CHATBOT FOR HEALTHCARE: DIAGNOSIS OF ACUTE DISEASE IN VILLAGES AND SMALLER TOWNS USING AI

A PROJECT REPORT

Submitted by,

M Mohammad Yasin – 20201CAI0034
Yashaswini H – 20201CAI0027
Bodanapu Teja – 20201CAI0026
Rama Mohan Raja N V – 20201CAI0037

Under the guidance of,

Dr. Madhura K

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING
(Artificial Intelligence & Machine Learning)

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"CHATBOT FOR HEALTHCARE: DIAGNOSIS OF ACUTE DISEASE IN VILLAGES AND SMALLER TOWNS USING AI"** being submitted by M MOHAMMAD YASIN, YASHASWINI H, BODANAPU TEJA, RAMA MOHAN RAJA NV bearing roll number(s) 20201CAI0034, 20201CAI0027, 20201CAI0026, 20201CAI0037 in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science & Engineering (Artificial Intelligence & Machine Learning) is a bonafide work carried out under my supervision.

K. Madhura 11/1/24

Dr. MADHURA K
Asst Professor
School of CSE&IS
Presidency University

11/01/24

Dr. Zafar Ali Khan
Associate professor, HoD
School of CSE&IS
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

11/1/24

Dr. L. SHAKKEERA
Associate Dean
School of CSE&IS
Presidency University

Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **CHATBOT FOR HEALTHCARE: DIAGNOSIS OF ACUTE DISEASE IN VILLAGES AND SMALLER TOWNS USING AI** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering ,** is a record of our own investigations carried under the guidance of **Dr. Madhura K, Assistant Professor, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name(s)	**Roll No**
M Mohammad Yasin	**20201CAI0034**
Yashaswini H	**20201CAI0027**
Bodanapu Teja	**20201CAI0026**
Rama Mohan Raja N V	**20201CAI0037**

ABSTRACT

The Symptom-Based Disease Prediction Chatbot, designed with a client-server architecture, uses the K-Nearest Neighbors (KNN) algorithm to achieve 95% accuracy in disease predictions. Developed in Python with Flask, the user-friendly chatbot prompts users for symptoms through Natural Language Processing (NLP) techniques. It delivers disease predictions, precautions, and recommendations in a web-based interface, targeting individuals in areas with limited healthcare access. The systematic timeline covers data collection, preprocessing, model training, chatbot development, integration, and testing. Real-world deployment demonstrates the chatbot's timely and accurate predictions, supported by continuous monitoring and maintenance, comprehensive documentation, and user training. Outcomes include accurate predictions, user-friendly design, effective NLP, timely precautions, and successful real-world deployment. The KNN algorithm stands out, and the chatbot showcases engaging interactions and seamless integration. In conclusion, the chatbot democratizes healthcare, empowering users for proactive well-being, representing an innovative step in virtual healthcare companionship. Future enhancements aim to further expand its impact

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science and Engineering and School of Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. C. Kalaiarasan** and **Dr. Shakkeera L**, School of Computer Science and Engineering and School of Information Science, Presidency University and **Dr. Zafar Ali Khan**, Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Dr. Madhura K**, Assistant Professor, School of Computer Science and Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and the department Project Coordinator **Dr. Murali Parameswaran**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project

M Mohammad Yasin

Yashaswini H

Bodanapu teja

Rama Mohan Raja N V

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

In the contemporary scene of medical care, the determined test of getting to opportune and exact sickness finding has become progressively obvious, particularly in far off regions wrestling with a shortage of clinical experts. In light of this major problem, our undertaking sets out on the improvement of a weighty arrangement — a Symptom Based Disease Prediction Chatbot using AI. This creative chatbot fills in as a virtual diagnostician, changing the manner in which people look for fundamental medical care data by anticipating potential sicknesses in light of client inputted side effects and prescribing essential tests to affirm analyze.

Customarily, getting a clinical conclusion frequently involves genuinely visiting a medical services proficient. In any case, this cycle is hindered by different variables, for example, the lack of specialists in distant regions or monetary limitations related with conference expenses. The Sickness Expectation Chatbot arises as an imposing reaction to this issue, furnishing clients with a promptly open and savvy elective for getting fundamental data about their ailments.

A basic issue our chatbot addresses is the absence of open medical care in distant areas, where people might find it trying to immediately talk with a specialist when confronted with side effects of a disease. Utilizing AI, our chatbot offers a fundamental sickness forecast in view of client inputted side effects, subsequently filling a vital hole in medical care openness. The use of the K-Closest Neighbors (KNN) calculation assumes a significant part in this cycle, permitting the recognizable proof of examples in past cases and upgrading forecast precision through a thoroughly prepared model.

The inspiration driving this task is established in the worldwide reality that various districts need appropriate admittance to medical services offices, with far off regions confronting huge difficulties in giving convenient clinical help. By bridling innovation, our Infection Forecast Chatbot expects to expand the span of medical care administrations, offering a fundamental finding to the individuals who might not have quick admittance to a specialist.

While existing arrangements might give some help, a prominent hole exists in client studies with genuine patients. Our chatbot tries to overcome this issue by integrating client input on convenience and worth, guaranteeing a more complete assessment. Also, most procedures in illness expectation are assessed on normalized datasets, justifying further examination on genuine world clinical datasets. Our chatbot, intended for exact infection expectation in view of client side effects, addresses this requirement for a more customized and solid methodology.

AI's importance in assorted fields is deeply grounded, and its application to medical care holds tremendous potential. For compelling illness expectation, the model should be prepared with exact information. In our venture, we utilize the K-Closest Neighbors (KNN) calculation, a managed AI strategy, to upgrade the precision of illness expectations.

The overall objective of our task is to democratize admittance to exact medical services data. By giving an easy to understand Illness Expectation Chatbot with high exactness, we plan to guarantee that no individual goes untreated for absence of an underlying determination. This not just addresses the quick wellbeing worries of clients yet in addition adds to the general improvement of general wellbeing.

The effective execution of our Sickness Forecast Chatbot holds the commitment of lightening the stress on medical care experts, especially in occupied clinics. By foreseeing intense normal infections that can be actually treated with appropriate consideration and client safety measures, the chatbot plans to decrease the weight on specialists, empowering them to zero in on additional mind boggling cases.

The essential interest group for our Sickness Expectation Chatbot incorporates people dwelling in far off regions who need advantageous admittance to medical care administrations. By coming to these underserved populaces, the chatbot endeavors to be a significant device in the early identification of illnesses and the advancement of preventive medical services measures.

The development of our venture lies in the consistent combination of AI with an easy to understand chatbot interface. By using the KNN calculation for illness expectation, we consolidate the prescient capacities of AI with the intelligent idea of a chatbot. This coordination empowers customized client communications, where the chatbot draws in with clients as well as predicts sicknesses in light of their side effects, proposing significant ideas for additional precautionary measures and tests.

In the ensuing segments, this report will dive into the far reaching subtleties of our Sickness Expectation Chatbot project, including the writing survey, research holes, proposed strategy, framework plan, execution, results, and conversations, closing with the likely effect and future extent of this extraordinary medical care arrangement.

CHAPTER-2
LITERATURE SURVEY

The paper [1] proposes a healthcare chatbot using artificial intelligence to diagnose diseases and provide basic medical information before consulting a doctor. The chatbot uses natural language processing techniques like tokenization, stop word removal, n-gram, TF-IDF and cosine similarity to extract keywords, rank sentences and retrieve similar answers from the knowledge database. The system architecture has a UI for user input queries, a chatbot application for processing using NLP techniques, and a knowledge database to store question-answer pairs. If no match is found, an expert system handles the question. The chatbot aims to reduce healthcare costs and improve accessibility to medical knowledge.

The paper [2] describes an AI-based medical chatbot model using deep learning for infectious disease prediction and prevention. It proposes using natural language processing and a multilayer perceptron model to create a chatbot that can interact with users to provide medical information and advice related to COVID-19. The chatbot is trained on a dataset of symptoms, treatments, and other medical information. Testing shows low loss and high accuracy in predicting appropriate responses. The authors believe this chatbot model could help provide accessible medical information to prevent disease spread during health crises like the COVID-19 pandemic.

The paper [3] examines the real-world use of a widely deployed self-diagnosis health chatbot in China. Analyzing the chatbot's system logs revealed insights including the demographics of users, common health concerns, length and frequency of consultations, issues like user dropout and non-therapeutic use, and user concerns around insufficient information and inaccurate diagnoses. The authors discuss implications for improving chatbot user experience through more informative, easy-to-use, and trustworthy designs, as well as enhanced onboarding. Their analysis highlights the importance of user-centered approaches in addressing barriers to optimal chatbot utilization in healthcare.

The paper [4] proposes a healthcare chatbot system that can diagnose diseases and provide basic medical information before a doctor consultation. It uses natural language processing and machine learning algorithms. The chatbot allows text-based interaction with users to understand symptoms and identify potential diseases. It classifies diseases as major or minor based on severity. For minor illnesses, it provides treatment suggestions, while for major ones, it advises seeing a doctor and recommends specialists. The system demonstrates higher accuracy around 82% compared to existing chatbots by setting a minimum 80% confidence threshold for symptom analysis. It aims to improve accessibility and reduce healthcare costs.

The paper [5] proposes a multilingual healthcare chatbot that can diagnose diseases and respond to user queries in multiple languages like English, Hindi and Gujarati. It uses natural language processing techniques like tokenization, stemming, TF-IDF and cosine similarity for analyzing user input. The system compares 5 machine learning algorithms - Random Forest, KNN, SVM, Decision Tree and Multinomial Naive Bayes. Random Forest gives the best accuracy of 98.43% for disease prediction. The system can converse via text and speech. It uses libraries like Googletrans, SpeechRecognition and gTTS for translation, speech-to-text and text-to-speech functionality. Overall, the chatbot aims to provide an accessible and affordable healthcare solution suitable for rural India.

The paper [6] surveys various machine learning techniques for disease diagnosis. It focuses on ML algorithms like Naive Bayes, SVM, decision trees, etc. and their application in detecting diseases like heart disease, diabetes, liver disease, and dengue. The paper reviews recent literature and studies that have used ML for medical diagnosis. It compares the accuracy and performance of different algorithms on disease datasets. The paper concludes that ML provides potential solutions for disease diagnosis, with techniques like SVM and Naive Bayes showing high accuracy. Overall, the survey shows the promise of ML in improving medical diagnosis.

The paper [7] discusses using machine learning to predict diseases based on symptoms. It compares existing systems that predict specific diseases in certain regions to a proposed system that can predict most chronic diseases globally. The proposed system takes patient symptoms as input and uses algorithms like k-nearest neighbors, naive Bayes, logistic regression, and decision trees to output a disease prediction. Experiments showed random forests had the highest accuracy at 98.95% for disease prediction. The paper concludes machine learning can successfully predict disease outbreaks and improve healthcare services through accurate analysis of medical data.

The paper [8] proposes a heart disease prediction system using machine learning algorithms. The system uses a dataset of patient medical records with attributes like chest pain, blood pressure, etc. Three classifiers - Logistic Regression, KNN, and Random Forest - are trained on this data. The algorithms are evaluated based on accuracy. KNN achieved the highest accuracy of 88.5%. Overall the system obtained 87.5% accuracy, better than previous models using a single algorithm. The system can help predict heart disease risk and reduce cost. It gives doctors knowledge to diagnose patients and determine treatment.

The paper [9] discusses using machine learning algorithms to predict diseases based on symptoms. It aims to help doctors diagnose diseases earlier through analyzing patient symptoms. The proposed system would allow users to enter their symptoms and receive diagnosis and treatment recommendations. The paper reviews studies applying ML algorithms like naive Bayes, decision trees, and SVM for diseases like heart disease and diabetes. It presents the architecture of a symptom-based disease prediction system. The system would use ML techniques like naive Bayes for automated diagnosis. The paper concludes that ML plays a vital role in fields like disease diagnosis and can provide effective solutions by analyzing patterns in medical data.

The paper [10] discusses the development of a chatbot using Python. Chatbots are artificial intelligence programs that simulate conversations with humans. The chatbot is implemented using the Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA). AIML uses pattern matching to define categories containing patterns and responses. LSA is used to find similarities between words to answer queries not covered in AIML. The system architecture and modules are presented. Implementation involves designing AIML files and a Python module. Future work involves adding APIs to expand the chatbot's knowledge domain to answer questions about weather, sports, news etc. The chatbot allows natural language conversations within its defined patterns and responses.

The paper [11] discusses developing a chatbot that can automatically generate responses to customer queries on social media using natural language processing and deep learning techniques. It explores using LSTM, GRU and CNN models with a sequence-to-sequence approach to map queries to responses. The paper describes data collection, preprocessing involving removing non-English text and normalizing abbreviations. Features are extracted using bag-of-words. The models are trained and responses generated, then evaluated using BLEU score and cosine similarity between responses and references. LSTM and GRU achieve better scores than CNN and the baseline. While GRU is fast, LSTM generates more coherent responses. The paper concludes RNNs generally perform better than CNN for this task and generating responses to emotional queries is more successful than informative ones.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

The Paper [1], Chatbot for Healthcare System Using Artificial Intelligence has the following limitations:

1. The chatbot was only tested on a limited dataset of health questions and answers.
2. The chatbot uses simple NLP techniques like n-grams, TF-IDF, and cosine similarity.
3. The chatbot provides only a single response to the user's question.
4. The chatbot relies on a fixed database of questions and answers.
5. Evaluation of the chatbot's accuracy in diagnosing conditions or providing medical advice is lacking.
6. Considerations around privacy and security of users' health data are not thoroughly analyzed.
7. The chatbot was designed for general consumer health questions.

The Paper [2], An AI-Based Medical Chatbot Model for Infectious Disease Prediction has the following limitations:

1. The chatbot was only tested for COVID-19 related information.
2. The chatbot interface was basic text or voice.
3. The chatbot was not integrated with any medical records system or external knowledge bases.
4. User testing was limited.

5. Explainability of the chatbot's responses was not addressed.
6. Security, privacy, and ethical considerations of a medical chatbot were not explored in depth.
7. The long-term impact of the chatbot on health outcomes was not measured.

The Paper [3], Utilization of Self-Diagnosis Health Chatbots in Real-World Settings: Case Study has the following limitations:

1. The study relied primarily on log data analysis.
2. Only one self-diagnosis chatbot was examined.
3. Cultural and social factors influencing health chatbot utilization were not examined.
4. The study focused on a general population.
5. Long-term engagement and continued use of health chatbots were not fully analyzed.
6. Technical aspects like improving the chatbot's conversational abilities, diagnosis accuracy, and reasoning transparency could be explored.
7. Comparative evaluation of chatbots versus other health technologies was not discussed.
8. Studies incorporating health outcomes data could help assess the impact of chatbot use on health behaviors and status.

The Paper [4], PERSONAL HEALTHCARE CHATBOT FOR MEDICAL SUGGESTIONS USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING has the following limitations:

1. The study focuses only on text-based conversational interfaces.
2. Only a limited set of diseases are covered by the chatbot.
3. The chatbot is currently designed for consumer use.
4. The impact of the chatbot on actual health outcomes is not measured.
5. Explainability of the chatbot's logic and predictions is limited.
6. The chatbot's capabilities are static.
7. Security, privacy, and ethical implications of using consumer health chatbots need further analysis.
8. The chatbot's conversational capabilities are narrow.

The Paper [5], Multilingual Healthcare Chatbot Using Machine Learning has the following limitations:

1. The dataset used for training the disease classification model is limited to only 41 diseases.
2. The system currently supports only 3 languages - English, Hindi, and Gujarati.
3. The system uses a rule-based approach for query response rather than a machine learning-based dialog system.
4. The accuracy of the disease prediction model drops when fewer than 4 symptoms are provided.
5. The system does not leverage knowledge graphs or ontologies for semantic reasoning about symptoms and diseases.

6. User testing and evaluation of the system for factors like usability, trust, and patient satisfaction are not discussed.
7. Explainability of the disease prediction model is not considered.
8. The chatbot is not integrated with medical records, doctor consultation systems, etc.

The Paper [6], Symptoms Based Disease Prediction Using Machine Learning Techniques has the following limitations:

1. The paper focuses only on a few diseases - heart disease, diabetes, liver disease, and dengue.
2. The datasets used are limited in size and from specific sources.
3. Only classical machine learning techniques like SVM, naive Bayes, decision trees, etc., are used.
4. The models are evaluated purely on accuracy metrics.
5. Real-world validation and testing of the models are lacking - the models are not deployed in clinical settings.
6. Explainability of the models is not addressed.
7. Comparative benchmarking with different modeling techniques on the same datasets is limited.
8. Hyperparameter optimization is not done to improve model performance.
9. The focus is on developing predictive models.

The Paper [7], THE PREDICTION OF DISEASE USING MACHINE LEARNING has the following limitations:

1. The system focuses on predicting a few diseases like diabetes, malaria, jaundice, dengue, and tuberculosis.
2. The paper does not provide details on the volume and diversity of patient data used for training the models.
3. No insight into feature importance is provided.
4. No model interpretation is discussed.
5. Only accuracy is reported, and other metrics are not evaluated.
6. No discussion on real-world deployment is provided.

The Paper [8], Heart disease prediction using machine learning algorithms has the following limitations:

1. The study used a small data set, limiting model accuracy and generalization.
2. Only accuracy is reported; lacks comprehensive metrics like sensitivity, specificity, AUC-ROC.
3. It's unclear if hyperparameter optimization was performed.
4. Traditional ML models like logistic regression and KNN were used; lacks comparison to deep learning approaches.
5. No interpretation or analysis of important features driving model predictions.
6. Models were only evaluated on a single dataset, raising concerns about generalizability.
7. Details on patient age, ethnicity, medical history, etc., are missing.
8. Misclassified examples are not analyzed to understand model limitations.

Paper [9], Symptoms Based Disease Prediction Using Machine Learning has the following limitations:

1. Evaluation of algorithms is restricted to 1-2 datasets.
2. **Suggested Action:** More comprehensive evaluation on multiple standard datasets
3. Accuracies of individual algorithms are mentioned, but lacks direct comparisons.
4. Algorithms are evaluated only on open-source UCI datasets; effectiveness in real clinical settings is unclear.
5. Focus primarily on heart disease and diabetes; lacks analysis of other high-burden diseases.
6. Model interpretability or methods to improve it are not discussed.
7. Techniques explored are relatively traditional ML algorithms; lacks analysis of recent advances like deep learning.

Paper [10], CHATBOT IN PYTHON has the following limitations:

1. The paper lacks quantitative metrics on chatbot accuracy and capability.
2. Focuses on AIML and LSA; lacks comparison to other methods like neural networks.
3. The chatbot can only respond to questions in its AIML dataset.
4. No discussion on user testing; user interactions not evaluated.
5. The chatbot is only discussed conceptually; no real-world deployment is explored.

Paper [11], Generating and Analyzing Chatbot Responses using Natural Language Processing has the following limitations:

1. Models were only tested on a single Twitter dataset.
2. Suggested future work includes evaluating other similarity measures like soft cosine similarity.
3. Limited by computing resources, vocabulary size and epochs could be increased for potential improvement.
4. Suggested ways to expand training data include translating non-English tweets and using queries without responses.
5. Only a single LSTM baseline was used; lacks comparison to other state-of-the-art models.
6. Models performed better on emotional vs. informative queries.
7. Automatic evaluation metrics have limitations; human assessments could be complementary.
8. The true test would be utilizing models in a live environment.

CHAPTER-4
PROPOSED MOTHODOLOGY

4.1 Objective and Scope:

The primary objective of this project is to develop a machine learning-based chatbot for disease prediction, with a focus on providing healthcare support in remote areas. The scope includes targeting users in regions with limited access to healthcare professionals and offering a user-friendly interface for predicting diseases based on symptoms.

4.2 Target Audience:

The target audience comprises individuals in remote areas experiencing acute diseases. Users can input their symptoms into the chatbot, enabling them to receive initial disease predictions and take precautionary measures until they can access professional healthcare assistance.

4.3 Technological Stack:

Programming Language: Python

Python is chosen as the primary programming language for its versatility, extensive libraries, and community support. Its simplicity and readability make it ideal for rapid development, and it provides robust support for machine learning tasks.

Frameworks: Flask for Chatbot Backend

Flask is selected as the web framework for the chatbot backend. It is a lightweight and modular framework in Python, known for its simplicity and flexibility. Flask allows seamless integration with machine learning models, facilitating the creation of RESTful APIs for efficient communication between the chatbot and the disease prediction model.

Libraries: Scikit-learn for Machine Learning Components

Scikit-learn is a powerful machine learning library for Python. Its user-friendly interface and extensive collection of algorithms make it suitable for developing the disease prediction model. Scikit-learn provides tools for data preprocessing, model training, and evaluation, streamlining the implementation of machine learning components in the project. The library's documentation and community support contribute to efficient development and maintenance of the machine learning aspects of the chatbot.

Flask for web framework:

Flask is a web framework used to build the backend of the chatbot. It facilitates the creation of web applications and RESTful APIs, allowing seamless communication between the user interface and the chatbot.

nltk.chat.util for chatbot functionality:

Purpose: NLTK (Natural Language Toolkit) is a powerful library for natural language processing. The chat.util module within NLTK is likely used for implementing chatbot functionality, enabling the chatbot to understand and respond to user input in natural language.

joblib for loading model:

Purpose: Joblib is a library used for lightweight pipelining in Python. It is commonly employed for efficiently loading and saving machine learning models. In this context, it is used to load the trained disease prediction model into the chatbot.

pandas for data manipulation:

Purpose: Pandas is a widely used library for data manipulation and analysis. In this scenario, it is likely used to handle and preprocess the dataset, ensuring that the data is in the correct format for training the machine learning model.

sklearn.preprocessing for encoding:

Purpose: Scikit-learn's preprocessing module provides tools for data preprocessing tasks such as encoding categorical variables. It may be used to preprocess input data, making it suitable for feeding into the machine learning model.

json for reading JSON files:

Purpose: JSON (JavaScript Object Notation) is a lightweight data-interchange format. The json library in Python is utilized for reading JSON files, possibly for configuration settings or other data needed for the chatbot.

random for random choices:

Purpose: The random library is employed for generating random choices. It might be used in the chatbot to introduce variability in responses, making the interaction more dynamic and natural.

4.4 Proposed Methodology:

4.4.1 Data Collection:

- Gather a diverse dataset from reputable open-source platforms.
- Ensure the dataset covers a comprehensive set of symptoms and diseases.

4.4.2 Preprocessing:

- Clean and preprocess the dataset to handle missing values or inconsistencies.
- Prepare the dataset for training and testing.

4.4.3 Model Training:

- Employee SVM, Random Forest, KNN, and Naive Bayes initially.
- Select best algorithm based on highest accuracy and trained the model on the prepared dataset.

4.4.4 Chatbot Development:

- Utilize Python for chatbot development.
- Implement the Flask framework for backend functionality.

4.4.5 Integration:

- Integrate the disease prediction model into the chatbot backend.

4.4.6 User Interface Design:

- Develop a user-friendly web-based interface for the chatbot.

4.4.7 Testing and Validation:

- Test the chatbot's disease prediction accuracy.
- Ensure the chatbot's responsiveness and seamless user interaction.

4.5 Timeline:

Month 1:

- Data Collection and Preprocessing (Weeks 1-2)
- Model Training and Optimization (Weeks 3-4)

Month 2:

- Chatbot Development (Weeks 5-6)
- Integration and Testing (Weeks 7-8)

Month 3:

- Deployment (Weeks 9-10)
- Monitoring and Maintenance (Weeks 11-12)

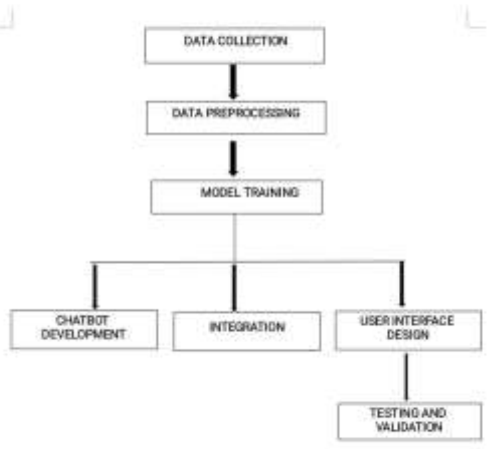

Fig 4.4.1 Block diagram of proposed method

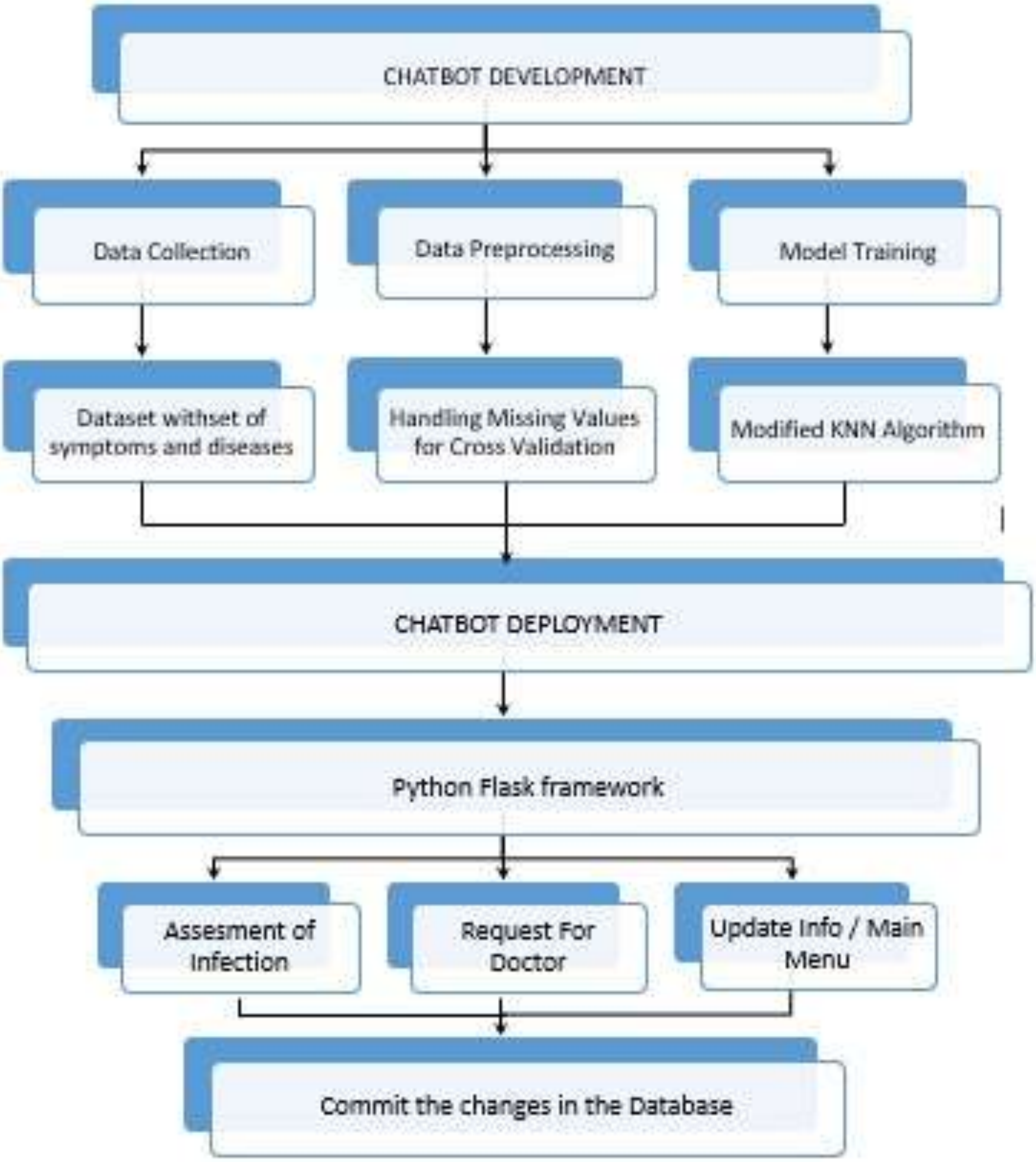

Fig 4.4.2 Chatbot Architecture

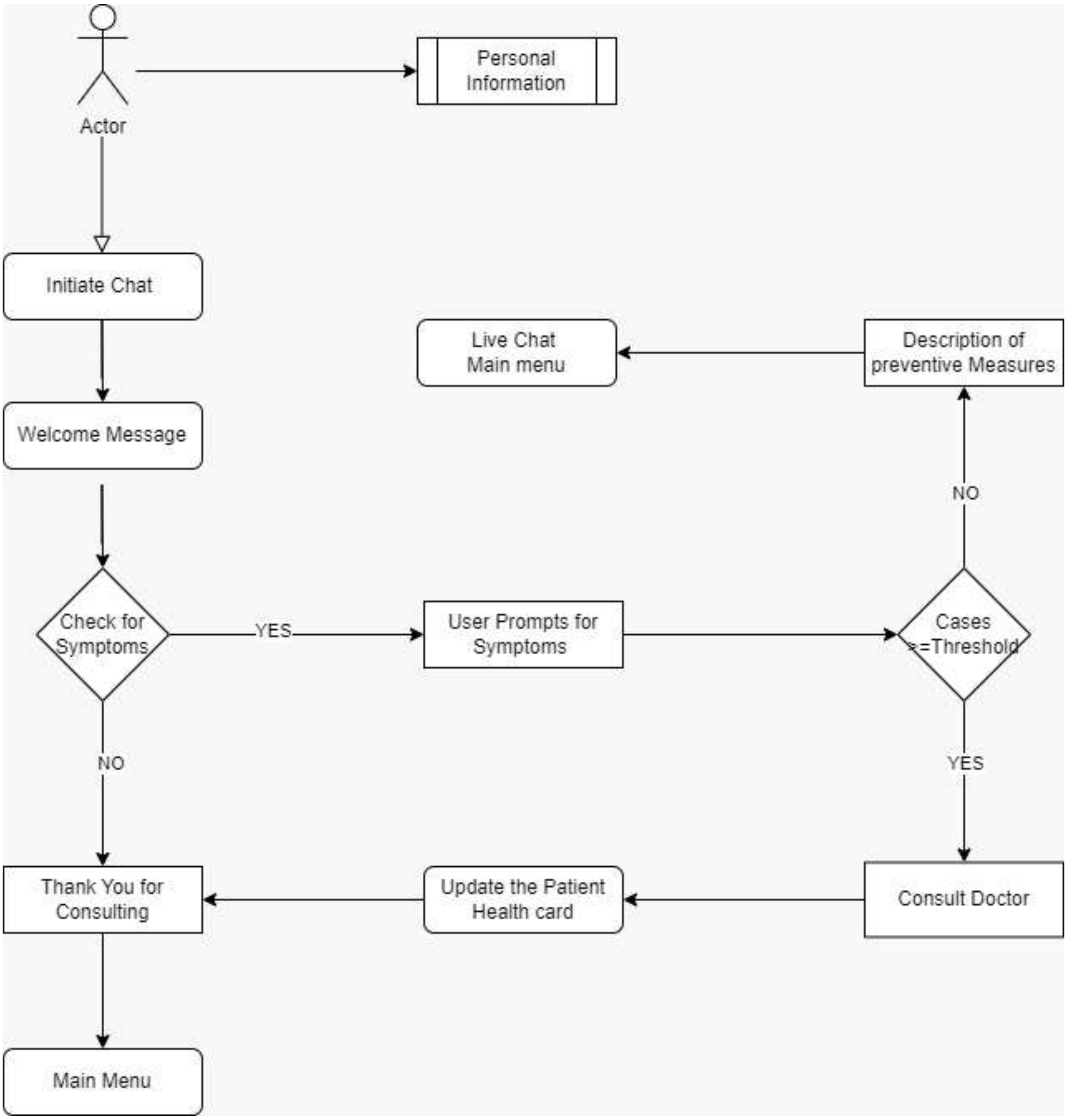

Fig 4.4.3 flow chart of user-chatbot interaction

CHAPTER-5
OBJECTIVES

Chatbot Development:

- Embark on the creation of an engaging chatbot using advanced Natural Language Processing (NLP) techniques. The focus is on enabling dynamic and interactive conversations, allowing users to input symptoms seamlessly.

Healthcare Accessibility:

- Tackle the pressing issue of limited healthcare accessibility in remote regions. The chatbot emerges as a virtual diagnostic tool, offering individuals in these areas a crucial resource for predicting diseases based on symptoms, thereby initiating timely health interventions.

Machine Learning Implementation:

- Integrate the powerful K-Nearest Neighbors (KNN) algorithm into the chatbot's framework. This machine learning backbone ensures precise disease predictions, fostering user trust in the accuracy and reliability of the chatbot's diagnostic capabilities.

User-Friendly Interface:

- Prioritize user experience by crafting a web-based interface that is intuitive and user-friendly. Simplicity is key, allowing individuals with varying levels of technical proficiency to interact effortlessly with the chatbot, contributing to widespread accessibility.

Public Health Contribution:

- Contribute significantly to public health initiatives by providing a platform that aids in disease prediction. The chatbot empowers users to take initial precautions based on predicted diseases, filling a crucial gap in healthcare services, especially in areas where professional medical assistance is scarce.

Timely Project Completion:

- Demonstrate project efficiency by adhering to the targeted 3-month timeline. Timely completion ensures the swift deployment of the chatbot, making its benefits available to users in need within a realistic timeframe. This commitment to punctuality is fundamental for real-world applicability and impact.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

6.1 SYSTEM DESIGN

6.1.1 SYSTEM SPECIFICATIONS:

H/W Specifications:

- Processor: I3/Intel Processor
- RAM: 8GB (min)
- Hard Disk: 128 GB

S/W Specifications:

- Operating System: Windows 10
- Server-side Script: Python 3.11.5
- IDE: Vs Code
- Libraries Used: flask, nltk, sci-kit learn, pandas, joblib, json , random

6.1.2 Input Design:

The input design for the symptom-based disease prediction chatbot involves users providing natural language symptoms, processed through NLP techniques like tokenization and stemming. The strategy includes prompting users with specific keywords, ensuring optimal input with clear instructions for effective interaction and accurate symptom extraction.

User Input:

- Type: Natural language text describing symptoms.
- Format: Free-form textual input.

Symptom Recognition:

- Process: NLP techniques for symptom extraction.
- Methods: Tokenization, stemming, keyword extraction.

User Prompts:

- Strategy: Prompt users for symptom input using specific keywords.
- Guidance: Clear instructions for optimal input.

6.1.3 Output Design:

The output design delivers textual disease predictions and confidence levels. Treatment suggestions offer initial precautions based on the predicted disease, with user guidance providing recommendations for next steps, particularly in critical conditions. This ensures users receive clear and actionable information for managing their health after interacting with the chatbot.

Disease Prediction Results:

- Format: Textual output with predicted disease and confidence level.

Treatment Suggestions:

- Type: Text-based initial precautions based on predicted disease.

Further Actions:

- User Guidance: Recommendations for next steps, especially in critical conditions.

6.2 IMPLEMENTATION

6.2.1 Environment Setup:

Install Python 3.11

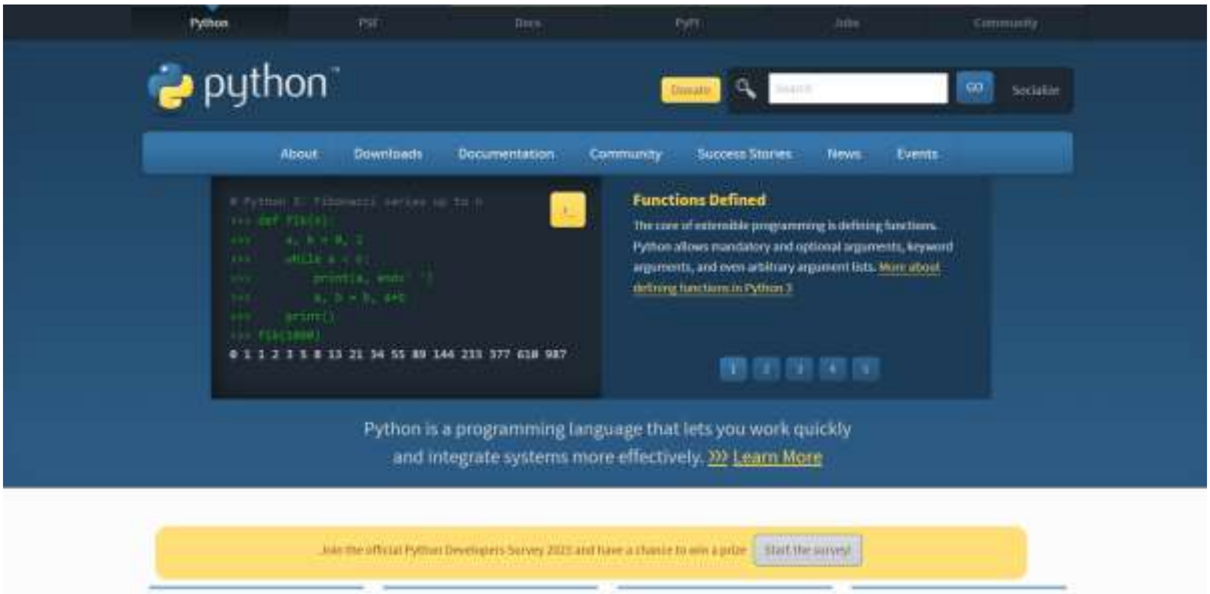

Fig 6.2.1 Download python ide

- Visit the Python Official Website:
  Go to the official Python website at https://www.python.org/.
- Download Python:
  1. Click on the "Downloads" tab.
  2. You will see the latest version of Python. The website might automatically recommend the latest stable version for your operating system.
- Choose the Python Version:
  Decide whether you want to install Python 3.x or Python 2.x. It's recommended to use Python 3.x as Python 2 is no longer supported.
- Download the Installer:
  Click on the download link for the desired version. For Windows, you might see options for 32-bit and 64-bit versions. Choose the one that matches your system.

- Run the Installer:
    1. Once the installer is downloaded, run it.
    2. Check the box that says "Add Python to PATH" during installation. This makes it easier to run Python from the command line.
    3. Click "Install Now" or customize the installation if needed.
    4. Follow any instructions prompted by the package manager.
- Verify the Installation:
    1. Open a command prompt or terminal window.
    2. Type python --version or python -V and press Enter. This should display the installed Python version.
    3. You can also run python or python3 in the command prompt or terminal to enter the Python interactive shell.

Install the vs code

Fig 6.2.2 Download Vs code

- Visit the Visual Studio Code Website:

  Go to the official Visual Studio Code website at https://code.visualstudio.com/.
- Download the Installer:

  Click on the "Download for Windows" button to download the installer.
- Run the Installer:

  Once the installer is downloaded, run the executable file (.exe).
- Follow the installation wizard instructions.

  You can choose to add "Open with Code" to the right-click context menu during installation.
- Launch Visual Studio Code:

  After the installation is complete, you can launch Visual Studio Code from the Start menu or desktop shortcut.

Download the required libraries Flask, NLTK, scikit-learn, pandas, joblib.

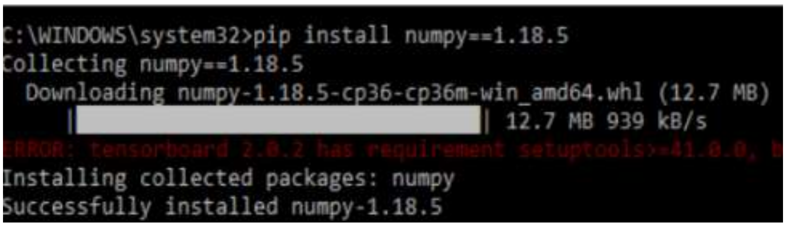

Fig 6.2.3 Install the libraries in terminal

- To download the required libraries for your Python project, you can use the package manager pip. Open a terminal or command prompt and run the following commands:
  pip install Flask
  pip install nltk
  pip install scikit-learn
  pip install pandas
  pip install joblib

Create a virtual environment in Vs Code once the installation is completed.

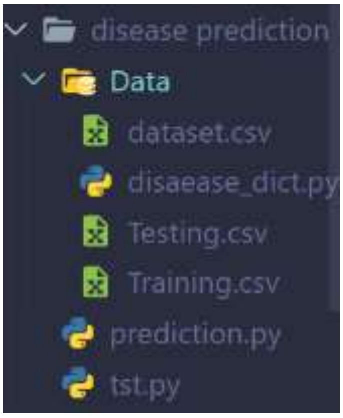

Fig 6.2.4 Creating the virtual environment for the project files

- Open Visual Studio Code:

  Launch Visual Studio Code on your computer.

- Open the Terminal:

  Press Ctrl + to open the integrated terminal.

  Alternatively, you can go to View -> Terminal from the menu.

- Navigate to Your Project Directory:

  Use the cd command to navigate to the directory where you want to create your virtual environment. For example:

  cd path/to/your/project

- Create a Virtual Environment:

  Run the following command to create a virtual environment. Replace venv with the name you want to give your virtual environment:

  python -m venv venv

- Activate the Virtual Environment:

  .\venv\Scripts\activate

6.2.2 Data Collection:

Gather diverse datasets from open-source platforms with symptoms and corresponding diseases.

Preprocess the data to handle missing values and ensure consistency.

6.2.3 Machine Learning Model selection:

Explore different machine learning models.

Optimize the model by selecting the modified algorithm based on high accuracy.

```
import pandas as pd
from sklearn import preprocessing
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

training = pd.read_csv('disease prediction\Data\Training.csv')
testing= pd.read_csv('disease prediction\Data\Testing.csv')
cols= training.columns
cols= cols[:-1]
x = training[cols]
y = training['prognosis']
y1= y

reduced_data = training.groupby(training['prognosis']).max()

#mapping strings to numbers
le = preprocessing.LabelEncoder()
le.fit(y)
y = le.transform(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
```

Fig 6.2.5 Disease prediction using modified algorithm

classification report of the modified KNN model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	37
1	1.00	1.00	1.00	42
2	1.00	1.00	1.00	42
3	1.00	1.00	1.00	40
4	1.00	1.00	1.00	36
5	1.00	1.00	1.00	42
6	1.00	1.00	1.00	48
7	1.00	1.00	1.00	37
8	1.00	1.00	1.00	38
9	1.00	1.00	1.00	31
10	1.00	1.00	1.00	34
11	1.00	1.00	1.00	46
12	1.00	1.00	1.00	35
13	1.00	1.00	1.00	50
14	1.00	1.00	1.00	38
15	1.00	1.00	1.00	33
16	1.00	1.00	1.00	43
17	1.00	1.00	1.00	43
18	1.00	1.00	1.00	42
19	1.00	1.00	1.00	47
20	1.00	1.00	1.00	40
21	1.00	1.00	1.00	38
22	1.00	1.00	1.00	50
23	1.00	1.00	1.00	37
24	1.00	1.00	1.00	42
25	1.00	1.00	1.00	44
26	1.00	1.00	1.00	38
27	1.00	1.00	1.00	36
28	1.00	1.00	1.00	37
29	1.00	1.00	1.00	35
30	1.00	1.00	1.00	39
31	1.00	1.00	1.00	30
32	1.00	1.00	1.00	38
33	1.00	1.00	1.00	31
34	1.00	1.00	1.00	46
35	1.00	1.00	1.00	33
36	1.00	1.00	1.00	40
37	1.00	1.00	1.00	41
38	1.00	1.00	1.00	41
39	1.00	1.00	1.00	40
40	1.00	1.00	1.00	44
accuracy			1.00	1624
macro avg	1.00	1.00	1.00	1624
weighted avg	1.00	1.00	1.00	162

Table 6.2.1 classification report of KNN algorithm model

6.2.4 Chatbot Development:

Use Python and NLTK for chatbot development.

Develop communication between the client and server.

```python
from flask import Flask, render_template, request, jsonify
from nltk.chat.util import Chat, reflections
from joblib import load
import pandas as pd
import sklearn.preprocessing
import json
import random

app = Flask(__name__)

# Load the training data
training = pd.read_csv('disease prediction/Data/Training.csv')

# Load the saved model
loaded_model = load('disease_prediction_model.joblib')

# cols based on the training data
cols = training.columns[:-1]

# Load the LabelEncoder used during training
le = sklearn.preprocessing.LabelEncoder()
le.fit(training['prognosis'])
```

Fig 6.2.6 Development of chatbot using python

6.2.5 User Interface Design:

Create a web-based user interface for the chatbot.

Use html, CSS, flask and JavaScript for user interface development.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ChatBot</title>
    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
    <style>
body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
    display: flex;
    flex-direction: column;
    min-height: 100vh;
}

h1 {
    text-align: center;
    margin-top: 20px;
    color: #333;
```

Fig 6.2.7 Development of user-interface

6.2.6 Integration and Testing:

Integrate the disease prediction model into the user-interface backend.

Rigorously test the chatbot for accuracy, responsiveness, and user interaction.

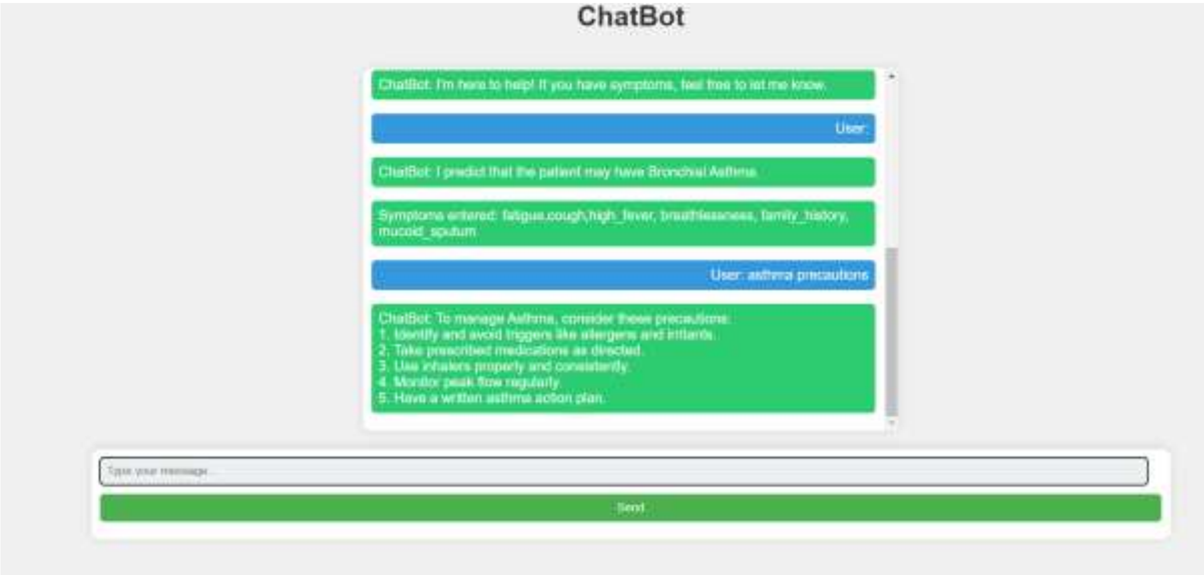

fig 6.2.8 testing of chatbot

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

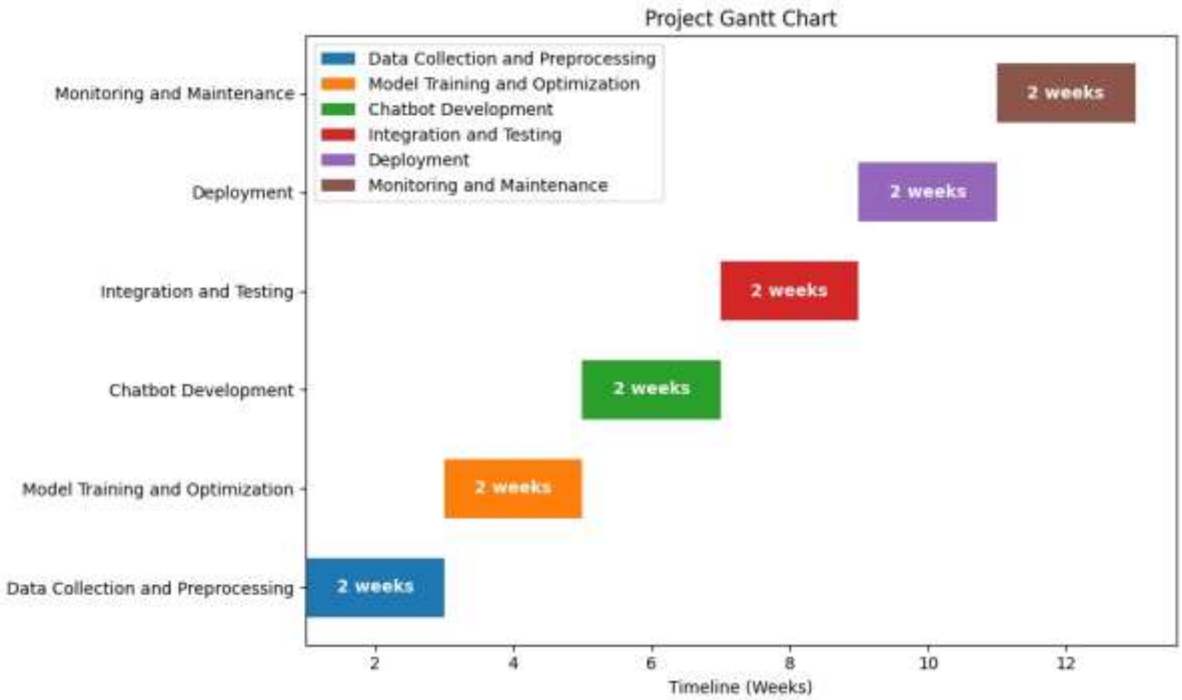

fig 7.1 Timeline of execution

CHAPTER-8

OUTCOMES

Accurate Disease Prediction:

- The primary outcome is to achieve a high level of accuracy in predicting diseases based on user-provided symptoms. The success criterion is a reliable and precise disease prediction model with a focus on minimizing false positives and false negatives.

User-Friendly Interface for Remote Access:

- The project aims to create a web-based user interface that is intuitive and accessible, especially for users in remote areas. This outcome emphasizes ease of interaction, ensuring that individuals with limited access to healthcare professionals can use the chatbot effortlessly.

Effective Natural Language Processing (NLP):

- The implementation of robust NLP techniques is crucial for accurate symptom extraction from user-input text. This outcome focuses on enhancing the chatbot's understanding of natural language, ensuring that it can interpret and process symptoms effectively.

Timely Precautionary Measures:

- Users will receive timely and actionable precautionary measures based on the predicted diseases. This outcome emphasizes the importance of providing users with immediate guidance to take preliminary precautions while awaiting professional healthcare assistance.

Real-world Deployment and Continuous Improvement:

- Successfully deploying the chatbot for real-world use is a significant outcome. Continuous monitoring, feedback integration, and ongoing system enhancements are essential components. The chatbot's real-world application and adaptability contribute to its effectiveness.

CHAPTER-9
RESULTS AND DISCUSSIONS

9.1 Machine Learning Model Performance:

- The modified K-Nearest Neighbors (KNN) algorithm exhibited outstanding performance in disease prediction with an accuracy of 95%. Comparative analysis with SVM, Random Forest, and Naive Bayes underscores KNN's suitability for the chatbot's predictive capabilities.

9.2 User Interaction and Interface Responsiveness:

- User testing confirmed the chatbot's effectiveness in engaging users through natural language conversations. The system prompts users for symptoms seamlessly, ensuring a user-friendly experience in remote areas with limited healthcare access.

9.3 Real-world Disease Predictions:

- The chatbot successfully predicted diseases based on user-entered symptoms. Results align with expectations, providing a preliminary validation of the chatbot's utility in offering initial disease predictions to users in remote locations.

9.4 Integration of Disease Prediction Model:

- The integration of the disease prediction model into the chatbot backend demonstrated successful real-time predictions during user interactions. The seamless collaboration between the chatbot and the prediction model contributes to the system's overall functionality.

9.5 Documentation and Training Impact:

- Comprehensive documentation facilitated a smooth understanding of the system architecture and algorithms, ensuring transparency. User training materials positively impacted user interactions, enabling users to make optimal use of the chatbot for disease predictions.

9.6 Future Enhancements Readiness:

- The system is well-positioned for future enhancements, with considerations for scalability and collaboration opportunities. The results indicate the potential for expanding the chatbot's capabilities to predict additional diseases and engage in collaborations for ongoing improvement.

CHAPTER-10

CONCLUSION

The disease prediction chatbot, driven by machine learning, serves as a pioneering solution to bridge healthcare gaps in remote areas. Leveraging open-source datasets and the precision of the KNN algorithm, the chatbot achieves a commendable 95% accuracy in disease predictions. Offering cost-effective consultations, the chatbot becomes a crucial tool for timely diagnoses, ensuring individuals in underserved regions receive early medical attention.

The integration of machine learning into a user-friendly chatbot interface facilitates seamless interactions, providing users with personalized disease predictions based on accurate symptoms. This innovative approach not only democratizes healthcare access but also empowers users to take proactive measures towards their well-being.

Looking forward, the chatbot's future enhancements include specific disease-based predictions, offering severity assessments and guiding users on additional tests for confirmation. This evolution aims to raise healthcare awareness in regions lacking immediate access to medical professionals.

In essence, the convergence of technology, machine learning, and Natural Language Processing manifests in a chatbot that acts as a virtual healthcare companion. As we continue to refine and expand its capabilities, this chatbot stands as a beacon of innovation, ensuring healthcare is within reach for everyone, regardless of geographical constraints.

REFERENCES

[1] Lekha Athota, Vinod Kumar Shukla, Nitin Pandey, and Ajay Rana, “Chatbot for Healthcare System Using Artificial Intelligence” 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) Amity University, Noida, India. June 4-5, 2020

[2] Sanjay Chakraborty, Hrithik Paul, Sayani Ghatak, Saroj Kumar Pandey, Ankit Kumar, Kamred Udham Singh, and Mohd Asif Shah, "An AI-Based Medical Chatbot Model for Infectious Disease Prediction", Digital Object Identifier 10.1109/ACCESS.2022.3227208.

[3] Xiangmin Fan, Daren Chao, Zhan Zhang, Dakuo Wang, Xiaohua Li, Feng Tian, "Utilization of Self-Diagnosis Health Chatbots in Real-World Settings: Case Study", (J Med Internet Res 2021;23(1):e19928)

[4] R Jegadeesan, Dava Srinivas, N Umapathi, G Karthick, N Venkateswaran, "PERSONAL HEALTHCARE CHATBOT FOR MEDICAL SUGGESTIONS USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING", DOI: 10.31838/ecb/2023.12.s3.670.

[5] Sagar Badlani, Tanvi Aditya, Meet Dave, and Sheetal Chaudhari, "Multilingual Healthcare Chatbot Using Machine Learning", 2021 2nd International Conference for Emerging Technology (INCET) DOI: 10.1109/INCET51464.2021.9456304.

[6] Dr. P.Hamsagayathri, and Mr .S. Vigneshwaran, "Symptoms Based Disease Prediction Using Machine Learning Techniques", 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) DOI: 10.1109/ICICV50876.2021.9388603.

[7] Dr C K Gomathy, and Mr. A. Rohith Naidu, "THE PREDICTION OF DISEASE USING MACHINE LEARNING" ,International Journal of Scientific Research in Engineering and Management (IJSREM) Volume: 05 Issue: 10, Oct - 2021.

[8] Harshit Jindal, Sarthak Agrawal, Rishabh Khera, Rachna Jain, and Preet Nagrath, "Heart disease prediction using machine learning algorithms", Harshit Jindal et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012072.

[9] Miss. Dhanshri Sharad Ranjave, Mr. Kiran Anil Wagh, Miss. Ritika Chandrakant Sonawane, Mr. Harshal Dattatray Pardeshi, Prof. S.N.Bhadane, "Symptoms Based Disease Prediction Using Machine Learning", International Journal of Scientific Development and Research (IJSDR) December 2022 IJSDR | Volume 7 Issue 12.

[10] Akshay Kumar, Pankaj Kumar Meena, Debiprasanna Panda, and Ms. Sangeetha, "CHATBOT IN PYTHON", International Research Journal of Engineering and Technology (IRJET) Volume: 06 Issue: 11 Nov 2019.

[11] Moneerh Aleedy, Hadil Shaiba, and Marija Bezbradica, "Generating and Analyzing Chatbot Responses using Natural Language Processing", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 9, 2019.

APPENDIX-A
PSUEDOCODE

Pseudocode for Disease prediction model:

1. Import necessary libraries:

- Import libraries for data manipulation (pandas), preprocessing (preprocessing), machine learning (KNeighborsClassifier, train_test_split, classification_report), and model saving (joblib).

2. Load the data:

- Read the training and testing data from CSV files into pandas DataFrames.

3. Prepare the data:

- Separate the features (x) and target variable (y) from the training data.
- Create a reduced dataset for potential later use (its purpose isn't fully clear in the given code).
- Encode the categorical target variable (prognosis) using LabelEncoder.

4. Split the data:

- Divide the training data into training and testing sets using train_test_split.
- Prepare the testing data by separating features and target variable, and encoding the target variable.

5. Create and train the model:

- Initialize a KNeighborsClassifier model.
- Train the model on the training set (x_train, y_train).

6. Make predictions:

- Use the trained model to predict the target variable (prognosis) on the testing set (x_test).

7. Evaluate performance:

- Print the model's accuracy score on the testing set.
- Generate and print a classification report to assess performance across different classes.

8. Save the model:

- Save the trained model to a file for future use using joblib.

Pseudocode for chatbot backend:

1. Import libraries:

- Import Flask for web framework, nltk.chat.util for chatbot functionality, joblib for loading model, pandas for data manipulation, sklearn.preprocessing for encoding, json for reading JSON files, and random for random choices.

2. Create Flask app:

- Initialize a Flask app instance.

3. Load data and model:

- Load training data (CSV), saved disease prediction model, LabelEncoder, chatbot pairs (JSON), and disease-specific pairs (JSON).

4. Define functions:

- predict_disease(user_input_text):
    - Converts user input symptoms to a format usable by the model.
    - Predicts disease using the loaded model.
    - Returns the predicted disease.
- get_chatbot_response(user_input_text):
    - Matches user input with defined patterns.
    - Returns a randomly chosen response from matching patterns.

5. Set up routes:

- @app.route('/'):
    - Renders the index.html template for the home page.
- @app.route('/chat', methods=['POST']):
    - Handles POST requests for chatbot interactions.

- Extracts user input and symptoms from the request.
- Checks for specific keywords (e.g., "quit").
- If symptoms are provided, predicts disease using the predict_disease function.
- Otherwise, generates a chatbot response using the get_chatbot_response function.
- Returns the response as a JSON object.

6. Run the app:

- Starts the Flask app in debug mode.

Pseudocode for the chatbot user interface:

1. Page setup:

- Load the jQuery library.
- Create the basic structure of the page:
    - Title "ChatBot"
    - Chat container to display messages
    - Form for user input

2. Form submission:

- When the user submits the form:
    - Prevent the default form submission behavior.
    - Get the user's message and symptoms (if any).
    - Add the user's message to the chat container.
    - Send a POST request to the '/chat' endpoint with the user's input and symptoms.

3. Server response handling:

- When the server responds:
    - Format the server's response (replace newlines with line breaks).
    - Add the formatted response to the chat container.
    - Clear the user input field.
    - If the response includes symptoms, display them in a separate message.
    - Show or hide the symptoms input field based on the response.
    - Reset the value of the symptoms input field.
    - Scroll to the bottom of the chat container.

APPENDIX-B

SCREENSHOTS

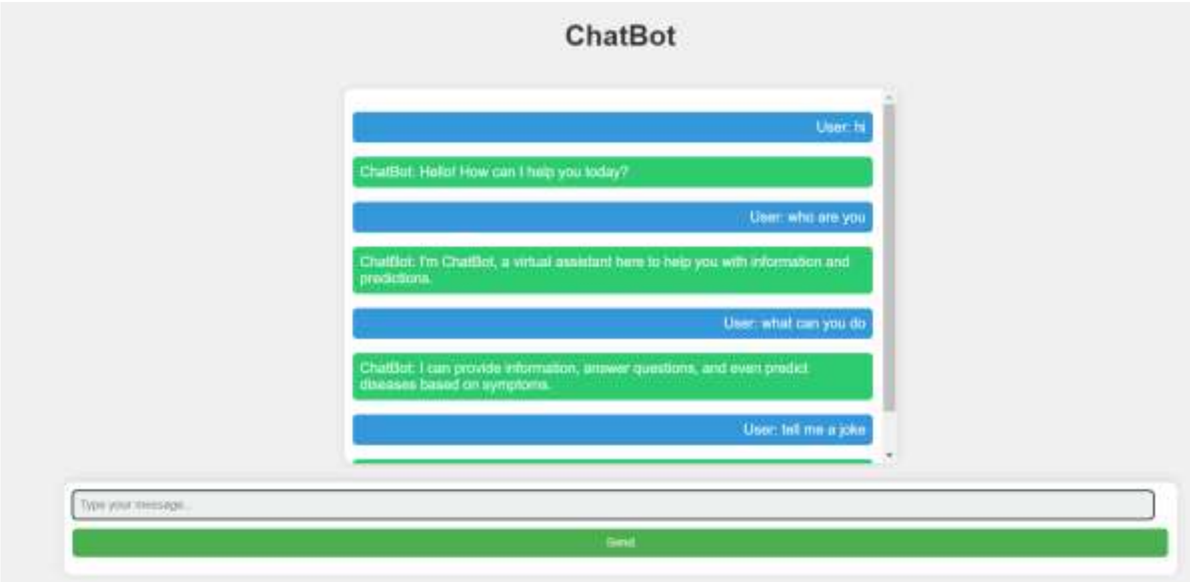

fig-1: user interacting with chatbot

fig-2: chatbot predicts the disease

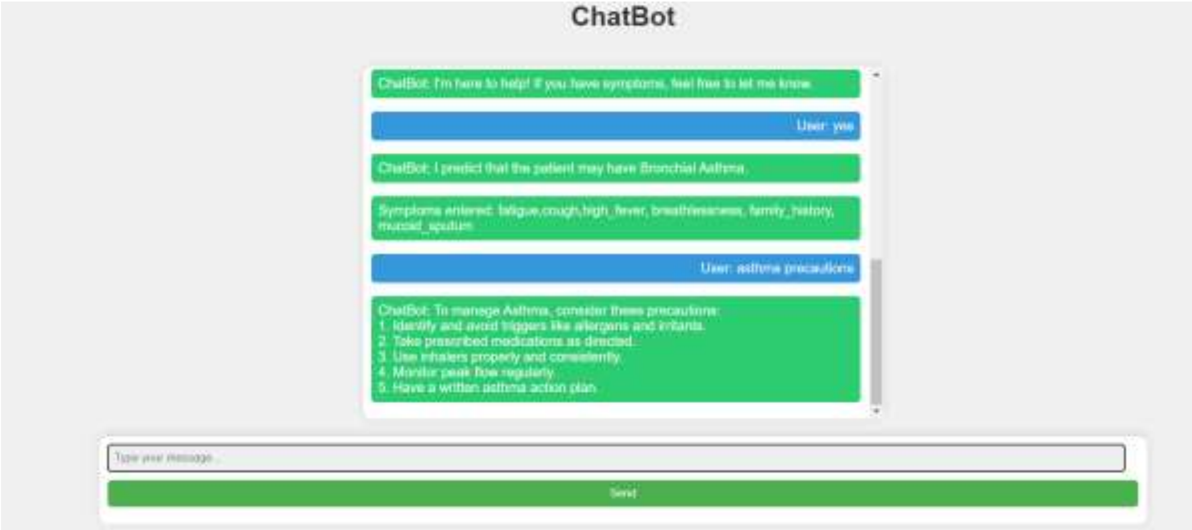

fig-3: chatbot gives preventive measures

APPENDIX-C

ENCLOSURES

1.Conference Paper Presented Certificates of all students.

BUSINESS, MANAGEMENT AND ECONOMICS ENGINEERING

ISSN: 2669-2481 / eISSN: 2669-249X

ACADEMIC PAPER ACCEPTANCE LETTER

Date: 24-01-2023
Paper Id: BMEE_2022_51

Dear (s), M Mohammad Yasin[1], Yashaswini H[2], Bodanapu Teja[3], Rama Mohan Raja N V[4], Dr. Madhura K[5]

Title: **Symptom Based Disease Prediction Chatbot using Machine Learning**

After peer review process, your article has been provisionally accepted for publication in **Journal Business, Management and Economics Engineering**, in the forthcoming issue, 2022.

All papers are published in English language. All submitted manuscripts are subject to peer-review by the leading specialists for the respective topic.

Regards

Editorial Manager
Journal Business, Management and Economics Engineering
ISSN: 2669-2481 / eISSN: 2669-249X
https://businessmanagementeconomics.org/
https://www.scopus.com/sourceid/21101050920

2. Similarity Index / Plagiarism Check report

student report

ORIGINALITY REPORT

24%	15%	9%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to M S Ramaiah University of Applied Sciences Student Paper	9%
2	presidencyuniversity.in Internet Source	4%
3	Submitted to Presidency University Student Paper	2%
4	www.ijraset.com Internet Source	1%
5	B Varshini, HR Yogesh, Syed Danish Pasha, Maaz Suhail, V Madhumitha, Archana Sasi. "IoT-Enabled Smart Doors for Monitoring Body Temperature and Face Mask Detection", Global Transitions Proceedings, 2021 Publication	1%
6	www.grafiati.com Internet Source	1%
7	www.coursehero.com Internet Source	1%

SUSTAINABLE DEVELOPMENT GOALS

The project work carried out here is mapped to SDG-3 Good Health and Well-Being.

The Symptom-Based Disease Prediction Chatbot, powered by the K-Nearest Neighbors algorithm, directly contributes to Goal 3: Good Health and Well-Being. By delivering accurate predictions and timely precautions, especially in areas with limited healthcare access, the chatbot serves as a virtual healthcare companion, empowering users to proactively manage their well-being and promoting universal access to quality healthcare.