

HealthCareChatbot

*A Project Report submitted in the partial fulfillment of the Requirements
for the award of the degree*

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

CH.LikithaBhavya	(20471A05K4)
A. Bhargavi	(20471A05J6)
SK. Meerabi	(20471A05M9)

Under the esteemed guidance

Dodda. Venkata Reddy, M.Tech,(Ph.D)
Assistant. Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

**(Affiliated to JNTUK, Kakinada, Approved by AICTE & Thrice Accredited by NBA)
2023 - 2024**

**NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project that is entitled with the name “HealthCareChatbot” is a bonafide work done by the team CH.LikithaBhavya (20471A05K4), A.Bhargavi(20471A05J6), Sk.Meerabi (20471A05M9) in Partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in Department of COMPUTER SCIENCE AND ENGINEERING during 2023-2024.

PROJECT GUIDE

**Dodda. Venkata Reddy, M.Tech(Ph.D)
Assistant Professor**

PROJECT CO-ORDINATOR

**M. Sireesha, M.Tech.Ph.D,
Assoc. Professor**

HEAD OF THE DEPARTMENT

**Dr. S. N. Tirumala Rao, M.Tech.,Ph.D,
Professor & HoD**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri M.V.Koteswara Rao,B.Sc., who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal Dr.M.Sreenivasa Kumar, M.Tech., Ph.D., MISTE., FIE(I.), for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards Dr.S.N.Tirumala Rao, M.Tech.,Ph.D., HoD of CSE department and also to our guide Dodda.Venkata Reddy, M.Tech, Associate professor of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards Ms Sireesha .M, M.Tech., Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

CH.LikithaBhavya	(20471A05K4)
A.Bhargavi	(20471A05J6)
SK.Meerabi	(20471A05M9)

ABSTRACT

To start a good life healthcare is more important. But it is very difficult to obtain the consultation with the doctor in case of any health issues. The proposed idea is to create a health care Chabot system using Artificial Intelligence that can diagnose the disease and provide basic details about the disease before consulting a doctor. The system provides text (or) voice assistance that means user can use own convenient language. Bot will provides which type of disease you have based on user symptoms and appeared doctor details respective to user disease. The Chabot will clarify the users symptoms with serious of questions and the symptom conformation will be done. The disease will be categorized as minor and major disease. Chatbot will reply whether it is a major or minor disease .if it is a major disease user will be suggested with the doctor details and analagesics for further treatment and also provides food suggestion that means which type of food you have to take. The user can achieve the real benefit of a chatbot only when it can diagnose all kind of disease and provide necessary information. A text-to-text diagnosis Bot engages patients in conversation about their medical issues and provides a personalized diagnosis based on their symptoms. Hence, people will have an idea about their health and have the right protection.

TABLE OF CONTENTS

	Page No
1. INTRODUCTION	1
2. LITERATURE SURVEY	4
3. SYSTEM ANALYSIS	7
3.1 Existing System	8
3.2 Proposed System	9
3.3 Feasibility Study	10
3.3.1 Economical feasibility	
3.3.2 Technical feasibility	
3.3.3 System feasibility	
4. SYSTEM REQUIREMENTS	12
4.1 Hardware Specification	
4.2 Software Specification	
4.3 Software Descriptions	13
5. MODULE DESCRIPTION	57
6. SYSTEM DESIGN	58
6.1 System architecture	60
6.2 Data flow diagram	61
6.3 Methodology	62
7. CODING	72
8. SYSTEM TESTING	88
8.1 Testing	
8.1.1 Unit Testing	
8.1.2 Integration Testing	
8.1.3 Functional Testing	

8.1.4 System Testing	
8.1.5 White box Testing	
8.1.6 Black Box Testing	
9. SCREENSHOTS	91
10. CONCLUSION	100
11. FUTURE ENHANCEMENT	101
12. REFERENCES	103

1.

INTRODUCTION

Now a days , health care is very important in our life. Today's people are busy with their works at home, office works and more addicted to Internet. They are not concerned about their health .So they avoid to go in hospitals for small problems. it may become a major problem. So we can provide an idea is to create a health care Chabot system using AI that can diagnosis the disease and provide basic information about the disease before consulting a doctor. Which helps the patients know more about their disease and improves their health User can achieve the all kind of disease information. The system application uses question and answer protocol in the form of chatbot to answer user queries. The response to the question will be replied based on the user query. The significant keywords are fetched from the sentence and answer to those sentences. If match is discovered or significant answer will be given or similar answers will be displayed. Bot will diagnosis which type of disease you have based on user symptoms and also gives doctor details of particular disease. It may reduce their health issues by using this application system. The system is developed to reduce the healthcare cost and time of the users as it is not possible for the users to visit the doctors or experts when immediately needed.

Artificial Intelligence is based on how any device perceives its Environment and takes actions based on the perceived data to achieve the result successfully. It is the study of intelligent agents. The term "artificial intelligence" is applied when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving. Artificial Intelligence gives the supreme power to mimic the human way of thinking and behaving to a computer. A chatbot (also known as a talkbot, chatterbot, Bot, IMbot, interactive agent, or Artificial(ConversationalEntity) is a computer program which conducts a conversation via auditory or textual methods. These programs are designed to provide a clone of how a human will chat and thereby it acts as a conversational partner rather than humans. For various practical purposes like customer service or information acquisition, , chatbot is being used in the dialog system. Mostly chatbots uses natural language processing for interpreting the user input and generating the corresponding response but certain simpler systems searches for the keyword within the text and then provides a reply based on the matching keywords or certain pattern. Today, chatbots are part of virtual assistants such as Google Assistant, and are accessed via many organizations' apps, websites, and on instant

messaging platforms. Non-assistant applications include chat bots used for entertainment purposes, for research, and social bots which promote a particular product, candidate, or issue.

Chatbot's are such kind of computer programs that interact with users using natural languages. For all kind of chat bots the flow is same, though each chatbot is specific in its own area knowledge that is one input from human is matched against the knowledge base of chatbot. Chatbot's work basically on Artificial intelligence so using this capability we have decided to add some contribution to the Health Informatics. The high cost of our healthcare system can often be attributed to the lack of patient engagement after they leave clinics or hospitals. Various surveys in this area have proved that that chatbot can provide healthcare in low costs and improved treatment if the doctors and the patient keep in touch after their consultation. To answer the questions of the user chatbot is used. There is very less number of chatbots in medical field.

The main purpose of the scheme is to build the language gap between the user and health providers by giving immediate replies to the Questions asked by the user. Today's people are more likely addicted to internet but they are not concern about their personal health. They avoid to go in hospital for small problem which may become a major disease in future. Establishing question answer forums is becoming a simple way to answer those queries rather than browsing through the list of potentially relevant document from the web. Many of the existing systems have some limitation such as There is no instant response given to the patients they have to wait for experts acknowledgement for a long time. Some of the processes may charge amount to perform live chat or telephony communication with doctors online. This system allows computer to communication between human to computer by using natural language processing (NLP). There are three analyses which understand natural language i.e. identification of main linguistic relations is completed to parse subject into object of the sentences. After that Chatbot is an Entity which imitate human discussion in its particular accepted set-up together with a text or vocal language with techniques such as Natural Language Processing (NLP). The aim of this system is to replicate a person's discussion. The development of chatbot application canbe done with making a user interface to send input and receive response. It is a system that interact with user by keeping the track of the state of interaction and recollecting the preceding commands to give functionality. The medical chat-bots can be developed by using artificial algorithms that scrutinize user's queries and recognize it and give reply to related query. A big disease can start from small problems such as headache which feels normal but it may beginning of big disease such as brain tumor

.most of the disease can be identified by common symptoms so the disease can be predicted if the patient body is analyzed periodically[6].The system give response by use of an efficient Graphical User Interface such that if actual person is chatting with the user. chatterbot that can be used in various fields like education, healthcare, and route assistance[8]The central part of the chat-bots includes MySQL. It is an interactive system solve users query regarding medicine. so they can get correct guidance for treatment through android app by using Google API.

Computers give us information; they engage us and help us in a lot of manners. A chatbot is a program intended to counterfeit smart communication on a text or speech. Yet, this paper concentrates only on text.

These systems can learn themselves and restore their knowledge using human assistance or using web resources. This application is incredibly fundamental since knowledge is stored in advance. The system application uses the question and answer protocol in the form of a chatbot to answer user queries. This system is developed to reduce the healthcare cost and time of the users, as it is not possible for the users to visit the doctors or experts when immediately needed.

The response to the question will be replied based on the user query and knowledge base. The significant keywords are fetched from the sentence and answer to those sentences. If the match is discovered or the significant answer will be given or similar answers will be displayed.

The complex questions and answers present in the database are viewed and answered by an expert. Here the users can personally ask any questions regarding healthcare, as not much time will be wasted by the user for consulting a doctor. The input sentence of the chat pattern is stored in an RDBMS. The chatbot would coordinate the input sentence from the user question with the knowledge base. Each query is compared with the knowledge database of the chatbot. The important keywords are extracted from the given input sentence and the sentence similarity is found. The keyword ranking and sentence similarity are found using the N-gram, TF-IDF, and cosine similarity. The interfaces are standalone built using the JAVA programming language.

2.

LITERATURE REVIEW

Simon Hoermann[1] discuss the current evidence for the feasibility and effectiveness of online one-on-one mental health interventions that use text-based synchronous chat. Synchronous written conversations (or -chats||) are becoming increasingly popular as Web-based mental health interventions. This review is based on an evaluation of individual synchronous Web-based chat technologies. Through the current evidence of the application of this technology, the tentative support for mode of intervention is seen. Interventions utilizing text-based synchronous communication showed better outcomes compared with Waitlist conditions and overall equivalent outcomes compared with Treatment As usual, and were at least as good as the comparison interventions. However, the issue of whether these technologies are cost effective in clinical practice remains a consideration for future research studies.

Saurav Kumar Mishra[2] says that the chatbot will act as a virtual doctor and makes possible for the patient to interact with virtual doctor. Natural language processing and pattern matching algorithm for the development of this chatbot. It is developed using the python Language. Based on the survey given it is found that the no of correct answer given by the chatbot is 80% and incorrect/ambiguous answer given is 20%. From this survey of chatbot and analysis of result suggested that this software can be used for teaching and as a virtual doctor for awareness and primary care.

DivyaMadhu[3] proposed an idea in which the AI can predict the diseases based on the symptoms and give the list of available treatments If a person's body is analyzed periodically, it is possible to predict any possible problem even before they start to cause any damage to the body. Some Challenges are research and implementation costs, and government regulations for the successful implementation of personalized medicine, they are not mentioned in the paper.

HameedullahKazi[4], describes the development of a chatbot for medical students, that is based on the open source AIML based Chatterbean. The AIML based chatbot is customized to convert natural language queries into relevant SQL queries. A total of 97 question samples were collected and then those questions were divided into categories depending on the type of question. According to the number of questions in each category the resultant categories were ranked. Questions were based on queries, where 47% are of posed questions

Other categories has less than 7%. The system has not been specially designed for the task of supporting natural dialog in chatbots or, providing responses to student queries Banks implement technology to strengthen their processing capacity, acquire a larger customer base and expand the services they could offer (Consoli2005). The use of internet banking has grown in demand enormously in the last decade. -15% of branch customers use online banking once a day, 59% once a week, 77% at least once a month and 53% were confident in carrying out the best part of their banking online|| Online banking has become more popular as it negates the need for customers to visit their local branch as they can manage their finances on the go to meet the demand of modern life. This is evident as branchless banks are now emerging from the industry such as Atom Bank and many banks now closing some of their branches. This is evident with the recent closure of 11 Ulster Bank Branches in NI due to the increased number of customers performing their banking online (The Belfast Telegraph). recent study by Ling et al., (2016), notes that most internet banking service providers struggle to get many of their customers to use their service. They identify lack of customer satisfaction when using online banking services to be a major cause. -Service quality, web design and content, security, privacy, speed and convenience|| (Ling et al., 2016) are as the top factors influencing customer satisfaction This suggests that there is a lack of technology in place to enhance the customer online banking experience which could be improved by integrating a chatbot to provide an efficient, convenient and personal service

Chat bots in Industry

Most businesses and organizations are understanding the potential benefits of machine learning and artificial intelligence to have a positive change on how they perform business. Artificial intelligence has progressed to allow the development of more sophisticated chatbots. Organizations are focusing on specific areas of user engagement that take up a lot of time but can be replaced through the use of a chatbot. Chatbots can understand what the customer needs from a single text instead of the customer having to follow a process of multiple steps. Chatbots are used to automate customer service and reduce manual tedious tasks performed by employees so they can spend their time more productively on higher priority tasks.

Most online banking services would benefit from having a chatbot integrated into their services. The uses of bots help internet banking service providers establish a better relationship with its customers. Customers can get answers to query's immediately, conduct e-commerce and banking operations all from within the one bot conversation. This is another

benefit of using an online banking system with an integrated chatbot as ~87% customers that bank online prefer to execute their personal finance operations within a single site (Dauda, et al., 2015).

WHAT IS CHATBOT?

In general terms, a bot is nothing but a software that will perform automatic tasks. In other terms, a bot is a computer program that is designed to communicate with human users through the internet. The most natural definition of a chatbot is – a developed a program that can have a discussion/conversation with a human. For example, any user could ask the bot an inquiry or a statement, and the bot will respond or perform an activity as appropriate. Chatbot interacts on a format similar to instant messaging. By artificially replicating the patterns of human interactions in machine learning allows computers to learn by themselves without programming natural language processing. While a bot is a computer's ability to understand human speech or text short for chat robot. A chatbot is merely a computer program that fundamentally simulates human conversations. It allows a form of interaction between a human and a machine the communication, which happens via messages or voice command. A chatbot is programmed to work independently from a human Operator. It can answer questions formulated to it in natural language and respond like a real person. It provides responses based on a combination of predefined scripts and machine learning applications. When it is asked a question, the chatbot will respond based on the knowledge database available to it at that point in time. If the conversation introduces a concept it is not programmed to understand, it will either deflect the conversation or potentially pass the communication to a human operator. Either way, it will also learn from that interaction as well as from future interactions. Thus, the chatbot will gradually grow in scope and gain relevance.

WHAT IS HEALTHCARE CHATBOT?

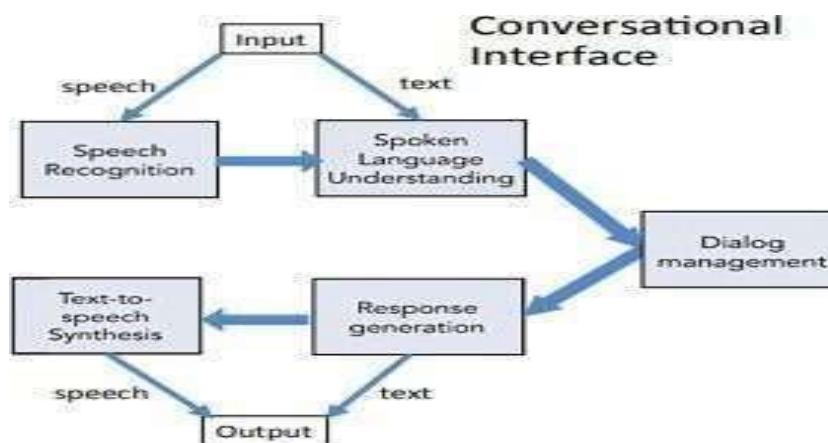
Enter chatbot, healthcare chatbots are software integrated into either a website or mobile app and possess any combination of the following attributes:

Natural language processing :The ability for a chatbot or other form of artificial intelligence to understand aspects of human language including syntax and semantics from users of various backgrounds.

Contextual AI: contextual AI takes into consideration the intent of the user as they ask questions and submit responses, thus ensuring the best relevant conversation takes place.

Chatbot Architecture

Previously chat bots solely supported a single adjacency pair, also known as a one-shot conversation. However, modern chatbots can sustain multiple adjacency pairs, remembering states and contexts between conversations and have the capability to associate data in different adjacency pairs which is related. This is the bots ability to preserve the conversation. A chatbot consists of four main parts: front- end, knowledge-base, back-end and corpus which is the training data. The front end is accountable for enabling communication between the bot and the user. The NLU utilises Artificial intelligence methods to identify the intent and context of the user input. An appropriate response is generated from the users' intent. The



knowledge base defines the chatbot knowledge, which is created within the NLU and supported by the back-end, the back-end applies the domains corpus to produce the knowledge base. Input can be supplied to the chatbot in the form of text or speech. The Input is sent to the dialog management system which is the NLU in this case, which determines an appropriate response and amends the chatbots state accordingly to carry out the required action. The chatbot will produce text and speech responses in the form of both text and speech.

3.

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Many of the existing systems have live chats through texts and some limitation such as there is no instant response given to the patients they have to wait for experts acknowledgement for a long time. Some of the processes may charge amount to perform live chat or telephony communication.

Here the studies are based on to recognize emotions classification using AI methods. The studies train emotions classification models from a lot of labelled data based on RNN, deep learning, convolutional neural network. Linguistic interaction is most important in counselling using NLP and NLG to understand dialogues of users. Here the multi-modal approach is used of emotion-recognition. They have collected corpuses to learn semantic information of words and represent as vector using the word vector, synonym knowledge of lexical are collected. [1]

In this paper a voice recognition chat-bot is developed, if the questions are not understood asked to the bot is further processed using the third party expert-system. The web-bots are created as text-based web-friends, an entertainer for the user. Here they focused on the improved system if the system is not only text-based but also voice-based trained. Here the voice recognition requires a 2 part process of capturing and analysis of an input signal. Server response recognition data retrieval and information output. The server used here is SOAP based on black box approach. The use of expert system allows unlimited and autonomous intelligence improvements. [2]

This chatbot aims to make a conversation between human and machine. Here the system stores the knowledge database to identify the sentence and making a decision to answer the question. The input sentence will get the similarity score of input sentences using bigram. The chatbot knowledge is stored in RDBMS. [3]

The chatbot implemented using pattern comparison in which the order of the sentence is recognized and saved response pattern. Here the author describes the implementation of the chatbot Operating system, software, programming language, and database. How results input and output is stored. Here the input is taken using text () function and other punctuation is removed using trim () function and random () function is used to choose a response from the database. The chatbot is used for an entertainment purpose. [4]

Here they use n-gram technique for extracting the words from the sentences. Here n-gram is used for comparison and deduction of the input with case data using Moro phonemes and phonemes as the deciding parameter. Probability analysis for the closest match is performed. The final expression is redirected through an expert system. [5]

The chatbot developed here for healthcare purposes for the android application. The user sends the text message or voice message using Google API. Here the user gets only related answer from the chatbot. SVM algorithm is used to classify the dataset. Here the Porter algorithm is used to discard unwanted words like suffixes or prefixes. [6]

The different documents served in web, the content is checked by tagging the dataset using n-gram based low dimensional demonstration, TF-IDF matrix that generates S, U, and V and finally multiplying the 3 matrices cosine similarity is calculated. [7]

Here the chatbot is created for the customer service that functions as public health service. The application uses N- gram, TF-IDF and cosine similarity. The knowledge base is created for storing the question and answer. The application clearly shows extracted the keyword from the question ad by using unigram, bigram, and trigram which helps in fast answering. [8]

Here says that the chatbot will act as a virtual doctor and makes possible for the patient to interact with virtual doctor. Natural language processing and pattern matching algorithm for the development of this chatbot. It is developed using the python Language. Based on the survey given it is found that the no of correct answer given by the chatbot is 80% and incorrect/ambiguous answer given is 20%.From this survey of chatbot and analysis of result suggested that this software can be used for teaching and as a virtual doctor for awareness and primary care.

Disadvantages in Existing system:

- It takes more time to response to the user question
- Pay some charges to perform live chat

3.2 PROPOSED SYSTEM:

The proposed system provides a text-to- text(or) voice conversational agent that asks the user about their health issue .The user can chat as if chatting with a human. The bot then ask the user a series of questions about their symptoms to diagnose the disease. It gives suggestions about the different symptoms to clarify the disease. Based on the reply from the user the accurate disease is found and it suggests the doctor who needs to be consulted in case of major

disease. The system remembers past responses and asks progressively more specific questions in order to obtain a good diagnosis.

- In our proposed system the user can chat with the bot regarding the query through voice or text.
- The system uses an expert system to answer the queries
- User can also view the available doctors and booking appointments for that particular disease.
- This system can be used by the multiple users to get the counselling sessions online.
- The data of the chatbot stored in the database in the form of pattern-template.
- Bot will provide analgesics and food suggestions that means which food you have to take based on the disease
- **Advantages in proposed system:**
- Reducing health care cost
- Save the user time
- Don't go to hospital for even any small problem

3.3 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

3.3.1 ECONOMICAL FEASIBILITY

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

3.3.2 TECHNICAL FEASIBILITY

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

3.3.3 SOCIAL FEASIBILITY

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

4.

SYSTEM REQUIREMENTS

4.1 HARDWARE SPECIFICATION:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 Mb.

4.2 SOFTWARE SPECIFICATION:

- Operating system : Windows 7/10.
- Coding Language : PYTHON ,HTML,CSS,JS,FLASK
- IDE : python 2.7.15

4.3 SOFTWARE DESCRIPTION:

Python language:

Python is the popular programming language. In-fact you will see the code of python language as English language. Python 2 and Python 3 are the two major versions of Python. But , Python Installation for both the versions are different. In addition ,Commands and path environment variables for both the versions are different. You can choose your python version according to your requirement. Beginners always faces challenges of installing Python in their operating system.

Python is a popular object-oriented programming language having the capabilities of high-level programming language. It's easy to learn syntax and portability capability makes it popular these days. The followings facts gives us the introduction to Python.

Python was developed by Guido van Rossum at StichtingMathematisch Centrum in the Netherlands. It was written as the successor of programming language named

‘ABC’. Its first version was released in 1991. The name Python was picked by Guido van Rossum from a TV show named Monty Python’s Flying Circus. It is an open source programming language which means that we can freely download it and use it to develop programs. It can be downloaded from www.python.org. Python programming language is having the features of Java and C both. It is having the elegant ‘C’ code and on the other hand, it is having classes and objects like Java for object-oriented programming. It is an interpreted language, which means the source code of Python program would be first converted into bytecode and then executed by Python virtual machine.

This article Python Installation tutorial is for them. Introduction to Python Programming article is a complete guide for beginners to know more about Python. You will know how to install python in Windows, Linux, MacOS. How to check your Python versions .

You make sure to install python carefully by reading all the steps described here otherwise it may be wrongly install.Python Installation Tutorial for Windows

In this section, you will know how to do Python installation in the Windows OS. First we will cover for the latest Python 3 version and then Python 2 version. You can either install both the versions or choose the version according to your end goal.

Python Features

Python is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.

In Python, we don’t need to declare the type of variable because it is a dynamic typed language.

Forexample,x=10

here x can be anything such as String, int etc.

Features in Python

There are many features in Python, some of which are discussed below –

1. Easy to code:

Python is high level programming language.Python is very easy to learn language as

compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

2. FreeandOpenSource:

Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword. Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3. Object-OrientedLanguage:

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4. GUIProgrammingSupport:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython (or)Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.

5. High-LevelLanguage:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensiblefeature:

Python is a **Extensible** language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. PythonisPortablelanguage:

Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. PythonisIntegratedlanguage:

Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

9. InterpretedLanguage:

Python is an Interpreted Language. because python code is executed line by line at a

time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code.The source code of python is converted into an immediate form called bytecode.

10. LargeStandardLibrary

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. DynamicallyTypedLanguage:

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance. because of this feature we don't need to specify the type of variable.

Applications of Python

These are some real-world Python applications:

- Web and Internet Development
- Desktop GUI Applications
- Science and Numeric
- Software Development
- Education
- Database Access
- Network Programming
- Games and 3D Graphics

Let's discuss each of them in detail:

1. Web and Internet Development

Python lets you develop a web application without too much trouble. It has libraries for internet protocols like HTML and XML, JSON, e-mail processing, FTP, IMAP, and easy-to-use socket interface. Yet, the package index has more libraries:

- **Requests** – An HTTP client library
- **BeautifulSoup** – An HTML parser
- **Feedparser** – For parsing RSS/Atom feeds
- **Paramiko** – For implementing the SSH2 protocol
- **Twisted Python** – For asynchronous network programmings

Desktop GUI Applications

Most binary distributions of Python ship with Tk, a standard GUI library. It lets you draft a user interface for an application. Apart from that, some toolkits are available:

- **wxWidgets**
- **Kivy** – for writing multitouch applications
- **Qt** via **pyqt** or **pyside**

And then we have some platform-specific toolkits:

- **GTK+**
- Microsoft Foundation Classes through the **win32** extensions
- **Delphi**

3. Science and Numeric Applications

This is one of the very common applications of Python programming. With its power, it comes as no surprise that Python finds its place in the scientific community. For this, we have:

- **SciPy** – A collection of packages for mathematics, science, and engineering.
- **Pandas** – A data-analysis and -modeling library
- **IPython** – A powerful shell for easy editing and recording of work sessions. It also supports visualizations and parallel computing.
- **Software Carpentry Course** – It teaches basic skills for scientific computing and running bootcamps. It also provides open-access teaching materials.
- Also, **NumPy** lets us deal with complex numerical calculations.

4. Software Development

Software developers make use of Python as a support language. They use it for build-control and management, testing, and for a lot of other things:

- **SCons** – for build-control
- **Buildbot, Apache Gump** – for automated and continuous compilation and testing
- **Roundup, Trac** – for project management and bug-tracking.
- Roster of Integrated Development Environments

5. Education

Thanks to its simplicity, brevity, and large community, Python makes for a great introductory programming language. Applications of Python programming in education has a huge scope as it is a great language to teach in schools or even learn on your own.

6. Business

Python is also a great choice to develop **ERP** and **e-commerce** systems:

- **Tryton** – A three-tier, high-level general-purpose application platform.
- **Odoo** – A management software with a range of business applications. With that, it's an all-rounder and forms a complete suite of enterprise-management applications in-effect.

7. Database Access

With Python, you have:

- Custom and ODBC interfaces to MySQL, Oracle, PostgreSQL, MS SQL Server, and others. These are freely available for download.
- Object databases like Durus and ZODB
- Standard Database API

8. Network Programming

With all those possibilities, how would Python slack in network programming? It does provide support for lower-level network programming:

- **Twisted Python** – A framework for asynchronous network programming.
We mentioned it in section 2.
- An easy-to-use **socket interface**

9. Games and 3D Graphics

Safe to say, this one is the most interesting. When people hear someone say they're learning Python, the first thing they get asked is – "So, did you make a game yet?"

PyGame, PyKyra are two frameworks for game development with Python. Apart from this, we also get a variety of 3D-rendering libraries.

10. Artificial Intelligence and Machine Learning

The IT industry went far beyond the science fiction ideas and our future is moving towards Artificial Intelligence. People are keen to move to technologies where machines can mimic human behaviours and most of the human work is being automated by machines. There is also a need to process and analyze huge volumes of data.

Statistically speaking that Python is the most popular language for Artificial intelligence and Machine Learning.

The several libraries and frameworks for Artificial intelligence are listed below.

- Tensorflow
- Scikit-learn
- NLTK
- Caffee
- PyTorch
- Keras
- Pandas
- Accord.NET

11. Prototyping

Programming in Python is easy when you compare it with other languages. It has easy syntax and it offers concise solutions for implementing all types of functionalities. This is the main reason why Python can be applied in prototyping stages of software development.

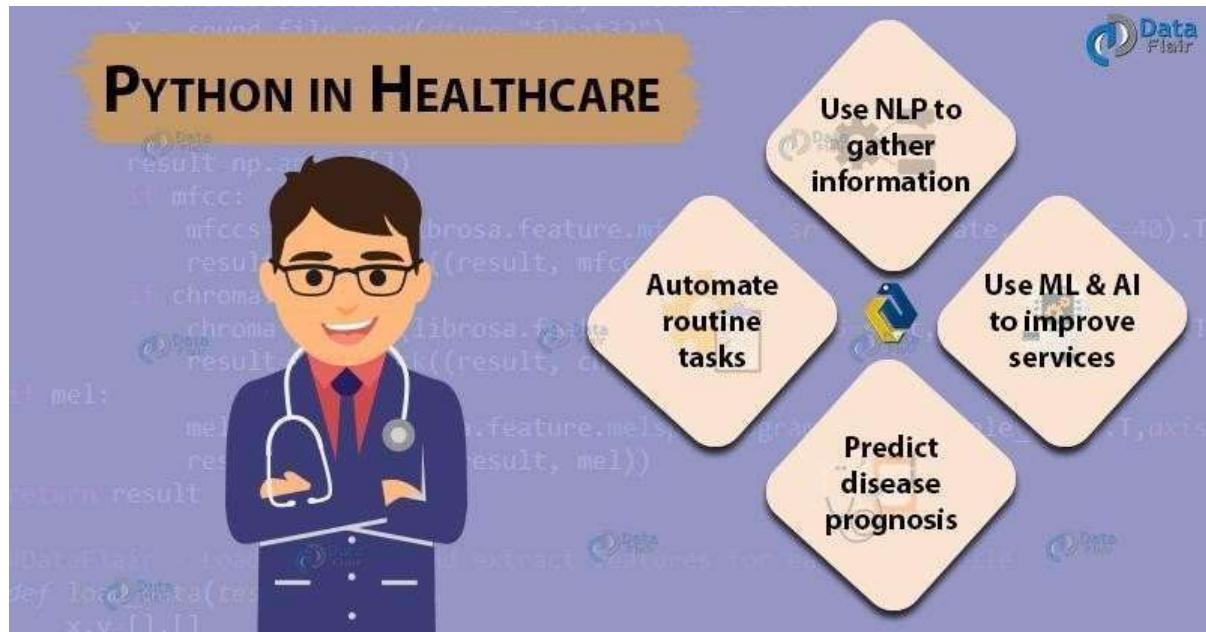
Python in Healthcare – Good Health now depends on Code too!!

With many functionalities to offer, Python lets you develop a prototype of applications faster. Many organizations and firms use Python at the prototyping stage of a project. Python in Healthcare – Good Health now depends on Code too!!

We have always known python as an object-oriented, high-level programming language with dynamic semantics. It has attracted users from different platforms for its advantages in rapid application development and its dynamic building options. Little did we know that it is considered to be one of the -safest^{ll} programming languages too apart from all the popularity because of its functionality. Also, Python plays an important role in the healthcare sector.

Although it is tough for a normal person to believe that programming languages like Python are important in healthcare. There's not only one reason that makes python in the limelight of healthcare. There are a number of reasons which make Python an important asset for the ones researching in the healthcare department.

Let's explore, what is the importance of Python in healthcare and why it is considered to be one of the safest languages.



Before moving ahead in Python in Healthcare article, read a little about Python Frameworks that are Django and Flask.

Django and Flask – The Python Frameworks

The performance of Python is appreciated against abilities like meeting deadlines, quality and amount of code. To achieve the same, Python is present with a framework Django.

Its trustworthy modules are so effective that you don't need to develop them by yourself. Also, the built-in maintenance against the web-app attack adds to its utility. Django framework allows developers to meet their requirements of any business idea related to eHealth or telemedicine projects.

There is also **Flask**, again a Python-based framework often used for building prototypes because of its great efficiency and can be extended to the full-stack framework with existing extensions. Its uses are still less than that of Django because it requires more time for configuration.

Why Python in Healthcare is Important?

Below are some important reasons for using Python in Healthcare sector, have a look

1. Python and its frameworks work on principles that are grounded equally and agreed upon the HIPAA checklist.
2. A full look at the big data healthcare allows the organization to exchange information for seeking patient outcomes.
3. The performance of the platforms made with Python is focused on their availability in phones and the web.

How Medical Startups are using Python?

- Roam Analysis

Uses machine learning (artificial intelligence) and comprehensive contextual data to take over biopharmaceutical and medical device companies, that need to make decisions, suggest treatments and get the best possible patient outcomes.

Roam's platform is powered by a proprietary data asset called Health Knowledge Graph which is continuously enriched using natural language processing to gather information and make connections with new data.

According to the official description, -Roam's machine learning and data platform powers rich analysis of patient journeys to reveal the factors affecting treatment decisions and outcomes.||

- AiCure

It is an NIH and VC-funded healthcare startup in New York. AiCure automates the process of ensuring that patients are taking their medicines, that too at their assigned time. It has combined artificial intelligence and mobile technology together.

For example, it uses computer vision to identify the patient (face recognition) and verify the right pill for the specific patient (using pill recognition) and they are actually consuming them (action recognition). They develop their backend applications using Django frameworks and use python base coding and research engineers to develop it.

Amazed by the magic of Python in Healthcare? Here are some more startups –

- Drchrono

It is a software as a service patient care platform that consists of a web and cloud-based app. This American company is for doctors and patients that make electronic health records available digitally and provides management and medical billing services. This is also a phone and web concentrated platform.

- Fathom Health

Again a Healthcare startup with deep learning NLP system for reading and understanding electronic health records. This startup with headquarters in San Francisco, California is backed up by Google. Their employees are familiar with Flask for API programming and data engineers prefer Python's NLTK.

Python Installation:

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platform

python download for windows 10

All Videos Images News Maps More Settings Tools

About 15,10,00,000 results (0.37 seconds)

www.python.org › downloads › windows ▾

Python Releases for Windows | Python.org

Download Windows help file; Download Windows x86-64 embeddable zip file; Download ...
Note that Python 3.6.10 cannot be used on Windows XP or earlier.
Python 3.7.4 · Python 3.7.3 · Python 2.7.16 · Python 3.8.0b4
You've visited this page 2 times. Last visit: 12/3/20

www.python.org › downloads ▾

Download Python | Python.org

18, 2019 Download Release Notes · Python 3.6.10 Dec. ... The same source code archive can also be used to build the Windows and Mac versions, and is the ...
Windows · Python 3.7.4 · Python 2.7.10 · Python 3.7.2

Here the above screen is for searching python download in Google.

Python PSF Docs PyPI

Donate Search

About Downloads Documentation Community Success Stories News

Python »» Downloads »» Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.8.2
- Latest Python 2 Release - Python 2.7.17

Stable Releases

- Python 3.7.7 - March 10, 2020

Note that Python 3.7.7 cannot be used on Windows X

Docs	Python's standard documentation: download or browse or watch a tutorial.	
Audio/Visual Talks	Get started below, or visit the Documentation page to browse by version.	
Beginner's Guide	Python 3.x Docs	Python 2.x Docs
Developer's Guide		
FAQ		
Non-English Docs		
PEP Index		
Python Books		
Python Essays		

After clicking python.org we get this window.

Python Releases for Windows

- [Latest Python 3 Release - Python 3.8.2](#)
- [Latest Python 2 Release - Python 2.7.17](#)

Stable Releases

- [Python 3.7.7 - March 10, 2020](#)

Note that Python 3.7.7 cannot be used on Windows XP or earlier.

- [Download Windows help file](#)
- [Download Windows x86-64 embeddable zip file](#)
- [Download Windows x86-64 executable installer](#)
- [Download Windows x86-64 web-based installer](#)
- [Download Windows x86 embeddable zip file](#)
- [Download Windows x86 executable installer](#)
- [Download Windows x86 web-based installer](#)

- [Python 3.8.2 - Feb. 24, 2020](#)

Note that Python 3.8.2 cannot be used on Windows XP or earlier.

After clicking downloads we get python releases for windows.

- [Python 2.7.15 - May 1, 2018](#)

- [Download Windows debug information files](#)
- [Download Windows debug information files for 64-bit binaries](#)
- [Download Windows help file](#)
- [Download Windows x86-64 MSI installer](#)
- [Download Windows x86 MSI installer](#)

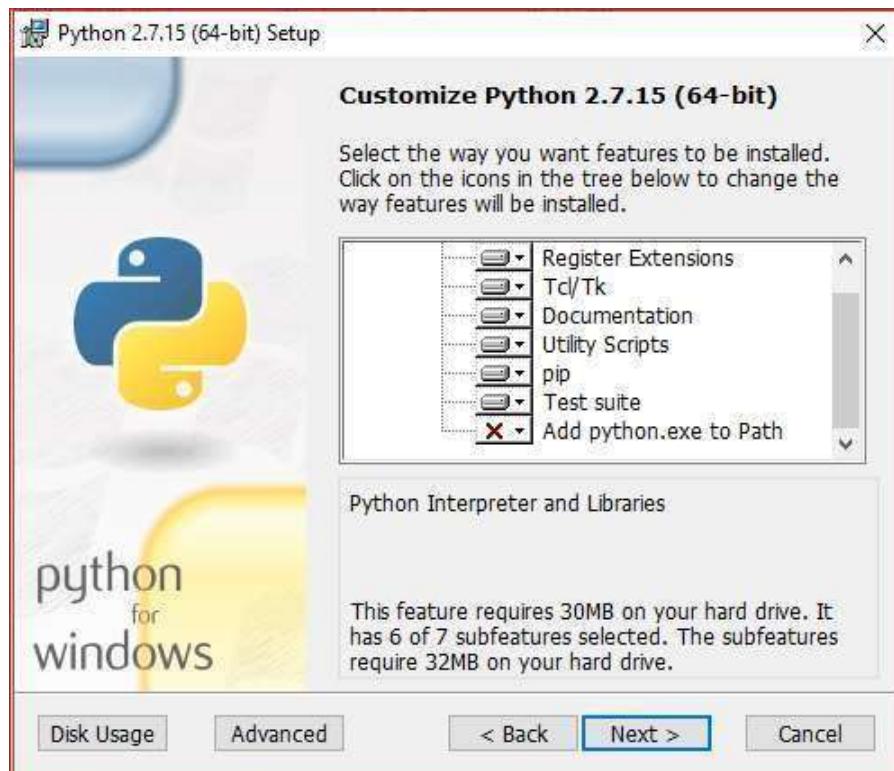
For downloading python 2.7.15, we have to download Windows x86-64 MSI installer.



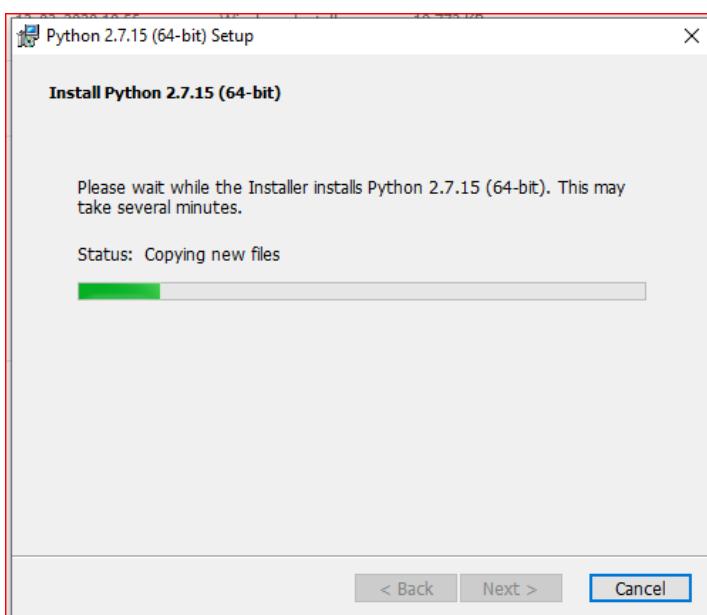
After downloading we have to install python 2.7.15.



Selecting destination directory.



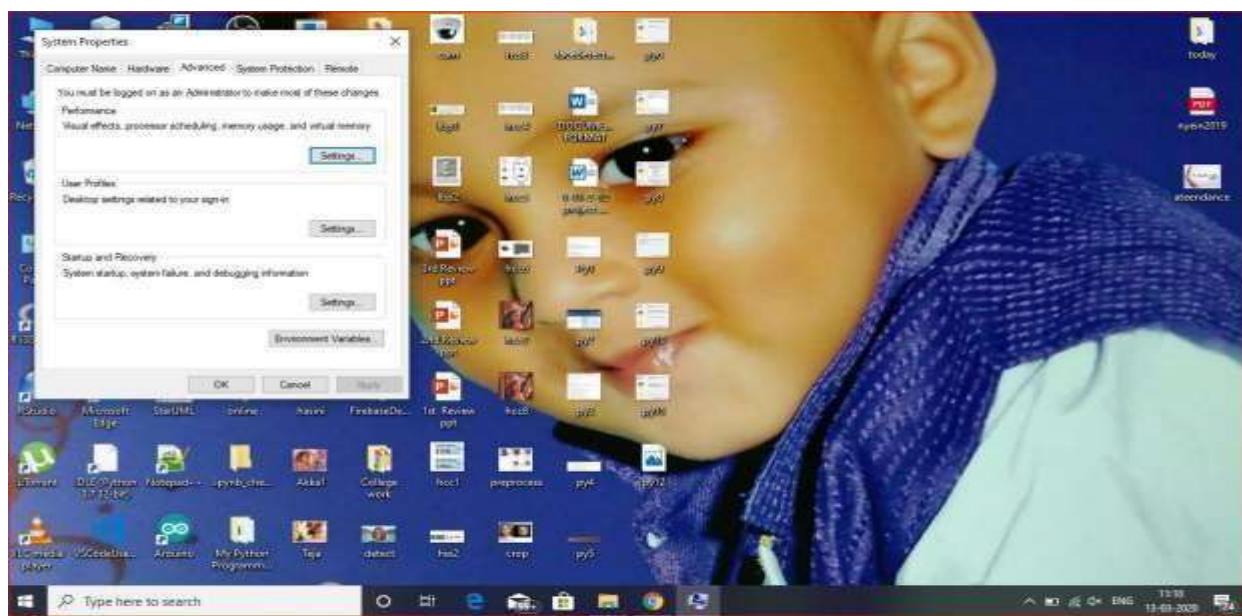
Select what features to be installed.

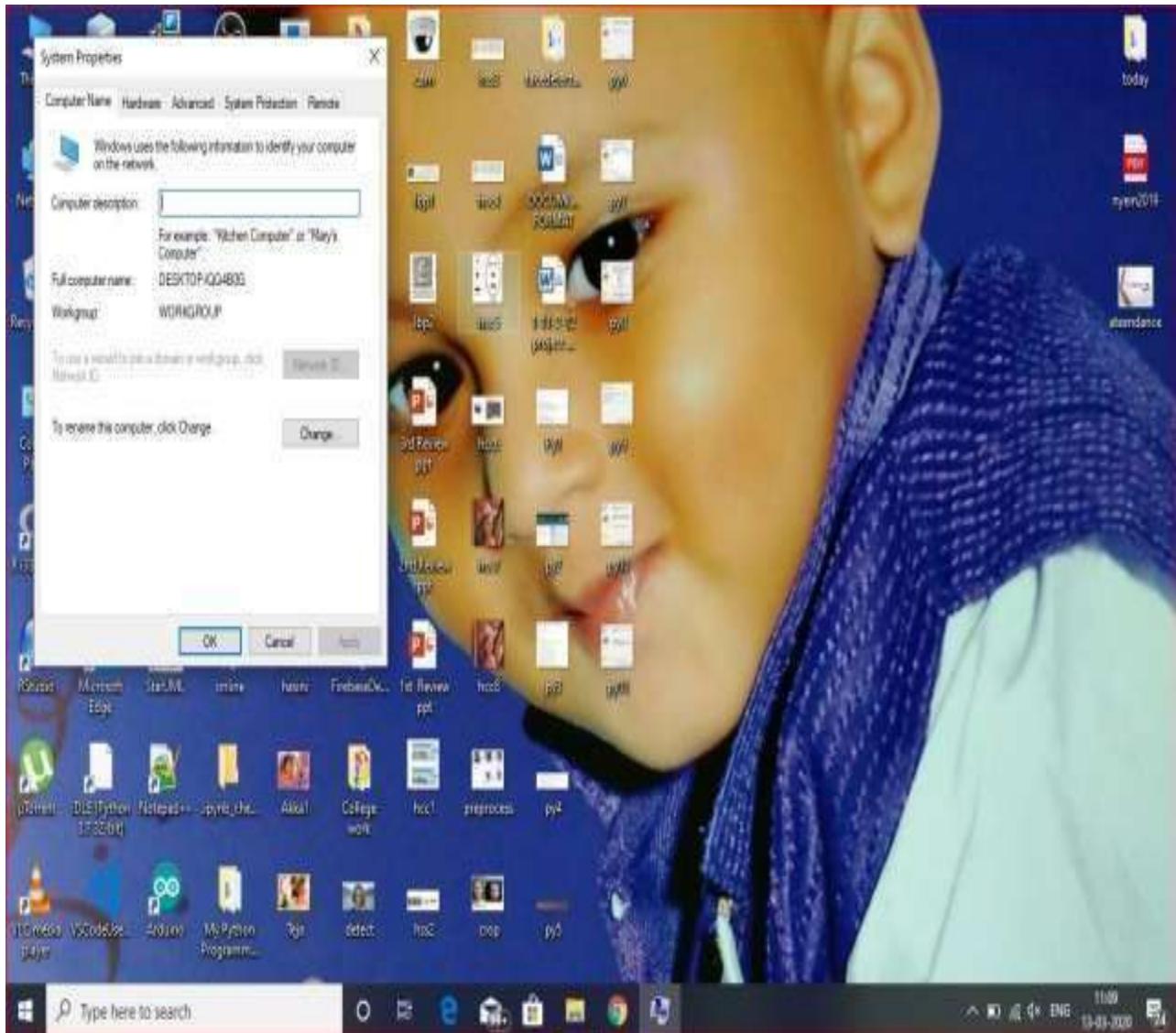


Installing python 2.7.15



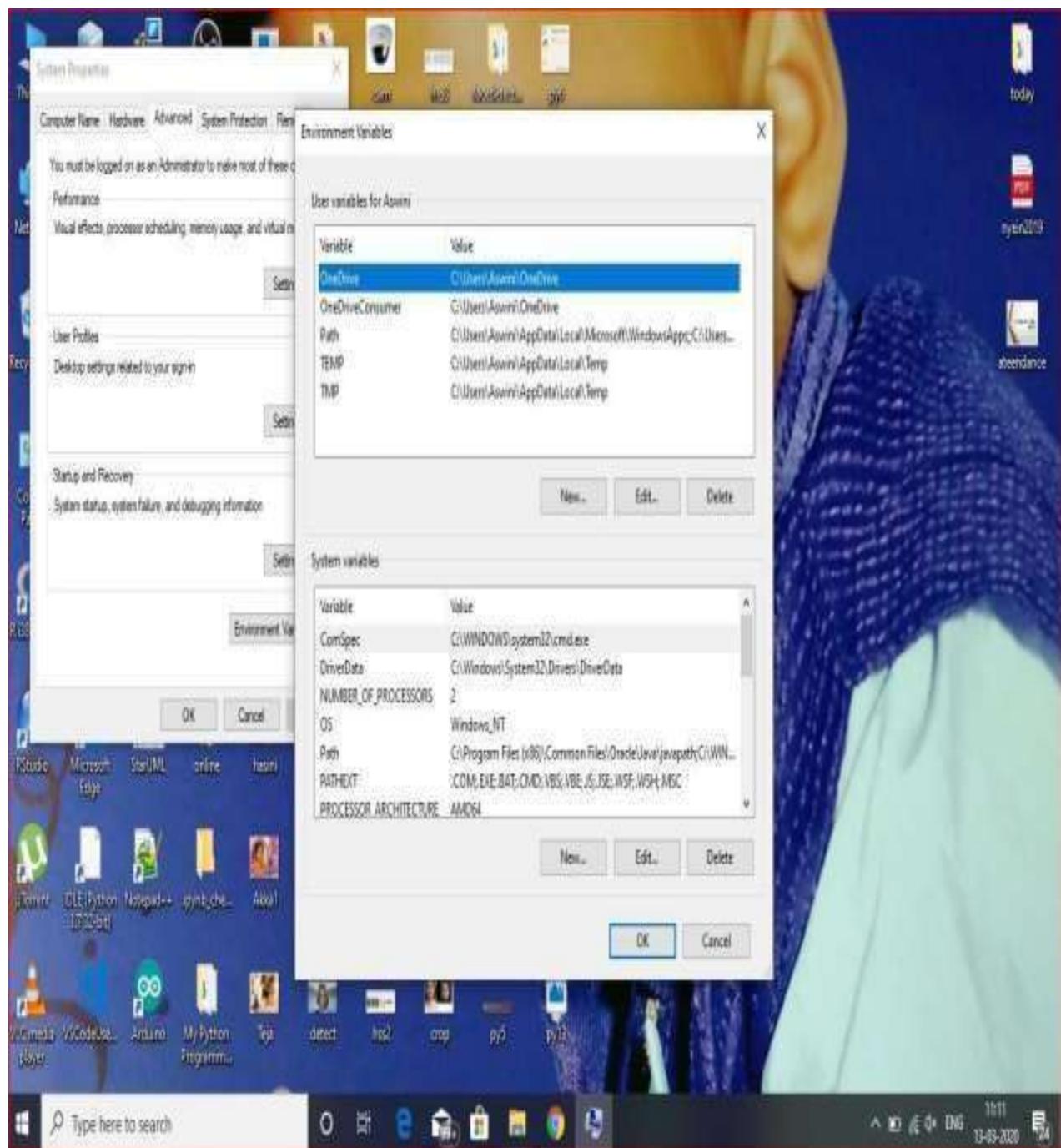
After installing click on finish



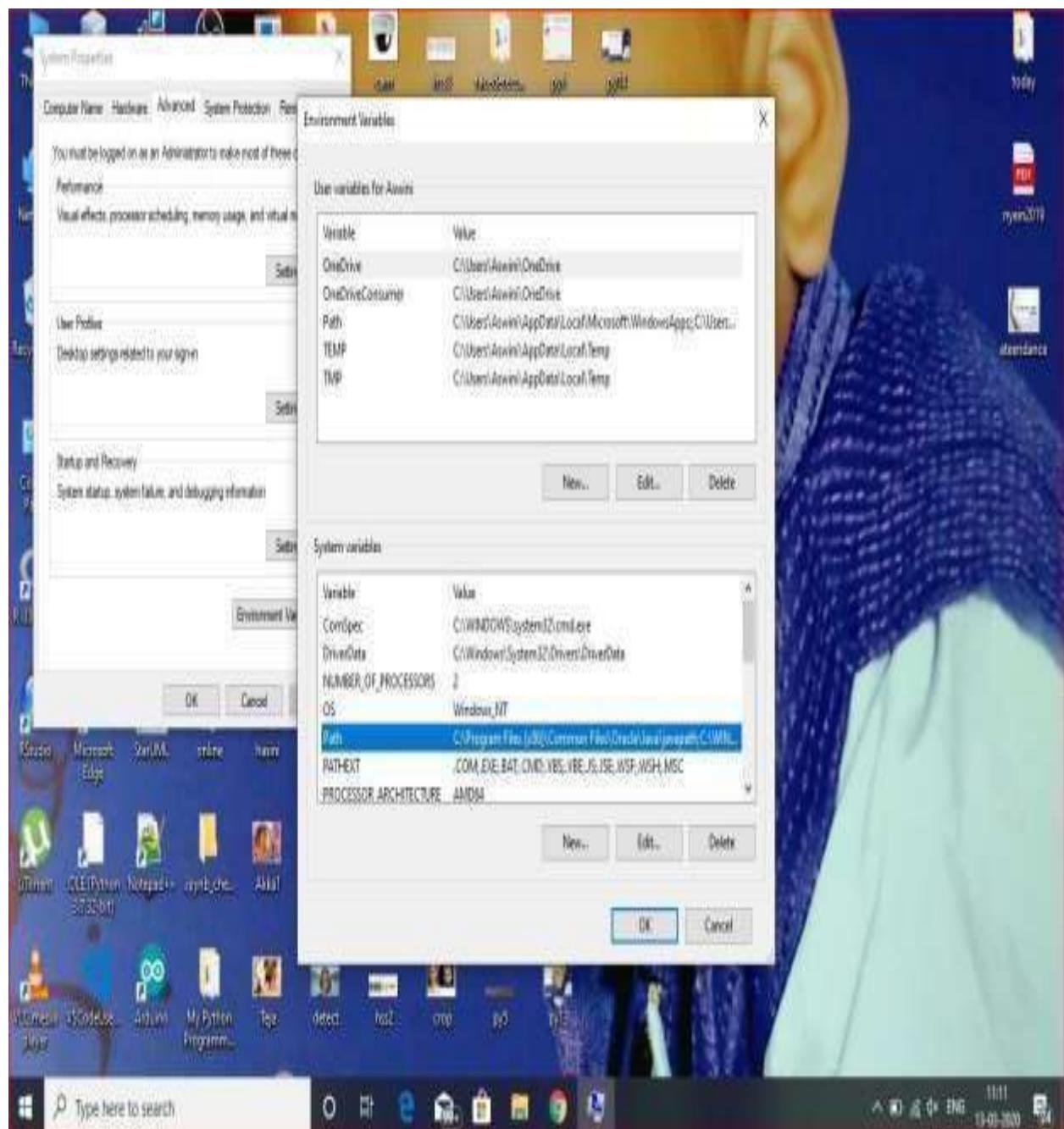


Here after installing the Python we have to set the specific path of it for the further process. For setting the path we go to the advanced settings in the system by clicking windows+r. Here we have the ability to set the path.

We have clicked on the Advanced to set the path for the further process. Here in the advances the settings can be changed to set the path of the specific version that is been installed and been using in our systems.



In the Advances settings we click on the Environmental Variables we get the System Variables. Where the path is allowed to set. In the System Variables we have to click on edit to set the path of the specific system that is been using in our systems.



Here in the System variables we can see the path data where it has to be edited. We have to click on the path settings to edit the path and further click on OK to continue.

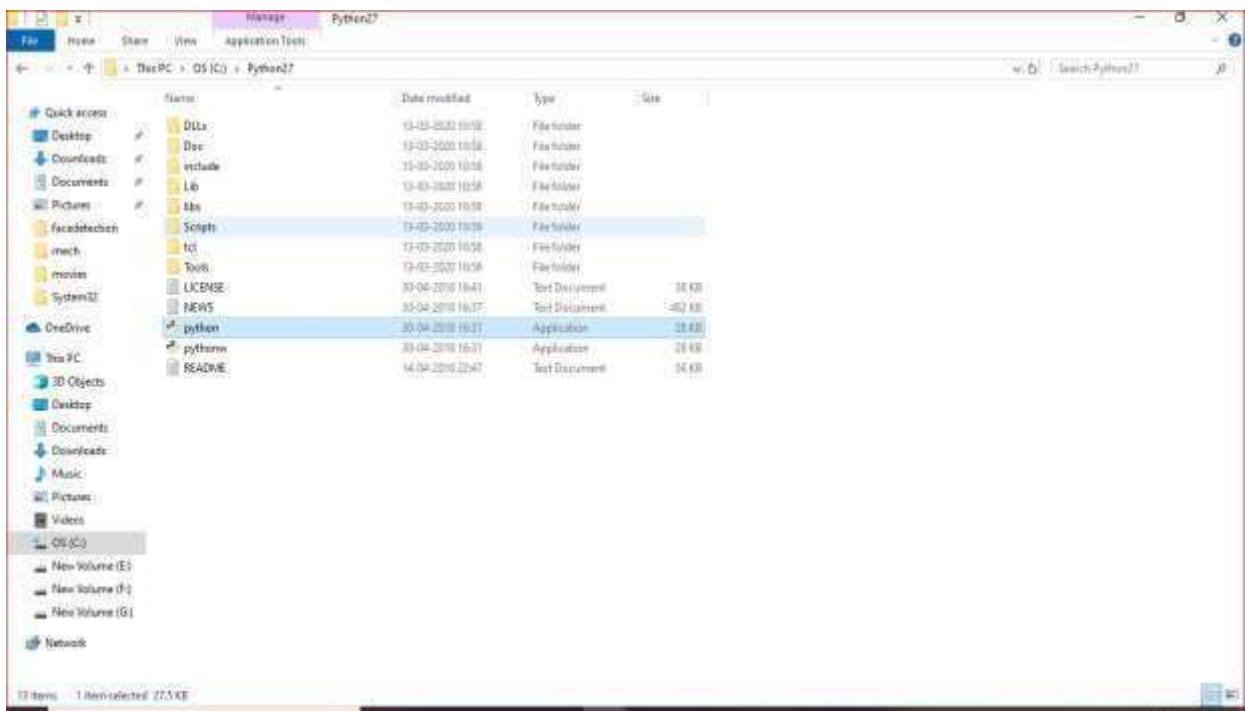


Fig: Selecting the path of the Python

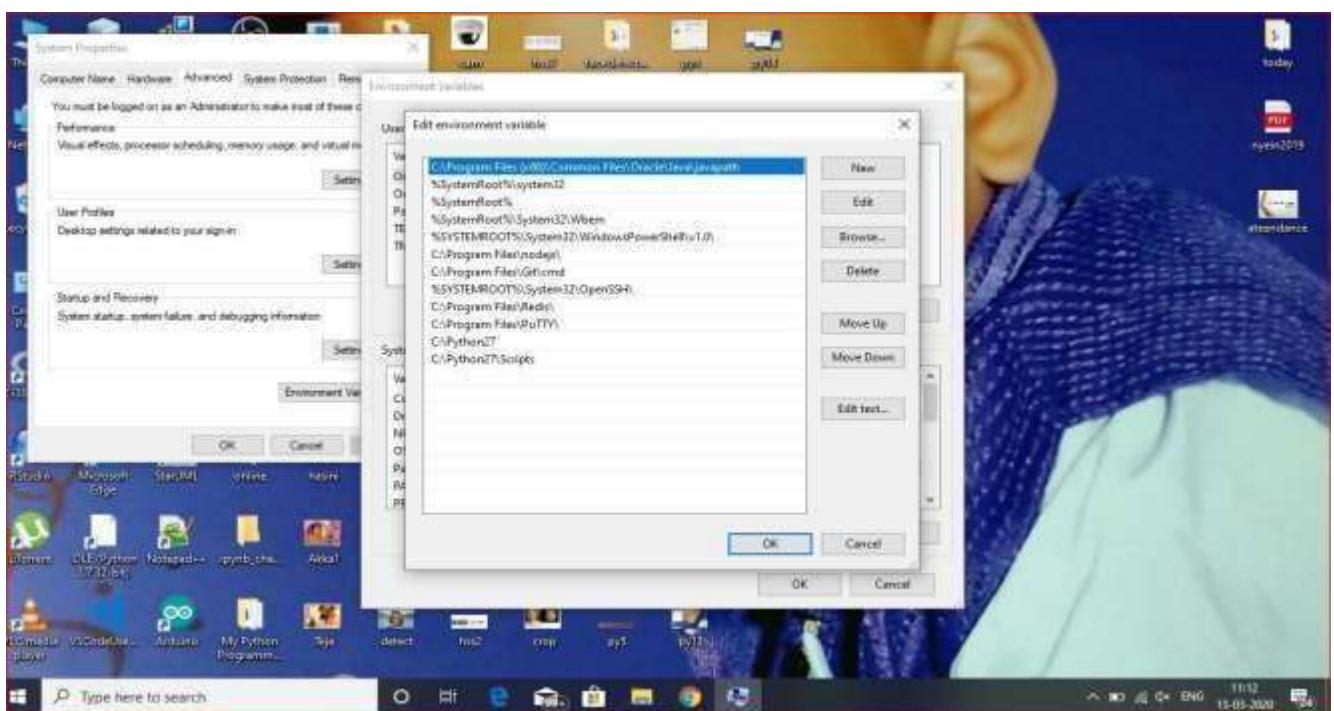


Fig: Entering the path of the Python

Here in this screen we can see the edit option where the path is to be set. We have to click on the NEW to set the path that is copied and should be pasted in the new row. The path of the Python is been copied in the new row.

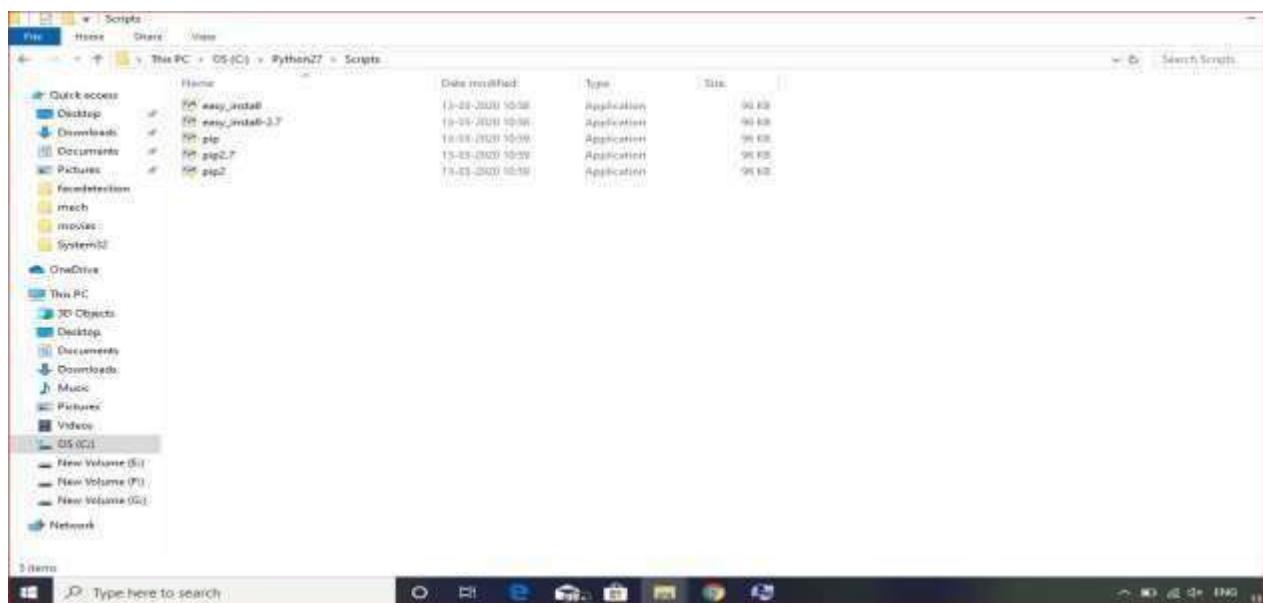


Fig: Selecting the path of the Script

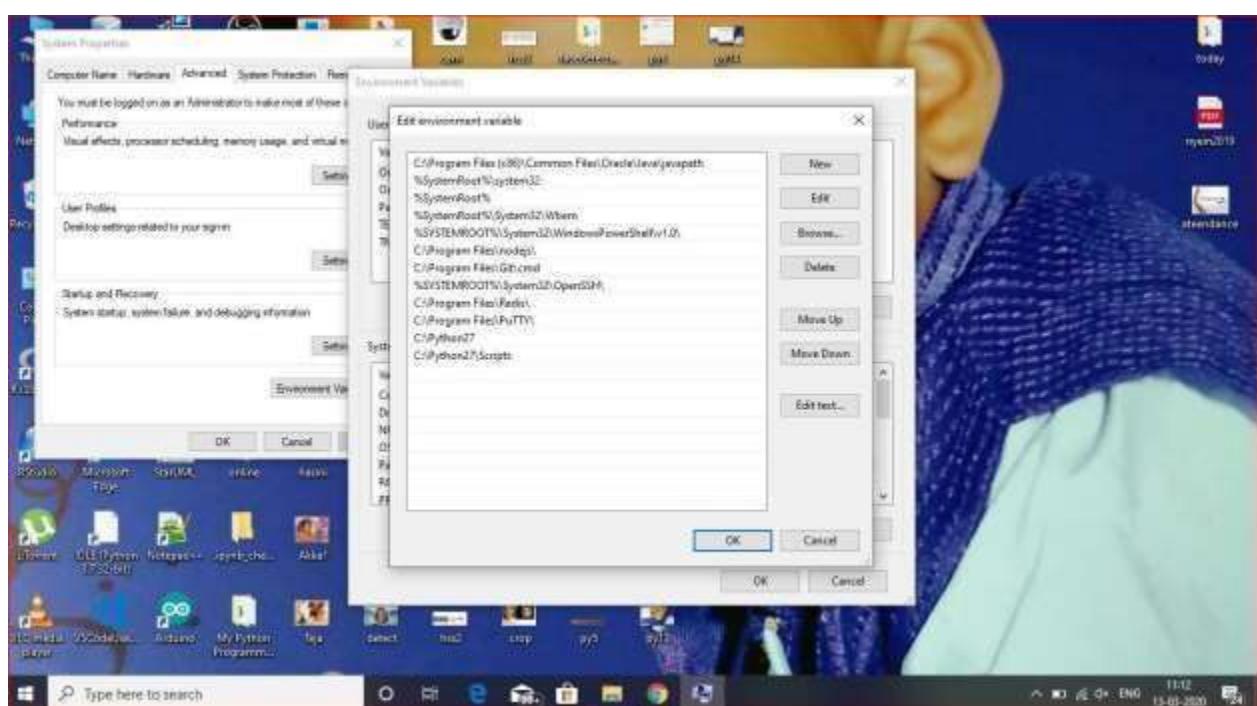


Fig: Entering the path of the Script

Here in this screen we can see the edit option where the path is to be set. We have to click on the NEW to set the path that is copied and should be pasted in the new row. The path of the Script is been copied in the new row.

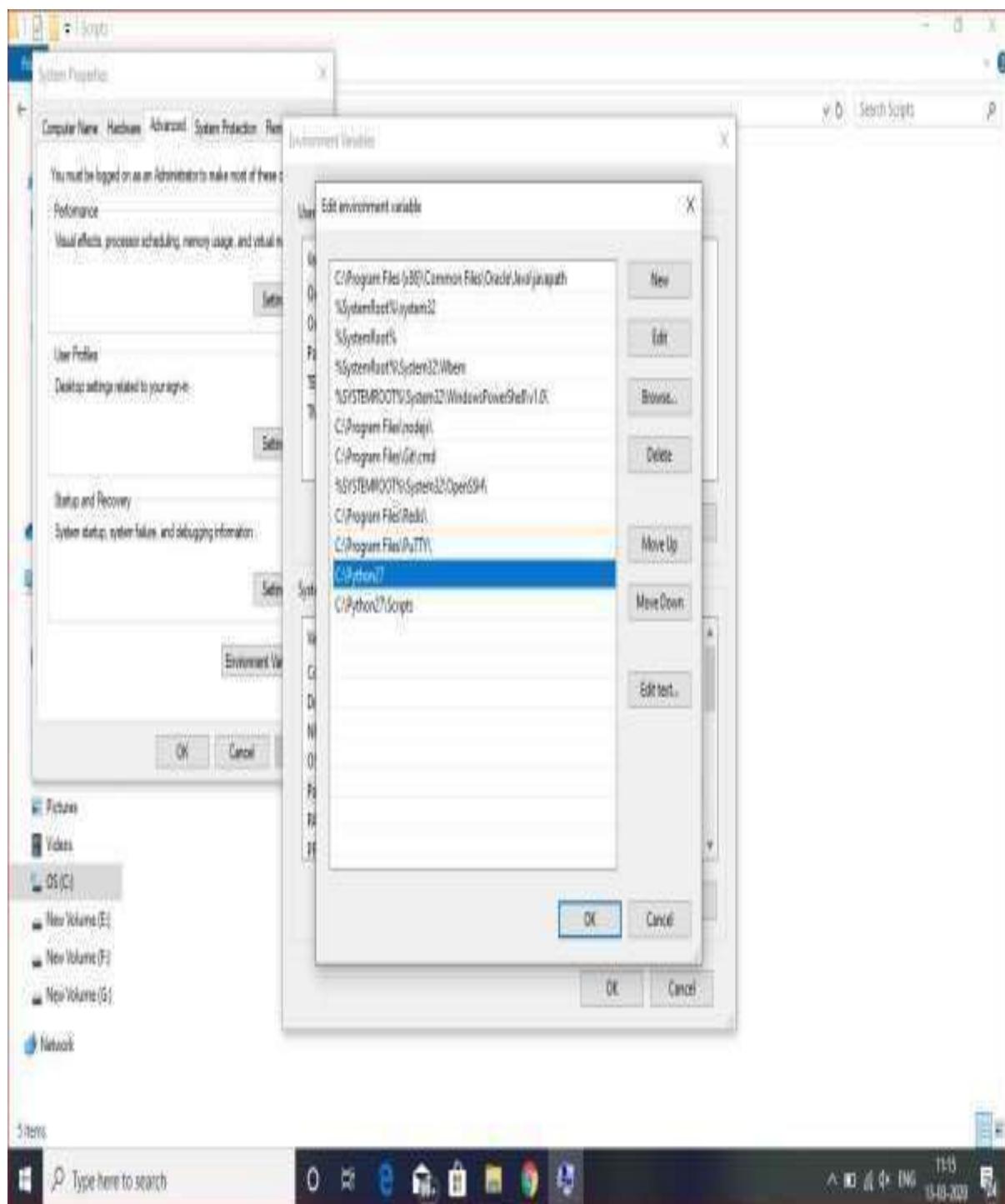


Fig: The path of the Python that is been pasted successfully

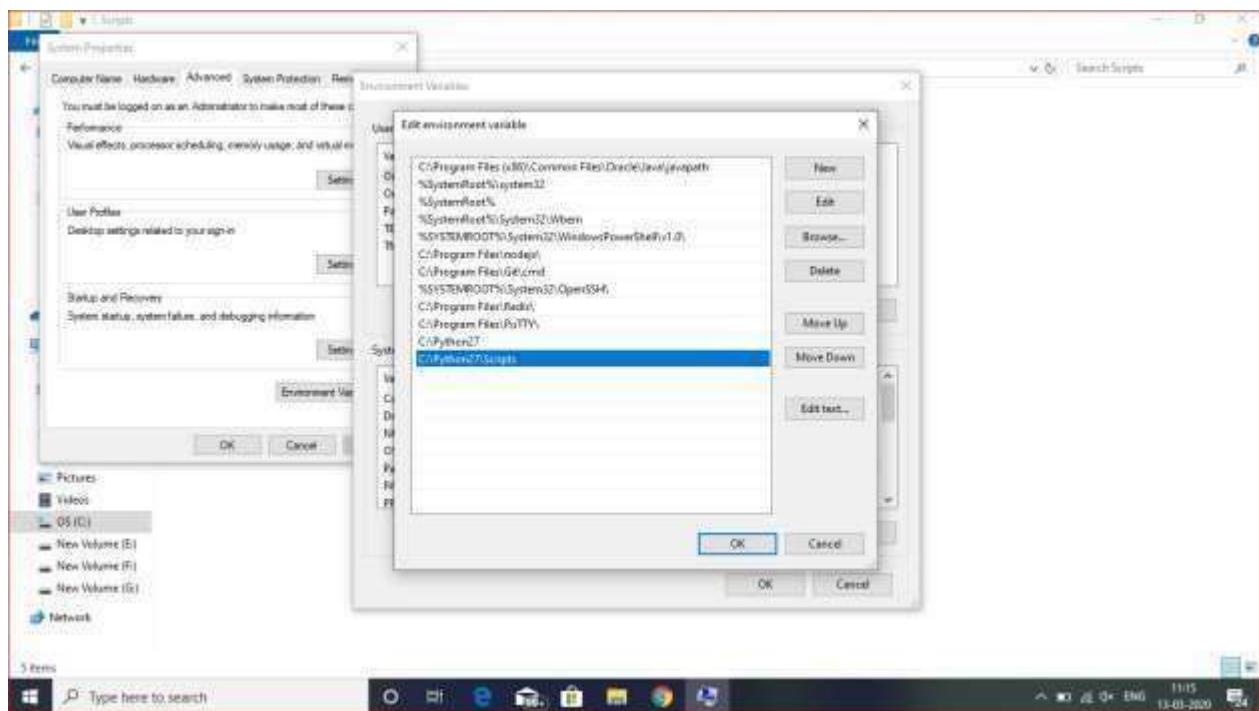


Fig: The path of the Script that is been pasted successfully

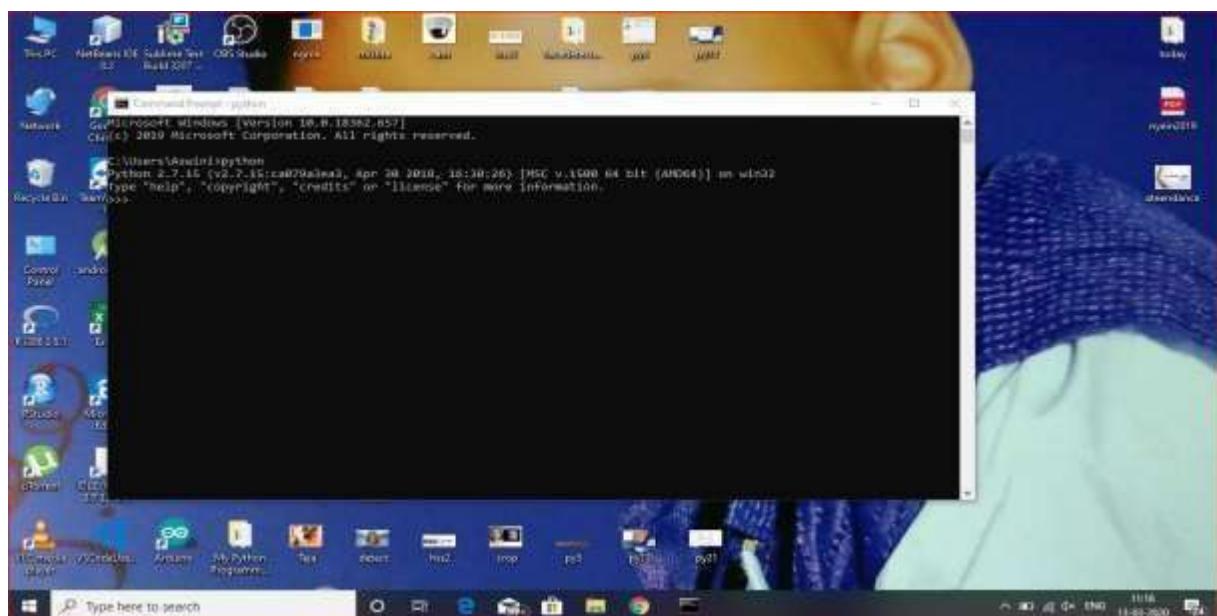


Fig: The version of the Python shown in the Command Prompt

Here the version of the Python is been shown in the command prompt by entering the word Python in the cmd. It further shows the version that is been installed and used in the systems for our project.

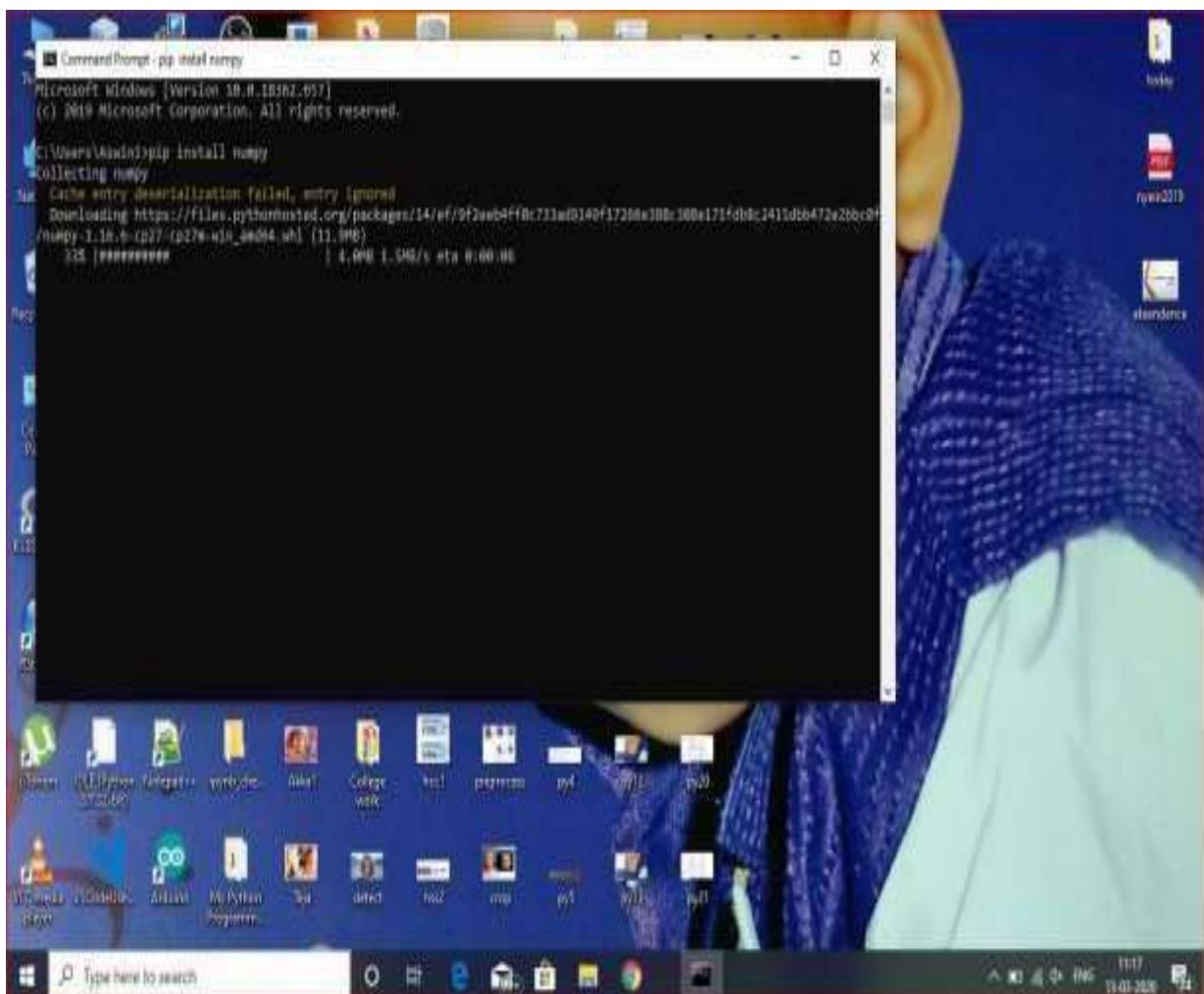


Fig: The library file of numpy is been installing

For installing the library file named numpy we have to type -`pip install numpy`|| and press enter. It further installs all the library files and functions that are needed to run the file.

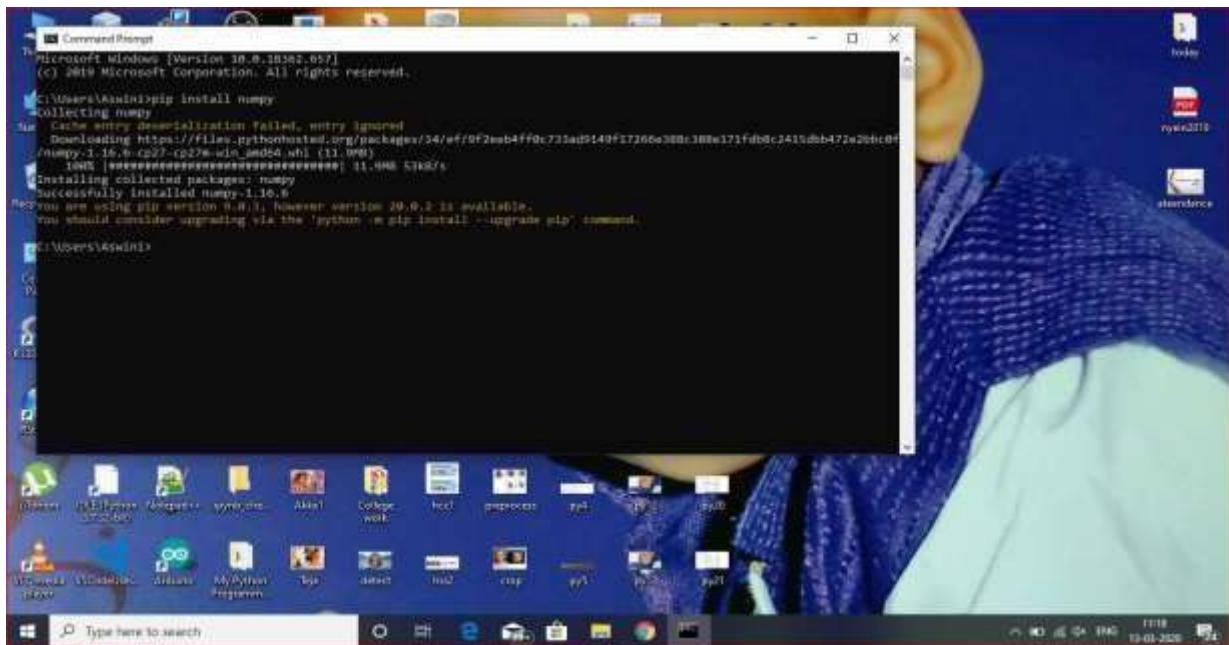


Fig: The successful installation of numpy

Here we can see the successful installation of the numpy library files that are been installed by using the command -`pip install numpy`|| in the command prompt.

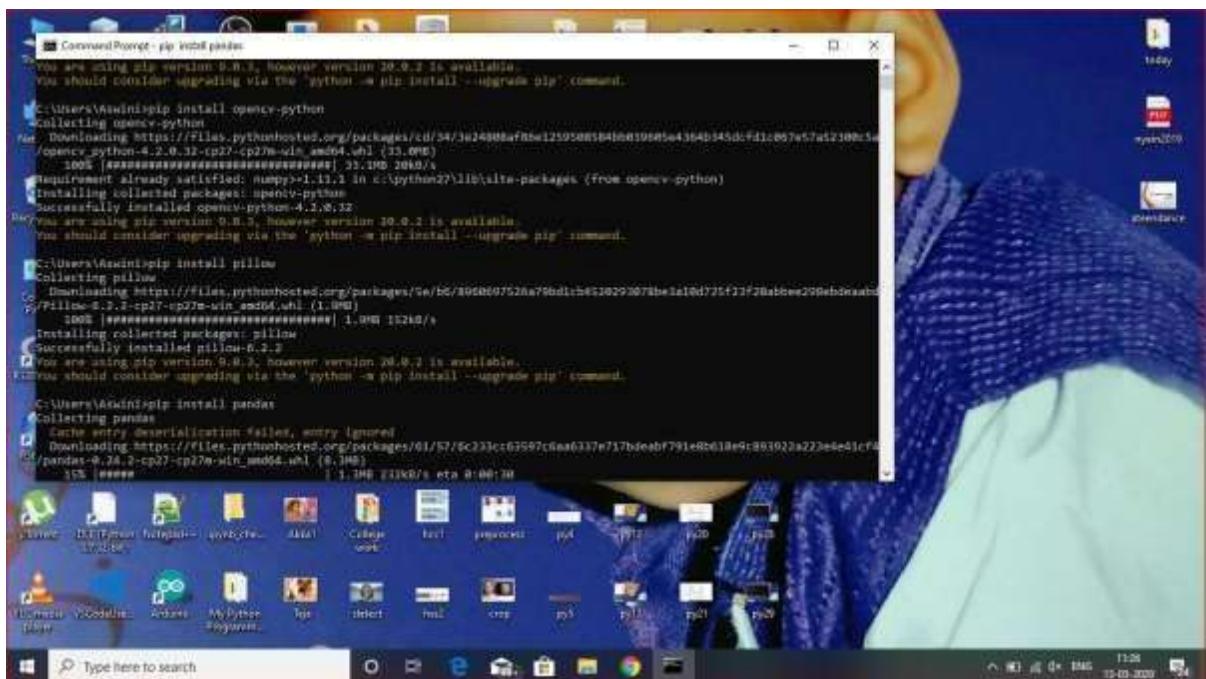


Fig: The file of pandas is been installing

Here the library files of the pandas are needed to be installed by writing the command as -`pip install pandas` in the command prompt where it installs all the related library files of the pandas for the project that are needed and for the further process.

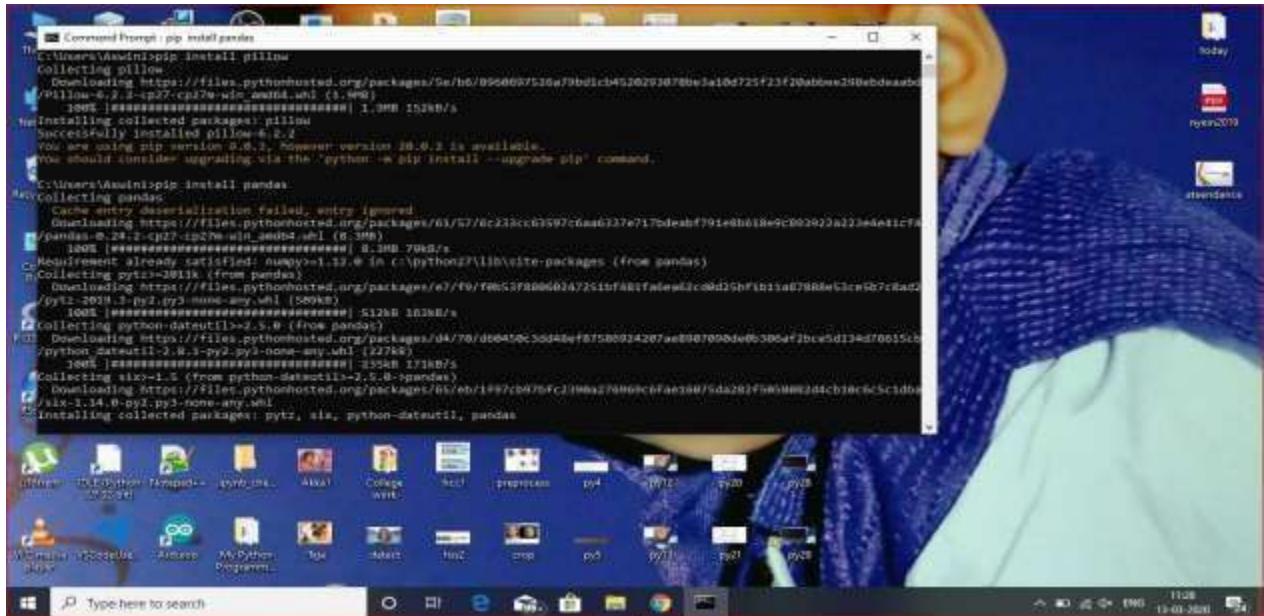


Fig: Installation of the related library files of pandas

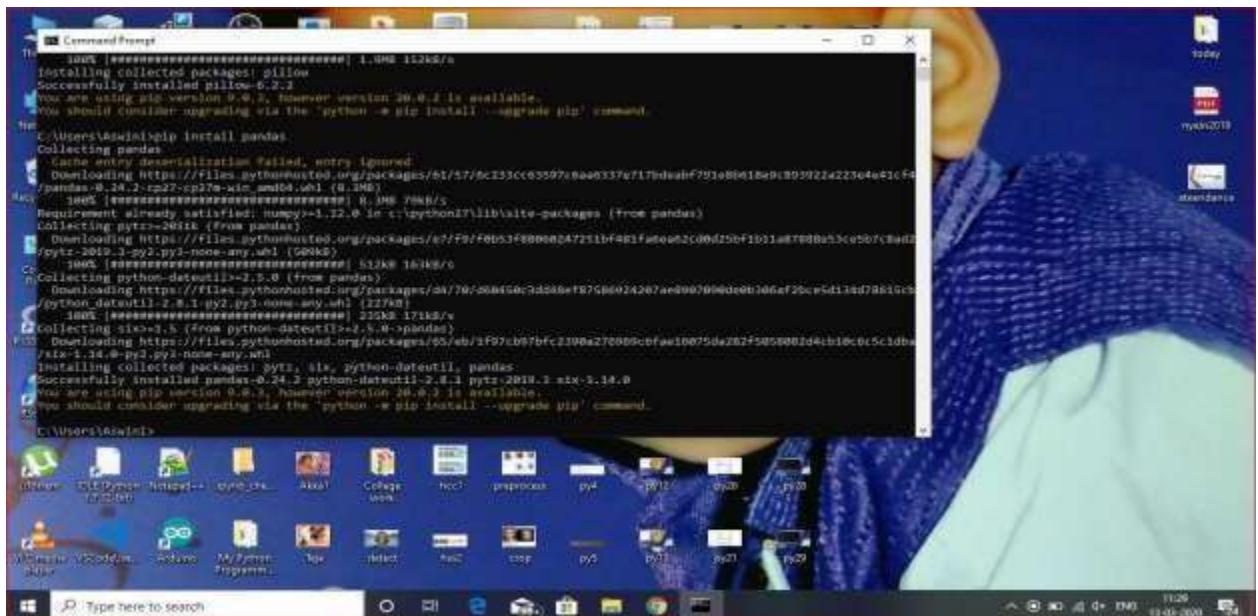


Fig: Successful installation of the pandas

Pandas in python

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice. Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

KeyFeatures of Pandas

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, pip

Pip install pandas

If you install Anaconda Python package, Pandas will be installed by default with the following –

Windows

- Anaconda (from <https://www.continuum.io>) is a free Python distribution for SciPy stack. It is also available for Linux and Mac.
- Canopy (<https://www.enthought.com/products/canopy/>) is available as free as well as commercial distribution with full SciPy stack for Windows, Linux and Mac.
- Python (x,y) is a free Python distribution with SciPy stack and Spyder IDE for Windows OS. (Downloadable from <http://python-xy.github.io/>)

Linux

Package managers of respective Linux distributions are used to install one or more packages in SciPy stack.

For Ubuntu Users

```
sudo apt-get install python-numpy python-scipy python-matplotlib python-ipython python-notebook  
python-pandas python-sympy python-nose
```

For Fedora Users

```
sudo yum install numpy scipy python-matplotlib python python-pandas sympy  
python-nose atlas-devel
```

pandas deals with the following three data structures –

- Series
- DataFrame
- Panel

These data structures are built on top of Numpy array, which means they are fast.

Dimension&Description

The best way to think of these data structures is that the higher dimensional data structure is a container of its lower dimensional data structure. For example, DataFrame is a container of Series, Panel is a container of DataFrame.

Data Structure	Dimensions	Description
Series	1	1D labeled homogeneous array, size immutable.
Data Frames	2	General 2D labeled, size-mutable tabular structure with potentially heterogeneously typed columns.
Panel	3	General 3D labeled, size-mutable array.

Building and handling two or more dimensional arrays is a tedious task, burden is placed on the user to consider the orientation of the data set when writing functions. But using Pandas data structures, the mental effort of the user is reduced.

For example, with tabular data (DataFrame) it is more semantically helpful to think of the index (the rows) and the columns rather than axis 0 and axis 1.

Mutability

All Pandas data structures are value mutable (can be changed) and except Series all are size mutable. Series is size immutable.

Note – DataFrame is widely used and one of the most important data structures. Panel is used much less.

Series

Series is a one-dimensional array like structure with homogeneous data. For example, the following series is a collection of integers 10, 23, 56, ...

10	23	56	17	52	61	73	90	26	72
----	----	----	----	----	----	----	----	----	----

Key Points

- Homogeneous data
- Size Immutable
- Values of Data Mutable

DataFrame

DataFrame is a two-dimensional array with heterogeneous data. For example,

Name	Age	Gender	Rating
Steve	32	Male	3.45
Lia	28	Female	4.6
Vin	45	Male	3.9
Katie	38	Female	2.78

The table represents the data of a sales team of an organization with their overall performance rating. The data is represented in rows and columns. Each column represents an attribute and each row represents a person.

Data Type of Columns

The data types of the four columns are as follows –

Column	Type
Name	String
Age	Integer
Gender	String
Rating	Float

Key Points

- Heterogeneous data
- Size Mutable
- Data Mutable

Panel

Panel is a three-dimensional data structure with heterogeneous data. It is hard to represent the panel in graphical representation. But a panel can be illustrated as a container of DataFrame.

Key Points

- Heterogeneous data
- Size Mutable
- Data Mutable

Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index.

pandas.Series

A pandas Series can be created using the following constructor –

`pandas.Series(data, index, dtype, copy)`

The parameters of the constructor are as follows –

S.No	Parameter & Description
1	Data data takes various forms like ndarray, list, constants
2	Index Index values must be unique and hashable, same length as data. Default np.arange(n) if no index is passed.
3	Dtype dtype is for data type. If None, data type will be inferred
4	Copy Copy data. Default False

A series can be created using various inputs like –

- Array
- Dict
- Scalar value or constant

FLASK:

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Why is Flask a good web framework choice?

Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

HTML:

HTML stands for Hypertext Markup Language. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications. HTML is not a programming language, meaning it doesn't have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word. When working with HTML, we use simple code structures (tags and attributes) to mark up a website page. For example, we can create a paragraph by placing the enclosed text within a starting `<p>` and closing `</p>` tag.

The History of HTML

HTML was invented by Tim Berners-Lee, a physicist at the CERN research institute in Switzerland. He came up with the idea of an Internet-based hypertext system. Hypertext means a text that contains references (links) to other texts that viewers can access immediately. He published the first version of HTML in 1991, consisting of 18 HTML tags. Since then, each new version of the HTML language came with new tags and attributes (tag modifiers) to the markup. According to Mozilla Developer Network's HTML Element Reference, currently, there are 140 HTML tags, although some of them are already obsolete (not supported by modern browsers). Due to a quick rise in popularity, HTML is now considered an official web standard. The HTML specifications are maintained and developed by the World Wide Web Consortium (W3C). You can check out the latest state of the language anytime on W3C's website. The biggest upgrade of the language was the introduction of HTML5 in 2014. It added several new semantic tags to the markup, that reveal the meaning of their own content, such as <article>, <header>, and <footer>.

How Does HTML Work?

HTML documents are files that end with a .html or .htm extension. You can view them using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it.

Usually, the average website includes several different HTML pages. For instance: home pages, about pages, contact pages would all have separate HTML documents.

Each HTML page consists of a set of tags (also called elements), which you can refer to as the building blocks of web pages. They create a hierarchy that structures the content into sections, paragraphs, headings, and other content blocks.

HTML:HTML stands for Hyper Text Markup Language. HTML describes the structure of web pages using mark up .

<!DOCTYPE html> :- Basically tells the browser what kind of document we're reloading

<a> :- Anchor also known as hyperlink or simply a link.

**** :-Forbolding or otherwise styling text without conveying any additional meaningsuch as with **** and **** etc.,

<base> :-Use to set a base URL.

<body> :-Establishes the body of an HTML document.

**
** :-Single line break.

<button>:-Specifies button that can be clicked. Commonly used in the forms.

<caption>:-Used for table caption.

<div> :- A divider or a generic contain.

<field set>:-Group related items in a form.

<form> :-Forms for User input.

<h1> to <h6> :-Level 1 to level 6 headings. With various sizes.

<head>:- Used mostly for metadata.

****:-For displaying images on a webpage.

<input>:- Used with various attributes to create form input elements.

<label>:-For assigning labels to interface elements such as forms.

<link>:- Used for linking to external resources.

****:- This represents a list item with an **** or ****.
<meta>:-Used for specifying various meta information about the document.

<nav>:- Used for specifying a navigational region within a document.

<option>:- Used within the **<select>** tag to specify an option.

<p>:- Paragraph. Enclose paragraphs in an article within **<p>** tags.

<script>:-Scripts inside of an HTML document are enclosed in the script tags.

<section>:-A nonspecific means of breaking up content within a webpage intosections.

<select>:- Used with <option> for selecting a particular option.

:- Useful for applying styles to or around text, especially inline.

<style>:- For declaring style sheets within document.

<table>:- For making tables in a HTML document.

<td>:- Table data. Represents a data cell within a table.

<th> :- Table header cell.

<title>:- The title of an HTML document.

<tr> :- A row within a table.

<textarea>:- For multiline text entry.

<u>:- Useful for annotating text for various reasons. no longer specifically for underlining.

:- Unordered list. Use with to make unordered list.

CASCADING STYLE SHEET

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS:-CSS stands for Cascading Style Sheet.CSS describes how Html elements are to be displayed on screen ,paper ,or in other media.

Font Properties :-

Font-Family :-Changes the font family of certain words, sentences, paragraphs, etc.

Font-Style :- Changes text: normal, oblique, and italics.

Font-Weight :-Used to specify the weight of the font.

Font-Size :-Used to modify the size of the displayed font.

Font:- Used to combine all properties of fonts.

Text Properties :-

Word-Spacing :-Defines an additional amount of space between words.

Letter-Spacing :-Defines an additional amount of space between characters.

Text-Decoration :-Allows text to be decorated through one of five properties: underline, overline, line-through, blink, none.

Vertical-Align :-Used to alter the vertical positioning of an inline element, relative to its parent element or to the element's line.

Text-Transform :-Allows for capitalizing the first letter of each word (capitalize), capitalizing all letters of a word(uppercase), using all small letters in each word(lowercase), and the initial value(None).

Text-Align :-Used to justify text left, centre, right, and justify.

Text-Indent :-Used to specify the amount of indentation prior to the first line of text.

Line-Height :-Used to control the spacing between baselines of text.

Color and Background Properties :-

Color :-Changes the color of text.

Background-Color :-Sets the background color of an element.

Background-Image :-Sets the background image of an element.

Background-Repeat :-Determines how a specified background image is repeated. The repeat-x value will repeat the image horizontally while the repeat-y value will repeat the image vertically.

Background-Attachment :-Determines if a specified background image will scroll with the content or be fixed with regard to the canvas.

Background :-Used to combine all properties of background.

Box Properties :-

Margin-Top :-Sets the top margin of an element by specifying a length or a percentage.

Margin-Right :-Sets the right margin of an element by specifying a length or a percentage.

Margin-Bottom :-sets the bottom margin of an element by specifying a length or a percentage.

Margin-Left :-sets the left margin of an element by specifying a length or a percentage.

Margin :-Sets the margins of an element by specifying top, bottom, left and right margins -- all either specifying length or percentage.

Padding-Top :-Describes the amount of space between the top border and the content of the selector.

Padding-Right :-Describes the amount of space between the right border and the content of the selector.

Padding-Bottom :-Describes the amount of space between the bottom border and the content of the selector.

padding-Left :-Describes the amount of space between the left border and the content of the selector.

Padding :-Shorthand for the padding-top, padding-right, padding-bottom, and padding-left properties.

Border-Width :-Used to set the width of an element's border (either all borders, or specifying top border, right border, bottom border, left border).

Border :-Sets the width, style, and color of an element's border.

Border-Radius:-Sets all the four border. Radius properties for rounded corners.

Width :-Each block-level or replaced element can be given a width,specified as a length, a percentage, or as auto.

Height :-Each block-level or replaced element can be given a height, specified as a length or as auto.

Float :-Allows text to wrap around an element (left, right, none).

Clear :-Specifies whether an element allows floating elements to its sides (left, right, none).

Outline:- An outline is a line that is drawn around elements.

Transition:-CSS transitions allow you to change property values smoothly ,over a given duration.

BOOTSTRAP :-Bootstrap is the most popular HTML, CSS, and javascript framework for faster and easier web development.

.btn-outline:- A button variation to have outlined buttons instead of a solid background.

.col-* :- This class is used for grid columns to determine the column width and the breakpoint you would like it to be active. The classes work from the breakpoint you set and everything larger.

.d-print-* :- Changes the display of elements when you print the document.

.fixed-* :- This class makes an element fixed to the top/bottom of the browser window. Here is what the CSS ruleset looks like.

.form-control-plaintext :- Use the class to remove the default form field styling and preserve the correct margin and padding.

.form-row :- Works similar to a grid. but is more compact to make the form look more uniform.

.nav-fill :- Makes all nav items use all available horizontal space. Nav items are different widths based on their content

.nav-justified :- Makes all nav items equal width and use all available horizontal space.

.navbar-collapse :- The nav links that are collapsed and shown when toggled on mobile widths.

.navbar-expand-* :- Since the navbar is displayed collapse on mobile first, this class specifies what breakpoint you want the navbar to not be collapsed

.btn-group-lg :- Increases the default button group size

.btn-group-sm :- decreases the default button group size

.btn-outline-* :- Transparent background with colored text and border
(danger|info|primary|secondary|success|warning)

.font-* :- (italic, weight-bold, weight-light, weight-normal, monospace)

.form-inline :- Use this class to have a series of labels and form elements on a single horizontal row

.form-text :- This class is used for help text alongside form elements. You can add .text-muted to make the text lighter in color

.input-group-text :- This class adds the background color and text styles to the text inside an input group

.list-inline :- Change ul or ol list to be listed horizontally with a little margin between each li

.media-left :- Add this class the div wrapping the media object image to align it to the left

.nav-item :- If your nav uses a list add this class to each list item for the proper spacing

.nav-link :- Each anchor link inside your nav is given this class in order to have the proper styling

.navbar-light :- Add this class to your navbar if you would like it to have a light background and dark text

.position-* :- Not responsive, but a group of utility classes to add common position values.
.position-(absolute, fixed, relative, static, sticky)

.container :- Fixed width container with widths determined by screen sites. Equal margin on the left and right

.container-fluid :- Spans the full width of the screen.

.media :- Media components are image heading and description text items. Blog comments, portfolio projects, album covers, etc.

.modal :- The parent wrapper class of modal content

.modal-body :- The modal body content class : Header - Body - Footer

.modal-content :- modal-content contains modal-body, modal-header, and modal-footer

.modal-header :- The header section of the modal that contains the title and close button

.navbar-brand :- Most navbars contain a logo or brand. This class is added to the anchor.

.navbar-nav :- The wrapper class of the navigation elements excluding the brand.

.align:- A set of utility classes that are equivalent to writing the css property. You can use this on inline and table cell elements.

.align-content:- Added to the parent flexbox container to determining how the elements are aligned horizontally.

.bg:- Background color utility classes.

.border:- A versatile border utility class that lets you add/remove borders on a side or change a border color.

.carousel-control:- When you have an image carousel with pagination you will use this class on the previous and next anchor links.

.justify-content:- Class specifies where the flex items will be positioned inside the container.

.navbar:- Navigation header bar.

.navbar-justified:- Makes all nav items equal width and use all available horizontal space.

.navbar-text:- Vertically centres text inside a navbar.

.display:- The set of classes increases the font size of headings in 4 stages. These classes are used for headings outside of the main content of the page like page headers.

.dropdown-item:- This class is added to each link item shown in a dropdown menu.

.form-text:- This class is used for help text alongside form elements. You can add .text-muted to make the text lighter in color.

.container:- Fixed width container with widths determined by screen sites. Equal margin on the left and right.

JAVA Script:

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems. JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server. Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

JAVASCRIPT:- JavaScript is the programming language of HTML and the web.

Event	Description
Onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element

onmouseout The user moves the mouse away from an HTML element

onkeydown The user pushes a keyboard key

onload The browser has finished loading the page

JavaScript String Methods

Methods	Description
charAt()	Returns the character at the specified index (position)
charCodeAt()	Returns the Unicode of the character at the specified index
concat()	Joins two or more strings, and returns a new joined strings
endsWith()	Checks whether a string ends with specified string/characters
includes()	Checks whether a string contains the specified string/characters
match()	Searches a string for a match against a regular expression, and returns the matches.
repeat()	Returns a new string with a specified number of copies of an existing string
replace()	Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced.
search()	Searches a string for a specified value, or regular expression, and returns the position of the match.
toLowerCase()	Converts a string to lowercase letters.
toString()	Returns the value of a String object.
toUpperCase()	Converts a string to uppercase letters.

Array Methods

concat() Joins two or more arrays, and returns a copy of the joined arrays

find() Returns the value of the first element in an array that pass a test

from() Creates an array from an object

includes() Check if an array contains the specified element

isArray() Checks whether an object is an array

join() Joins all elements of an array into a string

keys() Returns a Array Iteration Object, containing the keys of the original array

pop() Removes the last element of an array, and returns that element

push() Adds new elements to the end of an array, and returns the new length

reverse() Reverses the order of the elements in an array

shift() Removes the first element of an array, and returns that element

toString() Converts an array to a string, and returns the result

unshift() Adds new elements to the beginning of an array, and returns the new length

Date functions

Method Description

getFullYear() Get the year as a four digit number (yyyy)

getMonth() Get the month as a number (0-11)

getDate() Get the day as a number (1-31)

PHP:- PHP is a server scripting language, and a powerful tool for making dynamic and interactive web pages.

include():-It is used to insert the content of one PHP file into another PHP file. If _____ will only produce a warning(E_WARNING) and the script will continue.

Session _start():-A session is started with the session _start() function.

if: -This statement executes some code if one condition is true.

is set():-It is a function to find out if the cookie is set.

header():-This function sends a raw HTTP header to a client.

date():-It is a function that formats a timestamp to a more readable date and time.**echo():-** This function outputs one or more strings.

while(): -The while loop execute a block of code while the specified condition is true.

mysqli _query():- This function performs a query against the database.

mysqli _fetch _array():-This function fetches a result row as an associative array ,a numericarray or both.

rand():-This function generates a random integer.

mysqli _ connect():-This function opens a new connection to the MYSQL server.

mysqli_ error():-This function returns the last error description for the most recent function call ,if any.

ob_ start():-Turn on output buffering and no output is sent from the script.

error_ reporting():- This function specifies which errors are reported.

mysqli_fetch_assoc :-This function fetches a result row as an associative array.

\$_SERVER[,,PHP_SELF"]:-This super global variable returns the filename of the currently executing script.

\$_POST[]:- It is widely used to pass variables and to collect from data after submitting an HTML form with method=||post||.

\$_GET[]:- It is used to collect from data after submitting an HTML form with method=||get|| and also collect data sent in the URL.

mysqli_num_rows():-This function returns the number of rows in a result set.

unset():-The function unset only local variable and it does not return any values.

ob_end_flush():-This function flush(send) the output bufferand turn off output buffering.

substr():-It returns the portion of string specified by the start and length parameters.

md5:-This function calculates the MD5 hash of str and returns that hash.

session_destroy():-Destroys all of the data registered to a session.

Bind-param():-This function binds the parameters to the SQL query and tells the database what the parameters are.

5. MODULE DESCRIPTIONPROJECT MODULES:

USER VALIDATION AND EXTRACTION OF SYMPTOMS:

The validation of the user login details occurs here. Then Symptoms are extracted using String Searching Algorithm where substring representing the symptoms is identified in the natural language text input. When users give directly the symptom name such as(e.g. -I have cough, fever, and nauseal), the system will easily identify it. But however, the system should also be able to handle input like, -When I read, I'm okay at first, but over time, my eyes seem to get tired, and I start to see double. In this case, the system should extract substrings like -eyes tired and -see double (and not substrings like -read or -okay).

MAPPING EXTRACTED SYMPTOMS WITH TRAINED DATASETS:

Given some extracted substring from the user's input, we generate a list of suggested closest symptoms .We then ask the user to confirm if they have any of the suggested symptoms. Based on their reply few diseases are being shortlisted. Then further symptom clarification and symptom suggestions are being done by asking the users a series of questions and the mapping of the symptoms to the exact disease is done.

SPECIFYING THE DISEASE AND REFERRING A DOCTOR:

This process carries the list of diseases in the database and each symptom being entered is compared to the symptoms of the common diseases. Next symptom is checked until a matching one is found. The diseases are shortlisted based on the end users input on the question evaluation. The accurate disease is identified and specified to the end user by the chatbot. The chatbot checks whether the identified disease is a major issue or minor issue based on the conditions built in the chatbot. If it is a major issue the chatbot refers a specialist

6.

SYSTEM DESIGN

INPUT DESIGN

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

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

OBJECTIVES

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

OUTPUT DESIGN

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

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

2. Select methods for presenting information.

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

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

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

6.1 SYSTEM ARCHITECTURE:

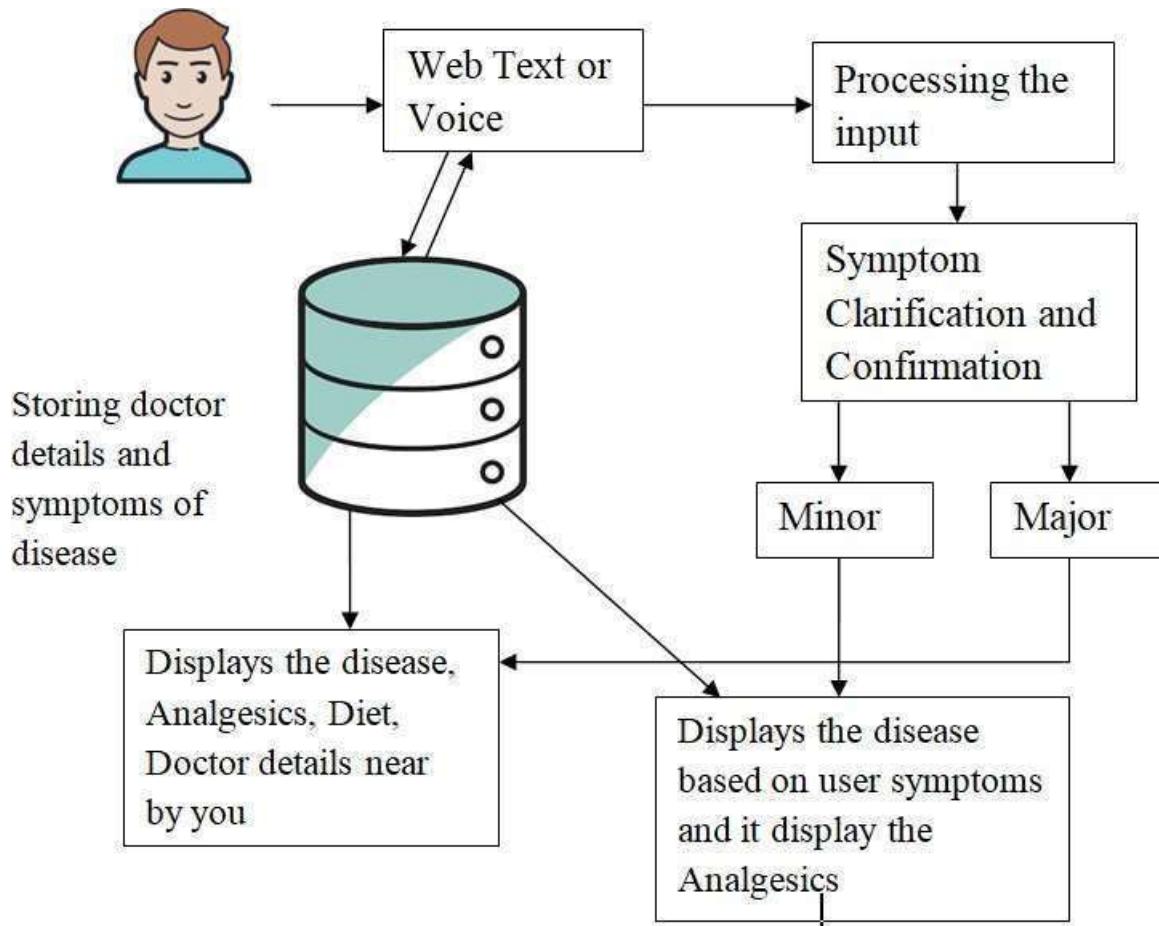


Fig: System Architecture

The above Figure proceeds with the user can start their conversation with the chatbot like user friendly and it will be stored in the database for future reference. The chatbot will clarify the users symptoms with serious of questions and the symptom conformation will be done. The disease will be categorized as minor and major disease. Chatbot will reply whether it's a major or minor disease. If it's a major one user will be suggested with the doctor details near by you for further treatment and display the analgesics and also provides food suggestions that means which food you have to take more to recover the disease. The chatbot user interface can chat with like user friendly. by using chatbot don't go to hospitals for even small problems.

6.2 DATA FLOW DIAGRAM:

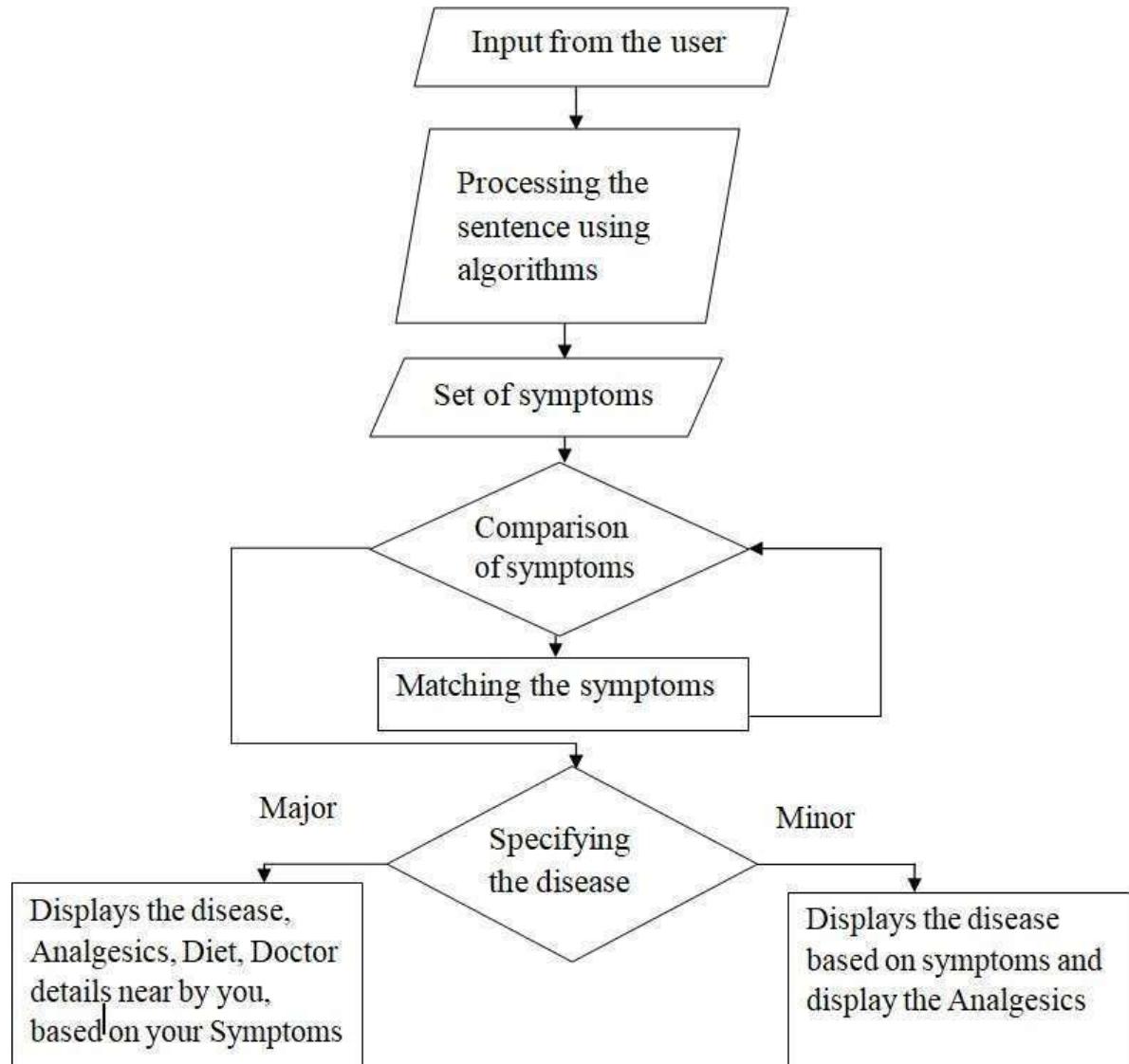


Fig: Data flow diagram of chatbot

The chatbot will take the input from the user and then processing the input by using algorithms. Bot will apply the algorithms on whatever the user give the input to the bot. It will understand the input by using algorithms, set of symptoms in the database. The chatbot will clarify the users symptoms with serious of questions and the symptom conformation will be done. The disease will be categorized as minor and major disease. Chatbot will reply whether it's a major or minor disease. If it's a major one user will be suggested with the doctor details near by you for further treatment and display the analgesics and also provides food suggestions that means which food you have to take more to recover the disease

6.3 METHODOLOGY:

Natural language processing:

Natural language processing involves the reading and understanding of spoken or written language through the medium of a computer. This includes, for example, the automatic translation of one language into another, but also spoken word recognition, or the automatic answering of questions. Computers often have trouble understanding such tasks, because they usually try to understand the meaning of each individual word, rather than the sentence or phrase as a whole. So for a translation program, it can be difficult to understand the linguistic nuance in the word 'Greek' when it comes to the examples 'My wife is Greek' and 'It's all Greek to me', for example. Through natural language processing, computers learn to accurately manage and apply overall linguistic meaning to text excerpts like phrases or sentences. But this isn't just useful for translation or customer service chat bots: computers can also use it to process spoken commands or even generate audible responses that can be used in communication with the blind, for example. Summarizing long texts or targeting and extracting specific keywords and information within a large body of text also requires a deeper understanding of linguistic syntax than computers had previously been able to achieve.

How does natural language processing work?

It doesn't matter whether it's processing an automatic translation or a conversation with a chat bot all natural language processing methods are the same in that they all involve understanding the hierarchies that dictate interplay between individual words. But this isn't easy – many words have double meanings. 'Pass' for example can mean a physical handover of something, a decision not to partake in something, and a measure of success in an exam or another test format. It also operates in the same conjugation as both a verb and a noun. The difference in meaning comes from the words that surround 'pass' within the sentence or phrase (*I passed the butter/on the opportunity/the exam*).

These difficulties are the main reason that natural language processing is seen as one of the most complicated topics in computer science. Language is often littered with double meanings, so understanding the differences requires an extensive knowledge of the content in which the different meanings are used. Many users have first-hand experience of failed communication with chat bots due to their continued use as replacements for live chat support in customer service. But despite these difficulties, computers are improving their

understanding of human language and its intricacies. To help speed this process up, computer linguists rely on the knowledge of various traditional linguistic fields: The term morphology is concerned with the interplay between words and their relationship with other words

Syntax defines how words and sentences are put together Semantics is the study of the meaning of words and groups of words Pragmatics is used to explain the content of spoken expressions And lastly, phonology covers the acoustic structure of spoken language and is essential for language recognition

We are using three algorithms to implement making health care chatbot using nlp technique

1.n-gram algorithm

2.TF-IDF(term frequency-inverse data

frequency)3.Cosine similarity algorithm

N-gram algorithm:

In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the items are words, n-grams may also be called shingles

Using Latin numerical prefixes, an n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram". English cardinal numbers are sometimes used, e.g., "four-gram", "five-gram", and so on. In computational biology, a polymer or oligomer of a known size is called a k-mer instead of an n-gram, with specific names using Greek numerical prefixes such as "monomer", "dimer", "trimer", "tetramer", "pentamer", etc., or English cardinal numbers, "one-mer", "two-mer", "three-mer", etc.

EXAMPLES:

Field	Unit	Sample sequence	1-gram sequence	2-gram sequence	3-gram sequence
Vernacular name			unigram	Bigram	Trigram
resulting <u>Markov model</u>			0	1	2

<u>Protein Order of sequencing</u>	<u>amino acid</u>	... Cys-Gly-Leu-Ser-Trp, Cys, Gly, Leu, Ser, Trp,, Cys-Gly, Gly-Leu, Leu-Ser, Ser-Trp,, Cys-Gly-Leu, Gly-Leu-Ser, Leu-Ser-Trp, ...
<u>DNA sequencing</u>	<u>base pair</u>	...AGCTTCGA...	..., A, G, C, T, T, C, G, A,, AG, GC, CT, TT, TC, CG, GA,, AGC, GCT, CTT, TTC, TCG, CGA, ...
<u>Computational linguistics</u>	<u>character</u>	...to_be_or_not_to_be...	..., t, o, _, b, e, _, o, r, _, n, o, t, _, t, o, _, b, e,, to, o, _, b, be, e, _, o, or, r, _, n, no, ot, t, _, t, to, o, _, b, be,, to_, o_b, _be, be_, e_o, _or, or_, r_n, _no, not, ot_, t_t, _to, to_, o_b, _be, ...
<u>Computational linguistics</u>	<u>word</u>	... to be or not to be, to, be, or, not, to, be,, to be, be or, or not, not to, to be,, to be or, be or not, or not to, not to be, ...

Applications:

n-gram models are widely used in statistical natural language processing. In speech recognition, phonemes and sequences of phonemes are modeled using a *n*-gram distribution. For parsing, words are modeled such that each *n*-gram is composed of *n* words. For language identification, sequences of characters/graphemes (*e.g.*, letters of the alphabet) are modeled for different languages. For sequences of characters, the 3-grams (sometimes referred to as "trigrams") that can be generated from "good morning" are "goo", "ood", "od ", "d m", " mo", "mor" and so forth, counting the space character as a gram (sometimes the beginning and end of a text are modeled explicitly, adding "__g", "_go", "ng_", and "g__"). For sequences of

words, the trigrams (shingles) that can be generated from "the dog smelled like a skunk" are "# the dog", "the dog smelled", "dog smelled like", "smelled like a", "like a skunk" and "a skunk #".

- ◆ N-Grams are one way to help machines understand a word in the content to get a better understanding of word.
- ◆ N-gram is a contiguous sequence of n-items from a given sample of text. N-items means we can have two items, three items and so on. So , it is a contiguous sequence of some items.it helped to predicting the next words in a sentence.
- ◆ Items can be characters , words , sentences. When n is 2 then we can call it as bigrams and n is 3 then we can call it as trigrams.Based on sentence we can change the value of n . Example1:I am suffering from fever.

Example2:I am suffering from headache.

$n=3$ (trigram)

example1:

(I‘, am‘, suffering‘) ->I am suffering

(am‘, suffering‘, from‘) ->I am suffering from

(suffering‘, from‘, fever‘) ->I am suffering from fever

example2:

(I‘, am‘, suffering‘) ->I am suffering

(am‘, suffering‘, from‘) ->I am suffering from

(suffering‘, from‘, headache‘) ->I am suffering from headache.

TF-IDF(term frequency-inverse data frequency):

Term frequency(tf):

Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "the brown cow". A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. However, in the case where the length of documents varies greatly, adjustments

are often made (see definition below). The weight of a term that occurs in a document is simply proportional to the term frequency.

Often times, when building a model with the goal of understanding text, you'll see all of stop words being removed. Another strategy is to score the relative importance of words using TF-IDF.

The number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

- Exmaple1: I am suffering from fever.
 - Example2:I am suffering from headache.
 - Tf=number of occurances/number of words in document
 - Example1: Example2:
 - $Tf(I) = 1/5 = 0.2$ $Tf(I) = 1/5 = 0.2$
 - $Tf(am) = 1/5 = 0.2$ $Tf(am) = 1/5 = 0.2$
 - $Tf(suffering) = 1/5 = 0.2$ $Tf(suffering) = 1/5 = 0.2$
 - $Tf(from) = 1/5 = 0.2$ $Tf(from) = 1/5 = 0.2$
 - $Tf(fever) = 1/5 = 0.2$ $Tf(headache) = 1/5 = 0.2$

Inverse Data Frequency (IDF):

Because the term "the" is so common, term frequency will tend to incorrectly emphasize documents which happen to use the word "the" more frequently, without giving enough weight to the more meaningful terms "brown" and "cow". The term "the" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less-common words "brown" and "cow". Hence an *inverse document frequency* factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

The log of the number of documents divided by the number of documents that contain the word w . Inverse data frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = \log\left(\frac{N}{df_t}\right)$$

Exmaple1: I am suffering from fever.

Example2:I am suffering from headache.

Idf=log(number of documents/number of documents that containing a word)

$$Idf(I) = \log(2/2) = 0$$

$$Idf(am) = \log(2/2) = 0$$

$$Idf(suffering) = \log(2/2) = 0$$

$$Idf(from) = \log(2/2) = 0$$

$$Idf(fever) = \log(2/1) = 0.301$$

$$Idf(headache) = \log(2/1) = 0.301$$

Tf -idf = Tf*idf

Documents	I	am	suffering	from	Fever	headache
Document1	0.2*0=0	0.2*0=0	0.2*0=0	0.2*0=0	0.2*0.301=0.012	0
Document2	0.2*0=0	0.2*0=0	0.2*0=0	0.2*0=0	0	0.2*0.301=0.012

Output of TF-IDF:

	_patches	abdominal_pain	abnormal_menstruation	acidity \
0	0.517716	0.0	0.0	0.0
1	0.546610	0.0	0.0	0.0
2	0.543606	0.0	0.0	0.0
3	0.605124	0.0	0.0	0.0
4	0.000000	0.0	0.0	0.0
...
4915	0.000000	0.0	0.0	0.0
4916	0.000000	0.0	0.0	0.0
4917	0.000000	0.0	0.0	0.0
4918	0.000000	0.0	0.0	0.0
4919	0.000000	0.0	0.0	0.0

The above fig represents the importance of keyword in the document which means calculate the term frequency and inverse data frequency which is used for find the keywords in the document .so tf-idf algorithm is used for calculate the keyword importance in the document

Cosine similarity algorithm:

- ◆ Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.
- ◆ The technique is also used to measure cohesion within clusters in the field of data mining.
- ◆ $\text{Cosine similarity} = \frac{\mathbf{AB}}{|\mathbf{A}||\mathbf{B}|}$.
- ◆ Chat Bot COSINE SIMILARITY DISTANCES
- ◆ Introduction on Chat bot techniques There are two main technique involved in chat bot to understand the user input such as Pattern matching and Intent classification. Types of response:
Static Response: The simplest way is to have a static response, with eventually a list of variants, for each user input. These static responses could be templates, such as -John is located in <Location details>||, where <Location details> is a variable computed by the chat bot.
Dynamic Response: A different approach would be to use resources, such as a knowledge base, to get a list of potential responses, and then score them to choose the better response. This is particularly appropriate if the chat bot acts mainly like a question-answering system.
Generated Response: If you have a huge corpus of examples of conversations, you could use a deep learning technique (Recurrent Neural Network) to train a generative model that, given an input, will generate the answer. You will need millions of examples to reach a decent quality and sometimes the results are going to be unexpected, but it could be interesting.
- ◆ Introduction on Text Similarity
1. Words are similar in two ways lexically and semantically.
2. Words are similar lexically if they have a similar character sequence.
3. Words are similar semantically if they have the same thing, are opposite of each other, used in the same way , used in the same context.
4. Lexical similarity is introduced through String based algorithms and semantic similarity is introduced through Corpus based and Knowledge based algorithms.
5. String based measures operates on string sequences and character composition.
- ◆ Text similarity approaches
1. Measuring the similarity between words, sentences, paragraphs and documents is an important component in various tasks such as information retrieval, document clustering, word-sense disambiguation, automatic essay scoring, short answer grading, machine translation, topic tracking, question generation, question answering and text summarization.
2. Text similarity is

partitioned into three approaches Text similarity String based similarity Corpus based similarity Knowledge based similarity

- String based Character based Jaro LCS Levenshtein Jaro - winkler Smith waterman Needleman Wunsch N gram Term based Dice's coefficient Block distance Cosine similarity Euclidean distance Matching coefficient Jaccard similarity Overlap coefficient String based similarity measures
- Term Based Example 1: Cosine similarity Term based Distances: Let us consider two example sentences to calculate the term based text similarity - Jenny loves burger more than Linda loves pizza - Jane likes pizza more than Jenny loves burger Cosine Similarity: Words = [burger, jenny, loves, linda, than, more, likes, jane, pizza] If you take the word count , you will get a matrix like mentioned on right side. Now calculating cosine similarity between a and b a : [1,1,2,1,1,1,0,0,1] b : [1,1,1,0,1,1,1,1,1] The cosine of the angle between vectors is their similarity , $\cos \alpha = \frac{a \cdot b}{\|a\| \|b\|}$ $a \cdot b = \text{dot product of vectors} / \text{Vectors magnitude}$ $\cos \alpha = 7 / \sqrt{9} = 0.782$ and $\cos \text{dissimilarity} = 1 - 0.782 = 0.217$ Terms Seq 1 Seq 2 Burger 1 1 Jenny 1 1 Loves 2 1 Linda 1 0 Than 1 1 More 1 1 Likes 0 1 Jane 0 1 pizza 1 1
- Term Based Example 2 : Jaccard Similarity Jaccard similarity : $J(a, b) = \frac{|a \cap b|}{|a \cup b|}$ a : { jenny, loves, burger, more, than, pizza, Linda} b : { jane, likes, pizza, more, than, burger, loves, jenny} $a \cap b = \{ \text{jenny, loves, burger, more, pizza} \}$ $a \cup b = \{ \text{jenny, loves, burger, more, pizza, than, likes, jane} \}$ Jaccard similarity = $6 / 9 = 0.666$ Removing stop words by parsing , Jaccard similarity = $5 / 8 = 0.625$ and Jaccard dissimilarity = $1 - 0.625 = 0.375$
- Cosine distance is nothing but getting distance between two vectors in n dimension space.Distance represent how words are related to each other
- Example:||fever|| and ||low fever|| will have a cosine distance very low

Disease	Fungal infection	Fungal infection	\
Disease			
Fungal infection	1.000000	0.947140	
Fungal infection	0.947140	1.000000	
Fungal infection	0.952373	0.891420	
Fungal infection	0.855553	0.776283	
Fungal infection	0.681132	0.559602	
...	
(vertigo) Paroxysmal Positional vertigo	0.000000	0.000000	
Acne	0.098174	0.103654	
Urinary tract infection	0.000000	0.000000	
Psoriasis	0.083380	0.088033	
Tinnitus	0.005923	0.101276	

The above diagram shows the similarity between the two sentences or documents. Cosine similarity has been used to check the similarity between two sentences. The similarity between the query and document is directly proportional to the number of query weights. The similarity calculation result of the two documents ranges from 0 to 1 since the term frequency cannot be negative. It calculates similarity between two sentences using cosine formula. The technique is also used to measure cohesion within clusters in the field of data mining. Cosine similarity = $AB / |A||B|$. Cosine distance is nothing but getting distance between two vectors in n dimension space. Distance represents how words are related to each other.

Algorithms:

- 1.DecisionTree
- 2.Svm
- 3.Knn
- 4.RandomForest

RandomForest:

Random Forest is a popular machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or the average prediction (regression) of the individual trees. We have various algorithms like SVM, KNN, Decision Tree and Random Forest. But Random Forest has produced more accuracy, precision, f1-scores compared to all algorithms.

```

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Data for the before and after scenarios
data_before = {

    'Model': ['DECISION TREE', 'RANDOM FOREST','K-NEAREST NEIGHBOUR','SUPPORT VECTOR
MACHINE'],
    'Accuracy': [97.1,98.4,97.8,96.2],
    'Precision': [96.9, 97.7,97.3,95.4],
    'F1_Score': [96.9, 97.8, 97.4, 95.8]
}

data_after = {
    'Model': ['DECISION TREE', 'RANDOM FOREST','K-NEAREST NEIGHBOUR','SUPPORT VECTOR
MACHINE'],
    'Accuracy': [99.5, 99.5, 99.4, 98.4],
    'Precision': [99.5, 99.6, 99.5, 98.7],
    'F1_Score': [99.5, 99.5, 99.4, 98.3]
}

# Create dataframes
dtfm_before = pd.DataFrame(data_before)
dtfm_after = pd.DataFrame(data_after)

# Set the Model column as the index for better plotting
dtfm_before.set_index('Model', inplace=True)
dtfm_after.set_index('Model', inplace=True)

# Create a bar graph
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(30, 20))

```

```

def plot_bargraph(ax, metric, title):
    before_values = dtfm_before[metric]
    after_values = dtfm_after[metric]
    models = dtfm_before.index

    index = np.arange(len(models))
    bar_width = 0.35

    ax.bar(index, before_values, width=bar_width, label='Before', color='skyblue')
    ax.bar(index + bar_width, after_values, width=bar_width, label='After', color='lightcoral')

    for i in range(len(models)):
        ax.text(index[i] - 0.05, before_values[i] + 1, f'{before_values[i]:.1f}', fontsize=20)
        ax.text(index[i] + bar_width - 0.05, after_values[i] + 1, f'{after_values[i]:.1f}', fontsize=20)

    ax.set_xlabel('Model')
    ax.set_ylabel(metric)
    ax.set_title(title)
    ax.set_xticks(index + bar_width / 2)
    ax.set_xticklabels(models)
    ax.legend()

plot_bargraph(axes[0, 0], 'Accuracy', 'Accuracy Comparison')
plot_bargraph(axes[0, 1], 'Precision', 'Precision Comparison')
plot_bargraph(axes[1, 0], 'F1_Score', 'F1 Score Comparison')

# Adjust layout for better appearance
plt.tight_layout()
plt.show()

import joblib
import pandas as pd

```

```

# Assuming you have a function to preprocess input data
def preprocess_input(symptoms_str, all_symptoms):
    # Create a DataFrame with all symptoms as columns
    symptoms_df = pd.DataFrame(columns=all_symptoms)

    # Add a row with zeros
    symptoms_df.loc[0] = 0

    # Set the corresponding symptoms to 1 based on the input
    for symptom in symptoms_str.split(','):
        symptoms_df.at[0, symptom] = 1

    return symptoms_df.values

# Save the model
joblib.dump(rf_model, "mymodel.pkl")

# Load the model
model = joblib.load("mymodel.pkl")

# Example symptoms string
input_symptoms = "itching,skin_rash,nodal_skin_eruptions,dischromic_patches"

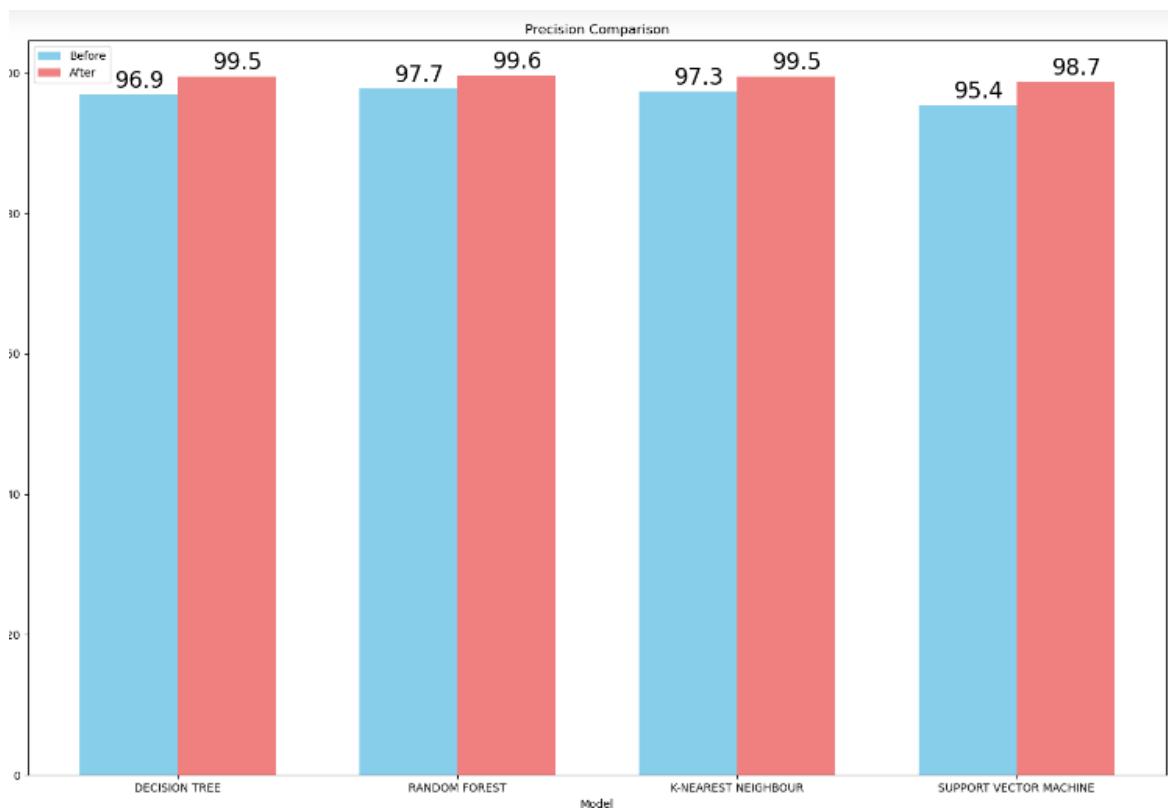
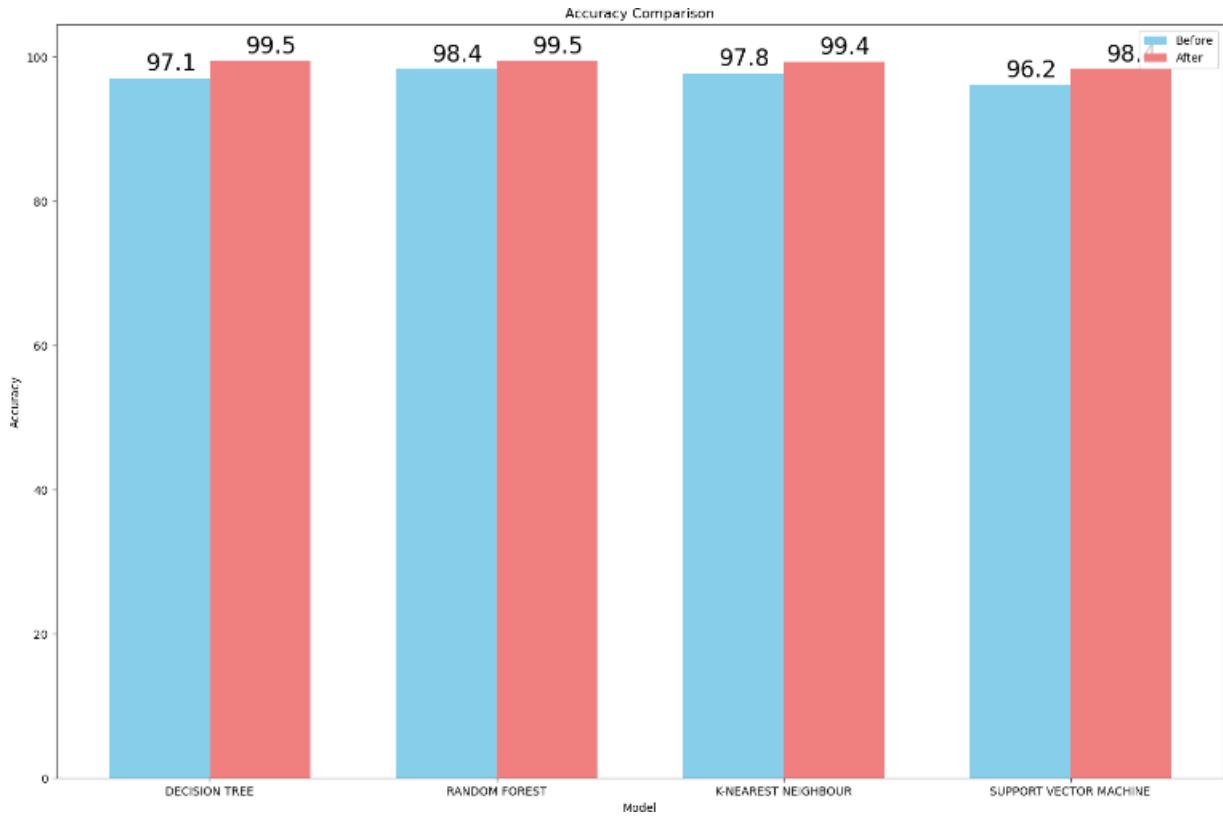
# List of all possible symptoms (replace with your actual list)
all_symptoms = ["itching", "skin_rash", "nodal_skin_eruptions",
                 "dischromic_patches", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0"]

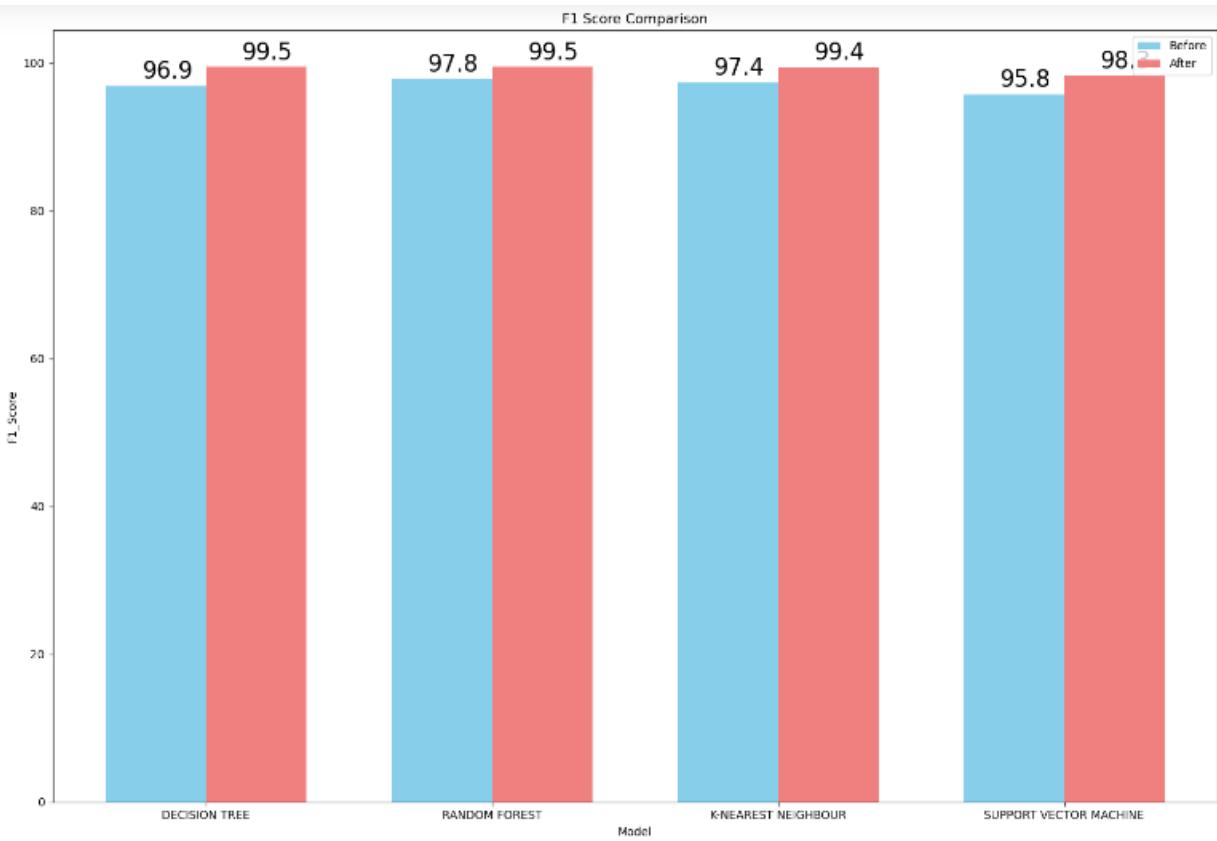
# Preprocess the input
processed_input = preprocess_input(input_symptoms, all_symptoms)

# Make a prediction
prediction = model.predict(processed_input)[0]

```

```
print("predictedDisease:"prediction)
```





7.

```
# Meet Pybot: your friend
```

```
import nltk

import warnings

warnings.filterwarnings("ignore")

# nltk.download() # for downloading
packages

#import tensorflow as tf

import numpy as np

import random

import string # to process standard
```

python strings

```
f=open('symptom.txt','r',errors = 'ignore')

m=open('pincodes.txt','r',errors = 'ignore')

checkpoint = "./chatbot_weights.ckpt"

#session = tf.InteractiveSession()

#session.run(tf.global_variables_initialize
r())

#saver = tf.train.Saver()

#saver.restore(session, checkpoint)

raw=f.read()

rawone=m.read()

raw=raw.lower()# converts to lowercase

rawone=rawone.lower()# converts to
lowercase

nltk.download('wordnet') # first-time use
only

sent_tokens = nltk.sent_tokenize(raw)#
converts to list of sentences
```

```
word_tokens = nltk.word_tokenize(raw)#
```

converts to list of words

```
sent_tokensone =
```

```
nltk.sent_tokenize(rawone)# converts to
```

list of sentences

```
word_tokensone =
```

```
nltk.word_tokenize(rawone)# converts to
```

list of words

```
sent_tokens[:2]
```

```
sent_tokensone[:2]
```

```
word_tokens[:5]
```

```
word_tokensone[:5]
```

```
lemmer =
```

```
nltk.stem.WordNetLemmatizer()
```

```
def LemTokens(tokens):
```

```
    return [lemmer.lemmatize(token)
```

```
for token in tokens]
```

```
remove_punct_dict = dict((ord(punct),
```

```
None) for punct in string.punctuation)
```

```
def LemNormalize(text):  
    return  
    LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))
```

Introduce_Ans = [" "]

GREETING_INPUTS = ("hello",
"hi","hiii","hii","hiiii","hiiii", "greetings",
"sup", "what's up","hey",)

GREETING_RESPONSES = ["hi,are
you suffering from any health
issues?(Y/N)", "hey,are you having any
health issues?(Y/N)", "hii there,are you
having any health issues?(Y/N)", "hi
there,are you having any health
issues?(Y/N)", "hello,are you having any
health issues?(Y/N)", "I am glad! You are
talking to me,are you having any health
issues?(Y/N)"]

Basic_Q = ("yes","y")

Basic_Ans = "okay,tell me about your
symptoms"

Basic_Om = ("no","n")

Basic_AnsM = "thank you visit again"

```
fev=("iam suffering from fever", "i  
affected with fever","i have  
fever","fever")
```

```
feve_r=("which type of fever you have?  
and please mention your symptoms then  
we try to calculate your disease.")
```

```
# Checking for greetings
```

```
def greeting(sentence):  
  
    """If user's input is a greeting, return a  
greeting response"""
```

```
for word in sentence.split():  
    if word.lower() in  
GREETING_INPUTS:  
  
        return  
random.choice(GREETING_RESPONSE  
S)
```

```
# Checking for Basic_Qs
```

```
def basic(sentence):  
  
    for word in Basic_Q:  
        if sentence.lower() == word:
```

```
    return Basic_Ans

def fever(sentence):
    for word in fev:
        if sentence.lower() == word:
            return feve_r

# Checking for Basic_QM

def basicM(sentence):
    """If user's input is a greeting, return a
    greeting response"""

    for word in Basic_Om:
        if sentence.lower() == word:
            return Basic_AnsM

# Checking for Introduce

def IntroduceMe(sentence):
    return random.choice(Introduce_Ans)
```

```
from sklearn.feature_extraction.text
import TfidfVectorizer

from sklearn.metrics.pairwise import
cosine_similarity

# Generating response

def response(user_response):

    robo_response="

    sent_tokens.append(user_response)

    TfidfVec =
        TfidfVectorizer(tokenizer=LemNormaliz
e, stop_words='english')

    tfidf =
        TfidfVec.fit_transform(sent_tokens)

    vals = cosine_similarity(tfidf[-1], tfidf)

    idx=vals.argsort()[0][-2]

    flat = vals.flatten()

    flat.sort()
```

```
req_tfidf = flat[-2]

if(req_tfidf==0):

    robo_response=robo_response+"I

    am sorry! I don't understand you"

    return robo_response

else:

    robo_response =

    robo_response+sent_tokens[idx]

    return robo_response
```

Generating response

```
# Generating response

def responseone(user_response):

    robo_response="

    sent_tokensone.append(user_response)

    TfidfVec =

    TfidfVectorizer(tokenizer=LemNormaliz

e, stop_words='english')

    tfidf =

    TfidfVec.fit_transform(sent_tokensone)

    vals = cosine_similarity(tfidf[-1], tfidf)
```

```
idx=vals.argsort()[0][-2]

flat = vals.flatten()

flat.sort()

req_tfidf = flat[-2]

if(req_tfidf==0):

    robo_response=robo_response+"I

    am sorry! I don't understand you"

    return robo_response

else:

    robo_response =

    robo_response+sent_tokensone[idx]

    return robo_response

def chat(user_response):

    user_response=user_response.lower()

    keyword = " module "

    keyworddone = " module"

    keywordsecond = "module "

    if(user_response!="bye"):
```

```
    if(user_response=='thanks' or
user_response=='thank you' ):

        flag=False

        #print("ROBO: You are
welcome..")

        return "You are welcome.."

    elif(basicM(user_response)!=None):

        return basicM(user_response)

    else:

        if(user_response.find(keyword) !=

-1 or user_response.find(keyworddone) !=

-1 or user_response.find(keywordsecond)

!= -1):


            #print("ROBO: ",end="")



#rint(responseone(user_response))

return

responseone(user_response)

sent_tokensone.remove(user_response)

elif(greeting(user_response)!=None):


    #print("ROBO:
```

```
+greeting(user_response))

    return greeting(user_response)

    elif(user_response.find("your
name") != -1 or user_response.find("your name") != -1 or
user_response.find("your name ") != -1 or
user_response.find(" your name ") != -1):
        return

    IntroduceMe(user_response)

    elif(basic(user_response)!=None):
        return basic(user_response)

    elif(fever(user_response)!=None):
        return fever(user_response)

    else:
        #print("ROBO: ",end="")

# print(response(user_response))

    return response(user_response)

sent_tokens.remove(user_response)

else:
```

```
flag=False  
#print("ROBO: Bye! take care..")  
  
return "Bye! take care.."
```

8.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

8.1.1 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. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

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

8.1.3 Functional testing

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

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.1.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.1.5 White Box Testinsg

White Box Testing is a testing in which in the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

8.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot -seel into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

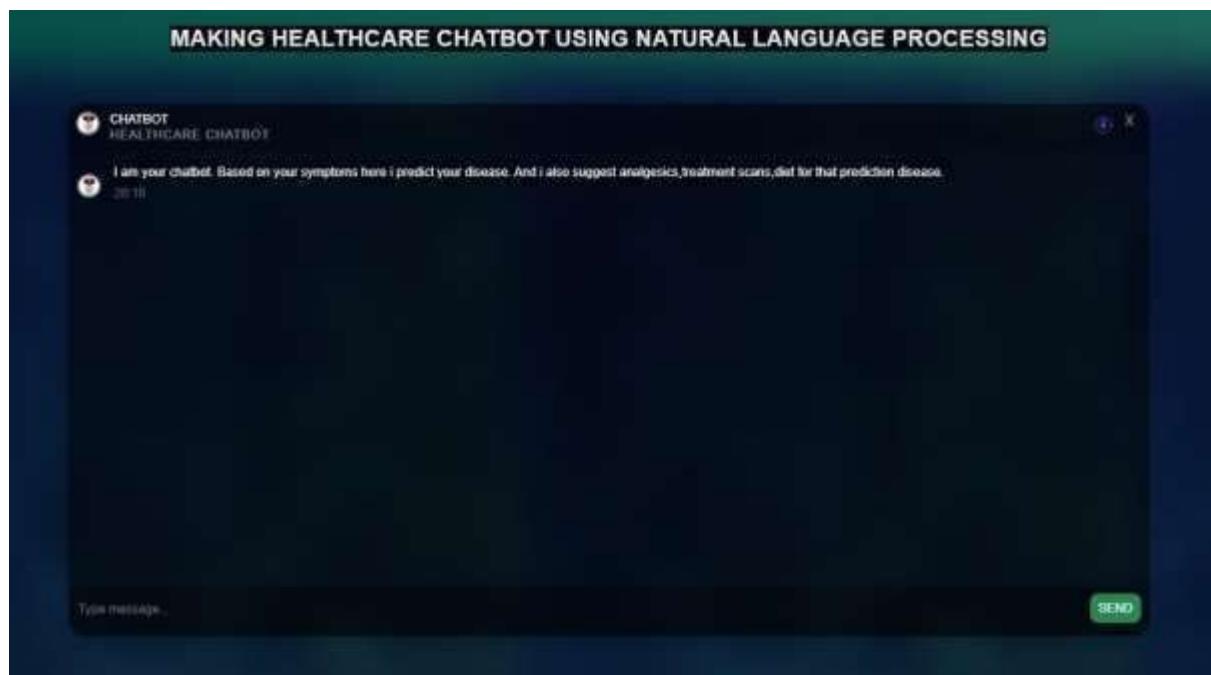
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

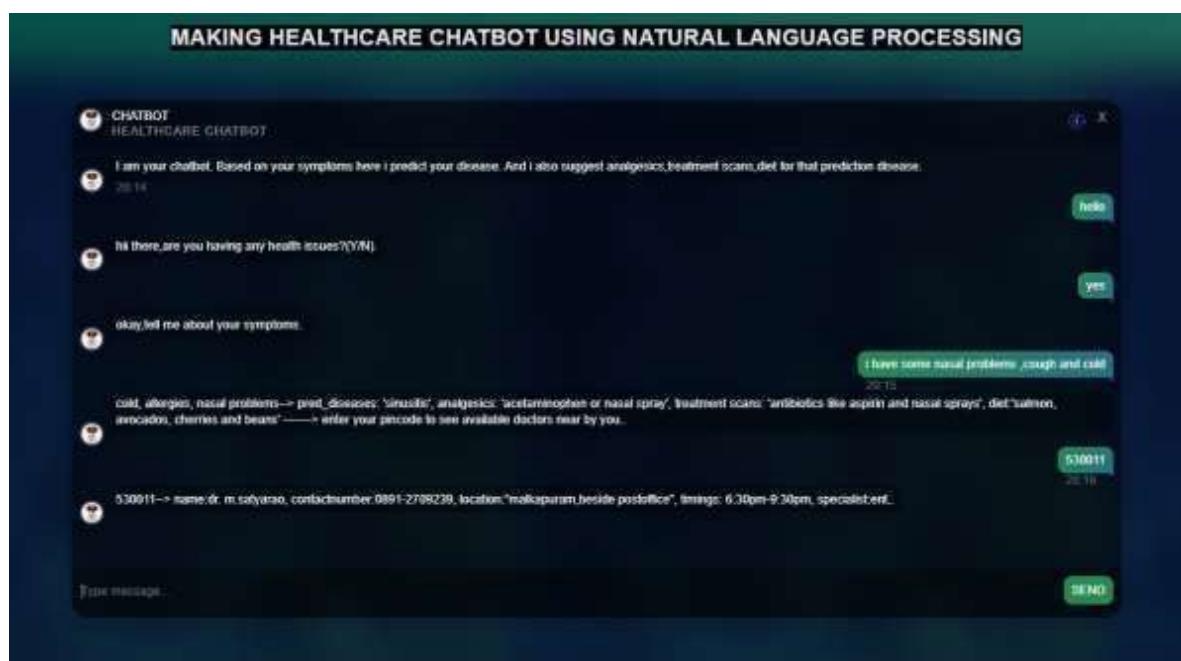
8.

SCREEN SHOTS

CHATBOT USER INTERFACE:



CHATBOT COMMUNICATION PAGE:



TRAINED DATASETS:

D	symptoms	disease																	
1	Allergy	nasal congestion, itchy and watery eyes, sneezing, stuffy or runny nose, scratchy or sore throat, throat clearing, cough from postnasal drip.																	
2	anemia	Feel tired or lightheaded (sometimes with fainting). Weakness Fatigue easily Have decreased energy Appear pale Develop palpitations or rapid heart rate Experience shortness of breath Headaches, body aches, fever, and flu-like symptoms Nasal congestion, runny nose, and sneezing Cough																	
3	cold/flu	Sore throat																	
4	asthma	shortness of breath chest pain cough loss of orientation (to person, place, or time), agitation, irritability, quarreling, and																	
5	alzheimers	a diminishing ability to care for him- or herself and to dress																	
6	pregnancy	absence of menstrual periods, breast swelling and tenderness, Food cravings																	
7	hyperthyroid	a rapid heart rate, excessive sweating, intolerance to heat, tremor, nervousness, or agitation. Other symptoms can include fatigue, weight loss, hair loss, increased appetit																	
8	hypothyroid	fatigue, depression, mild weight gain, cold intolerance, sleepiness, constipation dry and coarse hair, dry skin, and muscle cramps. Blood cholesterol levels may be elevat																	
9	diabetes	dehydration, increased urination, and increased thirst.																	
10	HIV AIDS	fever, swollen lymph nodes, joint and muscle aches, and sore throat, chills, night sweats, and mouth ulcers.																	
11	high blood pressure	dizziness, shortness of breath, headache, and blurred vision nosebleeds, blood in the urine, fatigue, chest pain,																	
12	Rheumatoid Arthritis	aches stiffness, muscle aches, low-grade fever, fatigue, lack of appetite, loss of energy . Joints can become warm, swollen, reddened, painful,																	
		fatigue, loss of appetite, nausea,																	
13	hepatitis B	jaundice (yellowing of the skin and eyes), and pain in the upper right abdomen (due to the inflamed																	
14	Dengue	headache, fever, exhaustion, severe muscle joint pain, swollen lymph nodes rash, fever, itchy rash, headache fever and chills, headaches, nauses and vomiting, and																	
15	Malaria	general weakness and body aches																	
16	Chicken Pox	general weakness, fever up to 102 F, and red spots that start on the same day or so as the fever																	
		Yellow discoloration of the skin, mucous membranes and the whites of the eyes. Light-colored skin Poor feeding Lethargy/excessive sleepiness																	
17	jaundice	Changes in muscle tone (either listless or stiff with arching of the back).																	
18	Diarrhea	bowel movements are frequent watery, no signs of inflammation, cramping abdominal pain																	
19	cataract	decrease in clarity of vision, not fully correctable with glasses, loss of contrast sensitivity, Disturbing glare in light																	
		fever, chills, cough,																	
20	pneumonia	shortness of breath, and fatigue.																	
		inability to urinate painful, urgent need to urinate																	
21	Urinary Retention	pain or discomfort in the lower abdomen bloating of the lower abdomen																	
		Waking up unrefreshed from lack of sleep Daytime sleepiness and fatigue Mood changes Poor concentration and attention Anxiety Headaches Lack of energy																	
22	Insomnia	Increased errors and mistakes																	
23	coronavirus	Difficulty breathing or shortness of breath, heavy cough, Persistent pain or pressure in the chest																	

Stopped Growth :--> pred_diseases:'Growth Disorder, Turner Syndrome',
analgesics: 'Thyroid hormone pills, growth hormone injections',
treatment scans:'Bone Age Xray, MRI scan',
'Meat, Seafood, Leafygreens, Diary, Spinach, Beef'.

Wheezing, Coughing and troubled breathing, chest pain -->pred_diseases: Asthma;
analgesics: Metered dose inhalers, nebulizers;
treatment scans: Asthma therapy;
diet: Fruits and vegetables.

baby too small, weight<5.5 pounds-->pred_diseases: 'Learning disabilities',
analgesics: 'iron supplements',
treatment scans: 'Temperature Control Incubator',
diet: 'iron supplements'.

Repetitive behaviour, prefers to be alone-->pred_diseases: 'Autism\r\n',
analgesics: 'Ketamine, Midazolem',
treatment scans: 'Teach social skills and cognitive behaviour',
diet: 'Only yellow or white foods such as rice, potatoes or pasta and strictly avoid diary'.

Self-destructiveness, sadness and being upset-->
pred_diseases: 'Depression',
analgesics: 'Prozac, Zoloft or Celexa',
treatment scans: 'Constantly praise him, teach skills, and give self-development talks',
diet: 'Fish and whole grains'.

Paleness, loss of energy, weightloss and easy bruising--pred_diseases: 'Cancer or Brain Tumor',

analgesics: 'Morphine or Ibuprofen',
treatment scans: 'Chemotherapy or Stem cell transplant',
diet: 'Cheese, Meat, Seafood, Diary and Eggs'.

Fever, fever up to 102f, Tiredness or Loss of Appetite-->
pred_diseases: 'Chickenpox',
analgesics: 'Acetaminophen',
treatment scans: 'lukewarm bath and apply calamine lotion to itchy areas',
diet: 'Raw fruits, Vegetables, Meat, Pastured eggs and avoid salty foods'.

easy tiredness and overweight-->
pred_diseases: 'Obesity',
analgesics: 'Not suggested to take analgesics for obesity',
treatment scans: 'building healthy eating and drinking habits and physical activities like exercise',
diet: 'Fruits and vegetables, and avoid skipping breakfast'.

Severe tooth ache-->
pred_diseases: 'Cavities\r\n',
analgesics: 'Ibuprofen or Aspirin',
treatment scans: 'Fluoride toothpaste',
diet: 'juices and avoid sweets, chocolates and unhealthy snacks'.

Often urinating, slow healing of bruises, weightloss-->
pred_diseases: 'Diabetes',
analgesics: 'Tylenol or Aspirin',
treatment scans: 'Insulin Treatment',
diet: 'brown rice or cereals with two eggs daily'.

High body temperature, severe headache and tiredness-->

pred_diseases: 'Fever',

analgesics: 'Paracetemol or Aspirin',

treatment scans: 'Tylenol, Ibuprofen to treat stomach irritation',

diet: 'fluid intake like gatorade, fruitjuices or milk'.

Depression, Eating Disorders-->

pred_diseases: 'Drug or Smoke or Alcohol Addiction',

analgesics: 'Baclofen, Gebapentin',

treatment scans: 'Visiting Rehabilitation Centers, Help teens choose good friends and learn to say NO',

diet: 'Green leafy vegetables'.

Swelling, irritation, breast or nipple pain-->

pred_diseases: 'Breast Cancer',

analgesics: 'Tamoxifen',

treatment scans: 'heart-smart diet and indulge in physical activity and avoid smoking',

diet: 'whole grains, beans, legumes'.

Sudden Confusion, Dizziness, Loss of Balance-->

pred_diseases: 'Stroke',

analgesics: 'Anti-platelet drugs like Plavix',

treatment scans: 'Clot-bursting therapy',

diet: 'Fruits and Vegetables, and avoid salt'.

Shortness of breath, or being inactive-->

pred_diseases: 'COPD',

analgesics: 'Tudorza or Brovana',
treatment scans:'Inhalers and oral steroids',
diet: 'Eggs, Cheese, Meat, fish, and Poultry'.

Forgetfulness or Confusion-->
pred_diseases: 'Alzheimer's disease',
analgesics: 'Aricept or Exelon',
treatment scans:'MRI and CT scans',
diet: 'berries, beans and whole grains'.

Chest pain, confusion, cough or fatigue-->
pred_diseases: 'Pneumonia or Influenza',
analgesics: 'Levofloxacin',
treatment scans:'Plenty of fluids and get lot of rest',
diet:'Citrus fruits, Oily fish and Leafy greens'.

Often urinating, slow healing of bruises, weightloss-->
pred_diseases: 'Diabetes',
analgesics: 'Tylenol or Aspirin',
treatment scans:'Insulin Treatment',
diet:'Brown rice or cereals with two eggs daily'.

reduced urine, swelling of legs or fatigue-->
pred_diseases: 'Kidney Disease',
analgesics: 'Naproxen or Ibuprofen',
treatment scans:'Dialysis',
diet:'Berries, bell peppers, onions and apples'.

Fever, chills, rapid breathing and heart rate-->
pred_diseases: 'Blood poisoning',
analgesics: 'Ceftriaxone or Azithromycin',
treatment scans:'Vasoconstriction to narrow blood vessels',
diet:'Leafy greens like spinach'.

Fragile bones-->
pred_diseases: 'Brittle bone disease',
analgesics: 'Biphosphonates, calcium and vitamin D supplements',
treatment scans: 'VitaminD supplements and avoid smoking and alcohol',
diet:'fruits and foods high on calcium'.

Skin wrinkles and aging-->
pred_diseases: 'No disease',
analgesics: 'Anti-aging creams',
treatment scans: 'Blepharoplasty surgery to get rid of aging',
diet: 'Berries, Broccoli, Papaya, Spinach, Nuts and Avocado'.

Blurry and loss of vision-->
pred_diseases: 'Muscular Degeneration',
analgesics: 'Lucentis or Macugen',
treatment scans: 'Regular exercise and avoid smoking',
diet:'Fish and vegetables'.

Blood pressure reading 140 or higher-->
pred_diseases: 'High blood pressure',
analgesics: 'Thiazide diuretics',
treatment scans:'Physical activity like exercise and avoid smoking and drinking',

diet:'low-fat diary, whole grains and Fruits'.

Trouble sleeping or sleep, mood swings, vaginal dryness-->

pred_diseases: 'Menopause',

analgesics: 'Isoflav capsules',

treatment scans: 'Estrogen levels test',

diet:'Fish, Fruits, Vegetables and Diary'.

Cold, Allergies, Nasal problems-->

pred_diseases: 'Sinusitis',

analgesics: 'Acetaminophen or Nasal spray',

treatment scans: 'Antibiotics like Aspirin and Nasal sprays',

diet:'Salmon, Avocados, Cherries and Beans'.

Pain while urine and ejaculation-->

pred_diseases: 'Prostate Cancer',

analgesics: 'Bicalutamide or Casodex',

treatment scans: 'Radiation therapy or prostatectomy',

diet: 'Veggies and fruits and low in meat'.

Trembling body parts and loss of balance-->

pred_diseases: 'Parkinson\'s disease',

analgesics: 'Vesoret capsules',

treatment scans: 'Cardibopa-levodopa',

diet:'Fruits and vegetables'.

Buildup of fluids in legs, ankles and legs, tiredness-->

pred_diseases: 'Heart failure',

analgesics: 'Carvedilol or Metoprolol',
treatment scans: 'ACE inhibitors',
diet: 'Plain rice, fish, egg, milk and yoghurt'.

CONCLUSION

Chatbot is great tool for conversation between human and machine. The application is developed for getting a quick response from the bot which means without any delay it gives the accurate result to the user.

It is concluded that, the usage of chatbot is user friendly and can be used by any person who knows how to type in their own language. Chatbot provides personalized diagnosis based on symptoms. From the review of various journals, it is concluded that, the usage of Chatbot is user friendly and can be used by any person who knows how to type in their own language in mobile app or desktop version. A medical chatbot provides personalized diagnoses based on symptoms. In the future, the bot's symptom recognition and diagnosis performance could be greatly improved by adding support for more medical features, such as location, duration, and intensity of symptoms, and more detailed symptom description. The implementation of Personalized Medical assistant heavily relies on AI algorithms as well as the training data. At last, the implementation of personalized medicine would successfully save many lives and create a medical awareness among the people. As said before, the future era is the era of messaging app because people going to spend more time in messaging app than any other apps. A text-to-text diagnosis Bot engages patients in conversation about their medical issues and provides a personalized diagnosis based on their symptoms. Hence, people will have an idea about their health and have the right protection.

Thus medical chatbot has wide and vast future scope. No matter how far people are, they can have this medical conversation. The only requirement they need is a simple desktop or smartphone with internet connection. The efficient of the chatbot can be improved by adding more combination of words and increasing the use of database so that of the medical chabot could handle all type of diseases. Even voice conversation can be added in the system to make it more easy to use.

FUTURE ENHANCEMENT

The future era is the era of messaging app because people going to spend more time in messaging app than any other apps. The implementation of personalized medicine would successfully save many lives and create a medical awareness among the people. No matter how far people are, they can have this medical conversation. The only requirement they need is a simple desktop or smartphone with internet connection. The efficient of chatbot can be improved by adding more combination of words and increasing the use of database so that of the medical chatbot could handle all type of diseases.

REFERENCES

1. SimonHoermann, Kathryn L McCabe, David N Milne, Rafael A Calvo1,- Application of Synchronous Text- Based Dialogue Systems in Mental Health Interventions: Systematic Review, Journal of Medical Internet Research ,volume: 19 , issue 8 , August 2017.
2. Saurav Kumar Mishra, DhirendraBharti, Nidhi Mishra, Dr.Vdoc: A Medical Chatbot that Acts as aVirtual Doctor, Journal of Medical Science and Technology ,Volume: 6, Issue 3,2017.
3. DivyaMadhu,Neeraj Jain C. J, ElmySebastain, ShinoyShaji, AnandhuAjayakumar, A Novel Approach for Medical Assistance Using Trained Chatbot,International Conference on Inventive Communication and Computational Technologies(ICICCT 2017).
4. HameedullahKazi,B.S. Chowdhry,ZeeshaMemon, MedChatBot: An UMLS based Chatbot for Medical Students, International Journal of Computer Applications (0975 – 8887)Volume 55– No.17, October 2016.
5. DoinaDrăgulescu,AdrianaAlbu,Medi cal Predictions System, International Journal of Engineering Research and Applications , ISSN: 2248-9622 ,Vol. 2, Issue 3 , pp.1988- 1996, May-Jun 2015.
6. Abbas SaliimiLokman, JasniMohamadZain,FakultiSistemKom puter, KejuruteraanPerisian, Designing a Chatbot for Diabetic Patients,ACM Transactions on Management Information Systems (TMIS), Volume 4, Issue 2, August 2015 .
7. PavlidouMeropi, Antonis S. Billis,Nicolas D. Hasanagas,CharalambosBratsas,Ioanni sAntoniou,Panagiotis D. Bamidis, Conditional Entropy Based Retrieval Model in Patient-Carer Conversational Cases,2017 IEEE 30th International conference on Computer-Based Medical System.
8. BenildaEleonor V. Comendador, Bien Michael B. Francisco, Jefferson S. Medenilla, Sharleen Mae T. Nacion, and Timothy Bryle E. Serac, -Pharmabot: A

Pediatric Generic Medicine Consultant Chatbot —,Journal of Automation and Control Engineering Vol. 3, No. 2, April 2015.

9. Gillian Cameron, David Cameron, Gavin Megaw, Raymond Bond, Maurice Mulvenna, Siobhan O'Neill, Cherie Armour, Michael McTear,—Towards a chatbot for digital counselling, Journal of Medical Internet Research, 4(1), 2020
10. Dinesh Kalla, Fnu Samaah, " Chatbot for Medical Treatment using NLTK Lib" IOSR Journal of Computer Engineering IOSR-JCE, Volume 22, Issue 1, Feb -2020
11. Ashwini Shangrapawar, Ankita Ravekar, " Artificial Intelligence based Healthcare Chatbot System", IRJET Vol:07, Issue-02:Feb-2020..
12. Mahajan, Papiya, et al. "Healthcare chatbot using natural language processing." 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions). 2020.
13. Mrs Rashmi Dharwadkar¹, Dr.Mrs. Neeta A. Deshpande, A Medical ChatBot, International Journal of Computer Trends and Technology (IJCTT) – Volume 60 Issue 1-June 2018 [8] N-gram Accuracy Analysis in the Method of Chatbot Response, International Journal of Engineering & Technology. (2018)

•

