**MINI PROJECT**

**FINAL REPORT**

**(2019-2020)**

**AI ChatBot for Health Care**

DEPARTMENT OF COMPUTER ENGINEERING AND APPLICATIONS



**Submitted To: Submitted By:**

Mr.Amir Khan AyushiSrivastava (171500080)

(***Technical Trainer***) KirtiVerma(171500162)

Nitya Nagpal (171500210)

**Declaration**

We here by declare that the work which is being presented in the B.Tech. Project **“AI Healthcare ChatbotSystem”**,in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** and submitted to the Department of Computer Science Engineering and Applications of GLA University, Mathura ,is an authentic record of our own work carried under the supervision of **Mr. AMIR KHAN , Assistant Professor of Computer Engineering Department.**

The contents of this project report, in full or in parts, have not been submitted to any other institute or university for the award of any degree.

|  |  |  |
| --- | --- | --- |
| AyushiSrivastava  (171500080) |  | Nitya Nagpal (171500210) |
| KirtiVerma  (171500162) |  |

**Certificate**

This is to certify that the above statements made by the candidates are correct to the best of my/our knowledge and belief.

#### ProjectSupervisor Date:

(Mr. Amir Khan)

#### Project Incharge Program Coordinator Head of Department

**(Mr. Sharad Gupta) (Mr. ShashiShekhar) (Prof. Anand Singh Jalal)**

## Acknowledgement

The satisfaction which accompanies the successful completion of the project, is incomplete without the mention of a few names. We take this opportunity to acknowledge the efforts of the many individuals who helped us to make this project possible.

Firstly we would like to express our heartfelt appreciation and gratitude to our project guide **Mr. AMIR KHAN, Assistant Professor, Computer Engineering Department.** His vision and execution aimed at creating a structure, definition and realism around the project and fostered the ideal environment for us to learn and do. This project is a result of his teaching, encouragement and inputs in the numerous meetings he had with us, despite his busy schedule.

We would also like to extend our immense gratitude to respected Head of Department **Prof. Anand Singh Jalal** who allowed us to choose the topic for our dissertation.

The experience was novel one and we would like to thank all the people ,who have lent their valuable time for the completion of the report. Without their consideration it would have been difficult to complete the report.

|  |  |
| --- | --- |
| AyushiSrivastava | (171500080) |
| KirtiVerma | (171500162) |
| Nitya Nagpal | (171500210) |

## Abstract

## Today in the fast running world, health of everyone is a leading topic to worry about because everyone is busy with their work in such an extent that they have no time to fix the problems related to their health. Everyone, today, tried to avoid their health issue until they reach at their worst condition only because they do not have time to go to the doctor personally. Normally the procedures (like regular check-ups, blood tests, ultrasounds etc) for even small health issues are so much hectic for a working person because it is so much time consuming.

## Heathcare chat bot plays in important role in the life of a working person.Normally Users are not aware of all the treatment or symptoms regarding the particular disease and going to the doctor for every problem is difficult task thensuch a problem can be solved by using medical ChatBot by giving proper guidance regarding healthy living.The chatbots thus try to handle several needs, such as personalized medical follow-up, communication and transmission of test results, dissemination of information, or even advice to patients or preliminary diagnosis which is preferable by the consumer.

## Index

## ● CHAPTER 1 - Introduction to HealthCare Chatbot

## ● CHAPTER 2 - Software Requirement Analysis

## ● CHAPTER 3 - System Design

## ● CHAPTER 4 – Software Testing

## ● CHAPTER 5 - Implementation

## •CHAPTER 6 - Tkinter Features and Functions

## •CHAPTER 7 - Data Sets

## •CHAPTER 8 – Code Section

## •Bibliography

## INTRODUCTION

**What is Intelligence?**

The word "[*Intelligence*](https://en.wiktionary.org/wiki/intelligence#English)" derives from the Latin [nouns](https://en.wikipedia.org/wiki/Noun) [*intelligentia*](https://en.wiktionary.org/wiki/intelligentia) or [*intellēctus*](https://en.wiktionary.org/wiki/intellectus), which in turn stem from the verb [*intelligere*](https://en.wiktionary.org/wiki/intelligere), to comprehend or perceive.Individuals differ from one another in their ability to understand complex ideas, to adapt effectively to the environment, to learn from experience, to engage in various forms of reasoning, to overcome obstacles by taking thought. Although these individual differences can be substantial, they are never entirely consistent: a given person's intellectual performance will vary on different occasions, in different domains, as judged by different criteria. Concepts of "intelligence" are attempts to clarify and organize this complex set of phenomena. Although considerable clarity has been achieved in some areas, no such conceptualization has yet answered all the important questions, and none commands universal assent.

**Intelligence** has been defined in many ways: the capacity for [logic](https://en.wikipedia.org/wiki/Logic), [understanding](https://en.wikipedia.org/wiki/Understanding), [self-awareness](https://en.wikipedia.org/wiki/Self-awareness), [learning](https://en.wikipedia.org/wiki/Learning), [emotional knowledge](https://en.wikipedia.org/wiki/Emotional_knowledge), [reasoning](https://en.wikipedia.org/wiki/Reason), [planning](https://en.wikipedia.org/wiki/Planning), [creativity](https://en.wikipedia.org/wiki/Creativity), [critical thinking](https://en.wikipedia.org/wiki/Critical_thinking), and [problem-solving](https://en.wikipedia.org/wiki/Problem-solving). More generally, it can be described as the ability to perceive or infer [information](https://en.wikipedia.org/wiki/Information), and to retain it as [knowledge](https://en.wikipedia.org/wiki/Knowledge) to be applied towards adaptive behaviors within an environment or context.

*Types of Intelligence-*

* Human Intelligence
* Artificial Intelligence
* Plant intelligence

**Human Intelligence-**

Human intelligence is the intellectual power of humans, which is marked by complex [cognitive](https://en.wikipedia.org/wiki/Cognition) feats and high levels of [motivation](https://en.wikipedia.org/wiki/Motivation) and [self-awareness](https://en.wikipedia.org/wiki/Self-awareness). Intelligence enables humans to remember descriptions of things and use those descriptions in future behaviors. It is a cognitive process. It gives humans the [cognitive](https://en.wikipedia.org/wiki/Cognition) abilities to [learn](https://en.wikipedia.org/wiki/Learning), [form concepts](https://en.wikipedia.org/wiki/Concept_learning), [understand](https://en.wikipedia.org/wiki/Understanding), and [reason](https://en.wikipedia.org/wiki/Reason), including the capacities to [recognize patterns](https://en.wikipedia.org/wiki/Pattern_recognition), comprehend ideas, [plan](https://en.wikipedia.org/wiki/Planning), [solve problems](https://en.wikipedia.org/wiki/Problem_solving), and use [language](https://en.wikipedia.org/wiki/Language) to [communicate](https://en.wikipedia.org/wiki/Human_communication). Intelligence enables humans to [experience](https://en.wikipedia.org/wiki/Experience) and [think](https://en.wikipedia.org/wiki/Thought).

**Artificial Intelligence-**

Artificial intelligence (or AI) is both the intelligence of machines and the branch of [computer science](https://en.wikipedia.org/wiki/Computer_science) which aims to create it, through "the study and design of [intelligent agents](https://en.wikipedia.org/wiki/Intelligent_agents)" or "rational agents", where an [intelligent agent](https://en.wikipedia.org/wiki/Intelligent_agent) is a system that perceives its environment and takes actions which maximize its chances of success. [Kaplan](https://en.wikipedia.org/wiki/Andreas_Kaplan) and Haenlein define artificial intelligence as “a system’s ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation”.

[Achievements in artificial intelligence](https://en.wikipedia.org/wiki/Progress_in_artificial_intelligence) include constrained and well-defined problems such as games, [crossword](https://en.wikipedia.org/wiki/Crossword)-solving and [optical character recognition](https://en.wikipedia.org/wiki/Optical_character_recognition) and a few more general problems such as [autonomous cars](https://en.wikipedia.org/wiki/Autonomous_cars). General intelligence or [strong AI](https://en.wikipedia.org/wiki/Artificial_general_intelligence) has not yet been achieved and is a long-term goal of AI research.

Among the traits that researchers hope machines will exhibit are [reasoning](https://en.wikipedia.org/wiki/Artificial_intelligence#Reasoning,_problem_solving), [knowledge](https://en.wikipedia.org/wiki/Artificial_intelligence#Knowledge_representation), [planning](https://en.wikipedia.org/wiki/Artificial_intelligence#Planning), [learning](https://en.wikipedia.org/wiki/Artificial_intelligence#Learning), [communication](https://en.wikipedia.org/wiki/Artificial_intelligence#Natural_language_processing), [perception](https://en.wikipedia.org/wiki/Artificial_intelligence#Perception), and the ability to [move](https://en.wikipedia.org/wiki/Artificial_intelligence#Motion_and_manipulation) and to manipulate objects. In the field of artificial intelligence there is no consensus on how closely the brain should be [simulated](https://en.wikipedia.org/wiki/Computer_simulation).

**Plant Intelligence**

It has been argued that plants should also be classified as intelligent based on their ability to sense and model external and internal environments and adjust their [morphology](https://en.wikipedia.org/wiki/Morphology_(biology)), [physiology](https://en.wikipedia.org/wiki/Plant_physiology) and [phenotype](https://en.wikipedia.org/wiki/Phenotypic_plasticity) accordingly to ensure self-preservation and reproduction.

A counter argument is that intelligence is commonly understood to involve the creation and use of persistent memories as opposed to computation that does not involve learning. If this is accepted as definitive of intelligence, then it includes the artificial intelligence of robots capable of "machine learning", but excludes those purely autonomic sense-reaction responses that can be observed in many plants. Plants are not limited to automated sensory-motor responses, however, they are capable of discriminating positive and negative experiences and of "learning" (registering memories) from their past experiences. They are also capable of communication, accurately computing their circumstances, using sophisticated [cost–benefit analysis](https://en.wikipedia.org/wiki/Cost%E2%80%93benefit_analysis) and taking tightly controlled actions to mitigate and control the diverse environmental stressors.

**What is bot?**

Bots are program that run continuously formulate decisions, act upon those decisions without human intervention, and are able adapt to the context they operate in.

The most advanced bots are powered by artificial intelligence, helping it to understand complex requests, personalize responses and improve interactions over time.

**What is chat bots?**

A **chatbot** is a [software](https://en.wikipedia.org/wiki/Software_agent) application used to conduct an on-line chat [conversation](https://en.wikipedia.org/wiki/Conversation) via text or text-to-speech, in lieu of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse or pass the industry standard [Turing test](https://en.wikipedia.org/wiki/Turing_test). The term "ChatterBot" was originally coined by [Michael Mauldin](https://en.wikipedia.org/wiki/Michael_Loren_Mauldin) (creator of the first [Verbot](https://en.wikipedia.org/wiki/Verbot)) in 1994 to describe these conversational programs.

Chatbots are typically used in [dialog systems](https://en.wikipedia.org/wiki/Dialog_system) for various purposes including customer service, request routing, or for information gathering.

While some chatbot applications use extensive word-classification processes, [Natural Language](https://en.wikipedia.org/wiki/Natural_language_processing) processors, and sophisticated [AI](https://en.wikipedia.org/wiki/Artificial_intelligence), others simply scan for general keywords and generate responses using common phrases obtained from an associated library or [database](https://en.wikipedia.org/wiki/Database).

Today, most chatbots are accessed on-line via website pop-ups, or through [virtual assistants](https://en.wikipedia.org/wiki/Virtual_assistant_(artificial_intelligence)) such as [Google Assistant](https://en.wikipedia.org/wiki/Google_Assistant), [Amazon Alexa](https://en.wikipedia.org/wiki/Amazon_Alexa), or [messaging apps](https://en.wikipedia.org/wiki/Messaging_apps) such as [Facebook Messenger](https://en.wikipedia.org/wiki/Facebook_Messenger) or [WeChat](https://en.wikipedia.org/wiki/WeChat). Chatbots are typically classified into usage categories that include: [commerce](https://en.wikipedia.org/wiki/Conversational_commerce) ([e-commerce](https://en.wikipedia.org/wiki/E-commerce) via chat), [education](https://en.wikipedia.org/wiki/Education), [entertainment](https://en.wikipedia.org/wiki/Entertainment), [finance](https://en.wikipedia.org/wiki/Finance), [health](https://en.wikipedia.org/wiki/Health), [news](https://en.wikipedia.org/wiki/News),

and [productivity](https://en.wikipedia.org/wiki/Productivity).

## Types of Chat Bot's

* Flow oriented chatbox
* Artificially oriented chatbot
* Human supported bots

**Artificial Intelligence HealthCare Chatbot System**

Through chatbots one can communicate with text or voice interface and get reply through artificial intelligence. Typically, a chat bot will communicate with a real person. Chat bots are used in applications such as ecommerce customer service, call centres and Internet gaming. Chatbots are programs built to automatically engage with received messages. Chatbots can be programmed to respond the same way each time, to respond differently to messages containing certain keywords and even to use machine learning to adapt their responses to fit the situation. A developing number of hospitals, nursing homes, and even private centres, presently utilize online Chatbots for human services on their sites. These bots connect with potential patients visiting the site, helping them discover specialists, booking their appointments, and getting them access to the correct treatment. In any case, the utilization of artificial intelligence in an industry where individuals’ lives could be in question, still starts misgivings in individuals. It brings up issues about whether the task mentioned above ought to be assigned to human staff. This healthcare chatbot system will help hospitals to provide healthcare support online 24 x 7, it answers deep as well as general questions. It also helps to generate leads and automatically delivers the information of leads to sales.

## Problem statement and objective

## Problem statement-

The existing system has lots of flaw, because peoples have to go to the doctor personally instead getting a solution online. And the peoples sometimes think about the time which will worsen the condition of health issues. Now the present system will save time as well as long processes.

* In this system people gives their symptoms of disease as input and according to that symptoms chatbot will tell the possible disease and also their medications .
* Chatbots also provide the name ,contact, and location of doctors for the treatment of particular disease

## Proposed System-

Though this new online AI healthcare chatbot system helps in getting the medications of a disease or idea about having a possible disease, it also helps the consumers to connect them to the medical practitioners directly.

This project used by two types of user:-

* Consumer
* Admin

Users can search for the diseases related to their symptomsjust by giving their symptoms as input in this healthcare chatbot system.

## Overview and motivations

## Overview of the system-

In this project, we will showcase how the user can interact with a chatbot to get a response to their queries. Chatbots are computer programs with the designed purpose to have conversations with human users over the world wide web and internet. Many tech programs use conversational agents in their services to help streamline communication with their target audience. Where human agents would have limits on how many customers they can service, chatbots can handle many more customers at once.

This is a basic project that is developed in Python. In this project, we use some Python Libraries.

A person just has to put their query to the chatbot which is used for chatting. The system will give appropriate answers to the user. If the answer is found invalid, then there is a system to declare the answer as invalid.

## Motivation for the system-

* Time consuming procedures followed the particular disease .
* Lots of time is wasted in direct meeting with a medical practitioner for even a small problem like headache, nausea etc .

## Objective -

The main objective of this system is to make a computer program designed to simulate an intelligent conversation with one or morehuman users.

The purpose of this project to provide the admin has to collect the patient’s medical history of records and filter it appropriately by applying data preprocessing techniques. Admin’s functionalities are to Collecting the appropriate medical records of the patients, handle missing values, handling categorical values, Creating sparse matrix representation, Feeding data to the autonomous pipeline for predictions, selecting and training an appropriate machine learning

algorithm.

The visitor can perform the basic task of the visitor is to access the chatbot from the front end and reply to its queries with a binary response (Yes/No). The visitor will be shown a confidence interval related to a certain prognosis which needs to be further investigated and experimented with for better results. The first step is to start their procedure, then one by one all the symptoms come in clients’ screens. They will have to reply with yes or no answer.

Once a problem is found then they will have to click yes, then the patient can see their problem on screen. The Best Part is that it will provide the doctor's information like the Doctor’s name and his/her website link. So that one can easily find their doctor with don’t face any type of problem, and start their treatment. This will prepare with the help of chatbot so that one can even check their problem at any time. You have to just reply with the clicking of button Yes or No.

**SOFTWARE REQUIREMENTANALYSIS**

System Analysis is a detailed study of the various operations performed by a system and their relationship within and outside the system .It is a systematic technique that defines goals and objectives the goal of the development is to deliver the system in the line with the user’s requirements, and analysis is this process.

System study has been conducted with the following objectives in mind: -

* + - Identify the client’sneed.
    - Evaluate the system concept forfeasibility.
    - Perform economical and technicalanalysis.
    - Allocate functional to hardware, software, people, database and other system elements
    - Establish cost and scheduleconstraints.
    - Both hardware and software expertise is required to successfully attain the objectives.

It may be divided into 5 areas of effort.

* Problem recognition
* Evaluation and synthesis
* Modeling
* Specification
* Review

## RequirementAnalysis

Information gathering is usually the first phase of the software development project.

The purpose of this phase is to identify and document the exact requirements for the system. The user’s request identifies the need for a new information system and on investigation re-defined the new problem to be based on MIS, which supports management. The objective is to determine whether the request is valid and feasible before a recommendation is made to build a new or existing manual system continue

The major steps are –

* + - Defining the user requirements.
    - Studying the present system to verify the problem.
    - Defining the performance expected by the candidate to user requirement

## Software requirement specification

Software Requirements Specification plays an important role in creating quality software solutions. The specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation.

Requirements may be specified in a variety of ways. However, there are some guidelines worth following: -

* Representation format and content should be relevant to the problem.
* Information contained within the specification should benested.
* Diagrams and other notational forms should be restricted in number and consistent inuse.
* Representations should be revisable.

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, and indication of performance requirements and design constraints, appropriate validation. Criteria and other data pertinent torequirements.

## HardwareRequirements

* i3 Processor Based Computer or higher
* Memory: 1 GB
* Hard Drive: 50 GB
* Monitor
* Internet Connection

## SoftwareRequirements

* Windows 7 or higher
* Google Chrome Browser

## Tools and Technology

#### Tools:

* + - * Windows 7 & and higher
      * PyCharm

## Technology:

* **Python:**

**Python** is an [open source](https://simple.wikipedia.org/wiki/Open_source) [programming language](https://simple.wikipedia.org/wiki/Programming_language) that was made to be easy-to-read and powerful. A [Dutch](https://simple.wikipedia.org/wiki/Netherlands) programmer named [Guido van Rossum](https://simple.wikipedia.org/wiki/Guido_van_Rossum) made Python in 1991. He named it after the television show [Monty Python's Flying Circus](https://simple.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus). Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be [compiled](https://simple.wikipedia.org/wiki/Compiled_language) to run. A program called an [interpreter](https://simple.wikipedia.org/wiki/Interpreter_(computing)) runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like [C](https://simple.wikipedia.org/wiki/C_(programming_language)), because it is not running [machine code](https://simple.wikipedia.org/wiki/Machine_code) directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

* **Machine learning:**

Machine Learning is said as a subset of **artificial intelligence.**Machine learning is a growing technology which enables computers to learn automatically from past data. With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

**A machine has the ability to learn if it can improve its performance by gaining more data.**Machine learning uses various algorithms for **building mathematical models and making predictions using historical data or information**. Currently, it is being used for various tasks such as **image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system**, and many more.

* **Import libraries**
* **label Encoder**

we will convert the country variables into categorical data by using **LabelEncoder()** class from **preprocessing** library.

## train\_test\_split

train\_test\_split is a function in **Sklearn model selection** for splitting data arrays into **two subsets**: for training data and for testing data. With this function, you don't need to divide the dataset manually.

By default, Sklearn**train\_test\_split** will make random partitions for the two subsets. However, you can also specify a random state for the operation.

“Splitting the dataset into training set and test set”

fromsklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)

* **Decision Tree Classifier**

Decision Trees can be used as classifier or regression models.A tree structure is constructed that breaks the dataset down into smaller subsets eventually resulting in a prediction.

**Decision Tree Algorithm**

A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node.

It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.



Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and number of attributes in the given data. The decision tree is a distribution-free or non-parametric method, which does not depend upon probability distribution assumptions. Decision trees can handle high dimensional data with good accuracy.

* **Tkinter**

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 is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

**read\_csv() function:**

Now to import the dataset, we will use read\_csv() function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL.

* **Scikit-learn**

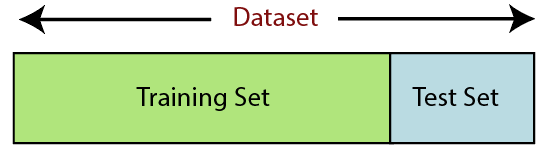
To handle missing values, we will use **Scikit-learn** library in our code, which contains various libraries for building machine learning models

* **Data Processing**

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:



**Training Set:** A subset of dataset to train the machine learning model, and we already know the output.

**Test set:** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output

## Feasibility study

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminateinawrittenandoralfeasibilityreport.Ihavetakenafixedtimeinfeasibility study with my co-developer. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to precede the project.

It helped in taking decisions such as which software to use, hardware combinations,etc.

#### Technical feasibility:

This is concerned with specifying equipment of software and hardware that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include:

* The facility to produce output in a given time.
* Response time under certain condition.
* Ability to produce a certain volume of transaction at a particular speed.
* In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the

complete picture about the system requirements. What speeds of input and output should be achieved at particular quality of printing.

#### operational feasibility:

It is mainly related to human organizational and political aspects. The points to be considered are:

* + - * What changes will be brought with the system?
      * What organization structures are distributed structures are distributed.
      * What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

At present stage all the work is done manually. So, throughput and response time is too much. Major problem is lack of security check that should have been applied.

Finding out the detail regarding user’s request was very difficult, because data store was in different registers and different places. In case of any problem, no one can solve the problem until the person responsible is not present.

Current communication is entirely on telephonic conversation or personal meetings. Post computerization staff can interact using internet.

Now, we will explain the last point of operational feasibility i.e. handling and keeping of software , at every point of designing I will take care that menu options arenottoocomplexandcanbeeasilylearnedandrequiredleastamountoftechnical skills as operators are going to be from non-computers background.

#### Economic feasibility:

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis: the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with cost. If benefits outweighs cost, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

At present Company has ten systems with following configuration:

* + - * Ram 4 GB or above for fast execution and reliability
      * MOTHER Board x64 based PC
      * Color Monitor 14” and17”
      * Hard Disk 100GB
      * Hence the economic feasibility is very good.

## Software Design

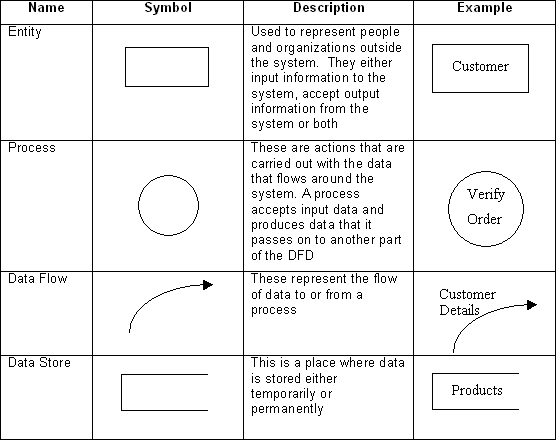
Software design is the most creative phase of system development.The question in system design is: How the problem is to be solved?.A software design document (SDD) is a written description of a [software](http://en.wikipedia.org/wiki/Software) product, that a software designer writes in order to give a [software development](http://en.wikipedia.org/wiki/Software_development) team overall guidance to the architecture of the software project.A systematic method has to achieve beneficial results in the end. It involves starting with a vague ideaand developing it into a series of steps. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, a design document is required to coordinate a large team under a single vision. A design document needs to be a stable reference, outlining all parts of the software and how they will work. The document is commanded to give a fairly complete description, while maintaining a high-level view of the software.

There are two kinds of design documents called HLDD (high-level design document) and LLDD (low-level design document).

The SDD contains the following documents:

* 1. The [**data design**](http://en.wikipedia.org/wiki/Data-driven_design) describes structures that reside within the software. Attributes and relationships between [data objects](http://en.wikipedia.org/wiki/Data_object) dictate the choice of [data structures](http://en.wikipedia.org/wiki/Data_structures).
  2. The [**architecture design**](http://en.wikipedia.org/wiki/Software_architecture)uses information flowing characteristics, and maps themintotheprogramstructure.Thetransformationmappingmethodisapplied
  3. to exhibit distinct boundaries between incoming and outgoing data. The data flow diagrams allocate control input, processing and output along three separate modules.
  4. The [**interface design**](http://en.wikipedia.org/wiki/Interface_design)describes internal and external program interfaces, as well as the design of human interface. Internal and external interface designs are based on the information obtained from the analysis model.
  5. The [**procedural design**](http://en.wikipedia.org/wiki/Procedural_design)describes structured programming concepts using graphical, tabular and textual notations. These design mediums enable the designer to represent procedural detail that facilitates translation to code. This blueprint for implementation forms the basis for all subsequent software engineeringworked.

## Data Flow Diagram (DFD)

It enables you to represent the processes in your information system from the viewpoint of data. The DFD lets you visualize how the system operates, what the system accomplishes and how it will be implemented, when it is refined with further specification.Data flow diagrams are used by systems analysts to design information-processing systems but also as a way to model whole organizations. You build a DFD at the very beginning of your business process modeling in order to model the functions your system has to carry out and the interaction between those functions together with focusing on data exchanges between process

P CHATBOT GIVE RESPONSE **LEVEL 0**

DOCTOR

PATIENT

ASSK QUES

F

CHATBOX

PATIENT

ANSWER

FWD QUERY

**LEVEL 1**

## Software testing

## Testing

* + - Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not.
    - Generally, no system is perfect due to communication problems between user and developer, time constraints, or conceptual mistakes by developer.
    - To purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved.
    - Testing is the fundamental process of software success.
    - Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase.
    - Testing is used to show incorrectness and considered to success when an error is detected.

## Objectives of Software Testing

* + - **Software Quality Improvement:** The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality ofsoftware.

#### Verification and Validation:

* + - * Verification means to test that we are building the product in right way .i.e. are we using the correct procedure for the development of software so that it can meet the user requirements.
      * Validation means to check whether we are building the right product or not.
    - **Software Reliability Estimation:**The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software liability

## Principles of Software Testing

* + - All tests should be traceable to end user requirements.
    - Tests should be planned long before testing begins
    - Testing should begin on a small scale and progress towards testing inlarge
    - To be most effective testing should be conducted by an independent third party

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

* + - White box testing.
    - Black box testing.

#### White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

#### Block-box testing:

Black box testing is designed to validate functional requirements without regardto the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions,interfaceerrors,errorsindatastructures,errorinfunctional logicarethe errors falling in thiscategory.

## Testing fundamentals

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

## Testing Information flow:

Information flow for testing flows the pattern. Two class of input provided to test the process. The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted andalltheresultsareevaluated.Thatistestresultsarecomparedwithexpectedresults. When erroneous data are uncovered, an error is implied and debugging commences.

**Overview of the testing process**

* Requirements traceability (to ensure that all requirements are tested)
* List of items to be tested
* Schedule
* Recording procedures so that test results can be audited
* Hardware and software requirements
* Constraints

**Implementation**

Implementation is the stage in the project where the theoretical design is turned into the working system and is giving confidence to the new system for the users i.e. will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of method to achieve the changeover, an evaluation, of change over methods. A part of planning a major task of preparing the implementation is the education of users. The more complex system is implemented, the more involved will be the system analysis and design effort required just for implementation. An implementation coordinating committee based on policies of the individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions may regarding the equipment that has to be acquired to implement the new system.

Implementation is the final and important phase. The most critical stage is in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or the inability to handle certain types of transaction while using the newsystem.

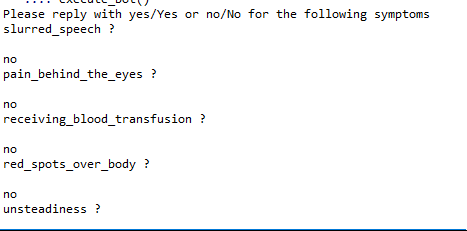
The major elements of the implementation plan are test plan, training plan, equipment installation plan, and a conversion plan.

### Interface

#### Console Based

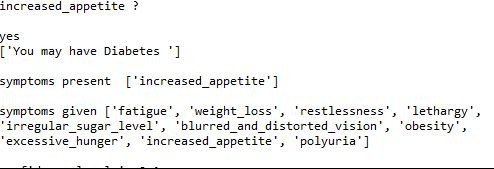
In this, we have to write Yes or No only.

If our Symptoms are not matched then we have to write no on our screen. When our Symptoms will be matched then we just have to write yes.

****

#### Symptoms Window

When we write Yes on our console screen, then our matched problem will be found on screen. And it willalso tell the Symptoms which may a patient have.

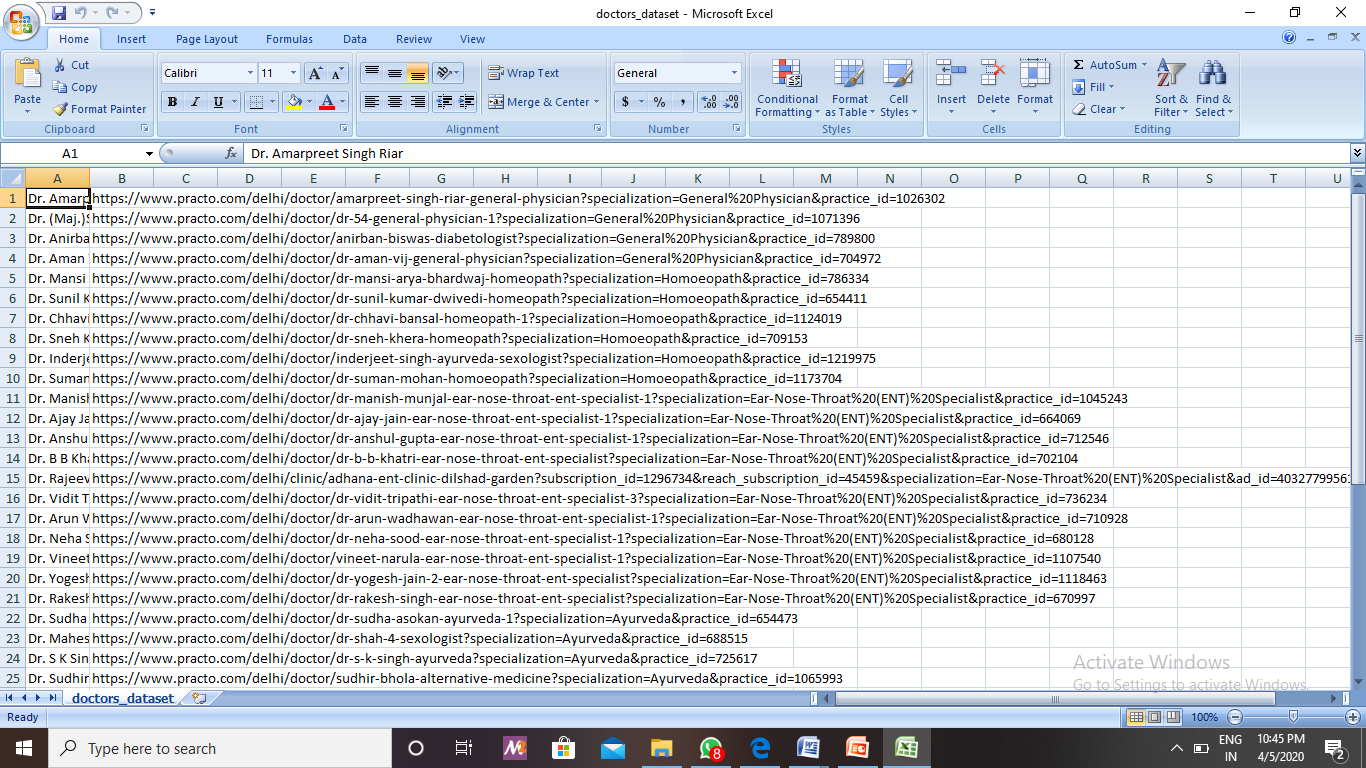
****

**Conclusion**

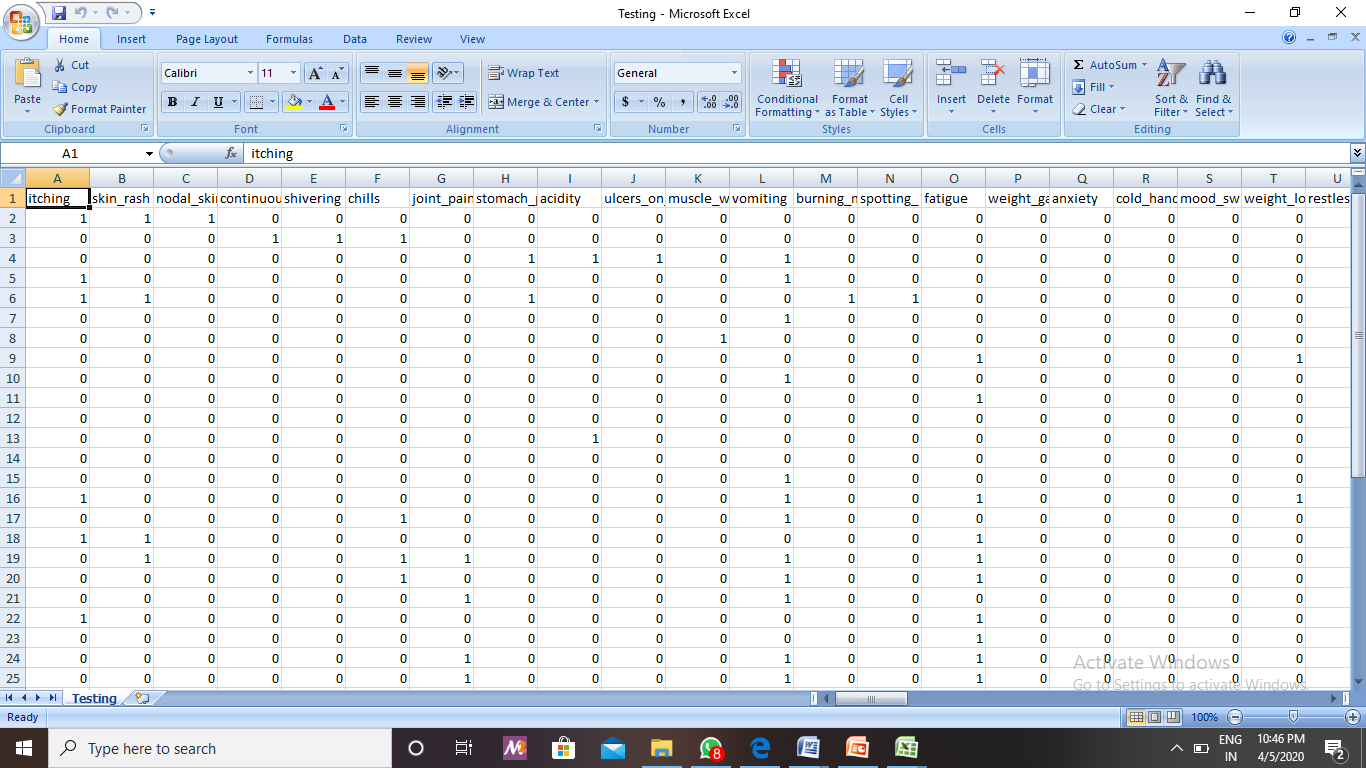
* THE KEY IS TO ENSURE THAT MAJORITY OF THE POPULATION CAN ACCESS AND THEREFORE BENEFIT FROM THIS TECHNOLOGICAL DISRUPTION. IF INDIA IS SUCCESSFUL IN DOING SO, THERE IS AN OPPORTUNITY TO BENEFIT FROM THE POSITIVE CHANGES THAT TECHNOLOGY WILL BRING TO HEALTHCARE.
* ARTIFICIAL INTELLIGENCE HAS A RANGE OF APPLICATIONS ACROSS THE HEALTHCARE SECTOR. BY PERFORMING DESCRIPTIVE, PREDICTIVE AND PRESCRIPTIVE FUNCTIONS, AI IN HEALTHCARE IN INDIA IS CURRENTLY AUGMENTING HUMAN CAPACITY RATHER THAN TO REPLACING HUMAN LABOUR ALTOGETHER.
* **Advantages**
* Save time and money
* Generate new leads
* Guide users
* It provides support 24 x
* **Limitation**
* It requires active internet connection.
* Not all business can use chatbot.
* **Application**
* This system can be used by the multiple peoples to get the counselling sessions online.

**Source Code:**

**Doctor’dataset:**

****

**Testing:-**

****

# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

training\_dataset = pd.read\_csv(r'E:\Chatbot\Training.csv')

test\_dataset = pd.read\_csv(r'E:\Chatbot\Testing.csv')

# Slicing and Dicing the dataset to separate features from predictions

X = training\_dataset.iloc[:, 0:132].values

y = training\_dataset.iloc[:, -1].values

# Dimensionality Reduction for removing redundancies

dimensionality\_reduction = training\_dataset.groupby(training\_dataset['prognosis']).max()

# Encoding String values to integer constants

from sklearn.preprocessing import LabelEncoder

labelencoder = LabelEncoder()

y = labelencoder.fit\_transform(y)

# Splitting the dataset into training set and test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=0)

# Implementing the Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier()

classifier.fit(X\_train, y\_train)

# Saving the information of columns

cols = training\_dataset.columns

cols = cols[:-1]

# Checking the Important features

importances = classifier.feature\_importances\_

indices = np.argsort(importances)[::-1]

features = cols

# Implementing the Visual Tree

from sklearn.tree import \_tree

# Method to simulate the working of a Chatbot by extracting and formulating questions

def execute\_bot():

print("Please reply with yes/Yes or no/No for the following symptoms")

def print\_disease(node):

# print(node)

node = node[0]

# print(len(node))

val = node.nonzero()

# print(val)

disease = labelencoder.inverse\_transform(val[0])

return disease

def tree\_to\_code(tree, feature\_names):

tree\_ = tree.tree\_

# print(tree\_)

feature\_name = [

feature\_names[i] if i != \_tree.TREE\_UNDEFINED else "undefined!"

for i in tree\_.feature

]

# print("def tree({}):".format(", ".join(feature\_names)))

symptoms\_present = []

def recurse(node, depth):

indent = " " \* depth

if tree\_.feature[node] != \_tree.TREE\_UNDEFINED:

name = feature\_name[node]

threshold = tree\_.threshold[node]

print(name + " ?")

ans = input()

ans = ans.lower()

if ans == 'yes':

val = 1

else:

val = 0

if val <= threshold:

recurse(tree\_.children\_left[node], depth + 1)

else:

symptoms\_present.append(name)

recurse(tree\_.children\_right[node], depth + 1)

else:

present\_disease = print\_disease(tree\_.value[node])

print("You may have " + present\_disease)

print()

red\_cols = dimensionality\_reduction.columns

symptoms\_given = red\_cols[dimensionality\_reduction.loc[present\_disease].values[0].nonzero()]

print("symptoms present " + str(list(symptoms\_present)))

print()

print("symptoms given " + str(list(symptoms\_given)))

print()

confidence\_level = (1.0 \* len(symptoms\_present)) / len(symptoms\_given)

print("confidence level is " + str(confidence\_level))

print()

print('The model suggests:')

print()

row = doctors[doctors['disease'] == present\_disease[0]]

print('Consult ', str(row['name'].values))

print()

print('Visit ', str(row['link'].values))

# print(present\_disease[0])

recurse(0, 1)

tree\_to\_code(classifier, cols)

# This section of code to be run after scraping the data

doc\_dataset = pd.read\_csv(r'E:\Chatbot\doctors\_dataset.csv', names=['Name', 'Description'])

diseases = dimensionality\_reduction.index

diseases = pd.DataFrame(diseases)

doctors = pd.DataFrame()

doctors['name'] = np.nan

doctors['link'] = np.nan

doctors['disease'] = np.nan

doctors['disease'] = diseases['prognosis']

doctors['name'] = doc\_dataset['Name']

doctors['link'] = doc\_dataset['Description']

record = doctors[doctors['disease'] == 'AIDS']

# Execute the bot and see it in Action

execute\_bot()# Importing the libraries

import numpy as np

import pandas as pd

# Importing the dataset

training\_dataset = pd.read\_csv(r'E:\Chatbot\Training.csv')

test\_dataset = pd.read\_csv(r'E:\Chatbot\Testing.csv')

# Slicing and Dicing the dataset to separate features from predictions

X = training\_dataset.iloc[:, 0:132].values

Y = training\_dataset.iloc[:, -1].values

# Dimensionality Reduction for removing redundancies

dimensionality\_reduction = training\_dataset.groupby(training\_dataset['prognosis']).max()

# Encoding String values to integer constants

from sklearn.preprocessing import LabelEncoder

labelencoder = LabelEncoder()

y = labelencoder.fit\_transform(Y)

# Splitting the dataset into training set and test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25, random\_state = 0)

# Implementing the Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier()

classifier.fit(X\_train, y\_train)

# Saving the information of columns

cols = training\_dataset.columns

cols = cols[:-1]

# Checking the Important features

importances = classifier.feature\_importances\_

indices = np.argsort(importances)[::-1]

features = cols

# Implementing the Visual Tree

from sklearn.tree import \_tree

# Method to simulate the working of a Chatbot by extracting and formulating questions

def print\_disease(node):

#print(node)

node = node[0]

#print(len(node))

val = node.nonzero()

#print(val)

disease = labelencoder.inverse\_transform(val[0])

return disease

def recurse(node, depth):

global val,ans

global tree\_,feature\_name,symptoms\_present

indent = " " \* depth

if tree\_.feature[node] != \_tree.TREE\_UNDEFINED:

name = feature\_name[node]

threshold = tree\_.threshold[node]

yield name + " ?"

# ans = input()

ans = ans.lower()

if ans == 'yes':

val = 1

else:

val = 0

if val <= threshold:

yield from recurse(tree\_.children\_left[node], depth + 1)

else:

symptoms\_present.append(name)

yield from recurse(tree\_.children\_right[node], depth + 1)

else:

strData=""

present\_disease = print\_disease(tree\_.value[node])

# print( "You may have " + present\_disease )

# print()

strData+="You may have " + present\_disease

red\_cols = dimensionality\_reduction.columns

symptoms\_given = red\_cols[dimensionality\_reduction.loc[present\_disease].values[0].nonzero()]

# print("symptoms present " + str(list(symptoms\_present)))

# print()

strData+="symptoms present " + str(list(symptoms\_present))

# print("symptoms given " + str(list(symptoms\_given)) )

# print()

strData+="symptoms given " + str(list(symptoms\_given))

confidence\_level = (1.0\*len(symptoms\_present))/len(symptoms\_given)

# print("confidence level is " + str(confidence\_level))

# print()

strData+="confidence level is " + str(confidence\_level)

# print('The model suggests:')

# print()

# strData+='The model suggests:'

# row = doctors[doctors['disease'] == present\_disease[0]]

# print('Consult ', str(row['name'].values))

# print()

# strData+='Consult ', str(row['name'].values)+"\n"

# print('Visit ', str(row['link'].values))

#print(present\_disease[0])

# strData+='Visit ', str(row['link'].values)+"\n"

yield strData

def tree\_to\_code(tree, feature\_names):

global tree\_,feature\_name,symptoms\_present

tree\_ = tree.tree\_

#print(tree\_)

feature\_name = [

feature\_names[i] if i != \_tree.TREE\_UNDEFINED else "undefined!"

for i in tree\_.feature

]

#print("def tree({}):".format(", ".join(feature\_names)))

symptoms\_present = []

# recurse(0, 1)

def execute\_bot():

# print("Please reply with yes/Yes or no/No for the following symptoms")

tree\_to\_code(classifier,cols)

# This section of code to be run after scraping the data

doc\_dataset = pd.read\_csv(r'E:\Chatbot\doctors\_dataset.csv', names = ['Name', 'Description'])

diseases = dimensionality\_reduction.index

diseases = pd.DataFrame(diseases)

doctors = pd.DataFrame()

doctors['name'] = np.nan

doctors['link'] = np.nan

doctors['disease'] = np.nan

doctors['disease'] = diseases['prognosis']

doctors['name'] = doc\_dataset['Name']

doctors['link'] = doc\_dataset['Description']

record = doctors[doctors['disease'] == 'AIDS']

record['name']

record['link']

# Execute the bot and see it in Action

#execute\_bot()

**Heathcare console**

from tkinter import \*

from tkinter import messagebox

import os

class QuestionDigonosis(Frame):

objIter=None

def \_\_init\_\_(self,master=None):

master.title("Question")

# root.iconbitmap("")

master.minsize(700,350)

super().\_\_init\_\_(master=master)

self["bg"]="light blue"

self.createWidget()

self.iterObj=None

def createWidget(self):

self.lblQuestion=Label(self,text="Question",width=12,bg="bisque")

self.lblQuestion.grid(row=0,column=0,rowspan=4)

self.lblDigonosis = Label(self, text="Digonosis",width=12,bg="bisque")

self.lblDigonosis.grid(row=4, column=0,sticky="n",pady=5)

# self.varQuestion=StringVar()

self.txtQuestion = Text(self, width=80,height=4)

self.txtQuestion.grid(row=0, column=1,rowspan=4,columnspan=10)

self.varDiagonosis=StringVar()

self.txtDigonosis =Text(self, width=80,height=8)

self.txtDigonosis.grid(row=4, column=1,columnspan=10,rowspan=8,pady=5)

self.btnNo=Button(self,text="No",width=12,bg="bisque", command=self.btnNo\_Click)

self.btnNo.grid(row=12,column=0)

self.btnYes = Button(self, text="Yes",width=12,bg="bisque", command=self.btnYes\_Click)

self.btnYes.grid(row=12, column=1,columnspan=10,sticky="e")

self.btnClear = Button(self, text="Clear",width=12,bg="bisque", command=self.btnClear\_Click)

self.btnClear.grid(row=14, column=0)

self.btnStart = Button(self, text="Start",width=12,bg="bisque", command=self.btnStart\_Click)

self.btnStart.grid(row=14, column=1,columnspan=10,sticky="e")

def btnNo\_Click(self):

global val,ans

global val,ans

ans='no'

str1=QuestionDigonosis.objIter.\_\_next\_\_()

self.txtQuestion.delete(0.0,END)

self.txtQuestion.insert(END,str1+"\n")

def btnYes\_Click(self):

global val,ans

ans='yes'

str1=QuestionDigonosis.objIter.\_\_next\_\_()

self.txtDigonosis.delete(0.0,END)

self.txtDigonosis.insert(END,str1+"\n")

def btnClear\_Click(self):

self.txtDigonosis.delete(0.0,END)

self.txtQuestion.delete(0.0,END)

def btnStart\_Click(self):

execute\_bot()

self.txtDigonosis.delete(0.0,END)

self.txtQuestion.delete(0.0,END)

self.txtDigonosis.insert(END,"Please Click on Yes or No for the Above symptoms in Question")

QuestionDigonosis.objIter=recurse(0, 1)

str1=QuestionDigonosis.objIter.\_\_next\_\_()

self.txtQuestion.insert(END,str1+"\n")

class MainForm(Frame):

main\_Root = None

def destroyPackWidget(self, parent):

for e in parent.pack\_slaves():

e.destroy()

def \_\_init\_\_(self, master=None):

MainForm.main\_Root = master

super().\_\_init\_\_(master=master)

master.geometry("300x250")

master.title("Account Login")

self.createWidget()

def createWidget(self):

self.lblMsg=Label(self, text="Select Your Choice", bg="blue", width="300", height="2", font=("Calibri", 13))

self.lblMsg.pack()

self.btnLogin=Button(self, text="Login", height="2", width="30", command=self.lblLogin\_Click)

self.btnLogin.pack()

self.btnRegister=Button(self, text="Register", height="2", width="30", command=self.btnRegister\_Click)

self.btnRegister.pack()

def lblLogin\_Click(self):

self.destroyPackWidget(MainForm.main\_Root)

frmLogin=Login(MainForm.main\_Root)

frmLogin.pack()

def btnRegister\_Click(self):

self.destroyPackWidget(MainForm.main\_Root)

frmSignUp = SignUp(MainForm.main\_Root)

frmSignUp.pack()

class Login(Frame):

main\_Root=None

def destroyPackWidget(self,parent):

for e in parent.pack\_slaves():

e.destroy()

def \_\_init\_\_(self, master=None):

Login.main\_Root=master

super().\_\_init\_\_(master=master)

master.title("Login")

master.geometry("300x250")

self.createWidget()

def createWidget(self):

self.lblMsg=Label(self, text="Please enter details below to login",bg="blue")

self.lblMsg.pack()

self.username=Label(self, text="Username \* ")

self.username.pack()

self.username\_verify = StringVar()

self.username\_login\_entry = Entry(self, textvariable=self.username\_verify)

self.username\_login\_entry.pack()

self.password=Label(self, text="Password \* ")

self.password.pack()

self.password\_verify = StringVar()

self.password\_login\_entry = Entry(self, textvariable=self.password\_verify, show='\*')

self.password\_login\_entry.pack()

self.btnLogin=Button(self, text="Login", width=10, height=1, command=self.btnLogin\_Click)

self.btnLogin.pack()

def btnLogin\_Click(self):

username1 = self.username\_login\_entry.get()

password1 = self.password\_login\_entry.get()

# messagebox.showinfo("Failure", self.username1+":"+password1)

list\_of\_files = os.listdir()

if username1 in list\_of\_files:

file1 = open(username1, "r")

verify = file1.read().splitlines()

if password1 in verify:

messagebox.showinfo("Sucess","Login Sucessful")

self.destroyPackWidget(Login.main\_Root)

frmQuestion = QuestionDigonosis(Login.main\_Root)

frmQuestion.pack()

else:

messagebox.showinfo("Failure", "Login Details are wrong try again")

else:

messagebox.showinfo("Failure", "User not found try from another user\n or sign up for new user")

class SignUp(Frame):

main\_Root=None

def destroyPackWidget(self,parent):

for e in parent.pack\_slaves():

e.destroy()

def \_\_init\_\_(self, master=None):

SignUp.main\_Root=master

master.title("Register")

super().\_\_init\_\_(master=master)

master.title("Register")

master.geometry("300x250")

self.createWidget()

def createWidget(self):

self.lblMsg=Label(self, text="Please enter details below", bg="blue")

self.lblMsg.pack()

self.username\_lable = Label(self, text="Username \* ")

self.username\_lable.pack()

self.username = StringVar()

self.username\_entry = Entry(self, textvariable=self.username)

self.username\_entry.pack()

self.password\_lable = Label(self, text="Password \* ")

self.password\_lable.pack()

self.password = StringVar()

self.password\_entry = Entry(self, textvariable=self.password, show='\*')

self.password\_entry.pack()

self.btnRegister=Button(self, text="Register", width=10, height=1, bg="blue", command=self.register\_user)

self.btnRegister.pack()

def register\_user(self):

# print(self.username.get())

# print("Hello")

file = open(self.username\_entry.get(), "w")

file.write(self.username\_entry.get() + "\n")

file.write(self.password\_entry.get())

file.close()

self.destroyPackWidget(SignUp.main\_Root)

self.lblSucess=Label(root, text="Registration Success", fg="green", font=("calibri", 11))

self.lblSucess.pack()

self.btnSucess=Button(root, text="Click Here to proceed", command=self.btnSucess\_Click)

self.btnSucess.pack()

def btnSucess\_Click(self):

self.destroyPackWidget(SignUp.main\_Root)

frmQuestion = QuestionDigonosis(SignUp.main\_Root)

frmQuestion.pack()

root = Tk()

frmMainForm=MainForm(root)

frmMainForm.pack()

root.mainloop()

**Bibliography**

#### Books

Learn PYTHON the HARD WAY(Third Edition) Introduction to Machine Learning

Machine Learning with Python Cookbook

**Website**

[**www.w3school.com**](http://www.w3school.com)

[**www.stackoverflow.com**](http://www.stackoverflow.com)

[**www.it-ebooks.com**](http://www.it-ebooks.com)