Symptom Based Disease Prediction Using Machine Learning Algorithms

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Information Technology

by

JAINI LAKSHMI SABARI PRIYA

16BIT0189

Under the guidance of

Prof. Dr. P.J.Kumar

School of Information technology,

VIT, Vellore.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May, 2020

<u>DECLARATION</u>

I hereby declare that the thesis entitled “Symptom based disease prediction using machine learning algorithms” submitted by me, for the award of the degree of *Bachelor of Technology in Information Technology* to VIT is a record of bonafide work carried out by me under the supervision of Prof. Dr. P.J.Kumar.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date : 14 May 2020

J.Lakshmi Sabari Priya

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled “Symptom based disease prediction using machine learning algorithms” submitted by Jaini Lakshmi Sabari Priya 16BIT0189, School of Information Technology, VIT, Vellore, for the award of the degree of *Bachelor of Technology in Information Technology*, is a record of bonafide work carried out by him under my supervision, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 14 May 2020

Kumar

Dr. P. J. KUMAR

Signature of the Guide

Internal Examiner **External Examiner**

Head of the Department

Information Technology

ACKNOWLEDGEMENTS

It is my pleasure to express with deep sense of gratitude to Prof. Dr. P.J.Kumar, Associate Professor, School of Information Technology and Engineering, Vellore Institute of Technology, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Computer Science.

I would like to express my gratitude to Dr. G.Viswanathan, Dr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. Anand A. Samuel, Dr. S.Narayanan, and Dr.Balakrushna Tripathy, School of Information Technology and Engineering, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Jasmine Norman, Head of the Department, SITE, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Jaini Lakshmi Sabari Priya

EXECUTIVE SUMMARY

Machine learning has a major impact on healthcare analytics and it have the capacity to reduce costs of treatment, predict outbreaks of epidemics, avoid preventable diseases and improve the quality of life. Accurate analysis of medical data benefits in early disease detection and well patient care in big data. The analysis accuracy is reduced when we have incomplete data. In this project, machine learning algorithms are used for effective prediction of diseases . Now-a-days, individuals face different infections due to the natural condition and their living propensities. So the prediction of disease at prior stage becomes significant assignment. Be that as it may, the exact forecast based on side effects turns out to be unreasonably hard for specialist. The right prediction of disease is the most testing assignment. To defeat this issue information mining plays a significant job to anticipate the infection. The examination exactness is diminished when we have fragmented information. In this project, Data visualization and predictive analysis of diseases using structured and unstructured data has been done by the machine learning algorithms like Principal Component Analysis, Support Vector Machine, artificial neural network, random forest, decision tree and Multinomial Naïve Bayes for predicting diseases from symptoms. For unstructured data, decision tree performs the best of all the three algorithms used with an accuracy of 89.93%. As the structured data is linear data, all the models perform the same. A website is developed using a python framework called flask based on this concept for patients to predict the diseases before reaching the medical facility using an accurate machine learning model and search for doctors also according to their required specialization. Doctors can register in the website and gain exposure to their expertise and their facilities.

TABLE OF CONTENTS

List Of Figures

List Of Tables

1. INTRODUCTION

1.1. OBJECTIVE

Early detection is the key of success in biomedical and healthcare communities. The accurate analysis of medical data will predict accurate results. The existing work covers structured data but unstructured data is not taken which lead to inaccurate results. We also have to consider the following scenarios: How to deal with the real life data, to deal with the missing data, to get the accurate prediction. The main objectives of the project are:

- Data visualisation and predictive analysis of diseases using structured data can be done by the machine learning algorithms like Principal Component Analysis, Support Vector Machine, artificial neural network and random forest.
- Data cleaning ,data visualization and predictive analysis of diseases using unstructured data can be done by the machine learning algorithms like Decision tree, Multinomial naive bayes classifier and Support Vector Machine
- Website for predicting diseases based on the symptoms using the most accurate model.

1.2. MOTIVATION

As the use of internet is growing every day, people are always curious to automate everything. People always prefer smart and easy solutions if any problem arises. People have more access to internet than hospitals and doctors these days. It is better to have an early detection of any disease than to wait till you reach a medical facility so that people may get an idea of what it is and can be more cautious. So, this system can be helpful to the people as most of the people have internet access all the time. My project is a web based application that predicts the disease of the user with respect to the symptoms given by the user. With the help of this web application, the user will be able to know the disease with the given symptoms and also search for doctors in their required field.

1.3. LITERATURE SURVEY:

1.Disease prediction by machine learning over big data from healthcare communities

AUTHOR: M. Chen, Y. Hao, K. Hwang, L. Wang

DESCRIPTION:

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. 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 collected from central China in 2013-2015. To overcome the difficulty of incomplete data, we use a latent factor model to reconstruct the missing data. We experiment on a regional chronic disease of cerebral infarction. We propose a new convolutional neural network (CNN)-based multimodal disease risk prediction algorithm using structured and unstructured data from hospital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed, which is faster than that of the CNN-based unimodal disease risk prediction algorithm.

2. Predictive analytics in health care using machine learning tools and techniques

AUTHOR: V Ilango B Nithya

DESCRIPTION:

When we have a huge data set on which we would like to perform predictive analysis or pattern recognition, machine learning is the way to go. Machine Learning (ML) is the fastest rising arena in computer science, and health informatics is of extreme challenge. The aim of Machine Learning is to develop algorithms which can learn and progress over time and can be used for predictions. Machine Learning practices are widely used in various fields and primarily health care industry has been benefitted a lot through machine learning prediction techniques. It offers a variety of alerting and risk management decision support tools, targeted at improving patients' safety and healthcare quality. With the need to reduce healthcare costs and the movement towards personalized healthcare, the healthcare industry faces challenges in the essential areas like, electronic record management, data integration, and computer …

3. Disease classification using machine learning algorithms-a comparative study

AUTHOR: S.Leoni Sharmila, C.Dharuman and P.Venkatesan

DESCRIPTION:

Machine learning technique is widely used in various fields of science and technology. They have been giving out meaningful and classified information this tool also explores in constructing and study of algorithms which can learn from data. Data mining in healthcare is an emerging field of high importance for providing prognosis and a deeper understanding of medical data. Its applications in healthcare include analysis and prevention of hospital errors, early detection , prevention of diseases, and for cost savings. The main problem arises to predict and diagnosis the disease in early stage, with the use of machine learning techniques. This paper gives a comparative study of different machine learning technique such Fuzzy logic, Fuzzy Neural Network and decision tree in classifying liver data set.

4. Disease prediction using machine learning over big data

AUTHOR: Shraddha Subhash Shirsath, Prof. Shubhangi Patil

DESCRIPTION:

Now a days big data is the fastest and more widely used in every field .With the help of big data medical and health care sectors achieves their growth and with help of big data benefit of a accurate medical data analysis , early disease prediction, accurate data of an patient can be securely stored and used .Moreover the accuracy of an analysis can be reduced due to an various reason like incomplete medical data, some regional disease characteristics which can be outbreaks the prediction. In this paper we can use a machine learning algorithm for the accurate disease prediction for that purpose we can collect the hospital data of a particular region. For missing data we can use latent factor model to achieve the incomplete data. In the previ-ous work for disease prediction Convolutional Neural Network Based Unimodel Disease Prediction (CNN-UDRP) Algorithm is used. Convolutional Neural Network Based Multimodal Disease Prediction(CNN-MDRP) algorithm is overcome the drawbacks of CNN-UDRP algorithm only focus work on a structured data but CNN-MDRP algorithm uses both structured and unstructured data from the hospital. None of the existing work focused on both data types in the area of medical big data analysis .CNN-MDRP algorithm prediction is more accurate than compared to the previous prediction algorithm.

5. Designing a disease prediction model using machine learning

AUTHOR: Ms.Jyoti Chandrashekhar Bambal1 , Prof. Roshani B. Talmale2

DESCRIPTION:

Now a day, people face various diseases due to the environmental condition and living habits of them. So prediction of disease at earlier stage becomes important task. But the prediction on the basis of symptoms becomes too difficult for doctor. The correctly prediction of disease is most challenging task. To overcome this problem data mining plays an important and efficient way to predict the disease. Medical science has huge amount of data growth per year. Due to increase amount of data growth in medical and healthcare field the accurate analysis on medical data which has been benefits from early patient care. With the help of disease data, data mining finds hidden pattern information in the large amount of medical data. We have designed the heart disease prediction system. We proposed multiple disease prediction based on symptoms of the patient. For the heart disease prediction, we used knn , naïve bayes machine learning algorithm for accurate prediction of disease. For disease prediction required disease symptoms dataset. Here we focused on heart disease prediction, because the heart disease is one of the leading causes of death among all other diseases. The heart disease prediction contains that whether the patient suffer from heart disease or not by using naïve bayes and KNN algorithm. In this heart disease prediction, the living habits of person and checkup information consider for the accurate prediction. The accuracy of heart disease prediction by using naïve bayes is 94.5% which is more than KNN algorithm. And the time and the memory requirement is also more in KNN than naïve bayes. After heart disease prediction, this system able to gives the risk associated with heart disease which is lower risk of heart disease or higher. For the risk prediction, we are using CNN algorithm.

6. A machine learning approach for prediction of diseases using unstructured datasets

AUTHOR: Mayedaarshi, Dr.Rekhapatil

DESCRIPTION:

The growth of big data in biomedical and healthcare communities has been rapidly increasing, early disease detection accurate analysis of medical data benefits, community services and patient care. Moreover, the analysis accuracy is reduced when the quality of medical data is incomplete. However, different regions exhibit unique characteristics of certain regional diseases that may weaken the prediction of disease outbreaks. Here, by streamlining the machine learning algorithms for effective prediction of chronic disease outbreak in disease-frequent communities. The modified prediction models over real life hospital data is collected and experimented. The difficulty of incomplete data, a latent factor model is used to reconstruct the missing data. A regional chronic disease is experimented. By proposing a new convolutional neural network based multimodal disease risk prediction (CNN-MDRP) algorithm using structured and unstructured data from hospital

7. Predictive analysis of diseases using machine learning and big data

AUTHOR: Sireesha Kilaru, D. Anupama

DESCRIPTION:

Today we must accept the truth that machine learning and big data are boon to healthcare industry, and many research works are already started to reduce the complications in biomedical and healthcare fields. In order to get the maximum benefit from these two sectors we need quality data as input. This can be obtained by the healthcare historical databases. Incomplete data may not predict the result accurately; to avoid this problem our proposed system will use machine learning algorithms to predict the accurate disease in the onset of chronic disease outbreaks in disease-frequent communities. For this we have collected a regional hospital data. We can use latent factor model to achieve incomplete data. The proposed system will use neuronal network based multimodal disease risk prediction (CNN-MDRP) algorithm that uses structured and unstructured data from the hospital. To the best of my knowledge, none of the existing work focused on the both types of data in the area of healthcare big data analysis .Our proposed algorithm will have a prediction accuracy of 94.8% with a convergence rate faster than CNN-based unimodal disease risk prediction algorithm (CNN-UDRP).

8. Disease prediction by using machine learning

AUTHOR: Sayali Ambekar and Dr.Rashmi Phalnikar

DESCRIPTION:

The rapid growth in the field of data analysis plays an important role in the healthcare research. Due to large amount of data growth in biomedical and healthcare field providing accurate analysis of medical data that has benefits from early detection, patient care, and community services. Previous system designed to analyze, manage and assimilate data produced by healthcare systems. Data analysis has been applied to help the disease-related information and treatment process. In this paper a decision tree is effectively used for predicting the outbreaks of diseases in society. The paper proposes to experiment with the modified predictive models with medical data which is related to the symptoms of the disease. For the disease prediction using unstructured data, we used a convolutional neural network which is based on multimodal disease risk prediction (CNN-MDRP) algorithm.

Users can post their queries in order to seek information regarding diseases so that user get the proper answer to any kind of question and solving any problem related to the disease.

9. Disease prediction by machine learning over big data from healthcare communities

AUTHOR: MIN CHEN1, (Senior Member, IEEE), YIXUE HAO, KAI HWANG, (Life Fellow, IEEE), LU WANG1, AND LIN WANG.

DESCRIPTION:

With big data growth in biomedical and healthcare communities, accurate analysis of medical data benets early disease detection, patient care, and community services. However, 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 collected from central China. To overcome the difficulty of incomplete data, we use a latent factor model to reconstruct the missing data. We experiment on a regional chronic disease of cerebral infarction. We propose a new convolutional neural network (CNN)-based multimodal disease risk prediction algorithm using structured and unstructured data from hospital. To the best of our knowledge, none of the existing work focused on both data types in the area of medical big data analytics. Compared with several typical prediction algorithms, the prediction accuracy of our proposed algorithm reaches 94.8% with a convergence speed, which is faster than that of the CNN-based unimodal disease risk prediction algorithm.

10. Prediction of probability of disease based on symptoms using machine learning algorithm

AUTHORS: Harini D K1, Natesh M2

DESCRIPTION: Big data has a major impact on healthcare analytics and it have the capacity to reduce costs of treatment, predict outbreaks of epidemics, avoid preventable diseases and improve the quality of life. Accurate analysis of medical data benefits in early disease detection and well patient care in big data. The analysis accuracy is reduced when we have incomplete data. In this paper, machine learning algorithms is used for effective prediction of diseases. Latent factor model is used to overcome the difficulty of missing data. A new convolutional neural network based multimodal disease risk prediction (CNN-MDRP) algorithm is proposed in this paper. It uses both structured and unstructured data from hospital for effective prediction of diseases.

2.PROJECT DESCRIPTION AND GOALS

2.1. PROJECT DESCRIPTION

In our current days people are busy with their own lifestyles, most people are owning the unhealthy lifestyle because of their work, unhealthy food and other habits. This may cause of many diseases like heart attacks. For these reasons we need more software‘s and apps to monitor and mobilize the health issues for users. By using machine learning concepts we can apply the prediction diseases based on symptoms. So in this project, a disease prediction system(web application) is proposed using machine learning algorithms and web technologies for users to be able to predict disease before reaching the medical facilities.

For prediction of diseases we use two datasets ,structured dataset and unstructured dataset.

2.2. GOALS

- Predict which disease is the user having based on symptoms using machine learning algorithms for the two datasets and compare the algorithms for the most accurate model for each dataset
- Create a website using the accurate machine learning model with 2 modules : patient login where the user can predict the disease based on symptoms, search for doctors with required specialization and doctor login where he can register as doctor with his specialization

3.TECHNICAL SPECIFICATION

3.1.PROJECT DEVELOPMENT REUQUIREMENTS

Hardware requirements:

- Processor – dual core
- Hard disk – 50 GB
- Memory – 1GB RAM

Software requirements:

- Windows 7 and above
- Anaconda navigator
- Python 3.5
- HTML
- CSS
- JavaScript
- JQuery
- MongoDB
- Boostrap
- Python Libraries-
    1. Pandas
    2. Pymongo
    3. Numpy
    4. Seaborn
    5. Sklearn
    6. Tensorflow
    7. Keras
    8. Json
    9. Matplotlib
    10. Pickle
    11. Flask

3.2 MODULE DESCRIPTION:

Front-end modules:

- **Patient module:** If Patient is a new user he will enter his personal details and he will user Id and password through which he can login to the system. If patient have already an account then he/she can log into the system. Patients can view only Doctor's little information. Patient will specify the symptoms caused due to his illness. System will ask certain question regarding his illness and system will predict the disease based on the symptoms specified by the patient and system allows patient to search for the related doctors based on the disease. Patient can search for doctor by specifying his specialisation.
- **Doctor module:** Doctor can register and enter his details and specialization so that patients can find them.

Backend modules:

- Data collection
- Data pre-processing
- Data visualization
- Model construction and validation
- Model testing
- Web application building based on most accurate model

3.3. DOMAIN REQUIREMENTS

Machine Learning Models:

1.Supporting Vector Machine (SVM):

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

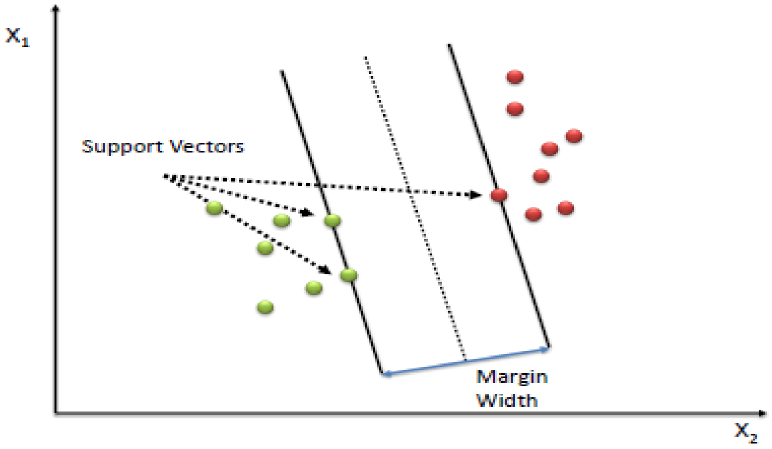


Figure 1 – Support Vector Machine Example Graph

Algorithm

- Define an optimal hyperplane: maximize margin.
- Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications.
- Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space

One-vs-the-rest (OvR) multiclass/multilabel strategy:

Also known as one-vs-all, this strategy consists in fitting one classifier per class. For each classifier, the class is fitted against all the other classes. In addition to its computational efficiency (only n_classes classifiers are needed), one advantage of this approach is its interpretability. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier. This is the most commonly used strategy for multiclass classification and is a fair default choice.

This strategy can also be used for multilabel learning, where a classifier is used to predict multiple labels for instance, by fitting on a 2-d matrix in which cell [i, j] is 1 if sample i has label j and 0 otherwise.

In the multilabel learning literature, OvR is also known as the binary relevance method.

One vs One multiclass/multilabel strategy :

This model considers each binary pair of classes and trains classifier on subset of data containing those classes. So it trains total n*(n-1)/2 classes. During the classification phases each classifier predicts one class. (This is contrast to one vs rest where each classifier predicts probability). And the class which has been predicted most is the answer.

2. Naive Bayes Classifier :

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

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

Using Bayes theorem, we can find the probability of **A** happening, given that **B** has occurred. Here, **B** is the evidence and **A** is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

Suppose in a data set X is input matrix and Y is output matrix. These are specified as

$$X = (x_1, x_2, x_3, ..., x_n)$$

Here x1,x2….xn represent the features. Y is a column matrix which contains two or more classes. So according to bayes theorem,

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Now, you can obtain the values for each by looking at the dataset and substitute them into the equation. For all entries in the dataset, the denominator does not change, it remain static. Therefore, the denominator can be removed and a proportionality can be introduced.

3.Decision Tree:

In medicinal choice like arrangement, diagnosing there are numerous circumstances where choice must be made successfully and dependably. Reasonable straightforward basic leadership models with the likelihood of programmed learning are the most fitting for performing such undertakings. Choice trees are a solid and successful basic leadership procedure that furnish high grouping exactness with a straightforward portrayal of accumulated learning and they have been utilized as a part of various zones of restorative basic leadership.

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

4.Artificial neural network:

A neural network will have

- Input layer, with the bias unit which is 1. It is also referred as the intercept.
- One or more hidden layers, each hidden layer will have a bias unit
- Output layer
- Weights associated with each connection
- Activation function which converts an input signal of a node to an output signal

Input layer, hidden layer and output layers are usually referred as dense layers

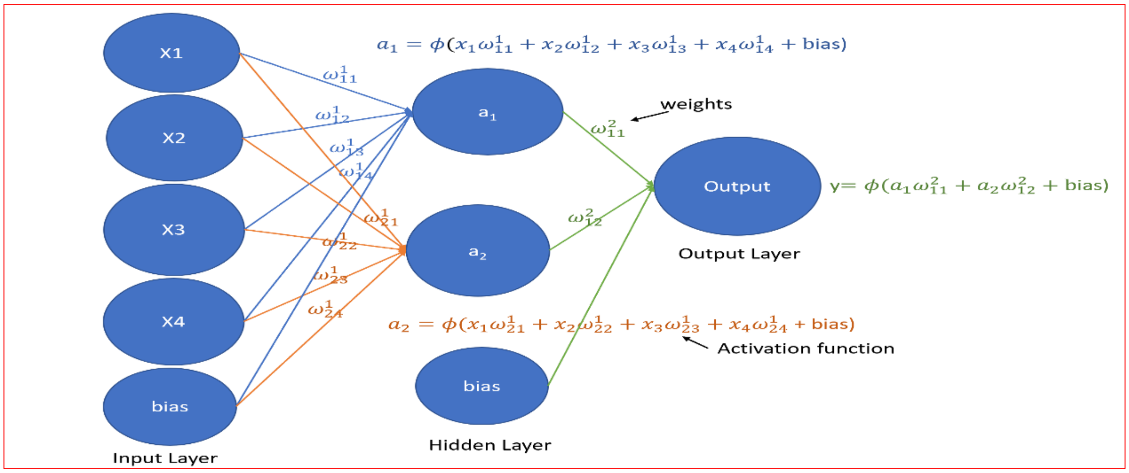

X1
X2
X3
X4
bias
Input Layer
$a_1 = \phi(x_1\omega^1_{11} + x_2\omega^1_{12} + x_3\omega^1_{13} + x_4\omega^1_{14} + \text{bias})$
ω^1_{11}
ω^1_{12}
ω^1_{13}
ω^1_{14}
ω^1_{21}
ω^1_{22}
ω^1_{23}
ω^1_{24}
a_1
a_2
bias
Hidden Layer
weights
ω^2_{11}
ω^2_{12}
Output
Output Layer
$y = \phi(a_1\omega^2_{11} + a_2\omega^2_{12} + \text{bias})$
$a_2 = \phi(x_1\omega^1_{21} + x_2\omega^1_{22} + x_3\omega^1_{23} + x_4\omega^1_{24} + \text{bias})$
Activation function

Figure 2 – Artificial Neural Network Example Graph

A neural network receives input, converts the input signal by changing state using an activation function to produce an output.

5.Random forest:

Random Forest is an extension over bagging. Each classifier in the ensemble is a decision tree classifier and is generated using a random selection of attributes at each node to determine the split. During classification, each tree votes and the most popular class is returned.

Implementation steps of Random Forest –

- Multiple subsets are created from the original data set, selecting observations with replacement.
- A subset of features is selected randomly and whichever feature gives the best split is used to split the node iteratively.
- The tree is grown to the largest.
- Repeat the above steps and prediction is given based on the aggregation of predictions from n number of trees.

3.4. FUNCTIONAL REQUIREMENTS

- Authentication is a must to login into the web page. Unauthenticated user will not be able to login.
- Patients can predict diseases based on the symptoms.
- Patients can search for doctors of their required specialization.
- Patients will only be able to see the doctors information and cannot edit the information.
- Doctors can register with their specializations and contact information.

3.5. NON-FUNCTIONAL REQUIREMENTS

- **Performance** : Under every circumstance our system responses are high measured from any point. But the accuracy of our project becomes unusually high when Decision tree is used.
- **Availability:** The application is likely accessible for a user at a given point in time
- **Security**: All data inside the system or its part will be protected against malware attacks or unauthorized access and password is encrypted and stored in the database.
- **Data privacy**: Patient health records are not saved inside the database as the symptom queries are responded at the moment and not saved into the database for the patients privacy.

4.DESIGN APPROACH

4.1. DESIGN APPROACH

PROPOSED ARCHITECTURE:

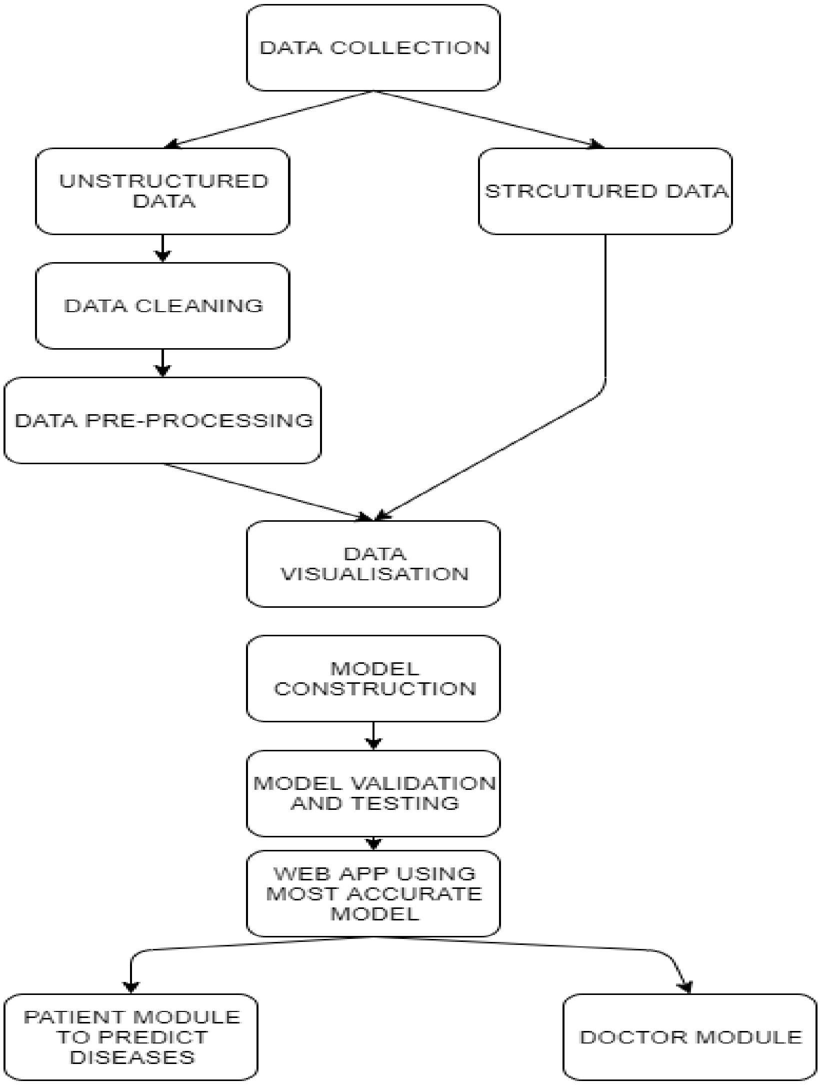


Figure 3 – Proposed Architecture

CLASS DIAGRAM:

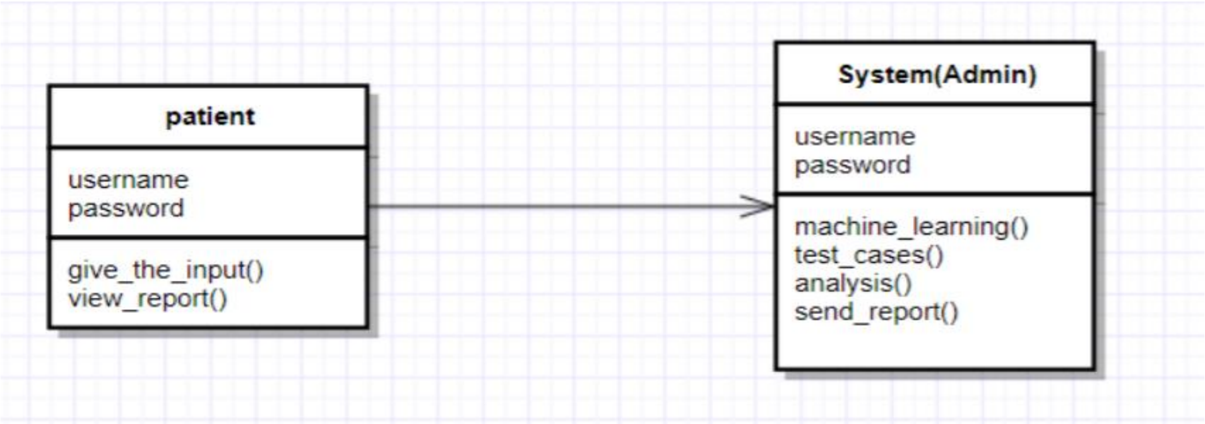


Figure 4 – Class Diagram

USE CASE DIAGRAM:

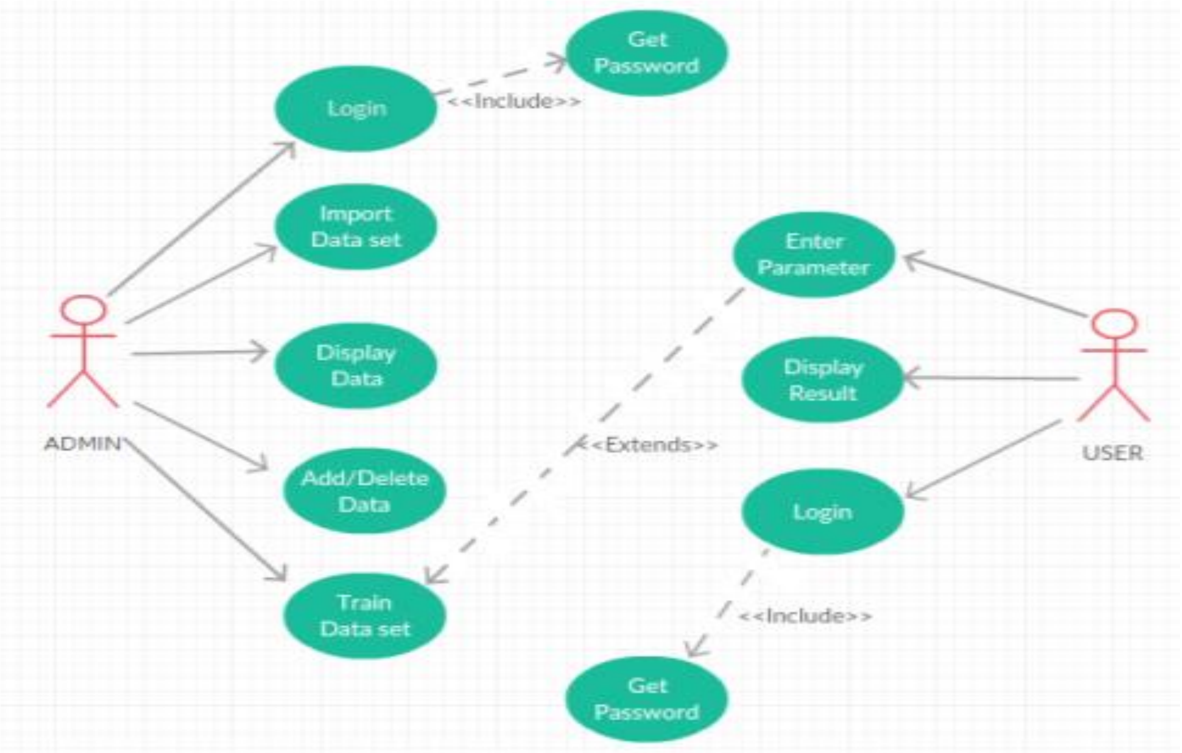


Figure 5 – Use Case Diagram

ACTIVITY DIAGRAM:

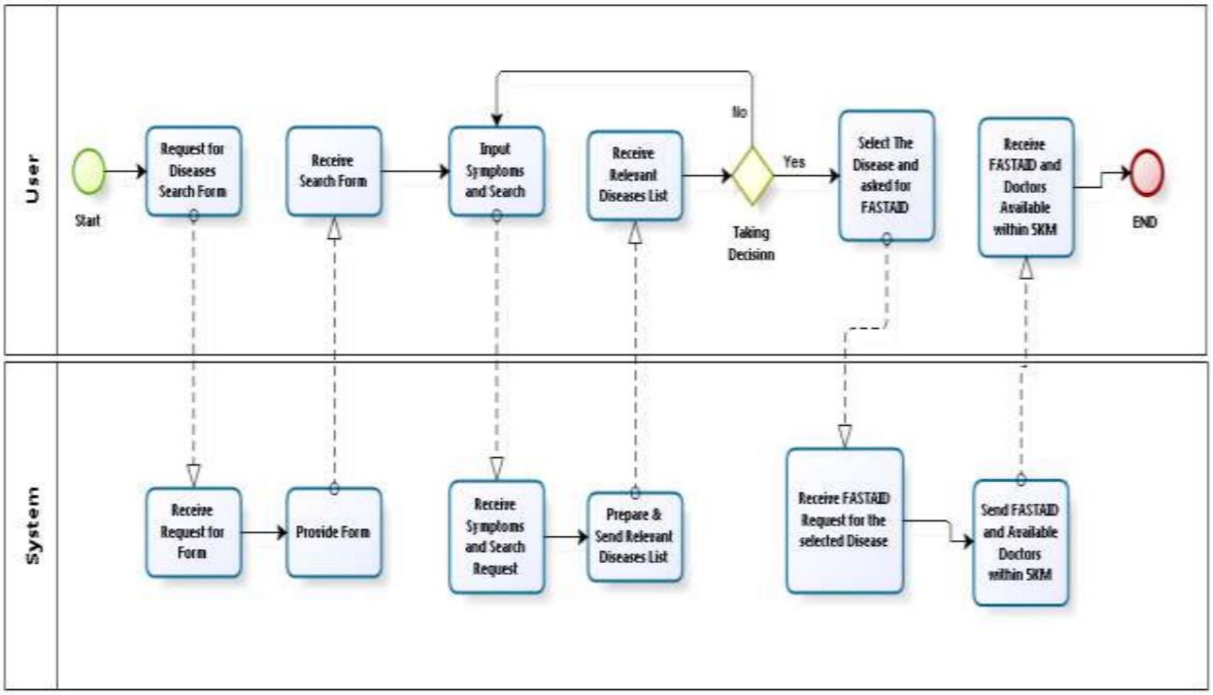

User
System
Start
Request for Diseases Search Form
Receive Search Form
Input Symptoms and Search
Receive Relevant Diseases List
No
Yes
Taking Decisinn
Select The Disease and asked for FASTAID
Receive FASTAID and Doctors Available within 5KM
END
Receive Request for Form
Provide Form
Receive Symptoms and Search Request
Prepare & Send Relevant Diseases List
Receive FASTAID Request for the selected Disease
Send FASTAID and Available Doctors within 5KM

Figure 6 – Activity Diagram

ER DIAGRAM:

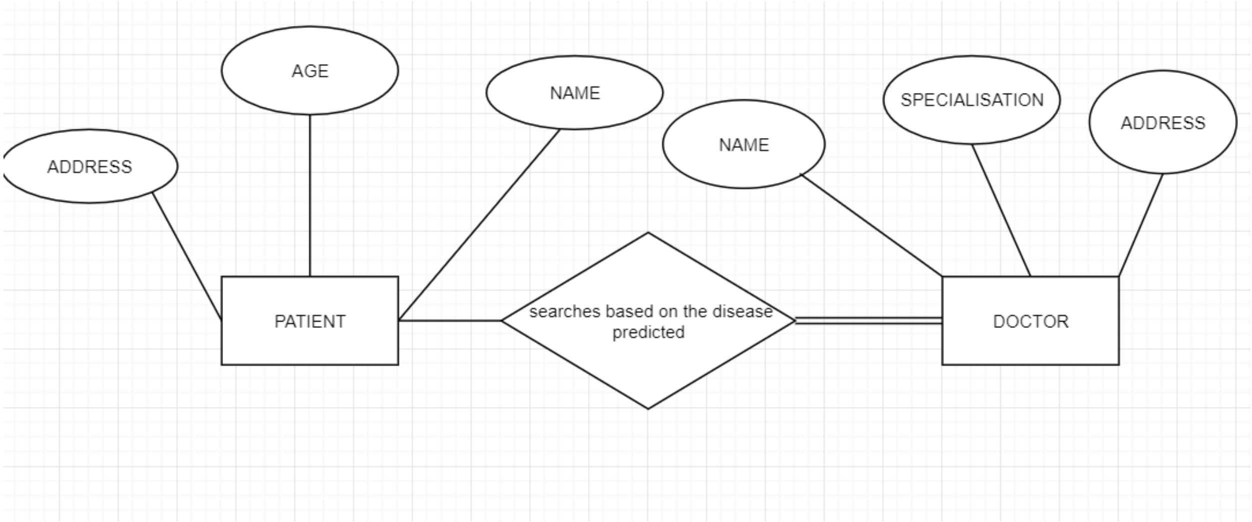


Figure 7 – ER Diagram

STATE DIAGRAM:

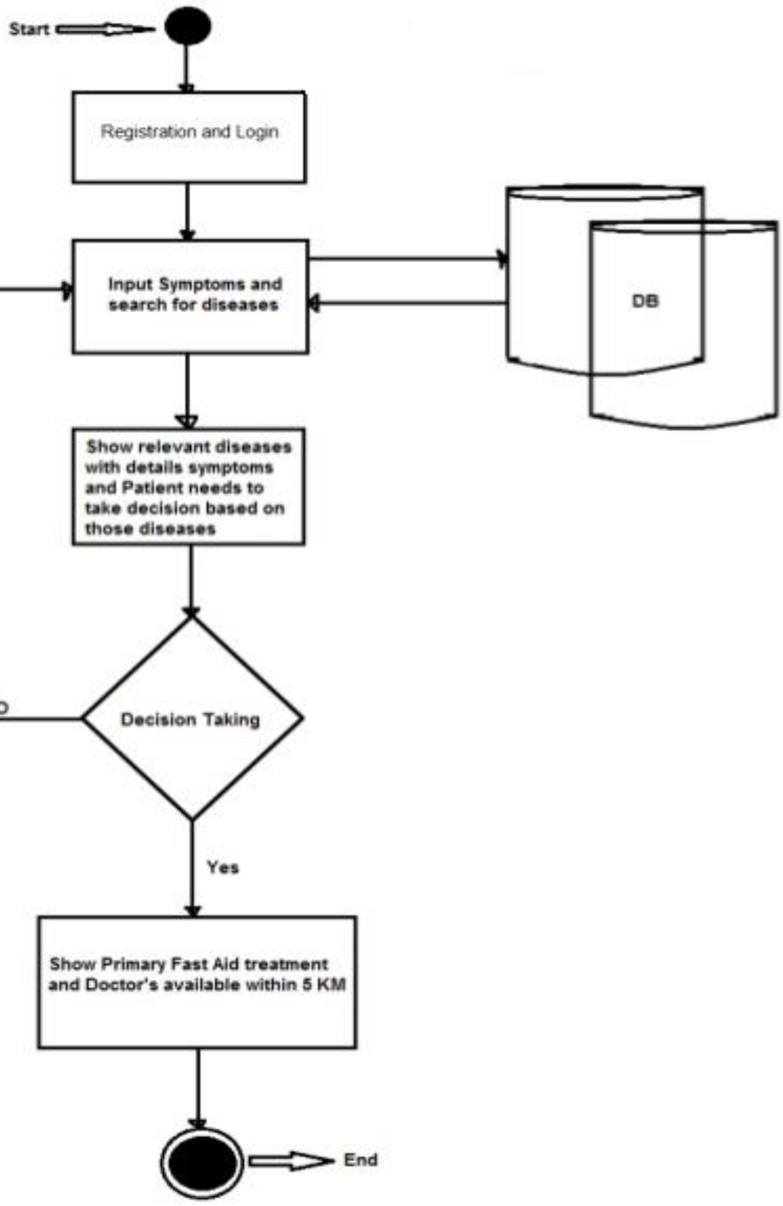

Start
Registration and Login
Input Symptoms and search for diseases
DB
Show relevant diseases with details symptoms and Patient needs to take decision based on those diseases
NO
Decision Taking
Yes
Show Primary Fast Aid treatment and Doctor's available within 5 KM
End

Figure 8 – State Diagram

SITEMAP:

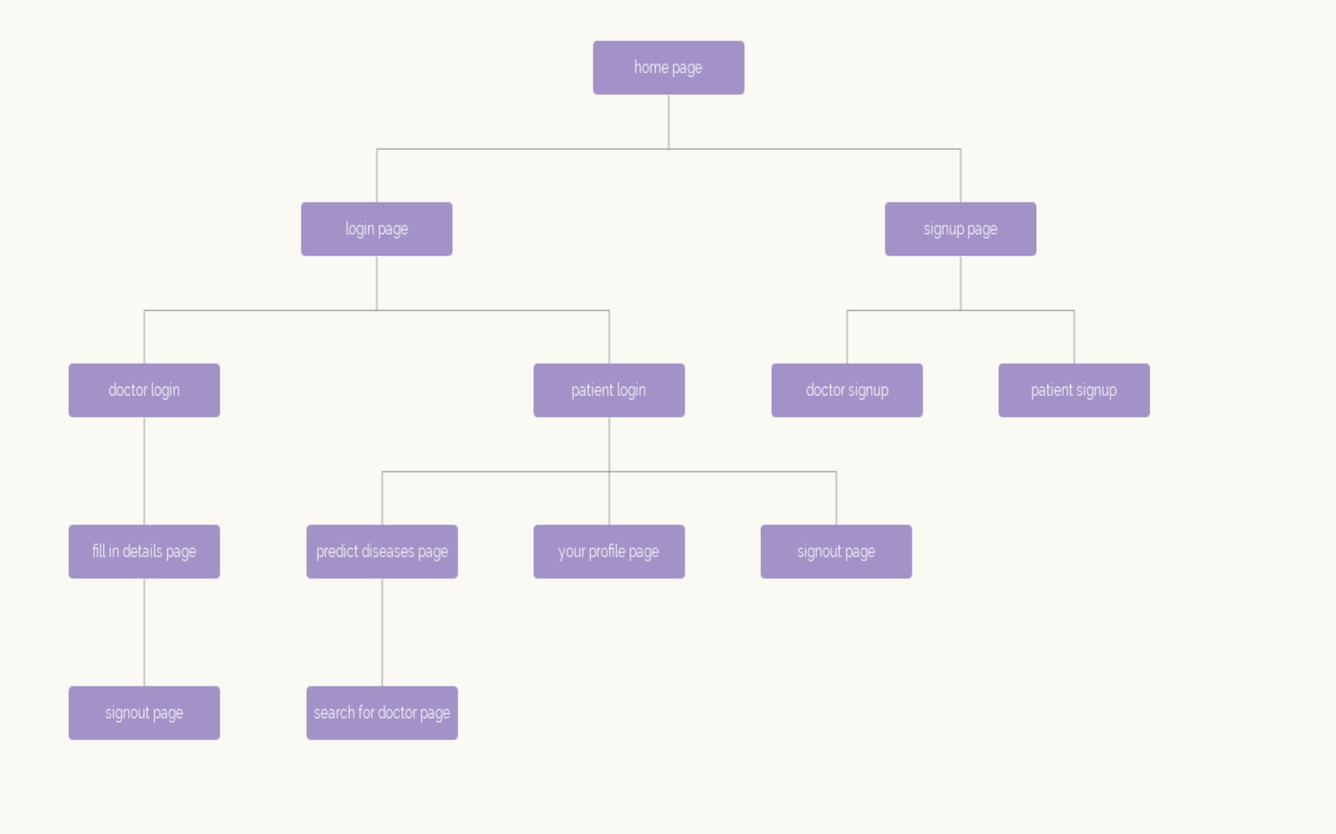


Figure 9 - Sitemap

4.2. CODES AND STANDARDS

Home.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>DISEASE PREDICTION SYSTEM</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/main.css') }}">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/util.css') }}">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/select2.min.css') }}">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/hamburgers.min.css') }}">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/animate.css') }}">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/font-awesome.min.css') }}">
   <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css') }}">
</head>
<body>

        <div class="limiter">
                <div class="container-login100">
                        <div class="wrap-login100">
                                <div class="login100-pic js-tilt" data-tilt>
```

```
                    <img src="{{ url_for('static', filename='images/img-01.png') }}" alt="IMG">

                </div>


                <form class="login100-form validate-form">

                    <span class="login100-form-title">

                        Login As

                    </span>


      <span>

        <a class="bt1" href={{ url_for('doctorlogin')}}>DOCTOR</a>

      </span>


      <span>

        <a class="bt2" href={{ url_for('patientlogin')}}>PATIENT</a>

      </span>


                    <div class="text-center p-t-136">

                        <a class="txt2" href={{ url_for('signup')}}>

                            New user? SIGNUP

                        </a>

                    </div>

                </form>
                </div>
            </div>
        </div>


    <script type=text/javascript src="{{url_for('static', filename='js/jquery-3.2.1.min.js') }}"></script>
    <script type=text/javascript src="{{url_for('static', filename='js/popper.js') }}"></script>
    <script type=text/javascript src="{{url_for('static', filename='js/bootstrap.min.js') }}"></script>
    <script type=text/javascript src="{{url_for('static', filename='js/select2.min.js') }}"></script>
    <script type=text/javascript src="{{url_for('static', filename='js/tilt.jquery.min.js') }}"></script>
    <script type=text/javascript src="{{url_for('static', filename='js/main.js') }}"></script>


        <script >
            $('.js-tilt').tilt({
                scale: 1.1
            })
        </script>
</body>
</html>
```

Main.css:

```
/*[ FONT ]*/
@font-face {
  font-family: Poppins-Regular;
  src: url('../fonts/poppins/Poppins-Regular.ttf');
}
* {
	margin: 0px;
	padding: 0px;
	box-sizing: border-box;
}

body, html {
	height: 100%;
	font-family: Poppins-Regular, sans-serif;
}
a {
	font-family: Poppins-Regular;
	font-size: 14px;
	line-height: 1.7;
	color: #666666;
	margin: 0px;
	transition: all 0.4s;
```

```
        -webkit-transition: all 0.4s;
  -o-transition: all 0.4s;
  -moz-transition: all 0.4s;
}


a:focus {
        outline: none !important;
}


a:hover {
        text-decoration: none;
  color: #57b846;
}
h1,h2,h3,h4,h5,h6 {
        margin: 0px;
}


p {
        font-family: Poppins-Regular;
        font-size: 14px;
        line-height: 1.7;
        color: #666666;
        margin: 0px;
```

```
}

ul, li {
        margin: 0px;
        list-style-type: none;
}
input {
        outline: none;
        border: none;
}

textarea {
  outline: none;
  border: none;
}

textarea:focus, input:focus {
  border-color: transparent !important;
}

input:focus::-webkit-input-placeholder { color:transparent; }
input:focus:-moz-placeholder { color:transparent; }
input:focus::-moz-placeholder { color:transparent; }
```

```
input:focus:-ms-input-placeholder { color:transparent; }


textarea:focus::-webkit-input-placeholder { color:transparent; }

textarea:focus:-moz-placeholder { color:transparent; }

textarea:focus::-moz-placeholder { color:transparent; }

textarea:focus:-ms-input-placeholder { color:transparent; }


input::-webkit-input-placeholder { color: #999999; }

input:-moz-placeholder { color: #999999; }

input::-moz-placeholder { color: #999999; }

input:-ms-input-placeholder { color: #999999; }


textarea::-webkit-input-placeholder { color: #999999; }

textarea:-moz-placeholder { color: #999999; }

textarea::-moz-placeholder { color: #999999; }

textarea:-ms-input-placeholder { color: #999999; }

button {

	outline: none !important;

	border: none;

	background: transparent;

}


button:hover {
        cursor: pointer;3
}


iframe {
        border: none !important;
}




.txt1 {
  font-family: Poppins-Regular;
  font-size: 13px;
  line-height: 1.5;
  color: #999999;
}


.txt2 {
  font-family: Poppins-Regular;
  font-size: 13px;
  line-height: 1.5;
  color: #666666;
}


.limiter {
  width: 100%;

  margin: 0 auto;

}



.container-login100 {

  width: 100%;

  min-height: 100vh;

  display: -webkit-box;

  display: -webkit-flex;

  display: -moz-box;

  display: -ms-flexbox;

  display: flex;

  flex-wrap: wrap;

  justify-content: center;

  align-items: center;

  padding: 15px;

  background: #9053c7;

  background: -webkit-linear-gradient(-135deg, #c850c0, #4158d0);

  background: -o-linear-gradient(-135deg, #c850c0, #4158d0);

  background: -moz-linear-gradient(-135deg, #c850c0, #4158d0);

  background: linear-gradient(-135deg, #c850c0, #4158d0);

}
```

```
.wrap-login100 {
  width: 960px;
  background: #fff;
  border-radius: 10px;
  overflow: hidden;

  display: -webkit-box;
  display: -webkit-flex;
  display: -moz-box;
  display: -ms-flexbox;
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  padding: 177px 130px 33px 95px;
}
.login100-pic {
  width: 316px;
}

.login100-pic img {
  max-width: 100%;
}
.login100-form {
  width: 290px;
}

.login100-form-title {
  font-family: Poppins-Bold;
  font-size: 24px;
  color: #333333;
  line-height: 1.2;
  text-align: center;

  width: 100%;
  display: block;
  padding-bottom: 54px;
}
.wrap-input100 {
  position: relative;
  width: 100%;
  z-index: 1;
  margin-bottom: 10px;
}
}
```

Main.js:

```
(function ($) {
    "use strict";
    [ Validate ]*/
    var input = $('.validate-input .input100');

    $('.validate-form').on('submit',function(){
        var check = true;

        for(var i=0; i<input.length; i++) {
            if(validate(input[i]) == false){
                showValidate(input[i]);
                check=false;
            }
        }

        return check;
    });


    $('.validate-form .input100').each(function(){
        $(this).focus(function(){
           hideValidate(this);
        });
    });


    function validate (input) {
        if($(input).attr('type') == 'email' || $(input).attr('name') == 'email') {
            if($(input).val().trim().match(/^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{1,5}|[0-9]{1,3})(\]?)$/) == null) {
                return false;
            }
        }
        else {
            if($(input).val().trim() == ''){
                return false;
            }
        }
    }


    function showValidate(input) {
        var thisAlert = $(input).parent();


        $(thisAlert).addClass('alert-validate');
    }
```

```
    function hideValidate(input) {
        var thisAlert = $(input).parent();


        $(thisAlert).removeClass('alert-validate');
    }


})(jQuery);
```

Python

Flask.py

```
from flask import Flask,redirect,url_for,render_template,request,session,flash
import bcrypt
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
import ex
from pymongo import MongoClient




client = MongoClient()
app = Flask(__name__)

```

```
@app.route('/')

def home():

    return render_template("home.html")



@app.route('/logout')

def logout():

    session.pop('login_user',None)

    return redirect(url_for('patientlogin'))



@app.route('/patientprofile')

def patientprofile():

    return render_template("patientprofile.html")




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

def doctorlogin():

    if request.method =='POST':

        db1 = client.doctors

        doctor=db1.my_collection

        login_user = doctor.find_one({'name' : request.form['username']})

        if login_user:

            if request.form['pass'] == login_user['password']:

                session['username'] = request.form['username']
                return redirect(url_for('home'))

        return 'Invalid credentials'
    return render_template('doctor.html')

@app.route('/patientlogin',methods=['POST','GET'])
def patientlogin():
    if request.method =='POST':
        db2 = client.patients
        patient=db2.my_collection
        login_user = patient.find_one({'name' : request.form['username']})
        if login_user:
            if request.form['pass'] == login_user['password']:
                session['username'] = request.form['username']
                return redirect(url_for('patientprofile'))

        return 'Invalid credentials'
    return render_template("patient.html")

@app.route('/prediction',methods=['POST','GET'])
def prediction():
    if request.method=='POST':
        to_predict_list=[]
    for key in request.form.keys():
        to_predict_list = request.form.getlist(key)
    standard_list = '''itching skin_rash nodal_skin_eruptions continuous_sneezing shivering
chills joint_pain stomach_pain acidity ulcers_on_tongue muscle_wasting vomiting
burning_micturition spotting_urination fatigue weight_gain anxiety cold_hands_and_feets
mood_swings weight_loss restlessness lethargy patches_in_throat irregular_sugar_level cough
high_fever sunken_eyes breathlessness sweating dehydration indigestion headache
yellowish_skin dark_urine nausea loss_of_appetite pain_behind_the_eyes back_pain
constipation abdominal_pain diarrhoea mild_fever yellow_urine yellowing_of_eyes
acute_liver_failure fluid_overload swelling_of_stomach swelled_lymph_nodes malaise
blurred_and_distorted_vision phlegm throat_irritation redness_of_eyes sinus_pressure
runny_nose congestion chest_pain weakness_in_limbs fast_heart_rate
pain_during_bowel_movements pain_in_anal_region bloody_stool irritation_in_anus neck_pain
dizziness cramps bruising obesity swollen_legs swollen_blood_vessels puffy_face_and_eyes
enlarged_thyroid brittle_nails swollen_extremeties excessive_hunger extra_marital_contacts
drying_and_tingling_lips slurred_speech knee_pain hip_joint_pain muscle_weakness stiff_neck
swelling_joints movement_stiffness spinning_movements loss_of_balance unsteadiness
weakness_of_one_body_side loss_of_smell bladder_discomfort foul_smell_of urine
continuous_feel_of_urine passage_of_gases internal_itching toxic_look_(typhos) depression
irritability muscle_pain altered_sensorium red_spots_over_body belly_pain
abnormal_menstruation dischromic_patches watering_from_eyes increased_appetite polyuria
family_history mucoid_sputum rusty_sputum lack_of_concentration visual_disturbances
receiving_blood_transfusion receiving_unsterile_injections coma stomach_bleeding
distention_of_abdomen history_of_alcohol_consumption fluid_overload blood_in_sputum
prominent_veins_on_calf palpitations painful_walking pus_filled_pimples blackheads scurring
skin_peeling silver_like_dusting small_dents_in_nails inflammatory_nails blister
red_sore_around_nose yellow_crust_ooze'''
    nlist=[]
    standard_list=standard_list.replace("\s+"," ")
```

```
        for x in standard_list.split(" "):

            if x in to_predict_list:

                nlist.append(1)

            else:

                nlist.append(0)

        result=ex.ValuePredictor(nlist)

        return render_template("prediction.html",result=result)



    return render_template("prediction.html")



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

def search():

    if request.method=='POST':

        data=client.doctors

        doctor=data.my_collection

        spl=request.form['specialisation']

        result=doctor.find_one({'specialisation':spl})

        return
render_template("search.html",name=result['name'],specialisation=result['specialisation'],info
=result['info'])

    return render_template("search.html")



@app.route('/signup')
```

```
def signup():
    return render_template("signup.html")


@app.route('/doctorsignup', methods=['POST', 'GET'])
def doctorsignup():
     if request.method == 'POST':
        db1 = client.doctors
        doctor=db1.my_collection
        existing_user = doctor.find_one({'name' : request.form['username']})
        if existing_user is None:
            doctor.insert({'name' : request.form['username'], 'password' :
request.form['pass'],'specialisation' : request.form['specialisation'],'info' :
request.form['contact']})
            session['username'] = request.form['username']
            return redirect(url_for('doctorlogin'))


        return ('Username already exists!')
     return render_template("doctorsignup.html")


@app.route('/patientsignup', methods=['POST', 'GET'])
def patientsignup():
    if request.method == 'POST':
        db2 = client.patients
        patient=db2.my_collection

        existing_user = patient.find_one({'name' : request.form['username']})

        if existing_user is None:

            patient.insert({'name' : request.form['username'], 'password' : request.form['pass']})

            session['username'] = request.form['username']

            return redirect(url_for('patientlogin'))


        return ('Username already exists!')

    return render_template("patientsignup.html")


if __name__ == '__main__':

    app.secret_key = 'mysecret'

    app.run(debug=True)
```

PREDICTIVE ANALYSIS OF STRUCTURED DATA:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.svm import SVC

from sklearn.metrics import mean_squared_error

from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler,LabelEncoder,LabelBinarizer

from sklearn.metrics import accuracy_score,confusion_matrix,log_loss

from sklearn.model_selection import cross_val_score

from sklearn.multiclass import OneVsRestClassifier,OneVsOneClassifier

%matplotlib inline

from sklearn.naive_bayes import MultinomialNB

import pickle
```

```
train_data=pd.read_csv(r'Training.csv')

test_data=pd.read_csv(r'Testing.csv')

train_data.info()

train_data.shape
```

```
Output: <class 'pandas.core.frame.DataFrame'>
RangeIndex: 4920 entries, 0 to 4919
Columns: 133 entries, itching to prognosis
dtypes: int64(132), object(1)
memory usage: 5.0+ MB
(4920, 133)
```

```
X_train=train_data.drop('prognosis',axis=1)

Y_train=enc.fit_transform(train_data['prognosis'])

X_test=test_data.drop('prognosis',axis=1)

Y_test=enc.fit_transform(test_data['prognosis'])

n_classes=np.unique(Y_test).shape[0]

n_classes
```

PRINCIPAL COMPONENT ANALYSIS

```
from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

import numpy as np

from numpy import linalg as LA

from sklearn.preprocessing import StandardScaler



S=np.dot(X_train.T,X_train)

#print(S)

eign_value,eign_vect=LA.eig(S)

print(len(eign_value))

eig_sort=np.argsort(eign_value)[::-1]

eig_vect_sort=None

for i in eig_sort:

  if eig_vect_sort is None:

    eig_vect_sort=eign_vect[i]

  else:

    eig_vect_sort=np.vstack((eig_vect_sort,eign_vect[i]))

eig_vect_sort=eig_vect_sort.T

S_=np.dot(S,eig_vect_sort)

#print(S_)

propotion_variance=[]

c=0
```

```
s=0
for v in eign_value:
    f=v
    if(c==0 or c==1):
     #print(v/sum(eign_value))
     propotion_variance.append(f/sum(eign_value))
    else:
      #print(s/sum(eign_value))
      propotion_variance.append(s/sum(eign_value))
    s=s+f
    c+=1
print(len(propotion_variance))
plt.figure(figsize=(50,10))
import matplotlib.pyplot as plt
cols=[i for i in range(len(eign_value))]
plt.plot(cols, propotion_variance)
plt.xticks(cols)
plt.show()
print(eig_vect_sort.shape)
```

ARTIFICIAL NEURAL NETWORK

```
from keras import Sequential
from keras.layers import Dense,Activation
model=Sequential()
model.add(Dense(120,input_dim=X_train.shape[1]))
model.add(Dense(60))
model.add(Dense(n_classes,activation='softmax'))
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
history=model.fit(X_train,Y_train_cat,epochs=100,validation_data=(X_test,Y_test_cat),verbose=1)
train_acc=history.history['acc']
val_acc=history.history['val_acc']
train_loss=history.history['loss']
val_loss=history.history['val_loss']
plt.figure(figsize=(20,10))
epochs=[i for i in range(100)]
plt.plot(epochs,train_acc)
```

SUPPORT VECTOR MACHINE

```
svm=cross_val_score(SVC(kernel='linear'),X_train,Y_train,cv=5)
svm_clf=SVC(kernel='linear')
svm_clf.fit(x_train,y_train)
y_pred=svm_clf.predict(x_test)
sns.heatmap(confusion_matrix(y_pred,y_test))
one_all= OneVsRestClassifier(estimator=SVC(kernel='linear'))
```

```
one_all.fit(X_train,Y_train)

def sqrror_loss(y_true,y_pred):

    loss=0

    for true,pred in zip(y_true,y_pred):

        loss+=(true-pred)**2

    return loss

confusion_matrix(rf_clf.predict(X_test),Y_test)

one_one= OneVsOneClassifier(estimator=SVC(kernel='linear'))

one_one.fit(X_train,Y_train)

one_one.score(X_test,Y_test)
```

RANDOM FOREST

```
rf_clf=RandomForestClassifier(n_estimators=100)

rf_clf.fit(X_train,Y_train)

confusion_matrix(rf_clf.predict(X_test),Y_test)

Y_pred=rf_clf.predict(X_test)
```

PREDICTIVE ANALYSIS OF UNSTRUCTURED DATA

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler,LabelEncoder,LabelBinarizer

from sklearn.metrics import accuracy_score,confusion_matrix,log_loss

from sklearn.model_selection import cross_val_score

from sklearn.multiclass import OneVsRestClassifier,OneVsOneClassifier

%matplotlib inline

from sklearn.naive_bayes import MultinomialNB

import pickle
```

DATA CLEANING

```
import csv

from collections import defaultdict



disease_list = []



def return_list(disease):
    disease_list = []

    match = disease.replace('^','_').split('_')

    ctr = 1

    for group in match:

        if ctr%2==0:

            disease_list.append(group)

        ctr = ctr + 1



    return disease_list



with open("Scraped-Data/dataset_uncleaned.csv") as csvfile:

    reader = csv.reader(csvfile)

    disease=""

    weight = 0

    disease_list = []

    dict_wt = {}

    dict_=defaultdict(list)

    for row in reader:



        if row[0]!="\xc2\xa0" and row[0]!="":

            disease = row[0]

            disease_list = return_list(disease)

            weight = row[1]
        if row[2]!="\xc2\xa0" and row[2]!="":
            symptom_list = return_list(row[2])


            for d in disease_list:
                for s in symptom_list:
                    dict_[d].append(s)
                dict_wt[d] = weight


    #print (dict_)
with open("Scraped-Data/dataset_clean.csv","w") as csvfile:
    writer = csv.writer(csvfile)
    for key,values in dict_.items():
        for v in values:
            #key = str.encode(key)
            key = str.encode(key).decode('utf-8')
            #.strip()
            #v = v.encode('utf-8').strip()
            #v = str.encode(v)
            writer.writerow([key,v,dict_wt[key]])
columns = ['Source','Target','Weight']
data = pd.read_csv("Scraped-Data/dataset_clean.csv",names=columns, encoding ="ISO-8859-1")
```

```
data.head()

data.to_csv("Scraped-Data/dataset_clean.csv",index=False)

slist = []

dlist = []

with open("Scraped-Data/nodetable.csv","w") as csvfile:

    writer = csv.writer(csvfile)



    for key,values in dict_.items():

        for v in values:

            if v not in slist:

                writer.writerow([v,v,"symptom"])

                slist.append(v)

        if key not in dlist:

            writer.writerow([key,key,"disease"])

            dlist.append(key)

nt_columns = ['Id','Label','Attribute']

nt_data = pd.read_csv("Scraped-Data/nodetable.csv",names=nt_columns, encoding ="ISO-8859-1",)

nt_data.to_csv("Scraped-Data/nodetable.csv",index=False)
```

```
data = pd.read_csv("Scraped-Data/dataset_clean.csv", encoding ="ISO-8859-1")

df = pd.DataFrame(data)
```

MODEL CLASSIFICATION - MULTINOMIAL NAIVE BAYES CLASSIFIER

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.naive_bayes import MultinomialNB
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
mnb_tot = MultinomialNB()
mnb_tot = mnb_tot.fit(x, y)
mnb_tot.score(x, y)
disease_pred = mnb_tot.predict(x)
disease_real = y.values
for i in range(0, len(disease_real)):
    if disease_pred[i]!=disease_real[i]:
        print ('Pred: {0} Actual:{1}'.format(disease_pred[i], disease_real[i]))
```

MODEL CONSTRUCTION - DECISION TREE

```
from sklearn.tree import DecisionTreeClassifier, export_graphviz
print ("DecisionTree")
dt = DecisionTreeClassifier()
clf_dt=dt.fit(x,y)
```

```
print ("Acurracy: ", clf_dt.score(x,y))

from sklearn import tree

from sklearn.tree import export_graphviz



export_graphviz(dt,

                out_file='DOT-files/tree.dot',

                feature_names=cols)

from IPython.display import Image

Image(filename='DOT-files/tree.png')
```

MODEL CONSTRUCTION – SUPPORT VECTOR MACHINE

```
from sklearn.cross_validation

import train_test_split x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)

svm_clf=SVC()

svm_clf.fit(x_train,y_train)

svm_clf.score(x,y)
```

5. SCHEDULE, TASKS AND MILESTONES

Table 1

TASK	START DATE	END DATE	DURATION (days)
Choosing problem statement	02-12-19	07-12-19	7
Analysing the scope of the selected problem statement	08-12-19	115-12-19	7
Objective, abstract and motivation	16-12-19	23-12-19	7
Literature review	24-12-19	31-12-19	7
Design Approach	1-1-20	5-1-20	5
Data collection and cleaning	6-1-20	8-1-20	2
Implementation-Prediction of diseases based on the symptoms for structured data	9-1-20	17-1-20	8
Review 1 documentation and presentation	18-1-20	22-1-20	4
Implementation-Prediction of diseases based on the symptoms for structured data and unstructured data	23-1-20	29-2-20	6
Review 2 documentation and presentation	1-3-20	3-2-20	2
Implementation - Prediction of diseases based on the symptoms for unstructured data	5-3-20	12-3-20	7
Implementation - Analysis and comparison of machine learning algorithms	13-3-20	17-3-20	4
Implementation -Website frontend	18-3-20	30-3-20	12
Implementation – Flask Integration	1-4-20	10-4-20	10
Final documentation and presentation	11-4-20	22-4-20	11

6. PROJECT DEMONSTRATION

HOME PAGE:

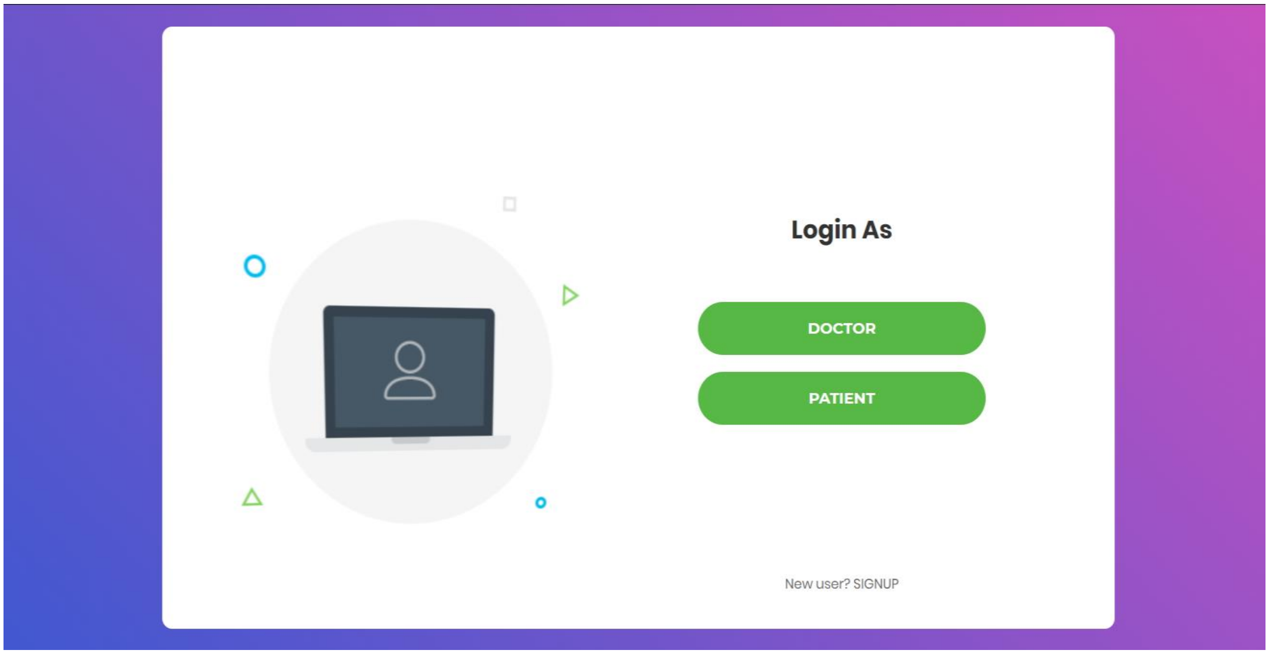


Figure 10 - Home Page

SIGNUP PAGE:

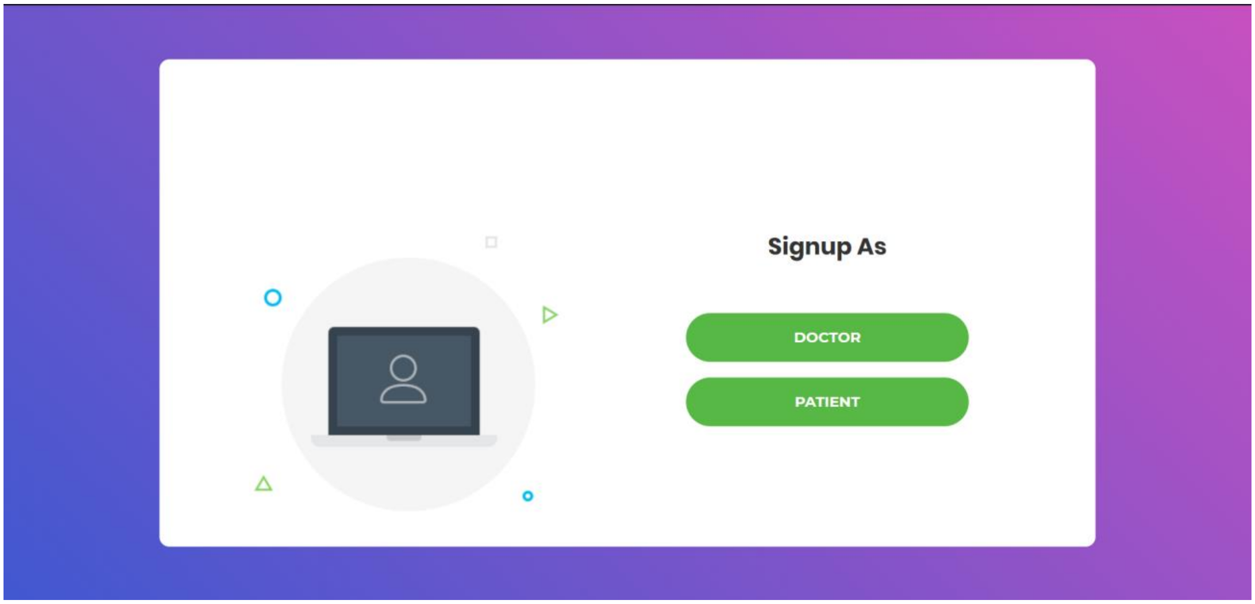
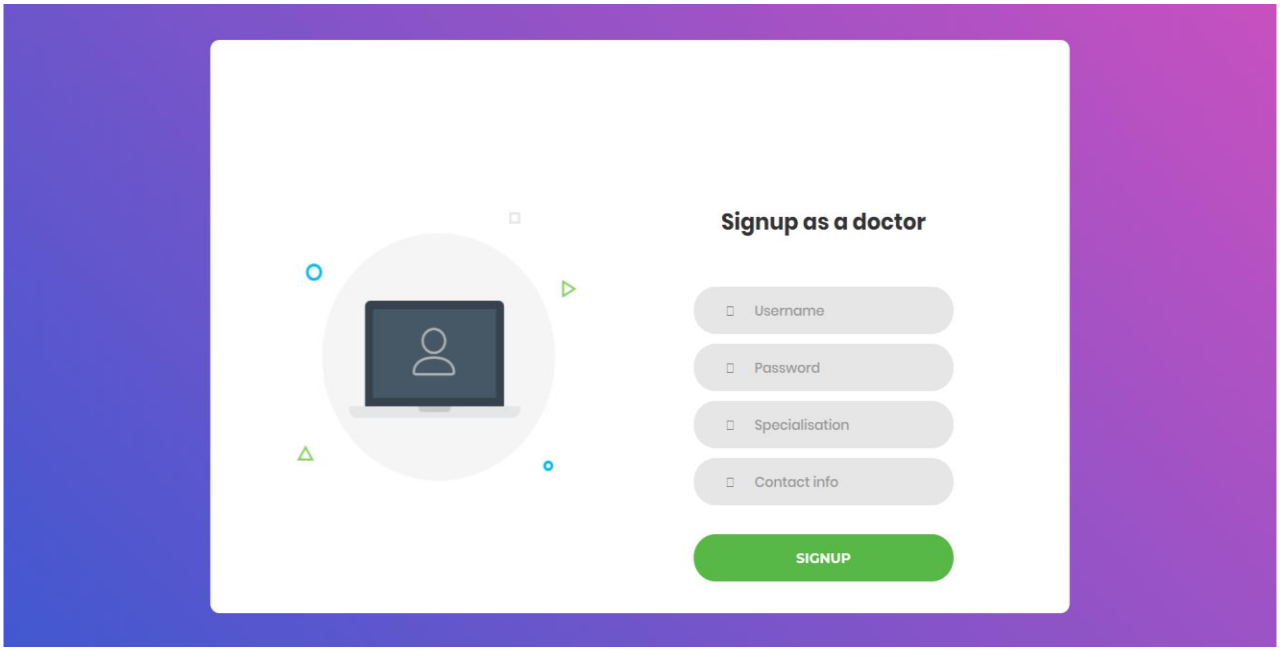


Figure 11 – Signup Page

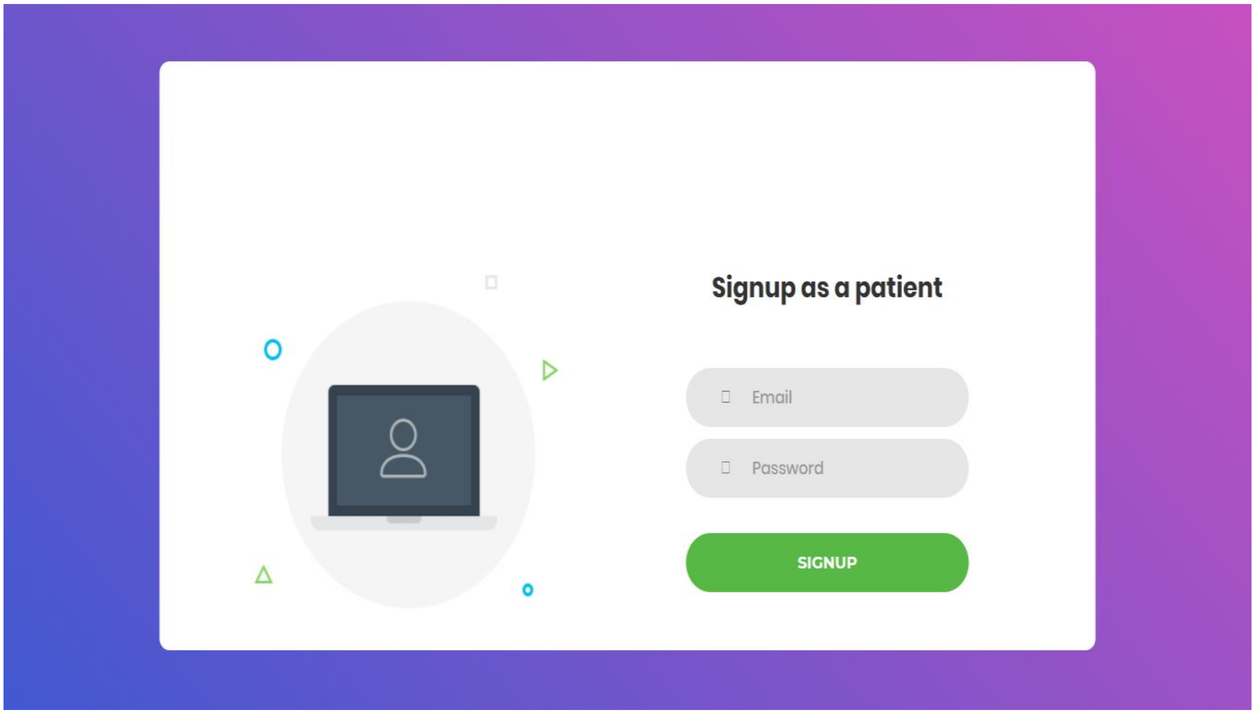
SIGNUP AS DOCTOR:



Signup as a doctor

Figure 12 – Doctor Signup Page

SIGNUP AS PATIENT:



Signup as a patient

Figure 13 – Patient Signup Page

DOCTOR LOGIN:

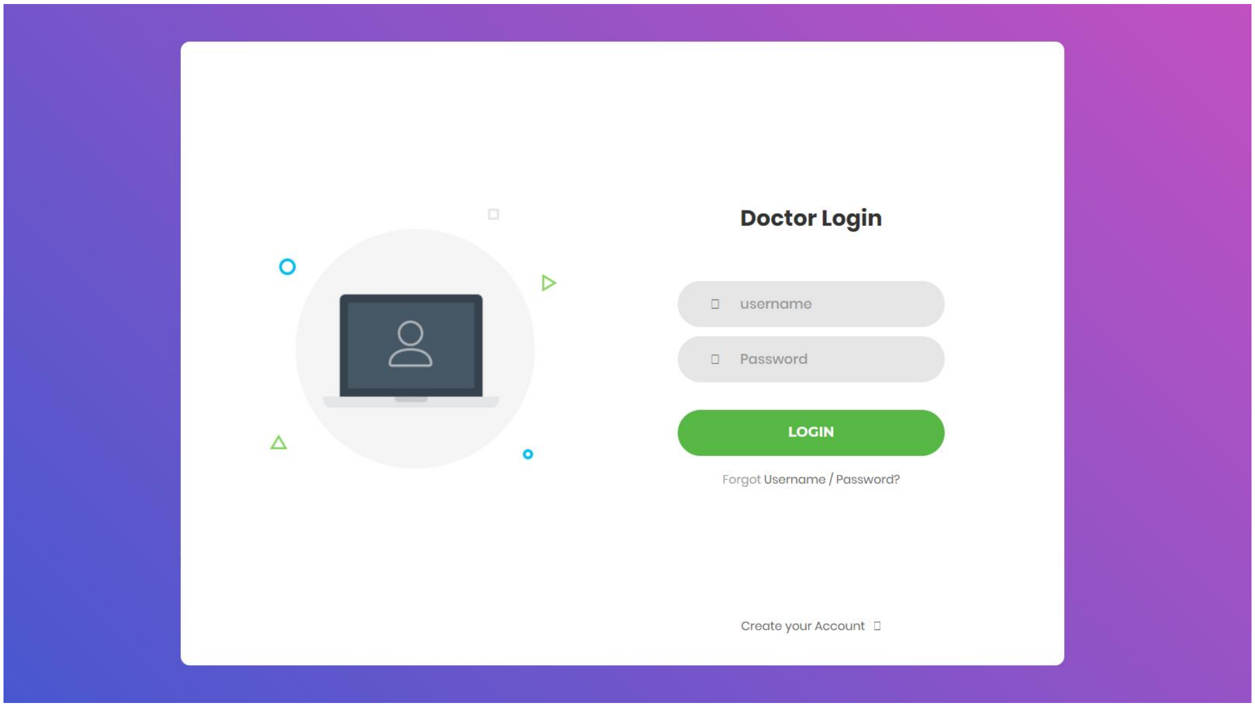


Figure 14 – Doctor Login Page

PATIENT LOGIN:

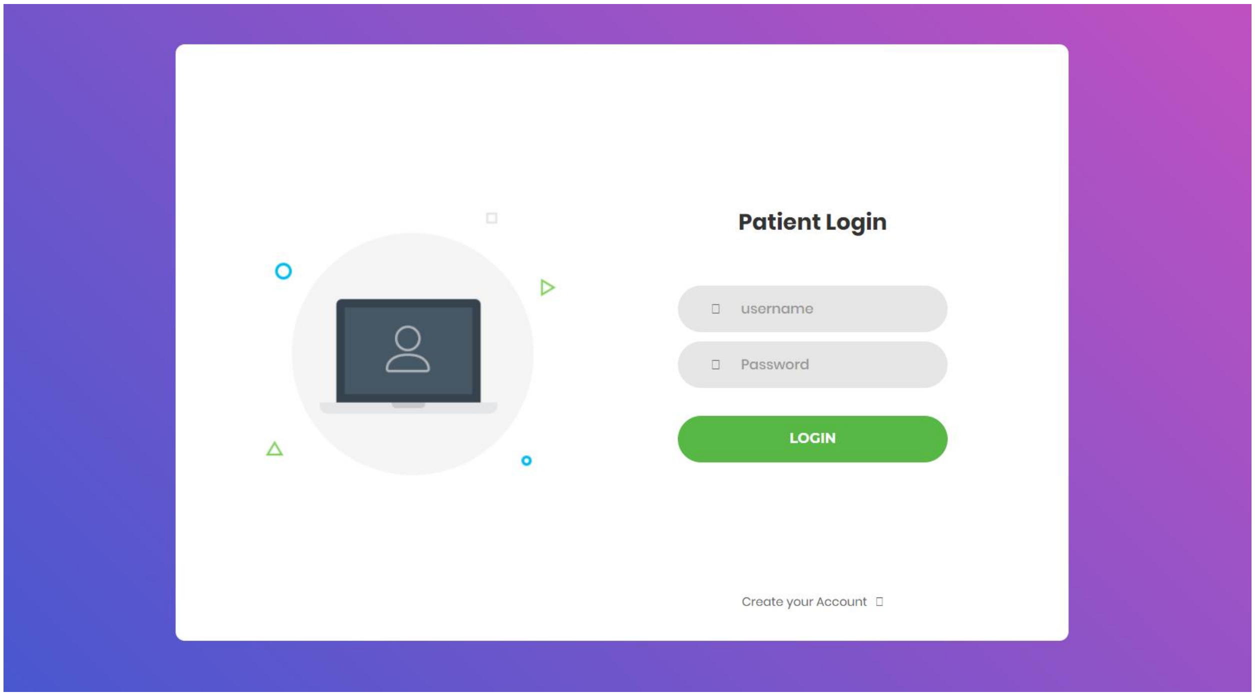


Figure 15 – Patient Login Page

PATIENT PROFILE PAGE:

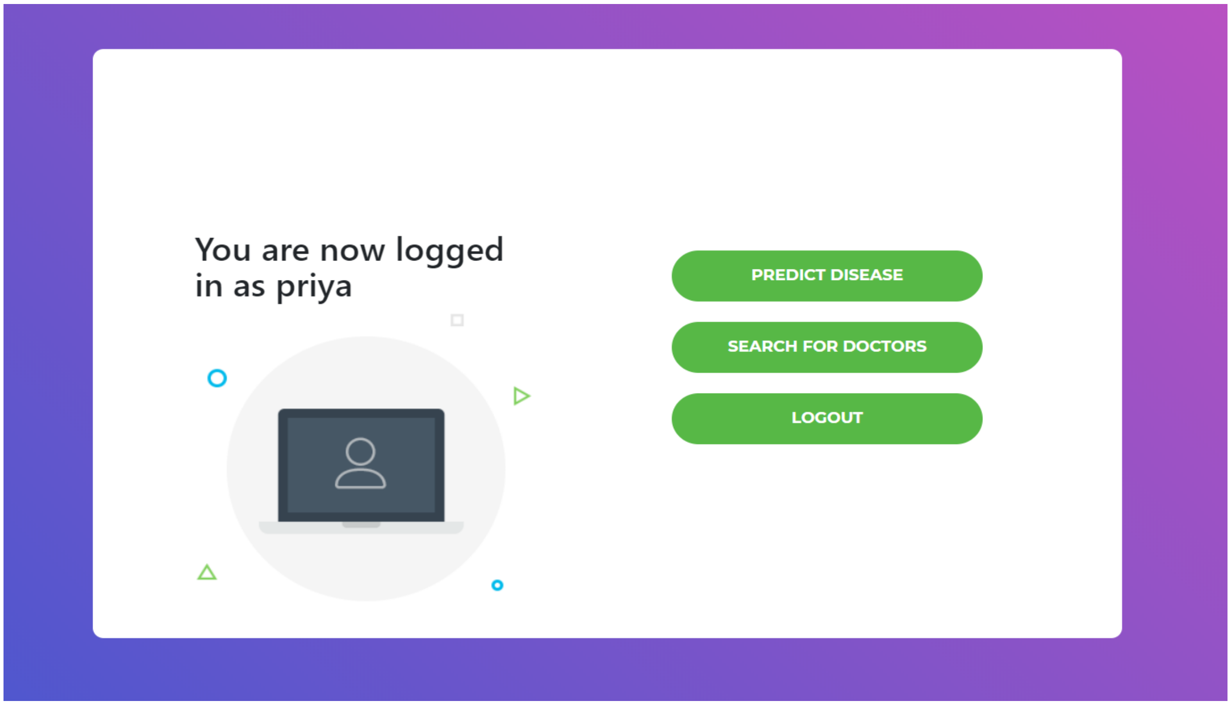
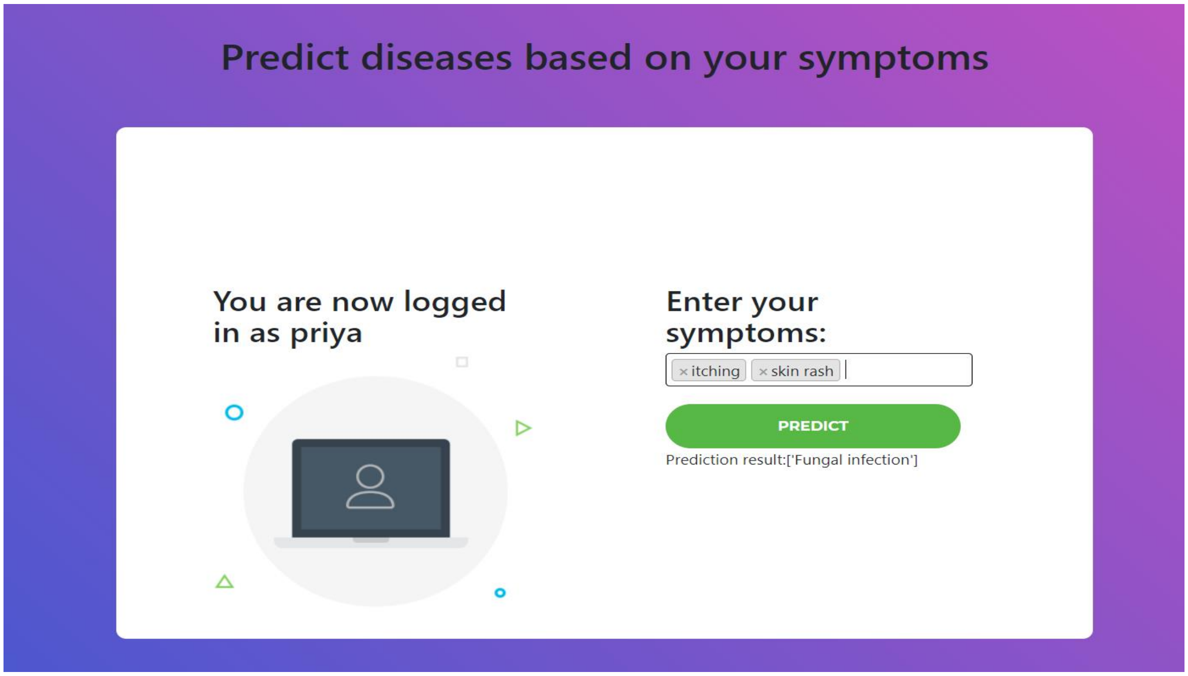

You are now logged in as priya

Figure 16 – Patient Profile Page

PREDICTION OF DISEASES:


Predict diseases based on your symptoms

You are now logged in as priya

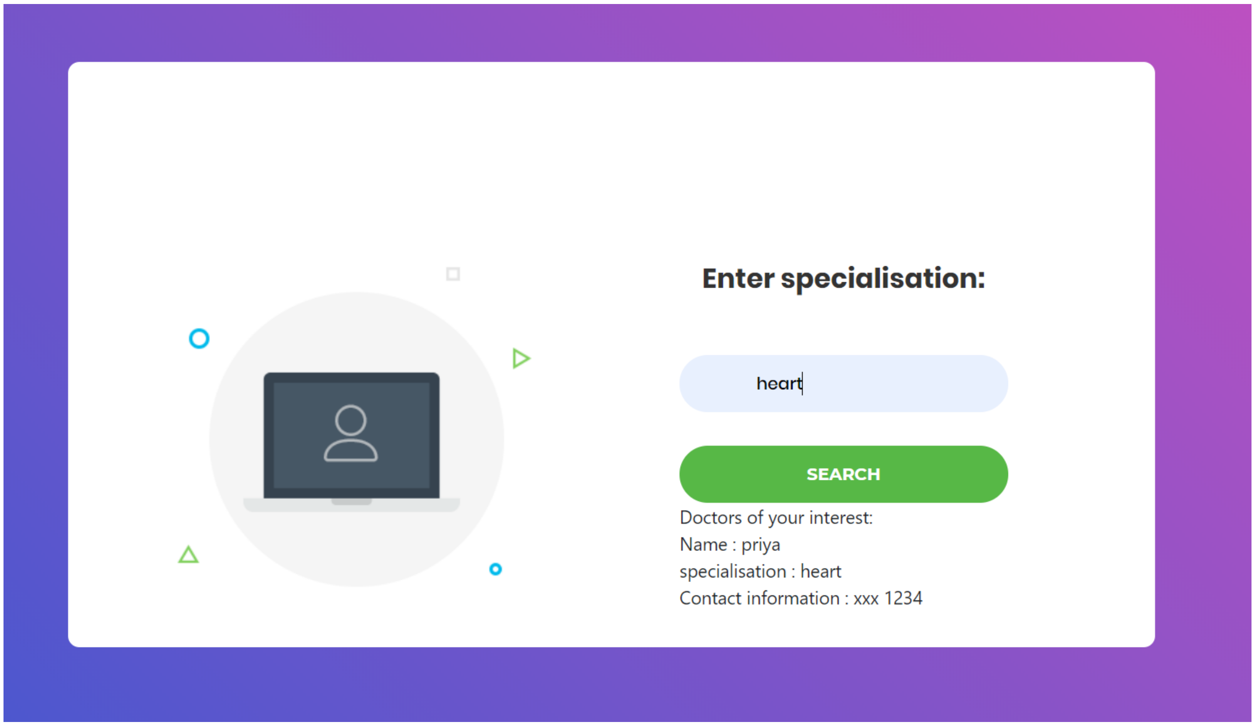
Enter your symptoms:

× itching × skin rash

Prediction result:['Fungal infection']

Figure 17 – Disease Prediction Page

SEARCH FOR DOCTORS:



Enter specialisation:

heart

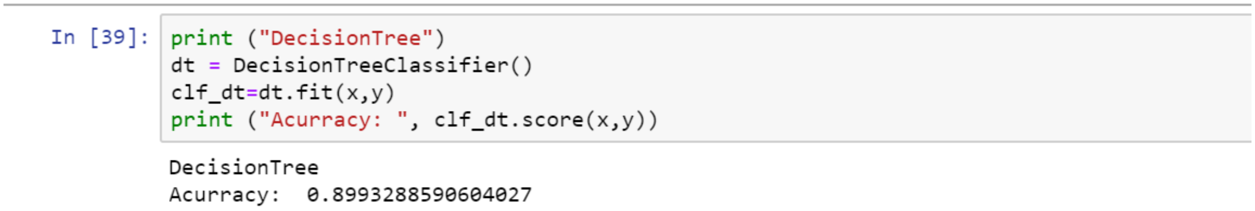
SEARCH

Doctors of your interest:
Name : priya
specialisation : heart
Contact information : xxx 1234

Figure 18 - Search for Doctors Page

UNSTRUCTURED DATA:

DECISION TREE:



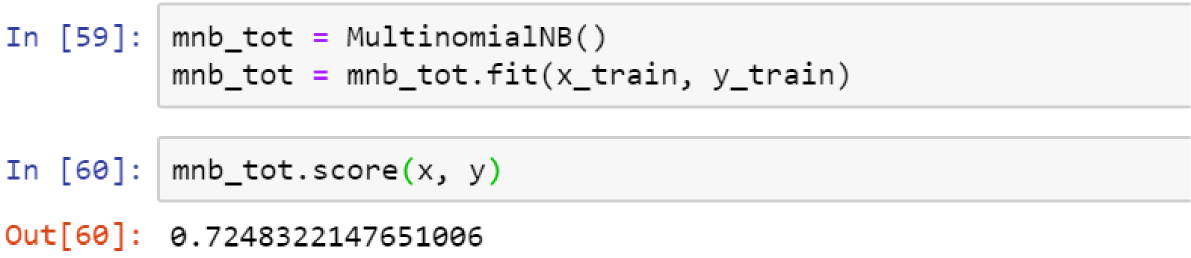
```
In [39]: print ("DecisionTree")
         dt = DecisionTreeClassifier()
         clf_dt=dt.fit(x,y)
         print ("Acurracy: ", clf_dt.score(x,y))

         DecisionTree
         Acurracy:  0.8993288590604027
```

Figure 19 - Decision Tree Accuracy

According to the plotted decision tree, Jugular venous distention is the attribute symptom that has the highest gini score of 0.9846. Thus this symptom would play a major role in predicting diseases.

MULTINOMIAL NAIVE BAYES CLASSIFIER:



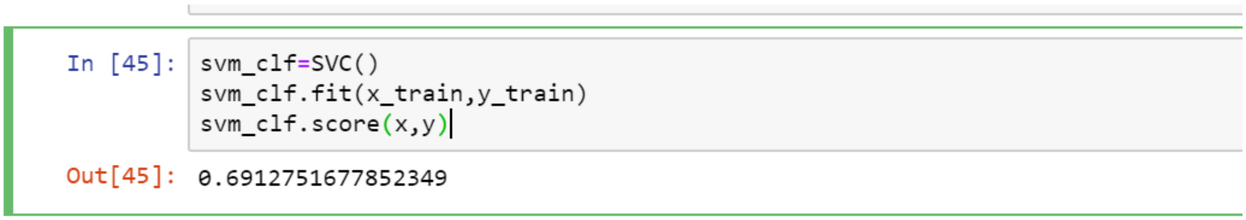
```
In [59]: mnb_tot = MultinomialNB()
         mnb_tot = mnb_tot.fit(x_train, y_train)

In [60]: mnb_tot.score(x, y)

Out[60]: 0.7248322147651006
```

Figure 20 - Multinomial Naive Bayes Accuracy

SUPPORT VECTOR MACHINE:



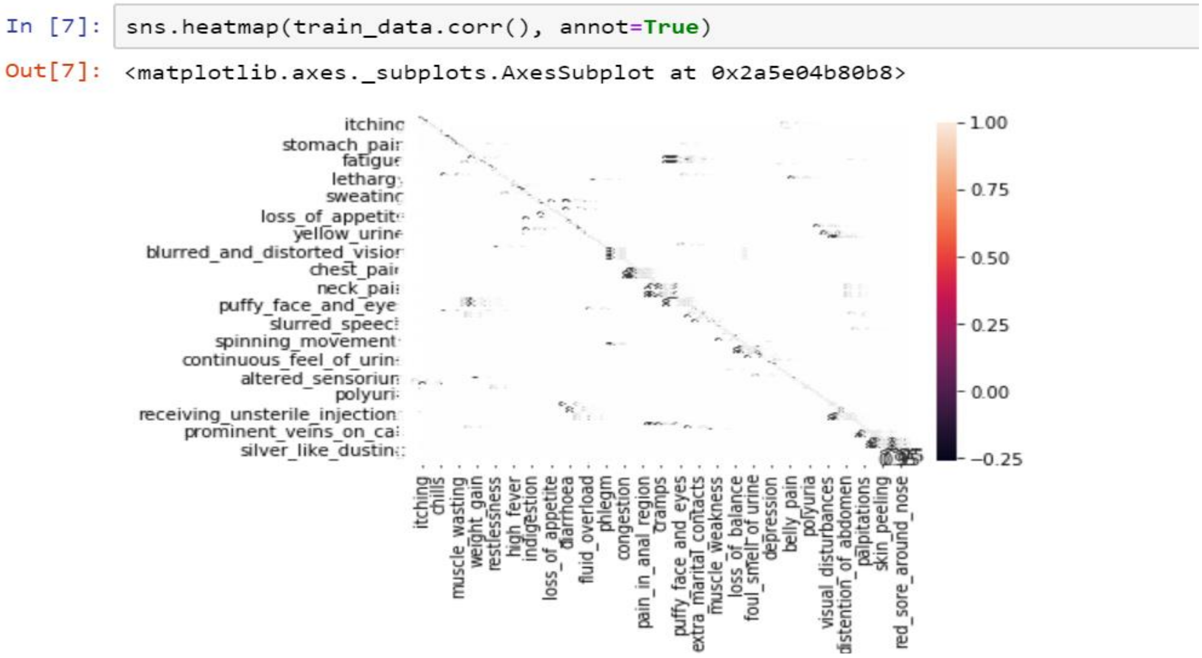
```
In [45]: svm_clf=SVC()
         svm_clf.fit(x_train,y_train)
         svm_clf.score(x,y)

Out[45]: 0.6912751677852349
```

Figure 21- Support Vector Machine Accuracy

STRUCTURED DATA

CORRELATION MATRIX:



```
In [7]: sns.heatmap(train_data.corr(), annot=True)
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x2a5e04b80b8>
```

Figure 22 - Correlation Matrix

PRINCIPAL COMPONENT ANALYSIS:

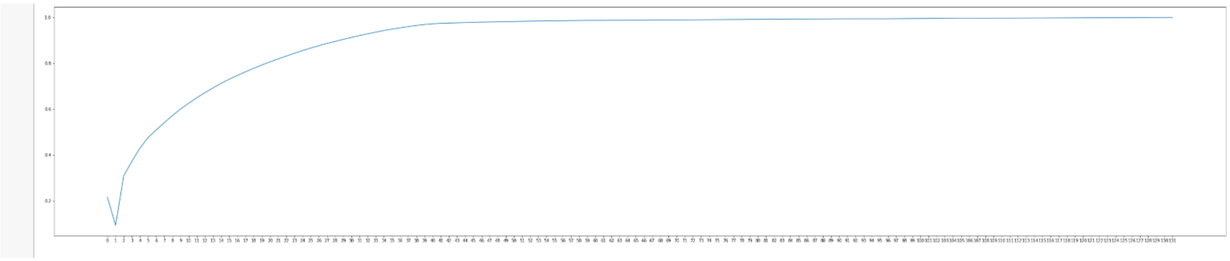
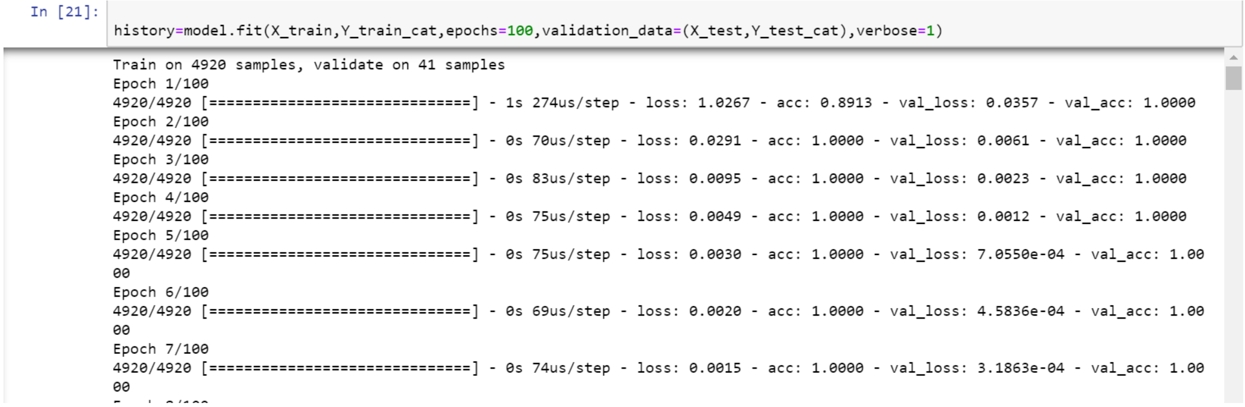


Figure 23 – Principal Component Analysis Graph

ARTIFICIAL NEURAL NETWORK:

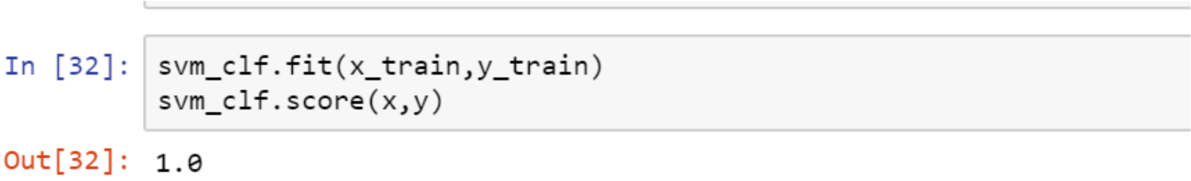


```
In [21]:
history=model.fit(X_train,Y_train_cat,epochs=100,validation_data=(X_test,Y_test_cat),verbose=1)

Train on 4920 samples, validate on 41 samples
Epoch 1/100
4920/4920 [==============================] - 1s 274us/step - loss: 1.0267 - acc: 0.8913 - val_loss: 0.0357 - val_acc: 1.0000
Epoch 2/100
4920/4920 [==============================] - 0s 70us/step - loss: 0.0291 - acc: 1.0000 - val_loss: 0.0061 - val_acc: 1.0000
Epoch 3/100
4920/4920 [==============================] - 0s 83us/step - loss: 0.0095 - acc: 1.0000 - val_loss: 0.0023 - val_acc: 1.0000
Epoch 4/100
4920/4920 [==============================] - 0s 75us/step - loss: 0.0049 - acc: 1.0000 - val_loss: 0.0012 - val_acc: 1.0000
Epoch 5/100
4920/4920 [==============================] - 0s 75us/step - loss: 0.0030 - acc: 1.0000 - val_loss: 7.0550e-04 - val_acc: 1.00
00
Epoch 6/100
4920/4920 [==============================] - 0s 69us/step - loss: 0.0020 - acc: 1.0000 - val_loss: 4.5836e-04 - val_acc: 1.00
00
Epoch 7/100
4920/4920 [==============================] - 0s 74us/step - loss: 0.0015 - acc: 1.0000 - val_loss: 3.1863e-04 - val_acc: 1.00
00
```

Figure 24 – Artificial neural Network Acuuracy

SUPPORT VECTOR MACHINE:

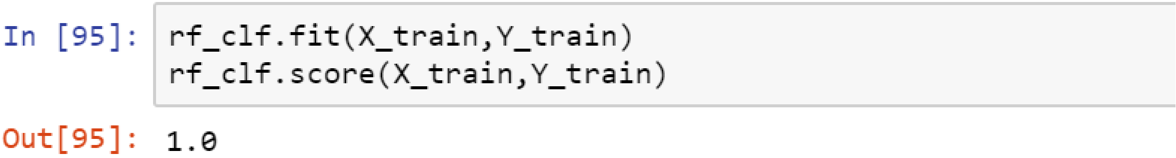


```
In [32]: svm_clf.fit(x_train,y_train)
         svm_clf.score(x,y)

Out[32]: 1.0
```

Figure 25 – Support Vector Machine Accuracy

RANDOM FOREST:



```
In [95]: rf_clf.fit(X_train,Y_train)
         rf_clf.score(X_train,Y_train)

Out[95]: 1.0
```

Figure 26 – Random Forest Accuracy

7. RESULT & DISCUSSION

COMPARISON GRAPH:

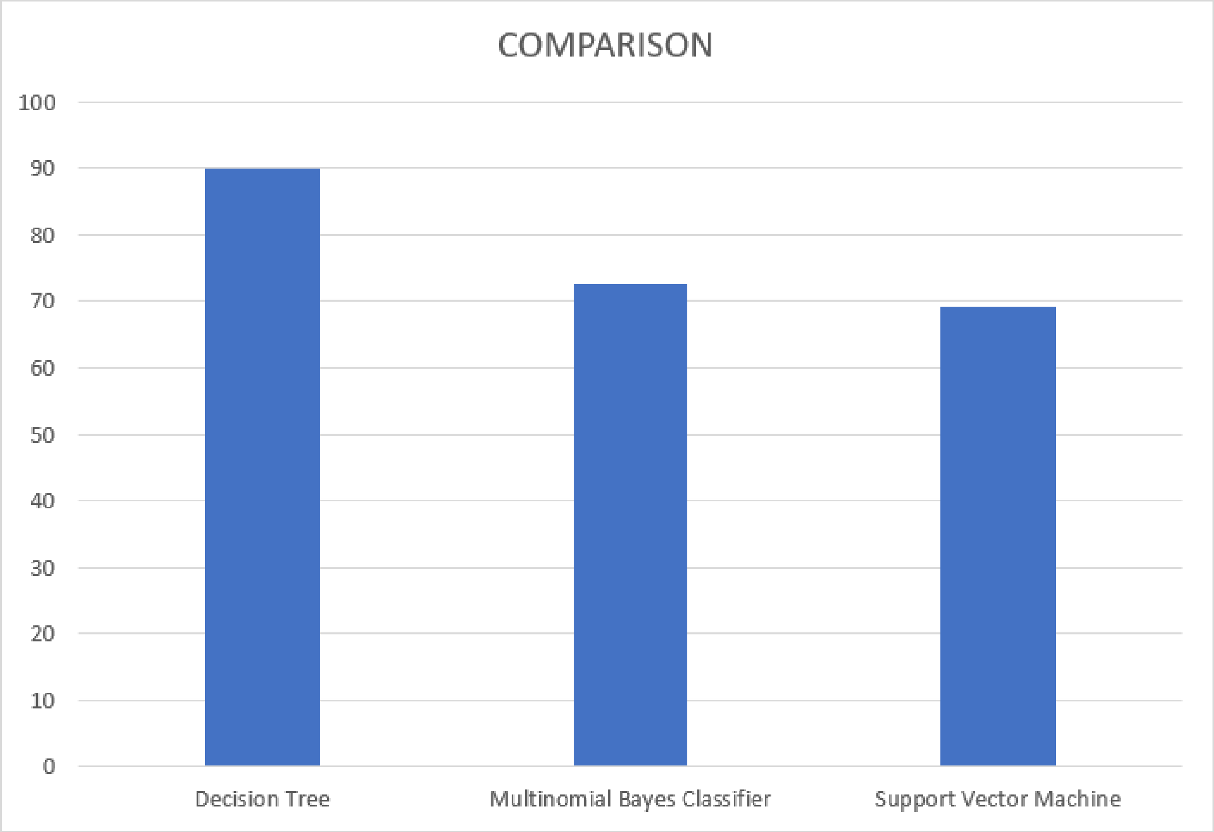


Figure 27 – Comparison Graph

For unstructured data, decision tree performs the best of all the three algorithms used with an accuracy of 89.93%. As the structured data is linear data, all the models perform the same. In the real life scenario, unstructured data is used which is why data cleaning has been performed on the data and used for model construction even though we already have structured dataset. The web application made in this project is useful for the patients to get an early idea of what the disease is based on their symptoms before reaching the medical facility and search for the doctors of their required specialization and the doctors can increase their exposure by registering here.

8.SUMMARY

Human face lots of problems related to the chronic disease. The main reason behind increase the chronic disease such as improper living habits, insufficient physical exercise, unhealthy diet, and irregular sleeping. 80% of people in the United States, spent more amount on the diagnosis of chronic disease. People give more aid for accurate prediction of disease. Due to preliminary disease prediction, it can reduce the risk of disease and patient gets diagnosed as early as possible.

In this project, a machine learning disease prediction system is proposed using structured and unstructured data from hospital for effective prediction of diseases. Existing work is not focused on both data types in the area of healthcare. Compared to the prediction algorithms used , the decision tree algorithm accuracy reaches 89.93%. A web application is also made based on the accurate model using flask for the patients to be able to predict the disease with most accuracy and doctors can also register for their increased exposure so that patients can search for them and contact them.

The future scope of this project would be to develop a mobile application based on this prediction system and this can also be made for one specific disease like heart disease prediction with the appropriate datasets. Dataset may be updated with many more diseases from hospital research data.

9.REFERENCES

[1] Sireesha Kilaru, D. Anupama, “Predictive Analysis of Diseases Using Machine Learning and Big Data”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 8, Issue 5, May 2019

[2] B. Qian, X. Wang, N. Cao, H. Li, and Y.-G. Jiang, “A relative similarity based method for interactive patient risk prediction,” Springer Data Mining Knowl. Discovery, vol. 29, no. 4, pp. 1070–1093, 2015.

[3] IM. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, “ Wearable 2.0: Enable human-cloud integration in next generation healthcare system,” IEEE Commun. , vol. 55, no. 1, pp. 54–61, Jan. 2017.

[4] Mayedaarshi, Dr.Rekhapatil ,“A Machine Learning Approach for Prediction of Diseases Using Unstructured Datasets”, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 6, Issue 6, June 2018.

[5] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang,“Disease prediction by machine learning over big data from healthcare communities”, IEEE Access, vol. 5, no. 1, pp. 8869–8879, 2017.

[6] Disease and symptoms Dataset –www.github.com.

[7] Ajinkya Kunjir, Harshal Sawant, Nuzhat F.Shaikh, “Data Mining and Visualization for prediction of Multiple Diseases in Healthcare,” in IEEE big data analytics and computational intelligence, Oct 2017 pp.2325.

[8] Shanthi Mendis, Pekka Puska, Bo Norrving, World Health Organization (2011), Global Atlas on Cardiovascular Disease Prevention and Control, PP. 3– 18. ISBN 978-92-4-156437-3.

[9] Amin, S.U.; Agarwal, K.; Beg, R., “Genetic neural network based data mining in prediction of heart disease using risk factors”, IEEE Conference on Information & Communication Technologies (ICT), vol., no.,pp.1227-31,11- 12 April 2013.

[10] Sayali Ambekar and Dr.Rashmi Phalnikar, “Disease Prediction By Using Machine Learning”, International Journal of Computer Engineering and Applications, Volume XII, Special Issue, May 18

[11] Prabhu. T, Darshana. J, Dharani Kumar. M, Hansaa Nazreen. M, “Health Risk Prediction by Machine Learning over Data Analytics”, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 02 Feb 2019.

[12] Harini D K1, Natesh M, “Prediction Of Probability Of Disease Based On Symptoms Using Machine Learning Algorithm”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 05 May-2018.