DISEASE PREDICTION BASED ON SYMPTOMS

A project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

submitted by

K SAI HARI PRIYA (18A31A0511)

K HARSHITHA SANTHOSHINI (19A35A0501)

U HEMA KUMAR (18A31A0557)

L PRAJWAL KUMAR (18A31A0542)

Under the Guidance of

Mrs.K.L.VIVEKA

Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRAGATI ENGINEERING COLLEGE
(AUTONOMOUS)

2020-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PRAGATI ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE & Permanently Affiliated to JNTUK & Accredited by NAAC)

1-378, ADB Road, Surampalem, E.G.Dist., A.P, Pin-533437.

CERTIFICATE

This is to certify that the report entitled **"DISEASE PREDICTION BASED ON SYMPTOMS"** *that is being submitted by* ***K S HariPriya, K H Santhoshini, U Hema Kumar, L Prajwal Kumar of III Year II Semester bearing Roll Numbers (18A31A0501 ,19A35A0501 ,18A31A0557 ,18A31A0542)****, in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering, Pragati Engineering College is a record of bonafide work carried out by them.*

Supervisor

Mrs.K.L.VIVEKA
Assistant Professor, CSE

Head of the Department

Dr.M.RadhikaMani
Professor & Head of Dept.of.CSE

ACKNOWLEDGMENT

It gives us an immense pleasure to express deep sense of gratitude, **Mrs.K.L.Viveka**, Assistant Professor, Department of Computer Science & Engineering because of her wholehearted and valuable guidance throughout the report. Without her sustained and sincere effort, this report would not have taken this shape. She encouraged and helped us to overcome various difficulties that we have faced at various stages of our report. We would like to sincierly thank **Dr. M Radhika Mani**, for providing all the necessary facilities that led to the successful completion of our report. We would like to take this opportunity to thank our beloved Principal and Vice-principal **Dr.S.Sambhu Prasad, Dr.K.Satyanarayana** for providing a great support to us in completing our project and for giving us the opportunity of doing the mini project report.

We would like to thank all the faculty members of the Department of Computer Science & Engineering for their direct or indirect support for helping us in completion of this report.

Finally, we would like to thank all our friends and family members for this continuous help and encouragement.

K S Hari Priya(18A31A0511)
K H Santhoshini(19A35A0501)
U Hema Kumar(18A31A0557)
L Prajwal Kumar(18A31A0542)

ABSTRACT

Disease Prediction using Machine Learning is a system which predicts the disease based on the symptoms he/she enter the system and provides the accurate results based on that information. If the patient is not much serious and the user just wants to know the type of disease, he/she has been through. It provides a way to find out the disease using this prediction. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. This Disease Prediction Using Machine Learning is completely done with the help of Machine Learning and Python Programming language with Tkinter Interface for it and using the dataset that is available previously by the hospitals using that we will predict the disease. The System Predicts using different machine learning algorithms such as RandomForest, DecisionTree, NaiveBayes. Hence the output is accurate.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Disease Prediction

Disease Prediction using Machine Learning is a system which predicts the disease based on the symptoms he/she enter into the system and provides the accurate results based on that information. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. This Disease Prediction UML is previously done by many other organizations, but our intention is to make it different and beneficial for the users who are using this system. This Disease Prediction Using Machine Learning is completely done with the help of Machine Learning and Python Programming language with Tkinter Interface for it and using the dataset that is available previously by the hospitals using that we will predict the disease. Doctors may sometimes fail to take accurate decisions while diagnosing the disease of a patient, therefore disease prediction systems which use machine learning algorithms assist in such cases to get accurate results. According to research there are 40% peoples who ignores about general disease which leads to harmful disease later. The main reason of ignorance is laziness to consult a doctor and time concern the peoples have involved themselves so much that they have no time to take an appointment and consult the doctor which later results into fatal disease. According to research there are 70% peoples in India suffers from general disease and 25% of peoples face death due to early ignorance. The main motive to develop this project is that a user can sit at their convenient place and have a check-up of their health. The UI is designed in such a simple way that everyone can easily operate on it and can have a check-up.

1.2 Problem Definition

Now a days in Health Industry there are various problems related to machines or devices which will give wrong or unaccepted results, so to avoid those results and get the correct and desired results we are building a program or project which will give the accurate predictions based on information provided by the user and also based on the datasets that are available in that machine. The health is rich in information yet and knowledge poor and this industry is very vast industry which has lot of work to be done. So, with the help of all machine learning algorithms, techniques and methodologies we have done this project which will help the peoples who are in the need. So the problem here is that many people goes to hospitals or clinic to know how is their health and how much they are improving in the given days, but they have to travel to get to know there answers and sometimes the patients may or may not get the results based on various factors such as doctor might be on leave or some whether problem so he might not have come to the hospital and many more reasons will be there so to avoid all those reasons and confusion we are making a project which will help all those person's and all the patients who are in need to know the condition of their health, and at sometimes if the person has been observing few symptoms and he/she is not sure about the disease he/she is encountered with so this will lead to various diseases in future. So, to avoid that and get to know the disease in early stages of the symptoms this disease prediction will help a lot to the various people's ranging from children to teenagers to adults and also the senior citizens.

1.3 Project Features

The features of Disease Prediction Using Machine Learning are as follows.

- This Project will predict the diseases of the patients based on the symptoms and using the datasets.
- This is done based on the previous datasets of the hospitals so after comparing it can provide up to 80% of accurate results, and the project is still developing further to get the 100% accurate results.
- With the help of Disease prediction, it can predict the disease of the patient and can solve various problems and prevents from various aspects.

The disease is predicted using the algorithms and the user has to enter the symptoms from the given drop-down menu, in order to get correct accuracy, the user has to enter all the symptoms.

CHAPTER 2

LITERATURE SURVEY

2.1 MACHINE LEARNING

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns.

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervisedlearning.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning tasks Machine learning tasks are typically classified into several broad categories:

Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Semi-supervised learning: The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Reinforcement learning: Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

2.1.1 FEATURES OF MACHINE LEARNING

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Machine leaning models involves machines learning from data without the help of humans or any kind of human intervention.
- Machine Learning is the science of making the computers learn and act like humans by feeding data and information without being explicitly program
- Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return it gives us the program which provides solution to the various problems. Below is the figure.

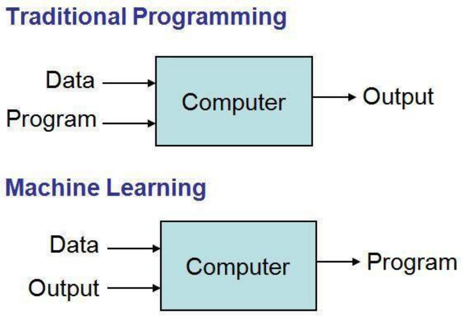


Fig 2.1.1 Traditional Programming vs Machine Learning

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There are Many Algorithms in Machine Learning through which will provide us the exact solution in predicting the disease of the patients.

2.2 EXISTING STSTEM

Prediction using traditional methods and models involves various risk factors and it consists of various measures of algorithms such as datasets, programs and much more to add on. High-risk and Low-risk patient classification is done on the basis of the tests that are done in group. But these models are only valuable in clinical situations and not in big industry sector. So, to include the disease predictions in various health related industries, we have used the concepts of machine learning and supervised learning methods to build the predictions system.

After doing the research and comparison of all the algorithms and theorems of machine learning we have come to conclusion that all those algorithms such as Decision Tree, Naïve Bayes, Regression and Random Forest Algorithm all are important in building a disease prediction system which predicts the disease of the patients from which he/she is suffering from and to do this we have used some performance measures like ROC, KAPPA Statistics, RMSE, MEA and various other tools. After using various techniques such as neural networks to make predictions of the diseases and after doing that we come to conclusion that it can predicts up to 90% accuracy rate after doing the experimentation and verifying the results. The information of patient statistics, results, disease history in recorded in EHR, which enables to identify the potential data centric solution, which reduces the cost of medical case studies. Existing system can predict the disease but not the sub type of the disease and it fails to predict the condition of the people, the predictions of disease have been indefinite and non-specific.

2.3 PROPOSED STSTEM

The proposed system of disease prediction using machine learning is that we have used many techniques and algorithms and all other various tools to build a system which predicts the disease of the patient using the symptoms and by taking those symptoms we are comparing with the system's dataset that is previously available. By taking those datasets and comparing with the patient's disease we will predict the accurate percentage disease of the patient. The dataset and symptoms go to the prediction model of the system where the data is pre-processed for the future references and then the feature selection is done by the user where he will enter the various symptoms. Then the classification of those data is done with the help of various algorithms and techniques such as Decision Tree, Naïve Bayes, Random Forest and etc. Then the data goes in the recommendation model, there it shows the risk analysis that is involved in the system and it also provides the probability estimation of the system such that it shows the various probability like how the system behaves when there are n number of predictions are done and it also does the recommendations for the patients from their final result and also from their symptoms like it can show what to use and what not to use from the given datasets and the final results. Here we have combined the overall structure and unstructured form of data for the overall risk analysis that is required for doing the prediction of the disease. Using the structured analysis, we can identify the chronic types of disease in a particular region and particular community. In unstructured analysis we select the features automatically with the help of algorithms and techniques. This system takes symptoms from the user and predicts the disease accordingly based on the symptoms that it takes and also from the previous datasets, it also helps in continuous evaluation of viral diseases, heart rate, blood pressure, sugar level and much more which is in the system and along with other external symptoms its predicts the appropriate and accurate disease.

2.4 SOFTWARE DESCRIPTION

2.4.1 PYTHON

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's developers strive to avoid premature optimization, and reject patches to non- critical parts of CPython that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter. An important goal of Python's developers is keeping it fun to use. Python's design offers some support for functional programming in the Lisp tradition. It has filter, map, and reduce functions, list comprehensions, dictionaries, sets, and generator expressions.

2.4.2 BENEFITS OF PYTHON

- Presence of Third-Party Modules
- Extensive Support Libraries
- Open Source and Community Development
- Learning Ease and Support Available
- User-friendly Data Structures
- Productivity and Speed
- Highly Extensible and Easily Readable Language.

2.4.3 LIBRAEIES USED

The following are the libraries used in this project.

2.4.3.1 <u>tkinter:</u>

It’s a standard GUI library of python. Python, when combined with tkinter provides a fast and easy way to create GUI. It provides a powerful object-oriented tool for creating GUI.

It provides various widgets to create GUI some of the prominent ones being:

· Button
· Canvas
· Label
· Entry
· Check Button
· List box
· Message
· Text
· Messagebox

Some of these were used in this project to create our GUI, namely messagebox, button, label, Option Menu, text and title. Using tkinter we were able to create an interactive GUI for our model.

2.4.3.2. Numpy:

Numpy is the core library of scientific computing in python. It provides powerful tools to deal with various multi-dimensional arrays in python. It is a general purpose array processing package.

Numpy's main purpose is to deal with multidimensional homogeneous arrays. It has tools ranging from array creation to its handling. It makes it easier to create a n dimensional array just by using np.zeros() or handle its contents using various other methods such as replace, arrange, random, save, load it also helps I array processing using methods like sum, mean, std, max, min, all, etc.

Arrays created with numpy also behave differently then arrays created normally when they are operated upon using operators such as +,-,*,/.

All the above qualities and services offered by numpy array make it highly suitable for our purpose of handling data. Data manipulation occurring in arrays while performing various operations needs to give the desired results while predicting outputs requires such high operational capabilities.

2.4.3.3. Pandas :

It is the most popular python library used for data analysis. It provides highly optimized performance with back-end source code purely written in C or python.

Data in python can be analyzed with 2 ways

· Series

· Dataframes

Series is one dimensional array defined in pandas used to store any data type.

Dataframes are two-dimensional data structures used in python to store data consisting of rows and columns.

Pandas dataframe is used extensively in this project to use datasets required for training and testing the algorithms. Dataframes make it easier to work with attributes and results. Several of

Its inbuilt functions such as replace were used in our project for data manipulation and preprocessing.

2.4.3.4. sklearn:

Sklearn is an open source python library which implements a huge range of machine learning, pre-processing, cross-validation and visualization algorithms. It features various simple and efficient tools for data mining and data processing. It features various classification, regression and clustering algorithms such as support vector machine, random forest classifier, decision tree, gaussian naïve-Bayes, KNN to name a few.

In this project we have used sklearn to take advantage of inbuilt classification algorithms like decision tree, random forest classifier and naive Bayes. We have also used inbuilt cross validation and visualization features such as classification report, confusion matrix and accuracy score.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 HARDWARE REQUIREMENTS

❖ System	: Pentium 4, Intel Core i3, i5, i7 and 2 GHz Minimum
❖ RAM	: 512Mb or above
❖ Hard Disk	: 10 GB or above
❖ Input Device	: Keyboard and Mouse
❖ Output Device	: Monitor or PC

3.2 SOFTWARE REQUIREMENTS

❖ Operating System	: Windows 7, 10 or Higher Versions
❖ Platform	: Jupiter Notebook
❖ Front End	: Python Tkinter
❖ Back End	: Python and Files
❖ Programming Lang	: Python

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

- We have designed our system in such a way that whenever user entered our system the user must enter the name. Otherwise, it will show message box saying please enter your name.
- After that user must select symptoms from the drop-down menu. User must select atleast two symptoms in order to predict the disease. Otherwise, a message warning will appear saying at least select two symptoms.
- After selecting symptoms, there will be three buttons named prediction1, prediction2, prediction3.
- We can click any of the button the result of the corresponding algorithm will be displayed. We can click all the three button to check result of all algorithms

4.2 DATA FLOW DIAGRAM

The dataflow diagram of the project disease prediction using machine learning consist of all the various aspects a normal flow diagram requires. This dataflow diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information.

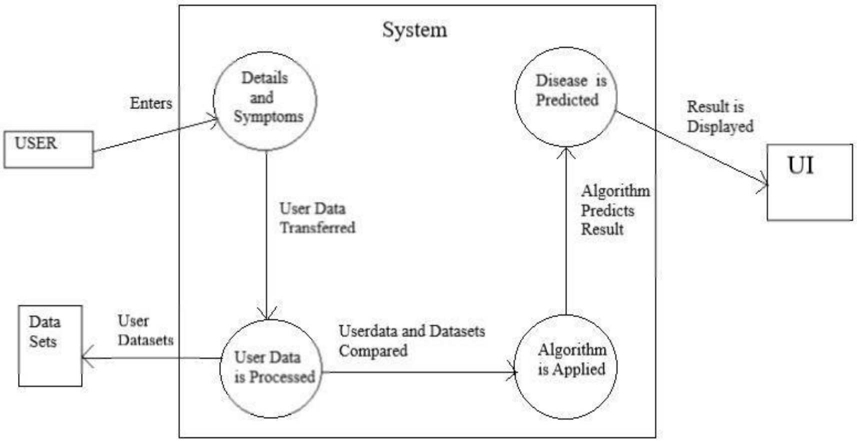


Fig 4.3 Data Flow Diagram

CHAPTER 5

IMPLEMENTATION

5.1 OVERVIEW

The user needs to the name and needs to select the symptoms from given drop-down menu, for more accurate result the user needs to enter all the given symptoms, then the system will provide the accurate result. This prediction is basically done with the help of 3 algorithms of machine learning such as Decision Tree, Random Forest and Naïve Bayes. When user enter all the symptoms then he needs to press the buttons of respective algorithm, for example there are 3 buttons for 3 algorithms, if user enters all symptoms and presses only Random forest’s button then the result will be provided only calculating using that algorithm, like this we have used 3 algorithms to provide clearer picture of the results and user needs to be satisfied with his predicted result.

In this project 3 different algorithms were used -

- Decision Tree Algorithm
- Random Forest Algorithm
- Naïve Bayes Algorithm
- 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.

The working and basic explanation of those 3 algorithms Random Forest, Decision Tree and Naïve Bayes is given below.

5.2 DECISION TREE ALGORITHM

Decision tree induction is the learning of decision trees from class-labelled training tuples. A decision tree is a flowchart-like tree structure,

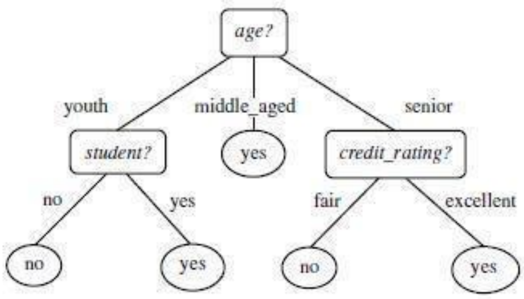
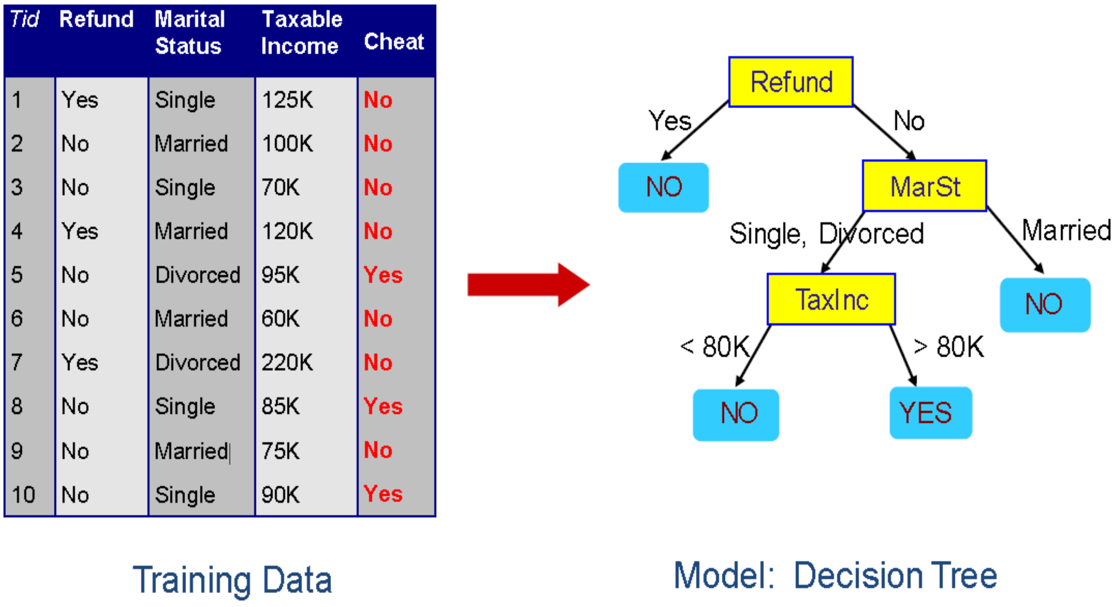


Fig 5.2.1 Decision Tree problem

- Decision tree induction is a non-parametric approach for building classification models.
- Finding an optimal decision tree is an NP-complete problem
- Techniques developed for constructing decision trees are computationally inexpensive, making it possible to construct models even when the training set size is verylarge.
- Decision trees, especially smaller-sized trees, are relatively easy to interpret.
- Decision tree provide an expressive representation for learning discrete- valued functions.
- Decision tree algorithms are quite robust to the presence of noise, especially when methods for avoidingoverfitting.



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married|	75K	No
10	No	Single	90K	Yes

Training Data

Model: Decision Tree

Fig 5.2.2 Decision Tree Example

- The presence of redundant attributes does not adversely affect the accuracy of decision tree.
- The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore I appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data.
- Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans.
- The learning and classification steps of decision tree induction are simple and fast.
- In general, decision tree classifiers have good accuracy.
- Decision tree induction algorithm shave been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.

5.3 RANDOM FOREST ALGORITHM

- It is an ensemble classifier using many decision trees models; it can be used for regression as well as classification.

- Accuracy and variable importance information can be provided with the results.

- A random forest is the classifier consisting of a collection of tree structured classifiers k, where the k is independently, identically distributed random trees and each random tree consist of the unit of vote for classification of input.

- Random forest uses the Gini index for the classification and determining the final class in each tree.

- The final class of each tree is aggregated and voted by the weighted values to construct the final classifier.

- The working of random forest is, A random seed is chosen which pulls out at a random, a collection of samples from the training datasets while maintaining the class distribution.

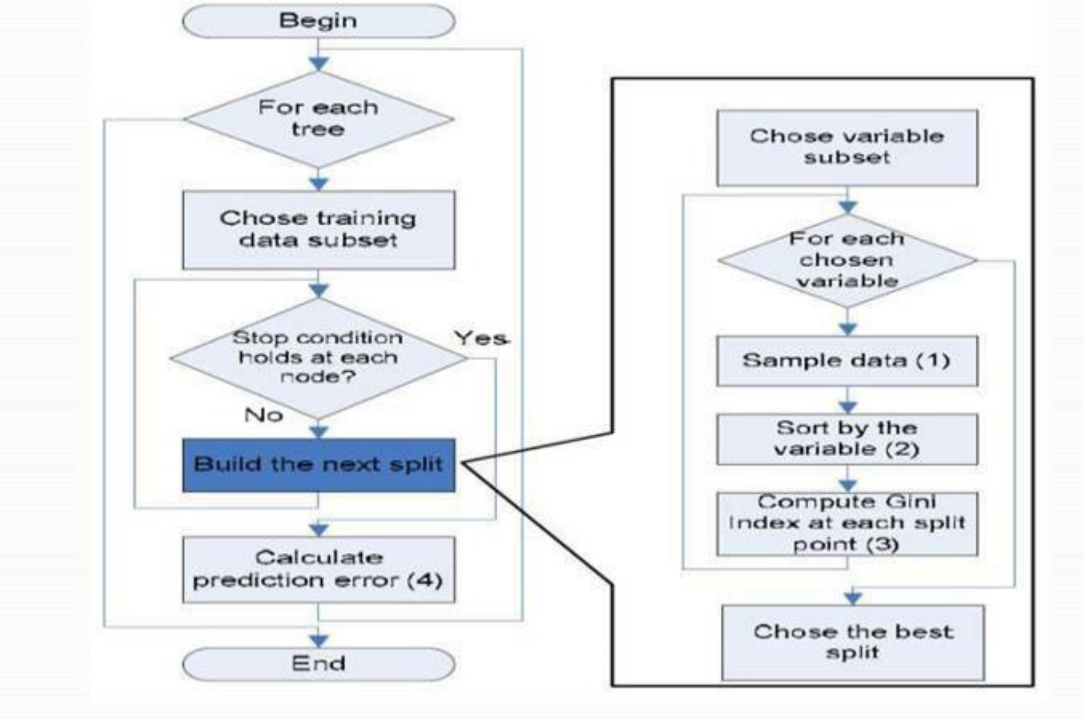
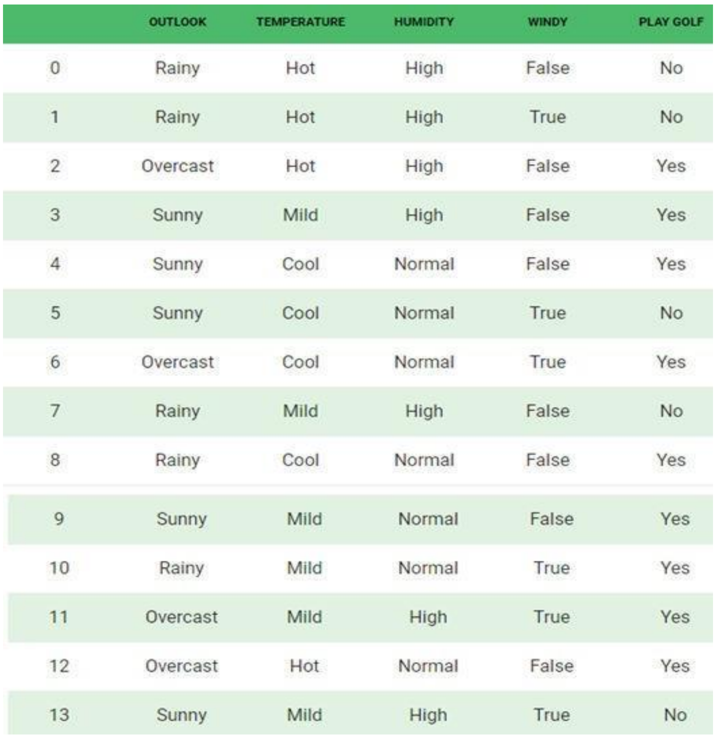


Fig 5.3 Random Forest Flowchart

5.4 NAIVE BAYES ALGORITHM

- It is used to predict the categorical class labels.
- It is a two-step process Model Construction and Model Usage
- It classifies the class data based on the training set and the values in a classifying attribute and uses it in classifying new data.
- This Bayes theorem is named after Thomas Bayes and it is statistical method for classification and supervised learning method.
- It can solve both categorical and continuous values attributes.
- Bayes theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes theorem is stated mathematically as the following equation.
- $P(A/B) = P(B|A)P(A)/P(B)$

Below is the example how this algorithm/theorem works with the dataset.



	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Fig 5.4 Naive Bayes Dataset

- The given dataset is divided into two parts namely feature matrix and response vector.

- Feature matrix contains all the vectors means rows of the dataset in which each vector consists of the values of dependent features. In the above dataset features are outlook, temperature, humidity andwindy.

- Response vector consist of values of class variables for each row of feature matrix. In the above dataset the class variable name is play golf.

- The fundamental naïve based assumption is that each feature makes an independent and equal contribution to the outcome.

CHAPTER 6

SOURCE CODE

```
#importing Libraries to create GUI
from tkinter import *

#Importing Libraries to perform calculations
import numpy as np
import pandas as pd

#List of the symptoms is listed here in list l1.
l1=['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_bow
el_movements','pain_in_anal_region','bloody_stool','irritation_in_anus','neck_pain','dizziness','cramps','bruisi
ng','obesity','swollen_legs','swollen_blood_vessels','puffy_face_and_eyes','enlarged_thyroid','brittle_nails','s
wollen_extremeties','excessive_hunger','extra_marital_contacts','drying_and_tingling_lips','slurred_speech','k
nee_pain','hip_joint_pain','muscle_weakness','stiff_neck','swelling_joints','movement_stiffness','spinning_mo
vements','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)','d
epression','irritability','muscle_pain','altered_sensorium','red_spots_over_body','belly_pain','abnormal_menstr
uation','dischromic_patches','watering_from_eyes','increased_appetite','polyuria','family_history','mucoid_sp
utum','rusty_sputum','lack_of_concentration','visual_disturbances','receiving_blood_transfusion','receiving_u
nsterile_injections','coma','stomach_bleeding','distention_of_abdomen','history_of_alcohol_consumption','flu
id_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']

#List of Diseases is listed in list disease.
disease=['Fungalinfection','Allergy','GERD','Chroniccholestasis','DrugReaction','Peptic ulcer disease','AIDS',
'Diabetes', 'Gastroenteritis','Bronchial Asthma’,‘Hypertension',
'Migraine','Cervicalspondylosis',‘Paralysis(brainhemorrhage)','Jaundice','Malaria','Chicken
pox','Dengue','Typhoid','hepatitis A','Hepatitis ','HepatitisC','HepatitisD','Hepatitis
E','Alcoholichepatitis','Tuberculosis','CommonCold','Pneumonia','Dimorphic
hemmorhoids(piles)','Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteo
arthristis','Arthritis','(vertigo) Paroymsal  Positional Vertigo’, 'Acne','Urinary tract
infection','Psoriasis','Impetigo']

l2=[]
for i in range(0,len(l1)):
l2.append(0)
print(l2)

#Reading the training .csv file
df=pd.read_csv("Training.csv")

#Replace the values in the imported file by pandas inbuilt function replace in pandas.
```

```
df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug



Reaction':4'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial
Asthma':9,'Hypertension ':10,'Migraine':11,
'Cervical spondylosis':12, 'Paralysis (brain hemorrhage)':13, 'Jaundice':14,
'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatitis':24,'Tuberculosis':
25,    'Common Cold':26,'Pneumonia':27,
'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Varicose
veins':30,'Hypothyroidism':31,'Hyperthyroidism':32,'Hypoglycemia':33,
'Osteoarthristis':34,'Arthritis':35,    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract
infection':38,'Psoriasis':39,
'Impetigo':40}},inplace=True)

#printing the top 5 rows of the training dataset
df.head()

X= df[l1]
y = df[["prognosis"]]

print(X)
print(y)

#Reading the  testing.csv file
tr=pd.read_csv("testing.csv")

#Using inbuilt function replace in pandas for replacing the values
tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug
Reaction':4'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial
Asthma':9,'Hypertension ':10,'Migraine':11,
'Cervical spondylosis':12, 'Paralysis (brain hemorrhage)':13, 'Jaundice':14,
'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic
hepatitis':24,'Tuberculosis':25,'Common Cold':26,'Pneumonia':27,
'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Varicose
veins':30,'Hypothyroidism':31,'Hyperthyroidism':32,'Hypoglycemia':33,
'Osteoarthristis':34,'Arthritis':35,    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract
infection':38,'Psoriasis':39,
'Impetigo':40}},inplace=True)

#printing the top 5 rows of the testing data
tr.head()

X_test= tr[l1]
y_test = tr[["prognosis"]]

print(X_test)
print(y_test)

root = Tk()
pred1=StringVar()

def DecisionTree():
    if len(NameEn.get()) == 0:
        pred1.set(" ")



comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two symptom")
        if sym:
            root.mainloop()
      else:
        print(NameEn.get())
        from sklearn import tree

        clf3 = tree.DecisionTreeClassifier()
        clf3 = clf3.fit(X,y)

        from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
        y_pred=clf3.predict(X_test)
        print("-------------------Decision Tree-------------------")
        print("Accuracy is:")
        print(accuracy_score(y_test, y_pred))
        #print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix:")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = clf3.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
```

```
if(predicted == a):
                h='yes'
                break


        if (h=='yes'):
            pred1.set(" ")
            pred1.set(disease[a])
        else:
            pred1.set(" ")
            pred1.set("Not Found")

pred2=StringVar()
def randomforest():
    if len(NameEn.get()) == 0:



pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.ensemble import RandomForestClassifier
        clf4 = RandomForestClassifier(n_estimators=100)
        clf4 = clf4.fit(X,np.ravel(y))

        # calculating accuracy
        from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
        y_pred=clf4.predict(X_test)
        print("-------------------Random Forest-------------------")
        print("Accuracy is:")
        print(accuracy_score(y_test, y_pred))
        #print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix:")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1
```

```
inputtest = [l2]
        predict = clf4.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
        if (h=='yes'):
            pred2.set(" ")
            pred2.set(disease[a])
        else:
            pred2.set(" ")
            pred2.set("Not Found")



pred3=StringVar()
def NaiveBayes():
    if len(NameEn.get()) == 0:
        pred1.set(" ")




comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb=gnb.fit(X,np.ravel(y))

        from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
        y_pred=gnb.predict(X_test)
        print("-------------------Naive Bayes-------------------")
        print("Accuracy is:")
        print(accuracy_score(y_test, y_pred))
        #print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix :")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
        for k in range(0,len(l1)):
```

```
for z in psymptoms:
            if(z==l1[k]):
               l2[k]=1

      inputtest = [l2]
      predict = gnb.predict(inputtest)
      predicted=predict[0]

      h='no'
      for a in range(0,len(disease)):
         if(predicted == a):
            h='yes'
            break
      if (h=='yes'):
         pred3.set(" ")
         pred3.set(disease[a])
      else:
         pred3.set(" ")
         pred3.set("Not Found")


#Tk class is used to create a root window
root.configure(background='Ivory')
root.title('Smart Disease Predictor System')
root.resizable(0,0)

#taking first input as symptom
Symptom1 = StringVar()



Symptom1.set("Select Here")

#taking second input as symptom
Symptom2 = StringVar()
Symptom2.set("Select Here")

#taking third input as symptom
Symptom3 = StringVar()
Symptom3.set("Select Here")

#taking fourth input as symptom
Symptom4 = StringVar()
Symptom4.set("Select Here")

#taking fifth input as symptom
Symptom5 = StringVar()
Symptom5.set("Select Here")
Name = StringVar()
```

```
#function to Reset the given inputs to initial position
prev_win=None
def Reset():
    global prev_win

    Symptom1.set("Select Here")
    Symptom2.set("Select Here")
    Symptom3.set("Select Here")
    Symptom4.set("Select Here")
    Symptom5.set("Select Here")

    NameEn.delete(first=0,last=100)

    pred1.set(" ")
    pred2.set(" ")
    pred3.set(" ")

    try:
        prev_win.destroy()
        prev_win=None
    except AttributeError:
        pass

#Exit button to come out of system
from tkinter import messagebox
def Exit():
    qExit=messagebox.askyesno("System","Do you want to exit the system")
    if qExit:
        root.destroy()
        exit()


#Headings for the GUI written at the top of GUI
w2 = Label(root, justify=LEFT, text="Disease Predictor using Machine Learning", fg="Brown",bg="white")



w2.config(font=("Times",30,"bold"))
w2.grid(row=1, column=0, columnspan=2, padx=100)


#Label for the name
NameLb = Label(root, text="Name of the Patient", fg="Black",bg="white")
NameLb.config(font=("Times",15,"bold italic"))
NameLb.grid(row=6, column=0, pady=15, sticky=W)

#Creating Labels for the symtoms
S1Lb = Label(root, text="Symptom 1", fg="Black",bg="ivory")
S1Lb.config(font=("Times",15,"bold italic"))
```

```
S1Lb.grid(row=7, column=0, pady=10, sticky=W)

S2Lb = Label(root, text="Symptom 2", fg="Black",bg="ivory")
S2Lb.config(font=("Times",15,"bold italic"))
S2Lb.grid(row=8, column=0, pady=10, sticky=W)

S3Lb = Label(root, text="Symptom 3", fg="Black",bg="ivory")
S3Lb.config(font=("Times",15,"bold italic"))
S3Lb.grid(row=9, column=0, pady=10, sticky=W)

S4Lb = Label(root, text="Symptom 4", fg="Black", bg="ivory")
S4Lb.config(font=("Times",15,"bold italic"))
S4Lb.grid(row=10, column=0, pady=10, sticky=W)

S5Lb = Label(root, text="Symptom 5", fg="Black", bg="ivory")
S5Lb.config(font=("Times",15,"bold italic"))
S5Lb.grid(row=11, column=0, pady=10, sticky=W)

#Labels for the different algorithms
lrLb = Label(root, text="DecisionTree", fg="Red", bg="ivory", width = 20)
lrLb.config(font=("Times",22,"bold "))
lrLb.grid(row=15, column=0, pady=10,sticky=W)

destreeLb = Label(root, text="RandomForest", fg="green", bg="ivory", width = 20)
destreeLb.config(font=("Times",22,"bold "))
destreeLb.grid(row=17, column=0, pady=10, sticky=W)

ranfLb = Label(root, text="NaiveBayes", fg="Blue", bg="ivory", width = 20)
ranfLb.config(font=("Times",22,"bold "))
ranfLb.grid(row=19, column=0, pady=10, sticky=W)

OPTIONS = sorted(l1)

#Taking name as input from user
NameEn = Entry(root, textvariable=Name)
NameEn.grid(row=6, column=1)

#Taking Symptoms as input from the dropdown from the user
S1 = OptionMenu(root, Symptom1,*OPTIONS)
S1.grid(row=7, column=1)




S2 = OptionMenu(root, Symptom2,*OPTIONS)
S2.grid(row=8, column=1)

S3 = OptionMenu(root, Symptom3,*OPTIONS)
S3.grid(row=9, column=1)
```

```
S4 = OptionMenu(root, Symptom4,*OPTIONS)
S4.grid(row=10, column=1)

S5 = OptionMenu(root, Symptom5,*OPTIONS)
S5.grid(row=11, column=1)

#Buttons for predicting the disease using different algorithms
dst = Button(root, text="Prediction 1", command=DecisionTree,bg="Red",fg="white")
dst.config(font=("Times",18,"bold "))
dst.grid(row=7, column=3,padx=10,pady=10)

rnf = Button(root, text="Prediction 2", command=randomforest,bg="light Green",fg="white")
rnf.config(font=("Times",18,"bold "))
rnf.grid(row=8, column=3,padx=10,pady=10)

lr = Button(root, text="Prediction 3", command=NaiveBayes,bg="Blue",fg="white")
lr.config(font=("Times",18,"bold "))
lr.grid(row=9, column=3,padx=10,pady=10)


rs = Button(root,text="Reset Inputs", command=Reset,fg="Black",width=15)
rs.config(font=("Times",15,"bold"))
rs.grid(row=15,column=3,padx=10)

ex = Button(root,text="Exit System", command=Exit,fg="black",width=15)
ex.config(font=("Times",15,"bold "))
ex.grid(row=17,column=3,padx=10)

#Showing the output of different algorithms
t1=Label(root,font=("Times",15,"bold italic"),text="Decision Tree",height=1
        ,width=40,fg="black",textvariable=pred1,relief="sunken").grid(row=15, column=1, padx=10)

t2=Label(root,font=("Times",15,"bold italic"),text="Random Forest",height=1
        ,width=40,fg="black",textvariable=pred2,relief="sunken").grid(row=17, column=1, padx=10)

t3=Label(root,font=("Times",15,"bold italic"),text="Naive Bayes",height=1
        ,width=40,fg="black",textvariable=pred3,relief="sunken").grid(row=19, column=1, padx=10)


#calling this function because the application is ready to run
root.mainloop()
```

CHAPTER 7

RESULTS

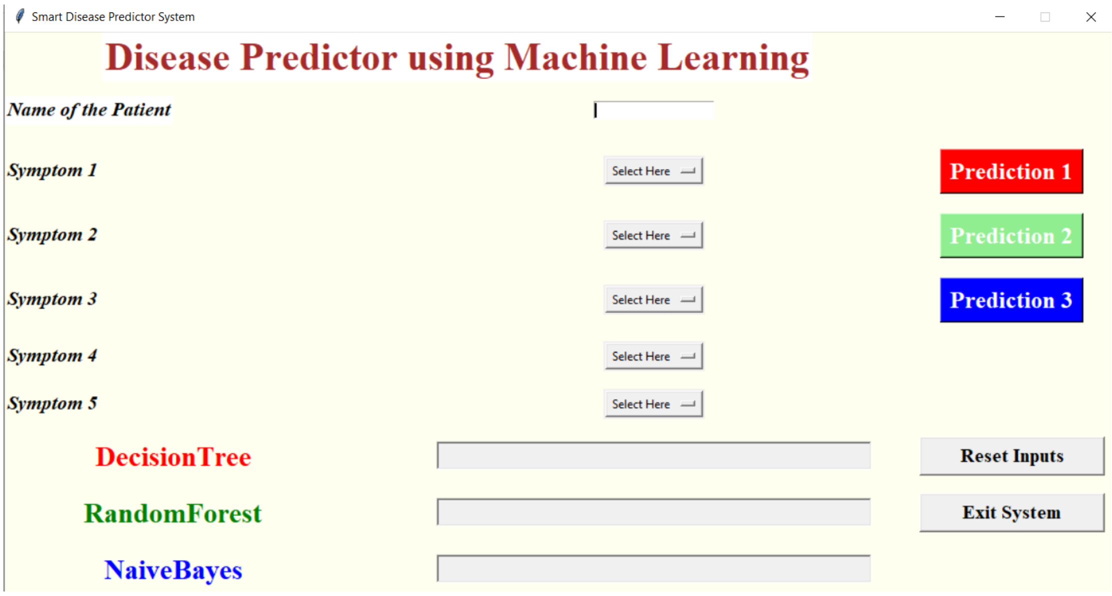
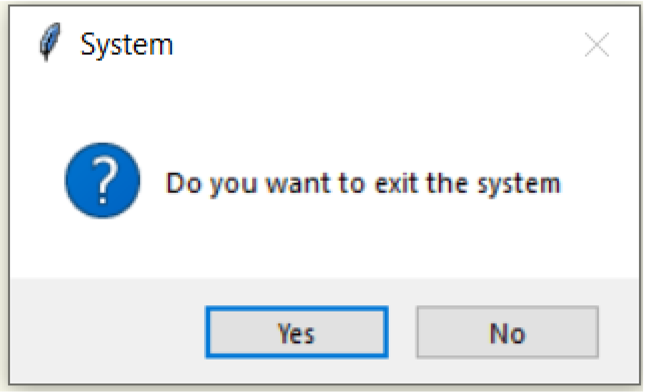


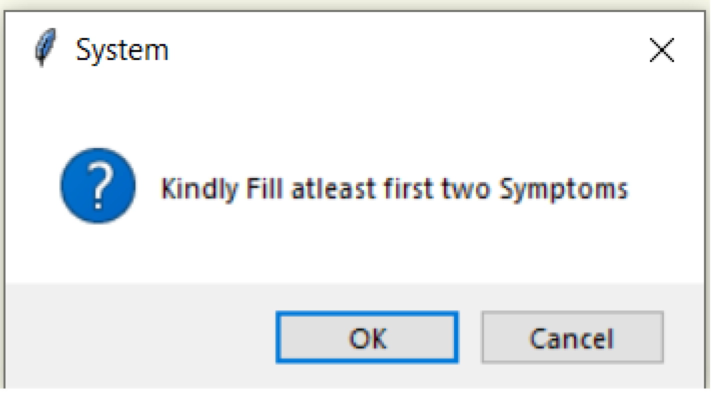
Fig 7.1 Main Page



System

Do you want to exit the system

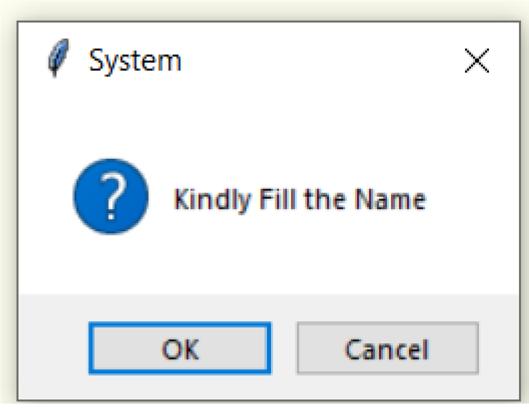
Fig 7.2 Message box1



System

Kindly Fill atleast first two Symptoms

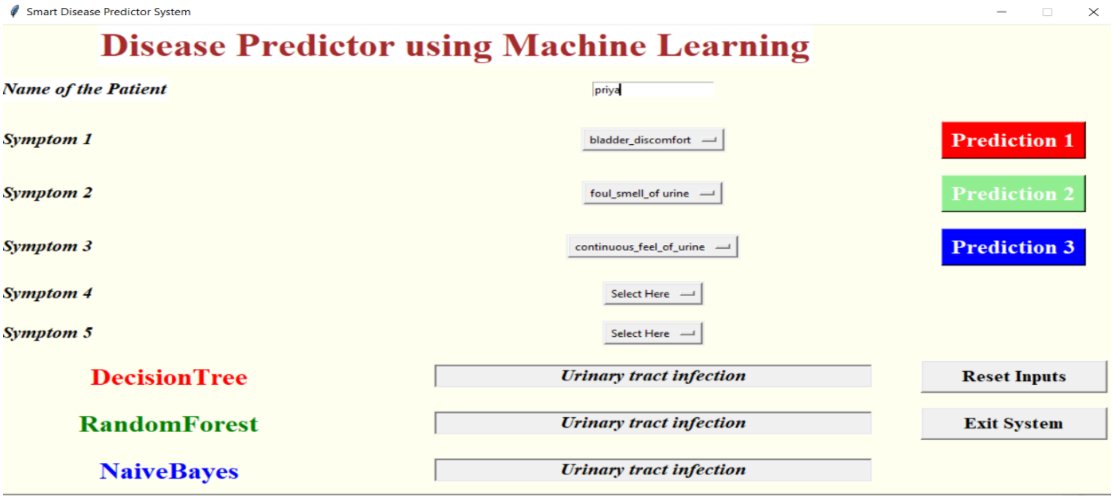
Fig 7.3 Message box2



System

Kindly Fill the Name

Fig 7.4 Message box3


Smart Disease Predictor System
Disease Predictor using Machine Learning
Prediction 1
Prediction 2
Prediction 3
Reset Inputs
Exit System

Name of the Patient: priya

Symptom 1: bladder_discomfort

Symptom 2: foul_smell_of urine

Symptom 3: continuous_feel_of_urine

Symptom 4: Select Here

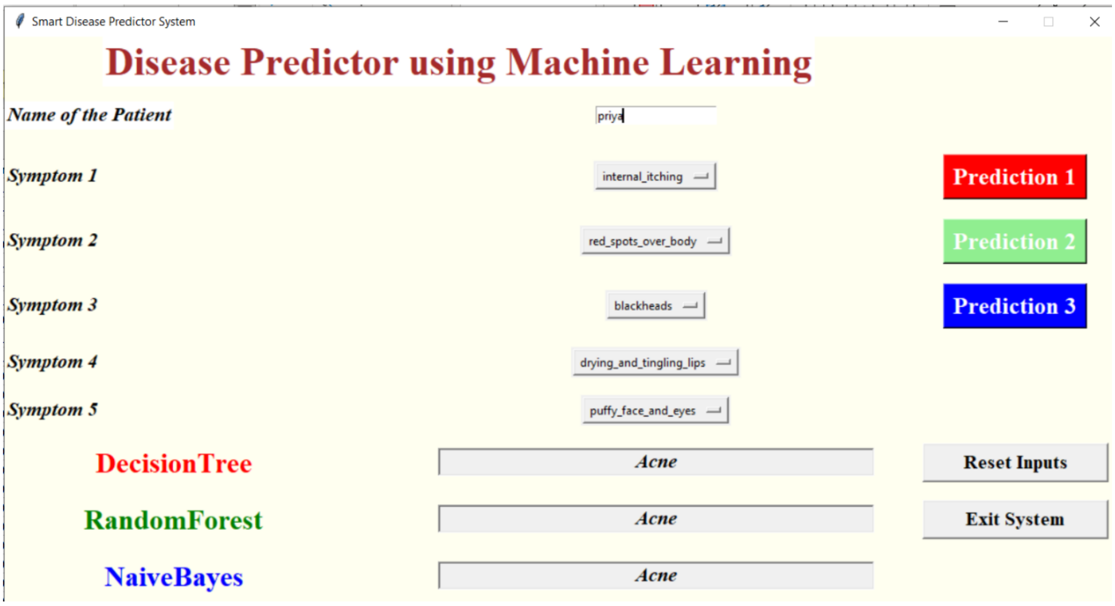
Symptom 5: Select Here

DecisionTree: Urinary tract infection

RandomForest: Urinary tract infection

NaiveBayes: Urinary tract infection

Fig 7.5 Predicted Result1 page


Smart Disease Predictor System
Disease Predictor using Machine Learning
Prediction 1
Prediction 2
Prediction 3
Reset Inputs
Exit System

Name of the Patient: priya

Symptom 1: internal_itching

Symptom 2: red_spots_over_body

Symptom 3: blackheads

Symptom 4: drying_and_tingling_lips

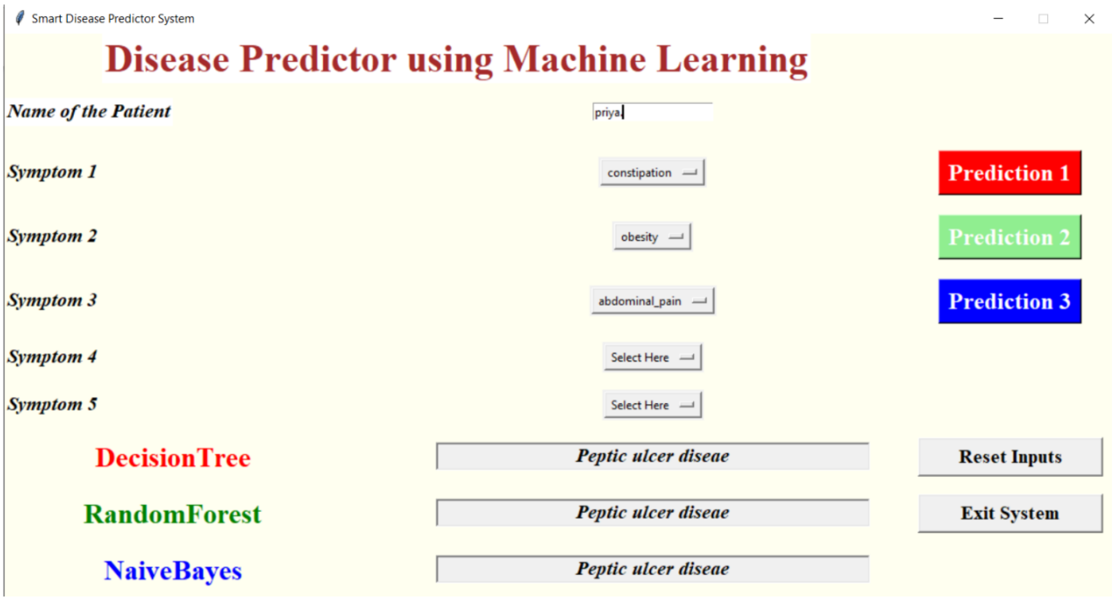
Symptom 5: puffy_face_and_eyes

DecisionTree: Acne

RandomForest: Acne

NaiveBayes: Acne

Fig 7.6 Predicted Result2 page



Smart Disease Predictor System

Disease Predictor using Machine Learning

Name of the Patient: priya

Symptom 1: constipation

Symptom 2: obesity

Symptom 3: abdominal_pain

Symptom 4: Select Here

Symptom 5: Select Here

DecisionTree: Peptic ulcer diseae

RandomForest: Peptic ulcer diseae

NaiveBayes: Peptic ulcer diseae

Fig 7.6 Predicted Result3 page

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

So, Finally I conclude by saying that, this project Disease prediction using machine learning is very much useful in everyone's day to day life and it is mainly more important for the healthcare sector, because they are the one that daily uses these systems to predict the diseases of the patients based on their general information and there symptoms that they are been through. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. If health industry adopts this project then the work of the doctors can be reduced and they can easily predict the disease of the patient. The Disease prediction is to provide prediction for the various and generally occurring diseases that when unchecked and sometimes ignored can turns into fatal disease and cause lot of problem to the patient and as well as their family members.

8.2 FUTURE ENHANCEMENT

- Facility for modifying user detail.
- Creating a database to store the details of patient.
- Develop a registration and login page.
- More interactive userinterface.
- Facilities for Backupcreation.
- Can be done as Web page.
- Can be done as Mobile Application.
- Add More Details and Latest Diseases

BIBLIOGRAPHY

1. Disease Prediction and Doctor Recommendation System by www.irjet.net
2. Disease Prediction Based on Prior Knowledge by www.hcup-us.ahrq.gov/nisoverview.jsp
3. GDPS - General Disease Prediction System by www.irjet.net
4. Disease Prediction Using Machine Learning by International Research Journal of Engineering and Technology (IRJET).
5. Kaveeshwar, S.A., and Cornwall, J., 2014, "The current state of disease mellitus in

   India". AMJ, 7(1), pp. 45-48.
6. Dean, L., McEntyre, J., 2004, "The Genetic Landscape of Disease [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); Chapter 1, Introduction to Disease. 2004 Jul 7.
7. Machine Learning Methods Used in Disease by www.wikipedia.com
8. https://www.researchgate.net/publication/325116774_disease_prediction_using_machine_learning_techniques
9. https://ieeexplore.ieee.org/document/8819782/disease_prediction
10. Algorithms Details from www.dataspirant.com
11. https://www.youtube.com/disease_prediction
12. https://www.slideshare.com/disease_prediction
13. https:/ en.wikipedia.org/machine_learning_algorithms
14. https://en.wikipedia.org/wiki/Python_(programming_language)
15. https://wiki.python.org/TkInter
16. https://creately.com/lp/uml-diagram-tool/
17. https://app.diagrams.net/

2

3

5