IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

# Ai Healthcare Bot System Using Python

1.T.Aruna Jyothi, 2Mohammad Abdul Rahaman, 3 Komepalli Ajay,4.Aravind Anugula
1Assisant Professor, 2Student, 3.Student,4.Student
1Department of Information Technology
Matrusri Engineering College, Hyderabad

**Abstract**: To lead a fulfilling life, access to healthcare is crucial. However, obtaining a doctor's consultation for health issues can often be challenging. To address this, the idea is to develop an AI-powered medical chat bot that can diagnose diseases and provide basic information about them before a doctor's visit. This approach aims to reduce healthcare costs and improve access to medical knowledge. Virtual assistants can assist patients and healthcare providers with a variety of medical-related tasks, enhancing the overall healthcare experience. Chat bots are software applications created to engage in conversations with individuals, offering assistance through text messages, applications, or instant messaging platforms. These bots can identify symptoms and provide diagnoses based on the identified symptoms. The methodology incorporates natural language processing (NLP) techniques to understand user queries, machine learning algorithms to predict diagnoses, and integration with healthcare databases to ensure accurate responses. This system shows promising results in both accuracy and efficiency, highlighting its potential to improve healthcare accessibility and effectiveness.

***Index Terms*** - Artificial Intelligence, Healthcare, Python, Chat bot, Natural Language Processing, Machine Learning, Diagnosis.

## I..INTRODUCTION

AI in healthcare refers to the application of artificial intelligence technologies within the healthcare sector to enhance patient care and outcomes. This involves creating and implementing intelligent systems capable of performing tasks typically carried out by humans, such as diagnosing illnesses, analyzing medical images, and offering personalized treatment suggestions.

Healthcare is essential in our lives, yet many people are preoccupied with their jobs and are often engrossed in online activities, neglecting their health. As a result, they tend to avoid visiting the hospital for minor issues. To address this, we propose developing a healthcare chat bot system using Python, NLP, and machine learning algorithms. This system will identify illnesses and provide detailed information about them before a doctor's consultation. It will help patients better understand their conditions and take steps to improve their health.

## II.LITERATURE SURVEY

[1] The aim of this paper is to enhance intelligent treatment using Machine Learning technology to streamline the decision support system. It comprehensively addresses the diagnosis of heart disease by monitoring an individual's heartbeat. The framework allows users to set pulse rate parameters. Once these limits are set, the system monitors the heartbeat and alerts the individual whenever their pulse exceeds a certain level, indicating a high pulse rate and the risk of a heart attack. Authors Ahmed
M . Alaa and Senthil Kumar Mohan experimented with a combination of different factors and achieved an 88.7% accuracy rate using a random hybrid forest.

[2] This paper focuses on classic supervised binary classification, utilizing various attributes in the data set. The data set includes plasma glucose concentration, blood pressure (mmHg), body mass index, age (years), and more. Several elements, each with specific features, are used to identify individuals affected by the

disease. To address the problem, the data must be analyzed, necessary adjustments made, an ML algorithm applied, a model trained, its performance evaluated, and various algorithms tested to achieve the most accurate results.

In developing software or websites, it is crucial to identify system requirements by accurately gathering expected data to facilitate communication between providers and customers.

[3] This paper emphasizes the need to develop a framework that simplifies disease prediction for end-users without requiring a doctor or specialist visit. It effectively identifies various diseases by analyzing patient symptoms using different Machine Learning models. This section of the paper presents the accuracy results of various algorithms, such as Decision Tree (DT) with 90.2% accuracy, Random Forest (RF) with 95.28% accuracy, and Naive Bayes (NB) with 88.08% accuracy. The paper also explains how advancements in technology within the health industry provide solutions for patients by offering recommendations from specialists and facilities, guiding them on where to seek treatment and which expert to consult for specific diseases. The healthcare industry collects data from patient databases by applying data mining and Machine Learning techniques.

This paper offers a heart disease prognosis using supervised learning algorithms, including SVM, KNN, and Naive Bayes. The data set contains 3000 objects with 14 features. From extensive literature review, it was observed that most research utilized a disease data set containing only

303 objects with 14 features. Naive Bayes produced the best results, with high accuracy (86.6%) and quick processing time, while Decision Tree achieved an accuracy of 78.69%, and KNN achieved 77.85% accuracy.

## III.METHODOLOGY

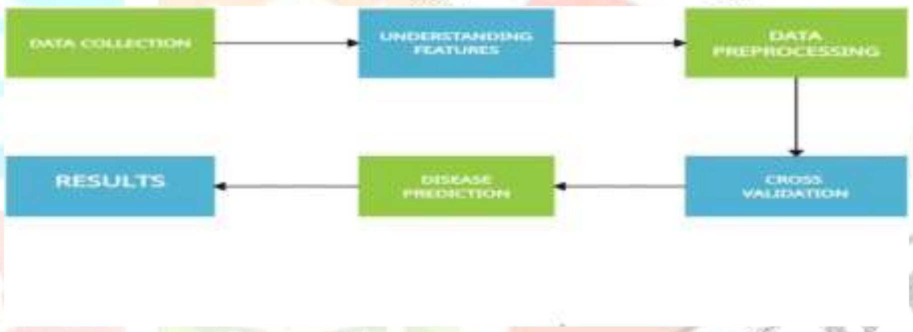


Fig 1: Flow chart for methodology

### A. Collection of Data Sets

In this stage, two datasets are used as input: one for multiple symptoms-based disease prediction and another text file data set for the chat bot application with question and answer pairs.

### B. *U*nderstanding Features of Data Sets

In the early lifestyle disease data set, various features are taken as input, with each disease classified based on the type of feature (status 1 or 0) and the disease name used as the label. For the chat bot application, the text file data set includes features as questions and labels as answers.

### C. Pre- processing the Data

At this stage, the disease data set is processed to extract features and labels, while the chat bot text file data set is pre-processed using NLP techniques.

### D. Splitting Data into Training and Testing Datasets

The data set is divided into two parts using the test- train split function, with 80% as the training set and 20% as the testing set. Training features are referred to as train_x and labels as train_y. These values are used to train the algorithm, and the test data is used to evaluate the accuracy of each disease data set.

### E. *Applying the ML Decision Tree Algorithm to the Data set*

At this stage, the pre- processed disease data set is used as input, and the trained features and labels are provided to the fit function to train the model. Through the web application, users can select symptoms and receive a predicted disease diagnosis.

### F. *Accuracy Results*

After training, the test set is input into the algorithm to assess the dataset's accuracy. The accuracy results from the testing phase are crucial for validating the effectiveness and utility of the developed AI healthcare bot system in aiding patient consultation and disease prediction.

### *Module Description:*

1. **Automated Disease Prediction through Machine Learning:**

This system employs machine learning algorithms such as random forest, decision tree, and logistic regression for automatic disease prediction. The medical data is trained using all three algorithms, and the best-performing algorithm is selected as the primary model for final predictions.

2. ***Chat bot Service:***

Through this service, users can register on the web application and utilize a chat bot to receive automatic responses generated from trained question-and-answer data. Natural language processing (NLP) techniques are employed for this purpose.

3. ***Online Doctor Booking Service:***

This service enables users to book appointments with doctors based on predicted diseases obtained from machine learning algorithms using input symptoms. Users can schedule appointments according to available timings and receive confirmation from the doctor.

4. ***Online Medicine Ordering Service:***

Users have the option to purchase medicines from an online store, add products to their cart, and proceed with payment using this service.

## IV.SYSTEM ARCHITECTURE

A System architecture is a conceptual model that defines the structure, behavior, and various views of a system. An architecture description is a formal representation of a system, organized to support reasoning about its structures and behaviors.

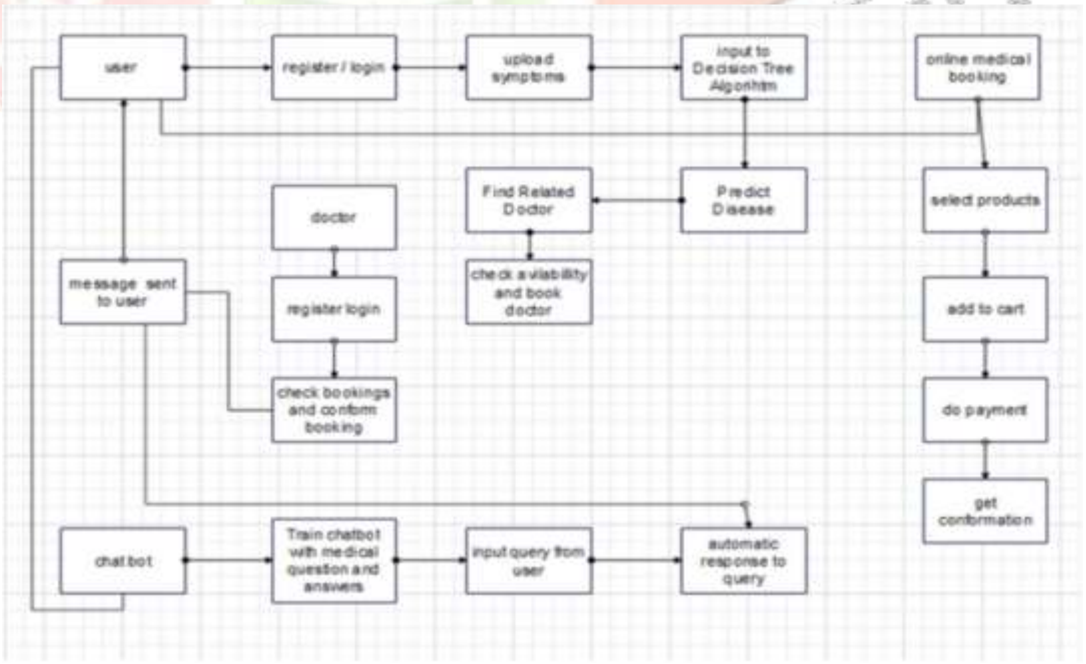


**Figure 2: System Architecture**

**3-Tier Architecture:**

The three-tier software architecture, which emerged in the 1990s, addresses the limitations of the two-tier architecture. This architecture introduces a middle tier (server) between the user interface (client) and data management (server) components. The middle tier handles process management, where business logic and rules are executed, and can support hundreds of users (compared to only 100 users in a two-tier architecture) by offering functions such as queuing, application execution, and database staging.

The three-tier architecture is employed in scenarios where an efficient distributed client/server design is required. Compared to the two-tier architecture, it offers enhanced performance, flexibility, maintainability, reusability, and scalability. Additionally, it conceals the intricacies of distributed processing from the user. These attributes have contributed to the widespread adoption of three-layer architectures, particularly in Internet applications and net- centric information systems.

***Advantages of Three-Tier:***

- Separates functionality from presentation.
- Clear separation – better understanding.
- Changes limited to well define components.
- Can be running on WWW.
- Effective network performance

## V.ANALYSIS OF RESULT

Once our model is trained, it can provide information on diseases, analgesics, dietary recommendations, and nearby doctor details based on symptoms. However, we lack evidence to verify that our healthcare chat bot has higher accuracy compared to others, as we lack parameters for measuring its performance against others.

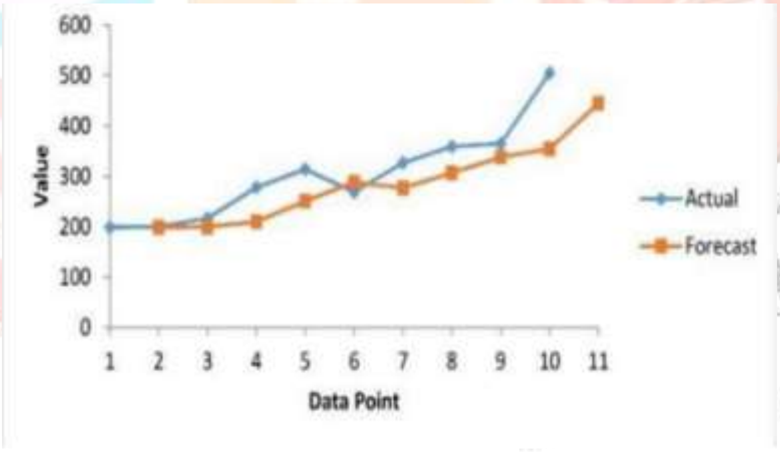
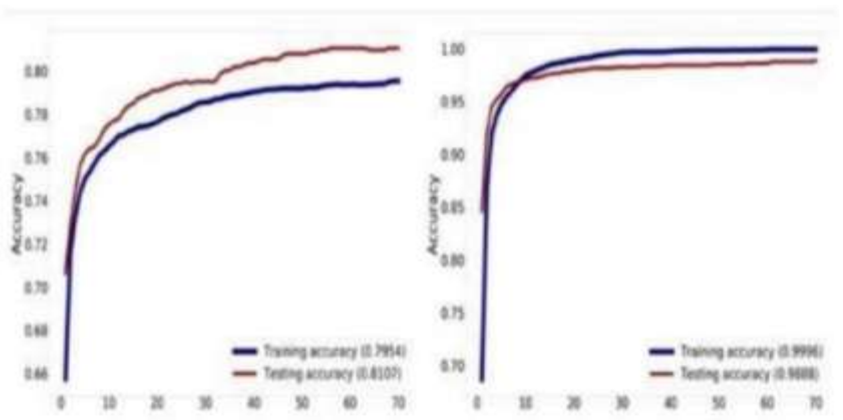


Figure 3: Classification of report

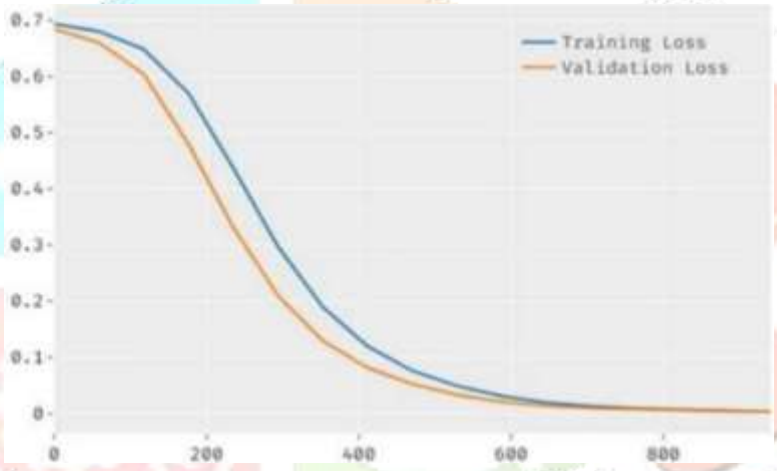
*Loss in Validation and Training*

*Accuracy in Training and Validation*

This figure illustrates the validation loss (depicted by the orange line) and the training loss (represented by the blue line) of our model.



The above figure contains the graph depiction between the training and testing data accuracy.



*Experimental Setup:*

**Train Dataset:**

This data set comprises a sample of data utilized to train the model. This process involves feature extraction and training the model. The observations within the training set provide the experience from which the algorithm learns. In supervised learning scenarios, each observation includes both an observed output variable and one or more observed input variables.

*Validation Data set:*

The validation data set serves as a sample of data used for unbiased evaluation of a model trained on the training data set. However, the evaluation can become more biased if the model configuration incorporates skills learned from the validation data set.

*Test data set:*

The test data set is a sample of data employed to provide an unbiased evaluation of a finalized model. Its purpose is to assess the accuracy of predictions made by the model. The test set comprises observations utilized to evaluate the model's performance using specific performance metrics. It's crucial to ensure that no observations from the training set are included in the test set. If the test set does contain examples from the training set, it becomes challenging to determine whether the algorithm has learned to generalize from the training data or has merely memorized it.

***Parameters given to model for training: Batch size:***

The batch size parameter should be an integer or None. It represents the number of samples used per gradient update. If not specified, the batch size defaults to 32. It's important not to specify the batch size if the data is in the form of datasets, generators, or instances of Keras .utils.Sequence.

***Epochs:***

The "epochs" parameter should be an integer indicating the number of epochs to train the model. An epoch refers to an iteration over the entire data set provided as input (both input features "x" and target labels "y").

It's important to note that when used in conjunction with the "initial_epoch" parameter, "epochs" should be interpreted as the "final epoch." This means that the model is trained until the specified epoch index is reached, rather than for a specific number of iterations.

***Verbose:***

The "verbosity" parameter can take the values 'auto', 0, 1, or 2, determining the level of output verbosity during training. A setting of 0 indicates silent mode, while 1 displays a progress bar. Meanwhile, setting it to 2 results in one line of output per epoch. In most cases, 'auto' defaults to 1, but switches to 2 when used with the Parameter Server Strategy. It's worth noting that the progress bar might not be useful when logging to a file; hence, setting verbose=2 is recommended for non-interactive environments like production setups.

***Validation Split:***

A float value between 0 and 1 represents the fraction of the training data reserved for validation purposes. This fraction of the training data is set aside and not used for training. Instead, the model evaluates the loss and any defined metrics on this data at the end of each epoch. In this context, 20% of the training data set is allocated as the validation data set, ensuring that the model, trained on 80% of the training data set, is effectively validated.

***Model Evaluation:***

Model evaluation is a vital aspect of the model development process, enabling the identification of the most suitable model for accurately representing the data and making reliable predictions for the future. It's widely recognized in data science that assessing model performance with the same data used for training is not acceptable. This practice often results in the creation of overly optimistic and over fitted models, which perform poorly when presented with new data. Hence, it's essential to employ separate datasets for training and evaluation to ensure the credibility and effectiveness of the selected model.

***Evaluation Metrics and Result:***

Model evaluation metrics are necessary to measure the performance of a model accurately. The selection of appropriate evaluation metrics depends on the specific machine learning task.

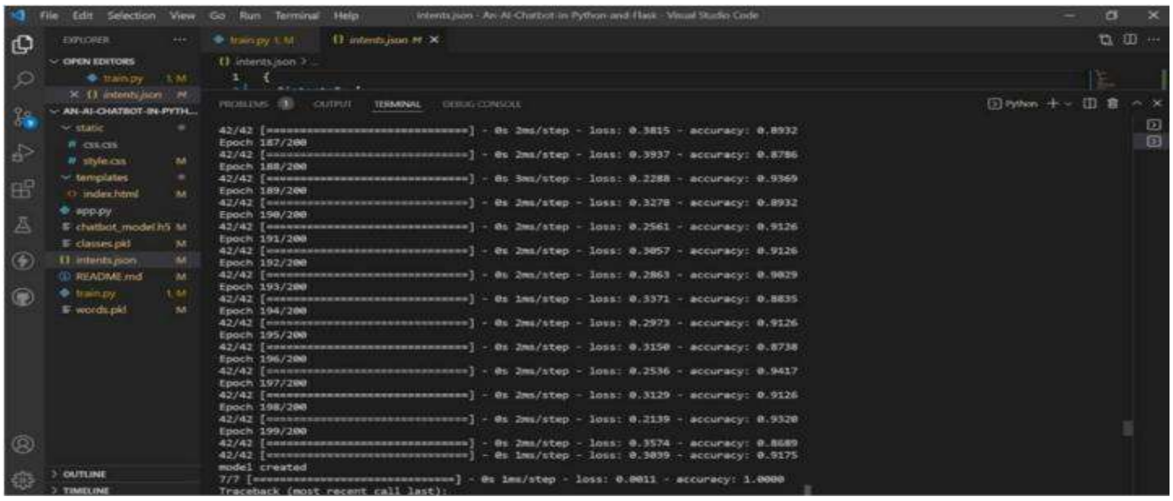
In this project evaluation metrics are:

1) Accuracy
2) Loss

Sample Code for Evaluating Model model.evaluate(np.array(train_x), np.array(train_y))

As a result of evaluation using model.evaluate() method,

1- Accuracy of the model is depicted as 1.0000
2- Loss of the model is depicted as 0.00073455



```
42/42 [==============================] - 0s 2ms/step - loss: 0.3815 - accuracy: 0.8932
Epoch 187/200
42/42 [==============================] - 0s 2ms/step - loss: 0.3937 - accuracy: 0.8786
Epoch 188/200
42/42 [==============================] - 0s 3ms/step - loss: 0.2288 - accuracy: 0.9369
Epoch 189/200
42/42 [==============================] - 0s 2ms/step - loss: 0.3278 - accuracy: 0.8932
Epoch 190/200
42/42 [==============================] - 0s 2ms/step - loss: 0.2561 - accuracy: 0.9126
Epoch 191/200
42/42 [==============================] - 0s 2ms/step - loss: 0.3057 - accuracy: 0.9126
Epoch 192/200
42/42 [==============================] - 0s 2ms/step - loss: 0.2863 - accuracy: 0.9029
Epoch 193/200
42/42 [==============================] - 0s 1ms/step - loss: 0.3371 - accuracy: 0.8835
Epoch 194/200
42/42 [==============================] - 0s 2ms/step - loss: 0.2973 - accuracy: 0.9126
Epoch 195/200
42/42 [==============================] - 0s 2ms/step - loss: 0.3150 - accuracy: 0.8738
Epoch 196/200
42/42 [==============================] - 0s 2ms/step - loss: 0.2536 - accuracy: 0.9417
Epoch 197/200
42/42 [==============================] - 0s 2ms/step - loss: 0.3129 - accuracy: 0.9126
Epoch 198/200
42/42 [==============================] - 0s 2ms/step - loss: 0.2139 - accuracy: 0.9320
Epoch 199/200
42/42 [==============================] - 0s 2ms/step - loss: 0.3574 - accuracy: 0.8689
42/42 [==============================] - 0s 1ms/step - loss: 0.3039 - accuracy: 0.9175
model created
7/7 [==============================] - 0s 1ms/step - loss: 0.0011 - accuracy: 1.0000
Traceback (most recent call last):
```

## OUTPUTS

Figure 5: Home Screen Interface

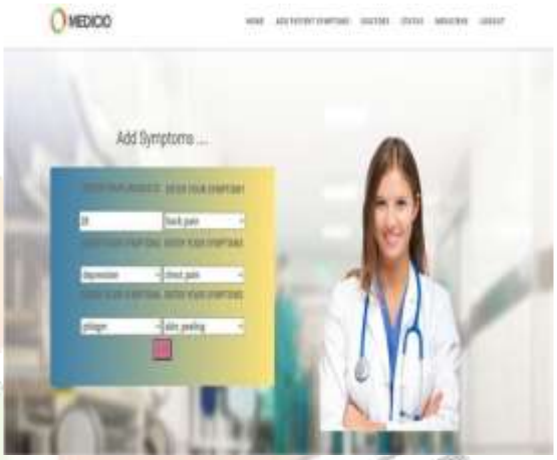
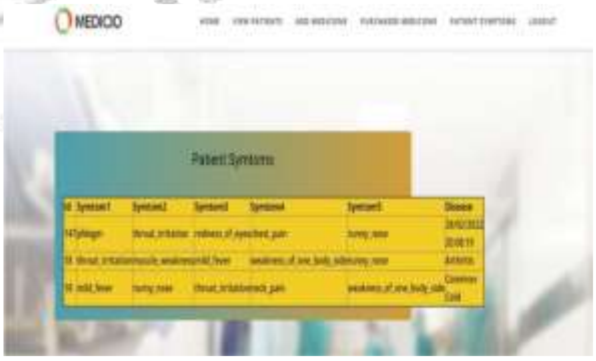


Figure 6: Patient Purchased Medicine

| Tablet | Price | Patient | Id | Phone |
|---|---|---|---|---|
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

Figure 7:Patient Login Interface



| Id | Symptom1 | Symptom2 | Symptom3 | Symptom4 | Symptom5 | Disease |
|---|---|---|---|---|---|---|
| 14 | phlegm | throat_irritation | redness_of_eyes | chest_pain | runny_nose | [illegible] |
| 18 | throat_irritation | muscle_weakness | mild_fever | weakness_of_one_body_side | runny_nose | Arthritis |
| 10 | mild_fever | runny_nose | throat_irritation | neck_pain | weakness_of_one_body_side | Common Cold |

Figure 8: Patient Symptoms UI

Figure 9: Chat bot Interface

## VI. CONCLUSION

We have successfully developed a machine learning algorithm capable of accurately predicting diseases based on input of six symptoms. Additionally, our application assists customers in booking appointments and purchasing medicine, as well as facilitating contact with doctors. Our application integrates a machine learning-based medical assistant that utilizes various algorithms to predict diseases. The most accurate algorithm is incorporated into a Flask web framework, allowing us to design a health website encompassing doctor appointment booking, chat bot assistance, medicine ordering, and disease prediction services all in one platform. In today's busy world, many individuals neglect to seek medical advice when unwell. Implementing a chat bot can address this issue by enabling users to diagnose their ailments without consulting a doctor directly. Acting as a virtual doctor, the chat bot analyzes users' symptoms and recommends appropriate healthcare steps. Our datasets contain comprehensive information on diseases and corresponding healthcare measures.