

REPORT

(COURSE CODE : INT404)

TOPIC : DISEASE PREDICTION SYSTEM

SUBMITTED

BY

NAME : SHUBHAM KUMAR

REG NO. : 11809368

ROLL NO. : 55

SECTION : K18NZ

EMAIL ADDRESS : shubham139kumar@gmail.com

GITHUB LINK : <https://github.com/shubhamkr139/Artificial-Intelligence-Project>



Department of Intelligent System
School of Computer Science and Engineering
Lovely Professional University
Phagwara , Punjab

CONTENT

1. INTRODUCTION.....	3
2. SCOPE.....	3
3. PURPOSE.....	3
4. OBJECTIVE.....	4
5. ALGORITHM	4-5
6. REQUIREMENT ANALYSIS.....	5
7. CODE.....	6-15
8. OUTPUT.....	15-17

INTRODUCTION

At present, when one suffers from particular disease, then the person has to visit to doctor which is time consuming and costly too. Also if the user is out of reach of doctor and hospitals it may be difficult for the user as the disease can not be identified. So, if the above process can be completed using a automated program which can save time as well as money, it could be easier to the patient which can make the process easier.

Healthcare industry has become big business. The healthcare industry produces large amounts of health-care data daily that can be used to extract information for predicting disease that can happen to a patient in future while using the treatment history and health data. This hidden information in the healthcare data will be later used for affective decision making for patient's health. Also, this area need improvement by using the informative data in healthcare. Major challenge is how to extract the information from these data because the amount is very large so some data mining and machine learning techniques can be used. Also, the expected outcome and scope of this project is that if disease can be predicted than early treatment can be given to the patients which can reduce the risk of life and save life of patients and cost to get treatment of diseases can be reduced up to some extent by early recognition.

SCOPE

Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. The user can select various symptoms and can find the diseases with their probabilistic figures.

PURPOSE

The purpose of this system is to provide prediction for the general and more commonly occurring disease that when unchecked can turn into fatal disease. The system applies data mining techniques and decision tree algorithms. This system will predict the most possible disease based on the given symptoms and measures required to avoid the aggression of disease, it will also help the doctors analyse the pattern of presence of diseases in the society. In this project, the disease prediction system will carry out data mining in its preliminary stages, the system will be trained using machine learning and data mining.

OBJECTIVE

- The main objective of this research is to develop a heart prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical data set.
- Provides new approach to concealed patterns in the data.
- Helps avoid human biasness.
- To implement Naïve Bayes Classifier that classifies the disease as per the input of the user.
- Reduce the cost of medical tests.

ALGORITHM WE USED

✓ Decision tree :

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called classification trees. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

✓ Random Forest :

Random forest is an ensemble and most accurate learning algorithm ,suitable for medical applications. Chi square feature selection measure is used to evaluate between variables and determines whether they are correlated or not. In this paper ,we propose a classification model which uses random forest as classifier ,chi square and genetic algorithm as feature selection measures to predict disease.

✓ Naves Bayes :

Naive Bayes or Bayes Rule is the basis for many machine-learning and data mining methods. The rule (algorithm) is used to create models with predictive capabilities. It provides new ways of exploring and understanding data.

The Naïve Bayes Classifier technique is mainly applicable when the dimensionality of the inputs is high . Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. Naïve Bayes model recognizes the characteristics of patients with heart disease. It shows the probability of each input attribute for the predictable state.

Naive Bayes algorithm is preferred in the following cases.

- When the dimensionality of data is high.
- When the attributes are independent of each other.

Otherwise, attributes are assumed to be independent in order to simplify the computations involved and, in this sense, is considered “naïve”.

- When we expect more efficient output, as compared to other methods output.
- Exhibits high accuracy and speed when applied to large

Requirement Analysis

Functional Requirement :

- Predict disease with the given symptoms.
- Compare the given symptoms with the input datasets.

Non-Functional Requirement :

- Display the list of symptoms where user can select the symptoms.
- Naïve Bayes Classifier is used to classify the data sets.

CODE for Disease Prediction System

```
from tkinter import *

import numpy as np

import pandas as pd

#List of the symptoms is listed here in list

l1=['back_pain','constipation','abdominal_pain','diarrhoea','mild_fever','
yellow_urine','yellowing_of_eyes','acute_liver_failure','fluid_overload','swellin
g_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','n
eck_pain','dizziness','cramps','bruising','obesity','swollen_legs','swollen_bloo
d_vessels','puffy_face_and_eyes','enlarged_thyroid','brittle_nails','swollen_ex
tremities','excessive_hunger','extra_marital_contacts','drying_and_tingling_li
ps','slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','stiff_nec
k','swelling_joints','movement_stiffness','spinning_movements','loss_of_balan
ce','unsteadiness','weakness_of_one_body_side','loss_of_smell','bladder_dis
comfort','foul_smell_of
urine','continuous_feel_of_urine','passage_of_gases','internal_itching','toxic_lo
ok_(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_disturbance
s','receiving_blood_transfusion','receiving_unsterile_injections','coma','stom
ach_bleeding','distention_of_abdomen','history_of_alcohol_consumption','flui
d_overload','blood_in_sputum','prominent_veins_on_calf','palpitations','painfu
l_walking','pus_filled_pimples','blackheads','scurring','skin_peeling','silver_li
ke_dusting','small_dents_in_nails','inflammatory_nails','blister','red_sore_around_nose','yellow_crust_ooze']

#List of Diseases is listed in list disease.

disease=['Fungal infection','Allergy','GERD','Chronic cholestasis','Drug
Reaction','Peptic ulcer disease','AIDS','Diabetes','Gastroenteritis','Bronchial
Asthma','Hypertension','Migraine','Cervical spondylosis','Paralysis (brain
hemorrhage)','Jaundice','Malaria','Chicken pox','Dengue','Typhoid','hepatitis
A','Hepatitis B','Hepatitis C','Hepatitis D','Hepatitis E','Alcoholic
hepatitis','Tuberculosis','Common Cold','Pneumonia','Dimorphic
```

hemorrhoids(piles)', 'Heart attack', 'Varicose veins', 'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia', 'Osteoarthritis', 'Arthritis', '(vertigo) Parosmal Positional Vertigo', 'Acne', 'Urinary tract infection', 'Psoriasis', 'Impetigo']

```
l2=[]
```

```
for i in range(0,len(l1)):
```

```
    l2.append(0)
```

```
df=pd.read_csv("Prototype.csv")
```

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

```
df.replace({'prognosis':{'Fungal infection':0, 'Allergy':1, 'GERD':2, 'Chronic cholestasis':3, 'Drug Reaction':4, 'Peptic ulcer disease':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, 'Dimorphic hemorrhoids(piles)':28, 'Heart attack':29, 'Varicose veins':30, 'Hypothyroidism':31, 'Hyperthyroidism':32, 'Hypoglycemia':33, 'Osteoarthritis':34, 'Arthritis':35, '(vertigo) Parosmal Positional Vertigo':36, 'Acne':37, 'Urinary tract infection':38, 'Psoriasis':39, 'Impetigo':40}}, inplace=True)
```

#check the df

```
#print(df.head())
```

```
X= df[l1]
```

```
#print(X)
```

```
y = df[["prognosis"]]
```

```
np.ravel(y)
```

```
#print(y)
```

#Read a csv named Testing.csv

```
tr=pd.read_csv("Prototype 1.csv")
```

#Use replace method in pandas.

```

tr.replace({'prognosis':{'Fungal
infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,
'Peptic ulcer disease':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,'Dimorphic
hemorrhoids(piles)':28,'Heart attack':29,'Varicose
veins':30,'Hypothyroidism':31,
'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthritis':34,'Arthritis':3
5,
'(vertigo) Paroxysmal Positional Vertigo':36,'Acne':37,'Urinary tract
infection':38,'Psoriasis':39,
'Impetigo':40}},inplace=True)

X_test= tr[l1]
y_test = tr[["prognosis"]]
#print(y_test)
np.ravel(y_test)

def DecisionTree():
    from sklearn import tree
    clf3 = tree.DecisionTreeClassifier()
    clf3 = clf3.fit(X,y)
    from sklearn.metrics import accuracy_score
    y_pred=clf3.predict(X_test)
    print(accuracy_score(y_test, y_pred))

```



```

print(accuracy_score(y_test, y_pred,normalize=False))

psymptoms =
[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom
5.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'):

    t1.delete("1.0", END)

    t1.insert(END, disease[a])

else:

    t1.delete("1.0", END)

    t1.insert(END, "Not Found")

def randomforest():

    from sklearn.ensemble import RandomForestClassifier

    clf4 = RandomForestClassifier()

    clf4 = clf4.fit(X,np.ravel(y))

    # calculating accuracy

    from sklearn.metrics import accuracy_score

```

```

y_pred=clf4.predict(X_test)

print(accuracy_score(y_test, y_pred))

print(accuracy_score(y_test, y_pred,normalize=False))


psymptoms =
[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom
5.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'):

    t2.delete("1.0", END)

    t2.insert(END, disease[a])

else:

    t2.delete("1.0", END)

    t2.insert(END, "Not Found")

def NaiveBayes():

    from sklearn.naive_bayes import GaussianNB

    gnb = GaussianNB()

```

```

gnb=gnb.fit(X,np.ravel(y))

from sklearn.metrics import accuracy_score

y_pred=gnb.predict(X_test)

print(accuracy_score(y_test, y_pred))

print(accuracy_score(y_test, y_pred,normalize=False))

psymptoms =
[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom
5.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'):

    t3.delete("1.0", END)

    t3.insert(END, disease[a])

else:

    t3.delete("1.0", END)

    t3.insert(END, "Not Found")

# GUI stuff.....

```

```

root = Tk()

root.configure(background='black')

Symptom1 = StringVar()
Symptom1.set("Select Here")

Symptom2 = StringVar()
Symptom2.set("Select Here")

Symptom3 = StringVar()
Symptom3.set("Select Here")

Symptom4 = StringVar()
Symptom4.set("Select Here")

Symptom5 = StringVar()
Symptom5.set("Select Here")

Name = StringVar()

w2 = Label(root, justify=LEFT, text="Disease Predictor System",
fg="White", bg="Green")

w2.config(font=("Times",30,"bold italic"))

w2.grid(row=1, column=0, columnspan=2, padx=100)

w2 = Label(root, justify=LEFT, text="A Project by Shubham Kumar",
fg="White", bg="Blue")

w2.config(font=("Times",30,"bold italic"))

w2.grid(row=2, column=0, columnspan=2, padx=100)

NameLb = Label(root, text="Name of the Patient", fg="White",
bg="Gray")

NameLb.config(font=("Times",15,"bold italic"))

NameLb.grid(row=6, column=0, pady=15, sticky=W)

S1Lb = Label(root, text="Symptom 1", fg="purple", bg="Pink")

S1Lb.config(font=("Times",15,"bold italic"))

```

```

S1Lb.grid(row=7, column=0, pady=10, sticky=W)
S2Lb = Label(root, text="Symptom 2", fg="purple", bg="Pink")
S2Lb.config(font=("Times",15,"bold italic"))
S2Lb.grid(row=8, column=0, pady=10, sticky=W)
S3Lb = Label(root, text="Symptom 3", fg="purple",bg="Pink")
S3Lb.config(font=("Times",15,"bold italic"))
S3Lb.grid(row=9, column=0, pady=10, sticky=W)
S4Lb = Label(root, text="Symptom 4", fg="purple", bg="Pink")
S4Lb.config(font=("Times",15,"bold italic"))
S4Lb.grid(row=10, column=0, pady=10, sticky=W)
S5Lb = Label(root, text="Symptom 5", fg="purple", bg="Pink")
S5Lb.config(font=("Times",15,"bold italic"))
S5Lb.grid(row=11, column=0, pady=10, sticky=W)
lrLb = Label(root, text="DecisionTree", fg="white", bg="red")
lrLb.config(font=("Times",15,"bold italic"))
lrLb.grid(row=15, column=0, pady=10,sticky=W)
destreeLb = Label(root, text="RandomForest", fg="White",
bg="Brown")
destreeLb.config(font=("Times",15,"bold italic"))
destreeLb.grid(row=17, column=0, pady=10, sticky=W)
ranfLb = Label(root, text="NaiveBayes", fg="White", bg="green")
ranfLb.config(font=("Times",15,"bold italic"))
ranfLb.grid(row=19, column=0, pady=10, sticky=W)
OPTIONS = sorted(l1)
NameEn = Entry(root, textvariable=Name)
NameEn.grid(row=6, column=1)
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)

dst = Button(root, text="Prediction 1",
command=DecisionTree,bg="Red",fg="White")

dst.config(font=("Times",15,"bold italic"))

dst.grid(row=8, column=3,padx=10)

rnf = Button(root, text="Prediction 2",
command=randomforest,bg="Brown",fg="white")

rnf.config(font=("Times",15,"bold italic"))

rnf.grid(row=9, column=3,padx=10)

lr = Button(root, text="Prediction 3",
command=NaiveBayes,bg="Green",fg="white")

lr.config(font=("Times",15,"bold italic"))

lr.grid(row=10, column=3,padx=10)

t1 = Text(root, height=1, width=40,bg="Light green",fg="red")

t1.config(font=("Times",15,"bold italic"))

t1.grid(row=15, column=1, padx=10)

t2 = Text(root, height=1, width=40,bg="Light green",fg="red")

t2.config(font=("Times",15,"bold italic"))

t2.grid(row=17, column=1 , padx=10)

t3 = Text(root, height=1, width=40,bg="Light green",fg="red")

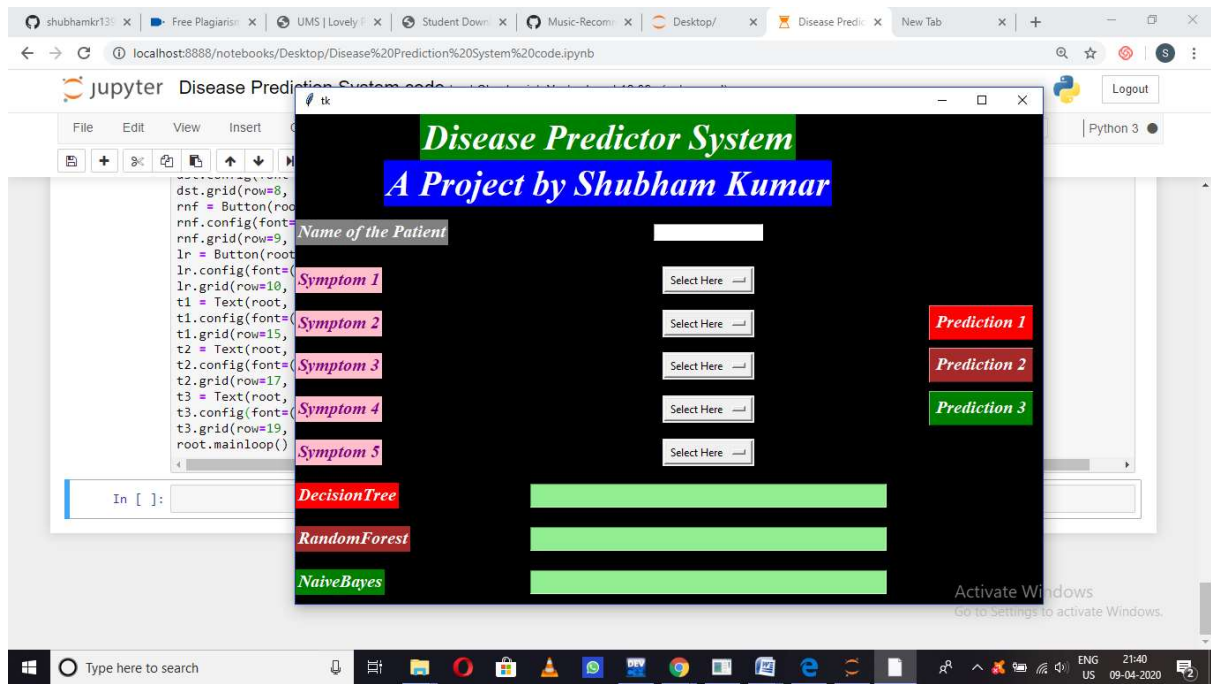
```

```
t3.config(font=("Times",15,"bold italic"))
```

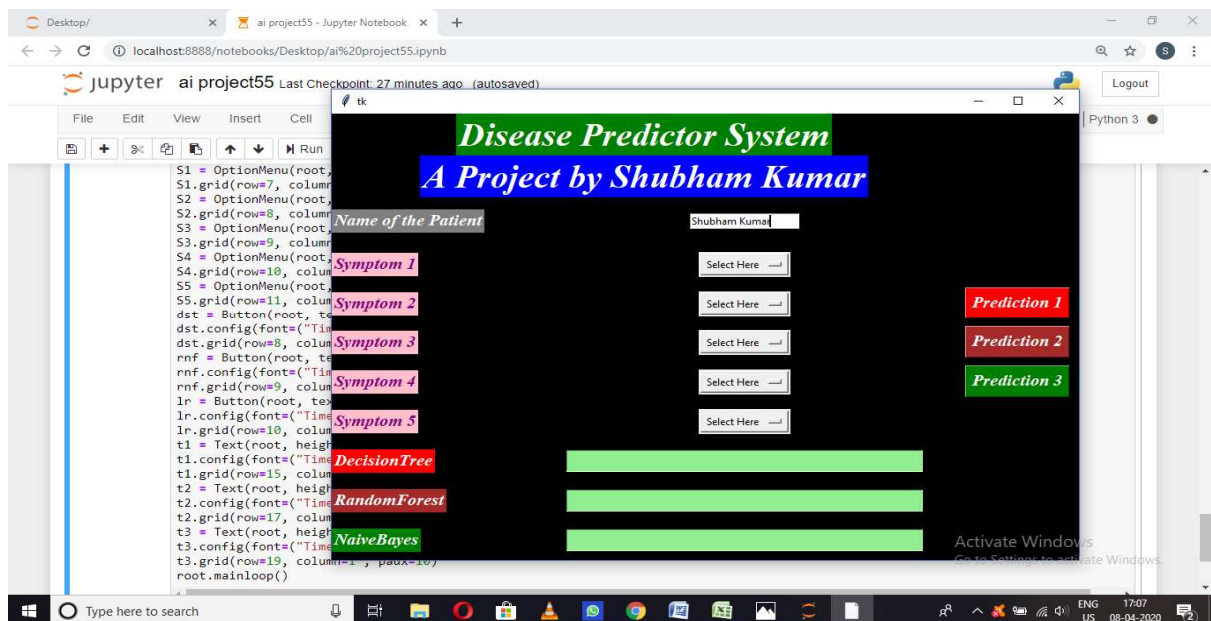
```
t3.grid(row=19, column=1 , padx=10)
```

```
root.mainloop()
```

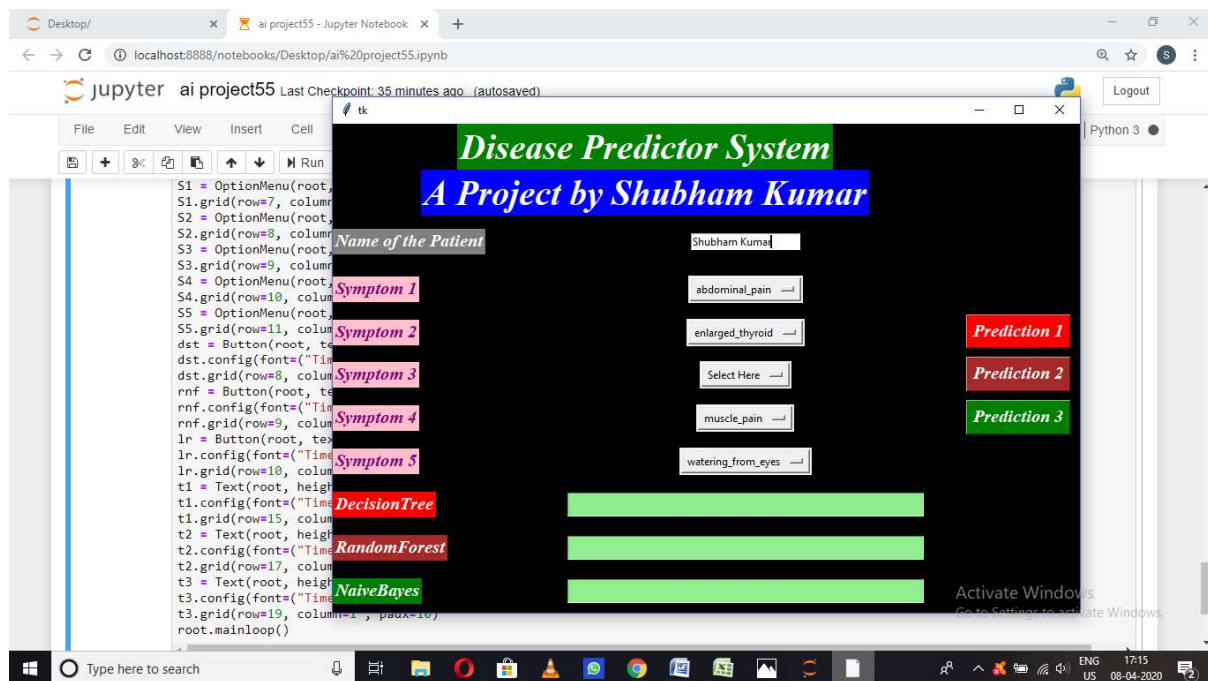
OUTPUT OF PROGRAM



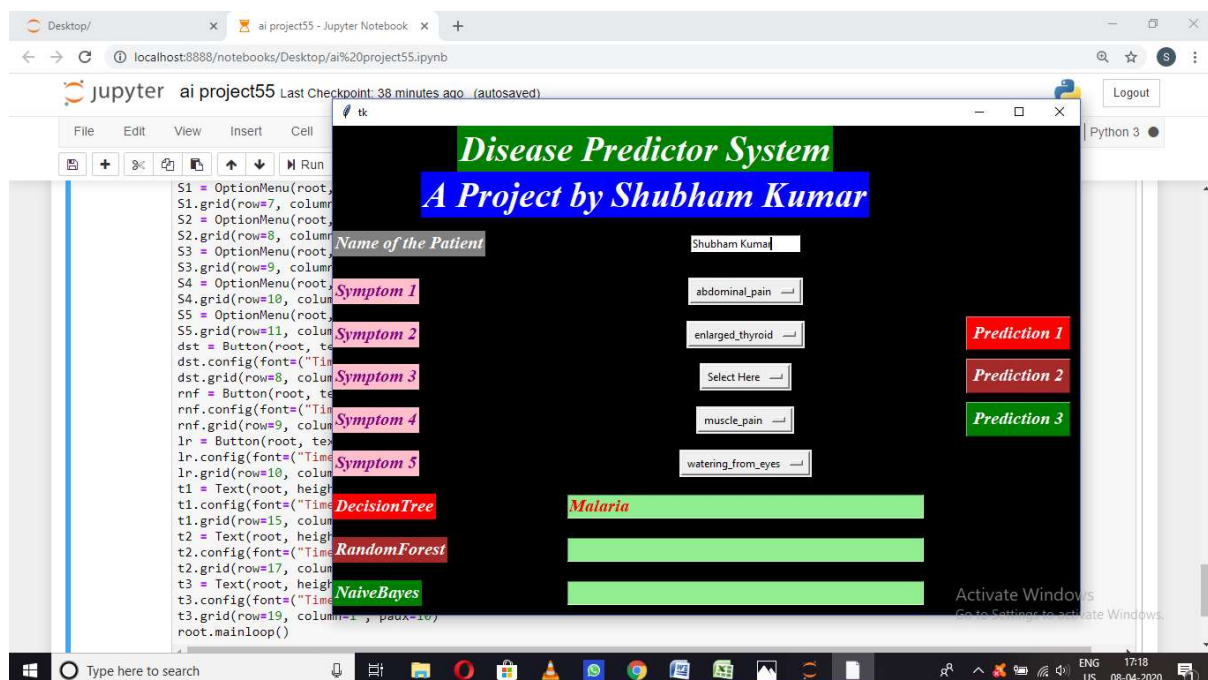
PAERT 1: Enter the name of the patient.



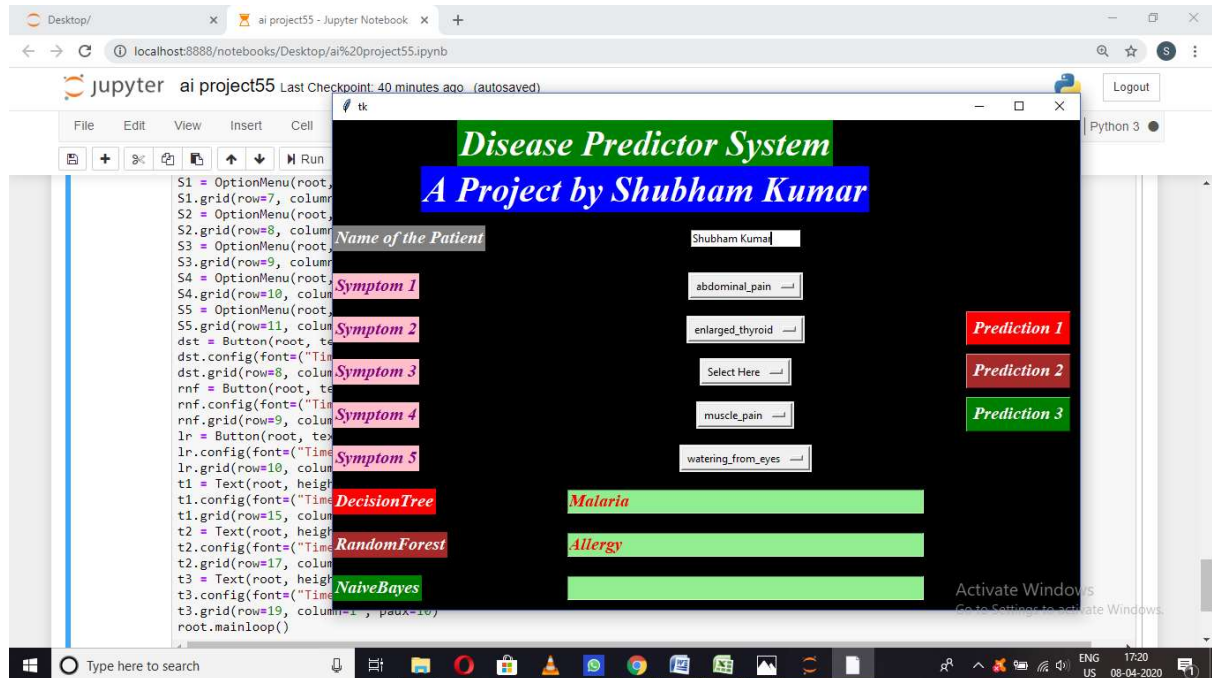
PART 2 : Enter the Symptom1, Symptom2, Symptom3, Symptom4, Symptom5.



PART 3 : Select the prediction 1. Then display the possible disease using Decision Tree technique.



PART4 : Select the prediction 2. Then display the possible disease using Random Forest technique.



PART 5 : Select the prediction 3 .Then display the possible disease using Naive Bayes technique.

