A Project Report

On

Disease Prediction Based On Symptom

Submitted in partial fulfilment of the
requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

SUBMITTED BY

GUNTU VEERA MANIKANTA

(20A91F0015)

Under the Esteemed Guidance of

Mrs. P.Anuradha,MCA,M.Tech(Ph.D)

Assistant Professor

DEPARTMENT OF MCA

ADITYA ENGINEERING COLLEGE

An Autonomous Institution

(Approved by AICTE, Affiliated to JNTUK & Accredited by NBA, NAAC with 'A' Grade) Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956

SURAMPALEM- 533 437, E.G.Dt,ANDHRA PRADESH

2020-2022

ADITYA ENGINEERING COLLEGE

An Autonomous Institution

(Approved by AICTE, Affiliated to JNTUK & Accredited by NBA, NAAC with 'A' Grade) Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956

SURAMPALEM- 533 437, E.G.Dt,ANDHRA PRADESH

DEPARTMENT OF MCA

CERTIFICATE

This is to certify that the project work entitled, **"DISEASE PREDICTION BASED ON SYMPTOM",** is a bonafide work carried out by **GUNTU VEERA MANIKANTA** bearing Regd.No:**20A91F0015** submitted to the requirements for the award of the Computer Applications in partial fulfilment of the requirements for the award **o**f degree of **MASTER OF COMPUTER APPLICATIONS** from **Aditya Engineering College**, Surampalem during the academic year **2020-2022**.

Project Guide

Mrs.P.Anuradha,MCA,M.Tech,(Ph.D)
Assistant Professor,
Department of MCA,
Aditya Engineering College,
Surampalem-533437

Head of the Department

Mrs.D.Beulah,M.Tech,(Ph.D)
Associate Professor,
Department of MCA,
Aditya Engineering College
Surampalem-533437.

EXTERNAL EXAMINER

DECLARATION

I here declare that the project entitled **“Disease prediction based on symptom”,** submitted to **Aditya Engineering College, Surampalem** has been carried out by me alone under the guidance of **Mrs.P.Anuradha**

Place:

Surampalem

Date:

GUNTU VEERAMANIKANTA

(20A91F0015)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first person I would like to thank is my guide **Mrs P.Anuradha, MCA, M.Tech,(Ph.D), Assistant Professor, Department of MCA**, **Aditya Engineering College, Surampalem.** Her wide knowledge and logical way of thinking have made a deep impression on me. Her understanding, encouragement and personal guidance have provided the basis for this project. She is a source of inspiration for innovative ideas and her kind support is well known to all her students and colleagues.

I wish to thank **Mrs D.Beulah, M.Tech, (Ph.D), Head of the Department, Aditya Engineering College, Surampalem,** who has extended her support for the success of this project.

I also wish to thank **Dr. M.Sreenivasa Reddy,** M.Tech, **Principal, Aditya Engineering College, Surampalem,** who has extended his support for the success of this project.

I Would like to thank all the **Management & Technical Supporting Staff** for their timely help throughout the project.

ABSTRACT

Healthcare is a significant industry that provides value-based care to millions of people. People require healthcare as a basic need. Many people today are afflicted with diseases for which they have no idea what they are or what medications they should take. In today's world less than half of world's population have basic health services. It is necessary for people to be able to diagnose their ailment and know which medications they should take. For the prevention and treatment of illness, timely and accurate analysis of any health-related problem is essential. In the case of a serious illness, the standard method of diagnosis may not be sufficient.

The development of a medical diagnosis system based on machine learning (ML) algorithms for disease prediction can help in a more accurate diagnosis than the conventional practice. Using numerous machine learning techniques, I created a disease prediction system.

The machine learning techniques used are Support Vector Machines, Decision Tree classifier. This diagnosis model can operate like a doctor in the early detection of a disease, allowing for timely treatment and the saving of lives. This model predicts the disease or illness accurately and also prescribes the best drug to take for curing the illness.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

1. INTRODUCTION

1.1 Brief Information about the Project

Medicine and healthcare are critical components of the financial system and human life. The world we live in now compared to the world we lived in a few weeks ago has changed exotic.In this implicit world, medico's and nurses are doing everything they can to save people's lives, even if it means putting their own lives in risk.There are also some seclusion villages without nursing services".

"In some cases, sufferers are unable to see a doctor for a medico symptomatic. The newest covid-19 emergency is a good instance of how separate are afraid to go to hospitals for checkup disease because they are nervous of deflate the illness.We need technology-based healthcare mode to deal with these kinds of happenings.Smart healthcare ease by technology is no longer a dream,as Internet-connected pharmaceutical gadgets are carry the health system from fall in under the weight of the residents.

"Computer science innovation abet in the capability of technology-based health care. Machine learning is a section of computer science that is rapidly improving in the medical sector. With the improving different of machine learning applications in the medical sector, I can see a up comming day's where data, analysis, and innovation work together to aid innumerable people without them even grasp it. Soon, ML-based applications embedded with real-time stoical data available from numerous medical management systems in different countries will be same place, upgrade the potency of earlier unavailable check-up alternatives.

1.2 Motivation and Contribution of the Project

The main motivation and contribution of this project is to save the sufferer healths information on the mode and suggestions to the sufferer. scheme is implement to create a therapeutic mode as a idiomatic agent which stir up users to respond to health problems on the basis of sign, returns the prognosis. This System framework can identify user interface sign's. forecasts the illness and suggests medication for these extracted symptoms. Random Forest and Decision Tree are the machine learning algorithm used here. This obviously illustrates that a medical System can identify patients healths to carry out a straightforward symptom diagnosis and chatty method.

The health community of the state is strongly feign by medical mode. It is less susceptible to human mistakes and more reliable. The Internet is more vulnerable this day's, but it doesn't matter for the well-being of the people. In minor problems which can become a big sickness in future they refuse hospital check-up. This idea is a solution to the errors. This concept focuses on providing the whole day with a free mode. The idea that the mode is free and convenient to any area or work field, leads the user to use it. It can be accessed. It saves the overall cost of consulting specialists.

1.3 Objective of the Project

In this system, suggest key neutral of the suggest framework is to encourage people to take action in the interest of well-being in life by ensuring that the mode is accessible to everone. mode and sap have a good history. It provides a healthy human experience to help the user and the machine communicate. The user speaks about their wellness in this method and it is a perfect way for people to monitor the healthier lifestyle. One of the fundamental aspects of this system is

that provides a sense of protection for non-person, particularly in mental well-being, because the communication remains only accessible in trust. It is purposeful to support and provide urgent steps that people can't go due to the easily on hand.

1.4 Organization of the Project

- **Chapter2: Literature Survey:** This chapter consists of Survey to this project how to implement to the system.
- **Chapter3: System Analysis:** This chapter consists of description of current system,proposed system,algorithms, and requirements specifications.
- **Chapter4: System Design:** This chapter mainly consists of modules description and unified modelling language diagrams use case diagrams, class diagrams.
- **Chapter5: Technology Description:** This chapter mainly consists of technology used in this project.
- **Chapter 6: Sample code:** This chapter mainly consists of sample code for few modules.
- **Chapter 7: Testing:** This chapter mainly consists of testing techniques and test cases for modules.
- **Chapter 8: Screenshots:** This chapter mainly consists of output screens of this project.

Conclusion: Main conclusion of this project.

CHAPTER-2
LITERATURE SURVEY

2. LITERATURE SURVEY

Dahiwade, Dhiraj; Patle, Gajanan; Meshram, Ektaa (2019). 3rd International Conference on Computing Methodologies and Communication (ICCMC) Designing Disease Prediction Model Using Machine Learning Approach. proposed a data mining approach to predict the disease. They developed a general disease prediction system based on symptoms of the patient. Forthe disease prediction they used algorithms such as K Nearest Neighbors(KNN) and Convolutional Neural Networks(CNN). The CNN algorithm achieved an accuracy of 84.5% which is more than the accuracy of KNN. Their model also gives the risk associated with general disease. Arumugam proposed a data mining approach for predicting heart disease of a patient. They used three machine learning algorithms such as decision trees, naïve bayes and support vector machines.The decision tree model beat the other two algorithms. Decision trees achieved accuracy of about 90%. SVM and Naïve Bayes achieved accuracies of 88% and 77% respectively.

Rayan Alanazi proposed a method of identification and prediction of the presence of chronic disease in an individual using the machine learning algorithms such as CNN and KNN. In this paper, the performance of the proposed model is compared with other algorithms such as Naïve Bayes decision tree, and logistic regression algorithms. The results show that the proposed system provides an accuracy of 95% that is higher than that of the other two algorithms. It is highly believed that the proposed system can reduce the risk of chronic diseases by diagnosing them earlier and also reduces the cost for diagnosis, treatment, and doctor consultation.

Sneha Grampurohit in this paper presents a comprehensive comparative study of three algorithms performance on a medical record each yielding an accuracy up to 95 percent. The performance is analyzed through confusion matrix and accuracy score. Artificial Intelligence will play even more important

role in data analysis in the future due to the availability of huge data produced and stored by the modern technology.

Suchith Reddy Vemula in this paper there has been sufficient study on the use of sentiment analysis, in the field of medicine, such nlp and sentimental analysis techniques must be given more weight, since the US Food and Medication Administration, has done multiple research on the consequences of adverse drug responses on patients. They have used SVM and logistic regression and they got an accuracy like 76%.They said that method of prediction of drug-disease association using Machine Learning and NLP can be next best method for implementation of disease drug prediction system with better accuracy.

Grampurohit this paper aim of developing classifier system using machine learning algorithms is to immensely help to solve the health-related issues by assisting the physicians to predict and diagnose diseases at an early stage. Their Sample data of 4920 patients' records diagnosed with 41 diseases was selected for analysis. A dependent variable was composed of 41 diseases. For disease prediction system developed using Machine learning algorithms such as Decision Tree classifier they got an accuracy upto 91%.

Huiqing Wang proposed a model for predicting drug-disease associations based on dense convolutional attention network.They proposed a novel deep learning model for drug-disease association prediction, called DCNN. The model introduces the Gaussian interaction profile kernel similarity for drugs and diseases, and combines them with the structural similarity of drugs and the semantic similarity of diseases to construct the feature space jointly. They have used Convolution Neural Networks here and they got an accuracy like 95%.

CHAPTER-3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 Existing System

In the existing mode, the sufferer may have to take medico's appointment and he/she is diagnosed by the doctor at the appointment time. In the meantime, the seriousness of the diseases/illness might increase. To get the better of this elegant mode is needed which acts a virtual doctor in predicting sufferer's illness.

Disadvantages:

- Increases the patients disease.
- Patient did'nt idea about their health problem.
- Patient have to take medico's appointment takes more time.

3.2 Proposed System

In the proposed mode, an online model is implemented which helps in finding the kind of illness the patient is suffering. It also report the illness and provides carefulness to decrease the effect of illness. The proposed mode also suggests the most used drug to take to cure the illness.

It collects all the most regular symptoms from the sufferer and accurately finds the kind of illness the patient is suffering from. my project also helps to compare the results obtained from different classifiers and hence more accurate result can be found. In the up comming day's this project can be further developed by considering sufferer's factors such as age, sex, health habits, longevity of symptoms etc.

Advantages

- It is used to predict the illness of the patient accuractly.
- To reduce the time to take awarness of the patient disease.
- Doctors can treat patients problems through online.

Algorithm-1: Random Forest

Random Forest algorithm start with the selection of random samples from the given dataset. Next, will construct a decision tree for every sample. Then it will get the prediction result from every decision tree. After that voting will be performed for every prediction then select the most voted prediction result as the final prediction result.

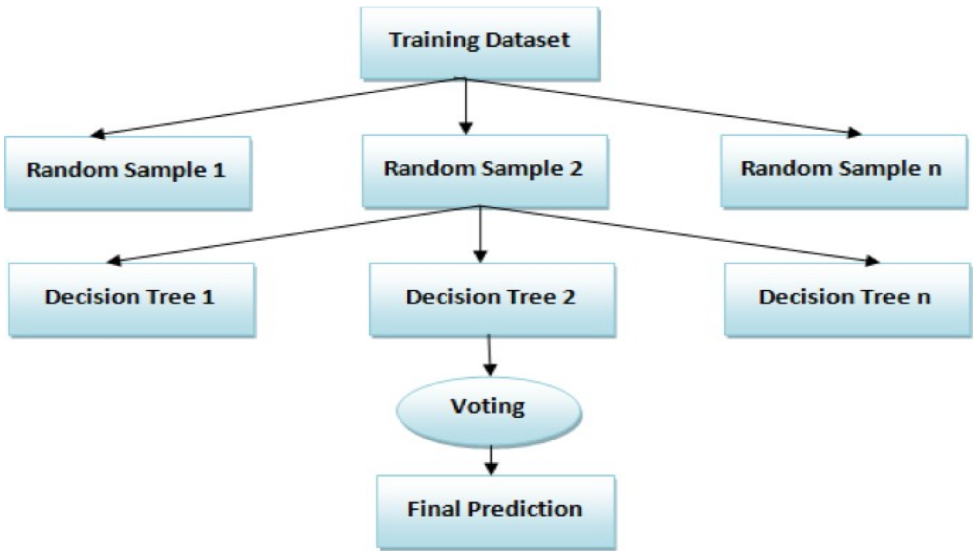
Pseudocode:

Step 1: Randomly select “k” features from total “m” features.

Step 2: Among the “k” features, calculate the node “d” using the best split point.

Step 3: Split the node into daughter nodes using the best split.

Step 4: Repeat 1 to 3 steps until “I” number of nodes has been reached.



Step 5: Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.

Algorithm-2 : Decision Tree Classifier

Decision tree is a well known ML algorithm. A decision tree model classifies data items into a tree-like structure. There are multiple levels and in each level there are multiple nodes. The top most node is a root node and the internal nodes represent input values of the tree. I have implements Decision tree in disease prediction System.

Pseudocode:

Step 1: Create a node N;

Step 2: If tuples in D are all of the same class, C, Then

Step 3: Return N as leaf node labeled with the class C;

Step 4: If attribute_list is empty then

Step 5: Return N as a leaf node labeled with the majority class in D; // majority voiting

Step 6: Apply attribute_selection_method(D, attrubute_list) to find the "best" splitting_criterion;

Step 7: label node N with splitting _criterion;

Step 8: If splitting_attribute is discrete-valued and multiway splits allowed then// notr estricted to binary trees.

Step 9: attribute_list – attribute_list – splitting _attribute; // remove splitting_attribute.

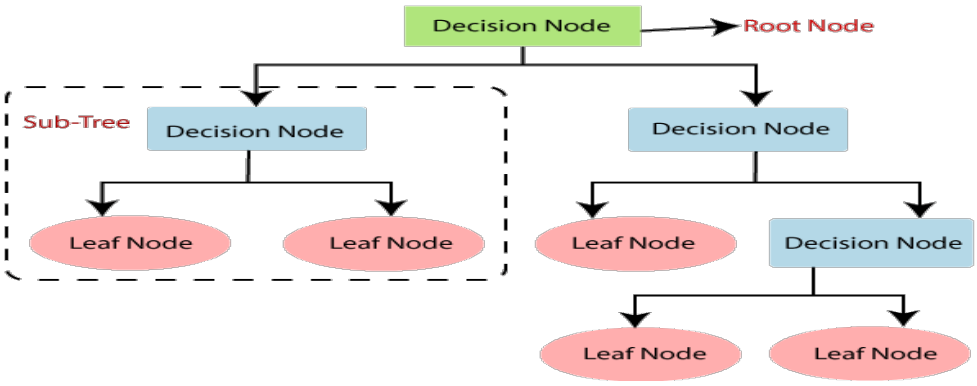
Step 10: for each outcome j of splitting_criterion
// partition the tuples and grow subtrees for each partition

Step 11: let Dj be the set of data tuples in D satisfying outcome j; // a partition.

Step 12: if Dj is empty then attach a leaf labeled with the majority class in D to node N;

Step 13: else attach the node returned by Generate_decision_tree(Dj,

Step 14: return N;



3.3 Feasibility Study:

An important outcome of initial investigation is the resolve that the mode request is feasible. This is possible only if it is feasible within restricted resource and time. The different feasibilities that have to be analysed are

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

Operational Feasibility

Operational possibility deals with the study of probability of the mode to be implemented. This mode of operationally get rid of all the tensions of the admin and helps him in effectively trace the project progress. This kind of automation will surely reduce the time and energy, which before consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility

Economic possibility or Cost-benefit is an evaluation of the economic defence for a computer-based project. Since the mode is a network based, any

number of workers connected to the program within that field can use this tool from at any time. So, the project is economically possibility.

Technical Feasibility

Technical possibility is the evaluation of the technical assets of the organization. The consortium needs IBM consistent machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform independent environment. Python used to develop the mode. The technical possibility has been carried out. The mode is technically possible for implement and can be implemented with the existing facility.

3.4 Functional Requirements:

- The model we used is built with sklearn using Random Forest and Decision Tree.
- We select the dataset which contain related data about the symptoms and the disease.
- Next, we do data pre-processing and remove irrelevant data.
- Then we train the data with different classifiers.
- Results of all the classifiers are produced.

3.5 Non-Functional Requirements:

Availability : The software for detecting disease based on symptoms can be available in all modes where this is an internet connection.

Performance: It is calculated in terms of the predictions of the model. Accuracy measures the performance of the model.

3.6 Requirement Specifications

3.6.1Software Requirements

- Operating System: Ubuntu or windows
- Domain : Machine learning
- Technologies : Python, Sqlite3
- Tools : Vscode

3.6.2Minimum Hardware Requirements

- Processor : Intel core i3 11^{th} Gen or 10
- RAM : 4GB to 16GB
- Hard Disk : 500MB

CHAPTER-4
SYSTEM DESIGN

4.SYSTEM DESIGN

4.1 INTRODUCTION

After look over the requirement of the mission to be accomplished, the next step is to research the problem and understand it's context. The first exercise in the phase is studying the existing mode and other is to understand the essential and domain of the new mode. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and essential of a new system is more difficult and needs creative thinking and understanding of existing running system is more difficult, improper understanding of present mode can lead diversion from solution.

4.2 System Architecture:

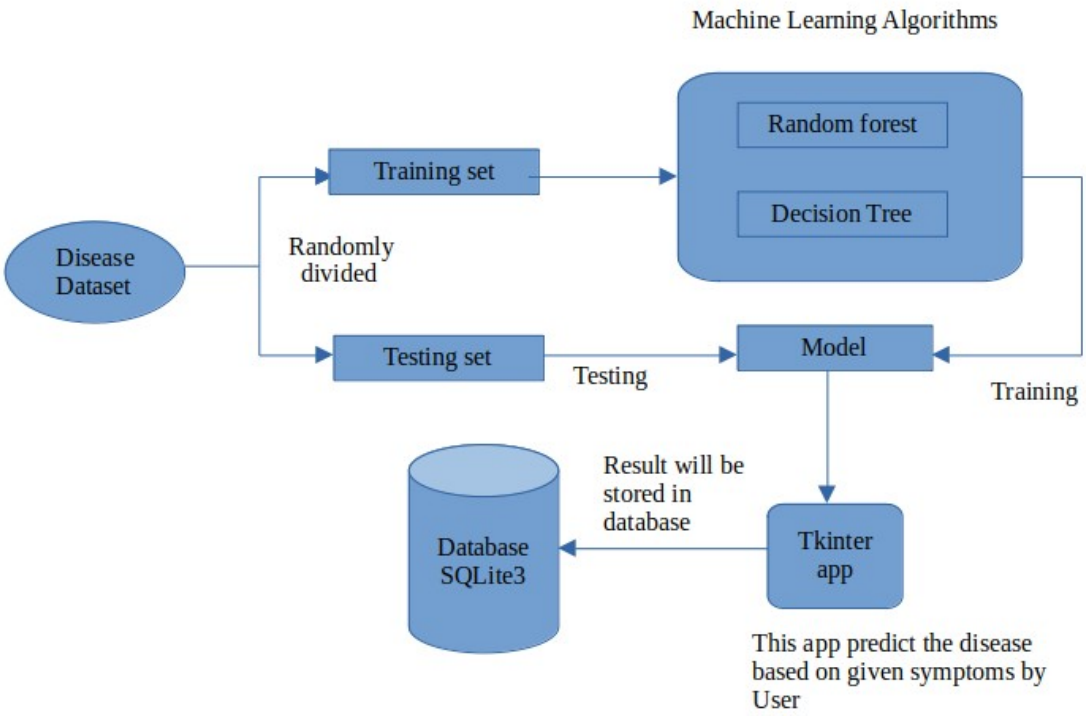


Fig 4.2 System Architecture for Disease prediction based on symptoms

Description:

The above architecture Disease dataset is randomly divided into two sets one is Training and other is Testing sets. Then Machine learning algorithms to train them and learn. Then after train the models then test the models through Testing set. Then performance the prediction to the model.

4.3 UML Diagrams

i) The combine modelling language allows the software engineer to express an exploration model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

ii) A UML system is represented using five different views that describe the system from distinctly different perspective. UML is specifically constructed through two different domains.

iii) UML Analysis modelling, this focuses on the user model and structural model views of the system.

iv) UML design modelling, which focuses on the behavioural modelling, implementation modelling and environmental model views.

These are divided into the following types.

- Use case diagram
- Class diagram
- Sequence
- Collaboration
- Activity

4.3.1 Use case Diagram

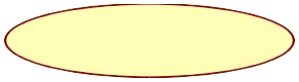
Use Case diagrams classify the functionality supplied by the mode (use cases), the users who interact with the system (actors), and the association between the users and the functionality. Use Cases are used in the classify phase of software development to articulate the high-level requirements of the system. The primary goals of Use Case diagrams include:

- Providing a high-level view of what the system does.
- Identifying the users ("actors") of the system.
- Determining areas needing human-computer interfaces.

Graphical Notation: The basic elements of Use Case diagrams are the Actor, the Use Case, and the Association.

Actor An Actor, as mentioned, is a user of the system, and is depicted using a stick figure. The role of the user is written beneath the icon. Actors are not limited to humans. If a system communicates with another application, and expects input or delivers output, then that application can also be considered an actor.

Use Case A Use Case is functionality provided by the system, Use Cases are depicted with an ellipse. The name of the use case is written within the case.



Directed Association These Associations are used to link Actors with Use Cases, and indicate that an Actor participates in the Use Case in some form.

Behind each Use Case is a series of actions to achieve the proper functionality, as well as alternate paths for instances where validation fails, or errors occur. These actions can be comming days defined in a Use Case description. Because this is not addressed in UML, there are no standards for Use Case descriptions.

However, there are some common templates can follow, and whole books on the subject writing of Use Case description.

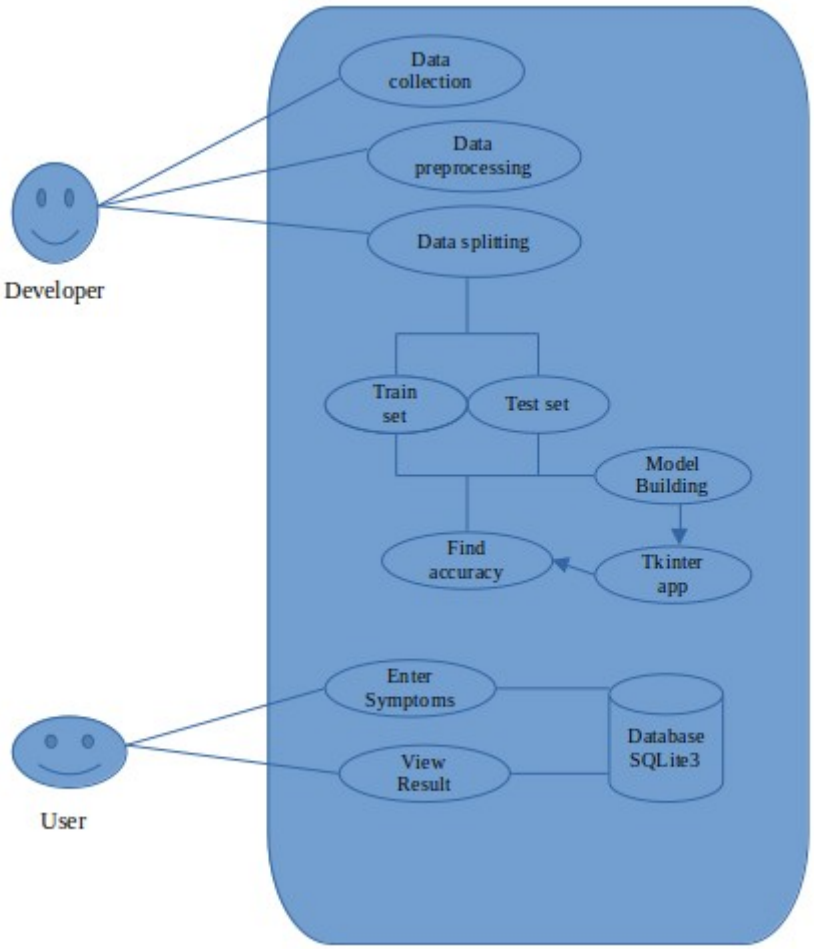


Fig: 4.3.1 Use case diagram for Disease prediction based on symptom

Description:

The above diagram represents the use case diagram of the project, and Disease prediction based on symptoms. The above use case diagram contains Data collection & preprocessing and train the models. The actors perform the performance in this use case diagram.

Use case Template

Use case name	Post information
Participating actors	User, System
Flow of events	• User should collect the data. • User should upload information to the system.
Entry condition	User must be used the webcam.
Exit condition	User successfully rang the alarm.

4.3.2 Class Diagram

Class diagrams identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes, such as an inheritance relationship. Part of the popularity of Class diagrams stems from the fact that many CASE tools, such as Rational XDE, will auto-generate code in a variety of languages, these tools can synchronize models and code, reducing the workload, and can also generate Class diagrams from object-oriented code.

Class: Classes are building blocks in object-oriented programming. A class is depicted using a rectangle divided into three sections. The top section is name of class, the middle section defines the properties of a class. The bottom section List the methods of class.

Association: An Association is a generic relationship between two classes and is modelled by a line connecting the two classes. This line can be qualified with the type of relationship and can also feature multiplicity rule for the relationship.

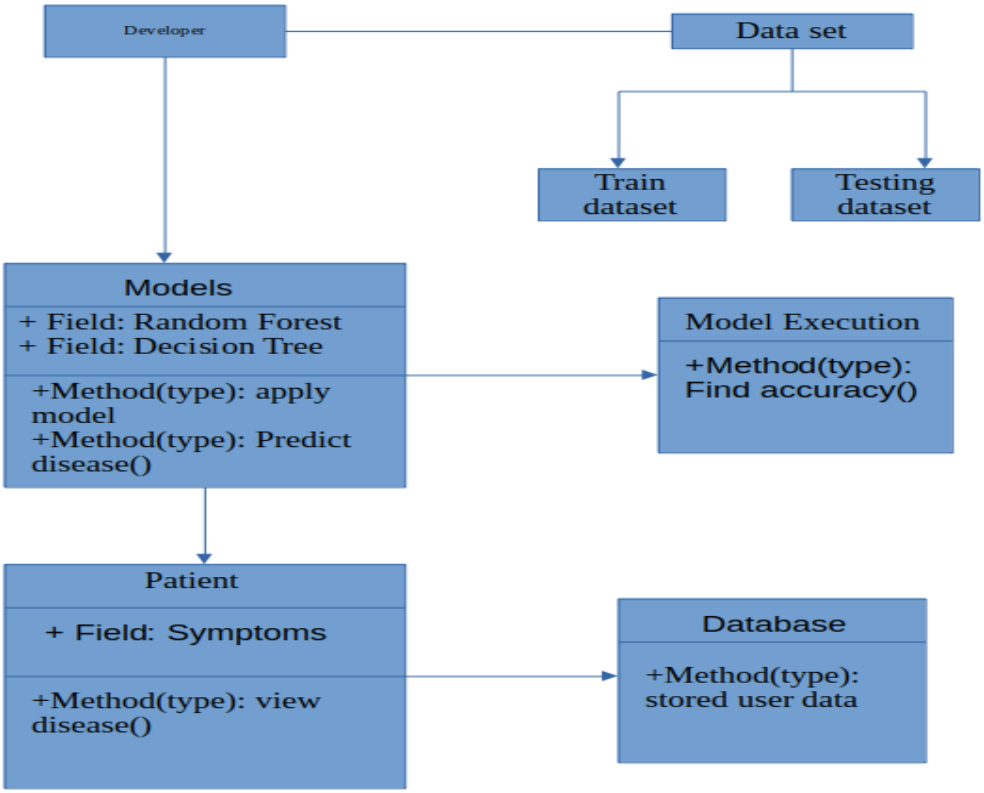


Fig: 4.3.2 Class diagram for Disease prediction based on symptoms

Description:

The above diagram represents the use case diagram of the project, and disease prediction based on symptoms. The above class diagram contains Data collection & Pre-processing and train the models. The actors perform the functionalities in this use case diagram.

4.3.3 Sequence:

Sequence diagrams document the interactions between classes to achieve a result, such as a use case. Because UML is designed for object-oriented programming, these communications between classes are known as messages. The Sequence diagram lists objects horizontally, and time vertically, and models these messages over time.

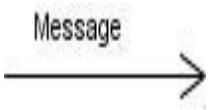
Graphical Notation: In a Sequence diagram, classes and actors are listed as columns, with vertical lifelines indicating the lifetime of the object over time.

Object: Objects are instances of classes and are arranged horizontally. The pictorial representation for an Object is a class (a rectangle) with the name prefixed by the object name (optional)and a semi-colon.

Lifeline: The Lifeline identifies the existence of the object over time. The notation 2for a Lifeline is a vertical dotted line extending from an object.

Activation: Activations, modeled as rectangular boxes on the lifeline, indicate when the object is performing an action.

Message: Messages, modelled as horizontal arrows between Activations, indicate the communications between objects



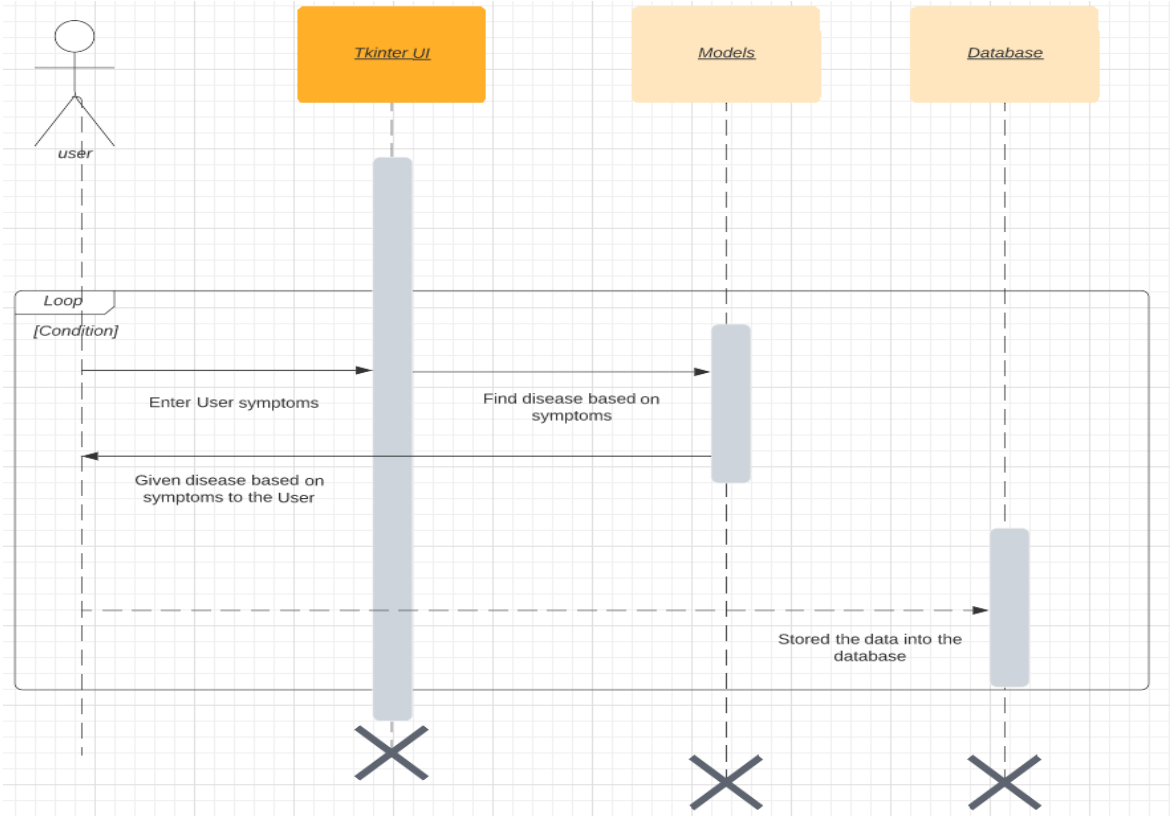


Fig: 4.3.3 Sequence diagram for Disease prediction based on symptom

Description: The above diagram represents the sequence diagram of the project, an efficient feedback control mechanism.

4.3.4 Collaboration Diagram :

A collaboration diagram is also known as communication diagram, is an illustration of the relationships and interactions among software objects in the UML. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. The four major components of a collaboration diagram.

Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name. If an object has a property or state that specifically influences the collaboration, this should also be noted.

Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.

Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.

messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

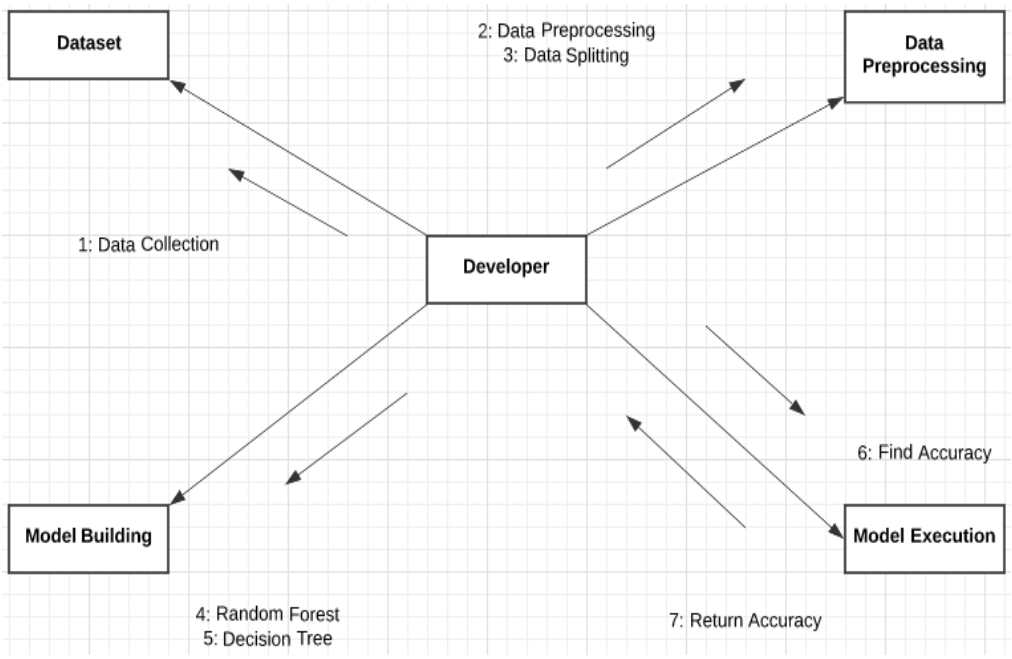


Fig: 4.3.4 Collaboration Diagram for Disease prediction based on symptoms

4.3.5 Activity Diagram

This shows the flow of events within the system. The activities that occur within a use case or within an objects behaviour typically occur in a sequence. An activity diagram is designed to be simplified look at what happens during an operation or a process.

Each activity is represented by a rounded rectangle the processing within an activity goes to compilation and then an automatic transmission to the next activity occurs. An arrow represents the transition from one activity to the next. An activity diagram describes a system in terms of activities. Activities are the state that represents the execution of a set of operations. These are like flow chart diagram and dataflow.

Initial state: which state is starting the process?

Action State: An action state represents the execution of an atomic action, typically the invocation of an operation. An action state is a simple state with an entry action whose only exit transition is triggered by the implicit event of completing the execution of the entry action.

Transition: A transition is a directed relationship between a source state vertex and a target state vertex. It may be part of a compound transition, which takes the static machine from one static configuration to another, representing the complete response of the static machine to a particular event instance.

Final state: A final state represents the last or "final" state of the enclosing composite state. There may be more than one final state at any level signifying that the composite state can end in different ways or conditions. When a final state is reached and there are no other enclosing states it means that the entire state machine has completed its transitions and no more transitions can occur.

Decision: Astate diagram (and by derivation an activity diagram) expresses decision when guard condition are used to indicate different possible transitions that depend on Boolean conditions of the following.

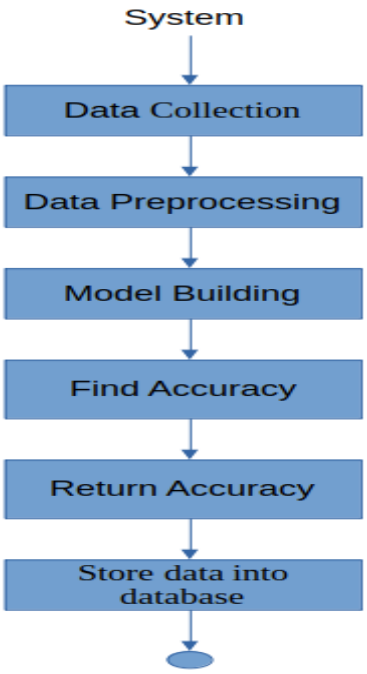


Fig: 4.3.5 Activity Diagram for Disease prediction based on symptoms

CHAPTER-5
TECHNOLOGY DESCRIPTION

5. TECHNOLOGY DESCRIPTION

5.1 Introduction to Python

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. You might have heard that python in popular in any domain. It is easy to understand. It is very simple syntax and have so many inbuilt funactions and librarys like numpy, pandas, mataplotlib.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Features:

Easy:

When we say the word ‘easy’, we mean it in different contexts.

Easy to Code

Python is very easy to code as compared to other popular languages like Java, c++, c- language etc.

Easy to Read

Being a high-level language, Python code is like English. Looking at it, you can tell what the code is supposed to do. Also, since it is dynamically typed, it mandates indentation. This aids readability.

Free and Open-Source

Firstly, Python is freely available. You can download it from the Python website. Secondly, it is open source. This means that its source code is available to the public. You can download it, change it, use it, and distribute it. This is called FLOSS (Free/Libre and Open-Source Software)**.** As the Python community, we're all headed toward one goal- an ever-bettering Python.

High-Level

Python is a high-level language. This means that as programmers, we don't need to remember the system architecture. Also, we need not manage memory. This makes it more programmer- friendly and is one of the key python features.

Portable

Let's assume you've written a Python code for your Windows machine. Now, if you want to run it on a Mac, you don't need to make changes to it for the same. In other words, you can take one code and run it on any machine. This makes Python a portable language. However, you must avoid any system-dependent features in this case.

Object-Oriented

A programming language that can model the real world is said to be object-oriented. It focuses on objects and combines data and functions. Contrarily, a procedure-oriented language revolves around functions, which are code that can be reused. Python supports both procedure- oriented and object-oriented programming which is one of the key python features. It also supports multiple inheritance, unlike Java. A class is a blueprint for such an object. It is an abstract data type and holds no values.

There are five pilliar's of objecte-oriented concepts are:

- classes & objects it is used for structure of the program or software.
- Abstraction is used for hidding of unnecessary information from the client side.
- Polymorphism means multiple forms or more than one form. Different types of polymorphism are:

  1) compile time – method over loading.

  2) run time – method over riding.
- Encapsulation is the process of combining attributes and methods with in the class. It is used for data protection and private.
- Inhertiance is the process of aquaring the properties of one class to the another class. It have different types are single, multilevel, multiple, hierarchical.

Embeddable

We just saw that we can put code in other languages in our Python source code. However, it is also possible to put our Python code in a source code in a different language like C++.This allows us to integrate scripting capabilities into our program of the other language.

Large Standard Library

Python downloads with a large library that you can use so you don't have to write your own code for everything. There are libraries for regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and a lot of other functionality. Python have different types of library's to implement their software's library are pygame, pyTorch, scikit-learn, pyBrain etc,.

GUI Programming

Software is not user-friendly until its GUI is made. A user can easily interact with the software with a GUI. Python offers various libraries for making Graphical user interface for your applications. For this, you can use Tkinter, wxPython or JPython. These toolkits allow you for easy and fast development of GUI.

Dynamically Typed

Python is dynamically typed. This means that the type for a value is decided at runtime, not in advance. This is why we don't need to particular the type of data while declaring it. This is all about the features of the python programming language.

5.2 Technologies used:

Scikit-learn:

Scikit-learn is probably the most useful library for machine learning in python. The sklearn Library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

It was originally called *scikits.learn* and was initially developed by David Cournapeau a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

Numpy:

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 was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas:

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

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. Relevant data is very important in data science.

Matplotlib:

In applied Statistics and Machine Learning, Data Visualization is one of the most important skills that helps in the qualitative understanding of the data at hand. This proves to help explore and extract relevant information from the data by identifying patterns, relationships, outliers, and much more. The article explores the concept of Matplotlib in Machine Learning.

Tkinter:

Tkinter tutorial provides basic and advanced concepts of Python Tkinter. Our Tkinter tutorial is designed for beginners and professionals. Python provides the standard library Tkinter for creating the graphical user interface for desktop based applications. Developing desktop based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps.

- Import the Tkinter module.
- Create the main application window.
- Add the widgets like labels, buttons, frames, etc. to the window.
- Call the main event loop so that the actions can take place on the user's computer screen.

SQLITE DATABASE:

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your needs with your application. SQLite accesses its storage files directly.

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is one of the fastest-growing database engines around, but that's growth in terms of acceptance, not anything to do with its size. The source code for SQLite is in the public area.

Databases offer countless functionalities by which one can manage large amounts of details easily over the web and high-volume data input and output over a typical file such as a text file. SQL is a query language and is very popular in databases. Many websites use MySQL. SQLite is a “light” version that works over syntax very much similar to SQL. SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine on the world wide web. Python has a library to access SQLite databases, called sqlite3, intended for working with this database which has been included with Python package since version 2.5. SQLite has the following features.

- Serverless
- Self-Contained
- Zero-Configuration
- Transactional
- Single-Database

The SQLite3 provides a standardized python DBI API 2.0 compliant interface to the SQLite database. If your application need to support not only the SQLite database but also other databases such as MySQL, PostgreSQL, and Oracle is a good choice.

CHAPTER-6
SAMPLE CODE

6. SAMPLE CODE

```
#Importing Libraries from matplotlib to visualize the data
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

#Importing Libraries to create GUI
from tkinter import *

#Importing Libraries to perform calculations
import numpy as np
import pandas as pd
import os
#List of the symptoms is listed here in list l1.
l1=['back_pain','constipation','abdominal_pain','diarrhoea','mild_fever','yellow_urine',
'yellowing_of_eyes','acute_liver_failure','fluid_overload','swelling_of_stomach',
'swelled_lymph_nodes','malaise','blurred_and_distorted_vision','phlegm','throat_irritation',
'redness_of_eyes','sinus_pressure','runny_nose','congestion','chest_pain','weakness_in_limbs',
'fast_heart_rate','pain_during_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','bell y_pain',

'abnormal_menstruation','dischromic_patches','watering_from_eyes','increased_appe tite','polyuria','family_history','mucoid_sputum','rusty_sputum','lack_of_concentratio n','visual_disturbances','receiving_blood_transfusion','receiving_unsterile_injections' ,'coma','stomach_bleeding','distention_of_abdomen','history_of_alcohol_consumptio n','fluid_overload','blood_in_sputum','prominent_veins_on_calf','palpitations','painfu l_walking','pus_filled_pimples','blackheads','scurring','skin_peeling','silver_like_dust ing','small_dents_in_nails','inflammatory_nails','blister','red_sore_around_nose','yell ow_crust_ooze']

#List of Diseases is listed in list disease.

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','Ost eoarthristis',

'Arthritis','(vertigo) Paroymsal Positional Vertigo','Acne','Urinary tract infection','Psoriasis',

'Impetigo']

l2=[]

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

l2.append(0)

print(l2)

#Reading the training .csv file

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

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

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,'Peptic ulcer 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)

#printing the top 5 rows of the training dataset

df.head()

 # Distribution graphs (histogram/bar graph) of column data

def plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow):

nunique = df1.nunique()

df1 = df1[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displaying purposes, pick columns that have between 1 and 50 unique values

nRow, nCol = df1.shape

columnNames = list(df1)

nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow

plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, facecolor = 'w',

edgecolor = 'k')

```
for i in range(min(nCol, nGraphShown)):


plt.subplot(nGraphRow, nGraphPerRow, i + 1)
columnDf = df.iloc[:, i]
if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
valueCounts = columnDf.value_counts()
valueCounts.plot.bar()
else:
columnDf.hist()
plt.ylabel('counts')
plt.xticks(rotation = 90)
plt.title(f'{columnNames[i]} (column {i})')
plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
plt.show()
```

 # Scatter and density plots

```
def plotScatterMatrix(df1, plotSize, textSize):
df1 = df1.select_dtypes(include =[np.number]) # keep only numerical columns
```

 # Remove rows and columns that would lead to df being singular

```
df1 = df1.dropna('columns')
df1 = df1[[col for col in df if df[col].nunique() > 1]] # keep columns where there are more than 1 unique values
columnNames = list(df)
if len(columnNames) > 10: # reduce the number of columns for matrix inversion of kernel density plots
columnNames = columnNames[:10]
df1 = df1[columnNames]
ax = pd.plotting.scatter_matrix(df1, alpha=0.75, figsize=[plotSize, plotSize], diagonal='kde')
```

```
corrs = df1.corr().values


for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):



ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes fraction',
ha='center', va='center', size=textSize)

plt.suptitle('Scatter and Density Plot')

plt.show()

plotPerColumnDistribution(df, 10, 5)

plotScatterMatrix(df, 20, 10)

X= df[l1]

y = df[["prognosis"]]

np.ravel(y)

print(X)

print(y)
```

#Reading the testing.csv file

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

#Using inbuilt function replace in pandas for replacing the values

```
tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug
Reaction':4,

'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial
Asthma':9,'Hypertension ':10,

'Migraine':11,'Cervical spondylosis':12,

'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken
pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,
```

'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatitis':24,'Tuberculosis':25,

'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Varicose veins':30,'Hypothyroidism':31,

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

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

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

#printing the top 5 rows of the testing data

tr.head()

plotPerColumnDistribution(tr, 10, 5)

plotScatterMatrix(tr, 20, 10)

X_test= tr[l1]

y_test = tr[["prognosis"]]

np.ravel(y_test)

print(X_test)

print(y_test)

#list1 = DF['prognosis'].unique()

def scatterplt(disea):

x = ((df.loc[disea]).sum())#total sum of symptom reported for given disease

x.drop(x[x==0].index,inplace=True)#droping symptoms with values 0

print(x.values)

y = x.keys()#storing nameof symptoms in y

print(len(x))

print(len(y))

plt.title(disea)

plt.scatter(y,x.values)

```
plt.show()
def scatterinp(sym1,sym2,sym3,sym4,sym5):

x = [sym1,sym2,sym3,sym4,sym5]#storing input symptoms in y
y = [0,0,0,0,0]#creating and giving values to the input symptoms
if(sym1!='Select Here'):
y[0]=1
if(sym2!='Select Here'):

y[1]=1
if(sym3!='Select Here'):
y[2]=1
if(sym4!='Select Here'):
y[3]=1
if(sym5!='Select Here'):
y[4]=1
print(x)
print(y)
plt.scatter(x,y)
plt.show()
root = Tk()
pred1=StringVar()
def DecisionTree():
if len(NameEn.get()) == 0:
pred1.set(" ")
comp=messagebox.askokcancel("System","Kindly Fill the Name")
```

```
if comp:
root.mainloop()
elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):

pred1.set(" ")

sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
if sym:
root.mainloop()
pred2=StringVar()
def randomforest():
if len(NameEn.get()) == 0:
pred1.set(" ")

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

 # calculating accuracy

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
y_pred=clf4.predict(X_test)
print("Random Forest")
print("Accuracy")
print(accuracy_score(y_test, y_pred))



print(accuracy_score(y_test, y_pred,normalize=False))
print("Confusion matrix")
conf_matrix=confusion_matrix(y_test,y_pred)
print(conf_matrix)
psymptoms =
[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
for k in range(0,len(l1)):
for z in psymptoms:
inputtest = [l2]
predict = clf4.predict(inputtest)

predicted=predict[0]
h='no'
for a in range(0,len(disease)):
if(predicted == a):
h='yes'
break
if (h=='yes'):
pred2.set(" ")
```

```
pred2.set(disease[a])

else:

pred2.set(" ")

pred2.set("Not Found")
```

#Creating the database if not exists named as database.db and creating table if not exists named as

```
RandomForest using sqlite3

import sqlite3

conn = sqlite3.connect('database.db')

c = conn.cursor()


c.execute("CREATE TABLE IF NOT EXISTS RandomForest(Name StringVar,Symtom1

StringVar,Symtom2 StringVar,Symtom3 StringVar,Symtom4 TEXT,Symtom5 TEXT,Disease StringVar)")

c.execute("INSERT INTO RandomForest(Name,Symtom1,Symtom2,Symtom3,Symtom4,Symtom5,Disease) VALUES(?,?,?,?,?,?,?)",
(NameEn.get(),Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get(),pred2.get()))

conn.commit()

c.close()

conn.close()
```

#printing scatter plot of disease predicted vs its symptoms

```


scatterplt(pred2.get())

t1=Label(root,font=("Times",15,"bold italic"),text="Decision Tree",height=1,bg="Light green"

,width=40,fg="red",textvariable=pred1,relief="sunken").grid(row=15, column=1, padx=10)

t2=Label(root,font=("Times",15,"bold italic"),text="Random Forest",height=1,bg="Purple"
```

```
,width=40,fg="white",textvariable=pred2,relief="sunken").grid(row=17, column=1,
padx=10)

root.mainloop()
```

CHAPTER-7
TESTING

7. TESTING

7.1 Introduction

In normally, software developers discriminate system faults or errors from software failures. In case of a defect, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. Software testing is nothing but an art of research software to ensure that its quality under test is in line with the needs of the client. Software testing is carried out in a systematic manner with the intent of finding defects in a mode. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

System Testing and Implementation

The purpose is to activity the different parts of the module code to finding coding errors. After this the modules are gradually fused into subsystems, which are then integrated themselves too the end forming the entire mode. During integration of module integration testing is fulfill. The aim of this is to detect designing defects,while focusing the interconnection between modules. After the system was put together, system testing is performed. Here the system is tested against the system requirements to see if all requirements were met, and the system performs as specified by the requirements. Finally accepting testing is performed to demonstrate to the client for the operation of the mode.

For the testing to be successful, proper picking of the test case is crucial. There are two varity of approaches for choicing test case. The software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called “black box testing”.

The focus here is on testing the external behaviour of the system. In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. The two forms of testing are complementary: one tests the external behaviour, the other tests the internal structure.

This plan identifies all testing related activities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested; different units to be tested, and the manner in which the module will be integrated together. Then for different test unit, a test case specification document is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual results are compared with the expected outputs. The final output of the testing phase is the testing report and the error report, or a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the errors encountered and the action taken to remove the error.

Testing Techniques

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected. To make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Black Box Testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

White Box Testing

White box testing the tester knows the code behind functionality and uses that knowledge for testing purpose, it is called white box testing. This technique is in the internal structure of the software system used data structures.

In this testing, the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. Ithas been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational.
- Execute internal data structures to ensure their validity.

Testing Strategies

Unit testing:

Unit testing is defined as a type of software testing individual ingredient of a software are tested.

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is

the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

This System consists of 3 modules. Those are Reputation module, route discovery module, audit module. Each module is taken as unit and tested. Identified errors are corrected and executable unit are obtained.

Integration testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items.

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes, and successive processes.

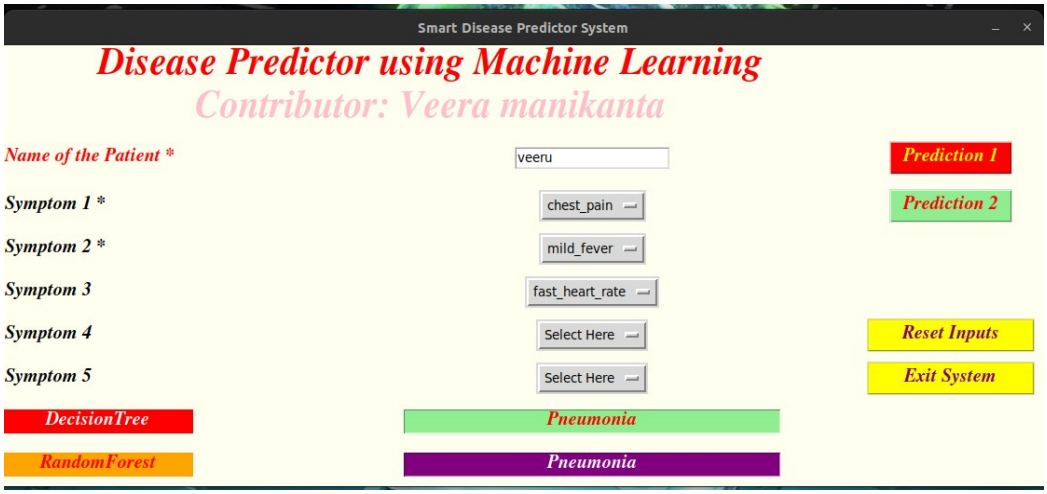
7.2 SAMPLE TEST CASES SPECIFICATION

Test case id	Test case name	Input	Expected Output	Observed Output	Result
T1	When user can enter the valid symptom at-least two.	Enter the symptom	It gives the predicted disease.	User can see the result successful.	Pass
T2	When user can enter the invalid input like not give symptoms.	Can’t gives the symptom	It’s can’t given the output.	User can’t see the result.	pass

Table 7.2: Sample Test Cases Specification

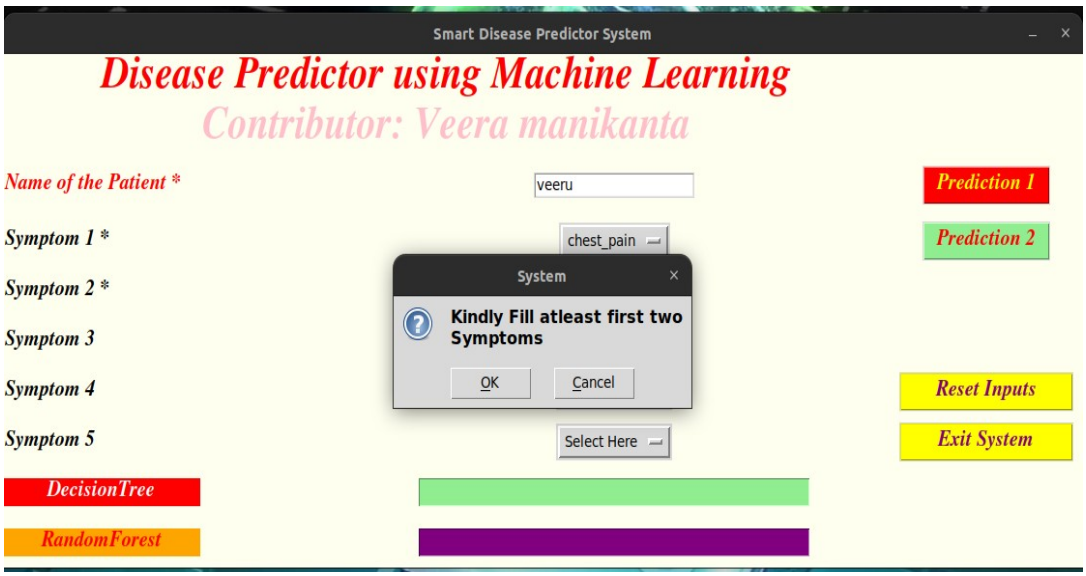
7.1 Test case Screenshots

Test case T1:



7.3.1 Test Screen for User give symptoms

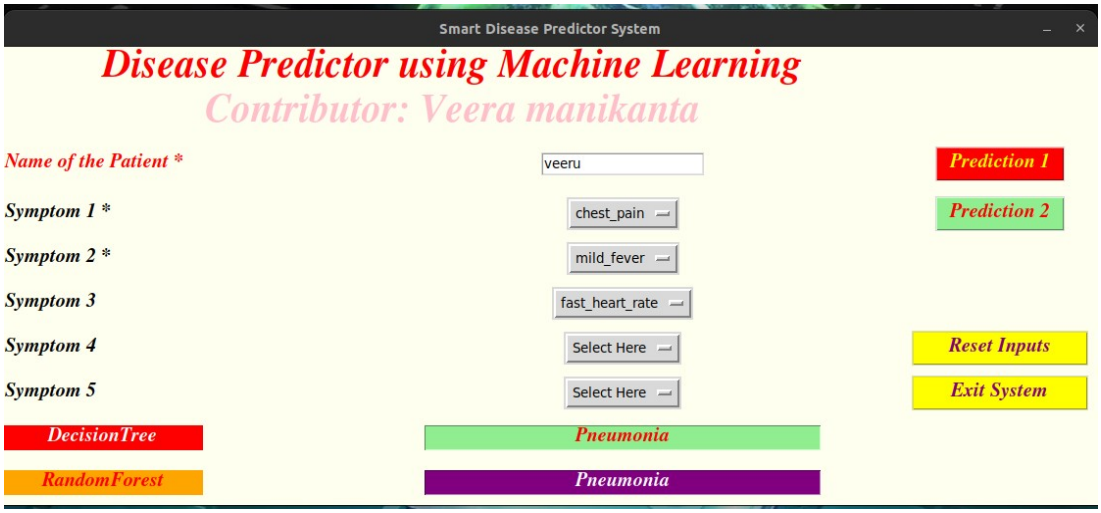
Test case T2:



7.3.2 Test Screen for User can not give symptoms

CHAPTER-8
SCREENSHOT

8. SCREENSHOTS



8.1: When user can check their disease based on symptom

Description: The above screenshot displays, when the user can see their disease based on the given symptoms. The above two models can gives accuract prediction.

prognosis	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_pimples	blackheads	scurring	skin_peeling
Fungal infection	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0
Fungal infection	0	1	1	0	0	0	0	0	0	0	...	0	0	0	0
Fungal infection	1	0	1	0	0	0	0	0	0	0	...	0	0	0	0
Fungal infection	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0
Fungal infection	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0

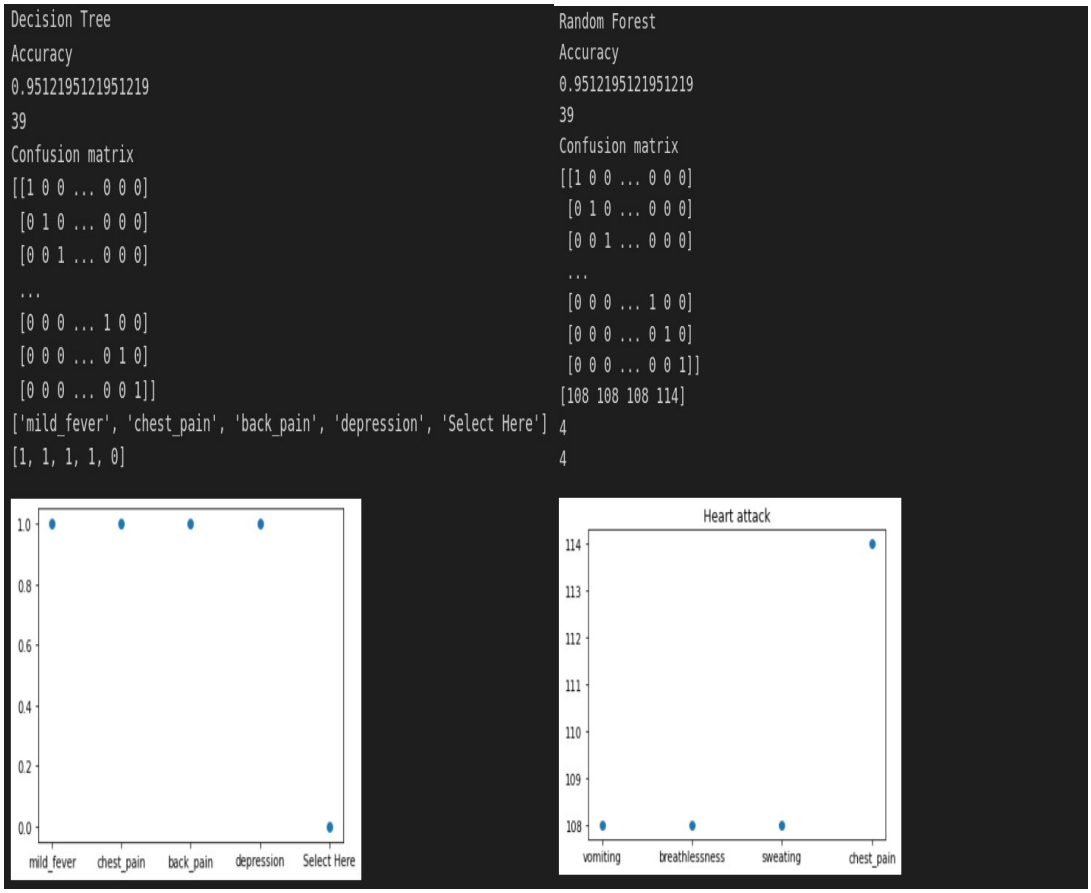
5 rows × 132 columns

8.2: First five rows in the dataset

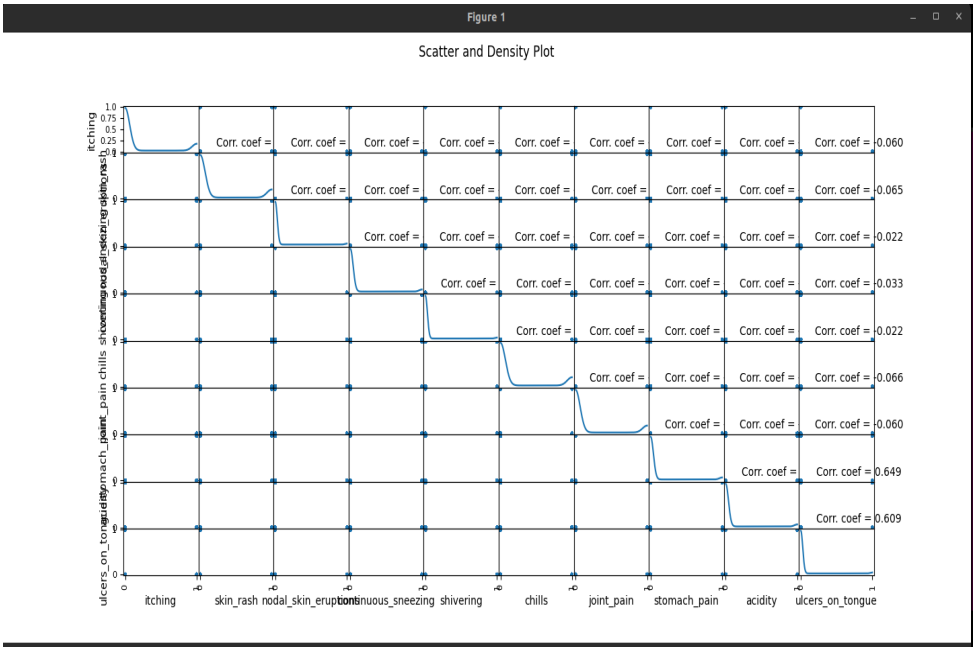
prognosis	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_pimples	blackheads	scurring	skin_peeling
(vertigo) Paroymsal Positional Vertigo	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
Acne	0	1	0	0	0	0	0	0	0	0	...	1	1	1	0
Urinary tract infection	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
Psoriasis	0	1	0	0	0	0	1	0	0	0	...	0	0	0	1
Impetigo	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0

5 rows × 132 columns

8.3: Last five rows in the dataset



8.4: Accuracy of the both models



8.5: Scatter and Density plot of the dataset

CONCLUSION

CONCLUSION

The system we implement is well trained with different machine learning algorithms and the best performed models are used for final predictions. The machine learning algorithms reached accuracy more than 90%. It is a pretty good accuracy for any machine learning model. So, there is a high chance for the model to predict the illness of a patient more accurately given their symptoms. Now the system can be deployed online and will be available to everyone across the web. The system is able is predict the outcome quicky within few minutes. By collecting new and more data of patient records from hospitals, the model can be further improved. The suggested model predicts disease merely based on patient symptoms. In up comming day's, this model can be greatly improved by considering other factors such as patients age, sex, medical-condition, severity of symptoms, cause of symptoms, demographics, climate etc.

BIBLIOGRAPHY

BIBLIOGRAPHY

Textbooks referred:

- Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Muller & Sarah Guido, Orielly Publications, 2019.
- Python Machine Learning, Sebastial Raschka & Vahid Mirjalili, 3rd Edition, 2019
- Machine Learning using Python, Manaranjan Pradhan, U Dinesh Kumar, Wiley, 1st Edition, 2019.
- Machine Learning, Tom M.Mitchell, Mc Graw-Hill Publication, 2017.
- Building Machine Learning System with Python, Luis Pedro Coelho, Willi Richert, 2nd Edition, 2015.
- Programming and Problem Solving with Python, Ashok Namdev Kamthane, Amit Ashok Kamthane, TMH, 2019.

Web sites referred:

- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://www.analyticsvidhya.com/blog/2021/06/understanding- random-forest/
- https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset

Reference Paper:

- Rayan Alanazi proposed a method of identification and prediction of the presence of chronic disease in an individual using the machine learning algorithms such as CNN and KNN.
- Sneha Grampurohit in this paper presents a comprehensive comparative study of three algorithms performance on a medical record each yielding an accuracy up to 95 percent.

- Grampurohit this paper aim of developing classifier system using machine learning algorithms

20A91F0015

ORIGINALITY REPORT

39%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.coursehero.com Internet	815 words — 9%
2	www.slideshare.net Internet	665 words — 7%
3	data-flair.training Internet	463 words — 5%
4	www.scribd.com Internet	316 words — 4%
5	deepnote.com Internet	265 words — 3%
6	grietinfo.in Internet	231 words — 3%
7	searchsoftwarequality.techtarget.com Internet	130 words — 1%
8	origin.geeksforgeeks.org Internet	126 words — 1%
9	academicscience.co.in Internet	102 words — 1%
10	core.ac.uk Internet	

102 words — 1%

11	comp786.blogspot.com Internet	78 words — 1%
12	www.hindawi.com Internet	76 words — 1%
13	www.tutorialspoint.com Internet	71 words — 1%
14	www.dotnetcoders.com Internet	70 words — 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	< 70 WORDS