

CARE-O-BOT USING NATURAL LANGUAGE PROCESSING

Submitted in partial fulfillment of the
requirements for the award of
Bachelor of Technology degree in Information Technology

By

SHARON SWETHA X (Reg No - 39120109)

PORKODI S (Reg No - 39120080)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI - 600119

MAY- 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

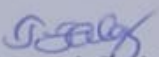
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

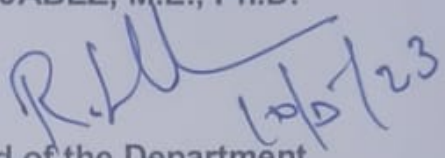
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SHARON SWETHA X (39120109)** and **PORKODI S (39120080)** who carried out the Project Phase-2 entitled "**CARE-O-BOT USING NATURAL LANGUAGE PROCESSING**" under my supervision from January 2023 to May 2023.

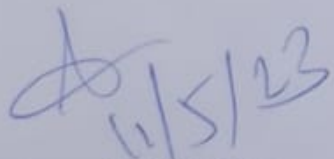

Internal Guide

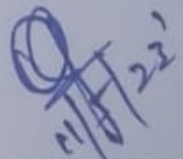
Dr. J. JABEZ, M.E., Ph.D.


Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.

Submitted for Viva voce Examination held on 11/5/23

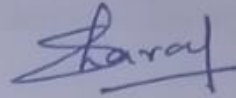

Internal Examiner


External Examiner

DECLARATION

I, **SHARON SWETHA X** (Reg No-39120109), hereby declare that the Project Phase-2 Report entitled "**CARE-O-BOT USING NATURAL LANGUAGE PROCESSING**" done by me under the guidance of **Dr. J. JABEZ, M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award Bachelor of Technology degree in Information Technology.

DATE: 11/5/2023



PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr.R.Subhashini M.E., Ph.D.** ,Head of the Department ,Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J.Jabez,M.E.,Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

In the world's population, there are 703 million people aged 60 to 70, which is a large figure, considering that by the year 2050 older adults would outnumber the children under the age of five, So the point is that there are more in number. With aging there are degenerative changes in many organs. After age forty there is an increase in cortical glomerulosclerosis and a decline in both glomerular filtration rate and renal plasma flow. These changes may be associated with an inability to excrete a concentrated or a dilute urine, ammonium, sodium, or potassium. Therefore, it is important to be aware of the potential electrolyte abnormalities in the elderly that can arise under these various conditions in order to prevent adverse outcomes. Hyponatraemia and hypokalaemia are the most common electrolyte abnormalities found among the elderly population taking diuretics. Both conditions are associated with short and long-term morbidity as well as mortality. A level under three can cause muscle weakness, spasms, cramps, paralysis and respiratory problems. The development of new technologies has left the elderly behind and unable to keep up. But in this age group only some number of people will have the knowledge of current technology so the remaining people will rely on the others. However, as a result of this, there is a significant digital divide among the elderly. So there is a necessity to develop a new technology to aid in their better care. Therefore, in order to address these issues, we have proposed a solution in which we want to develop a Voice AI chatbot using natural language processing model to help them to become independent. Additionally, a cloud hosting will be put up so that doctors may view their patients' prior diagnoses via graph.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	3
	2.1 Inferences from Literature Survey	3
	2.2 Open problems in Existing System	6
3	REQUIREMENTS ANALYSIS	8
	3.1 Feasibility Studies/Risk Analysis of the Project	8
	3.2 Software Requirements Specification Document	8
4	DESCRIPTION OF PROPOSED SYSTEM	9
	4.1 Selected Methodology or Process Model	9
	4.2 Architecture / Overall Design of Proposed System	12
	4.3 Description of Software for Implementation and Testing plan of the Proposed Model/System	13
	4.4 Project Management Plan	14
	4.5 Transition/ Software to Operations Plan	14
5	IMPLEMENTATION DETAILS	16
	5.1 Development and Deployment Setup	16
	5.2 Algorithms	17

	5.3 Testing	19
6	RESULTS AND DISCUSSION	20
7	CONCLUSION	25
	7.1 Conclusion	25
	7.2 Future work	25
	7.3 Research Issues	25
	7.4 Implementation Issues	26
	REFERENCES	27
	APPENDIX	30
	A. SOURCE CODE	30
	B. SCREENSHOTS	45
	C. RESEARCH PAPER	66

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No
4.2	System Architecture	12
4.4	Project Management Plan	14
6.1	Sodium	20
6.2	Diagnosing	21
6.3	PCA	21
6.4	BIRCH	22
6.5	DBSCAN	22
6.6	Sample Webpage	23
6.7	Register Page	23
6.8	Login Page	23
6.9	Cloud Page	24

CHAPTER 1

INTRODUCTION

The elderly population in the World continues to grow and is expected to double by 2050. Degenerative alterations occur in numerous organs as we age. GFR and renal plasma flow both decrease after the age of forty and there is an increase in cortical glomerulosclerosis. These alterations could be brought on by a failure to eliminate concentrated or diluted urine, ammonium, sodium, or potassium. Normally Minerals called electrolytes can be present in physiological fluids like urine, blood, and tissue. Electrolytes can also be obtained via foods, beverages, and supplements. Because they emit an electrical charge when they dissolve in bodily fluids, electrolytes receive their name. So the age, sex and preexisting health conditions can affect test results. And laboratories often use different methods for measuring electrolytes. This means results can vary from lab to lab.

Labs measure electrolytes by looking at the concentration of the substance in a specific amount of blood. In general, these are the normal ranges for electrolytes:

Sodium: 136 to 144 mmol/L.

Potassium: 3.7 to 5.1 mmol/L.

Calcium: In adults, 8.5 to 10.2 mg/dL.

Chloride: 97 to 105 mmol/L.

Magnesium: 1.7 to 2.2 mg/dL.

Phosphate: 2.5 to 4.8 mg/dL.

Bicarbonate: 22 to 30 mmol/L.

Older adults may be more susceptible to dehydration and electrolyte imbalances than younger adults. There are many reasons for this, including the following: The kidneys may lose some of their function with age. Older adults may take multiple medications, such as diuretics, that can change electrolyte levels.

Sodium. Low sodium, also called hyponatremia, causes water to move into cells. High sodium causes fluid to move out of the cells. When either of these things happens in brain cells, it can cause personality changes, headache, confusion

and lethargy. adults may be more susceptible to dehydration and electrolyte imbalances than younger adults. Electrolyte imbalance causes Hyponatremia and hyponatremia are the most common electrolyte abnormalities found in the elderly and both are associated with a high mortality.

The growth of new technologies has left the elderly behind and unable to keep up, but only a small percentage of those in this age group will be familiar with it, so the rest will depend on the others. But as a result, the elderly experience a considerable digital divide. Therefore, it is essential to create new technology to help with their better care.

Consequently, we have discovered that in order to address these concerns a remedy where we want to create a Voice AI chatbot utilizing a model of natural language processing to assist them to become independent.

So every elders will have family doctors as because of their old age their health may degrade and have to get checkup frequently .So of course the doctors will have their health chart with them. But most of the elders aren't aware of the electrolyte loss in their body. So, they will communicate their symptoms to the doctor through the voice chatbot that we create and using NLP remove the noises and detect whether humans are talking or not, so after that the doctor will diagnose their problem and contact them immediately and come up with their prescription.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY:

Mathew R B, Varghese S, Joy S E, Alex S S - 2019

Rohit Binu Mathew, Sandra Varghese, Sera Elsa Joy, Swanthana and Susan Alex conducted experiments we can observe that people don't value their health. This system's suggested medical chatbot is designed to be a conversational agent that encourages users to talk about their health problems and, depending on the symptoms they offer, returns the diagnosis. By interacting with users, this chatbot system will be able to recognise ailments. Chatbot diagnoses the illness and suggests a course of treatment using these extracted symptoms. The K-nearest neighbour algorithm is the machine learning algorithm used here (KNN).

Kandpal P, Jasnani K, Raut R, Bhorge S - 2020

Another research was conducted by Kandpal P, Jasnani K, Raut R and Bhorge S in which working on chatbots has provided us with a brief introduction to the field of virtual assistance and how it will affect our lives in the future. It is clear that chatbots will play a significant role in both large and small businesses. This paper covered the installation of various packages, code workflow, creating data in the intents file, training our model, and obtaining appropriate results.

Papiya Mahajan, Rinku Wankhade, Anup Jawade - 2020

Papiya Mahajan, Rinku Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge proceeded with an approach The user can communicate with the bot regarding their query via voice or text in our proposed system. The system responds to the queries using an expert system. The user can also see the doctors who are treating that specific illness. This solution enables numerous users to access online counselling sessions. The bot will offer analgesics and dietary suggestions.

Manthirathne, Wattaranthenne Pallemulle Walawwe Adithya Sriman - 2021

Manthirathne, Wattaranthenne Pallemulle Walawwe Adithya Sriman gathered data on different aspects study's technique proposes a more effective method for anticipating muscle cramping. To predict muscular anomalies, the system makes use of the wave patterns created by the muscles and analyses them using electromyograms. Real-time EMG data taken from a person is combined with a defined benchmark dataset and used with a trained algorithm to identify the wave patterns. The person will receive a warning or an alert if the identified wave patterns resemble wave patterns that cause muscle abnormalities. The patient will also receive recovery instructions for the muscle abnormalities. Time series prediction is the foundation of the suggested solution.

Kshitij Thakre , Dr. P. R. Rothe, Sakshi Kukade , Pranali Shinde - 2022

Kshitij Thakre, Dr. P. R. Rothe, Sakshi Kukade, Pranali Shinde, Komal Madame aimed to predict. The user can text chat with the bot in our proposed system to discuss the query. For query responses, the system makes use of an expert system. Additionally, users get access to a list of physicians who specialise in that condition. For online therapy sessions, this method can be used by many people. In the form of a pattern template, the chatbot's data is kept in the database..

Athulya N, Jeeshna K, SJ Aadithyan, U Sreelakshmi - 2021

Athulya N, Jeeshna K, S J Aadithyan, U Sreelakshmi, Hairunizha Alias Nisha Rose above sentiment analysis could be used for politics. In our system, the user can communicate with the chatbot via text, and it will respond to them both verbally and in writing. If a user is interacting with the chatbot, the bot responds to their questions by identifying the ailment. The bot suggests treatments for the user's problems and also suggests specialists that specialise in treating certain diseases.

Shivani Singh, Manmeet Kaur, Dr. Pooman Tanwar – 2022

This section of the article sought to analyse the available literature on COVID-19-related chatbots in healthcare, identify and characterise these cutting-edge technologies and their potential uses in the fight against COVID-19, and

outline associated difficulties. This research discusses a sophisticated AI medical chatbot for users, particularly during nCOV-19 and other unforeseen pandemics.

Kailash Chandra Bandhu, Binod Kumar Mishra, Mohit Patel – 2022

In the solution they have suggested, a text-based enquiry can be discussed in a chat session with the bot. Question answers are provided by the system. The user may also view the doctors who are qualified to treat their particular illness. Many users can find therapy appointments online by using this programme. In the form of a pattern template, chatbot data is saved on a website. The bot will offer analgesics and diet-related meals based on the patient's illness. Using a health care chatbot application that is based on Natural Language Processing, the SGD and ADAM optimizer's performance is assessed. Accuracy is used to gauge performance.

Nikita Vijay Shinde, Aniket Akhade, Pranali Bagad – 2021

In this work, they have introduced a healthcare chatbot that offers consumers medical assistance for a few prevalent disorders. The chatbot provides users with medical assistance when a doctor is not available, which will help to increase the availability & productivity of the medical industry. There are several medical chatbot models available, but they are generally expensive for the average person. With the healthcare chatbot system, they are working to reduce this type of annoyance. This technology actually functions as a sort of symptom checker and can accurately identify the patient's ailment 76% of the time.

Nithuna S and Laseena C.A – 2020

Integrated Systems made a substantial contribution in a variety of areas, including customer cooperation. This indicates that it will be difficult for each visitor to comprehend the structure in a useful way without any rules. The chatbot provides the observer with additional assistance in this case. Any AIML-based chatbots that are integrated into the article's architecture are added to it to foster client collaboration. several lightweight AIML files and packages that run the system to make it more adaptable and interactive for practise in various domains.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM:

Since the chatbot did not use cloud storage, the doctors are unable to consult the prior patient's health record that they diagnosed, and are forced to repeatedly ask the patient for details- Rohit Binu Mathew, Sandra Varghese, Sera Elsa Joy, Swanthana and Susan Alex(2019) an Papiya Mahajan, Rinku Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge(2020)

Seeing as chat bots are not humans, they have limited responses, so they can only respond to or prescribe things to patients based on the information that has been fed to them as input. As a result, using a chat bot as a health care assistant in this case without involving doctors on the server side is a major disadvantage- Kandpal P, Jasnani K, Raut R and Bhorge S(2020)

For patients to specifically contact doctors and describe their symptoms, no chat bot has been developed- Manthirathne, Wattaranthenne Pallemulle Walawwe Adithya Sriman(2022)

They have designed the online web application such that only persons with good literacy skills may use it, and neither the elderly nor those who lack basic literacy skills can use it-Kshitij Thakre, Dr. P. R. Rothe, Sakshi Kukade, Pranali Shinde, Komal Madame(2022)

They have created a voice chat bot that can only advise patients to consult with trained medical professionals; it cannot diagnose patients' ailments or use the patient's physician to communicate with the patients in the chat bot- Athulya N, Jeeshna K, S J Aadithyan, U Sreelakshmi, Hairunizha Alias Nisha Rose(2021)

They didn't utilise any models or demonstrate how accurate their predictions were in this study. - Shivani Singh, Manmeet Kaur, Dr. Pooman Tanwar – 2022

In this research report, only text-based chatbots have been employed as a communication channel, and no doctors were involved in the project. - Kailash Chandra Bandhu, Binod Kumar Mishra, Mohit Patel – 2022

For a few common diseases, they have developed a chatbot that offers medical aid and has an accuracy rate of only 76% which is less when determining the patient's condition. - Nikita Vijay Shinde, Aniket Akhade, Pranali Bagad – 2021

The healthcare chatbot in this research article was created without the usage of any web applications for deploying. - Nithuna S and Laseena C.A – 2020

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

According to my research, not a single author has focused on developing a chatbot for the elderly, even though some have done so. Instead, they have created chatbots that are only accessible to literate individuals. However, we are going to focus on the entire community so that even the elderly who are not literate can use the chatbot.

Therefore, in our project, we'll concentrate on patients who have electrolyte imbalance disorders and use NLP to create voice chat bots by gaining access to cloud storage.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT:

-  Python language is used
-  Google Colab as the environment in which the project is being built
-  Django as the python web framework for creating a front end website
-  ThingSpeak - is an IoT analytics platform service that allows us to analyze live data streams in the cloud

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

Taking into account the drawbacks of all the aforementioned approaches utilised in earlier systems, we are going to develop a voice chat bot employing Natural Language Processing and cloud storage as an IoT analytics platform service that enables cloud-based analysis. Here, we're going to concentrate on the patients who have electrolyte imbalance disorders and help them get diagnosed by the doctor via voice chat bot so that the next time they experience the same symptoms, the doctors will be able to see their health records in the cloud and be able to analyse them.

Since electrolyte imbalances are most common in elderly people, we are concentrating on them in order to enable them to use the chat bot independently and without assistance from others. Many people are unaware of the symptoms of electrolyte imbalance, such as fatigue or dizziness, until their health deteriorates to the point where they are at high risk of death.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

- **Data Collection** – The gathering of data is the initial step. Data collection is the process of gathering and analyzing information from a wide range of sources. So this project dataset contains 62991 rows(no of patients) and 18 columns(age, gender, glucose corrected sodium, albumin corrected calcium, department etc).
- **Data Analyzing** - In Machine Learning, Data analysis the initial step in completing the project is data analysis because learning about the inputs requires extensive data investigation. The data can even be compared during data exploration by using libraries like Matplotlib to display the data through some graphs, such as bar plots.
- **Data Preprocessing** - Data pre-processing is required following

investigation. Data pre-processing is an essential stage in machine learning that serves to improve the quality of the data to encourage the extraction of valuable insights from the data.

- **Analyzing Through Model-** As our dataset size is large and the fact that it is unlabeled, continuous, and has no output, we used an unsupervised clustering approach (PCA, BIRCH, DBSCAN) to reduce its dimension and conduct 3D and 2D analyses on it.
- **Creation of Chatbot** - The creation of a chatbot that uses pattern comparison to detect the sentence structure and remember response patterns. In this step, we will provide the chatbot with the anticipated patient questions as well as the signs of electrolyte imbalance. The discourse will then be designed after that. Consequently, there are two key perspectives when building a chatbot: the patient's and the doctor's. On the patient's perspective, the patient will use speech to communicate their symptoms in the chatbot, and on the doctor's perspective, the doctor will take the patient's voice and provide a diagnosis in the chatbot's text.
- **Integration of AI with NLP** – The development and integration of NLP come next. The first two steps in constructing a machine learning chatbot are building a speech bot and connecting it to the provider's API.
- **Training and Testing** - Predicting what people will say to your chatbot and what they expect from it is the main goal of bot training. You must currently have your use cases specified (specific purposes for your chatbot). After that, we must teach the AI chatbot how to comprehend the various ways that customers will phrase their inquiries. We begin asking the chatbot the questions we've taught it to respond to once it's ready. As usual, there are not many scenarios that need to be evaluated, allowing us to test manually. Your AI NLP chatbot's functionality can be checked through testing.

- **Cloud Hosting** - Following the development of the chatbot, the patient's past diagnosis is recorded, tracked, and saved in the cloud. In case the patient again displays the same symptoms, the doctor now has access to the patient's historical medical information stored in the cloud. When the patient is made aware that they have an electrolyte imbalance, the doctor will administer the proper therapy.
- **Web Development** - After building the chatbot, we developed the front-end Python web framework (Django) and an HTML page to demonstrate how it would look in future when improved.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

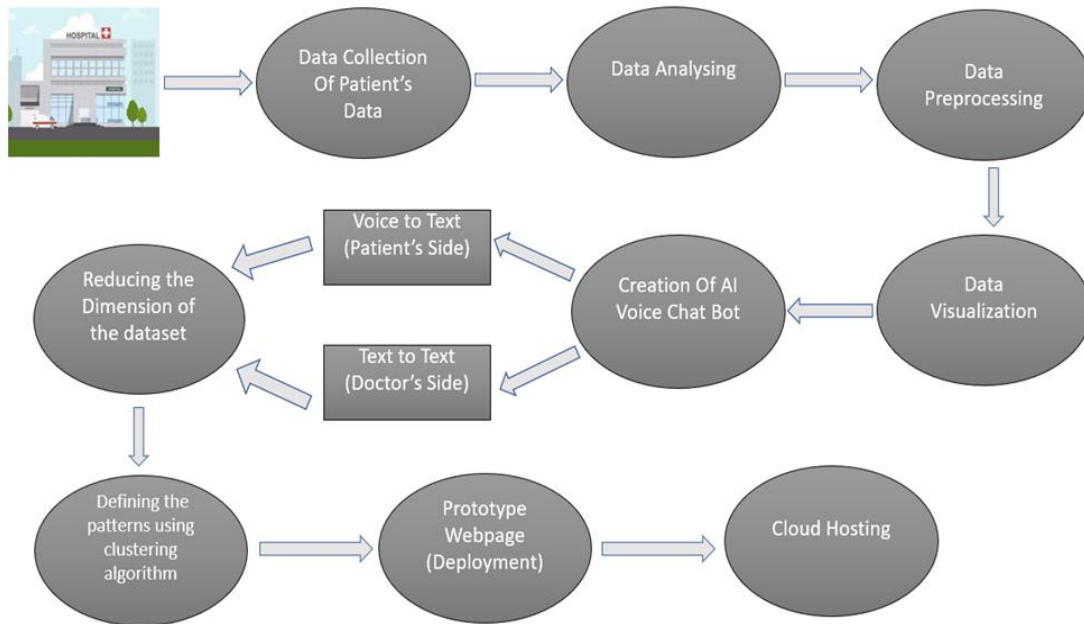


Fig 4.2: System Architecture

The above figures depict the proposed system's block diagram. We will initially collect information on people with electrolyte imbalance and their symptoms. The next phase is data analysis since learning about the data through data exploration is crucial. The raw data will then undergo pre-processing to make it suitable for the model. The building of an AI voice chat bot utilising pattern comparison is the next phase. The creating of the dialogue then takes place in order to supply it as input to the chatbot. Later, AI and NLP are combined, and a speech bot is built and connected to the provider's API. Following that, the chatbot's training and testing are finished. The next step is cloud hosting, where real-time analysis of data streams on the cloud takes place. Deployment is the very last stage.

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

First, we must gather patient data from various sources, such as Kaggle, etc., which will be unlabeled, necessitating the usage of an unsupervised learning technique. After the data preparation the following method of analyzing the data happens for the exploring the data which is done in google colab because we are taking it as the environment to build the project where we have used PCA, BIRCH, DBSCAN clustering algorithm which is an unsupervised model. Following data analyzation, an AI chatbot is created where the conversation design is completed in which we will have two sides one is the patient side prototype where the patient will speak to the chatbot to describe their symptoms and the other one is the doctor side prototype where the doctor will diagnose the patient while listening to their voice through the chatbot. After that integration of AI with Natural Language Processing happens Using this method estimates the cluster centers by constructing in rough. Testing is the following step, where we begin posing the queries, we have trained the chatbot to respond to. After that, we host our data in the cloud using things speak, an IoT analytics platform service that enables you to gather, view, and analyze data streams. We will use this cloud hosting to save the chats and previous patient records as a chart. At last, to demonstrate how it will look when improved in the future, we will first convert this into a front-end Python web framework and an HTML web page.

4.4 PROJECT MANAGEMENT PLAN

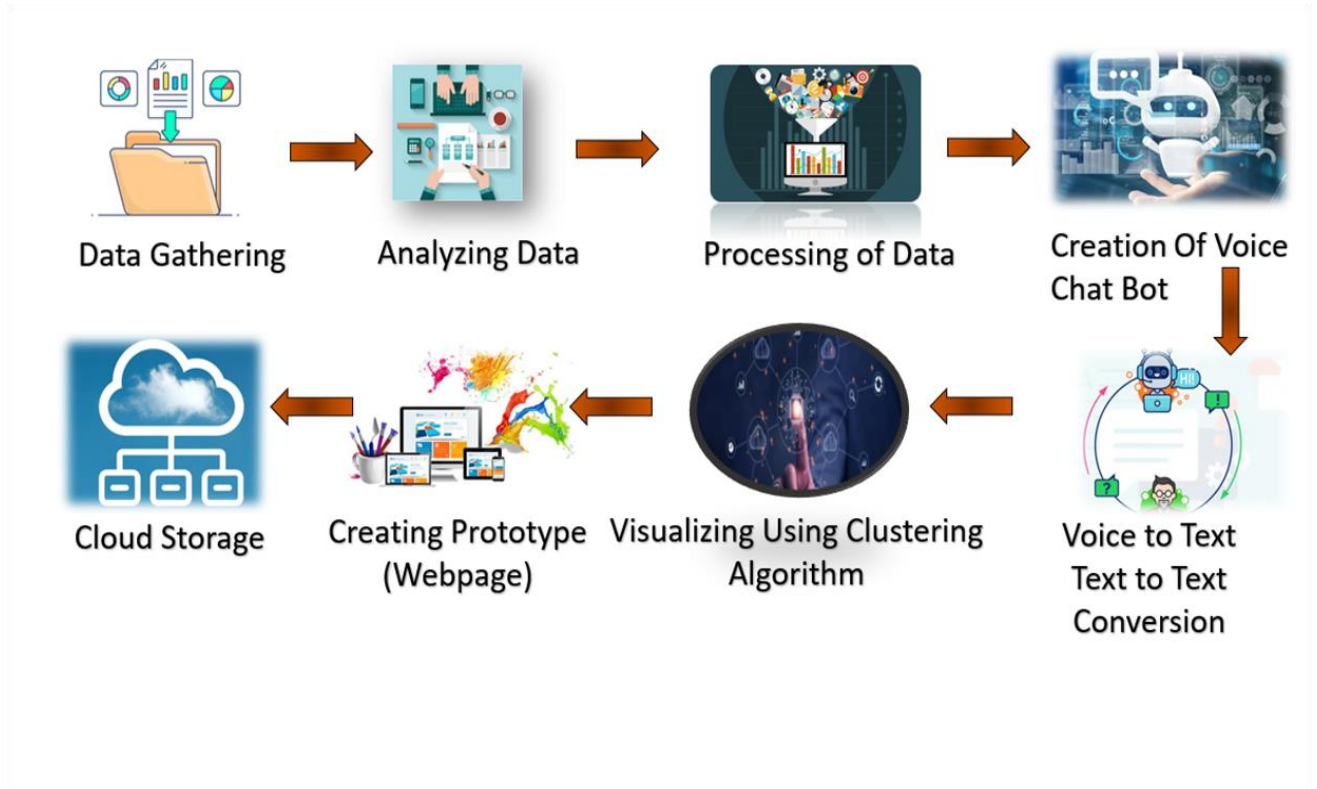


Fig 4.4: Project Management Plan

4.5 TRANSITION/SOFTWARE OR PROCESS MODEL

- Transitioning a healthcare chatbot for electrolyte imbalance from development to operations requires careful planning to ensure a smooth and successful launch.
- Identify key stakeholders: Identify the key stakeholders involved in the chatbot's development and deployment, including patients, doctors, IT staff, and project managers.
- Establish a deployment timeline: Develop a timeline for deploying the chatbot, including milestones for testing, training, and implementation.
- Identify technical requirements: Identify the technical requirements for deploying the chatbot, including software requirements, and ensure that the necessary infrastructure is in place.

- Develop a testing plan: Develop a testing plan to ensure that the chatbot functions as intended, and that it is accurate and effective
- Develop a training plan: Develop a training plan to ensure that healthcare professionals are trained on how to use the chatbot effectively, and that they understand its limitations and capabilities.
- Develop a maintenance plan: Develop a maintenance plan to ensure that the chatbot remains accurate and up-to-date with the latest medical patient's dataset
- Establish metrics for success: Establish metrics for success, such as user engagement, user satisfaction, and regularly measure and evaluate these metrics to ensure that the chatbot is meeting its objectives.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

To set up a healthcare chatbot using NLP for electrolyte imbalance, we will need to follow several steps:

Define the problem statement: Start by defining the scope of the chatbot and the problem statement it will solve. In this case, the chatbot will be designed to assist users in identifying and managing electrolyte imbalances.

Identify the target audience: Determine who the chatbot will be designed for. In this case, the target audience will include patients with electrolyte imbalances

Collect and organize data: Gather data related to electrolyte imbalances, including symptoms and causes. Organize this data into a format that can be easily used by the chatbot.

Train the chatbot: Use Natural Language Processing (NLP) to train the chatbot to understand user queries related to electrolyte imbalances. This involves identifying the intent behind the user's query and extracting relevant information from the query.

Develop the chatbot: Chatbot development will be done in google colab. The development of a chatbot that can recognize response patterns and recognize phrase structures through pattern comparison. In this stage, we will give the chatbot the questions we expect patients to ask as well as the symptoms of electrolyte imbalance. Following that, the dialogue will be developed. As a result, while developing a chatbot, it's important to consider both the patient's and the doctor's views. From the patient's point of view, the chatbot will be used to speak about their symptoms, and from the doctor's point of view, the doctor will listen to the patient speak and then type up a diagnosis.

Cloud Hosting: The patient's prior diagnosis is captured, tracked, and kept in the cloud after the chatbot development. The doctor now has access to the patient's prior medical information saved in the cloud in case the patient presents the same symptoms again. Informing the patient of their electrolyte imbalance will enable the doctor to provide the appropriate treatment.

Test the chatbot: Test the chatbot to ensure it is functioning correctly and providing accurate information to users. Use real user data to fine-tune the chatbot's responses and improve its performance.

Deploy the chatbot: Once the chatbot is tested and ready to use, deploy it to a suitable platform such as a website, social media, messaging app or mobile app.

Overall, developing a healthcare chatbot for electrolyte imbalances requires a robust data collection process, NLP training, and a well-planned conversation flow design. Deploying the chatbot to an accessible platform will increase the reach and impact of the chatbot on the intended audience.

5.2 ALGORITHMS

- **Natural Language Processing:** Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that deals with the interaction between computers and humans using natural language. NLP allows computers to understand, interpret, and generate human language, including spoken and written language. NLP involves various techniques and methods to analyze human language, including parsing, sentiment analysis, entity recognition, text classification, machine translation, and text summarization. NLP involves several stages, including data preprocessing, language modeling, and inference.
- **Principal Component Analysis:** Principal Component Analysis (PCA) is a

statistical technique used to identify patterns in data and reduce its dimensionality. PCA works by identifying the underlying structure of the data and transforming it into a set of new variables, known as principal components, which capture the most significant variations in the data. PCA can be used for various applications, such as data compression, data visualization, and feature extraction. PCA can reduce the dimensionality of the data, making it easier to visualize and analyze, while also reducing the noise and redundancy in the data. PCA can also be used for feature extraction, where the most significant features of the data can be identified and used for further analysis or modeling.

- ***Balanced Iterative Reducing and Clustering using Hierarchies:*** BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) is a hierarchical clustering algorithm used for large-scale data clustering. BIRCH is designed to be scalable and efficient, allowing it to handle datasets with a large number of data points. BIRCH has several advantages over other clustering algorithms. It is well-suited for large datasets with a large number of dimensions, as it can handle data with a high degree of sparsity. It also requires minimal memory and disk access, as it only needs to maintain a summary of the data distribution rather than the entire dataset. BIRCH can be used for a wide range of applications, including image segmentation, document clustering, and bioinformatics.
- ***Density-Based Spatial Clustering of Applications with Noise:*** DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm used to identify clusters of data points in a dataset. Unlike other clustering algorithms, DBSCAN does not require the user to specify the number of clusters beforehand. DBSCAN has several advantages over other clustering algorithms. It can identify clusters of arbitrary shape and handle noise effectively. It is also relatively insensitive to the initial parameter settings, and it does not require the number of clusters to be specified beforehand. However, DBSCAN can be sensitive to the choice of distance metric and the parameter settings, which can affect the quality of the clustering results.

5.3 TESTING

- Electrolyte imbalance occurs when there is an imbalance in the levels of electrolytes such as sodium, potassium, magnesium, and calcium in the body. Testing for electrolyte imbalance typically involves blood tests to measure the levels of these electrolytes in the blood.

- In a healthcare chatbot designed to help diagnose electrolyte imbalance using the doctors, the chatbot could ask the user questions about their symptoms and medical history, and then the doctor will suggest that they get a blood test to check their electrolyte levels. The doctor will use chatbot for providing information on what electrolyte imbalance is, what symptoms to look out for, and what the potential causes and treatments are.

CHAPTER 6

RESULT AND DISCUSSION

Thus, a trustworthy voice chatbot with Natural Language Processing and cloud hosting aids elders in a simple way to determine electrolyte imbalance. According to our research, a user-friendly assistant prototype is designed to make current technology more accessible to elders.

As a result, a powerful voice chatbot that is easy to use and offers a variety of advantages to old age people without making them reliant on others is developed. In future we may potentially connected our voice chatbot with a web application that uses the same Natural Language Processing and Cloud storage, therefore for that we have designed a sample prototype frontend web framework. This would enhance our project and give elderly people more independence and awareness of electrolyte loss. With the aid of this technology, senior citizens have greater accessibility.

PATIENTS SIDE PROTOTYPE:

```
[ ] filename = "sodium.wav"

[ ] # initialize the recognizer
r = sr.Recognizer() #performs speech recog using google specch recog API

# open the file
with sr.AudioFile(filename) as source:
    # listen for the data (load audio to memory)
    audio_data = r.record(source)
    # recognize (convert from speech to text)
    text = r.recognize_google(audio_data)
    print(text)

result2:
{ 'alternative': [ { 'confidence': 0.95515347,
                    'transcript': "hi I'm Sharon well I'm 82 years old "
                                'and a female I recently had some '
                                'symptoms like drowsiness headache '
                                'but I felt like I lost all of my '
                                'energy completely at first I did not '
                                'notice the symptoms seriously but '
                                'later on the symptoms got severe to '
                                'me while I felt like I have 40 '
                                'condition'},
                  { 'transcript': "hi I'm Sharon well I'm 82 years old "
                                'and a female I recently had some '
                                'symptoms like drowsiness headache '
                                'but I felt like I lost all of my '
                                'energy completely at first I did not '
                                'notice the symptoms seriously but '
                                'later on the symptoms got severe to '
                                'me while I felt like I have 40 '
                                'condition'}
```

Fig 6.1: SODIUM

DOCTOR SIDE PROTOTYPE:

```
[ ] list_trainer=ListTrainer(my_bot)
    for item in (small_talk,Math_talk_1,Math_talk_2):
        list_trainer.train(item)

List Trainer: [#####] 100%
List Trainer: [#####] 100%
List Trainer: [#####] 100%

[ ] print(my_bot.get_response("hi there!"))

What is your name?How old are you?Hello! This is Care0Bot

[ ] print(my_bot.get_response("Hello! This is Care0Bot"))

hi Iam Sharon well Iam 82 years old and a female I recently had some symptoms like drowsiness headache but I felt like I lost all of my energy completely at first I did not notice the
```

Fig 6.2: DIAGNOSING

ANALYZING THROUGH ALGORITHMS:

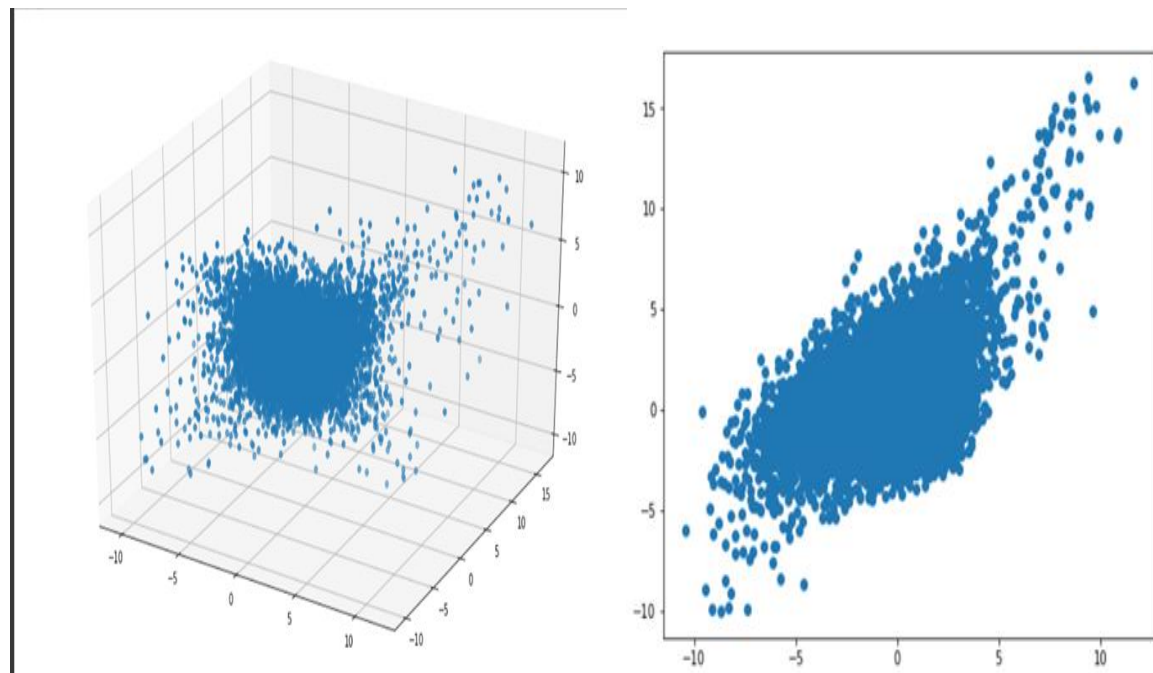


Fig 6.3: PCA

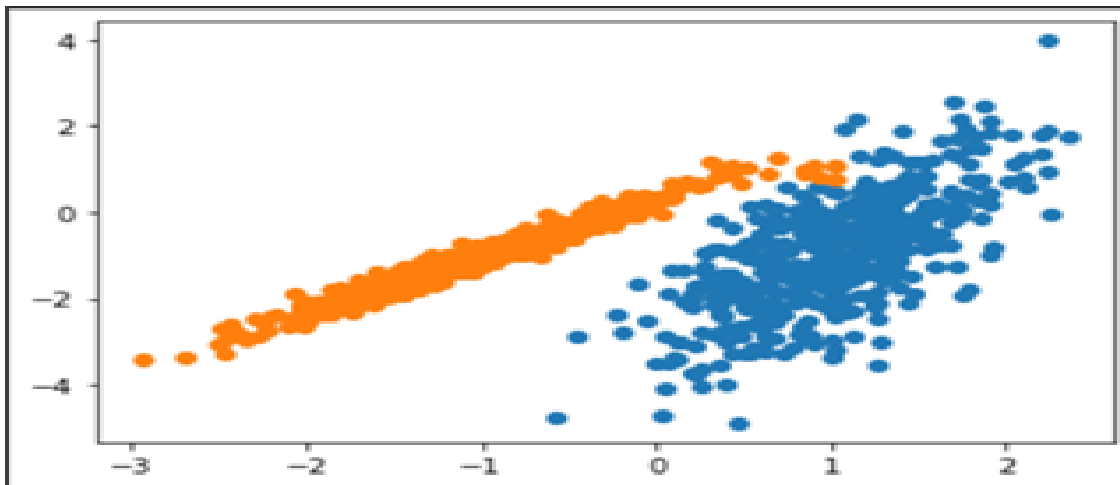


Fig 6.4: BIRCH

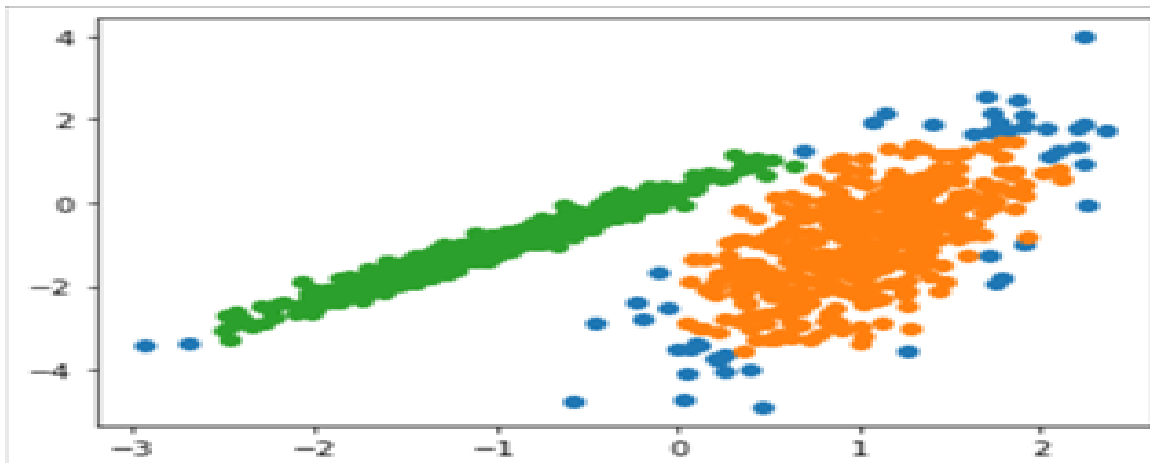


Fig 6.5: DBSCAN

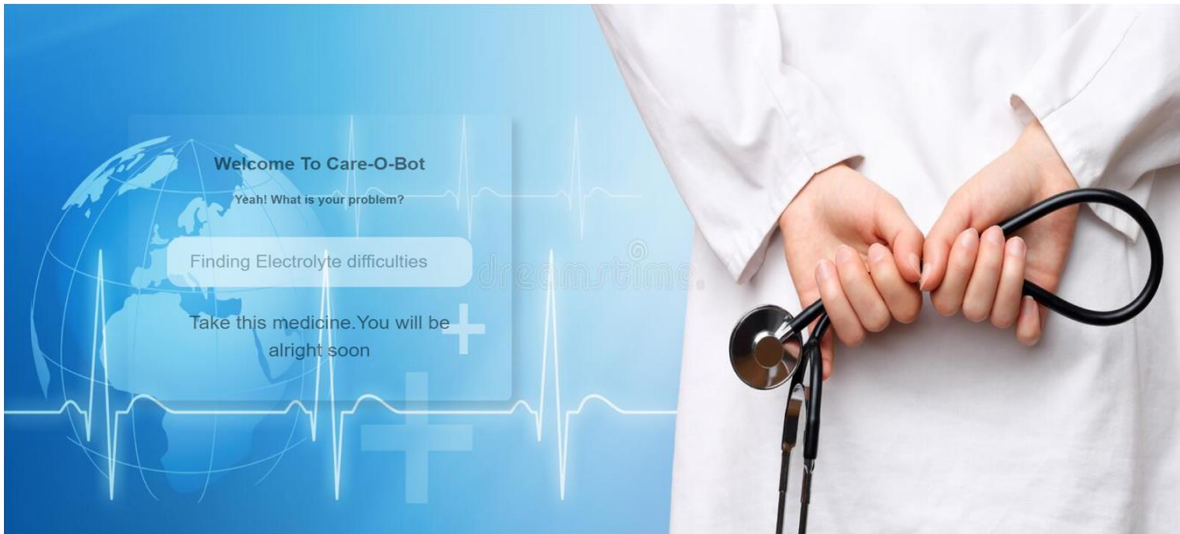


Fig 6.6: SAMPLE WEBPAGE

A 'REGISTER ACCOUNT' form is shown. It has a white background with a yellow border. The title 'REGISTER ACCOUNT' is at the top. Below it are four input fields: 'Username..', 'Email..', 'Enter password...', and 'Re-enter Password...'. Each field has a yellow square icon to its left. Below the fields is a green button labeled 'Register Account'. At the bottom, it says 'Already have an account? [Login](#)'.

Fig 6.7: REGISTER PAGE

A 'LOGIN' form is shown. It has a white background with a yellow border. The title 'LOGIN' is at the top. Below it are two input fields: 'Username...' and 'Password...'. Each field has a yellow square icon to its left. Below the fields is a green button labeled 'Login'. At the bottom, it says 'Don't have an account? [Sign Up](#)' and 'Forgot password? [Click here](#)'.

Fig 6.8: LOGIN PAGE

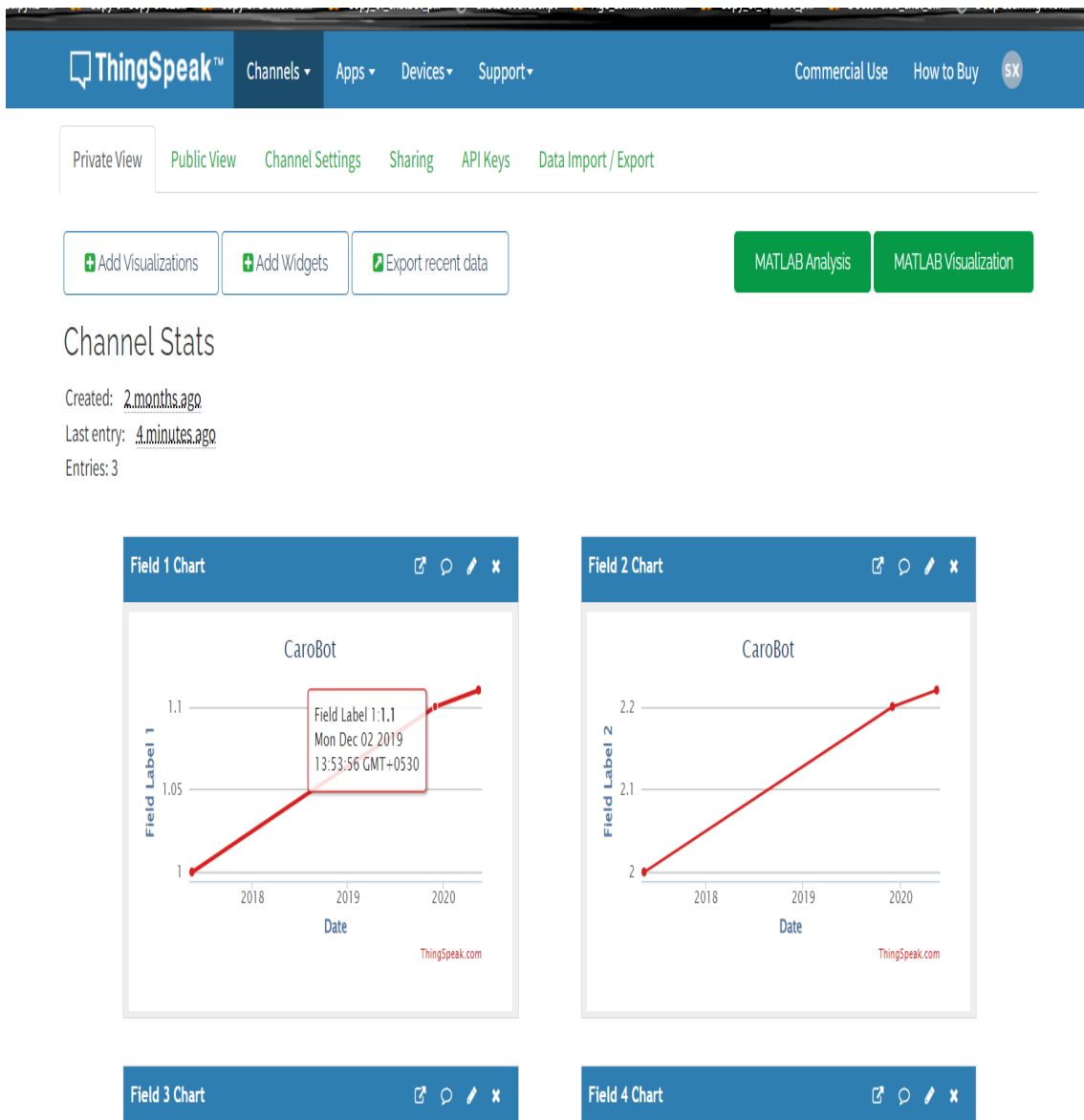


Fig 6.9: CLOUD PAGE

CHAPTER 7

CONCLUSION

Thus, an efficient voice chatbot that provides numerous benefits to older people who has electrolyte imbalance disorder in a user-friendly manner without making them dependent on others is created. According to our project, a user-friendly assistant is designed to make current technology more accessible to elders where the patient and doctor side prototype has been developed. It is also analysed with few clustering algorithms like Principal Component Analysis, BIRCH, DBSCAN algorithms and cloud hosting has been used for getting the prior details of the patient to save the doctors time

7.2 FUTURE WORK:

Future plans call for integrating our voice chatbot with a web application that uses the same Natural Language Processing and Cloud storage. This would enhance our project and give elderly people more independence and awareness of electrolyte loss. With the aid of this technology, senior citizens have greater accessibility.

7.3 RESEARCH ISSUES:

- Accuracy of diagnosis: One of the most critical issues in healthcare chatbots is accuracy. The chatbot should be able to diagnose electrolyte imbalance accurately and provide the right advice or recommendation to the user.
- Data quality and quantity: Developing a chatbot for healthcare requires a large amount of high-quality data. The data should be representative of the population, and the algorithms should be trained on diverse data sets to ensure that they are robust and effective.
- Integration with electronic health records (EHR): Healthcare chatbots need to be integrated with electronic health records to access the patient's medical history, laboratory results, medications, and other relevant information. This

integration ensures that the chatbot can provide personalized recommendations based on the patient's medical history.

- Natural language processing (NLP): Natural language processing is critical for healthcare chatbots, as it allows the chatbot to understand the user's queries and respond appropriately. Developing an NLP model that can understand the context and intent of the user's queries accurately is essential for the success of the chatbot.
- Privacy and security: Healthcare chatbots must comply with privacy and security regulations to ensure the patient's data is protected. The chatbot should have robust security measures in place to prevent data breaches or unauthorized access.
- Usability and user experience: The chatbot should be easy to use and provide a seamless experience to the user. The interface should be intuitive, and the chatbot should be able to provide relevant information quickly and efficiently.
- Clinical validation: Once the chatbot is developed, it should undergo clinical validation to ensure its accuracy and effectiveness in diagnosing electrolyte imbalances. Clinical validation involves testing the chatbot with real patients to evaluate its performance and identify any areas for improvement. Addressing these research issues can help ensure that the healthcare chatbot for electrolyte imbalance is accurate, effective, and safe for use by patients.

7.4 IMPLEMENTATION ISSUES:

Implementation issues for healthcare chatbot for electrolyte imbalance

Here are some implementation issues that could be considered when developing a healthcare chatbot for electrolyte imbalance:

- Integration with existing systems: The chatbot is designed to integrate with existing healthcare systems, such as electronic health records and

telemedicine platforms, to ensure that patient information is accurate and up-to-date. This would require coordination with IT departments and healthcare providers to ensure seamless integration.

- Natural Language Processing (NLP): The chatbot would need to be equipped with NLP capabilities to understand user input and respond appropriately. This would require the development and training data to ensure accurate interpretation of user input.
- Algorithm development: The chatbot would need to be equipped with accurate and up-to-date data for diagnosing electrolyte imbalances dataset. This would require collaboration with healthcare professionals and extensive research to ensure that the chatbot's recommendations are safe and effective.
- User interface design: The chatbot would need to be designed with a user-friendly interface that is easy to navigate and understand. This would require collaboration with UX designers and testing with potential users to ensure that the interface is intuitive and effective.
- Training data: The chatbot would require a large amount of training data to ensure accurate responses to user input. This would require collaboration with healthcare professionals and the development of a robust training dataset

REFERENCE:

[1] Mathew R B, Varghese S, Joy S E and Alex S Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning Third International Conference on Trends in Electronics and Informatics(2019)

[2] Kandpal P, Jasnani K, Raut R and Bhorge S Contextual Chatbot for Healthcare Purposes (using Deep Learning) 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) 625–34 (2020)

[3] Healthcare Chatbot using Natural Language Processing Papiya Mahajan, Rinku

Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 11 | Nov 2020

[4] HEALTHCARE CHATBOT Athulya N,Jeeshna K,S J Aadithyan,U Sreelakshmi,Hairunizha Alias Nisha Rose Volume 9, Issue 10 October 2021

[5] Manthrirathne, Wattaranthenne Pallemulle Walawwe Adithya Sriman Cloud based Muscle cramp prediction through machine learning by sensing EMG Pattern (2022)

[6] Health Care Chatbot Using NLP and Flask Kshitij Thakre, Dr. P. R. Rothe, Sakshi Kukade , Pranali Shinde , Komal Madame International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue V May 2022

[7] IoT based Chatbots using NLP and SVM Algorithms Astha Wahal, Muskan Aggarwal, Dr. T. Poongodi – 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM IEEE DOI:10.1109/ICIEM54221.2022.9853095

[8] Design and Development of Conversational Chatbot for Covid-19 using NLP: an AI application Shivani Singh, Manmeet Kaur, Dr. Pooman Tanwar – 2022 6th International Conference on Computing Methodologies and Communication (ICCMC)

[9] Health Care Chatbot using Natural Language Processing with SGD and ADAM Optimizer Parameter Optimization Kailash Chandra Bandhu, Binod Kumar Mishra, Mohit Patel – 2022 IEEE World Conference on Applied Intelligence and Computing

[10] Healthcare Chatbot System using Artificial Intelligence Nikita Vijay Shinde, Aniket Akhade, Pranali Bagad – 2021 Proceedings of the Fifth International Conference on Trends in Electronics and Informatics (ICOEI). IEEE Xplore Part Number:CFP21J32-ART; ISBN:978-1-6654-1571-2

[11] A Smart Chatbot Architecture based NLP and Machine learning for health care

assistance Soufyane Ayanouz, Boudhir Anouar Abdelhakim, Mohammed Benhmed
NISS2020, March 31-April 2, 2020, Marrakech, Morocco © 2020 Association for
Computing Machinery.

[12] Review on Implementation Techniques of Chatbot Nithuna S and Laseena C.A
International Conference on Communication and Signal Processing, July 28 - 30,
2020, India

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

[14] Healthcare Chatbot Aman Ranjan, Sagar Hukre, Sachin Rathore, Reetika
Kerkett – 2022 International Journal for Research in Applied Science & Engineering
Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538

[15] Healthcare Chatbot using Artificial Intelligence Anuj Patil International Journal for
Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-
9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue VIII Aug 2022

[16] Smart Medicare Chatbot Using Dialogflow and Support Vector Machine
Algorithm Sanika Dhavan International Journal for Research in Applied Science &
Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact
Factor: 7.429 Volume 9 Issue IX Sep 2021

APPENDIX

A. SOURCE CODE

PATIENT'S SIDE:

```
pip install speechrecognition #used for converting speech into text
!apt install libasound2-dev portaudio19-dev libportaudio2 libportaudiocpp0 ffmpeg #real
time audio,audio recording,noise reduction
pip install pytsx3 #used for conversion
!pip install PyAudio #binding the audio
pip install SpeechRecognition pydub
import speech_recognition as sr
filename = "sodium.wav"
# initialize the recognizer
r = sr.Recognizer() #performs speech recog using google specch recog API
# open the file
with sr.AudioFile(filename) as source:
    # listen for the data (load audio to memory)
    audio_data = r.record(source)
    # recognize (convert from speech to text)
    text = r.recognize_google(audio_data)
    print(text)
data="potassium.wav"
#open the file
with sr.AudioFile(data) as source:
    # listen for the data (load audio to memory)
    audio_data = r.record(source)
    # recognize (convert from speech to text)
    text = r.recognize_google(audio_data)
    print(text)
input="calcium1.wav"
#open the file
with sr.AudioFile(input) as source:
```

```

# listen for the data (load audio to memory)
audio_data = r.record(source)
# recognize (convert from speech to text)

text = r.recognize_google(audio_data)
print(text)

```

DOCTOR'S SIDE:

```

!pip install --upgrade chatterbot_corpus #export the content of the chatbot database
!pip install --upgrade chatterbot
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer #simplifying the process of loading the examples
my_bot=ChatBot(name='PyBot',read_only=True,logic_adapters=['chatterbot.logic.MathematicalEvaluation',
'chatterbot.logic.BestMatch']) #easy to generate automated
response to user input
small_talk=['hi there!',
            'What is your name?',
            'How old are you?',
            '\n Hello! This is CareOBot',
            '\n hi I am Sharon well I am 82 years old and a female I recently had some symptoms
like drowsiness headache but I felt like I lost all of my energy completely at first I did not
notice the symptoms seriously but later on the symptoms got severe to me while I felt like I
have fatigue condition',
            ]
Math_talk_1=['What is Your Gender', 'How long was this been happening?']
Math_talk_2=['Do You feel sleepy?', 'Is this the first time do you fee like this?']
list_trainer=ListTrainer(my_bot)
for item in(small_talk,Math_talk_1,Math_talk_2):
    list_trainer.train(item)
print(my_bot.get_response("hi there!"))
print(my_bot.get_response("Hello! This is CareOBot"))

```

IMPORTING THE DATASET:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/content/Electrolyte imbalances in an ED-datafile-Tazmini-032019.csv')
#df.shape
print(df)

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('/content/Electrolyte imbalances in an ED-datafile-Tazmini-032019.csv')
print(df)
df.shape
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import plotly.graph_objs as go
import plotly.offline as py
import random
'''from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA'''
pd.set_option('display.max_columns', None) #change the default no of rows to be displayed
df = pd.read_csv('/content/Electrolyte imbalances in an ED-datafile-Tazmini-032019.csv')
print(df)
rows = len(df.axes[0])
print(rows)
cols = len(df.axes[1])
print(cols)
df.head()
df.isna().sum().sort_values(ascending=False)
df= df.apply(lambda x: x.fillna(x.value_counts().index[0]))
df.isna().sum().sort_values(ascending=False)
df.dtypes
for column in df.columns:
    df[column] = df[column].astype('category')
df.dtypes
df['Age'] =df['Age'].cat.codes

df['Gender']=df['Gender'].cat.codes

df['SNA']=df['SNA'].cat.codes

df['SGLU']=df['SGLU'].cat.codes
```

```

df['Glucose corrected sodium']=df['Glucose corrected sodium'].cat.codes
df['SK']=df['SK'].cat.codes
df['SCA']=df['SCA'].cat.codes

df['SALB'] = df['SALB'].cat.codes
df['Albumin corrected calcium'] = df['Albumin corrected calcium'].cat.codes
df['SPO'] = df['SPO'].cat.codes
df['SMG'] = df['SMG'].cat.codes
df['Free calcium'] = df['Free calcium'].cat.codes
df['Department'] = df['Department'].cat.codes
df['Death nr days'] = df['Death nr days'].cat.codes
df['Readmission nr days'] = df['Readmission nr days'].cat.codes
df['Main condition'] = df['Main condition'].cat.codes
df['Other conditions'] = df['Other conditions'].cat.codes
df.head()

```

PCA ALGORITHM:

```

#3D PCA #reducing the dimension
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import plotly.graph_objs as go
import plotly.offline as py
import random
'''from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA'''
pd.set_option('display.max_columns', None)
#print(df)
from sklearn.preprocessing import StandardScaler

standardized = StandardScaler()

standardized.fit(df)

scaled_data = standardized.transform(df)

from sklearn.decomposition import PCA

pca = PCA(n_components=3)

pca.fit(scaled_data)

x_pca = pca.transform(scaled_data)

scaled_data.shape

x_pca.shape

```

```

"""from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline"""
x_pca[:1]
fig = plt.figure(figsize=(15, 8))

ax = fig.add_subplot(111, projection='3d')
ax.scatter(x_pca[:,0], x_pca[:,1], x_pca[:,2])

# 2D PCA
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import plotly.graph_objs as go
import plotly.offline as py
import random
"""from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA"""
pd.set_option('display.max_columns', None)
#df = pd.read_csv('/content/test.csv')
#print(df)
# Importing standardscaler module
from sklearn.preprocessing import StandardScaler

scalar = StandardScaler()

# fitting
scalar.fit(df)
scaled_data = scalar.transform(df)

# Importing PCA
from sklearn.decomposition import PCA
# Let's say, components = 2
pca = PCA(n_components = 2)
pca.fit(scaled_data)
x_pca = pca.transform(scaled_data)

x_pca.shape

# giving a larger plot
plt.figure(figsize =(8, 6))

plt.scatter(x_pca[:, 0], x_pca[:, 1])
sns.heatmap(df) #to better visualization

```

BIRCH ALGORITHM:

#unsupervised algorithm that used to perform clustering over particularly large dataset

```

from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import Birch
from matplotlib import pyplot
# define dataset
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# define the model
model = Birch(threshold=0.01, n_clusters=2)
# fit the model
model.fit(X)
# assign a cluster to each example
yhat = model.predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
    # get row indexes for samples with this cluster
    row_ix = where(yhat == cluster)
    # create scatter of these samples
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()

```

DBSCAN CLUSTERING ALGORITHM:

```

# dbscan clustering-finds clusters during noisy dataset circumstances
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import DBSCAN
from matplotlib import pyplot
# define dataset
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0,
n_clusters_per_class=1, random_state=4)
# define the model
model = DBSCAN(eps=0.30, min_samples=9)
# fit model and predict clusters
yhat = model.fit_predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
    # get row indexes for samples with this cluster
    row_ix = where(yhat == cluster)
    # create scatter of these samples
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()

```

HTML WEBPAGE:

```
<!DOCTYPE Html>
<Html Lang="En">
<Head>
<Meta Charset="UTF-8" />
<Meta Http-Equiv="X-UA-Compatible" Content="IE=Edge" />
<Meta Name="Viewport" Content="Width=Device-Width, Initial-Scale=1.0" />
<Title>Chatbot Javascript </Title>
<Link Rel="Stylesheet" Href="Style.Css" />
</Head>
<style>
body{
background-image:url('med.jpg');
background-repeat:no-repeat;
background-size:100% 100%;
background-attachment:fixed;

}
</style>

<Body>
<Div Class="Glass">
<center>
<H1>Welcome To Care-O-Bot</H1>
<H2>Yeah! What is your problem?</H2>
<Div Class="Input">
<Input
Type="Text"
Id="UserBox"
Onkeydown="If(Event.KeyCode == 13){ Talk()}"
Placeholder="Finding Electrolyte difficulties"
/>
</Div>
<P Id="ChatLog">Take this medicine.You will be alright soon</P>
</Div>
<Script Src="App.Js"></Script>
</Body>
</Html>

<!-- CSS -->
<Link Rel="Stylesheet" Href="Style.Css" />
//Body Section
<Script Src="Script.Js"></Script>
* {
Margin: 100;
Padding: 0;
Box-Sizing: Border-Box;
Background-Position: Center;
Align-Items: Center;

}
}
```

```

Body {
Width: 1000vw;
Height: 100vh;
Font-Family: Sans-Serif;
Padding: 10em 10em;
Background: Url(/Bg.Jpg);
Opacity:0.5;
Background-Position: Center;
Background-Repeat: No-Repeat;
Background-Position: 100% 200%;
Background-Size: Cover;
Display: Flex;
Align-Items: Center;
Justify-Content: Space-Between;
}
.Glass {
<center>
Width: 500px;
Height: 400px;
Background-Color: Rgba(255, 255, 255, 0.1);
Padding: 50px;
Color: #001;
Border-Radius: 9px;
Backdrop-Filter: Blur(50px);
Border: 2px Solid Transparent;
Background-Clip: Padding-Box;
Background-Position: Center;
Box-Shadow: 10px 10px 10px Rgba(45, 55, 68, 0.3);
Line-Height: 1.5;
Transform: Translatey(-5%);
Transition: Transform 0.5s;
}
.Glass-1 {
Width: 500px;
Height: 400px;
Background-Color: Rgba(255, 255, 255, 0.1);
Padding: 50px;
Color: Rgb(122, 82, 82);
Border-Radius: 9px;
Backdrop-Filter: Blur(50px);
Border: 2px Solid Transparent;
Background-Clip: Padding-Box;
Background-Position: Center;
Box-Shadow: 10px 10px 10px Rgba(45, 55, 68, 0.3);
Line-Height: 1.5;
Transform: Translatey(-5%);
Transition: Transform 0.5s;
Font-Size: 1.7rem;
}
.Glass H1 {
Font-Size: 1.5rem;

```

```

Text-Align: Center;
}
.Glass H2 {
Font-Size: 1rem;
Margin-Top: 20px;
}
.Input {
Width: 100%;
Height: 70px;
Overflow: Hidden;
Margin-Top: 40px;
}
.Input Input {
Width: 100%;
Height: 70px;
Border: None;
Padding-Left: 30px;
Padding-Top: 0;
Outline: None;
Font-Size: 1.5rem;
Border-Radius: 20px;
}
.Glass P {
Font-Size: 1.6rem;
Margin-Top: 30px;
}
* {
Margin: 0;
Padding: 0;
Box-Sizing: Border-Box;
}
Body {
Width: 100vw;
Height: 100vh;
Font-Family: Sans-Serif;
Padding: 10em 10em;
Background-Position: Center;
Background: Url(/Bg.Jpg);
Opacity: 0.5;
Background-Position: Center;
Background-Repeat: No-Repeat;
Background-Position: 100% 20%;
Background-Size: Cover;
Display: Flex;
Align-Items: Center;
Justify-Content: Space-Between;
}
.Glass {
Width: 500px;
Height: 400px;
Background-Color: Rgba(255, 255, 255, 0.1);

```

```

    Padding: 50px;
    Color: #000;
    Border-Radius: 9px;
    Backdrop-Filter: Blur(50px);
    Border: 2px Solid Transparent;
    Background-Clip: Padding-Box;
    Background-Position: Center;
    Box-Shadow: 10px 10px 10px Rgba(45, 55, 68, 0.3);
    Line-Height: 1.5;
    Transform: Translatey(-5%);
    Transition: Transform 0.5s;
}
.Glass-1 {
    Width: 500px;
    Height: 400px;
    Background-Color: Rgba(255, 255, 255, 0.1);
    Padding: 50px;
    Color: Rgb(122, 82, 82);
    Border-Radius: 9px;
    Backdrop-Filter: Blur(50px);
    Border: 2px Solid Transparent;
    Background-Clip: Padding-Box;
    Background-Position: Center;
    Box-Shadow: 10px 10px 10px Rgba(45, 55, 68, 0.3);
    Line-Height: 1.5;

    Transform: Translatey(-5%);
    Transition: Transform 0.5s;
    Font-Size: 1.7rem;
}
.Glass H1 {
    Font-Size: 1.5rem;
    Text-Align: Center;
}
.Glass H2 {
    Font-Size: 1rem;
    Margin-Top: 20px;
}
.Input {
    Width: 100%;
    Height: 70px;
    Overflow: Hidden;
    Margin-Top: 40px;
}
.Input Input {
    Width: 100%;
    Height: 70px;
    Border: None;
    Padding-Left: 30px;
    Padding-Top: 0;

```

```

Outline: None;
Font-Size: 1.5rem;
Border-Radius: 20px;
}
.Glass P {
Font-Size: 1.6rem;
Margin-Top: 30px;
}

```

DJANGO PAGE:

```

{% load static %}
<!DOCTYPE html>
<html>

<head>
<title>Login</title>
  <linkrel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-
9alt2nRpC12Uk9gS9baDI411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous"0>
  <link rel="stylesheet" href="{% static 'css/main.css' %}">

<style>
body,
html {
margin: 0;
padding: 0;
height: 100%;

}
.user_card {
width: 350px;
margin-top: auto;
margin-bottom: auto;
position: relative;
display: flex;
justify-content: center;
flex-direction: column;
padding: 10px;
box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
-webkit-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
-moz-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
border-radius: 5px;

}

.form_container {
margin-top: 20px;
}

```

```

.login_btn:focus {
box-shadow: none !important;
outline: 0px !important;
}
.login_container {
padding: 0 2rem;
}
.input-group-text {
background: #f7ba5b !important;
color: white !important;
border: 0 !important;
border-radius: 0.25rem 0 0 0.25rem !important;
}
.input_user,
.input_pass:focus {
box-shadow: none !important;
outline: 0px !important;
}
#messages{
background-color: grey;
color: #fff;
padding: 10px;
margin-top: 10px;
}
</style>
</head>
<body>
<div class="container h-100">
<div class="d-flex justify-content-center h-100">
<div class="user_card">
<div class="d-flex justify-content-center">
<h3 id="form-title">LOGIN</h3>
</div>

<div class="d-flex justify-content-center form_container">
<form method="POST" action="">
{% csrf_token %}
<div class="input-group mb-3">
<div class="input-group-append">
<span class="input-group-text"><i class="fas fa-user"></i></span>
</div>
<input type="text" name="username" placeholder="Username..." class="form-control">
</div>
<div class="input-group mb-2">
<div class="input-group-append">
<span class="input-group-text"><i class="fas fa-key"></i></span>
</div>
<input type="password" name="password" placeholder="Password..." class="form-control" >
</div>
<div class="d-flex justify-content-center mt-3 login_container">

```

```

<input class="btn btn-success" type="submit" value="Login">
</div>
</form>
</div>
{% for message in messages %}
<p id="messages">{{message}}</p>
{% endfor %}
<div class="mt-4">
<div class="d-flex justify-content-center links">
Don't have an account? <a href="{% url 'register' %}" class="ml-2">Sign Up</a>
</div>
<div class="d-flex justify-content-center links">
Forgot password? <a href="{% url 'reset_password' %}" class="ml-2">Click here</a>
</div>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

```

{% load static %}
<!DOCTYPE html>
<html>

```

```

<head>
<title>Login</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-
9alt2nRpC12Uk9gS9baDI411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">
<link rel="stylesheet" href="{% static 'css/main.css' %}">
<style>
body,
html {
margin: 0;
padding: 0;
height: 100%;
}
.user_card {
width: 350px;
margin-top: auto;
margin-bottom: auto;
position: relative;
display: flex;
justify-content: center;
flex-direction: column;
padding: 10px;
box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

```

```

-webkit-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
-moz-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
border-radius: 5px;
}
.form_container {
margin-top: 20px;
}
.login_btn:focus {
box-shadow: none !important;
outline: 0px !important;
}
.login_container {
padding: 0 2rem;
}
.input-group-text {
background: #f7ba5b !important;
color: white !important;
border: 0 !important;
border-radius: 0.25rem 0 0 0.25rem !important;
}
.input_user,
.input_pass:focus {
box-shadow: none !important;
outline: 0px !important;
}
</style>
</head>
<body>
<div class="container h-100">
<div class="d-flex justify-content-center h-100">
<div class="user_card">
<div class="d-flex justify-content-center">
<h3 id="form-title">REGISTER ACCOUNT</h3>
</div>
<div class="d-flex justify-content-center form_container">
<form method="POST" action="">
{% csrf_token %}
<div class="input-group mb-3">
<div class="input-group-append">
<span class="input-group-text"><i class="fas fa-user"></i></span>
</div>
{{form.username}}
</div>
<div class="input-group mb-2">
<div class="input-group-append">
<span class="input-group-text"><i class="fas fa-envelope-square"></i></span>
</div>
{{form.email}}
</div>
<div class="input-group mb-2">
<div class="input-group-append">

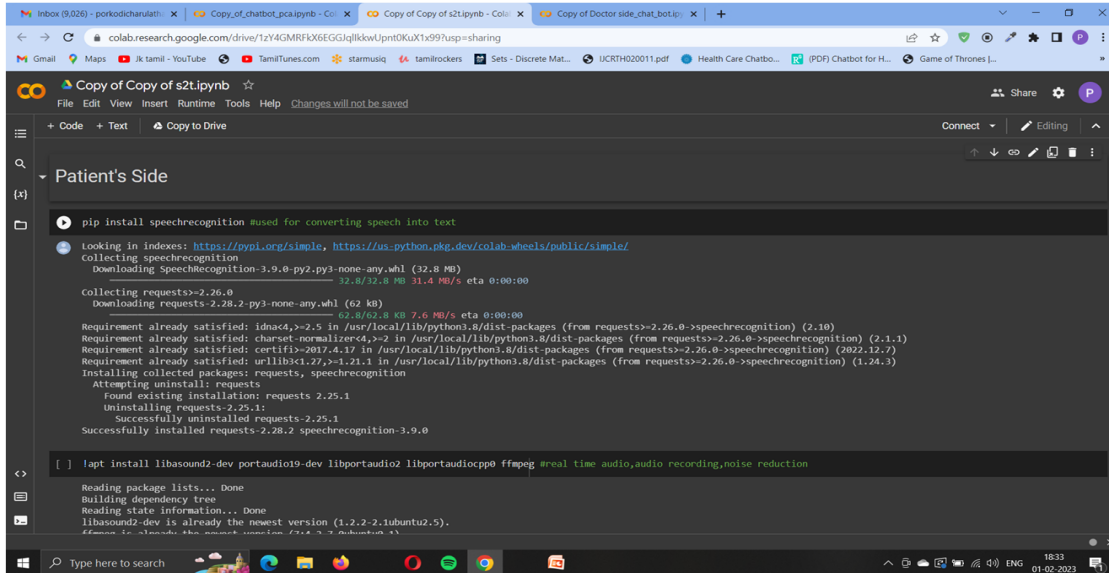
```

```

<span class="input-group-text"><i class="fas fa-key"></i></span>
</div>
{{form.password1}}
</div>
<div class="input-group mb-2">
<div class="input-group-append">
<span class="input-group-text"><i class="fas fa-key"></i></span>
</div>
{{form.password2}}
</div>
<div class="d-flex justify-content-center mt-3 login_container">
<input class="btn btn-success" type="submit" value="Register Account">
</div>
</form>
</div>
{{form.errors}}
<div class="mt-4">
<div class="d-flex justify-content-center links">
Already have an account? <a href="{% url 'login' %}" class="ml-2">Login</a>
</div>
</div>
</div>
</div>
</div>
</div>
<script>
/* Because i didnt set placeholder values in forms.py they will be set here using vanilla
Javascript
//We start indexing at one because CSRF_token is considered and input field */
//Query All input fields
var form_fields = document.getElementsByTagName('input')
form_fields[1].placeholder='Username..';
form_fields[2].placeholder='Email..';
form_fields[3].placeholder='Enter password...';
form_fields[4].placeholder='Re-enter Password...';
for (var field in form_fields){
form_fields[field].className += ' form-control'
}
</script>
</body>
</html>

```

B. SCREENSHOTS



The screenshot shows a Jupyter Notebook interface with the title 'Copy of Copy of s2t.ipynb'. The notebook is open to a cell titled 'Patient's Side'. The code in the cell is as follows:

```
pip install speechrecognition #used for converting speech into text
```

The output of the command is displayed below the code cell:

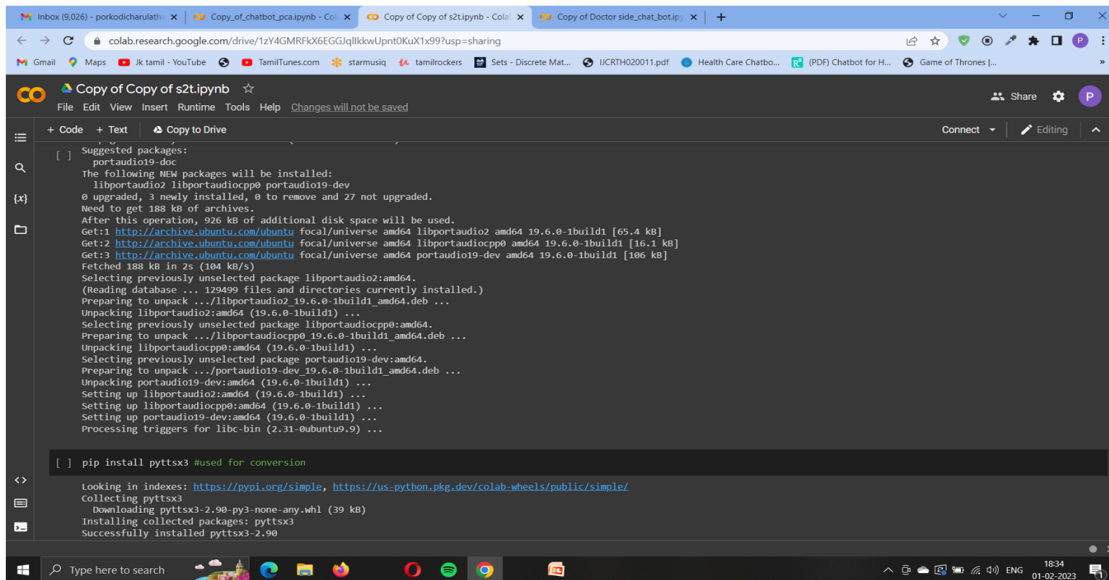
```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting speechrecognition
  Downloading SpeechRecognition-3.9.0-py2.py3-none-any.whl (32.8 MB)
    32.8/32.8 MB 31.4 MB/s eta 0:00:00
Collecting requests>=2.26.0
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
    62.0/62.0 kB 7.6 MB/s eta 0:00:00
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->speechrecognition) (2.10)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->speechrecognition) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->speechrecognition) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->speechrecognition) (1.24.3)
Installing collected packages: requests, speechrecognition
Attempting uninstall: requests
  Found existing installation: requests 2.25.1
  Uninstalling requests-2.25.1:
    Successfully uninstalled requests-2.25.1
Successfully installed requests-2.28.2 speechrecognition-3.9.0
```

Below the output, there is a code cell with the following code:

```
[ ] !apt install libasound2-dev portaudio19-dev libportaudio2 libportaudiocpp0 ffmpeg #real time audio, audio recording, noise reduction
```

The output of this command is also shown:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
libasound2-dev is already the newest version (1.2-2.2ubuntu2.5).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
```



The screenshot shows the same Jupyter Notebook interface, but now the code cell contains the following code:

```
[ ] pip install pyttss3 #used for conversion
```

The output of this command is displayed below the code cell:

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyttss3
  Downloading pyttss3-2.90-py3-none-any.whl (39 kB)
Installing collected packages: pyttss3
Successfully installed pyttss3-2.90
```

Below the output, there is a code cell with the following code:

```
[ ] !apt install libasound2-dev portaudio19-dev libportaudio2 libportaudiocpp0 ffmpeg #real time audio, audio recording, noise reduction
```

The output of this command is also shown:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
libasound2-dev is already the newest version (1.2-2.2ubuntu2.5).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
```

```
Copy of Copy of s2t.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
[ ] | pip install PyAudio #binding the audio
looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting PyAudio
  Downloading PyAudio-0.2.13.tar.gz (46 kB)
    46.0/46.0 KB 2.5 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: PyAudio
  Building wheel for PyAudio (pyproject.toml) ... done
  Created wheel for PyAudio: filename=PyAudio-0.2.13-cp38-cp38-linux_x86_64.whl size=69712 sha256=9f171ea54ea3e13244134248ac0138ec88181e05ec8c5ac0d05f5fb30f9b257e
  Stored in directory: /root/.cache/pip/wheels/c9/64/0b/9a48369872349f565ec45c3ec3f1efb2ce79464f6d9daf53
Successfully built PyAudio
Installing collected packages: PyAudio
Successfully installed PyAudio-0.2.13

[ ] | pip install SpeechRecognition pydub
looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: SpeechRecognition in /usr/local/lib/python3.8/dist-packages (3.9.0)
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.8/dist-packages (from SpeechRecognition) (2.28.2)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->SpeechRecognition) (2022.12.7)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->SpeechRecognition) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->SpeechRecognition) (1.24.3)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.26.0->SpeechRecognition) (2.10)
Installing collected packages: pydub
Successfully installed pydub-0.25.1

[ ] | import speech_recognition as sr
```

```
Copy of Copy of s2t.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
[ ] | import speech_recognition as sr

[ ] | filename = "sodium.wav"

[ ] | # initialize the recognizer
r = sr.Recognizer() #performs speech recog using google speech recog API

[ ] | # open the file
with sr.AudioFile(filename) as source:
    # listen for the data (load audio to memory)
    audio_data = r.record(source)
    # recognize (convert from speech to text)
    text = r.recognize_google(audio_data)
    print(text)

result2:
{ 'alternative': [ { 'confidence': 0.9551347,
  'transcript': "hi I'm Sharon well I'm 82 years old "
  'and a female I recently had some '
  'symptoms like drowsiness headache '
  'but I felt like I lost all of my '
  'energy completely at first I did not '
  'notice the symptoms seriously but '
  'later on the symptoms got severe to '
  'me while I felt like I have 40 '
  'condition'),
  ( 'transcript': "hi I'm Sharon well I'm 82 years old "
  'and a female I recently had some '
  'symptoms like drowsiness headache '
  'but I felt like I lost all of my '
  'energy completely at first I did not '
  'notice the symptoms seriously but '
  'later on the symptoms got severe to '
  'me while I felt like I have 40 '
  'condition') ] }
```



```
Copy of Copy of s2t.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
[ ] #open the file
with sr.AudioFile(input) as source:
    # listen for the data (load audio to memory)
    audio_data = r.record(source)
    # recognize (convert from speech to text)
    text = r.recognize_google(audio_data)
    print(text)

result:
{ 'alternative': [ { 'confidence': 0.95691472,
                    'transcript': 'hello I am Helena I'm a female '
                    '75-year-old I recently experience '
                    'some symptoms such as diarrhea or '
                    'vomiting where I always feel '
                    'nauseous after eating something and '
                    'even though I didn't engage in any '
                    'physical activity I just keep '
                    'sweating more than usual I also felt '
                    'my muscles were weak making it '
                    'impossible for me to lift anything '
                    'at first I assumed this this was due '
                    'to my older age but it was not the '
                    'symptoms are now becoming more '
                    'severe' },
                  { 'confidence': 0.95691472,
                    'transcript': 'hello I am Helena I'm a female '
                    '75-year-old I recently experience '
                    'some symptoms such as diarrhea or '
                    'vomiting where I always feel '
                    'nauseous after eating something and '
                    'even though I didn't engage in any '
                    'physical activity I just keep '
                    'sweating more than usual I also felt '
                    'my muscles were weak making it '
                    'impossible for me to lift anything '
                    'at first I assume this this was due '
                    'to my older age but it was not the '
                    'symptoms are now becoming more '
                    'severe' },
                  { 'confidence': 0.95691472,
                    'transcript': 'hello I am Helena I'm a female '
                    '75-year-old I recently experience '
                    'some symptoms such as diarrhea or '
                    'vomiting where I always feel '
                    'nauseous after eating something and '
                    'even though I didn't engage in any '
                    'physical activity I just keep '
                    'sweating more than usual I also felt '
                    'my muscles were weak making it '
                    'impossible for me to lift anything '
                    'at first I assume this this was due '
                    'to my older age but it was not the '
                    'symptoms are now becoming more '
                    'severe' } ] }
```

```
Copy of Copy of s2t.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
[ ] #open the file
with sr.AudioFile(input) as source:
    # listen for the data (load audio to memory)
    audio_data = r.record(source)
    # recognize (convert from speech to text)
    text = r.recognize_google(audio_data)
    print(text)

result:
{ 'alternative': [ { 'confidence': 0.95691472,
                    'transcript': 'hello I am Helena I'm a female '
                    '75-year-old I recently experience some symptoms such as diarrhea or vomiting where I always feel '
                    'nauseous after eating something and even though I did '
                    'not engage in any physical activity I just keep '
                    'sweating more than usual I also felt '
                    'my muscles were weak making it '
                    'impossible for me to lift anything '
                    'at first I assumed this this was due '
                    'to my older age but it was not the '
                    'symptoms are now becoming more '
                    'severe' },
                  { 'confidence': 0.95691472,
                    'transcript': 'hello I am Helena I'm a female '
                    '75-year-old I recently experience some symptoms such as diarrhea or vomiting where I always feel '
                    'nauseous after eating something and even though I did '
                    'not engage in any physical activity I just keep '
                    'sweating more than usual I also felt '
                    'my muscles were weak making it '
                    'impossible for me to lift anything '
                    'at first I assume this this was due '
                    'to my older age but it was not the '
                    'symptoms are now becoming more '
                    'severe' },
                  { 'confidence': 0.95691472,
                    'transcript': 'hello I am Helena I'm a female '
                    '75-year-old I recently experience some symptoms such as diarrhea or vomiting where I always feel '
                    'nauseous after eating something and even though I did '
                    'not engage in any physical activity I just keep '
                    'sweating more than usual I also felt '
                    'my muscles were weak making it '
                    'impossible for me to lift anything '
                    'at first I assume this this was due '
                    'to my older age but it was not the '
                    'symptoms are now becoming more '
                    'severe' } ] }
```

```
[ ]  
  
[  
  {  
    'transcript':  
      'trickling making it difficult for me  
      to breathe my family has noted that '  
      'I frequently have depression and '  
      'other memory loss symptoms such as '  
      'confusions and hallucinations')',  
      'I'm Zara I'm a female that is 67 '  
      'years old my body is aching all over '  
      'my lips tongue fingers and feet are '  
      'trickling making it difficult for me '  
      'to breathe my family has noted that '  
      'I frequently have depression and '  
      'other memory loss symptoms such as '  
      'confusions and allocation')',  
      'I'm Zara I'm a female that is 67 '  
      'years old my body is aching all over '  
      'my lips tongue fingers and feet are '  
      'trickling making it difficult for me '  
      'to breathe my family has noted that '  
      'I frequently have depression and '  
      'other memory loss symptoms such as '  
      'confusions and allocations')',  
      'I'm Zara I'm a female that is 67 '  
      'years old my body is aching all over '  
      'my lips tongue fingers and feet are '  
      'trickling making it difficult for me '  
      'to breathe my family has noted that '  
      'I frequently have depression and '  
      'other memory loss symptoms such as '  
      'confusions and Allsson Nations']',  
      'final': True  
    }  
  ]  
]  
  
I'm Zara I'm a female that is 67 years old my body is aching all over my lips tongue fingers and feet are trickling making it difficult for me to breathe my family has noted that I fre
```

```
[ ] !pip install --upgrade chatterbot_corpus #export the content of the chatbot database  
!pip install --upgrade chatterbot  
  
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/  
Collecting chatterbot_corpus  
  Downloading chatterbot_corpus-1.2.0-py2.py3-none-any.whl (117 kB)  
    117.3/117.3 KB 5.15 MB/s eta 0:00:00  
Collecting PyYAML<4.0,>=3.12  
  Downloading PyYAML-3.13.tar.gz (270 kB)  
    270.6/270.6 KB 18.4 MB/s eta 0:00:00  
Preparing metadata (setup.py) ... done  
Building wheels for collected packages: PyYAML  
  Building wheel for PyYAML (setup.py) ... done  
  Created wheel for PyYAML: filename=PyYAML-3.13-cp38-cp38-linux_x86_64.whl size=43898 sha256=18fc73bf6f24422a59aa825c9627d1a8dbf11772dafab63c411fbcba82a0d0  
  Stored in directory: /root/.cache/pip/wheels/db/f2/07/5e58b12bc1255c3f6a0aca8984905a08ec203db84a3c52c0  
Successfully built PyYAML  
Installing collected packages: PyYAML, chatterbot_corpus  
  Attempting uninstall: PyYAML  
    Found existing installation: PyYAML 6.0  
    Uninstalling PyYAML-6.0:  
      Successfully uninstalled PyYAML-6.0  
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.  
dask 2022.2.1 requires pyyaml>5.3.1, but you have pyyaml 3.13 which is incompatible.  
Successfully installed PyYAML-3.13 chatterbot_corpus-1.2.0  
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/  
Collecting chatterbot  
  Downloading chatterbot-1.0.5-m2-mc1-none-any.whl (67 kB)
```

```
Copy of Doctor side_chat_bot.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Successfully installed blis-0.2.4 chatterbot-1.0.5 mathparse-0.1.2 pint-0.20.1 plac-0.9.6 preshed-2.0.1 pymongo-3.13.0 python-dateutil-2.7.5 pyyaml-5.1.2 spacy-2.1.9 sqlalchemy-1.2.18

[]
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer #simplifying the process of loading the examples

my_bot=ChatBot(name='Pybot',read_only=True,logic_adapters=['chatterbot.logic.Mathematicalevaluation', 'chatterbot.logic.BestMatch']) #easy to generate automated response to user input

[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data] /root/nltk_data...
[nltk_data] Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

[] small_talk=['hi there!','What is your name?','How old are you?','\n Hello! This is CareBot','\n hi I am Sharon well I am 82 years old and a female I recently had some symptoms like drowsiness headache but I felt like I lost all of my energy completely at first I
Math_talk_1=['what is Your Gender','How long was this been happening?']
Math_talk_2=['Do You feel sleepy?','Is this the first time do you feel like this?']

[]
```

```
Copy of Doctor side_chat_bot.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
list_trainer=ListTrainer(my_bot)
for item in (small_talk,Math_talk_1,Math_talk_2):
    list_trainer.train(item)

List Trainer: [=====] 100%
List Trainer: [=====] 100%
List Trainer: [=====] 100%

[] print(my_bot.get_response("hi there!"))
What is your name/how old are you/hello! This is CareBot

[] print(my_bot.get_response("Hello! This is CareBot"))
hi Iam Sharon well Iam 82 years old and a female I recently had some symptoms like drowsiness headache but I felt like I lost all of my energy completely at first I did not notice the
```

```
Copy_of_chatbot_pca.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Knowing about the dataset

IMPORTING THE DATASET

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('/content/Electrolyte imbalances in an ED-datafile-tazmini-012019.csv')
df.shape
print(df)

ID-number Age Gender SMA SGU Glucose corrected sodium SK \
0 1 91.0 K 140.0 6.0 140.17 3.6
1 1 92.0 K 136.0 5.2 135.83 4.8
2 2 92.0 K 141.0 5.3 140.87 4.1
3 2 95.0 K 144.0 6.7 144.47 4.0
4 3 89.0 K 136.0 13.9 139.56 4.0
... ..
62986 31962 NaN M 139.0 5.0 139.74 4.7
62987 31963 NaN M 141.0 5.8 141.09 3.6
62988 31964 NaN K 143.0 4.4 142.49 3.8
62989 31965 NaN K 144.0 6.0 144.17 3.9
62990 31966 38.0 M 139.0 NaN NaN 4.5

SGA SGLB Albumin corrected calcium SPO SNG Free calcium \
0 2.53 37.0 2.316 1.09 0.75 NaN
1 2.39 41.0 2.396 1.46 0.82 NaN
2 2.39 37.0 2.476 NaN NaN NaN
```

Copy_of_chatbot_pca.ipynb

```
[ ] 2.36 39.0 2.406 1.13 0.89 1.18
    4 2.34 37.0 2.426 NaN NaN NaN
...
62986 2.24 39.0 2.286 NaN NaN 1.19
62987 2.23 45.0 2.156 NaN NaN 1.13
62988 2.39 39.0 2.436 NaN NaN NaN
62989 2.12 42.0 2.386 NaN NaN 1.14
62990 NaN NaN NaN NaN NaN NaN

Department Death no days Readmission no days Main condition \
0 KIR 2615.0 NaN S328
1 MED 1956.0 NaN F059
2 MED NaN NaN M189
3 MED NaN NaN M390
4 MED 2691.0 NaN I209
...
62986 NaN NaN NaN NaN
62987 NaN NaN NaN NaN
62988 NaN NaN NaN NaN
62989 NaN NaN NaN NaN
62990 KIR NaN NaN K601

Other conditions
0 I10
1 E43 N179 F067 I10 E86
2 H772
3 R296 E86
4
...
62986 NaN NaN
62987 NaN NaN
62988 NaN NaN
62989 NaN NaN
62990 K601
```

Copy_of_chatbot_pca.ipynb

```
[ ] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('/content/Electrolyte imbalances in an ED-datafile-Tazmini-032019.csv')
print(df)
df.shape
```

```
ID-number Age Gender SGA SGLU Glucose corrected sodium SK \
0 1 91.0 K 140.0 6.0 140.17 3.6
1 1 92.0 K 136.0 5.2 135.83 4.8
2 2 92.0 K 141.0 5.3 140.87 4.1
3 2 95.0 K 144.0 6.7 144.47 4.0
4 3 89.0 K 136.0 11.9 139.56 4.0
...
62986 31962 NaN M 139.0 5.0 138.74 4.7
62987 31963 NaN M 141.0 5.8 141.09 3.6
62988 31964 NaN K 143.0 4.4 142.49 3.8
62989 31965 NaN K 144.0 6.0 144.17 3.9
62990 31966 38.0 M 139.0 NaN NaN 4.5

SGA SGLU Albumin corrected calcium SPO SMC Free calcium \
0 2.23 37.0 2.316 1.09 0.75 NaN
1 2.39 41.0 2.396 1.46 0.82 NaN
2 2.39 37.0 2.476 NaN NaN NaN
3 2.36 39.0 2.406 1.13 0.89 1.18
4 2.34 37.0 2.426 NaN NaN NaN
...
62986 2.24 39.0 2.286 NaN NaN 1.19
62987 2.23 45.0 2.156 NaN NaN 1.13
```

```
Copy_of_chatbot_pca.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
62988 2.39 39.0 2.436 NaN NaN NaN
62989 2.32 42.0 2.306 NaN NaN NaN
62990 NaN NaN NaN NaN NaN NaN
Department Death nr days Readmission nr days Main condition \
0 KTR 2615.0 NaN 5328
1 MED 1956.0 NaN F059
2 MED NaN NaN R109
3 MED NaN NaN R190
4 MED 2691.0 NaN 1209
...
62986 NaN NaN NaN NaN
62987 NaN NaN NaN NaN
62988 NaN NaN NaN NaN
62989 NaN NaN NaN NaN
62990 KTR NaN NaN K603
[62991 rows x 18 columns]
(62991, 18)
```

```
Copy_of_chatbot_pca.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import plotly.graph_objs as go
import plotly.offline as py
import random
'''from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA'''
pd.set_option('display.max_columns', None) #change the default no of rows to be displayed
df = pd.read_csv('/content/Electrolyte Imbalances in an ED-datafile-Tazmini-032019.csv')
print(df)
ID-number Age Gender SMA SGU Glucose corrected sodium SE \
0 1 91.0 K 140.0 6.0 140.17 3.6
1 1 92.0 K 136.0 5.2 135.83 4.8
2 2 92.0 K 141.0 5.3 140.87 4.1
3 2 95.0 K 144.0 6.7 144.47 4.0
4 3 89.0 K 136.0 13.9 139.36 4.0
...
62986 31962 NaN M 139.0 5.0 138.74 4.7
62987 31963 NaN M 141.0 5.8 141.49 3.6
62988 31964 NaN K 143.0 4.4 142.49 3.8
62989 31965 NaN K 144.0 6.0 144.17 3.9
62990 31966 38.0 M 139.0 NaN NaN 4.5
SCA SABL Albumin corrected calcium SPO SMG Free calcium \
0 2.23 37.0 2.316 1.49 0.75 NaN
1 2.39 41.0 2.396 1.46 0.82 NaN
2 2.39 37.0 2.476 NaN NaN NaN
3 2.36 39.0 2.496 1.13 0.89 1.18
4 2.34 37.0 2.426 NaN NaN NaN
```

Copy of chatbot_pca.ipynb

```

4      2.34      37.0      2.426      NaN      NaN      NaN
...      ...      ...      ...      ...      ...      ...
62986      2.24      39.0      2.286      NaN      NaN      1.19
62987      2.23      45.0      2.156      NaN      NaN      1.13
62988      2.39      39.0      2.436      NaN      NaN      NaN
62989      2.32      42.0      2.386      NaN      NaN      1.14
62990      NaN      NaN      NaN      NaN      NaN      NaN

Department      Death nr days      Readmission nr days      Main condition
0      KIR      2615.0      NaN      S328
1      MED      1956.0      NaN      F059
2      MED      NaN      NaN      M109
3      MED      NaN      NaN      N390
4      MED      2691.0      NaN      I209
...      ...      ...      ...      ...
62986      NaN      NaN      NaN      NaN
62987      NaN      NaN      NaN      NaN
62988      NaN      NaN      NaN      NaN
62989      NaN      NaN      NaN      NaN
62990      KIR      NaN      NaN      K601

Other conditions
0      I10
1      E43 N179 F067 I10 E86
2      M772
3      R296 E86
4
...      ...
62986      NaN
62987      NaN
62988      NaN
62989      NaN
62990      K603

```

```

[62991 rows x 18 columns]

[ ] rows = len(df.axes[0])
print(rows)

62991

[ ] cols = len(df.axes[1])
print(cols)

18

[ ] df.head()

   ID-   Age  Gender  SNA  SGLU  corrected  Glucose  SK  SCA  SALB  Albumin  corrected  SPO  SMG  Free  Department  Death nr  Readmission nr  Main  Other
number  days  days  days  days  sodium    sodium  days  days  days  calcium    calcium  days  days  calcium  condition  condition  conditions
0      1    91.0    K   140.0    6.0      140.17    3.6    2.23    37.0      2.316    1.09    0.75    NaN    KIR      2615.0      NaN      S328      I10
1      1    92.0    K   136.0    5.2      135.83    4.8    2.39    41.0      2.396    1.46    0.82    NaN    MED      1956.0      NaN      F059      E43 N179 F067
2      2    92.0    K   141.0    5.3      140.87    4.1    2.39    37.0      2.476    NaN      NaN      NaN    MED      NaN      NaN      M109      I10 E86
3      2    95.0    K   144.0    6.7      144.47    4.0    2.36    39.0      2.406    1.13    0.89    1.18    MED      NaN      NaN      N390      R296 E86
4      3    89.0    K   136.0   13.9      139.56    4.0    2.34    37.0      2.426    NaN      NaN      NaN    MED      2691.0      NaN      I209

[ ] df.isna().sum().sort_values(ascending=False)

```

Copy_of_chatbot_pca.ipynb

```
[ ] SPO 55368
    SNG 54479
    Readmission nr days 50063
    Free calcium 48156
    Death nr days 41145
    Albumin corrected calcium 17316
    SCA 17264
    Glucose corrected sodium 19524
    SGU 15501
    SALB 10347
    Main condition 285
    Department 285
    Other conditions 285
    SK 261
    SNA 62
    Age 39
    Gender 0
    ID-number 0
    dtype: int64

[ ] df=df.apply(lambda x: x.fillna(x.value_counts().index[0]))

[ ] df.isna().sum().sort_values(ascending=False)

ID-number 0
Age 0
Main condition 0
Readmission nr days 0
Death nr days 0
Department 0
Free calcium 0
SNG 0
```

Copy_of_chatbot_pca.ipynb

```
[ ] Readmission nr days 0
    Death nr days 0
    Department 0
    Free calcium 0
    SNG 0
    SPO 0
    Albumin corrected calcium 0
    SALB 0
    SCA 0
    SK 0
    Glucose corrected sodium 0
    SGU 0
    SNA 0
    Gender 0
    Other conditions 0
    dtype: int64

df.dtypes

ID-number      int64
Age            float64
Gender         object
SNA            float64
SGU            float64
Glucose corrected sodium  float64
SK              float64
SCA             float64
SALB            float64
Albumin corrected calcium float64
SPO             float64
SNG             float64
Free calcium    float64
Department     object
```

Inbox (9:27) - porkodicharulati... x Copy_of_chatbot_pca.ipynb - Col... x Copy of Copy of k2t.ipynb - Col... x Copy of Doctor side_chat_bot.ip... x

colab.research.google.com/drive/1r1hIk-Vin7J5dGft-2qmr8TWfPX3arc?usp=sharing

Copy_of_chatbot_pca.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Connect Editing

```
[ ] df.dtypes
```

ID-number	int64
Age	float64
Gender	object
SMA	float64
SGLU	float64
Glucose corrected sodium	float64
SK	float64
SCA	float64
SALB	float64
Albumin corrected calcium	float64
SPO	float64
SMG	float64
Free calcium	float64
Department	object
Death nr days	float64
Readmission nr days	float64
Main condition	object
Other conditions	object
dtype:	object

```
[ ] for column in df.columns:  
    df[column] = df[column].astype('category')
```

```
[ ] df.dtypes
```

ID-number	category
Age	category
Gender	category
SMA	category
SGLU	category

Type here to search

18:54 01-02-2023

Inbox (9:27) - porkodicharulati... x Copy_of_chatbot_pca.ipynb - Col... x Copy of Copy of k2t.ipynb - Col... x Copy of Doctor side_chat_bot.ip... x

colab.research.google.com/drive/1r1hIk-Vin7J5dGft-2qmr8TWfPX3arc?usp=sharing

Copy_of_chatbot_pca.ipynb

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Connect Editing

```
[ ] df.dtypes
```

ID-number	int64
Age	float64
Gender	object
SMA	float64
SGLU	float64
Glucose corrected sodium	float64
SK	float64
SCA	float64
SALB	float64
Albumin corrected calcium	float64
SPO	float64
SMG	float64
Free calcium	float64
Department	object
Death nr days	float64
Readmission nr days	float64
Main condition	object
Other conditions	object
dtype:	object

```
[ ] for column in df.columns:  
    df[column] = df[column].astype('category')
```

```
[ ] df.dtypes
```

ID-number	category
Age	category
Gender	category
SMA	category
SGLU	category

Type here to search

18:55 01-02-2023

The screenshot shows a Jupyter Notebook titled 'Copy_of_chatbot_pca.ipynb'. The left sidebar lists variables: glucose corrected sodium, SK, SCA, SALB, Albumin corrected calcium, SPO, SNG, free calcium, Department, Death nr days, Readmission nr days, Main condition, Other conditions, and dtype: object. The main cell contains the following code:

```
df['ID-number'] = df['ID-number'].cat.codes
df['Age'] = df['Age'].cat.codes
df['Gender'] = df['Gender'].cat.codes
df['SMA'] = df['SMA'].cat.codes
df['SGLU'] = df['SGLU'].cat.codes
df['Glucose corrected sodium'] = df['Glucose corrected sodium'].cat.codes
df['SK'] = df['SK'].cat.codes
df['SCA'] = df['SCA'].cat.codes
df['SALB'] = df['SALB'].cat.codes
```

The screenshot shows the same Jupyter Notebook with the following code in the main cell:

```
df['SCA'] = df['SCA'].cat.codes
df['SALB'] = df['SALB'].cat.codes
df['Albumin corrected calcium'] = df['Albumin corrected calcium'].cat.codes
df['SPO'] = df['SPO'].cat.codes
df['SNG'] = df['SNG'].cat.codes
df['free calcium'] = df['free calcium'].cat.codes
df['Department'] = df['Department'].cat.codes
df['Death nr days'] = df['Death nr days'].cat.codes
df['Readmission nr days'] = df['Readmission nr days'].cat.codes
df['Main condition'] = df['Main condition'].cat.codes
df['Other conditions'] = df['Other conditions'].cat.codes
```

Below the code, the output of `df.head()` is displayed as a table:

	ID-number	Age	Gender	SMA	SGLU	Glucose corrected sodium	SK	SCA	SALB	Albumin corrected calcium	SPO	SNG	free calcium	Department	Death nr days	Readmission nr days	Main condition	Other conditions
0	0	78	0	38	48	1254	19	82	24	73	82	48	44	0	2417	0	2590	8169
1	0	79	0	34	40	950	31	98	28	81	119	55	44	1	1938	0	622	3829
2	1	79	0	39	41	1303	24	98	24	89	76	55	44	1	3	0	1788	15947
3	1	82	0	42	55	1555	23	95	26	82	86	62	45	1	3	0	2256	18428
4	2	76	0	34	127	1211	23	93	24	84	76	55	44	1	2449	0	1015	0

Copy_of_chatbot_pca.ipynb

```

df['SCA'] = df['SCA'].cat.codes
df['SALB'] = df['SALB'].cat.codes
df['Albumin corrected calcium'] = df['Albumin corrected calcium'].cat.codes
df['SPO'] = df['SPO'].cat.codes
df['SNG'] = df['SNG'].cat.codes
df['Free calcium'] = df['Free calcium'].cat.codes
df['Department'] = df['Department'].cat.codes
df['Death nr days'] = df['Death nr days'].cat.codes
df['Readmission nr days'] = df['Readmission nr days'].cat.codes
df['Main condition'] = df['Main condition'].cat.codes
df['Other conditions'] = df['Other conditions'].cat.codes

[ ] df.head()

```

ID-number	Age	Gender	SNA	SGLU	Glucose	corrected sodium	SK	SCA	SALB	Albumin corrected calcium	SPO	SNG	Free calcium	Department	Death nr days	Readmission nr days	Main condition	Other conditions
0	0	78	0	38	48	1284	19	82	24	73	82	48	44	0	2417	0	2590	8169
1	0	79	0	34	40	950	31	98	28	81	119	55	44	1	1938	0	622	3829
2	1	79	0	39	41	1303	24	98	24	89	76	55	44	1	3	0	1788	15947
3	1	82	0	42	55	1555	23	95	26	82	86	62	45	1	3	0	2256	18428
4	2	76	0	34	127	1211	23	93	24	84	76	55	44	1	2449	0	1015	0

Copy_of_chatbot_pca.ipynb

PCA

```

[ ] #30 PCA #reducing the dimension
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import plotly.graph_objs as go
import plotly.offline as py
import random

'''from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA'''
pd.set_option('display.max_columns', None)
#print(df)
from sklearn.preprocessing import StandardScaler

standardized = StandardScaler()

standardized.fit(df)

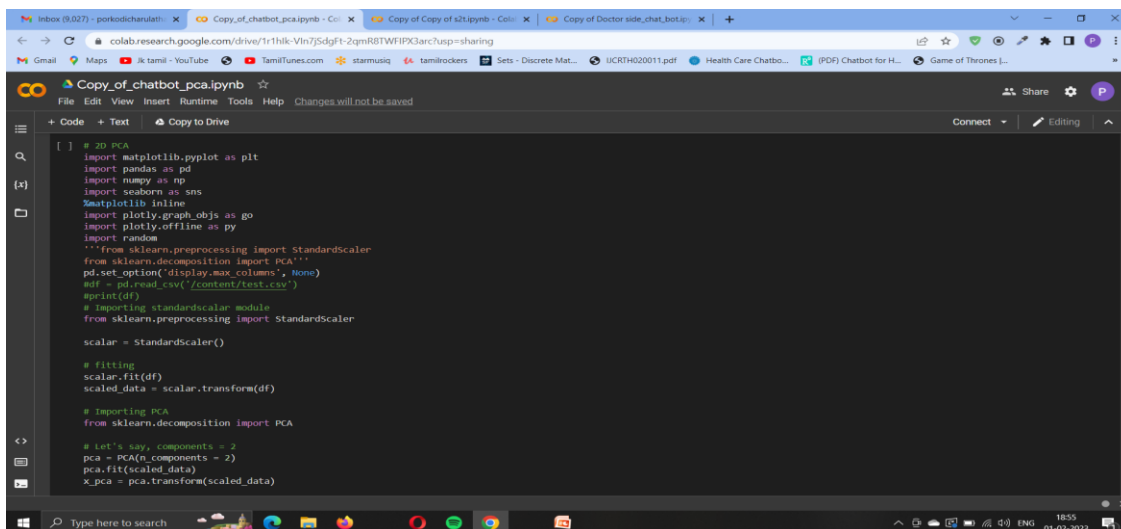
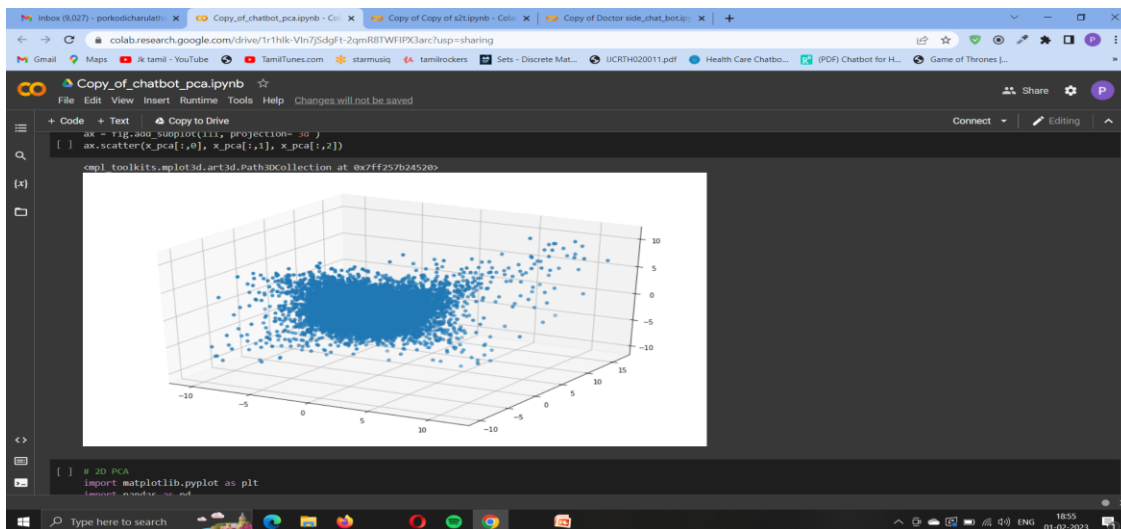
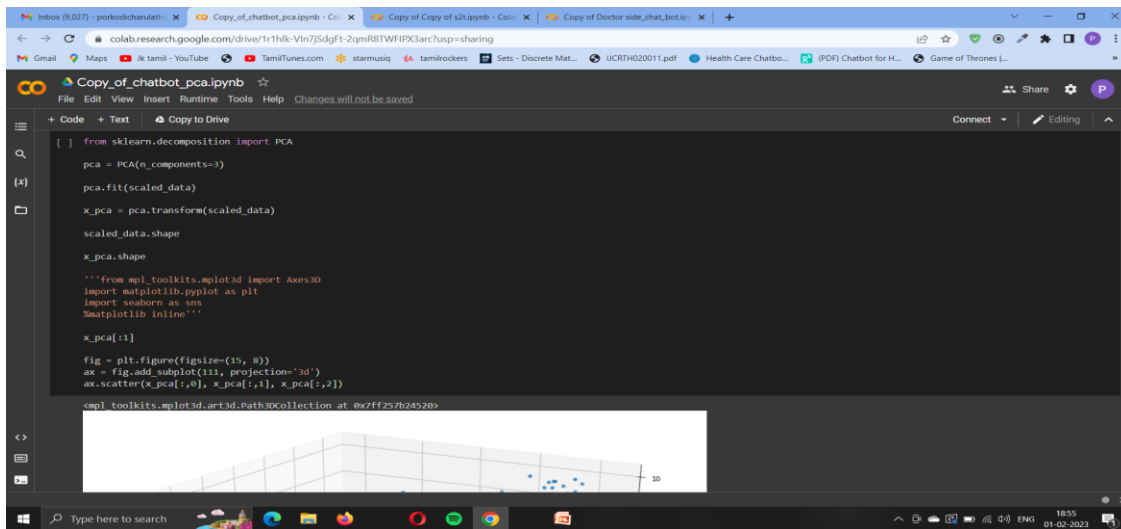
scaled_data = standardized.transform(df)

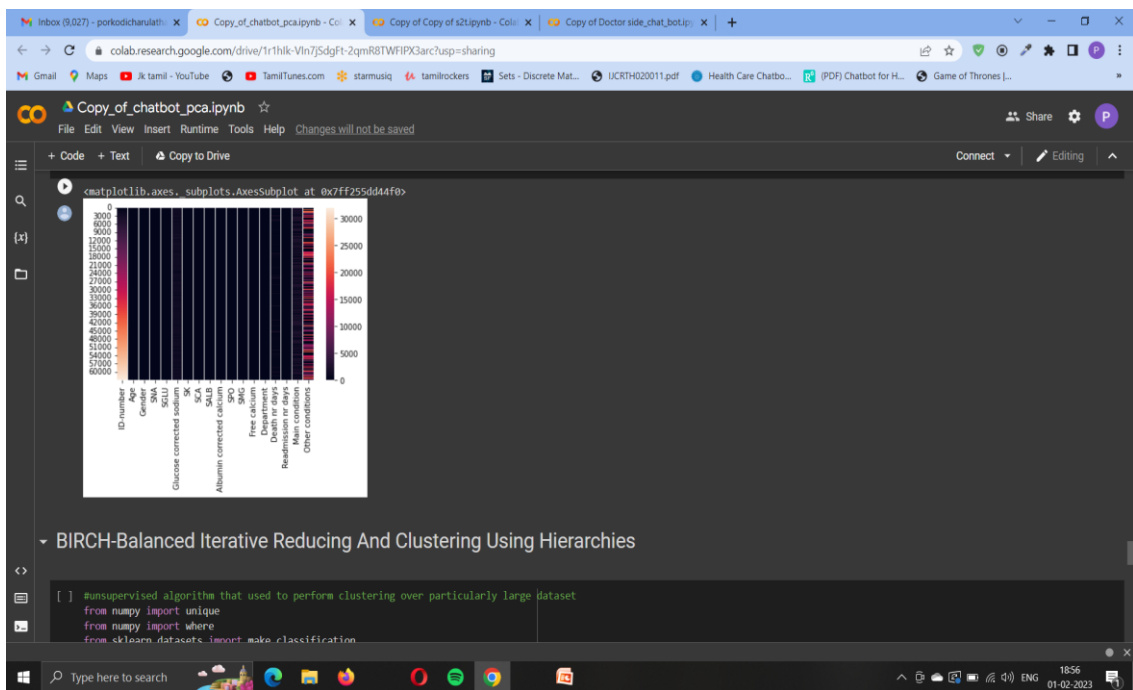
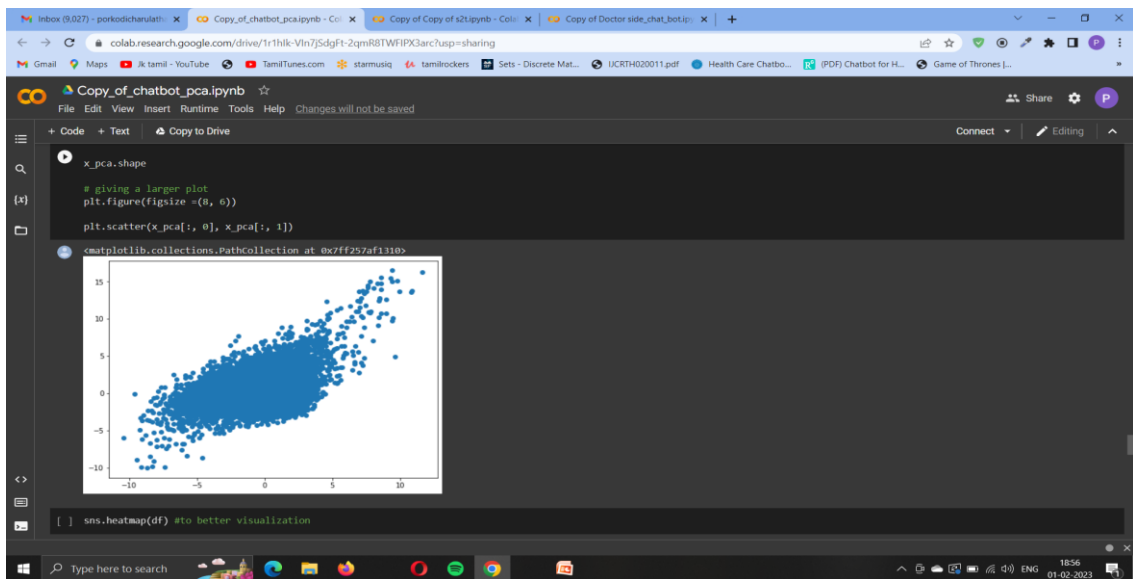
from sklearn.decomposition import PCA

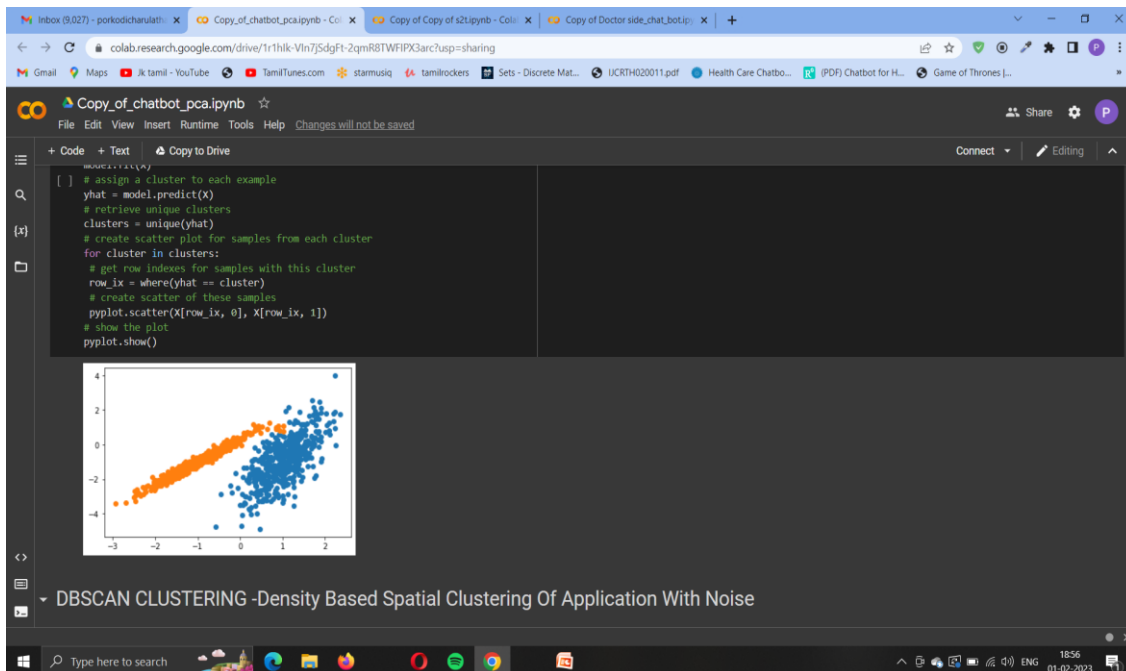
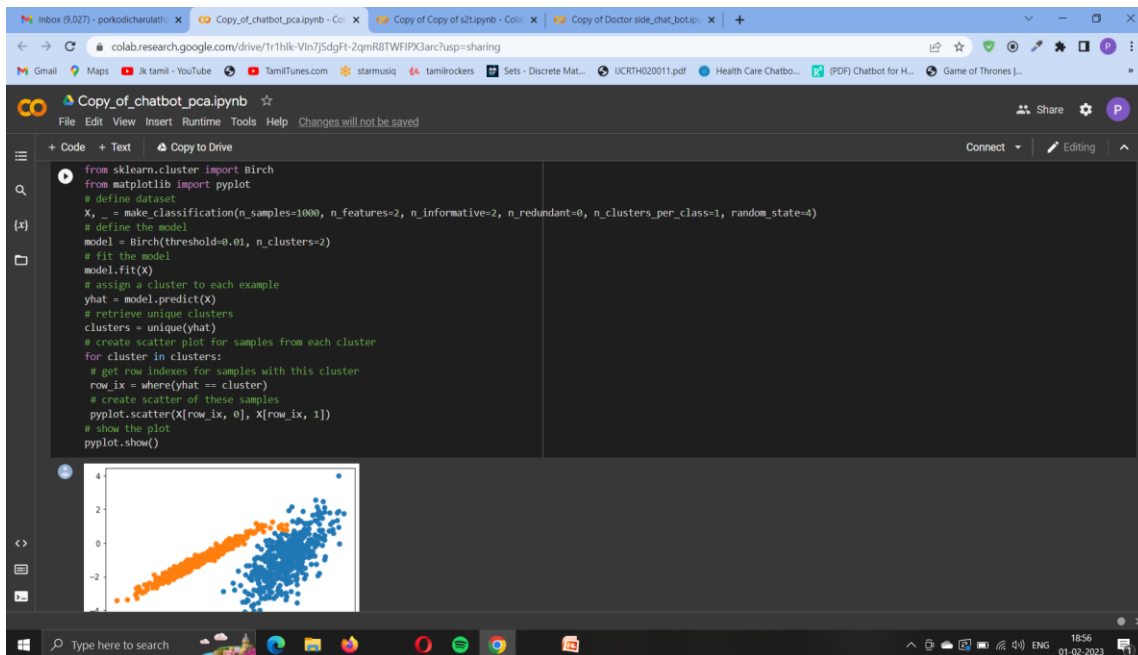
pca = PCA(n_components=3)

pca.fit(scaled_data)

```







Copy_of_chatbot_pca.ipynb

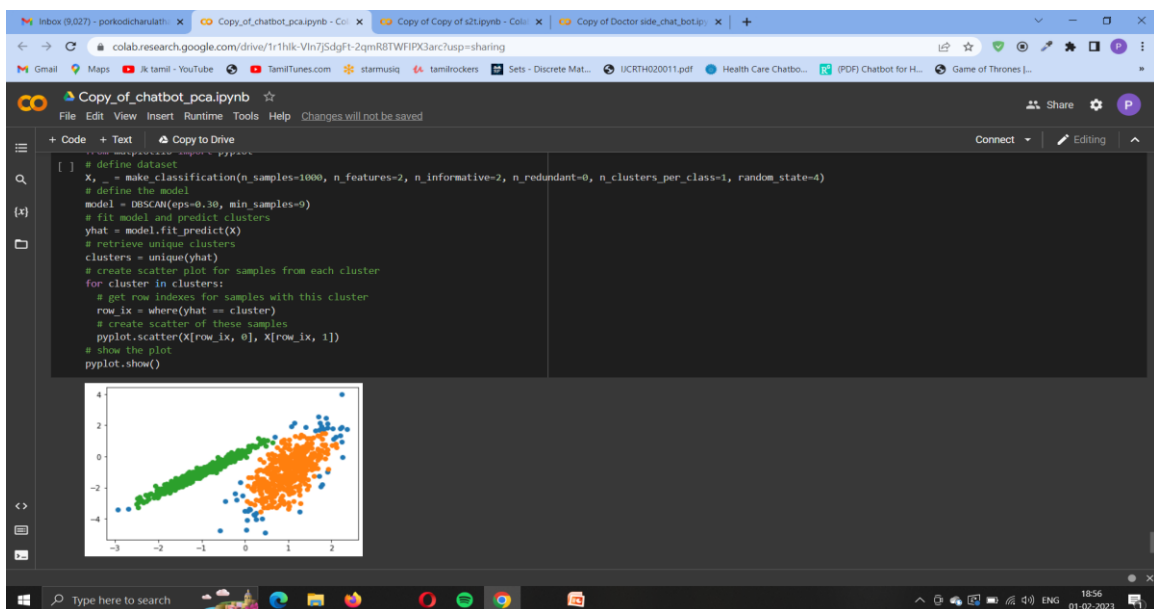
DBSCAN CLUSTERING -Density Based Spatial Clustering Of Application With Noise

```

# dbscan clustering finds clusters during noisy dataset circumstances
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import DBSCAN
from matplotlib import pyplot

# define dataset
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0, n_clusters_per_class=1, random_state=4)
# define the model
model = DBSCAN(eps=0.30, min_samples=9)
# fit model and predict clusters
yhat = model.fit_predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
    # get row indexes for samples with this cluster
    row_ix = where(yhat == cluster)
    # create scatter of these samples
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()

```



```

Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sharon>cd C:\Users\sharon\calorie

C:\Users\sharon\calorie>pip install django-filter
Collecting django-filter
  Downloading django-filter-23.1-py3-none-any.whl (91 kB)
    Requirement already satisfied: Django>=3.2 in c:\users\sharon\appdata\local\programs\python\python311\lib\site-packages (from django-filter) (4.2)
    Requirement already satisfied: asgiref<4,>=3.6.0 in c:\users\sharon\appdata\local\programs\python\python311\lib\site-packages (from Django>=3.2->django-filter) (3.6.0)
    Requirement already satisfied: sqlparse>=0.3.1 in c:\users\sharon\appdata\local\programs\python\python311\lib\site-packages (from Django>=3.2->django-filter) (0.4.3)
    Requirement already satisfied: tzdata in c:\users\sharon\appdata\local\programs\python\python311\lib\site-packages (from Django>=3.2->django-filter) (2022.7)
Installing collected packages: django-filter
Successfully installed django-filter-23.1

C:\Users\sharon\calorie>py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

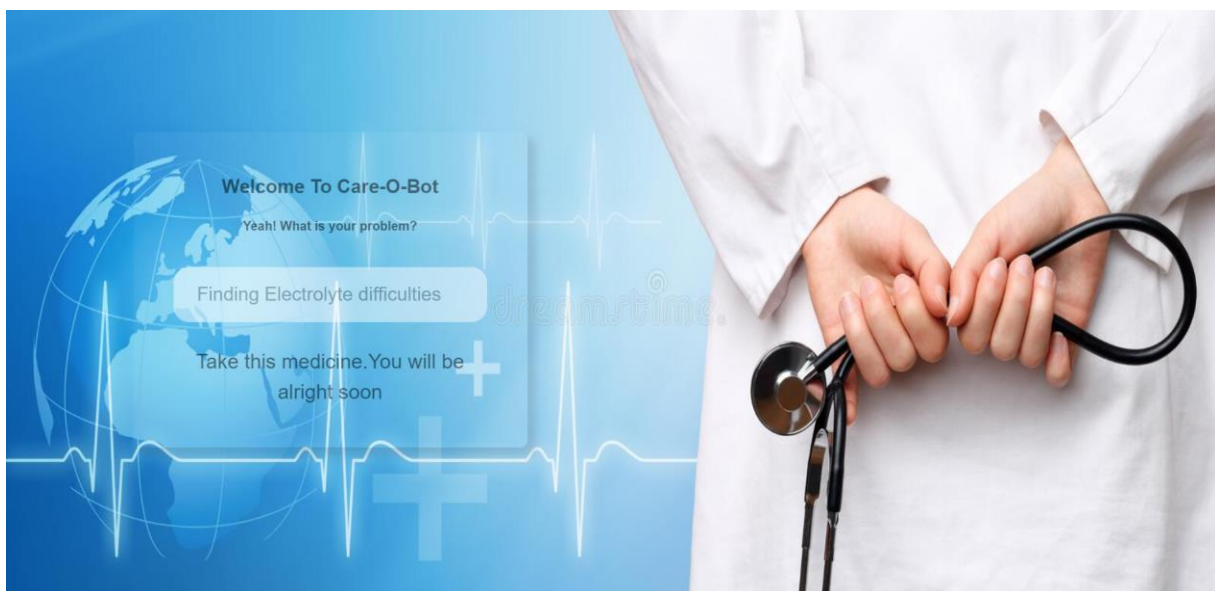
```

```
Command Prompt - py mana X + v
db.models.BigAutoField'.
Fityfeed.Customer: (models.WM42) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the FityfeedConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.
db.models.BigAutoField'.
Fityfeed.Fooditem: (models.WM42) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the FityfeedConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.
db.models.BigAutoField'.
Fityfeed.UserFooditem: (models.WM42) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
HINT: Configure the DEFAULT_AUTO_FIELD setting or the FityfeedConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.
db.models.BigAutoField'.

System check identified 4 issues (0 silenced).

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): auth.
Run 'python manage.py migrate' to apply them.
April 10, 2023 - 14:32:21
Django version 4.2, using settings 'calorieCalc.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[10/Apr/2023 14:32:40] "GET / HTTP/1.1" 302 0
[10/Apr/2023 14:32:40] "GET /login/texts/ HTTP/1.1" 200 3023
[10/Apr/2023 14:32:40] "GET /static/css/main.css HTTP/1.1" 200 289
Not Found: /favicon.ico
[10/Apr/2023 14:32:43] "GET /favicon.ico HTTP/1.1" 404 4285
[10/Apr/2023 14:32:57] "POST /login/texts/ HTTP/1.1" 200 3006
[10/Apr/2023 14:32:59] "GET /register/ HTTP/1.1" 200 3921
[10/Apr/2023 14:33:32] "POST /register/ HTTP/1.1" 302 0
[10/Apr/2023 14:33:32] "GET /login/ HTTP/1.1" 200 3001
[10/Apr/2023 14:33:32] "GET /static/css/main.css HTTP/1.1" 304 0
[10/Apr/2023 14:33:38] "POST /login/ HTTP/1.1" 302 0
[10/Apr/2023 14:33:38] "GET / HTTP/1.1" 302 0
[10/Apr/2023 14:33:39] "GET /user/ HTTP/1.1" 200 10012
```



REGISTER ACCOUNT

Register Account

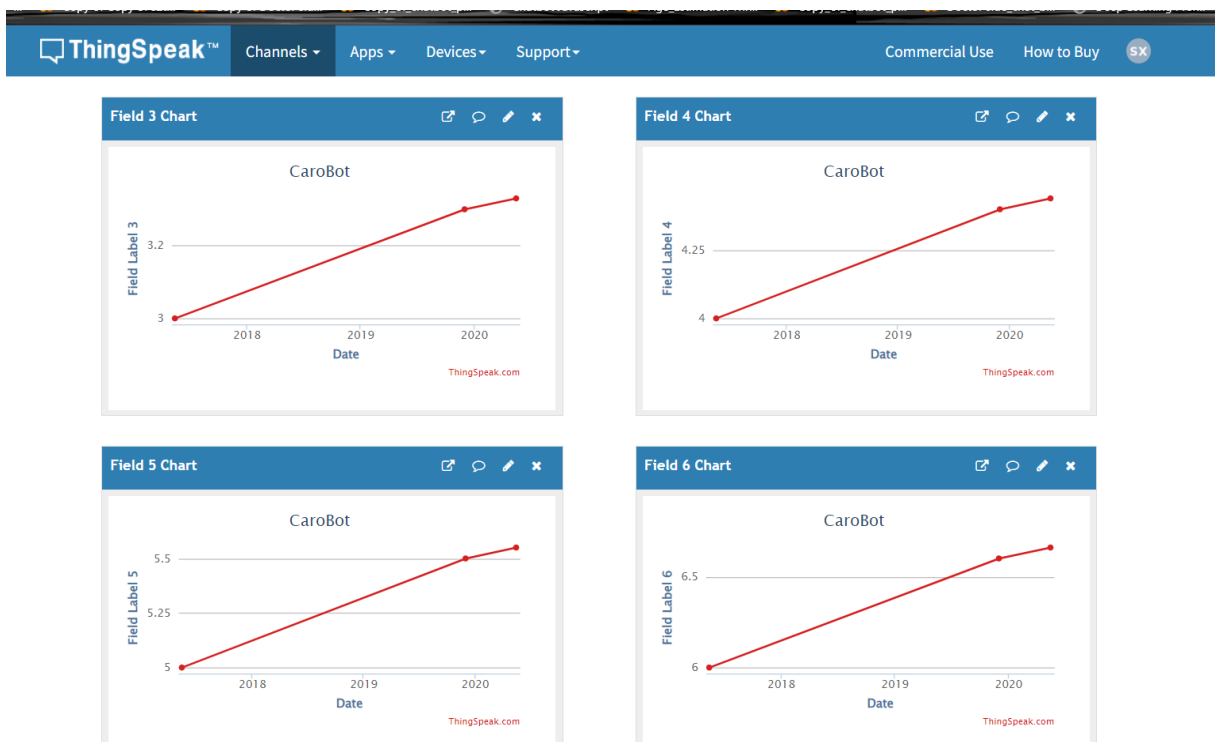
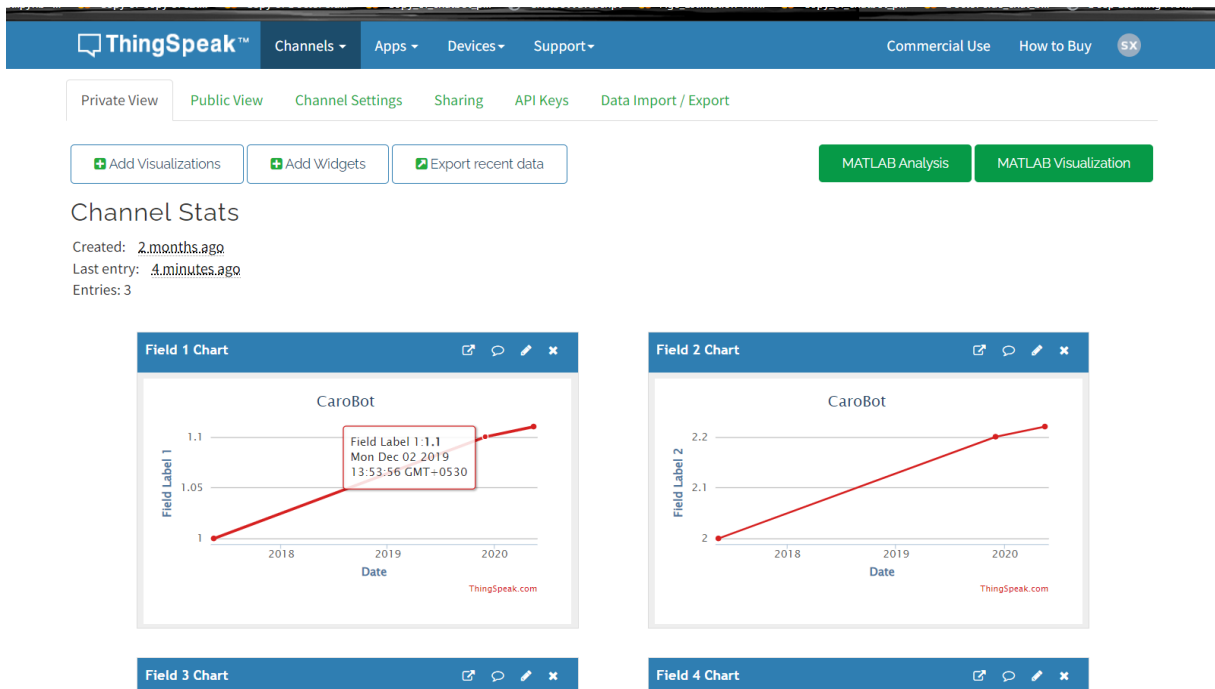
Already have an account? [Login](#)

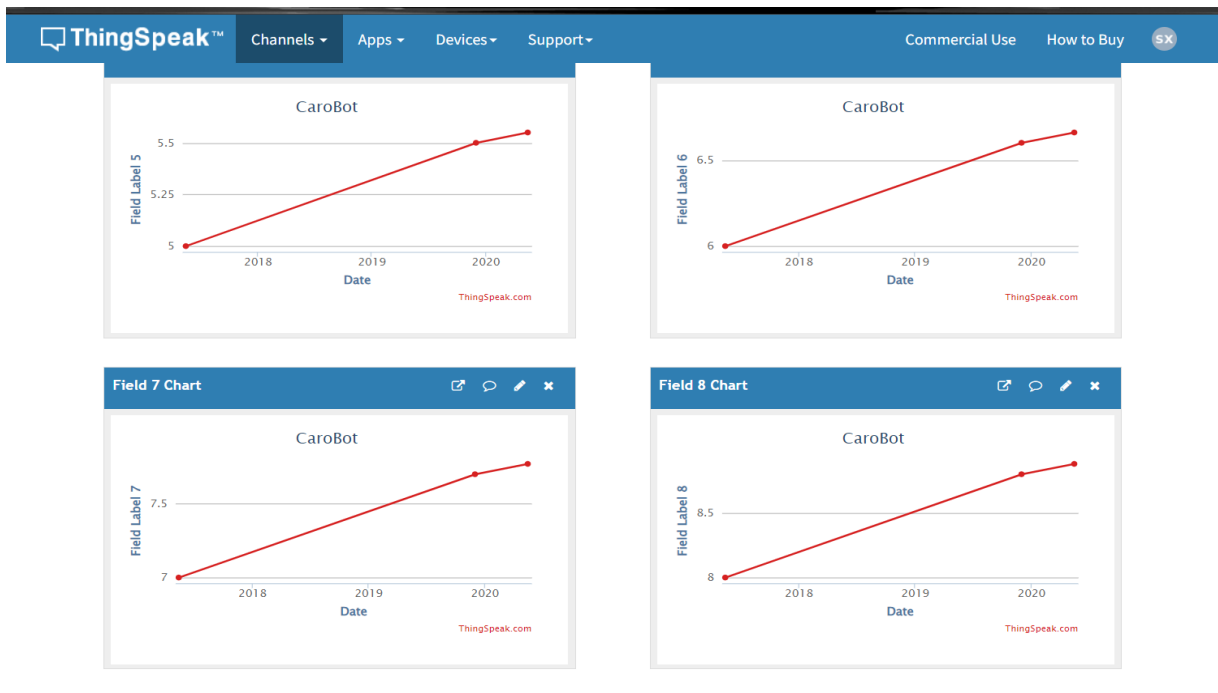


LOGIN

Login

Don't have an account? [Sign Up](#)
Forgot password? [Click here](#)





C. RESEARCH PAPER

HEALTHCARE CHATBOT FOR ELECTROLYTE IMBALANCE

Sharon Swetha X
Information Technology
Sathyabama University
Chennai, India
sharonswetha3131@gmail.com

Porkodi S
Information Technology
Sathyabama University
Chennai, India
porkodicharulatha@gmail.com

Dr.J.Jabez,M.E.,Ph.d.,
Dept. Of Information
Technology
Sathyabama University
Chennai, India
jabezme@gmail.com

ABSTRACT: In the world's population, there are 703 million people aged 60 to 70, which is a large figure, considering that by the year 2050 older adults would outnumber the children under the age of five, So the point is that there are more in number. With aging there are degenerative changes in many organs. These changes may be associated with an inability to excrete a concentrated or dilute urine, ammonium, sodium, or potassium. Hyponatremia and hypokalemia are the most common electrolyte abnormalities found among the elderly population taking diuretics. Both illnesses are linked to both short- and long-term morbidity and mortality. The development of new technologies has left the elderly behind and unable to keep up. But in this age group only some number of people will have the knowledge of current technology so the remaining people will rely on the others. However, as a result of this, there is a significant digital divide among the elderly. So, there is a necessity to develop a new technology to aid in their better care. Therefore, in order to address these issues, we have proposed a solution in which we want to develop a Voice AI chatbot using Natural Language Processing model to help them to become independent. Additionally, we'll use the cloud to put up

reminders for their doctors' recommended diet.

Keywords: *NLP, Care-O-Bot, Voice chat bot, Old age, Electrolyte Imbalance, Artificial Intelligence, Machine Learning, Deep Learning.*

1. INTRODUCTION

According to projections, the global senior population will double by 2050 as it keeps growing. As humans age, several organs experience degenerative changes. Glomerular filtration rate (GFR) and renal plasma flow both decline beyond the age of forty as cortical glomerulosclerosis advances. Failure to remove urine that is concentrated or dilute, ammonium, salt, or potassium could result in these changes. You can get electrolytes from food, drink, and supplements. Electrolytes get their name because they discharge an electrical charge when they dissolve in body fluids. So test results may be impacted by age, sex, and pre-existing medical issues. And several techniques are frequently used in laboratories to measure electrolytes. This implies that outcomes may differ. So, every elder will have family doctors as because of their old age their health may degrade and have to get check-up frequently. So of course, the doctors will have their health chart with them. But most of the elders aren't aware of the electrolyte loss in their

body. So, they will communicate their symptoms to the doctor through the voice chatbot that we create and using NLP remove the noises and detect whether humans are talking or not, so after that the doctor will diagnose their problem and contact them immediately and come up with their prescription. Suppose a patient has electrolyte imbalance; for example, if the patient has a low potassium level, the doctor will figure out their electrolyte imbalance chart in the cloud. and will notify them to consume more apple juice and banana to maintain the electrolyte balance, because the patient won't be aware that their body is experiencing an electrolyte imbalance, the chatbot reminds them.

2. LITERATURE REVIEW

Rohit Binu Mathew [1] et.al conducted experiments We can observe that people don't value their health. Normally, we find going to hospitals for checkups to be time-consuming. Chatbot diagnoses the illness and suggests a course of treatment using these extracted symptoms. The K-nearest neighbor algorithm is the machine learning algorithm used here (KNN). This demonstrates unequivocally that a medical chatbot can diagnose patients with a conversational approach, simple symptom analysis, and the aid of natural language processing.

Kandpal P [2] et.al in which working on chatbots has provided us with a brief introduction to the field of virtual assistance and how it will affect our lives in the future. It is clear that chatbots will play a significant role in both large and small businesses as well as other organizations after looking at the current work and development enterprises and other organizations are making in this area. This paper covered the installation of various packages, code workflow, creating data in the intents file,

training our model, and obtaining appropriate results.

Papiya Mahajan [3] et.al proceeded with an approach in our proposed system, the user can communicate with the bot regarding their query via voice or text. The system responds to the queries using an expert system. The user can also see the doctors who are treating that specific illness. This solution enables numerous users to access online counseling sessions. The chatbot's data is saved as pattern templates in the database. The bot will provide analgesics and diet plans, instructing you on the best items to eat in accordance with the ailment.

Manthirathne [4] et.al gathered data on different aspects of the study's technique and propose a more effective method for anticipating muscle cramping. To predict muscular anomalies, the system makes use of the wave patterns created by the muscles and analyses them using electromyograms. Real-time EMG data taken from a person is combined with a defined benchmark dataset and used with a trained algorithm to identify the wave patterns. The person will receive a warning or an alert if the identified wave patterns resemble wave patterns that cause muscle abnormalities. The patient will also receive recovery instructions for the muscle abnormalities. Time series prediction is the foundation of the suggested solution. Databases on both a local and cloud basis house the real-time data. Through user-friendly interfaces, the user is given the necessary information.

Kshitij Thakre [5] et.al aimed to predict. The user can text chat with the bot in our proposed system to discuss the query. For query responses, the system makes use of an expert system. Additionally, users get access to a list of physicians who specialize in that condition. For online therapy sessions, this method can be used by many people. In the

form of a pattern template, the chatbot's data is kept in the database.

Athulya N [6] et.al Above sentiment analysis could be used for politics. In our system, the user can communicate with the chatbot via text, and it will respond to them both verbally and in writing. If a user is interacting with the chatbot, the bot responds to their questions by identifying the ailment. The bot suggests treatments for the user's problems and also suggests specialists that specialize in treating certain diseases. There won't be any lagging when numerous users use this system simultaneously.

Asthha Wahal [7] et.al Their primary research and work was to increase this percentage because the accuracy rate of online results might range from 10 to 50%. As a result, HealthiMe has a lot of promise but has to expand in terms of its aspects and creativity. Compared to other similar apps and technologies, this attempts to provide us with findings that are 60–65% more accurate. By using electronic and telecom advances, the HealthiMe chatbot distributes health-related services. It enables patients who are at a substantial distance from the clinic to receive care, advice, updates, training, checking, and remote confirmations from professionals. It can speed up reaction times, reduce needless time wastage by avoiding repetition and asking questions the answer to which is obvious, and work from anywhere on a constant basis for 24 hours a day.

Shivani Singh [8] et.al This section of the article sought to analyze the available literature on COVID-19-related chatbots in healthcare, identify and characterize these cutting-edge technologies and their potential uses in the fight against COVID-19, and outline associated difficulties. This research discusses a sophisticated AI medical chatbot for users, particularly during nCOV-19 and

other unforeseen pandemics. With intentions to launch this app in the next months, the suggested chatbot in this paper is currently in the design phase. Watson will serve as the AIML platform for this chatbot engine.

Kailash Chandra Bandhu [9] et.al In their proposed fix, a text-based query can be discussed in the chat conversation with the bot. The system offers responses to questions. The user can see the medical professionals who are qualified to handle their specific disease. This program allows many people to find therapy appointments online. Data from chatbots is maintained online as a pattern template. The bot will recommend painkillers and meals that fit the patient's diet based on their condition. A healthcare chatbot application based on Natural Language Processing is used to assess the efficacy of the SGD and ADAM optimizer.

Nikita Vijay Shinde [10] et.al In this study, they introduce a healthcare chatbot that provides users with treatment options for a few common diseases. There are several medical chatbot models available, but they are generally expensive for the average person. With the healthcare chatbot system, they are working to reduce this type of annoyance. This technology actually functions as a sort of symptom checker and can accurately identify the patient's ailment 76% of the time. After the system has been successfully implemented, they will work on adding more features, while employing the proper security.

Soufyane [11] et.al According to the scientific community, chatbots are easy to use by anyone who knows how to text in their native language on the desktop version and in the mobile application. A chatbot offers a personalized analysis based on symptoms, particularly in the field of medical. With the addition of support for more medical features, like the severity,

duration, location, and a more thorough description of symptoms, the identification of symptoms and the performance of diagnosis will be much enhanced in the future.

Nithuna S and Laseena C.A [12] et.al The article's architecture is enhanced by any AIML-based chatbots that are incorporated therein to promote client communication. To make the system more flexible and interactive for practice in many fields, it is operated by a number of small AIML scripts and packages. A framework that associated with the customer using automated conversational operators further completed a number of important duties. According to this strategy, the specialised provides automated conversational agents at a lesser cost than a private conversational agent. In addition to the collegiate context, the chatbot is employed in a number of different sectors, such as forensics, sports, and medical.fast in terms of information acquisition.

Lekha Athota [13] et.al An excellent conversational tool is a chatbot. It lessens the load of the answer by employing an expert system to provide the solution straight to the user. By avoiding the need to consult any specialists or medical professionals when looking for a healthcare solution, the initiative is designed to save the user time. In this instance, the keyword from the user query was extracted using the N-gram and TF-IDF to develop the application.

Aman Ranjan [14] et.al Despite the fact that there are several laws and regulations in place, the gravity of the situation should not be disregarded. Despite maintaining basic precautions, many still avoid going to hospitals, which raises their chance of getting an illness. This practice will help build a healthy and safe workplace without having to worry about the possibility of becoming sick or developing an illness. Also

benefited from this endeavor may be people who are having trouble covering their medical expenses. We intend to add capabilities like translation in the future to our chatbot to make it simpler for folks who don't speak English well to utilize it. Other medical experts may be added to the dataset, and more data, such as the prescription of popular drugs, can be added.

Anuj Patil [15] et.al An analysis of the literature reveals that the conclusions drawn by this approach are accurate. We will do better than in the past due to the amount of the dataset we are employing. We create a mechanism as a result to allow users to ask queries regarding the medication dose in private medical facilities or hospitals. The system calls out and displays all of the names of the medications after receiving output for the drug API.

3. REQUIREMENTS

Python language is used, Google Colab and ThingSpeak - is an IoT analytics platform service that allows us to analyze live data streams in the cloud.

4. EXISTING SYSTEM

The doctors are obliged to continuously ask the patient for specifics because the chatbot did not employ cloud storage, making it impossible for them to check the preceding patient's health record from whom they made their diagnosis. No chat bot has been created for patients to particularly contact doctors and report their problems. The online web application has been created in such a way that the only people with strong reading abilities may use it; neither the elderly nor those with basic literacy skills are permitted to use it. Some researches involved the creation of the website, which allows users to pay for comprehensive prescriptions written by

doctors without having access to their long-term medical records.

5. PROPOSED ALGORITHM

Taking into account the drawbacks of all the aforementioned approaches utilized in earlier systems, we are going to develop a voice chat bot employing Natural Language Processing and cloud storage as an IoT analytics platform service that enables cloud-based live data stream analysis. Here, we're going to concentrate on the patients who have electrolyte imbalance disorders and help them get diagnosed by the doctor via voice chat bot so that the next time they experience the same symptoms, the doctors will be able to see their health records in the cloud and be able to analyze them. For example, if a patient loses potassium, the doctor may create reminders for the patient to consume bananas or apple juice to restore their electrolyte balance. We will alert them by establishing reminders by the doctors using the voice chat bot to remind them to take the prescription. Since electrolyte imbalances are most common in elderly people, we are concentrating on them in order to enable them to use the chat bot independently and without assistance from others. Many people are unaware of the symptoms of electrolyte imbalance, such as fatigue or dizziness, until their health deteriorates to the point where they are at high risk of death.

6. METHODOLOGY

Data Collection – The gathering of data is the initial step. It is the process of gathering and analyzing information from a wide range of sources.

Data Analyzing - In Machine Learning, Data Analysis The initial step in completing the project is data analysis because learning

about the inputs requires extensive data investigation. The data can even be compared during data exploration by using libraries like Matplotlib to display the data through some graphs, such as bar plots.

Data Preprocessing - Data pre-processing is required following investigation. Data pre-processing is an essential stage in machine learning that serves to improve the quality of the data to encourage the extraction of valuable insights from the data.

Creation of Chatbot - The creation of a chatbot that uses pattern comparison to detect the sentence structure and remember response patterns. In this step, we will provide the chatbot with the anticipated patient questions as well as the signs of electrolyte imbalance. The discourse will then be designed after that.

Design the Conversation - Then the next step is reducing the dimension of the dataset due to the large size of the dataset, for that we have used the principal component analysis for dimension reduction. Then two other top clustering algorithms have been used, namely BIRCH and DBSCAN, for defining the patterns of the dataset. Since clustering algorithms helps in large quantity of data Once the work is complete, we may integrate the chat bot with the html webpage just to show the prototype for the future work.

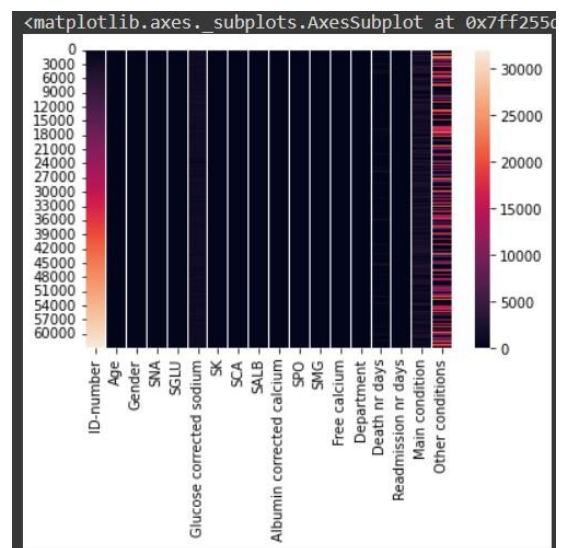
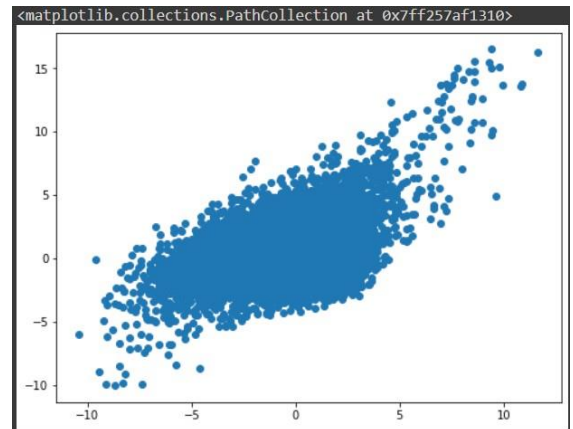
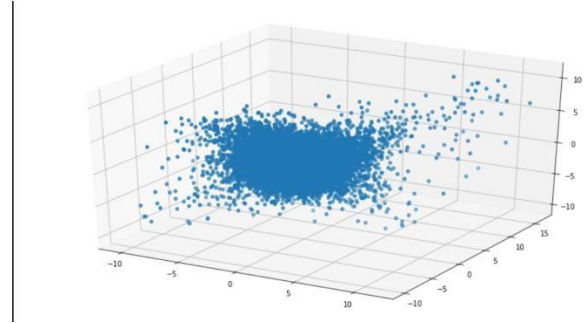
Training and Testing - Predicting what people will say to your chatbot and what they expect from it is the main goal of bot training. You must currently have your use cases specified (specific purposes for your chatbot). After that, we must teach the AI

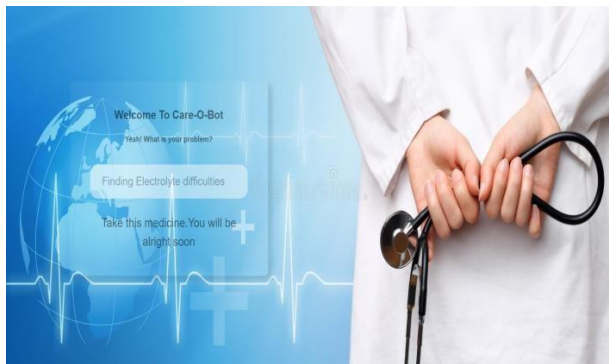
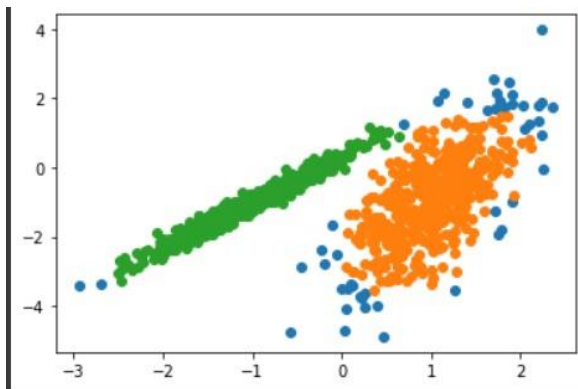
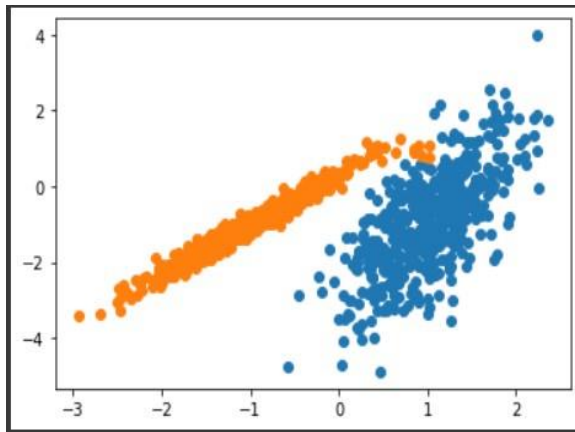
chatbot how to comprehend the various ways that customers will phrase their inquiries. We begin asking the chatbot, the questions we've taught it to respond to once it's ready. As usual, there are not many scenarios that need to be evaluated, allowing us to test manually. So, the AI NLP chatbot's functionality can be checked through testing.

Cloud Storage - The symptoms and diagnosis of the patient are noted, tracked, and stored in the cloud after the chatbot has been developed. This gives the doctor access to the patient's past medical records in the cloud in case the patient recurrently exhibits the same symptoms. As soon as the patient is informed that they have an electrolyte imbalance, they can forget to take their medication. The doctor can inform or remind them via the chat bot.

7. RESULT

Therefore, a trustworthy voice chatbot using Natural Language Processing and cloud storage aids seniors in a simple way to determine electrolyte imbalance. According to our research, a user-friendly assistant is developed using a dataset containing 62991 rows and 18 columns to increase elderly citizens' access to contemporary technology. A few clustering techniques, including as Principal Component Analysis, BIRCH, and DBSCAN algorithms, are also used to analyze it.





1. CONCLUSION

Thus, a trustworthy voice chatbot with Natural Language Processing and cloud storage aids seniors in a simple way to determine electrolyte imbalance. According to our research, a user-friendly assistant is

designed to make current technology more accessible to elders. As a result, a powerful voice chatbot that is easy to use and offers a variety of advantages to seniors without making them reliant on others is developed. Future plans call for integrating our voice chatbot with a web application that uses the same Natural Language Processing and Cloud storage. This would enhance our project and give elderly people more independence and awareness of electrolyte loss. With the aid of this technology, senior citizens have greater accessibility.

8. REFERENCES

- [1] Mathew R B, Varghese S, Joy S E and Alex S Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning Third International Conference on Trends in Electronics and Informatics (2019)
- [2] Kandpal P, Jasnani K, Raut R and Bhorge S Contextual Chatbot for Healthcare Purposes (using Deep Learning) 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4) 625–34 (2020)
- [3] Healthcare Chatbot using Natural Language Processing Papiya Mahajan, Rinku Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 11 | Nov 2020
- [4] HEALTHCARE CHATBOT Athulya N,Jeeshna K,S J Aadithyan,U Sreelakshmi,Hairunizha Alias Nisha Rose Volume 9, Issue 10 October 2021

[5] Manthirathne, Wattaranthenne Pallemulle Walawwe Adithya Sriman Cloud based Muscle cramp prediction through machine learning by sensing EMG Pattern (2022)

[6] Health Care Chatbot Using NLP and Flask Kshitij Thakre, Dr. P. R. Rothe, Sakshi Kukade , Pranali Shinde , Komal Madame International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue V May 2022

[7] IoT based Chatbots using NLP and SVM Algorithms Asthha Wahal, Muskan Aggarwal, Dr. T. Poongodi – 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM) IEEE DOI:10.1109/ICIEM54221.2022.9853095

[8] Design and Development of Conversational Chatbot for Covid-19 using NLP: an AI application Shivani Singh, Manmeet Kaur, Dr. Pooman Tanwar – 2022 6th International Conference on Computing Methodologies and Communication (ICCMC)

[9] Health Care Chatbot using Natural Language Processing with SGD and ADAM Optimizer Parameter Optimization Kailash Chandra Bandhu, Binod Kumar Mishra, Mohit Patel – 2022 IEEE World Conference on Applied Intelligence and Computing

[10] Healthcare Chatbot System using Artificial Intelligence Nikita Vijay Shinde, Aniket Akhade, Pranali Bagad – 2021 Proceedings of the Fifth

International Conference on Trends in Electronics and Informatics (ICOEI). IEEE Xplore Part Number:CFP21J32-ART; ISBN:978-1-6654-1571-2

[11] A Smart Chatbot Architecture based NLP and Machine learning for health care assistance Soufyane Ayanouz, Boudhir Anouar Abdelhakim, Mohammed Benhmed NISS2020, March 31-April 2, 2020, Marrakech, Morocco © 2020 Association for Computing Machinery.

[12] Review on Implementation Techniques of Chatbot Nithuna S and Laseena C.A International Conference on Communication and Signal Processing, July 28 - 30, 2020, India

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

[14] Healthcare Chatbot Aman Ranjan, Sagar Hukre, Sachin Rathore, Reetika Kerkett – 2022 International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538

[15] Healthcare Chatbot using Artificial Intelligence Anuj Patil International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue VIII Aug 2022