### LUNG CANCER PREDICTION USING DECISION TREE CLASSIFIER ALGORITHM-SMLT

###### A PROJECT REPORT

***Submitted by***

RAAGUL S [REGISTER NO: 211419104206] RAM KUMAR R [REGISTER NO: 211419104216]

VINOTH KUMAR R [REGISTER NO: 211419104307]

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

COMPUTER SCIENCE AND ENGINEERING



**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

APRIL 2023

### PANIMALAR ENGINEERING COLLEGE

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report **“LUNG CANCER PREDICTION USING DECISON TREE CLASSIFIER ALGORITHM- SMLT”** is the bonafide work of “**RAAGUL S (211419104206), RAM KUMAR R**

**(211419104216), VINOTH KUMAR R (211419104307)”** who carried out the

project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr. L. JABASHEELA, M.E., Ph.D., Mrs. P. VIJAYALAKSHMI, HEAD OF THE DEPARTMENT, ASSISTANT PROFESSOR,** DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE

NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above mentioned students were examined in the End Semester project viva-voice held on

###### INTERNAL EXAMINER EXTERNAL EXAMINER

**DECLARATION BY THE STUDENT**

###### We RAAGUL S (211419104206), RAM KUMAR R (211419104216),

**VINOTH KUMAR R (211419104307), hereby** declare that this project report titled

###### “LUNG CANCER PREDICTION USING DECISION TREE

**CLASSIFIER ALGORITHM-SMLT”,** under the guidance of **Mrs. P. VIJAYALAKSHMI,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

1. RAAGULS
2. RAM KUMAR R
3. VINOTH KUMAR R

**ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D**. and **Dr.SARANYASREE SAKTHI KUMAR**

**B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities toundertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.**

who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA, M.E., Ph.D.,**

for the support extended throughout the project.

We would like to thank my project coordinator **Mr M.MOHAN M.E., (Ph.D).,** and **Project Guide Mrs. P. VIJAYALAKSHMI** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

VINOTH KUAMAR R RAM KUMAR R

RAAGUL S

###### ABSTRACT

Lung cancer is a type of cancer that begins in the lungs. Your lungs are two spongy organs in your chest that take in oxygen when you inhale and release carbon dioxide when you exhale. The aim is to predict machine learning-based techniques for Lung cancer prediction results with the best accuracy. Lung cancer is the leading cause of cancer deaths worldwide. The analysis of dataset by supervised machine learning technique (SMLT) to capture several information like, variable identification, univariate analysis, bi-variate, and multi-variate analysis, missing value treatments and analyze the data validation, data cleaning/preprocess and data visualization will be done on the entire given dataset. To propose a machine learning-based method to accurately predict Lung cancer by prediction results in the form of pulmonary disease classification of best accuracy from comparing supervise classification machine learning algorithms. 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 the best accuracy with precision, Recall and F1 Score.

iii

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | iii |
|  | **LIST OF FIGURES** | vi |
|  | **LIST OF SYMBOLS, ABBREVIATIONS** | vii |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 OVERVIEW | 2 |
|  | 1.2 Problem Definition | 11 |
| **2.** | **LITERATURE SURVEY** | 13 |
| **3.** | **SYSTEM ANALYSIS** |  |
|  | 3.1 Existing System | 21 |
|  | 3.2 Proposed system | 22 |
|  | 3.3 Feasibility Study | 23 |
|  | 3.4 Hardware Environment | 25 |
|  | 3.5 Software Environment | 25 |
| **4.** | **SYSTEM DESIGN** |  |
|  | 4.1 ER diagram | 27 |
|  | 4.2 Data Flow Diagram | 28 |
|  | 4.3 Use Case Diagram | 29 |
|  | 4.4 Class Diagram | 30 |
|  | * 1. Activity Diagram   2. Sequence Diagram | 31  32 |

|  |  |  |
| --- | --- | --- |
| **5.** | **SYSTEM ARCHITECTURE** |  |
|  | 5.1 Module Design Specification | 34 |
|  | 5.2 Algorithms | 58 |
| **6.** | **SYSTEM IMPLEMENTATION** |  |
|  | 6.1 Client-side coding | 67 |
|  | 6.2 Server-side coding | 71 |
| **7.** | **SYSTEM TESTING**  7.1 Performance Metrics | 74 |
|  | 7.2 Test Cases & Reports | 77 |
| **8.** | **CONCLUSION** |  |
|  | 8.1 Results & Discussion | 79 |
|  | 8.2 Conclusion and Future Enhancements | 81 |
|  |  | 82 |
|  | **REFERENCES** |  |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Fig 1.1 | Process of Machine Learning | 9 |
| Fig 1.2 | Process of data flow diagram | 24 |
| Fig 3.1 | ER Diagram | 27 |
| Fig 3.2 | Data Workflow Diagram | 28 |
| Fig 3.3 | Use Case Diagram | 29 |
| Fig 3.4 | Class Diagram | 30 |
| Fig 3.5 | Activity Diagram | 31 |
| Fig 3.6 | Sequence Diagram | 32 |
| Fig 5.1 | System Architecture | 34 |
| Fig 5.2 | HTML Representation | 48 |
| Fig 5.3 | Accuracy of Logistic Regression | 60 |
| Fig 5.4 | Accuracy of Decision tree classifier | 61 |
| Fig 5.5 | Accuracy of Random Forest classifier | 62 |
| Fig 5.6 | Accuracy of Support vector machine | 64 |
| Fig 7.1 | Classification report | 76 |
| Fig 8.1 | Output page 1 | 79 |
| Fig 8.2 | Output page 2 | 80 |

iv

**LIST OF ABBREVATIONS**

|  |  |
| --- | --- |
| **SHORTCUT** | **ABBREVATION** |
| GB | Giga Byte |
| ERD | Entity Relation Diagram |
| API | Application Programming Interface |
| URL | Uniform Resource Locator |
| Str | String |
| Func | Function |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| FP | False Positive |
| FN TP TN | False Negatives True Positive True Negative |

v

**CHAPTER 1**

INTRODUCTION

1.1 OVERVIEW

One of the most lethal types of the disease, lung cancer, is responsible for the passingaway of about one million people every year. The current state of affair in the world of medicine makes it absolutely essential to perform lung nodule identification on chest CT scans. This is due to the fact that lung nodules are becoming increasingly common. As a direct result of this, the deployment of CAD systems is required in order to accomplish the objective of early lung cancer identification. When doing a CT scan, sophisticated X-ray equipment is utilized in order to capture images of the human body from a number of different angles. Following this, the images are fed into a computer, which processes them in such a way as to produce a cross-sectional view of the internal organs and tissues of the body. If lung cancer is detected at an early stage, the American Cancer Society estimates that a patient has a 47 percent chance of surviving the disease. It is quite unlikely that X-ray pictures may accidentally reveal lung cancer in its earlier stages. It is famously difficult to detect lesions that are round and have a diameter of 510 millimeters or less.

**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 proposedit 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 Hammer bacher, 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 science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which help us in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decisions.

###### 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.

###### 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 agent" any system that perceives itsenvironment 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.

Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems. Specific applications of AI include expert systems, natural language processing, and speech recognition and machine vision.

AI applications include advanced web search engines, recommendation systems (used by YouTube, Amazon and Netflix), Understanding human speech (such as Siri or Alexa), self-driving cars (e.g. Tesla), and competing at the highest level in strategic game systems (such as chess and Go), As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the AI effect. For instance, optical character recognition is frequently excluded from things considered to be AI, having become a routine technology.

Artificial intelligence was founded as an academic discipline in 1956, and in the years since has experienced several waves of optimism, followed by disappointment and the loss of funding (known as an "AI winter"), followed by new approaches, success and renewed funding. 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.

The various sub-fields of AI research are centered on goals and the use of particular tools. The traditional goals of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception and the ability to move and manipulate objects. General intelligence (the ability to solve an arbitrary problem) is among the field's long-term goals. To solve these problems, AI researchers use versions of search and mathematical optimization, formal logic, artificial neural networks, and methods based on statistics, probability and economics. AI also draws upon computer science, psychology, linguistics, philosophy, and many other fields.

The field was founded on the assumption that human intelligence "can be so precisely described that a machine can be made to simulate it". This raises philosophical arguments about the mind and the ethics of creating artificial beings endowed with human-like intelligence. These issues have been explored by myth, fiction and philosophy since antiquity. Science fiction and futurology have also suggested that, with its enormous potential and power, AI may become an existential risk to humanity.

As the hype around AI has accelerated, vendors have been scrambling to promote how their products and services use AI. Often what they refer to as AI is simply one component of AI, such as machine learning. AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms. No one programming language is synonymous with AI, but a few, including Python, R and Java, are popular.

In general, AI systems work by ingesting large amounts of labeled training data, analyzing the data for correlations and patterns, and using these patterns to make predictions about future states. In this way, a chatbot that is fed examples of text chats can learn to produce life like exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of examples.

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.

AI is important because it can give enterprises insights into their operations that they may not have been aware of previously and because, in some cases, AI can perform tasks better than humans. Particularly when it comes to repetitive, detail-oriented tasks like analyzing large numbers of legal documents to ensure relevant fields are filled in properly, AI tools often complete jobs quickly and with relatively few errors.

###### 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 access 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. Beyondsemantic NLP, the goal of "narrative" NLP is to embody a fullunderstanding of common-sense reasoning. By 2019, transformer-based deep learning architectures could generate coherent text.

###### 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 correspondinglabelling to learn data must be labelled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm must 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.

Data scientists use many kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y).

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 the

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.

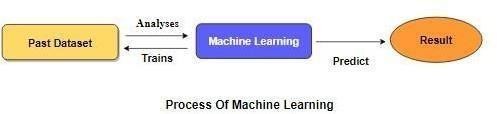


Fig 1.1

**Supervised machine learning is most of the** 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 thatdata. 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 labelled 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. A classification problem is when the output variable is a category, such as “red” or “blue”.

###### Proposed System

In our proposed system we have used machine learning algorithm for implementation. In this model, Decision tree classifier is used for the deployment in order to predict the output. So basically this proposed model can be accessed by anyone at any time in order to find out whether they are affected by lung cancer or not. For the betterment of the usage, we have created a user interface, where the user can update about their health status and can get to know whether they are affected by lung cancer or not. We take 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 obtain results with maximum accuracy. 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 usingmachine learning algorithms are applied on the Training set and based on the test resultaccuracy, Test set prediction is done. We are using more than two machine learning algorithms for getting better results in classification problem. Some irrelevant data should be removed by the preprocessing technique, because initially our dataset will contain some noise like duplicate values, null values and etc. Finally implementing algorithmsand getting better outcome.

###### Advantages

* Machine Learning Technology is implemented.
* Performance metrics will be calculated.
* Deployment is done.

PROBLEM STATEMENT

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. We are using more than two machine learning algorithms for getting better results in classification problem. Some irrelevant data should be removed by the preprocessing technique, because initially our dataset will contain some noise like duplicate values, null values and etc. After the preprocessing we completed the implementation of visualization technique for diagrammatical representation. Finally implementing algorithms and getting better outcome. We take 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 obtain results with maximum accuracy. 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.

**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 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.

1. **Deep Learning-Based Classification of Reduced Lung Ultrasound Data From COVID-19 Patients**

The application of lung ultrasound (LUS)imaging for the diagnosis of lung diseases has recently captured significant interest within the research community. With the ongoing COVID-19 pandemic, many efforts have been made to evaluate LUS data. A four-level scoring system has been introduced to semi quantitatively assess the state of the lung, classifying the patients. Various deep learning (DL) algorithms supported with clinical validations have been proposed to automate the stratification process. However, no work has been done to evaluate the impact on the automated decision by varying pixel resolution and bit depth, leading to the reduction in size of overall data. This article evaluates the performance of DL algorithm over LUS data with varying pixel and gray- level resolution. The algorithm is evaluated over a dataset of 448 LUS videos capturedfrom 34 examinations of 20 patients. All videos are resampled by a factor of 2, 3, and 4of original resolution, and quantized to 128, 64, and 32 levels, followed by score prediction. The results indicate that the automated scoring shows negligible variation in accuracy when it comes to the quantization of intensity levels only. Combined effect of intensity quantization with spatial down-sampling resulted in a prognostic agreement ranging from 73.5% to 82.3%. These results also suggest that such level of prognostic agreement can be achieved over evaluation of data reduced to 32 times of its original size. Thus, laying foundation to efficient processing of data in resource constrained environments.

AUTHOR:

Umair Khan Federico Mento

1. **Diagnosis of Lung Cancer Prediction System Using Data Mining Classification Techniques**

Cancer is the most important cause of death for both men and women. The early detection of cancer can be helpful in curing the disease completely. So the requirement of techniques to detect the occurrence of cancer nodule in early stage is increasing. A disease that is commo nly misdiagnosed is lung cancer. Earlier diagnosis of Lung Cancer saves enormous lives, failing which may lead to other severe problems causing sudden fatal end. Its cure rate and prediction depend mainly on the early detection and diagnosis of the disease. Valuable knowledge can be discovered from application of data mining techniques in healthcare system. In this study, we briefly examine the potential use of classification-based data mining techniques such as Rule based, Decision tree, Naïve Bayes and Artificial Neural Network to massive volume of healthcare data. The healthcare industry collects huge amounts of healthcare data which, unfortunately, are not “mined” to discover hidden information. For data preprocessing and effective decision making One Dependency Augmented Naïve Bayes classifier (ODANB) and naive creedal classifier 2 (NCC2) are used. This is an extension of naïve Bayes to imprecise probabilities that aims at delivering robust classifications also when dealing with small or incomplete data sets. Discovery of hidden patterns and relationships often goes unexploited. Using generic lung cancer symptoms such as age, sex, Wheezing, Shortness of breath, Pain in shoulder, chest, arm, it can predict the likelihood of patients getting a lung cancer disease. Aim of the paper is to propose a model for early detection and correct diagnosis of the disease which will help the doctor in saving the life of the patient.

***Author:*** V.Krishnaiah Dr.G.Narsimha

###### Lung disease detection using deep learning models: a comparative analysis.

Now a days for identifying or predict any diseases on human beings, we should have proper diagnosis for predicting the disease which is present in that human body. In general, for prediction of diseases we try to use either X-Ray, CT or MRI scan techniques for taking decision on that appropriate disease. In general, medical person need complete knowledge on that appropriate domain to find out the abnormality which is present in human beings. As we all know that India tops the world for having more deaths due to lung diseases. After the second highest cause of deaths in India due to heart disease, this ling disease is one which is increasing its rank more and more. In order to reduce that problem early diagnosis and treatment of lung diseases is critical to prevent complications including death. Normally for finding the abnormality present in lung, chest X-ray is playing very important role to detect the complete information about the lungs. In this current article we try to present an effective way for expert diagnosis of lung diseases using deep learning models. It focuses on creating a system for assistance of Radiologists in detection of lung diseases. This will especially benefit ruralareas where radiologists aren’t easily available. We use two models like Vgg16 and Vgg19 for predicting the lung disease from chest X ray images and then tell which model gives high accuracy and performance. We conclude by discussing research obstacles, emerging trends, and possible future directions for improving some more advancement.

Authors:

Dr. Raja Sarath Kumar Boddu

###### Classification and Pathology of Lung Cancer

Significant progress has been made in the understanding of lung cancer biology, due in large part to advancement in the understanding of tumor biology and pathogenesis. Acquisition of key somatic mutations acts as a sentinel event in lung carcinogenesis, essential for tumor cell growth and division.1 Molecular detection of driver mutations in specific histologic types of lung cancer can predict favorable response to targeted therapy. The essence of personalized medicine is to tailor individual lung cancer treatment based on accurate histologic classification and biomarker information. Therefore, characterization of histologic type of lung cancer plays an increasingly pivotal role in the multidisciplinary approach in the diagnosis and management of lung cancer. Recognizing the biological diversity of lung cancer, a comprehensive and accurate tumor classification has been developed, which is important for treatment and prognosis. Pathology of lung cancer has expanded to cover both tissue diagnosis.

Author:

Min Zheng, MD, PhD

###### Lung Cancer Prediction Using Robust Machine Learning and Image Enhancement Methods on Extracted Gray-Level Co-Occurrence Matrix Features

In the present era, cancer is the leading cause of demise in both men and women worldwide, with low survival rates due to inefficient diagnostic techniques. Recently, researchers have been devising methods to improve prediction performance. In medical image processing, image enhancement can further improve prediction performance. This study aimed to improve lung cancer image quality by utilizing and employing various image enhancement methods, such as image adjustment, gamma correction, contrast stretching, thresholding, and histogram equalization methods. We extracted the gray-level co-occurrence matrix (GLCM) features on enhancement images, and applied and optimized vigorous machine learning classification algorithms, such as the decision tree (DT), naïve Bayes, support vector machine (SVM) with Gaussian, radial base function (RBF), and polynomial. Without the image enhancement method, the highest performance was obtained using SVM, polynomial, and RBF, with accuracy of (99.89%). The image enhancement methods, such as image adjustment, contrast stretching at threshold (0.02, 0.98), and gamma correction at gamma value of 0.9, improved the prediction performance of our analysis on 945 images provided by the Lung Cancer Alliance MRI dataset, which yielded 100% accuracy and 1.00 of AUC using SVM, RBF, and polynomial kernels. The results revealed that the proposed methodology can be very helpful to improve the lung cancer prediction for further diagnosis and prognosis by expert radiologists to decrease the mortality rate.

Author:

Lal Hussain Hadeel Alsolai

###### The 2015 World Health Organization Classification of Lung Tumors

The 2015 World Health Organization (WHO) Classification of Tumors of the Lung, Pleura, Thymus and Heart has just been published with numerous important changes from the 2004 WHO classification. The most significant changes in this edition involve.

(1) use of immunohistochemistry throughout the classification, (2) a new emphasis on genetic studies integration of molecular testing to help personalize treatment strategies for advanced lung cancer patients, (3) a new classification for small biopsies and cytology similar to that proposed in the 2011 Association for the Study of Lung Cancer/American Thoracic Society/European Respiratory Society classification,

(4) a completely different approach to lung adenocarcinoma as proposed by the 2011 Association for the Study of Lung Cancer/American Thoracic Society/European Respiratory Society classification.

Author:

William D. Travis MD.Elisabeth Brambilla

**CHAPTER 3**

# SYSTEM ANALYSIS

### Existing System

The application of lung ultrasound (LUS) imaging for the diagnosis of lung diseases has recently captured significant interest within the research community. With the ongoing COVID-19 pandemic, many efforts have been made to evaluate LUS data.A four-level scoring system has been introduced to semi quantitatively assess the state of the lung, classifying the patients. This evaluates the performance using CNN algorithm over LUS data with varying pixel and gray-level resolution by Deep Learning. This algorithm isevaluated over a dataset of 448 LUS videos captured from 34 examinations of 20 patients. All videos are resampled by a factor of 2, 3, and 4 of original resolution, and quantized to 128, 64, and 32 levels, followed by score prediction. Since the outbreak ofCOVID-19, LUS emerged as a promising technology to monitor the condition of the affected patients. In this context, it is important to highlight the fact that LUS cannot diagnose COVID-19. In fact, positive patients may show no signs of alteration at the lung surface. Similarly, post-COVID-19 patients canstill present severe alterations of the lung.

###### Disadvantages

* + - They are classifying lung ultrasound image-based prediction it is not a properway to classifying the disease.
    - They are not implementing any machine learning algorithms.
    - They are not calculating the performance metrics like confusion matrix.

### Proposed System

We take 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 obtain results with maximum accuracy. 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. We are using more than two machine learning algorithms for getting better results in classification problem. Some irrelevant data’s should be removed by the preprocessing technique, because initially our dataset will contains some noise like duplicate values, null values and etc. After the preprocessing, we completed the implementation of visualization technique for diagrammatical representation. Finally implementing algorithms and getting better outcome.

###### Advantages

* Machine Learning Technology is implemented.
* Performance metrics will be calculated.
* Deployment is done.

### 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

* + - This dataset contains 950 records of features extracted from Vehicles, which were then used to find the smog rating from the vehicles.

###### Preprocessing

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

###### Building the classification model

The prediction of CO2 emission rating, 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’tbe 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.



Data Gathering

Data Pre-Processing

Choose model

Train model

Test model

Tune model

Prediction

Fig: 3.1 Process of dataflow diagram

### Functional requirements

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like Sk-learn, pandas, NumPy, matplotlib and seaborn.

### Non-Functional Requirements:

Process of functional steps,

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

### Hardware Environment

Processor : Intel i3 or later

Hard disk : minimum 10 GB

RAM : minimum 4 GB

### Software Environment

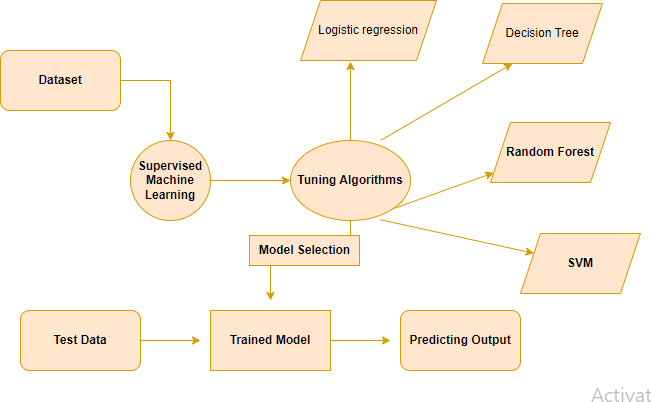
Operating System : Windows 10 or later

Tool : Anaconda with Jupyter Notebook

**CHAPTER 4**

# SYSTEM DESIGN

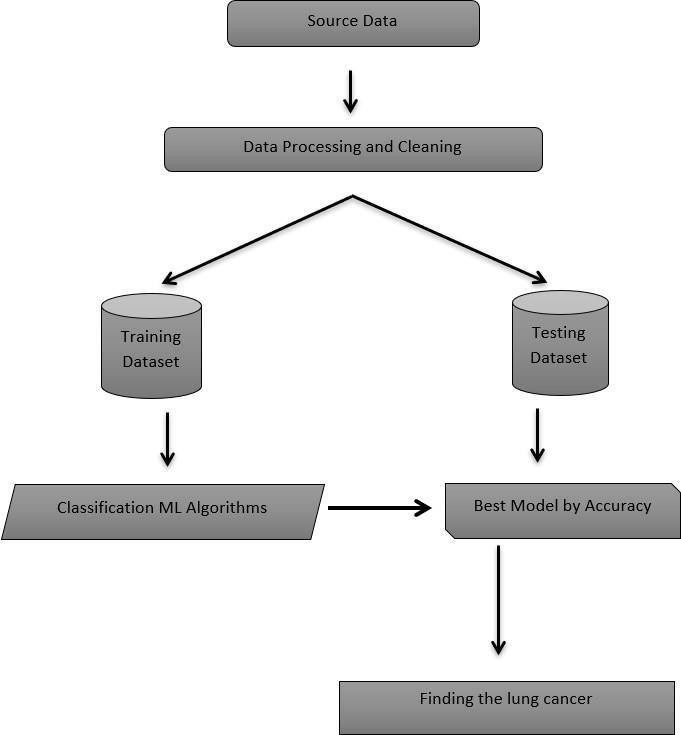
### 4.1. ER Diagram



**Fig 4.1 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 are 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 processor-engineering be needed later.

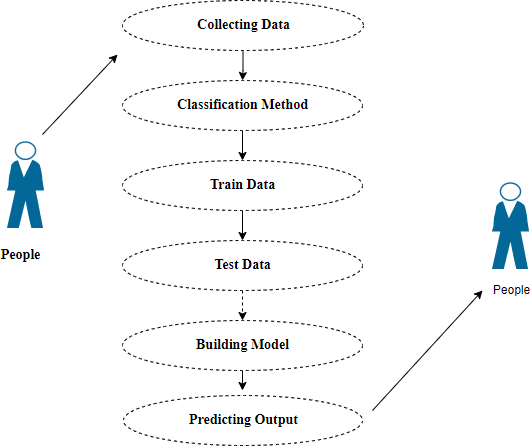
### Data flow diagram



**Fig. 4.2 Data Workflow Diagram**

The above Data flow diagram contains a source data where it is passed to the data processing and cleaning in order to clear the null and unwanted data. Then the processed data is trained and tested with multiple ML algorithms and the algorithm which gives the best accuracy is used for the deployment of the model.

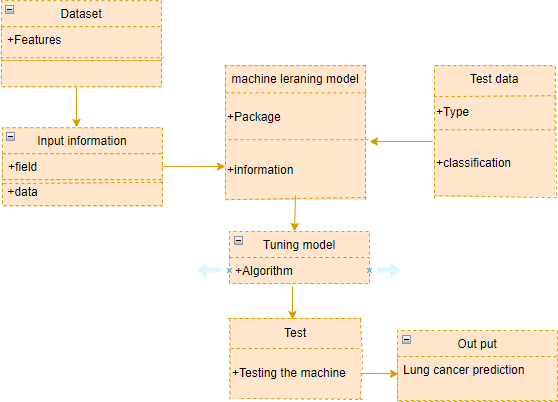
### USE CASE DIAGRAM



**Fig 4.3: Use Case Diagram**

The above use case diagram consists of people who are considered as actors. Here, the data is collected from the people and the collected data is classified according to their needs. The collected data is trained according to the machine language and then tested. The testing process is done in order to receive the best accuracy rate, where the model is built, and the output is predicted.

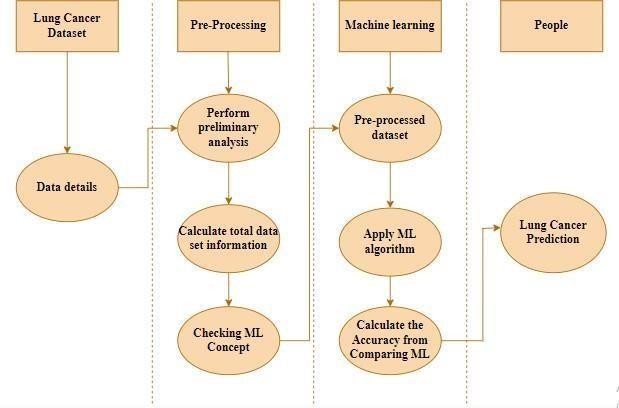
### CLASS DIAGRAM



**Fig 4.4: 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 representsthe whole system. From the above class diagram the inputs from the dataset have been sent to the machine learning algorithms, where the dataset is trained and tested. After the testing is done the algorithm which provides best accuracy has been used to predict the output..

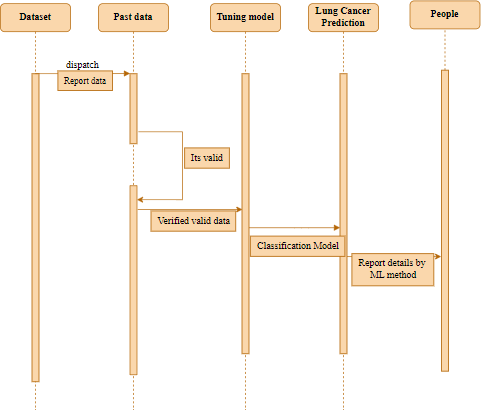
### ACTIVITY DIAGRAM



**Fig 4.5: 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 look like a flow chart, but it is not. It shows different flow like parallel, branched, concurrent and single.

### SEQUENCE DIAGRAM

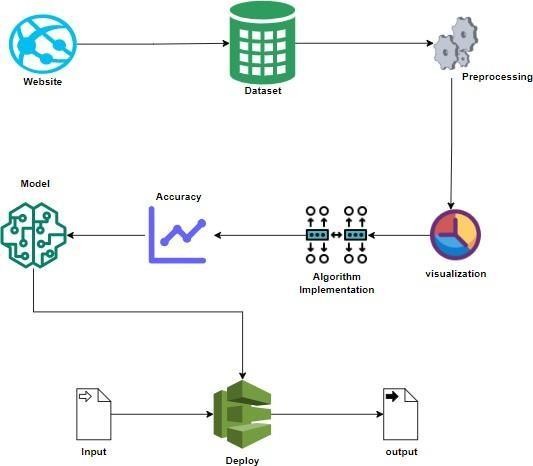


**Fig 4.6 Sequence Diagram**

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. From the above activity diagram, the collected dataset is fed to the pre- processing phase where it goes through the preliminary analysis, and the total dataset information is calculated. To the processed dataset the ML algorithms are applied, and the accuracy is calculated. The algorithm which gives the best accuracy rate is used for predicting lung cancer.

**CHAPTER 5**

**SYSTEM ARCHITECTURE**



### 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.

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 uses 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**](https://www.dataoptimal.com/data-cleaning-with-python-2018/).

It wants to **spend less time cleaning data**, and more time exploring and modeling. Some of these sources are just simple random mistakes. Other ti mes, there can be a deeper reason why data is missing. It’s important to understand these different types.

It’s important to understand these different types. of missing data from a statistics point of view. The type of missing data will influence howto 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 intocode, 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 Univariate, 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.
  + 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

MODULE DIAGRAM



Fig 5.1 Module Diagram I

GIVEN INPUT EXPECTED OUTPUT

input: data

output: removing noisy data.

###### Data 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.

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.

MODULE DIAGRAM

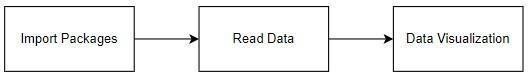


Fig 5.2 Module Diagram II

GIVEN INPUT EXPECTED OUTPUT

input: data

output: visualized data

### Deployment

###### Deploying the model in Django Framework and predicting output

In this module the trained machine learning model is converted into pickle data format file (.pkl file) which is then deployed in our django framework for providing better user interface and predicting the output of how much the given data is emitting Co2.

MODULE DIAGRAM

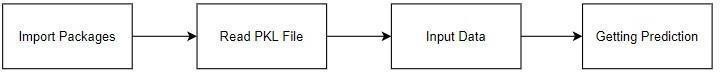


Fig 5.3 Module Diagram III

GIVEN INPUT EXPECTED OUTPUT

input: data values

output: predicting output.

###### 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 componentswhere pre-existing third-party libraries provide common functions.
    - However, Flask supports extensions that can add application features as if theywere implemented in Flask itself.
    - Extensions exist for object-relational mappers, form validation, upload handling,various open authentication technologies and several common framework related tools.
    - Flask was created by Amrin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. According to Ronacher, the idea was originally an April fool’s joke that was popular enough to make into a serious application. The name is a play on the earlier Bottle framework.
    - When Ronacher and Georg Brand created a bulletin board system written in Python, the Pocoo projects Werkzeug and Jinja were developed.
    - In April 2016, the Pocoo team was disbanded, and development of Flask and related libraries passed to the newly formed Pallet’s project.
    - 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 Flask, 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.

Flask **is** based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD license. It was developed at pocoo by Armin Ronacher. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let’s take a closer look into Flask, so-called “micro” framework for Python.

###### 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. Flask is great for all kinds of projects. It's especially good for prototyping. Flask depends on two external libraries: the Jinja2 template engine and the Werkzeug WSGI toolkit.

Still the question remains why use Flask as your web application framework if we have immensely powerful Flask, Pyramid, and don’t forget web mega- framework Turbo gears? Those are supreme Python web frameworks BUT out-of-the-box Flask is impressive too with it’s:

* 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
* Google App Engine Compatibility
* Extensions available to enhance features desired.

Plus, Flask gives you so much more **CONTROL** on the development stage of **your project**. It follows the principles of minimalism and let you decide how you will build your application.

* Flask has a lightweight and modular design, so it easy to transform it to the web framework you need with a few extensions without weighing it down.
* ORM-agnostic: you can plug in your favorite ORM e.g., SQL alchemy.
* Foundation API is nicely shaped and coherent.
* Flask documentation is comprehensive, full of examples and well structured. You can even try out some sample application to really get a feel of Flask.
* It is super easy to deploy Flask in production (Flask is 100% WSGI 1.0 compliant”)
* HTTP request handling functionality
* High Flexibility

The configuration is even more flexible than that of Flask, giving you plenty of solution for every production need.

To sum up, Flask is one of the most polished and feature-rich micro frameworks, available. Still young, Flask has a thriving community, first-class extensions, and an **elegant API**. Flask comes with all the benefits of fast templates, strong WSGI features, **thorough unit testability** at the web application and library level, **extensive documentation**. So next time you are starting a new project where you need some good features and a vast number of extensions, definitely check out Flask.

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Flask framework and is also easier to learn because it has less base code to implement a simple web-Application.

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.

Overview of Python Flask Framework Web apps are developed to generate content based on retrieved data that changes based on a user’s interaction with the site. The server is responsible for querying, retrieving, and updating data. This makes web

applications to be slower and more complicated to deploy than static websites for simple applications.

Flask is an excellent web development framework for REST API creation. It is built ontop of Python which makes it powerful to use all the python features.

Flask is used for the backend, but it makes use of a templating language called Jinja2 which is used to create HTML, XML or other markup formats that are returned to the user via an HTTP request.

Flask is considered to be more popular because it provides many out of box features and reduces time to build complex applications. Flask is a good start if you are getting into web development. Flask is a simple, un-opinionated framework; it doesn't decide what your application should look like developers do.

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, and a wiki or go as big as a web-based calendar application ora 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.
  + Easy to build a quick prototype.
  + Routing URL is easy.
  + Easy to develop and maintain applications.

Framework Flask is a web framework from Python language. Flask provides a library and a collection of codes that can be used to build websites, without the need to do everything from scratch. But Framework Flask still doesn't use the Model View Controller (MVC) method.

Flask-RESTful is an extension for Flask that provides additional support for building REST APIs. You will never be disappointed with the time it takes to develop an API. Flask-Restful is a lightweight abstraction that works with the existing ORM/libraries. Flask-RESTful encourages best practices with minimal setup.

Flask Restful is an extension for Flask that adds support for building REST APIs in Python using Flask as the back end. It encourages best practices and is very easy to set up. Flask restful is very easy to pick up if you're already familiar with Flask.

Flask is a web framework for Python, meaning that it provides functionality for building web applications, including managing HTTP requests and rendering templates and also,we can add to this application to create our API.

###### Start Using an API

1. Most APIs require an API key. ...
2. The easiest way to start using an API is by finding an HTTP client online, like REST- Client, Postman, or Paw.
3. The next best way to pull data from an API is by building a URL from existing API documentation.

The Flask object implements a WSGI application and acts as the central object. It is passed the name of the module or package of the application. Once it is created it

will act as a central registry for the view functions, the URL rules, template configuration and much more.

The name of the package is used to resolve resources from inside the package or the folder the module is contained in depending on if the package parameter resolves to an actual python package (a folder with an init .py file inside) or a standard module (justa

.py file).

For more information about resource loading, see [open resource().](https://flask.palletsprojects.com/en/2.0.x/api/#flask.Flask.open_resource)

Usually you create a [Flask](https://flask.palletsprojects.com/en/2.0.x/api/#flask.Flask) instance in your main module or in the init .py file of your package.

###### Parameters

* **rule** ([str](https://docs.python.org/3/library/stdtypes.html#str)) – The URL rule string.
* **endpoint** (Optional[[str](https://docs.python.org/3/library/stdtypes.html#str)]) – The endpoint name to associate with the rule and view function. Used when routing and building URLs. Defaults to view\_func. name.
* **view\_func** (Optional [Callable]) – The view function to associate with the endpoint name.
* **provide\_automatic\_options** (Optional[bool*]*) – Add the OPTIONS method and respond to OPTIONS requests automatically.
* **options** (Any) – Extra options passed to the [Rule](https://werkzeug.palletsprojects.com/en/2.0.x/routing/#werkzeug.routing.Rule) object.Return type -- None.

After\_Request(f)

Register a function to run after each request to this object.

The function is called with the response object, and must return a response object. This allows the functions to modify or replace the response before it is sent.

If a function raises an exception, any remaining after request functions will not be called. Therefore, this should not be used for actions that must execute, such as to close resources. Use [teardown\_request()](https://flask.palletsprojects.com/en/2.0.x/api/#flask.Flask.teardown_request) for that.

###### Parameters:

**f** (Callable[[[Response](https://flask.palletsprojects.com/en/2.0.x/api/#flask.Response)], [Response](https://flask.palletsprojects.com/en/2.0.x/api/#flask.Response)]) Return type

Callable[[Response], Response] after\_request\_funcs: t.Dict[AppOrBlueprintKey, t.List[AfterRequestCallable]]

A data structure of functions to call at the end of each request, in the format

{scope: [functions]}. The scope key is the name of a blueprint the functions are active for, or None for all requests.

To register a function, use the [after\_request()](https://flask.palletsprojects.com/en/2.0.x/api/#flask.Flask.after_request) decorator.

This data structure is internal. It should not be modified directly, and itsformat may change at any time.

app\_context()

Create an [AppContext](https://flask.palletsprojects.com/en/2.0.x/api/#flask.ctx.AppContext). Use as a with block to push the context, which will make [current\_app](https://flask.palletsprojects.com/en/2.0.x/api/#flask.current_app) point at this application.

An application context is automatically pushed by [RequestContext.push()](https://flask.palletsprojects.com/en/2.0.x/api/#flask.ctx.RequestContext.push) when handling a request, and when running a CLI command. Use this to manually create a context outside of these situations.

With app.app\_context():

Init\_db()

###### 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 itand 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.

Basic Construction of an HTML Page

These tags should be placed underneath each other **at the top of every HTML page** that you create.



Fig 5.4 HTML Representation

<!DOCTYPE html> — This tag **specifies the language** you will write on the page. In this case, the language is HTML 5.

<html> — This tag signals that from here on we are going to write in HTML code.

<head> — This is where all the **metadata for the page** goes — stuff mostly meant for search engines and other computer programs.

<body> — This is where the **content of the page** goes.

Further Tags

Inside the <head> tag, there is one tag that is always included: <title>, but there are others that are just as important:

<title>

This is where we **insert the page name** as it will appear at the top of the browser window or tab.

<meta>

This is where information *about* the document is stored: character encoding, name (page context), description.

###### Head Tag

<head>

<title>My First Webpage</title>

<meta charset="UTF-8">

<meta name="description" content="This field contains information about your page. It is usually around two sentences long.">.

<meta name="author" content="Conor Sheils">

</header>

Adding Content

Next, we will make<body> tag.

The HTML <body> is where we add the content which is designed for viewing by human eyes.

This includes **text, images, tables, forms** and everything else that we see on the internet each day.

Add HTML Headings To Web Page

In HTML, headings are written in the following elements:

* <h1>
* <h2>
* <h3>
* <h4>
* <h5>
* <h6>

As you might have guessed <h1> and <h2> should be used for the most important titles, while the remaining tags should be used for sub-headings and less important text.

**Search engine bots use this order** when deciphering which information is most important on a page.

###### Creating Your Heading

Let’s try it out. On a new line in the HTML editor, type:

<h1> Welcome to My Page </h1>

And hit save. We will save this file as “index.html” in a new folder called “my webpage.”

###### Add Text In HTML

Adding text to our HTML page is simple using an element opened with the tag

<p> which **creates a new paragraph**. We place all of our regular text inside the element <p>.

When we write text in HTML, we also have a number of other elements we can use **to control the text or make it appear in a certain way.**

Add Links In HTML

As you may have noticed, the internet is made up of lots of links.

Almost everything you click on while surfing the web is a link **takes you to another page** within the website you are visiting or to an external site.

Links are included in an attribute opened by the [**<a>**](https://html.com/tags/a/) tag. This element is the first that we’ve met which uses an attribute and so it **looks different to previously mentioned tags.**

<a href= [http://WWW.google.com](http://www.google.com/) >Google</a>

###### Image Tag

In today’s modern digital world, [images](https://html.com/blog/100-legal-sources-free-stock-images/) are everything. The [**<**img**>**](https://html.com/tags/img/) tag has everything you need to display images on your site. Much like the <a> anchor element,

<img> also contains an attribute.

The attribute features information for your computer regarding the source, height, width and alt text of the image

<img src=”yourimage.jpg” alt=”Describe the image” height=“X” width=“X”>

###### 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

CSS Syntax

Selector { Property 1 : value; Property 2 : value; Property 3 : value;

}

For example: h1

{

Color: red;

Text-align: center;

}

#unique

{

color: green;

}

* + Selector: selects the element you want to target
  + Always remains the same whether we apply internal or external styling
  + There are few basic selectors like tags, id’s, and classes
  + All forms this key-value pair
  + Keys: properties(attributes) like color, font-size, background, width, height,etc
  + Value: values associated with these properties

CSS Comment

* + Comments don’t render on the browser
  + Helps to understand our code better and makes it readable.
  + Helps to debug our code
  + Two ways to comment:
    - Single line

CSS How-To

* + There are 3 ways to write CSS in our HTML file.
    - Inline CSS
    - Internal CSS
    - External CSS
  + Priority order
    - Inline > Internal > External

###### Inline CSS

* + Before CSS this was the only way to apply styles
  + Not an efficient way to write as it has a lot of redundancy
  + Self-contained
  + Uniquely applied on each element
  + The idea of separation of concerns was lost
  + Example:

<h3 style = “color:red”> Have a great day </h3>

<p style = “color:green”> I did this, I did that </p>

###### Internal CSS

* + With the help of style tag, we can apply styles within the HTML file.
  + Redundancy is removed.
  + But the idea of separation of concerns still lost.
  + Uniquely applied on a single document
  + Example:

<style> H1{

Color:red;

}

</style>

<h3> Have a great day </h3>

###### External CSS

* + With the help of <link> tag in the head tag, we can apply styles.
  + Reference is added.
  + File saved with .css extension.
  + Redundancy is removed.
  + The idea of separation of concerns is maintained.
  + Uniquely applied to each document
  + Example:

<head>

<link rel= “stylesheet” type= “text/css” href= “name of the CSS file”>

</head> h1{

color:red; //.css file

}

CSS Selectors

* + The selector is used to target elements and apply CSS.
  + Three simple selectors
    - Element Selector
    - Id Selector
    - Class Selector
  + Priority of Selectors

CSS Colors

* + There are different colouring schemes in CSS.
  + **RGB**-This starts with RGB and takes 3 parameters.
  + **HEX**-Hex code starts with # and comprises of 6 numbers which are further divided into 3 sets.
  + **RGBA**-This starts with RGB and takes 4 parameters.

###### CSS Background

* + There are different ways by which CSS can have an effect on HTML elements.
  + Few of them are as follows:
    - Color – used to set the color of the background.
    - Repeat – used to determine if the image must repeat or not and if it is repeating then how it should do that.
    - Image – used to set an image as the background.
    - Position – used to determine the position of the image.
    - Attachment – It basically helps in controlling the mechanism of scrolling.

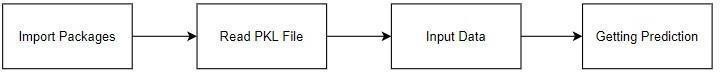
CSS Box Model

* + Every element in CSS can be represented using the BOX model.
  + It allows us to add a border and define space between the content.
  + It helps the developer to develop and manipulate the elements.
  + It consists of 4 edges.
    - Content edge – It comprises of the actual content.
    - Padding edge – It lies in between content and border edge.
    - Border edge – Padding is followed by the border edge.
    - Margin edge – It is an outside border and controls the margin of the element.

###### Deploying the model predicting output

In this module the trained machine learning model is converted into pickle data format file (.pkl file) which is then deployed for providing better user interface and predicting the output of Human Air Pollution.

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input: data values

output: predicting output.

### Algorithms

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 few 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.

The below 4 different algorithms are compared:

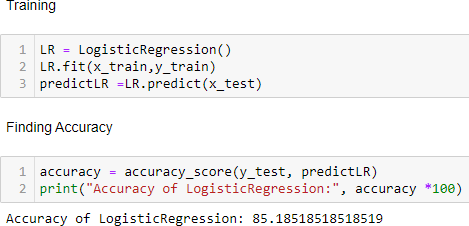
* + - Logistic Regression
    - Decision Tree
    - Random Forest
    - Support Vector Machine

###### 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. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

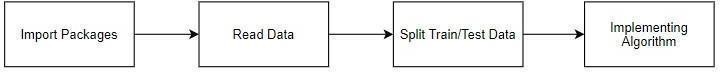
In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic regression Assumptions:

* 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 5.5 Accuracy of Logistic Regression**

###### MODULE DIAGRAM



**GIVEN INPUT EXPECTED OUTPUT**

**input:** data

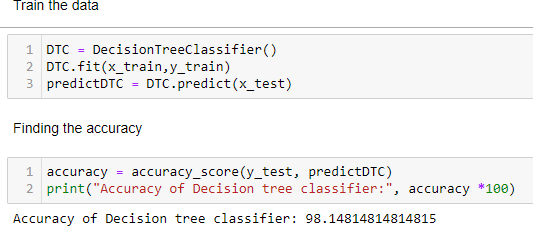
**output:** getting accuracy

###### Decision Tree Classifier

Decision tree learning is a supervised learning approach used in [statistics](https://en.wikipedia.org/wiki/Statistics), [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning.](https://en.wikipedia.org/wiki/Machine_learning) In this formalism, a classification or regression [decisiontree](https://en.wikipedia.org/wiki/Decision_tree) is used as a [predictive model](https://en.wikipedia.org/wiki/Predictive_model) to draw conclusions about a set of observations.

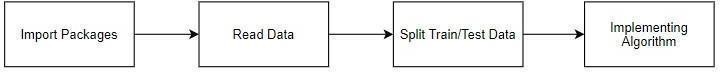
Tree models where the target variable can take a discrete set of values are called [classification trees](https://en.wikipedia.org/wiki/Classification); in these tree structures, [leaves](https://en.wikipedia.org/wiki/Leaf_node) represent class labels and branches represent [conjunctions](https://en.wikipedia.org/wiki/Logical_conjunction) of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically [real numbers](https://en.wikipedia.org/wiki/Real_numbers)) are called [regression trees](https://en.wikipedia.org/wiki/Regression_analysis).

Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.



MODULE DIAGRAM

**Fig 5.6 Accuracy of Decision tree classifier**



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

* Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model.
* The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest".
* The random forest algorithm can be used for both regression and classification tasks.

The [random](https://en.wikipedia.org/wiki/Random_forest) [forest](https://en.wikipedia.org/wiki/Random_forest) algorithm combines multiple algorithm of the same type i.e. multiple decision *trees,* resulting in a *forest of trees,* hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

* 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.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks.

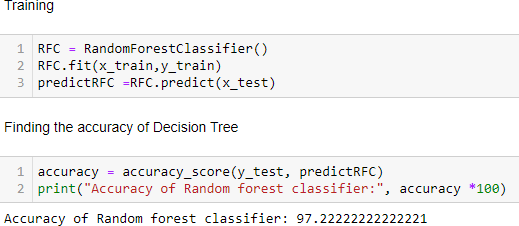
62

forests correct for decision trees’ habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on [ensemble learning.](https://en.wikipedia.org/wiki/Ensemble_learning) Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The [random](https://en.wikipedia.org/wiki/Random_forest) [forest](https://en.wikipedia.org/wiki/Random_forest) algorithm combines multiple algorithm of the same type i.e. multiple decision *trees,* resulting in a *forest of trees,* hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

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

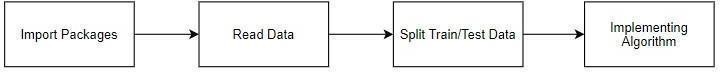
* Pick N random records from the dataset.
* Build a decision tree based on these N records.
* Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.



**Fig 5.7 Accuracy of Random Forest classifier**

MODULE DIAGRAM



GIVEN INPUT EXPECTED OUTPUT

input: data

output: getting accuracy

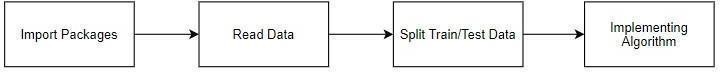
###### Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

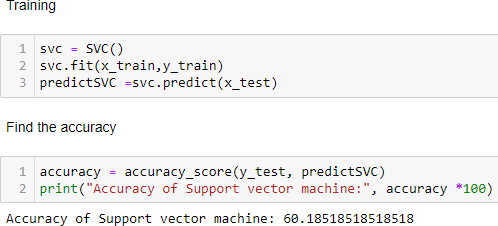
###### MODULE DIAGRAM



**GIVEN INPUT EXPECTED OUTPUT**

input: data

output: getting accuracy



**Fig 5.8 Accuracy of Support vector machine**

**CHAPTER 6**

### SYSTEM IMPLEMENTATION

* 1. **Client-side coding**

###### Django Deploy

from django.shortcuts import render

from django.shortcuts import render, redirect import numpy as np

import joblib

model = joblib.load('C:/Users/SPIRO- PYTHON1/Desktop/ITPML04/kidney\_django/kidney/rf.pkl')

# Create your views here. def home(request):

return render(request, "index.html")

def predict(request):

if request.method == "POST":

int\_features = [x for x in request.POST.values()] int\_features = int\_features[1:] print(int\_features)

final\_features = [np.array(int\_features)] print(final\_features)

prediction = model.predict(final\_features)

print(prediction)

output = prediction[0] if output == 1:

return render(request, 'index.html', {"prediction\_text":"Affected By Chronic Kidney Disease"})

else:

return render(request, 'index.html', {"prediction\_text":"Not Affected By Kidney Disease"})

print(output)

FRONT-END:

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Index</title>

</head>

<style>

:root {

--dark-purple: #2a2a49;

--light-purple: #6564b0;

}

body {

padding-block: 5vh; margin: 0;

font-family: Arial, Helvetica, sans-serif; text-align: center;

line-height: 1.5;

color: rgb(245, 243, 243); background-image: linear-gradient(

115deg,

rgba(58, 58, 158, 0.8),

rgba(136, 136, 206, 0.7)

),

url(static/img/1.jpg);

}

h1 {

margin-top: 0;

}

#description {

font-size: 1.2em; font-style: italic;

}

#survey-form { display: flex;

flex-flow: column; gap: 0.5em;

width: min(40em, 70vw); padding: 1.5em 2em; border-radius: 1.5em; margin: 0 auto;

color: white;

background-color: var(--dark-purple);

### Server-side coding

#### Data preprocessing

*# Import the libraries* **import** pandas **as** pd **import** numpy **as** np

**import** warnings warnings**.**filterwarnings('ignore')

*# Load the Datasets*

data **=** pd**.**read\_csv('Dataset.csv') data**.**head()

data**.**shape data**.**columns

df **=** data**.**dropna() df**.**describe() df**.**isnull() df**.**isnull()**.**sum()

print("Maximum Age :",df["AGE"]**.**max())

print("Minimum Age :",df["AGE"]**.**min())

print("Mean of Age :","%.2f" **%** df["AGE"]**.**mean())

print("Median of Age :","%.2f" **%** df["AGE"]**.**median())

print("Mode of Age :","%.2f" **%** df["AGE"]**.**mode())

df **=** df**.**rename({'ALCOHOL CONSUMING':'Alcohol\_Consuming','CHRONIC DISEASE':'Chronic\_disease','SHORTNESS OF BREATH':'Shortness\_of\_breath',

'SWALLOWING DIFFICULTY':'Swallowing\_difficulty','CHEST PAIN':'chest\_pain','GENDER':'Gender','AGE':'Age','SMOKING':'Smoking',

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

*# Load the Datasets*

data **=** pd**.**read\_csv('Dataset.csv') data**.**head()

data**.**shape data**.**columns

df **=** data**.**dropna() df**.**describe() df**.**isnull() df**.**isnull()**.**sum()

'YELLOW\_FINGERS':'Yellow\_fingers','ANXIETY':'Anxiety','PEER\_PRESSURE':'Peer\_pres sure','FATIGUE ':'Fatigue','ALLERGY ':'Allergy',

'WHEEZING':'Wheezing','COUGHING':'Coughing','LUNG\_CANCER':'Lung\_cancer'},axis**=**1)

In [ ]:

print("Maximum Age :",df["AGE"]**.**max())

print("Minimum Age :",df["AGE"]**.**min())

print("Mean of Age :","%.2f" **%** df["AGE"]**.**mean())

print("Median of Age :","%.2f" **%** df["AGE"]**.**median())

print("Mode of Age :","%.2f" **%** df["AGE"]**.**mode())

72

df**.**groupby(["Age","Alcohol\_Consuming"])**.**groups df**.**groupby(["Age","Anxiety"])**.**groups df['Age']**.**unique()

df**.**head() df['Lung\_cancer'] df**.**tail() df**.**duplicated() sum(df**.**duplicated()) df**=** df**.**drop\_duplicates() sum(df**.**duplicated())

### Implementing Random Forest classifier

*# import the libraries* **import** pandas **as** pd **import** numpy **as** np

**import** warnings warnings**.**filterwarnings('ignore')

*# Load the dataset*

data **=** pd**.**read\_csv('Dataset.csv') data**.**head()

data**.**columns

df **=** data**.**rename({'ALCOHOL CONSUMING':'Alcohol\_Consuming','CHRONIC DISEASE':'Chronic\_disease','SHORTNESS OF BREATH':'Shortness\_of\_breath',

'SWALLOWING DIFFICULTY':'Swallowing\_difficulty','CHEST

PAIN':'chest\_pain','GENDER':'Gender','AGE':'Age','SMOKING':'Smoking',

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

'YELLOW\_FINGERS':'Yellow\_fingers','ANXIETY':'Anxiety','PEER\_PRESSURE':'Peer\_pres sure','FATIGUE ':'Fatigue','ALLERGY ':'Allergy',

'WHEEZING':'Wheezing','COUGHING':'Coughing','LUNG\_CANCER':'Lung\_cancer'},axis**=**1)

In [ ]:

df**.**head() df**.**size

In [ ]:

In [ ]:

In [ ]

df**.**dropna() In [ ]:

df**.**describe()

*# Label encode the dataset*

**from** sklearn.preprocessing **import** LabelEncoder le **=** LabelEncoder()

var **=** ['Lung\_cancer','Gender']

**for** i **in** var:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(int)

*#preprocessing, split test and dataset, split response variable*

X **=** df**.**drop(labels**=**'Lung\_cancer', axis**=**1)

*#Response variable*

In [ ]:

In [ ]:

Y **=** df**.**loc[:,'Lung\_cancer'] In [ ]:

**import** imblearn

**from** imblearn.over\_sampling **import** RandomOverSampler

**from** collections **import** Counter

ros **=**RandomOverSampler(random\_state**=**1) x\_ros,y\_ros**=**ros**.**fit\_resample(X,Y)

print("OUR DATASET COUNT : ", Counter(Y))

print("OVER SAMPLING DATA COUNT : ", Counter(y\_ros)) In [ ]:

**from** sklearn.model\_selection **import** train\_test\_split

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(x\_ros, y\_ros, test\_size**=**0.20, random\_state**=**1, stratify**=**y\_ros)

print("Number of training dataset : ", len(x\_train)) print("Number of test dataset : ", len(x\_test))

print("Total number of dataset : ", len(x\_train)**+**len(x\_test))

Random forest classifier In [ ]:

**from** sklearn.metrics **import** confusion\_matrix, classification\_report, accuracy\_score,plot\_confusion\_matrix

**from** sklearn.ensemble **import** RandomForestClassifier

Training In [ ]:

RFC **=** RandomForestClassifier() RFC **=**RFC**.**predict(x\_test)

Finding the accuracy of Decision Tree In [ ]:

accuracy **=** accuracy\_score(y\_test, predictRFC) print("Accuracy of Random forest classifier:", accuracy **\***100)

Find the classification report In [ ]:

cr **=** classification\_report(y\_test, predictRFC) print("Classification report \n\n:", cr)

cm **=** confusion\_matrix(y\_test, predictRFC) print("Confusion matrix:", cm)

*# Label encode the dataset*

**from** sklearn.preprocessing **import** LabelEncoder le **=** LabelEncoder()

var **=** ['Lung\_cancer','Gender']

**for** i **in** var:

df[i] **=** le**.**fit\_transform(df[i])**.**astype(int)

*#preprocessing, split test and dataset, split response variable*

X **=** df**.**drop(labels**=**'Lung\_cancer', axis**=**1)

*#Response variable*

RFC **=** RandomForestClassifier()

RFC **=**RFC**.**predict(x\_test)

Finding the accuracy of Decision Tree In [ ]:

accuracy **=** accuracy\_score(y\_test, predictRFC)

print("Accuracy of Random forest classifier:", accuracy **\***100)

75

**CHAPTER 7**

# SYSTEM TESTING

#### Performance Metrics

**False Positives (FP):** A person who will pay predicted as defaulter. When actual class is no and predicted class is yes

**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

True Positive Rate(TPR) = TP / (TP + FN) False Positive rate(FPR) = FP / (FP + TN)

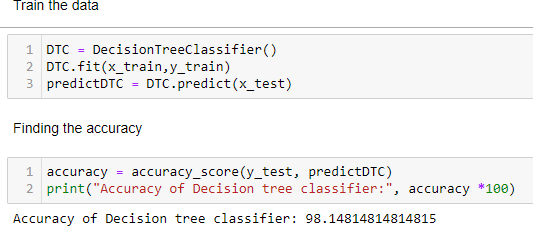
**Accuracy:** The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy 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 labelled 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)

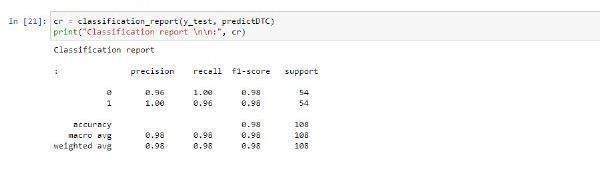
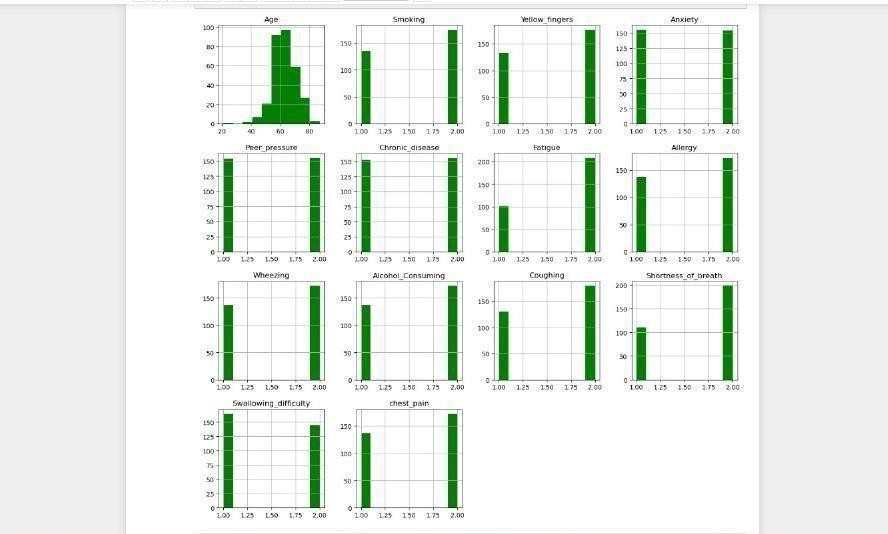


Fig 7.1 Classification report

#### TEST CASES



The test cases that are used in this project are shown in the above figure.

**CHAPTER 8**

# CONCLUSION

### Results & Discussion

The best combination of supervision algorithm has been used and it’s been decided that Decision Tree Classifier has showed the best accuracy rate and this project has used this algorithm for the further process. This model has showed the accuracy rate of 98.148%.

The output for this project can be divided has into two types:

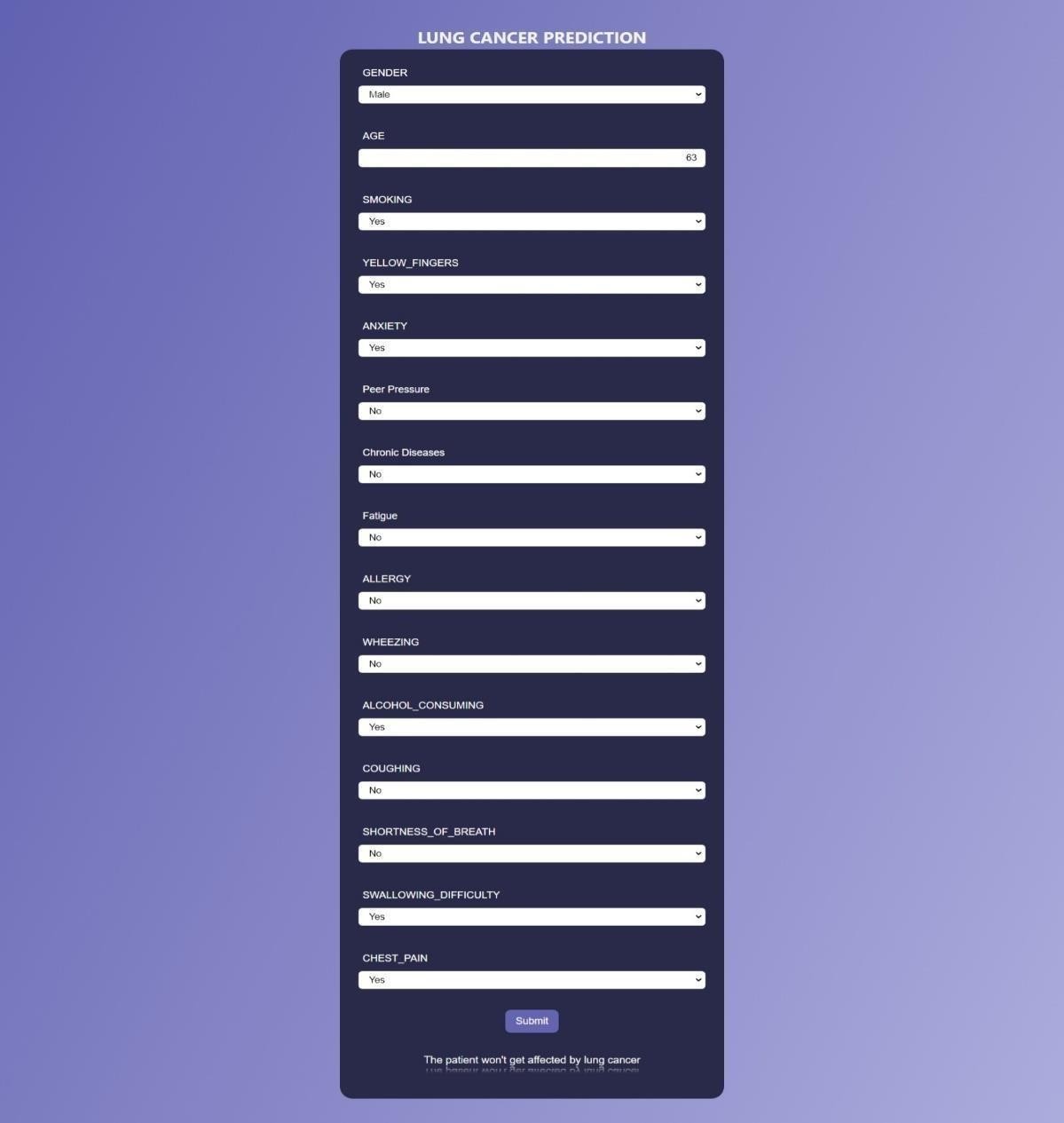
* + 1. Person not - affected with Lung Cancer
    2. Person may develop Lung Cancer Person not - affected with Lung Cancer

Fig 8.1 Output page 1

79

##### Person may develop Lung Cancer



Fig 8.2 Output screen 2

### Conclusion and Future Enhancements

In Our project, the analytical process started from data cleaning by data Pre- processing technique and processing and visualizing our collected data set, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set of higher accuracy score algorithm will be find out. By comparing the results and accuracy score we got from all 4 algorithms, in most cases our best accuracy with sample data set is Decision tree classifier Algorithm. This Algorithm’s higher accuracy score is 98.2%. Which is way higher than CNN Algorithm which was used in the existing system. Decision Tree classifier is used in the application which can help to find the lung cancer prediction.

###### Future Work

* Deploying the project in the cloud.
* To optimize the work to implement in the IOT system.

### 9. REFERENCES

1. P. Joshi, R. K. Tyagi, and K. M. Agarwal, “Technological resources for fighting COVID-19 pandemic health issues,” J. Ind. Integr. Manage.,

vol. 6, no. 2, pp. 271–285, Jun. 2021.

1. Y. Yang et al., “Laboratory diagnosis and monitoring the viral shed- ding of SARS-CoV-2 infection,” Innov., vol. 1, no. 3, Nov. 2020,

Art. no. 100061.

1. S. Woloshin, N. Patel, and A. S. Kesselheim, “False negative tests for SARS-CoV-2 infection—Challenges and implications,” New England

J. Med., vol. 383, no. 6, p. e38, Aug. 2020.

1. S. Salehi, A. Abedi, S. Balakrishnan, and A. Gholamrezanezhad, “Coro- navirus disease 2019 (COVID-19): A systematic review of imaging find- ings in 919 patients,” Amer. J. Roentgenol., vol. 215, no. 1, pp. 87–93,

Jul. 2020.

1. G. Soldati et al., “Is there a role for lung ultrasound during the COVID-19 pandemic?” J. Ultrasound Med., vol. 7 no. 39,pp. 1459–1462, 2020.
2. L. Demi, “Lung ultrasound: The future ahead and the lessons learned from COVID-19,” J. Acoust. Soc. Amer., vol. 148, no. 4, pp. 2146–2150, Oct. 2020.
3. G. Soldati, M. Demi, A. Smargiassi, R. Inchingolo, and L. Demi, “The role of ultrasound lung artifacts in the diagnosis of respiratory diseases,” Expert Rev. Respirat. Med., vol. 13, no. 2, pp. 163–172, 2019.
4. 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. 2097–2104, 2016.
5. G. Soldati, M. Demi, R. Inchingolo, A. Smargiassi, and L. Demi, “On the physical basis of pulmonary sonographic interstitial syndrome,” J. Ultrasound Med., vol. 35, no. 10, pp. 2075–2086, Oct. 2016.
6. R. Copetti, G. Soldati, and P. Copetti, “Chest sonography: A useful

tool to differentiate acute cardiogenic pulmonary edema from acute respiratory distress syndrome,” Cardiovascular Ultrasound, vol. 6, no. 1, pp. 1–10, Apr. 2008.

1. L. Demi, M. Demi, R. Prediletto, and G. Soldati, “Real-time multi- frequency ultrasound imaging for quantitative lung ultrasound–first

clinical results,” J. Acoust. Soc. Amer., vol. 148, no. 2, pp. 998–1006, 2020.

1. F. Mento, G. Soldati, R. Prediletto, M. Demi, and L. Demi, “Quantitative lung ultrasound spectroscopy applied to the diagnosis of pulmonary

fibrosis: The first clinical study,” IEEE Trans. Ultrason., Ferroelectr., Freq. Control, vol. 67, no. 11, pp. 2265–2273, Nov. 2020.

1. F. Mento and L. Demi, “On the influence of imaging parameters on lung ultrasound B-line artifacts, in vitro study,” J. Acoust. Soc. Amer., vol. 148, no. 2, pp. 975–983, Aug. 2020.
2. L. 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 lung ultrasound spectroscopy,” Sci. Rep., vol. 7, no. 1, pp. 1–7, Dec. 2017.

1. K. Mohanty, J. Blackwell, T. Egan, and M. Müller, “Characterization of the lung parenchyma using ultrasound multiple scattering,” Ultrasound.

Med. Biol., vol. 43, no. 5, pp. 993–1003, 2017.