CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

With the advancement in technology, Machine Learning is becoming more popular and commonly used technology by industry experts for solving problems faced in real life. Machine Learning is the scientific study of algorithms and statistical models that computer use to perform a specific task without using explicit instructions, relying on patterns and inference instead. Machine Learning is also used by the healthcare industry to bring advancement in their techniques so that they can provide better services to their patients. The disease prediction system predicts diseases based on patient’s symptoms.

1.1 Problem statement

Most people in our digital age are prone to ailments as a result of a lack of good food, adequate sleep, and daily exercise. It is critical to know if we are afflicted with sickness at an early stage rather than discovering it later. As a result, illness prediction systems play a significant role since they predict diseases based on symptoms. This disease prediction system employs the Machine Learning algorithms. This approach also recommends drug that are widely used to treat the illness.

1.2 Research Objective

There is a require to ground work and evolve a system that will enable end users to predict diseases without having to visit a physician or doctor for diagnosis. To identify the patient sickness by analyzing symptoms and suggest the drug by using various machine learning algorithms. Machine learning can improve the accuracy of predictions

1.3 Project scope and limitations

The project aim is to predict the disease and suggest the drug based on the symptoms. Increases the efficiency, accuracy for patients to help them in future and manages the information related to diseases.

Limitations:

- Disease Prediction System Provides only possible outcomes it does not guarantee that it will predict the disease correctly but it has significantly higher accuracy for predicting possible diseases.

- If we use large amount of data set related to diseases and their symptoms is very time consuming and it cannot be dine within 1 or 2 years it requires multiple years to collect those data sets and trained the system using those data searches.
- This system predicts the disease and suggests the drug according to the disease prediction but it will not recommend the food.

CHAPTER 2

BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Vitamin Deficiency Detection Using Image Processing and Artificial Intelligence

2.1.1 Introduction

Vitamin deficiency is a problem that affects over two billion people around the world. The WHO said that one in three children do not get enough vitamins. Vitamin deficiency is a global problem that affects over two billion people around the world. The WHO said that one in three children do not get vitamin. 33% of children under the age of five have deficiency of vitamin A. This deficiency causes low immunity and night blindness. Vitamin deficiencies affect all ages and frequently co-exist with mineral (zinc, iron, iodine) deficiencies. The groups most susceptible to vitamin deficiencies are pregnant women, children, because of their needs for these compounds and susceptibilities to their absence. Most common deficiencies relate to vitamin A, vitamin B, folate and vitamin D. Supplementation programs have made diseases like scurvy and pellagra rare. Many of deficiencies are preventable through consumption of a healthy diet containing diverse foods, as well as food fortification and supplementation, where needed. Most vitamin and mineral deficiencies can be picked up with a blood test, like a venous blood test and finger-prick blood test. In venous blood test a trained professional will use a needle to puncture a vein, usually in your arm, to collect a blood sample and in finger-prick blood test using lancet, you can prick your own finger and collect blood sample. In hospitals these blood tests can be done or we can also order home vitamin and mineral test kits online and do it our self.

2.1.2 Merits, Demerits and challenges

Merits

- Blood samples are not required
- No need to visit a physician

Demerits

- This system will not work for all deficiencies
- It will capture only 4 types of Image symptoms

2.1.3 Implementation of Vitamin Deficiency Detection Using Image Processing and Artificial Intelligence

The System is capable to diagnosis the vitamin deficiency spectrum from the images of user's tongue, nails, lips and eyes using Artificial Intelligence. Application uses the Neural Network Training to detect symptoms and Natural Language Processing to extract features. Fuzzy logic algorithm is used to specify the type of deficiency. After specifying visual symptoms through pathological research, a Tensor Flow classifier trained using number of labeled images of segmented symptoms. One more layer of decision making algorithm shows a list of nutrients as well as suited medications and supplementary products.

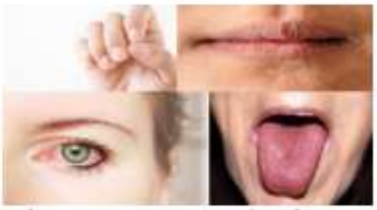


Fig 2.1 Sample Dataset Images

The system is an innovative approach that allows self-diagnosis in a short span of time without any blood sample. The accuracy of proposed system can be improved by adding more data with contribution from Doctors, medical researchers and experts. The proposed solution's capabilities are not limited to vitamin deficiencies only, but they can be extended to detect other health problems.

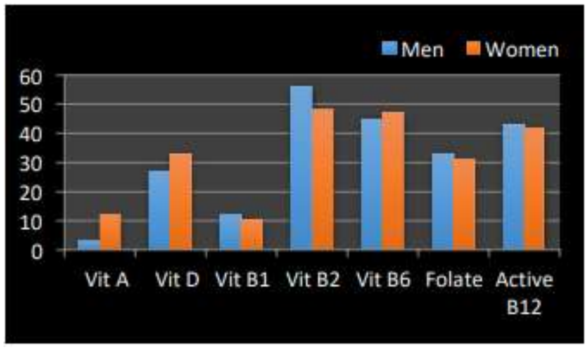


Fig 2.2: Pervasiveness of vitamin deficiencies in an urban adult population

2.2 Vitamin Deficiency and Food Recommendation System Using Machine Learning

2.2.1 Introduction

For people, balanced diet is a vital part of a healthy lifestyle. A healthy lifestyle requires a balanced diet as well as regular physical activity. Nutrition and health are frequently disregarded nowadays. The majority of people have diabetes, heart disease, cancer, stroke, and other diseases. The diseases are virtually always linked to poor dietary habits. So, in order to stay healthy, our bodies require nutrition, and food provides these necessary elements. Vitamins, minerals, protein, healthy fats, proteins, carbs, and fiber are often found in a healthy, balanced diet. Plant foods with a moderate amount of animal products make up a healthy dietary pyramid. Vegetables, grains, fruits, oils and sweets, dairy, meat, and legumes are all included. In most cases, a person is unaware of the major causes of deficiency or illness. Excesses of several critical nutrients, such as calcium, proteins, and vitamins, and how to restore equilibrium with a well-balanced diet People can live a healthier lifestyle thanks to technological advancements. The goal of this project is to create a system that will advise users on proper nutrition intake based on their BMI and grocery data preferences. Underweight, healthy weight, overweight, and obese are all weight status categories calculated by BMI. Seasonal foods, user-in treated foods, plant foods, and animal products are all included in grocery data.

2.2.2 Merits, Demerits and challenges

Merits

- Automates process of vitamin deficiency detection and food recommendation
- Previous datasets are used to training and testing.
- Accuracy of model is improved compare to existing methods.

Demerits

- Not suggests the medicine to the disease.
- Blood samples repots are required.

Challenges

- Problems are encountered in relation to the User.
- Algorithm related difficulties encountered.
- Difficulties in describing the link between various types of dietary components.

2.2.3 Implementation of Vitamin Deficiency and Food Recommendation System Using Machine Learning

A website that recommends foods and predicts vitamin deficit and implemented prediction by using information such as vitamins and their deficiency. The dataset development of food items dependent on the vitamin deficit is the first step in training the system. The prediction of various meal recommendations based on which vitamins are required for each type of deficiency. Following clustering, the closest food items that are most suited for the proper diet are suggested using the Random Forest classifier. Our diet advice method simply allows customers to receive the healthy food they want based on vitamin insufficiency.

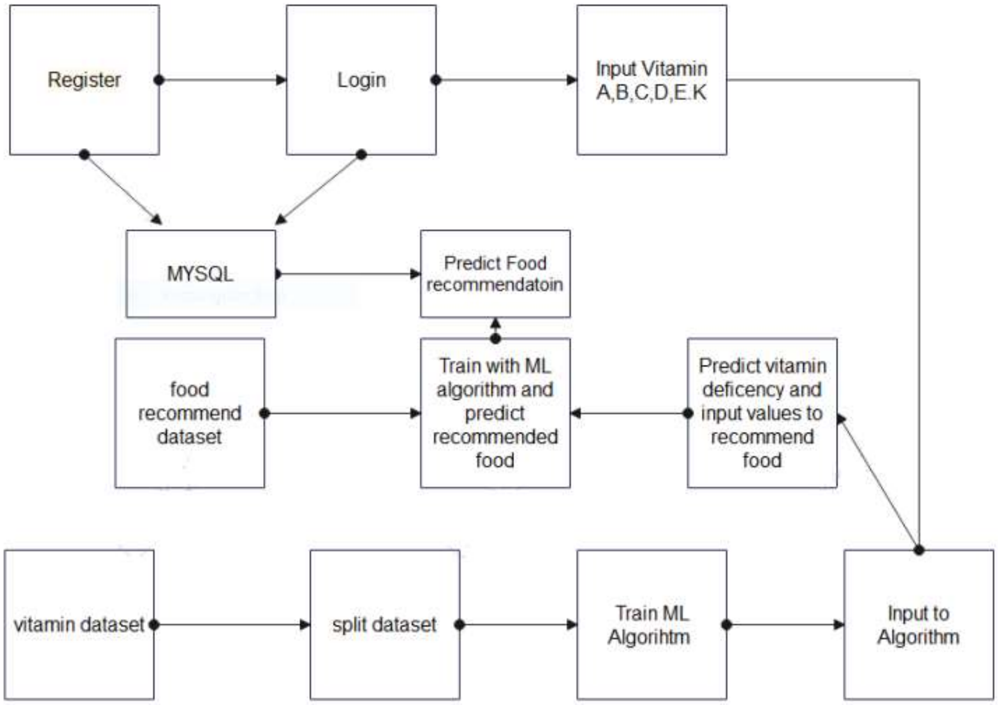

Register
Login
Input Vitamin A,B,C,D,E.K
MYSQL
Predict Food recommendatoin
food recommend dataset
Train with ML algorithm and predict recommended food
Predict vitamin deficency and input values to recommend food
vitamin dataset
split dataset
Train ML Algorihtm
Input to Algorithm

Fig 2.3 Architecture of the system

In this stage new data is taken as input and trained models are loaded using pickle and then values are preprocessed and passed to predict function to find out result which is showed on web application

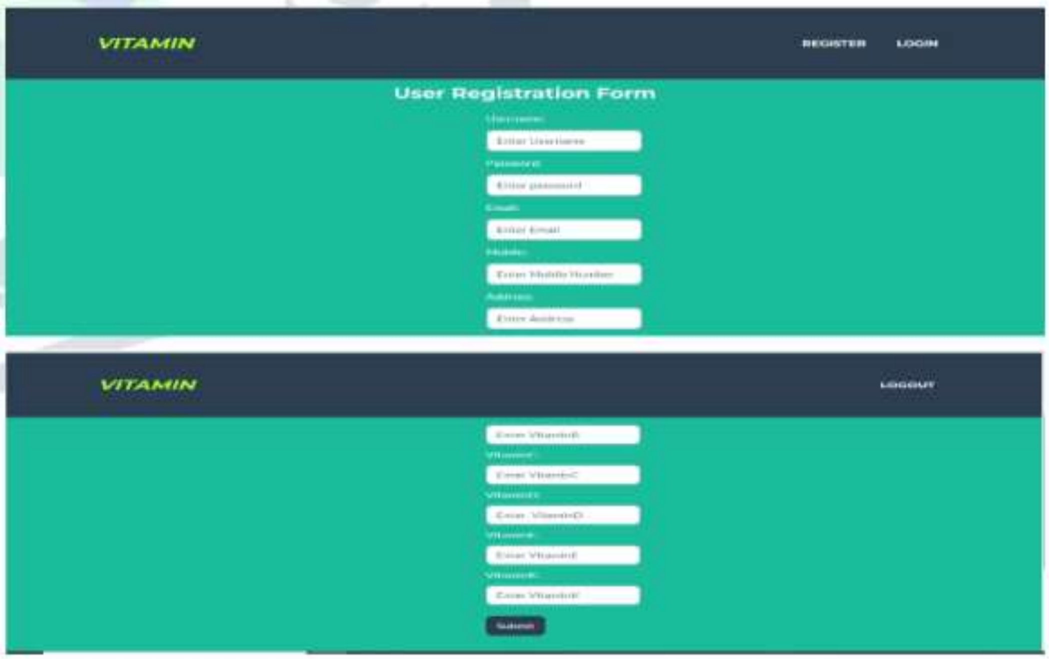


Fig 2.4 Upload vitamins range page

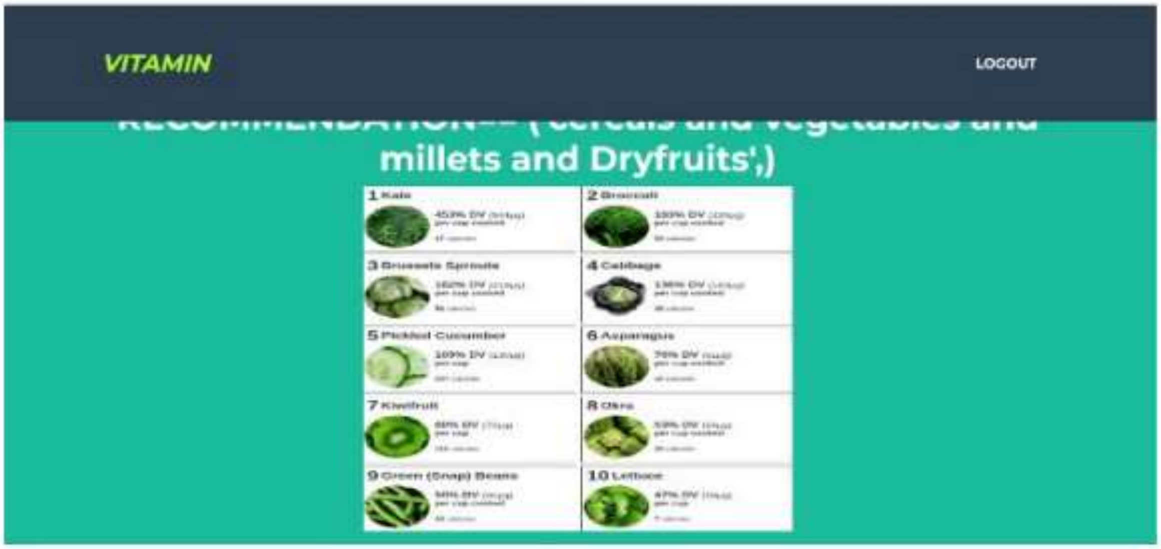


Fig 2.5 Food recommendation page

2.3 Heart Disease Prediction system using Machine Learning

2.3.1 Introduction

Heart is an important organ of the human body. It pumps blood to every part of our anatomy. If it fails to function correctly, then the brain and various other organs will stop working, and within few minutes, the person will die. Change in lifestyle, work related stress and bad food habits contribute to the increase in the rate of several heart-related diseases. Heart diseases have emerged as one of the most prominent causes of death all around the world. According to World Health Organization, heart related diseases are responsible for taking 17.7 million lives every year, 31% of all global deaths. In India too, heart-related diseases have become the leading cause of mortality. Heart diseases have killed 1.7 million Indians in 2016, according to the 2016 Global Burden of Disease Report, released on September 15, 2017.

Medical organizations, all around the world, collect data on various health-related issues. These data can be exploited using various machine learning techniques to gain useful insights. But the data collected is very massive and, many times, this data can be very noisy

2.3.2 Merits, Demerits and challenges

Merits

- Increased accuracy for effective heart disease diagnosis.
- Handles roughest (enormous) amount of data using random forest algorithm and feature selection.
- Reduce the time complexity of doctors.
- Cost effective for patients.

Demerits

- This system works for only heart diseases
- It does not suggests the drugs
- Prediction of cardiovascular disease results is not accurate.
- Data mining techniques does not help to provide effective decision making.
- Cannot handle enormous datasets for patient records.

Challenges

- A major challenge facing healthcare organization is the provision of quality services at affordable cost.
- Quality service implies diagnosing patients correctly and administering treatments that are effective.
- Algorithm related difficulties encountered.

2.3.3 Implementation of Heart Disease Prediction using Machine Learning

This project aims to know whether the patient has heart disease the training set and test sets. After preprocessing the data, the data classification technique namely support vector machine, artificial neural network, random forest were applied. The project involved analysis of the heart disease patient dataset with proper data processing. Then, 3 models were trained and tested with maximum scores as follows:

1. Support Vector Classifier: 84.0 %
2. Neural Network: 83.5 %
3. Random Forest Classifier: 80.0 %

This project provides the deep insight into machine learning techniques for classification of heart diseases. The role of classifier is crucial in healthcare industry so that the results can be used for predicting the treatment which can be provided to patients. The existing techniques are studied and compared for finding the efficient and accurate systems. Machine learning techniques significantly improves accuracy of cardiovascular risk prediction through which patients can be identified during an early stage of disease and can be benefitted by preventive treatment. It can be concluded that there is a huge scope for machine learning algorithms in predicting cardiovascular diseases or heart related diseases. Each of the above-mentioned algorithms have performed extremely well in some cases but poorly in some other cases.

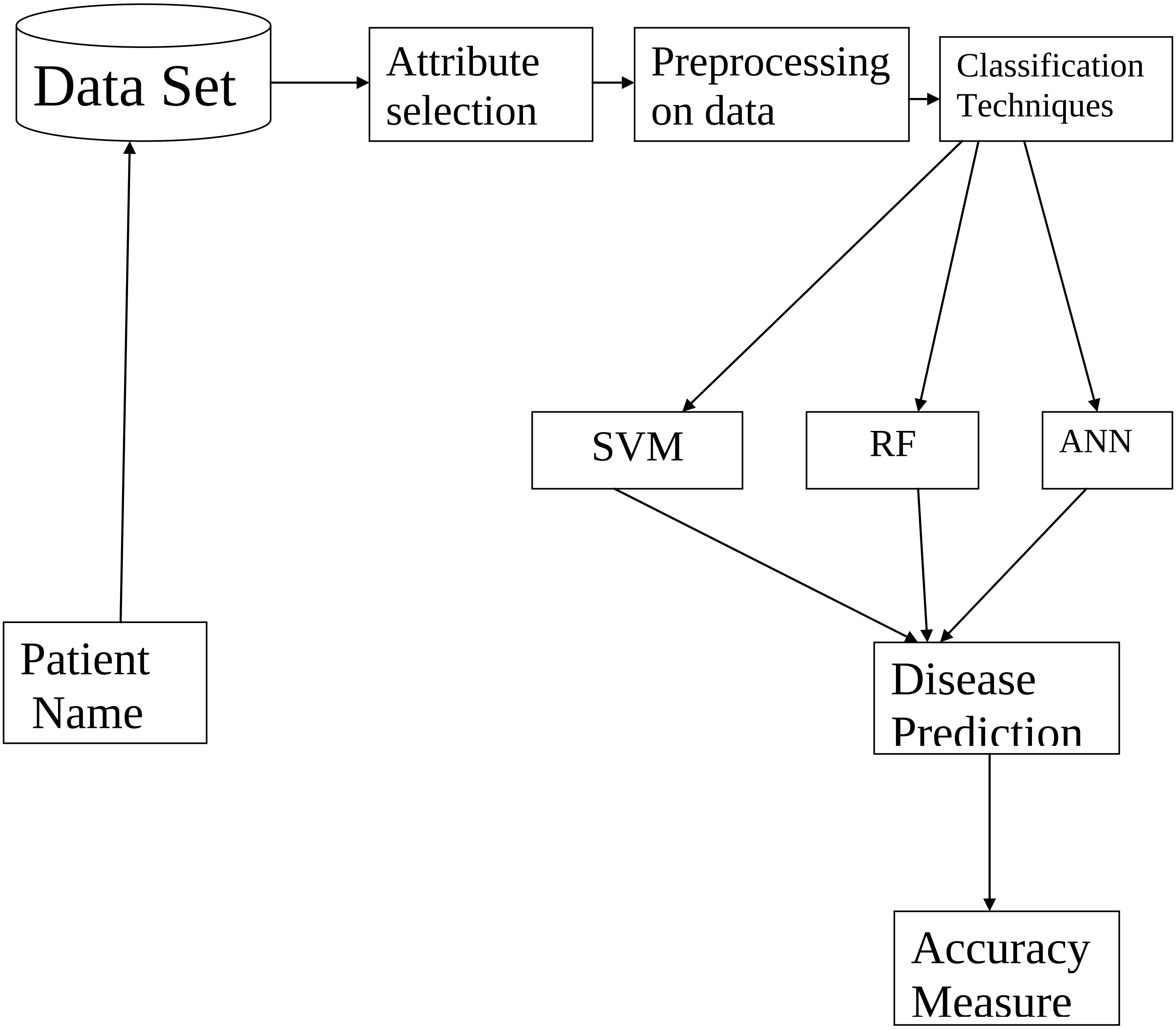


Fig 2.6 Architecture of the system

2.4 SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 3

PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of Proposed Model

Disease Prediction using Machine learning is the System that is used to predict the diseases from the Symptoms which are given by the Patients or any user.The System Processes the Symptoms Provided by the user as input and gives the output and gives the output as the Probability of the disease. Naïve Bayes classifier is used in the prediction of the disease which is a supervised machine learning algorithm. The probability of the disease is calculated by the Naïve Bayes algorithm. With an increase in biomedical and healthcare data, accurate analysis of medical data benefits early disease detection and patient care. By using linear regression and decision tree we are predicting diseases like Diabetes, Malaria, Jaundice, Dengue, and Tuberculosis

3.2 Algorithms Used in the Proposed Model

The machine learning algorithms were used in this project is

- Naive Bayes
- Random Forest
- Decision Tree

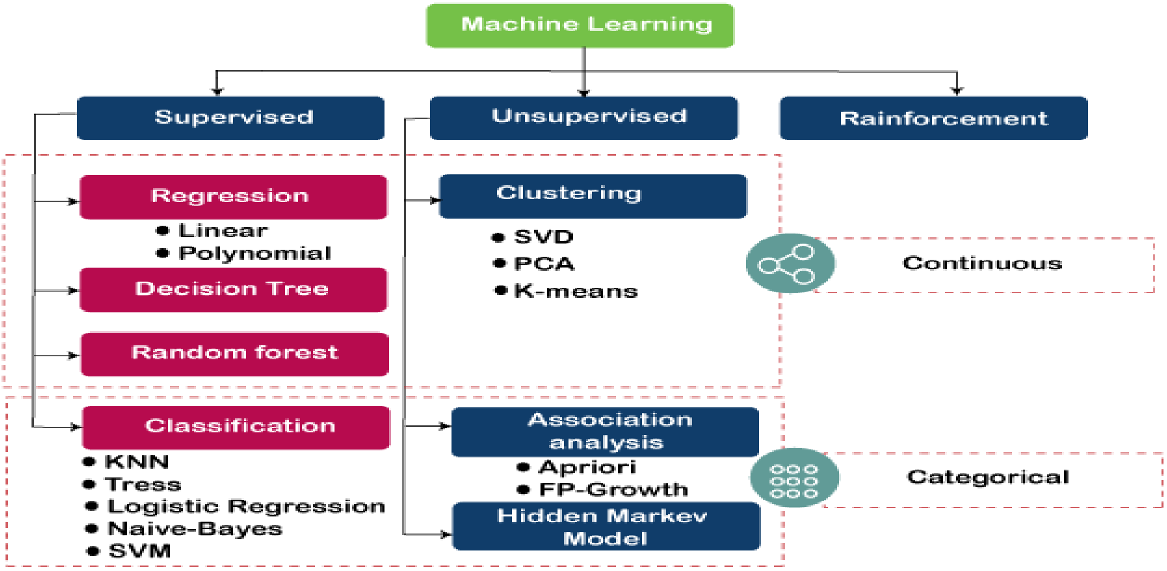


Fig 3.1 Machine Learning Algorithms

Random Forest

- 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.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting.

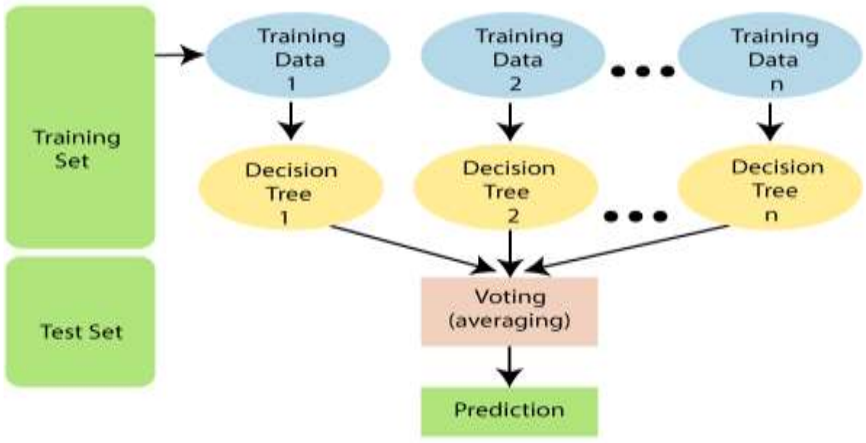


Fig 3.2 Working of Random Forest

Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems.

In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node**. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub trees.

Different Types of Decision Tree are:

Categorical Variable Decision Tree: These types of decision tree works on thewhole value i.e. suppose we have to choose between 0 and 1 or something which istaken as a whole.

Continuous Variable Decision tree: These types of the decision tree works on the continuous values that's why they are known as the continuous variable decision tree.

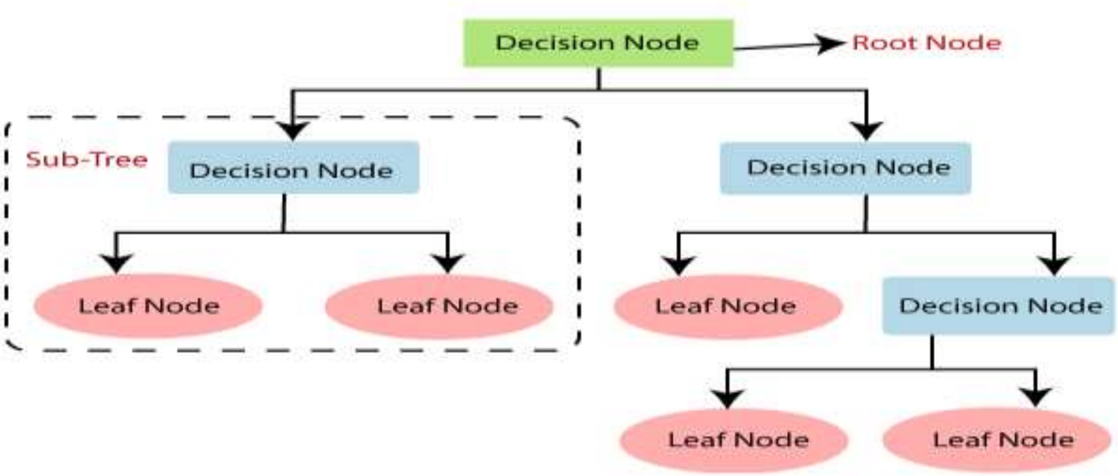


Fig 3.3 Structure of Decision Tree

Naive Bayes

- Naïve Bayes algorithm is a **supervised learning algorithm**, which is based on **Bayes theorem** and used for solving **classification problems**.
- It is mainly used in **text classification** that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can **make quick predictions**.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of

an object.

- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.
- There are three different types of naive bayes classifier:
- Multinomial Naive Bayes: This is mainly used for the classification of big filesand documents by dividing them into different categories as weather a file or adocument is of sports category, politics, technology or something else. This methodof classification is very widely used in Machine Learning algorithms.
- Bernoulli Naive Bayes: This method of classification is very similar to the multinomial naive bayes but the result of this type of naive bayes is only in either yes or no.
- Gaussian Naive Bayes: This type of naive bayes is used for the prediction of thecontinuous values while the other two types were used for the discrete value prediction.

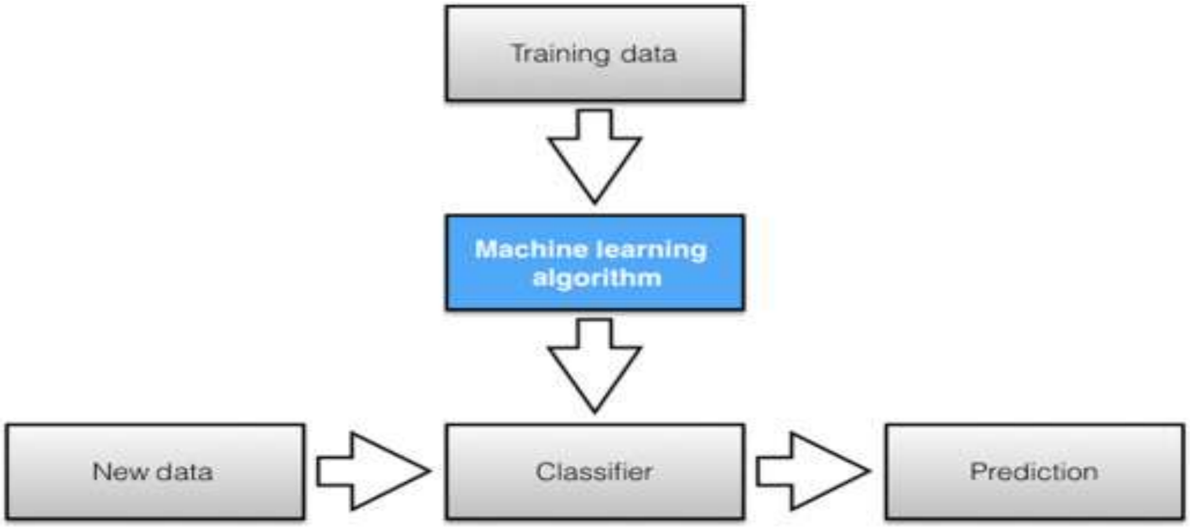


Fig 3.4 Flow Chart of Naïve Bayes Classifier

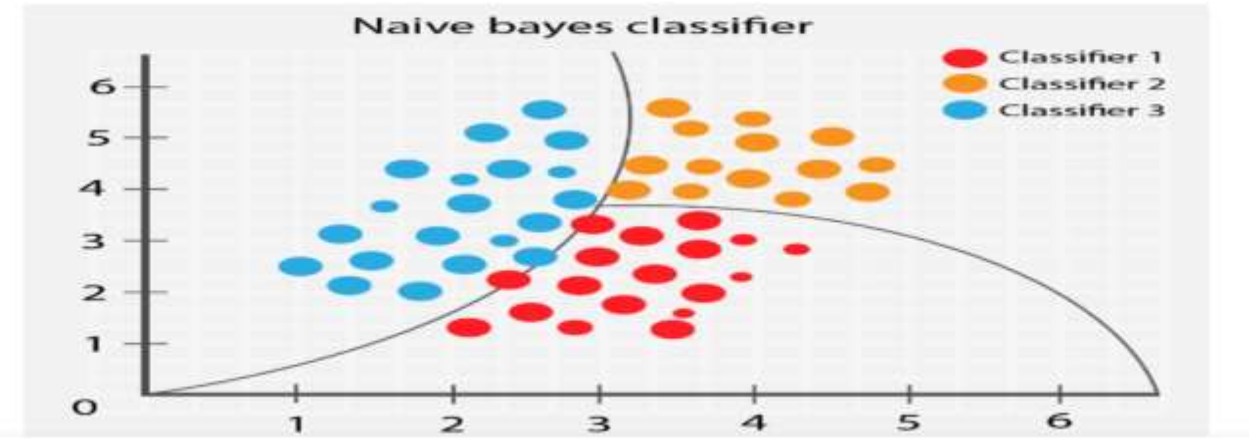


Fig 3.5 Naïve Bayes Classifier

3.3 Designing

3.3.1 UML Diagram

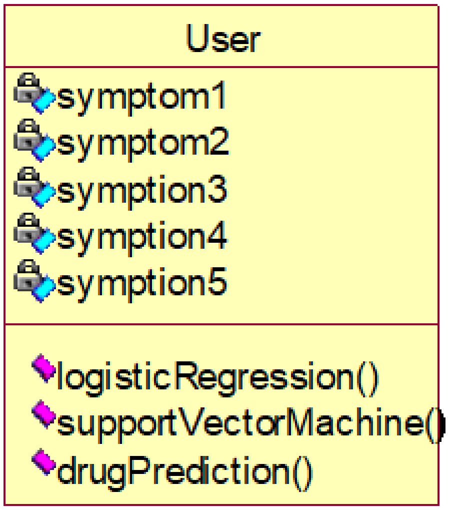


Fig 3.6 Class Diagram

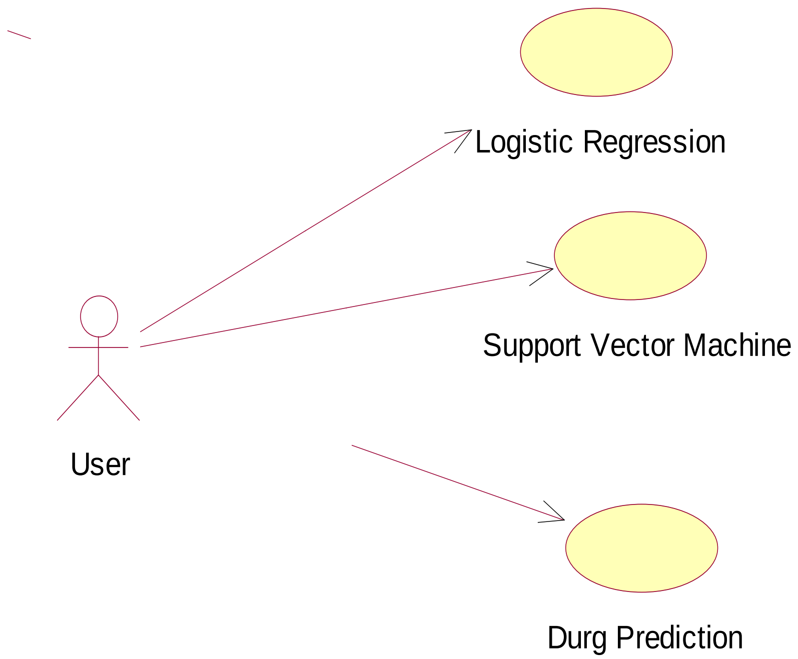


Fig 3.7 Use Case Diagram

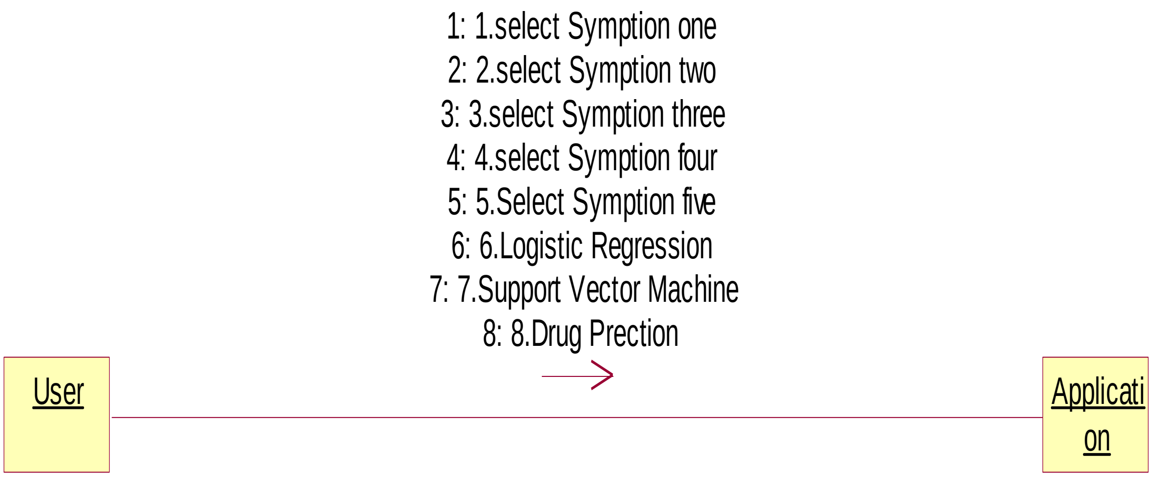


Fig 3.8 Collaboration Diagram

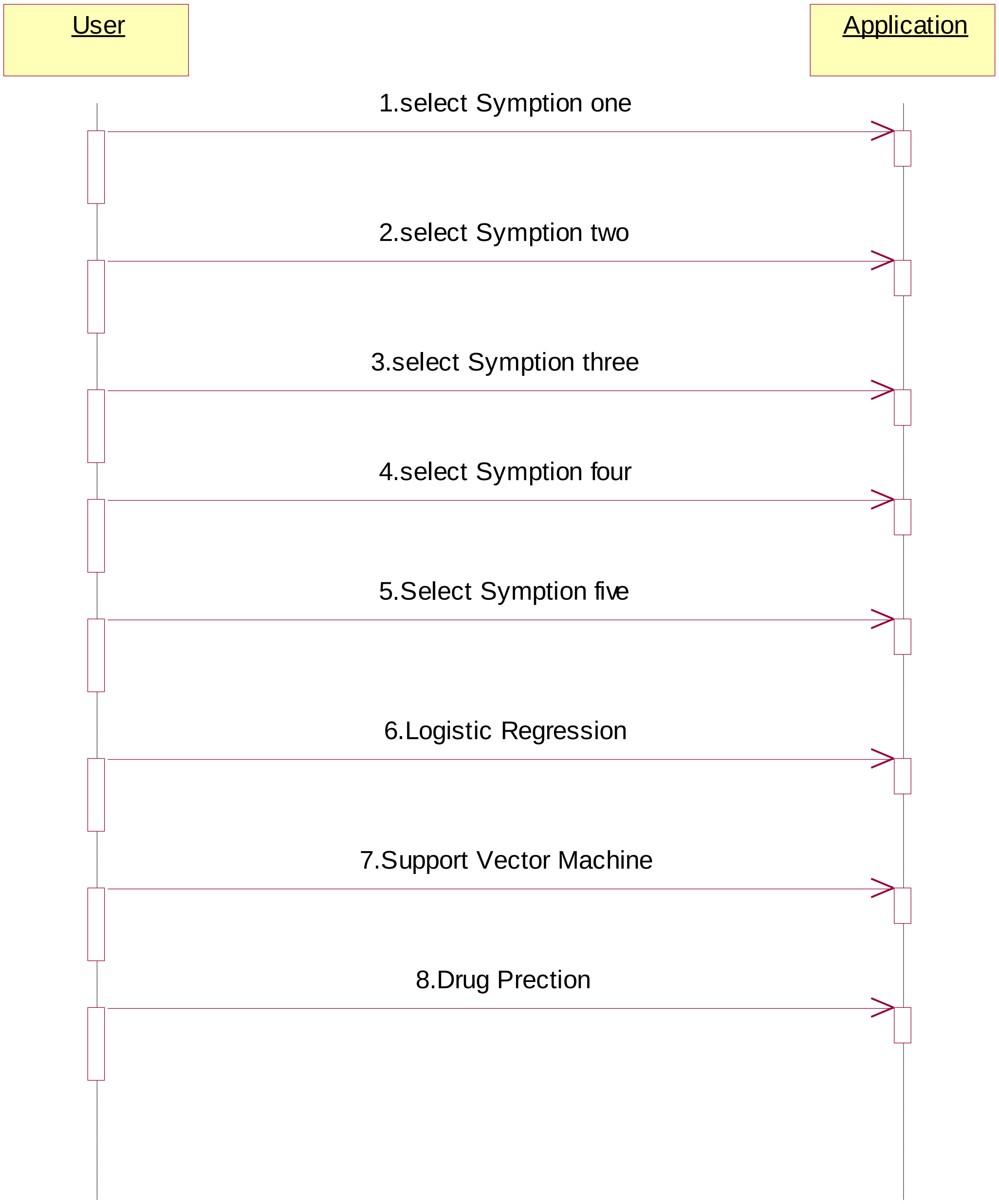


Fig 3.9 Sequence Diagram

3.3.2 SYSTEM SPECIFICATIONS

HARDWARE REQUIREMENTS:

- **System** **:** Pentium IV 2.4 GHz.
- **Hard Disk** **:** 40 GB.
- **Floppy Drive** **:** 1.44 Mb.
- **Monitor** : 14’ Colour Monitor.
- **Mouse** **:** Optical Mouse.
- **Ram** **:** 512 Mb.

SOFTWARE REQUIREMENTS:

- **Operating system** **:** Windows 7 Ultimate.
- **Coding Language** **:** Python.
- **Front-End** **:** Python.
- **Designing** **:** Tkinter

Packages:

1. **Sklearn** : This stands for Scikit learn and is built on the Scipy package. It is the primarypackage being used in this project. It is used for providing interface for supervised andunsupervised learning algorithms. Following groups of models are provided by sklearnClustering, Cross Validation, Datasets, Dimensionality Reduction, Ensemble methods,Feature extraction, Feature selection, Parameter Tuning, Manifold Learning, SupervisedModels.

2. **Numpy** : It is a library for the Python programming language, adding support for multi-dimensional arrays and matrices, along with a large collection of high-level mathematicalfunctions to operate on these arrays. It provides functions for Array Objects,

Routines,Constants, Universal Functions, Packaging etc. In this project it is used for performingmulti-dimensional array operations.

3. **Pandas** : This library is used to provide high-performance, easy-to-use data structures anddata analysis tools for the Python programming language. It provides functionalities liketable manipulations, creating plots, calculate summary statistics, reshape tables, combinedata from tables, handle times series data, manipulate textual data etc. In this project it isused for reading csv files, comparing null and alternate hypothesis etc.

4. **Matplotlib** : It is a library for creating static, animated, and interactive visualizations inPython programming language. In this project it is used for creating simple plots, sub-plots and its object is used alongside with the seaborn object to employ certain functionssuch as show, grid etc. A %matplotlib inline function is also used for providing moreconcise plots right below the cells that create that plot.

3.3.3 INPUT AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

3.3.4 REQUIREMENT ANALYSIS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENT SPECIFICATION

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application the following are the Software Requirements:

- Python
- Django

Operating Systems supported

- Windows 7
- Windows XP
- Windows 8

Technologies and Languages used to Develop

- Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor: Pentium IV or higher
- RAM: 256 MB
- Space on Hard Disk: minimum 512MB

Django Framework

The Django is a high-level Python framework that helps in the rapid development and clean, pragmatic design, django makes it easier to build applications more quickly, efficiently and with less code. Django is used for creating the User Interface (UI) for the application. The UI created by Django is easy to use so that the person which are from the non-technical field can also use the application for the prediction of disease without going anywhere any saving time and money.

3.4 Stepwise Implementation and code

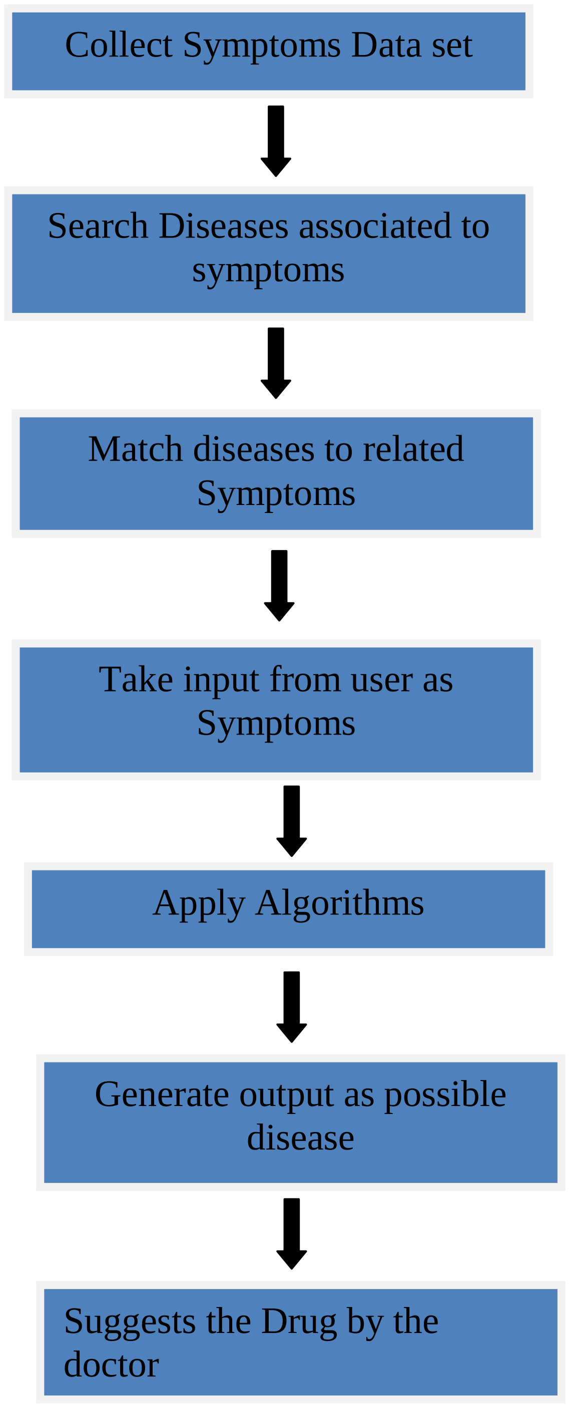

Collect Symptoms Data set
Search Diseases associated to symptoms
Match diseases to related Symptoms
Take input from user as Symptoms
Apply Algorithms
Generate output as possible disease
Suggests the Drug by the doctor

Fig 3.10 Stepwise Implementation

CODE

```
from tkinter import *
import numpy as np
import pandas as pd
# from gui_stuff import *

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_bowel_movements','pain_in_anal_region','bloody_stool',
'irritation_in_anus','neck_pain','dizziness','cramps','bruising','obesity','swollen_legs',
'swollen_blood_vessels','puffy_face_and_eyes','enlarged_thyroid','brittle_nails',
'swollen_extremeties','excessive_hunger','extra_marital_contacts','drying_and_tingling_lips',
'slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','stiff_neck','swelling_joints',
'movement_stiffness','spinning_movements','loss_of_balance','unsteadiness',
'weakness_of_one_body_side','loss_of_smell','bladder_discomfort','foul_smell_of urine',
'continuous_feel_of_urine','passage_of_gases','internal_itching','toxic_look_(typhos)',
'depression','irritability','muscle_pain','altered_sensorium','red_spots_over_body','belly_pain',
'abnormal_menstruation','dischromic
_patches','watering_from_eyes','increased_appetite','polyuria','family_history','mucoid_sputum',
'rusty_sputum','lack_of_concentration','visual_disturbances','receiving_blood_transfusion',
'receiving_unsterile_injections','coma','stomach_bleeding','distention_of_abdomen',
'history_of_alcohol_consumption','fluid_overload','blood_in_sputum','prominent_veins_on_calf',
'palpitations','painful_walking','pus_filled_pimples','blackheads','scurring','skin_peeling',
'silver_like_dusting','small_dents_in_nails','inflammatory_nails','blister','red_sore_around_nose',
'yellow_crust_ooze']

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','Osteoarthristis',
'Arthritis','(vertigo) Paroymsal  Positional Vertigo','Acne','Urinary tract infection','Psoriasis',
'Impetigo']
```

```
l2=[]
for x in range(0,len(l1)):
   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 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,'Osteoarthristis':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[l1]

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,'Osteoarthristis':34,'Arthritis':35,
'(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract
infection':38,'Psoriasis':39,
'Impetigo':40}},inplace=True)
```

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

def DecisionTree():

    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(l1)):
        # 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():
    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():
    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='blue')
```

```
# 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="white", bg="blue")
w2.config(font=("Elephant", 30))
w2.grid(row=1, column=0, columnspan=2, padx=100)
w2 = Label(root, justify=LEFT, text="A Project by J.Mamatha,V.Vaishnavi,P.Deepthi", fg="white", bg="blue")
w2.config(font=("Aharoni", 20))
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(l1)

NameEn = Entry(root, textvariable=Name)
NameEn.grid(row=6, column=1)

S1En = OptionMenu(root, Symptom1,*OPTIONS)
S1En.grid(row=7, column=1)

S2En = OptionMenu(root, Symptom2,*OPTIONS)
S2En.grid(row=8, column=1)

S3En = OptionMenu(root, Symptom3,*OPTIONS)
S3En.grid(row=9, column=1)

S4En = OptionMenu(root, Symptom4,*OPTIONS)
S4En.grid(row=10, column=1)

S5En = OptionMenu(root, Symptom5,*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()
```

Dataset:

```
    itching,
    skin_rash
    nodal_skin_eruptions
```

continuous_sneezing

shivering

chills

joint_pain

stomach_pain

acidity

ulcers_on_tongue

muscle_wasting

vomiting

burning_micturition

spotting_ urination

fatigue

weight_gain

anxiety

cold_hands_and_feets

mood_swings

weight_loss

restlessness

lethargy

patches_in_throat

irregular_sugar_level

cough

high_fever

sunken_eyes

breathlessness

sweating

dehydration

indigestion

headache

yellowish_skin

dark_urine

nausea

loss_of_appetite

pain_behind_the_eyes

back_pain

constipation

abdominal_pain

diarrhoea

mild_fever

yellow_urine

yellowing_of_eyes

acute_liver_failure

fluid_overload

swelling_of_stomach

swelled_lymph_nodes

malaise

blurred_and_distorted_vision

phlegm

throat_irritation

redness_of_eyes

sinus_pressure

runny_nose

congestion

chest_pain

weakness_in_limbs

fast_heart_rate

pain_during_bowel_movements

pain_in_anal_region

bloody_stool

irritation_in_anus

neck_pain

dizziness

cramps

bruising

obesity

swollen_legs

swollen_blood_vessels

puffy_face_and_eyes

enlarged_thyroid

brittle_nails

swollen_extremeties

excessive_hunger

extra_marital_contacts

drying_and_tingling_lips

slurred_speech

knee_pain

hip_joint_pain

muscle_weakness

stiff_neck

swelling_joints

movement_stiffness

spinning_movements

loss_of_balance

unsteadiness

weakness_of_one_body_side

loss_of_smell

bladder_discomfort

foul_smell_of urine

continuous_feel_of_urine

passage_of_gases

internal_itching

toxic_look_(typhos)

depression

irritability

muscle_pain

altered_sensorium

red_spots_over_body

The raw data is gathered from the Previous patients. The raw dataset which is a file contains two columns the first one is disease and the other one is the related symptoms for that particular disease. Every disease contains at least 4-5 symptoms for the same and the data is then sent for the pre-processing so that the python code can be implemented efficiently.

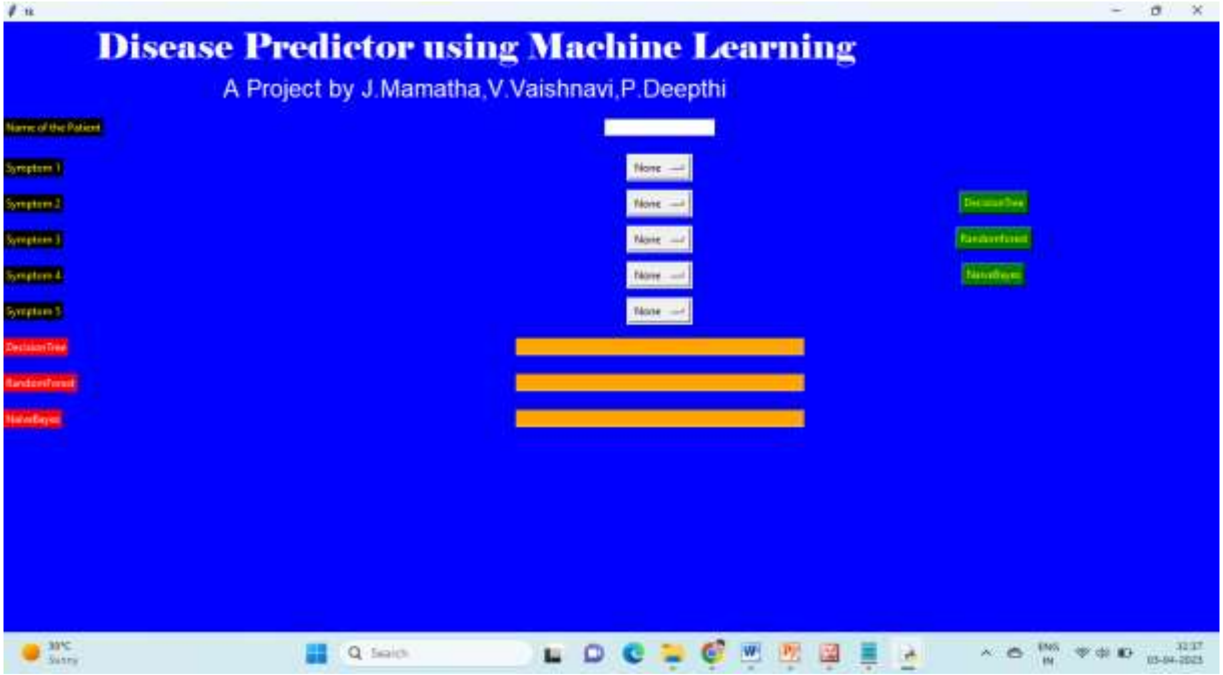
CHAPTER 4

RESULTS AND DISCUSSION

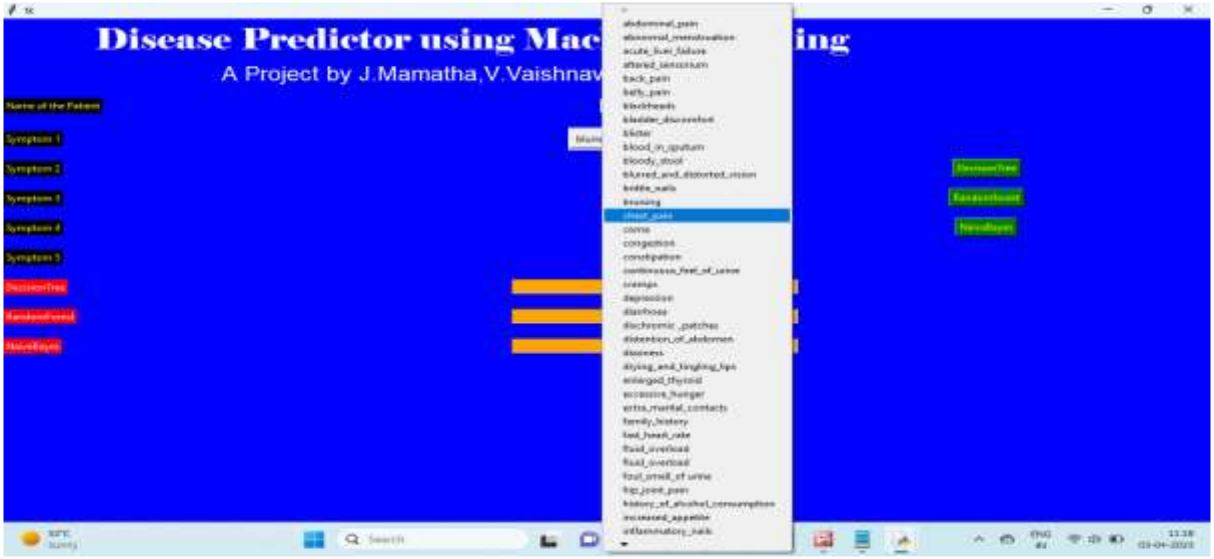
CHAPTER 4

RESULTS AND DISCUSSION

Click on run.bat file in your project directory

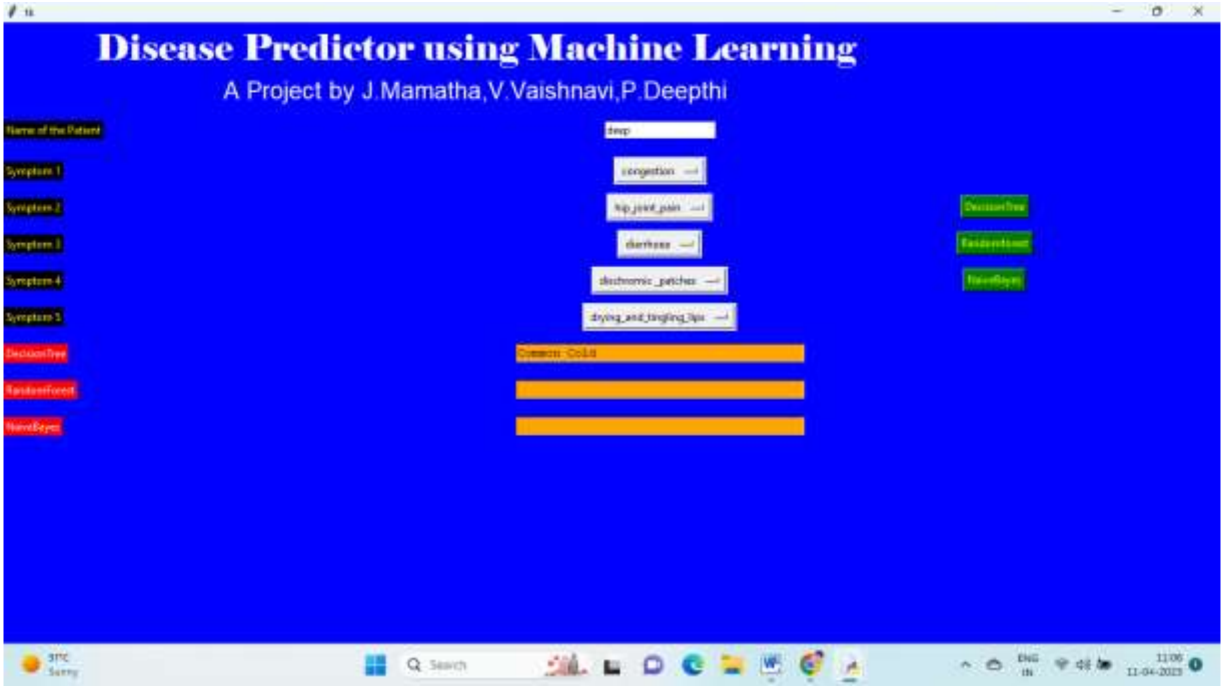


Screenshot 4.1 Enter the name of patient



Screenshot 4.2 Select the symptoms

Enter the Name of patient and enter the symptoms of the patient to prediction the disease. And then click on algorithm from which you want to predict.



Disease Predictor using Machine Learning

A Project by J.Mamatha,V.Vaishnavi,P.Deepthi

Name of the Patient: deep

Symptom 1: congestion

Symptom 2: hip_joint_pain

Symptom 3: diarrhoea

Symptom 4: dischromic _patches

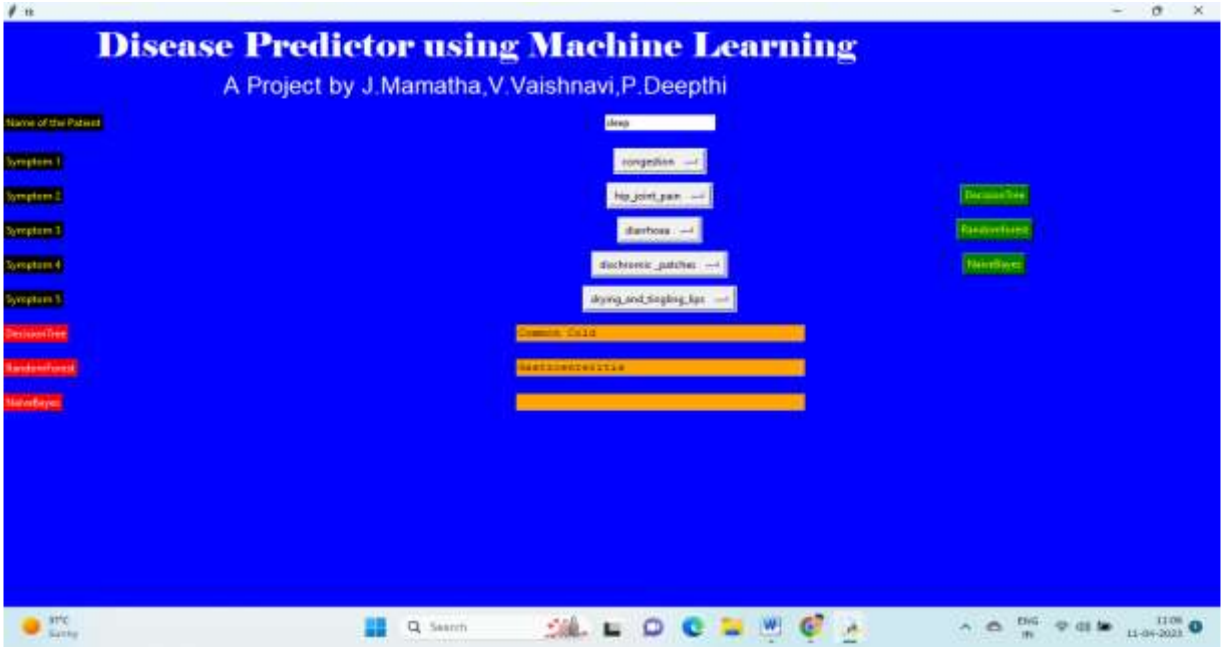
Symptom 5: drying_and_tingling_lips

DecisionTree: Common Cold

RandomForest:

NaiveBayes:

Screenshot 4.3 From the above figure for given symptoms it predicted Common cold by using Decision Tree



Disease Predictor using Machine Learning

A Project by J.Mamatha,V.Vaishnavi,P.Deepthi

Name of the Patient: deep

Symptom 1: congestion

Symptom 2: hip_joint_pain

Symptom 3: diarrhoea

Symptom 4: dischromic _patches

Symptom 5: drying_and_tingling_lips

DecisionTree: Common Cold

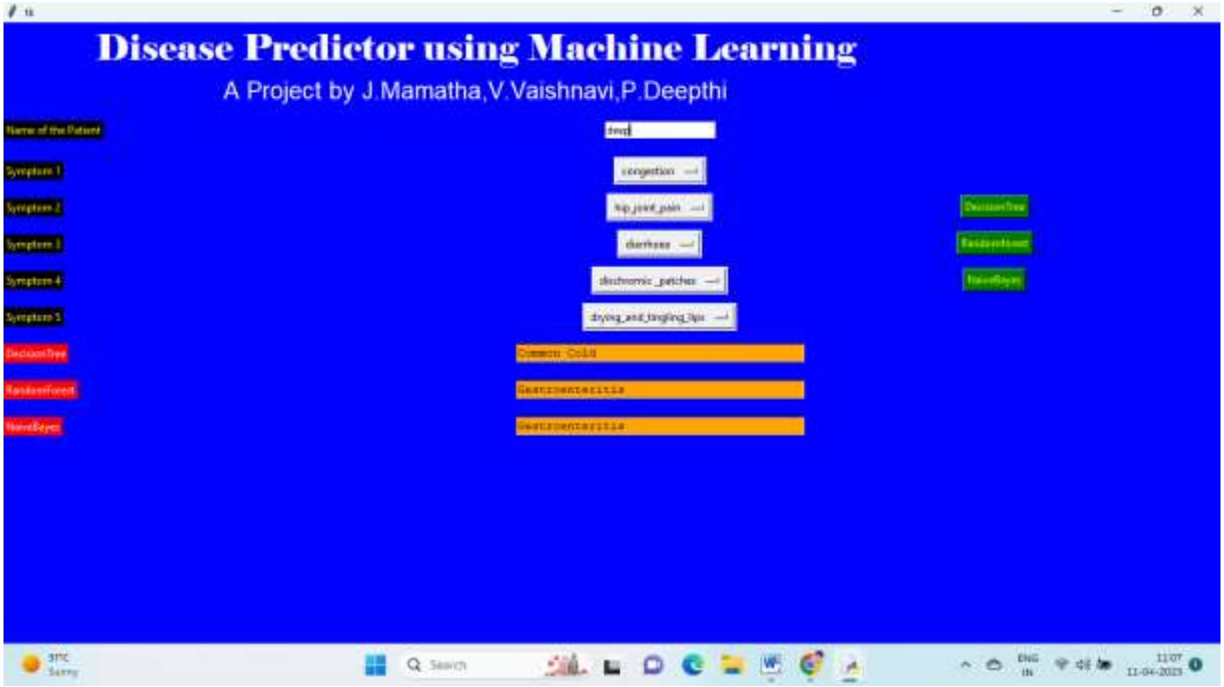
RandomForest: Gastroenteritis

NaiveBayes:

Screenshot 4.4 For Random Forest for the given symptoms it predicted Gastroenteritis

From the above figure for given symptoms it predicted Gastroenteritis by using Random Forest.

Now test for Naïve Bayes also.



Disease Predictor using Machine Learning

A Project by J.Mamatha,V.Vaishnavi,P.Deepthi

Name of the Patient: deep

Symptom 1: congestion

Symptom 2: hip_joint_pain

Symptom 3: diarrhoea

Symptom 4: dischromic _patches

Symptom 5: drying_and_tingling_lips

DecisionTree: Common Cold

RandomForest: Gastroenteritis

NaiveBayes: Gastroenteritis

Screenshot 4.5 For Naïve Bayes also for the given symptoms it is predicted as Gastroenteritis

For Navie Bayes also for the given symptoms it predicted Hypertension. Now predict the Drug for the disease.

4.1 Performance metrics

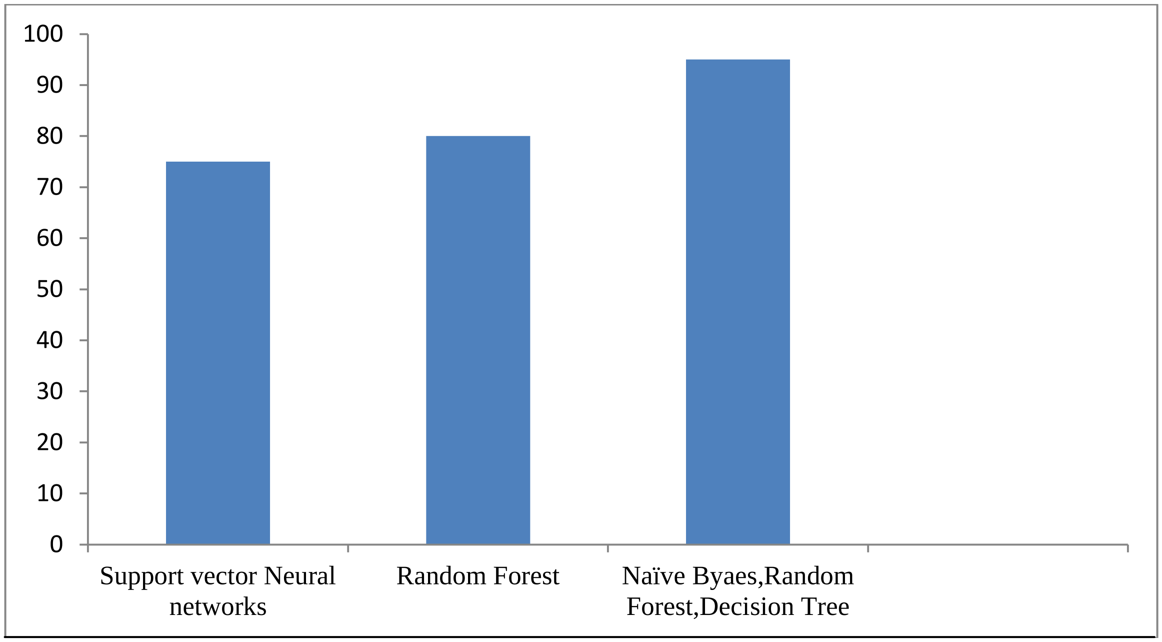


Fig 4.6 Performance Metrics

The graph describes the comparison between the algorithms which and states the best algorithm among all the applied algorithms from which Apriority is the best option and it could be further improved if there are more data included in the dataset and less one to one mappings between the disease and the symptom. The accuracy could also be improved by taking feedback from the client as how accurate the prediction was by the application and the correction could be updated and the dataset could be improved and through consulting a real doctor after using the application could improve the application to the point that consulting a real doctor would become a rare case and most of the diseases could be predicted by the application.

CHAPTER 5
CONCLUSION

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

The aim of this project is to predict disease based on symptoms. The project is set up in such a way that the device takes the user's symptoms as input and generates an output, which is disease prediction. A prediction accuracy probability of 95% is obtained on average.

To overcome the future diseases faced by human, Disease prediction is important. Hence, our Project will predict the disease based on the symptoms and drug will be suggested by Doctor or Pharmacist.

5.2 Future Enhancement

Today's, world most of the data is computerized, the data is distributed, and it is not utilizing properly. With the help of the already present data and analysing it, we can also use for un-known patterns. The primary motive of this project is the prediction of diseases with high rate of accuracy. For predicting the disease, we can use logistic regression algorithm, naive Bayes, sklearn in machine learning. The future scope of the paper is the prediction of diseases by using advanced techniques and algorithms in less time complexity.

A technology called CAD is more beneficial as sometimes systems are better diagnostics than Doctors. Machine Learning and its different branches are used in Cancer detection as well. It helps or can say assist in making decisions on critical cases or on therapies. Artificial intel-ligence plays an important role in development of many health related procedure or methods.

Artificial intelligence is very common now a days in surgeries, like Robotics surgery. Since we are in the circumstances of growing population, we must need technology which can help us to meet the expectations of the patients, their flawless cure, their better health and their smooth and easy approachable access to health care industries to heal and get well soon!!

CHAPTER 6
REFERENCES

CHAPTER 6
REFERENCES

[1] Ms. A.Bhavana, Ms. T.Navya & Ms. B.Veena, "Vitamin Deficiency and Food Recommendation System Using Machine Learning" , Journal of Emerging Technologies and Innovative Research (JETIR), Volume:09 Issue:05 |May 2022

[2] Dr. Arati Dandavate, Priyanka Gore, Namita Naikwadi, 4 Shrushti Sable & Muskan Tilwani, "Vitamin Deficiency Detection Using Image Processing and Artificial Intelligence", International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 04 | Apr 2021

[3] Palle Pramod Reddy, Dirisinala Madhu Babu, Hardeep Kumar & Dr.Shivi Sharma "Disease Prediction using Machine Learning" , International Journal of Creative Research Thoughts (IJCRT) , Volume:09, Issue:05 |May 2021

[4] Vivek Mudaliar, P.Savaridaasan & Sachin Garg, "Disease Prediction and Drug Recommendation Android Application using Data Mining (Virtual Doctor)", International Journal of Recent Technology and Engineering (IJRTE) , Volume:08 Issue:03 | September 2019

[5] Anjali Bhatt , Shruti Singasane & Neha Chaube, ''Disease prediction using machine learning", International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), Volume:04 Issue:01|January-2022

GitHub Link

https://github.com/Vaishnavi-Vadlamudi/Disease-drug-prediction-