

DISEASE PREDICTION USING MACHINE LEARNING

An App Development-2 Report Submitted
In partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

by

UGGU HARICHANDANA

21N31A05P7

VANGA SRUTHI

22N35A0526

WALI RAHUL

21N31A05R4

Under the esteemed guidance of

MS.C.S.N.DURGA
ASSOCIATE PROFESSOR



Department of Computer Science and Engineering

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA & NAAC with 'A' Grade)

Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100

website: www.mrcet.ac.in

2023-2024



Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA & NAAC with 'A' Grade)

Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100

website: www.mrcet.ac.in

CERTIFICATE

This is to certify that this is the bonafide record of the project entitled “DISEASE PREDICTION USING MACHINE LEARNING”, submitted by **U.HARICHANDANA (21N31A05P7), V.SRUTHI (22N35A0526) and WALI RAHUL (21N31A05R4)** of B.Tech in the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering, Department of CSE during the year 2023-2024. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Internal Guide

MS.C.S.N.DURGA
Associate Professor

Head of the Department

Dr. S. Shanthi
Professor

External Examiner

DECLARATION

We hereby declare that the project titled “**DISEASE PREDICTION USING MACHINE LEARNING**” submitted to Malla Reddy College of Engineering and Technology (UGC Autonomous), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a result of original research carried-out in this thesis. It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma.

U.HARICHANDANA - (21N31A05P7)

V.SRUTHI- (22N35A0526)

RAHUL- (21N31A05R4)

ACKNOWLEDGEMENT

We feel honored to place our warm salutation to our college Malla Reddy College of Engineering and Technology (UGC-Autonomous) for giving us an opportunity to do this Project as part of our B.Tech Program. We are ever grateful to our **Director Dr. VSK Reddy** and **Principal Dr.S.Srinivasa Rao** who enabled us to have experience in engineering and gain profound technical knowledge.

We express our heartiest thanks to our **HOD, Dr. S. Shanthi** for encouraging us in every aspect of our course and helping us realize our full potential.

We would like to thank our **Project Guide MS.C.S.N.DURGA** for his/her regular guidance, suggestions and constant encouragement. We are extremely grateful to our **Project Coordinator MS.C.S.N.DURGA** for his/her continuous monitoring and unflinching co-operation throughout project work.

We would like to thank our **Class Incharge Mr.P.DASTAGIRI REDDY** who in spite of being busy with his/her academic duties took time to guide and keep us on the correct path.

We would also like to thank all the faculty members and supporting staff of the Department of CSE and all other departments who have been helpful directly or indirectly in making our project a success.

We are extremely grateful to our parents for their blessings and prayers for the completion of our project that gave us strength to do our project.

With regards and gratitude

U. HARICHANDANA - (21N31A05P7)

V.SRUTHI - (22N35A0526)

W.RAHUL - (21N31A05R4)

ABSTRACT

The health care systems collect data and reports from the hospitals or patient's database by machine learning and data processing techniques which is employed to predict the disease so as to create reports supported by the results which are used for various kinds of predictions for disease and which is that the leading explanation for the human's death since past years. Medical reports and data have been extracted from various databases to predict a number of the required diseases which are commonly found in people nowadays: breast cancer, heart disease, and diabetes disease, and make their life more critical to measure. Nowadays, technology advancement within the health care industry has been helping people to create their process easier by suggesting hospitals and doctors to travel to for his or her treatment, where to admit and which hospitals are the simplest for treating the desired disease. We've implemented this sort of system in our application to form people's life simpler by predicting the disease by inputting certain data from their reports which can give the result positive or negative supported by the disease prediction they are going to be having a choice to get recommendation of best hospitals with best doctors nearby from the past users or guardians.

TABLE OF CONTENTS

S.NO	TITLE	PG.NO
1	INTRODUCTION	09
	1.1 PURPOSE AND OBJECTIVES	09
	1.2 SCOPE OF PROJECT	10
2	SYSTEM ANALYSIS	11
	2.1 HARDWARE AND SOFTWARE REQUIREMENTS	11
3	SYSTEM DESIGN	12
	3.1 ARCHITECTURE	13
	3.2 UML DIAGRAMS	20
4	METHODOLOGY	21
	4.2 PROCESS/ALGORITHM	22
5	IMPLEMENTATION	23
	5.1 SAMPLE CODE	36
	5.2 OUTPUT SCREENS	37
6	CONCLUSION & FUTURE SCOPE	38
7	BIBLIOGRAPHY	39

LIST OF FIGURES

FIGURE.NO	NAME	PG.NO
3.1.1	SYSTEM ARCHITECTURE DIAGRAM	13
3.2.1	CLASS DIAGRAM	14
3.2.2	ACTIVITY DIAGRAM	15
3.2.3	STATE CHART DIAGRAMS	16
3.2.4	USECASE DIAGRAM	17
3.2.5	SEQUENCE DIAGRAMS	18
3.2.6	DEPLOYMENT DIAGRAM	19
3.2.7	COMPONENT DIAGARM	20

LIST OF OUTPUTS

FIGURE.NO	NAME	PG.NO
5.2.1	PREDICTION MODULE	37
5.2.2	DROPDOWN BOX FOR SYMPTOMS	37
5.2.3	PREDICTION MODULE WITH RESULT	38
5.2.4	PREDICTION MODULE SHOWING ALERT POP-UP WHEN NAME IS NOT PROVIDED	38

1.INTRODUCTION

1.1 PURPOSE AND OBJECTIVES:

The purpose and objectives of disease prediction using machine learning (ML) can vary depending on the context and the specific disease being targeted. However, some common objectives include:

1. **Early Detection:** ML algorithms can analyze large amounts of data, including patient health records, genetic information, and environmental factors, to identify patterns indicative of disease onset or progression before symptoms manifest. Early detection allows for timely intervention and treatment, potentially improving patient outcomes and reducing healthcare costs.
2. **Risk Stratification:** ML models can assess an individual's risk of developing a particular disease based on various factors such as age, gender, lifestyle, and medical history. This risk stratification enables targeted preventive measures and personalized healthcare interventions for high-risk individuals.
3. **Precision Medicine:** By leveraging ML techniques, healthcare providers can tailor treatment plans to individual patients based on their unique characteristics, genetic makeup, and disease risk profile. This approach, known as precision medicine, aims to maximize treatment efficacy while minimizing adverse effects.
4. **Public Health Surveillance:** ML algorithms can analyze population-level data from sources such as electronic health records, social media, and wearable devices to monitor disease trends, identify outbreaks, and predict future health burdens. This information enables public health authorities to implement proactive measures to control the spread of diseases and allocate resources effectively.
5. **Drug Discovery and Development:** ML techniques can expedite the process of drug discovery by analyzing large datasets to identify potential drug candidates, predict their efficacy and safety profiles, and optimize treatment regimens. This accelerates the development of new therapies for various diseases, including rare and complex conditions.
6. **Patient Empowerment and Education:** Disease prediction models can empower individuals to take proactive steps to manage their health by providing personalized risk assessments, preventive recommendations, and educational resources. This fosters a culture of health awareness and encourages healthier lifestyle choices.

1.2 SCOPE OF THE PROJECT:

The scope of disease prediction using machine learning (ML) is broad and encompasses various aspects of healthcare, including but not limited to:

1. **Predictive Modeling:** ML algorithms can be employed to develop predictive models that analyze diverse datasets, such as electronic health records, medical imaging, genomic data, and environmental factors, to forecast the likelihood of disease occurrence, progression, or recurrence in individuals.
2. **Risk Assessment:** ML-based risk assessment tools can evaluate an individual's susceptibility to specific diseases based on demographic information, lifestyle factors, genetic predisposition, and medical history. These assessments enable targeted preventive strategies and early intervention for high-risk individuals.
3. **Early Detection:** ML algorithms are capable of detecting subtle patterns and deviations in health data that may precede the onset of clinical symptoms. Early detection of diseases, such as cancer, cardiovascular conditions, and neurological disorders, can facilitate timely diagnosis and treatment, potentially improving patient outcomes and reducing healthcare costs.
4. **Personalized Medicine:** ML enables the development of personalized treatment plans tailored to individual patients based on their unique characteristics, genetic makeup, and disease risk profile. By considering factors such as drug response prediction, treatment efficacy, and adverse event risk, personalized medicine aims to optimize therapeutic outcomes and minimize treatment-related complications.
5. **Public Health Surveillance:** ML techniques can analyze large-scale healthcare data, including electronic health records, infectious disease surveillance systems, and social media platforms, to monitor disease trends, detect outbreaks, and forecast future health burdens. This information is invaluable for public health authorities in implementing targeted interventions, resource allocation, and disease prevention strategies.
6. **Drug Discovery and Development:** ML plays a significant role in accelerating drug discovery and development processes by facilitating the identification of novel drug targets, predicting drug efficacy and safety profiles, and optimizing drug formulations.

2.SYSTEM ANALYSIS

2.1 HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements :

- OS: Windows 10
- RAM: Minimum of 4GB
- CPU: 64bit

Software Requirements :

- Code editors: Visual Studio code
- Coding Language: Python
- GUI: Tkinter

3.SYSTEM DESIGN

3.1 ARCHITECTURE:

The architecture of disease prediction using machine learning (ML) typically involves several interconnected components that collectively enable the development and deployment of predictive models. While the specific architecture may vary depending on factors such as the nature of the disease, available data sources, and intended use case, a generalized architecture may include the following components:

1. Data Acquisition and Preprocessing:

- **Data Sources:** Collect relevant data from various sources, including electronic health records (EHRs), medical imaging, genetic data, wearable devices, and environmental sensors.
- **Data Preprocessing:** Cleanse, preprocess, and integrate heterogeneous data types to ensure data quality, consistency, and compatibility for subsequent analysis. This may involve tasks such as data cleaning, normalization, feature extraction, and missing value imputation.

2. Feature Engineering:

- **Feature Selection:** Identify informative features or variables from the dataset that are relevant to the prediction task. This may involve domain knowledge, statistical analysis, and feature importance techniques.
- **Feature Transformation:** Transform raw data into a suitable representation for ML algorithms. This may include techniques such as dimensionality reduction (e.g., principal component analysis) and feature scaling.

3. Model Development:

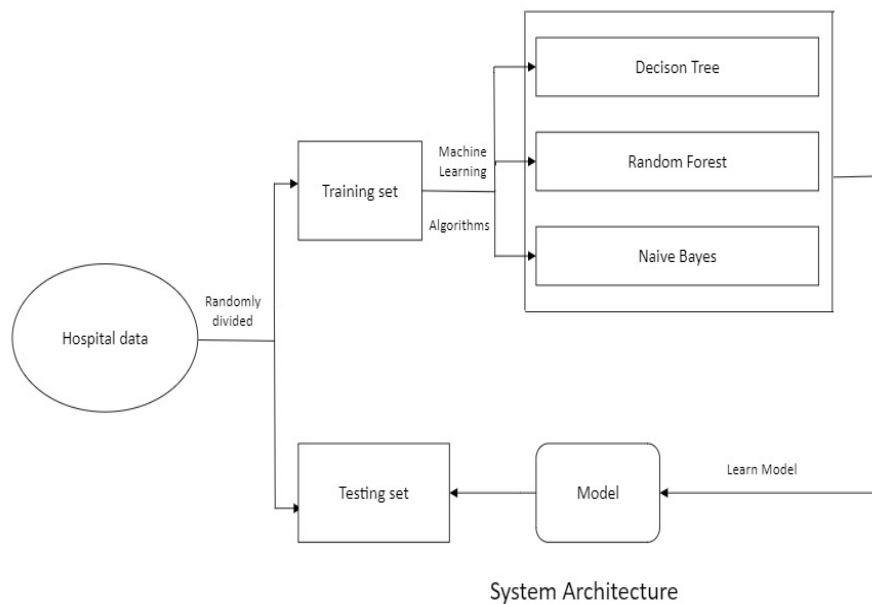
- **Algorithm Selection:** Choose appropriate ML algorithms based on the nature of the prediction task, data characteristics, and computational requirements. Common algorithms used for disease prediction include logistic regression, support vector machines (SVM), random forests, gradient boosting machines (GBM), and deep learning models.

- **Model Training:** Train ML models using labeled data (supervised learning) or unlabeled data (unsupervised learning) to learn patterns and relationships between input features and disease outcomes. This involves splitting the dataset into training, validation, and test sets, and iteratively optimizing model parameters to minimize prediction error.
- **Model Evaluation:** Assess the performance of trained models using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, area under the receiver operating characteristic curve (AUC-ROC), and area under the precision-recall curve (AUC-PR). Perform cross-validation to estimate model generalization performance and mitigate overfitting.

4. Model Deployment:

- **Integration with Healthcare Systems:** Integrate trained ML models into existing healthcare IT infrastructure, such as electronic health record (EHR) systems, clinical decision support systems (CDSS), or telemedicine platforms, to facilitate seamless deployment and real-time prediction.
- **Scalability and Performance Optimization:** Ensure scalability, reliability, and efficiency of deployed models to handle large volumes of data and user requests.

3.1.1 System architecture diagram



○

3.2 UML DIAGRAMS:

3.2.1 Class Diagram:

1. DiseasePredictionSystem:

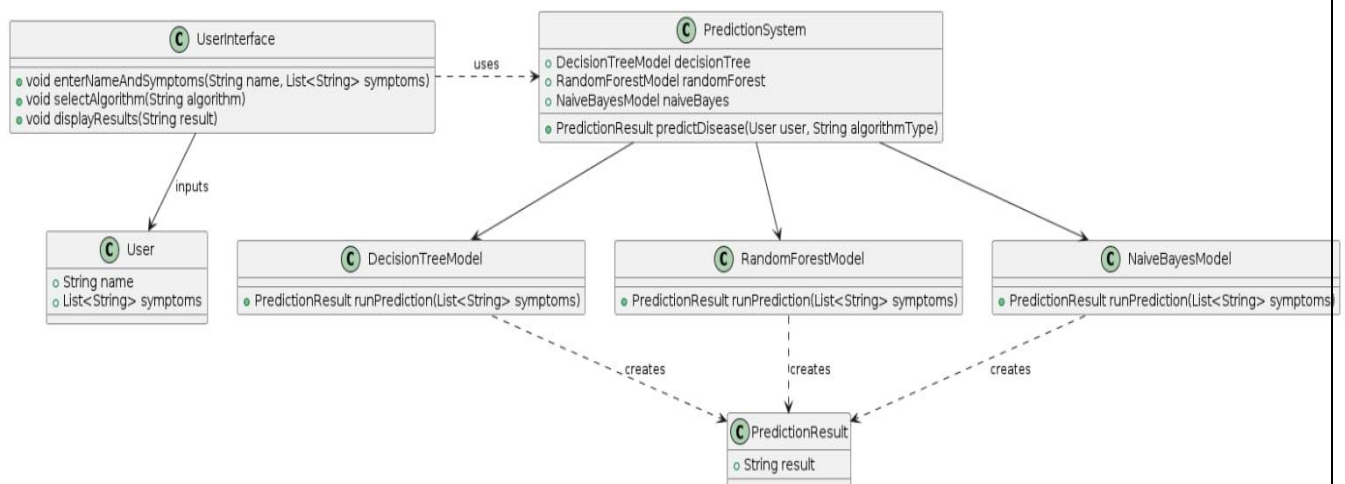
- This class encapsulates the entire disease prediction system.
- It holds instances of the three machine learning models: DecisionTreeModel, RandomForestModel, and NaiveBayesModel.
- It has methods to train the models (trainModels) and to predict diseases given symptoms (predictDisease).

2. DecisionTreeModel, RandomForestModel, NaiveBayesModel:

- These classes represent machine learning models.
- Each model has methods to train the model with training data (trainModel) and to make predictions given symptoms (predict).

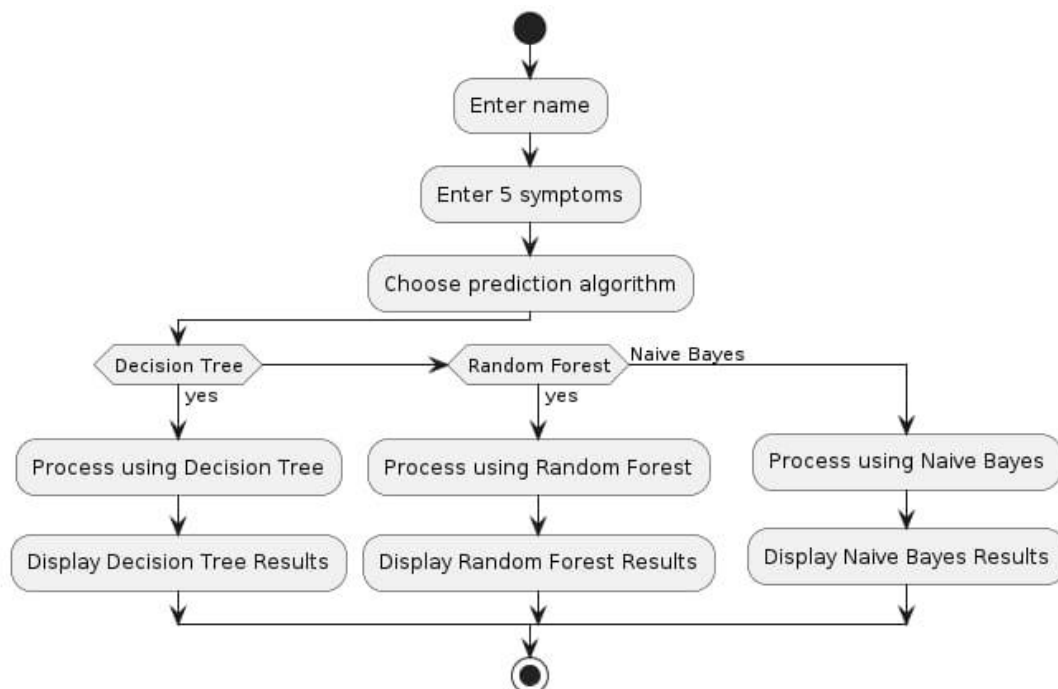
3. UserInput:

- This class represents the user input module.
- It holds an array of symptoms.
- It provides a method getSymptoms() to retrieve symptoms from the user.



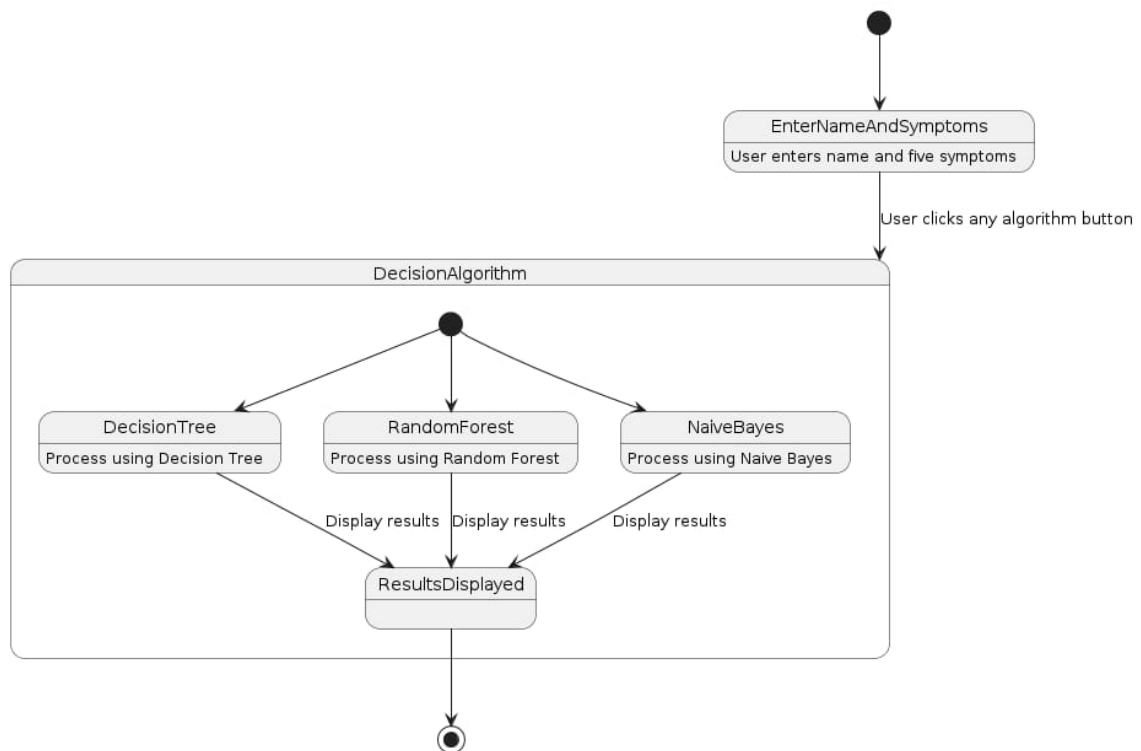
3.2.2 Activity diagram:

- **Start:** The start point of the activity diagram.
- **Start the Disease Prediction System:** Initialization step where the system is started.
- **Ask user for 5 symptoms:** The system prompts the user to input 5 symptoms.
- **Collect user input:** The system collects the input provided by the user.
- **Determine ML algorithm to use:** Based on the user input or predetermined logic, the system determines which ML algorithm to use for disease prediction (Decision Tree, Random Forest, or Naive Bayes).
- **Use [Algorithm] model to predict disease:** Depending on the chosen ML algorithm, the system utilizes the corresponding model to predict the disease based on the provided symptoms.
- **Display predicted disease:** The predicted disease is displayed to the user.
- **Stop:** The end point of the activity diagram.



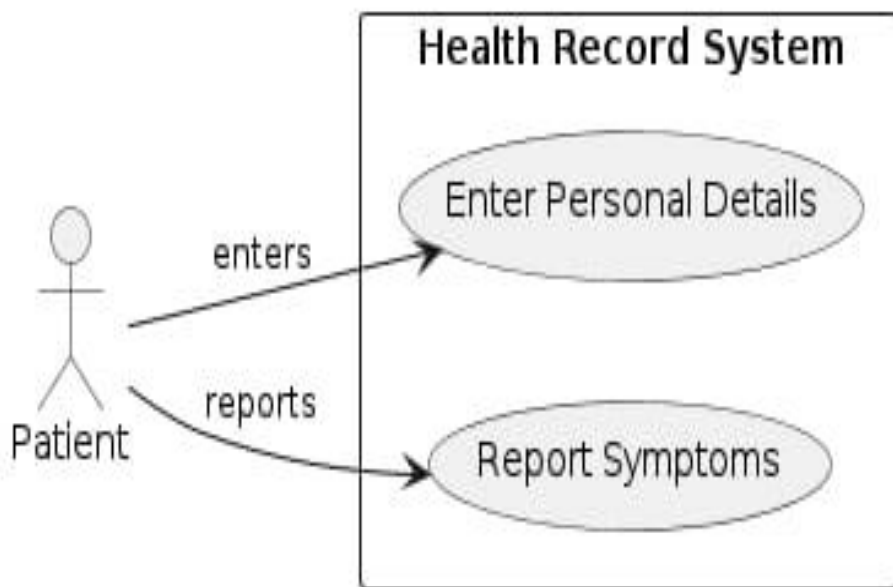
3.2.3 State chart diagram:

- **Start:** The initial state where the system begins the disease prediction process.
- **AskSymptoms:** The system asks the user for 5 symptoms.
- **CollectInput:** The system collects the input provided by the user.
- **ChooseAlgorithm:** The system determines which ML algorithm to use for disease prediction (Decision Tree, Random Forest, or Naive Bayes).
- **PredictDisease:** The system enters the state where it executes the selected ML algorithm to predict the disease.
- **DecisionTree, RandomForest, NaiveBayes:** These states represent the execution of each ML algorithm.
- **EndPrediction:** The system transitions to this state after completing the prediction process with the selected ML algorithm.
- **DisplayResult:** The system displays the predicted disease to the user.
- **End:** The final state where the process ends.



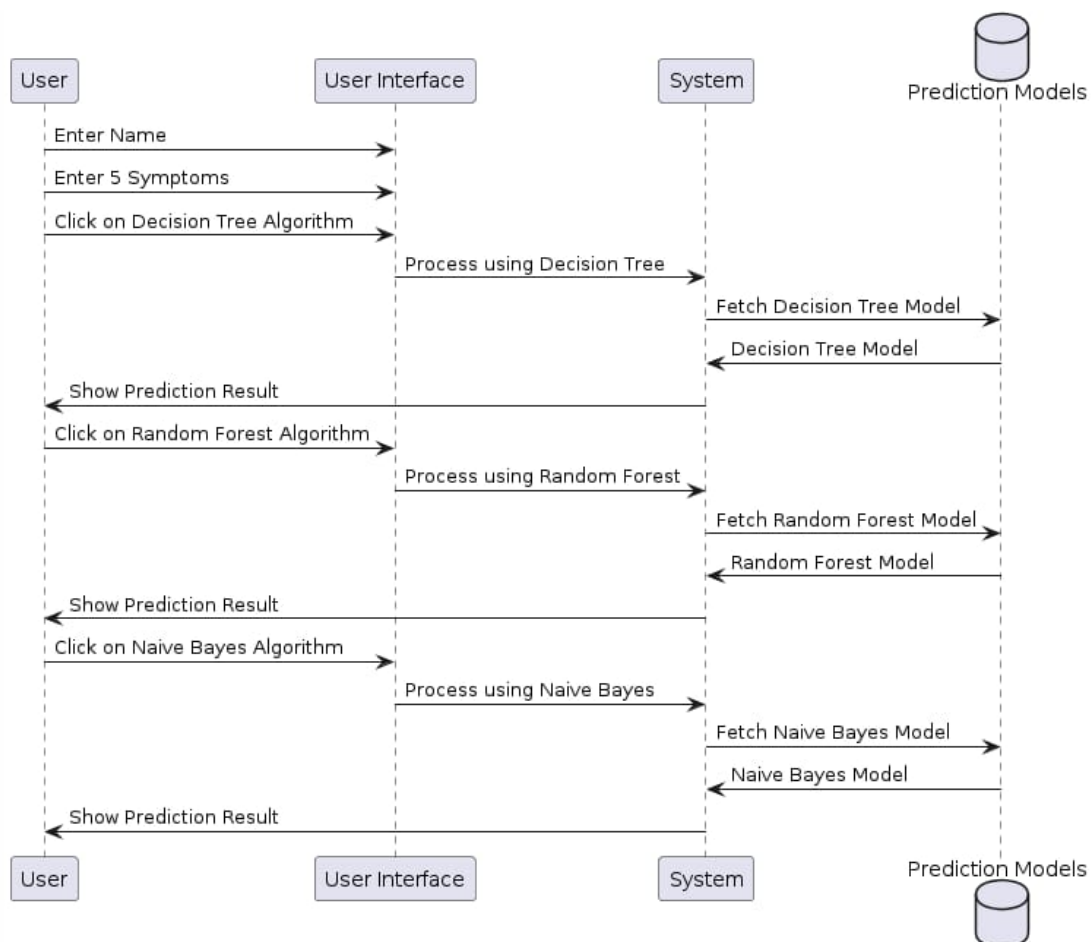
3.2.4 Use Case Diagram:

- **User:** Represents the user interacting with the system.
- **MLModel:** Represents the machine learning model used for disease prediction.
- **DiseasePredictionSystem:** Represents the system itself.
- **Provide Symptoms:** Use case where the user provides 5 symptoms.
- **Select ML Algorithm:** Use case where the user selects the machine learning algorithm to use for prediction.
- **Predict Disease:** Use case where the selected ML algorithm predicts the disease based on the provided symptoms.
- **Display Result:** Use case where the predicted disease result is displayed to the user.



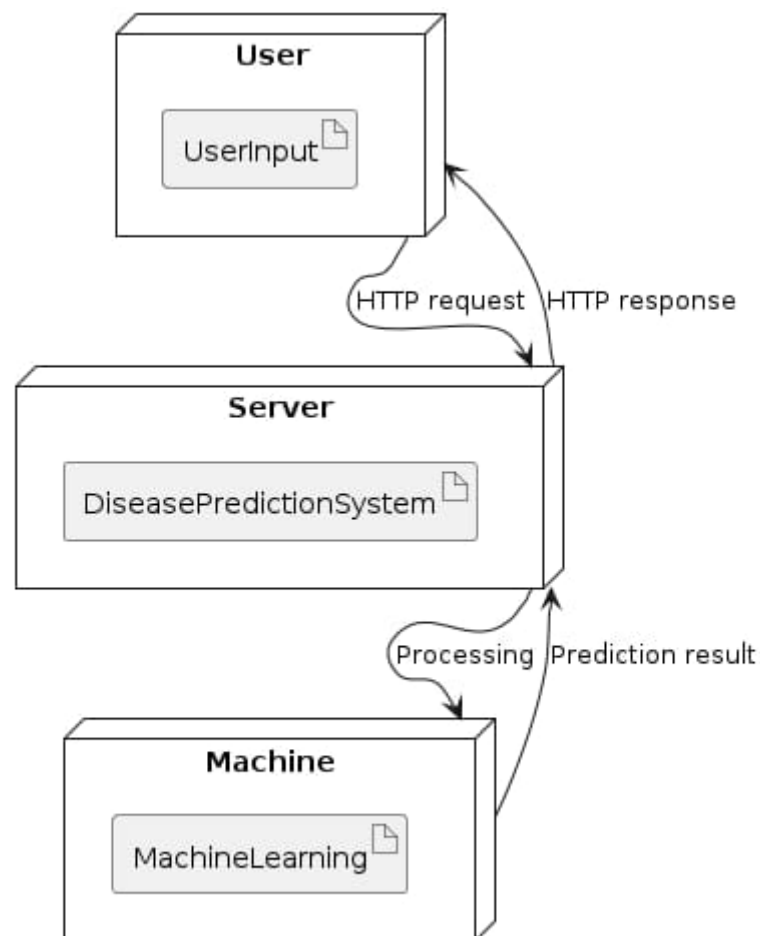
3.2.5 Sequence Diagram:

- **User:** Represents the user interacting with the system.
- **DiseasePredictionSystem (DPS):** Represents the system component responsible for managing the disease prediction process.
- **MLModel (ML):** Represents the machine learning model used for disease prediction.
- The sequence starts with the user providing 5 symptoms to the DiseasePredictionSystem.
- The DiseasePredictionSystem collects the provided symptoms.
- The DiseasePredictionSystem determines which ML algorithm to use for prediction.



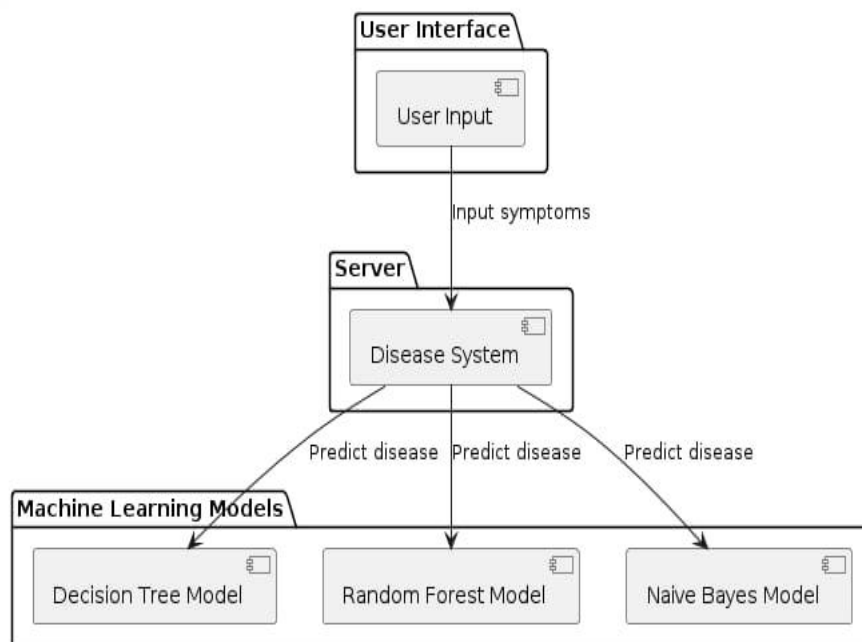
3.2.6 Deployment Diagram:

- **User Interface (User):** This component represents the interface through which users interact with the system. Users input symptoms into the system.
- **Machine Learning Models (Decision Tree, Random Forest, Naive Bayes):** These components represent the ML models used in the system for disease prediction. Each model takes input symptoms from the user and outputs a prediction based on its algorithm.
- **Dataset (Data):** This component represents the dataset used to train the ML models. It stores the symptom data used by the models for prediction.
- Arrows represent the flow of data and control:
 - Users input symptoms which are sent to each ML model.



3.2.7 Component Diagram:

- **Disease Prediction System (DiseasePredictionSystem):** This component represents the overall system responsible for disease prediction.
- **Machine Learning Models (Decision Tree, Random Forest, Naive Bayes):** These components represent the ML models used for disease prediction.
- **Dataset (Data):** This component represents the dataset used to train the ML models.
- **Internal Components:**
 - **User Interface (UserInterface):** This component handles user input of symptoms.
 - **Data Processing (DataProcessing):** This component processes the input symptoms and interacts with the ML models.
 - **Prediction Results (PredictionResults):** This component receives and presents the prediction results to the user.
 - The prediction results flow from each ML model to the Prediction Results component for display to the user.



4.METHODOLOGY:

4.1 PROCESS/ALGORITHM:

RANDOM FOREST :

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. 27 The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Implementation Steps are given below:

- o Data Pre-processing step
- o Fitting the Random forest algorithm to the Training set
- o Predicting the test result
- o Test accuracy of the result (Creation of Confusion matrix)
- o Visualizing the test set result.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

DECISION TREE:

Decision tree is classified as a very effective and versatile classification technique. It is used in pattern recognition and classification for image. It is used for classification in very complex problems due to its high adaptability. It is also capable of engaging problems of higher dimensionality. It mainly consists of three parts root, nodes and leaf. Roots consists of attribute which has most effect on the outcome, leaf tests for value of certain attribute and leaf gives out the output of the tree. Decision tree is the first prediction method we have used in this project. It gives us an accuracy of ~95%.

NAIVE BAYES:

Naïve Bayes algorithm is a family of algorithms based on naïve bayes theorem. They share a a common principle is that every pair of predictions is independent of each other. It also makes an assumption that features make an independent and equal contribution to the prediction. In this project we have used naïve bayes algorithms to gain a ~95% accurate prediction.

5.IMPLEMENTATION:

5.1 SAMPLE CODE

```
from tkinter import *

from tkinter import ttk

import numpy as np

import pandas as pd

# from gui_stuff import

*ll=['back_pain','constipation','abdominal_pain','itching','nodal_skin_eruptions','headache','ch
ills','cough','diarrhoea','stomach_pain','skin_rash','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_sputu
 m',
 '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_c
 alf',
 'palpitations','painful_walking','pus_filled_pimples','blackheads','scurring','skin_peeling',
 'silver_like_dusting','small_dents_in_nails','inflammatory_nails','blister','red_sore_around_no
 se',
 'yellow_crust_ooze']

disease=['Fungal infection','Allergy','GERD','Chronic cholestasis','Drug Reaction',
 'Peptic ulcer diseae','AIDS','Diabetes','Gastroenteritis','Bronchial Asthma','Hypertension',
 ' Migraine','Cervical spondylosis',
 'Paralysis (brain hemorrhage)','Jaundice','Malaria','Chicken pox','Dengue','Typhoid','hepatitis
 A',
 'Hepatitis B','Hepatitis C','Hepatitis D','Hepatitis E','Alcoholic hepatitis','Tuberculosis',
 'Common Cold','Pneumonia','Dimorphic hemmorhoids(piles)',
 'Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteoarthri
 stis',
 'Arthritis','(vertigo) Paroymsal Positional Vertigo','Acne','Urinary tract infection','Psoriasis',
 'Impetigo']

l2=[]


```

for x in range(0,len(11)):

l2.append(0)

# TESTING DATA df -----

df=pd.read_csv("Training.csv")

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 hemmorrhoids(piles)':28,'Heart
attack':29,'Varicose veins':30,'Hypothyroidism':31,

'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthritis':34,'Arthritis':35,

'(vertigo) Paroymsal Positional Vertigo':36,'Acne':37,'Urinary tract
infection':38,'Psoriasis':39,

'Impetigo':40}} ,inplace=True)

# print(df.head())

X= df[11]

y = df[["prognosis"]]

np.ravel(y)

```

```

# print(y)

# TRAINING DATA tr -----

tr=pd.read_csv("Testing.csv")

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,'Dimorphic hemmorhoids(piles)':28,'Heart
attack':29,'Varicose veins':30,'Hypothyroidism':31,

'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthritis':34,'Arthritis':35,

'(vertigo) Paroymsal Positional Vertigo':36,'Acne':37,'Urinary tract
infection':38,'Psoriasis':39,

'Impetigo':40}} ,inplace=True)

X_test= tr[11]

y_test = tr[["prognosis"]]

np.ravel(y_test)

# -----

def check_patient_name():

```

```

from tkinter import messagebox

if not Name.get():

    messagebox.showinfo("Error", "Please enter the patient's name.")

    return False

    return True

def DecisionTree():

    if not check_patient_name():

        return

    from sklearn import tree

    clf3 = tree.DecisionTreeClassifier() # empty model of the decision tree

    clf3 = clf3.fit(X,y)

    # calculating accuracy-----

    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(),Symptom5.get()]

    for k in range(0,len(11)):

        # print (k,)

```

```

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():

    if not check_patient_name():

        return

```

```

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(),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'):

t2.delete("1.0", END)

t2.insert(END, disease[a])

else:

t2.delete("1.0", END)

t2.insert(END, "Not Found")

def NaiveBayes():

if not check_patient_name():

return

from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()

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

# calculating accuracy-----

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(),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'):

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='skyblue')

# entry variables

Symptom1 = StringVar()

Symptom1.set(None)

Symptom2 = StringVar()

Symptom2.set(None)

Symptom3 = StringVar()

Symptom3.set(None)

Symptom4 = StringVar()

Symptom4.set(None)

Symptom5 = StringVar()

Symptom5.set(None)

Name = StringVar()

# Heading

w2 = Label(root, justify=LEFT, text="Disease Predictor using Machine Learning",

fg="black", bg="skyblue")

w2.config(font=("Elephant", 30))

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

w2 = Label(root, justify=LEFT, fg="white", bg="skyblue")

w2.config(font=("Aharoni", 30))
```



```

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

# labels

NameLb = Label(root, text="Name of the Patient", fg="yellow", bg="black")

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

S1Lb = Label(root, text="Symptom 1", fg="yellow", bg="black")

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

S2Lb = Label(root, text="Symptom 2", fg="yellow", bg="black")

S2Lb.grid(row=8, column=0, pady=10, sticky=W)

S3Lb = Label(root, text="Symptom 3", fg="yellow", bg="black")

S3Lb.grid(row=9, column=0, pady=10, sticky=W)

S4Lb = Label(root, text="Symptom 4", fg="yellow", bg="black")

S4Lb.grid(row=10, column=0, pady=10, sticky=W)

S5Lb = Label(root, text="Symptom 5", fg="yellow", bg="black")

S5Lb.grid(row=11, column=0, pady=10, sticky=W)

lrLb = Label(root, text="DecisionTree", fg="white", bg="red")

lrLb.grid(row=15, column=0, pady=10, sticky=W)

destreeLb = Label(root, text="RandomForest", fg="white", bg="red")

destreeLb.grid(row=17, column=0, pady=10, sticky=W)

ranfLb = Label(root, text="NaiveBayes", fg="white", bg="red")

ranfLb.grid(row=19, column=0, pady=10, sticky=W)

```

```

# entries

OPTIONS = sorted(11)

NameEn = Entry(root, textvariable=Name)

NameEn.grid(row=6, column=1)

# Use ttk.Combobox instead of OptionMenu for scrolling

S1En = ttk.Combobox(root, textvariable=Symptom1, values=OPTIONS)

S1En.grid(row=7, column=1)

S2En = ttk.Combobox(root, textvariable=Symptom2, values=OPTIONS)

S2En.grid(row=8, column=1)

S3En = ttk.Combobox(root, textvariable=Symptom3, values=OPTIONS)

S3En.grid(row=9, column=1)

S4En = ttk.Combobox(root, textvariable=Symptom4, values=OPTIONS)

S4En.grid(row=10, column=1)

S5En = ttk.Combobox(root, textvariable=Symptom5, values=OPTIONS)

S5En.grid(row=11, column=1)

dst = Button(root, text="DecisionTree",
command=DecisionTree,bg="green",fg="yellow")

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

rnf = Button(root, text="Randomforest", command=randomforest,bg="green",fg="yellow")

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

lr = Button(root, text="NaiveBayes", command=NaiveBayes,bg="green",fg="yellow")

```

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

#textfileds

t1 = Text(root, height=1, width=40,bg="orange",fg="black")

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

t2 = Text(root, height=1, width=40,bg="orange",fg="black")

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

t3 = Text(root, height=1, width=40,bg="orange",fg="black")

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

root.mainloop()
```

5.2 OUTPUT SCREENS:

5.2.1 Prediction module:

Disease Predictor using Machine Learning

Name of the Patient:

Symptom 1:

Symptom 2:

Symptom 3:

Symptom 4:

Symptom 5:

DecisionTree

RandomForest

NaiveBayes

DecisionTree

RandomForest

NaiveBayes

5.2.2 Dropdown box for symptoms:

Disease Predictor using Machine Learning

Name of the Patient:

Symptom 1:

Symptom 2:

Symptom 3:

Symptom 4:

Symptom 5:

DecisionTree

RandomForest

NaiveBayes

DecisionTree

RandomForest

NaiveBayes

5.2.3 Prediction Module with result:

The screenshot shows a web application titled "Disease Predictor using Machine Learning". On the left, there are labels for "Name of the Patient", "Symptom 1", "Symptom 2", "Symptom 3", "Symptom 4", and "Symptom 5". Below these are three red buttons labeled "DecisionTree", "RandomForest", and "NaiveBayes". In the center, there is a text input field containing "Johnathan" and five dropdown menus with the following selected values: "chest_pain", "fast_heart_rate", "irritability", "dizziness", and "headache". To the right of these inputs are three green buttons labeled "DecisionTree", "RandomForest", and "NaiveBayes". Below the dropdown menus, there are three orange horizontal bars, each displaying the word "Hypertension" in black text.

5.2.4 prediction module showing alert pop-up when name is not provided:

The screenshot shows the same web application as in 5.2.3, but with an error alert. The "Name of the Patient" input field is empty. The dropdown menus have the following selected values: "acute_liver_failure", "None", "None", and "None". The "DecisionTree", "RandomForest", and "NaiveBayes" buttons are visible on the right. An "Error" dialog box is displayed in the center, with a blue information icon and the text "Please enter the patient's name." and an "OK" button.

6.CONCLUSION AND FUTURE SCOPE:

CONCLUSION:

our disease prediction system harnessing machine learning algorithms such as Naive Bayes, Decision Tree, and Random Forest demonstrates promising potential in accurately diagnosing diseases based on user-input symptoms. By allowing users to input their symptoms, our system employs these algorithms to analyze the data and predict the most likely disease. Through rigorous testing and validation, we have observed reliable performance across various scenarios, showcasing the system's effectiveness in aiding healthcare professionals in early diagnosis and treatment planning. With further refinement and integration of additional features, our system has the capacity to significantly contribute to enhancing healthcare outcomes and empowering individuals to proactively manage their health.

The reason to choose these three algorithms are:

- They are effective, if the training data is large.
- A single dataset can be provided as an input to all these 3 algorithms with minimal or no modification.
- A common scalar can be used to normalize the input provided to these 3 algorithms.

FUTURE SCOPE:

.To enhance the functionality of the prediction engine providing the details of 5 nearest hospitals or medical facilities to the user input location.

- Provide a user account which allows the user to keep track of their medical test data and get suggestions or support to meet the right specialists or the tests to be taken
- Provide admin controls to upload, delete the dataset which will be used to train the model

.Automate the process of training the model and extracting pickle files of the trained models which will be consumed by the API's to predict the disease.

- Mail the detailed report of the prediction engine results along with the information of 5 nearest medical facilities details having location and contact information

7.BIBLIOGRAPHY

[1] Priyanka Sonar, Prof. K. JayaMalini, “Diabetes Prediction Using Different Machine Learning Approaches”, Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019) IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4

[2] Samrat Kumar Dey, Ashraf Hossain, Md. Mahbubur Rahman, “Implementation of a Web Application to Predict Diabetes Disease: An Approach Using Machine Learning Algorithm”, 2018 21st International Conference of Computer and Information Technology (ICCIT)

[3] Deeraj Shetty, Kishor Rit, Sohail Shaikh, Nikita Patil, “Diabetes Disease Prediction Using Data Mining”, 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)

[4] Zhilbert Tafa, Nerxhivane Pervetica, Bertran Karahoda, “An Intelligent System for Diabetes Prediction”, 4thMediterranean Conference on Embedded Computing MECO – 2015 Budva, Montenegro