VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(15CSP85)** ON

"DISEASE RISK PREDICTION"

Submitted in Partial fulfillment of the Requirements for the VII Semester of the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

C. Chaitra (1CR16CS038)

S. Gayathri (1CR16CS052)

J. Pooja Preetha (1CR16CS058)

N. Janani (1CR16CS060)

Under the Guidance of

Mrs. Jagadishwari Palani

Asst. Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"DISEASE RISK PREDICTION"** carried out by Ms.C.Chaitra, USN 1CR16CS038, Ms. S.Gayathri, USN 1CR16CS052, Ms. J.Pooja Preetha, USN 1CR16CS058, Ms. Janani, USN 1CR16CS060, bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

.

Signature of Guide
Mrs. Jagadishwari Palani
Asst.Professor
Dept. of CSE, CMRIT

Signature of HOD
Dr. Prem Kumar Ramesh
Professor & HOD
Dept. of CSE, CMRIT

DECLARATION

We the team members of 8th semester of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the project entitled "**DISEASE RISK PREDICTION**" has been successfully completed under the guidance of Asst. Prof. Mrs. Jagadishwari Palani, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bangalore

Date: 06/06/2020

Team members:

C. Chaitra (1CR16CS038)

S. Gayathri (1CR16CS052)

J. Pooja Preetha (1CR16CS058)

N. Janani (1CR16CS060)

ABSTRACT

The Disease Risk Prediction system is an end user support and online consultation project. Here we propose a system that allows users to get instant guidance on their health issues through an intelligent health care system online. The system is fed with various symptoms and the disease/illness associated with those symptoms. The system allows user to share their symptoms and issues. It then processes user's symptoms to check for various illness that could be associated with it. Here we use some intelligent data mining techniques to guess the most accurate illness that could be associated with patient's symptoms.

Disease Prediction system has data sets collected from different health related sites. With the help of Disease Predictor the user will be able to know the probability of the disease with the given symptoms. The Disease Prediction System using data mining techniques that analyzes the risk level of the patient.

Finally, we provide a model which can be used for predictive analytics using data mining and machine learning algorithms to predict the chances of a person to be prone to a disease.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project.

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr. Prem Kumar Ramesh,** HOD, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide
Mrs. Jagadishwari Palani, Asst. Prof, Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of the project.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Prediction using traditional disease risk models usually involves a machine learning algorithm and by the use of training data with labels to train the model. In the test set, patients can be classified into groups of either high risk or low-risk. Patient's statistical information, test results and disease history are recorded in the EMR, enabling us to identify potential data-centric solutions to reduce the costs of medical case studies.

Now-a-days, people face various diseases due to the environmental condition and their living habits. So the prediction of disease at earlier stage becomes important task. But the accurate prediction on the basis of symptoms becomes too difficult for doctor. The correct prediction of disease is the most challenging task. To overcome this problem this system plays an important role to predict the disease.

In this project, we use K-Nearest Neighbor (KNN), Decision tree, CNN and Random forest machine learning algorithms for accurate prediction of disease. For disease prediction required disease symptoms dataset. In this general disease prediction the living habits of person and checkup information consider for the accurate prediction.

After general disease prediction, this system able to give us the risk associated with general disease which is lower risk of general disease or higher.

1.1 Relevance of the Project

Through this system of Risk prediction, patients need not visit the hospital instead using this system he/she can contact doctor to know risk of disease within a minutes.In case the risk is high, the patient can book the appointment with doctor and get started with diagnosis.

1.2 Problem statement

In this general disease prediction the living habits of person and checkup information consider for the accurate prediction. After general disease prediction, this system able to give us the risk associated with general disease which is lower risk of general disease or higher.

1.3 Objectives

The main objective of this project is to get the risk level (low, high) of the diseases namely (COPD, lung cancer, diabetes, heart disease) using multiple machine learning algorithms namely KNN, CNN ,Random forest , decision tree .After prediction of risk by the doctor, the report is generated and mailed to patient’s valid email-id, also can book appointment with specialized doctor.

1.4 Scope of the Project

This web application can be additionally upgraded in an Android application. This will be accessible to clients on versatile premise and its utilization can be additionally expanded. Likewise, highlight like getting the specialist online on a visit with the goal that patients can straightforwardly converse with the concerned specialists.

The modules doing malignant growth examination can be coordinated to discover how close the individual related with disease is.This will make this web application unsurprising in obvious sense.

1.5 Methodology

CNN algorithm: A convolutional neural network (CNN) is a specific type of artificial neural network that uses perceptrons, a machine learning unit algorithm, for supervised learning, to analyse data. CNNs apply to image processing, natural language processing and other kinds of cognitive tasks.

KNN algorithm: KNN is the one of the simplest algorithm used in machine learning for regression and classification problem. KNN uses data and classify the new data points based on the similarity measures (distance function).

Decision tree: Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning. Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions

Random forest: The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

User stories:

1. As a registered user I want to change my password so I can keep my account secure.
2. As a website visitor I want to subscribe to the mailing list so I can get updates through mail.
3. As an admin user I want to disable a user so I can prevent unauthorized access.

CHAPTER 2

LITERATURE SURVEY

Literature survey is mainly carried out to analyse the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

[2.1] Convolutional Neural Network (CNN)-based multimodal disease

With big data growth in biomedical and healthcare communities, accurate analysis of medical data benefits early disease detection, patient care and community services. However, the analysis accuracy is reduced when the quality of medical data is incomplete .We propose a new convolutional neural network (CNN)-based multimodal disease of cerebral algorithm using structured and unstructured data from hospital[1]. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8%.

The CNN-MDRP algorithm, the training process is divided into two parts:

1) Training word embedding: the text data of all patients is extracted in the hospital.

2) Training parameters of CNN-MDRP: In CNN-MDRP algorithm, the specific training parameters are set and finally risk assessment of whether the patient suffers from the disease.

[2.2] Gradient-Boosted (GDB) and Random forests

This study proposed an ECG (Electrocardiogram) classification approach using machine learning based on several ECG features. The proposed approach is implemented using ML-libs and Scala language on Apache Spark framework. MLlib is Apache Spark’s scalable machine learning library.

Discrete Wavelet Transform (DWT) was used to get features from downloaded annotations. All chosen features were used in classification model. Algorithms used in this project is Gradient-Boosted (GDB) and Random forests or random decision forests[2]. Classification process is using some features of heartbeats and machine learning classification algorithms with local host pc working using only one node, which are crucial for diagnosis of cardiac arrhythmia.

[2.3] RGB to Grayscale Conversion and Bilateral Filtering

Cancer is the most sacrificed disease all over the globe which is clinically referred as a malevolent neoplasm, it is a multifarious genetic disease that is caused primarily by the environmental factors. For tumour detection various techniques such as MRI(Magnetic Resonance Imaging), CT(Computerised tomography) scan and Microwave are CT(Computerised tomography) scan available among mentioned techniques MRI delivers the best images as it has higher resolution.

In this paper a method is introduced to detect bone cancer by using machine learning algorithm. The main objective is to detect the tumour present in the bone. Stages for the proposed system are RGB to Grayscale Conversion and Bilateral Filtering[3].After the tumour identification process it is last step to detect the tumour which can be carried out by using the MATLAB function for connected components which will simply select out the area with maximum connected component and the remaining area will be discarded.

[2.4] CART (Classification and Regression Trees) machine learning algorithm

Diabetes Mellitus or Diabetes has been describe as worse than Cancer and HIV (Human Immunodeficiency Virus). It develops when there are high blood glucose levels over a protracted period. Recently, it's been quoted as a risk factor for developing Alzheimer, and a number one cause for blindness & renal failure. Prevention of the disease may be a hot topic for research within the healthcare community. Many techniques are discovered to seek out the causes of diabetes and cure it.

This research paper may be a discussion on establishing a relationship between diabetes risk likely to be developed from an individual's daily lifestyle activities like his/her eating habits, sleeping habits, physical activity alongside other indicators like BMI (Body Mass Index), waist circumference etc. Initially, a Chi-Squared Test of Independence was performed followed by application of the CART (Classification and Regression Trees) machine learning algorithm[4] on the info and eventually using Cross-Validation, the bias within the results was removed.

[2.5] K-Nearest Neighbors (KNN) and Convolutional neural network (CNN)

Now-a-days, people suffer from various diseases due to their living habits. Therefore the prediction of disease at earlier stage becomes important task. But the accurate prediction on the idea of symptoms becomes too difficult for doctor. The right prediction of disease is that the most challenging task to beat this problem data processing plays a crucial role to predict the disease. With the assistance of disease data, data processing finds hidden pattern information within the huge amount of medical data.

They proposed general disease prediction supported symptoms of the patient. For the disease prediction, we use K-Nearest Neighbors (KNN) and Convolutional neural network (CNN) [5]machine learning algorithm for accurate prediction of disease. For disease prediction required disease symptoms dataset. During this general disease prediction the living habits of person and check-up information consider for the accurate prediction. The accuracy of general disease prediction by using CNN is 84.5% which is quite KNN algorithm and therefore the reform the time and the memory requirement is additionally more in KNN than CNN. After general disease prediction, this technique ready to gives the danger related to general disease which is lower risk of general disease or higher.

[2.6] Cloud to end fusion big health application system architecture

The world is facing problems, such as uneven distribution of medical resources, the growing chronic diseases, and the increasing medical expenses. Blending the latest information technology into the healthcare system will greatly mitigate the problems. This paper presents the big health application system based on the big data. This paper presents the cloud to end fusion big health application system[6] architecture which is supported by health big data.

The whole system includes health perception layer, transport layer, big health cloud service layer. Cloud service application sub-layer is responsible for the control of all sensor nodes, the visual presentation of data, the scheme of work and business. In many developing countries, rural health resources are of serious shortage, so it is necessary using advanced detection technology to integrate rural medical resources and establishing ``big health project" suitable for rural area.

In this paper, we present a big health application system which is based on the health big data. We introduce the big health care system's architecture, technology challenges and several typical large health applications.

[2.7] Predictions of outbreaks in diseases using the decision tree

Author has presented the data mining concept “Disease Prediction by using Machine Learning”. The data mining best growth of the stage is develops that technique into the healthcare basis, the data analysis is an important part of every field. The data mining is predicts the information for healthcare is called rapid growth of medical care field. The existing one is designed the purpose of (i) analyse, (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[7], 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 choose the raining set like medical patient symptoms, than, use the decision tree, then, predicted, finally give the symptoms of patient and get the accurate result for disease prediction.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. In order to fully understand one’s project, it is very important that they come up with an SRS listing out their requirements, how are they going to meet it and how will they complete the project. SRS also functions as a blueprint for completing a project with as little cost growth as possible.

SRS is often referred to as the parent document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of source.

3.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

• Collecting Date sets and data pre-processing is performed for that data set

• Developing a probabilistic modelling and Machine learning approach for disease prediction.

• Training and experimentation on datasets. Development and analysis on real life scenario.

3.2 Non Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge using CNN in the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and performance. Non-functional requirements arise through the user needs , organizational policies and the need for interoperability with other software and hardware systems.

3.2.1 Some Non-Functional Requirements are as follows

- Reliability: The structure must be reliable and strong in giving the functionalities. The movements must be made unmistakable by the structure when a doctor has revealed a couple of enhancements. The progressions made by the Programmer must be Project pioneer and in addition the Test designer.
- Maintainability: The system watching and upkeep should be fundamental and focus in its approach. There should not be an excess of occupations running on diverse machines such that it gets hard to screen.
- Performance: The framework will be utilized by numerous representatives all the while. Since the system will be encouraged on a single web server with a lone database server outside of anyone's ability to see, execution transforms into a significant concern. The structure should not capitulate when various customers would use everything the while. It should allow brisk accessibility to each and every piece of its users. For instance, if two test specialists are all the while attempting to report the vicinity of a bug, then there ought not to be any irregularity at the same time.
- Portability: The framework should to be effectively versatile to another framework. This is obliged when the web server, which is facilitating the framework gets adhered because of a few issues, which requires the framework to be taken to another framework.
- Scalability: The framework should be sufficiently adaptable to include new functionalities at a later stage. There should be a run of the mill channel, which can oblige the new functionalities.

- Flexibility: Flexibility is the capacity of a framework to adjust to changing situations and to adapt to changes to business approaches and rules.

3.3 Hardware system Configuration

System	: Pentium IV 2.4 GHz.
Hard Disk	: 500 GB.
Floppy Drive	: 1.44 Mb.
Monitor	: 15 VGA Colour.
Mouse	: Logitech.
Ram	: 4 GB/8 GB for faster output

3.4 Software Requirements

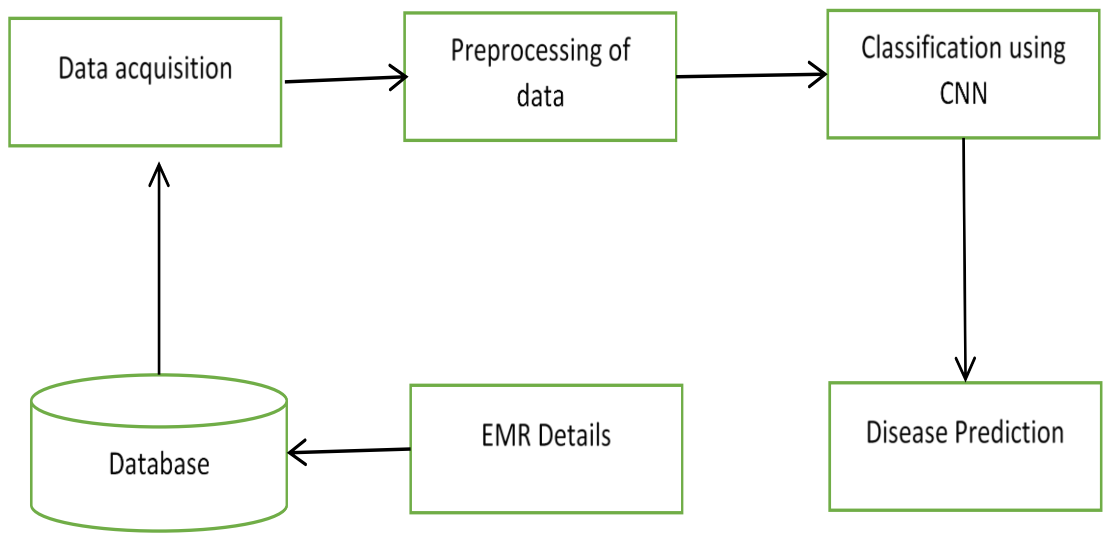
Operating system	: Windows
Coding Language	: Python, Django
Data Base	: Sqlite3
Tools	:Anaconda Distribution
IDE	: Pycharm IDE Community Edition 2018.3.3
Technology used	: Machine Learning
Front End	: HTML5, CSS, Bootstrap4

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Architecture

Fig4.1. Represent system architecture of proposed system. Initially we take disease dataset from Kaggle machine learning website and different diseases have different attributes and with its symptoms. After that pre-processing is performed on that dataset for cleaning that is removing unnecessary attributes and that is used as training dataset. After that feature extracted and selected. Then we classify that data using classification techniques such as CNN, DT, KNN and Random forest. Based on machine learning we can predict accurate disease prediction.



4.2 Datasets

The analysis accuracy is reduced when the quality of medical data is incomplete. Moreover, different regions exhibit unique characteristics of certain regional diseases, which may weaken the prediction of disease outbreaks. In this paper, we streamline machine learning algorithms for effective prediction of chronic disease outbreak in disease-frequent communities. We experiment the modified prediction models over real-life hospital data downloaded from Kaggle.

4.2.1Sample Dataset of Heart Disease

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

Attribute Information:

Age ,sex,chest pain type (4 values),resting blood pressure,serum cholestoral in mg/dl,fasting blood sugar > 120 mg/dl,resting electrocardiographic results (values 0,1,2),maximum heart rate achieved,exercise induced angina,oldpeak = ST depression induced by exercise relative to rest,the slope of the peak exercise ST segment,number of major vessels (0-3) colored by flourosopy,thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

4.2.2 Sample Dataset of Diabetes

Pregnant	Glucose	Bp	Skin	Insulin	Bmi	Pedigree	Age	Label
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

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, Outcome. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

4.2.3 Sample Dataset of Lung Cancer

Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity	Smoking	Passive Smoker	Chest Pain
33	1	2	4	5	4	3	2	2	4	3	2	2
17	1	3	1	5	3	4	2	2	2	2	4	2
35	1	4	5	6	5	5	4	6	7	2	3	4
37	1	7	7	7	7	6	7	7	7	7	7	7
46	1	6	8	7	7	7	6	7	7	8	7	7

Coughing of Blood	Fatigue	Weight Loss	Shortness of Breath	Wheezing	Swallowing Difficulty	Clubbing of Finger Nails	Frequent Cold	Dry Cough	Snoring	Level	total
4	3	4	2	2	3	1	2	3	4	Low	61
3	1	3	7	8	6	2	1	7	2	Medium	70
8	8	7	9	2	1	4	6	7	2	High	105
8	4	2	3	1	4	5	6	7	5	High	121
9	3	2	4	1	4	2	4	2	3	High	111

The attributes considered to predict lung cancer are the age ,gender ,alcohol consumption , direct to dust allergy , heredity ,pain in chest , cough intensity etc.

4.2.4 Sample Dataset of Copd

age	gender	weight	lipcolor	FEV1	smoking intensity	temperature	label
20	0	48	1	0.24	0.8	103	severe
70	1	58	0	0.54	0.6	98	moderate
25	0	50	1	0.3	0.7	104	severe
30	1	45	1	0.45	0.9	103	severe
50	0	48	0	0.8	0.1	97	mild
56	1	54	1	0.5	0.6	101	moderate
73	1	67	0	0.56	0.5	100	moderate
46	1	44	0	0.36	0.7	104	severe
87	0	65	1	0.43	0.9	104	severe
22	0	32	1	0.45	1	103	severe
24	0	38	1	0.5	0.4	99	moderate
34	1	45	0	0.6	0.5	100	moderate
62	1	56	1	0.63	0.4	99	moderate

Chronic obstructive pulmonary disease (COPD) is a common lung disease. Having COPD makes it hard to breathe. Attributes considered for risk prediction are age , gender, weight index ,fever , temperature , smoking intensity of a person.

4.3 Use Case Diagram

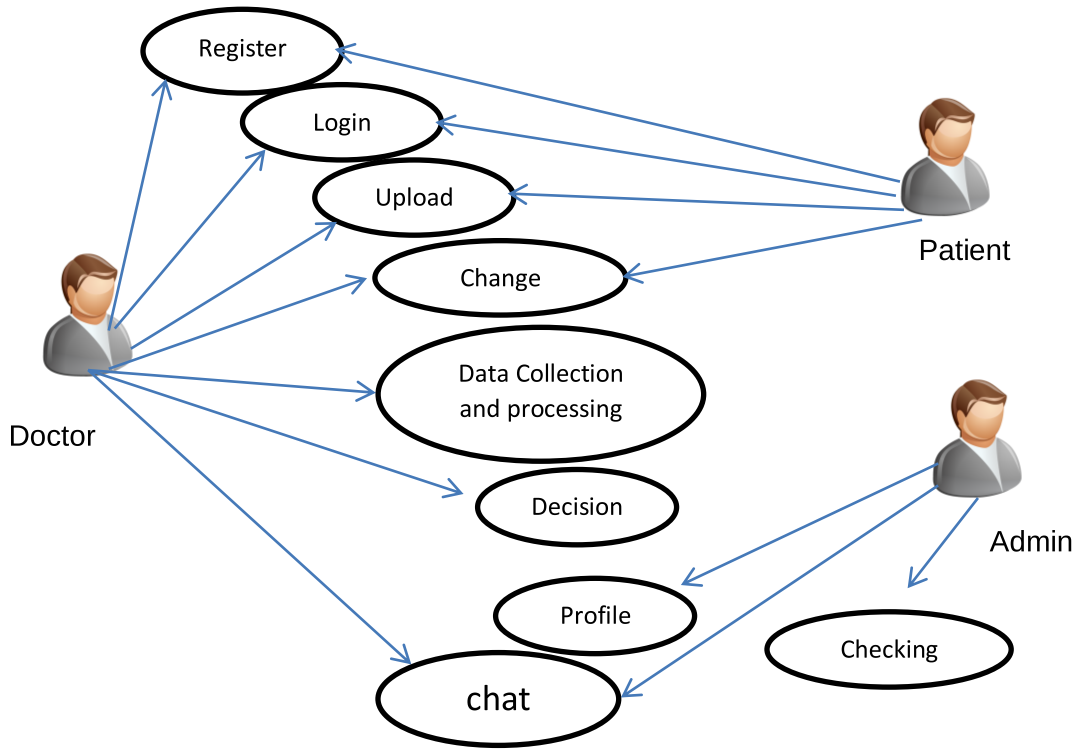


Fig 4.3 use case diagram

In figure 4.3, doctor can register, login, upload EMR data, collect data and make decisions. Patient can also register, login, upload lab test results and chat with doctor for any other queries. Admin will keep track of profiles from both patient and doctor side. Admin can also view previous treatments given by the doctor.

CHAPTER 5

IMPLEMENTATION

Step 1: Data collection and dataset Preparation: This will involve collection of medical information artifacts from various sources like hospitals, discharge slips of patients to form an EMR dataset

Step 2: Developing a probabilistic modeling and Machine learning approach for Disease Prediction In this step probabilistic modeling and deep learning approach based on this to be developed it will run effectively on extensive databases of healthcare.

Step 3: Training and experimentation on datasets The Disease Prediction model will be trained on the dataset of diseases to do the prediction accurately and produce Confusion matrix.

Step 4: Deployment and analysis on real life scenario. The trained and tested prediction model will be deployed in a real-life scenario made by the human experts & will be leveraged for further improvement in the methodology and will follow the above architecture.

5.1 Random Forest (Heart Disease Prediction)

```
def predHeart(request):
    cp = request.POST['cp']
    trestbps = request.POST['trestbps']
    chol = request.POST['chol']
    fbs = request.POST['fbs']
    exang = request.POST['exang']
    ca = request.POST['ca']
    pemail1 = request.POST['pemail']
    docname1 = request.POST['docname']
    reportof = request.POST['reportof']
    df2 =
pd.read_csv(r"G:\emradmin1\emrnew\csv\heart.csv")
    X = df2[['cp', 'trestbps', 'chol',
'fbs','exang','ca']]
    Y = df2[['target']]
   X_train, X_test, Y_train, Y_test = train_test_split(X,
Y, test_size = 0.25,      random_state = 1)
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)
    forest = RandomForestClassifier(n_estimators = 10,
criterion = 'entropy',     random_state = 1)
    forest.fit(X_train, Y_train)
    model = forest
    model.score(X_train, Y_train)
    new_input2 =np.array([[cp, trestbps, chol, fbs,
exang,ca]])
    pred  = model.predict(new_input2)
    if pred==[[0]]:
        pred1="LOW"
    else:
        pred1="HIGH"
    return
render(request,'doctor/heartpred.html',{"data2":pred1,"cp
":cp,"trestbps":
trestbps,"chol":chol,"fbs":fbs,"exang":exang,"ca":ca,"pem
ail1":pemail1,"docname1":docname1,"reportof":reportof})
```

Random Forest algorithm is a supervised classification algorithm. First, start with the selection of random samples from a given dataset. Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree. In this step, voting will be performed for every predicted result. At last, select the most voted prediction result as the final prediction result.

5.2 Convolution Neural Network (Diabetes Prediction)

```
def predicDiabetes(request):
    glucose = request.POST['Glucose']
    Blood_Pressure = request.POST['Blood_Pressure']
    Insulin = request.POST['Insulin']
    BMI = request.POST['BMI']
    dpedgree=request.POST['DiabetesPedigreeFunction']
    pemail1 = request.POST['pemail']
    docname1 = request.POST['docname']
    reportof = request.POST['reportof']
    df1 =
pd.read_csv(r"G:\emradmin1\emrnew\csv\diabetes.csv")
    X = df1[['Glucose', 'BloodPressure', 'Insulin',
'BMI', 'DiabetesPedigreeFunction']]
    Y = df1[['Outcome']]
    min_max_scaler = preprocessing.MinMaxScaler()
    X_scale = min_max_scaler.fit_transform(X)
    X_train, X_test, y_train, y_test =
train_test_split(X_scale, Y, test_size=0.2, random_state
= 4)
    model = Sequential()
    model.add(Dense(15, input_dim=5, activation='relu'))
    model.add(Dense(10, activation='relu'))
    model.add(Dense(21, activation='relu'))
    model.add(Dense(8, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy',
optimizer='adam', metrics =          ['accuracy'] )
    model.fit(X, Y, epochs=1000, batch_size=10)
    scores=model.evaluate(X,Y)
    print("\n%s: %.2f%%"
%(model.metrics_names[1],scores[1]*100))
    new_input2 =np.array([[glucose, Blood_Pressure,
Insulin, BMI,dpedgree]])
    predictions2 = model.predict_classes(new_input2)
    if predictions2==[[0]]:
        pred="LOW"
    else:
        pred="HIGH"
    return
render(request,'doctor/diaPred.html',{"data22":pred,"gluc
ose":glucose,"Blood_Pressure":Blood_Pressure,"Insulin":In
sulin,"BMI":BMI,"dpedgree":dpedgree,"pemail1":pemail1,"do
cname1":docname1,"reportof":reportof})
```

At each connection, you are feeding the value forward, while the value is multiplied by a weight and a bias is added to the value. It is generally common to use a rectified linear unit (ReLU) for hidden layers, a sigmoid function for the output layer in a binary classification problem. This is done by using optimization methods (also called optimizer) like the gradient descent in order to reduce the error between the computed and the desired output . The Sequential model is a linear stack of layers, where you can use the large variety of available layers in Keras. Before you can start with the training of the model, you need to configure the learning process. This is done with the .compile() method. This method specifies the optimizer and the loss function. Iterations are commonly called epochs. Another parameter you have to your selection is the batch size. The batch size is responsible for how many samples we want to use in one epoch, which means how many samples are used in one forward/backward pass. the .evaluate() method to measure the accuracy of the model.

5.3 Decision Tree (Diabetes Prediction)

```
def predicDiabetes_dec(request):
    a = request.POST['Glucose']
    b = request.POST['Blood_Pressure']
    c = request.POST['Insulin']
    d = request.POST['BMI']
    e = request.POST['DiabetesPedigreeFunction']
    pemail1= request.POST['pemail']
    docname1 = request.POST['docname']
    reportof = request.POST['reportof']
    df = pd.read_csv(r"G:\EmrNew - 
Copy\EmrNew\emrnew\csv\diabetes.csv")
    X = df[['Glucose', 'BloodPressure', 'Insulin', 'BMI', 
'DiabetesPedigreeFunction']]
    Y = df[['Outcome']]
    X_train, X_test, Y_train, Y_test = 
train_test_split(X, Y, test_size=0.3, random_state=1)
    tree = DecisionTreeClassifier(max_leaf_nodes=6, 
random_state=0)
    tree.fit(X_train, Y_train)
    prediction = tree.predict([[a,b,c,d,e]])
    predi=tree.predict(X_test)
    print("Accuracy Score of decision tree:", 
accuracy_score(Y_test, predi)*100)
    if prediction ==[1]:

        pred="HIGH"
    else :
        pred="LOW"
```

```
return render(request, 'doctor/diaPred_dec.html',{
"data22": pred, "glucose": a, "Blood_Pressure": b,
"Insulin": c,"BMI": d, "dpedgree": e, "pemail1": pemail1,
"docname1": docname1, "reportof": reportof})
```

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. Divide given columns into two types of variables dependent(or target variable) and independent variable(or feature variables). To understand model performance, dividing the dataset into a training set and a test set is a good strategy.Create a Decision Tree Model using Scikit-learn. Accuracy can be computed by comparing actual test set values and predicted values. Supported criteria are “gini” for the Gini index and “entropy” for the information gain. In Scikit-learn, optimization of decision tree classifier performed by only pre-pruning. Maximum depth of the tree can be used as a control variable for pre-pruning. attribute selection measures such as Information Gain, Gain Ratio, and Gini Index, decision tree model building, visualization and evaluation on diabetes dataset using Python Scikit-learn package.

5.4 Random Forest (Diabetes Prediction)

```
def predicDiabetes_ran(request):
    glucose = request.POST['Glucose']
    Blood_Pressure = request.POST['Blood_Pressure']
    Insulin = request.POST['Insulin']
    BMI = request.POST['BMI']
    dpedgree = request.POST['DiabetesPedigreeFunction']
    pemail1 = request.POST['pemail']
    docname1 = request.POST['docname']
    reportof = request.POST['reportof']
    df = pd.read_csv(r"G:\EmrNew -
Copy\EmrNew\emrnew\csv\diabetes.csv")
    x = df[['Glucose', 'BloodPressure', 'Insulin', 'BMI',
'DiabetesPedigreeFunction']]
    y = df[['Outcome']]
    Standard = StandardScaler()
    x = Standard.fit_transform(x)
    x_train, x_test, y_train, y_test =
train_test_split(x, y, test_size=0.20, random_state=4)
    randomforest = RandomForestClassifier()
    randomforest.fit(x_train, y_train)
    ss = randomforest.predict(x_test)
    score_randomforest = round(accuracy_score(ss, y_test)
* 100, 2)
    print("The accuracy score of Random Forest is ",
score_randomforest, "%")
    new_input2 =
np.array([[glucose,Blood_Pressure,Insulin,BMI,dpedgree]])
    pred = randomforest.predict(new_input2)
    if pred == [[1]]:
        pred1 = "HIGH"
    else:
        pred1 = "LOW"
    return render(request, 'doctor/diaPred_ran.html',
{"data22": pred1, "glucose":glucose , "Blood_Pressure":
Blood_Pressure, "Insulin": Insulin,"BMI": BMI,
"dpedgree": dpedgree, "pemail1": pemail1, "docname1":
docname1, "reportof": reportof})
```

5.5 Decision Tree (Lung Cancer)

```
def predict(request):
    a = request.POST['Genetic_Risk']
    b = request.POST['Occupational_hazard']
    c = request.POST['chest_pain']
    d = request.POST['chronic_lung_cancer']
    e = request.POST['clubbing_of_finger_nail']
    f = request.POST['coughing_of_blood']
    g = request.POST['dry_cough']
    h = request.POST['fatigue']
    i = request.POST['passive_smoker']
    j = request.POST['smocking']
    k = request.POST['weight_loss']
    pemail1 = request.POST['pemail']
    docname1 = request.POST['docname']
    reportof = request.POST['reportof']
    lists =[a,b,c,d,e,f,g,h,i,j,k]
    df =
pd.read_csv(r"G:\emradmin1\emrnew\csv\lungcancer.csv")
    X_train = df[['Genetic Risk','OccuPational
Hazards','Chest Pain','chronic Lung Disease','Clubbing of
Finger Nails','Coughing of Blood','Dry
Cough','Fatigue','Passive Smoker','Smoking','Weight
Loss']]
    Y_train = df[['Level']]
    tree = DecisionTreeClassifier(max_leaf_nodes=6,
random_state=0)
    tree.fit(X_train, Y_train)
    prediction = tree.predict([[a,b,c,d,e,f,g,h,i,j,k]])
    return
render(request,'doctor/predict.html',{"data":prediction,"
lists":lists,"a1":a,"b1":b,"c1":c,"d1":d,"e1":e,"f1":f,"g
1":g,"h1":h,"i1":i,"j1":j,"k1":k,"pemail1":pemail1,"docna
me1":docname1,"reportof":reportof})
```

Attribute selection measures such as Information Gain, Gain Ratio, and Gini Index, decision tree model building, visualization and evaluation on diabetes dataset using Python Scikit-learn package. Feed a dataset, containing a number of training instances, with a set of features and a target.Train the decision tree classification or regression models with the help of DecisionTreeClassifier () or DecisionTreeRegressor () methods, and add the required criterion while building the decision tree model .DecisionTreeClassifier (): It is nothing but the decision tree classifier function to build a decision tree model in Machine Learning using Python. The DecisionTreeClassifier() function looks like this:

DecisionTreeClassifier (criterion = ‘gini’, random_state = None, max_depth = None, min_samples_leaf =1)

5.6 K-Nearest Neighbour (COPD)

```
def predCopd(request):
    lipcolor = request.POST['lipcolor']
    FEV1 = request.POST['FEV1']
    smoking_intensity = request.POST['smoking_intensity']
    temperature = request.POST['temperature']
    pemail1 = request.POST['pemail']
    docname1 = request.POST['docname']
    reportof = request.POST['reportof']
    dataset = pd.read_csv( r"G: \emradmin1 \emrnew \ csv
\copd.csv")
    X = dataset[[ 'lipcolor','FEV1','smoking
intensity','temperature' ]]
    y = dataset[['label']]
    x_train, x_test, y_train, y_test =
train_test_split(X, y, test_size=0.3, random_state=43)

    model = KNeighborsClassifier(n_neighbors=14)
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    print(y_pred)
    new_input3 = np.array([[lipcolor, FEV1,
smoking_intensity , temperature]])
    y_pred1 = model.predict(new_input3)
    print(y_pred1)


    return render(request, 'doctor/predCopd.html'
,{"data3":y_pred1,"lipcolor":lipcolor,"FEV1":FEV1,"smokin
g_intensity":smoking_intensity,"temperature":temperature,
"pemail1":pemail1,"docname1":docname1,"reportof":reportof
})
```

The number of neighbors(K) in KNN is a hyperparameter that you need choose at the time of model building. Let's first create our dataset. Here we need two kinds of attributes or columns in our data: Feature and label. The reason for two type of column is "supervised nature of KNN algorithm".First, import the KNeighborsClassifier module and create KNN classifier object by passing argument number of neighbors in KNeighborsClassifier() function.Let's split dataset by using function train_test_split(). You need to pass 3 parameters features, target, and test_set size. Additionally, you can use random_state to select records randomly.Then, fit your model on the train set using fit() and perform prediction on the test set using predict().

Accuracy can be computed by comparing actual test set values and predicted values.

5.7 Front-End / Back-End implementation Details

5.7.1 Report Generation:

```
a=PatientReg.objects.get(pemail = pemail1)
fname = a.pname
basename = "copdReport"
suffix =
datetime.datetime.now().strftime("%y%m%d_%H%M%S")
filename2 = "_".join([basename, suffix])
loc="static/report/"+filename2+".pdf"
#file naming is in above
doc =
SimpleDocTemplate(loc,pagesize=letter,rightMargin=72,left
Margin=72 ,topMargin=72,bottomMargin=18) #output contents
to a file
Story=[]
logo = "static/images/seal.png"
formatted_time = time.ctime()#current time is returned
full_name = fname
address_parts = [pemail1]
im = Image(logo, 2*inch, 2*inch)
Story.append(im)
styles=getSampleStyleSheet()
styles.add(ParagraphStyle(name='Justify',
alignment=TA_JUSTIFY))
ptext = '<font size="12">%s</font>' % formatted_time
Story.append(Paragraph(ptext, styles["Normal"]))
Story.append(Spacer(1, 12))
# Create return address
for part in address_parts:
    ptext = '<font size="12">%s</font>' % part.strip()
    Story.append(Paragraph(ptext, styles["Normal"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">Dear <b> %s </b>:</font>' %
full_name.split()[0].strip()
Story.append(Paragraph(ptext, styles["Normal"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">We have generated the report
of<b> %s </b>, we found the your risk of <b> %s </b> is \
                    =<b> %s </b>, we recommend you to
care for your health, because your this health will\
                    help you to live the happy life. We
```

```
are attaching the report here</font>' %
(reportof1,reportof1,data3)
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">\
    ------------------------------------------------------
---------------------------------------------------------
---\
    <b>Patient name</b> = %s    || <b>Doctor name</b>=%s
\
    </font>' % (pemail1,docname1)
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">\
---------------------------------------------------------
--------------------------------------------------------\
<b>Report of</b> = %s                  \
</font>' % (reportof1)
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">\
---------------------------------------------------------
--------------------------------------------------------\
<b>Lip color</b>= %s     ||<b> FEV1</b>= %s   ||   <b>
smoking_intensity</b>=%s               \
</font>' % (lipcolor1,FEV1,smoking_intensity)
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">\
---------------------------------------------------------
---------------------------------------------------------
\
<b>temperature</b>= %s                  \
</font>' % (temperature)
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">\
---------------------------------------------------------
--------------------------------------------------------\
Your rishk about the <b>%s</b>= <b>%s </b>
\
</font>' % (reportof1 , data3)
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">\
---------------------------------------------------------
--------------------------------------------------------\
</font>'
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">Thank you very much and we look
forward to serving you.</font>'
Story.append(Paragraph(ptext, styles["Justify"]))
Story.append(Spacer(1, 12))
ptext = '<font size="12">Sincerely,</font>'
Story.append(Paragraph(ptext, styles["Normal"]))
Story.append(Spacer(1, 48))
ptext = '<font size="12">%s</font>' %(docname1)
Story.append(Paragraph(ptext, styles["Normal"]))
Story.append(Spacer(1, 12))
doc.build(Story)
```

Report generation part takes care of generating a formatted report which includes the details of the patient , specialized doctor incharge ,attribute and its measured values , test result . The entire report is formatted under pdf format . The entire procedure is done using python ReportLab module .In the above code we are explicitly storing the generated copy within the doctor system and one that has to be sent to patient for further diagnosis process .

5.7.2 Email Generation:

```
#applying the smtp server here
fromaddr = "healthcaremail1998@gmail.com"
toaddr = pemail1
msg = MIMEMultipart()
msg['From'] = fromaddr #storing senders mail address
msg['To'] = toaddr  #storing receivers mail address
msg['Subject'] = "This is your report" #store the subject
body = "Kindly check the attachment"
msg.attach(MIMEText(body, 'plain'))#attach the body with
message instance
filename = filename2+".pdf"
```

```
attachment = open(loc, "rb")#open file to be sent in 
rb(binary read)
p = MIMEBase('application', 'octet-stream')#mimebase 
instance octet-stream->is a binary file...it will be a 
document that must be opened in an application
p.set_payload((attachment).read())#encoding
encoders.encode_base64(p)#encode
p.add_header('Content-Disposition', "attachment; 
filename= %s" % filename)
msg.attach(p)
s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login(fromaddr, "cgpjcmr1998@")
text = msg.as_string()#convert multipart msg into string
s.sendmail(fromaddr, toaddr, text)
print("Msg sent successful")
s.quit()
```

The report generated by doctor has to be sent to respective patient ie via their email id . The entire process of sending the mail is done using smtp server ,the from and to mail addresses are taken from the form where the doctor has entered patient mail id (by default the from id is that the developer has created for the proposed system).

The details including subject,body are encrypted using encode_base64() and is prepared to send vis smtp port numbered as 587.The entire encrypted multipart message is converted to string and sent to destined receiver and the acknowledge for the same is sent to the doctor.

5.7.3 Database (model.py)

```
from django.db import models

class PatientReg(models.Model):
    pname = models.CharField(max_length=100)
    pemail = models.CharField(max_length=100)
    pphone = models.CharField(max_length=100)
    paddress = models.TextField()
    password = models.CharField(max_length=100)
```

```
class DoctorReg(models.Model):
    pname = models.CharField(max_length=100)
    pemail = models.CharField(max_length=100)
    pphone = models.CharField(max_length=100)
    paddress = models.TextField()
    password = models.CharField(max_length=100)

class ReportPredict(models.Model):
    Occupational_hazard = models.IntegerField()
    Genetic_Risk = models.IntegerField()
    chronic_lung_cancer = models.IntegerField()
    smocking = models.IntegerField()
    passive_smoker = models.IntegerField()
    chest_pain = models.IntegerField()
    coughing_of_blood = models.IntegerField()
    fatigue = models.IntegerField()
    weight_loss = models.IntegerField()
    dry_cough = models.IntegerField()
    clubbing_of_finger_nail = models.IntegerField()
```

Each model maps to single database table. Each field takes set of field specific arguments .The column type which tells the database what kind of data to store(eg : INTEGER ,VARCHAR) . Above code is the table creation part ,where each disease algo variables are declared to be stored as the column name in database.

CHAPTER 6

RESULTS AND DISCUSSION

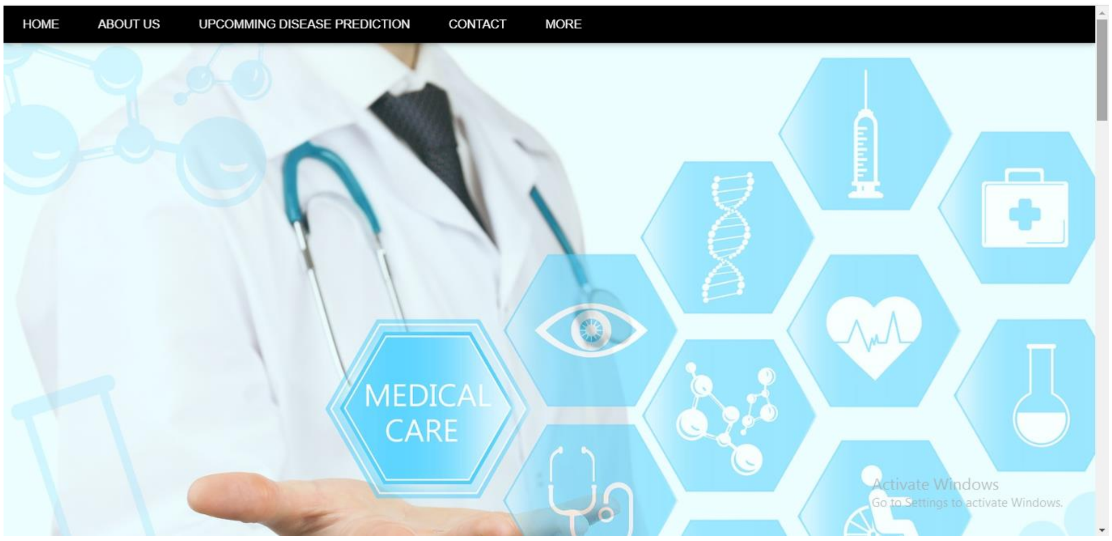

HOME
ABOUT US
UPCOMMING DISEASE PREDICTION
CONTACT
MORE
MEDICAL CARE
Activate Windows
Go to Settings to activate Windows.

Fig 6.1 Home page

Home page contains the details on developers, keys for users to login with the system and share their feedback to the developers

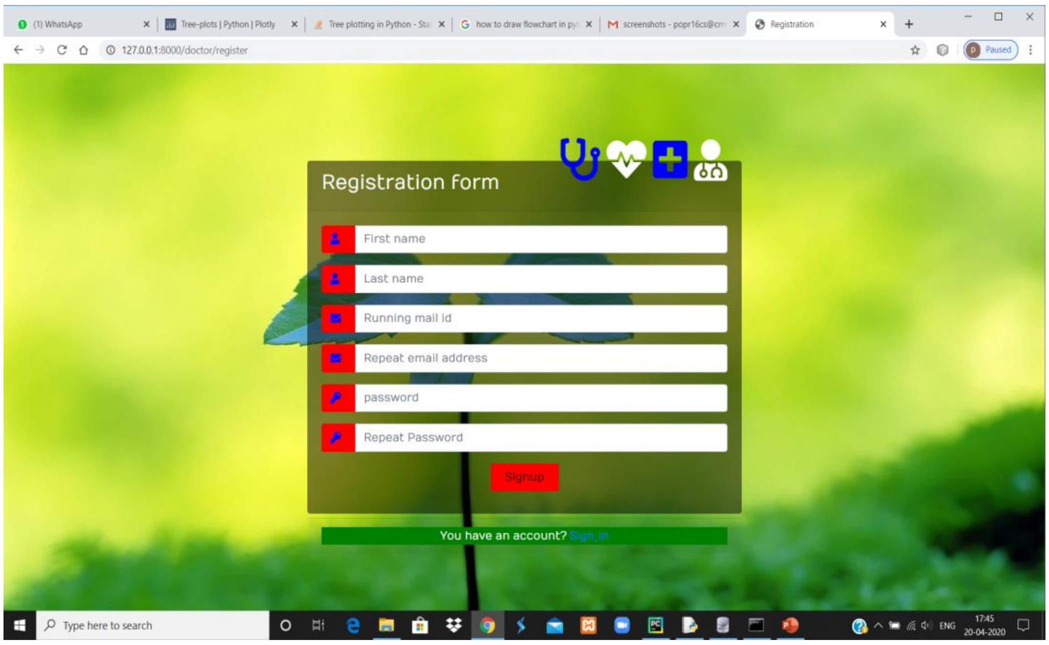
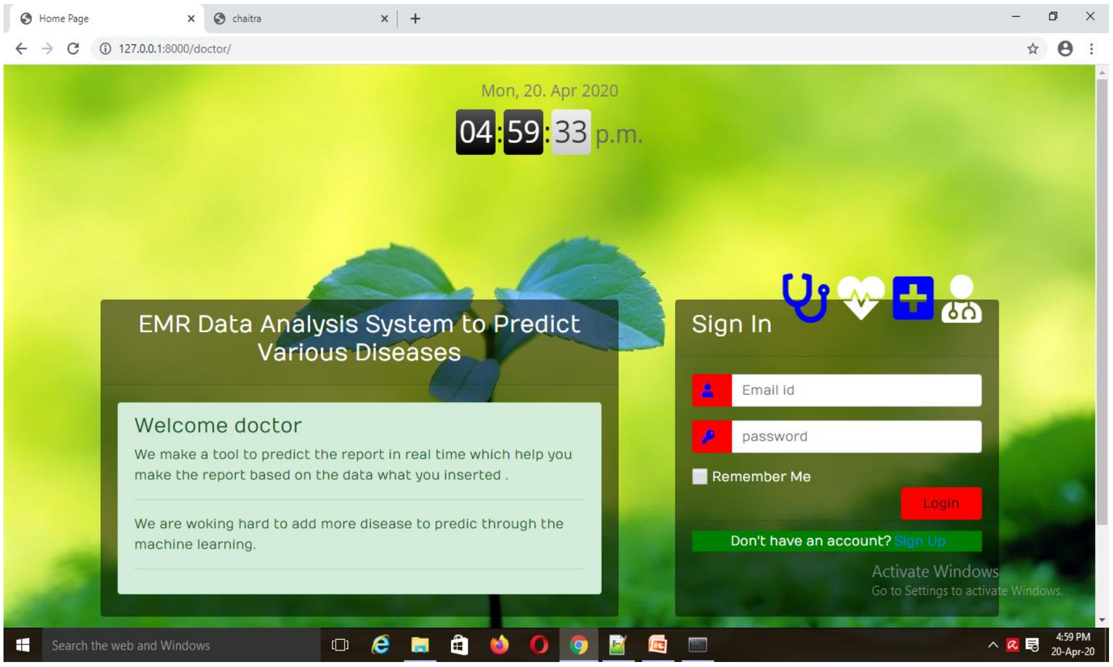


Fig 6.2 Registration page of doctor


Home Page
chaitra
127.0.0.1:8000/doctor/
Mon, 20. Apr 2020
04:59:33 p.m.
EMR Data Analysis System to Predict Various Diseases
Sign In
Email id
password
Remember Me
Login
Don't have an account? Sign Up
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
4:59 PM
20-Apr-20

Welcome doctor

We make a tool to predict the report in real time which help you make the report based on the data what you inserted .

We are woking hard to add more disease to predic through the machine learning.

Fig 6.3 Login page of doctor

Doctor panel being active and important module, the doctors as user can register with the system using valid credentials and login to the system using the same.

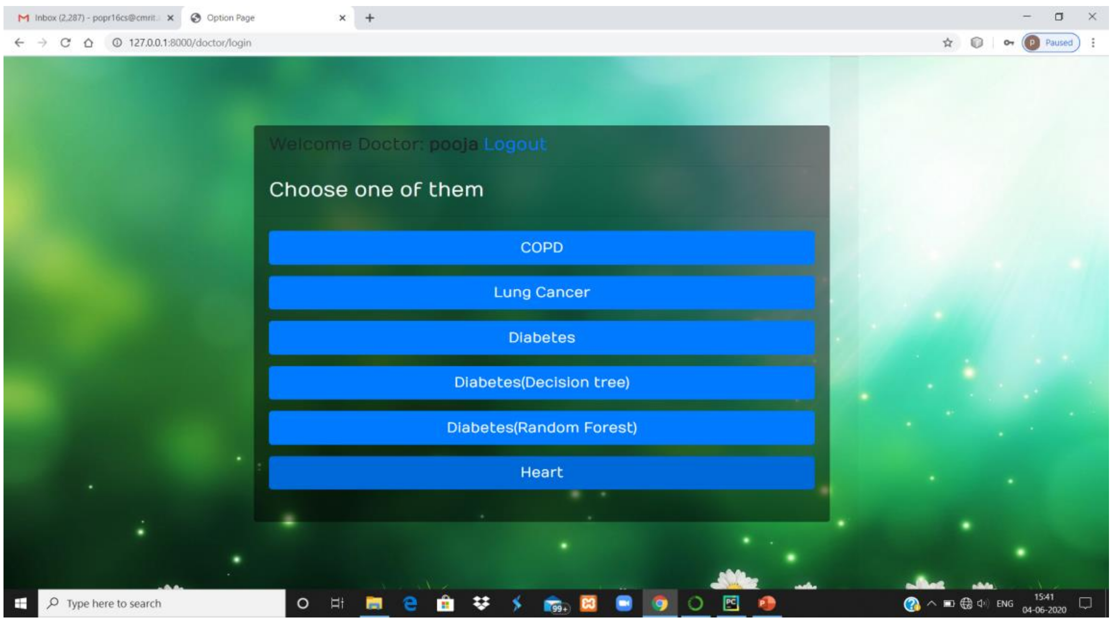
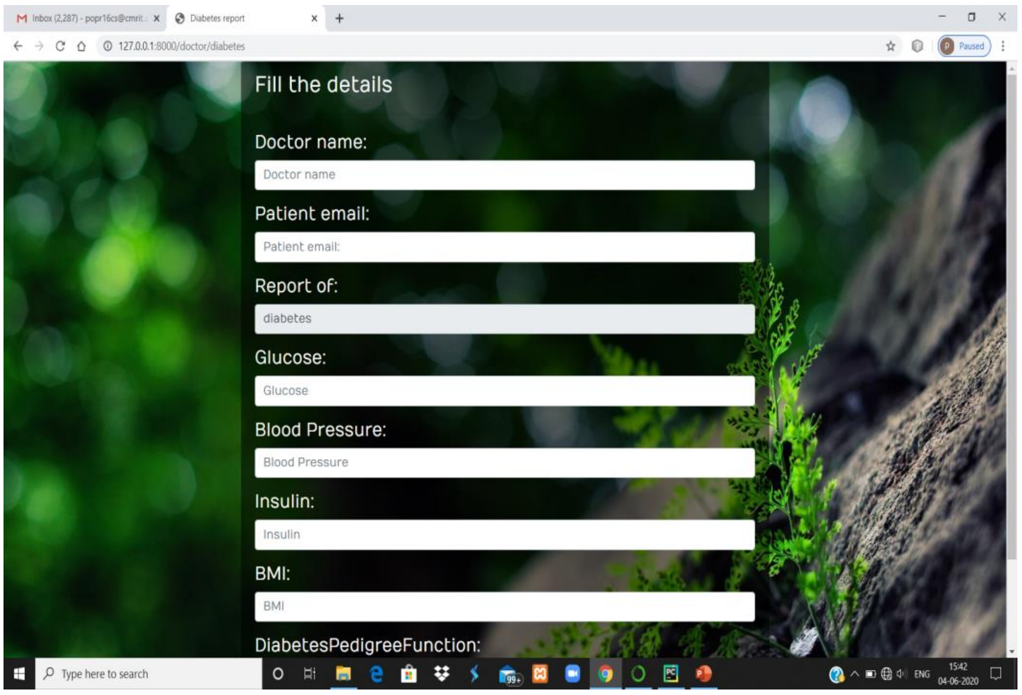


Fig 6.4 Choosing a particular disease for prediction

Fig 6.4 shows the choices of a diseases where doctor choose a particular disease risk to be predicted based on the lab report of the patient.


Fill the details

Doctor name: Doctor name

Patient email: Patient email:

Report of: diabetes

Glucose: Glucose

Blood Pressure: Blood Pressure

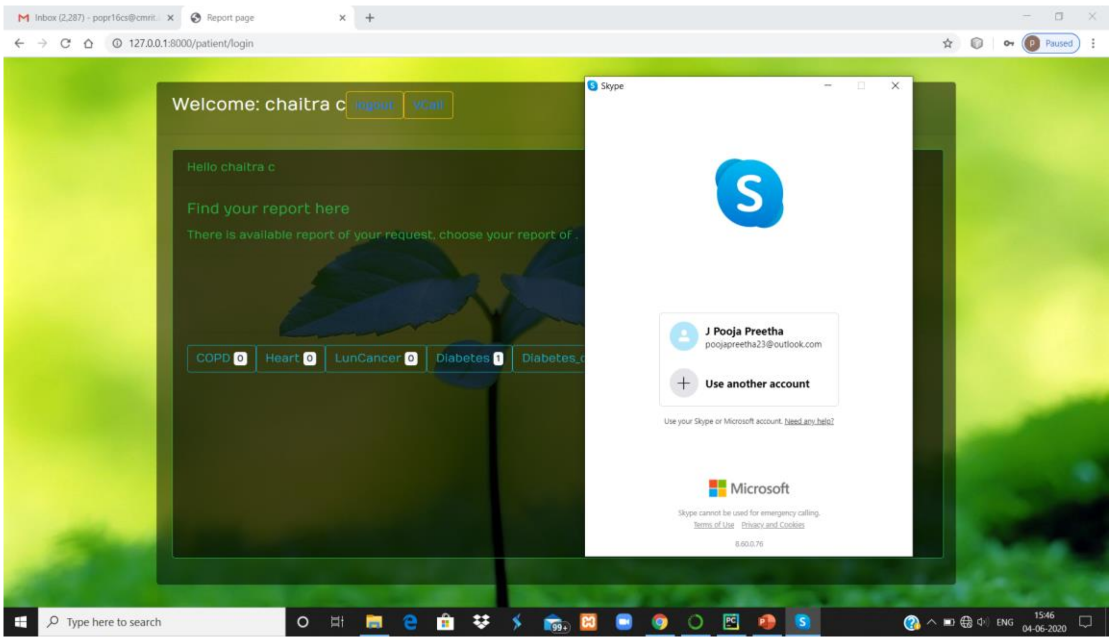
Insulin: Insulin

BMI: BMI

DiabetesPedigreeFunction:

Fig 6.5 Report values to be entered

Fig 6.5 here doctor can enter the details based on the report of the patient to predict the risk of the disease.


Welcome: chaitra c

Hello chaitra c

Find your report here

There is available report of your request, choose your report of .

COPD 0 Heart 0 LunCancer 0 Diabetes 1 Diabetes

Fig 6.6 Patient page

Fig 6.6here patient can able to download the report sent by the doctor after the prediction of a risk of the disease and also he/she can a call doctor if required.

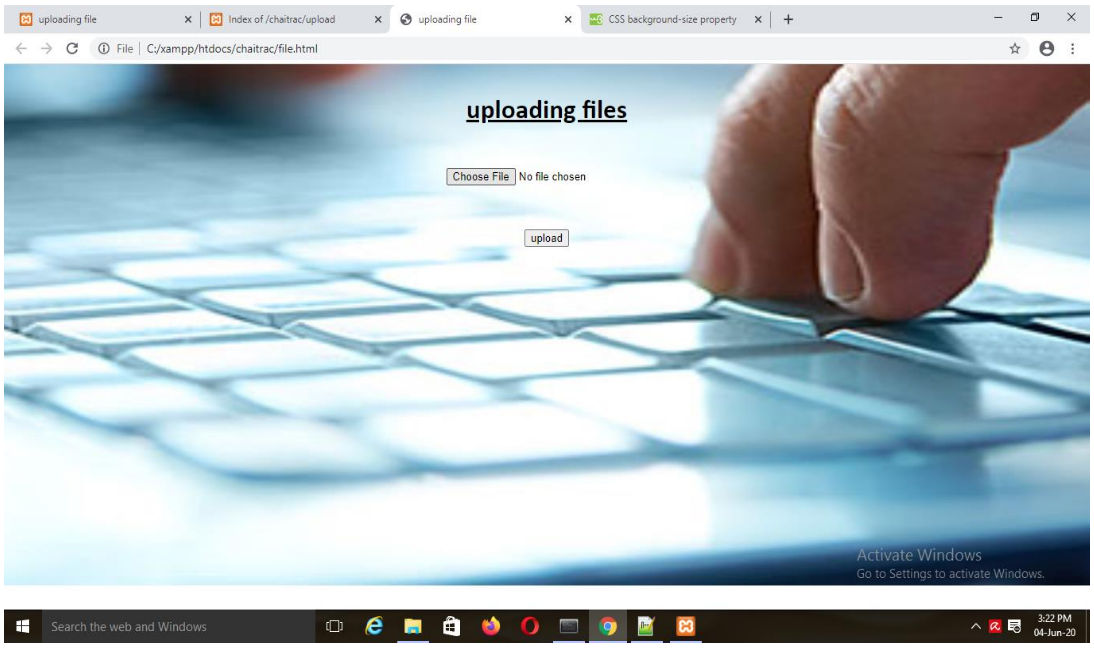
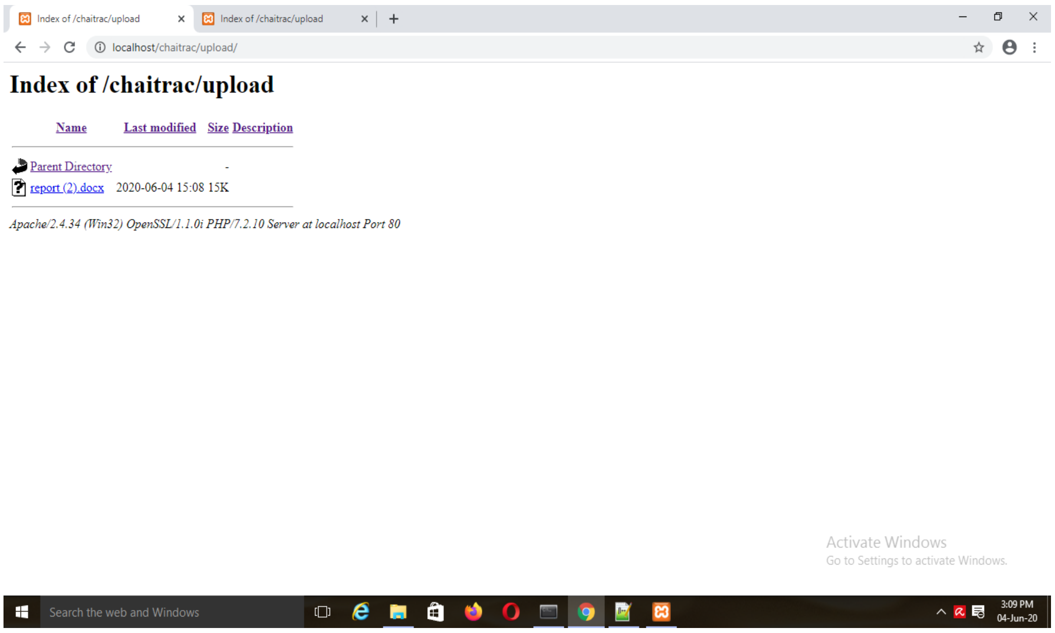

uploading file
Index of /chaitrac/upload
uploading file
CSS background-size property
File | C:/xampp/htdocs/chaitrac/file.html
uploading files
Choose File
No file chosen
upload
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
3:22 PM
04-Jun-20

Fig 6.7 Uploading lab report

Fig 6.7 here patient can upload the lab report, once patient uploads the report it can be seen by the doctor.


Index of /chaitrac/upload
Index of /chaitrac/upload
localhost/chaitrac/upload/
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
3:09 PM
04-Jun-20

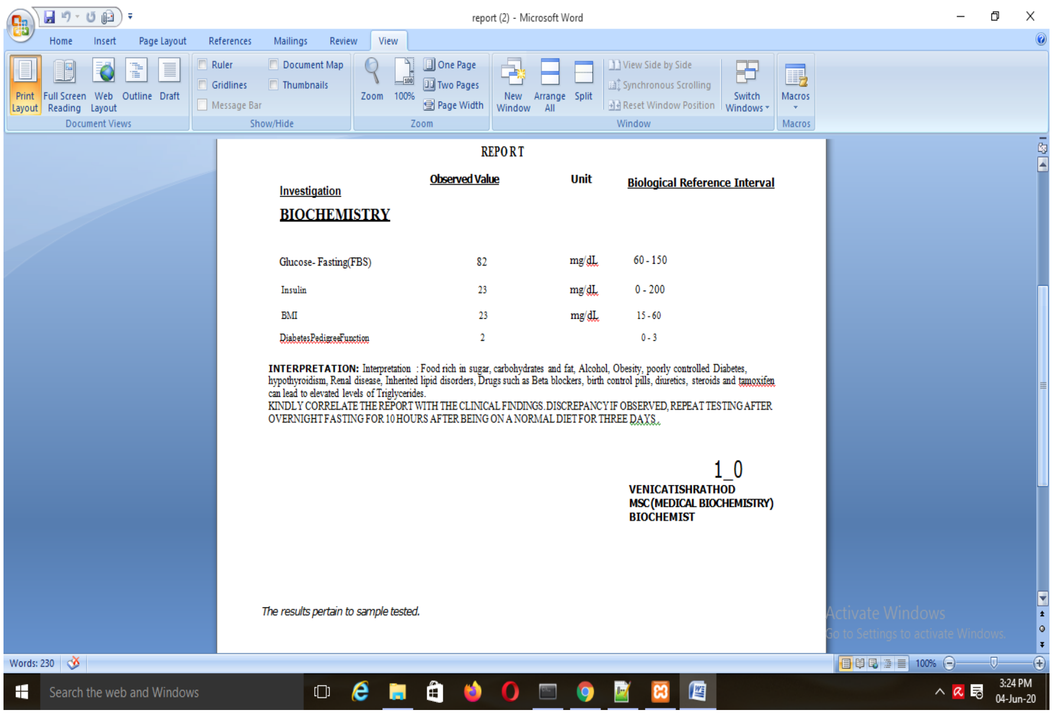
Index of /chaitrac/upload

Name	Last modified	Size	Description
Parent Directory		-	
report (2).docx	2020-06-04 15:08	15K	

Apache/2.4.34 (Win32) OpenSSL/1.1.0i PHP/7.2.10 Server at localhost Port 80

Fig 6.8 Lab Reports of the patients

Fig 6.8 here doctor can able to see the lab reports of all the patient.



REPORT

Investigation	Observed Value	Unit	Biological Reference Interval
BIOCHEMISTRY			
Glucose- Fasting(FBS)	82	mg/dL	60 - 150
Insulin	23	mg/dL	0 - 200
BMI	23	mg/dL	15 - 60
DiabetesPedigreeFunction	2		0 - 3

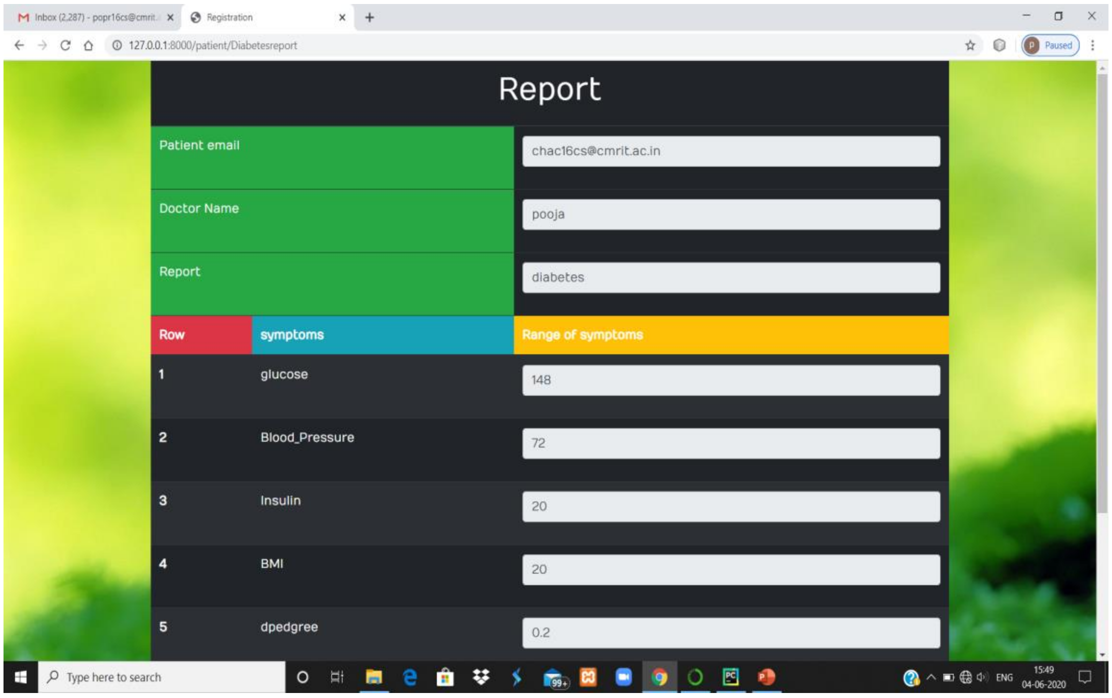
INTERPRETATION: Interpretation : Food rich in sugar, carbohydrates and fat, Alcohol, Obesity, poorly controlled Diabetes, hypothyroidism, Renal disease, Inherited lipid disorders, Drugs such as Beta blockers, birth control pills, diuretics, steroids and tamoxifen can lead to elevated levels of Triglycerides.
KINDLY CORRELATE THE REPORT WITH THE CLINICAL FINDINGS. DISCREPANCY IF OBSERVED, REPEAT TESTING AFTER OVERNIGHT FASTING FOR 10 HOURS AFTER BEING ON A NORMAL DIET FOR THREE DAYS.

1_0
VENICATISHRATHOD
MSC (MEDICAL BIOCHEMISTRY)
BIOCHEMIST

The results pertain to sample tested.

Fig 6.9 Lab report of the patient

Fig 6.9 shows the lab report of the patient which is seen by the doctor.



127.0.0.1:8000/patient/Diabetesreport

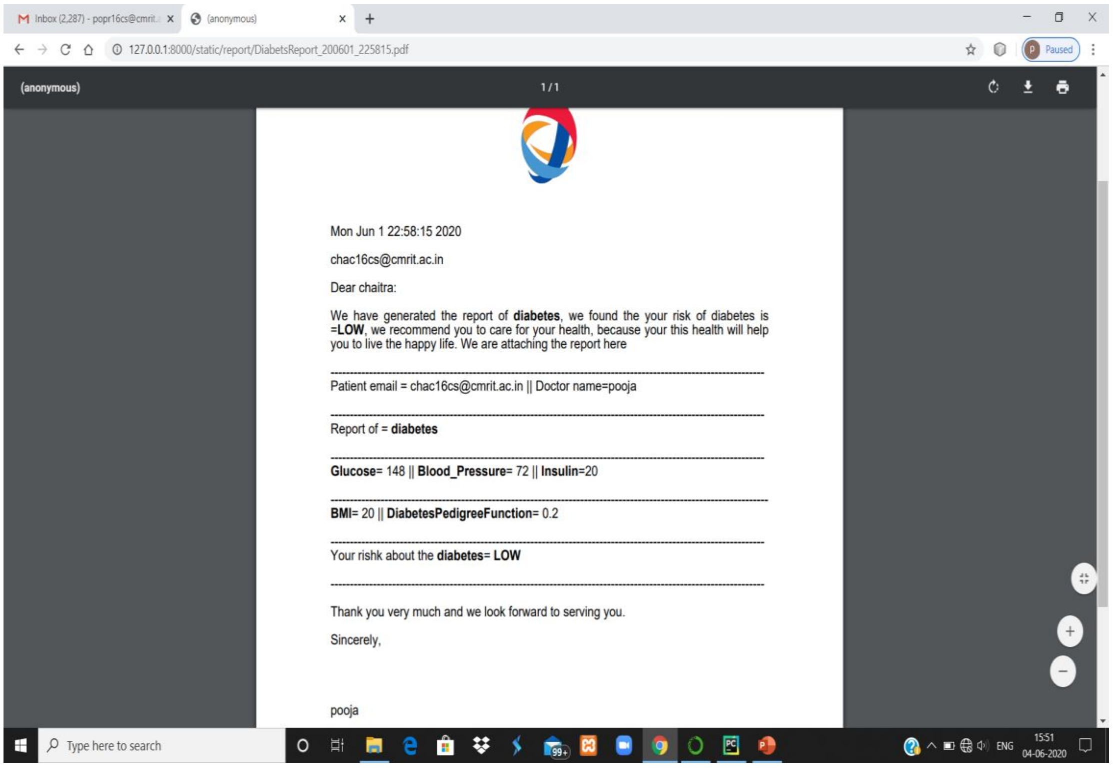
Report

Patient email	chac16cs@cmrit.ac.in
Doctor Name	pooja
Report	diabetes

Row	symptoms	Range of symptoms
1	glucose	148
2	Blood_Pressure	72
3	Insulin	20
4	BMI	20
5	dpedgree	0.2

Fig 6.10 Result of the report

Fig 6.10 shows the result of a lab report generated, when the report details submitted by the doctor.



Mon Jun 1 22:58:15 2020

chac16cs@cmrit.ac.in

Dear chaitra:

We have generated the report of **diabetes**, we found the your risk of diabetes is =**LOW**, we recommend you to care for your health, because your this health will help you to live the happy life. We are attaching the report here

Patient email = chac16cs@cmrit.ac.in || Doctor name=pooja

Report of = **diabetes**

Glucose= 148 || **Blood_Pressure**= 72 || **Insulin**=20

BMI= 20 || **DiabetesPedigreeFunction**= 0.2

Your rishk about the **diabetes= LOW**

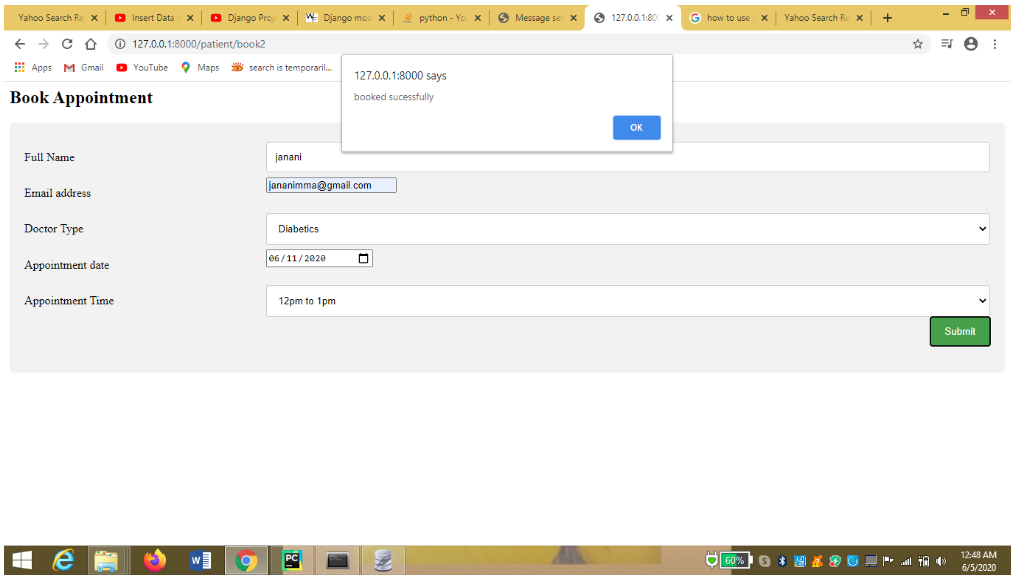
Thank you very much and we look forward to serving you.

Sincerely,

pooja

Fig 6.11 Result of the patient disease

Fig 6.11 shows the result of the patient report i.e. the risk of the disease is severe, moderate or low.



127.0.0.1:8000 says

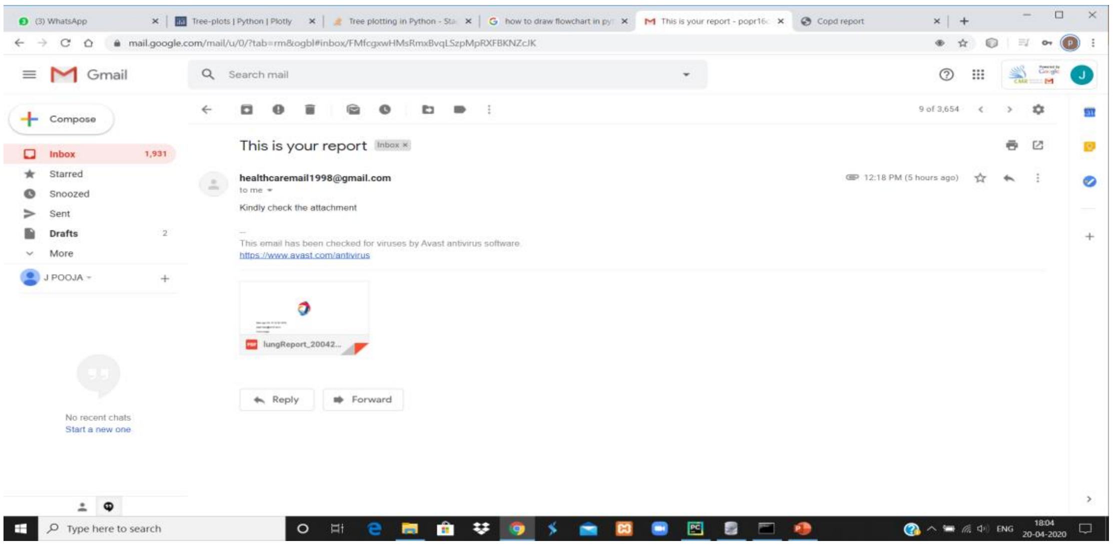
booked sucessfully

Book Appointment

Full Name	janani
Email address	jananimma@gmail.com
Doctor Type	Diabetics
Appointment date	06/11/2020
Appointment Time	12pm to 1pm

Fig 6.12 Appointment booking

Fig 6.12 here the patient can able to book the appointment with doctor for diagnosis.



This is your report Inbox

healthcaremail1998@gmail.com
to me

12:18 PM (5 hours ago)

Kindly check the attachment

--
This email has been checked for viruses by Avast antivirus software.
https://www.avast.com/antivirus

lungReport_20042...

Fig 6.13 Mail to a patient

Fig 6.13 shows the mail of a patient received from a doctor after the prediction of the risk of a disease.

CHAPTER 7

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

There are various types of test. Each test type addresses a specific testing requirement.

For the proposed system, we have executed multiple algorithms on these four diseases to choose those algorithm that is suitable for comparing. Further, the comparison is done between these algorithms to get the one that has high accuracy.

Each modules are tested separately on all cases (high, low) and checked for errors that users may face during the practical usage of this system.

Unit testing

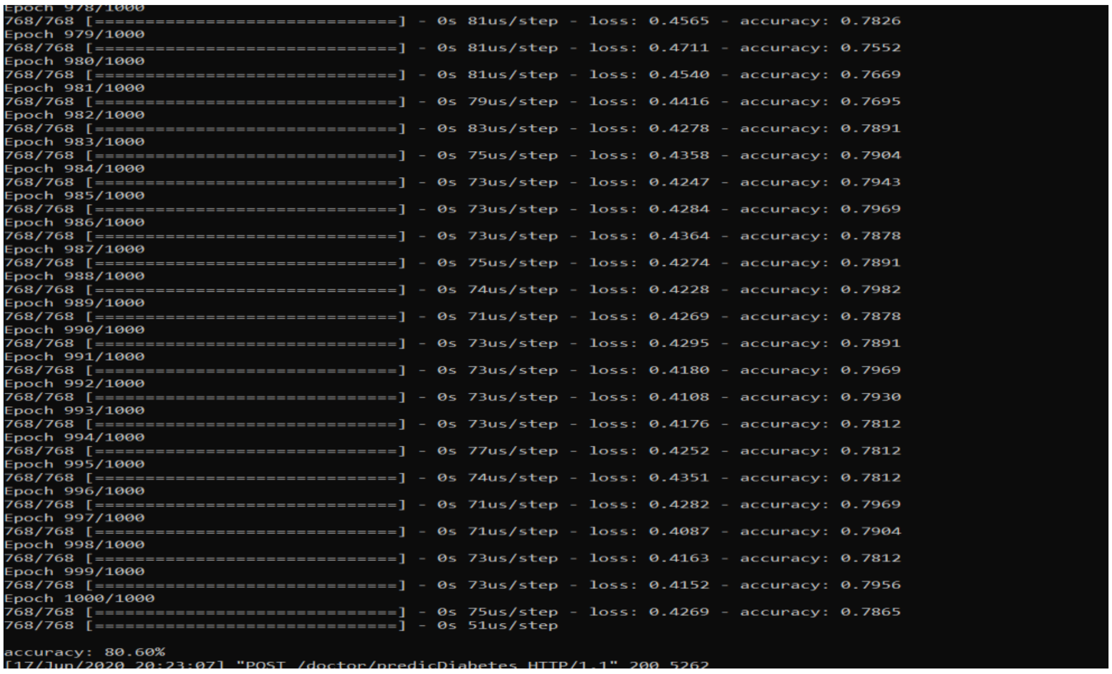
Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.



```
Epoch 978/1000
768/768 [==============================] - 0s 81us/step - loss: 0.4565 - accuracy: 0.7826
Epoch 979/1000
768/768 [==============================] - 0s 81us/step - loss: 0.4711 - accuracy: 0.7552
Epoch 980/1000
768/768 [==============================] - 0s 81us/step - loss: 0.4540 - accuracy: 0.7669
Epoch 981/1000
768/768 [==============================] - 0s 79us/step - loss: 0.4416 - accuracy: 0.7695
Epoch 982/1000
768/768 [==============================] - 0s 83us/step - loss: 0.4278 - accuracy: 0.7891
Epoch 983/1000
768/768 [==============================] - 0s 75us/step - loss: 0.4358 - accuracy: 0.7904
Epoch 984/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4247 - accuracy: 0.7943
Epoch 985/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4284 - accuracy: 0.7969
Epoch 986/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4364 - accuracy: 0.7878
Epoch 987/1000
768/768 [==============================] - 0s 75us/step - loss: 0.4274 - accuracy: 0.7891
Epoch 988/1000
768/768 [==============================] - 0s 74us/step - loss: 0.4228 - accuracy: 0.7982
Epoch 989/1000
768/768 [==============================] - 0s 71us/step - loss: 0.4269 - accuracy: 0.7878
Epoch 990/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4295 - accuracy: 0.7891
Epoch 991/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4180 - accuracy: 0.7969
Epoch 992/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4108 - accuracy: 0.7930
Epoch 993/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4176 - accuracy: 0.7812
Epoch 994/1000
768/768 [==============================] - 0s 77us/step - loss: 0.4252 - accuracy: 0.7812
Epoch 995/1000
768/768 [==============================] - 0s 74us/step - loss: 0.4351 - accuracy: 0.7812
Epoch 996/1000
768/768 [==============================] - 0s 71us/step - loss: 0.4282 - accuracy: 0.7969
Epoch 997/1000
768/768 [==============================] - 0s 71us/step - loss: 0.4087 - accuracy: 0.7904
Epoch 998/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4163 - accuracy: 0.7812
Epoch 999/1000
768/768 [==============================] - 0s 73us/step - loss: 0.4152 - accuracy: 0.7956
Epoch 1000/1000
768/768 [==============================] - 0s 75us/step - loss: 0.4269 - accuracy: 0.7865
768/768 [==============================] - 0s 51us/step

accuracy: 80.60%
[17/Jun/2020 20:23:07] "POST /doctor/predicDiabetes HTTP/1.1" 200 5262
```

Running tests on CNN algorithm to get maximum accuracy.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

Prediction of multiple diseases using CNN machine learning Algorithm is successfully implemented with 83% accuracy.

There are many possible improvements that could be explored to improve the scalability and accuracy of this Disease Risk prediction system.

As we have developed a system for few diseases to predict the Risk, in future we can use this system for implementing other diseases like corona, brain tumour etc.

The performance of health's diagnosis can be improved significantly by handling numerous class labels in the prediction process and it can be another positive direction of research.

8.2 Future Work

There are many possible improvements that could be exposed to improve the scalability and accuracy of this Disease Risk prediction system. We know our project have some limitations and complex. In future our project will be more suitable and more reliable.

As we have developed a system for few diseases to predict the Risk, in future we can use this system for implementing other diseases like Corona, Brain tumour etc.

The performance of health’s diagnosis can be improved significantly by handling numerous class labels in the prediction process and it can be another positive direction of research.

REFERENCES

[1] Disease Prediction by Machine Learning Over Big Data From Healthcare Communities , Min Chen , Yixue Hao , Kai Hwang, Lu Wang and Lin Wang"DNA June , 2018.

[2] "Analysis and classification of heart diseases using heartbeat features and machine learning algorithms."Pattekari, S.A. and Parveen, A., 2012. Prediction system for heart disease using Naïve Bayes. International Journal of Advanced Computer and Mathematical Sciences, 3(3), pp. 290–294.

[3] Journal (May 2018) Bone Tumor Detection from MRI Images Using Machine Learning Sonal S. Ambalkar, S. S. Thorat

[4] "Prediction of Diabetes Supported Personal Lifestyle Indicators." AYUSH ANAND , DIVYA SHAKTI,2015 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015 .

[5] Designing Disease Prediction Model Using Machine Learning Approach 1Dhiraj Dahiwade, 2 Prof. Gajanan Patle, 2 Prof. Ektaa Meshram. Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019).

[6] "Big Health Application System based on Health Internet of Things and Big Data Data YUJUN MA1 , (Member, IEEE), YULEI WANG1 , JUN YANG2 , YIMING MIAO2 , AND WEI LI2 date of publication December 13, 2016

[7] "Disease Prediction by using Machine Learning". Sayali Ambekar and Dr.Rashmi Phalnikar." Department of Information Technology MIT-COE, Pune