University of Mumbai

HealthCare Web Application Using Machine Learning

Submitted at the end of semester VIII in fulfillment of requirements

For the degree of

Bachelors in Electronics and Telecommunications

by

Nakul Chamariya
Roll No: 1813068

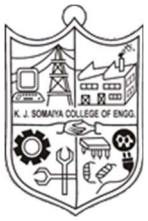
Rahul Doshi
Roll No: 1813074

Merediya Shabdarali
Roll No: 1813101

Smit Mehta
Roll No: 1813104

Guide

Prof. Deepak Kulkarni



Department of Electronics and Telecommunication Engineering

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **HealthCare Web Application** is bonafide record of the dissertation work done by **Nakul Chamariya, Rahul Doshi, Marediya Shabdarali** and **Smit Mehta** in the year 2021-22 under the guidance of Prof. Deepak Kulkarni of Department of Electronics and Telecommunication Engineering in partial fulfillment of requirement for the Bachelor of Technology degree in Electronics and Telecommunication Engineering of University of Mumbai.

_____ _____

Guide Head of the Department

_____

Principal

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **HealthCare Web Application** is bonafide record of project work done by **Nakul Chamariya, Rahul Doshi, Marediya Shabdarali** and **Smit Mehta** during semester VIII.

This project is approved for the award of Bachelor of Technology Degree in Electronics and Telecommunication Engineering of University of Mumbai.

_____

Internal Examiner

_____

External Examiner

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written thesis submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student Nakul Chamariya **Roll No. 1813068**	**Signature of the Student** Rahul Doshi **Roll No. 1813074**
Signature of the Student Marediya Shabdarali **Roll No. 1813101**	**Signature of the Student** Smit Mehta **Roll No. 1813104**

Date:

Place: Mumbai-77

Dedicated to

My family and friends

Abstract

With the improvement of AI, data innovation, the idea of savvy medical care has bit by bit come to the front. Savvy medical services utilizes another age of data innovations, like the web of things (IOT), AI, large information, distributed computing, and man-made reasoning, to change the customary clinical framework in an inside and out manner, making medical care more productive, more advantageous, and more customized.

AI is tracking down a wide scope of medical services applications, going from case the board of common persistent sicknesses to utilizing patient wellbeing information, related to outside impacts like contamination openness and weather conditions factors.

By crunching enormous volumes of information, AI innovation can assist medical services experts with creating exact medication arrangements tweaked to individual attributes. AI and AI is supposed to assume a basic part in Central Nervous System (CNS) clinical preliminaries later on, as per a report in the Mercury News. The innovation is supposed to assist with crunching CNS infection treatment information gathered over the long haul in clinical preliminaries to assist with estimating the reaction by patients.

Other potential AI improvements in medical care incorporate investigating ways of involving the innovation in telemedicine, the report notes. AI organizations are concentrating on the capacity to arrange and furnish specialists with patient data during a telemedicine meeting, as well as catching data during the virtual visit to help with expanded productivity and work process.

Contents

List of Figures

CHAPTER 1

Introduction

This chapter present about the background of Healthcare Web application, its relation with machine learning, types of system, technologies used. Further we discussed about motivation and scope of the project. And finally, gave brief description about the project

1.1 Background

Ascend in the area of innovation AI is broadly utilized in different fields. Presently it has different applications on the field of wellbeing industry. It fills in as some assistance for the field of wellbeing industry. By the assistance of different AI calculations, we can make different models for foreseeing the outcomes through the huge measure of dataset present in clinical field. This paper includes effective AI calculations utilized in anticipating illness through side effects. As, the wellbeing business has an immense measure of information for different fields in this way, we need to create a framework where we can utilize different uses of AI on wellbeing industry. This all had been done to pursue the better clinical choices and furthermore for ascend in the accuracy.[1] As exact examination of the early expectation of sickness helps in the patient consideration and the general public administrations. These everything difficulties can be more straightforward by the assistance of different apparatuses, calculations and system given by the AI. Notwithstanding this multitude of expectations, we are making a chatbot for everything that could be added the side effects that are useful to anticipate the illness and furthermore check their diabetes status through the different data gave to framework by the patients.[2]

1.2 Motivation

The interest for medical care administrations is truly expanding and numerous nations are encountering a lack of medical care experts, particularly doctors. Medical care establishments are likewise battling to stay aware of the multitude of new innovative turns of events and the exclusive standards of patients regarding levels of administration and results as far as they might be concerned from purchaser items including those of Amazon and Apple. The advances in remote innovation and cell phones have given chances to on-request medical care administrations utilizing wellbeing following applications and search stages and have additionally empowered another type of medical care conveyance, by means of far off communications, accessible anyplace and whenever. Such administrations are pertinent for underserved locales and spots lacking trained professionals and assist with diminishing expenses and forestall superfluous openness to infectious sicknesses at the center. Telehealth innovation is additionally pertinent in agricultural nations where the medical care framework is growing and where medical care foundation can be intended to meet the ongoing requirements. While the idea is clear, these arrangements actually need significant autonomous approval to demonstrate patient security and efficacy.[3]

1.3 Scope of the Project

First decide the major constant infections in the area. Close to deal with organized information, talk with emergency clinic specialists to remove valuable highlights. For unstructured text information, choosing the elements consequently utilizing CNN algorithm.[2] Finally, a clever CNN-based calculation for organized and unstructured information is proposed. The illness risk model is acquired by the blend of both organized and unstructured elements. The attributes are chosen through experience. Nonetheless, these pre-chosen qualities perhaps not fulfill the progressions in the

sickness and its impacting factors. With the advancement of huge information examination innovation, more consideration has been paid to illness expectation according to the viewpoint of enormous information investigation, different explores have been directed by choosing the attributes consequently from countless information to work on the exactness of hazard arrangement as opposed to the recently chosen characteristics.[6] The educational record is pretty much nothing, for patients and afflictions with specific circumstances; the characteristics are picked through understanding. Anyway, these pre-picked credits potentially not satisfy the changes in the disease and its affecting variables. With the headway of enormous data assessment development, more thought has been paid to contamination assumption according to the perspective of colossal data examination, different investigates have been coordinated by picking the properties normally from a broad number of data to upgrade the accuracy of risk request, rather than the in advance picked characteristics. Nevertheless, that ongoing work commonly remembered to be coordinated data. For unstructured data, for example, using Convolutional Neural Network (CNN) to isolate content characteristics subsequently has quite recently pulled in wide thought and moreover refined extraordinary outcomes.[3]

1.4 Brief Description of Project Undertaken

The wellbeing expectation framework is an end client support and online meeting project. Here, we propose a framework that permits client to help moment direction on their medical problems through a clever medical care framework on the web. The framework is taken care of with different side effects and different sicknesses/ailment related with those side effects. The framework permits client to share their side effects and issues. It then, at that point, processes client side effects to check for Body Mass Index (BMI), pulse and different infections related with the side effects of the client. On the off chance that client side effects match any sickness in the information base, it shows the likely infection client could have.

Approach:

a) For Pneumonia Prediction
    1) Collect the x-ray image
    2) Convert it to greyscale image
    3) Scale the image
    4) Pass the image to the CNN model
    5) Print the result
b) For diabetes, kidney, liver, heart prediction
    1) Collect the necessary information from the patient.
    2) Create a data frame of all these inputs
    3) Convert it to NumPy array
    4) Scale the data
    5) Pass the data to the model
    6) Print the result

Compatibility:

The platform should be compatible with multiple browsers including:

- Chrome version 87.0.4280.141
- Firefox version 84
- Safari version 11.1.2
- Internet Explorer 11.0

Availability:

Application is available for the users all the time to test their health based issues and give them the right course of action.

Performance:

- The system is lag free and fast for best user experience.
- Application is easy to understand and more user-friendly.

1.5 Organization of the Project

We elaborated till now all the essential parts of our system, its scope, motivation, requirements, compatibility and performance of the project.

In the follow up part we have discussed the background details regarding the program that were going on continuously throughout the course of our project. We discuss about the technologies, programming languages that we have used to create the Web Application. Nonetheless we also discuss about the methodology to our approach, algorithms that we have used to make the machine learning more robust. We have presented the implementation of various web pages. We later on discuss about our results and conclude with our learning and describe the scope for future work. And References at the end.

CHAPTER 2

Literature Survey

This chapter presents about the past work on this project and research done in area of machine learning and its applications in healthcare.

In this paper [1] author has presented the concept namely, "Disease prediction using Machine Learning over Big Data". The big data is fastest concept in current trend, so this concept is applied in more fields. The big data is most widely used in every field because it is very large. The big data is applied in medical field both side developing the better growth in both fields, that is big data is applied in medical fields develops the medical fields at the same time increase the growth in big data field. The big data helps to achieve the better growth in medical and health care sectors. It additionally, provides the more merits gives, (i) medical data analysis with accuracy, (ii) early prediction for disease, (iii) patient oriented data with accuracy, (iv) The medical data, is securely stored and used in many places, (v) incomplete regional data are reduced and give the accuracy result. Goal of the concept is choosing the region and collects the hospital data or medical data of particular selected region; this process is using the machine learning algorithm. This term based on the data mining technique is used for disease prediction with accuracy. Then, finding the missing data based on latent factor get the incomplete data and it is reduced. The previous system uses the CNN-UDRP (Unimodal Disease Risk Prediction), then continuously implements the next level use the CNN-MDRP (Multimodal Disease Risk Prediction). The CNN-MDRP is overcome the drawback of CNN-UDRP.

The CNN-MDRP is uses the hospital data, that is structured and unstructured data. The CNN-MDRP algorithm based prediction is produce more accurate, this accuracy is compared with previous system. The advantages of the concept are, better feature description and better accuracy, and the disadvantages of this system is, this feature is

only applicable for the structured data so it is not good in disease description.

The paper [2] author has presented the data mining concept "Disease Prediction by using Machine Learning". The data mining best growth of the stage is developing that technique into the healthcare basis, the data analysis is an important part of every field. The data mining is predicting the information for healthcare is called rapid growth of medical care field. The existing one is designed the purpose of (i) analyze, (ii) manage, (iii) predict of healthcare data, it is described the overall healthcare systems. The concept of machine learning is applied into the disease-related information retrievals and the treatment processes in these types of process are achieved by using the data analysis. The predictions of outbreaks in diseases are using the decision tree, because it is very effective. This concept based experimental shows that result is related to the disease symptoms, so that data is described medical data using modified prediction model. If the concept chooses the raining set like medical patient symptoms, then, use the decision tree, then, predicted, finally give the symptoms of patient and get the accurate result for disease prediction. This concept only performs, that is predicts only the patient related information with low time and low cost.

Authors, presents the survey paper [3] for "prediction of disease using machine learning over big data". Can develop the medical specialty basis this concept is applied to produce the medical data in to mass medical data, which means the data which is enlarged. The goal of this concept is targeted the simplest data is stored into the space of medical massive data analysis, called "medical data analysis in massive collection". It produces the accuracy and it reaches the 4.8% speed faster the CNN-UDRP. It only focuses these three data, (a) structured data, (b) text data, (c) structured and text data. In this proposed system is improves the medical data oriented term.

Paper [4] gives the survey for Disease prediction in big data healthcare using extended CNN. This concept is applied in the medical field to implements the hospital. It provides

the (i) high accuracy, (ii) high performance, (iii) high convergence speed. To select the particular region and then, analyzed the chronic diseases, that holds the structured data (extracted useful features), the unstructured data is use the CNN technique, so automatically selects the features. The novel CNN is proposed the medical data, and disease risk model is combined this data. The characteristic behavior of this system is selecting the data via previous term. This term is previously applied is possible but not satisfied the disease changes, because disease level is not standard, it is changed in every seconds. To take the selected data from large number of data and improves the accuracy by using risk classification term. The proposed system aim is to predict the risk in liver oriented disease. So, the hospital dataset is related to the liver oriented disease and it collects only the structured data from liver disease information. In the proposed system is use the disease risk modeling and get the accuracy. But the risk prediction is depending on the different feature of medical data with higher accuracy.

This paper [5] author has presented the concept is, “Improving disease prediction by machine learning”, that is using the machine learning and improving the disease prediction. The big data is expanding the medical data, so improving this type of information. This concept uses the genetic algorithm, it is utilizing the recover data, that is the missing data, then, its dataset includes the medical data. In this system using the two calculation terms namely, (i) KNN, (ii) SVM. The chronic diseases every increasing the data CNN-MDRP technique use the medical data. The database includes the medical data, and personal data and detailed history of patient is stored. The RNN based techniques are easily find out the logical data. This system uses the online and offline methods.

Paper No.	Technique	Advantages	Disadvantages
1	Multi-model Disease Risk Prediction (CNN-MDRP)	(i) Medical data analysis with accuracy, (ii) early prediction for disease, (iii) patient oriented data with accuracy, (iv) The medical data, is securely stored and used in many places, (v) incomplete regional data are reduced and give the accuracy result	This feature is only applicable for the structured data so it is not good in disease description.
2	Decision Tree	(i) analyze, (ii) manage, (iii) predict of healthcare data	It predicts only the patient related information
3	Big data	The goal of this concept is targeted the simplest data is stored into the space of medical massive data	It only focuses these three data, (a) structured data, (b) text data, (c) structured and text data
4	Using extended CNN	(i) high accuracy, (ii) high performance, (iii) high convergence speed	The risk prediction is depending on the different feature of medical data
5	(i) KNN, (ii) SVM, Genetic algorithm.	This concept uses the genetic algorithm, it is utilizing the recover data, that is the missing data, then, its dataset includes the medical data	Take more time.

CHAPTER 3

Project Design

This chapter presents in brief the tools used to build the application. Also, it contains detailed analysis from data collection, data preprocessing to model building along with results on unseen data

3.1 Introduction

Technologies Used: -

- **Python**

  Python is a deciphered, significant level and universally useful programming language. Python's plan theory underscores code comprehensibility with its eminent utilization of critical space. Its language develops and object-situated approach expect to assist software engineers with composing clear, legitimate code for little and huge scope projects. Python is progressively composed, and trash gathered. It upholds numerous standards, including organized (especially, procedural), object-arranged and utilitarian programming. Python is frequently depicted as a "batteries included" language because of its extensive standard library. In our venture we have involved python for our suggestion framework in view of cosine similarity.[14]

- **Machine Learning**

  AI (ML) is the investigation of PC calculations that can work on consequently through experience and by the utilization of information. It is viewed as a piece of man-made reasoning. AI calculations assemble a model in view of test information, known as preparing information, to pursue expectations or choices without being expressly customized to do as such. AI calculations are utilized in a wide assortment of uses, for example, in medication, email separating,

discourse acknowledgment, and PC vision, where it is troublesome or impossible to foster customary calculations to play out the required undertakings. [14]

- **Flask**

  Flask is a miniature web structure written in Python. It is delegated a microframework on the grounds that it doesn't need specific apparatuses or libraries. It has no information base reflection layer, structure approval, or whatever other parts where previous outsider libraries give normal capacities. In any case, Flask upholds augmentations that can add application highlights as though they were executed in Flask itself. Expansions exist for object-social mappers, structure approval, transfer dealing with, different open confirmation innovations and a few normal system related devices. [14]

- **API**

  Programming interface is the abbreviation for Application Programming Interface, which is a product delegate that permits two applications to converse with one another. Each time you utilize an application like Facebook, send a text, or check the climate on your telephone, you're utilizing an API. [14]

- **HTML5**

  The Hyper Text Markup Language, or HTML is the standard markup language for archives intended to be shown in an internet browser. Internet browsers get HTML reports from a web server or from nearby capacity and render the records into sight and sound pages. HTML depicts the design of a site page semantically and initially included signals for the presence of the report. HTML components are the structure squares of HTML pages. With HTML develops, pictures and different items, for example, intuitive structures might be implanted into

the delivered page. HTML gives a way to make organized archives by indicating underlying semantics for text like headings, sections, records, connections, quotes and different things. [14]

- **CSS**

  Cascading Style Sheets (CSS) is a template language utilized for portraying the introduction of a report written in a markup language like HTML. CSS is a foundation innovation of the World Wide Web, close by HTML and JavaScript. CSS is intended to empower the partition of show and content, including design, varieties, and textual styles. This partition can work on satisfied availability, give greater adaptability and control in the detail of show attributes, empower various pages to share organizing by determining the important CSS in a different .CSS record which diminishes intricacy and reiteration in the underlying substance as well as empowering the .CSS document to be stored to further develop the page load speed between the pages that share the record and its designing. [14]

- **JavaScript**

  JavaScript frequently truncated JS, is a programming language that is one of the center innovations of the World Wide Web, close by HTML and CSS.[11] Over 97% of sites use JavaScript on the client side for page conduct, frequently integrating outsider libraries. All significant internet browsers have a committed JavaScript motor to execute the code on the client's gadget. [14]

- **React.js**

  React (otherwise called React.js or ReactJS) is a free and open-source front-end JavaScript library for building UIs in light of UI parts. It is kept up with by Meta (previously Facebook) and a local area of individual designers and organizations. Respond can be utilized as a base in the advancement of single-page, versatile, or server-delivered applications with structures like Next.js. Nonetheless, respond is just worried about state the board and delivering that

state to the DOM, so making React applications typically requires the utilization of extra libraries for directing, as well as specific client-side usefulness.

- **Tailwind CSS**

  Tailwind CSS is essentially a utility-first CSS system for quickly constructing custom UIs. It is a profoundly adaptable, low-level CSS structure that gives you all of the structure blocks you want to construct customized plans with practically no irritating stubborn styles you need to battle to supersede. The excellence of this thing called tailwind is it doesn't force plan particular or how your site ought to seem to be, you essentially unite minuscule parts to build a UI that is one of a kind. What Tailwind just does is take a 'crude' CSS record, processes this CSS document over a setup record, and creates a result.

Software's Used: -

- **VS Code**

  Visual Studio Code is a source-code supervisor made by Microsoft for Windows, Linux and macOS. Highlights incorporate help for investigating, sentence structure featuring, canny code consummation, bits, code refactoring, and installed Git. Clients can change the subject, console alternate routes, inclinations, and introduce augmentations that add extra usefulness.

- **Google Colab**

  Colab is a free Jupyter note pad climate that runs totally in the cloud. In particular, it doesn't need an arrangement and the scratch pad that you make can be all the while altered by your colleagues - simply the manner in which you alter records in Google Docs. Colab upholds numerous well known AI libraries which can be effortlessly stacked in your note pad.

- **Jupyter Notebook**

  The Jupyter Notebook is an open-source web application that permits you to make and share records that contain live code, conditions, representations and account message. Utilizes incorporate information cleaning and change, mathematical recreation, factual displaying, information representation, AI, and substantially more.[14]

- **Post Man**

  Postman is an API stage for building and utilizing APIs. Mailman improves on each progression of the API lifecycle and smoothens out joint effort so you can make better APIs — quicker.[14]

- **Heroku**

  Heroku is a cloud stage as an assistance (PaaS) supporting a few programming dialects. One of the primary cloud stages, Heroku has been being developed since June 2007, when it upheld just the Ruby programming language, yet presently upholds Java, Node.js, Scala, Clojure, Python, PHP, and Go.[3] For this explanation, Heroku is supposed to be a bilingual stage as it has highlights for an engineer to construct, run and scale applications likewise across most dialects. Heroku was procured by Salesforce in 2010 for $212 million.[4]

- **Firebase**

  Firebase is an application advancement stage that helps you fabricate and develop applications and games clients love. Upheld by Google and trusted by a great many organizations all over the planet.

  Services by firebase:

  - Cloud Firestore
  - Machine Learning
  - Cloud Functions
  - Authentication

- Hosting
- Cloud Storage
- Realtime Database
- Performance Monitoring
- Test Lab
- Google Analytics
- Remote Config
- Dynamic Links

- **Netlify**

  Netlify is a San Francisco-based distributed computing organization that offers facilitating and serverless backend administrations for web applications and static websites.[5] The organization gives facilitating to sites whose source records are put away in the rendition control framework Git and afterward produced into static web content files[b 1] served by means of a Content Delivery Network.[b 2][b 3] Given the impediments of the absolutely static model, the organization later extended administrations to incorporate substance the board frameworks, and highlights of serverless computing[6] to deal with sites with intuitive elements.[7]

3.2 Problem Statement

- Using X-ray Images to detect pneumonia using CNN (Deep Learning)
- Using patients' medical data to predict the risk of diabetes, kidney disease, heart disease or liver disease.
- Using Symptoms of patients to predict common disease like diabetes, hypertension.
- Building applications which are user accessible and user friendly.
- Developing software to answers complex problems in a click.

3.3 Pneumonia Detection:

1) Introduction

In today's time, Pneumonia is a sensitive and has become common disease because of the air pollution and increase in garbage drainage in our nearby surroundings. Major cause of Pneumonia is the increase in air pollution in the world.

As per WHO, nearly 4 million premature deaths occur annually due to disease related to air pollution. Every year nearly 150 million people gets infected with pneumonia and due to lack of medical resource in certain areas people die with this disease.

One important factor for the cause of death is the early detection of pneumonia. With the advancement in AI early detection of pneumonia is possible and thus can result in decrease in fatality.[4-7]

2) Dataset: -

Dataset contains two folders: -

a) Pneumonia (containing Images of Pneumonia x-rays)

b) Normal (containing Images of non-Pneumonia x-rays)

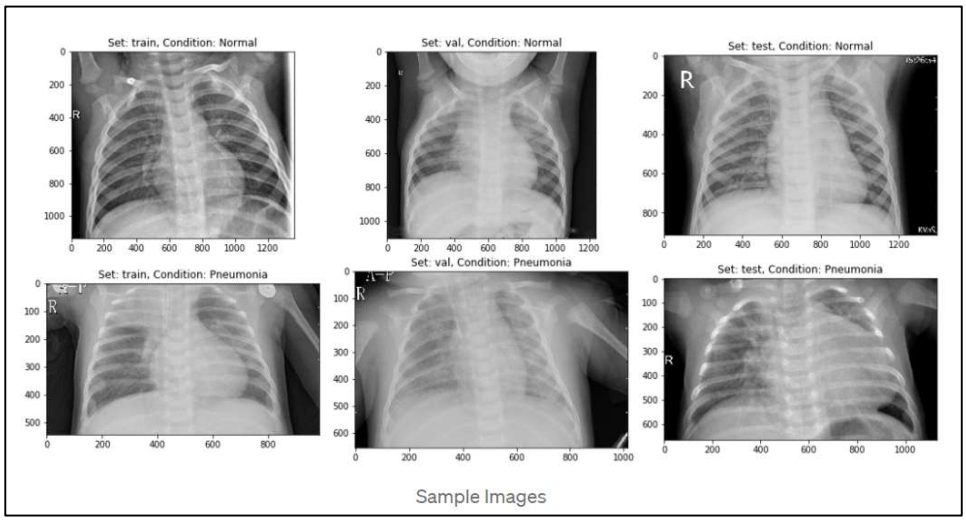


Fig 3.1: Sample 3 Images of Pneumonia and normal class x-rays

3) Packages used

a) TensorFlow

b) Keras

c) Sklearn

d) NumPy

e) Pandas

f) Matplotlib

g) Seaborn

4) Label Preprocessing

Here we used sklearn's 'LabelEncoder' to convert y-labels to integers.

For e.g.: Pneumonia class was mapped to 1 and Normal class was mapped to 0.

5) Loading the Images

We used keras Image Generator to Generate Image of batch size 64 from the directory.

Image Generator is used to load Images from the directory and pass it to the model for training sequentially. So, at a time only batch of 64 Images is loaded into the ram avoiding memory issues.

6) CNN (Convolutional Neural Network)

A CNN is used to process Images and convert it to feature vectors so as to differentiate it from other Images category.

So here CNN will convert Pneumonia x-ray images to some feature vectors such that it is distinguishable from normal class x-ray images feature vector.

CNN is designed to process Images and extract feature vectors from the Images.

For e.g.: A dog feature vectors will be shape, size, color, hair type, pattern of eyes, nose, etc.

7) Convolutional Neural Network (CNN) for Pneumonia Detection

Important points for CNN: -

1. Start with small number of filters like 32 and increase the number of filters with each passing CNN layer.
2. Use Max-pooling or 'Average pooling' appropriately after CNN layer.
3. Use 'relu' or 'variants of relu' as activation functions for hidden layers sinch they are less likely to go under vanishing gradient problem.
4. Flatten the CNN layer to pass it through the ANN layer.
5. Use sigmoid activation function as activation function for last layer with one neuron for binary classification problem.
6. Use binary_cross_entropy as loss function for binary classification problem.

```
cnn.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 498, 498, 32)      320
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 249, 249, 32)      0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 247, 247, 32)      9248
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 123, 123, 32)      0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 121, 121, 32)      9248
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 60, 60, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 58, 58, 64)        18496
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 29, 29, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 27, 27, 64)        36928
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 13, 13, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 10816)             0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               1384576
_________________________________________________________________
dense_3 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 65
=================================================================
Total params: 1,467,137 Trainable params: 1,467,137 Non-trainable
params: 0
```

Fig 3.2: CNN architecture for Pneumonia Prediction.[10]

8) Assigning Class Weights

For Imbalance classification problem we try to give class weights to both classes to make them unbiased for the model training.

9) Training The Model

We trained the model for 25 epochs but due to early stopping our best test results

were obtained as epoch no 10.

After 10 epochs our validation loss was 14.9% and validation accuracy was 94.6%.

10) Evaluate

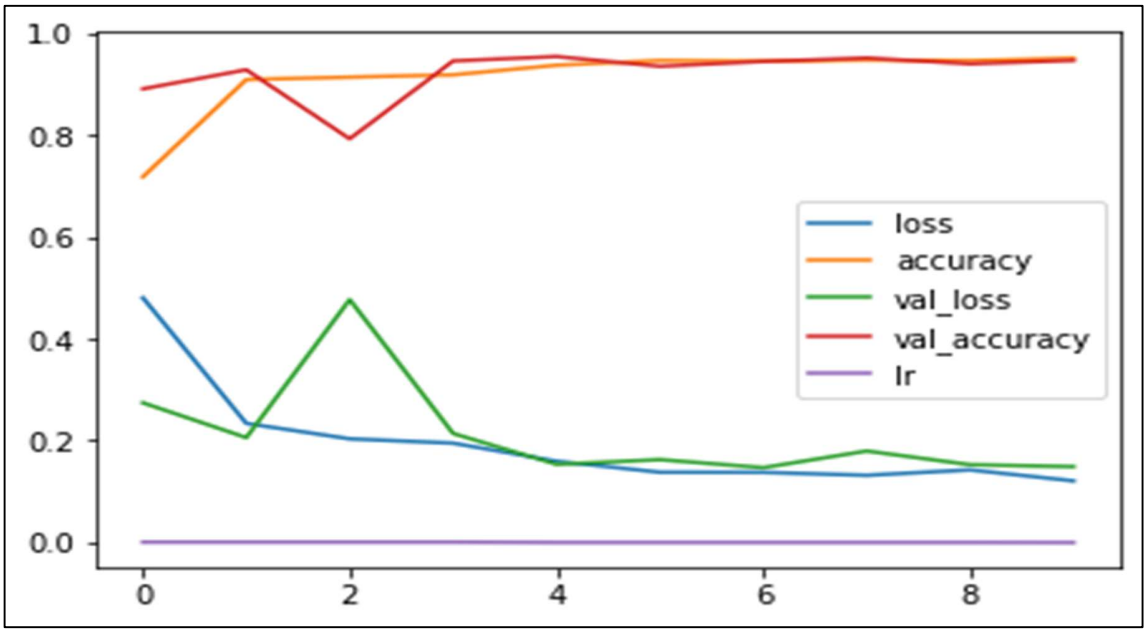


Fig 3.3: Train vs test accuracy and train vs test loss

The accuracy we are getting on Test dataset is of 91%

Confusion matrix: -

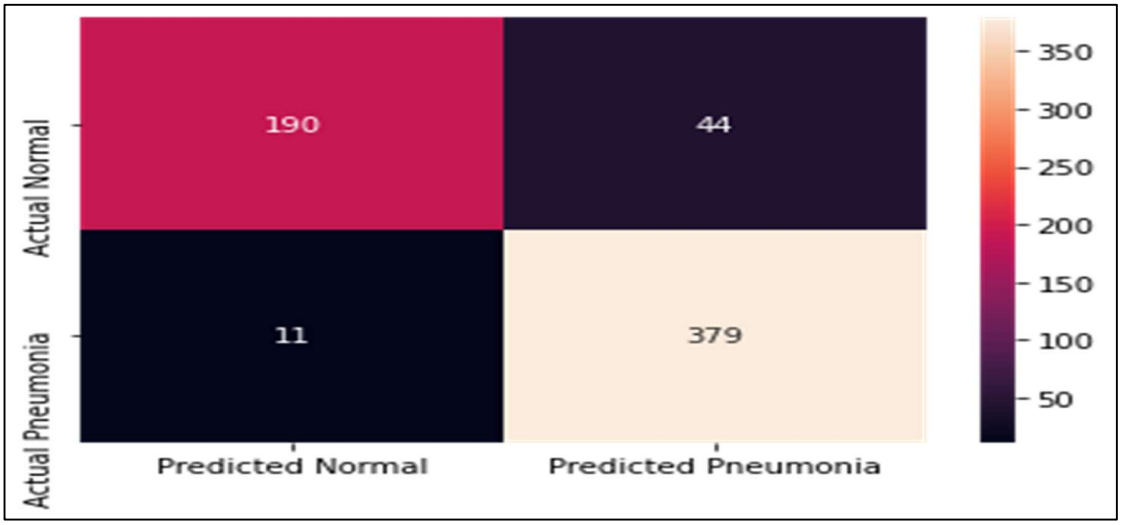


Fig 3.4: Confusion matrix of test data

Classification Report: -

	precision	recall	f1-score	support
NORMAL	0.95	0.81	0.87	234
PNEUMONIA	0.90	0.97	0.93	390
accuracy			0.91	624
macro avg	0.92	0.89	0.90	624
weighted avg	0.91	0.91	0.91	624

Fig 3.5: Classification report of test data

3.4 Diabetes

1) Introduction

India has large no of diabetes patients in the world. Diabetes is one of the major causes of increase in deaths in India every year.

Major cause of diabetes is intake of sugar. Since India is one of the large manufacturers of sugar in the world it also has high number of diabetes patients.

With the help of AI and ML detection of diabetes is possible and, in this article, we have tried to build a simple ML model to detect the risk of diabetes in patients.[11]

2) Dataset

Dataset is obtained from Kaggle website. Source of the data is 'National Institute of Diabetes and Digestive and Kidney Diseases'. [11]

3) Packages Used

a. NumPy
b. Pandas
c. Sklearn
d. Seaborn
e. Matplotlib
f. Yellowbrick

4) EDA

The dataset contains 9 columns viz. pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree function, age, and outcome. The outcome is the target variable, the rest all are the predictor variables.

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Fig 3.6: Columns Information of diabetes dataset

```
In [7]: df.isna().sum()

Out[7]: Pregnancies                 0
        Glucose                     0
        BloodPressure               0
        SkinThickness               0
        Insulin                     0
        BMI                         0
        DiabetesPedigreeFunction    0
        Age                         0
        Outcome                     0
        dtype: int64
```

Fig 3.7: Missing Values Distribution

The dataset has no missing values as can be seen from the figure

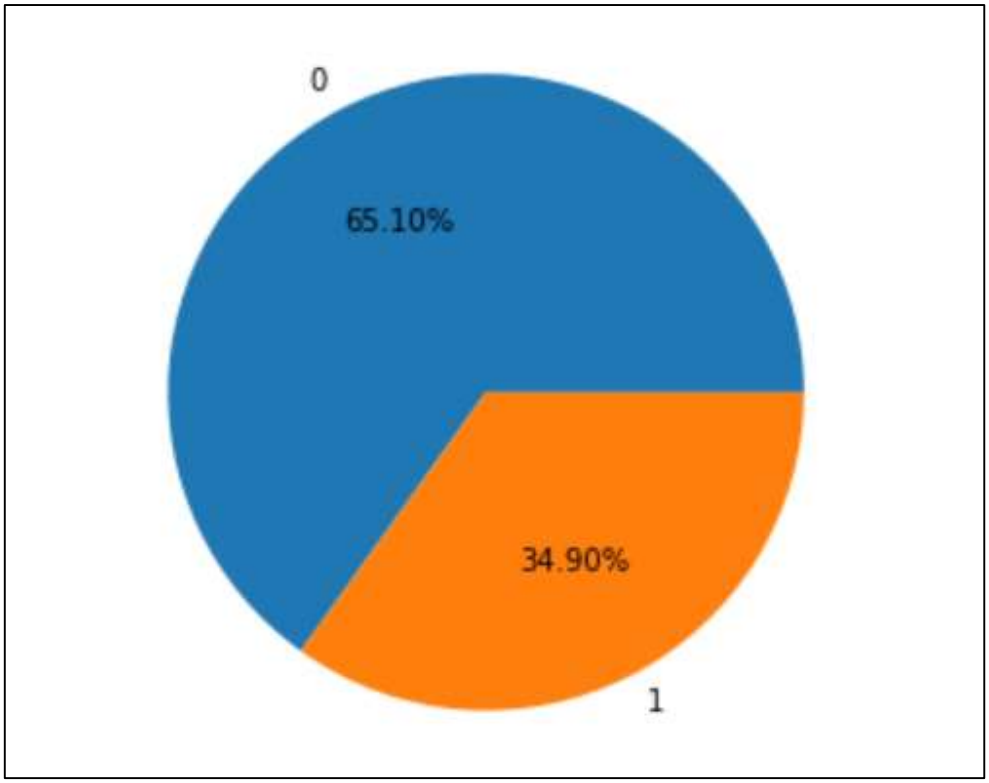


Fig 3.8: Class distribution of outcome variable

As we can see from the figure class is imbalance as there are 65% i.e., class 0 data of no diabetes patients and 35% i.e., class 1 data containing diabetes patients

5) Data Preprocessing

Dealing with Imbalanced Data

SMOTE: Synthetic Minority Oversampling Technique

SMOTE is an oversampling technique used for creating synthetic samples for minority class.

It is mainly used to remove class Imbalance so that model is not biased towards majority class while training on the data.

It used different algorithms like KNN, SVM to generate new samples for minority class.

Working Procedure:

a) Consider one minority samples and used KNN to find the k-nearest neighbors for that particular sample.

b) Now once we have the neighbors, connect that minority sample to its neighbors.

c) Now arbitrarily put one sample on the line joining the minority sample and its neighbors.

d) Repeat this procedure for all minority samples in the dataset till the classes are balanced.

6) Assigning Class Weights

For Imbalance classification problem we try to give class weights to both classes to make them unbiased for the model training.

7) Training and Results

Here we used Logistic Regression, K-nearest neighbours, support vector machine, decision tree, random forest and ada-boost classifier.[8]

```
In [44]: result
```

Out[44]:

	Classifiers	Precision	Recall	F-measure	Accuracy
0	DT	0.784091	0.780	0.779205	0.780
1	RF	0.815032	0.815	0.814995	0.815
2	LR	0.723214	0.720	0.718988	0.720
3	KNN	0.759384	0.755	0.753960	0.755
4	AB	0.758371	0.750	0.747958	0.750
5	SVM	0.719949	0.715	0.713388	0.715

Fig 3.9: Final result of all classifiers

As we can see that random forest gives the highest accuracy with predicting 81% times correct class label

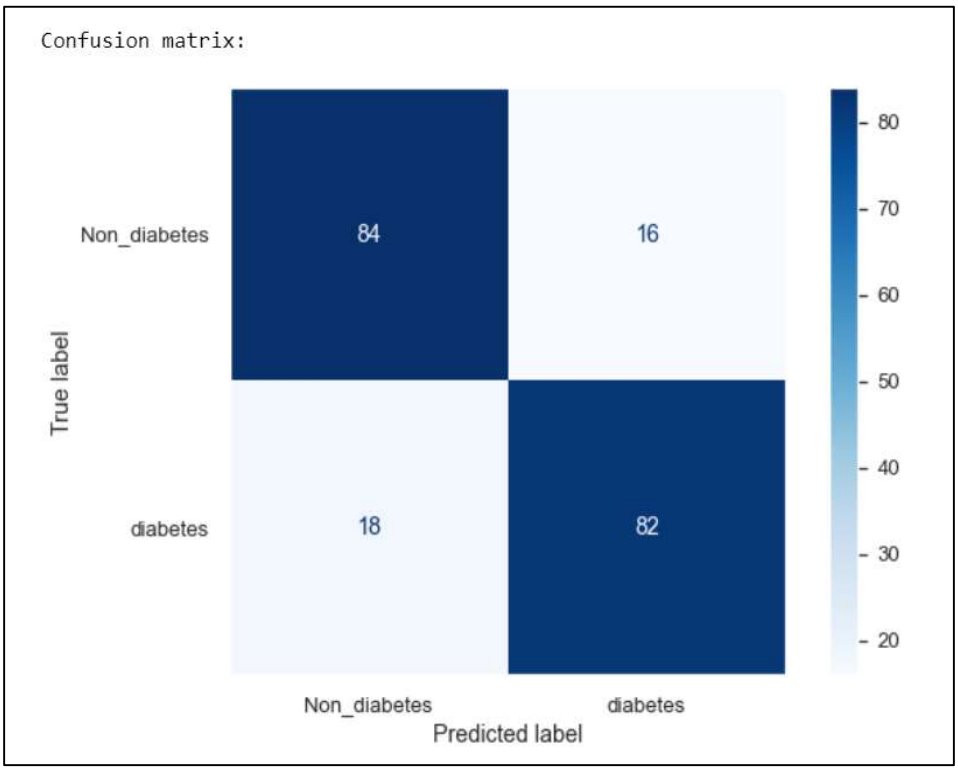


Fig 3.10: Random Forest confusion matrix

Classification report of Random forest classifier

Classification report:-

	precision	recall	f1-score	support
Non_diabetes	0.82	0.84	0.83	100
diabetes	0.84	0.82	0.83	100
accuracy			0.83	200
macro avg	0.83	0.83	0.83	200
weighted avg	0.83	0.83	0.83	200

Fig 3.11: Classification report of random forest.

3.5 Heart Disease Prediction:

1) Introduction

In today's world, heart disease is one of the major causes of early deaths. As the age increases the risk of heart disease also increases. Studies shows that majority of deaths are due to late detection of the disease. So early detection of heart disease becomes very important.

As with advancement in Machine learning and AI early detection is possible along with very high accuracy.[9-11]

2) Dataset used

The data is collected from UCI repository and contains 303 individual data.

Following are the variables from the dataset: -

a) Age

b) Sex (Male or Female)

c) Chest pain type

d) *Resting Blood Pressure*

e) *Serum Cholesterol*

f) *Fasting Blood Sugar*

g) *Resting ECG*

h) *Max heart rate achieved*

i) *Exercise induced angina*

j) ST depression induced by exercise relative to rest

k) *Peak exercise ST segment*

l) *Number of major vessels (0–3) colored by fluoroscopy*

m) *Thal*

n) *Diagnosis of heart disease*

3) Packages Used

a) NumPy

b) Pandas

c) Sklearn

d) Seaborn

e) Matplotlib

f) Yellowbrick

4) Data Pre-Processing

```
Out[2]:
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
target      0
dtype: int64
```

Fig 3.12: Null values in each column of the data

In total we only have 6 null values (4 from ca and 2 from thal).

Since null values are less, we drop the null values.

We also splitted the dataset into 80/20 ration for train and test data.

Dealing with Imbalanced Data

SMOTE: Synthetic Minority Oversampling Technique

SMOTE is an oversampling technique used for creating synthetic samples for minority class.

It is mainly used to remove class Imbalance so that model is not biased towards majority class while training on the data. It used different algorithms like KNN, SVM to generate new samples for minority class.

Working Procedure:

a) Consider one minority samples and used KNN to find the k-nearest neighbors for that particular sample.

b) Now once we have the neighbors, connect that minority sample to its neighbors.

c) Now arbitrarily put one sample on the line joining the minority sample and its

neighbours.

d) Repeat this procedure for all minority samples in the dataset till the classes are balanced.

5) Training and Results:

We trained several models such as Logistic regression, LDA, QDA, Linear-SVC, Kernel-SVC, KNN, Decision tree classifier, Extra tree classifier, Random Forest classifier, Extra trees classifier, voting classifier and Stacking classifier and evaluated the results on test data.

Below table summarizes all models and their results on test data: -

	Classifiers	Train score	Test score	No of Missclassification	% of Missclasification	Training time	Prediction time
0	LR	0.847328	0.818182	12	18.181818	8.239647	0.000995
1	LDA	0.839695	0.833333	11	16.666667	0.003997	0.001000
2	QDA	0.862595	0.787879	14	21.212121	0.016003	0.000997
3	LSVC	0.851145	0.833333	11	16.666667	0.020007	0.000000
4	SVC	0.912214	0.818182	12	18.181818	0.008006	0.005014
5	KNN	0.858779	0.878788	8	12.121212	0.009021	0.127268
6	DTC	1.000000	0.772727	15	22.727273	0.008032	0.000000
7	ETC	1.000000	0.803030	13	19.696970	0.007999	0.000999
8	RFC	0.896947	0.848485	10	15.151515	0.428743	0.110207
9	ET	1.000000	0.878788	8	12.121212	0.547042	0.109422
10	VTC	0.969466	0.863636	9	13.636364	2.150225	0.217731
11	STC	1.000000	0.863636	9	13.636364	8.159245	0.454857

Fig 3.13: Result summary table for all classifiers

From above table we can see that Extra Tress Classifier performs best on test data compared to other models. So, we choose this model as our final model for future predictions.

Below are the results of Extra Tress Classifier on test data: -

```
No of missclassified for class 0 (test data) = 4
No of missclassified for class 1 (test data) = 4
------------------------------------------------------------
Total no of missclassified points (test data) =  8
Total % of missclassified points (test data) =  12.121212121212121
```

Performance of Extra trees classifier on test data

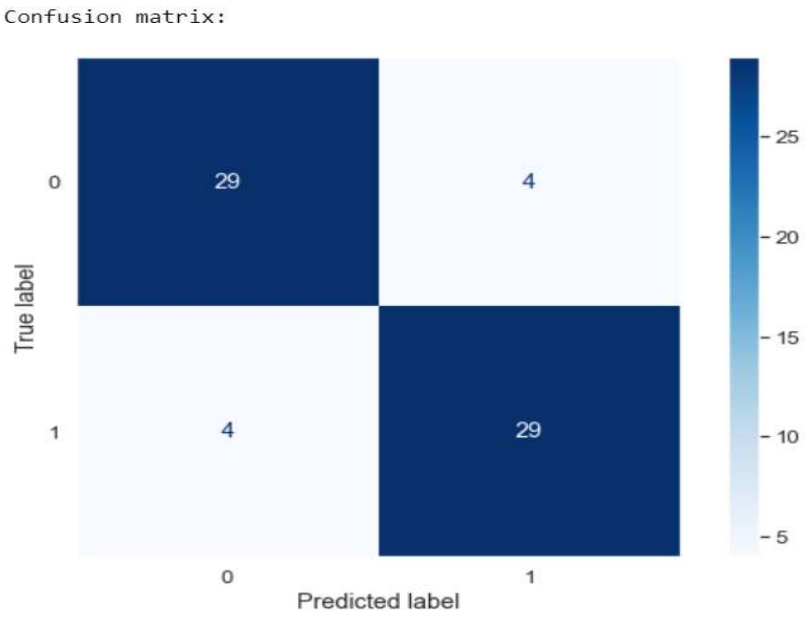


Fig 3.14: Confusion matrix of Extra trees classifier on test data

Classification report:-

	precision	recall	f1-score	support
0	0.88	0.88	0.88	33
1	0.88	0.88	0.88	33
accuracy			0.88	66
macro avg	0.88	0.88	0.88	66
weighted avg	0.88	0.88	0.88	66

Fig 3.15: Classification Report of Extra trees classifier on test data

3.6 Kidney Disease Prediction:

1) Introduction

In today's world, kidney diseases are one of the major central problems in healthcare domain. As the age increases the risk of kidney disease also increases. Studies shows that majority of deaths are due to late detection of the kidney disease. So early detection of kidney disease becomes very important.

As with advancement in Machine learning and AI early detection is possible along with very high accuracy.

So, in this section we will try to build a simple machine learning model to detect kidney disease.[12]

2) Dataset used

Here we have used chronic kidney disease dataset from Kaggle.

Following are the columns in the dataset: -

1) age
2) blood pressure
3) specific gravity
4) albumin

5) sugar
6) red blood cells
7) pus cell
8) pus cell clumps
9) bacteria
10) blood glucose random
11) blood urea
12) serum creatinine
13) sodium
14) potassium
15) hemoglobin
16) packed cell volume
17) white blood cell count
18) red blood cell count
19) hypertension
20) diabetes mellitus
21) coronary artery disease
22) appetite
23) pedal edema
24) anemia
25) class

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 25 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       391 non-null    float64
 1   blood_pressure            388 non-null    float64
 2   specific_gravity          353 non-null    float64
 3   albumin                   354 non-null    float64
 4   sugar                     351 non-null    float64
 5   red_blood_cells           248 non-null    object
 6   pus_cell                  335 non-null    object
 7   pus_cell_clumps           396 non-null    object
 8   bacteria                  396 non-null    object
 9   blood_glucose_random      356 non-null    float64
 10  blood_urea                381 non-null    float64
 11  serum_creatinine          383 non-null    float64
 12  sodium                    313 non-null    float64
 13  potassium                 312 non-null    float64
 14  haemoglobin               348 non-null    float64
 15  packed_cell_volume        330 non-null    object
 16  white_blood_cell_count    295 non-null    object
 17  red_blood_cell_count      270 non-null    object
 18  hypertension              398 non-null    object
 19  diabetes_mellitus         398 non-null    object
 20  coronary_artery_disease   398 non-null    object
 21  appetite                  399 non-null    object
 22  pedal_edema               399 non-null    object
 23  anemia                    399 non-null    object
 24  class                     400 non-null    object
dtypes: float64(11), object(14)
memory usage: 78.2+ KB
```

Fig 3.16: Column data-type and null values distribution

3) Packages Used

0. NumPy
1. Pandas
2. Sklearn
3. Seaborn
4. Matplotlib

5. TensorFlow
6. Keras

4) Data Preprocessing

Dealing with Imbalanced Data

SMOTE: Synthetic Minority Oversampling Technique

SMOTE is an oversampling technique used for creating synthetic samples for minority class.

It is mainly used to remove class Imbalance so that model is not biased towards majority class while training on the data.

It used different algorithms like KNN, SVM to generate new samples for minority class.

Working Procedure:

A) Consider one minority samples and used KNN to find the k-nearest neighbour for that particular sample.

B) Now once we have the neighbours, connect that minority sample to its neighbours.

C) Now arbitrarily put one sample on the line joining the minority sample and its neighbours.

D) Repeat this procedure for all minority samples in the dataset till the classes are balanced.

5) Training and Results

Here we used simple deep learning based ANN (Artificial Neural Network).

Below is the architecture of the ANN: -

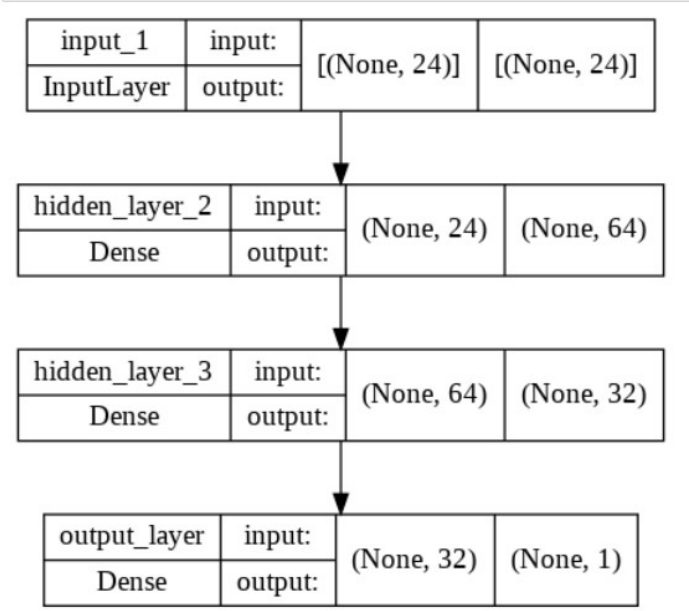


Fig 3.17: Architecture of ANN used for training

Here we trained the model for 30 epochs and used 'balanced accuracy' as our performance measure and 'binary cross entropy' as our loss function.

Within 30 epochs model was able to achieve 100% accuracy on test data.

Below are the results of ANN on test data: -

Confusion matrix:

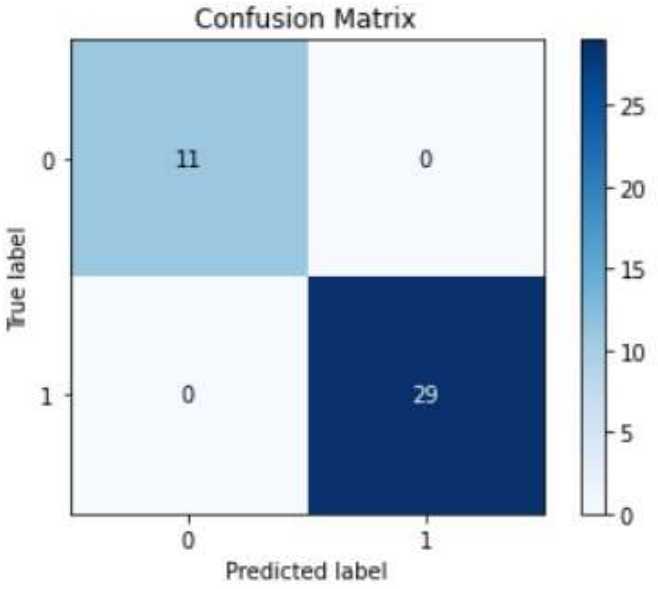


Fig 3.18: Confusion matrix of ANN on test data

Classification report:-

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
1	1.00	1.00	1.00	29
accuracy			1.00	40
macro avg	1.00	1.00	1.00	40
weighted avg	1.00	1.00	1.00	40

****

Fig 3.19: Classification report of ANN on test data

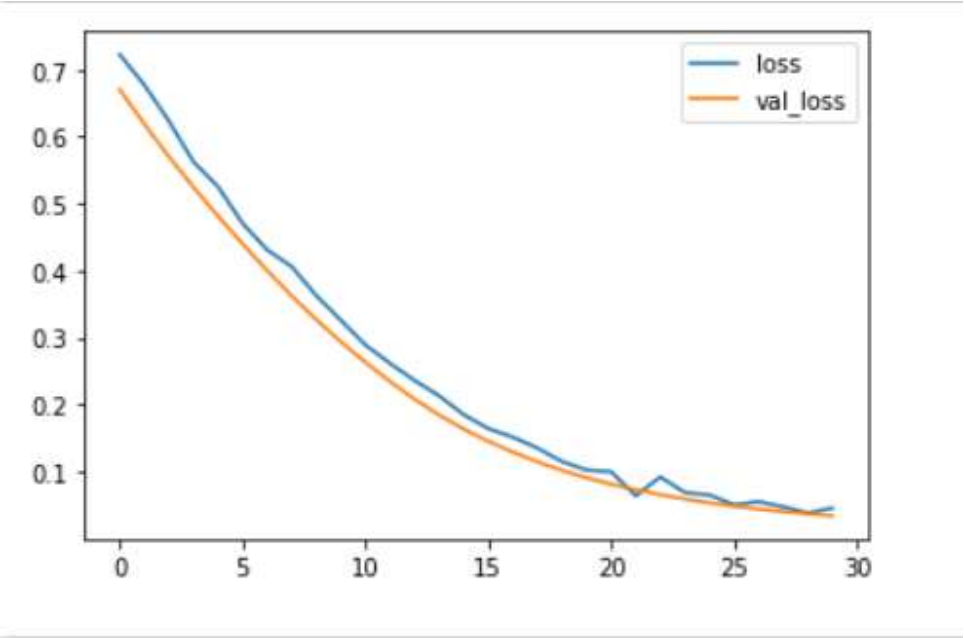


Fig 3.20: Loss vs No-of-epochs of ANN

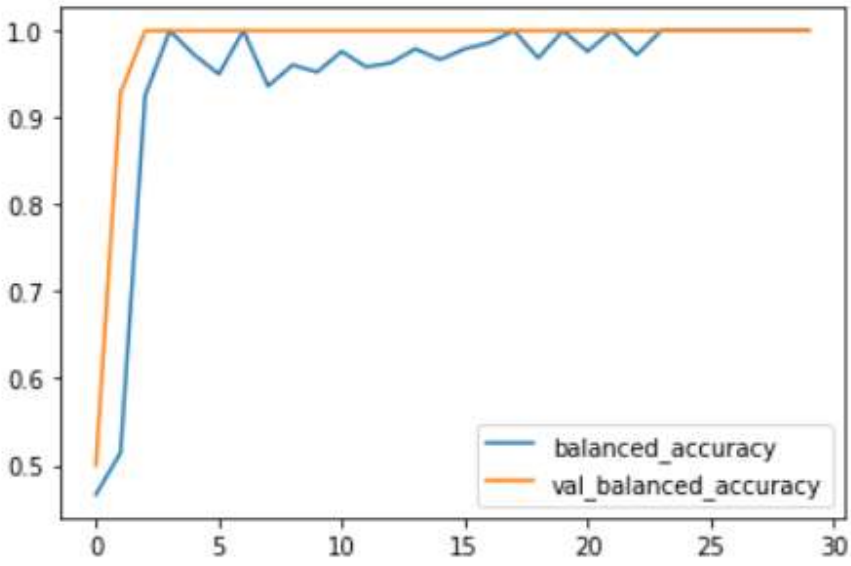


Fig 3.21: Balanced-accuracy vs No-of-epochs of ANN

3.7 Liver Disease Prediction:

1) Introduction

In today's world, diseases related to liver are increasing day by day. From studies it is found that one out of 10 people suffers from liver disease.

Due to lack of awareness about the importance of treatment of liver disease people are dying at higher rate compared to past few years.

With the help of AI and machine learning detection of liver disease has become one of the key areas of research and development.

So here we have built a simple machine learning based model to detect liver disease.

2) Dataset used

Data used is 'Indian Liver disease' dataset and is downloaded from Kaggle.

It contains 416 liver patients and 167 non liver patients.

Here column named 'Dataset' consist of 1 and 0 (1 means liver disease is present and 0 means liver disease is not present).[13]

- Following are the columns from the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         583 non-null    int64
 1   Gender                      583 non-null    object
 2   Total_Bilirubin             583 non-null    float64
 3   Direct_Bilirubin            583 non-null    float64
 4   Alkaline_Phosphotase        583 non-null    int64
 5   Alamine_Aminotransferase    583 non-null    int64
 6   Aspartate_Aminotransferase  583 non-null    int64
 7   Total_Protiens              583 non-null    float64
 8   Albumin                     583 non-null    float64
 9   Albumin_and_Globulin_Ratio  579 non-null    float64
 10  Dataset                     583 non-null    int64
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
```

Fig 3.22: Column data-types and null values distribution

3) **Packages Used**

1. NumPy
2. Pandas
3. Sklearn
4. Seaborn
5. Matplotlib
6. TensorFlow
7. Keras

4) **Data Preprocessing**

Dealing with Imbalanced Data

SMOTE: Synthetic Minority Oversampling Technique

SMOTE is an oversampling technique used for creating synthetic samples for minority class.

It is mainly used to remove class Imbalance so that model is not biased towards majority class while training on the data.

It used different algorithms like KNN, SVM to generate new samples for minority class.

Working Procedure:

i. Consider one minority samples and used KNN to find the k-nearest neighbour for that particular sample.
ii. Now once we have the neighbours, connect that minority sample to its neighbours.
iii. Now arbitrarily put one sample on the line joining the minority sample and its neighbours.
iv. Repeat this procedure for all minority samples in the dataset till the classes are balanced.

5) **Training and Results:**

We trained several models such as Logistic regression, LDA, QDA, Linear-SVC, Kernel-SVC, KNN, Decision tree classifier, Extra tree classifier, Random Forest classifier, Extra trees classifier, voting classifier and Stacking classifier and evaluated the results on test data.

Below table summarizes all models and their results on test data: -

	Classifiers	Train score	Test score	No of Missclassification	% of Missclasification	Training time	Prediction time
0	LR	0.714502	0.716867	47	28.313253	8.479547	0.000000
1	LDA	0.705438	0.698795	50	30.120482	0.008996	0.000000
2	QDA	0.672205	0.650602	58	34.939759	0.024997	0.002028
3	LSVC	0.717523	0.722892	46	27.710843	0.068997	0.000000
4	SVC	0.735650	0.734940	44	26.506024	0.045001	0.022027
5	KNN	0.796073	0.680723	53	31.927711	0.010985	0.121365
6	DTC	1.000000	0.867470	22	13.253012	0.019005	0.001009
7	ETC	1.000000	0.873494	21	12.650602	0.019998	0.000996
8	RFC	1.000000	0.819277	30	18.072289	0.553645	0.108918
9	ET	1.000000	0.867470	22	13.253012	0.445190	0.111113
10	VTC	1.000000	0.873494	21	12.650602	4.847528	0.110261
11	STC	1.000000	0.897590	17	10.240964	3.169696	0.236332

Fig 3.23: Result summary table for all classifiers

From above table we can see that Stacking Classifier performs best on test data compared to other models. So, we choose this model as our final model for future predictions.

Below are the results of Stacking Classifier on test data: -

```
No of missclassified for class 0 (test data) = 11
No of missclassified for class 1 (test data) = 6
------------------------------------------------------------
Total no of missclassified points (test data) =  17
Total % of missclassified points (test data) =  10.240963855421686
```

Fig 3.24: Performance of Stacking Classifier on test data

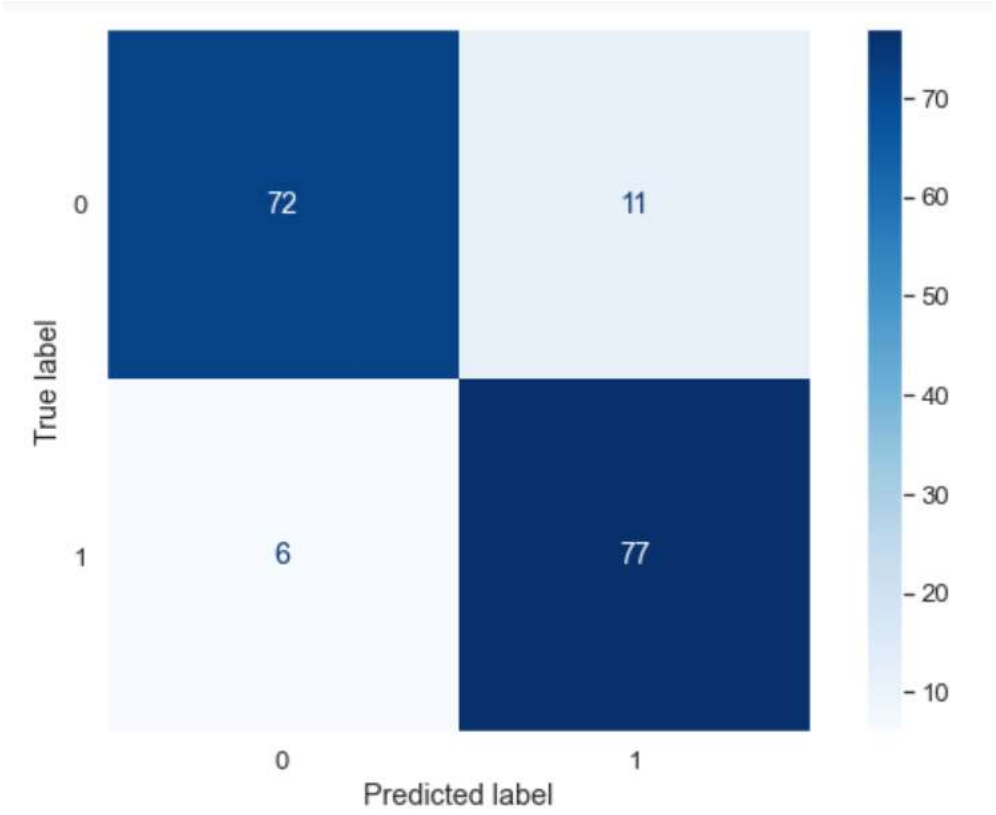


Fig 3.25: Confusion matrix of Stacking Classifier on test data

Classification report:-

	precision	recall	f1-score	support
0	0.92	0.87	0.89	83
1	0.88	0.93	0.90	83
accuracy			0.90	166
macro avg	0.90	0.90	0.90	166
weighted avg	0.90	0.90	0.90	166

Fig 3.26: Classification Report of Stacking Classifier on test data

3.8 General Disease Prediction:

1) **Introduction**

During the time when Machine Learning and Deep Learning are booming so much, it is very important to understand that all this knowledge is not of any use if we can't apply it to different areas and impact the humanity.

2) **Dataset Used**

The dataset has 132 parameters on which 42 different types of diseases can be predicted. Complete Dataset consists of 2 CSV files. One of them is training and other is for testing your model.

Each CSV file has 133 columns. 132 of these columns are symptoms that a person experiences and last column is the prognosis.

These symptoms are mapped to 42 diseases you can classify these set of symptoms to.

```
array(['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
       'Dimorphic hemorrhoids(piles)', 'Heart attack', 'Varicose veins',
       'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo'], dtype=object)
```

Fig 3.27: List of all Disease in the dataset

3) **Libraries Used**

- Pandas and NumPy

  Pandas and NumPy were used for preprocessing the data and handling missing values so that the data is ready for the input to the model.

- Sklearn

  Sklearn contains all the models that were used for this dataset and also contains some preprocessing function such as standardscaler.

4) Data Preprocessing

We clubbed all symptoms from different columns to one column as can be seen from below image.

	Disease	Symptoms
0	Fungal infection	['itching', 'skin_rash', 'nodal_skin_eruptions...
1	Fungal infection	['skin_rash', 'nodal_skin_eruptions', 'dischro...
2	Fungal infection	['itching', 'nodal_skin_eruptions', 'dischromi...
3	Fungal infection	['itching', 'skin_rash', 'dischromic_patches']
4	Fungal infection	['itching', 'skin_rash', 'nodal_skin_eruptions']

Fig 3.28: Dataset after processing all symptoms

To convert all symptoms to feature vector we used multilabel binarizer.

We also splitted the dataset into 80-20 ratio as train and test data.

5) Training and Results

We trained Logistic regression on top of the symptoms feature with "Class weight = balanced" as parameter to the model for handling imbalance data.

We achieved 100% accuracy on test data using this model and saved this model for future predictions.

Below are the results on test data: -

```
Training accuracy =  1.0
Testing accuracy =  1.0
```

Fig 3.29: Train and test accuracy results

```
Training log_loss =  0.0224348833929566 56
Testing log_loss =  0.0240698447238168 54
```

Fig 3.30: Train and test loss results

3.9 Building and Testing Prediction APIs

Using the above models, we created API to predict the diseases like diabetes and pneumonia. To create APIs, we used flask framework and multiple python libraries.

Below Mentions are steps towards building Flask restful API –

Step1:

Create the main project directory at preferred location and in that directory create file "app.py".

Step2:

Open terminal and install virtual environment using command "pip install virtualenv". After this commands execution create virtual environment using command "python -m venv env".

Step3:

Activate virtual environment using following commands: "cd env"- to enter folder env and then "Scripts\activate" – to activate virtual environment.

```
C:\Users\smitm\Desktop\FinalProject_APIs\Diabetes_api>cd env

C:\Users\smitm\Desktop\FinalProject_APIs\Diabetes_api\env>scripts\activate

(env) C:\Users\smitm\Desktop\FinalProject_APIs\Diabetes_api\env>flask run
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
```

Fig 3.31: Creating Virtual Environment

Step4:

Now return to the main project directory and add machine learning code and jolib files of respective models in directory. And add all the additional packages using command "pip install r- requirements.txt"

Step5:

Create Flask-app and Routes to point API to URL in app.py and write the code to execute the task.

```
@app.route('/', methods=['GET', 'POST'])
def pneumonia():
    if request.method == 'GET':
        return jsonify({'message': 'Upload X-ray Image ',
                        'POST-Key for uploading file': 'file'})

    elif request.method == 'POST':
        img = request.files['file'].read()
        prob, pred = prediction(img)

        if pred == 1:
            return jsonify({'message': 'Pneumonia Detected', 'probability': (prob*10)})
        elif pred == 0:
            return jsonify({'message': 'Pneumonia Not Detected', 'probability': (1-prob)*10})
```

Fig 3.32: Creating API Sample Code

Now the app is created and ready to use API. In above image "/pneumonia" represents the URL to access the API. "GET & POST" requests both are created and after file is uploaded POST request is used is to return the message in JSON format using Jsonify module about the result predicted using machine learning model.

'file' is used to represent corresponding variable while request is made.

Step6:

Testing of API Using Postman application. (Only shown for Diabetes and Pneumonia)

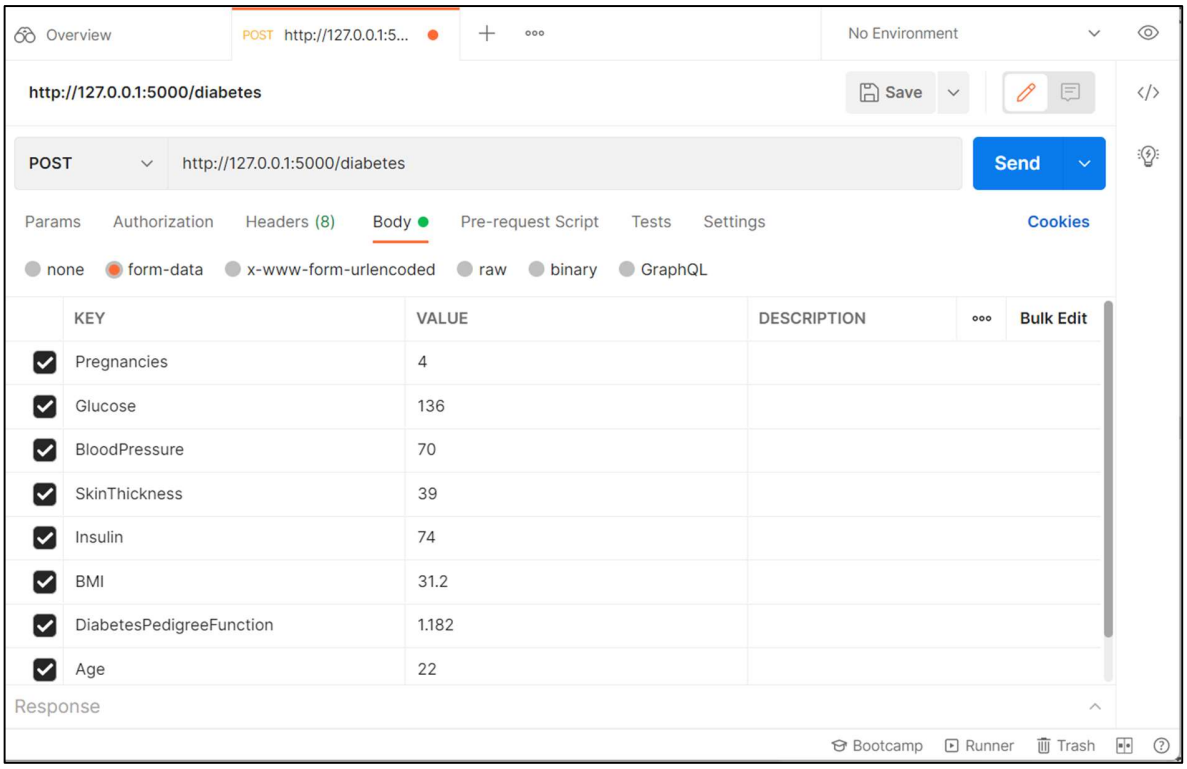


Fig 3.33: Testing Diabetes API (input value shown)

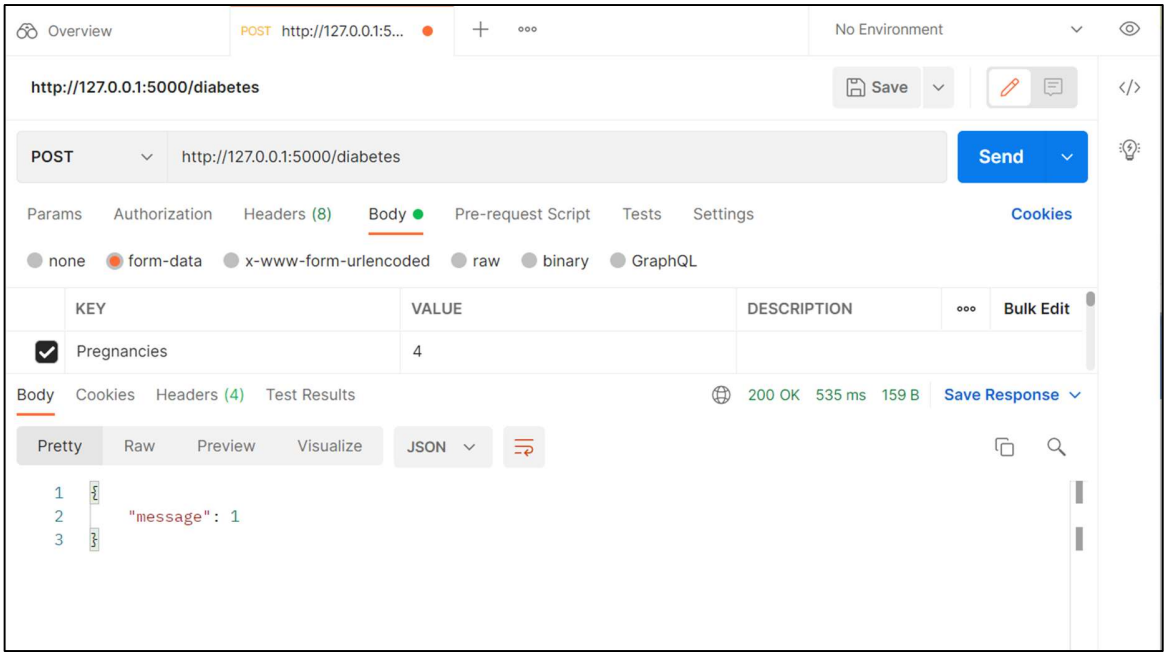


Fig 3.34: Testing Diabetes API (result shown)

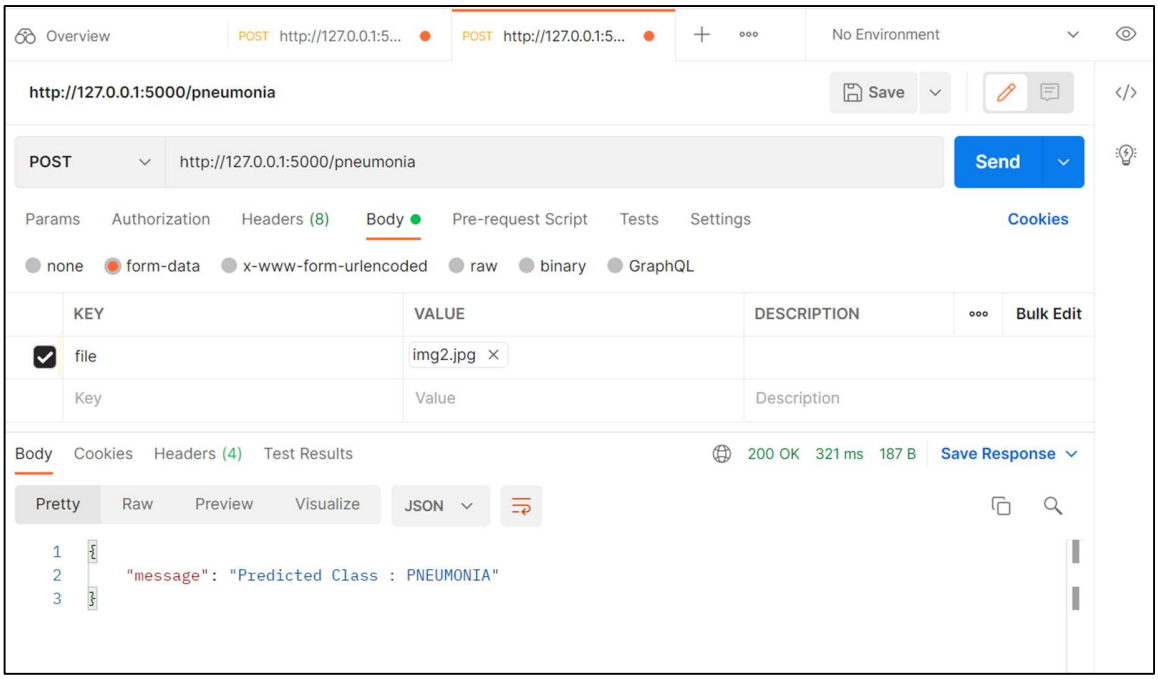


Fig 3.35: Testing Pneumonia API (input value and result shown)

Here we understood how to build API on a machine learning model and then testing of API using postman.

3.10 Objective

- The objective of Diabetes, Kidney, Liver and Heart disease is to predict whether a person has that particular disease or not based on factors like age, gender and other medical details.
- The objective of General disease prediction is to predict the general diseases based on just symptoms of the patient.
- This application lets user verify the data on its own accords and helps in taking decisions as a 2nd person.
- Easy automation of hospital records with the help of APIs created. Helps to make faster decisions.

CHAPTER 4

Implementation and Experimentation

This chapter presents layout and implementation of the Application

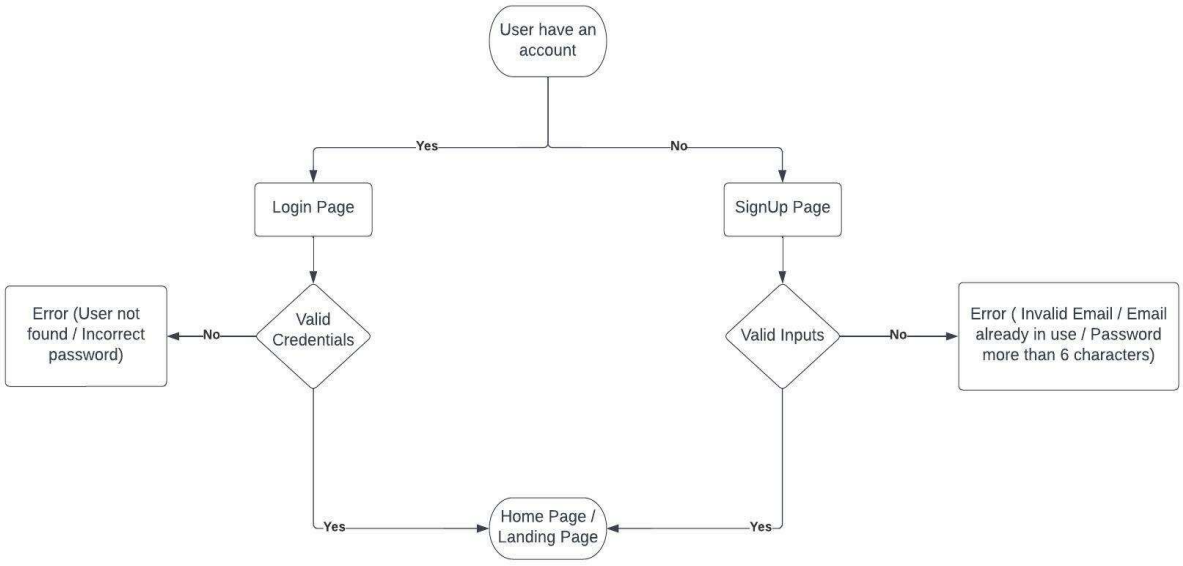
4.1 Implementation

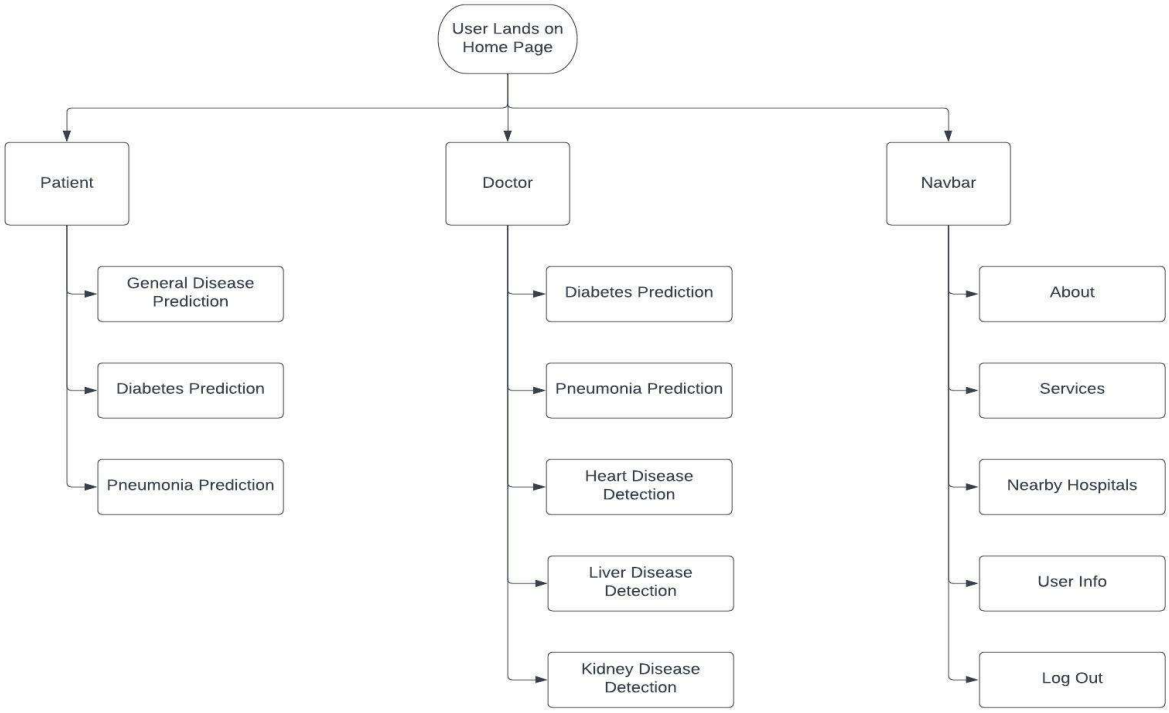
The design cycle starts by recognizing what the framework ought to do and what the sources of info and results of every module. Essentially, the framework ought to have the option to distinguish the traffic signs with greatest precision. Most importantly, we have introduced and stacked the vital bundles that will be utilized for this project. We then, at that point, begin with composing the code and the way to deal with building this traffic sign recognition model as examined in stages:

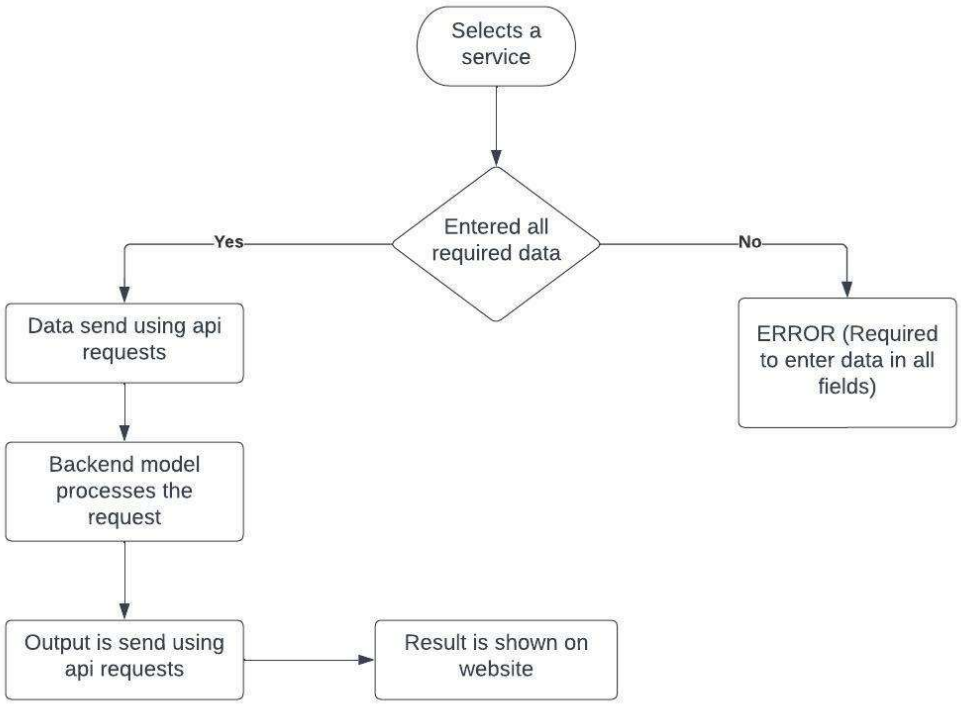
a. Explore the dataset
b. Assemble & Build the models
c. Train and validate the models
d. Test the model with test dataset
e. Build APIs to access the models
f. Building Web Application.
g. Hosting APIs and Application on cloud

Flow of Application:-

- User goes to website and registers using email-id and password.
- If user already have an account user logs-in.
- After Sign-in\Sign-up user lands on home page.
- Multiple disease prediction/detection services are available on this website user can get access to all of them.
- User are required to enter correct inputs.
- Then the data is send to servers using POST method and models processes the data.
- Output from models in shown to user on web-application.




User Lands on Home Page
Patient
Doctor
Navbar
General Disease Prediction
Diabetes Prediction
Pneumonia Prediction
Diabetes Prediction
Pneumonia Prediction
Heart Disease Detection
Liver Disease Detection
Kidney Disease Detection
About
Services
Nearby Hospitals
User Info
Log Out


Selects a service
Entered all required data
Yes
No
Data send using api requests
ERROR (Required to enter data in all fields)
Backend model processes the request
Output is send using api requests
Result is shown on website

4.2 Application Layouts

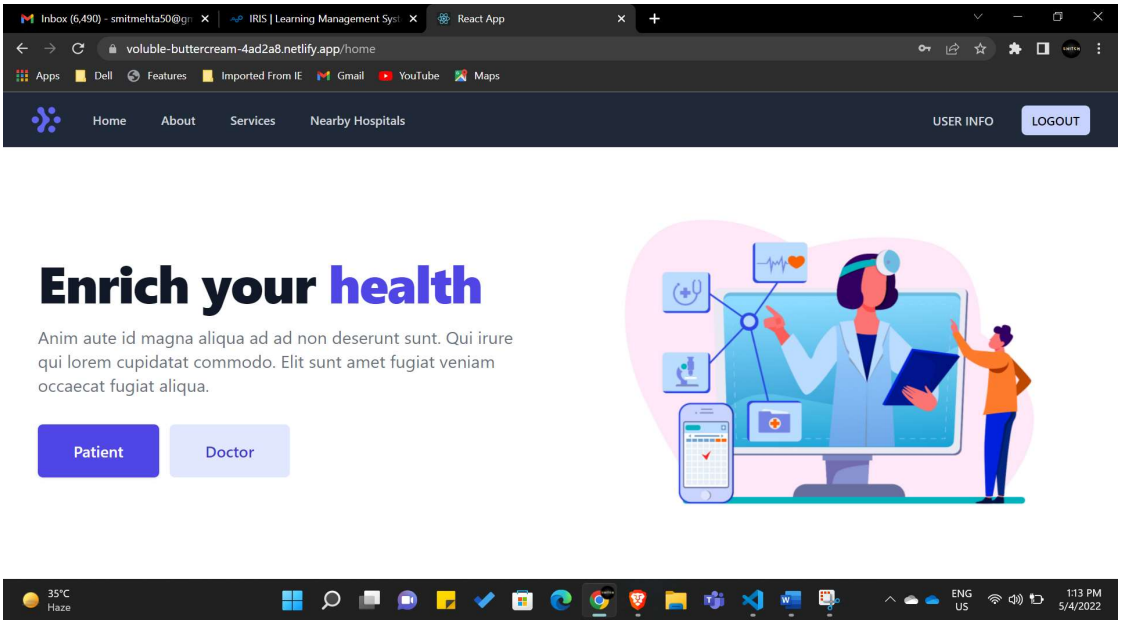


Fig 4.1: Home/Landing Page

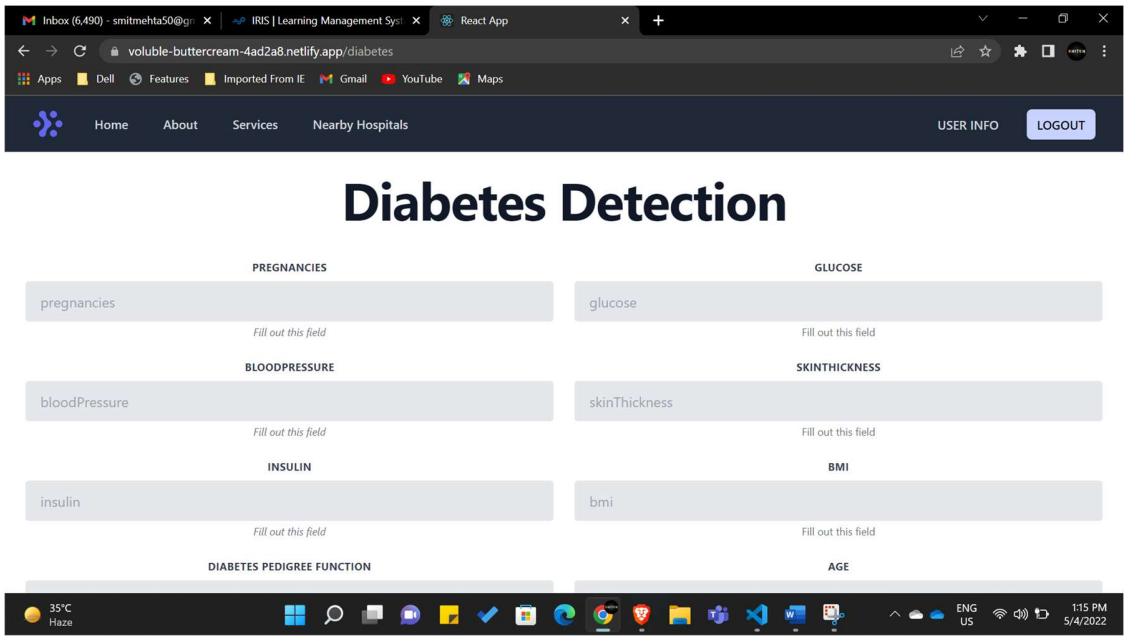


Fig 4.2: Diabetes Prediction Page

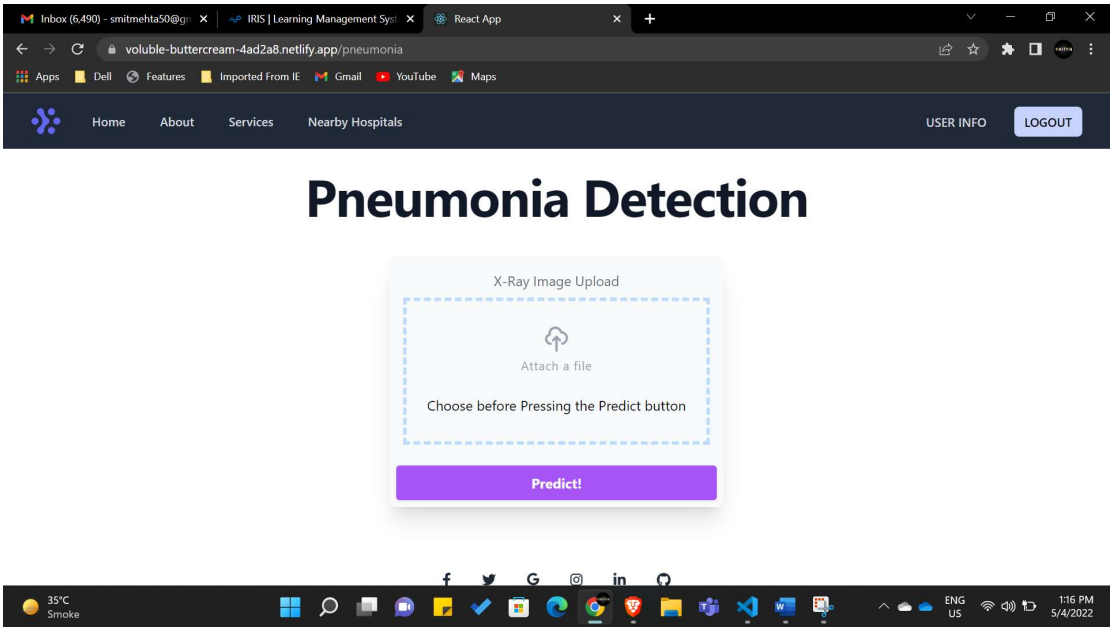


Fig 4.3: Pneumonia Detection Page

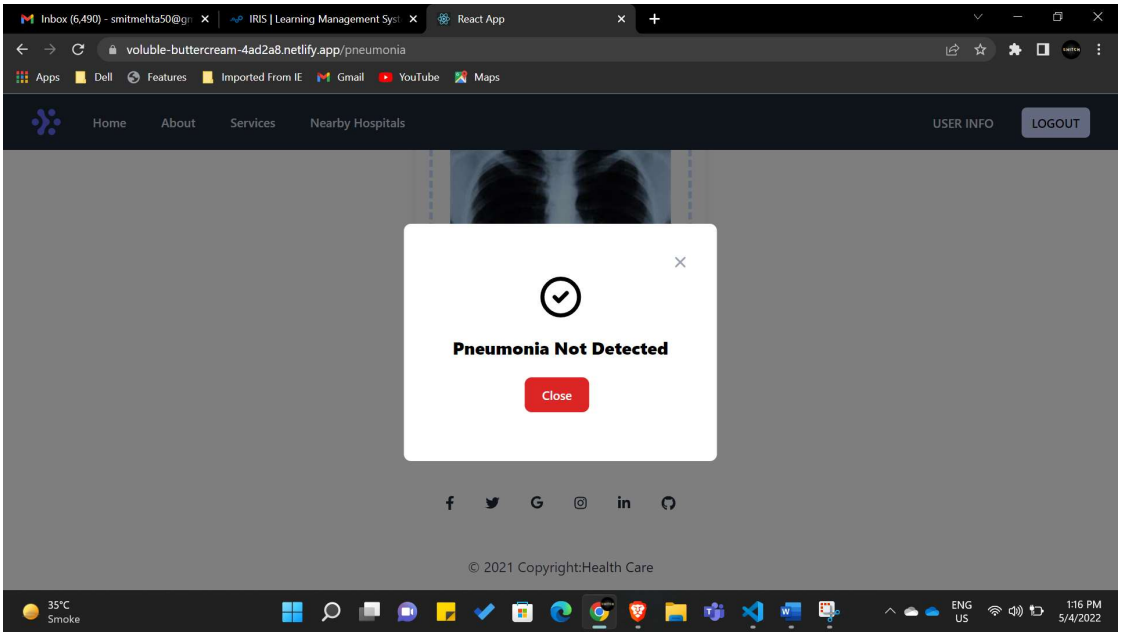


Fig 4.4: Result

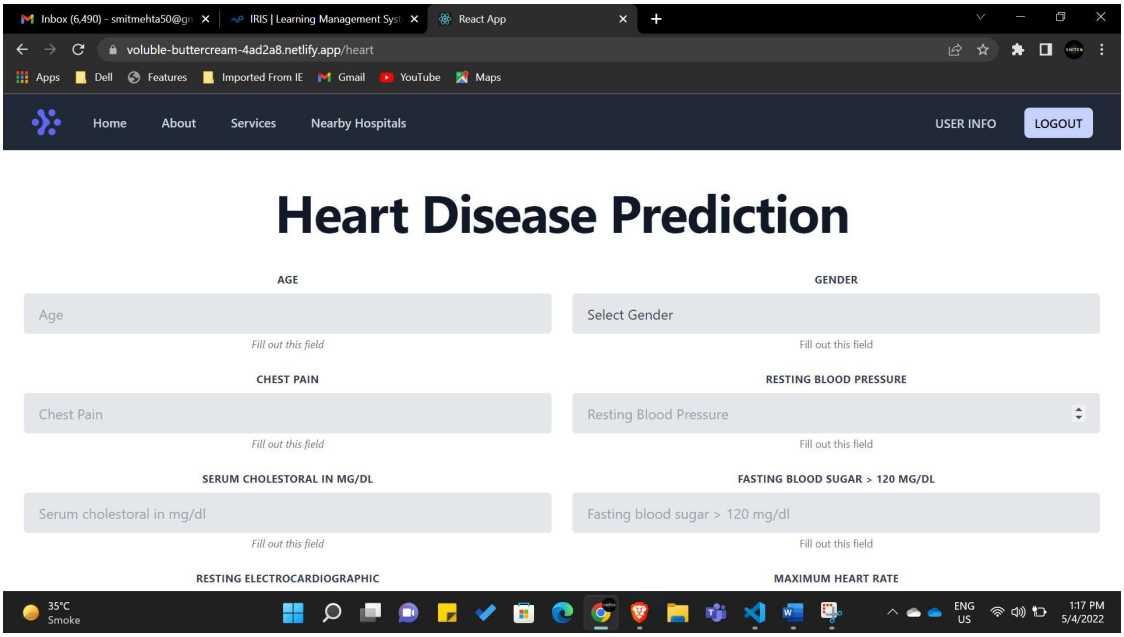


Fig 4.5: Heart Disease Prediction Page

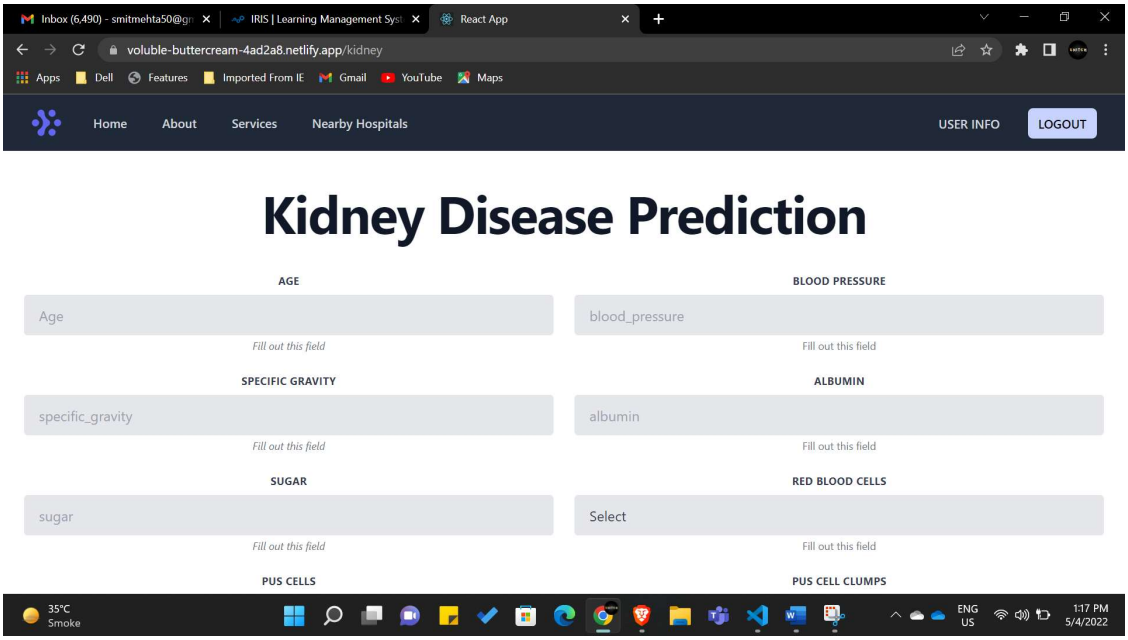


Fig 4.6: Kidney Disease Detection Page

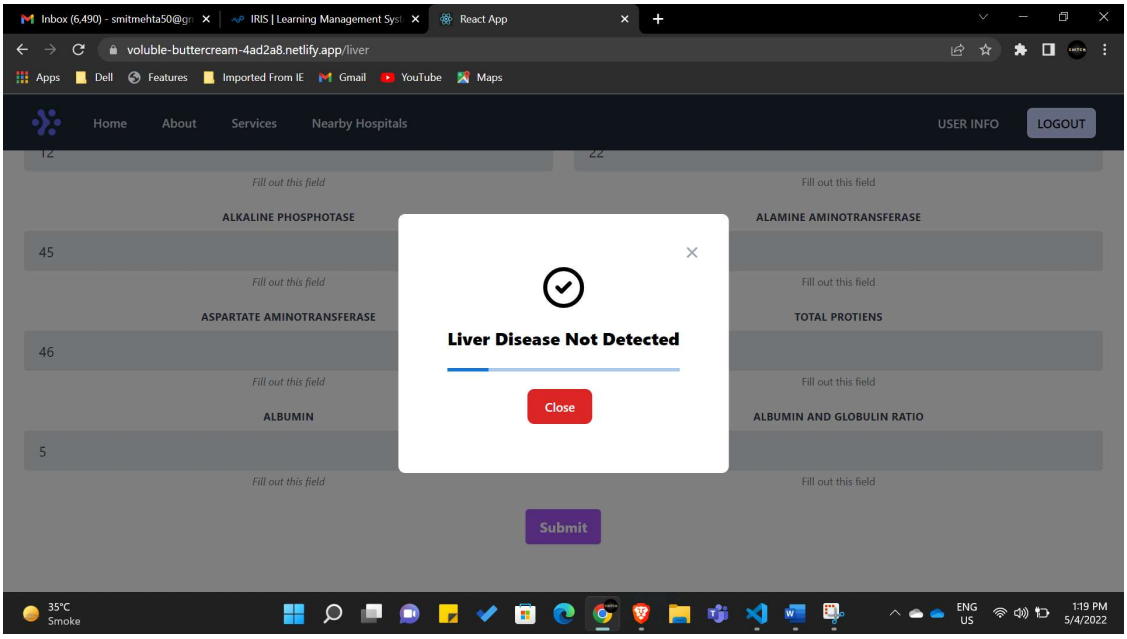


Fig 4.7: Result with Percentage-bar

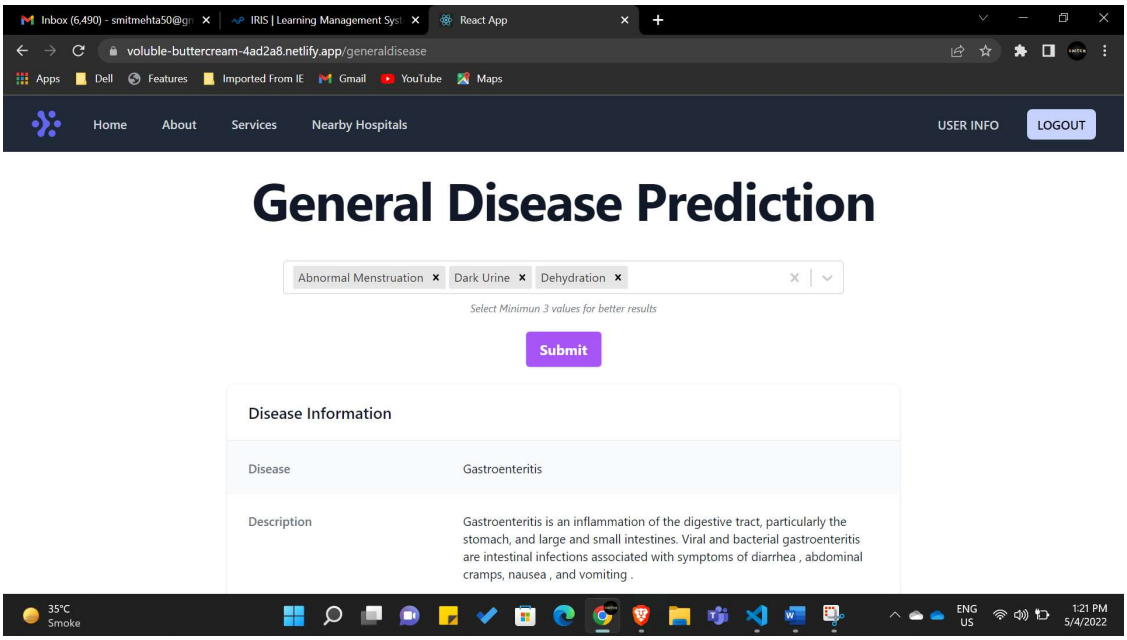


Fig 4.8: General Disease Prediction

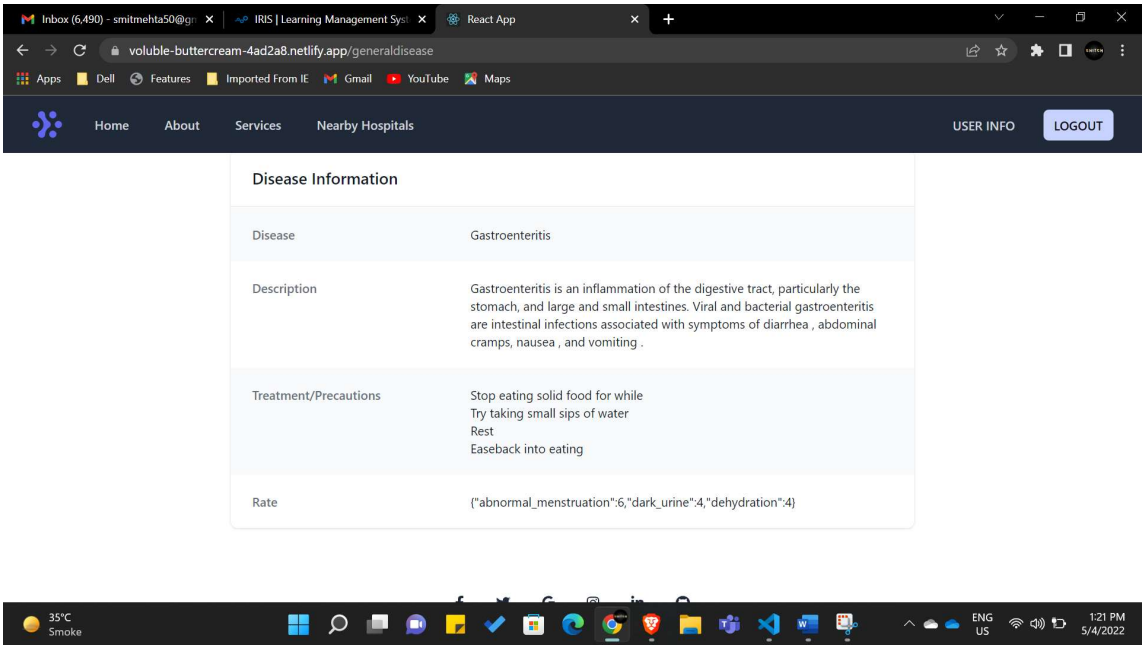


Fig 4.9: Result of General Prediction

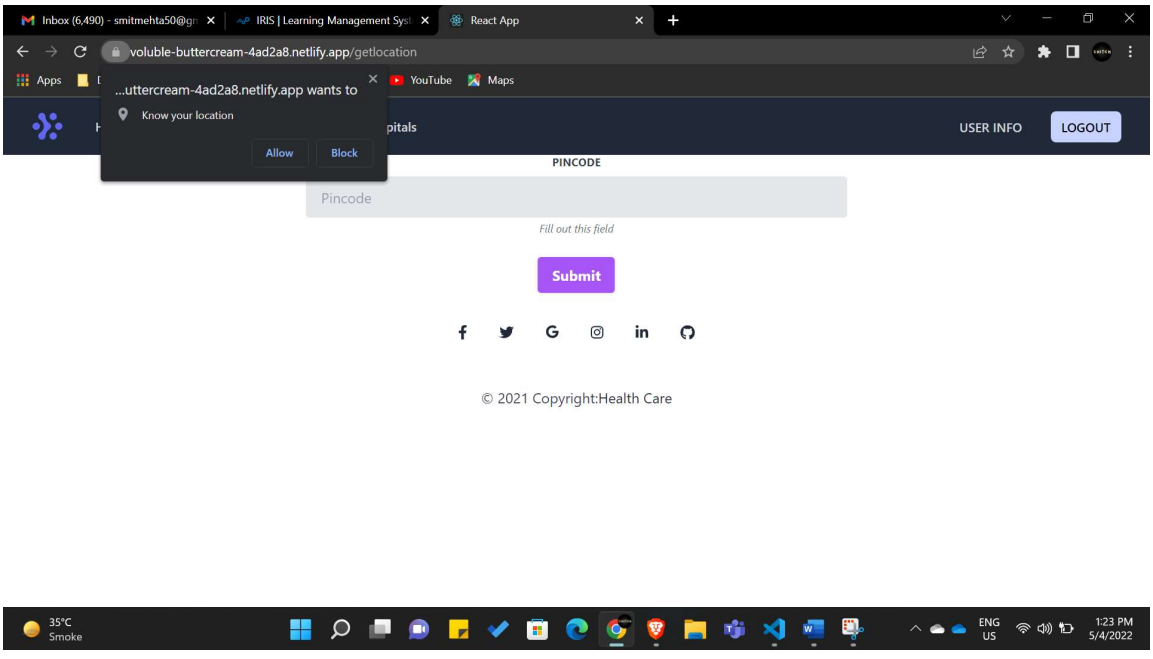


Fig 4.10: Location Access for Nearby Hospitals

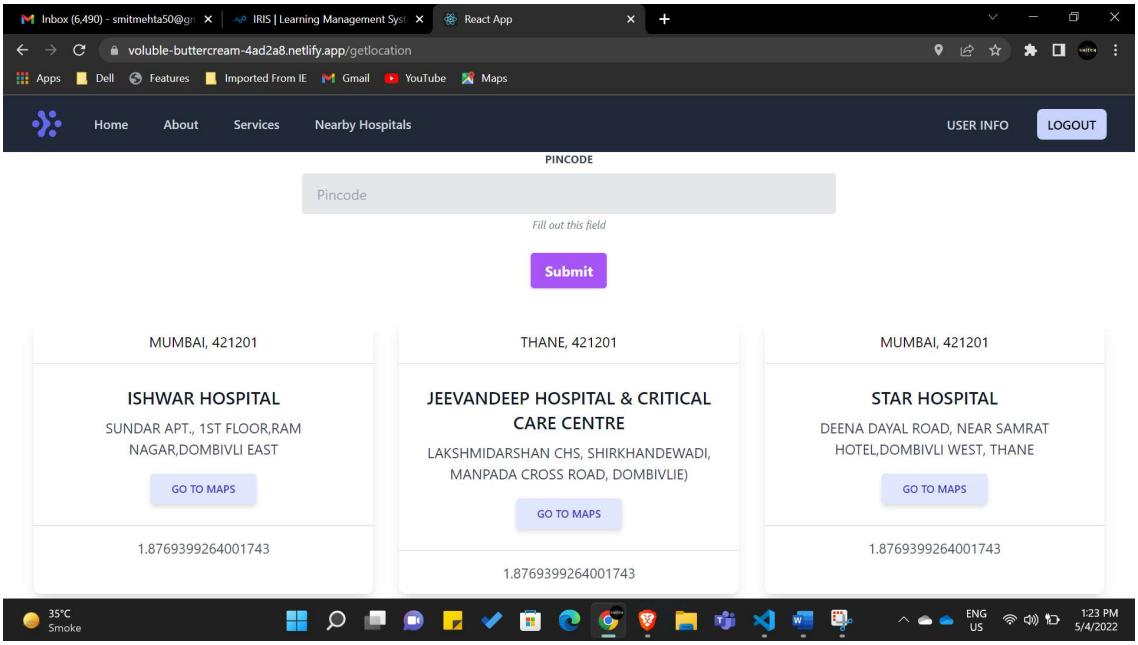


Fig 4.11: Nearby Hospitals and Hospitals Based on Pin-code Provided

CHAPTER 5

Conclusion and Scope for Future Work

This chapter leads to final conclusion and scope of the disease prediction system and future word in this field.

5.1 Conclusion

The health care web app for pneumonia, diabetes, kidney, liver, heart and general disease prediction
was successfully implemented and deployed to the server. It includes
1) Managing the data and building an API for ease of use for a patient
2) Doing proper validation of data before passing it to the model
3) Learn about other technologies and ongoing research in healthcare domain
4) Converted large deep learning models to small models using tflite for quick response and less space on cloud.
5) Added location based navigation so that based on user's location nearby hospitals will be shown.

5.2 Scope for Future Work

The 21st century has been a time of information driven choices. It is said that the sections or businesses which create more information will develop quicker and the associations which use this information to take significant choices might remain on the ball. Viable AI execution empowers medical care experts in better independent direction, recognizing patterns and advancements, and working on the proficiency of examination and clinical preliminaries.

The proposed framework is easy to use, adaptable, solid and an expandable framework.

The model can effectively train apparatus for clinical understudies and will be a delicate analytic instrument accessible for doctor.

There are numerous potential enhancements that could be investigated to work on the adaptability and precision of this forecast framework. As we have fostered a summed up framework, in later we can involve this framework for the examination of various informational collections. The presentation of the wellbeing's finding can be improved altogether by taking care of various class names in the forecast cycle, and it tends to be one more certain course of examination.

References

[1] Jobeda Jamal Khanam, Simon Y. Foo, "A comparison of machine learning algorithms for diabetes prediction", ICT Express, Volume 7, Issue 4, 2021, Pages 432-439, ISSN 2405-9595, https://doi.org/10.1016/j.icte.2021.02.004.

[2] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan and A. Mittal, "Pneumonia Detection Using CNN based Feature Extraction," 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2019, pp. 1-7, doi: 10.1109/ICECCT.2019.8869364.

[3] A. Singh and R. Kumar, "Heart Disease Prediction Using Machine Learning Algorithms," 2020 International Conference on Electrical and Electronics Engineering (ICE3), 2020, pp. 452-457, doi: 10.1109/ICE348803.2020.9122958.

[4] I. U. Ekanayake and D. Herath, "Chronic Kidney Disease Prediction Using Machine Learning Methods," 2020 Moratuwa Engineering Research Conference (MERCon), 2020, pp. 260-265, doi: 10.1109/MERCon50084.2020.9185249.

[5] L. A. Auxilia, "Accuracy Prediction Using Machine Learning Techniques for Indian Patient Liver Disease," 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 2018, pp. 45-50, doi: 10.1109/ICOEI.2018.8553682.

[6] C.L. Huang, M.C. Chen, C.J. Wang, Credit scoring with a data mining approach based on support vector machines, Expert Syst. Appl. 33 (4) (2007) 847–856, http://dx.doi.org/10.1016/j.eswa.2006.07.007.

[7] Nicholas E Ross, Charles J Pritchard, David M Rubin, and Adriano G Duse. 2006. Automated image processing method for the diagnosis and classification of malaria on thin blood smears. Medical and Biological Engineering and Computing 44, 5 (2006), 427436.

[8] Pahulpreet Singh Kohli and Shriya Arora, “Application of Machine Learning in Diseases Prediction”, 4th International Conference on Computing Communication And Automation(ICCCA), 2018.

[9] Aditi Gavhane, Gouthami Kokkula, Isha Panday, Prof. Kailash Devadkar, “Prediction

of Heart Disease using Machine Learning", Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology(ICECA), 2018.

[10] P. Yildirim, "Chronic kidney disease prediction on imbalanced data by multilayer perceptron: Chronic kidney disease prediction," 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), vol. 02, pp. 193–198, 2017.

[11] A. J. Aljaaf, D. Al-Jumeily, H. M. Haglan, M. Alloghani, T. Baker, A. J. Hussain, and J. Mustafina, "Early prediction of chronic kidney disease using machine learning supported by predictive analytics," in 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2018, pp. 1–9.

[12] Prasad Babu et. al. (2014). An implementation of hierarchical clustering on Indian Liver Patient Dataset. International Journal of Emerging Technologies in Computational and Applied Sciences. 8(6): 543-547.

[13] P. Mazaheri, A. Narouzi and A. Karimi (2015), Using Algorithms to Predict Liver Disease Classification, Electronics Information and Planning. 3 :255-259.

[14] A.S.Aneeshkumarand C.JothiVenkateswaran, "Estimating the Surveillance of Liver Disorder using Classification Algorithms", International Journal of Computer Applications (0975 – 8887), Volume 57– No.6, November 2012.

Acknowledgements

The completion of this undertaking couldn't have been possible without the participation and assistance of numerous individuals. The group members would like to express their sincere appreciation to Professor Dipak Kulkarni, from the **Electronics & Telecommunication Department**. We would like to thank principal ma'am Dr. Shubha Pandit for giving us this opportunity. This practical knowledge we gained will help us in projects assigned during corporate working. This project has helped us in developing our cognitive skills and team management abilities.