**The Project Report on**

# Multi disease prediction uisng machiline leanring and deep learning

**Submitted To**

**Acharya Nagarjuna University**

**A project report submitted in the partial fulfillment of the requirements for the**

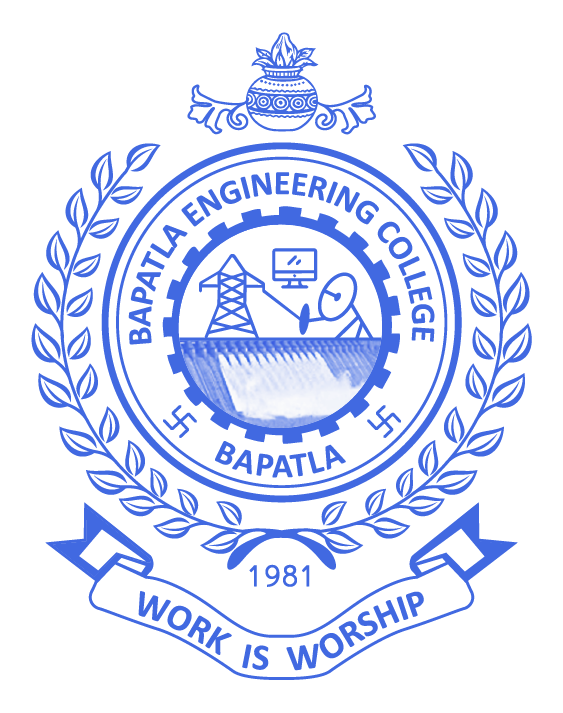
**Award of the Degree of**

**MASTER OF COMPUTER APPLICATIONS**

**Submitted By**

**ARUNURU NAVEEN**

**Reg.No: L20MC23014**

****

**Under The Esteemed Guidance Of**

**Sri N. KIRAN KUMAR M. Tech**

**Assistant Professor**

**DEPARTMENT OF MCA**

**BAPATLA ENGINEERING COLLEGE**

**(Approved by A.I.C.T.E)**

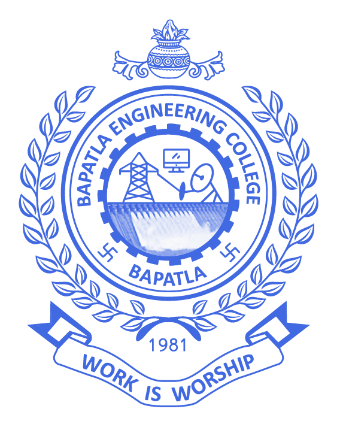
**(Affiliated to ACHARYA NAGARJUNA UNIVERSITY)**

**2019-2021**

**DEPARTMENT OF MCA**

**BAPATLA ENGINEERING COLLEGE**

**BAPATLA-522101**

****

**CERTIFICATE**

This is to certify that this project work entitled “**Secure Cloud Storage based on RLWE Problem”** is the bonafide work carried out by **ARUNURU NAVEEN**, **Reg.No: L20MC23014** submitted in Partial fulfillment of the requirement for the Award of Degree of “**Master of** **Computer Applications**”, during the academic year 2019-2021.

The results submitted in this project have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other university for the award of the any other degree/diploma.

**Project Guide: Head of the Department:**

**Sri N. Kiran Kumar M. Tech Sri K.N. Prasad M. Tech**

**Assistant Professor Associate Professor & Head**

**Dept. of MCA Dept. of MCA**

**Bapatla Engineering College Bapatla Engineering College**

**Bapatla. Bapatla.**

**Signature of External Examiner**

# ACKNOWLEDGEMENT

I sincerely thank the following distinguished personalities who have given their advice and support for successful completion of this work.

I would like to thank **Dr. V. Damodara Naidu**, Principal, Bapatla Engineering College, Bapatla for providing the online resources and allowing me to carry out this work.

I extended my sincere thanks to **Sri K.N. Prasad**, Head of the Department for extending their cooperation and providing the required resources.

I am deeply indebted to most respect and my project guide **Sri N. Kiran Kumar**, Assistant Professor Dept. of MCA for his valuable and inspiring guidance, comments, suggestions and encouragements.

I extended my sincere thanks to all other teaching and non-teaching staff of the department who helped directly or indirectly for their cooperation and encouragement.

**ARUNURU NAVEEN**

**(L20MC23014)**

**DECLARATION**

This is to declare that the project **“Secure Cloud Storage based on RLWE Problem”** at Bapatla Engineering College has been presented by me during the academic year **2019-2021** in partial fulfillment of the requirements for the **“Master of Computer Application”**.

I also declare that this project is the result of my own efforts and that it has not been submitted to any other universities for the award of degree or diploma.

**ARUNURU NAVEEN**

**(L20MC23014)**

Table of Contents

[ACKNOWLEDGEMENT iii](#_Toc80180563)

[ABSTRACT viii](#_Toc80180564)

[CHAPTER-1 1](#_Toc80180565)

[INTRODUCTION 1](#_Toc80180566)

[1.1 INTRODUCTION 1](#_Toc80180567)

[CHAPTER- 2 5](#_Toc80180568)

[LITERATURE SURVEY 5](#_Toc80180569)

[2.1 LITERATURE REVIEW 5](#_Toc80180570)

[CHAPTER-3 7](#_Toc80180571)

[THEORETICAL BACKGROUND 7](#_Toc80180572)

[3.1 INTRODUCTION: 7](#_Toc80180573)

[3.2 INTRODUCTION TO PYTHON 12](#_Toc80180574)

[3.3 BENFITS OF PYTHON 21](#_Toc80180575)

[CHAPTER-4 30](#_Toc80180576)

[SYSTEM ANALYSIS 30](#_Toc80180577)

[4.1 EXISTING SYSTEM: 30](#_Toc80180578)

[4.1.1 DISADVANTAGES OF EXISTING SYSTEM: 30](#_Toc80180579)

[4.2 PROPOSED SYSTEM: 30](#_Toc80180580)

[4.2.1 ADVANTAGES OF PROPOSED SYSTEM: 31](#_Toc80180581)

[CHAPTER- 5 32](#_Toc80180582)

[SYSTEM DESIGN 32](#_Toc80180583)

[5.1 INTRODUCTION 32](#_Toc80180584)

[5.2 MODULES 32](#_Toc80180585)

[5.2.1 DATASET: 32](#_Toc80180586)

[5.2.2 PREPROCESSING: 32](#_Toc80180587)

[5.2.3 GRAPHS: 32](#_Toc80180588)

[5.2.4 PREDICTION: 32](#_Toc80180589)

[5.3 SYSTEM ARCHITECTURE 33](#_Toc80180590)

[5.4 UML DAIGRAMS 34](#_Toc80180591)

[5.4.1 CONSTRUCTION OF USE CASE DIAGRAMS: 37](#_Toc80180592)

[5.4.2 SEQUENCE DIAGRAMS: 40](#_Toc80180593)

[5.4.3 CLASS DIAGRAM: 42](#_Toc80180594)

[5.4.4 ACTIVITY DIAGRAM: 42](#_Toc80180595)

[CHAPTER-6 44](#_Toc80180596)

[SYSTEM REQUIREMENTS 44](#_Toc80180597)

[6.1 SYSTEM REQUIREMENTS 44](#_Toc80180598)

[6.1.1 HARDWARE REQUIREMENTS: 44](#_Toc80180599)

[6.1.2 SOFTWARE REQUIREMENTS: 44](#_Toc80180600)

[CHAPTER-7 45](#_Toc80180601)

[SYSTEM IMPLEMENTATION 45](#_Toc80180602)

[7.1 INPUT AND OUTPUT DESIGNS 45](#_Toc80180603)

[7.1.1 LOGICAL DESIGN 45](#_Toc80180604)

[7.1.2 PHYSICAL DESIGN 45](#_Toc80180605)

[7.2 INPUT & OUTPUT REPRESENTATION 46](#_Toc80180606)

[7.2.1 INPUT DESIGN 46](#_Toc80180607)

[7.2.2 OBJECTIVES 47](#_Toc80180608)

[7.2.3 OUTPUT DESIGN 47](#_Toc80180609)

[CHAPTER-8 69](#_Toc80180610)

[SYSTEM TESTING 69](#_Toc80180611)

[8.1 INTRODUCTION: 69](#_Toc80180612)

[8.2 LEVELS OF TESTING 69](#_Toc80180613)

[8.2.1 BLACK BOX TESTING 70](#_Toc80180614)

[8.2.2 WHITE BOX TESTING 72](#_Toc80180615)

[CHAPTER-9 74](#_Toc80180616)

[OUTPUT SCREENS 74](#_Toc80180617)

[CONCLUSION 75](#_Toc80180618)

[REFERENCES 76](#_Toc80180619)

**List of Figures**

**Name of the figure Pg.no**

[Figure 3. 1 Structure of cloud computing 7](#_Toc80179715)

[Figure 3. 2 Characteristics of cloud computing 9](#_Toc80179716)

[Figure 3. 3 Structure of service models 10](#_Toc80179717)

[Figure 5. 1 System Architecture 33](#_Toc80179735)

[Figure 5. 2 Use Case Diagram 39](#_Toc80179736)

[Figure 5. 3 Sequence diagram 41](#_Toc80179737)

[Figure 5. 4 Class Diagram 42](#_Toc80179738)

[Figure 5. 5 Activity Diagram 43](#_Toc80179739)

# ABSTRACT

Our main Motivation is to predict the disease at most early stage, so that the disease diagnosed person can able to get the treatment at early stage. So, we tried to predict as much as diseases as we can those are Breast Cancer, Kidney disease, Diabetes, Heart Disease Breast cancer accounts for 14% of cancers in Indian women. It is reported that with every four minutes, an Indian woman is diagnosed with breast cancer. Breast cancer is on the rise, both in rural and urban India. A 2018 report of Breast Cancer statistics recorded 1,62,468 new registered cases and 87,090 reported deaths. Cancer survival becomes more difficult in higher stages of its growth, and more than 50% of Indian women suffer from stage 3 and 4 of breast cancer. Post cancer survival for women with breast cancer was reported 60% for Indian women, as compared to 80% in the U.S. Kidney Disease, per year over 100,000 patients are diagnosed with End Stage Kidney Disease (ESKD) in India. The common kidney disease is chronic kidney disease and it affects about 10% of the world's population. Due to the lack of accurate national data collection, the incidence of CKD in India is not clear but studies estimate that the number of new patients diagnosed with End Stage Kidney Disease (ESKD) who are started on dialysis or transplantation is over 100,000 per year. India has an estimated 77 million people (1 in 11 Indians) formally diagnosed with diabetes, which makes it the second most affected in the world, after China. Furthermore, 700,000 Indians died of diabetes, hyper glycemia, kidney disease or other complications of diabetes in 2020. An estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Of these deaths, 85% were due to heart attack and stroke. n 2016 India reported 63% of total deaths due to NCDs, of which 27% were attributed to CVDs. CVDs also account for 45% of deaths in the 40–69-year age group. 11 Individuals at risk of CVD may demonstrate raised blood pressure, glucose, and lipids as well as overweight and obesity. And finally coming to Malaria, with approximately 4.2 million estimated malaria cases and 7341 estimated malaria- deaths, India accounted for a total of 83% estimated malaria cases and 82% estimated malaria-deaths in the WHO South-East Asia Region. Detection of these diseases at early stages of infection can, in many cases, drastically increase the chances of survival and can prevent a large number of deaths. Some of the common examinations done to determine the presence of these diseases. The presence of trained professionals who can read these scans and determine the infection are necessary in every locality. But in most cases, some of the doctors cover vast areas and their presence at all times is not possible.

.

# CHAPTER-1

# INTRODUCTION

## Introduction

During a lot of analysis over existing systems in health care analysis considered only one disease at a time. For example, article [1] is used to analyses diabetes, article [2] is used to analyses diabetes retinopathy, article [3] is used to predict heart disease [11]. Maximum articles focus on a particular disease. When any organization wants to analyses their patient’s, health reports then they have to deploy many models. The approach in the existing system is useful to analyses only particular disease. Now a day’s mortality got increased due to exactly not identifying exact disease. Even the patient got cured from one disease may be suffering from another disease. In real life, I faced that situation. My father got cured from the accident. My father got discharged from hospital but after a few days he got expired. Internally suffering from heart issues which is not identified. Like this many instances observed in many people’s life stories. Some existing systems used few parameters while analyzing the disease. Due to that may be not possible to identify the diseases which will be caused due to the effect of that disease. For example, due to diabetes, there may be chance of heart disease, neuropath, retinopathy, hearing loss, and dementia. In this article considered Diabetes analysis, Diabetes Retinopathy, heart disease and cancer detection data sets. In future many other diseases like skin diseases can be included, fever related diseases and many more. This analysis is flexible that later included many diseases for analysis. While adding any new disease analysis to this existing API, the developer has to add the model file related to the analysis of the new disease. When developing new disease the developer have to prepare python picking to save model behavior. When using this Flask API, the developer can load pickled file to retrieve the model behavior. When user wants to analyses the patient’s health condition either then can predict a particular disease or if the report contains parameters which are used to predict other diseases then this analysis will produce maximum identification of relevant diseases. The aim of this article is used to prevent mortality ratio increasing day by day by warning the patients in advance based on their health conditions. Due to many diseases models and predictions done at one place cost of patient analysis can be reduced.

**Problem Identification & Objectives**

video cameras have become a resource for controlling and regulating traffic in urban areas. They make it possible to analyze and monitor the traffic flowing within the city [5]. However, the number of cameras needed to perform these tasks has been increasing significantly over time, which makes control difficult if automation mechanisms are not implemented because the number of professionals needed to comply with all the points also increases. Several approaches have been proposed to automate tasks within the control and follow-up process. An example of this is a system based on video camera surveillance in traffic. Through these, it is possible to estimate the speeds and trajectories of the objects of interest [6], with the objective of predicting and controlling the occurrence of traffic accidents in the area. The scientific community has presented different approaches to detect traffic accidents [7]. These include statistics-based methods [8–10], social network data analysis [11,12], sensor data [13,14], machine learning, and deep learning [15–18]. These latest techniques have presented improvements in various fields of science, including video-based problem solving (video processing). Therefore, it is important to study these tech niques in order to approach a solution to the detection and classification of traffic accidents based on video

**Objective:**

The main objective of this project is to classify various types of Diseases using the Machine Learning Algorithms. We plan to predict different types of diseases like Breast Cancer, Diabetes, Kidney Diseases, Heart Disease and Malaria, with the help of the data sets which we will be getting from an online website called “Kaggle.” We will try to use different types of Algorithms under the Supervised Learning Category and in the end, we will compare the Algorithms with their achieved Accuracies.

**AIM**:

We are proposing such a system that will flaunt a simple, cost effective, elegant User Interface and also be time efficient. Our proposed system bridges the gap between doctors and patients which will help both classes of users to achieve their goal. This system is used to predict below mentioned diseases - Diabetes - Breast Cancer - Heart Disease - Kidney Disease - Liver Disease - Malaria.

**Limitations:**

An estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Of these deaths, 85% were due to heart attack and stroke. n 2016 India reported 63% of total deaths due to NCDs, of which 27% were attributed to CVDs. CVDs also account for 45% of deaths in the 40–69-year age group. 11 Individuals at risk of CVD may demonstrate raised blood pressure, glucose, and lipids as well as overweight and obesity. And finally coming to Malaria, with approximately 4.2 million estimated malaria cases and 7341 estimated malaria- deaths, India accounted for a total of 83% estimated malaria cases and 82% estimated malaria-deaths in the WHO South-East Asia Region. Detection of these diseases at early stages of infection can, in many cases, drastically increase the chances of survival and can prevent a large number of deaths. Some of the common examinations done to determine the presence of these diseases. The presence of trained professionals who can read these scans and determine the infection are necessary in every locality. But in most cases, some of the doctors cover vast areas and their presence at all times is not possible

# CHAPTER- 2

# literature survey

## 2.1 literature review

The American Diabetes Association’s (ADA’s) Standards of Medical Care in Diabetes are published each year in a supplement to the January issue of Diabetes Care. The ADA’s Professional Practice Committee develops the Standards and updates them annually, or more frequently online should it determine that new evidence or regulatory changes (e.g., drug approvals, label changes) merit immediate incorporation. The Standards include the most current evidence-based recommendations for diagnosing and treating adults and children with diabetes. ADA’s grading system uses A, B, C, or E to show the evidence level that supports each recommendation. A—Clear evidence from well-conducted, generalizable randomized controlled trials that are adequately powered B—Supportive evidence from well-conducted cohort studies C—Supportive evidence from poorly controlled or uncontrolled studies E—Expert consensus or clinical experience This is an abridged version of the Standards containing the evidence-based recommendations most pertinent to primary care. The tables and figures have been renumbered from the original document to match this version. All of the recommendations (bulleted text) are precisely the same as in the full Standards of Care. The complete 2018 Standards of Care document, including all supporting references, is available at professional.diabetes.org/standards. Go to: IMPROVING CARE AND PROMOTING HEALTH IN POPULATIONS Over the past 10 years, the proportion of patients with diabetes who achieve recommended A1C, blood pressure, and LDL cholesterol levels has increased. The mean A1C nationally among people with diabetes has declined from 7.6% (60 mmol/mol) in 1999–2002 to 7.2% (55 mmol/mol) in 2007–2010 based on the National Health and Nutrition Examination Survey, with younger adults less likely to meet treatment targets than older adults. This has been accompanied by improvements in cardiovascular outcomes and has led to substantial reductions in end-stage microvascular complications. Nevertheless, 33–49% of patients still do not meet targets for glycemic, blood pressure, or cholesterol control, and only 14% meet targets for all three measures while also avoiding smoking. Optimal diabetes management requires an organized, systematic approach and the involvement of a coordinated team of dedicated health care professionals working in an environment where patient-centered high-quality care is a priority. Recommendations Ensure treatment decisions are timely, rely on evidence-based guidelines, and are made collaboratively with patients based on individual preferences, prognoses, and comorbidities. B Align approaches to diabetes management with the Chronic Care Model, emphasizing productive interactions between a prepared proactive care team and an informed activated patient. A Care system should facilitate team-based care, patient registries, decision support tools, and community involvement to meet patient needs. B Efforts to assess the quality of diabetes care and create quality improvement strategies should incorporate reliable data metrics, to promote improved processes of care and health outcomes, with simultaneous emphasis on costs. E.

**Feature Importance Score-Based Functional Link Artificial Neural Networks for Breast Cancer Classification**

Growth of malignant tumors in the breast results in breast cancer. It is a cause of death of many women across the world. As a part of treatment, a woman might have to go through painful surgery and chemotherapy that may further lead to severe side effects. However, it is possible to cure it if it is diagnosed in the initial stage. Recently, many researchers have leveraged machine learning (ML) techniques to classify breast cancer. However, these methods are computationally expensive and prone to the overfitting problem. A simple single-layer neural network, i.e., functional link artificial neural network (FLANN), is proposed to overcome this problem. Further, the F-score is used to reduce the issue of overfitting by selecting features having a higher significance level. In this paper, FLANN is proposed to classify breast cancer using Wisconsin Breast Cancer Dataset (WBCD) (with 699 samples) and Wisconsin Diagnostic Breast Cancer (WDBC) (with 569 samples) datasets. Experimental results reveal that the proposed models can diagnose breast cancer with higher performance. The proposed model can be used in the early breast cancer diagnosis with 99.41% accuracy.

**The utilisation of machine learning approaches for medical data classification and personal care system mangementfor sickle cell disease**

The expert systems and smart devices played a key role in the development of health care in terms of continuous monitoring of patient’s treatment and preservation of E-medication system. The basic challenge that patients faced is the fact of difficulty in contacting physician specialists. The problem is there was no direct contact with the physician. This paper proposes an intelligent system that can offer self-care and monitoring system that can simulate the patient based on the application installed on his smartphone. The procedure will be whenever a patient sends his information about his blood test and other tests, the expert system will decide whether the situation is critical or not. In non-critical condition, the intelligent system will provide the recommendations and treatment directly. Otherwise, it will contact the physician directly to suggest the proper action that the patient should follow. Further expert system will update information regularly with patient information. A machine-learning algorithm was conductor perform the classification process.

# Chapter-3

# Theoretical background

## 3.1 Introduction:

**3.1 Software Used**

**3.1.1 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. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

**Keras Keras** is an API designed for human beings, now no longer machines. Keras follows best practices for lowering cognitive load: it gives consistent & easy APIs, it minimizes the quantity of consumer movements required for not un-usual place use cases, and it offers clean and actionable comments upon consumer error. This makes Keras smooth to research and smooth to use. As a Keras consumer, you are greater productive, permitting you to strive greater thoughts than your competition, faster which in flip facilitates you win system getting to know competitions. This ease of use does now no longer come on the price of decreased flexibility: because Keras integrates deeply with low-level TensorFlow capability, it enables you to increase extraordinarily hackable workflows in which any piece of capability may be customized.

**Jupyter Jupyter** is, in a nutshell: it is a device for collaborating. It is constructed for writing and sharing code and text, in the context of an internet web page. The code runs on a server, and the outcomes are become HTML and included in the web page you are writing. That server may be anywhere: to your laptop, in the back of your firewall, or on the general public internet. Your web page carries your thoughts, your code, and the outcomes of running the code. Code is in no way simply code. It is a part of a concept process, an argument, even a test. This is especially authentic for facts evaluation, however it is authentic for nearly any application. Jupyter helps you to construct a “lab notebook” that indicates your work: the code, the facts, the outcomes, at the side of your rationalization and reasoning. As IBM places it, Jupyter helps you to construct a “computational narrative that distills facts into insights.” Data means nothing in case you cannot flip it into insight, in case you cannot discover it, proportion it, and speak it. Data evaluation manner little in case you cannot discover and test with a person else’s 14 outcomes. Jupyter is a device for exploring, sharing, and discussing.

**Flask Flask** is a micro internet framework written in Python. It is classed as a microframework as it does now no longer requires precise equipment or libraries. It has no database abstraction layer, shape validation, or some other additives wherein pre-present third-birthday birthday celebration libraries offer not un usual place functions. However, Flask helps extensions which could upload utility capabilities as though they had been carried out in Flask itself. Extensions exist for object-relational mappers, shape validation, add handling, diverse open authentication technology and numerous un usual place framework associated equipment.

## 3.2 Introduction to PYTHON

**Python Technology**

Python technology is both a programming language and a platform.

**The python Programming Language**

THE PYTHON PROGRAMMING LANGUAGE IS A HIGH-LEVEL LANGUAGE THAT CAN BE CHARACTERIZED BY ALL OF THE FOLLOWING BUZZWORDS:

* + - Simple
    - Architecture neutral
    - Object oriented
    - Portable
    - Distributed
    - High performance
    - Interpreted
    - Multithreaded
    - Robust
    - Dynamic
    - Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Python programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Python byte codes —the platform-independent codes interpreted by the interpreter on the Python platform. The interpreter parses and runs each Python byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

FEATURES OF MACHINE LEARNING

• It is nothing but automating the Automation.

• Getting computers to program themselves.

• Writing Software is bottleneck.

• Machine leaning models involves machines learning from data without the help of humans or any kind of human intervention.

• Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.

• Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return it gives us the program which provides solution to the various problems. Below is the figure.

**Traditional Programming vs Machine Learning**

• Machine Learning is a combination of Algorithms, Datasets, and Programs.

• There are Many Algorithms in Machine Learning through which we will provide us the exact solution in predicting the disease of the patients.

• How Does Machine Learning Works?

• Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.

**Machine Learning Model**

• There are various applications in which machine learning is implemented such as Web search, computing biology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.

• There are 3 types of machine learning supervised, unsupervised, and reinforcement.

**BENEFITS OF PYTHON**

• Presence of Third-Party Modules

• Extensive Support Libraries

• Open Source and Community Development

• Learning Ease and Support Available

• User-friendly Data Structures

• Productivity and Speed

• Highly Extensible and Easily Readable Language.

**Python**

Python is high level language and it is also integrated version of the program. Python is an object-oriented approach and its main aim to help programmers to write the code clearly, logical code for small and large scale of project.

Pytrhon is dynamically typed and garbage collected it also support multiple programming and it is both procedure and object oriented and also functional programming. And structural programming also supported. It has many built in function it also supports filter, map and reduce function. All the machine learning algorithm and the libraries are being supported by the python programming language. Python also support list, dict, sets and other generators. Python code can be run in different platform such as anaconda, PyCharm etc.

The main goal of this programing language is as follows:

• Python is simple, object-oriented programming language.

• The language and implementation should provide support for software engineering principles such as strong type library preset for different machine learning algorithm, and all other algorithm in simple manner.

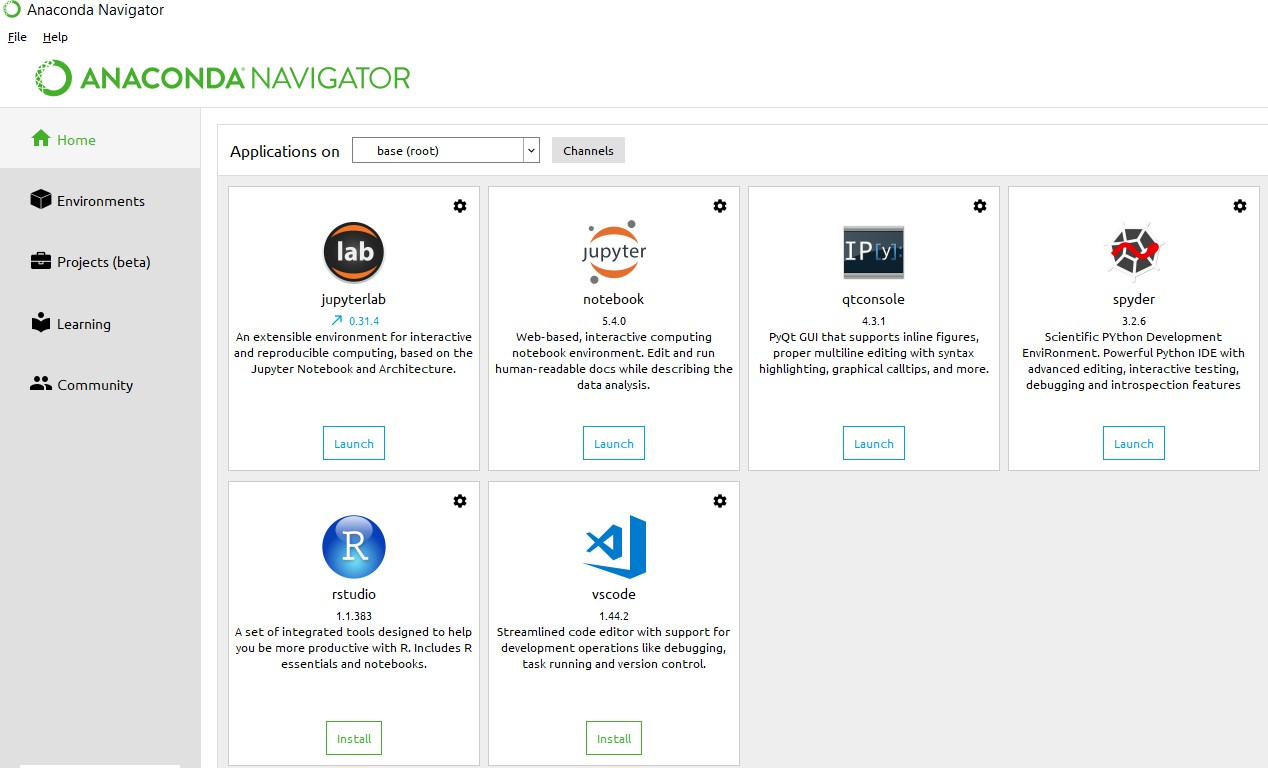
• Coding will be smooth in python and the data analysis can be easily done in python.

This is so much so to the point where we now have modules and APIs at our disposal, and you can engage in machine learning very easily without almost any knowledge at all of how it works. With the defaults from Scikit-learn, you can get 90-95% accuracy on many tasks right out of the gate. Machine learning is a lot like a car, you do not need to know much about how it works in order to get an incredible amount of utility from it.

Despite the apparent age and maturity of machine learning, I would say there's no better time than now to learn it, since you can actually use it. Machines are quite powerful, the one you are working on can probably do most of this series quickly. Data is also very plentiful lately.

**Anaconda**

Anaconda is free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine Learning applications, Large- scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. It is developed and maintained by Anaconda, Inc. The distribution incudes data-science packages suitable for Windows, Linux, and macOS. Packaged versions are required and are managed by the package management system anaconda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depends on, and a small number of other packages.



**Anaconda Console**

**Jupyter notebook**

Jupiter Notebook or so called IPython Notebook is an interactive web based computational mean for starting with Jupiter Notebook documents. The term notebook itself is a huge entity to represent the integration with different entity sets. JSON is the main document form from the same for the execution which follows the brief on the schema and the input and output means. It has high integration with several language set and has various flexibilities with the choices. The extension used for the same is “.ipynb” which runs in this platform. It’s an open-source software package with interactive communication means. It has it’s open standards for the same. It’s an open community best for budding programmers . The flexibility of the same is phenomenon and splendidly done the configuration and integration of the same is simplest and easy on hold so that no prior distortion is generated and the efficiency of the same is measured through out any system of choice.

It’s the best software sets that been used across cross for designing and developing of the products and support wide help support. Not only to that, it provides scalability in the code and the deployment of the same. Various Language can be changed and the project can be undertaken on the same. The created notebook files can be shared and stored in various means for further utilization. It supports cultivated and interactive output sets. Easily crossed over for graphing, plotting and visualizing of the elements. Data Integration of the same is to it’s best. The integration of big data and it can process chunks of values in an approx. time which gives a better performance and the higher computational means. Various works on data like cleaning, cleansing, transforming modeling and visualizing can be done by the same

Machine learning is the ability that gives the computer to learn without being explicitly programmed. There are two types of machine learning:

Supervised Learning: supervised learning is the learning of the labelled data. It is the types of machine learning that maps the input and output based on the examples input-output pairs. In supervised learning each training data having pairs of input and desired outputs values. Supervised learning algorithm analyzes the training data and produces a function which can be used for mapping of new data.

Fig 2.1 Supervised Learning The output to solve the supervised learning algorithm are as:

• Determine the types of data, before doing anything else the user should understand which types of data set is to be used for training the data.

• Gathered the training data sets either in form of human experts or from measurements.

• Determine the feature of inputs from the learned data and depends on the inputs it changed into feature vector; number of features should not be large but should contains enough information to accurately predict the outputs.

• Check the learned function and the learned algorithm for example we use support vector machines or decisions tree.

• Complete the design and run the trained data sets.

• Analyzed the output and verify the data sets to get the accurate outputs.

Unsupervised Learning:

Unsupervised learning is a type of machine learning that helps in finding the previously unknown patterns in the data set without any known labels. It is known as self- organization and allows modelling probability densities of given inputs.

Fig 2.2 unsupervised Learning Some of the algorithm used in unsupervised learning are:

• Clustering

• Anomaly detection

• Neural networks

• Approach for learning latent variable models

• Non labelled data

Semi Supervised Machine Learning algorithm: It’s like the middle man which have some labeled data and some unlabeled which can be prosed by the both the structured and unsupervised learning.

The algorithms have been compared based upon the parameters: Size of the dataset and Number of technical indicators used. Accuracy and F-measure values have been computed for each algorithm. Long term model has been used to compute the accuracy and F-measure.

Reinforcement Learning: This type of learning is used to reinforce or strengthen the network based on critic information. That is, a network being trained under reinforcement learning, receives some feedback from the environment. However, the feedback is evaluative and not instructive as in the case of supervised learning. Based on this feedback, the network performs the adjustments of the weights to obtain better critic information in future.

This learning process is similar to supervised learning but we might have very less information. The following figure gives the block diagram of reinforcement learning:

**import numpy as np**

* NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
* At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:
  + NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.
  + The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
  + NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python’s built-in sequences.
  + A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today’s scientific/mathematical Python-based software, just knowing how to use Python’s built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

**import time**

This module provides various time-related functions. For related functionality, see also the datetime and calendar modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

An explanation of some terminology and conventions is in order.

The epoch is the point where the time starts, and is platform dependent. For Unix, the epoch is January 1, 1970, 00:00:00 (UTC). To find out what the epoch is on a given platform, look at time.gmtime(0).

The term seconds since the epoch refers to the total number of elapsed seconds since the epoch, typically excluding leap seconds. Leap seconds are excluded from this total on all POSIX-compliant platforms.

The functions in this module may not handle dates and times before the epoch or far in the future. The cut-off point in the future is determined by the C library; for 32-bit systems, it is typically in 2038.

Function strptime() can parse 2-digit years when given %y format code. When 2-digit years are parsed, they are converted according to the POSIX and ISO C standards: values 69–99 are mapped to 1969–1999, and values 0–68 are mapped to 2000–2068.

UTC is Coordinated Universal Time (formerly known as Greenwich Mean Time, or GMT). The acronym UTC is not a mistake but a compromise between English and French.

DST is Daylight Saving Time, an adjustment of the timezone by (usually) one hour during part of the year. DST rules are magic (determined by local law) and can change from year to year. The C library has a table containing the local rules (often it is read from a system file for flexibility) and is the only source of True Wisdom in this respect.

The precision of the various real-time functions may be less than suggested by the units in which their value or argument is expressed. E.g. on most Unix systems, the clock “ticks” only 50 or 100 times a second.

On the other hand, the precision of time() and sleep() is better than their Unix equivalents: times are expressed as floating point numbers, time() returns the most accurate time available (using Unix gettimeofday() where available), and sleep() will accept a time with a nonzero fraction (Unix select() is used to implement this, where available).

The time value as returned by gmtime(), localtime(), and strptime(), and accepted by asctime(), mktime() and strftime(), is a sequence of 9 integers. The return values of gmtime(), localtime(), and strptime() also offer attribute names for individual fields.

See struct\_time for a description of these objects.

Changed in version 3.3: The struct\_time type was extended to provide the tm\_gmtoff and tm\_zone attributes when platform supports corresponding struct tm members.

Changed in version 3.6: The struct\_time attributes tm\_gmtoff and tm\_zone are now available on all platforms.

**import os**

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the fileinput module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

Notes on the availability of these functions:

The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function os.stat(path) returns stat information about path in the same format (which happens to have originated with the POSIX interface).

Extensions peculiar to a particular operating system are also available through the os module, but using them is of course a threat to portability.

All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.

On VxWorks, os.popen, os.fork, os.execv and os.spawn\*p\* are not supported.

# 

# Chapter-4

# System analysis

## 4.1 EXISTING SYSTEM:

“Disease Prediction System” used Decision tree, Random Forest and Naïve Bayes algorithms to predict a disease on the basis of systems and to enable synchronized and well-versed medical systems ensuring maximum patient satisfaction

Heart Disease Prediction with Machine Learning Approaches" made use of LR, NB,KNN,SVM,DT and RF algorithms for prediction of heart disease with proper data processing and implementation of ML algorithm with different parameters and among all Machine Learning algorithms, the highest accuracy is achieved by KNN with 87%.

”Heart Attack Prediction By Using Machine Learning Techniques" has compared various Machine Learning models with the help of performance metrics and to detect heart related problems with highest accuracy of 89.34% by SVM.

”Disease Prediction Using Machine Learning over Big Data” has proposed a CNN-MDRP algorithm which combines structured and unstructured data and proved that CNN-MDRP is more accurate than previous prediction algorithm.

.

### 4.1.1 DISADVANTAGES OF EXISTING SYSTEM:

* Existing system works for only single disease and for each type of disease use should visit multiple websites to get predict disease
* Existing methods use same algorithms for all disease predictions.

## 4.2 PROPOSED SYSTEM:

* We are proposing such a system that will flaunt a simple, cost effective , elegant User Interface and also be time efficient . Our proposed system bridges the gap between doctors and patients which will help both classes of users to achieve their goal. This system is used to predict below mentioned diseases
* Diabetes
* - Breast Cancer
* - Heart Disease
* - Kidney Disease
* - Liver Disease
* - Malaria
* - Pneumonia
* . In this proposed system we are going to take down six disease dataset from the Kaggle website and evaluate them by applying algorithms such as Decision Tree, Random Forest, Naïve bayes and KNN which will help in getting accurate prediction .Our system will explore and merge more datasets which includes large diversity of population to get more effective results and thus our system will improve and enhances the accuracy of the results. Along with the increased accuracy rate, we will proliferate the reliability of our system for this job and can gain the trust of patient in this system. Apart from all these, our system will comprise of a Database for storing the data entered by the users and the name of the disease the patient is suffering from which can be used as a reference in future for further treatment. Hence this system will contribute in easier health management with better satisfaction to the users.

### 4.2.1 ADVANTAGES OF PROPOSED SYSTEM:

* Single website is used to predict six type of diseases.
* Each disease is trained with different algorithms
* Time taken for training and prediction is less with high actuary**.**

.

**FEASIBILITY STUDY**

**FEASIBILITY STUDY:**

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

1. Technical Feasibility
2. Economic Feasibility
3. Operation Feasibility

**Technical Feasibility:**

In the feasibility study first step is that the organization or company has to decide that what technologies are suitable to develop by considering existing system.

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system.

Here in this application used the technologies like Visual Studio 2012 and SqlServer 2014. These are free software that would be downloaded from web.

Visual Studio 2013 –it is tool or technology.

**ECONOMICAL FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

### Determining Economic Feasibility:

Assessing the economic feasibility of an implementation by performing a cost/benefit analysis, which as its name suggests compares the full/real costs of the application to its full/real financial benefits.  The alternatives should be evaluated on the basis of their contribution to net cash flow, the amount by which the benefits exceed the costs, because the primary objective of all investments is to improve overall organizational performance.

|  |  |  |
| --- | --- | --- |
| **Type** | **Potential Costs** | **Potential Benefits** |
| Quantitative | 1. Hardware/software upgrades 2. Fully-burdened cost of labor (salary + benefits) 3. Support costs for the application 4. Expected operational costs 5. Training costs for users to learn the application 6. Training costs to train developers in new/updated technologies | 1. Reduced operating costs 2. Reduced personnel costs from a reduction in staff 3. Increased revenue from additional sales of your organizations products/services |
| Qualitative | 1. Increased employee dissatisfaction from fear of change | 1. Improved decisions as the result of access to accurate and timely information 2. Raising of existing, or introduction of a new, barrier to entry within your industry to keep competition out of your market 3. Positive public perception that your organization is an innovator |

 The table includes both qualitative factors, costs or benefits that are subjective in nature, and quantitative factors, costs or benefits for which monetary values can easily be identified.  I will discuss the need to take both kinds of factors into account when performing a cost/benefit analysis.

**OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

Not only must an application make economic and technical sense, it must also make operational sense.

|  |  |
| --- | --- |
| **Operations Issues** | **Support Issues** |
| 1. What tools are needed to support operations? 2. What skills will operators need to be trained in? 3. What processes need to be created and/or updated? 4. What documentation does operations need? | 1. What documentation will users be given? 2. What training will users be given? 3. How will change requests be managed? |

Very often you will need to improve the existing operations, maintenance, and support infrastructure to support the operation of the new application that you intend to develop.  To determine what the impact will be you will need to understand both the current operations and support infrastructure of your organization and the operations and support characteristics of your new application. To operate this application END-TO-END VMS. The user no need to require any technical knowledge that we are used to develop this project is Asp.net C#.net. That the application providing rich user interface by user can do the operation in flexible manner.

# CHAPTER- 5

# SYSTEM design

## 5.1 introduction

System Design Introduction:

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

## 5.2 modules

### 5.2.1 Data COLLECTION:

Heart, kidney, diabetic, liver disease datasets are collected form Kaggle website which are in the form of csv format. These datasets have features and labels based on type of disease dataset we are using features and labels are changed.

**Understanding features of dataset.**

**Cancer Dataset:**

**Features:**

id radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean compactness\_mean concavity\_mean concave points\_mean symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se perimeter\_se area\_se smoothness\_se compactness\_se concavity\_se concave points\_se symmetry\_se fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst area\_worst smoothness\_worst compactness\_worst concavity\_worst concave points\_worst symmetry\_worst fractal\_dimension\_worst

Labels:

Diagnosis

**Diabetic Disease:**

**Features:**

Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome

**Labels:**

Outcome

### 5.2.2 Pre-processing:

In this stage data analysis of each dataset is performed to check relation between features and labels with graphical representation. Null values are removed from the dataset and balanced dataset is prepared for all diseases datasets.

### 5.2.3 Split Data:

Data set is split in to two parts using test train split function ( 80 and 20 ) as test and train datasets. Train features are called as train x and labels as train y. These values are used to train algorithm and test data is used to check accuracy of each disease dataset**.**

**5.2.4** **Apply ML algorithms**

In this stage pre processed dataset is taken as input of each disease dataset and trained features and labels are given as input to fit function to train model and model is saved in to system in the form of pkl file. The model is used in web application for prediction results based on user given input.

**5.2.5** **Accuracy results**

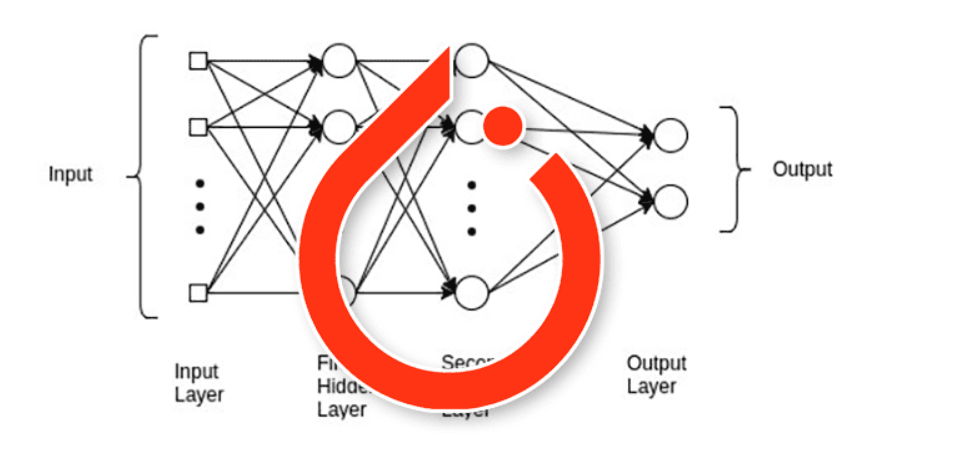
After training is done test set is given and input to algorithm to test accuracy of each dataset.

**5.2.6** **Flask Web framework:**

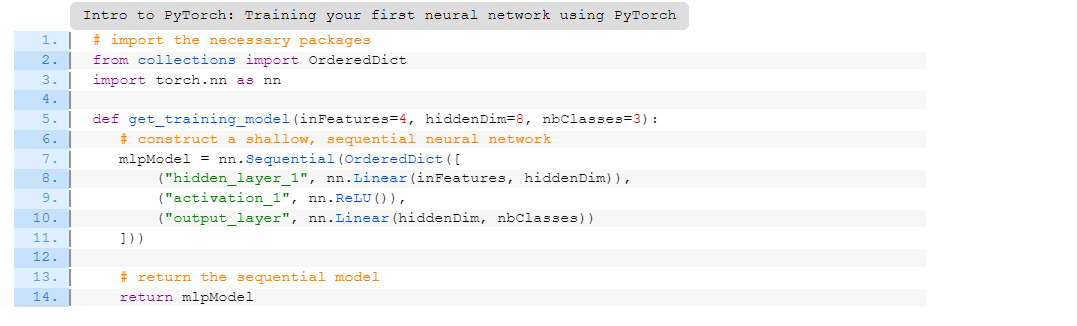
For this project web application is developed using flask framework which takes trained model as input and html, css for web page design. Using this application own input is given to webpage and disease is predicted.

**Algorithms**

**Implementing our neural network with PyTorch**

****

This network is a very simple feedforward neural network called a **multi-layer perceptron (MLP)** (meaning that it has one or more hidden layers).



OrderedDict: A dictionary object that remembers the order in which objects were added — we use this ordered dictionary to provide human-readable names to each layer in the network

nn: PyTorch’s neural network implementations

We then define the get\_training\_model function (Line 5) which accepts three parameters:

The number of input nodes to the neural network

The number of nodes in the hidden layer of the network

The number of output nodes (i.e., dimensionality of the output prediction)

Based on the default values provided, you can see that we are building a 4-8-3 neural network, meaning that the input layer has 4 nodes, the hidden layer 8 nodes, and the output of the neural network will consist of 3 values.

The actual neural network architecture is then constructed on Lines 7-11 by first initializing a nn.Sequential object (very similar to Keras/TensorFlow’s Sequential class).

Inside the Sequential class we build an OrderedDict where each entry in the dictionary consists of two values:

A string containing the human-readable name for the layer (which is very useful when debugging neural network architectures using PyTorch)

The PyTorch layer definition itself

The Linear class is our fully connected layer definition, meaning that each of the inputs connects to each of the outputs in the layer. The Linear class accepts two required arguments:The number of inputs to the layer

The number of outputs

On Line 8, we define hidden\_layer\_1 which consists of a fully connected layer accepting inFeatures (4) inputs and then producing an output of hiddenDim (8).

From there, we apply a ReLU activation function (Line 9) followed by another Linear layer which serves as our output (Line 10).

Notice that the second Linear definition contains the same number of inputs as the previous Linear layer did outputs — this is not by accident! The output dimensions of the previous layer must match the input dimensions of the next layer, otherwise PyTorch will error out (and then you’ll have the quite tedious task of debugging the layer dimensions yourself).

PyTorch is not as forgiving in this regard (as opposed to Keras/TensorFlow), so be extra cautious when specifying your layer dimensions.

The resulting PyTorch neural network is then returned to the calling function

With our neural network architecture implemented, we can move on to training the model using PyTorch.

To accomplish this task, we’ll need to implement a training script which:

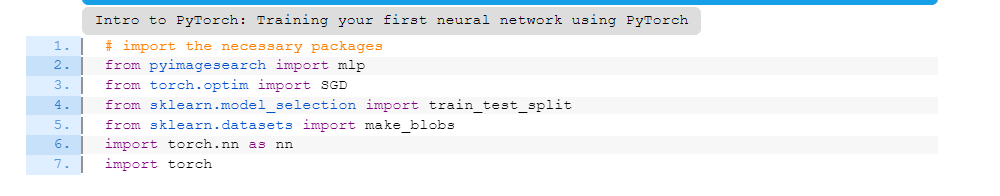
Creates an instance of our neural network architecture

Builds our dataset

Determines whether or not we are training our model on a GPU

Defines a training loop (the hardest part of our script)

Open train.py, and lets get started:

****

Lines 2-7 import our required Python packages, including:

mlp: Our definition of the multi-layer perceptron architecture, implemented in PyTorch

SGD: The Stochastic Gradient Descent optimizer that we’ll be using to train our model

make\_blobs: Builds a synthetic dataset of example data

train\_test\_split: Splits our dataset into a training and testing split

nn: PyTorch’s neural network functionality

torch: The base PyTorch library

## 5.3 system architecture

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system. Organized in a way that supports reasoning about the structures and behaviors of the system.

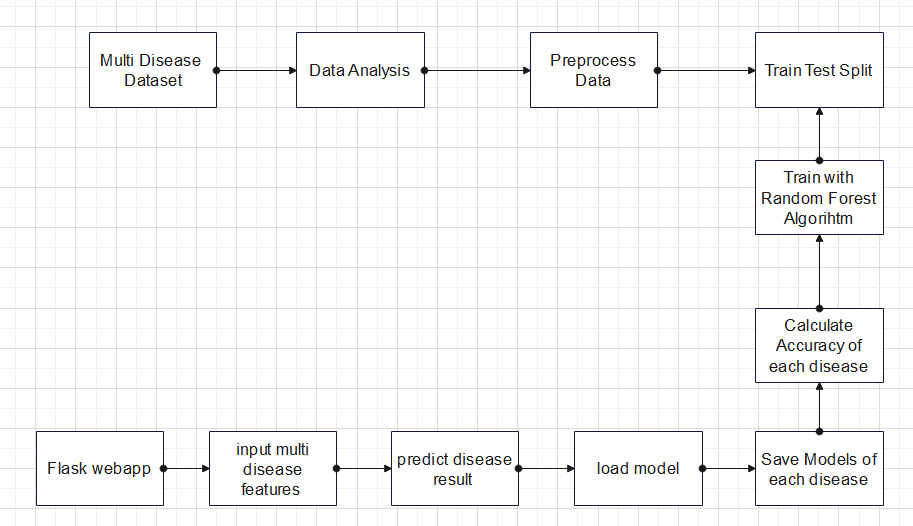


Figure 5. 1 System Architecture

3-Tier Architecture:

The three-tier software architecture (a three-layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems.

**Advantages of Three-Tier:**

* Separates functionality from presentation.
* Clear separation – better understanding.
* Changes limited to well define components.
* Can be running on WWW.
* Effective network performance.

## 5.4 UML DAIGRAMS

Global Use Case Diagrams:

Identification of actors:

Actor: Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation:



<<Actor name>>

An actor is someone or something that:

Interacts with or uses the system.

* Provides input to and receives information from the system.
* Is external to the system and has no control over the use cases.

Actors are discovered by examining:

* Who directly uses the system?
* Who is responsible for maintaining the system?
* External hardware used by the system.
* Other systems that need to interact with the system.

Questions to identify actors:

* + Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?
  + Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.
  + Which external hardware or systems (if any) use the system to perform tasks?
  + What problems does this application solve (that is, for whom)?
  + And, finally, how do users use the system (use case)? What are they doing with the system?

The actors identified in this system are:

1. System Administrator
2. Customer
3. Customer Care

Identification of use cases:

Use case: A use case can be described as a specific way of using the system from a user’s (actor’s) perspective.

Graphical representation:



A more detailed description might characterize a use case as:

* Pattern of behavior the system exhibits
* A sequence of related transactions performed by an actor and the system
* Delivering something of value to the actor

Use cases provide a means to:

* capture system requirements
* communicate with the end users and domain experts
* test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guide lines for identifying use cases:

* For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
* Name the use cases.
* Describe the use cases briefly by applying terms with which the user is familiar.

This makes the description less ambiguous

Questions to identify use cases:

* What are the tasks of each actor?
* Will any actor create, store, change, remove or read information in the system?
* What use case will store, change, remove or read this information?
* Will any actor need to inform the system about sudden external changes?
* Does any actor need to inform about certain occurrences in the system?
* What usecases will support and maintains the system?

**1.2 Flow of Events**

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

* When and how the use case starts and ends
* Use case/actor interactions
* Data needed by the use case
* Normal sequence of events for the use case
* Alternate or exceptional flows

### 5.4.1 Construction of Use case diagrams:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

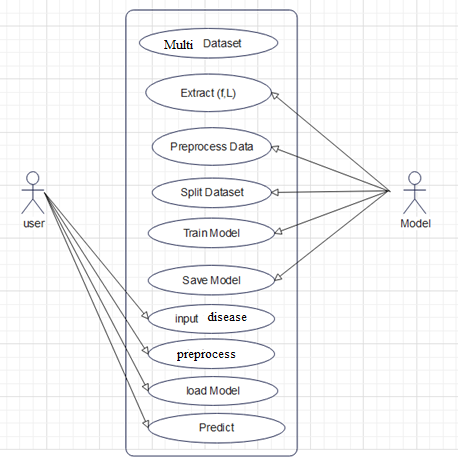


Figure 5. 2 Use Case Diagram

### 5.4.2 SEQUENCE DIAGRAMS:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

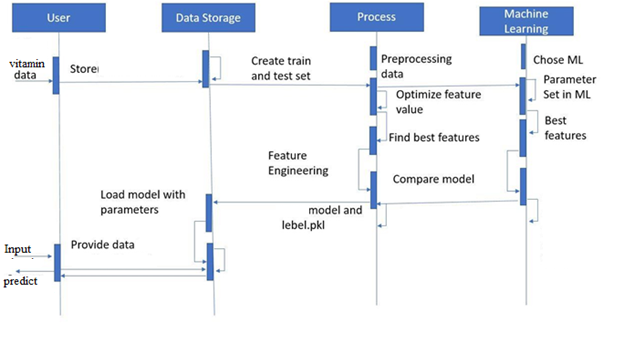


Figure 5. 3 Sequence diagram

### 5.4.3 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

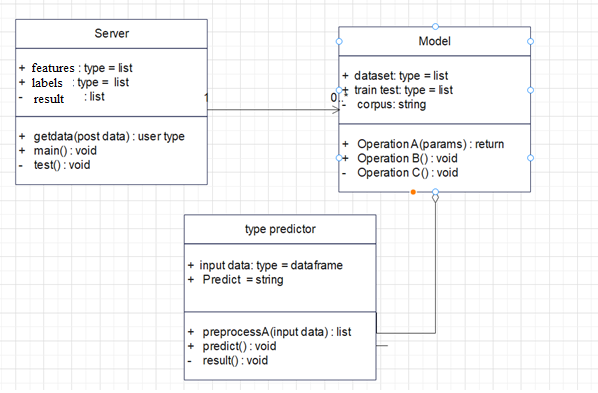


Figure 5. 4 Class Diagram

### 5.4.4 ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

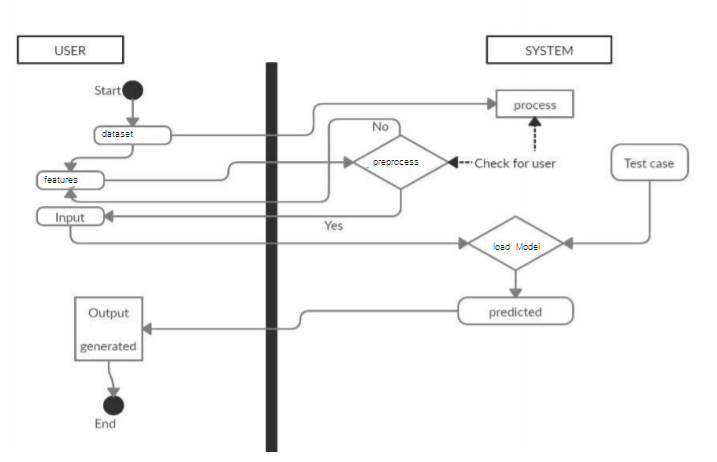
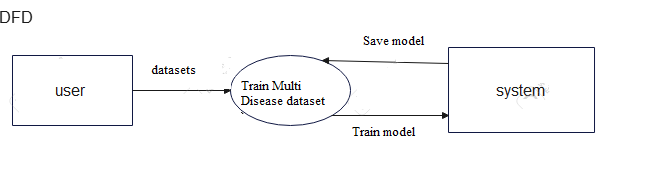
****

Figure 5. 5 Activity Diagram

**DFD:**

****

# 

# CHAPTER-6

# system requirements

## 6.1 SYSTEM REQUIREMENTS

### 6.1.1 HARDWARE REQUIREMENTS:

* System : Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz
* Hard Disk : 1 TB.
* Input Devices : Keyboard, Mouse
* Ram : 4 GB.

### 6.1.2 SOFTWARE REQUIREMENTS:

* Operating system : Windows XP/7/10.
* Coding Language : Python
* Tool :Anaconda
* Interface : Flask

# Chapter-7

# System implementation

To conduct studies and analyses of an operational and technological nature, and To promote the exchange and development of methods and tools for operational analysis as applied to defense problems.

## 7.1 Quality Requirement

### 7.1.1 Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams

### 7.1.2 Physical design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output. In Physical design, following requirements about the system are decided.

1. Input requirement,
2. Output requirements,
3. Storage requirements,
4. Processing Requirements,
5. System control and backup or recovery.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

1. User Interface Design
2. Data Design
3. Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

## 7.2 INPUT & OUTPUT REPRESENTATION

### 7.2.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

### 7.2.2 Objectives

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

### Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

* 1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
  2. Select methods for presenting information.
  3. Create document, report, or other formats that contain information produced by the system.

Algorithm Complexity:

**Decision Tree Algorithm**

The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem. The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

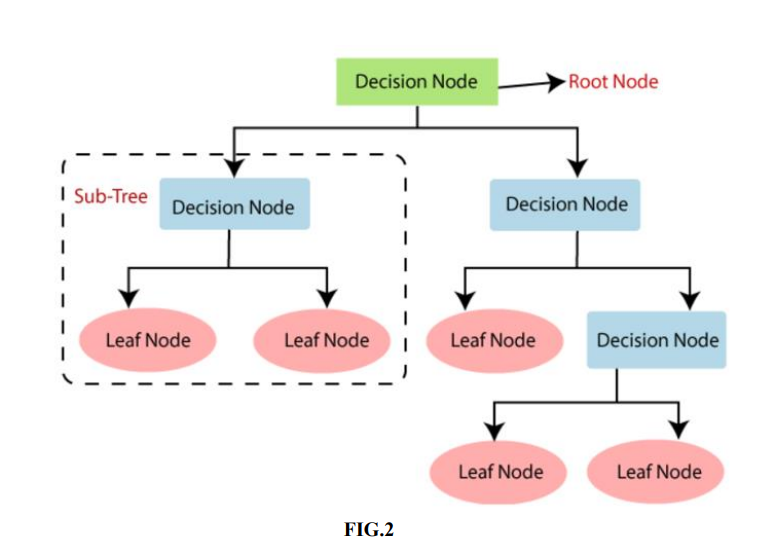
➢ Decision Tree is a Supervised learning technique

➢ That can be used for both classification and Regression problems

➢ It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome

➢ It is simple to understand as it follows the same process which a human follow while making any decision in real-life.

➢ It may have an overfitting issue, which can be resolved using the Random Forest algorithm



**Random Forest Algorithm:**

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

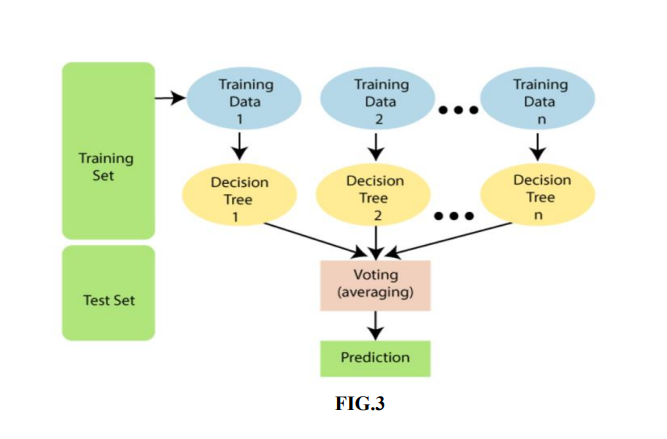
➢ Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.

➢ It can be used for both Classification and Regression problems in ML.

➢ Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

➢ It is capable of handling large datasets with high dimensionality and it enhances the accuracy of the model and prevents the overfitting issue.

➢ Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks

****

**Naïve Bayes Algorithm:**

Naive Bayes classifiers are a collection of classification algorithms based on Bayes’ Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

➢ Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.

➢ It is mainly used in text classification that includes a high-dimensional training dataset.

➢ Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

➢ It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

➢ Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

**K- Nearest neighbour Algorithm:**

➢ K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

➢ K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

17 ➢ K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

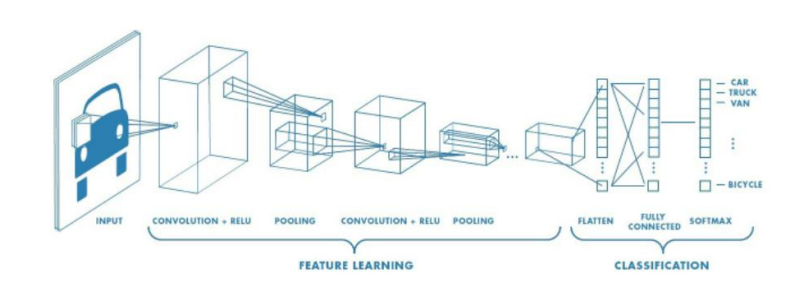
➢ K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

➢ K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

➢ It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

➢ KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data

**CNN Algorithm Convolutional Neural Network (CNN):**



is a Deep Learning set of rules that can absorb an enter photograph, assign significance to numerous aspects/gadgets withinside the photograph and have the ability to distinguish one from the different. The pre-processing required in a CNN is a lot decrease in comparison to different type algorithms. While in primitive strategies filters are hand-engineered, with sufficient training, CNN have the capacity to examine those filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area

**Methodology:**

Design Approach and Details

**4.1 Collection of datasets**. Heart, kidney, diabetic, liver and malaria disease datasets are collected form Kaggle website which are in the form of csv format. These datasets have features and labels based on type of disease dataset we are using features and labels are changed.

**4.2. Understanding features of dataset**. To understand the features which are given in the dataset and able to study that.

**4.3 Pre-processing the data.** In this stage data analysis of each dataset is performed to check relation between features and labels with graphical representation. Null values are removed from the dataset and balanced dataset is prepared for all diseases datasets.

**4.4 Split data into training dataset and testing dataset**. Data set is split in to two parts using test train split function (80 and 20) as test and train datasets. Train features are called as train x and labels as train y. These values are used to train algorithm and test data is used to check accuracy of each disease dataset.

**4.5 Apply ML algorithms** to dataset to predict which type of disease In this stage pre-processed dataset is taken as input of each disease dataset and trained features and labels are given as input to fit function to train model and model is saved in to system in the form of pkl file. The model is used in web application for prediction results based on user given input.

**4.6 Accuracy results** After training is done test set is given and input to algorithm to test accuracy of each dataset.

**4.7 Flask Web framework**: For this project web application is developed using flask framework which takes trained model as input and html, CSS for web page design. Using this application own input is given to webpage and disease is predicted.

**Code**

from flask import Flask, render\_template, request, flash, redirect

import pickle

import numpy as np

from PIL import Image

from tensorflow.keras.models import load\_model

app = Flask(\_\_name\_\_)

def predict(values, dic):

if len(values) == 8:

model = pickle.load(open('models/diabetes.pkl','rb'))

values = np.asarray(values)

return model.predict(values.reshape(1, -1))[0]

elif len(values) == 26:

model = pickle.load(open('models/breast\_cancer.pkl','rb'))

values = np.asarray(values)

return model.predict(values.reshape(1, -1))[0]

elif len(values) == 13:

model = pickle.load(open('models/heart.pkl','rb'))

values = np.asarray(values)

return model.predict(values.reshape(1, -1))[0]

elif len(values) == 18:

model = pickle.load(open('models/kidney.pkl','rb'))

values = np.asarray(values)

return model.predict(values.reshape(1, -1))[0]

elif len(values) == 10:

model = pickle.load(open('models/liver.pkl','rb'))

values = np.asarray(values)

return model.predict(values.reshape(1, -1))[0]

@app.route("/")

def home():

return render\_template('home.html')

@app.route("/diabetes", methods=['GET', 'POST'])

def diabetesPage():

return render\_template('diabetes.html')

@app.route("/cancer", methods=['GET', 'POST'])

def cancerPage():

return render\_template('breast\_cancer.html')

@app.route("/heart", methods=['GET', 'POST'])

def heartPage():

return render\_template('heart.html')

@app.route("/kidney", methods=['GET', 'POST'])

def kidneyPage():

return render\_template('kidney.html')

@app.route("/liver", methods=['GET', 'POST'])

def liverPage():

return render\_template('liver.html')

@app.route("/malaria", methods=['GET', 'POST'])

def malariaPage():

return render\_template('malaria.html')

@app.route("/pneumonia", methods=['GET', 'POST'])

def pneumoniaPage():

return render\_template('pneumonia.html')

@app.route("/predict", methods = ['POST', 'GET'])

def predictPage():

try:

if request.method == 'POST':

to\_predict\_dict = request.form.to\_dict()

to\_predict\_list = list(map(float, list(to\_predict\_dict.values())))

pred = predict(to\_predict\_list, to\_predict\_dict)

except:

message = "Please enter valid Data"

return render\_template("home.html", message = message)

return render\_template('predict.html', pred = pred)

@app.route("/malariapredict", methods = ['POST', 'GET'])

def malariapredictPage():

if request.method == 'POST':

try:

if 'image' in request.files:

img = Image.open(request.files['image'])

img = img.resize((36,36))

img = np.asarray(img)

img = img.reshape((1,36,36,3))

img = img.astype(np.float64)

model = load\_model("models/malaria.h5")

pred = np.argmax(model.predict(img)[0])

except:

message = "Please upload an Image"

return render\_template('malaria.html', message = message)

return render\_template('malaria\_predict.html', pred = pred)

@app.route("/pneumoniapredict", methods = ['POST', 'GET'])

def pneumoniapredictPage():

if request.method == 'POST':

try:

if 'image' in request.files:

img = Image.open(request.files['image']).convert('L')

img = img.resize((36,36))

img = np.asarray(img)

img = img.reshape((1,36,36,1))

img = img / 255.0

model = load\_model("models/pneumonia.h5")

pred = np.argmax(model.predict(img)[0])

except:

message = "Please upload an Image"

return render\_template('pneumonia.html', message = message)

return render\_template('pneumonia\_predict.html', pred = pred)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = True)

            valid\_loss\_min,

            valid\_loss))

        torch.save(model.state\_dict(), 'sed.pt')

        valid\_loss\_min = valid\_loss

# 

# Chapter-8

# System testing

## 8.1 INTRODUCTION:

Testing is the debugging program is one of the most critical aspects of the computer programming triggers, without programming that works, the system would never produce an output of which it was designed. Testing is best performed when user development is asked to assist in identifying all errors and bugs. The sample data are used for testing. It is not quantity but quality of the data used the matters of testing. Testing is aimed at ensuring that the system was accurately an efficiently before live operation commands.

Testing objectives:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, testing is a process of executing a program with intent of finding an error.

1. A successful test is one that uncovers an as yet undiscovered error.
2. A good test case is one that has probability of finding an error, if it exists.
3. The test is inadequate to detect possibly present errors.
4. The software more or less confirms to the quality and reliable standards.

## 8.2 Levels of Testing

Code testing:

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

Specification Testing:

Executing this specification starting what the program should do and how it should performed under various conditions. Test cases for various situation and combination of conditions in all the modules are tested.

Unit testing:

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during programming stage itself. In the testing step each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks for fields also. For example the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find error debut the system.

Each Module can be tested using the following two Strategies:

1. Black Box Testing
2. White Box Testing

### 8.2.1 BLACK BOX TESTING

What is Black Box Testing?

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.



The above Black Box can be any software system you want to test. For example : an operating system like Windows, a website like Google ,a database like Oracle or even your own custom application. Under Black Box Testing , you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Black box testing - Steps

Here are the generic steps followed to carry out any type of Black Box Testing.

* Initially requirements and specifications of the system are examined.
* Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.
* Tester determines expected outputs for all those inputs.
* Software tester constructs test cases with the selected inputs.
* The test cases are executed.
* Software tester compares the actual outputs with the expected outputs.
* Defects if any are fixed and re-tested.

Types of Black Box Testing

There are many types of Black Box Testing but following are the prominent ones -

* Functional testing – This black box testing type is related to functional requirements of a system; it is done by software testers.
* Non-functional testing – This type of black box testing is not related to testing of a specific functionality, but non-functional requirements  such as performance, scalability, usability.
* Regression testing – Regression testing is done  after code fixes , upgrades or any other system maintenance to check the new code has not affected the existing code.

### 8.2.2 WHITE BOX TESTING

White Box Testing is the testing of a software solution's internal coding and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.White box testing is also known as clear, open, structural, and glass box testing.

It is one of two parts of the "box testing" approach of software testing. Its counter-part, blackbox testing, involves testing from an external or end-user type perspective. On the other hand, Whitebox testing is based on the inner workings of an application and revolves around internal testing. The term "whitebox" was used because of the see-through box concept. The clear box or whitebox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "black box testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested

WHAT DO YOU VERIFY IN WHITE BOX TESTING?

White box testing involves the testing of the software code for the following:

* Internal security holes
* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops
* Testing of each statement, object and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

HOW DO YOU PERFORM WHITE BOX TESTING?

  To give you a simplified explanation of white box testing, we have divided it into **two basic steps**. This is what testers do when testing an application using the white box testing technique:

**STEP 1) UNDERSTAND THE SOURCE CODE**

The first thing a tester will often do is learn and understand the source code of the application. Since white box testing involves the testing of the inner workings of an application, the tester must be very knowledgeable in the programming languages used in the applications they are testing. Also, the testing person must be highly aware of secure coding practices. Security is often one of the primary objectives of testing software. The tester should be able to find security issues and prevent attacks from hackers and naive users who might inject malicious code into the application either knowingly or unknowingly.

**Step 2) CREATE TEST CASES AND EXECUTE**

The second basic step to white box testing involves testing the application’s source code for proper flow and structure. One way is by writing more code to test the application’s source code. The tester will develop little tests for each process or series of processes in the application. This  method requires that the tester must have intimate knowledge of the code and is often done by the developer. Other methods include manual testing, trial and error testing and the use of testing tools as we will explain further on in this article.

Unit testing:

|  |  |
| --- | --- |
| Sl # Test Case : ­ | UTC­1 |
| Name of Test: ­ | Load dataset |
| Items being tested: ­ | Dataset features and labels are displayed or not |
| Sample Input: ­ | Dataset csv file |
| Expected output: ­ | All features and labels should be displayed |
| Actual output: ­ | Total data is displayed |
| **Remarks: ­** | **Pass.** |

|  |  |
| --- | --- |
| Sl # Test Case : ­ | UTC­2 |
| Name of Test: ­ | Split data |
| Items being tested: ­ | Data is divided in to train and test set |
| Sample Input: ­ | Test and train size |
| Expected output: ­ | Dataset is divided in to 2 parts |
| Actual output: ­ | Based on given test size data is divided and stored in train and test sets |
| Remarks: ­ | pass |

**Integration Testing:**

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom­up integration testing and Top­down integration testing.

* + 1. **Bottom­up Integration**

This testing begins with unit testing, followed by tests of progressively higher­level combinations of units called modules or builds.

* + 1. **Top­down Integration**

In this testing, the highest­level modules are tested first and progressively, lower­level modules are tested thereafter.

In a comprehensive software development environment, bottom­up testing is usually done first, followed by top­down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations. Table 6.5 shows the test cases for integration testing and their results

|  |  |
| --- | --- |
| Sl # Test Case : ­ | ITC­1 |
| Name of Test: ­ | Train Model |
| Item being tested: ­ | Model fit is performed |
| Sample Input: ­ | Train x and train y |
| Expected output: ­ | Fit is performed |
| Actual output: ­ | Training is done and accuracy is displayed |
| Remarks: ­ | Pass. |

|  |  |
| --- | --- |
| Sl # Test Case : ­ | ITC­2 |
| Name of Test: ­ | Accuracy calculation |
| Item being tested: ­ | If accuracy of each algorithm is calculated |
| Sample Input: ­ | Test x and test y |
| Expected output: ­ | Accuracy of each algorithm |
| Actual output: ­ | Accuracy of each model |
| Remarks: ­ | Pass. |

**System testing**:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black­box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

The application is tested thoroughly to verify that it meets the functional and technical specifications.

The application is tested in an environment that is very close to the production environment where the application will be deployed.

System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

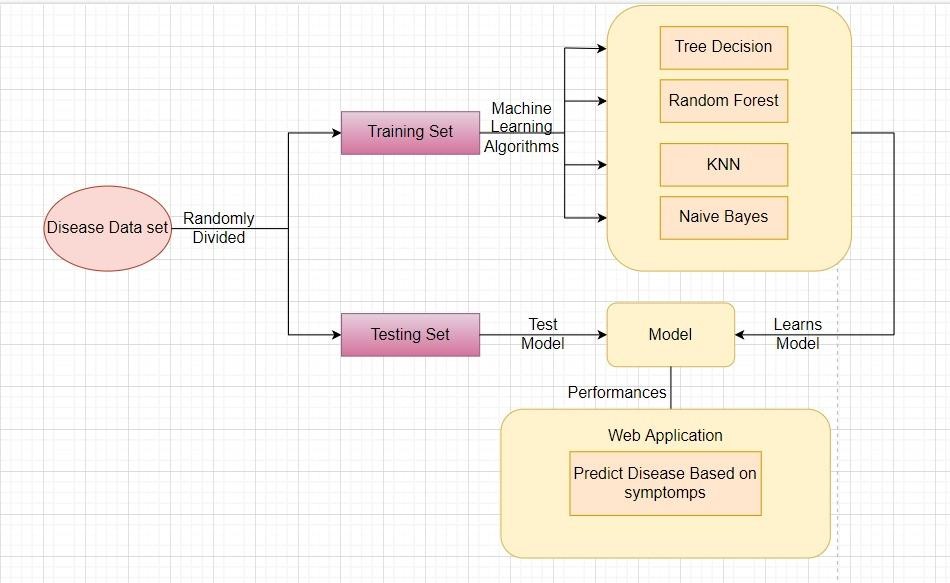
System Testing is shown in below tables

|  |  |
| --- | --- |
| Sl # Test Case : ­ | STC­1 |
| Name of Test: ­ | System testing in various versions of OS |
| Item being tested: ­ | OS compatibility. |
| Sample Input: ­ | Execute the program in windows XP/ Windows­7/8 |
| Expected output: ­ | Performance is better in windows­7 |
| Actual output: ­ | Same as expected output, performance is better in windows­7 |
| Remarks: ­ | Pass |

# CHAPTER-9

# Output Screens

* 9.1 The dataset is collected from the website called “Kaggle” and the training model is designed.
* The dataset will be randomly divided in to Training set and Testing set.
* The model is trained with the dataset and classifies the different sample test cases which have not been seen by the model yet.
* The accuracy of the model is obtained by the respective algorithms used.
* The models are taken into the web application in order to predict the outcome for given input.



CNN

### FIG.4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | We will be taking the dataset from the Kaggle. Coming to the Breast cancer Dataset we have taken the data where the total patients are 569 and, in those breast cancer Diagnosed patients are 212 in number and the remaining i.e., 357 patients are not Diagnosed with Breast Cancer. In this taken dataset, Breast Cancer Diagnosed patients are indicated with “m” which Is known for “Malign” and the patients that are not Diagnosed with cancer are indicated with “b” which is known as “Benign”.  For Breast Cancer. (Diagnosed=Malign=+, Normal=Benign=-) Total =569; Diagnosed=212; Normal =357 | | | | | | | | | | | | |  |
|  | id | dia | R\_m | T\_m | P\_m | A\_m | S\_m | C\_m | Co\_m | Co\_p\_m | Sy\_m | radius\_se | F\_m |  |
|  | 842302 | M | 17.99 | 10.38 | 122.8 | 1001 | 0.1184 | 0.2776 | 0.3001 | 0.1471 | 0.2419 | 1.095 | 0.07871 |  |
|  | 842517 | M | 20.57 | 17.77 | 132.9 | 1326 | 0.08474 | 0.07864 | 0.0869 | 0.07017 | 0.1812 | 0.5435 | 0.05667 |  |
|  | 84300903 | M | 19.69 | 21.25 | 130 | 1203 | 0.1096 | 0.1599 | 0.1974 | 0.1279 | 0.2069 | 0.7456 | 0.05999 |  |
|  | 84348301 | M | 11.42 | 20.38 | 77.58 | 386.1 | 0.1425 | 0.2839 | 0.2414 | 0.1052 | 0.2597 | 0.4956 | 0.09744 |  |
|  | 84358402 | M | 20.29 | 14.34 | 135.1 | 1297 | 0.1003 | 0.1328 | 0.198 | 0.1043 | 0.1809 | 0.7572 | 0.05883 |  |
|  | 843786 | M | 12.45 | 15.7 | 82.57 | 477.1 | 0.1278 | 0.17 | 0.1578 | 0.08089 | 0.2087 | 0.3345 | 0.07613 |  |
|  | 844359 | M | 18.25 | 19.98 | 119.6 | 1040 | 0.09463 | 0.109 | 0.1127 | 0.074 | 0.1794 | 0.4467 | 0.05742 |  |
|  | 84458202 | M | 13.71 | 20.83 | 90.2 | 577.9 | 0.1189 | 0.1645 | 0.09366 | 0.05985 | 0.2196 | 0.5835 | 0.07451 |  |
|  | 844981 | M | 13 | 21.82 | 87.5 | 519.8 | 0.1273 | 0.1932 | 0.1859 | 0.09353 | 0.235 | 0.3063 | 0.07389 |  |
|  | 84501001 | M | 12.46 | 24.04 | 83.97 | 475.9 | 0.1186 | 0.2396 | 0.2273 | 0.08543 | 0.203 | 0.2976 | 0.08243 |  |
|  | 845636 | M | 16.02 | 23.24 | 102.7 | 797.8 | 0.08206 | 0.06669 | 0.03299 | 0.03323 | 0.1528 | 0.3795 | 0.05697 |  |
|  | 84610002 | M | 15.78 | 17.89 | 103.6 | 781 | 0.0971 | 0.1292 | 0.09954 | 0.06606 | 0.1842 | 0.5058 | 0.06082 |  |
|  | 846226 | M | 19.17 | 24.8 | 132.4 | 1123 | 0.0974 | 0.2458 | 0.2065 | 0.1118 | 0.2397 | 0.9555 | 0.078 |  |
|  | 846381 | M | 15.85 | 23.95 | 103.7 | 782.7 | 0.08401 | 0.1002 | 0.09938 | 0.05364 | 0.1847 | 0.4033 | 0.05338 |  |
|  | 84667401 | M | 13.73 | 22.61 | 93.6 | 578.3 | 0.1131 | 0.2293 | 0.2128 | 0.08025 | 0.2069 | 0.2121 | 0.07682 |  |
|  | 84799002 | M | 14.54 | 27.54 | 96.73 | 658.8 | 0.1139 | 0.1595 | 0.1639 | 0.07364 | 0.2303 | 0.37 | 0.07077 |  |
|  | 848406 | M | 14.68 | 20.13 | 94.74 | 684.5 | 0.09867 | 0.072 | 0.07395 | 0.05259 | 0.1586 | 0.4727 | 0.05922 |  |
|  | 84862001 | M | 16.13 | 20.68 | 108.1 | 798.8 | 0.117 | 0.2022 | 0.1722 | 0.1028 | 0.2164 | 0.5692 | 0.07356 |  |
|  | 849014 | M | 19.81 | 22.15 | 130 | 1260 | 0.09831 | 0.1027 | 0.1479 | 0.09498 | 0.1582 | 0.7582 | 0.05395 |  |
|  | 8510426 | B | 13.54 | 14.36 | 87.46 | 566.3 | 0.09779 | 0.08129 | 0.06664 | 0.04781 | 0.1885 | 0.2699 | 0.05766 |  |
|  | 8510653 | B | 13.08 | 15.71 | 85.63 | 520 | 0.1075 | 0.127 | 0.04568 | 0.0311 | 0.1967 | 0.1852 | 0.06811 |  |
|  | 8510824 | B | 9.504 | 12.44 | 60.34 | 273.9 | 0.1024 | 0.06492 | 0.02956 | 0.02076 | 0.1815 | 0.2773 | 0.06905 |  |
|  | 8511133 | M | 15.34 | 14.26 | 102.5 | 704.4 | 0.1073 | 0.2135 | 0.2077 | 0.09756 | 0.2521 | 0.4388 | 0.07032 |  |
|  | 851509 | M | 21.16 | 23.04 | 137.2 | 1404 | 0.09428 | 0.1022 | 0.1097 | 0.08632 | 0.1769 | 0.6917 | 0.05278 |  |
|  | 852552 | M | 16.65 | 21.38 | 110 | 904.6 | 0.1121 | 0.1457 | 0.1525 | 0.0917 | 0.1995 | 0.8068 | 0.0633 |  |
|  | 852631 | M | 17.14 | 16.4 | 116 | 912.7 | 0.1186 | 0.2276 | 0.2229 | 0.1401 | 0.304 | 1.046 | 0.07413 |  |
|  | 21 | | | | | | | | | | | | |  |

The Diabetes Dataset we have taken from the Kaggle contains the data where the total patients are 768 and, in those Diabetes, diagnosed patients are 268 in number and the remaining i.e., 500 patients are not Diagnosed with Diabetes.

In this taken dataset, Diabetes Diagnosed patients are indicated with “1” and the patients that are not Diagnosed with cancer are indicated with “0”.

For Diabetes. (Diagnosed=1=+, Normal=0=-) Total=768; Diagnosed = 268; Normal =500.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pregnancies | Glucose | BP | ST | Insulin | BMI | Diabe\_ped-  fun | Age | Outcome |
| 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 |
| 3 | 78 | 50 | 32 | 88 | 31 | 0.248 | 26 | 1 |
| 10 | 115 | 0 | 0 | 0 | 35.3 | 0.134 | 29 | 0 |
| 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 8 | 125 | 96 | 0 | 0 | 0 | 0.232 | 54 | 1 |
| 4 | 110 | 92 | 0 | 0 | 37.6 | 0.191 | 30 | 0 |
| 10 | 168 | 74 | 0 | 0 | 38 | 0.537 | 34 | 1 |
| 10 | 139 | 80 | 0 | 0 | 27.1 | 1.441 | 57 | 0 |
| 1 | 189 | 60 | 23 | 846 | 30.1 | 0.398 | 59 | 1 |
| 5 | 166 | 72 | 19 | 175 | 25.8 | 0.587 | 51 | 1 |
| 7 | 100 | 0 | 0 | 0 | 30 | 0.484 | 32 | 1 |
| 0 | 118 | 84 | 47 | 230 | 45.8 | 0.551 | 31 | 1 |
| 7 | 107 | 74 | 0 | 0 | 29.6 | 0.254 | 31 | 1 |
| 1 | 103 | 30 | 38 | 83 | 43.3 | 0.183 | 33 | 0 |
| 1 | 115 | 70 | 30 | 96 | 34.6 | 0.529 | 32 | 1 |
| 3 | 126 | 88 | 41 | 235 | 39.3 | 0.704 | 27 | 0 |
| 8 | 99 | 84 | 0 | 0 | 35.4 | 0.388 | 50 | 0 |
| 7 | 196 | 90 | 0 | 0 | 39.8 | 0.451 | 41 | 1 |
| 9 | 119 | 80 | 35 | 0 | 29 | 0.263 | 29 | 1 |
| 11 | 143 | 94 | 33 | 146 | 36.6 | 0.254 | 51 | 1 |
| 10 | 125 | 70 | 26 | 115 | 31.1 | 0.205 | 41 | 1 |

Coming to the Kidney Disease Dataset we have taken the data where the total patients are 400 and, in those Kidney, Disease Diagnosed patients are 250 in number and the remaining i.e., 150 patients are not Diagnosed with Kidney Disease.

In this taken dataset, Kidney Disease Diagnosed patients are indicated with “Ckd” which is a representation for “chronic kidney disease” and the patients that are not Diagnosed with Kidney Disease are indicated with “Notckd” which is a representation for “No Chronic Disease.”

For Kidney Disease. (Diagnosed= Ckd=+, Normal= Notckd=-) Total =400; Diagnosed =250; Normal=150

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | age | bp | sg | al | bu | sc | hemo | classification |
| 0 | 48 | 80 | 1.02 | 1 | 36 | 1.2 | 15.4 | ckd |
| 1 | 7 | 50 | 1.02 | 4 | 18 | 0.8 | 11.3 | ckd |
| 2 | 62 | 80 | 1.01 | 2 | 53 | 1.8 | 9.6 | ckd |
| 3 | 48 | 70 | 1.005 | 4 | 56 | 3.8 | 11.2 | ckd |
| 4 | 51 | 80 | 1.01 | 2 | 26 | 1.4 | 11.6 | ckd |
| 5 | 60 | 90 | 1.015 | 3 | 25 | 1.1 | 12.2 | ckd |
| 6 | 68 | 70 | 1.01 | 0 | 54 | 24 | 12.4 | ckd |
| 7 | 24 |  | 1.015 | 2 | 31 | 1.1 | 12.4 | ckd |
| 8 | 52 | 100 | 1.015 | 3 | 60 | 1.9 | 10.8 | ckd |
| 9 | 53 | 90 | 1.02 | 2 | 107 | 7.2 | 9.5 | ckd |
| 10 | 50 | 60 | 1.01 | 2 | 55 | 4 | 9.4 | ckd |
| 11 | 63 | 70 | 1.01 | 3 | 60 | 2.7 | 10.8 | ckd |
| 12 | 68 | 70 | 1.015 | 3 | 72 | 2.1 | 9.7 | ckd |
| 13 | 68 | 70 | 1.005 | 2 | 86 | 4.6 | 9.8 | ckd |
| 14 | 68 | 80 | 1.01 | 3 | 90 | 4.1 | 5.6 | ckd |
| 15 | 40 | 80 | 1.015 | 3 | 162 | 9.6 | 7.6 | ckd |
| 16 | 47 | 70 | 1.015 | 2 | 46 | 2.2 | 12.6 | ckd |
| 17 | 47 | 80 | 1.05 | 1 | 87 | 5.2 | 12.1 | ckd |
| 18 | 60 | 100 | 1.025 | 0 | 27 | 1.3 | 12.7 | ckd |
| 19 | 62 | 60 | 1.015 | 1 | 31 | 1.6 | 10.3 | ckd |
| 20 | 61 | 80 | 1.015 | 2 | 148 | 3.9 | 7.7 | ckd |
| 21 | 60 | 90 | 1.02 | 3 | 180 | 76 | 10.9 | ckd |
| 22 | 48 | 80 | 1.025 | 4 | 163 | 7.7 | 9.8 | ckd |
| 23 | 21 | 70 | 1.01 | 0 | 57 | 3.5 | 10.5 | ckd |
| 24 | 42 | 100 | 1.015 | 4 | 50 | 1.4 | 11.1 | ckd |
| 25 | 61 | 60 | 1.025 | 0 | 75 | 1.9 | 9.9 | ckd |

And after collecting all the datasets the data will be divided in to the Training and Test sets and the Training set will be get trained to the particular Algorithm that we choose. We will be using

different types of machine learning algorithms under the supervised learning category. then we need the will be getting their accuracies and after that we will be comparing the Training Data and the Test data in-order to get the results.

In the Liver Disease Dataset, we have taken the data where the total patients are 583 and, in those Liver, Disease Diagnosed patients are 167 in number and the remaining i.e., 416 patients are not Diagnosed with Liver Disease.

In this taken dataset, Liver Disease Diagnosed patients are indicated with “2” and the patients that are not Diagnosed with Liver Disease are indicated with “1”.

For Liver Disease. (Diagnosed= 2=+, Normal= 1=-) Total =583; Diagnosed =167; Normal=416

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Age | Gender | T\_B | D\_B | A\_P | A\_A | A\_Am | Total\_Protiens | Albumin | A\_G | Dataset |
| 65 | Female | 0.7 | 0.1 | 187 | 16 | 18 | 6.8 | 3.3 | 0.9 | 1 |
| 62 | Male | 10.9 | 5.5 | 699 | 64 | 100 | 7.5 | 3.2 | 0.74 | 1 |
| 62 | Male | 7.3 | 4.1 | 490 | 60 | 68 | 7 | 3.3 | 0.89 | 1 |
| 58 | Male | 1 | 0.4 | 182 | 14 | 20 | 6.8 | 3.4 | 1 | 1 |
| 72 | Male | 3.9 | 2 | 195 | 27 | 59 | 7.3 | 2.4 | 0.4 | 1 |
| 46 | Male | 1.8 | 0.7 | 208 | 19 | 14 | 7.6 | 4.4 | 1.3 | 1 |
| 26 | Female | 0.9 | 0.2 | 154 | 16 | 12 | 7 | 3.5 | 1 | 1 |
| 29 | Female | 0.9 | 0.3 | 202 | 14 | 11 | 6.7 | 3.6 | 1.1 | 1 |
| 17 | Male | 0.9 | 0.3 | 202 | 22 | 19 | 7.4 | 4.1 | 1.2 | 1 |
| 55 | Male | 0.7 | 0.2 | 290 | 53 | 58 | 6.8 | 3.4 | 1 | 1 |
| 57 | Male | 0.6 | 0.1 | 210 | 51 | 59 | 5.9 | 2.7 | 0.8 | 1 |
| 72 | Male | 2.7 | 1.3 | 260 | 31 | 56 | 7.4 | 3 | 0.6 | 1 |
| 64 | Male | 0.9 | 0.3 | 310 | 61 | 58 | 7 | 3.4 | 0.9 | 1 |
| 74 | Female | 1.1 | 0.4 | 214 | 22 | 30 | 8.1 | 4.1 | 1 | 1 |
| 61 | Male | 0.7 | 0.2 | 145 | 53 | 41 | 5.8 | 2.7 | 0.87 | 1 |
| 25 | Male | 0.6 | 0.1 | 183 | 91 | 53 | 5.5 | 2.3 | 0.7 | 1 |
| 38 | Male | 1.8 | 0.8 | 342 | 168 | 441 | 7.6 | 4.4 | 1.3 | 1 |
| 33 | Male | 1.6 | 0.5 | 165 | 15 | 23 | 7.3 | 3.5 | 0.92 | 1 |
| 40 | Female | 0.9 | 0.3 | 293 | 232 | 245 | 6.8 | 3.1 | 0.8 | 1 |
| 40 | Female | 0.9 | 0.3 | 293 | 232 | 245 | 6.8 | 3.1 | 0.8 | 1 |
| 51 | Male | 2.2 | 1 | 610 | 17 | 28 | 7.3 | 2.6 | 0.55 | 1 |
| 51 | Male | 2.9 | 1.3 | 482 | 22 | 34 | 7 | 2.4 | 0.5 | 1 |

In the taken Heart Disease Dataset it contains total patients of 397 and in those Heart, Disease Diagnosed patients are 160 in number and the remaining i.e., 137 patients are not Diagnosed with Heart Disease.

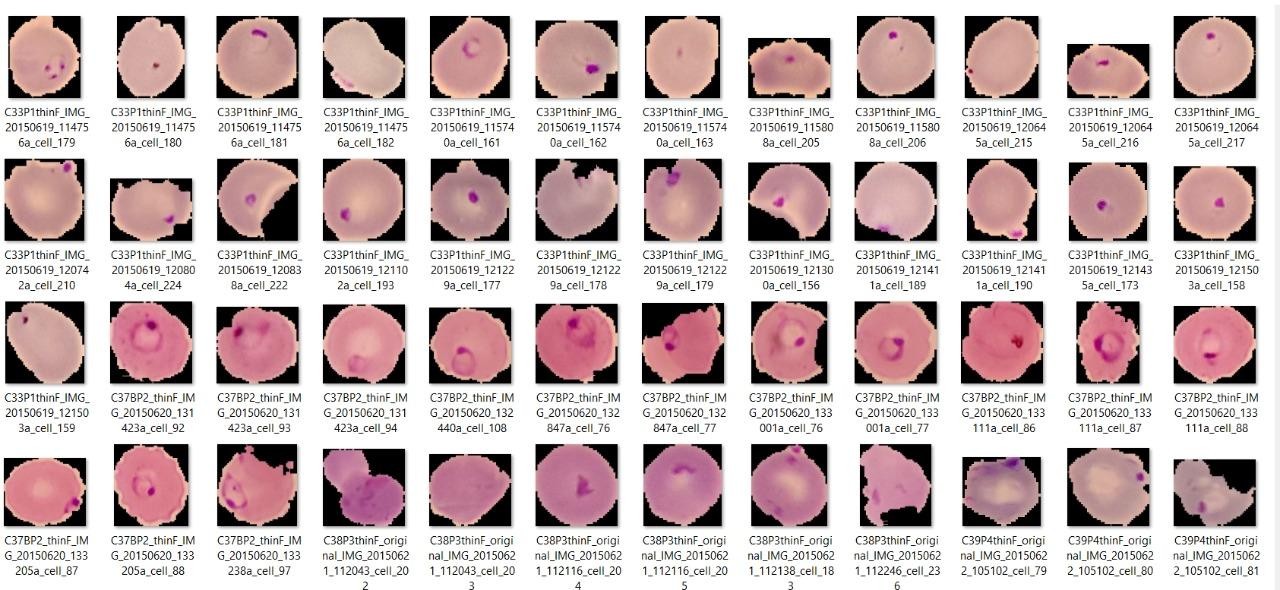
In this taken dataset, Heart Disease Diagnosed patients are indicated with “0” and the patients that are not Diagnosed with Heart Disease are indicated with “1”.

For Heart Disease. (Diagnosed= 0 =+, Normal= 1=-) Total =397; Diagnosed =160; Normal=137

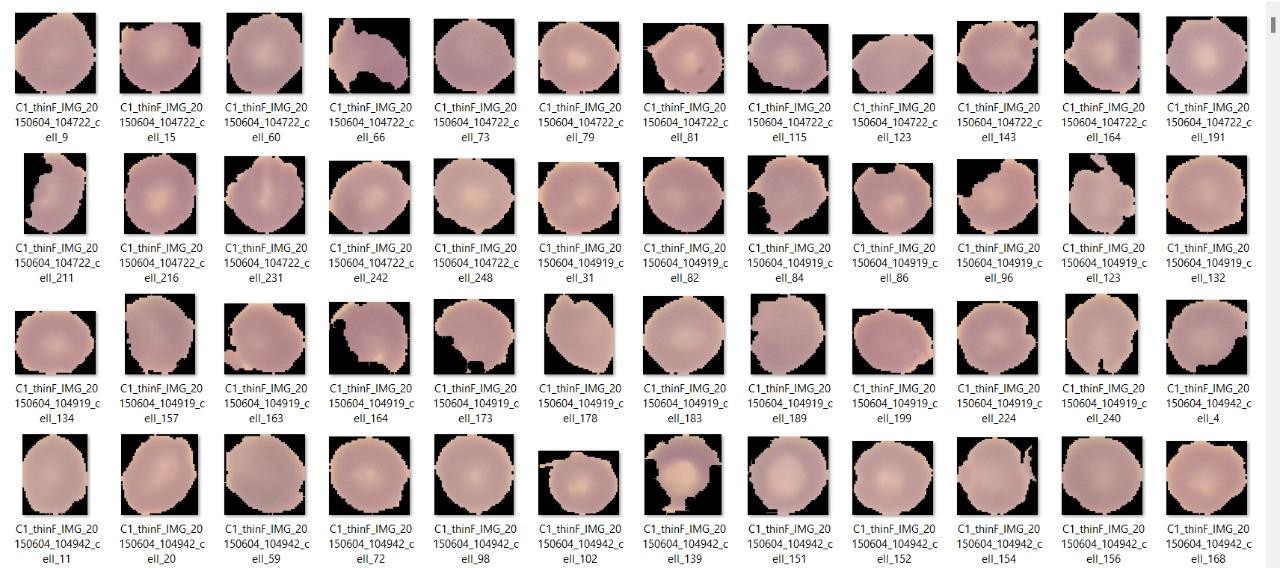
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| age | sex | c | t | c | f | r | t | ex | oldpeak | slope | ca | thal | condition |
| 69 | 1 | 0 | 160 | 234 | 1 | 2 | 131 | 0 | 0.1 | 1 | 1 | 0 | 0 |
| 69 | 0 | 0 | 140 | 239 | 0 | 0 | 151 | 0 | 1.8 | 0 | 2 | 0 | 0 |
| 66 | 0 | 0 | 150 | 226 | 0 | 0 | 114 | 0 | 2.6 | 2 | 0 | 0 | 0 |
| 65 | 1 | 0 | 138 | 282 | 1 | 2 | 174 | 0 | 1.4 | 1 | 1 | 0 | 1 |
| 64 | 1 | 0 | 110 | 211 | 0 | 2 | 144 | 1 | 1.8 | 1 | 0 | 0 | 0 |
| 64 | 1 | 0 | 170 | 227 | 0 | 2 | 155 | 0 | 0.6 | 1 | 0 | 2 | 0 |
| 63 | 1 | 0 | 145 | 233 | 1 | 2 | 150 | 0 | 2.3 | 2 | 0 | 1 | 0 |
| 61 | 1 | 0 | 134 | 234 | 0 | 0 | 145 | 0 | 2.6 | 1 | 2 | 0 | 1 |
| 60 | 0 | 0 | 150 | 240 | 0 | 0 | 171 | 0 | 0.9 | 0 | 0 | 0 | 0 |
| 59 | 1 | 0 | 178 | 270 | 0 | 2 | 145 | 0 | 4.2 | 2 | 0 | 2 | 0 |
| 59 | 1 | 0 | 170 | 288 | 0 | 2 | 159 | 0 | 0.2 | 1 | 0 | 2 | 1 |
| 59 | 1 | 0 | 160 | 273 | 0 | 2 | 125 | 0 | 0 | 0 | 0 | 0 | 1 |
| 59 | 1 | 0 | 134 | 204 | 0 | 0 | 162 | 0 | 0.8 | 0 | 2 | 0 | 1 |
| 58 | 0 | 0 | 150 | 283 | 1 | 2 | 162 | 0 | 1 | 0 | 0 | 0 | 0 |
| 56 | 1 | 0 | 120 | 193 | 0 | 2 | 162 | 0 | 1.9 | 1 | 0 | 2 | 0 |

At last, coming to the Malaria Disease, in this take dataset it totally contains of 27,559 images combined of both Parasitized and Uninfected and in those Parasitized are “13,779” and the Uninfected are “13,780”.

Now we will be showing a glympse of the dataset of Parasitized and Uninfected.



### FIG. 5



**FIG. 6**

For Malaria Disease.

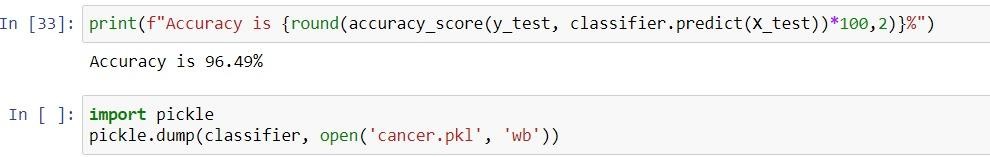
Total =27,558; Parasitized =13,779; Normal= 13,780

# Result and Discussion

**For Breast Cancer.**

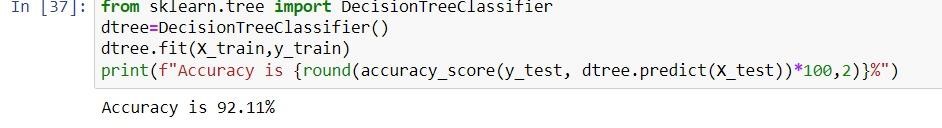
Here we will be writing the particular code for Breast cancer of their particular Algorithms in order to calculate their accuracies. These Code will be written in the Jupyter Notebook.

# Random Forest: -



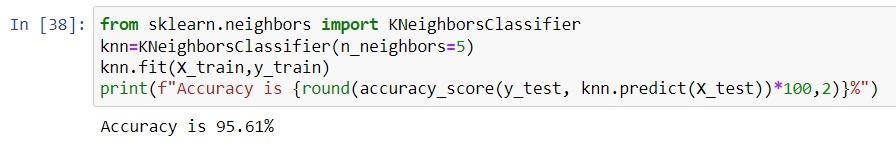
**FIG. 7**

# Decision Tree: -



**KNN: -**

### FIG. 8

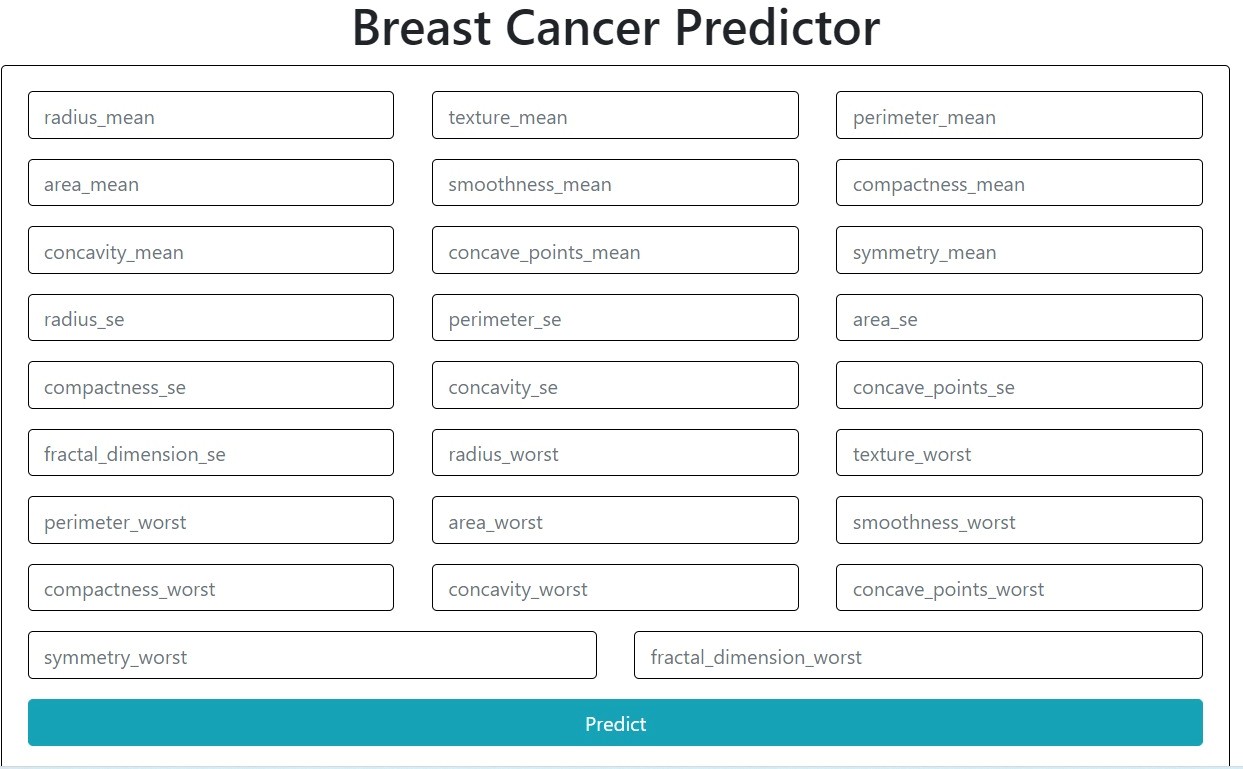


**FIG. 9**

# Naïve Bayes: -

### FIG. 10

And after that we will be aiming ourselves to the Flask to gain the results. We will also develop the code in Jupyter Notebook for the particular appearances that we want in Flask.



### FIG. 11

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with that Breast Cancer or not.

These parameters can be taken from the dataset or our Own.

We will be doing the same thing for the all-other disease that we want to predict.

# For Diabetes

Same as what we did for the Breast Cancer here also, we will be writing the particular code for Diabetes of their particular Algorithms in order to calculate their accuracies. These Code will be written in the Jupyter Notebook.

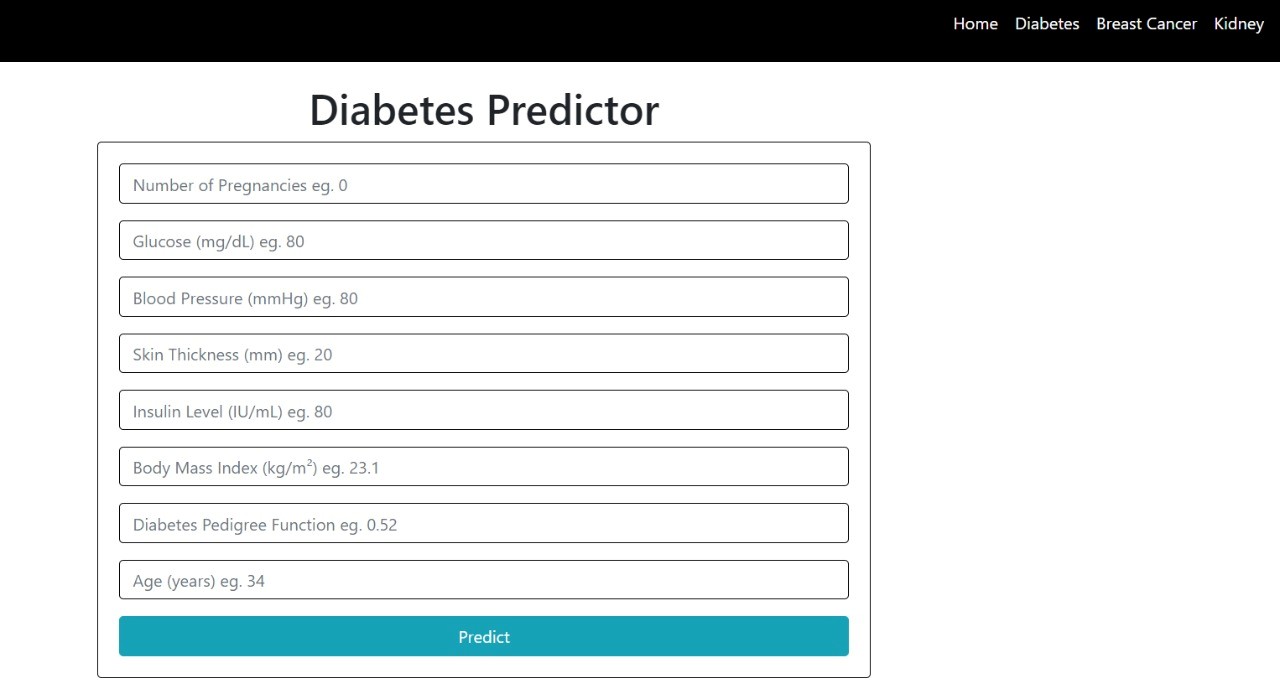
The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 36, 38, 39 and 40 respectively in the below given Figure.



### FIG.12

As we can see in the Above Figure that it shows for the Random Forest Algorithm shows the highest accuracy comparing to the all-other Algorithms.

And after that we will be aiming ourselves to the Flask to gain the results



### FIG.13

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with Diabetes or not. These parameters can be taken from the dataset or our Own.

# For Kidney Disease

Same as what we did for the Above Diseases here also, we will be writing the particular code for kidney disease for their particular Algorithms in order to calculate their accuracies. These Code will be written in the Jupyter Notebook.

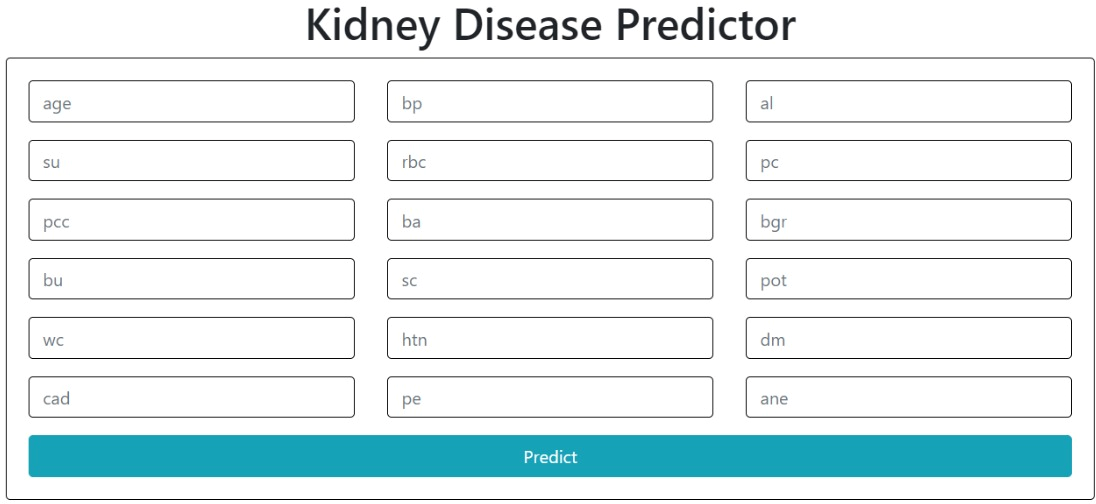
By Comparing the all accuracies, we conclude that the Random Forest Algorithm attains the best accuracy. So, we will be writing only for random forest algorithm.

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 35, 37, 38 and 39 respectively in the below given Figure.



### FIG. 14

And after that we will be aiming ourselves to the Flask to gain the results.



### FIG.15

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with kidney disease or not. These parameters can be taken from the dataset or our Own.

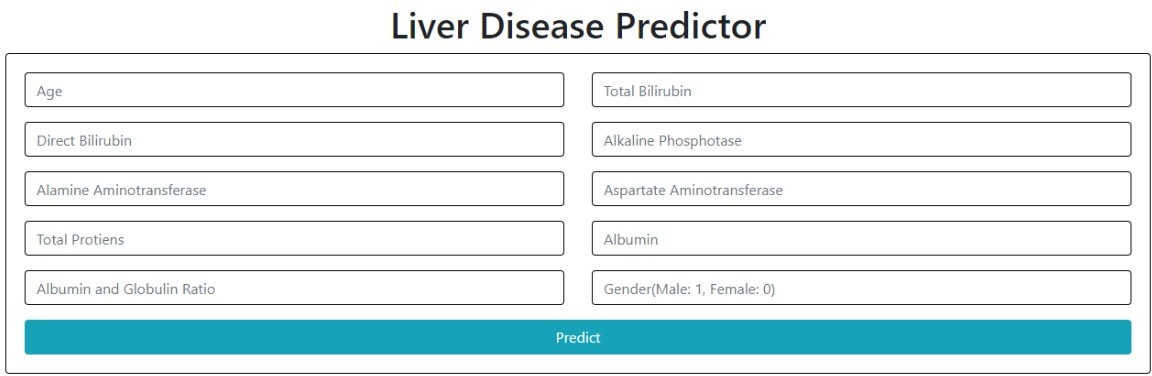
# For Liver Disease

Coming to the Liver Disease we will be writing the code we will be writing the particular code for their particular Algorithms in Jupyter Notebook in order to calculate their accuracies.



### FIG. 16

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 27, 29, 30 and 31 respectively in the below given Figure.



### FIG.17

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with Liver Disease or not. These parameters can be taken from the dataset or our Own.

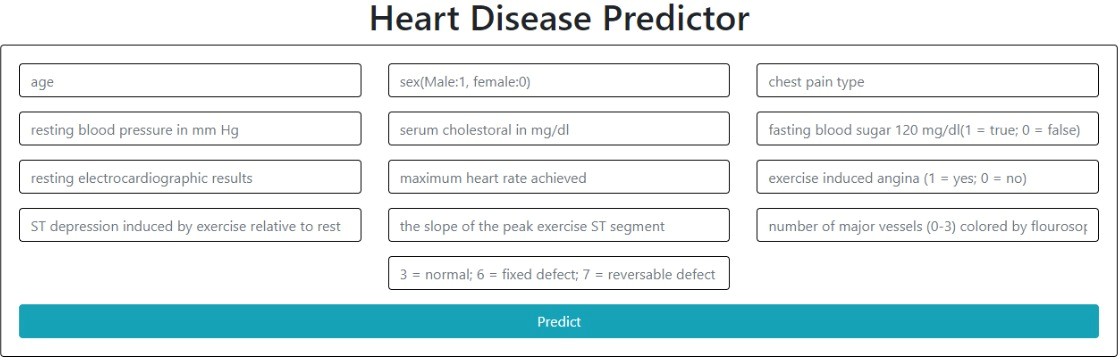
# For Heart Disease

For Heart Disease also like same as the others we will be writing the code we will be writing the particular code for their particular Algorithms in Jupyter Notebook in order to calculate their accuracies.



### FIG. 18

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 29, 31, 32 and 33 respectively in the below given Figure.

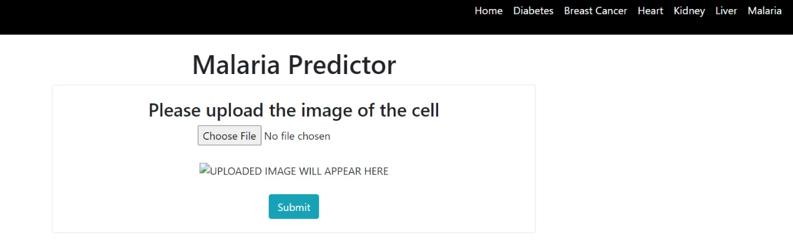


### FIG.19

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with heart disease or not. These parameters can be taken from the dataset or our Own.

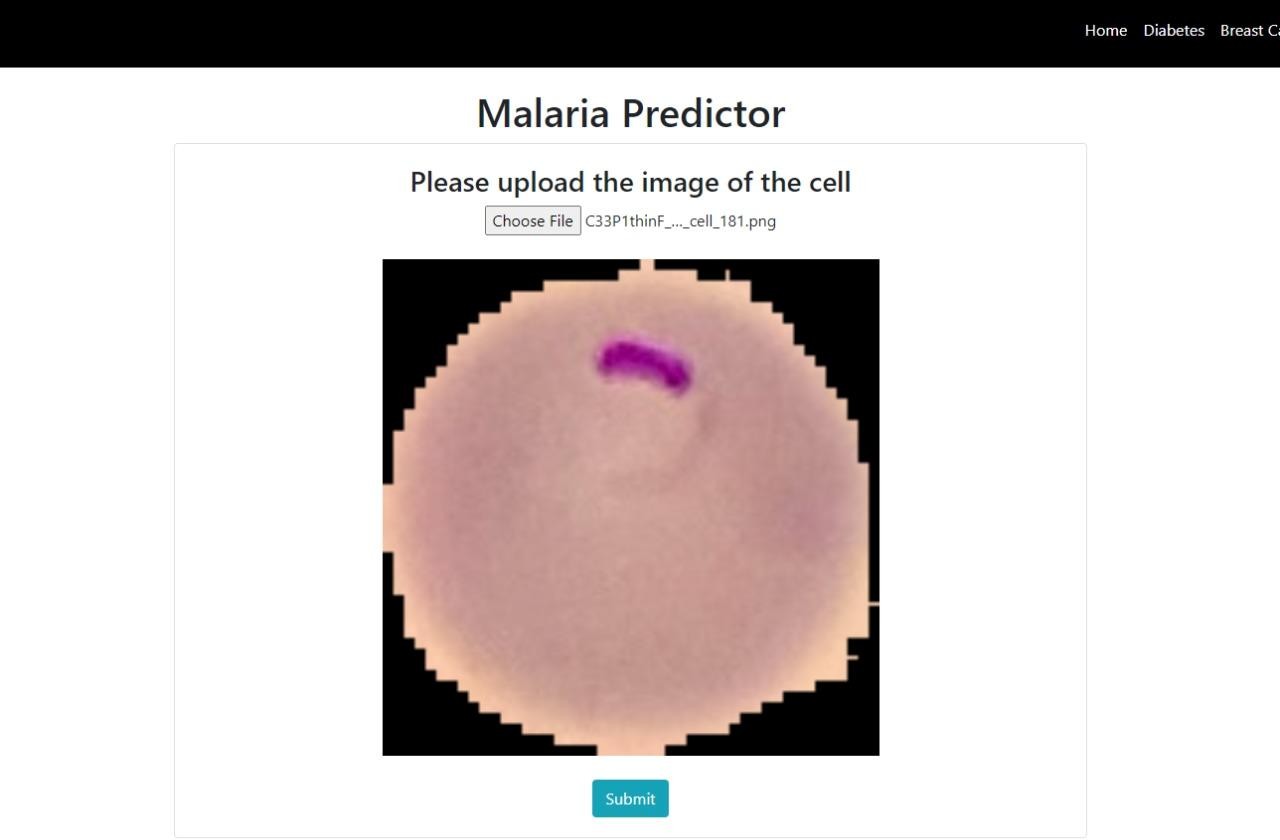
# For Malaria Disease

Coming to the Malaria Disease it is a bit different cause unlike all the other disease we will be using Deep Learning for this. So, the all collected data i.e., the image data will be divided into the training set and testing set and after we train the training data, we will compare that data to the Testing data inorder to know whether the patient is diagnosed with the Malaria Disease or not.

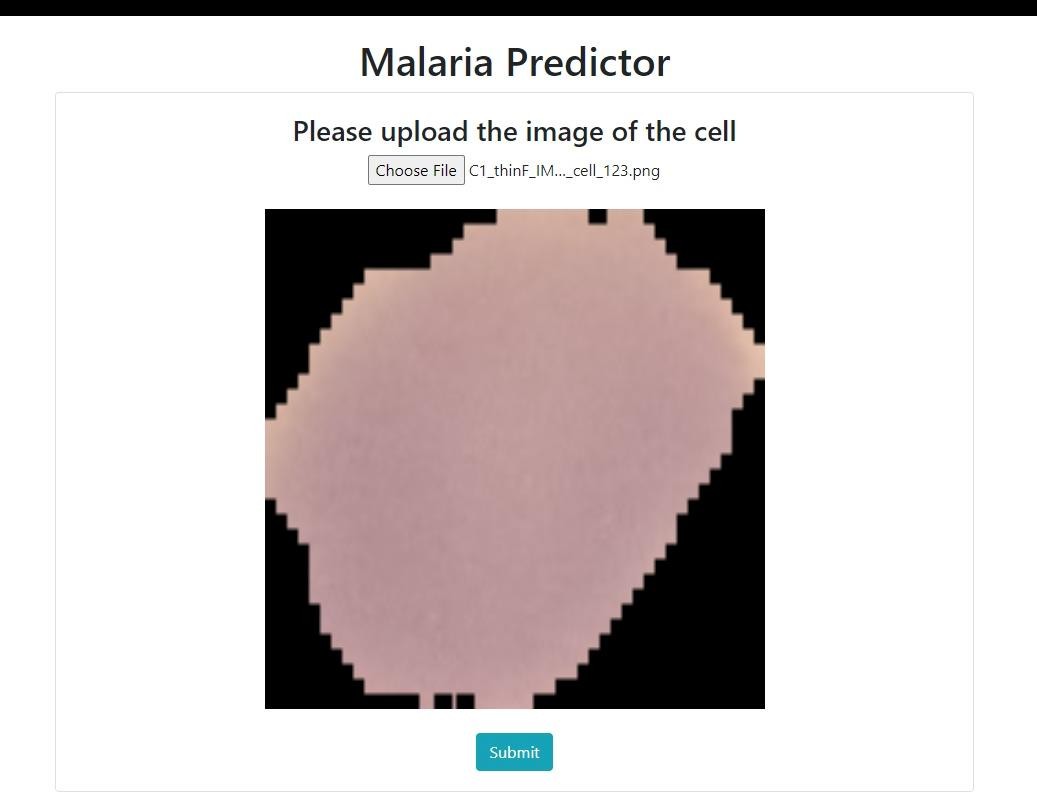


### FIG. 20

In the given Above Figure we need to upload the image of the cell of a particular person whom we want to check whether he is diagnosed with malaria disease or not at the Given choose File option.



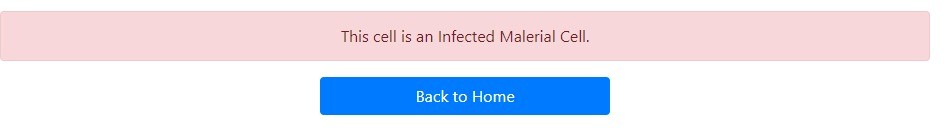
### FIG. 21



**FIG. 22**

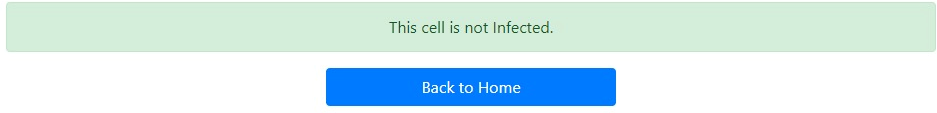
After uploading the image, we will get to see the image that we uploaded in the website as shown in the above Figure.

In The Above Figures, Fig .no is Infected with the malaria disease so the result will be represented like as figure given below.



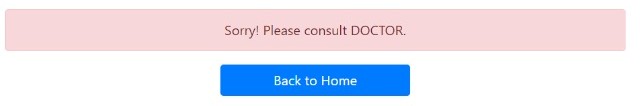
### FIG. 23

In The Above Figures, Fig .no is not Infected with the malaria disease so the result will be represented like as figure given below.



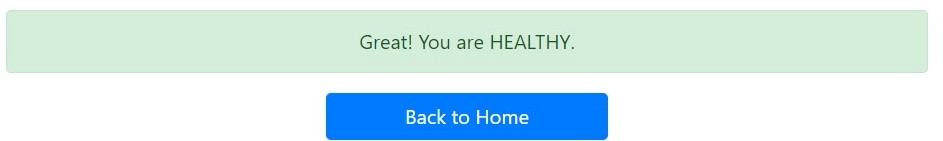
### FIG. 24

When the Particular Patient is Diagnosed with the Disease except the Malaria Disease then It will represent like the below given Figure in the Flask

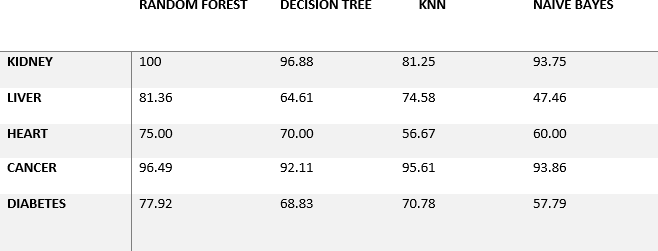


### FIG. 25

When the Particular Patient is not Diagnosed with the Disease Excluding the Malaria Disease i.e., the Person is in Healthy Condition then It will represent like the below given Figure in the Flask.



### FIG. 26



**FIG. 27**

# CONCLUSION and Future work

.

Multi disease prediction model is used to predict multiple diseases at a time. Here based on the user input disease will be predicted. The choice will be given to user. If the user want to predict particular disease or if the user don’t enter any disease type then based on user entered inputs corresponding disease model will be invoked and predicted. The advantage of multi disease prediction model in advance can predict the probability of occurrence of various disease and also can reduce mortality ratio.

In future work live data can be used in as input for predicting disease by integration hardware to detect heal condition of patient.

# REFERENCES

[1] Khurana, Sarthak . , Jain, Atishay ., Kataria ,Shikhar. ,Bhasin ,Kunal . , Arora ,Sunny . ,& Gupta , Dr.Akhilesh . Das. (2019). Disease Prediction System.International Research Journal Of Engineering and Technology , 6(5) , 5178-5184.

[2] Kamboj ,Mgha. (2020).Heart Disease Prediction with Machine Learning Approaches.International Journal Of Science and Research , 9(7) , 1454-1458.

[3]Ware,Miss.Sangya . , Rakesh,Mrs.Shanu. K.,&Choudhary,Mr.Bharat . (2020). Heart Attack Prediction By Using Machine Learning Techniques. International Journal Of Recent Technology and Engineering , 8(5), 1577- 1580. Turkish Journal of Computer and Mathematics Education \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4023 Research Article Vol.12 No.6 (2021), 4013-4023

[4] Shirsath ,Shraddha.Subhash .,& Patil , Prof. Shubhangi . (2018).Disease Prediction Using Machine Learning over Big Data .International Journal Of Innovative Research in Science and Technology , 7(6), 6752-6757

[5]Marimuthu , M. , Abinaya, M. ,Hariesh,K.S., Madhan,K.,& Pavithra, Kumar. V.(2018).A Review of Heart Disease Prediction Using Machine Learning and Data Analytics Approach .International Journal of Computer Application , 181(18), 20-25.

[6] Battineni ,Gopi. , Sagaro,Getu.Gamo. ,Chinatalapudi, Nalini. ,&Amenta,Francesco. (2020). Application Of Machine Learning Predictive Models in the Chronic Disease .International of PersonalisedMedicine , 10(21), 1- 11

7. Parsa, A.B.; Chauhan, R.S.; Taghipour, H.; Derrible, S.; Mohammadian, A. Applying Deep Learning to Detect Traffic Accidents in Real Time Using Spatiotemporal Sequential Data. arXiv 2019, arXiv:1912.06991.

8. Joshua, S.C.; Garber, N.J. Estimating Truck Accident Rate and Involvements Using Linear and Poisson Regression Models. Transp. Plan. Technol. 1990, 15, 41–58. [CrossRef]

9. Arvin, R.; Kamrani, M.; Khattak, A.J. How Instantaneous Driving Behavior Contributes to Crashes at Intersections: Extracting Useful Information from Connected Vehicle Message Data. Accid. Anal. Prev. 2019, 127, 118–133. [CrossRef] [PubMed] [12] Sh. Ataei et al. ‘Sensor fusion of a railway bridge load test using neural networks’. In: Expert Syst. Appl. 29 (3 2005), pp. 678–683.

[13] A.S. Zaher and S.D.J. McArthur. ‘A Multi-Agent Fault Detection System for Wind Turbine Defect Recognition and Diagnosis’. In: 2007 IEEE Lausanne Powertech. 2007.

[14] Alice E. Smith, David W. Coit and Yun-chia Liang. A Neural Network Approach to Condition Based Maintenance: Case Study of Airport Ground Transportation Vehicles.

[15] Mark Owen. Practical Signal Processing. Cambridge University Press, 2007.