Psychological Chatbot

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

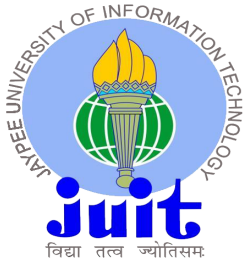
Submitted by

Sai Hritvik (201168)

Tanvi Thakur(201164)

Under the guidance & supervision of

Dr. Ekta Gandotra



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, Solan - 173234 (India)

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ __Department: _____ Enrolment No _____

Contact No. _____E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><td>Copy Received on</td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="3"></td><td colspan="2">Word Counts</td></tr>
<tr><td>Report Generated on</td><td colspan="2">Character Counts</td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td colspan="2">Total Pages Scanned</td></tr>
<tr><td></td><td colspan="2">File Size</td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Candidate's Declaration

We hereby declare that the work presented in this report entitled **Psychological Chatbot** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Ekta Gandotra** (Associate Professor, Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)
Student Name: Sai Hritvik
Roll No.: 201168

(Student Signature with Date)
Student Name: Tanvi Thakur
Roll No.: 201164

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)
Supervisor Name: Dr. Ekta Gandotra
Designation: Associate Professor
Department: Computer Science & Engineering and Information Technology
Dated:

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing to make it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Ekta Gandotra, Assosiate Professor(SG),** Department of CSE Jaypee University of Information Technology, Waknaghat Deep Knowledge & keen interest of my supervisor in the field of "**Machine Learning**" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Ekta Gandotra,** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Sai Hritvik (201168)

Tanvi Thakur (201164)

Table of content

List of abbreviations

Derivations	Meaning
NLP	Natural language processing
NLU	Natural language understanding
RNN	Recurrent neural network
CNN	Convolutional Neural Network
Chatbot	Chatting Robot
AI	Artificial Intelligence

List of tables

List of figures

ABSTRACT

In this era the majority people are suffering from psychological disease and due to this it is affecting the world very badly. In this world people want to discuss their problems with someone whom they can trust. Due to the loneliness increasing in this world the more people are suffering from it. For this problem we have developes a psychological chatbot which will help the people from getting through it and discuss their problem in a secures environment and make them feel good in life so that the problem will decrease from this world and world should be more healthier and happy. Psychological state is an important element in our everyday life in this world. Individuals still don't feel comfortable this usually causes them to talk to a person about their mental condition but normally shut down on themselves. stress accumulation in their minds which then hinder their productivity at work. We end up conceiving that such cases will last for a long time. This would be achieves via developing a therapy chatbot, who could dialogue with the person on how he was feeling emotionally wise. This way, the user is able to express his. The second one is the fact that she feels okay for now without worrying about being judges. Cutting down on death by suicide causes by severe depression.

The chatbot has been traines on a descriptive data set which covers a wide range of psychological topics and situations. We have achieves good accuracy with naive bayes algorithm – 88% thus suggesting the model's ability to learn the query understanding and answers. The web app have also developes to make things easier and to enhance accessibility we are using html, css, and javascript. In our project we uses the two algorithm first one is random forest and the another one is naïve bayes algorithm where the accuracy of the naïve bayes is better than the random forest and then we uses the naïve bayes to train our model further. The accuracy of the model has increased from 79% to 88% where the 79% is of the earlier model which is random forest. We made the web application so that the user will interact with the chatbot easily. The working of chatbot is as follow firstly the chatbot will give the greeting to the user for example it will say hello how can I help you then the user will tell his mood or what he is feeling at that moment and with this the chatbot will analyze the emotions of the user and give the answer accordingly the very severe case of depression will not be treated and those user will get the reply from the chatbot that you should contact nearby psychiatrist as soon as possible at the the chatbot will ask for another helpif user says yes then the conversation will continue and if the user says no it will conclude the chat by saying bye.

CHAPTER 1: INTRODUCTION

1.1) Introduction

Mental health is usually associated with stigma, it is unnoticed, or even overlooked, despite being a vital factor in general welfare. It is unfortunate that intellectual problems like depression, anxiety, and other stresses affect the population extensively. Although therapy or counseling is highly important for resolving such situations that many people suffer from, many humans choose not to seek any help from the mental health specialist.

This has also led to increased number of supplemental treatment together with internet counseling, self-aid books and mental health programmes. The use of bots is among the answers to chat. Chatbots are computer programmes that interact with users via speech or text-based inputs, pretending to be in conversation with real humans. Human beings suffering from intellectual ill health, should receive company, guidance, and direction. The aim of our project is to develop a chat-bot psychiatrist for people suffering from depression. Those unscrupulous of seeking traditional advice or counseling could also employ chatbot. This offers a confidential space where users can express their worries and seek help through a virtual secretary. Being a highly flexible bot enables us to design specific content material as well as shows according to every person's individual tastes.

1.2) Problem Statement

They developed a health chatBot powered by Artificial Intelligence through which they could interact with their customers over mental health matters. There are tools powered by artificial intelligence and natural language processing for psychoeducation, coping, emotional support and even symptom spotting of psychiatric disorders.

Usability and Uses:

The role that a simple psychological-based chatbot plays towards the users' mental health can be seen in various aspects. It can talk of caring and non-judgmental way as it tailors its recommendations for each individual's problem of stress, depression or anxiety. Also, the

chatbots would provide psychoeducation for the users on what the illnesses are as well as the emotions associated with the mental disorders. Some of the modern chat services employ deep sentiment analysis to determine one's emotion based on the language they use.

1.3) Objectives

1) To detect the emotion of the user and responds accordingly.
2) To make an AI-driven chat bot using Natural Language Processing approach for a healthy conversation. This web app will help in managing emotions and releasing stress

1.4) Motivation

1.4.1) Transforming Mental Health Support

In an unstable setting such as that of mental health services, the development of psychological chatbots is driven by a need for changing mental health services. This stimulus results from the recognized requirement for technological interventions aimed at addressing challenges observed in traditional service delivery systems of the mental health field. For instance, some of the prominent factors that contributed to emergence and evolution of psychoboats include the following.

1.4.2) Accessibility and Reach: Breaking Down Barriers

The challenge of traditional mental health services are mainly related to issues pertaining to access. The limitations to accessing such support include long waiting times, geographical barriers and a shortage of mental health professionals or services. Therefore, psychological chatbots aspire at overcoming them through ready-access-24 hrs support. Nowadays, various people use smartphones and have access to the internet. Therefore, different groups in society can interact with chatbots and help populations who might be unable to get mental help services.

1.4.3) Stigma Reduction: Fostering Anonymity and Privacy

Despite this improvement, stigma still discourages people from seeking of mental help. People are afraid of what others would think about them and their decision towards seeking help making many refrain from asking for what they need. Psychological chatbots provide a platform where users can speak their minds openly while still being anonymous and safe against prejudice. Chatbots operate in a virtual space thus de-stigmatising mental health problems hence more people can seek for intervention and share experiences.

1.4.4) Early Intervention: Detecting Signs of Distress

Early identification of mental health issues is important. Nevertheless, people do not always acknowledge or express their emotional discomfort to the extent that it becomes crucial.The ability to identify signs of a person's mental health through continuous interaction with these users and analysis of the provided inputs. Timely diagnosis facilitates early intervention as opposed to handling the cases while they have grown wilder with more complex implications on mental health as a whole.

1.4.5) Scalability: Meeting Growing Demands

Demand for mental health care has increased and most cases are too great for the traditional systems to address individual needs. Scalable psychological chatbots that can handle concurrent conversations address a growing need for more mental health resources. Reaching out to many people makes sure more people get help when they need it.

1.5) Organization

Chapter 1 We have discusses the introduction to our major project, it also describes our objective and motivation of this project.

Chapter 2 Reviewing the literature survey and finding the key gaps in it

Chapter 3 In the system development we have describes that how the project works and explaines the important code.

Chapter 4 We have performes the testing and its method

Chapter 5 Discusses the result and final outcome

Chapter 6 We have discusses the conclusion and future scope of the project

CHAPTER 2 :LITERATURE REVIEW

2.1) Overview of Literature Review

This Prior studies have explored the integration of AI-based tools in mental health care, emphasizing their potential to provide support and address various mental health issues. For instance, the ALIZ-E project, funded by the EU, introduced "Making New "New AI" Friends: Designing a Social Robot for Diabetic Children from an Embodied AI Perspective" to develop Robin, a social robot tailored for type 1 diabetic children prone to depression due to their strict dietary requirements and ongoing psychological stress [1]. In another study, "The Next Generation: Chatbots in Clinical Psychology and Psychotherapy to Foster Mental Health," researchers discuss the significance of chatbots in mental health care, highlighting their role in offering emotional support and encouragement to individuals [7]. Additionally, research on "Using AI chatbots to provide self-help depression interventions for university students" demonstrates the effectiveness of AI chatbots in delivering self-help interventions for depression among university students [8]. Moreover, the study "A review of mobile chatbot apps for anxiety and depression and their self-care features" provides insights into the features and functionalities of mobile chatbot apps designed to alleviate anxiety and depression [9]. Similarly, "Towards developing a pocket therapist: An intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation" emphasizes the importance of developing intelligent chatbots to provide psychological support during challenging times, such as the COVID-19 pandemic [6]. Furthermore, "An Overview of the Features of Chatbots in Mental Health: A Scoping Review" offers a comprehensive overview of the features of chatbots in mental health care, underscoring their potential to enhance patient engagement and support [2]. Additionally, "Chatbot features for anxiety and depression: A scoping review" presents a detailed examination of chatbot features tailored to address anxiety and depression, highlighting their potential benefits [3]. Continuing the exploration of AI-driven solutions in mental health, "Artificial Intelligence Chatbot for Depression: Descriptive Study of Usage" presents a descriptive study on the usage patterns of AI chatbots for depression, shedding light on their potential impact and user engagement [4]. Similarly, "Therapy Chatbot: A Relief From Mental Stress And Problems" discusses the role of therapy chatbots in alleviating mental stress and offering support to individuals facing psychological challenges [5]. 6 Furthermore, "Development of a chatbot for depression: adolescent perceptions and recommendations" delves into adolescent perceptions and recommendations

regarding the development of chatbots for depression, providing valuable insights for improving chatbot interventions in this domain [9]. Additionally, "A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators" demonstrates the application of deep learning techniques in innovative security solutions, which could potentially be adapted for mental health chatbot systems [10].

Table 1: Comparison of Selected Previous work done in the Domain

S.No.	**Paper Title**	**Tools/Techniques/ Dataset**	**Results**	**Limitations**
1.	A review of mobile chatbot apps for anxiety and depression and their self- care features[1]	Reviewed apps for depression and tension that encompass chatbots. Examined apps that had been available on the Apple Store and Google Play Store (Android).	Apps met the inclusion and exclusion criteria These eight apps were included in the review	The study used limited search keywords. App store exclusions, changing app landscape, subjective criteria, and lack of user feedback were limitations.
2.	Chatbot features for anxiety and depression: A scoping review [2]	Data which include chatbot names, goals, dialogues, input formats, and have a look at specifics have been extracted with the aid of reviewers. A type of classes, consisting of	The initial search across six Databases resulted in 1302 citations. After eliminating duplicates and conducting	The study did not report detailed engagement metrics due to its scope. It focused on English articles published between 2015 and 2022.

		schooling, remedy, prognosis, and counseling, were used to classify chatbot purposes.	title/abstract and full-text screenings,37 studies met the inclusion criteria.	
3.	Artificial Intelligence Chatbot for Depression: Descriptive Study of Usage [3]	354 users' interactions with Tess depression modules have been tested on this have a look at. It assessed person engagement, crowning glory prices, and quantity of time spent on each module the use ofdescriptive records.	User sent a total of 6,220 messages which contained 86,298 characters.	There is a significant gap in understanding how chatbots function at both the item and module levels. The design and implementation of chatbots lack guiding models.
4.	Therapy Chatbot: A Relief From Mental Stress And Problems [4]	This therapy chatbot takes four entities: Personal loss Relationship problems Career / Job Problems	The chatbot is able to as it should be become aware of the troubles that users are dealing with and offer pointers on a way to enhance their basic properly being. It is	The text's limited geographic scope may hinder a comprehensive understanding of cultural and regional factors influencing mental health and chatbot usage. Insufficient detail about study

			particularly powerful in addressing issues relating courting issues, profession/task difficulties, instructional concerns,and personal loss.	methodologies, sample sizes, and findings reduces the ability to evaluate the rigor and relevance of the research.
5.	Towards developing a pocket therapist: an intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation [5]	Draws attention to how crucial customized conversational agent design is. Affords a chatbot therapist designed to provide intellectual fitness help to people affected by the COVID-19	Collected diverse mental health data from Reddit. Trained an empathetic chatbot using a sequence- to sequence reinforcement model.	Chatbot had issues with context understanding andunanswered questions. New healthcare chatbots like OneRemission and Babylon Health use advanced machine learning techniques.
6.	The Next Generation: Chatbots in Clinical Psychology and Psychotherapy to Foster Mental Health – A Scoping Review [6]	Review of the literature on Chatbots in psychotherapy and medicalpsychology. Covered research at the advertising of mental fitness through chatbots.	Chatbots in mental health are experimental. Lack strong evidence from controlled trials. Show promise for practicality and feasibility.	Lack of specific details on ethical, legal, and data privacy concerns. Need for clear quality criteria for chatbots

7.	Using AI chatbots to provide self help depression interventions for university students: A randomized trial of effectiveness[7]	Three Chinese universities had been used to recruit participants. 37 is the desired pattern length for every institution. PHQ-nine is used to assess despair discount with ethical approval.	Recruitment between February 11th and April 15th , 2021. 83 eligible university student participants in a 16-week trial	Limited chatbot content due to professional approval constraints. Potential incorporation of DST and DPO modules for improved personalization.
8.	Development of a chatbot for depression: adolescent perceptions and recommendations[8]	Focused American contributors between the ages of 13 and 18. Used Facebook Messenger's Beth Bot, a rule-based chatbot.	Targeted participants aged 13 to 18 in the United States. Utilized a rulebased chatbot named Beth Bot on Facebook Messenger.	Limited by a small sample size primarily composed of California adolescents. Potential biases introduced by attrition during the study.
9.	An overview of the features of chatbots in mental health: A scoping review[9]	Used a radical technique that made use of seven databases and referencing techniques. Covered independent reviewers inside	Reviewed a total of 53 studies from an initial pool of 1039 citations. Examined 41 different chatbots predominantly	Focused on reviewing chatbots specifically designed for standalone software and web browser platforms,

		the manner of selecting the studies and extracting the statistics.	used for therapy, training, and screening.	excluding those controlled by human operators. Acknowledged the possibility of missing some studies due to practical constraints in the search process.

2.2) Key Gaps In The Literature

2.2.1) The study used limited search keywords:

A small collection of phrases limited the examine's seek approach, which would possibly have affected how thorough the literature evaluation turned into. Which articles are covered in the look at and the way large the quest may be are each encouraged with the aid of the key phrases used. Employing a greater diversity of key phrases could have yielded a extra various collection of pertinent articles and a extra thorough comprehension of the subject.

2.2.2) App store exclusions, changing app landscape, subjective criteria, and lack of user feedback were limitations:

The observe findings' robustness was impacted with the aid of some shortcomings. Potentially essential apps may had been left out because of app save exclusions, specifically ones that are not handy on famous platforms. The dynamic nature of cell packages makes it tough to keep the have a look at findings applicable ultimately. The observe generalizability can be impacted by using bias added by means of subjective app selection criteria. Furthermore, the lack of person enter makes it more hard for the have a look at to assess person options and reports, which are vital additives in figuring out how a success mental health apps are.

2.2.3) The study did not report detailed engagement metrics due to its scope:

Although the take a look at yielded insightful records, one downside is the dearth of precise engagement indicators. More precise records, which include user retention quotes, app utilization frequency, and interplay time, could have given upward push to a more complex photo of ways users interact with mental fitness apps. Reporting these signs, but, could necessitate a extra involved and useful resource-in depth records collection process, which might be outside the purview of the research.

2.2.4) It focused on English articles published between 2015 and 2022:

The chronological and linguistic barriers of the have a look at constrain how a ways the consequences may be applied. Restricting the focal point to English-language articles posted between 2015 and 2022 may additionally result in the exclusion of pertinent research posted earlier or in other languages. Although the problem of intellectual fitness programs is developing quick, in advance research can also provide historical background or insights into how these technologies have evolved.

2.2.5) This focus may have resulted in the omission of relevant interdisciplinary database results, such as those from Web of Science:

Due to the observe 's dependence on a slim range of databases—which does now not consist of multidisciplinary resources like Web of Science—crucial contributions from associated domains may fit omitted. Multidisciplinary databases frequently embody a greater variety of studies throughout several fields, enhancing the multiplicity of viewpoints and conclusions. The take a look at 's capability to offer a comprehensive information of the topic may be hampered by way of the exclusion of sure sources.

CHAPTER 3 SYSTEM DEVELOPMENT

3.1 Requirements and Analysis

3.1.1) Processor (CPU):

Requirement: Efficient parallel processing with a Multi-core processor.

Analysis: The brains of the systems are the processors. For instance, an Intel Core i5 or any good multi-core processor will enable the chatbot to conduct a few tasks at the same time.

3.1.2) Memory (RAM):

Requirement: Adequate memory for simultaneous user sessions and big sets of data. Analysis: Chatbots use RAM to store data which they can access fast for operations such as conversations with users. The system should include minimally 16 GB of RAM so that it manages the complexity of natural language processing effortlessly even during peaks.

3.1.3) Storage (SSD):

Requirement: Rapid memory access for fast model loading/data retrieval.

Analysis: Solid state drive(SSD) can fetch data much quickly than hard disk drives. Responsiveness demands prompt retrieval of information from the storage while loading the machine learning models as well as accessing the user data during the chats.

3.1.4) Platform Used:

Platform: Google Collab

Analysis: The cloud environment of the Google Collab can host GPU for the running machine learning models and neural network training. Although the primary hardware specifications are given by Google Collab, the efficiency of code execution is dependent on your local machine's specification like i5 10th Gen processor and 16GB RAM.

3.1.5) System Architecture:

Requirement: 64-bit architecture for dealing with big datasets and memory needs.

Analysis: Sixty four bit hardware address capacity enables the system cope with large data sets and complex ML models that are typical features of NLP tasks.

3.1.6) Operating System:

Requirement: This project uses macOS.

3.2 Project Design and Architecture

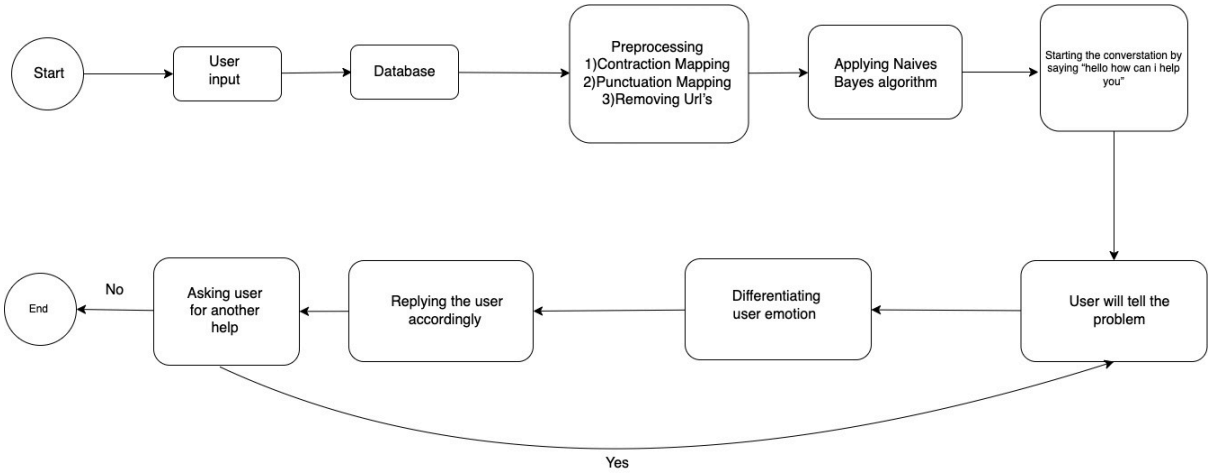


Figure 1: Flow Diagram of Chatbot

Figure 1 depicts that this flow chart it is shown that this flowchart outlines a conversation with a psychological chatbot, a virtual companion offering mental health support. The process starts with you, the user, expressing yourself through text. The chatbot then preprocesses your input. This preprocessing involves three steps: Abbreviation Mapping: Similar to websites, the chatbot might expand shortened words you use. Punctuation Mapping: Just like a website, the chatbot might adjust your punctuation for clarity. Identifying Emotional Language: This differs from website interaction. Here, the chatbot might identify and categorize any words or phrases that suggest your emotional state. After preprocessing, the chatbot employs a Naive Bayes algorithm, a classification tool, to understand the intent behind your message. Perhaps you're expressing feelings of anxiety, seeking coping mechanisms for stress, or simply wanting to vent. Based on this classification, the chatbot initiates a conversation tailored to your needs. For example, if you express anxiety, the chatbot might begin with an empathetic response like "It sounds like you're feeling anxious. Would you like to talk about what's on your mind?" The chatbot then replies to you, with the content depending on your input and its classification. The chatbot might offer resources, suggest relaxation techniques, or simply provide a listening ear. Here, the flowchart suggests the chatbot can potentially differentiate your emotional state further, perhaps by analyzing the progression of your words and phrasing. Finally, the chatbot asks if you have anything else you'd like to discuss. If you do, the cycle restarts. If not, the conversation concludes.

3.3)Methodology

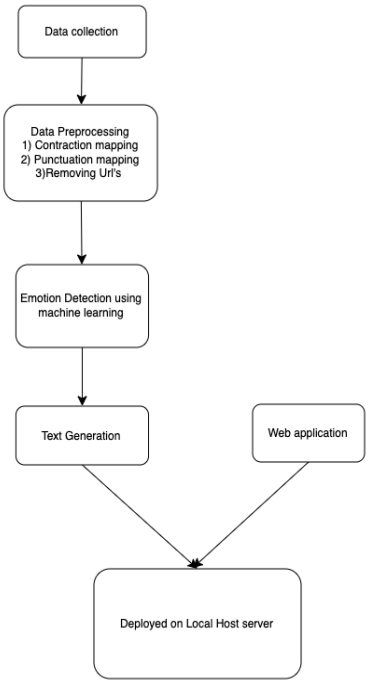


Figure 2: Work flow diagram

Figure 2 depicts that firstly we have collected the dataset and then we have prepared the dataset by preprocessing it in this preprocessing we have used contraction mapping punctuation mapping and removing the url's and we have worked on the emotion detection using two machine learning algorithm first is naïve bayes and second is random forest and then we have worked on the text generation and lastly we made a web application for the chatbot and deployed in the local host server.

3.4) Data Collection

Firstly we have collected the data from the Kaggle the dataset and in this dataset comprises a predefined set of questions derived from various psychological domains, designed to elicit insightful responses from patients. Alongside these questions are the corresponding answers from experienced psychologists, offering valuable insights into appropriate responses and therapeutic approaches. In addition to sourcing data from Kaggle, we've also collected extensive datasets in the form of text files, categorized based on distinct emotional states such as sadness, anger, joy, and fear. Each text file encapsulates a rich array of expressions, experiences, and narratives corresponding to the respective emotion. These datasets serve as invaluable resources for training the psychological chatbot to recognize, understand, and appropriately respond to a diverse range of emotional cues expressed by users. At last after collecting the dataset we combined all of these into one text file.

Figure 3: Descriptive Dataset

3.5) Data Preparation

After colleting the data from the different sources we have combined all of these in one text file and then we have started the procedure of data pre processing in the data pre processing we have performed the different steps like contraction mapping, punctuation mapping, removing url's

3.5.1) Data pre processing :

3.5.1.1) Performed Contraction mapping

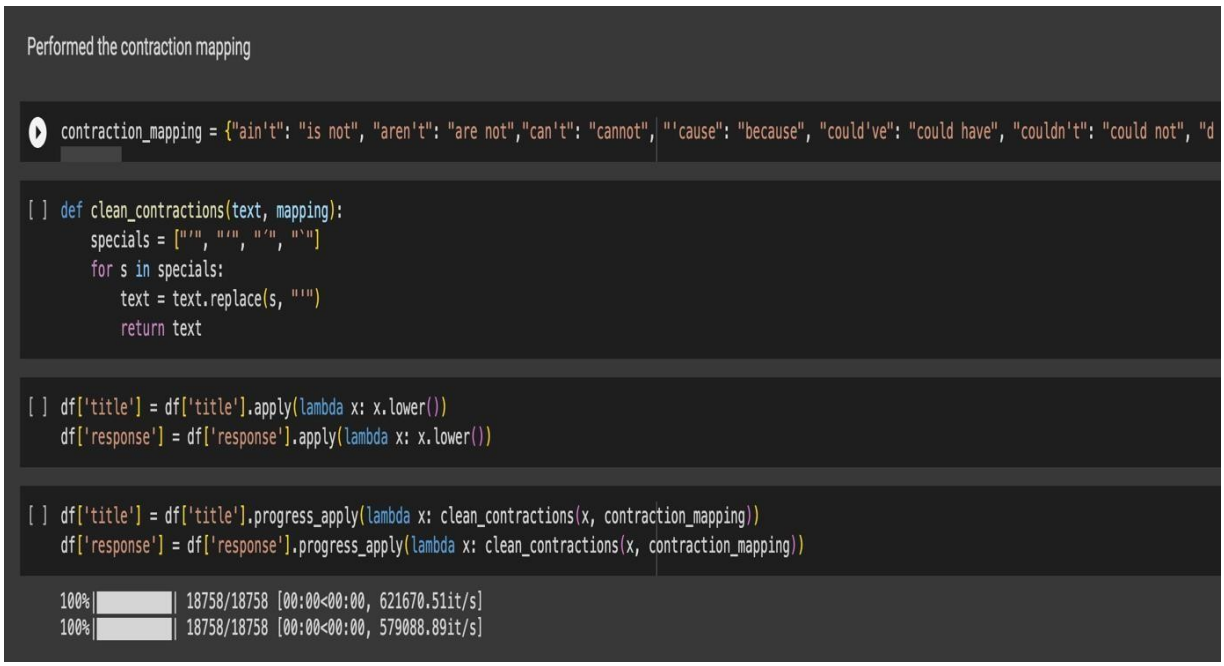


Figure 4: Contraction mapping

Figure 4 depicts that contraction mapping is the process where the contracted forms are converted to the full forms for example where "ain't" is translated as "is not', "don't" becomes "do not", and so on. These words are useful in mapping contractions to the text data while performing the natural language processing tasks for the same purpose. Therefore, the short forms are converted to the full form.

3.5.1.2) Performed Punctuation mapping:

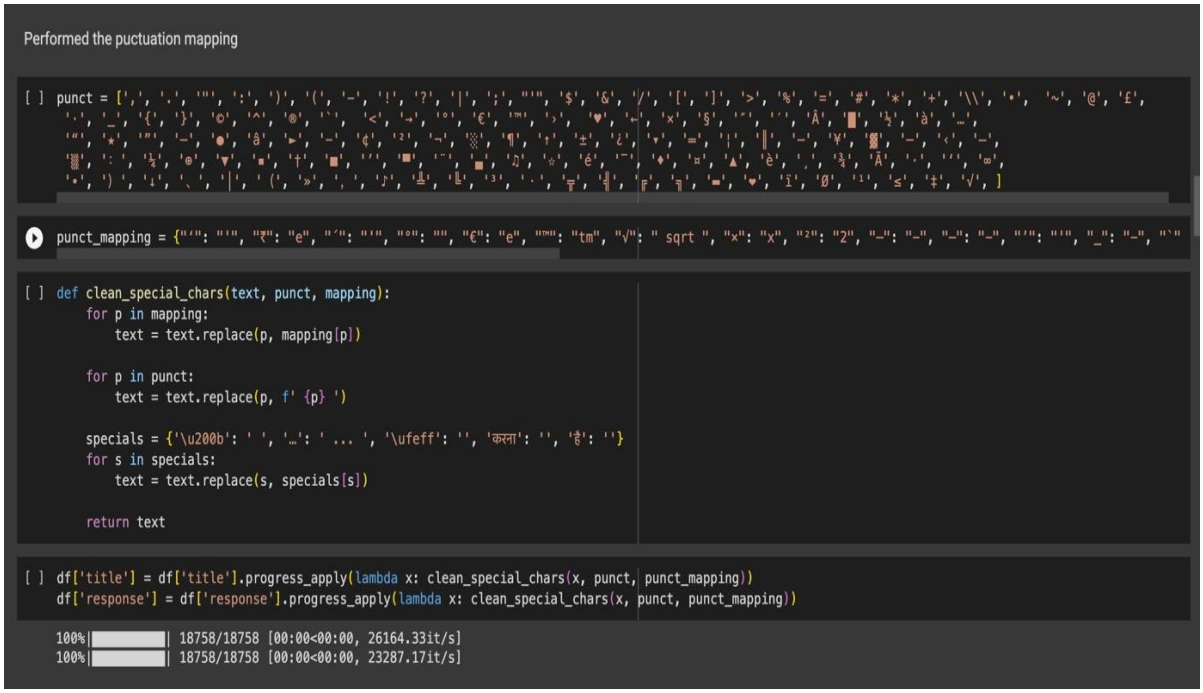


Figure 5: Punctuation mapping

Figure 5 depicts that this Python dictionary named as ‘punct_mapping’ serves as a replacement tool for converting certain punctuation and other characters in textural information to their respective substitutes. Each dictionary’s key cost pair links a specific person with the corresponding equivalent. Some instances of special varieties of single and dual quote markers such as “’”, “’”, “”,” and “” will be changed into ordinary “’”. This dictionary aims at pre-processing and cleaning text by substituting additional common characters in place of perhaps harder or uncommon ones. Such may be very helpful especially with respect to herb-based jobs where perfect analysis and explanation require uniform illustration of text. There are also words with letters like “√” changed into “sqrt” and “θ” to “theta”. This proves that there is no single substitution for each punction mark.

3.5.1.3) Removing Url's:

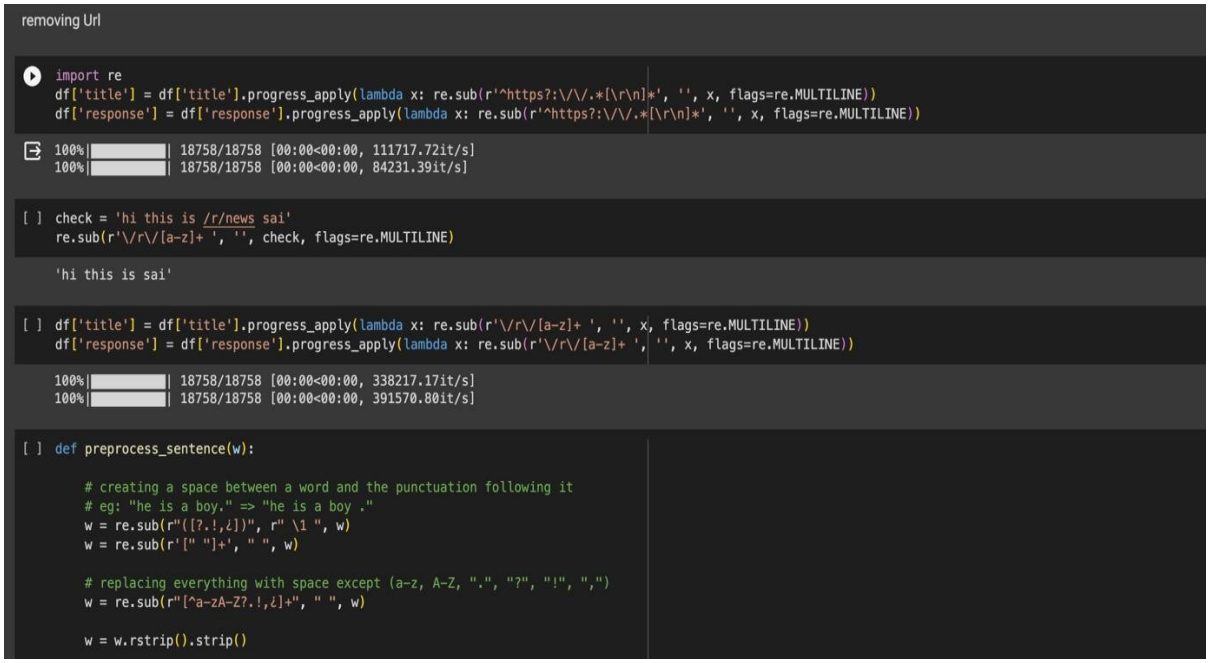


Figure 6: Removing URL's

Figure 6 depicts that A small chunk of code that pulls out URLs from the user response.To discover and replace URLs, use the ordinary expression `r'^https?: //.*[rn]*'. Starting of a line, wherein "http" or "https" is observed by ": Any character following "/" and any newline characters, even carriage returns. The second step is then to use the `progress _ apply approach to apply the 're'. To sub function to "identify" and "response" columns respectively that replace the noted URLs with blank. This cleansing operation aims to improve the clarity and the targeting of the examination as it is a pre-processing stage in many natural language processing works. After performing the removing Url's the Url's given by the user will be replaced bye the blank space.

3.5.1.4) Final pre processed dataset:

```
[ ] def preprocess(x, y):

        preprocess_x, x_tk = tokenize(x)
        preprocess_y, y_tk = tokenize(y)

        preprocess_x = pad(preprocess_x)
        preprocess_y = pad(preprocess_y)

        # Keras's sparse_categorical_crossentropy function requires the labels to be in 3 dimensions
        preprocess_y = preprocess_y.reshape(*preprocess_y.shape, 1)

        return preprocess_x, preprocess_y, x_tk, y_tk
```

```
preproc_title, preproc_response, title_tokenizer, response_tokenizer =\
    preprocess(title, response)

max_title_sequence_length = preproc_title.shape[1]
max_response_sequence_length = preproc_response.shape[1]
title_vocab_size = len(title_tokenizer.word_index)
response_vocab_size = len(response_tokenizer.word_index)

print('Data Preprocessed')
print("Max Title sentence length:", max_title_sequence_length)
print("Max Response sentence length:", max_response_sequence_length)
print("Title vocabulary size:", title_vocab_size)
print("Response vocabulary size:", response_vocab_size)
```

```
Data Preprocessed
Max Title sentence length: 1
Max Response sentence length: 1
Title vocabulary size: 9
Response vocabulary size: 12
```

Figure 7: Final Pre processed

Figure 7 depicts that it is a final preprocess dataset in which the following actions are done first is contraction mapping second is punctuation mapping third is removing URL's after doing this it's state that the dataset is fully pre-processed and there is no same questions in the dataset. The contraction mapping is the mapping which associates the special character by the simple character's which are easily read by the users then we perform the punctuation mapping in the punctuation mapping we associate the punctuations with the alphabets. In this all the rows and columns are pre-processed and finally it got a new dataset which will be used for the training of the Chabot.

3.6) Emotion detection using Machine Learning

3.6.1) Applying Random Forest Algorithm

After pre processing of the dataset we are applying random forest algorithm to train our model for emotion detection and we have categorised emotion into 4 parts sad, joy, fear, anger

```
import matplotlib.pyplot as plt
from nltk.stem.lancaster import LancasterStemmer
from string import punctuation
import nltk
nltk.download('punkt')
anger_training_set = []
fear_training_set = []
sadness_training_set = []
joy_training_set = []

anger_test_set = []
fear_test_set = []
sadness_test_set = []
joy_test_set = []
stemmer = LancasterStemmer()
all_words=[]

# Here I am loading the dataset from stored folder. The training data is stored as text file and each tweet is accompanied
# by the magnitude of its sentiment (0 to 1). I had to go through the tweets myself and observed that a threshold of 0.5 is
# good enough to classify a tweet according to its sentiment. Tweets with lesser threshold were not definitive to be trained as per their mentioned classif
# I only read those tweets that have a dominant classification factor i.e. above 0.5
# Here i am setting each tweet's threshold magnitude accordingly
def load_training_data(sentiment):
    data = open("/content/anger_training_set.txt")
    if sentiment == "anger":
        threshold = 0.5
    elif sentiment == "fear":
        threshold = 0.6
    elif sentiment == "sadness":
        threshold = 0.5
    elif sentiment == "joy":
        threshold = 0.5
    else:
        pass
```

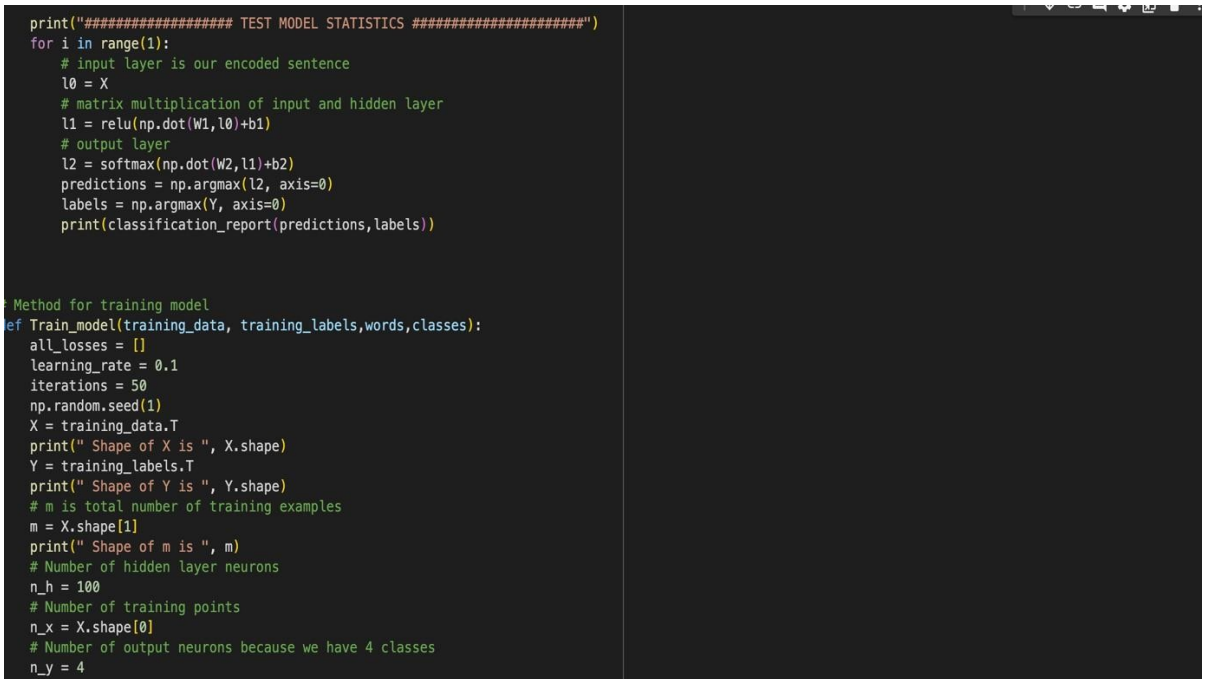
Figure 8: Training chatbot to emotions

Figure 8 depicts that a tiny fraction of the larger problem of sentiment analysis with respect to NLP of herbal language. The goal here is to place within feeling categories like anger, sadness, joy, and diss appointment each tweet which does correspond. Two important libraries are used in the implementation: matplotlib is employed to produce charts, while NLTK is utilized for dealing with natural language affairs. In the initialization phase for both training and test, there is an attachment of awesome lists for each sentiment category. It includes an implementation of a Lancaster Stemmer from NLTK that reduces phrases to their base/root form in order to build a system called phrase stemming.

```
ef main():
   bag = []
   all_data = []
   all_test_data = []
   labels = []
   classes = []
   labels = []
   test_labels = []
   words=[]
   test_words = []

   ######## Here we read the whole training data for each class and the threshold we will use for its classification
   anger_training_data,threshold = load_training_data("anger")
   anger_training_set = clean_data(anger_training_data,threshold)
   print(anger_training_set[0])

   fear_training_data,threshold = load_training_data("fear")
   fear_training_set = clean_data(fear_training_data,threshold)

   sadness_training_data,threshold = load_training_data("sadness")
   sadness_training_set = clean_data(sadness_training_data,threshold)

   joy_training_data,threshold = load_training_data("joy")
   joy_training_set = clean_data(joy_training_data,threshold)


   ######### Here we read the whole test data for each class and the threshold we will use for its classification
   anger_test_data = load_test_data("anger")
   anger_test_set = clean_data(anger_test_data,threshold)
   #print(anger_test_set[0])
   print(len(anger_test_set))
```

Figure 9: Training data about emotion

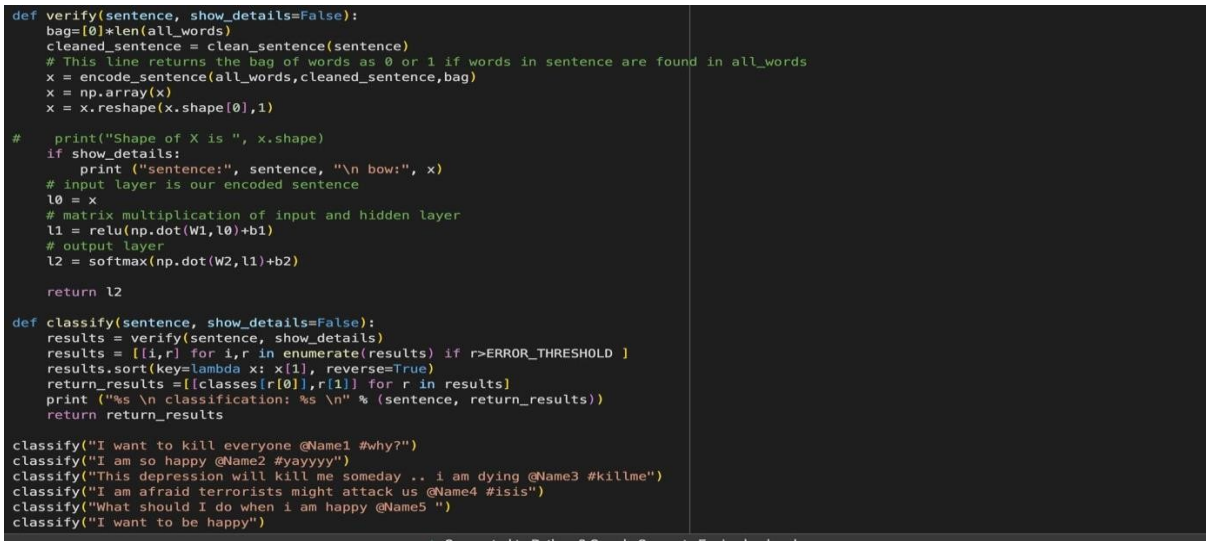
Figure 9 depicts that special sentiment class-specific statistical data to be stored for use by the `important` function is initialized with some lists and variables. bag, all_data, all_test_data, labels, instructions, and phrases. There might also be redundant declarations of test labels as well as labels. This component should make it easier to load and train information in different feelings such as anger, fear, sadness, and joy. With the help of `load_training_data`, one is able to retrieve each sentient magnitides's training data together with its corresponed classification threshold. Clean data attribute helps in cleaning of final training information. In fact, upon studying what is contained in the published material for the trained data set, one observes that the attribute scans were for "angry" appearing first in the cleansed training dataset (anger_training_set). Eventually, this is important in generating educational fact



```
print("################## TEST MODEL STATISTICS ####################")
for i in range(1):
    # input layer is our encoded sentence
    l0 = X
    # matrix multiplication of input and hidden layer
    l1 = relu(np.dot(W1,l0)+b1)
    # output layer
    l2 = softmax(np.dot(W2,l1)+b2)
    predictions = np.argmax(l2, axis=0)
    labels = np.argmax(Y, axis=0)
    print(classification_report(predictions,labels))



# Method for training model
def Train_model(training_data, training_labels,words,classes):
    all_losses = []
    learning_rate = 0.1
    iterations = 50
    np.random.seed(1)
    X = training_data.T
    print(" Shape of X is ", X.shape)
    Y = training_labels.T
    print(" Shape of Y is ", Y.shape)
    # m is total number of training examples
    m = X.shape[1]
    print(" Shape of m is ", m)
    # Number of hidden layer neurons
    n_h = 100
    # Number of training points
    n_x = X.shape[0]
    # Number of output neurons because we have 4 classes
    n_y = 4
```

Figure 10: Testing Model Accuracy

Figure 10 depicts that a simplified neural network model follows as an educational illustration and framework for evaluating emotion oriented information in the source code enclosed above. The print-statement at the beginning represents the objective of this phase that is computing respective records to assess the model's ability. In one stride, there is one variety that helps in forward skipping of the neural network. An input layer (l0) feeds into a hidden layer (bl) via ReLU = MAX(0, x) units; these units are activated by multiplying weights (Wl) with input layer values. Softmax activation feature involves multiplication of the weights of W2 with the concealed layer values plus bias b2. Subsequently, the performance of a model is created through the `scikit-learn classifications report`. Remember, this also implies F1-rating , precisions and recalls. This implies that the way version of sentiment detection must be compared using this crucial measure. The next part referred to as 'Train_model' explains the training process. the parameters of hyperit information, the number of the hidden-layer-neuron, iterations, and the learning rate.



```
def verify(sentence, show_details=False):
    bag=[0]*len(all_words)
    cleaned_sentence = clean_sentence(sentence)
    # This line returns the bag of words as 0 or 1 if words in sentence are found in all_words
    x = encode_sentence(all_words,cleaned_sentence,bag)
    x = np.array(x)
    x = x.reshape(x.shape[0],1)

#    print("Shape of X is ", x.shape)
    if show_details:
        print ("sentence:", sentence, "\n bow:", x)
    # input layer is our encoded sentence
    l0 = x
    # matrix multiplication of input and hidden layer
    l1 = relu(np.dot(W1,l0)+b1)
    # output layer
    l2 = softmax(np.dot(W2,l1)+b2)

    return l2

def classify(sentence, show_details=False):
    results = verify(sentence, show_details)
    results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD ]
    results.sort(key=lambda x: x[1], reverse=True)
    return_results =[[classes[r[0]],r[1]] for r in results]
    print ("%s \n classification: %s \n" % (sentence, return_results))
    return return_results

classify("I want to kill everyone @Name1 #why?")
classify("I am so happy @Name2 #yayyyy")
classify("This depression will kill me someday .. i am dying @Name3 #killme")
classify("I am afraid terrorists might attack us @Name4 #isis")
classify("What should I do when i am happy @Name5 ")
classify("I want to be happy")
```

Figure 11: Classifying the terms

Figure 11 depicts the look of it is as if the given code fragment loads and employs a sentiment assessment model for classifying sentences under several opinion classes. In this case, weights.json loads the weights and biases for versions. This “clean_sentence” attribute cleans numbers and punctuations, converts all words into lowercase and deletes first letter of “@”starting twitter tags. Second, it processes the cleansed term as a bag of words running the forward pass through neural networks before providing output-level activations for sentiments evaluation by aff. It then specifies the error threshold (ERROR_THRESHOLD) and clears away information whose statistical evidence falls below To this end, we present some pattern statements that can be used to illustrate the outcome of this sentimental analysis on some instances. The classify function is fed with sentences which reflect a particular emotion such as fear, joy, sad, anger, and so forth. The output comprises the input sentence coupled with respective sentiment classes together with probability of each class included in this set. In this context, code structure refers to the arrangement of related words based on their sentiments by a supervised model of sentiment analysis.

3.6.2) Applying Naïve Bayes Algorithm

After applying the random forest algorithm we are using naïve bayes algorithm for the emotion detection.

```
def main():
    all_train_data = []
    all_test_data = []
    train_labels = []
    test_labels = []

    for sentiment in ["anger", "fear", "sadness", "joy"]:
        training_data, threshold = load_training_data(sentiment)
        training_set = clean_data(training_data, threshold)
        all_train_data.extend(training_set)

        test_data = load_test_data(sentiment)
        test_set = clean_data(test_data, threshold)
        all_test_data.extend(test_set)

    train_documents, train_labels = zip(*all_train_data)
    test_documents, test_labels = zip(*all_test_data)

    vectorizer = CountVectorizer()
    X_train = vectorizer.fit_transform(train_documents)
    X_test = vectorizer.transform(test_documents)

    classifier = MultinomialNB()
    classifier.fit(X_train, train_labels)
    predictions = classifier.predict(X_test)

    print(classification_report(test_labels, predictions))
```

Figure 12: Appling the Naive bayes classifier

Figure 12 depicts that the script designed in python was explained in the textbook. It was Naive Bayes classifier text mining approach in Python. It begins by initializing empty lists to store training and testing data alongside their corresponding labels, iterating over four sentiment categories: "anger," "fear," "sadness," and "joy" were sentiments indicated. Different Paraphrasing tools will go through the training data and the test data, and will mine them by a certain type of mining (tokenization and stopwords removal). Forming these datasets by scraping from preprocessed text, both text data and labels are extracted. The word count of texts is changed into document - term matrix with the help of the CountVectorizer class which is a language stopped value representer for how many words there are in text corpora per the document. The classifier which uses multinomial naïve Bayes algorithm is formulated to recognize whether the words frequencies are associated with positive or negative sentiment labels. When the training is done, then what the classifier can do is to guess the sentiment of data from the testing dataset.

```
    # Extract metrics
    precision = [report[label]['precision'] * 100 for label in classifier.classes_]
    recall = [report[label]['recall'] * 100 for label in classifier.classes_]
    f1_score = [report[label]['f1-score'] * 100 for label in classifier.classes_]
    classes = classifier.classes_

    # Plotting
    x = np.arange(len(classes))  # the label locations
    width = 0.2  # the width of the bars

    fig, ax = plt.subplots()
    rects1 = ax.bar(x - width, precision, width, label='Precision')
    rects2 = ax.bar(x, recall, width, label='Recall')
    rects3 = ax.bar(x + width, f1_score, width, label='F1-Score')

    # Add some text for labels, title and custom x-axis tick labels, etc.
    ax.set_ylabel('Scores (%)')
    ax.set_title('Scores by class')
    ax.set_xticks(x)
    ax.set_xticklabels(classes)
    ax.legend()

    fig.tight_layout()

    plt.show()

if __name__ == '__main__':
    main()
```

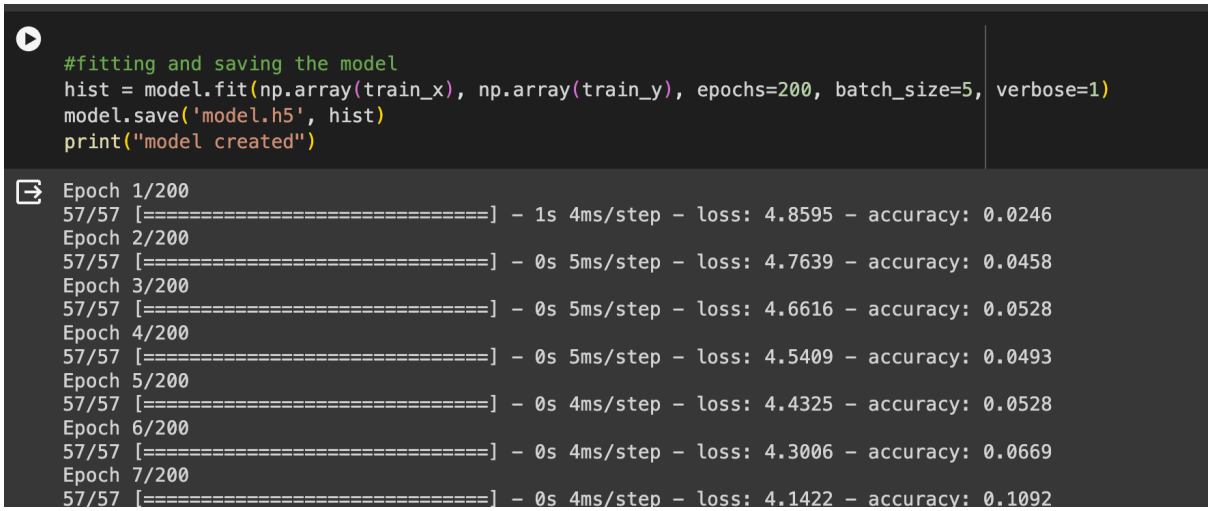
Figure 13: Representing the Metrics of Naïve bayes algorithm

Figure 13 depicts that algorithm produces a graphical representation of classification metrics in case of sentiment analysis evaluation with a multinomial Naive Bayes classifier. The code then runs a classification report which includes metrics like precision, recall and F1-score for each sentiment class and subsequently portrays these metrics and the corresponding sentiment labels. Finally, it draws these measures as bar charts where the columns depict a specific sentiment group. Within each column, there are three types of bars which show precision, recall, and F1-score separately. Axes are consisting of the sentiment classes on the x-axis and the scores in percentages on the y-axis. Through visualization the code provides a clear overlay of the performance of classifier over different categories of the sentiment into which it was classified, a critical interpretational and evaluation tool for performance evaluations. Next is the execution of sentiment analysis and visualization by the main() function.

```
# Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer co
# equal to number of intents to predict output intent with softmax
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))
```

Figure 14: Creating the model using Naive Bayes Algorithm

Figure 14 depicts that the contextualized Python code comprises a neural network model built on the Keras Sequential API for the sake of intent classification which is leveraged often in natural language processing tasks like chatbots or virtual assistants. To start with, the model architecture has three densely connected layers. A group of 128 neurons is used for the first layer, each neuron being a computing unit along with learning patterns in input targets. The activation function employed here is ReLU (Rectified Linear Unit), that is responsible for disabling the linearity displayed in the model, so that it can learn complex interdependencies, present in the data. This layer's input shape is defined by the number of dimensions of the input features (train_x), standing for the flattened representation of the text data. In the second layer, turn the dropout layer on with 0. 2 dropout rate. 5, in which 50% of neurons are randomly dropped off during the training procedure to prevent the network from overfitting and thus lead to the generalization ability of the model being improved. The layer-2 containing 64 neurons with ReLU activation is an abstraction from the input data with the help of another transformation and extraction. In addition to the first layer, the second layer is equipped with another dropout layer of the same probability rate meant for regularization. The final layer , which is tolerant, contains the number of neurons equal to the number of intents which the model is targeted to predict. A softmax activation function is introduced in this layer to compute respective probabilities of intents, making sure that the output probabilities add up to 1. 0 across all intents. This way, it can act like a classifier that is capable of generating a probability distribution from which the most probable intent can be predicted. In summary, this neural network architecture offers a powerful implementation for the task of intent learning and text classification from the input text data. This makes the solution appropriate for various problems of natural language processing.



```
#fitting and saving the model
hist = model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5, verbose=1)
model.save('model.h5', hist)
print("model created")
```

```
Epoch 1/200
57/57 [==============================] - 1s 4ms/step - loss: 4.8595 - accuracy: 0.0246
Epoch 2/200
57/57 [==============================] - 0s 5ms/step - loss: 4.7639 - accuracy: 0.0458
Epoch 3/200
57/57 [==============================] - 0s 5ms/step - loss: 4.6616 - accuracy: 0.0528
Epoch 4/200
57/57 [==============================] - 0s 5ms/step - loss: 4.5409 - accuracy: 0.0493
Epoch 5/200
57/57 [==============================] - 0s 4ms/step - loss: 4.4325 - accuracy: 0.0528
Epoch 6/200
57/57 [==============================] - 0s 4ms/step - loss: 4.3006 - accuracy: 0.0669
Epoch 7/200
57/57 [==============================] - 0s 4ms/step - loss: 4.1422 - accuracy: 0.1092
```

Figure 15: Epoch from dataset and then creating the model

Figure 15 depicts that the python code implementes neural network model training for intent classification uisimg training data(train_x and train_y) withe 200 periods and batch size of five. When training the model, the metrics such as loss and accuracy per epoch are visible enhancing our understanding of how well the algorithm is detecting the trends from the data. When defined, the loss function pinpoints the difference between the actual and predicted results. It aims at minimizing this gap during the training process by making the correct predictions over time. Prediction correctness is the number of samples or instances in train data set among chosen ones that were correctly classified. The model's training under multiple epochs leads to the decrease in the loss value while accuracy is growing, which consequently demonstrates that the model is getting more likely to accurately predict intents. First of all, the model is provisioned after the training and then saved as a. h5 data file for future use is mentioned. Finally the transposed outcome informs about the model`s training dynamics on the specified number of epochs, supplying useful information on its performance and convergence.

3.7) Text generation for the user

After applying the two models we got the good results of the naïve bayes algorithm and hence we are using this algorithm for the text generation and in the text generation the user will interact with the chatbot.

```
    p = np.array(bag)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i, r] for i, r in enumerate(res) if r > ERROR_THRESHOLD]
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})

    # Get response based on predicted intent
    tag = return_list[0]['intent']
    list_of_intents = intents['intents']
    for i in list_of_intents:
        if i['tag'] == tag:
            result = random.choice(i['responses'])
            break
    return result

# Main function to interact with the chatbot
def main():
    print("Welcome to the chatbot!")
    while True:
        user_input = input("You: ")
        if user_input.lower() == 'bye':
            print("Chatbot: Goodbye have a nice day")
            break
        else:
            response = chatbot_response(user_input)
            print("Chatbot:", response)
```

Figure 16: Interaction with the user

Figure 16 depicts that python code is creating basis for an interactive chatbot, specialised with initiating and maintaining conversation with its users upon provided text input. The design of the model is based on neural network which is responsible for determining the intentions of users that seek information and it also generating the detailed responses accordingly. NLTK facilitating tokenization and lemmatization in processing text provides an intervention that ensures input submitted by users for analysis is properly formatted. The chatbot_response function orchestrates the core functionality: it parses user sentences into a vocabulary or word count form via a bag-of-words modeling approach, the number is then presented as the neural network's input for intent prediction, and one of the intent returns is chosen based on the most possible intend rank. The error threshold used has bots predict only confident analysis instances, thus bolstering the system reliability. Principally, the function does the part of talking, which means asking for information from a user and displaying the bot's replies.

3.8) Making the user interface

At last after using the two algorithm the naïve bayes has a good results so we have used the naïve bayes algorithm and then we performed the text generation at last we made a web application and connected our model named ‘model.h5’ with a web application using flask at the local host server

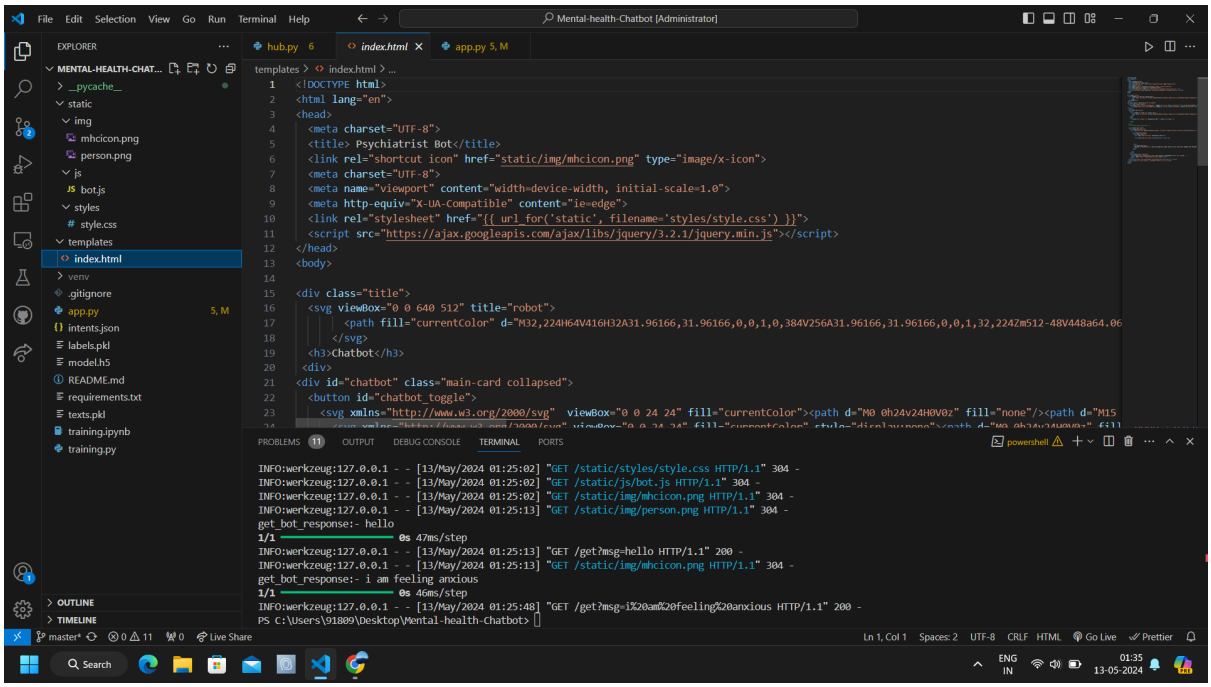


Figure 17: Using html for web app

Figure 17 depicts that code provides a structurial basis for web page guiding users through the working modes of a Psychiatrist AI wellversed in mental health. The document type as HTML5 is started by a line that declares the document to be of this type, while the HTML tag that follows it indicates the beginning of the document and the language as English. Beneath the head section, some vital data are being provided: character encoding (UTF-8) and the website title being set as "Psychiatrist Bot". From here the icon that will appear on browser tabs is also linked in. Generally, meta tags which configuring viewport for different devices and are IE versions compatible ones also exist here in order to have good looking screens when working on various devices and browsers versionsOn the one hand, an external CSS stylesheet is used to link it to the document, thus it is the guarantee of the consistent styling and layout management throughout the document. For example, JavaScript library is loaded from Google CDN so that the code will be executing on the clients' computers. The body solution starts with a div container which is used for the webpage title thereafter.

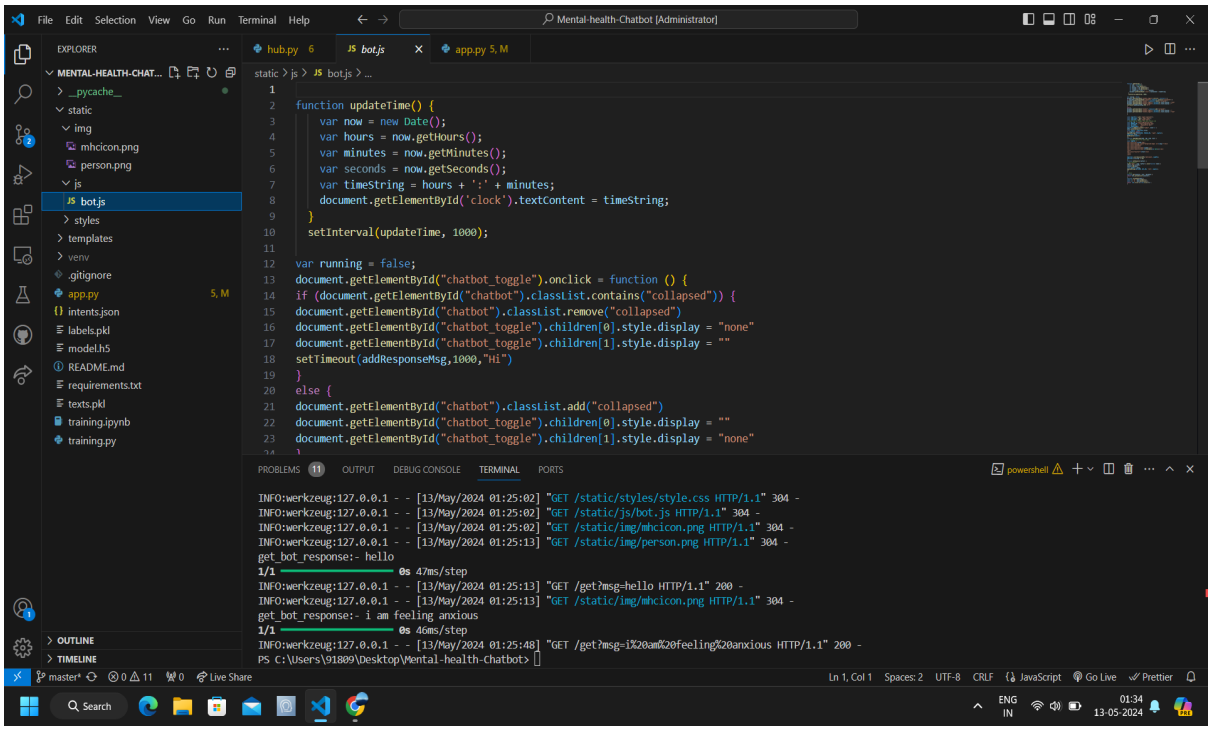


Figure 18:Using java script for web app

Figure 18 depicts that javascript code that implements the function called "updateTime()" as its primary purpose is to change the time show in a webpage to current time continuously. It first envelops some Date object which is current with the time and obtains its hours, minutes and seconds components. These elements are then combined to form a string which reflects the current time. Thanks to function, this time formatted string turns to the text content of an HTML element whose ID is 'clock'. Furthermore the algorithm sets up "updateTime"+ function to be called every second with the setInterval() function. The program also provides event handling functionality for displaying or hiding the chatbot interface. On clicking the button having id = 'chatbot_toggle', the function looks at the status of the chatbot to determine whether its interface is collapsed. If it is collapsed, it expands the interface by using addEventListener() to remove the class 'collapsed' from the chatbox HTML element and show the hidden button, and then it uses setTimeout() to show a message after the duration. On the contrary, if the interface is already expanded, it collapses it, hides the button icon and clears any pending timeouts. Nevertheless, there are some syntax errors and inconsistencies as well which are important to be avoided so that the code may be executed properly (missing parentheses, incorrect method call and incorrect use of properties etc .

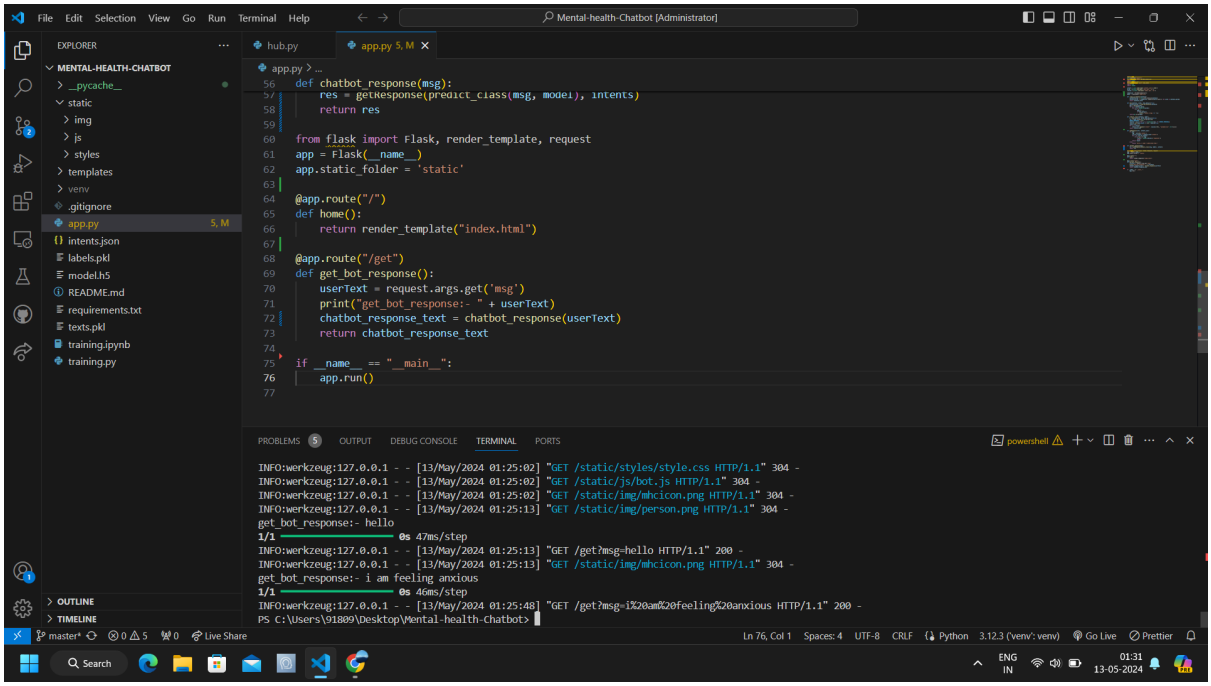


Figure 19: Final Code web app formation

Figure 18 depicts that the final working of the web app and in this code we connect html files css files and java script file together to run the web app on a local host server.

3.5 Key Challenges:

3.5.1) Connection with the Server:

This creates one serious issue in setting up a strong and reliable chain with the server. A powerful server linkage is indispensable for the particular psychological chat bot that is meant to converse with users regarding sensitive issues. For instance, the problems could be related to latency, down servers or chat breakdowns that may make it virtually useless or result in bad user experience.

3.5.2) Lack of Dataset:

Consequently, for instance, in machine learning models like a chatbot, quality and quantity of datasets are important features. Psychological chatbots should have access to multiple databases that ought to be carefully selected and there should also be conversation regarding mentally related issues of interest. Such data sets are crucial; but, if the chatbot does not have them it may be unable to understand more than one client input. Secondly, confirmatory measures should be taken to ascertain the ethical methods employed during sourcing of the data and avoiding breaching customer's privacy.

3.5.3) Generative AI Problem:

Generative AI means training a version to give contextually fitting and empathetic replies which in this case is mainly about how to position psychological discussions. Among some core ventures is creating a chatbot that understands the nuance as well as typically elaborate dynamics of mental conversation. These generative AI dresses should be well scrutinized, corrected and altered to meet ethical considerations for the sake of friendly reactions without negative information, and enormous support.

There are several multifaceted problems surrounding this issue that must be dealt with using all means necessary incorporating knowledge about data technology, psychology, NLP, and server infrastructure.

Chapter 4:TESTING

4.1) Testing Strategy:

4.1.1) User Testing

Software development manner: Testing man or woman elements, features, or modules of the chatbot code one after the other is an vital step. Unit testing is a manner that includes testing every thing of the device one at a time to make certain it features as meant. Developers can locate and fix troubles early on by segmenting the code into smaller, greater possible chunks. Developers create test instances for precise functions or modules all through unit testing, providing inputs and confirming that the outputs suit the meant consequences. They can find and deal with insects, mistakes, or uncommon conduct in discrete regions of the codebase thanks to this targeted method.

4.2.2) Integration Testing

Ensuring the general functionality and dependability of the device calls for verifying the smooth interaction among the diverse components of the chatbot. This sort of checking out, also known as integration testing, is focused on evaluating how different components interact to produce the preferred end result.

Integration trying out for a chatbot might entail examining how essential components work together, just like the database, the natural language processing (NLP) engine, and any 1/3birthday party APIs.

4.1.3) Functional Testing

The comprehensive system of testing the chatbot's universal functionality assesses how properly the gadget performs in real situations. This sort of testing, additionally called ceaseto-give up trying out, evaluates how properly the chatbot can manipulate consumer interactions from starting to stop. End-to-quit checking out incorporates providing a lot of consumer inputs to the chatbot on the way to validate its natural language processing (NLP) abilities. To assure that the chatbot efficiently translates consumer motive, check instances have to encompass quite a number query sorts, including requests, commands, and questions

Chapter 5: RESULT AND EVALUATION

5.1) Results:

After collecting the dataset we have pre processed the data by applying the various techniques like Contraction mapping Punctuation mapping Removing the url's Then we have started implementing the code in which we have applied the random forest algorithm along with the activation functions like sigmoid function reLu fuction . We have separated the user input into four categories Anger Joy Sadness Fear Then the user will elaborate the feelings about himself and our chatbot will reply accordingly to that

Accuracy anger fear sadness joy anger fear sadness joy

	precision	recall	f1-score	support
0	0.78	0.75	0.76	71
1	0.75	0.86	0.80	70
2	0.76	0.70	0.73	71
3	0.90	0.88	0.89	98
accuracy			0.80	310
macro avg	0.80	0.80	0.79	310
weighted avg	0.80	0.80	0.80	310

Figure 20: Result of random forest algorithm

Figure 20 depicts that the results of the random forest algorithm which shows the accuracy precision recall score different emotions which are denoted by the number where 0 denotes anger 1 denotes fear 2 denotes sadness and 3 denotes the joy.

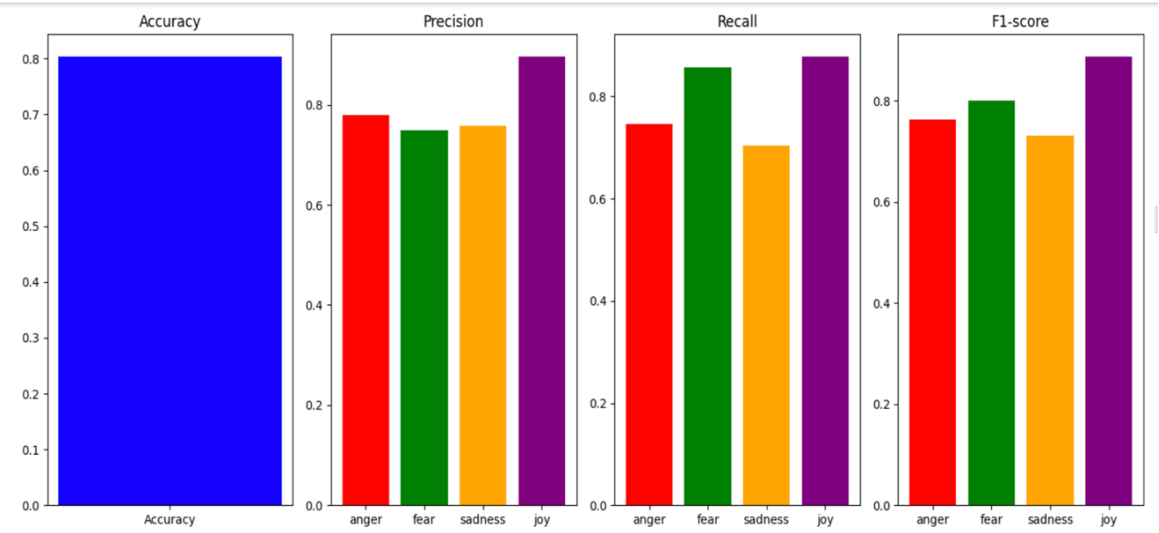


Figure 20: Results of various categories

Figure 21 depicts that the score of the differ parameters which are the different emotion like anger fear joy sadness these scores are the score of accuracy Precision recall F1-Score which is obtained by applying the random forest which is a supervised learning technique

Table 2: Results of Random forest

Category	**Precision**	**Recall**	**F1-Score**
0 (Anger)	0.78	0.75	0.76
1 (Fear)	0.75	0.86	0.80
2 (Sadness)	0.76	0.70	0.73
3 (Joy)	0.90	0.88	0.89
Weighted average	0.80	0.80	0.80

Overall accuracy is 80%

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
              precision    recall  f1-score   support

       anger       0.89      0.94      0.91       760
        fear       0.97      0.79      0.87       995
         joy       0.91      0.87      0.89       714
     sadness       0.76      0.95      0.84       673

    accuracy                           0.88      3142
   macro avg       0.88      0.89      0.88      3142
weighted avg       0.89      0.88      0.88      3142
```

Figure 21: Results of Naïve Bayes algorithm

Figure 22 depicts that the results of the naïve bayes algorithm which shows the accuracy precision recall score different emotions anger, fear, joy ,sadness

Table 3 Results of naïve bayes algorithm

Category	**Precision**	**Recall**	**F1-Score**
Anger	0.89	0.94	0.91
Fear	0.97	0.79	0.87
Sadness	0.76	0.95	0.84
Joy	0.91	0.87	0.89
Weighted average	0.89	0.88	0.88

Overall Accuracy is 88%

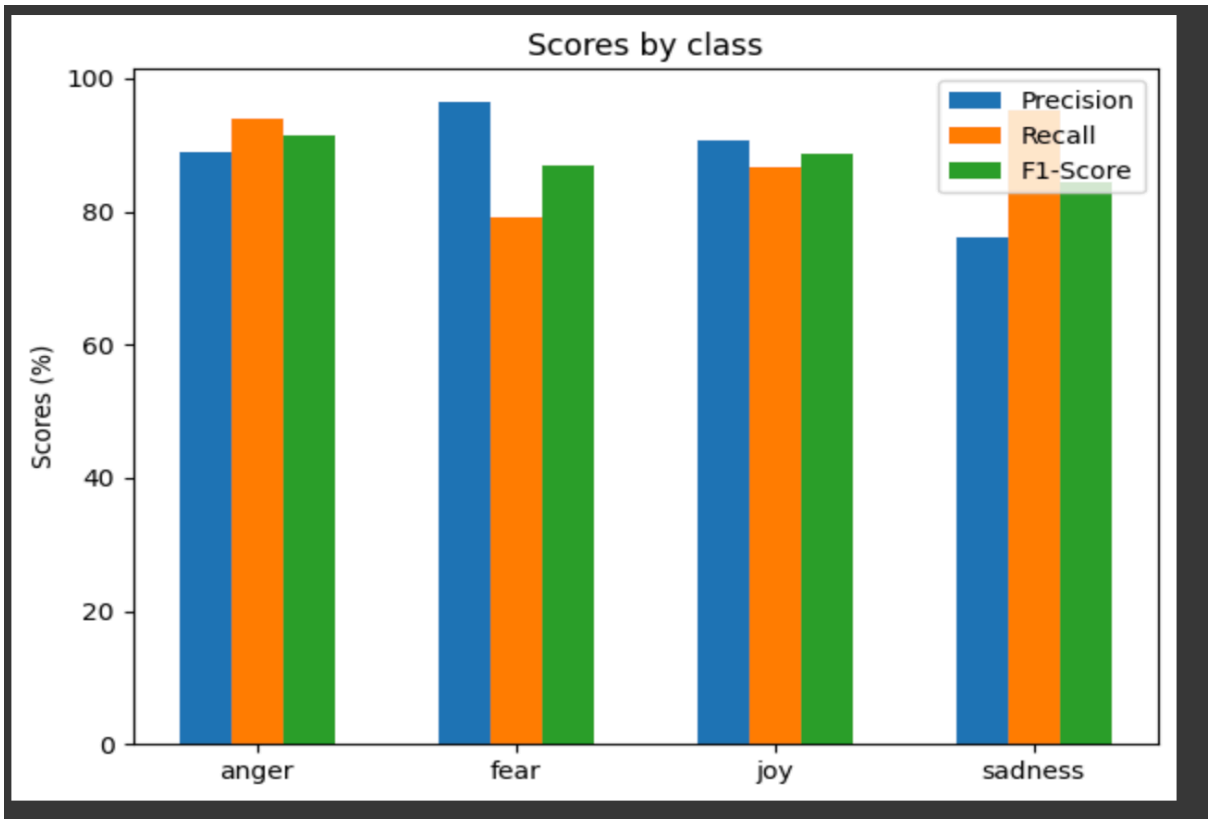


Figure 22: Results when using the naïve bayes algorithm

Figure 22 depicts that the score of the differ parameters which are the different emotion like anger fear joy sadness these scores are the score of accuracy Precision recall F1-Score which is obtained by applying the naïve bayes algorithm

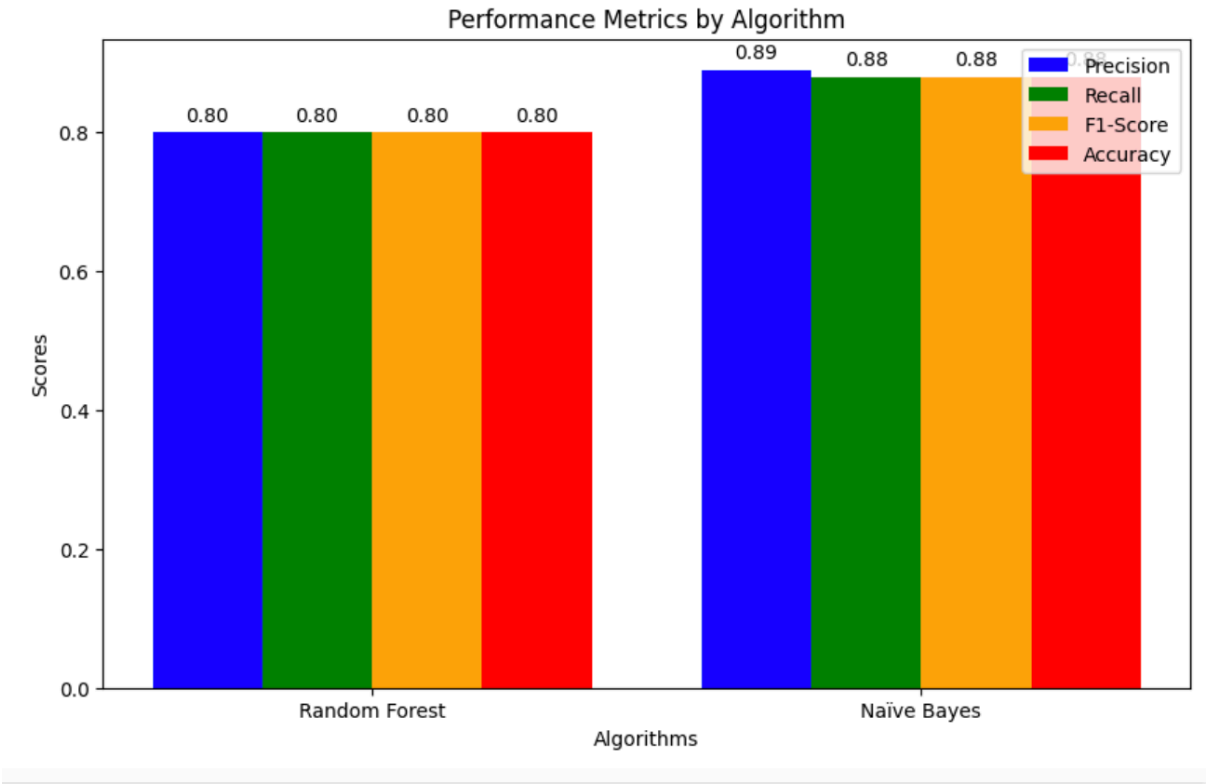


Figure 23: Metrics Comparison of Naïve Bayes And Random Forest

Figure 23 Depicts the metrics comparison of naïve bayes and random forest these are two models which are applied in the detection of the emotion which are categorized into four parts anger fear joy sadness and these metrics are in a bar chart which shows precision recall and f1-score of these two algorithm where the accuracy of naïve bayes algo is 88 % and the random forest has 79 percent

Table 4 Metrics Comparison of Naïve Bayes And Random Forest

Algorithm	Precision	Recall	F1-Score	Accuracy
Random Forest	0.80	0.80	0.80	0.80
Naïve bayes	0.89	0.88	0.88	0.88

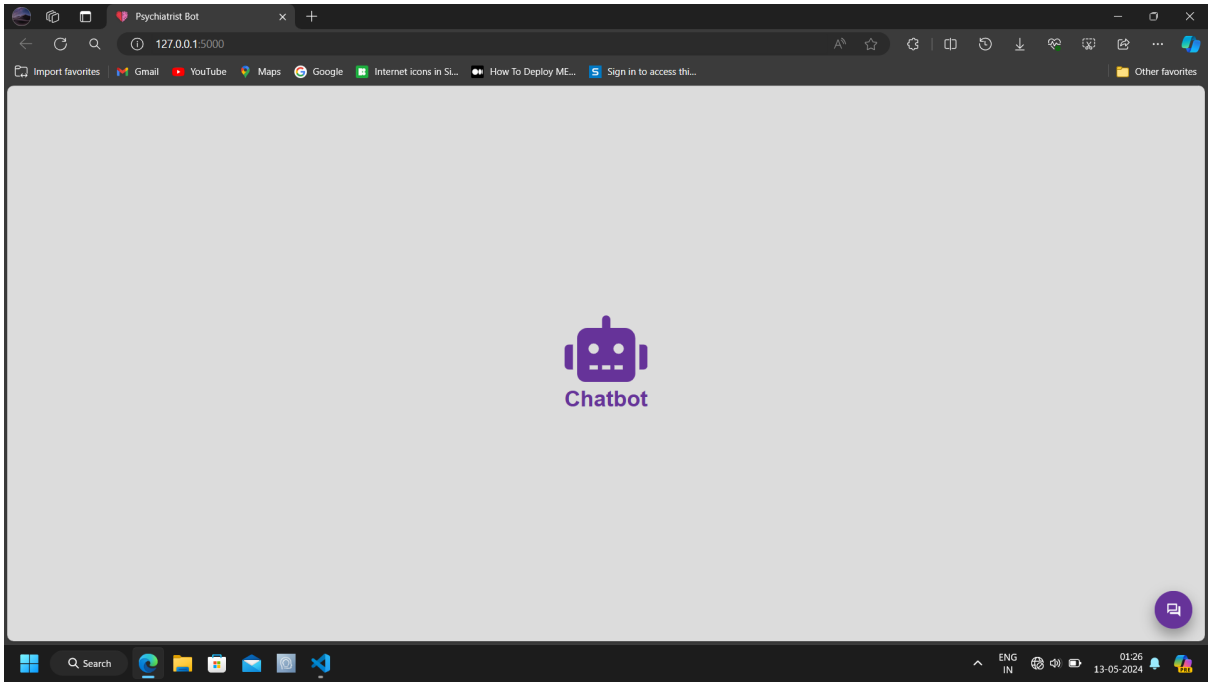


Figure 24: Starting webpage of the psychological chatbot

Figure 24 depicts that the starting webpage of psychological chatbot where by clicking the messaging icon we can chat with the chatbot

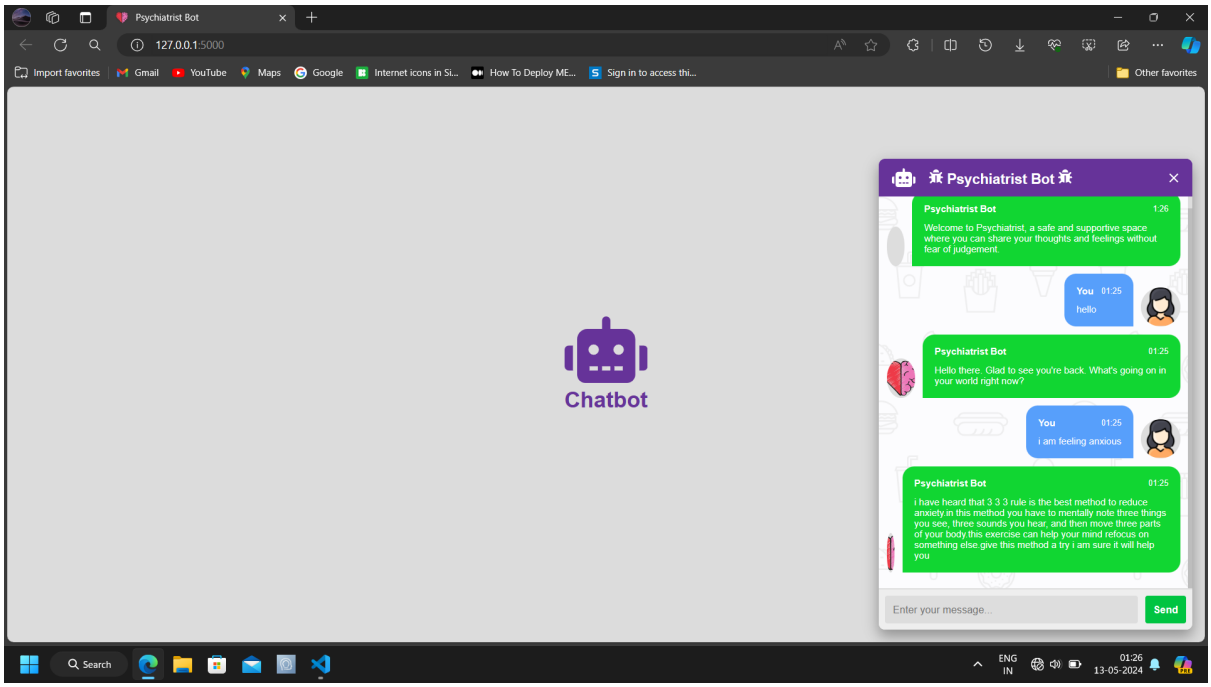


Figure 25:Final Output of the psychological chatbot

Figure 25 depicts that after clicking the messaging icon we can start the conversation with a chatbot.

Chapter 6: CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

To sum up we made the psychological Chatbot in the psychological chabot it is a platform which is easily accessible by the users and the user can share his/her feelings in a safe environment where no one can judge. Firstly we have taken dataset which is descripted dataset used to train a model where model can answer on its own there are no predefined answers and we have pre-processed the dataset in the pre-processing firstly we have removed the same question answers and then we performed the contraction mapping after applying the contraction mapping we have performed punctuation mapping, in the punctuation mapping we mapped the different punctuation with the different alphabets so that it can be easily ridable and second thing we have removed the URL's where user put the URL then it will be removed automatically. We have coded the chatbot in this model we have applied the random forest algorithm and we have applied the relu and sigmoid function and we have also applied the back progression method by applying this all our project work like following: Firstly, the chatbot will ask the user about his emotions and then the user tell his emotion then the Chabot categorise the emotion into four parts these are anger, fear, joy, sad. After depicting the emotion the chatbot will ask a question related to it, then continuing with a word for "may I help you" if the user said yes the conversation will go on and it will search from the dataset that which paragraph is suitable for the use and by analysing it's emotion it finally gives the answer if the user is satisfied with the answer then he/she can reply after his/her reply the chatbot will continue the conversation and after the satisfaction of the user the chatbot will say good bye. In this chatbot it is a very safe environment where no one can judge and user will get the answer clearly without any stigma pressure. The Overall Accuracy Is 79% and after that we have used the naïve bayes algorithm where we got the better accuracy than the previous random forest . The accuracy is of 88% and after using naïve bayes we made a generative chatbot in this chatbot the user will get more answers and the user will do the conversation more freely. In this chatbot we made a web application where the user can interact and the conversation will be more judgment free. The working of the chatbot is as follow firstly the chatbot will say hello and say how may I help you the the user will tell the mood of him and then the chatbot will ask some more question and the chatbot will give the answer accordingly.

6.2 Future Scope

Advanced Natural Language Processing (NLP): We will apply the advance Natural language processing model in the future to make it more intelligent

Scalability and Performance: We will try to increase the scalability and performance of the model

Model Training and Enhancement: We will apply more models so that it can train well and more efficiently

Diversity in Training Data: We will extract more data so that it trains more

References

1. A. Ahmed, N. Ali, S. Aziz, A. A. Abd-alrazaq, A. Hassan, M. Khalifa, B. Elhusein, M. Ahmed, M. A. S. Ahmed, and M. Househ, "A review of mobile chatbot apps for anxiety and depression and their self-care features," Computer Methods and Programs in Biomedicine Update, Update 1, 2021

2. A. A. Abd-alrazaq, A. A. Alalwan, B. M. Bewick, and P. H. Gardner, "An Overview of the Features of Chatbots in Mental Health: A Scoping Review," International Journalof Medical Informatics, Sep. 2019, doi: 10.1016/j.ijmedinf.2019.103978.

3. A. Ahmed, A. Hassan, S. Aziz, A. A. Abd-alrazaq, N. Ali, M. Alzubaidi, D. AlThani, B. Elhusein, M. A. Siddig, M. Ahmed, and M. Househ, "Chatbot features for anxiety and depression: A scoping review," Health Informatics Journal, vol. 1-17, 2023, doi: 10.1177/14604582221146719.

4. G. Dosovitsky, B. S. Pineda, N. C. Jacobson, C. Chang, M. Escoredo, and E. L. Bunge, "Artificial Intelligence Chatbot for Depression: Descriptive Study of Usage," Palo Alto University, Palo Alto, CA, United States, Dartmouth College, Lebanon, NH, United States, X2AI, San Francisco, CA, United States, 2023.

5. P. Kapoor, P. Agrawal, and Z. Ahmad, "Therapy Chatbot: A Relief From Mental Stress And Problems," VIT University, May 2021.

6. I. Salhi, K. el Guemmat, M. Qbadou, and K. Mansouri, "Towards developing a pocket therapist: An intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation," Indonesian Journal of Electrical Engineering and Computer Science, vol. 23, no. 2, pp. 1200-1211, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1200-1211.

7. E. Bendig, B. Erb, L. Schulze-Thuesing, and H. Baumeister, "The Next Generation: Chatbots in Clinical Psychology and Psychotherapy to Foster Mental Health – A Scoping Review," Verhaltenstherapie, vol. 32, suppl. 1, pp. 64-76, Aug. 20, 2019

8. H. Liu, H. Peng, X. Song, C. Xu, and M. Zhang, "Using AI chatbots to provide self-help depression interventions for university students: A randomized trial of effectiveness," Internet Interventions, vol. 27, p. 100495, Jan. 6, 2022, doi: 10.1016/j.invent.2022.100495.

9. G. Dosovitsky and E. Bunge, "Development of a chatbot for depression: adolescent perceptions and recommendations," Child and Adolescent Mental Health, vol. 28, no. 1, pp. 124–127, 2023, doi: 10.1111/camh.12627.

10. E. A. Aldakheel, M. Zakariah, G. A. Gashgari, F. A. Almarshad and A. A. Alzahrani, "A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators," 2023 Sensors 23, no. 9: 4403. https://doi.org/10.3390/s23094403

11. J. Rashid, T. Mahmood, M. W. Nisar and T. Nazir, "Phishing Detection Using Machine Learning Technique," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 2020, pp. 43-46, doi: 10.1109/SMART TECH49988.2020.00026

12. M. A. Adebowale, K.T. Lwin, M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," 2020 Journal of Enterprise Information Management 36, no. 3 (2020): 747-766.

13. P. Yang, G. Zhao, and P. Zeng, “Phishing website detection based on multidimensional features driven by deep learning,” IEEE Access, 7:15196–15209, 2019

14. T. Nathezhtha, D. Sangeetha, and V. Vaidehi, “WC-PAD: Web crawling based phishing attack detection,” In 2019 International Carnahan Conference on Security Technology (ICCST), pages 1–6, 2019.

15. Y. Huang, Q. Yang, J. Qin, and W. Wen, “Phishing url detection via CNN and attention-based hierarchical RNN,” In 2019 18th IEEE International Conference On 55 Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)

16. M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, “An adaptive machine learning based approach for phishing detection using hybrid features,” In 2019 5th International Conference on Web Research (ICWR), pages 281–286, 2019.

17. S. Parekh, D. Parikh, S. Kotak and S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 949-952, doi: 10.1109/ICICCT.2018.8473085.

18. V. Gupta, A. Mehta, A. Goel, U. Dixit and A. C. Pandey, "Spam Detection Using Ensemble Learning," 2018 Advances in Intelligent Systems and Computing, vol 741. Springer, Singapore.

19. C. Barata et al., "Chatbots for Mental Health Support: A Review of the Literature," in Proceedings of the 10th International Conference on Pervasive Technologies Related to Health and Care (PervasiveHealth), 2016, pp. 1-4.

20. V. Mori et al., "Effectiveness of Chatbots for Mental Health Interventions: A Systematic Review," Journal of Medical Internet Research, vol. 22, no. 1, p. e11473, 2020.

21. R. Xu et al., "Using Chatbots in Mental Healthcare: A Review of the Literature," JMIR Mental Health, vol. 5, no. 3, p. e10262, 2022.

22. S. Breazeal et al., "Conversational Agents and Robots for Emotional Support: A Review of Design Methods and System Capabilities," ACM Transactions on Human-Computer Interaction (TOCHI), vol. 23, no. 5, pp. 1-61, 2016.

23. R. Xu et al., "Designing Chatbots for Effective Mental Health Interventions: Applying a Human-Centered Design Approach," International Journal of Human-Computer Interaction, vol. 38, no. 11, pp. 1222-1241, 2022.

g

ORIGINALITY REPORT

4%	4%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijisae.org Internet Source	1%
2	pubs2.ascee.org Internet Source	1%
3	Alaa A. Abd-alrazaq, Mohannad Alajlani, Ali Abdallah Alalwan, Bridgette M. Bewick, Peter Gardner, Mowafa Househ. "An overview of the features of chatbots in mental health: A scoping review", International Journal of Medical Informatics, 2019 Publication	<1%
4	formative.jmir.org Internet Source	<1%
5	Pratham Taneja, Ronak Bhatia, Shivam Gaur, Juhi Priyani. "Beyond Words: Extracting Emotions from Speech with AI Techniques", 2023 IEEE 4th Annual Flagship India Council International Subsections Conference (INDISCON), 2023 Publication	<1%