Detection of diseased portion in lung images

A Project

Report

Submitted in partial fulfillment of the requirements for

the award of the degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

T.SAI NANDA KUMAR	180030287
D.SURYA TEJA	180030496
E. KAMAL	180030520

Under the guidance of

Mr. Dr. C. KARTHIKEYAN

Department of CSE

K L E F, Green Fields,

Vaddeswaram – 522 502, Guntur (Dist.), Andhra Pradesh, India.

November 2021

KONERU LAKSHMAIAH EDUCATIONAL FOUNDATION

Green fields, Vaddeswaram.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

This is to certify that the project report entitled **"Detection of diseased portion in lung images"** the Bonafede work carried and done by **T.SAI NANDA KUMAR (180030287), D. SURYA TEJA (180030496) and E. KAMAL (180030520)** in partial fulfillment of the requirements for the award of the degree in "**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**" to **KLEF Deemed to be University** during the academic year 2021-2022. We also declare that this Project report is of our effort and it has not been copied from any other university/department/institute for the award of any degree.

T.SAI NANDA KUMAR

(180030287)

D.SURYA TEJA

(180030496)

E. KAMAL

(180030520)

K L UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to be certified that the project report entitled **"Detection of diseased portion in lung images"** the Bonafede work carried and done by **T.SAI NANDA KUMAR (180030287)** , **D.SURYA TEJA (180030496) and E.KAMAL(180030520)** in partial fulfilment of the requirements for the award of the degree in "**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**" to **KLEF (Deemed to beUniversity)** during the academic year 2021-2022.

PROJECT GUIDE

Mr. Dr. C. KARTHIKEYAN

Assistant Professor, CSE.

HEAD OF THE DEPARTMENT

Dr. HARI KIRAN VEGE

PROJECT INCHARGE

Mrs. VASANTHA BHAVANI

Associate Professor, CSE.

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I greatly indebted to our K L Deemed to be university that has provided a healthy environment to drive me to achieve my ambitions and goals. I would like to express my sincere thanks to my project in charges, for the guidance, support, and assistance they have provided in completing this project.

I express my sincere gratitude and thanks to my project guide **Mr. Dr. C. KARTHIKEYAN** for his novel affiliation of thoughts, support, appreciation, and mental energy which spurred us to wander this project effectively.

I express sincere gratitude to our Coordinator **MR. Dr. C. KARTHIKEYAN** for her leadership and constant motivation provided in successful completion of our academic semester.

I record it as my benefit to profoundly thank our pioneer **Dr. HARI KIRAN VEGE** HOD- CSE for giving us the productive workforce and facilities and faculty to create our thoughts into reality.

Finally, it is satisfied to acknowledge the obligation to all those who given themselves specifically or in a roundabout way to create this project report success.

ABSTRACT

The later advancements of profound learning bolster the distinguishing proof and classification of respiratory organ diseases in restorative pictures. Thus, changed work on the discovery of respiratory organ ailment exploitation deep learning are frequently found inside the writing. This paper presents a overview of profound learning for respiratory organ ailment discovery in therapeutic pictures. There has exclusively been one overview paper revealed within the final 5 a long time concerning profound learning coordinated at respiratory organ infections discovery. However, their study is missing inside the introduction of scientific categorization and examination of the drift of later work. The objectives of this paper are to blessing a scientific categorization of the dynamic profound learning fundamentally based respiratory organ sickness location frameworks, picture the patterns of later work on the space and decide the remaining issues and potential future bearings amid this space. xcviii articles uncovered from 2016 to2020 were thought-about amid this study Machine learning based ailing lung prediction models have been proposed to help clinicians in overseeing accidental or screen recognized uncertain pulmonary knobs. Such frameworks may be able to diminish inconstancy in knob classification, make strides choice making and eventually decrease the number of generous knobs that are unnecessarily taken after or worked-up. In this article, we offer an diagram of the most lung forecast approaches proposed to date and highlight a few of their relative qualities and shortcomings. We talk about a few of the challenges within the improvement and approval of such strategies and layout the way to clinical appropriation. In this project we compare also the multiple models to show which model is better for Detection of diseased portion in lung images.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER-1: INTRODUCTION

1.1 INTRODUCTION

Diseases characterized by the uncontrolled development and unfurl of anomalous cells. In the event that the unfold isn't controlled, it may result in death. carcinoma was the foremost common cancer in worldwide, tributary 2,093,876 of the whole variety of recent cases diagnosed in 2018.

From 2005 to 2015, respiratory organ diseased incidence rates attenuate by two.5% per annum in men and one.2% per annum in girls. Symptoms don't typically occur till the cancer is advanced, and should embrace persistent cough, liquid body substance streaky with blood, chest pain, voice modification, worsening shortness of breath, and perennial respiratory disorder or respiratory illness.

Cigarette smoking is far and away the foremost necessary risk issue for respiratory organ pathologic; eightieth of respiratory organ diseased deaths within the U.S.A. still caused by smoking. Risk will increase with each amount and period of smoking. roll of tobacco and pipe smoking conjointly increase risk. Exposure to noble gas free from soil and building materials is assumed to be the second-leading explanation for carcinoma within the U.S.

Other risk factors embrace activity or environmental exposure to second-hand smoke, amphibole (particularly among smokers), sure metals (chromium, cadmium, arsenic), a few natural chemicals, radiation, contamination, and diesel deplete. A few particular action exposures that increment chance grasp elastic creating, clearing, material, portray, and chimney clearing. Risk is additionally in all probability hyperbolic among individuals with a history of TB. Genetic status (e.g., family history) plays a job within the development of carcinoma, particularly in those that develop the illness at a young age.

We can cure respiratory organ illness, as long as you distinguishing the yearly stage. So here, we tend to use machine learning algorithms to discover the carcinoma. this will be created quicker and a lot of correct. during this study we tend to propose machine learning methods to boost disease characterization.

CHAPTER-2: LITERATURE SURVEY

“In Depth Examination of Lung Disease Forecast Utilizing Machine Learning Algorithms”

The objective of this examination is to examine and predict the Lung Illnesses with help from Machine Learning Calculations. The most common lung infections are Asthma, Sensitivities, Inveterate obstructive pulmonary disease (COPD), bronchitis, emphysema, lung cancer and so on. It is important to anticipate the chances of lung ailments some time recently it happens and by doing that individual can be causes and make crucial strides some time recently it happens. In this paper, we have worked with a collection of information and classified it with various machine learning calculations. We have collected 323 occasions at the side 19 attributes. These data have been collected from patients enduring from numerous lung infections together with other indications. The Lung maladies property contains two sorts of category which are ‘Positive ‘and ‘Negative’. ‘Positive ‘means that the individual contains a lung malady and so forward. The preparing of the dataset has been done with K-Fold Cross Approval Method and specifically, five Machine Learning calculations

“**Deep Learning Approaches in chest X-Rays Analysis”**

Later improvements in Profound Neural Systems lead to major changes in therapeutic imaging. The productivity of dimensionality diminishment calculations like lung division was illustrated within the chest X-Ray picture investigation. As of late analysts pointed at progressing tuberculosis discovery on generally little information sets of less than 103 pictures per course by joining profound learning division and classification methods from We are going advance investigate these strategies in this paper.

In this paper we combine two moderately little datasets containing less than 103 pictures per course for classification (pneumonia and tuberculosis discovery) and division purposes. We chosen 306 cases per “disease” lesson (306 pictures with tuberculosis and 306 pictures with pneumonia) and 306 of solid patients yielding the set of 918 tests from distinctive patients.

All information come from the patient’s schedule clinical care. The volume of the total dataset incorporates thousands of approved optical coherence tomography (OCT) and X-ray pictures however for our examination we needed to keep the dataset modest and equitably conveyed hence as it were 153 pictures were chosen (other 153 pictures come from the tuberculosis dataset) from the assets labeled as solid and 306 as pneumonia - both chosen arbitrarily We made lung illnesses classification pipeline based on exchange learning that was connected to little datasets of lung pictures. We assessed its execution in classification of non-segmented and portioned chest X-Ray pictures. In our best performing system we utilized U-net division organize and InceptionV3 profound show classifier. Our systems were compared with the existing models. We illustrated that models pre-trained by exchange learning approach and basic classifiers such as shallow neural systems can effectively compete with the complex frameworks.

Purpose of review

Reason of survey the application of manufactured insights within the determination of obstructive lung maladies is an energizing wonder. Fake insights calculations work by finding designs in information gotten from symptomatic tests, which can be utilized to anticipate clinical results or to distinguish obstructive phenotypes. The reason of this survey is to depict the most recent trends and to talk about long term potential of counterfeit insights within the determination of obstructive lung diseases.

Recent Findings

Machine learning has been effectively utilized in robotized translation of pneumonic work tests for differential conclusion of obstructive lung maladies. Profound learning models such as convolutional neural arrange are state-of-the craftsmanship for obstructive design acknowledgment in computed tomography. Machine learning has moreover been connected in other demonstrative approaches such as constrained wavering test, breath investigation, lung sound investigation and telemedicine with promising comes about in small-scale studies.

By and large, the application of manufactured insights has created empowering comes about within the conclusion of obstructive lung maladies. In any case, large-scale thinks about are still required to approve current discoveries and to boost its selection by the therapeutic community.

CHAPTER-3: REQUIREMENT & DESIGN ANALYSES

3.1 FUNCTIONAL REQUIREMENTS

A functional requirement is a requirement concerning a result of behavior that shall be provided by a function of the system. A useful necessity could be a prerequisite concerning a result of behavior that shall be given by a work of the framework. FRs are prerequisites that demonstrates a work that a framework or framework component must be able to perform. Functional requirement depicts a usefulness to be made accessible to the clients of the system, characterizing somewhat its behavior as a reply to the jolt that it is subject to. This sort of prerequisite ought to not say any mechanical issue, that is, in a perfect world useful necessity must be autonomous of plan and implementation perspectives.

3.2 NON-FUNCTIONAL REQUIREMENTS

- **Availability:** Describes however seemingly the system is accessible for a user at a given purpose in time

- **Fault Tolerance:** Degree to that a framework, item or portion works as implied in spite of the nearness of hardware or package issues.

- **Operability:** Degree to that a item or framework has traits that construct it basic to work and control.

- **Performance:** Execution relative to the number of assets utilized underneath communicated conditions.
- **Portability:** Degree of effectiveness and power with that a framework, item or portion may be exchanged from one equipment, package or distinctive operational or utilization environment to a diverse.
- **Usability:** Degree to that a item or framework may be utilized by particular clients to accomplish particular objectives with adequacy, productivity, and fulfilment amid a particular setting of utilize.

- **Scalability:** Degree to that a item or framework will successfully and speedily be custom-made for different or advancing equipment, bundle or diverse operational or utilization situations.

3.3 SOFTWARE REQUIREMENT SPECIFICATIONS

- **Software:** Jupyter Notebook (Anaconda)
- **Operating System:** Windows 7 or Above
- **Programming Language:** Python 3.8

3.4 HARDWARE SPECIFICATIONS

- **Computer:** Laptop
- **RAM:** 8GB
- **Processor:** Intel Core i5

3.5 SYSTEM ARCHITECTURE

System architecture could be a conceptual demonstrate that depicts the structure and behavior of different components and subsystems like numerous computer program applications, arrange gadgets, equipment, and indeed other apparatus of a framework. It focuses on the entire system. System architecture incorporates components of both software and equipment and is utilized to empower plan of such a composite framework.

3.5.1 Fundamental Design Concepts

- **Data Abstraction -** It is an act of representing fundamental highlights without counting the foundation points of interest or clarifications.
- **Refinement -** It is the process of elaboration A progression is created by breaking down a macroscopic articulation of work in a stepwise design until programming dialect articulations are come to. Abstraction and are complementary concepts.
- **Modularity -** Software architecture is divided into components called modules.
- **Software Architecture -** It refers to the in general structure of the computer program and the ways in which that structure gives conceptual astuteness for a system.
- **Control Hierarchy -** A program structure that speaks to the organization of a program component and suggests a pecking order of control.
- **Data Structure -** It could be a representation of the coherent relationship among person components of information.
- **Software Procedure -** It centers on the handling of each module exclusively.
- **Information Hiding -** Modules ought to be indicated and planned so that data contained inside a module is blocked off to other modules that have no require for such data.

3.6 MODULE DESCRIPTION

There are few common steps involved in implementing the algorithms as

- **Data Collection**: The data we have Gathered from the meteorological datasets to classify the diseased lungs.
- **Data Preprocessing**: It involves removing the outliers of the data, Eliminating the unnecessary data
- **Data Transformation**: It is the process of converting data from one format or structure into another format or structure.
- **Applying Algorithms**: It involves the application of the data mining algorithms to the given dataset.
- **Predicting the data**: It is the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends.

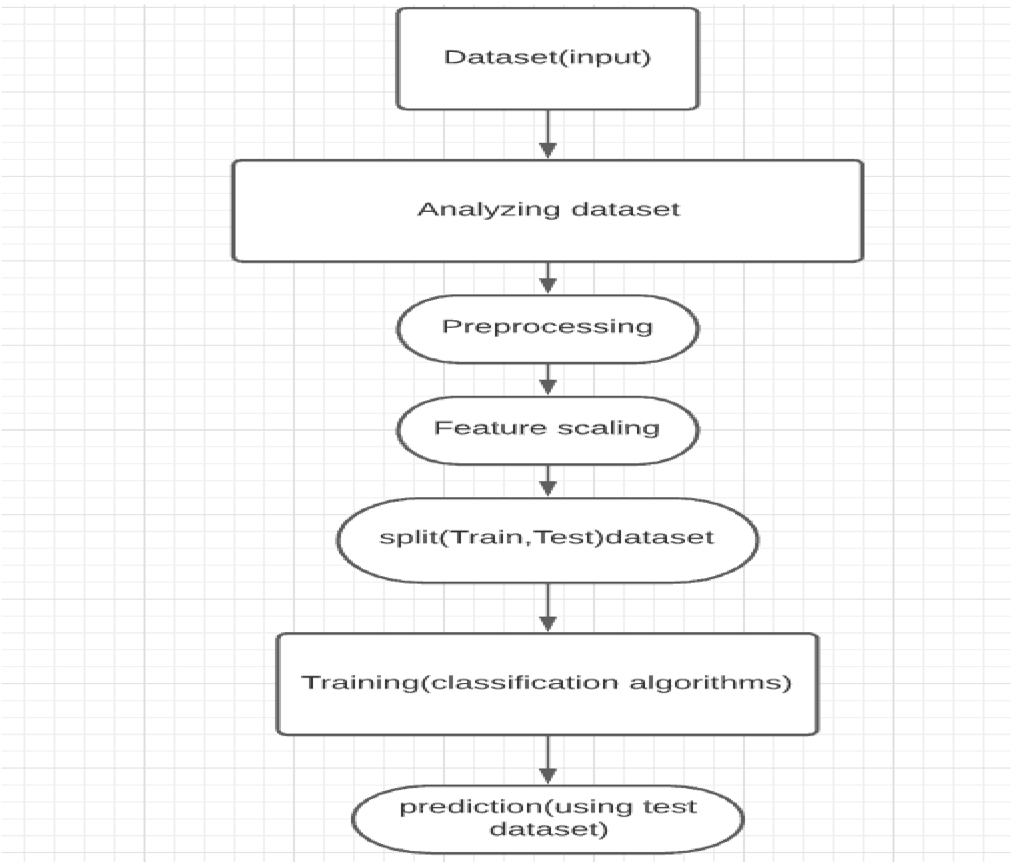


Fig3.6.1: Picture depicting the modules of the project

3.7 METHODOLOGY

1. The first step is taking an input which is collection of data called dataset,
2. Then the second step is analyzing the dataset, we will observe the relationships between features and how they are contributing for the prediction of lung cancer, and which one of them is contributing more for the prediction, then perform reduction on features which are removed based on the analysis.
3. hen the third step is preprocessing, which first, search for the null values present in the dataset, then if there is null values present then perform two operations

a) Either deleting the particular row(which is not suitable method).

b) Perform mean, mode, median based on type of the data which is present in the particular feature of the dataset.

4. Then the fourth step is feature scaling, which is method used to normalize the range of independent variables or features of data.

a) StandardScaler: standardize the data, values are centered around mean.

b) MinMaxScaler: Normalize the data, values are shifted and rescaled so that they end up ranging between 0 and 1.

5. Then the fifth step is for splitting the data for testing and training.
6. Then the sixth step is to train the data using classification algorithms.
7. Then the seventh step is for predicting the values using predictor test data, Then observe the efficiency between predicted data and predicting test data.

CHAPTER-4: THEORETICAL ANALYSIS

4.1 TECHNOLOGY DESCRIPTION

4.1.1 PROBLEM STATEMENT

Lung cancer unwellness caused due to smoking and different risk factors embody activity or environmental exposure to second-hand smoke, amphibole (particularly among smokers), bound metals (chromium, cadmium, arsenic), a few natural chemicals, radiation, contamination, and diesel deplete. carcinoma results in death. So, in our system we have a tendency to square measure planning to analyse carcinoma patient's dataset and use totally different machine learning algorithms and opt for acceptable models.

4.1.2 EXISTING SYSTEM

Lung unwellness was foreseen supported taking Associate in Nursing input dataset of carcinoma patients, for out dataset there was K Nearest Neighbour machine learning demonstrate was utilized that was appearing less precision. The KNN machine learning model can predict the respiratory organ caner unwellness supported dataset was not that correct.

4.1.3 DISADVANTAGES

- The prediction of the KNN model may not be correct due to less accuracy.
- Only one machine learning model is used, don't know how other ML models are performing.
- Accuracy of K Nearest Neighbors is less accurate compared to other ML models.

4.1.4 PROPOSED SYSTEM

For predicting the lung disease, we use the lung disease patient's dataset, then analysis that dataset and apply some preprocessing techniques on that data set, we are going to divide the dataset into train set and test set, then use those datasets for training. We are going to use Machine Learning Algorithms. In ML algorithms we utilize Classification algorithms since our dataset out is yes/no sort. Classification calculations such as Logistic Regression, Support Vector Machine, Decision Tree, Random Forest are utilized on lung malady dataset, we prepare the models 75% Of dataset and after that we test with 25% of dataset information. The trained models will predict whether particular person have lung cancer or not based on the trained model on the lung disease patient's dataset then, we going to calculated the accuracy of every machine learning model used, and we are going to advise that Machine Learning model which is performing best on predicting the lung disease of a patient.

4.1.5 ADVANTAGES

- Reduce the taken a toll of treatment by identifying in early arrange of the Lung Cancer.
- We will know which algorithms performing best in detecting lung cancer.
- The accuracy of the model is increased.
- The risk of patient life will be reduced.
- Prevent the patient from getting silverly affected by Lung cancer.

4.1.6 DATA MINING

The data mining handle includes a number of steps from data collection to visualization to extricate profitable data from expansive information sets. data mining methods are utilized to create portrayals and expectations almost a target data set. Data researchers portray information through their perceptions of designs, affiliations, and relationships. They moreover classify and cluster information through classification and relapse strategies, and distinguish exceptions for utilize cases, like spam location. Information mining more often than not comprises of four fundamental steps: setting targets, information gathering and planning, applying information mining calculations, and assessing comes about.

4.1.7 MACHINE LEARNING

Machine learning (ML) is the think about of computer algorithms that move forward naturally through encounter and by the utilize of information. It is seen as a portion of artificial intelligence. Machine learning algorithms construct a demonstrate based on test information, known as "training data", in arrange to form expectations or choices without being expressly modified to do so. Machine learning algorithms are utilized in a wide assortment of applications, such as in medication, mail sifting, discourse acknowledgment, and computer vision, where it is troublesome or unfeasible to create customary calculations to perform the required assignments.

4.1.7.1 SUPERVISED LEARNING

Supervised learning employments a preparing set to educate models to abdicate the required yield. This training dataset incorporates inputs and rectify outputs, which permit the model to memorize over time. The algorithm measures its precision through the misfortune function, altering until the blunder has been adequately minimized. Supervised learning can be disconnected into two sorts of issues when data mining -classification and regression.

4.1.7.1.1 CLASSIFICATION

Classification issues utilize an calculation to precisely assign test information into particular categories, such as isolating apples from oranges. Or, inside the veritable world, managed learning calculations can be utilized to classify spam in a confined envelope from your inbox. Linear classifiers, support vector machines, decision trees and random forest are all common sorts of classification algorithms. Uses an calculation to precisely allot test information into particular categories. It recognizes particular substances inside the dataset and endeavors to draw a few conclusions on how those substances ought to be labeled or characterized

4.1.7.1.1.1 LOGISTIC REGRESSION

Logistic could be a administered learning classification calculation utilized to anticipate the likelihood of a target variable. foresee the probability of a target variable. The nature of target or subordinate variable is dichotomous, which suggests there would be as it were two conceivable classes. In straightforward words, the subordinate variable is parallel in nature having information coded as either1 (stands for success/yes) or (stands for failure/no). Mathematically, a logistic regression show predicts P(Y=1) as a work of X. It is one of the best ML algorithms that can be utilized for different classification issues such as spam location, Diabetes forecast, cancer discovery etc

1. Binary or Binomial

In such a kind of classification, a subordinate variable will have as it were two conceivable sorts either 1 and 0. For case, these factors may speak to victory or disappointment, yes or no, win or misfortune etc.

2. Multinomial

In such a kind of classification, subordinate variable can have 3 or more conceivable unordered sorts or the sorts having no quantitative importance.

For illustration, these factors may speak to “Type A” or “Type B” or “Type C”.

3. Ordinal

In such a kind of classification, subordinate variable can have 3 or more conceivable requested sorts or the sorts having a quantitative noteworthiness. For case, these factors may speak to “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

4.1.7.1.1.2 SUPPORT VECTOR MACHINE

Support Vector Machine can be a overseen classification calculation where we draw a line between two distinctive categories to recognize between them. SVM to boot known as the support vector organize. The most assignment of the algorithm is to discover the foremost rectify line, or hyperplane, which isolates information into two classes. An SVM is an calculation that gets input information and returns such a isolating line.

4.1.7.1.1.3 DECISION TREE

A decision tree may be a flowchart-like structure in which each inward hub represents a test on a incorporate (e.g., whether a coin flip comes up heads or tails), each leaf hub talks to a course name (choice taken after computing all highlights) and branches represent conjunctions of highlights choice trees are built by implies of an algorithmic approach that recognizes ways to portion a information set based on distinctive conditions. It is one of the preeminent broadly utilized and common-sense methodologies for supervised learning. Decision Trees are a non-parametric administered learning procedure utilized for both classification and backslide assignments. Tree models where the target variable can take a discrete set of values are called classification trees. Choice trees where the target variable can take ceaseless values (frequently honest to goodness numbers) are called relapse trees. Classification And Relapse Tree (CART) is common term for this.at lead to those course names. The ways from root to leaf speak to classification run the show.

4.1.7.1.1.4 RANDOM FOREST

Random forest could be a supervised learning calculation. The "forest" it builds, is an outfit of decision trees, more often than not prepared with the “bagging” strategy. The common thought of the sacking strategy is that a combination of learning models increments the by and large result. Put basically random forest builds multiple decision trees and consolidates them together to induce a more exact and steady prediction.

One huge advantage of random forest is that it can be utilized for both classification and regression issues, which shape the larger part of current machine learning frameworks. Let's see at random forest in classification, since classification is in some cases considered the building piece of machine learning.

4.1.7.1.1.5 K NEAREST NEIGHBHORS

K nearest neighbours (KNN) is a supervised machine learning algorithm. A supervised machine learning algorithm's goal is to learn a function such that f(X) = Y where X is the input, and Y is the output. KNN can be used both for classification as well as regression. In this article, we will only talk about classification. Although for regression, there is just a minute change.

The properties of KNN is that it could be an apathetic learning algorithm and a non-parametric strategy. Sluggish learning means the calculation takes nearly zero time to memorize since it as it were storing the data of the training portion (no learning of a work). The put away information will at that point be utilized for the assessment of a modern query point.

The non-parametric method refers to a method that does not assume any distribution. Therefore, KNN does not have to find any parameter for the distribution.

4.1.7.1.1.6 NAIVE BAYES

Bayes' Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

Naïve Bayes Classifier Algorithm

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

Were,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

4.1.3

- **PYTHON**

  Python is an interpreted, high-level and general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms including structured, object-oriented and functional programming.

- **INSTALLING PYTHON MODULES:**

  As a popular open source development project, Python has an active supporting community of contributors and users that also make their software available for other Python developers to use under open source license terms. pip is the preferred installer program. Starting with Python 3.4, it is included by default with the Python binary installers.

- **WHY MACHINE LEARNING?**

  Machine Learning algorithms are capable of learning the tasks with the available historical data and are able to predict the output using those historical data. It can perform the assignments without being unequivocally modified. In addition to this, we can use deep learning techniques to perform computer vision tasks such as image processing, object detection etc. Since image processing and object detection can be performed efficiently using machine learning and as they are part of our project, we used machine learning techniques.

4.2 INSTALLING STEPS

- **Anaconda**

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

Installing Anaconda

- Go to Anaconda website
- Check for windows Individual Edition
- Click on download button
- Now run the executable file and install it.

Installing required packages

There are some python packages to be installed before executing the project. This can be done using pip (preferred installer program) command. To install a particular python package its respective pip command has to be run in Anaconda command prompt. The python package and its respective pip command were mentioned below.

- Open anaconda prompt.
- Type “pip install numpy” and click enter.
- Type “pip install pandas” and click enter.
- Type “pip install matplotlib” and click enter.
- Type “pip install scikit-learn” and click enter.
- Type “pip install seaborn” and click enter.

4.3 PROCEDURE FOR EXECUTION

1. Open the Anaconda prompt.
2. Change the location of prompt to destination file
3. Now type “jupyter notebook”.
4. Then jupyter notebook will open.
5. Now click on the file need to be executed.
6. Then file will be opened in the jupyter note book.
7. In the Menu bar, click on kernel.
8. After clicking kernel, some options will be displayed, then click on Restart and Run all option.
9. Then that Executable file will be executed

4.4 BLACK BOX TESTING

Black-box testing may be a strategy of program testing that analyzes the usefulness of an application based on the details. It is additionally known as Determinations based testing. Free Testing Group more often than not performs this sort of testing amid the computer program testing life cycle. This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

- **Behavioral Testing Techniques**

  There are different techniques involved in Black Box testing.

  1. Equivalence Class
  2. Boundary Value Analysis
  3. Domain Tests
  4. Orthogonal Arrays
  5. Decision Tables
  6. State Models
  7. Exploratory Testing

8. All-pairs testing

   Black box testing could be a sort of program testing in which the usefulness of the software isn't known. The testing is done without the inside information of the products.

- **Black box testing can be done in following ways**

1. **Syntax Driven Testing:** This sort of testing is connected to frameworks that can be linguistically spoken to by a few dialects. For case- compilers, language that can be represented by setting free language structure. In this, the test cases are created so that each linguistic use run the show is utilized at slightest once.
2. **Equivalence partitioning:** It is regularly seen that numerous sorts of inputs work so also so rather than giving all of them independently we will bunch them together and test as it were one input of each bunch. The thought is to segment the input space of the framework into a number of proportionality classes such that each part of lesson works in a comparative way, i.e., in case a test case in one lesson comes about in a few mistakes, other individuals of course would moreover result into same mistake.
3. **Identification of equivalence class:** Partition any input domain into minimum two sets: valid values and invalid values. For example, if the valid range is 0 to 100 then select one valid input like 49 and one invalid like 104.
4. **Generating test cases:**

   (i) To each valid and invalid class of input assign unique identification number.

   (ii) Write test case covering all valid and invalid test case considering that no two invalid inputs mask each other.

   (iii) To calculate the square root of a number, the equivalence classes will be:

(a) Valid inputs:

- Whole number which is a perfect square- output will be an integer.
- Whole number which is not a perfect square- output will be decimal number.
- Positive decimals

(b) Invalid inputs :

- Negative numbers (integer or decimal).
- Characters other that numbers like “a”,”!”,”;”, etc.

5. Boundary value analysis:

Boundaries are exceptionally great places for blunders to happen. Consequently in case test cases are outlined for boundary values of input space, at that point the effectiveness of testing moves forward and likelihood of finding blunders moreover increment. For illustration – On the off chance that substantial extend is 10 to 100 at that point test for 10,100 too separated from substantial and invalid input.

6. Cause effect Graphing:

This method sets up relationship between logical input called causes with comparing activities called effect. The causes and effects are represented using Boolean graphs. The following steps are followed:

- Identify inputs (causes) and outputs (effect).
- Develop cause effect graph.
- Transform the graph into decision table.
- Convert decision table rules to test cases

7. **Requirement based testing:** It includes validating the requirements given in SRS of software system.
8. **Compatibility testing:** The test case comes about not as it were depend on item but too infrastructure for conveying functionality. When the framework parameters are changed it is still anticipated to work appropriately. A few parameters that for the most part influence compatibility of software is:

- Processor (Pentium 3, Pentium 4) and number of processors.
- Architecture and characteristic of machine (32 bit or 64 bit).
- Back-end components such as database servers.
- Operating System (Windows, Linux, etc.

CHAPTER-5: EXPERIMENTAL INVESTIGATION

5.1 PICTURE DEPICTING

```
import numpy as np

import cv2

import pandas as pd

img=cv2.imread('Covid2.png')

x_size=300

y_size=237

img=cv2.resize(img,(x_size,y_size))

img=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

df=pd.DataFrame()

img2=img.reshape(-1)

df['Original image']=img2

df.head()

df.info()

len(df)
```

5.2 APPLYING FILTERS AND EDGE TECHNIQUES

```
num = 1

    kernels = []

    for theta in range(2):

        theta = theta / 4. * np.pi

        for sigma in (1, 3):

            for lamda in np.arange(0, np.pi, np.pi / 4):

                for gamma in (0.05, 0.5):

                    #           print(theta, sigma, , lamda, frequency)
```

```
                    gabor_label = 'Gabor' + str(num)
#                     print(gabor_label)
                    ksize=5
                    kernel = cv2.getGaborKernel((ksize, ksize), sigma, theta, lamda, gamma,
0,ktype=cv2.CV_32F)
                    kernels.append(kernel)
                    #Now filter image and add values to new column
                    fimg = cv2.filter2D(img2, cv2.CV_8UC3, kernel)
                    filtered_img = fimg.reshape(-1)
                    df[gabor_label] = filtered_img  #Modify this to add new column for each gabor
                    print(gabor_label,': theta=',theta,': sigma=',sigma,':lamda=',lamda,':
gamma=',gamma)
                    num += 1
      ########################################
#Geerate OTHER FEATURES and add them to the data frame
#Feature 3 is canny edge
    edges = cv2.Canny(img, 100,200)   #Image, min and max values
    edges1 = edges.reshape(-1)
    df['Canny Edge'] = edges1 #Add column to original dataframe
    from skimage.filters import roberts, sobel, scharr, prewitt
#Feature 4 is Roberts edge
    edge_roberts = roberts(img)
    edge_roberts1 = edge_roberts.reshape(-1)
    df['Roberts'] = edge_roberts1
#Feature 5 is Sobel
    edge_sobel = sobel(img)
    edge_sobel1 = edge_sobel.reshape(-1)
    df['Sobel'] = edge_sobel1
#Feature 6 is Scharr
    edge_scharr = scharr(img)
    edge_scharr1 = edge_scharr.reshape(-1)
    df['Scharr'] = edge_scharr1
    #Feature 7 is Prewitt
    edge_prewitt = prewitt(img)
```

```
#Feature 8 is Gaussian with sigma=3
    from scipy import ndimage as nd
    gaussian_img = nd.gaussian_filter(img, sigma=3)
    gaussian_img1 = gaussian_img.reshape(-1)
    df['Gaussian s3'] = gaussian_img1
    #Feature 9 is Gaussian with sigma=7
    gaussian_img2 = nd.gaussian_filter(img, sigma=7)
    gaussian_img3 = gaussian_img2.reshape(-1)
    df['Gaussian s7'] = gaussian_img3
    #Feature 10 is Median with sigma=3
    median_img = nd.median_filter(img, size=3)
    median_img1 = median_img.reshape(-1)
    df['Median s3'] = median_img1
```

5.3 creating dataset or dataframe from the image from

```
labeled_img=cv2.imread('Covidlabel1.png')
x_size=300
y_size=237
labeled_img=cv2.resize(labeled_img,(x_size,y_size))
labeled_img=cv2.cvtColor(labeled_img,cv2.COLOR_BGR2GRAY)
labeled_img1=labeled_img.reshape(-1)
df['Labels']=labeled_img1
print(df.head())
```

5.4 Trainning and Testing

```
from sklearn.model_selection import train_test_split
Y=df['Labels'].values
X=df.drop(labels=['Labels'],axis=1)
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.4, random_state=42)
```

5.4.1 Decision Tree Classifier

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics
from sklearn.model_selection import cross_val_score

clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(X_train,Y_train)
result=clf.predict(X)
print(result)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
print("Train Accuracy:", metrics.accuracy_score(Y_train, clf.predict(X_train)))

print("Test Accuracy:",metrics.accuracy_score(Y_test, y_pred))

print(cross_val_score(clf, X, Y, scoring="accuracy", cv = 7))
mean_score = cross_val_score(clf, X, Y, scoring="accuracy", cv = 7).mean()

print('Validation',mean_score)
import cv2
from matplotlib import pyplot as plt
image1 = cv2.imread("Covid2.png")
#Show the image with matplotlib
plt.imshow(image1)
plt.show()
#result=clf.predict(X)
segmented=result.reshape((img.shape))
plt.imshow(segmented,cmap='jet')
```

5.4.2 Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
model=RandomForestClassifier(n_estimators=10,random_state=42)
model.fit(X_train,Y_train)
prediction_test=model.predict(X_test)

print("Train Accuracy:", metrics.accuracy_score(Y_train, model.predict(X_train)))

print("Test Accuracy:",metrics.accuracy_score(Y_test, prediction_test))
print(cross_val_score(model, X, Y, scoring="accuracy", cv = 7))
mean_score = cross_val_score(model, X, Y, scoring="accuracy", cv = 7).mean()
#std_score = cross_val_score(model, X, Y, scoring="accuracy", cv = 7).std()
print('Validation',mean_score)

image2 = cv2.imread("Covid2.png")
#Show the image with matplotlib
plt.imshow(image2)
plt.show()
result1=model.predict(X)
segmented=result1.reshape((img.shape))
from matplotlib import pyplot as plt
plt.imshow(segmented,cmap='jet')
```

5.4.3 Linear Regression Model

```
from sklearn import linear_model
from sklearn.metrics import mean_squared_error
from  sklearn.metrics import r2_score
```

```
from sklearn import metrics

reg = linear_model.LinearRegression()

reg.fit(X_train,Y_train)

prediction_test=reg.predict(X_test)

test_set_rmse = (np.sqrt(mean_squared_error(Y_test, prediction_test)))

test_set_r2 = r2_score(Y_test, prediction_test)

#print("Accuracy=",metrics.accuracy_score(Y_test,prediction_test))

print(test_set_rmse)

print(test_set_r2)

print("Train Accuracy:", r2_score(Y_train, reg.predict(X_train)))

print("Test Accuracy:",test_set_r2)

print( cross_val_score(reg, X_train, Y_train, scoring='r2', cv=5))

mean_score = cross_val_score(reg, X_train, Y_train, scoring='r2', cv=5).mean()

print('cross_Validation',mean_score )

image3 = cv2.imread("Covid2.png")

#Show the image with matplotlib

plt.imshow(image3)

plt.show()

result=reg.predict(X)

segmented=result.reshape((img.shape))

from matplotlib import pyplot as plt

plt.imshow(segmented,cmap='jet')

#image1 = cv2.imread("Covid2.png")
```

5..5 Creating Prediction Model

```
import pickle
filename='Prediction_Model'
pickle.dump(reg,open(filename,'wb'))
```

5.6 Testing a Lung Images with the Best prediction Model

```
import numpy as np
import cv2
import pandas as pd


def  feature_extraction(img):
    df=pd.DataFrame()
    img2=img.reshape(-1)
    df['Original image']=img2
    num = 1
    kernels = []
    for theta in range(2):
        theta = theta / 4. * np.pi
        for sigma in (1, 3):
            for lamda in np.arange(0, np.pi, np.pi / 4):
                for gamma in (0.05, 0.5):
#                print(theta, sigma, , lamda, frequency)


                    gabor_label = 'Gabor' + str(num)
#                     print(gabor_label)
                    ksize=5
                    kernel = cv2.getGaborKernel((ksize, ksize), sigma, theta, lamda, gamma, 0,
ktype=cv2.CV_32F)
                    kernels.append(kernel)
```

```
                    #Now filter image and add values to new column
                  fimg = cv2.filter2D(img2, cv2.CV_8UC3, kernel)
                  filtered_img = fimg.reshape(-1)
                  df[gabor_label] = filtered_img  #Modify this to add new column for each gabor
                  print(gabor_label,': theta=',theta,': sigma=',sigma,':lamda=',lamda,':
gamma=',gamma)
                  num += 1
      ########################################
#Geerate OTHER FEATURES and add them to the data frame
#Feature 3 is canny edge
    edges = cv2.Canny(img, 100,200)   #Image, min and max values
    edges1 = edges.reshape(-1)
    df['Canny Edge'] = edges1 #Add column to original dataframe

    from skimage.filters import roberts, sobel, scharr, prewitt

#Feature 4 is Roberts edge
    edge_roberts = roberts(img)
    edge_roberts1 = edge_roberts.reshape(-1)
    df['Roberts'] = edge_roberts1

#Feature 5 is Sobel
    edge_sobel = sobel(img)
    edge_sobel1 = edge_sobel.reshape(-1)
    df['Sobel'] = edge_sobel1

#Feature 6 is Scharr
    edge_scharr = scharr(img)
    edge_scharr1 = edge_scharr.reshape(-1)
    df['Scharr'] = edge_scharr1

    #Feature 7 is Prewitt
    edge_prewitt = prewitt(img)
    edge_prewitt1 = edge_prewitt.reshape(-1)
```

```
#Feature 8 is Gaussian with sigma=3
    from scipy import ndimage as nd
    gaussian_img = nd.gaussian_filter(img, sigma=3)
    gaussian_img1 = gaussian_img.reshape(-1)
    df['Gaussian s3'] = gaussian_img1
    #Feature 9 is Gaussian with sigma=7
    gaussian_img2 = nd.gaussian_filter(img, sigma=7)
    gaussian_img3 = gaussian_img2.reshape(-1)
    df['Gaussian s7'] = gaussian_img3

    #Feature 10 is Median with sigma=3
    median_img = nd.median_filter(img, size=3)
    median_img1 = median_img.reshape(-1)
    df['Median s3'] = median_img1

    #Feature 11 is Variance with size=3
    #variance_img = nd.generic_filter(img, np.var, size=3)
    #variance_img1 = variance_img.reshape(-1)
    #df['Variance s3'] = variance_img1  #Add column to original dataframe

    return df
```

5.7 COMPARING MODELS

```
import pandas

import matplotlib.pyplot as plt

from sklearn import model_selection

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

from sklearn.naive_bayes import GaussianNB

from sklearn.svm import SVC

models = []

models.append(('LR', LogisticRegression()))

models.append(('LIR', LinearRegression()))

models.append(('KNN', KNeighborsClassifier()))

models.append(('CART', DecisionTreeClassifier()))

models.append(('NB', GaussianNB()))

results = []

names = []

scoring = 'accuracy'

for name, model in models:

  kfold = model_selection.KFold(n_splits=10, random_state=None)

 cv_results = model_selection.cross_val_score(model, X, Y, cv=kfold, scoring=scoring)

  results.append(cv_results)

  names.append(name)
  msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
print(msg)
```

```
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
data_ef={
    'Classification algorithms':['Linear Regression', 'Decision Tree','Random Forest'],
    'Accuracy':[r,q,a],
    'Train Accuracy':[r1,q1,a1],
     'validation':[mean_score,mean_score1,mean_score2]
}
df=pd.DataFrame(data_ef)
df
```

CHAPTER-6 : EXPERIMENTAL & DISCUSSION RESULTS

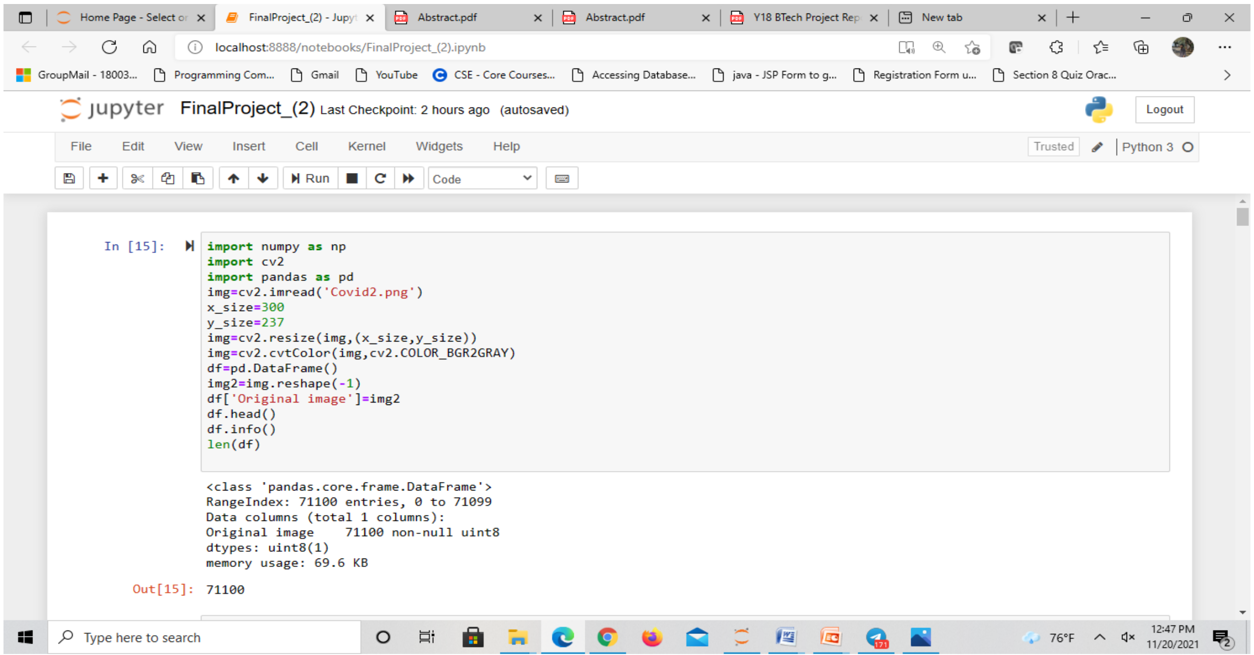


Fig 6.1: Picture depicting execution of project

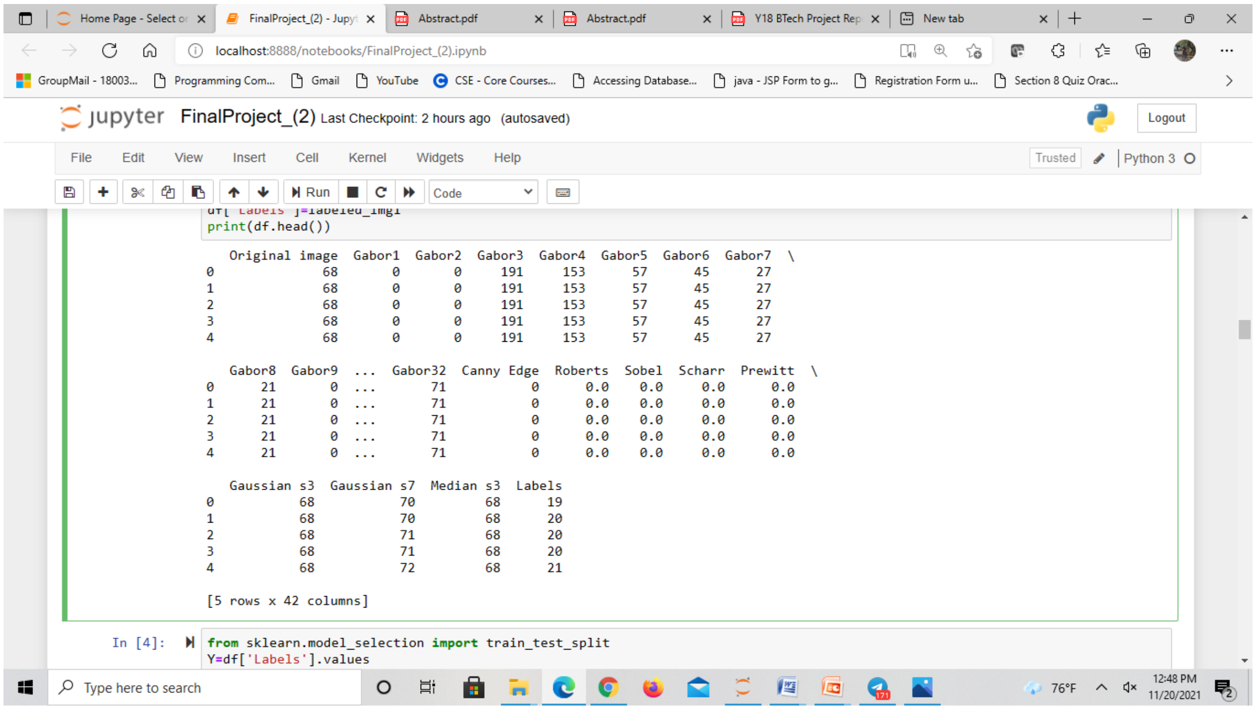


Fig 6.2: Dataset for the model

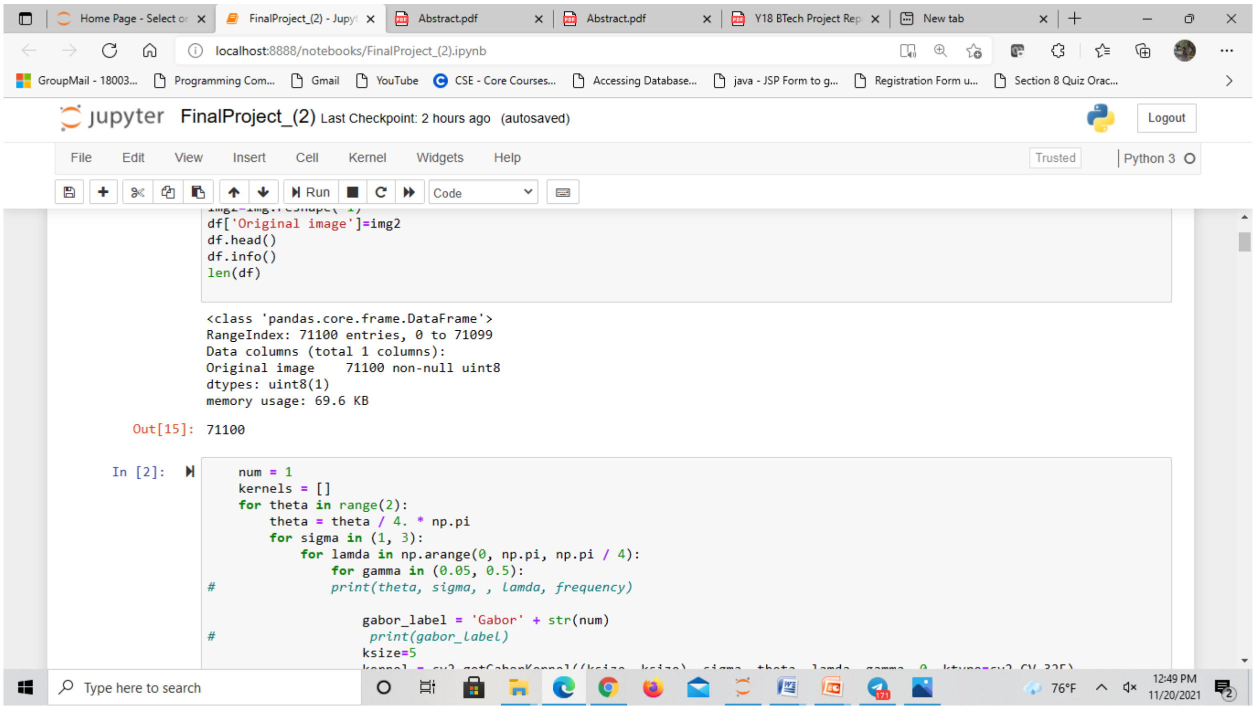


Fig 6.3: Information about dataset

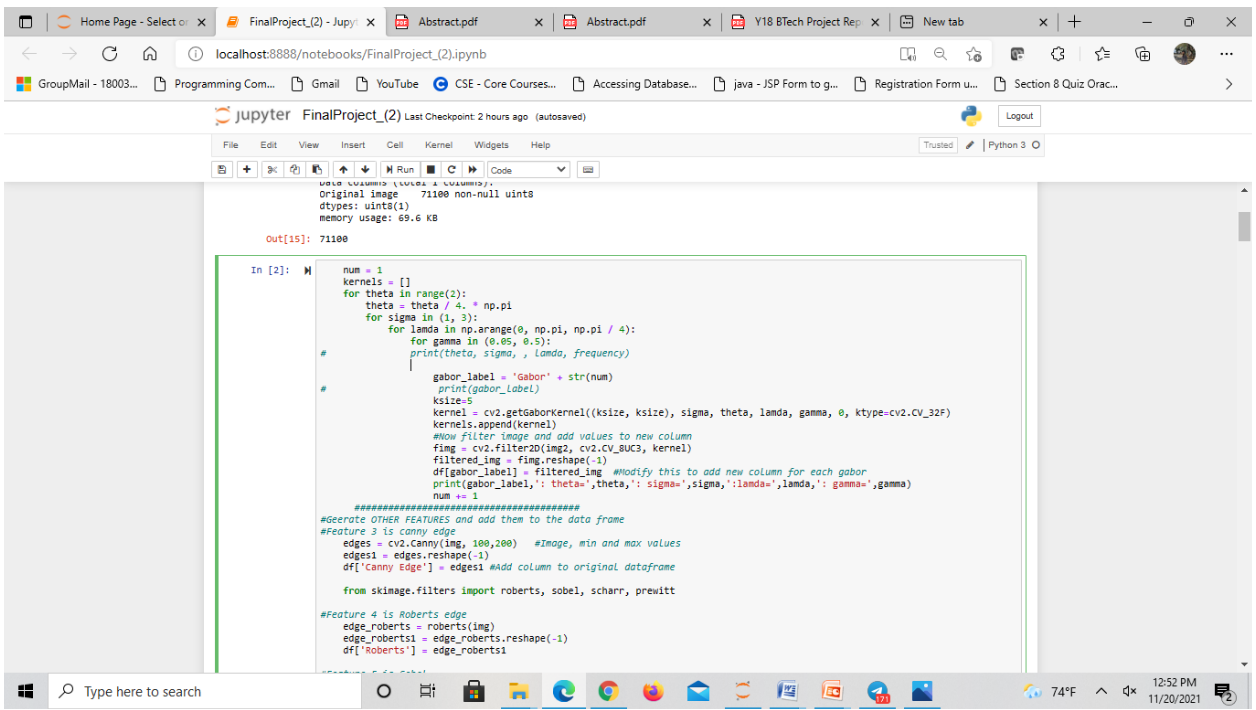
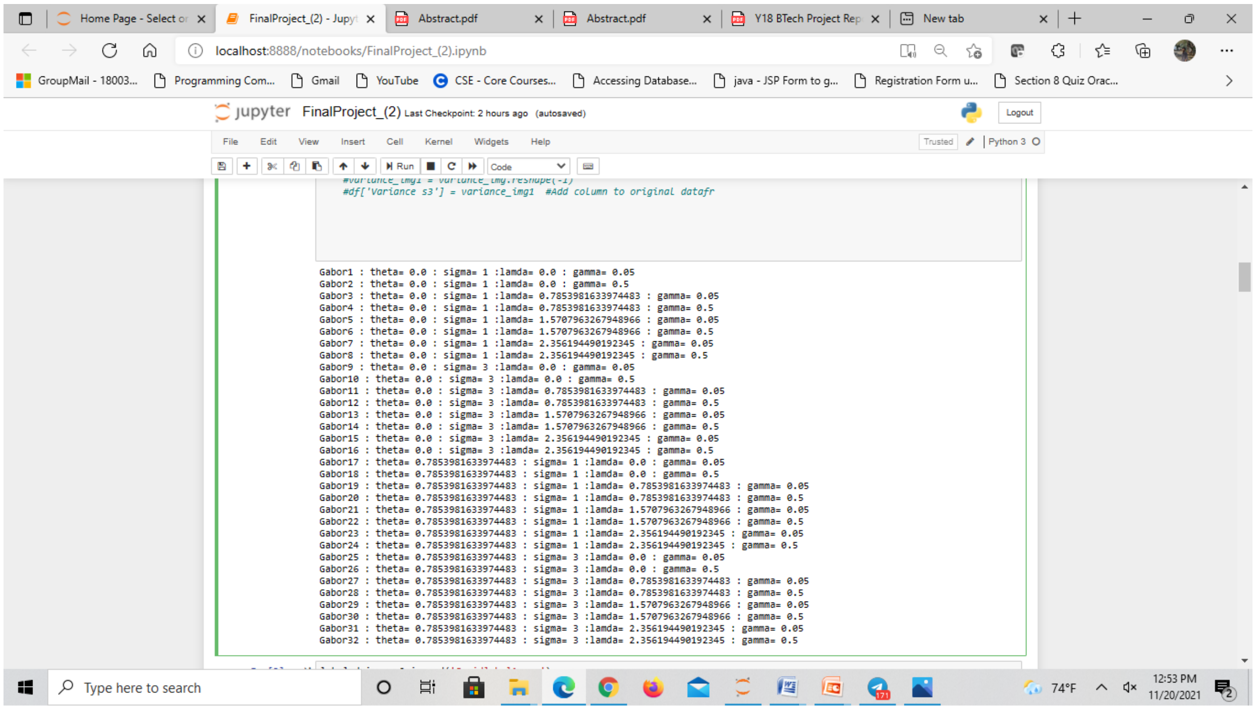


Fig 6.4: Applying filters



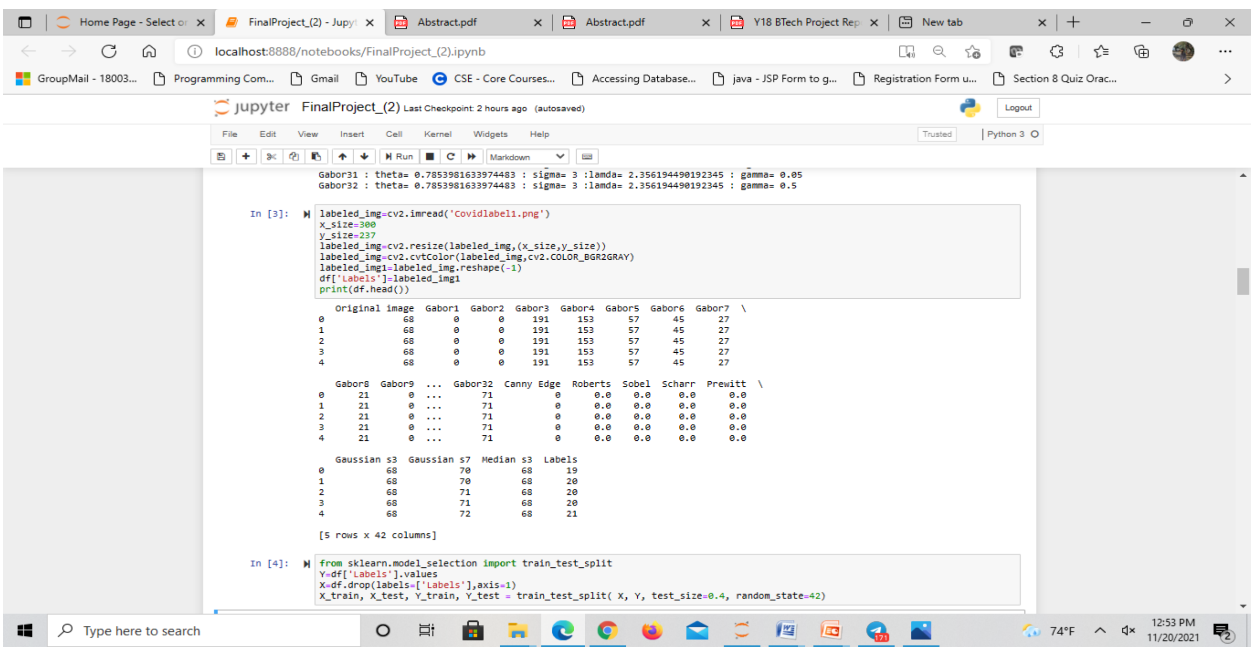


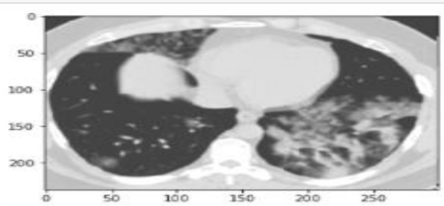
Fig 6.5: Applying filters and edge techniques

```
In [5]:  import pandas as pd
         from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
         from sklearn.model_selection import train_test_split # Import train_test_split function
         from sklearn import metrics
         from sklearn.model_selection import cross_val_score

         clf = DecisionTreeClassifier()

         # Train Decision Tree Classifer
         clf = clf.fit(X_train,Y_train)
         result=clf.predict(X)
         print(result)
         #Predict the response for test dataset
         y_pred = clf.predict(X_test)
         print("Train Accuracy:", metrics.accuracy_score(Y_train, clf.predict(X_train)))

         print("Test Accuracy:",metrics.accuracy_score(Y_test, y_pred))

         print(cross_val_score(clf, X, Y, scoring="accuracy", cv = 7))
         mean_score = cross_val_score(clf, X, Y, scoring="accuracy", cv = 7).mean()

         print('Validation',mean_score)
```

```
[ 28  28  28 ...  41 128 128]
Train Accuracy: 0.9469995311767464
Test Accuracy: 0.150210970464135

C:\Users\vs\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:657: Warning: The least populated cl
only 1 members, which is too few. The minimum number of members in any class cannot be less than n_splits=7.
  % (min_groups, self.n_splits)), Warning)

[0.07689301 0.103732   0.10123723 0.09176841 0.07979091 0.07735681
 0.07091776]

C:\Users\vs\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:657: Warning: The least populated cl
only 1 members, which is too few. The minimum number of members in any class cannot be less than n_splits=7.
  % (min_groups, self.n_splits)), Warning)

Validation 0.08585430031026454
```

Fig 6.6 Implementing Decision Tree Classifier

```
#show the image with matplotlib
plt.imshow(image1)
plt.show()
#result=clf.predict(X)
segmented=result.reshape((img.shape))
plt.imshow(segmented,cmap='jet')
```




```
Out[16]: <matplotlib.image.AxesImage at 0x18e80d09248>
```

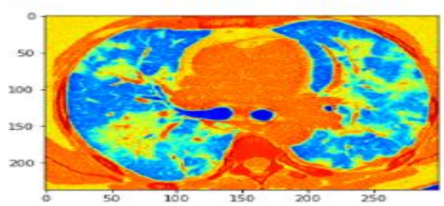
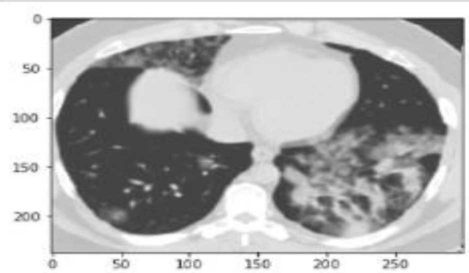


Fig 6.7 output of Decision Tree Classifier

```
8]:  from sklearn.ensemble import RandomForestClassifier
     from sklearn import metrics
     model=RandomForestClassifier(n_estimators=10,random_state=42)
     model.fit(X_train,Y_train)
     prediction_test=model.predict(X_test)

     print("Train Accuracy:", metrics.accuracy_score(Y_train, model.predict(X_train)))

     print("Test Accuracy:",metrics.accuracy_score(Y_test, prediction_test))
     print(cross_val_score(model, X, Y, scoring="accuracy", cv = 7))
     mean_score = cross_val_score(model, X, Y, scoring="accuracy", cv = 7).mean()
     #std_score = cross_val_score(model, X, Y, scoring="accuracy", cv = 7).std()
     print('Validation',mean_score)
```

```
Train Accuracy: 0.9380215658696671
Test Accuracy: 0.1338959212376934

C:\Users\vs\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:657: Warning:
only 1 members, which is too few. The minimum number of members in any class cannot be
  % (min_groups, self.n_splits)), Warning)

[0.07767465 0.09893231 0.09622938 0.08615597 0.07249236 0.074587
 0.06803735]

C:\Users\vs\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:657: Warning:
only 1 members, which is too few. The minimum number of members in any class cannot be
  % (min_groups, self.n_splits)), Warning)

Validation 0.08201557296557334
```

Fig 6.8 Implementing Random Forest Classifier




```
Out[9]: <matplotlib.image.AxesImage at 0x18eb57cf908>
```

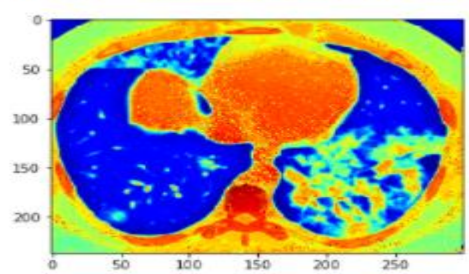
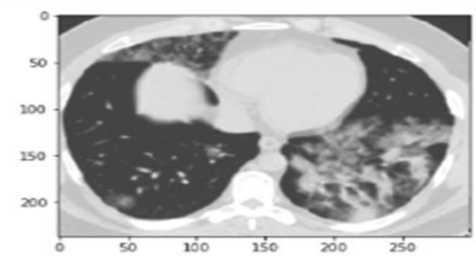


Fig 6.9 output of Random Forest

linear_model

```
In [11]:

from sklearn import linear_model
from sklearn.metrics import mean_squared_error
from  sklearn.metrics import r2_score
from sklearn import metrics
reg = linear_model.LinearRegression()
reg.fit(X_train,Y_train)

prediction_test=reg.predict(X_test)
test_set_rmse = (np.sqrt(mean_squared_error(Y_test, prediction_test)))
test_set_r2 = r2_score(Y_test, prediction_test)
#print("Accuracy=",metrics.accuracy_score(Y_test,prediction_test))
print(test_set_rmse)
print(test_set_r2)
print("Train Accuracy:", r2_score(Y_train, reg.predict(X_train)))

print("Test Accuracy:",test_set_r2)
print( cross_val_score(reg, X_train, Y_train, scoring='r2', cv=5))
mean_score = cross_val_score(reg, X_train, Y_train, scoring='r2', cv=5).mean()

print('cross_Validation',mean_score)

10.66560444600108
0.9615536771627278
Train Accuracy: 0.9618185770399349
Test Accuracy: 0.9615536771627278
[0.96236389 0.96060525 0.96180678 0.96207556 0.96199572]
cross_Validation 0.9617694379474859
```

Fig 6.10 implementing linear Regression Model



```
Out[13]: <matplotlib.image.AxesImage at 0x18ee65eaa08>
```

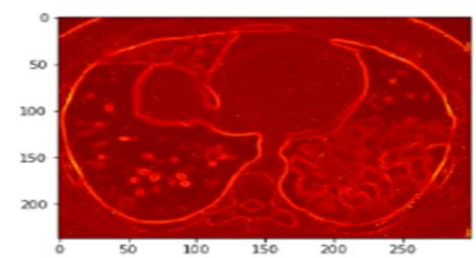


Fig 6.11 output of linear regression model

```
LR: 0.046456 (0.015131)
LDA: 0.054895 (0.020439)
KNN: 0.068664 (0.013576)
CART: 0.074515 (0.017170)
NB: 0.026948 (0.013139)
```

```python
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

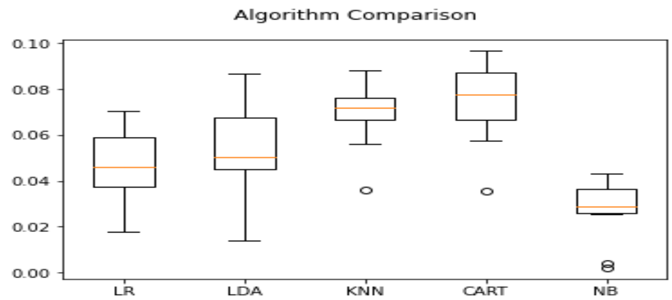


Fig 6.12: boxplot image comparing various models and their accuracy values

	Classification algorithms	Accuracy	Train Accuracy	validation
0	Linear Regression	0.961617	0.961873	0.085767
1	Decision Tree	0.153165	0.947023	0.079212
2	Random Forest	0.130802	0.938514	0.961824

Fig 6.13: data frame to check the best ccuracy value of model.

From above, we can observe different evaluation measures values as follows

CHAPTER-7: CONCLUSION AND FUTER SCOPE

7.1 CONCLUSION

In recent times detecting the Lung Disease Using Software has enhanced in more larger Way Through Machine Learning Algorithms Which gives us the data in more understandable image segmentation format to analyses and predict by the image shown through the algorithms it will be easier to predict the type of disease that is having in the Lungs by showing the type of depicted color images displayed to show the diseased portion of Lungs .and also from comparison of various models for detection the linear regression model got the best accuracy value,

7.2 FUTURE SCOPE

The scope of Machine Learning isn't restricted to the investment sector. Rather, it's increasing across all fields like banking and finance, data technology, media & diversion, gaming, and therefore the automotive business. because the Machine Learning scope is incredibly high, there square measure a number of the square measure as wherever researchers are operating toward revolutionizing the planet for the longer term.

CHAPTER- 8: REFERENCES

[1]. C. Clément-Duchêne, C. Carnin, F. Guillemin and Y. Martineta, "How Accurate Are Physicians in the Prediction of Patient Survival in Advanced Lung Cancer?", *The Oncologist*, vol. 15, pp. 782-789, 2010.

[2] Gunasinghe, A.D., Alphonso, A.C., Thirimanne, H.: Early prediction of lung diseases. Conference Paper, March 2019

[3]. Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012

[4] Kadir, T., Gleeson, F.: Lung cancer prediction using machine learning and advanced imaging techniques (2018). https://doi.org/10.21037/tlcr.2018.05.15

[5]. NCI_SEER_Training_Lung_Cancer_Stats. Introduction to Lung Cancer: SEER training modules—National Cancer Institute; [2015]. Available from: http://training.seer.cancer.gov/lung/.

[6]. NCI_Lung_Cancer_Info. What You Need To Know About Lung Cancer: National Cancer Institute; [2015]. Available from: http://www.cancer.gov/publications/patient-education/wyntk-lung-cancer.

[7] Oku, J.L., et al.: Development and validation of clinical prediction models to risk stratify patients presenting with small pulmonary nodules: a research protocol. Diagn. Prong. Res. 2,22 (2018)

[8] Paul Paisitkriangkrai, Linear Regression and Support Vector Regression modules, The University of Adelaide, October 2012, [online] Available: https://cs.adelaide.edu.au/~chhshen/teaching/ML_SVR.pdf.

[9]. SEER Program. Surveillance, Epidemiology, and End Results (SEER) Program (www.seer.cancer.gov) Research Data (1973–2009), National Cancer Institute, DCCPS, Surveillance Research Program, Surveillance Systems Branch, released April 2012, based on the November 2011 submission.

[10] Sadia, A., et al.: Differential diagnosis of tuberculosis and pneumonia using machine learning. Int. J. Inova. Technol. Explore. Eng. (IJITEE) 8(684), 245–250 (2019). ISSN 2278-3075

[11] Wildman, M.J., et al.: Implications of prognostic pessimism in patients with chronic obstructive pulmonary disease (COPD) or asthma admitted to intensive care in the UK within the COPD and asthma outcome study (CAOS): multicenter observational cohort study (2007). https://doi.org/10.1136/bmj.39371.524271.55

[12] The burden of lung disease. https://www.erswhitebook.org/chapters/the-burden-of-lung-disease/

PaperPass .net Upload Report Account English Log out

The time it takes to process a paper depends on its length. Normally, the plagiarism check report will be completed within an hour.

	Title	State	Similarity	Report	Submit Date	
	Detection of diseased portion in lung images	Completed	13%	View Report	2021-11-22 06:53	

delete

Warning: The system only keeps the report within 100 days. Please download your report as soon as possible.

Basic Information

Title	Detection of diseased portion in lung images
Time	2021-11-22 06:53:41 (+00:00 UTC)
Words	5865

Overall Similarity

Overall Similarity		13%
├ Similarity degrees in local database		7%
└ Similarity degrees in internet database		8%

Similarity Calculation Formula

(Sentence1 similarity + Sentence2 similarity........ / n)

Sentence similarity scale from 0.0~1.0   Green sentence reprensent similarity-free

Sentences similarity distribution

≥70% 40-70% <40%

Unqualificted sentences dotted distribution map