who diagnose the required parameter which you collect data from the doctor and add in excel in their respected field and then, you use your classification for your dataset to preprocess your data and then you decide the deciding factors of detection of COPD. Tuned model involved by tuned time to time with improving the accuracy. Then you try the trained system with custom input for checking results, this involves the algorithms used to implement the machine learning part, which is tested with the custom input. The output is then displayed and verified and by the user.

## 4.4 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Data Collection Module: This module collects data from patients that includes demographic information, medical history, lifestyle habits, and lung function tests.

Data Pre-processing Module: This module cleans and pre-processes the collected data to remove any missing values, outliers, or redundant information.

Feature Selection Module: This module selects relevant features from the pre-processed dataset to improve the accuracy of the prediction model.

Machine Learning Algorithm Selection Module: This module selects the appropriate machine learning algorithm for the prediction task. Popular algorithms for classification tasks include Decision Trees, Random Forests, Logistic regression, Naïve Bayes.

Model Training Module: This module trains the selected machine learning algorithm on the pre-processed dataset using a suitable training strategy, such as k-fold cross-validation.

User Interface Module: This module provides a user-friendly interface for the software system, allowing patients to input their data and receive predictions about their COPD risk.

Deployment Module: This module deploys the software system in a secure and scalable environment, ensuring that it complies with ethical and legal regulations related to patient data privacy and security.

Continuous Improvement Module: This module continuously monitors the performance of the software system and improves it through updates and improvements.

### *4.4.1 Testing Plan of the proposed system*

Unit Testing: This step involves testing individual components of the software system, such as the data pre-processing module, feature selection module, machine learning algorithm selection module, model training module, model evaluation module, and user interface module. Unit testing helps to identify and fix bugs in the code early in the development process.

Integration Testing: This step involves testing the integration of the individual components of the software system. Integration testing helps to ensure that the different modules of the software system work together as expected.

Acceptance Testing: This step involves testing the software system with real-world data to ensure that it meets the requirements and specifications of the stakeholders. Acceptance testing helps to validate the accuracy and reliability of the software system.

Performance Testing: This step involves testing the performance of the software system under different conditions, such as varying data sizes, network speeds, and user loads. Performance testing helps to identify any performance issues and bottlenecks in the software system.

Security Testing: This step involves testing the security of the software system to ensure that it complies with ethical and legal regulations related to patient data privacy and security.

Usability Testing: This step involves testing the usability and user experience of the software system. Usability testing helps to ensure that the software system is user-friendly and easy to use.

## 4.5 PROJECT MANAGEMENT PLAN

Define the project scope: Determine the purpose and objectives of the project, and the expected outcomes. Identify the stakeholders who will be involved in the project, including patients, medical professionals, researchers, and others.

Identify project risks: Determine the potential risks associated with the project, including technical, financial, and other risks. Develop a risk management plan to address these risks.

Define project deliverables: Identify the key deliverables of the project, including the data sources, models, algorithms, and other tools that will be used to predict COPD.

Develop a project timeline: Create a timeline for the project, including milestones and deadlines for each deliverable. Determine the resources needed to complete the project within the designated timeframe.

Establish project communication plan: Establish a communication plan to ensure that all stakeholders are informed about the progress of the project. This may include regular meetings, progress reports, and other communication channels.

Implement the project: Execute the project plan, following the timeline and milestones established. Monitor progress and adjust the plan as needed to ensure successful completion of the project.

Evaluate project outcomes: Evaluate the effectiveness of the project outcomes, including the accuracy and reliability of the predictions. Identify areas for improvement and opportunities for further research.

Document project results: Document the results of the project, including the data sources, models, and algorithms used, as well as the project timeline, milestones, and outcomes. Make this information available to stakeholders and other interested parties.

### 4.6 FINANCIAL REPORT ON ESTIMATED COSTING

- Hardware Costs: The hardware required for developing and training CNN models can be one of the biggest expenses. This can include high-end GPUs or specialized hardware such as TPUs, which can be expensive to purchase and maintain.

- Software Costs: There are several software tools and frameworks available for developing and training ML model. These software tools are generally free to use, but may require additional paid plugins or services for optimal performance.

- Data Collection and Preparation Costs: Collecting and preparing large datasets of authentic and manipulated images can be time-consuming and costly. This may involve hiring data annotators or purchasing datasets from third-party providers.

- Model Development Costs: Developing ML models for diagnosing COPD requires a team of skilled data scientists and machine learning engineers, which can be expensive. The cost may vary on the complexity and size of the models developed.

- Maintenance and Upgrades Costs: After developing the system, it is essential to maintain and upgrade it to keep up with the latest technology trends and developments. This may involve additional costs for hiring dedicated staff or outsourcing maintenance services.

The total estimated cost for diagnosing COPD using supervised machine learning can vary widely based on several factors, including the size and complexity of the dataset, the expertise and experience of the team involved, and the hardware and software requirements.

# CHAPTER 5
# IMPLEMENTATION DETAILS

## 5.1 DEVELOPMENT DEPLOYMENT AND SETUP

Generally, several phases are involved in the development stage such as research, data collection and preparation, model development and training, evaluation and validation, and deployment.

During the research phase, a thorough investigation of existing literature and techniques related to diagnosing COPD using supervised machine learning will be conducted by the project team. This will involve reviewing academic papers, online resources, and other relevant sources to gain a deep understanding of the topic and identify potential approaches.

In the data collection and preparation phase, the dataset that will be used to train and evaluate the model will be acquired and cleaned by the team. This may involve sourcing and labeling images that are both real and manipulated and then pre- processing them to ensure they are suitable for training the model.

In the model development and training phase, the architecture of the ML model will be designed by the team and trained using the prepared dataset. This may involve experimenting with different model architectures and hyperparameters to achieve optimal performance. In the evaluation and validation phase, the model will be tested on a separate dataset to evaluate its performance and ensure that it is not overfitting. This may involve measuring metrics such as accuracy, precision, recall, and F1 score.

Finally, in the deployment phase, the trained model will be integrated into a larger system or application for practical use by the team. The development stage for a project on diagnosing COPD using supervised machine learning can take several months or longer, depending on the resources available, the complexity of the project,and the expertise of the team involved.

### *5.1.1 Dataset Description*

The lung cancer dataset by StaceyInRobert, which is available on Kaggle, contains information related to COPD. The dataset consists of 59 records, with each record representing a single patients record. The dataset is relatively small, but it provides a starting point for exploring the use of machine learning algorithms for predicting lung cancer based on lung nodule features.

## 5.2 ALGORITHM

### *5.2.1. Logistic Regression*

Factual techniques for examining inserted data incorporate one from another free deciding outcomes. Results are guessed with two factors . The motivation behind the review, to find a decent model to clarify the connection which joins two helpful variables (contingent upon change = reaction or change reaction) and free level (indicator or clarification). Calculated lapse is an AI machine that rundowns the calculations used to uncover conceivable class-based changes. In calculated relapse, changes depend on two factors that incorporate data composed as 1 (indeed, achievement, and so forth) or 0 (no, disappointment, and so on)

### *5.2.2. Random Forest Classifier*

Common woodlands or normal timberlands are an approach to figuring out how to sort, retreat and different exercises, which is finished by building a great deal of choice trees during the preparation and delivering classes and strategies (traditional) or and that implies anticipating (withdrawing) the trees exclusively. Standard affirmation timberlands right for certificate past preparation. An average timberland is a kind of AI calculation that depends on outfit learning. Concentrating together is a sort of realizing where you enter various calculations or calculations commonly to have a solid feeling of prescience. The novel timberland calculation consolidates numerous calculations of a similar sort for example various choice trees, bringing about a tree backwoods, thus the name "Ordinary Forest". The backwoods calculation can be utilized for accumulation and sequencing assignments.

### *5.2.3. Decision Tree Classifier*

You would require a collection of patient data, including pertinent medical history, demographic data, and test results, to create a decision tree classifier for predicting COPD. It would be necessary to label the dataset, which would categorise each patient as either having COPD or not.After you have the dataset, you may create a decision tree classifier using a machine learning toolkit, such as scikit-learn in Python. The decision tree algorithm divides the data recursively into subsets according to the values of the input characteristics, maximizing the information gain or minimizing the entropy at each split.

#### *5.2.4. Naive Bayes*

One method for estimating the chance of developing COPD is the Naive Bayes algorithm . The naive Bayes method, which is based on the Bayes theorem, makes the assumption that the presence of one feature does not rely on the presence of other characteristics.We would first require a dataset comprising data on individuals with and without COPD in order to utilize Naive Bayes for predicting COPD. Age, gender, smoking status, family history of COPD, and the results of pulmonary function tests are common elements of this dataset.

### 5.3 TESTING

- Unit Testing: This step involves testing individual components of the software system, such as the data pre-processing module, feature selection module, machine learning algorithm selection module, model training module, model evaluation module, and user interface module

- Integration Testing: This step involves testing the integration of the individual components of the software system. Integration testing helps to ensure that the different modules of the software system work together as expected

- Performance Testing: This step involves testing the performance of the software system under different conditions, such as varying data sizes, network speeds, and user loads. Performance testing helps to identify any performanceissues and bottlenecks in the software system.

- Security Testing: This step involves testing the security of the software system to ensure that it complies with ethical and legal regulations related to patient data privacy and security.

- Usability Testing: This step involves testing the usability and user experience of the software system. Usability testing helps to ensure that the software system is user-friendly and easy to use.

# CHAPTER 6
# RESULTS AND DISCUSSION

Start by reporting the performance metrics of the machine learning model, including accuracy, precision, recall, and F1-score. These metrics provide an overall assessment of the model's ability to predict COPD and help to evaluate the tradeoff between false positives and false negatives.

Confusion matrix gives a detailed breakdown of the model's performance. The confusion matrix shows the number of true positives, false positives, true negatives, and false negatives, which can help to identify which types of errors the model is making and where it needs improvement.

The most important features for predicting COPD identified by the machine learning model. This can help to provide insights into the underlying risk factors for COPD and guide future research.

Limitations of the project, such as potential sources of bias or variability in the data, limitations of the machine learning algorithm or approach used, or generalizability of the results to other populations or settings.

Comparing the performance of the machine learning model to a baseline model, such as a simple rule-based algorithm or random guessing. This helps to provide context for the performance of the machine learning model and assess its added value compared to simpler approaches.

Discussing the potential clinical implications of the machine learning model for predicting COPD. This could include identifying high-risk individuals for targeted screening or interventions, improving early detection and diagnosis of COPD, or guiding treatment decisions.

# CHAPTER 7
# CONCLUSION

## 7.1 CONCLUSION

In conclusion, machine learning can be a valuable tool for predicting COPD and identifying high-risk individuals for targeted screening and interventions. The use of machine learning models can potentially improve early detection and diagnosis of COPD, guide treatment decisions, and improve patient outcomes.

However, the development and deployment of machine learning models for predicting COPD require careful consideration of data quality, model selection, and performance evaluation. It's also important to keep in mind that machine learning models are not a substitute for clinical judgment and expertise, and any predictions or decisions based on these models should be made in conjunction with healthcare providers.

Overall, the results of this study demonstrate the potential of machine learning for predicting COPD and highlight the need for further research and development in this area. By continuing to refine and improve machine learning models for predicting COPD, we can potentially make significant strides in improving the diagnosis and management of this debilitating disease.

## 7.2 FUTURE WORKS

The current study may have identified some important features for predicting COPD, there may be other features or risk factors that have not been explored yet. Future research can focus on identifying and validating new features that can improve the accuracy and generalizability of machine learning models for predicting COPD.

One important area of research is Machine learning models can be enhanced by incorporating multi-omics data, such as genomics, transcriptomics, and proteomics. Integrating these data types can provide a more comprehensive view of the biological processes underlying COPD and improve the accuracy of machine learning models.

In addition, Machine learning models can be complex and difficult to interpret, making it challenging for clinicians to understand how the model arrived at its predictions. Future work can focus on developing methods to improve the explainability and interpretability of machine learning models, allowing for better integration into clinical decision-making.

Finally, future work can focus on implementing machine learning models for predicting COPD in clinical practice. This would require addressing several challenges, such as integrating the models into electronic health record systems, ensuring data privacy and security, and assessing the impact of the models on clinical outcomes and patientcare.

Overall, there are many exciting areas of research that could help to advance the field of detecting COPD using supervised machine learning, helps to ensure that these technologies are effective, reliable, and trustworthy in real-world applications.

## REFERENCES:

[1] Sharma, Anushka, et al. "Artificial Intelligence in Oncology- Technologies being used and scope in India." UNIVERSITY JOURNAL MAXILLOFACIAL SURGERY AND ORAL SCIENCES 1.2 (2021).

[2] Jiang, Feng. Sputum Biomarkers to Improve CT Screening for the Early Detection of Lung Cancer in Veterans. University of Maryland, Baltimore Baltimore United States, 2020.

[3] Batra, Usha, NiharRanjan Roy, and Brajendra Panda, eds. Data Science and Analytics: 5th International Conference on Recent Developments in Science, Engineering and Technology, REDSET 2019, Gurugram, India, November 15-16, 2019, Revised Selected Papers, Part II. Vol. 1230. Springer Nature, 2020.

[4] Chandra, E. YatishVenkata, K. Ravi Teja, and M. Hari Chandra Siva Prasad. "Lung Cancer Prediction using Data Mining Techniques." International Journal of Recent Technology and Engineering 8 (2019).

[5] R. Moshavegh, K. L. Hansen, H. Moller-Sorensen, M. B. Nielsen, and J. A. Jensen, "Automatic Detection of B-Lines in In Vivo Lung Ultrasound," IEEE Trans Ultrason Ferroelectr Freq Control, vol. 66, no. 2, pp. 309-317,Feb 2019.

[6] Niranjan, S. K. "Second International Conference on Green Computing and Internet of Things (ICGCIoT 2018).

[7] M. H. Miglioranza et al. "Pulmonary congestionevaluated by lung ultrasound predicts decompensation in heart failure outpatients," Int JCardiol, vol. 240, pp. 271-278, Aug 1 2017.

[8] ML Demi, W. van Hoeve, R. J. G. van Sloun, G.Soldati, and M. Demi, "Determination of a potential quantitative measure of the state of the lung using lang ultrasound spectroscopy," Sci Rep, vol. 7, no. 1, p. 12746, Oct. 6 2017.

[9] N. Anantrasirichai, W. Hayes, M. Allinovi, D. Bull, and A. Achim, "Line Detection as an Inverse Problem: Application to Lung Ultrasound Imaging." IEEE Trans Med Imaging, vol. 36, no. 10, pp. 2045-2056, Oct 2017.

[10] X. Zhang et al., "Lung ultrasound surface wave elastography: a pilot clinical study," IEEE transactions on ultrasonics, ferroelectrics, and frequency control, vol. 64, no. 9,

pp. 1298-1304,2017.

[II] M. Jozwiak, J.-L. Teboul, and X. Monnet, "Extravascular lung water in critical care: recent advances and clinical applications." Annals of intensive care, vol. 5, no. 1. p. 38, 2015.

[12] C. M. WRITING et al., "2013 ACCF/AHA guideline for the management of heart failure: a report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines," Circulation, vol. 128, no. 16, p e240, 2013.

[13] D. Lichtenstein and G. Meziere. "A lung ultrasound sign allowing bedside distinction between pulmonary edema and COPD: the comet-tail artifact," Intensive care medicine, vol. 24, no. 12, pp. 1331-1334, 2011.

# APPENDIX

## A. SOURCE CODE

### Module – 1

**Pre-Processing**

```
#import library packages

import pandas as pd

import numpy as np

import warnings

warnings.filterwarnings("ignore")

#Load given dataset

data = pd.read_csv("Lung Cancer.csv")
```

Before drop the given dataset

```
data.head()

#shape

data.shape
```

After drop the given dataset

```
df = data.dropna()

df.head()

df.shape

#show columns

df.columns

#To describe the dataframe

df.describe()

#Checking datatype and information about dataset
```

```
df.info()

#Checking sum of missing values

df.isnull().sum()

df.GENDER.unique()

df.AGE.unique()

df.YELLOW_FINGERS.unique()

df.CHRONIC_DISEASE.unique()

df.ALCOHOL_CONSUMING.unique()

df.COUGHING.unique()

df.SHORTNESS_OF_BREATH.unique()

df.CHEST_PAIN.unique()

print("Minimum value of Age of patient is:", df.AGE.min())

print("Maximum value of Age of patient is:", df.AGE.max())

print("Age of patient ranges :", sorted(df['AGE'].unique()))

df.corr()
```

Before Pre-Processing

```
df.head()
```

After Pre-Processing

```
from sklearn.preprocessing import LabelEncoder

var_mod = ['GENDER','LUNG_CANCER']

le = LabelEncoder()
```

```
for i in var_mod:

    df[i] = le.fit_transform(df[i]).astype(int)

df.head()
```

## Module – 2

## Visualization

```
#import library packages

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

data = pd.read_csv("Lung Cancer.csv")

df=data.dropna()

df.columns

pd.crosstab(df.GENDER,df.LUNG_CANCER)

pd.crosstab(df.CHRONIC_DISEASE,df.LUNG_CANCER)

df["AGE"].hist(figsize=(10,8), color="red", alpha=0.8)

plt.title("AGE DISTRIBUTION")

plt.xlabel("AGE")
plt.ylabel("COUNT")

plt.show()

#Propagation by variable

def PropByVar(df, variable):

    dataframe_pie = df[variable].value_counts()

    ax = dataframe_pie.plot.pie(figsize=(10,10), autopct='%1.2f%%', fontsize = 12)

    ax.set_title(variable + ' \n', fontsize = 15)

    return np.round(dataframe_pie/df.shape[0]*100,2)

PropByVar(df, 'AGE')

fig, ax = plt.subplots(figsize=(10,8))

sns.barplot(df.LUNG_CANCER, df.CHRONIC_DISEASE, ax=ax)

plt.show()

fig, ax = plt.subplots(figsize=(15,6))

sns.boxplot(df.AGE, ax =ax)

plt.title("Age distribution")

plt.show()

fig, ax = plt.subplots(figsize=(10,6))

sns.stripplot(df.LUNG_CANCER, df.AGE)

plt.title("Lung Cancer predicted for a age")

plt.show()
```

```
# Heatmap plot diagram

fig, ax = plt.subplots(figsize=(15,10))

sns.heatmap(df.corr(), ax=ax,  annot=True)

from sklearn.preprocessing import LabelEncoder

var_mod = ['GENDER','LUNG_CANCER']

le = LabelEncoder()

for i in var_mod:

    df[i] = le.fit_transform(df[i]).astype(int)

df.head()
```

## Module – 3

### Logistic Regression Algorithm

```
#import library packages

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import warnings

warnings.filterwarnings('ignore')

#Load given dataset

data = pd.read_csv("Lung.csv")
```

```
df=data.dropna()

df.columns

#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, roc_auc_score

from sklearn.preprocessing import LabelEncoder

var_mod = ['GENDER','LUNG_CANCER']

le = LabelEncoder()

for i in var_mod:

    df[i] = le.fit_transform(df[i]).astype(int)

X = df.drop(labels='LUNG_CANCER', axis=1)

#Response variable

y = df.loc[:,'LUNG_CANCER']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
```

Logistic Regression :

```
from sklearn.metrics import accuracy_score, confusion_matrix

from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score

logR= LogisticRegression()
logR.fit(X_train,y_train)
predictR = logR.predict(X_test)

print("")
print('Classification report of Logistic Regression Results:')
print("")
print(classification_report(y_test,predictR))

print("")
cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Logistic Regression is:\n',cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")



accuracy = cross_val_score(logR, X, y, scoring='accuracy')

print('Cross validation test results of accuracy:')

print(accuracy)

#get the mean of each fold

print("")

print("Accuracy result of Logistic Regression is:",accuracy.mean() * 100)

LR=accuracy.mean() * 100

def graph():

    import matplotlib.pyplot as plt

    data=[LR]

    alg="Logistic Regression"

    plt.figure(figsize=(5,5))

    b=plt.bar(alg,data,color=("b"))

    plt.title("Accuracy comparison of Lung disease",fontsize=15)

    plt.legend(b,data,fontsize=9)

graph()

TP = cm[0][0]

FP = cm[1][0]
```

```
FN = cm[1][1]
TN = cm[0][1]
print("True Positive :",TP)
print("True Negative :",TN)
print("False Positive :",FP)
print("False Negative :",FN)
print("")
TPR  =  TP/(TP+FN)
TNR  =  TN/(TN+FP)
FPR  =  FP/(FP+TN)
FNR = FN/(TP+FN)
print("True Positive Rate :",TPR)
print("True Negative Rate :",TNR)
print("False Positive Rate :",FPR)
print("False Negative Rate :",FNR)
print("")

PPV  =  TP/(TP+FP)
NPV = TN/(TN+FN)
print("Positive Predictive Value :",PPV)
print("Negative predictive value :",NPV)

def plot_confusion_matrix(cm2, title='Confusion matrix-LogisticRegression', cmap=plt.cm.Blues):

    target_names=['Predict','Actual']

    plt.imshow(cm2, interpolation='nearest', cmap=cmap)

    plt.title(title)

    plt.colorbar()

    tick_marks = np.arange(len(target_names))

    plt.xticks(tick_marks, target_names, rotation=45)

    plt.yticks(tick_marks, target_names)

    plt.tight_layout()

    plt.ylabel('True label')

    plt.xlabel('Predicted label')



cm2=confusion_matrix(y_test, predictR)

print('Confusion matrix-LogisticRegression:')

print(cm2)

plot_confusion_matrix(cm2)
```

**Module – 4**

**Random Forest Algorithm**

```
#import library packages
```

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import warnings

warnings.filterwarnings('ignore')

#Load given dataset

data = pd.read_csv("Lung.csv")

df=data.dropna()

df.columns

#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, roc_auc_score

from sklearn.preprocessing import LabelEncoder

var_mod = ['GENDER','LUNG_CANCER']

le = LabelEncoder()

for i in var_mod:

    df[i] = le.fit_transform(df[i]).astype(int)

X = df.drop(labels='LUNG_CANCER', axis=1)

#Response variable
```

```
y = df.loc[:,'LUNG_CANCER']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
```

RandomForestClassifier:

```
from sklearn.metrics import accuracy_score, confusion_matrix

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import cross_val_score



rfc = RandomForestClassifier()

rfc.fit(X_train,y_train)

predictR = rfc.predict(X_test)



print("")

print('Classification report of Random Forest Classifier Results:')

print("")

print(classification_report(y_test,predictR))



print("")
```

```
cm=confusion_matrix(y_test,predictR)

print('Confusion Matrix result of Random Forest Classifier is:\n',cm)

print("")

sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])

print('Sensitivity : ', sensitivity )

print("")

specificity = cm[1,1]/(cm[1,0]+cm[1,1])

print('Specificity : ', specificity)

print("")



accuracy = cross_val_score(rfc, X, y, scoring='accuracy')

print('Cross validation test results of accuracy:')

print(accuracy)

#get the mean of each fold

print("")

print("Accuracy result of Random Forest Classifier is:",accuracy.mean() * 100)

LR=accuracy.mean() * 100

def graph():

    import matplotlib.pyplot as plt

    data=[LR]
    alg="Random Fores tClassifier"

    plt.figure(figsize=(5,5))

    b=plt.bar(alg,data,color=("b"))

    plt.title("Accuracy comparison of Lung disease",fontsize=15)

    plt.legend(b,data,fontsize=9)

graph()

TP = cm[0][0]

FP = cm[1][0]

FN = cm[1][1]

TN = cm[0][1]

print("True Positive :",TP)

print("True Negative :",TN)

print("False Positive :",FP)

print("False Negative :",FN)

print("")

TPR  =  TP/(TP+FN)

TNR  =  TN/(TN+FP)

FPR  =  FP/(FP+TN)

FNR = FN/(TP+FN)

print("True Positive Rate :",TPR)
print("True Negative Rate :",TNR)

print("False Positive Rate :",FPR)

print("False Negative Rate :",FNR)

print("")

PPV = TP/(TP+FP)

NPV = TN/(TN+FN)

print("Positive Predictive Value :",PPV)

print("Negative predictive value :",NPV)

def plot_confusion_matrix(cm2, title='Confusion matrix-RandomForestClassifier',
cmap=plt.cm.Blues):

    target_names=['Predict','Actual']

    plt.imshow(cm2, interpolation='nearest', cmap=cmap)

    plt.title(title)

    plt.colorbar()

    tick_marks = np.arange(len(target_names))

    plt.xticks(tick_marks, target_names, rotation=45)

    plt.yticks(tick_marks, target_names)

    plt.tight_layout()

    plt.ylabel('True label')

    plt.xlabel('Predicted label')
```

```
cm2=confusion_matrix(y_test, predictR)

print('Confusion matrix-RandomForestClassifier:')

print(cm2)

plot_confusion_matrix(cm2)

import joblib

joblib.dump(rfc,"model.pkl")

['model.pkl']
```

## Module – 5

### Decision Tree Algorithm

```
#import library packages

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import warnings

warnings.filterwarnings('ignore')

#Load given dataset

data = pd.read_csv("Lung.csv")

df=data.dropna()

df.columns
```

```
#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.

from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score, roc_auc_score

from sklearn.preprocessing import LabelEncoder

var_mod = ['GENDER','LUNG_CANCER']

le = LabelEncoder()

for i in var_mod:

    df[i] = le.fit_transform(df[i]).astype(int)

X = df.drop(labels='LUNG_CANCER', axis=1)

#Response variable

y = df.loc[:,'LUNG_CANCER']

#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1,
stratify=y)
```

Decision Tree Classifier:

```
from sklearn.metrics import accuracy_score, confusion_matrix

from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import cross_val_score
```

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train,y_train)
predictR = dtc.predict(X_test)

print("")
print('Classification report of Decision Tree Classifier Results:')
print("")
print(classification_report(y_test,predictR))

print("")
cm=confusion_matrix(y_test,predictR)
print('Confusion Matrix result of Decision Tree Classifier is:\n',cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")
```

```
accuracy = cross_val_score(dtc, X, y, scoring='accuracy')

print('Cross validation test results of accuracy:')

print(accuracy)

#get the mean of each fold

print("")

print("Accuracy result of Decision Tree Classifier is:",accuracy.mean() * 100)

LR=accuracy.mean() * 100

def graph():

    import matplotlib.pyplot as plt

    data=[LR]

    alg="Decision Tree Classifier "

    plt.figure(figsize=(5,5))

    b=plt.bar(alg,data,color=("b"))

    plt.title("Accuracy comparison of Lung disease",fontsize=15)

    plt.legend(b,data,fontsize=9)

graph()

TP = cm[0][0]

FP = cm[1][0]

FN = cm[1][1]

TN = cm[0][1]
```

```
print("True Positive :",TP)

print("True Negative :",TN)

print("False Positive :",FP)

print("False Negative :",FN)

print("")

TPR  =  TP/(TP+FN)

TNR  =  TN/(TN+FP)

FPR  =  FP/(FP+TN)

FNR = FN/(TP+FN)

print("True Positive Rate :",TPR)

print("True Negative Rate :",TNR)

print("False Positive Rate :",FPR)

print("False Negative Rate :",FNR)

print("")

PPV  =  TP/(TP+FP)

NPV = TN/(TN+FN)

print("Positive Predictive Value :",PPV)

print("Negative predictive value :",NPV)

def plot_confusion_matrix(cm2, title='Confusion matrix-DecisionTreeClassifier', cmap=plt.cm.Blues):

    target_names=['Predict','Actual']
    plt.imshow(cm2, interpolation='nearest', cmap=cmap)

    plt.title(title)

    plt.colorbar()

    tick_marks = np.arange(len(target_names))

    plt.xticks(tick_marks, target_names, rotation=45)

    plt.yticks(tick_marks, target_names)

    plt.tight_layout()

    plt.ylabel('True label')

    plt.xlabel('Predicted label')


cm2=confusion_matrix(y_test, predictR)

print('Confusion matrix-DecisionTreeClassifier:')

print(cm2)

plot_confusion_matrix(cm2)
```

**Module – 6**

**Naïve Bayes Algorithm**

```
#import library packages

import pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import warnings

warnings.filterwarnings('ignore')

#Load given dataset

data = pd.read_csv("Lung.csv")

df=data.dropna()

df.columns

#According to the cross-validated MCC scores, the random forest is the best-performing model, so now let's evaluate its performance on the test set.

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, roc_auc_score

from sklearn.preprocessing import LabelEncoder

var_mod = ['GENDER','LUNG_CANCER']

le = LabelEncoder()

for i in var_mod:

    df[i] = le.fit_transform(df[i]).astype(int)

X = df.drop(labels='LUNG_CANCER', axis=1)

#Response variable

y = df.loc[:,'LUNG_CANCER']
```

```
#We'll use a test size of 30%. We also stratify the split on the response variable, which is very important to do because there are so few fraudulent transactions.

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=1, stratify=y)
```

Naive Bayes:

```
from sklearn.metrics import accuracy_score, confusion_matrix

from sklearn.naive_bayes import GaussianNB

from sklearn.model_selection import cross_val_score



nb = GaussianNB()

nb.fit(X_train,y_train)

predictR = nb.predict(X_test)



print("")

print('Classification report of Naive Bayes Results:')

print("")

print(classification_report(y_test,predictR))



print("")

cm=confusion_matrix(y_test,predictR)
```

```
print('Confusion Matrix result of Naive Bayes is:\n',cm)
print("")
sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
print('Sensitivity : ', sensitivity )
print("")
specificity = cm[1,1]/(cm[1,0]+cm[1,1])
print('Specificity : ', specificity)
print("")

accuracy = cross_val_score(nb, X, y, scoring='accuracy')
print('Cross validation test results of accuracy:')
print(accuracy)
#get the mean of each fold
print("")
print("Accuracy result of Naive Bayes is:",accuracy.mean() * 100)
LR=accuracy.mean() * 100
def graph():
    import matplotlib.pyplot as plt
    data=[LR]
    alg="GaussianNB"
    plt.figure(figsize=(5,5))

    b=plt.bar(alg,data,color=("b"))

    plt.title("Accuracy comparison of Lung disease",fontsize=15)

    plt.legend(b,data,fontsize=9)

graph()

TP = cm[0][0]

FP = cm[1][0]

FN = cm[1][1]

TN = cm[0][1]

print("True Positive :",TP)

print("True Negative :",TN)

print("False Positive :",FP)

print("False Negative :",FN)

print("")

TPR  =  TP/(TP+FN)

TNR  =  TN/(TN+FP)

FPR  =  FP/(FP+TN)

FNR = FN/(TP+FN)

print("True Positive Rate :",TPR)

print("True Negative Rate :",TNR)
print("False Positive Rate :",FPR)

print("False Negative Rate :",FNR)

print("")

PPV = TP/(TP+FP)

NPV = TN/(TN+FN)

print("Positive Predictive Value :",PPV)

print("Negative predictive value :",NPV)

def plot_confusion_matrix(cm2, title='Confusion matrix-Naive Bayes',
cmap=plt.cm.Blues):

    target_names=['Predict','Actual']

    plt.imshow(cm2, interpolation='nearest', cmap=cmap)

    plt.title(title)

    plt.colorbar()

    tick_marks = np.arange(len(target_names))

    plt.xticks(tick_marks, target_names, rotation=45)

    plt.yticks(tick_marks, target_names)

    plt.tight_layout()

    plt.ylabel('True label')

    plt.xlabel('Predicted label')



cm2=confusion_matrix(y_test, predictR)
```

```
print('Confusion matrix-Naive Bayes:')

print(cm2)

plot_confusion_matrix(cm2)
```

**HTML Code:**

```
<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

  <meta charset="UTF-8">

  <title>TITLE</title>

<link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">

  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/css'>

<style>

.back{
 background-image: url("{{ url_for('static', filename='image/img.png') }}");

 background-repeat: no-repeat;

 background-attachment: fixed;

 background-size: 100% 100%;

}

.white{

color:white;

}

.space{

margin:10px 30px;

padding:10px 10px;

background: palegreen;

width:500px

}

.gap{

padding:10px 20px;

}

</style>



</head>
```

```
<body class="back">

 <div>

          <div class="jumbotron">

          <h1 style="text-align:center">DIOGNOSIS OF LUNG DISEASE</h1>



          </div>

    <!-- Main Input For Receiving Query to our ML -->

   <form class="form-group" action="{{ url_for('predict')}}"method="post">



                    <div class="row">

                    <div class="gap col-md-6 ">



                    <label class="white" for="">GENDER</label>

                    <select class="space form-control" name="GENDER" id="GENDER">

                              <option value=0>FEMALE</option>

                              <option value=1>MALE</option>

                    </select>



                    <label class="white"  for="">AGE</label>
                    <input type="number" class="space form-control" step="0.01" name="AGE" placeholder="AGE" required="required" /><br>


                    <label class="white"  for="">SMOKING</label>

                    <input type="number" class="space form-control" step="0.01" name="SMOKING" placeholder="SMOKING" required="required" /><br>


                    <label class="white" for="">YELLOW FINGERS</label>

                    <input type="number" class="space form-control" step="0.01" name="YELLOW FINGERS" placeholder="YELLOW FINGERS" required="required" /><br>


                    <label class="white" for="">ANXIETY</label>

                    <input type="number" class="space form-control" step="0.01" name="ANXIETY" placeholder="ANXIETY" required="required" /><br>


                    <label class="white"  for="">PEER PRESSURE</label>

                    <input type="number" class="space form-control" step="0.01" name="PEER PRESSURE" placeholder="PEER PRESSURE" required="required" /><br>

                     <label class="white"  for="">CHRONIC DISEASE</label>

                    <input type="number" class="space form-control" step="0.01" name="CHRONIC DISEASE" placeholder="CHRONIC DISEASE" required="required" /><br>
```

```
                <label class="white"  for="">FATIGUE</label>

                <input type="number" class="space form-control" step="0.01"
name="FATIGUE" placeholder="FATIGUE" required="required" /><br>

        </div>

<div class="gap col-md-6">



                <label class="white"  for="">ALLERGY</label>

                <input type="number" class="space form-control" step="0.01"
name="ALLERGY" placeholder="ALLERGY" required="required" /><br>



                <label class="white"  for="">WHEEZING</label>

                <input type="number" class="space form-control" step="0.01"
name="WHEEZING" placeholder="WHEEZING" required="required" /><br>



                <label class="white"  for="">ALCOHOL CONSUMING</label>

                <input type="number" class="space form-control" step="0.01"
name="ALCOHOL CONSUMING" placeholder="ALCOHOL CONSUMING"
required="required" /><br>

                <label class="white"  for="">COUGHING</label>

                <input type="number" class="space form-control" step="0.01"
name="COUGHING" placeholder="COUGHING" required="required" /><br>
```

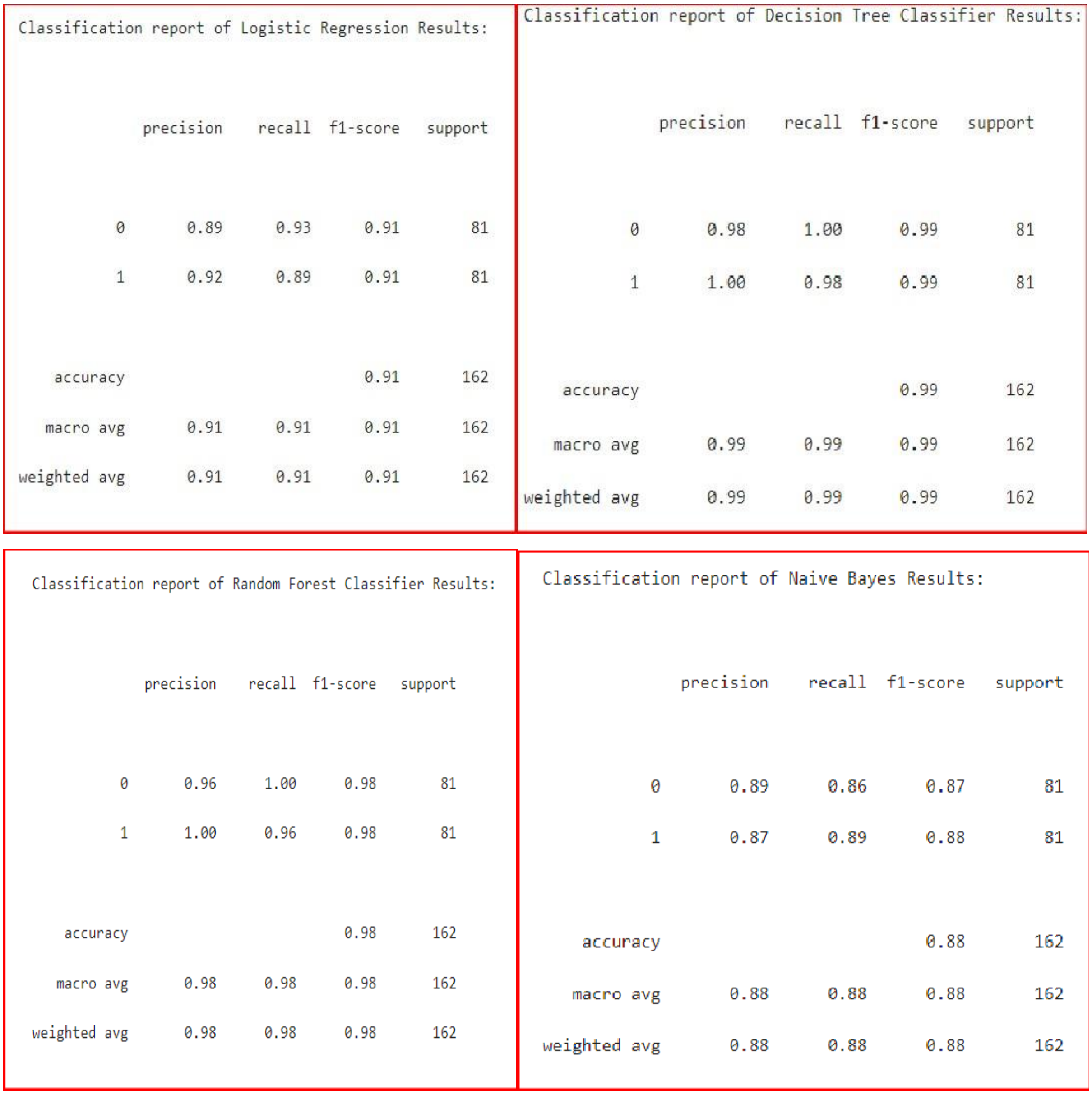
```
                    <label class="white"  for="">SHORTNESS OF BREATH</label>

                    <input type="number" class="space form-control" step="0.01"
name="SHORTNESS OF BREATH" placeholder="SHORTNESS OF BREATH"
required="required" /><br>



                    <label class="white"  for="">SWALLOWING DIFFICULTY</label>

                    <input type="number" class="space form-control" step="0.01"
name=SWALLOWING DIFFICULTY" placeholder="SWALLOWING DIFFICULTY"
required="required" /><br>

                    <label class="white"  for="">CHEST PAIN</label>

                    <input type="number" class="space form-control" step="0.01"
name="CHEST PAIN" placeholder="CHEST PAIN" required="required" /><br>



</div>

</div>



<div style="padding:2% 35%">

  <button type="submit" class="btn btn-success btn-block"
style="width:350px;padding:20px">Predict</button>

</div>



    </form>
```

```
   <br>

   <br>

<div  style="background:skyblue;padding:2% 40%">

   {{ prediction_text }}

</div>

 </div>

</body>

</html>
```

**Flask Deploy:**

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import joblib

app = Flask(__name__)
model = joblib.load('model.pkl')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    print(final_features)
    prediction = model.predict(final_features)
    print(prediction)
    output = prediction[0]
    if output == 1:
        return render_template('index.html', prediction_text='LUNG AFFECTED')
    else:
        return render_template('index.html', prediction_text='LUNG NOT AFFECTED')
    print(output)

if__name__== "__main__":
    app.run(host="localhost", port=8000)
```

## B. SCREENSHOTS



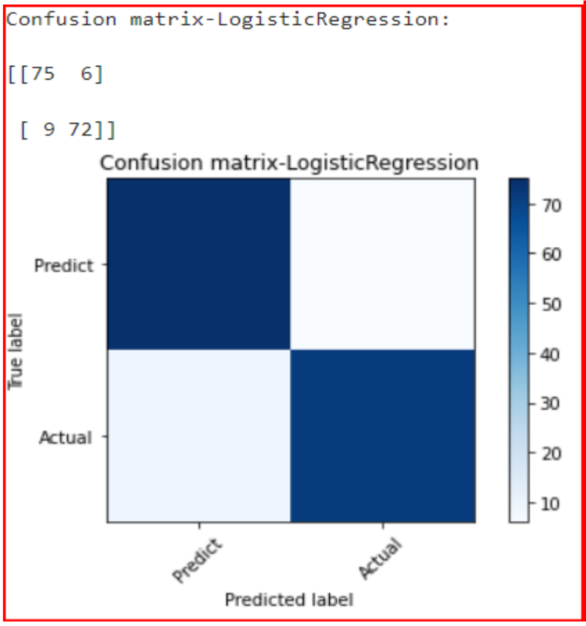
```
Classification report of Logistic Regression Results:

              precision    recall  f1-score   support

           0       0.89      0.93      0.91        81
           1       0.92      0.89      0.91        81

    accuracy                           0.91       162
   macro avg       0.91      0.91      0.91       162
weighted avg       0.91      0.91      0.91       162
```

```
Classification report of Decision Tree Classifier Results:

              precision    recall  f1-score   support

           0       0.98      1.00      0.99        81
           1       1.00      0.98      0.99        81

    accuracy                           0.99       162
   macro avg       0.99      0.99      0.99       162
weighted avg       0.99      0.99      0.99       162
```

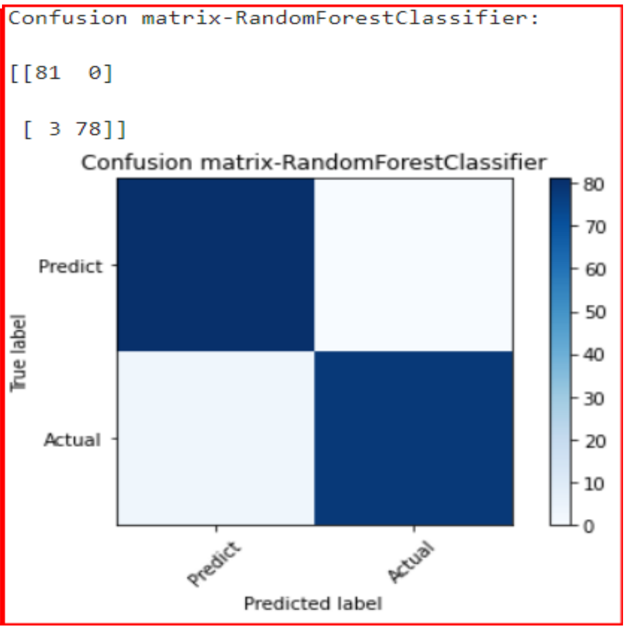
```
Classification report of Random Forest Classifier Results:

              precision    recall  f1-score   support

           0       0.96      1.00      0.98        81
           1       1.00      0.96      0.98        81

    accuracy                           0.98       162
   macro avg       0.98      0.98      0.98       162
weighted avg       0.98      0.98      0.98       162
```

```
Classification report of Naive Bayes Results:

              precision    recall  f1-score   support

           0       0.89      0.86      0.87        81
           1       0.87      0.89      0.88        81

    accuracy                           0.88       162
   macro avg       0.88      0.88      0.88       162
weighted avg       0.88      0.88      0.88       162
```

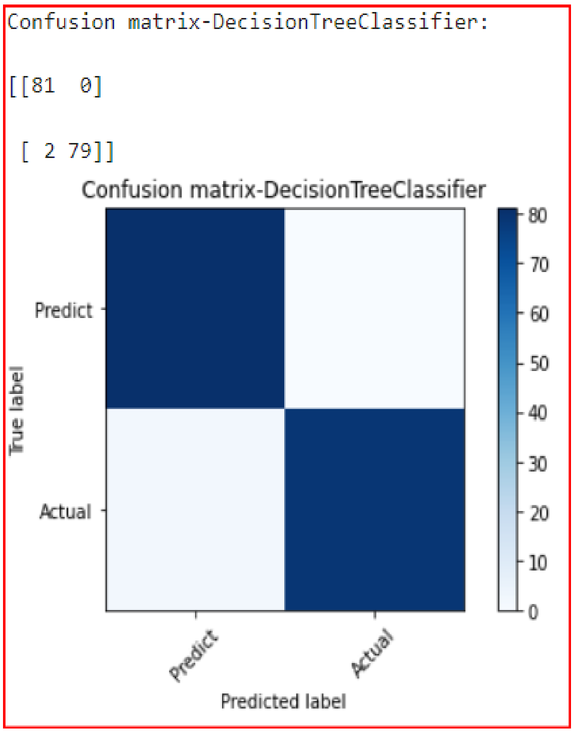
***Fig B.1 Classification report***



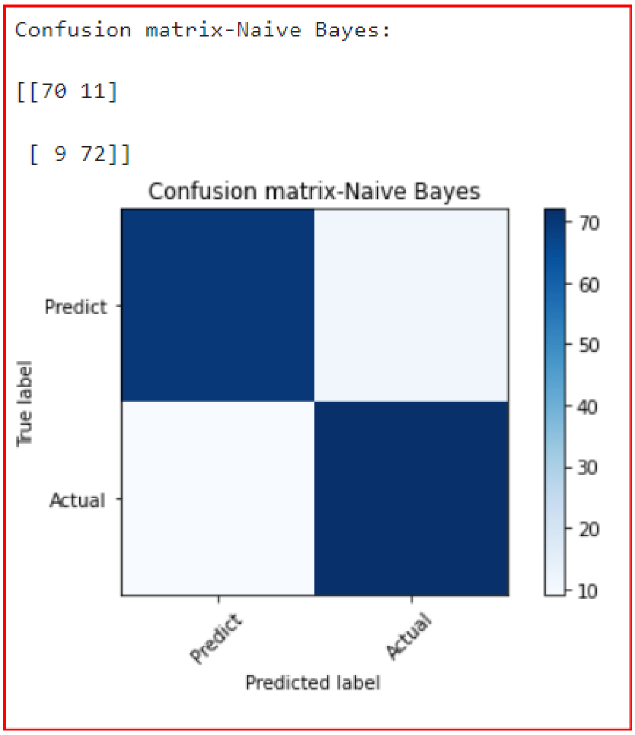
```
Confusion matrix-LogisticRegression:

[[75  6]

 [ 9 72]]
```



```
Confusion matrix-RandomForestClassifier:

[[81  0]

 [ 3 78]]
```



```
Confusion matrix-DecisionTreeClassifier:

[[81  0]

 [ 2 79]]
```



```
Confusion matrix-Naive Bayes:

[[70 11]

 [ 9 72]]
```

*Fig : B.2.Confussion matrix*

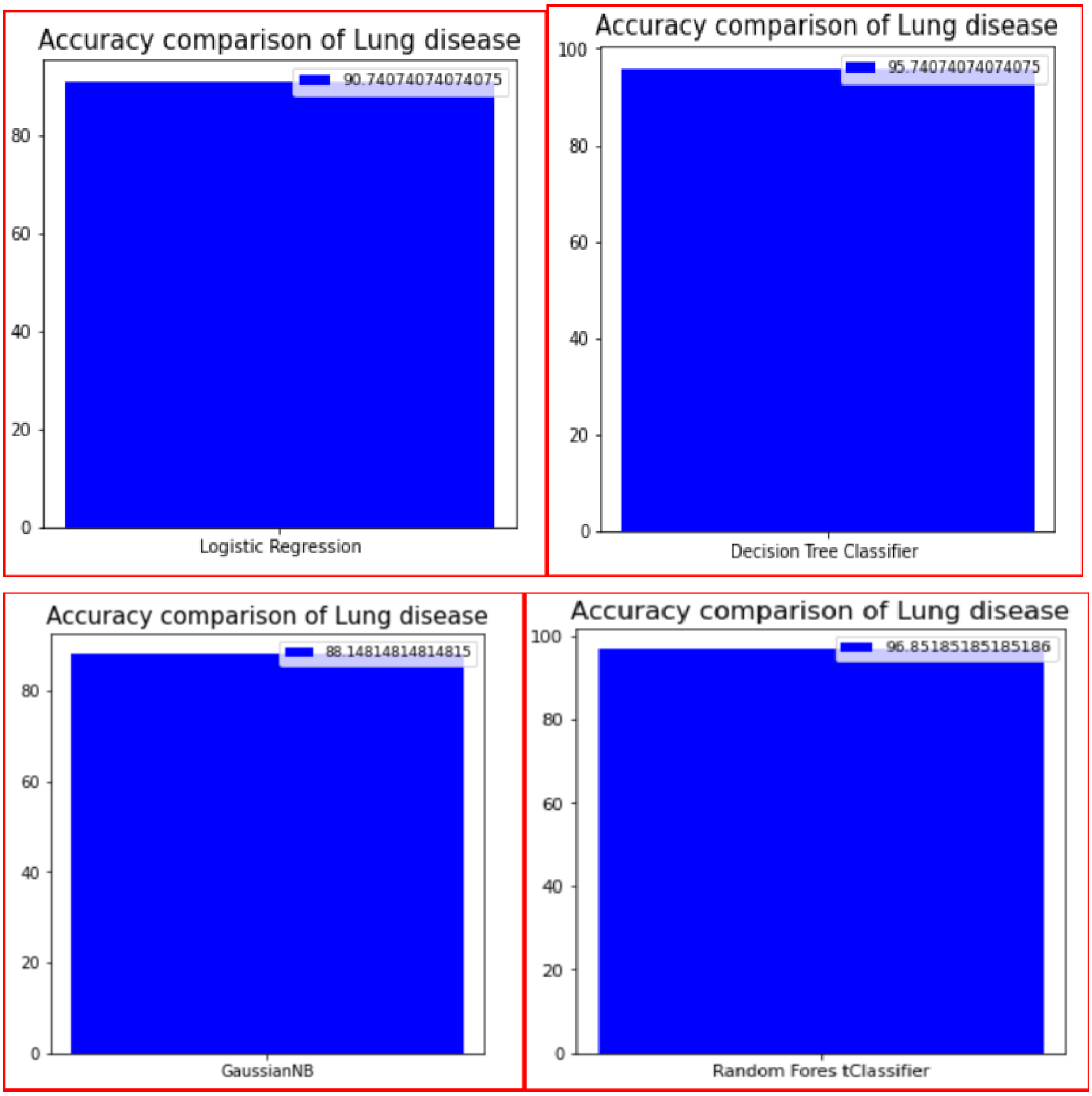


***Fig B.3 Accuracy***

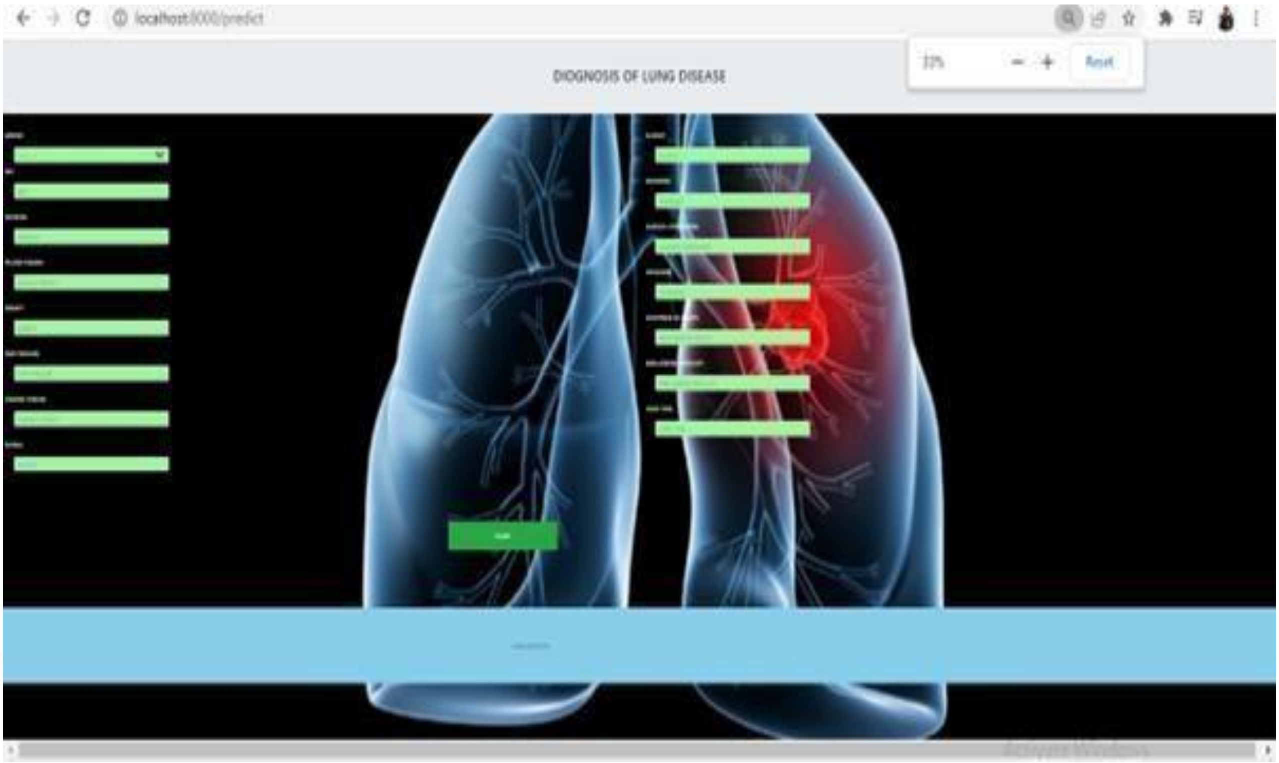


*Fig :B.4 User interface*

## C. RESEARCH PAPER

# DIAGNOSING COPD USING SUPERVISED MACHINE LEARNING

Manoj.M
Mail.id: manojhariharank7@gmail.com
Sathyabama Institute of Science and Technology,
Chennai,India

Mohammed Afrid.S
Mail.id: s.mohdafrid@gmail.com
Sathyabama Institute of Science and Technology,
Chennai,India

Dr.S.Rajashree
Mail.id: rajashree.cse@sathyabama.ac.in
Sathyabama Institute of Science and Technology,
Chennai,India

***Abstract*—Our study delves into the realm of clinical decision support systems within the healthcare industry with a specific focus on respiratory diseases such as Asthma and chronic obstructive pulmonary disease (COPD). The prevention, diagnosis, and treatment of these ailments are of utmost importance, particularly in regard to preventing exacerbation and determining the severity of the disease during hospitalization. The need for such measures is a global initiative, especially for COPD patients and is only available during the stable-phase of the disease. This is where AI systems come into play, as traditional methods take too long for accurate prognosis. Machine-learning techniques have been proven effective in crucial healthcare applications and we have applied four supervised machine-learning algorithms - Random Forest Classifier, Naive Bayes, Decision Tree classifier and Logistic Regression - to aid respiratory physicians in estimating the severity of COPD patients in the early stages, thus guiding the cure strategy. It is crucial to detect and manage COPD in its early stages to greatly improve the quality of life for patients and reduce the burden on healthcare systems. By utilizing advanced machine-learning techniques, we can provide better care for those suffering from respiratory diseases and ultimately improve the overall health of the population.**

**Keywords: Chronic Obstructive Pulmonary Disease(COPD);Supervised Machine learning; Random Forest Classifier, Naive Bayes, Decision Tree Classifier and Logistic Regression**

## 1. INTRODUCTION

Chronic Obstructive Pulmonary Disease (COPD), a respiratory condition, can cause considerable morbidity and death, if not identified and treated right once. Early detection of Chronic Obstructive Pulmonary Disease is crucial in avoiding the illness from advancing to a more severe state. Today, spirometry is the gold standard diagnostic test for Chronic Obstructive Pulmonary Disease. Spirometry is difficult to use in environments with limited resources because it needs specialized equipment and trained personnel. Hence, the creation of precise, non-invasive, and affordable diagnostic techniques is crucial.

For the diagnosis of Chronic Obstructive Pulmonary Disease, supervised machine learning algorithms have shown promise. Large data sets may be analyzed by machine learning algorithms to find patterns and generate precise predictions. The application of supervised machine learning algorithms for the diagnosis of COPD will be reviewed .

Persistent lung illness is the quickest developing reason for death for individuals beyond 65 a year old, a huge expansion in the United States populace. Fostering a treatment program for COPD requires a great deal of reasoning on various issues, including diagnosing COPD, depending on physiological tests, testing results, and infection. COPD (ongoing lung sickness) is a long illness that is challenging to inhale and increments over the long haul. Regularly, the respiratory plot and the sacs in your lungs are simple or nitty gritty. During inward breath, the respiratory plot conveys air into the sacs. The air pack is loaded up with air like a little air pocket. Whenever you inhale, the air pack goes out and the air get away.

In the event that you have COPD, your wind current is diminished because of at least one issues. Lessening the seriousness of respiratory diseases and pneumonia. The divider between the numerous airbags imploded. The dividers of the street are thick and thick. The aviation routes produce a larger number of breaks than ordinary and are interfered. Emphysema influences the air sacs in the lungs and the dividers between them. They are harmed and won't change. Ongoing bronchitis is a sickness where the respiratory lot is continually consuming and consuming. She would balloon and conceive an offspring. The reason for COPD is the harm to your lungs and your breath and the drawn-out openness to it. Tobacco smoke is a significant reason in the United States. Smoke, tobacco, and different kinds of tobacco smoke can cause COPD, particularly when breathed in. Openness to other respiratory variables can add to COPD.

## 2. LITERATURE SURVEY

[1] F. A. Marcano-Belisario et al 's "Predictive models for exacerbation risk in chronic obstructive pulmonary disease using electronic health record data: A systematic review," was published in 2020. This review of the literature offers a systematic analysis of studies that employed supervised machine learning to forecast the probability of an exacerbation in COPD patients using data from electronic health records. The writers go over the benefits and drawbacks of various strategies.

[2] Tandel et al. (2020) published a study titled "Prediction of COPD using machine learning algorithms using spirometry data." This study used spirometry data to examine the effectiveness of several machine learning algorithms for the prediction of COPD. The decision tree method, which had an accuracy of 94.4%, was determined to perform the best by the authors.

[3] Sánchez-Morillo et al. (2019) published "A machine learning technique for COPD identification using spirometry and demographic data." This study created a machine learning method that uses spirometry and demographic information to identify COPD. For detecting COPD, the algorithm had an accuracy of 87.6%.

[4] Shen and others in 2018 published a study titled "Prediction of COPD hospitalisation using machine learning and administrative data from electronic health records." This study used administrative data from electronic health records to build a machine learning model to forecast COPD hospitalisation. The authors' forecast of hospitalisation had an accuracy of 84.6%.

[5] Rahimi et al. (2017) published "COPD exacerbation prediction using machine learning technique" - This study used demographic, clinical, and physiological data to create a machine learning model to predict COPD exacerbations. In regard to predicting exacerbations, the authors had an accuracy of 80%.

[7] Sivapalan et al. (2017) published a study titled "Prediction of COPD exacerbations using an artificial immune recognition system." In order to forecast COPD flare-ups, this team created an artificial immune recognition system (AIRS).

[9] Martnez-Camblor et al. (2016) released an article titled "Prediction of COPD exacerbation using a Bayesian network technique." A Bayesian network technique was used in this study to forecast COPD flare-ups. The authors' forecast of exacerbations had a 72.4% accuracy rate.

[10] "Predicting the onset of COPD: an application of machine learning" by Boutou et al. (2015) - In this study, machine learning

algorithms were employed to foretell when a group of smokers will develop COPD. The prediction of the beginning of COPD was accurate to 70.9% by the authors.

[11] Tan et al. (2015) released an article titled "Predicting COPD exacerbations using an ensemble of supervised machine learning algorithms and electronic health record data." Using information from electronic health records, this study created a collection of machine learning algorithms to forecast COPD exacerbations. The authors' forecast of exacerbations had a 75.2% accuracy rate.

[12] Interobserver agreement in the evaluation of B-lines using bedside ultrasound by John Gullett, John P Donnelly, et al (2015) assessed the level of B-line appearance on bedside ultrasonography in patients coming to the emergency department (ED) with acute undifferentiated dyspnea to see if there was agreement among qualified emergency doctors. In order to measure sonographic B-lines, we also identified which thoracic zones had the best level of interobserver reliability.

## 3. PROPOSED SYSTEM

The framework's main purpose is to deconstruct information, forecast, exchange data, acquire data, and illustrate the many stages of research into lung disease. Techniques used in AI and weaponry innovation: These studies are geared towards developing AI tools that can predict lung illness while an employee is at work. Then, at that moment, it tracks some of the research topics, problems, and demands that will be present in the future.

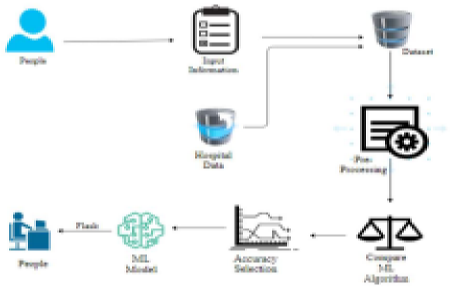


**Figure 1: Describes the System Architecture Data Pre-processing**

The AI system used to determine the AI error rate may be compared to the information error rate for intentionally set out information. You might not need an observation technique if the amount of information available is sufficiently large to address the population. In any case, work with commendable facts that might not be consistent with the public presentation in everyday life. Provide two characteristics and definitions, regardless of whether the type of data is a floating point or a whole integer, and locate the missing value. The model that consolidates the preparatory informational index to meet the hyper model was evaluated for reasonableness using the information model.

### Data Validation/ Cleaning/Preparing Process

Pressing the data imported the library pack. Analyze to determine modifications based on the type and sort of media, determine what's going on, and imitate the attributes. In order to show and manage the model plan, model approval depends on the model information that was stored during model preparation. You may use this information to more easily approve and test the laid-out information when evaluating the model. Renaming the provided data, unloading portions to analyse one change, two factors, or a variety of other factors are just a few ways to erase or edit information. Variations in steps and data innovation will occur between stores. The primary objective of data destruction is to identify and get rid of errors and failures in order to enlarge the value of the data for independent research with decision-making.

### Exploration data analysis of visualization

Scientific evidence for observational research Display mathematical and AI skills and data. Measurements really focus on information correlation and fine detail. Video clips provide information that is essential for a person's success. This may help in determining the design, damaged data, exterior, and much more when you're going through and differentiating datasets. Data perception may be used to describe and display essential connections in locations and plans that are more obvious and applicable to partners than practical or helpful ways when there is limited knowledge about the place. Media analysis and diary research are fields in and of themselves, requiring deep immersion in some of the concluding texts.

### Comparing algorithms and predictions in terms of their level of accuracy

It is crucial to consider how a variety of AI calculations will be presented, and we'll look at using test tools to assess various AI computation in Python along with scikit-learn. You may use this test along starting point for your AI problems before adding further data to examine. There will be a secondary capacity for each model. You may gain a general idea of how each model might be compatible with improbable numbers by using a recursive approach, such cross approval. It must to include the choice for using models to select a few of the excellent models from the model suite you created. Every time you receive new knowledge, you must grasp it by using multiple advancements to examine the material in unique ways. Selecting a model is one idea. To choose a few of your AI computations to accomplish, you should examine at the similar parts using a variety of ways. Using various imaging techniques will let you to demonstrate the real contrast, differentiation, and other model proliferation components.

## 4.ALGORITHM AND TECHNIQUES:

### Logistic Regression

Factual techniques for examining inserted data incorporate one from another free deciding outcomes. Results are guessed with two factors . The motivation behind the review, to find a decent model to clarify the connection which joins two helpful variables (contingent upon change ▪ reaction or change reaction) and free level (indicator or clarification). Calculated lapse is an AI machine that rundowns the calculations used to uncover conceivable class-based changes. In calculated relapse, changes depend on two factors that incorporate data composed as 1 (indeed, achievement, and so forth) or 0 (no, disappointment, and so on)

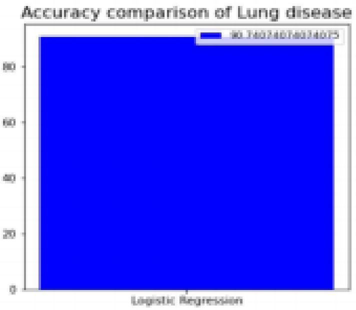


**Figure 2: Comparison of cellular breakdown in the lungs with Logistic Regression.**

### Random Forest Classifier

Common woodlands or normal timberlands are an approach to figuring out how to sort, retreat and different exercises, which is finished by building a great deal of choice trees during the preparation and delivering classes and strategies (traditional) or and that implies anticipating (withdrawing) the trees exclusively. Standard affirmation timberlands right for certificate past preparation. An average timberland is a kind of AI calculation that depends on outfit learning. Concentrating together is a sort of realizing where you enter various calculations or calculations commonly to have a solid feeling of prescience. The novel timberland calculation consolidates numerous calculations of a similar sort for example various choice trees, bringing about a tree backwoods, thus the name "Ordinary Forest". The backwoods calculation can be utilized for accumulation and sequencing assignments.

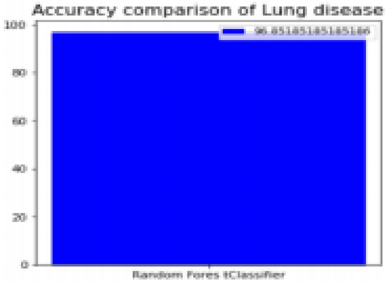


**Figure 3: Proper examination of cellular breakdown in the lungs with Random Forest Classifier.**

**Decision Tree Classifier**

You would require a collection of patient data, including pertinent medical history, demographic data, and test results, to create a decision tree classifier for predicting COPD. It would be necessary to label the dataset, which would categorise each patient as either having COPD or not.

After you have the dataset, you may create a decision tree classifier using a machine learning toolkit, such as scikit-learn in Python. The decision tree algorithm divides the data recursively into subsets according to the values of the input characteristics, maximizing the information gain or minimizing the entropy at each split.

Decision tree classifiers include drawbacks, such as overfitting to the training data and being sensitive to minute changes in the data, which should be kept in mind. Hence, to avoid overfitting and enhance the classifier's generalizability to fresh data, it's essential to use the right approaches, such cross-validation and feature selection.

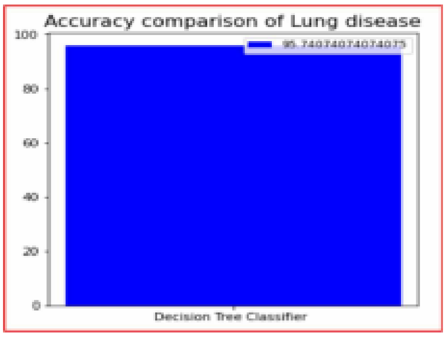


**Figure 4: Comparison of cellular breakdown in the lungs with Decision Tree Classifier**

**Naive Bayes**

One method for estimating the chance of developing COPD is the Naive Bayes algorithm . The naive Bayes method, which is based on the Bayes theorem, makes the assumption that the presence of one feature does not rely on the presence of other characteristics.

We would first require a dataset comprising data on individuals with and without COPD in order to utilize Naive Bayes for predicting COPD. Age, gender, smoking status, family history of COPD, and the results of pulmonary function tests are common elements of this dataset.

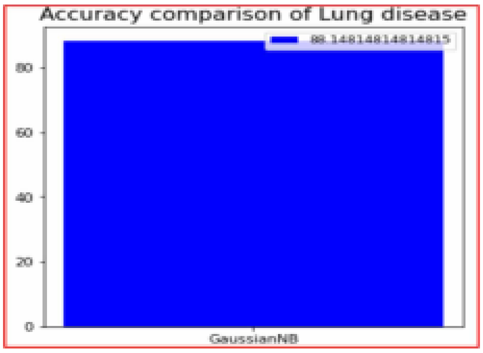


**Figure 5: Comparison of cellular breakdown in the lungs with Naive Bayes**

## 5.RESULT

**Prediction result by accuracy:**

The strategic relapse calculation likewise utilizes straight correlations with free indicators to anticipate values. A given worth can be anyplace in the center of endlessness. This requires the result of the calculation to be variable. A definitive significance of forecasting results is an impeccably adjusted relapse model.
Genuine positive rate (TPR) = TP/(TP + FN)
Positive Ratio (FPR) = FP/(FP + TN)
Reality: The proportion of the anticipated worth to the all out example size is a decent indicator of how regularly the indicator will precisely foresee the payers and non-payers.
Compute:
Valid = (TP + TN)/(TP + TN + FP + FN)
Truth is a proportion of profound activity, a proportion of cautious perception and study. Assuming we are straightforward, we might imagine that our model is great. Indeed, being straightforward is no joking matter, however it is to the point of being practical regarding the worth of untruths and malevolence.

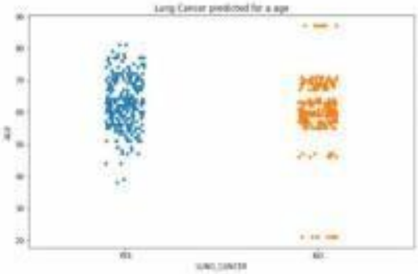


**Figure 5: Lung malignant growth is anticipated at various ages**

Clarification: An illustration of a decent prediction.
Note = TP/(TP + FP)
If it's not too much trouble, note that this is an all around arranged and all around arranged correlation. The inquiry is the number of these companions made due. The top quality is because of the low level of untruths. We got an exactness of 0.788, which is generally excellent.
Keep in mind: Predictable evident worth. (The model of genuine givers will be a decent indicator)
Recollect = TP/(TP + FN)
Memory (Sensation) - Memory is a amount of consistency and cautious approach in all classifications - yes.

The F1 score is an exact equilibrium of Accuracy and Memory. So these focuses get both great and awful. It's difficult all things considered, however F1 has high concern than devotion, especially assuming you have an alternate allotment. In the event that the fact of the matter is equivalent to the cost is very similar. Assuming the cost of good and awful is altogether different, you should check Accuracy and Memory out.

General cycle:
F-rate = (2TP)/(2TP + FP + FN)
F1 focuses:
F1 score = 2 * (memory * Note )/(memory + note)

Misleading Benefit (FP): The payer is thought to be the moneylender. At the point when there is no genuine arrangement and the speculative characterization is so. For instance, on the off chance that the actual order says that this accomplice didn't make due, however the accepted classifications say that this accomplice will get by.

Bogus Evil (FN): some unacceptable individual anticipated vengeance. At the point when the genuine order is thought to be thus, however the characterization isn't.

Genuine Benefit (TP): An individual who doesn't pay is anticipated to be a bank. These are accurately anticipated values, and the anticipated class esteem is yes and the anticipated classification esteem is yes.

Genuine Evil (TN): some unacceptable individual anticipated vengeance. These are accurately expected values, and that implies the worth of the accepted class isn't equivalent to the worth of the expected classification.

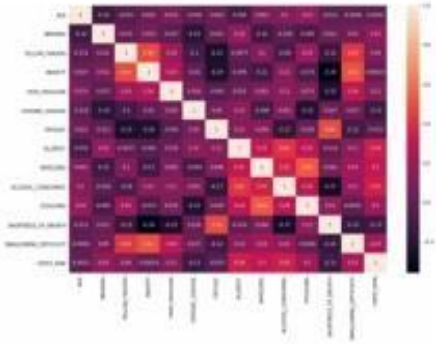


**Figure 6: Use measurements to track down reality**

## 6. CONCLUSION:

In conclusion, supervised machine learning algorithms may be employed as precise and affordable techniques considering the diagnosis of COPD. To judge the effectiveness of these algorithms in broader and more varied patient groups, more study is necessary. Efforts should also be made to render these algorithms visible and understandable so that doctors may comprehend the underlying reasoning and decide on patient treatment with confidence. In the long run, supervised machine learning algorithms have the potential to improve COPD diagnosis and therapy, leading to better patient outcomes.

## 7. REFERENCES:

[1] F. A. Marcano-Belisario et al 's "Predictive models for exacerbation risk in chronic obstructive pulmonary disease using electronic health record data: A systematic review," was published in 2020.

[2] Tandel et al. (2020) published a study titled "Prediction of COPD using machine learning algorithms using spirometry data". View at :Google Scholar.

[3] Sánchez-Morillo et al. (2019) published "A machine learning technique for COPD identification using spirometry and demographic data."

[4] Shen and others in 2018 published a study titled "Prediction of COPD hospitalisation using machine learning and administrative data from electronic health records."View at :Google Scholar.

[5] Rahimi et al. (2017) published "COPD exacerbation prediction using machine learning technique"

[6] N. Anantrasirichai, W. Hayes, M. Allinovi, D. Bull, and A. Achim, "Line Detection as an Inverse Problem: Application to Lung Ultrasound Imaging," IEEE Trans Med Imaging, vol. 36, no. 10, pp. 2045-2056, Oct 2017, doi: 10.1109/TMI.2017.2715880.

[7] Sivapalan et al. (2017) published a study titled "Prediction of COPD exacerbations using an artificial immune recognition system." View at :Google Scholar.

[8] X. Zhang "Lung ultrasound surface wave elastography: a pilot clinical study," IEEE transactions on ultrasonics, ferroelectrics, and frequency control, vol. 64, no. 9, pp. 1298-1304, 2017, doi: 10.1109/TUFFC.2017.2707981.

[9] Martnez-Camblor et al. (2016) released an article titled "Prediction of COPD exacerbation using a Bayesian network technique."

[10]"Predicting the onset of COPD: an application of machine learning" by Boutou et al. (2015) View at :Google Scholar.

[11] Tan et al. (2015) released an article titled "Predicting COPD exacerbations using an ensemble of supervised machine learning algorithms and electronic health record data."

[12] J. Gullett "Interobserver agreement in the evaluation of B-lines using bedside ultrasound," Journal of critical care, vol. 30, no. 6, pp. 1395-1399, 2015.

[13]H. Sekiguchi "Critical care ultrasonography differentiates ARDS, pulmonary edema, and other causes in the early course of acute hypoxemic respiratory failure," Chest, vol. 148, no. 4, pp. 912-918, 2015.

[13] L. J. Brattain, B. A. Telfer, A. S. Liteplo, and V. E. Noble, "Automated B-line scoring on thoracic sonography," J Ultrasound Med, vol. 32, no. 12, pp. 2185-90, Dec 2013, doi: 10.7863/ultra.32.12.2185.

[14] G. Baldi "Lung water assessment by lung ultrasonography in intensive care: a pilot study," Intensive care medicine, vol. 39, no. 1, pp. 74-84, 2013.