PREDICTION OF COPD USING MACHINE LEARNING TECHNIQUE

A PROJECT REPORT

Submitted By

SIVA KUMAR G - 111518106150

VIGNESH R - 111518106173

VINOD KUMAR P - 111518106177

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

R.M.D. ENGINEERING COLLEGE

(AN AUTONOMOUS INSTITUTION)

R.S.M. NAGAR, KAVARAIPETTAI – 601 206

MAY 2022

R.M.D. ENGINEERING COLLEGE

(AN AUTONOMOUS INSTITUTION)

R.S.M. NAGAR, KAVARAIPETTAI-601 206

BONAFIDE CERTIFICATE

Certified that this project report **“PREDICTION OF COPD USING MACHINE LEARNING”** is the bonafide work of **“SIVA KUMAR G, VIGNESH R, VINOD KUMAR P”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.K. HELEN PRABHA	**Mr. JAI GANESH B**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
Department of Electronics and Communication Engineering	Department of Electronics and Communication Engineering
R.M.D. Engineering College	R.M.D. Engineering College
Kavaraipettai – 601 206	Kavaraipettai - 601 206

The Viva-Voce Examination of the students who have submitted this project work is held on ___________________

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

At this outset, we would like to express our gratitude to our beloved and respected **Thiru. R.S. MUNIRATHINAM, Chairman, R.M.D Engineering College**.

We would like to thank **Thiru. R.M. KISHORE, Vice chairman** for his encouragement, and our deepest gratitude for **Dr. N. ANBUCHEZHIAN B.E, M.S, M.B.A, M.E, Ph.D Principal** for his support during course of the project.

We express our sincere gratitude **to Dr. K. HELEN PRABHA, M.E, Ph. D Professor, Head of the Department of Electronics and Communication Engineering**, who has been a guiding force and constant source of inspiration to us.

We are very thankful to our Supervisor **Mr.JAI GANESH B, M.E, Ph.D** for having extended her fullest co-operation and guidance. We also thank for his constant support and patience. We also take this opportunity to thank all the staff of our department for having encouraged us in completing this project successfully.

We also thank our **Parents** for their unparalleled love and moral support and finally the **Almighty** for showering his generous blessing on us, without whom we would have not gone this far

ABSTRACT

Ultrasound insonation of lungs that are dense with extra vascular 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. 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 learning algorithms

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	CONVOLUTIONAL NEURAL NETWORK.
AI	ARTIFICIAL INTELLIGENCE
ML	MACHINE LEARNING
NLP	NATURAL LANGUAGE PROCESSING
ER	ENTITY RELATION
HTML	HYPER TEXT MARKUP LANGUAGE
CSS	CASCADING STYLE SHEETS

CHAPTER 1

INTRODUCTION

1.1 DATA SCIENCE

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

The term “data science” was first coined in 2008 by D.J. Patil, and Jeff Hammerbacher, the pioneer leads of data and analytics efforts at LinkedIn and Facebook. In less than a decade, it has become one of the hottest and most trending professions in the market.

Data Scientist:

Data scientists examine which questions need answering and where to find the related data. They have business acumen and analytical skills as well as the ability to mine, clean, and present data. Businesses use data scientists to source, manage, and analyze large amounts of unstructured data.

Required Skills for a Data Scientist:

- **Programming**: Python, SQL, Scala, Java, R, MATLAB.
- **Machine Learning**: Natural Language Processing, Classification, Clustering.
- **Data Visualization**: Tableau, SAS, D3.js, Python, Java, R libraries.
- **Big data platforms**: MongoDB, Oracle, Microsoft Azure, Cloudera.

1.2 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals. Leading AI textbooks define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers.

AI research has tried and discarded many different approaches during its lifetime, including simulating the brain, modeling human problem solving, formal logic, large databases of knowledge and imitating animal behavior. In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly successful, helping to solve many challenging problems throughout industry and academia.

AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

Learning processes. This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Reasoning processes. This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

Self-correction processes. This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.

Artificial neural networks and deep learning artificial intelligence technologies are quickly evolving, primarily because AI processes large amounts of data much faster and makes predictions more accurately than humanly possible.

Natural Language Processing (NLP):

Natural language processing (NLP) allows machines to read and understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straightforward applications of natural language processing include information retrieval, text mining, question answering and machine translation. Many current approaches use word co-occurrence frequencies to construct syntactic representations of text. "Keyword spotting" strategies for search are popular and scalable but dumb; a search query for "dog" might only match documents with the literal word "dog" and miss a document with the word "poodle". "Lexical affinity" strategies use the occurrence of words such as "accident" to assess the sentiment of a document. Modern statistical NLP approaches can combine all these strategies as well as others, and often achieve acceptable accuracy at the page or paragraph level.

1.3 MACHINE LEARNING

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

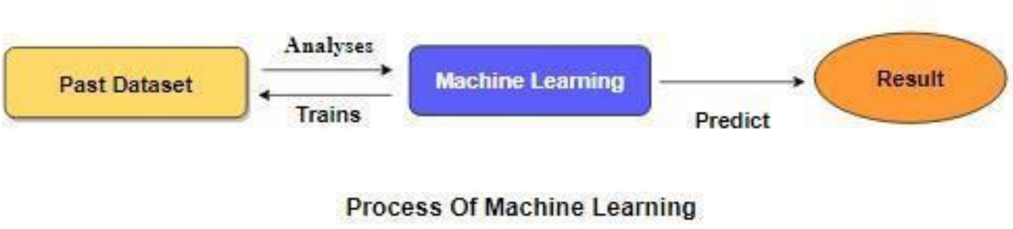


Fig. 1.1 Process of Machine Learning

Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output **is y = f(X).** The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include **logistic regression**, **multi-class classification**, **Decision Trees** and **support vector machines etc**. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into **Classification** problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes.

1.4 PREPARING THE DATASET

This dataset contains 541 records of features extracted from Lung Patients, which were then classified into 2 classes:

- Affected
- Not affected

1.5 PROPOSED SYSTEM

Exploratory Data Analysis of pulmonary disease Prediction

Pulmonary lung datasets from different sources would be combined to form a generalized dataset, and then different machine learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using machine learning algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Building the classification model

The predicting the pulmonary disease problem, ML algorithms 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.

1.6ADVANTAGES

- These reports are to the investigation of applicability of machine learning techniques for pulmonary disease prediction in operational conditions.
- Finally, it highlights some observations on future research issues, challenges, and needs.

1.7. EXISTING SYSTEM

They are used the surface elastic properties of the lung in patients presenting with symptomatic pulmonary edema. Surface wave speeds of the lung were quantified at bedside using a novel LUSWE technique in combination with standard 2DLUS. We observed significant differences in lung surface wave speed at three frequencies between baseline and follow-up testing. We speculate that these differences are driven by a reduction in lung stiffness due to a decrease in EVLW from diuretic therapy. The calculated reduction in surface wave speed magnitude correlated with other markers of decreased EVLW including a decline in the burden of B-lines and successful fluid removal by diuretics. B-Mode 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. B-mode lung ultrasound and LUSWE assessment of the lungs were performed using anterior and lateral intercostal spaces in the supine patient. 14 patients were evaluated at admission with reassessment performed 1-2 days after initiation of diuretic therapy. Each exam recorded the total lung B-lines, lung surface wave speeds (at 100, 150, and 200 Hz) and net fluid balance. The patient cohort experienced effective diuresis (average net fluid balance of negative 2.1 liters) with corresponding decrease in pulmonary edema visualized by B-mode ultrasound (average decrease of 13 B-Lines).

1.8 DRAWBACKS

- 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.

CHAPTER 2
LITERATURE SURVEY

General

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

Review of Literature Survey

Title : Lung Cancer Prediction using Data Mining Techniques

Author: E. Yatish Venkata Chandra, K. Ravi Teja, M.Hari Chandra Siva Prasad, Mohammed.Ismail.B

Year :2019

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 early cancer 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 two types of small cell lung carcinoma (SCLC), non-small cell lung carcinoma (NSCLC). The main reason for lung cancer is smoking of cigarette. There are many researches targeting on exact approaches for treating cancer. To predict the survival rate for NSCLC patients data mining techniques can be used with selection of algorithms. The algorithms used to detect the lung cancer are Support vector machine (SVM), Decision tree, k-Nearest neighbour, Random forest, Logistic regression.

Title : Lung disease prediction using image processing and CNN algorithm
Author: Shivani Kasar1, Darshan Hujband2, Abhijeet Ahire3, Jyoti Rahade4

Year :2020

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. In numerous parts of the planet far reaching screening by CT or MRI isn't yet pragmatic, in order that midsection radiology stays in starting and most elementary system. Firstly, we'll utilize some systems are key to the errand of medicinal picture mining, Lung Field Segmentation, processing, Feature Extraction, Classification utilizing neural system and SVMs. The routines utilized as part of this paper work states to group computerized X-beam midsection movies into two classes: ordinary and weird. Diverse learning examinations were performed on two distinctive information sets, made by method for highlight choice and SVMs prepared with diverse parameters; the outcomes are checked out and reported.

Title : Lung Cancer Prediction using Feed Forward Back Propagation Neural Networks with Optimal Features

Author: Dr. S. Senthil, B. Ayshwarya

Year :2018

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. In this paper a computer aided classification method for lung cancer prediction based an evolutionary system by a combination of architectural evolution with weight learning using neural network and Particle Swarm Optimization is implemented. This method proposed different variants and hybridize it with evolutionary algorithm to improve its performance and uses global searching of PSO and local searching ability of neural network gives better lung cancer prediction as cancerous and non-cancerous. 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.

Title :A Computer Aided Diagnosis of Lung Disease using Machine Learning Approach

Author: Subapriya V, Jaichandran R, Shunmuganathan K.L, Abhiram Rajan, Akshay T, Shibil Rahman

Year :2020

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 lungs can also be impacted by cancer from other bodies. The metastases are named as cancer cells migrate from organ to organ. Lung cancers are normally grouped into two major cell and non-small cell types. In this study we predict a Computer Aided Diagnosis (CAD) for lung cancer prediction using Convolutional Neural Network (CNN) and ML approach.

Title : Detection and Prediction of Lung Cancer Using Different Algorithms
Author: N. Sudhir Reddy, V Khanaa

Year :2019

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. In this work, decision tree algorithm is used for prediction of lung cancer wherein the important pattern with their corresponding weightage and score is studied. Processes such as preprocessing of images and feature extraction are done using Histogram Equalization and using neural network classifier does normal or abnormality check of the patient. Therefore, an easy, cost effective and time saving method will produce promising result for detection and prediction of lung cancer

CHAPTER 3

SYSTEM STUDY

3.1 OBJECTIVES

The goal is to develop a machine learning model for Pulmonary Disease Prediction, to potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

3.2 PROJECT GOALS

- Exploration data analysis of variable identification
  - Loading the given dataset
  - Import required libraries packages
  - Analyze the general properties
  - Find duplicate and missing values
  - Checking unique and count values

- Uni-variate data analysis
  - Rename, add data and drop the data
  - To specify data type

- Exploration data analysis of bi-variate and multi-variate
  - Plot diagram of pairplot, heatmap, bar chart and Histogram

- Method of Outlier detection with feature engineering
  - Pre-processing the given dataset
  - Splitting the test and training dataset
  - Comparing algorithm to predict the result based on the best accuracy

3.3 SCOPE OF THE PROJECT

Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

CHAPTER 4

PROJECT OUTLINE

4.1 PROJECT REQUIREMENTS

General:

Requirements are the basic constraints that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
   - A. Hardware requirements
   - B. software requirements

4.2 FUNCTIONAL REQUIREMENTS

The software requirements specification is a technical specification of requirements for the software product. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

4.3 NON-FUNCTIONAL REQUIREMENTS:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

4.4. ENVIRONMENTAL REQUIREMENTS

1. Software Requirements:

Operating System	: Windows
Tool	: Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor	: Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

4.5 FEASIBILITY STUDY

Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

Building the classification model

The prediction of Heart attack, 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.

Construction of a Predictive Model

Machine learning needs data gathering have lot of past data. 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.

CHAPTER 5

SYSTEM ARCHITECTURE

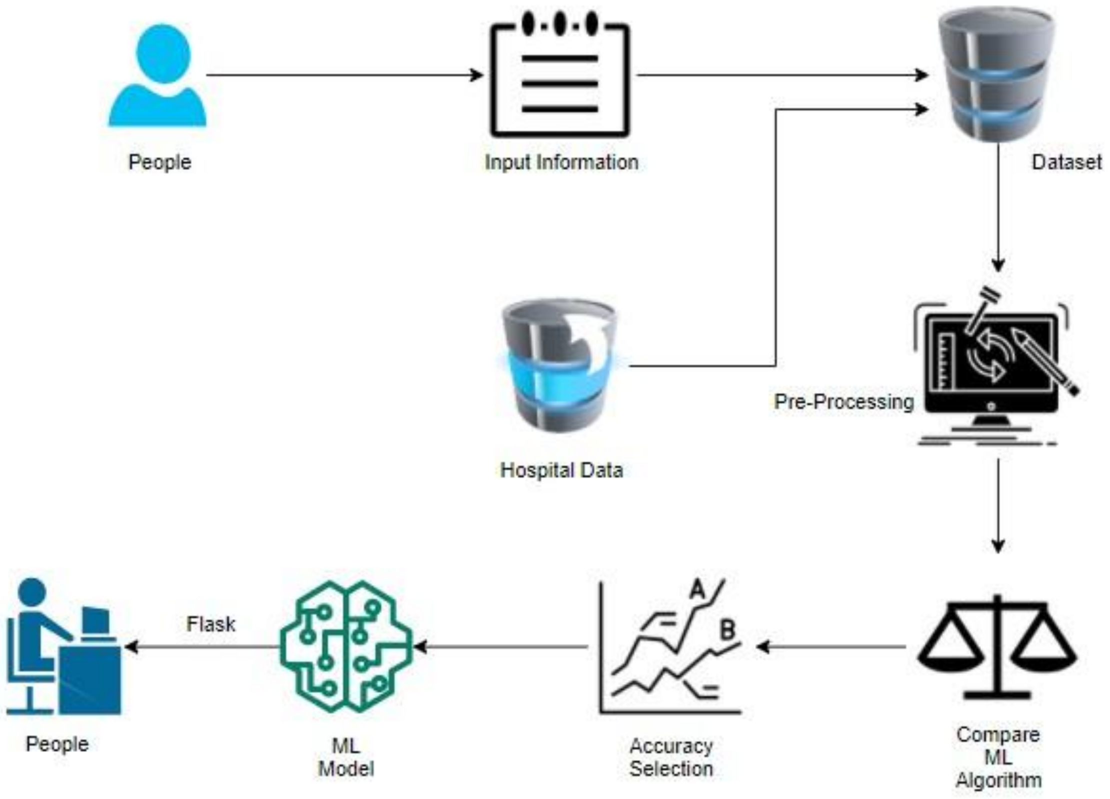


Fig 5.1 System Architecture

5.1 WORKFLOW DIAGRAM

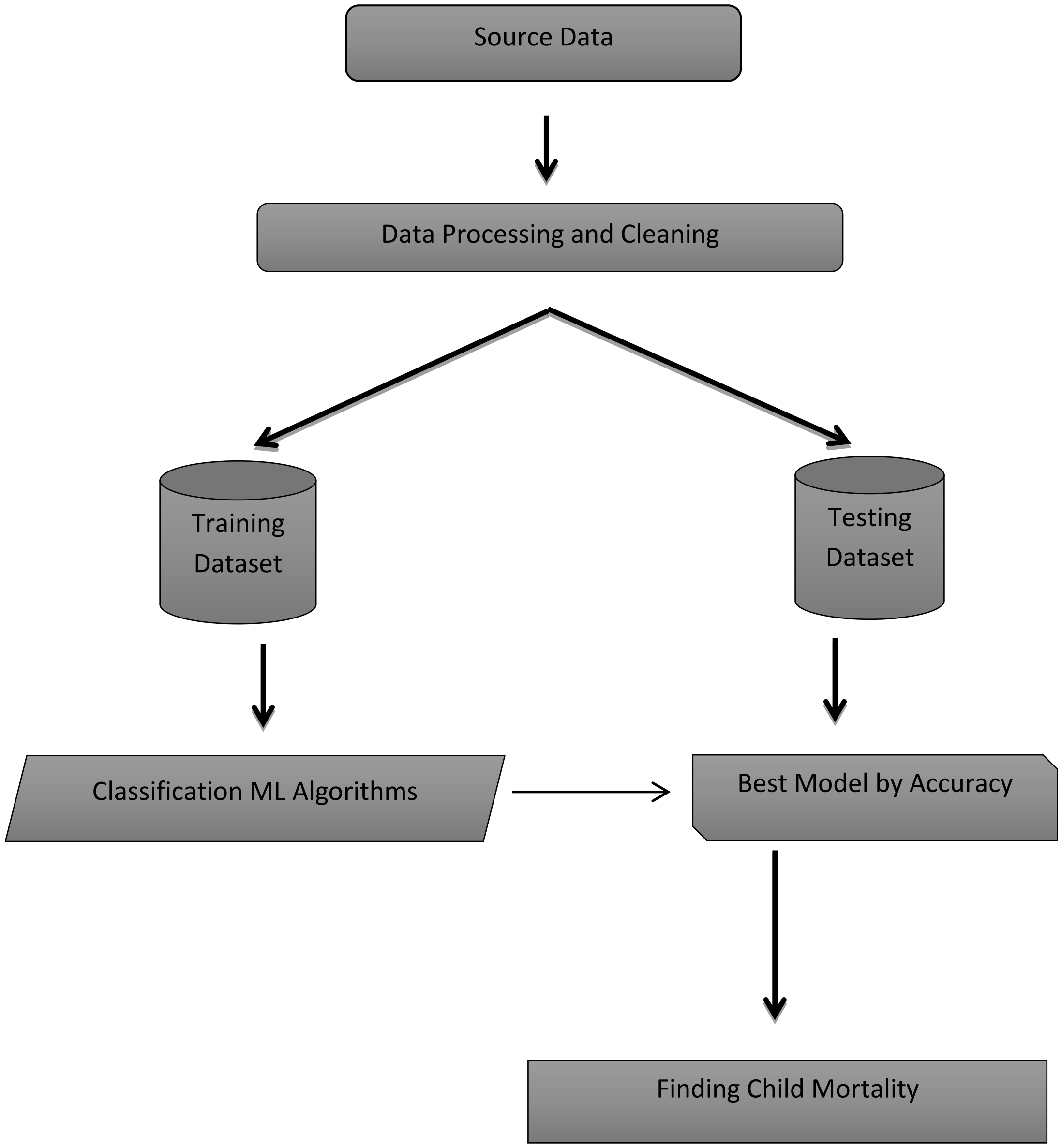


Fig 5.2 Workflow Diagram

5.2 USE CASE DIAGRAM

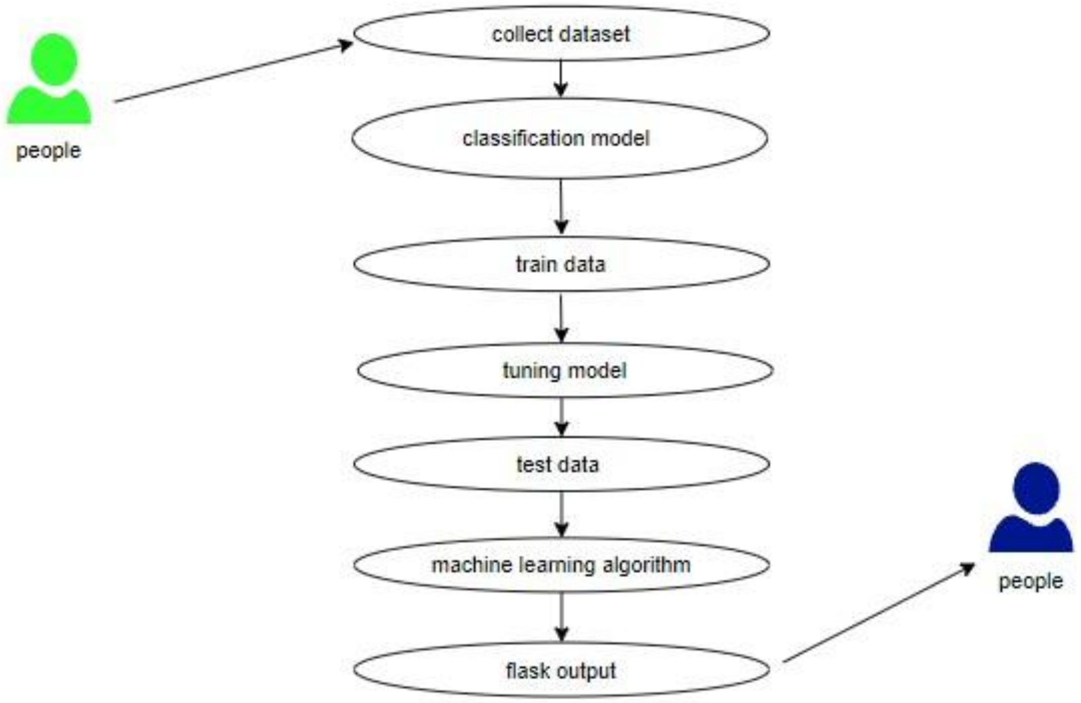


Fig 5.3 Use Case Diagram

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

5.3 CLASS DIAGRAM

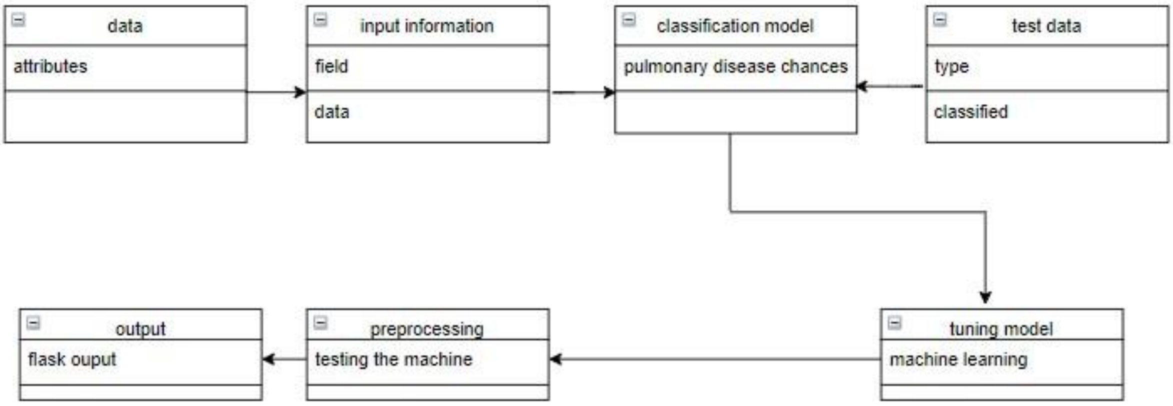


Fig 5.4 Class Diagram

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

5.4 ACTIVITY DIAGRAM

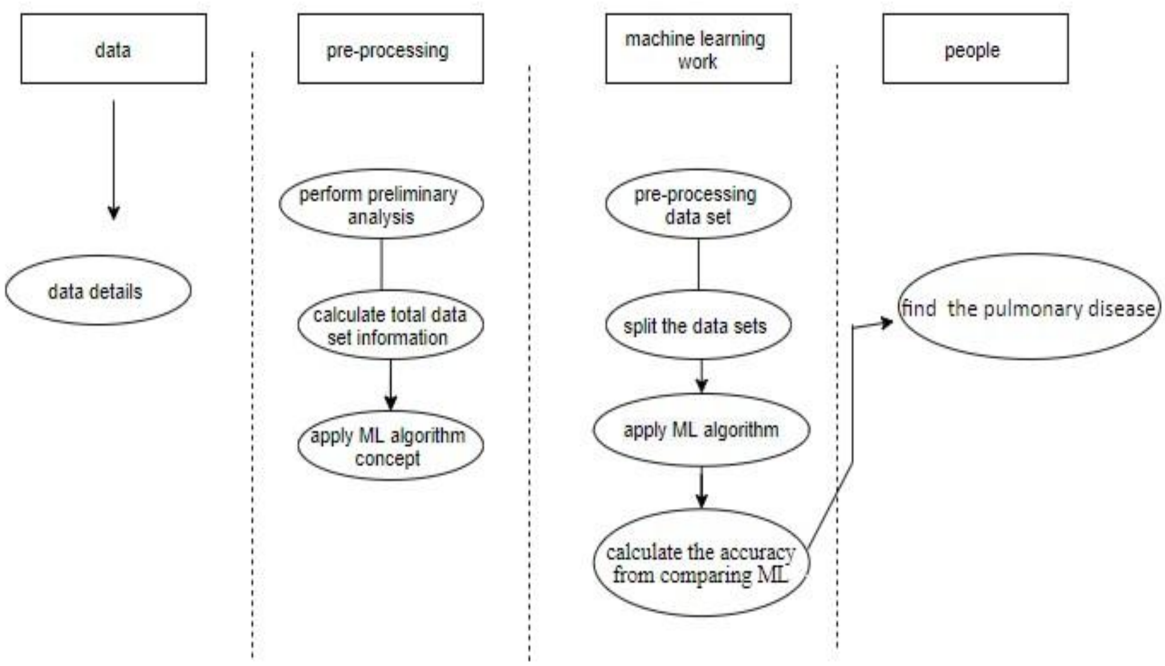

data
pre-processing
machine learning work
people
data details
perform preliminary analysis
calculate total data set information
apply ML algorithm concept
pre-processing data set
split the data sets
apply ML algorithm
calculate the accuracy from comparing ML
find the pulmonary disease

Fig 5.5 Activity Diagram

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

5.5 SEQUENCE DIAGRAM

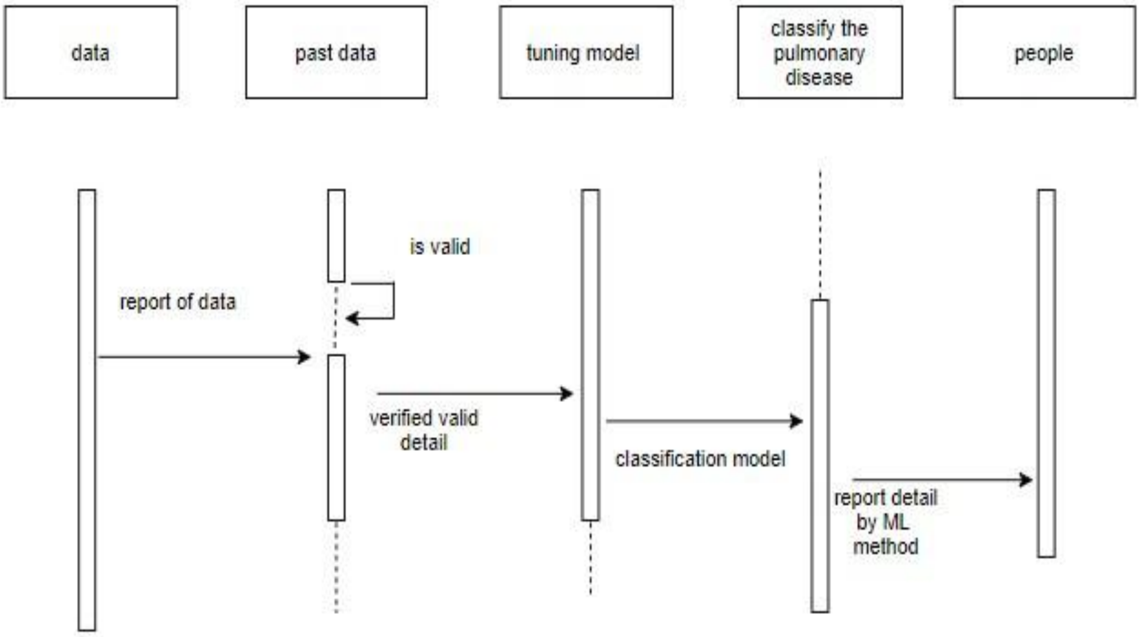


Fig 5.6 Sequence Diagram

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design-level models for modern business application development.

5.6 ENTITY RELATIONSHIP DIAGRAM (ERD)

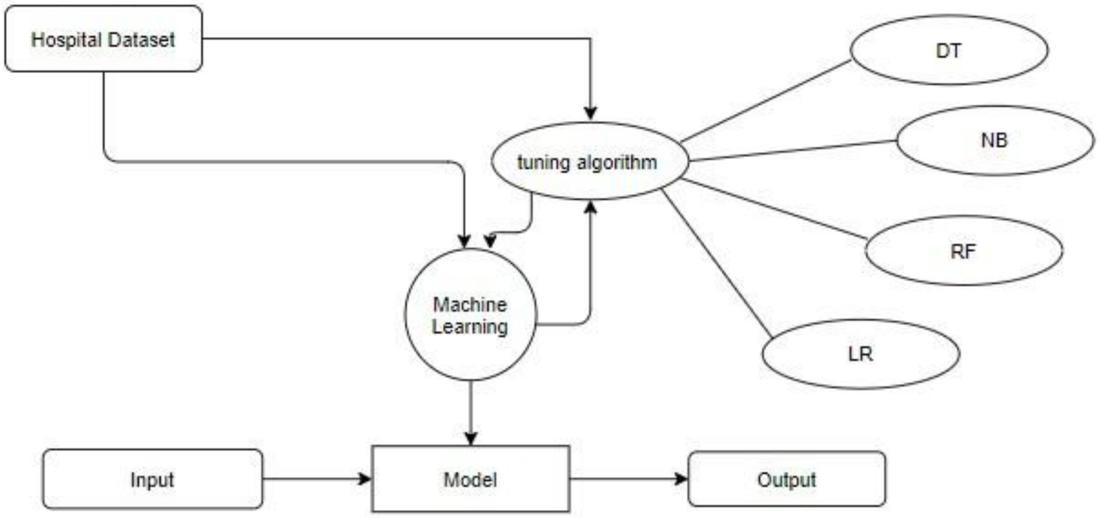


Fig 5.7 ER Diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

CHAPTER 6

SOFTWARE DESCRIPTION

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system "Conda". The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda.

6.1 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)

Navigator allows you to launch common Python programs and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository.

Conda:

Conda is an open source, cross-platform, language-agnostic package manager and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

6.2 JUPYTER NOTEBOOK

This website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started.

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird's eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well-known steps:

- Define Problem.
- Prepare Data.
- Evaluate Algorithms.
- Improve Results.
- Present Results.

6.3 PYTHON

Introduction:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Pythonis dynamically-typed and garbage-collected.It supportsmultiple programming paradigms,including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Python consistently ranks as one of the most popular programming languages

History:

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a 5-member "Steering Council" to lead the project. As of 2021, the current members of this council are Barry Warsaw, Brett Cannon, Carol Willing, Thomas Wouters, and Pablo Galindo Salgado.

Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues, leading to possible remote code execution and web cache poisoning.

Design Philosophy & Feature

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta-programming and meta-objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

Python's developers aim to keep the language fun to use. This is reflected in its name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (a reference to a Monty Python sketch) instead of the standard foo and bar.

Syntax and Semantics:

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but are rarely, if ever, used. It has fewer syntactic exceptions and special cases than C or Pascal.

Indentation:

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block.

Statements and control flow:

Python's statements include:

- The assignment statement, using a single equals sign =.
- The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
- The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The while statement, which executes a block of code as long as its condition is true.
- The Try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
- The raise statement, used to raise a specified exception or re-raise a caught exception.
- The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The def statement, which defines a function or method.
- The break statement, exits from a loop.
- The continue statement, skips this iteration and continues with the next item.
- The del statement, removes a variable, which means the reference from the name to the value is deleted and trying to use that variable will cause an error.

Expressions:

Some Python expressions are similar to those found in languages such as C and Java, while some are not:

- Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division (or integer division) // and floating-point/division. Python also uses the ** operator for exponentiation.
- From Python 3.5, the new @ infix operator was introduced. It is intended to be used by libraries such as NumPy for matrix multiplication.
- From Python 3.8, the syntax :=, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.
- In Python, == compares by value, versus Java, which compares numerics by value and objects by reference. (Value comparisons in Java on objects can be performed with the equals() method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example A<=B<=C.
- Python uses the words and, or, not for or its boolean operators rather than the symbolic &&, ||, ! used in Java and C.
- Python has a type of expression termed a list comprehension as well as a more general expression termed a generator expression.
- Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.
- Conditional expressions in Python are written as x if c else y (different in order of operands from the c ? x : y operator common to many other languages).
- Python makes a distinction between lists and tuples. Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be immutable in Python).
- In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality. For example:

- List comprehensions vs. for-loops
- Conditional expressions vs. if blocks
- The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as a=1 cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c==1) {…} is syntactically valid (but probably unintended) C code but if c=1: … causes a syntax error in Python.

Methods:

Methods on objects are functions attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self parameter access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

Typing:

Python uses duck typing and has typed objects but untyped variable names. Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically-typed, Python is strongly-typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

MODULE DESCRIPTION:

Data Pre-processing

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.

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.

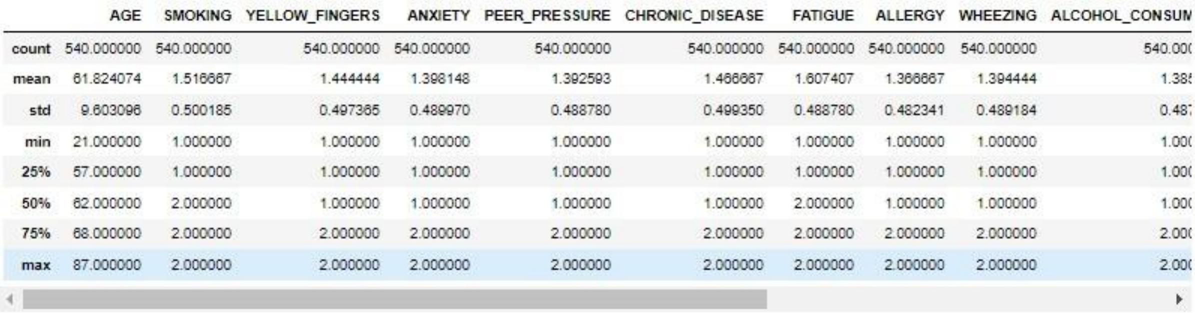
Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- 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

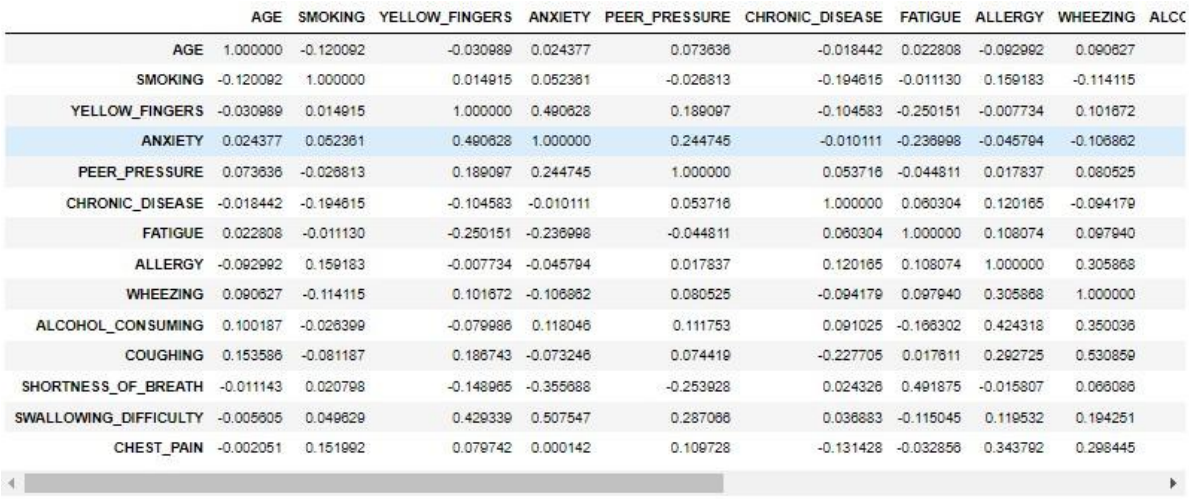
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To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

Data Validation/ Cleaning/Preparing Process

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.

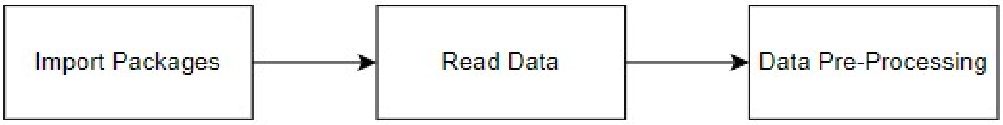


	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC_DISEASE	FATIGUE	ALLERGY	WHEEZING	ALCOHOL_CONSUM
count	540.000000	540.000000	540.000000	540.000000	540.000000	540.000000	540.000000	540.000000	540.000000	540.00
mean	61.824074	1.516667	1.444444	1.398148	1.392593	1.466667	1.607407	1.366667	1.394444	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



	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	

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : removing noisy data

Exploration data analysis of visualization

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.

How to summarize data distributions with histograms and box plots.

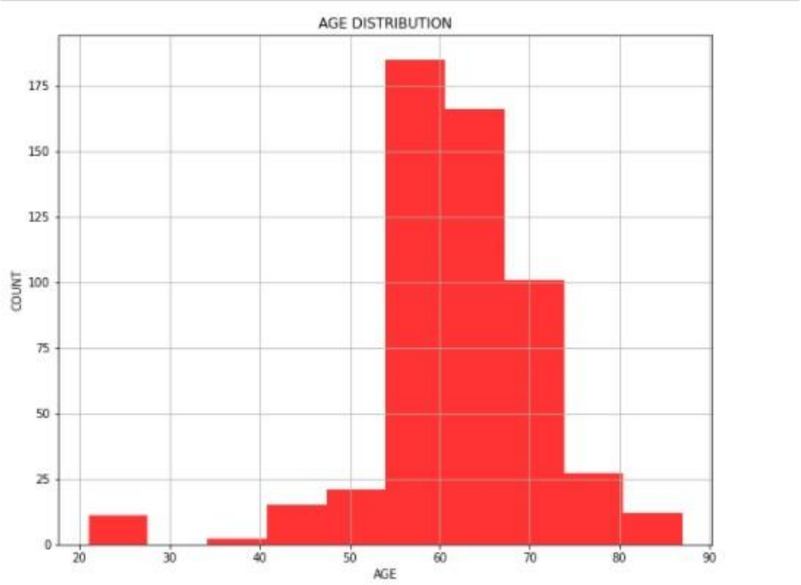
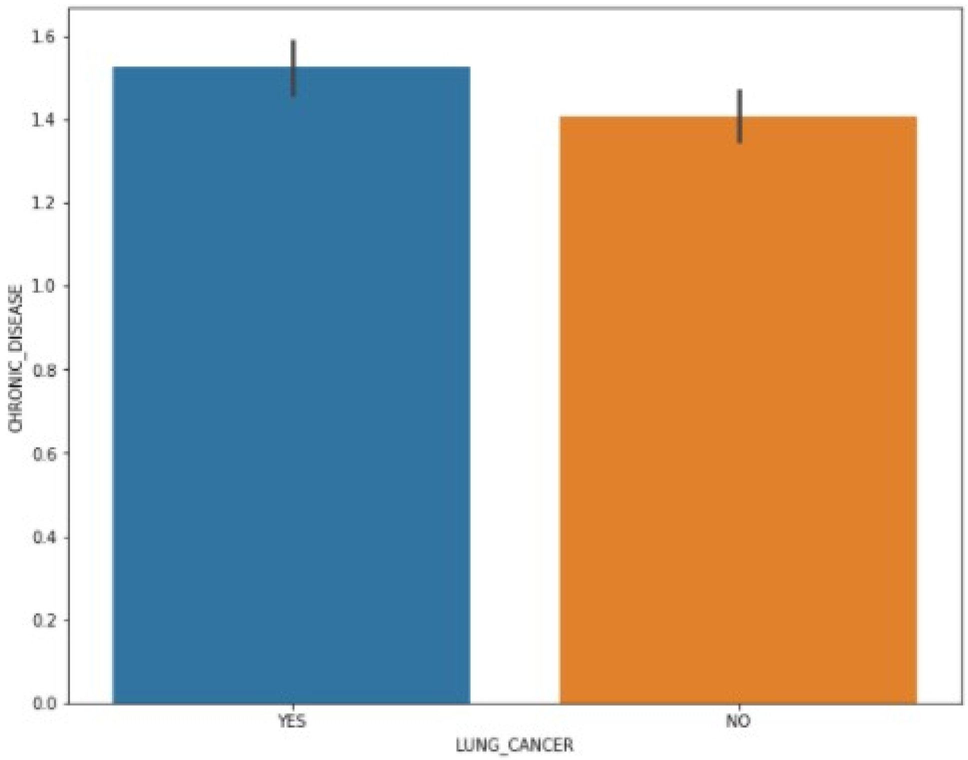
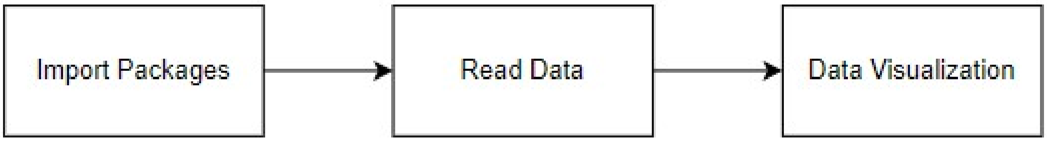


Fig 6.1 Age Distribution



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.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : visualized data

Comparing Algorithm with prediction in the form of best accuracy result

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 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.

In the example below 4 different algorithms are compared:

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

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate(TPR) = TP / (TP + FN)

False Positive rate(FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

Precision = TP / (TP + FP)

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

General Formula:

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

ALGORITHM AND TECHNIQUES

Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.

Logistic Regression

It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.

In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.
- Logistic regression requires quite large sample sizes.

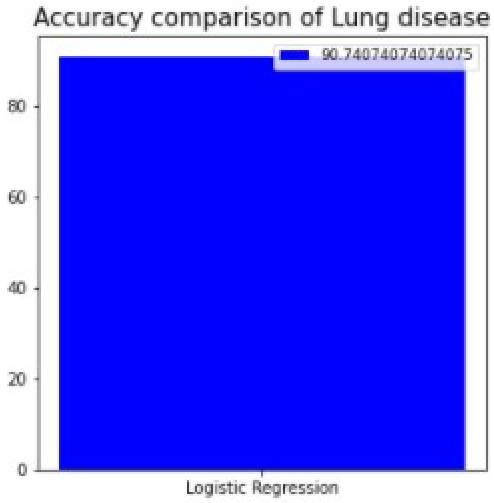
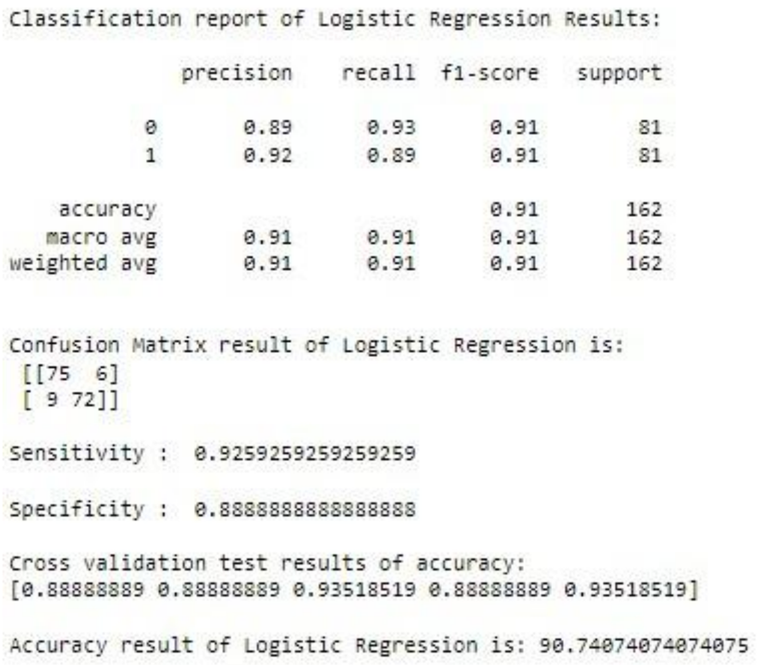
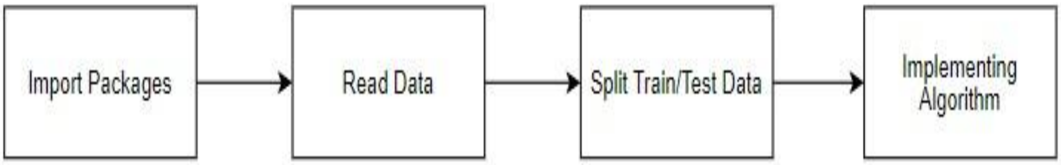


Fig. 6.2 Accuracy comparison – Logistic Regression



```
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


Confusion Matrix result of Logistic Regression is:
 [[75  6]
 [ 9 72]]

Sensitivity :  0.9259259259259259

Specificity :  0.8888888888888888

Cross validation test results of accuracy:
[0.88888889 0.88888889 0.93518519 0.88888889 0.93518519]

Accuracy result of Logistic Regression is: 90.74074074074075
```

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

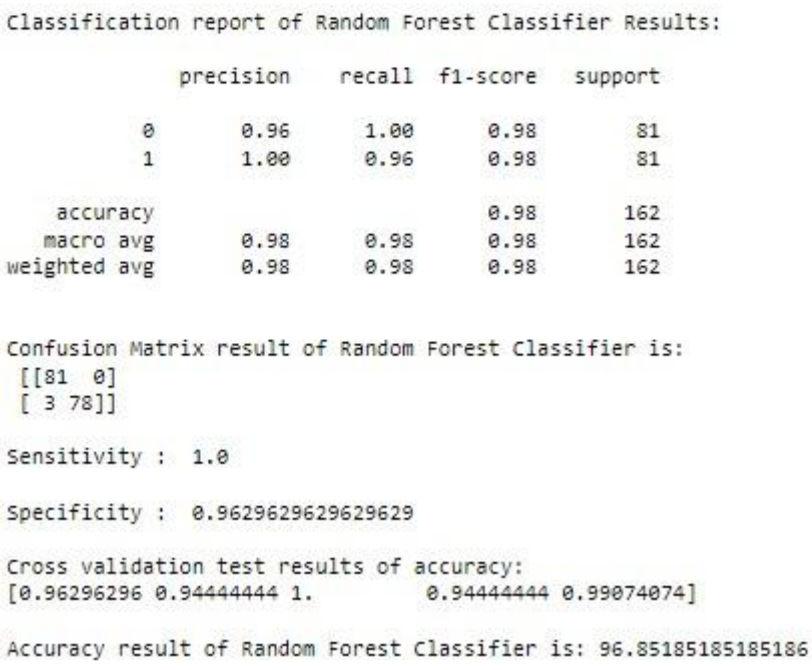
output : getting accuracy

Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps 1 and 2.



```
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

Confusion Matrix result of Random Forest Classifier is:
 [[81  0]
 [ 3 78]]

Sensitivity :  1.0

Specificity :  0.9629629629629629

Cross validation test results of accuracy:
[0.96296296 0.94444444 1.         0.94444444 0.99074074]

Accuracy result of Random Forest Classifier is: 96.85185185185186
```

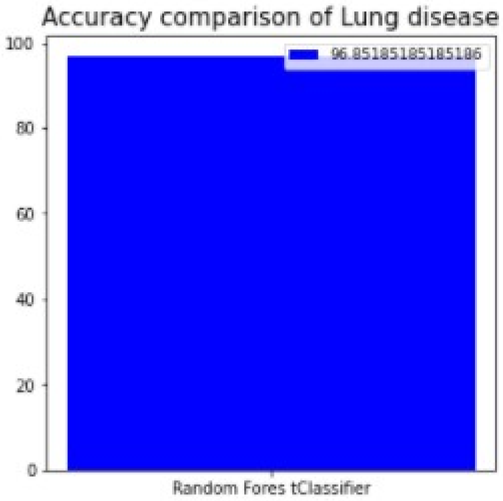
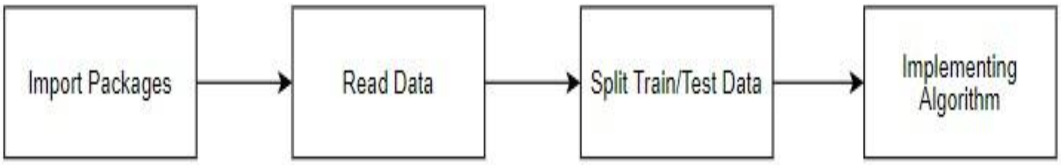


Fig. 6.3 Accuracy comparison – Random Forest Classifier

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

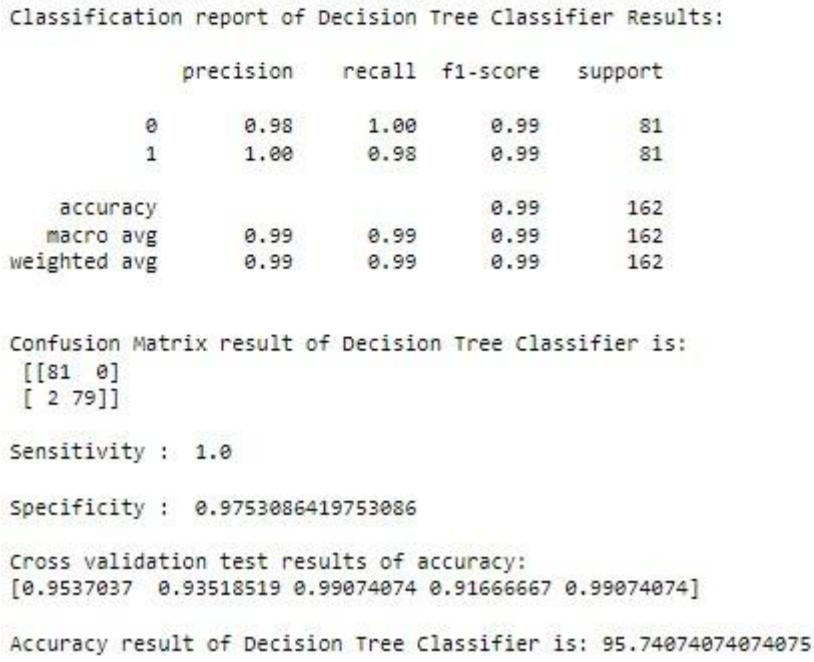
output : getting accuracy

Decision Tree Classifier

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

- At the beginning, we consider the whole training set as the root.
- Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
- On the basis of attribute values records are distributed recursively.
- We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.



```
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


Confusion Matrix result of Decision Tree Classifier is:
 [[81  0]
 [ 2 79]]

Sensitivity :  1.0

Specificity :  0.9753086419753086

Cross validation test results of accuracy:
[0.9537037  0.93518519 0.99074074 0.91666667 0.99074074]

Accuracy result of Decision Tree Classifier is: 95.74074074074075
```

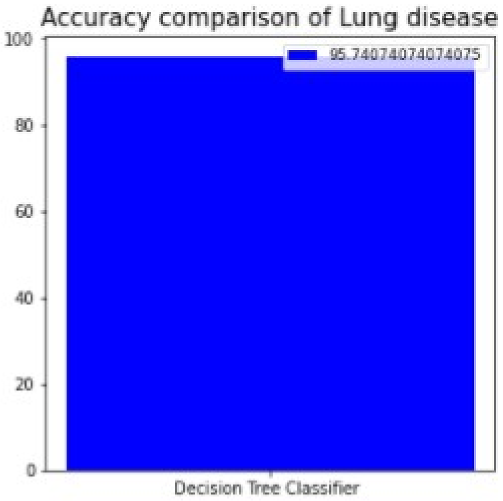
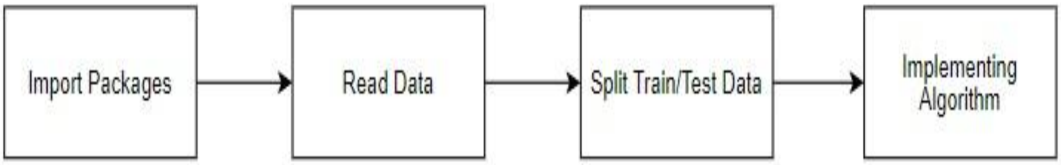


Fig. 6.4 Accuracy comparison – Decision Tree Classifier

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

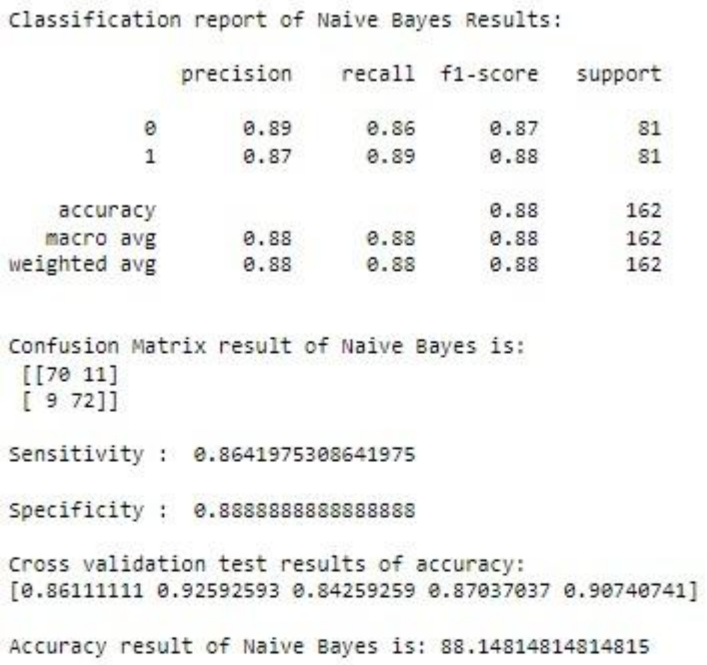
input : data

output : getting accuracy

Naive Bayes algorithm:

- The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically.

- Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location.



```
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

Confusion Matrix result of Naive Bayes is:
 [[70 11]
 [ 9 72]]

Sensitivity :  0.8641975308641975

Specificity :  0.8888888888888888

Cross validation test results of accuracy:
[0.86111111 0.92592593 0.84259259 0.87037037 0.90740741]

Accuracy result of Naive Bayes is: 88.14814814814815
```

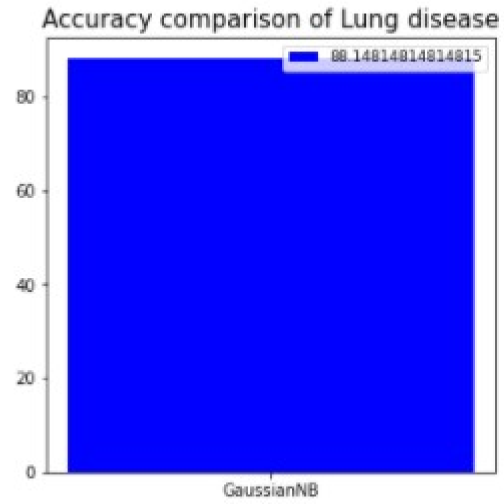
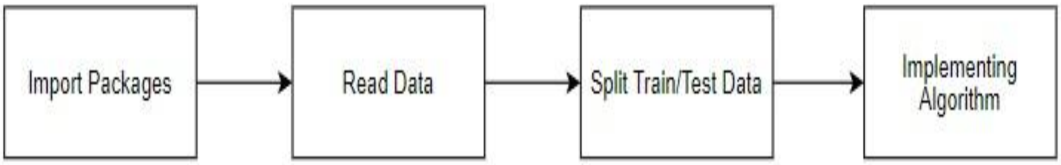


Fig. 6.5 Accuracy comparison - GuassianNB

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data

output : getting accuracy

DEPLOYMENT

Flask (Web FrameWork) :

Flask is a micro web framework written in Python.

It is classified as a micro-framework because it does not require particular tools or libraries.

It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Flask has become popular among Python enthusiasts. As of October 2020, it has second most stars on GitHub among Python web-development frameworks, only slightly behind Django, and was voted the most popular web framework in the Python Developers Survey 2018.

The micro-framework Flask is part of the Pallets Projects, and based on several others of them.

FEATURES:

Flask was designed to be **easy to use and extend**. The idea behind Flask is to build a solid foundation for web applications of different complexity. From then on you are free to **plug in any extensions** you think you need. Also you are free to build your own modules.

Those are supreme Python web frameworks BUT out-of-the-box Flask is pretty impressive too with its:

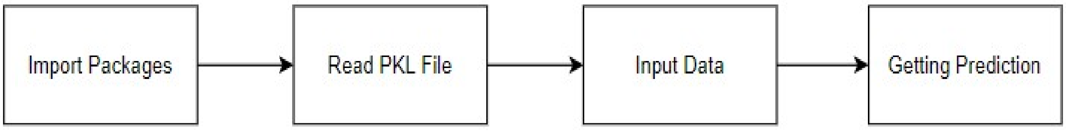
- Built-In Development server and Fast debugger
- integrated support for unit testing
- RESTful request dispatching
- Uses Jinja2 Templating
- support for secure cookies
- Unicode based
- Extensive Documentation

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Advantages of Flask:

- Higher compatibility with latest technologies.
- Technical experimentation.
- Easier to use for simple cases.
- Codebase size is relatively smaller.
- High scalability for simple applications

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input : data values

output : predicting output

HTML Introduction

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text

CSS

CSS stands for Cascading Style Sheets. It is the language for describing the presentation of Web pages, including colours, layout, and fonts, thus making our web pages presentable to the users.CSS is designed to make style sheets for the web. It is independent of HTML and can be used with any XML-based markup language. Now let’s try to break the acronym:

- Cascading: Falling of Styles
- Style: Adding designs/Styling our HTML tags
- Sheets: Writing our style in different documents

CHAPTER 7

CONCLUSION

7.1 RESULT

The entire step of detecting COPD (Chronic Obstructive Pulmonary Disease) as proposed earlier in abstract is carefully handled for better precision and accuracy for easier detection of the disease.

The algorithm is trained with dataset in the backend and same is being binded along the front-end system. So based on the trained dataset the input details of the patients such as gender, age, smoking, yellow finger, anxiety, wheezing, alcohol consumption, coughing, allergy, fatigue, peer pressure, shortness of breath, chest pain, chronic disease and swallowing difficulty are used to predict the presence of the disease in patients.

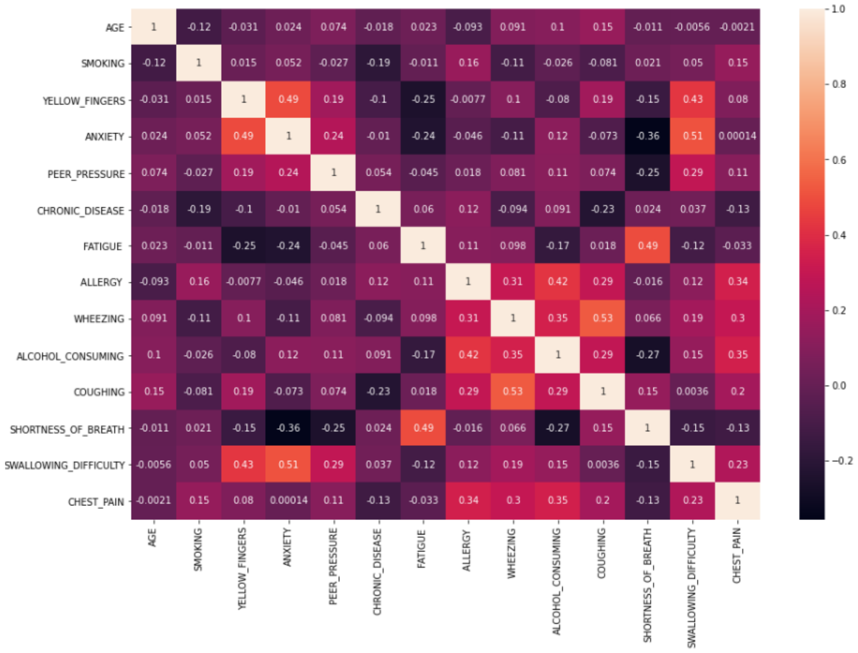
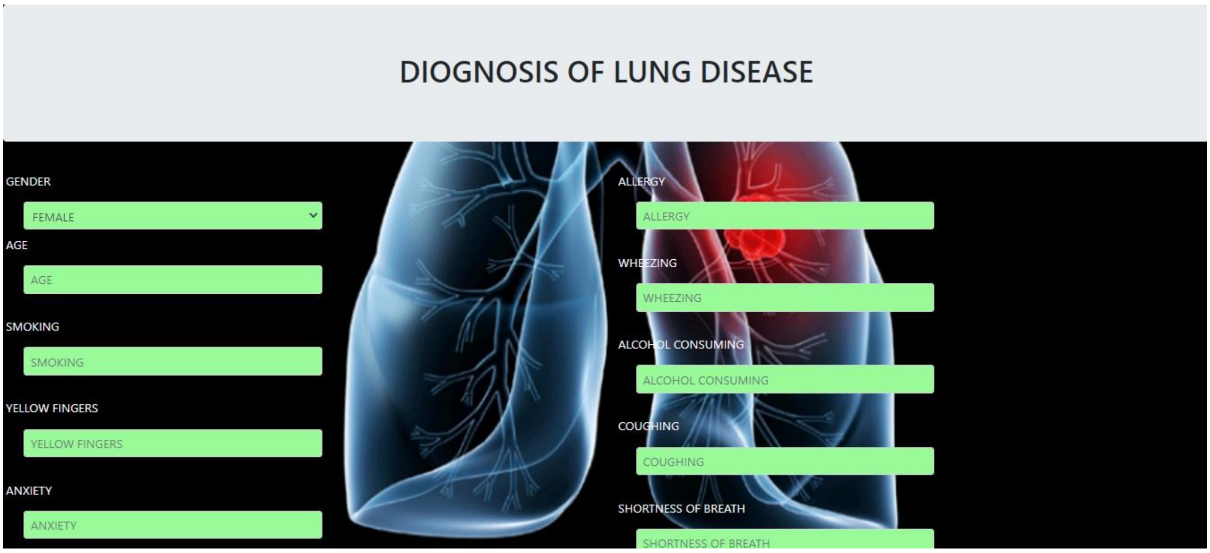


Fig 7.1 Heat map

The above figure shows the heat map which is the two-dimensional representation of the given dataset.



DIOGNOSIS OF LUNG DISEASE

GENDER

FEMALE

AGE

AGE

SMOKING

SMOKING

YELLOW FINGERS

YELLOW FINGERS

ANXIETY

ANXIETY

ALLERGY

ALLERGY

WHEEZING

WHEEZING

ALCOHOL CONSUMING

ALCOHOL CONSUMING

COUGHING

COUGHING

SHORTNESS OF BREATH

SHORTNESS OF BREATH

Fig 7.2 Final Output

The above figure shows us the user interface for the end user, who will use the deployed system to detect the presence of COPD in patients. In this the user needs to enter basic details of the patient such as gender, age, smoking, yellow finger, anxiety, wheezing, alcohol consumption, coughing, allergy, fatigue, peer pressure, shortness of breath, chest pain, chronic disease and swallowing difficulty which is used by the backend to detect the presence of disease in patients.

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be found out. Thus this application helps us to predict Pulmonary Disease in patients.

7.2 FUTURE WORK

- Pulmonary Disease prediction to connect with cloud.
- To optimize the work to implement in an Artificial Intelligence environment.

7.3 REFERENCES

1. M. H. Miglioranza et al., "Pulmonary congestion evaluated by lung ultrasound predicts decompensation in heart failure outpatients," *Int J Cardiol, vol. 240*, pp. 271-278, Aug 1 2017, doi: 10.1016/j.ijcard.2017.02.150.

2. E. Picano and P. A. Pellikka, "Ultrasound of extravascular lung water: a new standard for pulmonary congestion," *Eur Heart J, vol. 37*, no. 27, pp. 2097-104, Jul 14 2016, doi: 10.1093/eurheartj/ehw164

3. E. Picano and P. A. Pellikka, "Ultrasound of extravascular lung water: a new standard for pulmonary congestion," (in eng), *Eur Heart J, Review vol. 37*, no. 27, pp. 2097-104, Jul 14 2016, doi: 10.1093/eurheartj/ehw164.

4. L. Gargani et al., "Persistent pulmonary congestion before discharge predicts rehospitalization in heart failure: a lung ultrasound study," *Cardiovasc Ultrasound, vol. 13*, p. 40, Sep 4 2015, doi: 10.1186/s12947-015-0033-4

5. P. Enghard et al., "Simplified lung ultrasound protocol shows excellent prediction of extravascular lung water in ventilated intensive care patients," *Critical Care, vol. 19*, no. 1, p. 36, 2015.

6. E. Pivetta et al., "Lung Ultrasound-Implemented Diagnosis of Acute Decompensated Heart Failure in the ED: A SIMEU Multicenter Study," *Chest, vol. 148*, no. 1, pp. 202-210, Jul 2015, doi: 10.1378/chest.14-2608.

7. J. L. Martindale, V. E. Noble, and A. Liteplo, "Diagnosing pulmonary edema: lung ultrasound versus chest radiography," *European Journal of Emergency Medicine, vol. 20,* no. 5, pp. 356-360, 2013.

8. G. Volpicelli et al., "International evidence-based recommendations for point-of-care lung ultrasound," *Intensive care medicine, vol. 38,* no. 4, pp. 577-591, 2012.

9. G. Soldati, V. Giunta, S. Sher, F. Melosi, and C. Dini, "“Synthetic” comets: a new look at lung sonography," *Ultrasound in Medicine and Biology, vol. 37,* no. 11, pp. 1762-1770, 2011.

10. 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, 1998.