A PROJECT REPORT

on

MACHINE LEARNING FOR DISEASE PREDICTION

A report submitted in partial fulfilment of the requirement for the award of

The degree of

BACHELOR OF TECHNOLOGY

In

Computer Science Engineering

By

Adhyyan Shrivastava 18SCSE1010615

Vaibhav Saxena 18SCSE1010124

Under the Guidance of

Dr.A.Suresh Kumar

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

OCTOBER,2021

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We would like to thank our project guide Dr.A.Suresh Kumar for guiding us at every step that we need.

We are also thankful to the teammates who are collectively putting their efforts in order to make this project successful.

We would also like to thank our overall project coordinator for helping us out and giving your valuable time.

ABSTRACT

Lack of knowledge and time causes problems to grow. Medical records of human history that can be used to analyse patterns can help diagnose the disease if it is present or future consequences depending on the disease in the individual. One such use of machine learning algorithms is within the scope of health care. Medical facilities need to be upgraded to make better patient diagnosis and treatment options. Machine learning in health care helps people to analyse complex and complex data sets and analyse them with clinical understanding. This can also be used by doctors to provide medical care. Therefore, machine learning in the field of health care can lead to increased patient satisfaction. We will try to apply the machine learning capabilities to healthcare in one program. Instead of diagnosing it, when the diagnosis of a disease is made using the study technique of a particular machine medical care can be made smarter. Other rare cases may occur when a diagnosis is made that has not been diagnosed immediately. Therefore, disease prognosis can be effectively enforced. As the saying goes, “Prevention is better than cure”, predicting disease and epidemics can lead to early prevention of disease. The project focuses primarily on program development or in other words, the provision of immediate medical care that will incorporate the collected data into medical data and store it in a health database. This database will then be analysed using K-mean machine learning algorithms to deliver results with high accuracy. It hopes to accelerate the pace of information delivery and improve the quality of health care.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Predicting the disease using patient medical history and health data through mine data and machine learning strategies has been an ongoing struggle for decades. Many businesses have tried to use data mining methods in pathological data or medical profiles to diagnose specific diseases. These methods sought to predict how recurrent diseases would occur. Also, other methods have tried to make predictions for disease control and progression. The recent success of in-depth learning in various machine learning environments has led to a shift to machine learning models that can read rich and consistent presentations of raw data with minimal pre-processing and accurate results. With the development of high-tech data, more attention is being paid to disease assessment from the perspective of big data analysis; Various studies have been conducted by selecting automated features from many data to improve the accuracy of risk classification compared to previously selected indicators. The focus is on the use of learning tools in disease assessment to visualize this particular disease in relation to a specific area. Mechanical studies have made it easier to identify various diseases and diagnose them appropriately. Predictability analysis with the help of multi-machine learning algorithms can help in accurately diagnosing the disease and help in better treatment of the patients. The health care industry produces many daily health data that can be used to gather information to predict future disease in a patient using medical history and health data. The information hidden in the health care data is then used to make effective decisions for patients, government, and NGOs. In addition, the health care sector needs to be improved through the use of informational healthcare information. The use of machine learning algorithms is relevant in the field of health care. Medical institutions need to improve to make better decisions about patient diagnosis and treatment options. Machine learning in health care allows people to analyse large and complex data sets and analyse with clinical understanding. Physicians can also use it in providing medical care. Therefore, machine learning increases patient satisfaction when involved in health care. K means the algorithm used to diagnose diseases using the patient's medical history and their medical details. Ting the disease helps us to assess what is the proximal area of a particular disease and what is not. These types of predictions help us to control the early stages of a particular disease and to control global infections.

Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) that helps us analyse data structure and take data from models. It is one of the fields of computer science, and it differs from other computer technologies in the form of computer training in terms of data provided as input and uses mathematical analysis to obtain the desired result. For this reason, Machine Learning is used in automated decision-making models such as face recognition, recommendation engines, OCR and driving apps.

Supervised learning:

Supervised learning is where the model is trained in labelled databases. The database labelled has both input and output components. Supervised reading is the most popular form of machine learning. It is easy to understand and easy to use. Supervised Reading is, where you might think the reading is guided by a teacher. We have a dataset that works as a teacher and its role is to train a model or machine. When a training model can begin to make predictions or decisions when given new data.

When training a model, the data is usually divided into an 80:20 scale i.e., 80% as training data and then rested as test data. In training data, we feed input and output data of 80%. The model learns from training data only.

It is further get classified into two parts:

1. **Regression:** It is a Targeted Reading activity where the output is a continuous value. The Wind Speed Model does not have a different value but continues at a certain distance. The purpose here is to predict the approximate value of the actual output as our model can then test is performed by calculating the error value. A bit of an error grows with the accuracy of our regression model.

2. **Classification:** It is a Targeted Reading activity where the output has a defined label Example: The goal here is to predict the divided values of a particular class and analyse it accurately.

Example of Supervised Learning Algorithms:

- Linear Regression
- Nearest Neighbor
- Gaussians Naive Bayes
- Decision Trees
- Support Vector Machine (SVM)

Unsupervised learning:

Unsupervised learning is a form of machine learning where users do not need to monitor the model. Instead, it allows the model to work on its own to discover patterns and information that were not previously available. Works great on unlabelled data. Unsupervised learning machine training uses data that is not separated or labelled and allows the algorithm to work on that information without guidance.

Prediction Models

The predictive model is seen as a model that provides a way to assess the potential risk to the patient with the outcome of the disease. The question is when, how, and how to use these species with the growth of such speculative models. These types can be taught over time, providing company needs, responding to new information or ideas.

CHAPTER-2

PROJECT DESCRIPTION

2.1 Purpose

The purpose of Sathya is to predict the possible disease by predicting symptoms an individual currently have. Swasthaya is a platform to provide service with an ease so that people do not have to visit hospitals and wait long standing in a queue, they can simply tell us their symptoms and Swasthaya will help them to know what all possible disease they might have. People can then take necessary actions based on the outcome which is 85.45% accurate, as expected. Swasthaya also provides an opportunity for a routine check-up of health, that to sitting at home and free of cost!

2.2 Problem Statement

Due to COVID-19, our doctors are doing a terrific job by helping all of us 24*7 and our hospitals are also quite equipped. Senior citizens and infants are advised to stay at home as their immune system is weak comparatively and they might be at high risk. Since, patients suffering from corona are being treated on priority basis, it is more likely for the rest of the public to stay at home and be healthy and safe.

2.3 Motivation

Machine learning techniques have been around us and has been compared and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore behind the models, and further implement Logistic Regression model to train the obtained data. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid to early detection of heart disease, in this project we have developed a model which classifies if patient will have heart disease in ten years or not based

on various features (i.e. potential risk factors that can cause heart disease) using logistic regression. Hence, the early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine.

2.4 Problem Perspective

Keeping the problem statement in mind, we have introduced Swasthaya website to help each and every citizen so that nobody has to take risk of visiting hospitals and take any sort of risk. Instead, they can use Swasthaya platform for their health check-ups and predict the disease if they have any symptoms and are in doubt.

2.5 System Requirements

- Operating System like Windows, Linux
- Web browser

  A web browser is a software application for accessing information on the internet. When a user asks for any web site it show on the browser. It helps user to interact with world wide web as a gateway to internet.

- Machine learning algorithms

  Machine learning (ML) is the study of computer algorithms that improve automatically through experience. Machine-learning algorithms use statistics to find patterns in massive amounts of data. And data, here, encompasses a lot of things—numbers, words, images, clicks, what have you. If it can be digitally stored, it can be fed into a machine-learning algorithm. There are many predefined algorithms which will use in this project like: CNN, tree.

- Python 3

  Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. It is most popular language in the world right now. Vast community support makes it best suitable language for machine learning.

- Tkinter frontend

  Tkinter is a standard Python GUI library. Python when integrated with Skinter provides a quick and easy way to customize GUI programs. Tkinter provides a powerful visual-focused interface to the Tk GUI tool kit.

  Structural or unstructured database

  Structured data is data which is organised or have pre-defined model. It helps user in extract needed data without making much effort.

  Unstructured data is a data which is not organized in a predefined manner or does not have a predefined data model.

- IDE Jupiter Notebook

  Jupyter notebook is a web-based interactive development environment. Jupyter Notebook is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning.

  - Visual Studio

  Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. Visual studio used to develop all type of apps, websites, data science etc. It provides direct fetch in with our project like webform, GitHub or website.

2.6 Literature Review

1. There are numerous works has been done related to disease prediction systems using different machine learning algorithms in medical centres.

2. Machine learning is a part of artificial intelligence. It is used to extract useful and meaningful information from complex data. The main motive of Machine learning is to compute the methods. This can be anything like- identification of patterns, mapping between symptoms and diagnoses.

3. There are mainly two steps to Predict the disease accurately and in an efficient manner:
    - ❖ Firstly, it will predict the disease from the dataset database then measure it accurately by using some algorithms like- naive bayes, decision tree and Random Forest.
    - ❖ Maximum accuracy among all the algorithms is the strength of the relationship between the training data set & testing data set by the Scikit library.

4. It is interesting to use all the algorithms at a time & find out the results in a confident manner which is based upon the tendency of each algorithm. It would be interesting for the future to finely tune the parameters of the algorithms and to test more techniques.

5. A major research effort in this field is to automatically classify disease and predict future outcomes for patients. The data set is divided into two parts that is 70% of the data are used for training and 30% used for testing. Based on the results, it is clear that the classification accuracy of Regression algorithm is better compared to others algorithms.

CHAPTER-3

TOOLS & TECHNOLOGIES

❖ Machine learning is an astonishing technology. Machine learning comes with collection of ML tools, platforms, and software.

1. NumPy

NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ND array, it provides a lot of supporting functions that make working with `ND array` very easy.

Features:

- Array creation
- Basic operation
- Universal function
- Linear algebra
- Tensors
- Incorporation with OpenCV
- Nearest Neighbor Search - Iterative Python algorithm and vectorized NumPy version

2. Pandas

Pandas is a Python library. Pandas is used to analyze data. Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets and make them readable and relevant.

Fast and efficient DataFrame object with default and customized indexing.

Tools for loading data into in-memory data objects from different file formats.

Data alignment and integrated handling of missing data.

Reshaping and pivoting of date sets.

Label-based slicing, indexing and subsetting of large data sets.

Columns from a data structure can be deleted or inserted.

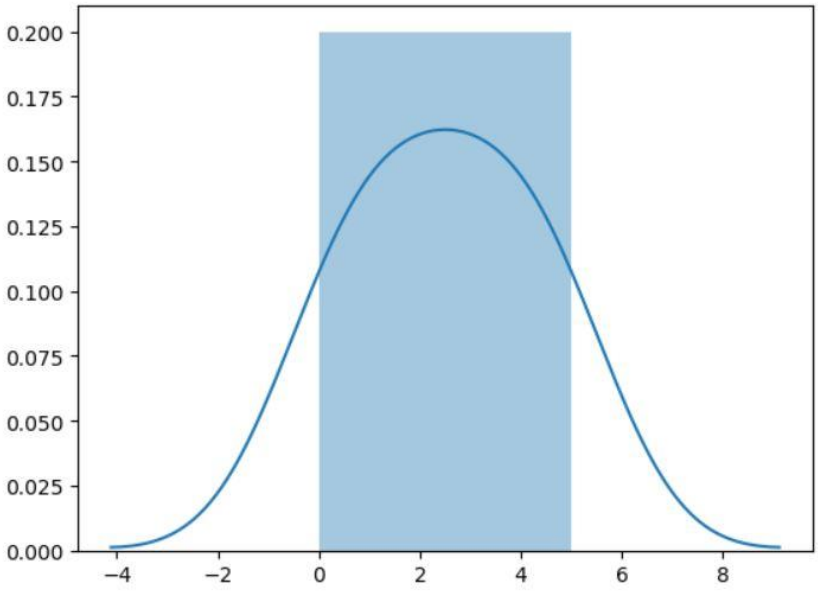
Group by data for aggregation and transformations.

High performance merging and joining of data.

Time Series functionality.

3. Sea born

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. The main idea of Seaborn is that it provides high-level commands to create a variety of plot types useful for statistical data exploration, and even some statistical model fitting.



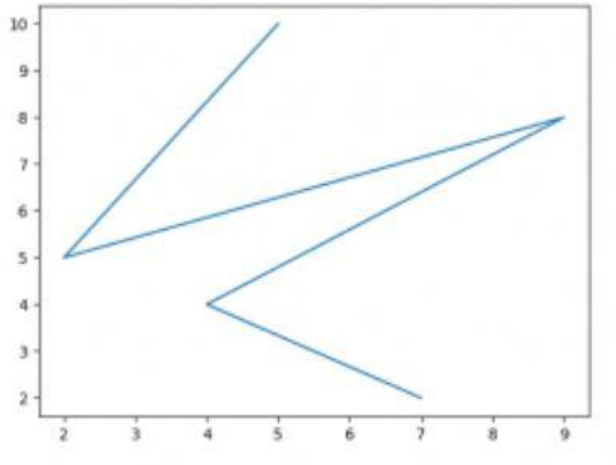
3.1 Sea Born

4. Matplotlib

Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. matplotlib. pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels.

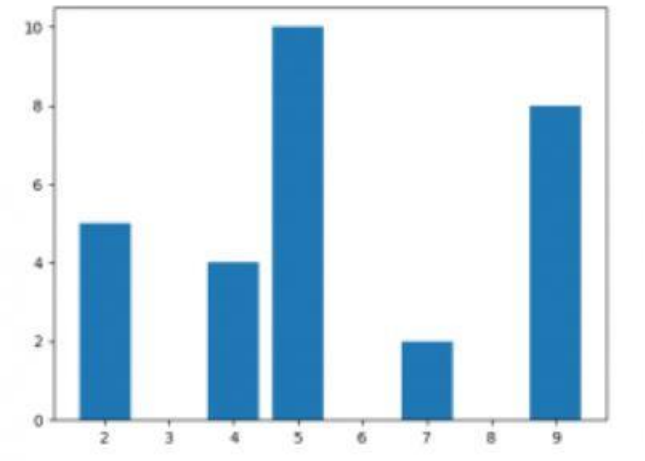
There are four types of plots-

1. **Line Plot:**



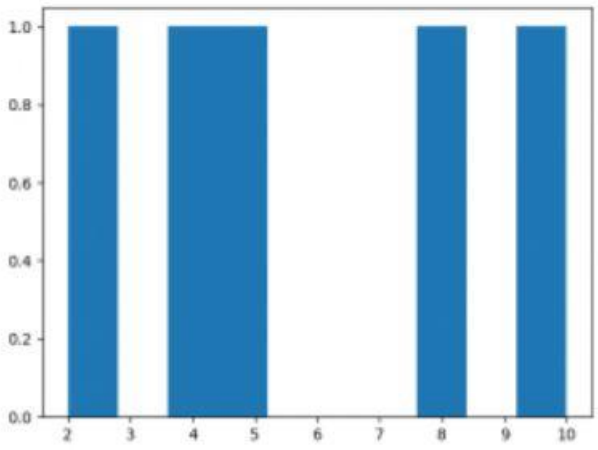
3.2 Matplotlib

2. **Bar Plot:**



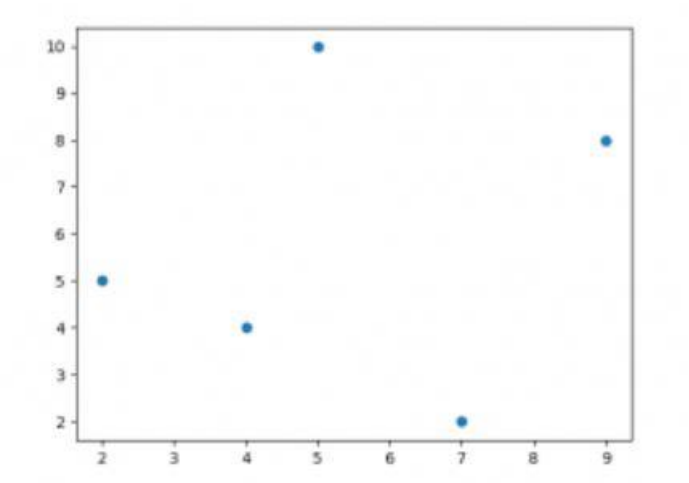
3.3 Bar Plot

3. **Histogram:**



3.4 Histogram

4. **Scatter Plot:**



3.5 Scatter Plot

5. Jupyter notebook

Jupyter notebook is one of the most widely used machine learning tools among all. It is a very fast processing as well as an efficient platform.

Thus, the name of Jupyter is formed by the combination of these three programming languages. Jupyter Notebook allows the user to store and share the live code in the form of notebooks. One can also access it through a GUI. For example, win python navigator, anaconda navigator, etc.

6. Tkinter

The tkinter package ("Tk interface") is a standard Python interface in the Tcl / Tk GUI toolbar. Both Tk and Tkinter are available on many Unix platforms, including macOS, and Windows systems.

To activate the python -m tkinter from the command line should open a window that displays a simple Tk interface, let us know that the tkinter is properly installed on our system, and also shows which version of Tcl / Tk is installed, so you can read Tcl / Tk texts specific to that version.

Tkinter supports a wide range of Tcl / Tk versions, built with or without support. Official Python Tcl / Tk 8.6 string binary release threads. See the module code tkinter module for more details on supported versions.

Tkinter is not a small threat, but it adds a fair amount of its own idea to make the experience Pythonic. These documents will focus on these additions and modifications, and refer to the official Tcl / Tk text for unchanged details

.

To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

6. GUI

There are many graphical user interface (GUI) tools that you can use in the Python editing language. The top three are Tkinter, wxPython, and PyQt. Each of these tools will work with Windows, MacOS, and Linux, and PyQt has additional mobile capabilities.

A graphical user interface is an app with buttons, windows, and many other widgets that a user can use to interact with your app. A good example would be a web browser. It has buttons, tabs, and a large window when uploading all content.

METHODS AND MATERIALS

User

User is a people who want to know the disease outbreak or study any disease with great interest and prevent it from any doing further damage or read disease profoundly for their self-project or other self-interest.

Machine Learning

We will used different model to predict accurate result using different provided structural and non-structural database. We collect data from different government website and medical company or hospital website.

Interactive web page

Page named (Swasthya) will provide interactive website to visualize different type of disease formatted in different entity w.r.t different Symptoms category. Each disease will be curated on different entity of symptoms.

Requirement

- Web browser, e.g. chrome, Edge, morzilla etc
- Machine learning algo for model prediction
- Python 3, HTML, CSS, JavaScript
- Structural or unstructured database
- Code Editor e.g. Visual Basic code editor, IDE Jupiter
- Tinkers, GUI, Numpy and Pandas

Major Task or Methodology

1. Use different structural dataset visualization to predict special disease using machine learning algorithm.
2. Use Naïve bayes, Decision tree and random forest machine learning algorithms for accurate prediction of disease. For disease prediction required disease symptoms dataset.

3. In this general disease prediction, the living habits of an individual and check-up information are considered for the accurate prediction.
4. Used tinker for user interface, where user select their different symptoms and predict the disease with different accuracy based on machine learning algorithm.

CHAPTER-4

IMPLEMENTATION

Decision Tree

Decision Tree is a Supervised learning approach that can be used for both segregation and Regression problems, but it is especially preferred to solve classification problems. A tree-shaped separator, where the internal nodes represent the elements of the database, the branches represent the rules of decision and the node of each leaf represents the result.

The deciding tree is a flowchart-like structure in which each internal node represents a "test" in the attribute (eg: a category label (a decision taken after applying all the attributes).

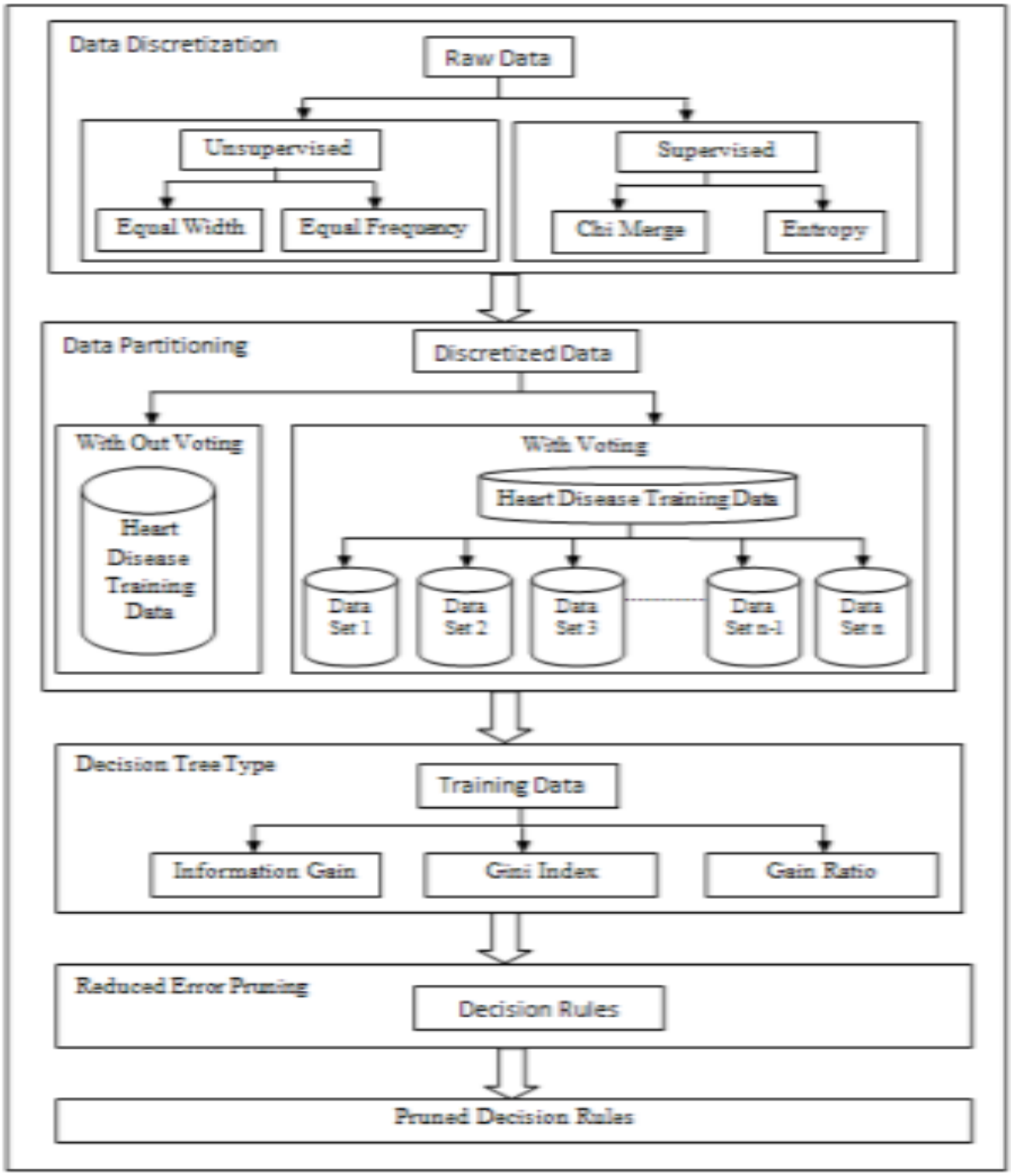
The decision tree has three types of nodes:

Decision nodes – typically represented by squares

Chance nodes – typically represented by circles

End nodes – typically represented by triangles

Decision trees are often used in research and project operations. If, in fact, decisions have to be made online without having to remember under incomplete information, the decision tree should be compared to a possible model such as the best choice model or the online selection algorithm. descriptive methods for calculating the probability of conditions.



4.1 Decision tree

Random Forest

Random Forest is a combination method that is able to perform retrofitting and separation tasks using multiple decision-making trees and a process called Bootstrap and Aggregation, more commonly known as bagging. The basic premise of this is to combine multiple decision trees in determining the final outcome rather than relying on individual decision trees.

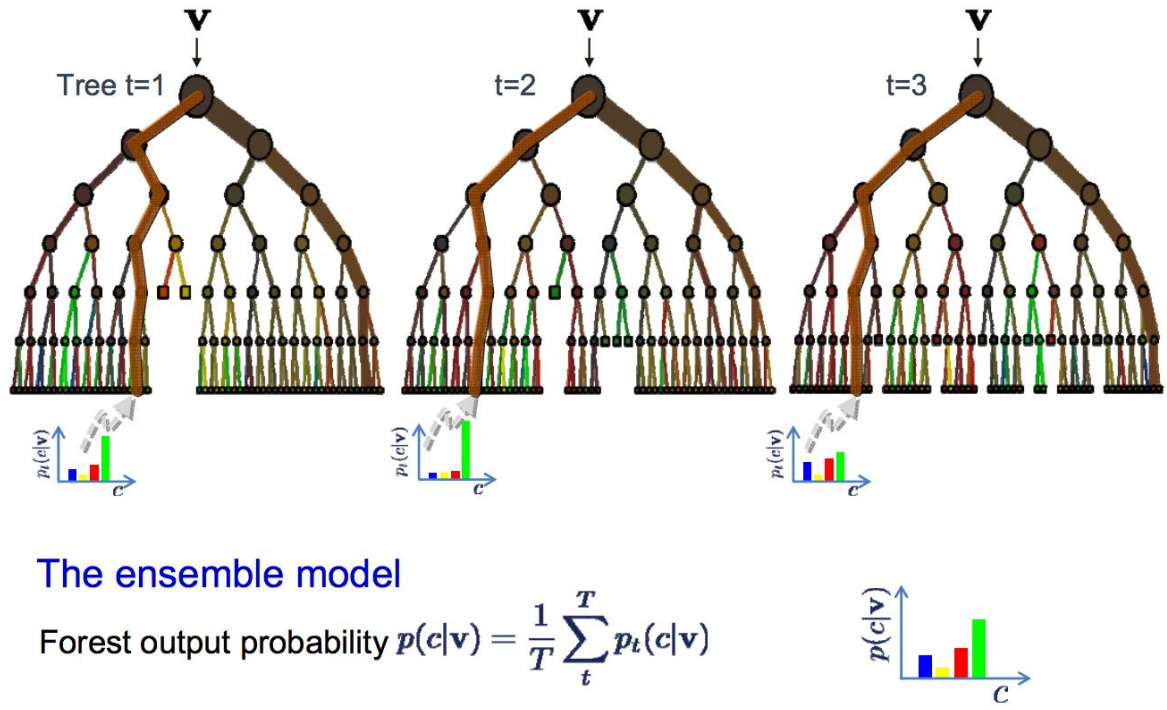
The Random Forest has many trees for decision-making as basic learning models. We randomly process the sample and then extract the sample from the database that creates the data samples for all models.

We need to look at the process of deforestation in a random forest like any other machine learning process

- Create a specific query or data and find a source to determine the required data.
- Make sure the data is in an accessible format and convert it to the required format.
- Specify all visible faults and missing data points that may be required to complete the required data.
- Create a machine learning model
- Set the basic model you want to achieve
- Train data machine learning model.
- Provide model understanding with test details
- Now compare performance metrics for both test data and predicted data from the model.
- If expectations do not meet your expectations, you can try to improve your model appropriately or fall in love with your data or use another data modelling process.

Steps to implement the random forest:

1. Import the required libraries
2. Import and print the dataset
3. Select all rows and column from dataset to x and all rows and column 2 as y
4. Fit random forest regressor to the dataset
5. Predicting a new result
6. Visualising the result



4.2 Random Forest

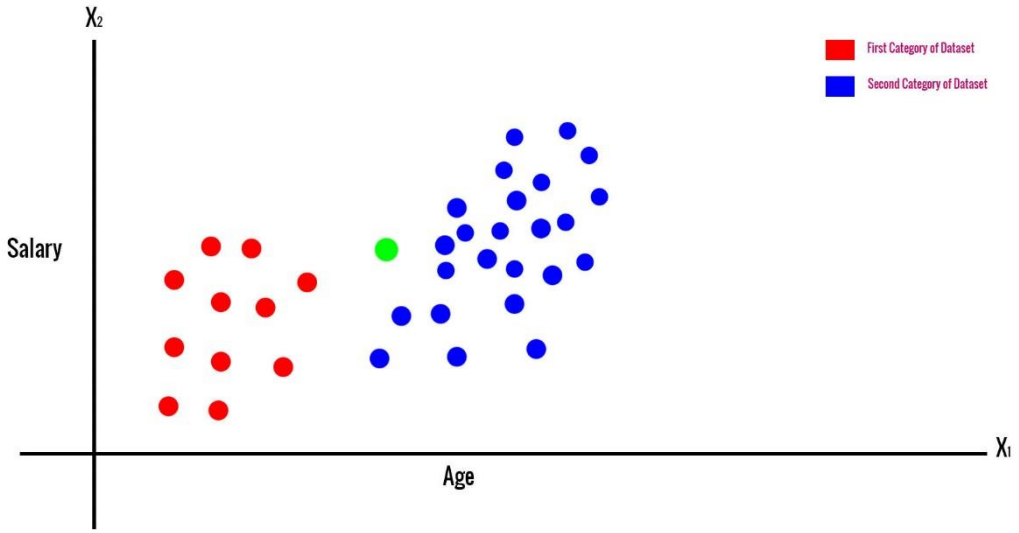
Naive Bayes

The Naive Bayes are an easy way to create classifiers: models that offer class labels in problematic situations, are represented as velvety values, where class labels are drawn on a limited set. There is no single algorithm for training such classifiers, but a family of algorithms based on the same principle: all Bayes classifiers are classified as the value of a particular element independent of any other element, given the class flexibility. For example, a fruit can be considered an apple if it is red, round, and about 10 cm wide. The inexperienced Bayes editor considers each of these features to be independent of the possibility that the fruit is an apple, regardless of the combination of color, rotation, and width.

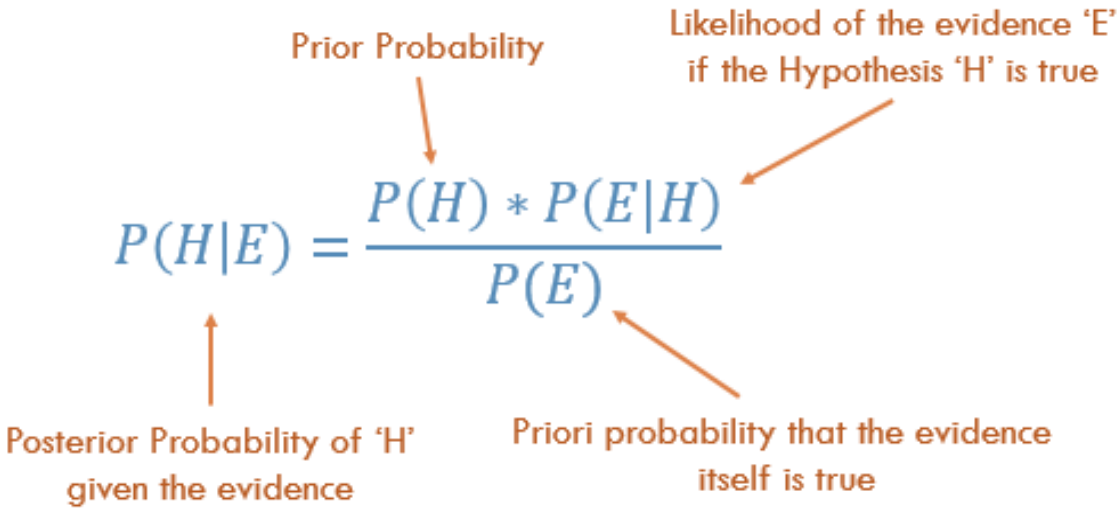
In other types of possible models, Bayes classifiers can be trained very well in a supervised learning setting. In many operating systems, the parameter measurement of the inexperienced Bayes models uses the high probability method; In other words, one can work with an inexperienced Bayé model without accepting the Basesi opportunities or using any Basezi methods.

In addition to their irrational design and extremely obvious speculation, the inexperienced Bayes dividers have worked well in many of the complex realities of the real world. In 2004, an analysis of the Basesian segregation problem showed that there were plausible theoretical reasons for making it irrefutably obvious to unsuspecting Bayes rescuers. However, a complete comparison of other class algorithms in 2006 showed that the planning of the Bayes was done over other alternatives, such as developed trees or informal forests.

The advantage of the inexperienced Bayes is that they require only a small number of training data to measure the parameters required for the division



4.3 Naïve bayes



4.4 naïve bayes formula

CHAPTER-6
Deployment

Backend

Importing libraries

In [1]:
```
from tkinter import *
import numpy as np
import pandas as pd
from tkinter import messagebox
```

Dataset

In [2]:
```
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_s
'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)
```

6.1 Importing lib & Dataset

jupyter Disease prediction using machine learning Last Checkpoint: Last Monday at 7:45 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Testing data

In [3]:

```
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[11]

y = df[["prognosis"]]
np.ravel(y)
print(y)
```

```
   itching  skin_rash  nodal_skin_eruptions  continuous_sneezing  shivering  \
0        1          1                     1                    0          0
1        0          1                     1                    0          0
2        1          0                     1                    0          0
3        1          1                     0                    0          0
4        1          1                     1                    0          0

   chills  joint_pain  stomach_pain  acidity  ulcers_on_tongue  ...  scurring  \
0       0           0             0        0                 0  ...         0
1       0           0             0        0                 0  ...         0
2       0           0             0        0                 0  ...         0
3       0           0             0        0                 0  ...         0
4       0           0             0        0                 0  ...         0

   skin_peeling  silver_like_dusting  small_dents_in_nails  \
0             0                    0                     0
1             0                    0                     0
2             0                    0                     0
3             0                    0                     0
4             0                    0                     0

   inflammatory_nails  blister  red_sore_around_nose  yellow_crust_ooze  \
0                   0        0                     0                  0
1                   0        0                     0                  0
```

```
   prognosis  Unnamed: 133
0          0           NaN
1          0           NaN
2          0           NaN
3          0           NaN
4          0           NaN

[5 rows x 134 columns]
      prognosis
0             0
1             0
2             0
3             0
4             0
...         ...
4915         36
4916         37
4917         38
4918         39
4919         40

[4920 rows x 1 columns]
```

6.2 Testing dataset

Training data

In [4]:
```
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)
```

Out[4]:
```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40,  0], dtype=int64)
```

6.3 Training Dataset

Decision tree

In [5]:
```
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")
```

6.4 Decision Tress algorithm

Randon forest

In [6]:

```
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")
```

6.5 Random Forest algorithm

Naive bayes

In [7]:

```
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")
```

6.6 Naïve bayes algorithm

GUI(Tkinter)

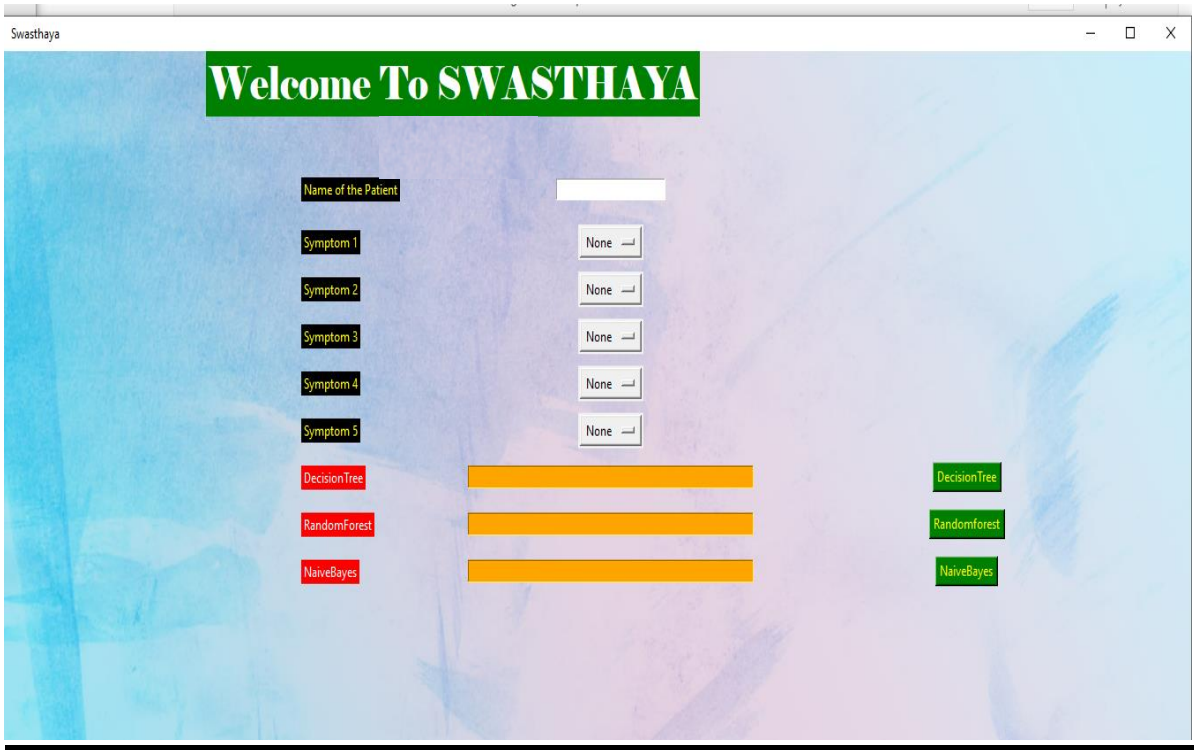
In [*]:

```
root = Toplevel()
root.configure(background='blue')
root.title("Swasthaya")

root.geometry("800x700")

# Add image file
c=Canvas(root, bg="gray16",height=200,width=200)
filename = PhotoImage(file = "bg.png")
background_label=Label(root, image=filename)
background_label.place(x=0,y=0,relwidth=1,relheight=1)
c.pack

# 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="Welcome To SWASTHAYA", fg="white", bg="green")
w2.config(font=("Elephant", 30))
w2.grid(row=1, column=0, columnspan=2, padx=250)
w2 = Label(root, justify=LEFT, text="IT-G7", fg="white", bg="green")
w2.config(font=("Aharoni", 20))
w2.grid(row=2, column=0, columnspan=2, padx=250)

# labels
NameLb = Label(root, text="Name of the Patient", fg="yellow", bg="black")
NameLb.grid(row=6, column=1, pady=15, sticky=W)


S1Lb = Label(root, text="Symptom 1", fg="yellow", bg="black")
S1Lb.grid(row=7, column=1, pady=10, sticky=W)

S2Lb = Label(root, text="Symptom 2", fg="yellow", bg="black")
S2Lb.grid(row=8, column=1, pady=10, sticky=W)
```

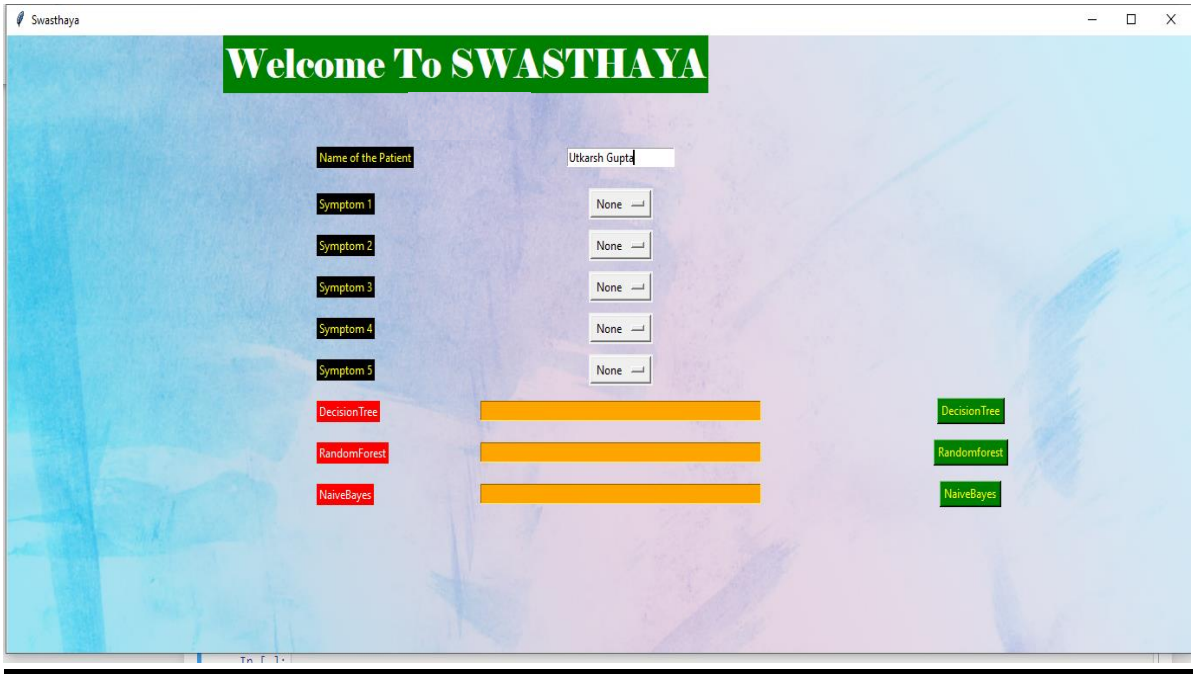
```
# entries
OPTIONS = sorted(l1)

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=15, column=3,padx=10)

rnf = Button(root, text="Randomforest", command=randomforest,bg="green",fg="yellow")
rnf.grid(row=17, column=3,padx=10)

lr = Button(root, text="NaiveBayes", command=NaiveBayes,bg="green",fg="yellow")
lr.grid(row=19, 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()
```

6.7 GUI Tkinter (Front End Code)

Front End

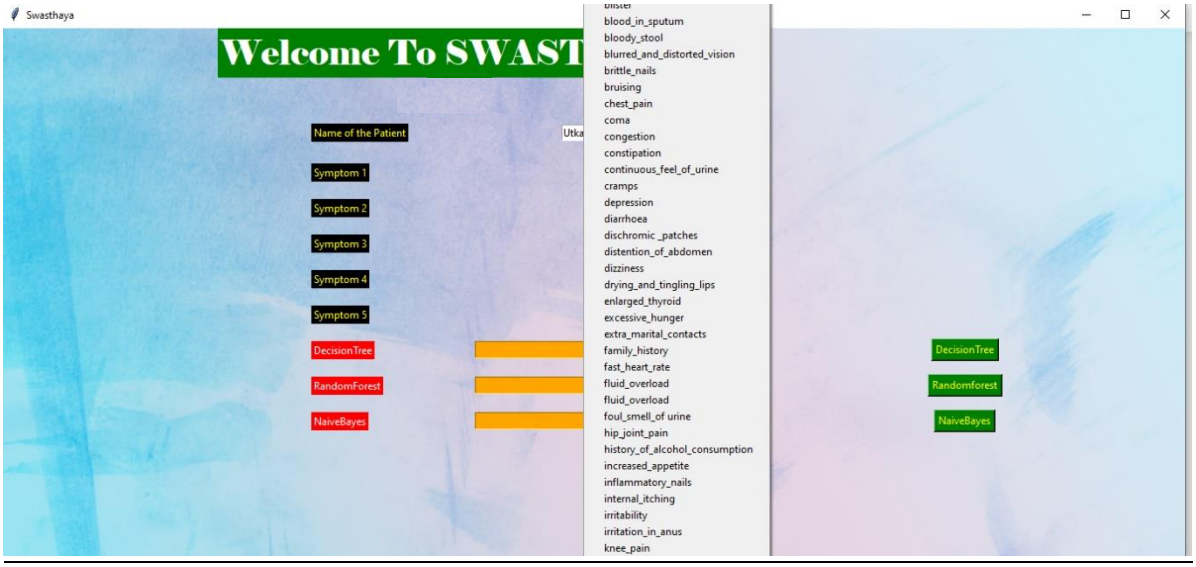


6.8 Home Page

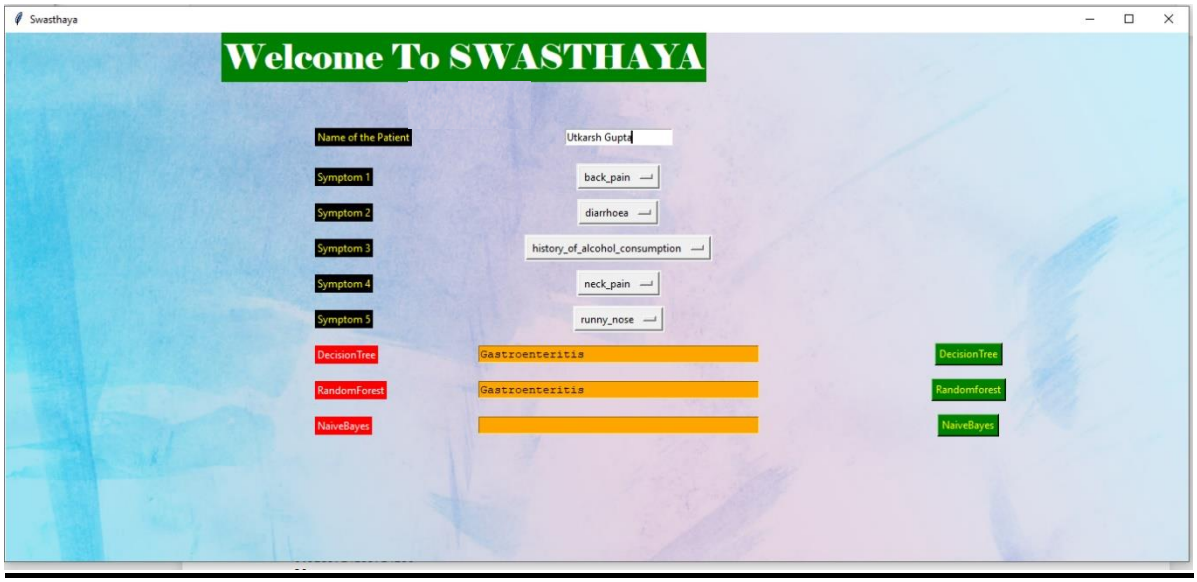


6.9 Input name

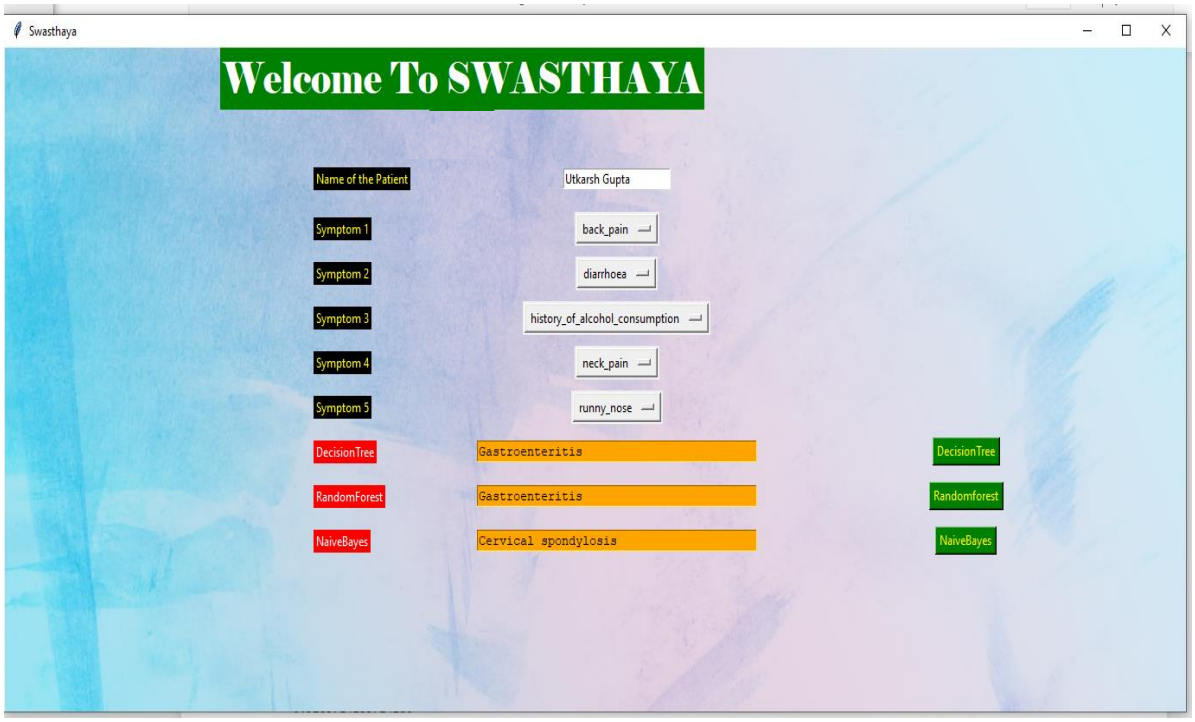
6.10 Symptoms 1



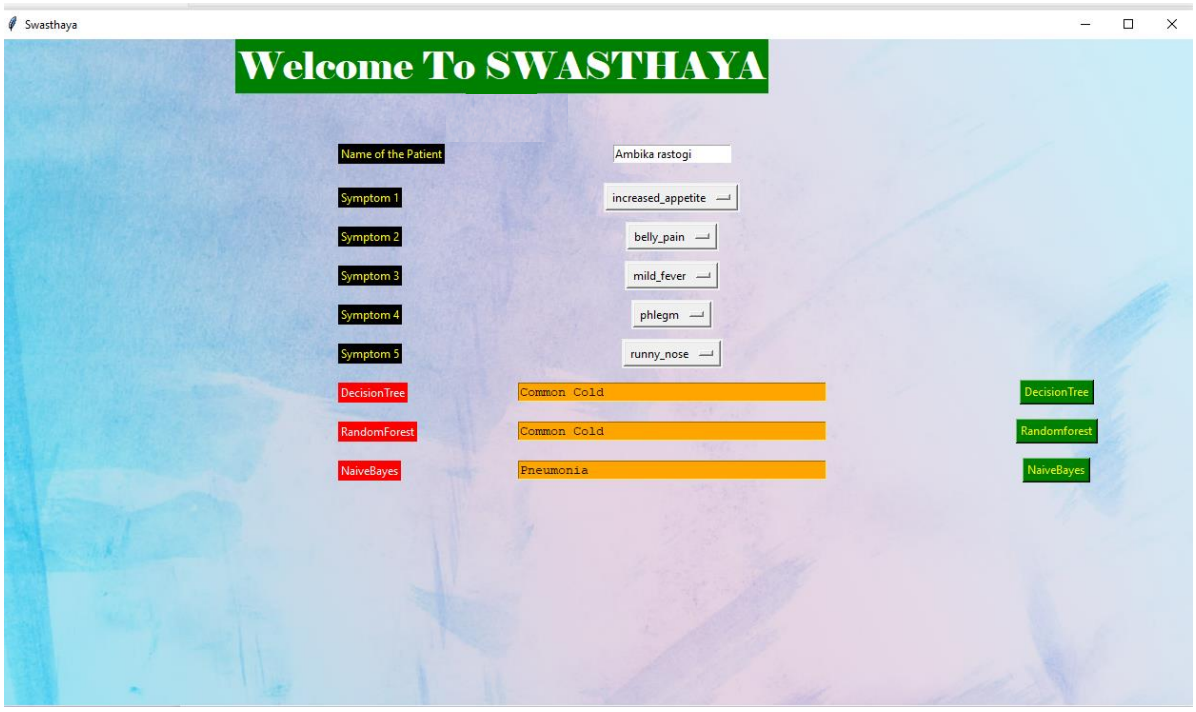
6.11 Symptoms 2



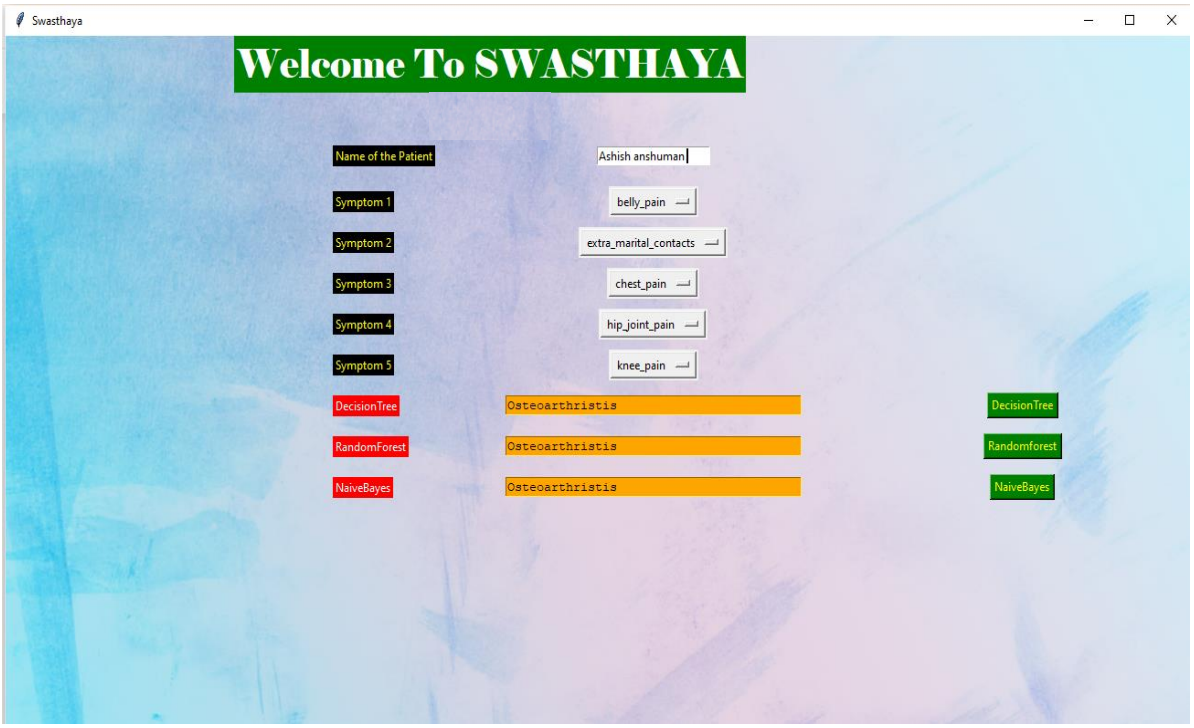
6.12 Prediction using Decision tree and Random forest



6.13 Prediction using decision tree, Random forest and naïve bayes



6.14 Prediction 1



6.15 Prediction 2

CHAPTER-7

CONCLUSION

1. The primary purpose of this activity is to extract and classify data using machine learning techniques.
2. During the research we continue to set many details and many algorithms that make the best accuracy in our predicted diseases.
3. It is not expected that the models will be accurate in every detail; it is expected that exemplary predictions can be taken as a sound basis for action by participants.
4. A sound basis for action 'seems to be a reasonable desire to predict the dangers of future diseases.
5. The use of formal, measurable methods of predicting future risk of infectious disease has become more complex and widely accepted in recent years.
6. A key challenge for the future is to develop interdepartmental frameworks to provide reliable and useful predictions for future disease risks.

FUTURE SCOPE

Our project is specifically designed for tough situations like COVID wherein people avoid physical contact and visiting areas with huge gathering. In such situations, there is an urge for people who are not well to visit doctors/hospitals for their well-being and unwillingly, will have to join huge gathering where every other person in gathering might be infected with one virus or the other.

Here comes the picture of our project which will serve the purpose without the people in need having to join any such gathering. Our project will ask about the symptoms that the patient is suffering from, it will diagnose based on the data set we have collected from different research papers, Government websites and many more. It will help them know what disease they might have so that they can take proper medication accordingly.

Reference

1 Wes McKinney, Python for Data Analysis, Shroff Publishers & Distributors Pvt. Ltd, Third Edition, Indian Reprint, July 2015.

2 Wesley J Chun, Core Python Application Programming, Third Edition, Pearson.

3 Reema Thareja, Python Programming Using Problem Solving Approach, Fourth Edition, Oxford University Press.

4 Sebastian Raschka & Vahid Mirjalili, Python Machine Learning, Second Edition, Packet.

5 https://www.coursera.org/learn/machine-learning

6 https://www.kaggle.com/tags/healthcare

7 https://www.who.int/en.

8 https://data.world/datasets/health

9 https://archive.ics.uci.edu/ml/datasets/Heart+Disease

10 https://data.gov.in/search/site?query=health

11 www.irjet.net

12 G. Saranya, A. Pravin. "A comprehensive study on disease risk predictions in machine learning", International Journal of Electrical and Computer Engineering (IJECE), 2020

13 aaaproductreviews.com

14 www.ijcaonline.org

15 ieeexplore.ieee.org