

DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY
KUDLU GATE
BANGALORE - 560068



MINOR PROJECT REPORT

ON

“Disease-Prediction Using Naive Bayes”

**SUBMITTED TO THE Vth SEMESTER
MINOR PROJECT (19CS3505)-2021**

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

PRARTHAN P - (ENG19CS0228)

Under the supervision of

Dr. BONDU VENKATESWARLU

-Associate Professor, DSU

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068



CERTIFICATE

This is to certify that PRARTHAN P - (ENG19CS0228) has satisfactorily completed his/her Mini Project Report as prescribed by the University for the 5 THsemester B.Tech. program in Computer Science & MINOR PROJECT course during the year 2021-22 at the School of Engineering, Dayananda Sagar University., Bangalore.

Date: _____

Signature of the faculty in-charge

Signature of Chairman

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this minor project entitled - “Diseases Prediction using Naïve Bayes”, has been carried out by us and it has not been submitted for the award of any degree, diploma or the minor project of any other college or university.

PRARTHAN P - (ENG19CS0228)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of a task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our Dean & Chairman **Dr. Srinivas A**, Department of Computer Science & Engineering, for providing necessary departmental facilities, moral support and encouragement.

We are very much thankful to **Dr. Bondu Venkateswarlu**, Associate Professor, Department of Computer Science & Engineering, for providing help and suggestions in completion of this minor project successfully.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directly or indirectly helped us in the successful completion of this project work.

PRARTHAN P - (ENG19CS0228)

ABSTRACT

Disease Prediction using Machine Learning is a system which predicts the disease based on the information or the symptoms he/she enters into the system and provides the accurate results based on that information. If the patient is not very serious and the user just wants to know the type of disease, he/she has been through. It is a system which provides the user the tips and tricks to maintain the health system of the user and it provides a way to find out the disease using this prediction. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. This Disease Prediction Using Machine Learning is completely done with the help of Machine Learning and Python Programming language with Tkinter Interface for it and also using the dataset that is available previously by the hospitals using that we will predict the disease

CONTENTS

COVER PAGE	I
CERTIFICATE	II
DECLARATION	III
ACKNOWLEDGEMENT	IV
ABSTRACT	V
1. INTRODUCTION	
1.1. DISEASE PREDICTION	1
1.2. PROBLEM DEFINITION	2
1.3. PROJECT PURPOSE	3
1.4. PROJECT FEATURES	3
2. LITERATURE SURVEY	
2.1. MACHINE LEARNING	5
2.1.1. FEATURES OF MACHINE LEARNING	6
2.2. EXISTING SYSTEM	8
2.3. PROPOSED SYSTEM	9
2.4. SOFTWARE DESCRIPTION	10
2.4.1. PYTHON	10
2.4.2. BENEFITS OF PYTHON	10
2.4.3. TKINTER INTERFACE	11
3. REQUIREMENT ANALYSIS	
3.1. FUNCTIONAL REQUIREMENTS	12
3.2. NON-FUNCTIONAL REQUIREMENTS	13
3.3. HARDWARE REQUIREMENTS	15
3.4. SOFTWARE REQUIREMENTS	15

4. DESIGN

4.1. DESIGN GOALS	16
4.2. SYSTEM ARCHITECTURE	17
4.3. DATA FLOW DIAGRAM	18
4.4. CLASS DIAGRAM	19
4.5. SEQUENCE DIAGRAM	20
4.6. USE CASE DIAGRAMS	21
4.7. ACTIVITY DIAGRAM	22
4.8. COMPONENT DIAGRAM	23
4.9. STATE CHART DIAGRAM	24
4.10. COLLABORATION DIAGRAM	25
4.11. DEPLOYMENT DIAGRAM	26
4.12. INTERFACE AND FRAMEWORK DIAGRAM	27

5. IMPLEMENTATION

5.1 OVERVIEW	29
5.2 NAÏVE BAYES ALGORITHM	31
5.3 CODE SNIPPET	33

6. TESTING

6.1 UNIT TESTING	33
6.2 INTEGRATION TESTING	33
6.3 VALIDATION TESTING	33
6.4 SYSTEM TESTING	34
6.5 TESTING OF INITIALIZATION AND UI COMPONENTS	35

7. SNAPSHOTS

8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION	43
8.2 FUTURE ENHANCEMENT	43

CHAPTER 1

INTRODUCTION

1.1 DISEASE PREDICTION

Disease Prediction using Machine Learning is a system which predicts the disease based on the information provided by the user. It also predicts the disease of the patient or the user based on the information or the symptoms he/she enters into the system and provides the accurate results based on that information. If the patient is not very serious and the user just wants to know the type of disease, he/she has been through. It provides a way to find out the disease using this prediction. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. This Disease Prediction Using Machine Learning is completely done with the help of Machine Learning and Python Programming language with Tkinter Interface for it and also using the dataset that is available previously by the hospitals using that we will predict the disease. Nowadays doctors are adopting many scientific technologies and methodology for both identification and diagnosing not only common diseases, but also many fatal diseases. The successful treatment is always attributed to the right and accurate diagnosis. Doctors may sometimes fail to make accurate decisions while diagnosing the disease of a patient, therefore disease prediction systems which use machine learning algorithms assist in such cases to get accurate results. The project disease prediction using machine learning is developed to overcome general disease in earlier stages as we all know in the competitive environment of economic development mankind has been involved so much that he/she is not concerned about health. According to research there are 40% of people who ignore general disease which leads to harmful disease later. The main reason for ignorance is laziness to consult a doctor and time concerns. The people have involved themselves so much that they have no time to take an appointment and consult the

doctor which later results in fatal disease. According to research there are 70% peoples in India suffers from general disease and 25%of peoples face death due to early ignorance the main motive to develop this project is that a user can sit at their convenient place and have a check-up of their health the UI is designed in such a simple way that everyone can easily operate on it and can have a check-up.

1.2 PROBLEM DEFINITION

Now a days in Health Industry there are various problems related to machines or devices which will give wrong or unaccepted results, so to avoid those results and get the correct and desired results we are building a program or project which will give the accurate predictions based on information provided by the user and also based on the datasets that are available inthat machine. The health industry is information yet and knowledge poor and this industry isa very vast industry which has a lot of work to be done. So, with the help of Naive Bayes algorithm we have done this project which will help the people who are in need. So the problemhere is that many people goes to hospitals or clinic to know how is their health and how muchthey are improving in the given days, but they have to travel to get to know there answers andsometimes the patients may or may not get the results based on various factors such as doctormight be on leave or some whether problem so he might not have come to the hospital and many more reasons will be there so to avoid all those reasons and confusion we are making a project which will help all those person's and all the patients who are in need to know the condition of their health, and at sometimes if the person has been observing few symptoms andhe/she is not sure about the disease he/she is encountered with so this will lead to various diseases in future. So, to avoid that and get to know the disease in early stages of the symptoms,this disease prediction will help a lot to the various people ranging from children to teenagersto adults and also the senior citizens.

1.3 PROJECT PURPOSE

The purpose of making this project called “Disease Prediction Using Machine Learning” is to predict the accurate disease of the patient using all their general information and also the symptoms. Using this information, there we will compare with our previous datasets of the patients and predict the disease of the patient he/she has been through. If this Prediction is done at the early stages of the disease with the help of this project and all other necessary measures the disease can be cured and in general this prediction system can also be very useful in the health industry. If the health industry adopts this project, then the work of the doctors can be reduced and they can easily predict the disease of the patient. The general purpose of this Disease prediction is to provide predictions for the various and generally occurring diseases that when unchecked and sometimes ignored can turn into fatal disease and cause a lot of problems to the patient and as well as their family members. This system will predict the most possible disease based on the symptoms. The health industry is information yet and knowledge poor and this industry is a very vast industry which has a lot of work to be done. So, with the help of Naive Bayes algorithm we have done this project which will help the people who are in need.

1.4 PROJECT FEATURES

The features of Disease Prediction Using Machine Learning are as follows.

- This Project will predict the diseases of the patients based on the symptoms and other general information using the datasets.
- With the help of Disease prediction, it can predict the disease of the patient and can solve various problems and prevent various aspects.
- It provides security for the system so that no one can break into that and no one can make any changes in the system.

- The disease is predicted using the algorithm and the user has to enter the symptoms from the given drop-down menu, in order to get correct accuracy, the user has to enter all the symptoms.
- Here we can easily prepare the data and transform that data into algorithms, which will reduce the overall work of the project.
- To make users more application friendly rather than discussing with others about their disease.
- Here the user has to register first, in order to use the prediction and then login to the system using the credentials such as username and password.
- Once a user opens the system to login, the user needs to register by clicking on the register/signup button.
- After which the user needs to provide some basic details of signup and then the details of the user are saved in the system.

CHAPTER 2

LITERATURE SURVEY

2.1 MACHINE LEARNING

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is a combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationships between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Trees, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning tasks Machine learning tasks are typically classified into several broad categories:

Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback.

Semi-supervised learning: The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

Active learning: The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

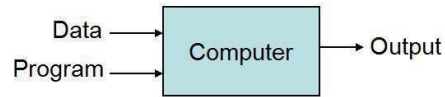
Reinforcement learning: Data (in form of rewards and punishments) are given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

2.1.1 FEATURES OF MACHINE LEARNING

- It is nothing but Automation.
- Getting computers to program themselves.
- Writing Software is a bottleneck.
- Machine learning models involve machines learning from data without the help of humans or any kind of human intervention.
- Machine Learning is the science of making computers learn and act like humans by feeding data and information without being explicitly programmed.

- Machine Learning is totally different from traditional programming, here data and output is given to the computer and in return it gives us the program which provides solutions to the various problems. Below is the figure.

Traditional Programming



Machine Learning

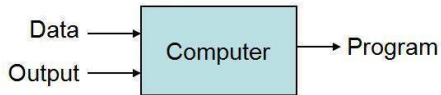
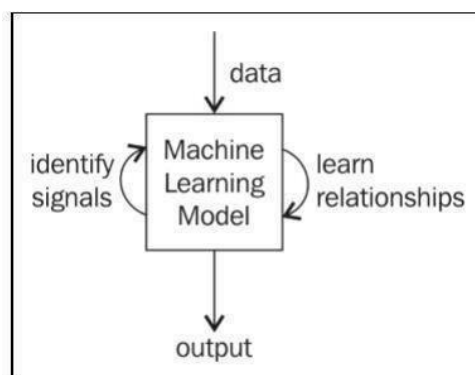


Fig 2.1.1 Traditional Programming vs Machine Learning

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There are Many Algorithms in Machine Learning through which we will provide us the exact solution in predicting the disease of the patients.
- How Does Machine Learning Works?
- Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.



An overview of machine learning models

Fig 2.1.2 Machine Learning Model

- There are various applications in which machine learning is implemented such as Web search, computing biology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.
- There are 3 types of machine learning: supervised, unsupervised, and reinforcement.

2.2 EXISTING SYSTEM

Prediction using traditional methods and models involves various risk factors and it consists of various measures of algorithms such as datasets, programs and much more to add on. High-risk and Low-risk patient classification is done on the basis of the tests that are done in groups. But these models are only valuable in clinical situations and not in big industry sectors. So, to include the disease predictions in various health related industries, we have used the concepts of machine learning and supervised learning methods to build the predictions system.

After doing the research and comparison of all the algorithms and theorems of machine learning we have come to conclusion that Naïve Bayes Algorithm is important in building a disease prediction system which predicts the disease of the patients from which he/she is suffering from. After using various techniques such as neuralnetworks to make predictions of the diseases and after doing that we come to the conclusion that **it can predict up to 90% accuracy rate** after doing the experimentation and verifying the results. The information of patient statistics, results, and disease history is recorded, which enables us to identify the potential data centric solution, which reduces the cost of medical case studies. Existing system can predict the disease but not the subtype of the disease and it fails to predict the condition of the people, the predictions of disease have been indefinite and non-specific.

2.3 PROPOSED SYSTEM

The proposed system of disease prediction using machine learning is that we have used many techniques and algorithms and all other various tools to build a system which predicts the disease of the patient using the symptoms and by taking those symptoms we are comparing with the system's dataset that is previously available. By taking those datasets and comparing with the patient's disease we will predict the accurate percentage disease of the patient. The dataset and symptoms go to the prediction model of the system where the data is pre-processed for the future references and then the feature selection is done by the user where he will enter the various symptoms. Then the classification of those data is done with the help of an algorithm by Naïve Bayes. Then the data goes in the recommendation model, there it shows the risk analysis that is involved in the system and it also provides the probability estimation of the system such that it shows the various probability like how the system behaves when there are n number of predictions are done and it also does the recommendations for the patients from their final result and also from their symptoms like it can show what to use and what not to use from the given datasets and the final results. Here we have combined the overall structure and unstructured form of data for the overall risk analysis that is required for doing the prediction of the disease. Using structured analysis, we can identify the chronic types of disease in a particular region and particular community. In unstructured analysis we select the features automatically with the help of algorithms and techniques.

2.4 SOFTWARE DESCRIPTION

2.4.1 PYTHON

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects. Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution. An important goal of Python's developers is keeping it fun to use. Python's design offers some support for functional programming.

2.4.2 BENEFITS OF PYTHON

- Presence of Third-Party Modules
- Extensive Support Libraries
- Open Source and Community Development
- Learning Ease and Support Available
- User-friendly Data Structures
- Productivity and Speed
- Highly Extensible and Easily Readable Language.

2.4.3 TKINTER INTERFACE

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and Mac OS X installs of Python. The name Tkinter comes from the Tk interface. Tkinter was written by Fredrik Lundh. Tkinter is free software released under a Python license.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete Tool Command Language (TCL) interpreter embedded in the Python interpreter. Tkinter calls are translated into TCL commands which are fed to this embedded interpreter, thus making it possible to mix Python and TCL in a single application. In Tkinter, the Frame widget is the basic unit of organization for complex layouts. A frame is a rectangular area that can contain other widgets. When any widget is created, a parent- child relationship is created. For example, if you place a text label inside a frame, the frame is the parent of the label.

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications.

To create a tkinter:

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.

Importing tkinter is the same as importing any other module in the python code. Note that The name of the module in Python 2.x is 'Tkinter' and in Python 3.x is 'tkinter'.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

- Functional Requirements concerns the specific functions delivered by the system.
So, Functional requirements are statements of the services that the system must provide.
- The functional requirements of the system should be both complete and consistent
- Completeness means that all the services required by the user should be defined.
- Consistency means that requirements should not have any contradictory definitions.
- The requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.
- Take user id and password match it with corresponding file entries. If a match is found then continue else raise an error message.

3.2 NON-FUNCTIONAL REQUIREMENTS

- Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.
 - The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies, etc.
1. **Performance requirement:** All data entered shall be up to mark and no flaws shall be there for the performance to be 100%.
 2. **Platform constraints:** The main target is to generate an intelligent system to predict the adult height.
 3. **Accuracy and Precision:** Requirements are accuracy and precision of the data
 4. **Modifiability:** Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).
 5. **Portability:** Since a mobile phone is handy it is portable and can be carried and used whenever required.
 6. **Reliability:** Requirements about how often the software fails. The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error Prediction, and a strategy for correction.
 7. **Security:** One or more requirements about protection of your system and its data.
 8. **Usability:** Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the cloud can access the cloud to store and retrieve their data with the help of a secret key sent to their email ids. User interface is simple and efficient and easy to use.

MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities that can be added in the project based on the user requirements just by adding the appropriate files to existing projects using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

SCALABILITY:

System is capable of handling increased total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meets its minimum configurations. Only system files and dependent assemblies would have to be configured in such a case.

VALIDATION:

It is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e The software meets the user requirements, not as specification artifacts or as needs of those who will operate the software only; but, as the needs of all the stakeholders.

3.3 HARDWARE REQUIREMENTS

- ❖ System : Pentium 4, Intel Core i3, i5, i7 and 2 GHz Minimum
- ❖ RAM : 2GB or above
- ❖ Hard Disk : 10 GB or above

3.4 SOFTWARE REQUIREMENTS

- ❖ Operating System : Windows 7, 10 or Higher Versions
- ❖ Platform : Jupyter Notebook
- ❖ Front End : Python Tkinter
- ❖ Back End : Python, Files and SQL Database
- ❖ Programming Language : Python

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

The Design goals consist of various designs which we have implemented in our system disease prediction using machine learning. This system is built with various designs such as data flow diagram, sequence diagram, class diagram, use case diagram, component diagram, activity diagram, state chart diagram, deployment diagram. After doing these various diagrams and based on these diagrams we have done our project.

We have designed our system in such a way that whenever a user logs in into the system, the user has to register to the system, and new users cannot use the system without registering in the system. After that for registration the user requires basic credentials such as username, age, email, phone, password. Then the user has to login to the system using the same username and password. Here are the things that this system can perform.

- a. Entering Symptoms
- b. Disease Prediction

Entering Symptoms: Once a user successfully logs in to the system then he/she has to select the symptoms from the given drop-down menu.

Disease prediction: The predictive model predicts the disease of a person he might have, based on the user entered symptoms.

4.2 SYSTEM ARCHITECTURE

Disease prediction using machine learning predicts the presence of the disease for the user based on various symptoms and the information the user gives such as sugar level, hemoglobin level and many more such general information through the symptoms. The architecture of the system disease prediction using machine learning consist of various datasets through which we will compare the symptoms of the user and predicts it, then the datasets are transformed into the smaller sets and from there it gets classified based on the classification algorithms later on the classified data is then processed into the machine learning technologies through which the data gets processed and goes in to the disease prediction model using all the inputs from the user that is mentioned above. Then after the user enters the above information and overall processed data combines and compares in the prediction model of the system and finally predicts the disease. An architecture diagram is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements and components. The diagram explains about the system software in perception of overview of the system.

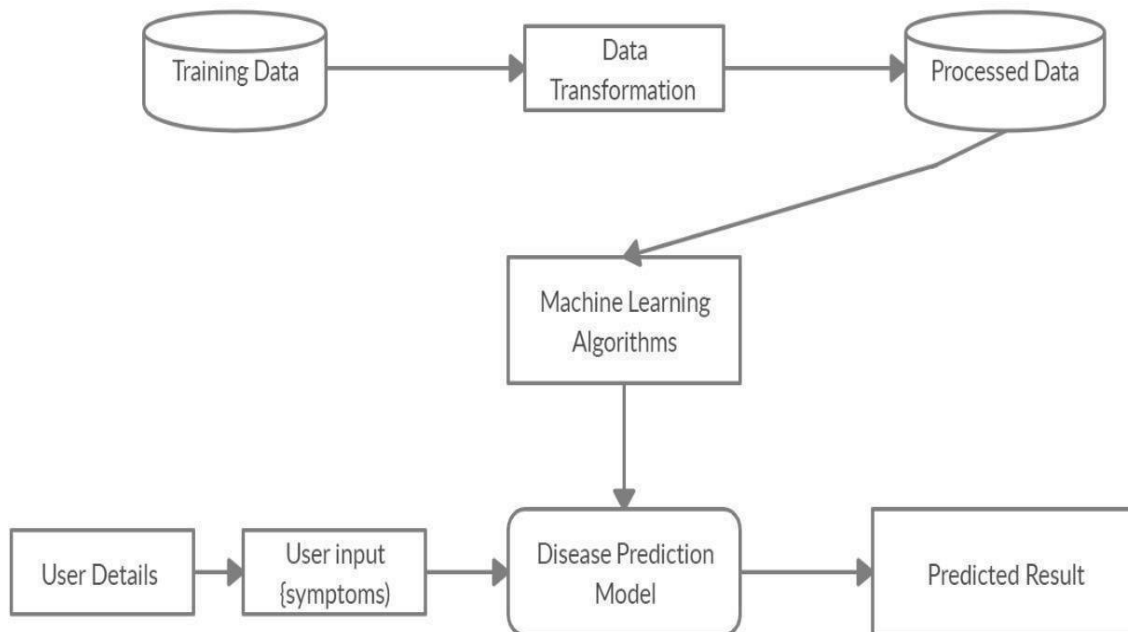


Fig 4.2 System Architecture

4.3 DATA FLOW DIAGRAM

The data flow diagram of the project disease prediction using machine learning consists of all the various aspects a normal flow diagram requires. This data flow diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information.

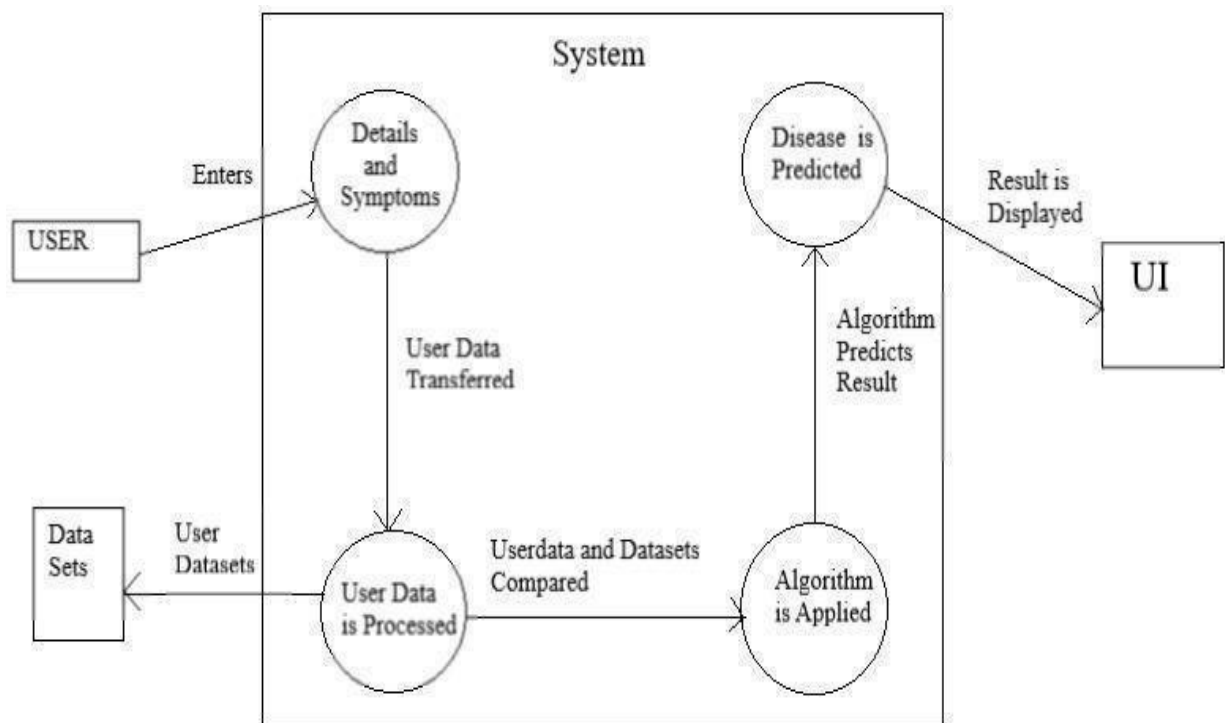


Fig 4.3 Data Flow Diagram

4.4 CLASS DIAGRAM

Disease prediction using machine learning consists of a class diagram that all the other applications that consist of the basic class diagram, here the class diagram is the basic entity that is required in order to carry on with the project. Class diagram consist information about all the classes that is used and all the related datasets, and all the other necessary attributes and their relationships with other entities, all these information is necessary in order to use the concept of the prediction, where the user will enter all necessary information such as user name, email, phone number, and many more attributes that is required in order to login into the system and using the files concept we will store the information of the users who are registering into the system and retrieves those information later while logging into the system.

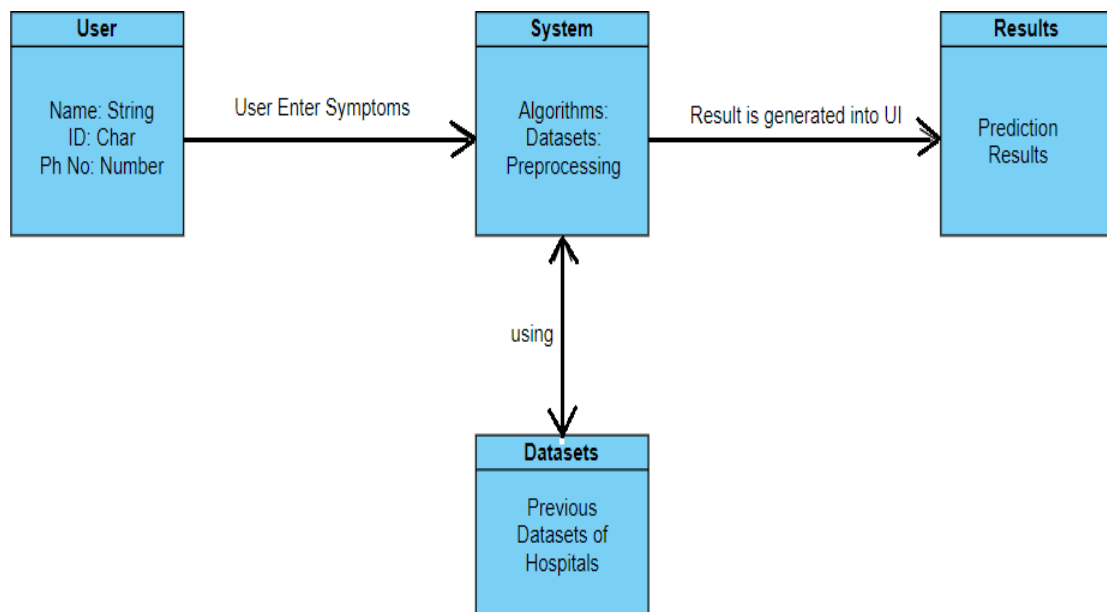


Fig 4.4 Class Diagram

4.5 SEQUENCE DIAGRAM

The Sequence diagram of the project disease prediction using machine learning consists of all the various aspects a normal sequence diagram requires. This sequence diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the sequence of all the entities are linked to each other where the user gets started with the system.

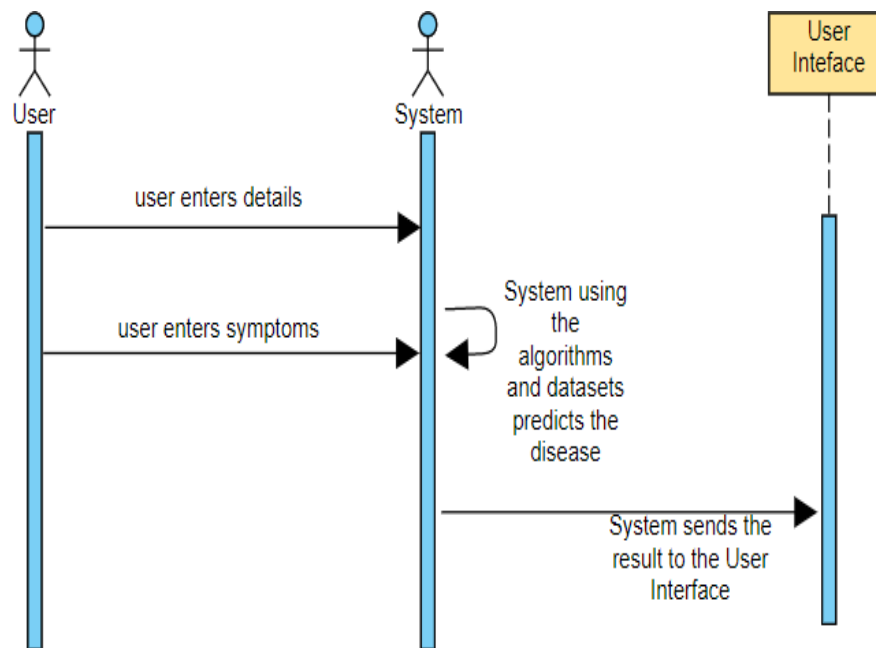


Fig 4.5 Sequence Diagram

4.6 USE CASE DIAGRAM

The Use Case diagram of the project disease prediction using machine learning consists of all the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities is linked to each other where the user gets started with the system.

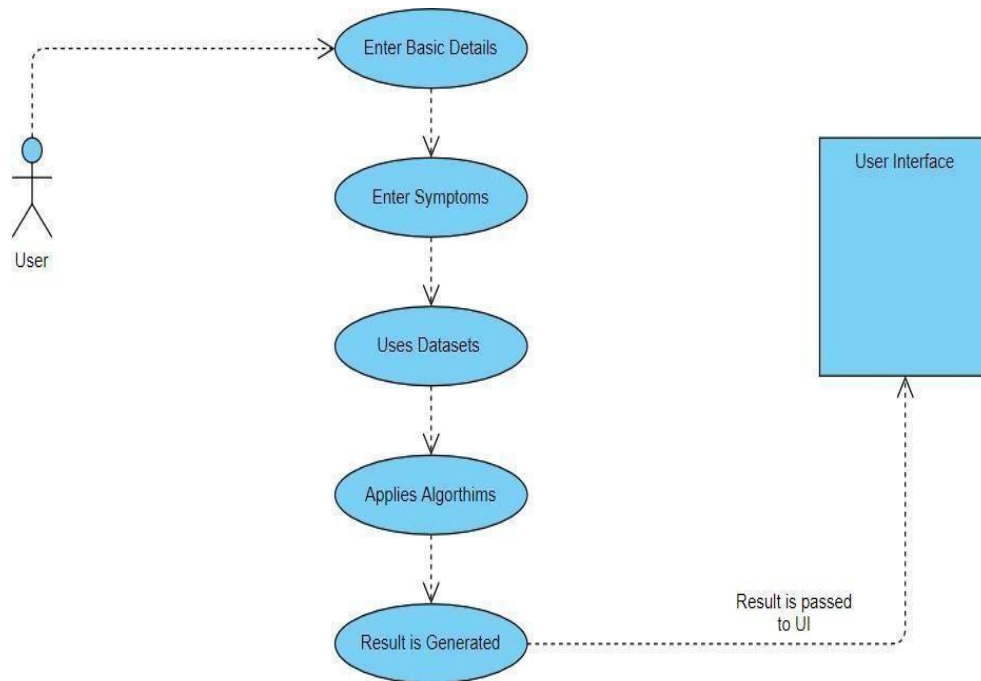


Fig 4.6 Use Case Diagram

4.7 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. Here in this diagram the activity starts from the user where the user registers into the system then login using the credentials and then the credentials are matched in the system and if it's true, then the user proceeds to the prediction phase where the prediction happens. Then finally after processing the data from datasets the analysis will happen then the correct result will be displayed that is nothing but the Output.

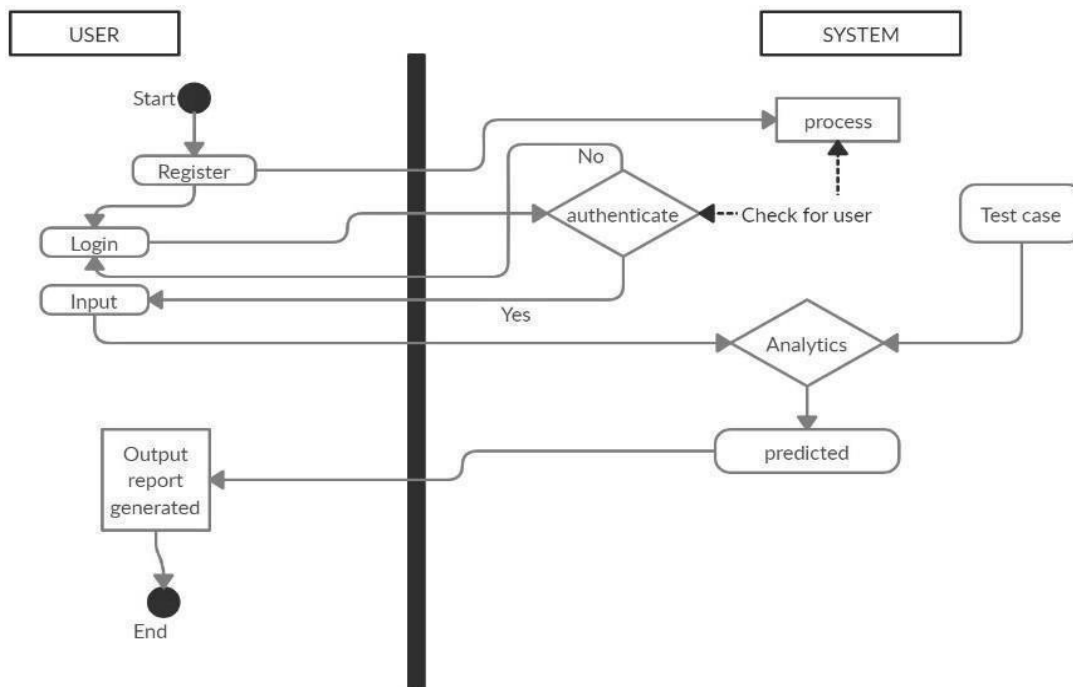


Fig 4.7 Activity Diagram

4.8 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development. Here the component diagram consists of all major components that are used to build a system. So, Design, Algorithm, File System and Datasets all are linked to one another. Datasets are used to compare the results and an algorithm is used to process those results and give a correct accuracy and design UI is used to show the result in an appropriate way in the system and file system is used to store the user data. So, like this all components are interlinked to each other.

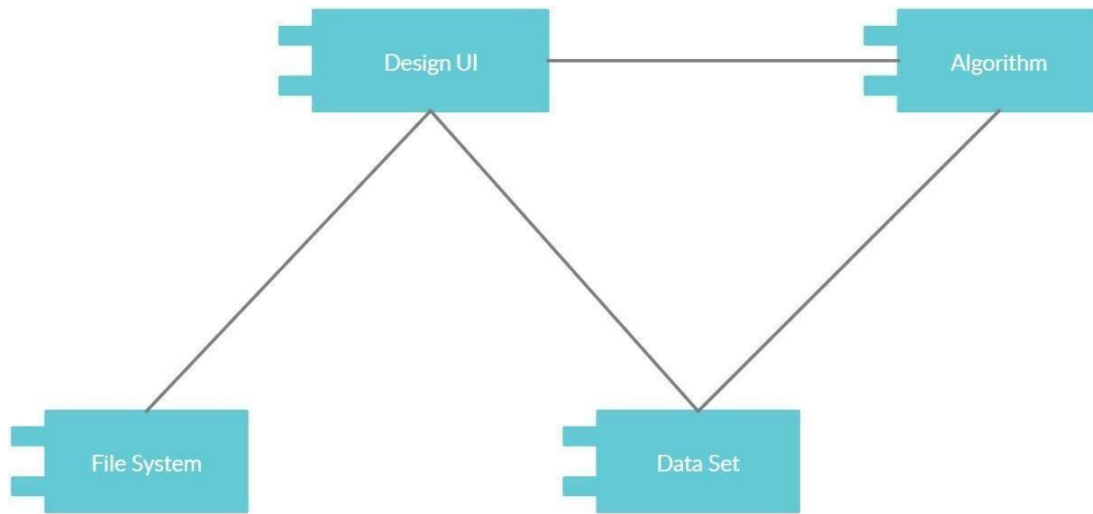


Fig 4.8 Component Diagram

4.9 STATE CHART DIAGRAM

A State chart diagram describes the behavior of a single object in response to a series of events in a system. Sometimes it's also known as a Harel state chart or a state machine diagram. This UML diagram models the dynamic flow of control from state to state of a particular object within a system. It is similar to activity diagram but here there are only few rules like how it starts and how it ends all are denoted with the help of the symbol given below, the system starts with the registration and then login comes, if the login is successful then it will go to the next step and if login is incorrect then comes back to same page stating incorrect details. After the successful login the user needs to enter the symptoms and then press the prediction button, at the same time the backend process will do their work and the correct result is predicted.

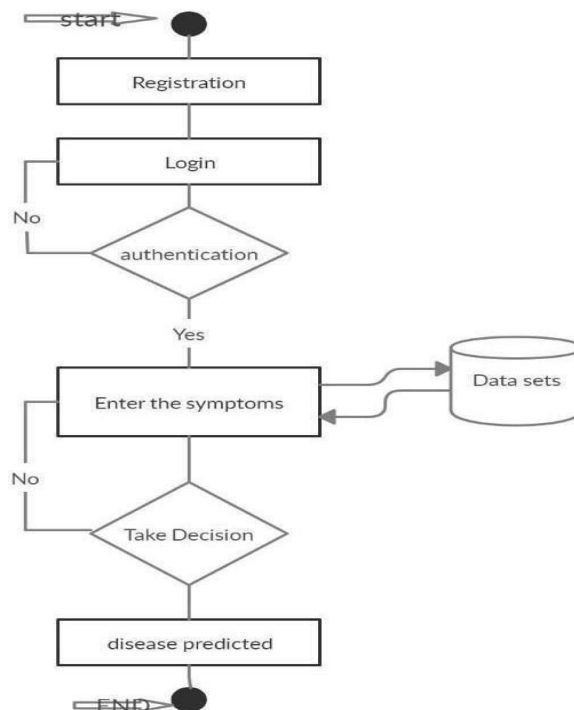


Fig 4.9 State Chart Diagram

4.10 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Here this diagram shows how all the models are connected to show the correct result starting from user, where he opens the system then using the system, he does registration and that registration data is saved into file system and the using those data user logs in to the system and then he provides all the necessary information in order to get the accurate result, then system evaluates the user entered information and finally gives the correct result.

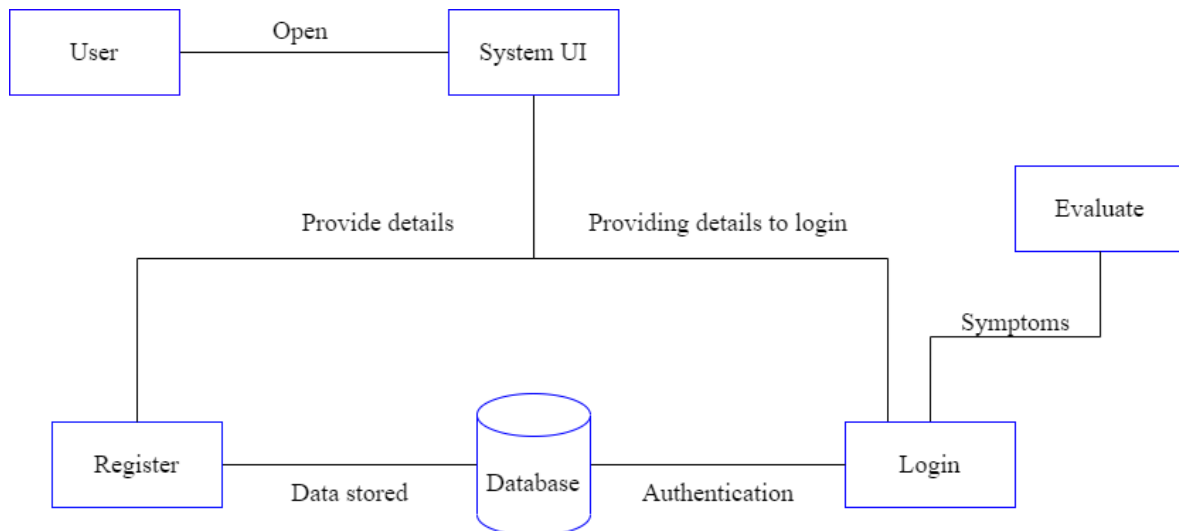


Fig 4.10 Collaboration Diagram

4.11 DEPLOYMENT DIAGRAM

A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. Here the deployment diagram shows the final stage of the project and it also shows how the model looks after doing all the processes and deploying in the machine. Starting from the system, how it processes the user entered information and then comparing that information with the help of datasets, then training and testing that data using the algorithms such as decision tree, naïve Bayes, random forest. Then finally processing all that data and information the system gives the desired result in the interface.

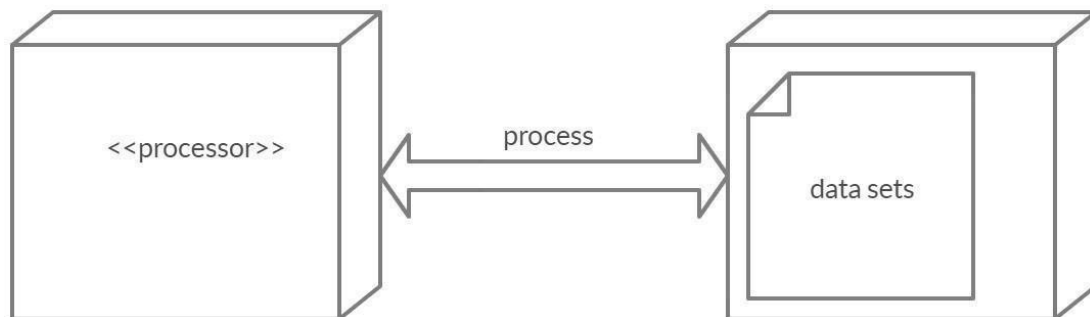


Fig 4.11 Deployment Diagram

4.12 INTERFACE AND FRAMEWORK DIAGRAM

- Below is the structure which we will use in our project Disease Prediction using Machine learning.
- The User Interface of this system consists of Python's library interface called tkinter.
- Then it goes into the framework model where all the actions and services are combined and then the result is processed.
- It also consists of a file system or SQL database where all the user related information is stored such as username, password, age, phone, email.
- Below is the structure of the User Interface along with necessary implementations.

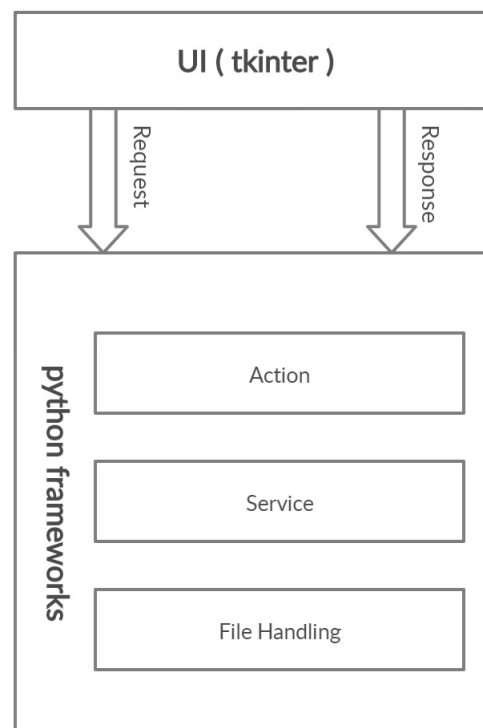


Fig 4.12 Interface and Framework Diagram

- After the User Interface it consists of the framework in which the system works accordingly using all the technologies, algorithms and various tools in which the project works accordingly.

- The framework consists of all the modules starting from the data preparation, data building and assessment stage.
- All these three factors are then going into the data collection phase, where the data is classified accordingly using the appropriate models and algorithms such as decision tree, naïve bayes, random forest.
- Then all those algorithms use the datasets and it forms the sets where all the previous data is stored, then using that data it compares with the new data and the result is generated.
- Then pre-processing work will happen to reduce and analyze the data that is present in the system.
- Then with the help of UI the data is transferred into the main screen.
- Then later all those data are analyzed and validated then the final result is generated.
- Finally, after the user enters the symptoms, all backend mechanisms work and the predicted result is displayed in the User Interface.

CHAPTER 5

IMPLEMENTATION

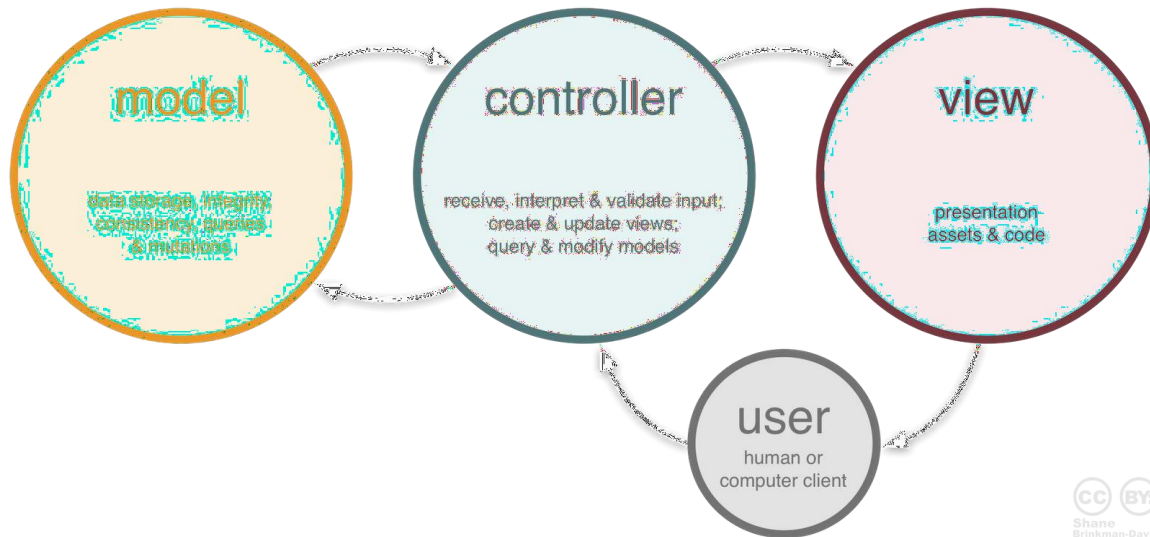
5.1 OVERVIEW

The Project Disease Prediction using Machine Learning is developed to overcome general disease in earlier stages as we all know in the competitive environment of economic development mankind has been involved so much that he/she is not concerned about health. According to research there are 40% of people who ignore general disease which leads to harmful disease later. The Project “Disease Prediction using Machine Learning” is implemented using python completely. Even the interface of this project is done using python’s library interface called Tkinter. Here first the user needs to register into the system in order to use the prediction, user needs to register with username, email-id, phone, age and password. All these values are stored into the file system respectively, then the user has the option to move forward or leave, then the user needs to login to the system using the username and password which he/she provided during the time of registration. If he/she enters incorrect username and correct password then the error message will prompt stating incorrect username and if he/she enters incorrect password and correct username then the error message will prompt stating incorrect password, so both username and password is necessary in order to login to the system. After logging in the user needs the name and needs to select the symptoms from the given drop-down menu, for more accurate results the user needs to enter all the given symptoms, then the system will provide the accurate result. This prediction is basically done with the help of 3 algorithms of machine learning such as Decision Tree, Random Forest and Naïve Bayes. When user enter all the symptoms then he needs to press the buttons of respective algorithm, for example there are 3 buttons for 3 algorithms, if user enters all symptoms and presses only Random forest’s button then the result will be provided only calculating using that algorithm, like this we have used 3 algorithms to provide clear picture of the results and user needs to be satisfied with his predicted result.

The project is designed user friendly and also secure to use ever user requires a authentication to enter into the system after which it provides the result based on the user input let me explain the complete implementation and working of project step wise below:

- Once user open the system to login user needs to register by clicking on register/signup button
- After which user needs to provide some basic details of signup and then the details of user are saved in system
- Then user needs to login to have a checkup of his/her health
- When user tries to login if he provides wrong user name the system will provide a prompt message stating that the user is not found
- And if user tries to enter the wrong password the system will prompt stating that password is incorrect hence the user needs to enter the correct user id and password to get in to the system
- After user enters the system user has to provide the symptoms which he/she is going through based on which we have several algorithms which predict the disease and also displays the percentage of accuracy
- The user needs to enter all the columns of symptoms to get the accurate result.
- Data collection and dataset preparation This will involve collection of medical information from various sources like hospitals, then pre-processing is applied on dataset which will remove all the unnecessary data and extract important features from data.
- Developing a probabilistic model and deep learning approach (RNN) for Disease Prediction in this step probabilistic model and deep learning approach based on RNN is to be developed it will run effectively on extensive databases of healthcare. And generating a decision tree also can deal with a huge number of information variables without variable deletion.
- Training and experimentation on datasets The Disease Prediction model will be trained on the dataset of diseases to do the prediction accurately and produce Confusion matrix. In this project Naïve Bayes Algorithms were used.

- Deployment and analysis on real life scenarios the trained and tested prediction model will be deployed in a real-life scenario made by the human experts & will be leveraged for further improvement in the methodology.



5.2 NAÏVE BAYES ALGORITHM

- It is used to predict the categorical class labels.
- It classifies the class data based on the training set and the values in a classifying attribute and uses it in classifying new data.
- It is a two-step process Model Construction and Model Usage.
- This Bayes theorem is named after Thomas Bayes and it is a statistical method for classification and supervised learning method.
- It can solve both categorical and continuous values attributes.
- Bayes theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes theorem is stated mathematically as the following equation:

$$P(A/B) = P(B|A) P(A)/P(B)$$

- Below is the pseudo code of the Naive Bayes Algorithm:

Input:

Training dataset T,

$F = (f_1, f_2, f_3, \dots, f_n)$ // value of the predictor variable
in testing dataset.

Output:

A class of testing dataset.

Step:

1. Read the training dataset T;
2. Calculate the mean and standard deviation of the predictor variables in each class;
3. Repeat

Calculate the probability of f_i using the gauss density equation in each class;

Until the probability of all predictor variables ($f_1, f_2, f_3, \dots, f_n$) has been calculated.

4. Calculate the likelihood for each class;
5. Get the greatest likelihood;

Fig 5.2.1 Naïve Bayes Algorithm Pseudo code

- The given dataset is divided into two parts namely feature matrix and response vector.
- Feature matrix contains all the vectors, which means rows of the dataset in which each vector consists of the values of dependent features. In the above dataset features are outlook, temperature, humidity and windy.
- Response vectors consist of values of class variables for each row of the feature matrix. In the above dataset the class variable name is play golf.
- The fundamental naïve based assumption is that each feature makes an independent and equal contribution to the outcome.

5.3 CODE SNIPPET

```
def NaiveBayes():
    from sklearn.naive_bayes import MultinomialNB
    gnb = MultinomialNB()
    gnb=gnb.fit(X,np.ravel(y))
    from sklearn.metrics import accuracy_score
    y_pred = gnb.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred, normalize=False))

psymptoms=[Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),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(disease[predicted] == disease[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, "No Disease")
```

CHAPTER 6

TESTING

TYPES OF TESTS

6.1 UNIT TESTING

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.

6.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually 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 satisfactory, 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.

6.3 VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications. It is important in identifying design problems, and solving them as early in the design cycle as possible, is the key to keeping projects on time and within budget. Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The old adage holds true: It costs a penny to make a change in engineering, a dime in production and a dollar after a product is in the field.

Verification is a Quality control process that is used to evaluate whether or not a product, service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process.

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

The testing process overview is as follows:

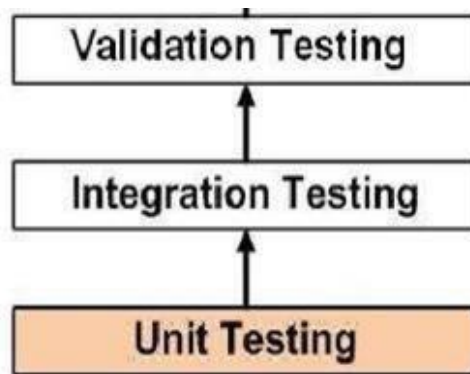


Fig 6.1 The Testing Process

6.4 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) or System Requirement Specification (SRS).

6.5 TESTING OF INITIALIZATION AND UI COMPONENTS

Serial Number of Test Case	TC 01
Module Under Test	User Registration
Description	A user enters their details for registering themselves to the System
Input	Details of Users such as username, email, phone, age, password.
Output	If the user's details are correct, the user is registered. If the user's details are incorrect, Displays an error message. If the user is already registered, Displays an error message.
Remarks	Test Successful.

Table 6.5.1 Test Case for User Registration

Serial Number of Test Case	TC 02
Module Under Test	User Login
Description	When the user tries to log in, details of user are verified in the system
Input	Username and Password.
Output	If the login details are correct, the user is logged in and the user page is displayed. If the login details are incorrect, Displays an error message.
Remarks	Test Successful.

Table 6.5.2 Test Case for User Login

Serial Number of Test Case	TC 03
Module Under Test	Prediction Result
Description	User needs to enter the name and symptoms to get the prediction result.
Input	Name and Symptoms
Output	If the user enters all 5 correct symptoms then the accuracy will be high. If a user enters only a few symptoms then accuracy will be low.
Remarks	Test Successful.

Table 6.5.3 Test Case for Prediction Result

LOGIN TESTING

SINo	Input		Expected Output	Actual Output	Status
	User Name	Password			
1	None	None	Enter User Name and Password	Enter User Name and Password	Pass
2	Rohit	None	Enter User Name and Password	Enter User Name and Password	Pass
3	None	Rohit123	Enter User Name and Password	Enter User Name and Password	Pass
4	Rohit	<u>Rohit125</u>	Invalid Username And Password	Invalid Username And Password	Pass
5	Rohit2	Rohit123	Invalid Username And Password	Invalid Username And Password	Pass
6	Rohit	Rohit123	Successfully Login	Successfully Login	Pass

SIGNUP TESTING

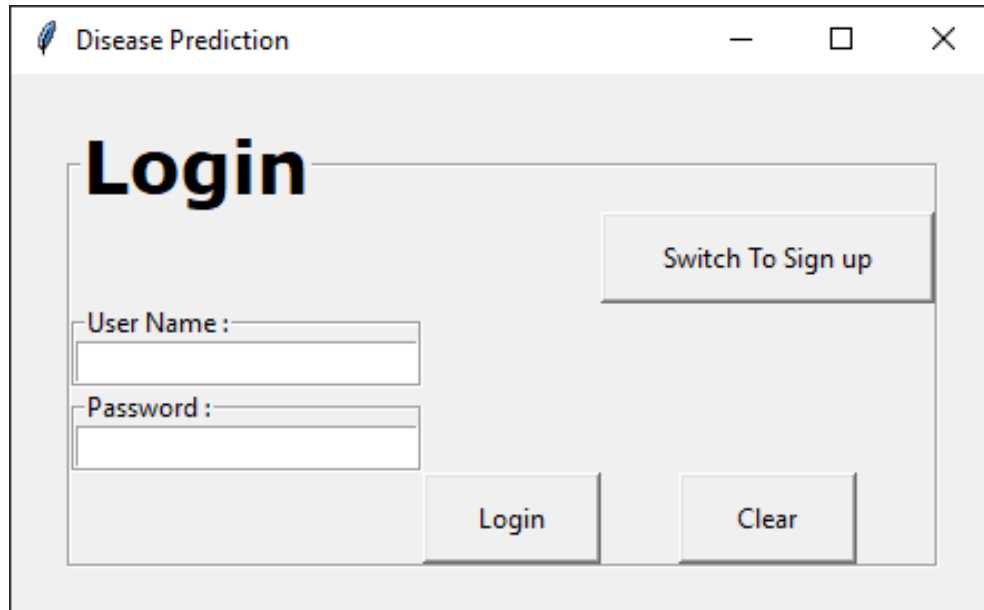
SINo	Expected Input	Expected Output	Actual Output	Status
1	Zero Fields Entered	All Fields Are Required	All Fields Are Required	Pass
2	Partial Fields Entered	All Fields Are Required	All Fields Are Required	Pass
3	Password Confirmation Field Entered wrong	Password And Confirm Password Should Be Same	Password And Confirm Password Should Be Same	Pass
4	Password Confirmation Field Entered correct	Registration Successful	Registration Successful	Pass

DISEASE PREDICTION TESTING

SlNo	Input					Expected Output	Actual Output	Status
	Symptom 1	Symptom 2	Symptom 3	Symptom 4	Symptom 5			
1	itching	skin_rash	nodal_skin_eruptions	dischromic_patches	None	Fungal infection	Fungal infection	PASS
2	vomiting	chest_pain	None	None	None	Gastroenteritis	Heartattack	FAIL
3	stomach_pain	acidity	ulcers_on_tongue	vomiting	chest_pain	GERD	GERD	PASS
4	itching	vomiting	yellowish_skin	nausea	loss_of_appetite	Chronic cholestasis	Chronic cholestasis	PASS
5	itching	skin_rash	stomach_pain	burning_micturition	spotting_urination	Drug Reaction	Drug Reaction	PASS
6	vomiting	indigestion	loss_of_appetite	abdominal_pain	passage_of_gases	Peptic ulcer disease	Peptic ulcer disease	PASS
7	muscle_wasting	patches_in_throat	high_fever	extra_marital_contacts	None	AIDS	AIDS	PASS
8	throat_irritation	chills	acidity	None	None	Dengue	Common Cold	FAIL
9	fatigue	weight_loss	restlessness	lethargy	obesity	Diabetes	Diabetes	PASS
10	fatigue	cough	high_fever	breathlessness	family_history	Bronchial Asthma	Bronchial Asthma	PASS
11	headache	chest_pain	dizziness	loss_of_balance	lack_of_concentration	Hypertension	Hypertension	PASS
12	throat_irritation	chills	acidity	back_pain	None	Dengue	Dengue	PASS
13	back_pain	weakness_in_limbs	neck_pain	dizziness	loss_of_balance	Cervical spondylosis	Cervical spondylosis	PASS
14	continuous_sneezing	shivering	chills	watering_from_eyes	None	Allergy	Allergy	PASS
15	breathlessness	None	None	None	None	Bronchial Asthma	Heartattack	FAIL
16	vomiting	sunken_eyes	dehydration	diarrhoea	None	Gastroenteritis	Gastroenteritis	PASS
17	acidity	indigestion	headache	blurred_and_distorted_vision	excessive_hunger	Migraine	Migraine	PASS
18	vomiting	high_fever	None	None	None	Paralysis	Jaundice	FAIL

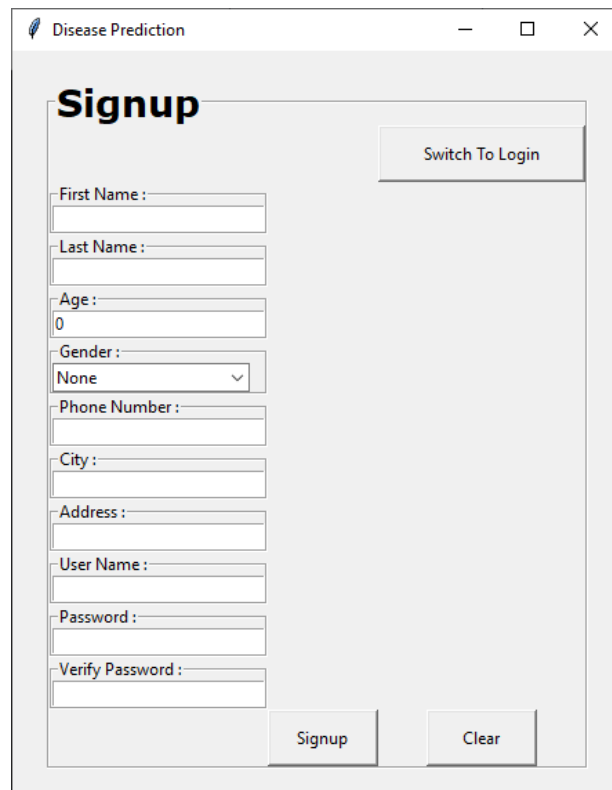
CHAPTER 7

SNAPSHOTS



The image shows a web application window titled "Disease Prediction". The main heading is "Login". On the right side, there is a button labeled "Switch To Sign up". Below the heading, there are two input fields: "User Name :" and "Password :". At the bottom right, there are two buttons: "Login" and "Clear".

Fig 7.1 Login Page



The image shows a web application window titled "Disease Prediction". The main heading is "Signup". On the right side, there is a button labeled "Switch To Login". Below the heading, there are several input fields: "First Name :", "Last Name :", "Age :", "Gender :" (with a dropdown menu showing "None"), "Phone Number :", "City :", "Address :", "User Name :", "Password :", and "Verify Password :". At the bottom right, there are two buttons: "Signup" and "Clear".

Fig 7.2 Signup Page

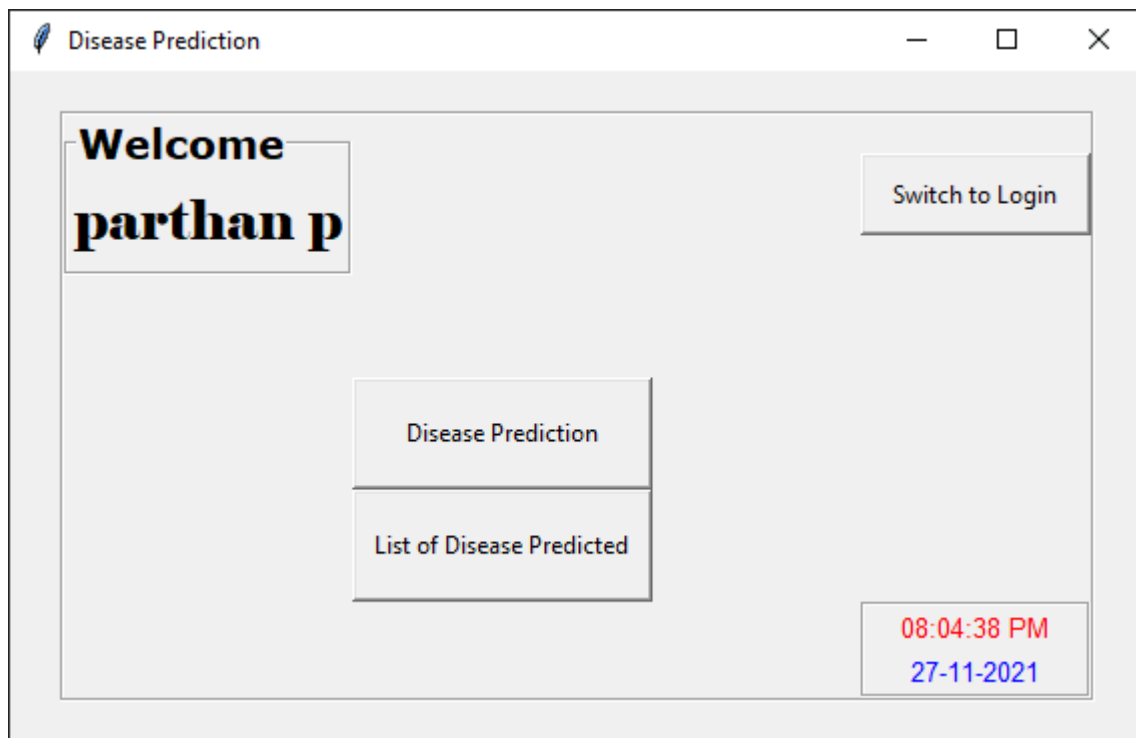


Fig 7.3 Menu Page

	id	first_name	last_name	age	gender	city	address	username	password	phone_number
►	41	Mohammed	Taheer	22	Male	chintamani	chintamani	taheer	taheer@123	9353474769
	43	shaik	safi	20	Male	bengaluru	banashankari	safi	safi@123	8965471235
	44	parthan	p	21	Male	bengaluru	jayadeva	parthan	parthan@123	6547892658
	45	dsa	dsdadsda	19	Female	adsadasd	asdadasd	sad	sad	9008154698
	46	sai	kireeti	25	Male	bengaluru	bommanahalli	kireeti	kireeti@123	5689741235
	47	mno	pqr	20	Female	brgjgdjjj	fdsgfdbfd	mno	mno@123	9353647895

Fig 7.4 User Details Page

	id_2	full_name	age	gender	id	patient_date	patient_time
►	5000	kireeti	20	Male	43	2021-11-07	07:19:20
	5002	parthana	21	Female	43	2021-11-07	10:00:20
	5003	xyz	15	Male	41	2021-11-07	10:30:20
	5005	kireeti	25	Male	41	2021-11-07	10:45:20
	5008	pqr	21	Female	43	2021-11-07	22:13:31
	5010	lmn	50	Female	43	2021-11-09	00:18:28
	5011	pqr	23	Male	41	2021-11-09	15:44:15
	5012	lop	30	Male	44	2021-11-20	20:23:49
	5013	efg	50	Female	44	2021-11-21	09:51:07
	5014	rohan	30	Male	46	2021-11-26	18:54:17

Fig 7.5 Patient Details Page

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	inflammatory_nails	blister	red_sore_around_nose	yellow_crust_ooze
count	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000
mean	0.166667	0.190476	0.023810	0.047619	0.023810	0.023810	0.023810	0.047619	0.023810
std	0.377195	0.397437	0.154303	0.215540	0.154303	0.154303	0.154303	0.215540	0.154303
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig 7.6 Testing Data

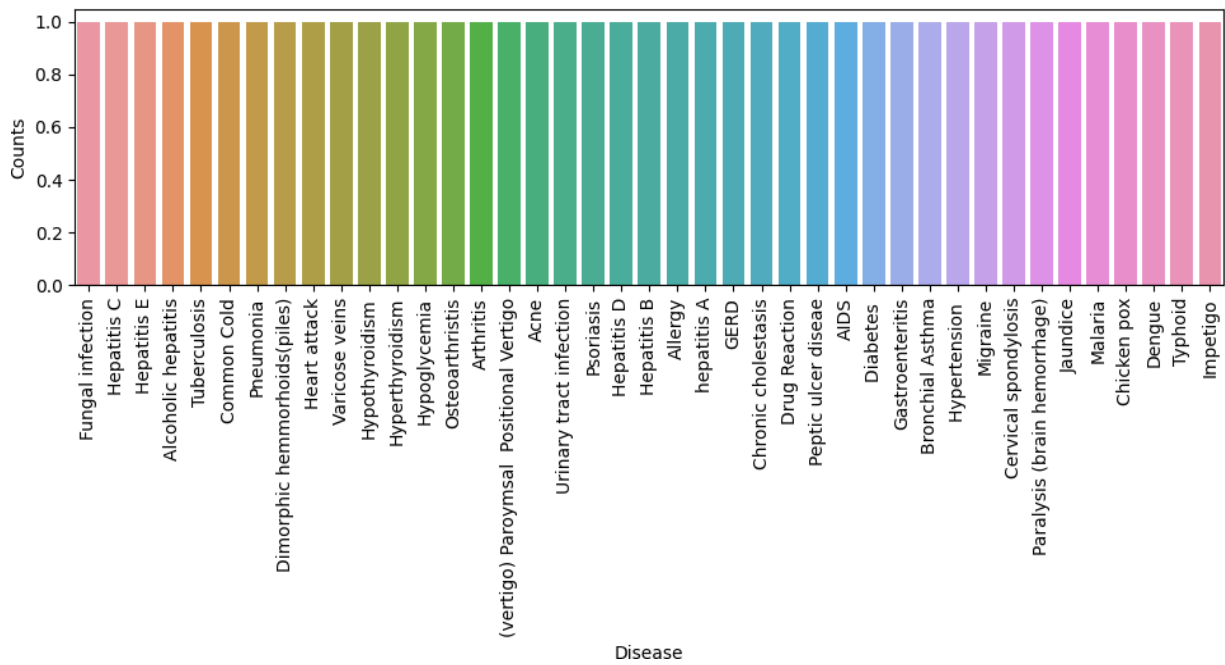
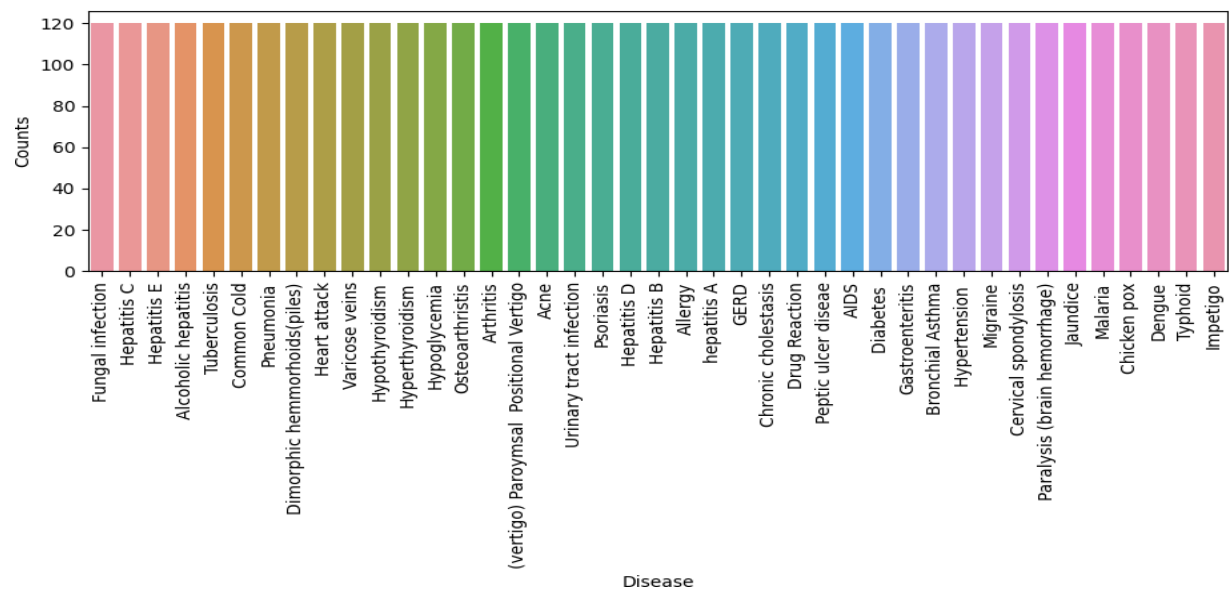


Fig 7.7 Bar Plot chart of Testing Data

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	inflammatory_nails	blister	red_sore_around_nose	yellow_crust_ooze
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000
mean	0.137805	0.159756	0.021951	0.045122	0.021951	0.023171	0.023171	0.023171	0.023171
std	0.344730	0.366417	0.146539	0.207593	0.146539	0.150461	0.150461	0.150461	0.150461
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig 7.8 Training Data



**Fig 7.9 Bar Plot chart
Training Data**

Disease Prediction

Patient Details

Name: kireeti
Age: 20
Gender: Male
Already Entered Patients: kireeti
Switch to Menu

Enter Symptoms

Symptom 1
Symptom 2
Symptom 3
Symptom 4
Symptom 5

anxiety
cough
headache
abdominal_pain
None

Predict Disease
Clear

Disease Predicted

Typhoid

Fig 7.10 Prediction Page

id_3	id_2	symptom_1	symptom_2	symptom_3	symptom_4	symptom_5	disease
10020	5008	mood_swings	unsteadiness	dark_urine	yellow_crust_ooze	None	Impetigo
10021	5005	joint_pain	chills	yellowing_of_eyes	diarrhoea	None	hepatitis A
10022	5014	itching	skin_rash	nodal_skin_eruptions	dischromic_patches	None	Fungal infection
10023	5014	continuous_sneezing	shivering	chills	watering_from_eyes	None	Allergy
10024	5014	stomach_pain	acidity	ulcers_on_tongue	vomiting	None	GERD
10025	5014	itching	vomiting	yellowish_skin	nausea	None	Chronic cholestasis
10026	5014	itching	skin_rash	stomach_pain	burning_micturition	spotting_urination	Drug Reaction
10027	5014	vomiting	indigestion	loss_of_appetite	abdominal_pain	passage_of_gases	Peptic ulcer diseae
10028	5012	muscle_wasting	patches_in_throat	high_fever	extra_marital_conta...	None	AIDS
10029	5012	vomiting	sunken_eyes	dehydration	diarrhoea	None	Gastroenteritis
10030	5012	fatigue	weight_loss	restlessness	lethargy	None	Diabetes
10031	5012	fatigue	cough	high_fever	breathlessness	family_history	Bronchial Asthma
10032	5012	anxiety	blackheads	abnormal_menstruation	back_pain	anxiety	Acne
10033	5011	back_pain	bruising	dischromic_patches	None	None	Fungal infection

Fig 7.11 Symptoms and Disease table

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

So, Finally I conclude by saying that, this project Disease prediction using machine learning is very much useful in everyone's day to day life and it is mainly more important for the healthcare sector, because they are the one that daily uses these systems to predict the diseases of the patients based on their general information and their symptoms that they are been through. Now a day's health industry plays major role in curing the diseases of the patients so this is also some kind of help for the health industry to tell the user and also it is useful for the user in case he/she doesn't want to go to the hospital or any other clinics, so just by entering the symptoms and all other useful information the user can get to know the disease he/she is suffering from and the health industry can also get benefit from this system by just asking the symptoms from the user and entering in the system and in just few seconds they can tell the exact and up to some extent the accurate diseases. If the health industry adopts this project, then the work of the doctors can be reduced and they can easily predict the disease of the patient. The Disease prediction is to provide predictions for the various and generally occurring diseases that when unchecked and sometimes ignored can turn into fatal disease and cause a lot of problems to the patient and as well as their family members.

8.2 FUTURE ENHANCEMENT

Facility for modifying user detail, more interactive user interface, facilities for backup creation, can be done as a web page, can be done as a mobile application, more details and latest diseases.

REFERENCES

1. Kaveeshwar, S.A., and Cornwall, J., 2014, "The current state of disease mellitus in India". AMJ, 7(1), pp. 45-48.
2. Anjan Nikhil Repaka, Sai Deepak Ravikanti, Ramya G Franklin, "Design And Implementation Heart Disease Prediction Using Naives Bayesian", International Conference on Trends in Electronics and Information(ICOEI 2019).
3. Fahd Saleh Alotaibi," Implementation of Machine Learning Model to Predict Heart Failure Disease", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, 2019.
4. P. S. Kohli and S. Arora, "Application of machine learning in disease prediction," in 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018.
5. S. Jadhav, R. Kasar, N. Lade, M. Patil, and S. Kolte, "Disease Prediction by Machine Learning from Healthcare Communities," International Journal of Scientific Research in Science and Technology, 2019.
6. M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities", IEEE Access, vol. 5, 2017.
7. Ajinkya Kunjir, Harshal Sawant, Nuzhat F.Shaikh, "Data Mining and Visualization for prediction of Multiple Diseases in Healthcare", in IEEE big data analytics and computational intelligence, 2017.
8. B. Qian, X. Wang, N. Cao, H. Li, and Y.-G. Jiang, "A relative similarity based method for interactive patient risk prediction", Data Mining Knowl.Discovery, vol. 29, 2015.
9. IM. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, " Wearable 2.0: Enable human-cloud integration in next generation healthcare system", IEEE Commun. , vol. 55, Jan. 2017.

