# WeCare – A Medical Chatbot

**A PROJECT REPORT**

***Submitted by***

**Madhav Singh Rawat- 19BCS35413**

***in partial fulfillment for the award of the degree of***

## Bachelor of Engineering

**IN**

CSE- Information Security



**Chandigarh University**

May 2022



## BONAFIDE CERTIFICATE

Certified that this project report **“WeCare – A Medical Chatbot”** is the bonafide work of “**Madhav Singh Rawat”** who carried out the project work under my/our supervision.

**SIGNATURE SIGNATURE**

**SUPERVISOR HEAD OF THE DEPARTMENT**

Submitted for the project viva-voce examination held on

##### INTERNAL EXAMINER EXTERNAL EXAMINER

## 

**Table of Contents**

TITLE PAGE i

BONAFIDE CERTIFICATE ii

ABSTRACT iv

LIST OF TABLES v

TIMELINE/GANTT CHART vi

1. **INTRODUCTION 1**

Problem Definition 1

Project Overview/Specifications 1

Hardware Specification 2

Software Specification 2

1. **LITERATURE SURVEY 3**

Existing System 3

Proposed System 4

Literature Review Summary 5

1. **DESIGN FLOW/PROCESS 6**

**Proposed Model 7**

**Methodology Used 8**

**\**

**ABSTRACT**

To easily detect diseases, we are designing a Chatbot system, this work explores prediction of diseases by taking various symptoms related to a disease. For this purpose, we use different datasets, we apply various Machine Learning classification and ensemble Techniques to predict diseases. Machine Learning is a method that is used to train computers or machines explicitly. Various machine learning Techniques provide efficient result to collect knowledge by building various classification and ensemble models from collected dataset. Such collected data can be useful to predict diseases. Various techniques of Machine Learning can capable to do prediction, however it’s tough to choose best technique.

***List of Tables***

### *Figure Title page*

### *Table 1 Gantt Chart v*

### *Table 2.1 Literature Review Summary 5*

### Table 1 Gantt Chart

Gantt Chart - WeCare

# 1 INTRODUCTION

**1.1 PROBLEM DEFINITION:** The 21st century has now become a century of hustle and fast-paced life, where people are hardly taking time for oneself. Most of the humans are not living a healthy lifestyle i.e., not eating proper diet, not having enough sleep hours and moreover environmental conditions are making it worse. These factors are making human’s body prone to many diseases such as diabetes, cancer, heart attacks, etc. Therefore, it is important to keep a check on your health on a weekly/monthly basis to avoid a potential life-threatening conditions or diseases at an early stage. Regular checkups also reduce risk of getting sick and increases chances for treatment.

**1.2 PROJECT OVERVIEW/SPECIFICATIONS:** To easily detect diseases we are designing a prediction system; this work explores prediction of diseases by taking various symptoms related to a disease. For this purpose, we use different datasets, we apply various Machine Learning classification and ensemble Techniques to predict diseases. Machine Learning is a method that is used to train computers or machines explicitly. Various machine learning Techniques provide efficient result to collect knowledge by building various classification and ensemble models from collected dataset. Such collected data can be useful to predict diseases. Various techniques of Machine Learning can capable to do prediction, however it’s tough to choose best technique.

## 1.3 HARDWARE SPECIFICATIONS

* RAM – 2 GB or more
* i3 or i5 Intel Core Processor

**1.4 SOFTWARE SPECIFICATIONS**

* Windows 10 x64
* Google Colaboratory/ Jupyter Notebook
* Python 3.9
* MySQL

# 2 LITERATURE REVIEW

**2.1 Existing System:**

Krishnendu Rarhi et al. [1] provides an idea for a medical chatbot using AIML (Artificial Intelligence Mark-up Language), based on XML. To use the proposed system, the user has to provide the preliminary symptoms faced by them. The system then collects keywords and shortlist the diseases. The design also has a concept of threshold level that helps to detect the intensity of a particular disease. It also provides the facility to connect directly with a doctor if the problem is serious.

Another project by Mrs. Asha Rani K P et al [2] focuses on a medical chatbot using a machine learning algorithm i.e., k-nearest algorithm (KNN). The implementation of the system includes NLP and NLTK. The proposed system is a web application and has options for both online and offline consultancy. If a doctor is online, the user can directly consult with him else user can use an automated chatbot.

The major drawback of the previous systems is that they do not provide nor store the user reports for future reference.

**2.2 Proposed System:**

Due to the recent advancements in the computer science field, there are several studies showing us that diseases can be accurately predicted beforehand by using machine learning algorithms and making a proper generated result using the persons previous medical records. This system will not only help us rule out and predict an outcome which is correct and useful, but also help us identify early symptom and rectify the problem occurring before handedly and treat the patient before he/she reaches a chronic state of the disease.

The following are the advantages of the proposed system:

* Automation along with self-reporting system
* Cost- effective
* Increased security
* Time saving
* Easy to manage

## Literature Review Summary

Table 2.1: Literature review summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Year and citation** | **Article Title** | **Purpose of the study** | **Tools/ Software used** | **Source (Journal/ Conference)** |
| 2017 | Automated Medical Chatbot | Medical Chatbot that tells threshold. | XML, AIML | ResearchGate |
| 2021 | Medical Chatbot for Disease  Prediction using  Machine Learning | Chatbot that provides both online and offline consultancy. | KNN, NLP, NLTK | IJARESM |
| 2019 | A Personalized  Medical Assistant  Chatbot: MediBot  [3] | Chatbot that takes input as speech only. | Google Speech Recognition API | IJSTE |
| 2019 | EMERGENCY  PATIENT CARE  SYSTEM USING CHATBOT [4] | Option for SOS for emergency services. | Google Map API | IJTRE |
| 2020 | Doctor Chatbot:  Heart Disease  Prediction System [5] | SVM, SNN | Dialogflow | ResearchGate |

**3 DESIGN FLOW/PROCESS**

**3.1 Proposed Model**

There are many medical chatbots proposed in the past but they do have certain drawbacks. The major drawback that most of the chatbots are having is that they neither store any information about the patient nor they store the diagnosis provided by the respective chatbot. The previous systems also don’t tell the users about the doctors who come for online check-ups.

To overcome these limitations of the existing system, the proposed system stores all the personal details of the user in the database and it also stores the diagnosis provided by the chatbot so that it can be used in future while consulting a doctor. User can also save the chat with the date and time mentioned in the file. It also recommends the user some online websites, where doctors come on the daily basis for regular check-ups. The system is built using a set of large databases which makes it more efficient as compared to the existing systems. Moreover, it is cost efficient and also saves a lot of time required for the preliminary diagnosis of the disease. By using this system, a user can know about most probable health issue they are facing by just knowing the symptoms of the disease.

The chatbot application will converse with the user in the same way humans do. The user needs to first register themselves on the application (or login with the credentials used while registering). By using text chat, a user needs to answer/ tell some symptoms which he/she is facing to the system. For the smooth conduct, some predefined questions and answers are stored in the database. The chat processing in the system is done through Natural Language Processing (NLP). The chatbot after receiving an input from a user, will try to diagnose the disease using the database where it already trained. After the diagnosis, a probable health issue is told to the user with some online websites link, where doctors come for online check-ups and we can take advice from them. The figure below shows the working of the chatbot –

Figure: Working of Chatbot

Working of the System:

A. Registration/ Login

The users using the chatbot for the very first time needs to register themselves by providing personal details, and login credentials to login in future. All the details of the user are stored in MySQL database. The users who have already registered them just need to enter their correct login credentials to use the system. The system will then check the login credentials with that that of present in the MySQL database. If the credentials are correct then the system will redirect the user to chatbot else it will show an error. [12]

B. Medical Chatbot

The user gets the access to medical chatbot, which is trained using a TensorFlow and Keras. TensorFlow is an open-source library available free of cost. It is basically used to implement machine learning and artificial intelligence. The medical chatbot contains two major phases namely symptom extraction and diagnosis which is based on the symptoms. In the first phase, the system enters the loop to get the details of symptoms faced by the user. The system then replies to the user using different modules in which it is trained. Therefore, the system shows the probable health issue a user may be facing.

C. Doctor Suggestion

Based on the outcome i.e., a probable disease a user might be facing, some websites (where doctors come for online check-ups) will be recommended to the user, which enables the user to contact the doctors for consultation.

D. Previous Record

The medical chatbot after showing the preliminary diagnosis to the user, saves the probable disease in a database. A user can view it anytime he/she wants to. This can also be helpful while consulting a doctor.

**3.2 METHODOLOGY USED**

## Natural Language Processing:

The extension of NLP function is quite tough because all the computer systems need humans to interact with them in a certain programming language. But human language contains a lot of composite variables which makes it very less accurate. To solve this problem, NLP is used by computers to analyze, understand, and derive meaning from human language in a useful way [13]. Therefore, NLP permits a user to ask queries from the system. By using NLP, system understands important elements from user’s text that can be related to information in a dataset and thus provides with an answer. The stored information in the datasets include greeting messages, user’s details and symptoms related to a particular disease on the basis of which we predict the disease.

To implement Natural Language Processing, most commonly Natural Language Toolkit (NLTK) is used. NLTK is an open-source library for python programming language that work with human language data. Some steps to implement NLP in chatbot –

*a. Tokenization:* NLTK divides the large quantity of text into words called tokens. This is the base step to implement NLP. These tokens are further used to match the pattern.

Example- “I am feeling feverish” is divided into [‘I’, ‘am’, ‘feeling, ‘feverish’]

*b. Lemmatization:* It is the process in which words (tokens made in tokenization) are converted to its meaningful base forms. In this chatbot, we have used wordnet lemmatizer. [11]

Wordnet is a large and freely available lexical database for English language.

Example – Tokens made above will be converted as [‘feel’, ‘fever’]

*c. Pattern Matching:* After all the words are lemmatized, the symptoms are matched with the information present in the dataset and accordingly output is generated and stored.

This concludes the creation of training data in our chatbot. To further implement our chatbot, we created models using Keras.

## Keras:

It is basically an open-source library that provides an interface for both artificial neural networks (Used in deep learning) in python and TensorFlow library. [14,15] To implement the chatbot, we made models, defined different layers and optimizer. Therefore, Keras compiles our model with optimizer functions and training process with fit function. [16] The implementation of the said is done in the following steps:

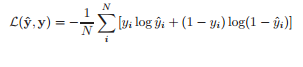
*a) Create model -*

We create a model of 3 layers to implement the chatbot. First layer is of 128 neurons, second layer is of 64 neurons and the third which is also the last layer contains the number of neurons which is equal to the number of intents to produce output intent. The first and the second layer is activated using Rectified Linear Unit (ReLu) which returns 0 in case of negative input and returns value itself if it is a positive input [9]. The third layer is activated using softmax which converts a vector of values to a probability distribution. This makes it easier to interpret the result on basis of their probability. [10]

*b) Compile model –*

We have used Stochastic Gradient Descent (SGD) as an optimizer in the chatbot. Keras provides SGD class that implements stochastic gradient descent optimizer with a learning rate and momentum. In this chatbot, we have applied learning rate as 0.01, momentum as 0.9 and decay rate as 1e-6 (0.000001).

To compile the model, we have used categorical cross entropy loss function which is used to evaluate the quantity that the model should seek to minimize during the training part. Since we have converted the values to a probability distribution while creating a model, the best loss function to use is categorical cross entropy. Mathematical formula for cross entropy –



Mathematical formula to calculate cross entropy

*c) Fitting and saving model –*

For fitting the model, we have used 200 epochs. An epoch is basically an iteration over the entire data. We have also specified batch size as 5 which means that 5 samples are taken per gradient update. Verbosity mode is also given as 1 which includes both progress bar and one line per epoch.



Progress line per epoch

This is the screenshot of the system training the project to work properly, so that it can interact with humans easily.

To validate the working of the model, Term Detection Test by General Word Percentage approach is used.

By using General Word Percentage approach, we can get a ratio of number of unrelated words used over the total number of words in a particular message. [17] This approach helps us to understand about how our Chatbot could deal with unrelated words that are mixed with medical terminologies. We would take some sample corpus to proceed with the analysis. We would check whether our Chatbot is able to detect medical terminologies for our sample corpus.

Sample corpus for analysis:

1. Hey Chatbot! I am not able to handle this headache and I feel like vomiting. My digestion is also not okay and feeling lazy.

2. Can we cure brain cancer? I think I am suffering from it.

3. I think I have cancer. I also feel like vomiting since yesterday. I want medicines for typhoid.

4. I am feeling good today! But I am having a headache.

5. I am feeling like nausea since yesterday.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Corpus No.** | **GWP Ratio** | **Terms Present** | **Terms Detected** | **Term Detection Ratio** |
| 1 | 0.833 | 4 | 3 | 0.75 |
| 2 | 0.916 | 1 | 0 | 0.00 |
| 3 | 0.823 | 3 | 3 | 1.00 |
| 4 | 0.909 | 1 | 2 | 0.50 |
| 5 | 0.857 | 1 | 1 | 1.00 |
| Average | 0.867 | 2 | 1.8 | 0.65 |

Table: Illustrating Term Detection Ratio

*Calculations Taken:*

GWP Ratio – Number of unrelated medical terms/ Number of total words in corpus

Term Detection Ratio – Terms Detected/ Terms Present

Figure: Visual Representation of GWP Ratio

We notice that with an average 86.7% of the unrelated terms present in our sample corpus, our Chatbot performed Terminology detection with 65% of accuracy. We can observe through the above table (Table 2) and Figure 1 that with the decrease in GWP ratio and the increase in the terms present in a message, our Chatbot performed efficiently and was successfully able to detect correct medical terminologies from the corpus. However, the efficiency of terminology detection would also depend on the way users type and preciseness of the messages.