VALLEY VIEW UNIVERSITY

INSTITUTE OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE (BSC.) IN COMPUTER SCIENCE DEGREE

TOPIC:

MENTAL HEALTH CARE USING MACHINE LEARNING (NLU AND NLP)

BY:

KANOR KIM KATEY

ID:

220CS01000325

STUDENT:	**SUPERVISOR:**
KANOR KIM KATEY	**JOHN KINGSLEY ARTHUR**
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _	_ _ __ _ _ _ _ _ _ _
SIGNATURE	**SIGNATURE**
DATE:	**DATE:**

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.0 Introduction

Mental health refers to cognitive, behavioral, and emotional well-being (Felman, 2022). According to a recent report, as evidenced by the inclusion of mental health in the Sustainable Development Goals, there has been growing recognition of the important role mental health plays in achieving global development goals in recent years (World Health Organization [WHO]; n.d.). Even though mental health issues are quite common, stigma, expense, and availability of services hinder access to help. Reports show that, as of 2015, 14.3% of deaths worldwide, or approximately 8 million deaths each year, are attributable to mental disorders (Walker, 2015). These problems and their grossly adverse effects are significantly more widespread in low- and middle-income nations. A solution to this presents itself through artificial intelligence (AI)- chatbots, for instance. Chatbots could provide a scalable, less expensive solution than the traditional method, owing to the advances in technology and increase in internet access. Chatbots operate with a conversational format and have been used to give psychoeducation, self-care practices, and skills training in the mental health field, with high satisfaction rates (Suganuma et al., 2018). As such, the focus of this study is the development of a mental health assessment system that employs a conversational AI chatbot, deployed as a web application, to provide a psychological evaluation, diagnostic report as well as various resources in an attempt at a first step to treating one’s mental health issues.

The chatbot would be developed using Natural language processing (NLP) techniques that subject data received from the user through conversation undergoes several stages of preprocessing, as well as classification algorithms such as SVM and KNN for optimal results of a psychological evaluation.

1.1 Subject and Field of Study

The subject of study is mental disorder detection using machine learning and natural language processing, and the field of study is computer science, specifically artificial intelligence.

1.2 Problem Statement

Though effective as companion applications, existing works and systems lack the capability of mental health screening/assessment. Although some works have proven helpful in identifying anxiety-depressive symptoms and providing resources, they act solely as support and are unable to detect or provide a diagnosis in the case of a user seeking to identify (potential) issues concerning their psychological wellbeing. Previous attempts failed due to neglect of datasets for more severe mental health issues. Therefore, this study aims to provide a means of evaluation using various mental health assessment tools such as questionnaires tailored to identifying stress, depression, anxiety, addiction, and other common ailments, use cognitive behavioral therapy (CBT) using NLP as a means of treatment, and supply various resources in the case of more severe diagnoses.

1.3 Study Objectives

1.3.1 Global Objective:

This project aims to create an artificially intelligent chatbot that offers support through CBT, as well as a machine learning model that assesses a user's mental health.

1.3.2 Specific Objectives:

The study's specific objectives will be dependent on achieving the following goals:

i. To assess the mental health status of the user via SCL-90.

ii. To provide, upon request, cognitive behavioral therapy using techniques such as cognitive restructuring and coping strategies.

iii. To provide resources regarding topics such as CBT, depression, anxiety, and stress.

1.4 Background of Study

The Farmer and Stevenson research illuminates a major mental health issue that the United Kingdom faces at work (Stevenson et al., 2017). According to the report, around 300,000 people with long-term mental health problems lose their jobs each year, and approximately 15% of persons already employed have indications of a mental health condition. Moreover, research published by Investors in People shows that charities were ranked third among the top five industries with the most stressed staff (Investors in People, 2018). In the field of mental health care, chatbots are starting to develop. People who live in remote areas or who work multiple shifts may have difficulty getting mental health care visits, and chatbots could be a viable answer. Chatbots have already been deployed to help pupils cope with exam anxiety. Woebot, a chatbot therapist, has recently made news and is gaining popularity. Every week, two million messages are sent. Stanford University conducted a randomized control experiment. Students who used Woebot had much fewer symptoms of depression within two weeks.

1.5 Scope of the Study

The scope of this study focuses on creating a chatbot and a separate machine learning model in a dedicated application to allow users a means of evaluating their own mental wellness and seeking treatment through text-based interaction with the chatbot.

1.6 Significance of the Study

With the current rate of advancement of technology, certain issues and daily tasks are all but obscure, due in large part to the clever and convenient solutions brought about by the latest technological innovations. However, as far as mental wellness and psychology, the traditional method of visiting a therapist/counsellor/psychologist is still the main and, in some parts of the world, only, means of intervention or treatment. This problem is, however, curbed by the introduction of artificial intelligence by way of chatbots in this field. People in distress can now reach out to mental health "chatbots" via the internet, as the demand for mental health services continues to outstrip supply. Artificial intelligence is used in various cases to provide responses. There is a human factor in others. Moreover, chatbots holds numerous automation options that can significantly accelerate services, allowing for an exponential increase in the number of individuals receiving care in this way compared to a human doctor. Seeking mental health treatment by means of a chatbot is also highly cost efficient to the user. The standard rate per hour to visit a therapist is absurdly high in most countries. The problem of availability is highly mitigated as the majority of the population has access to both a smart device and the internet.

1.7 Expected Results of the Study and Possible Use

Upon concluding this project, the expected result is a web-based chatbot application that can accurately provide the user with basic CBT to help manage their mental health. Furthermore, the machine learning model should generate a diagnosis of a user’s mental health after an assessment has been made.

1.8 Presentation of Thesis

Chapter 1:

The general introduction, study objectives, subject and field of study, the background of the study, study scope, significance of the study, methodology used, and expected results are all analyzed in this part.

Chapter 2:

This chapter focuses on the literature review of the study, including an extensive and detailed feature-based analysis of various systems of similar nature.

Chapter 3:

This chapter details the theoretical analysis of the proposed system.

Chapter 4:

The proposed application is studied in depth in chapter four, with context-level diagrams explaining the proposed application.

Chapter 5:

Chapter five includes algorithm and flowchart, data flow diagrams, entity relational charts.

Chapter 6:

This chapter details the system implementation and testing of the application.

Chapter 7:

The results of training of the models and how they are used are covered in Chapter 7.

Chapter 8:

This chapter details the limitations of the study, recommendations, and conclusions.

CHAPTER TWO

LITERATURE REVIEW

2.0 Literature Review

Chatbots have a huge influence in a variety of sectors, particularly in psychology. Although creators warn that these technologies should not be used in place of human contact with specialists, chatbots are available 24 hours a day, seven days a week, unlike human health practitioners. A psychological therapy chatbot seeks to be the initial point of contact for mental health patients, with a focus on privacy and anonymity. These apps were created to proactively monitor patients, be available to listen and converse at any time and from any location, and offer activities to improve users' well-being. Of course, they will never be able to replace a genuine psychologist, but they perform an excellent job and give valuable assistance. This chapter focuses on reviewing existing works that have been developed for the above purposes.

In the case of Mohr et al.,2017, their work shed light on how smaller sample sizes may expose deep learning to over-fitting, which frequently favors more conventional machine-learning techniques. Deep learning, on the other hand, demonstrates superior capability at capturing the intricate characteristics of data that traditional machine-learning methods fail to capture once an adequate sample size has been attained. Deep learning performs significantly better than other approaches as the sample size grows.

The results produced by Shatte et al., 2018 advise to use a variety of strategies to find the algorithm that performs the best for the specific dataset and task since there is no one technique that works best for all problems (Wolpert, 1997). Various machine learning techniques have proven effective in dealing with mental disorders. For instance, support vector machine (SVM), regression, recurrent neural networks and naïve bayes provide improved are applied in cases of depression, emotional and behavioral problems, anxiety, and borderline personality disorder.

A study by Vaishnavi et al., 2022 analyzed the accuracy of five machine learning algorithms in detecting mental health concerns using a variety of accuracy criteria. Logistic Regression, K-NN Classifier, Decision Tree Classifier, Random Forest, and Stacking are the five machine learning algorithms. We compared these methods, put them into practice, and also found that the stacking method, based on a forecast accuracy of 81.75%, was the most accurate (Vaishnavi et al., 2022). The results showed the various ML algorithms and the respective accuracy rates as follows: Logistic regression – 79.63%, K Neighbors Classifier – 80.42%, Decision Tree Classifier – 80.69%, Random Forest – 81.22%, Stacking – 81.75%.

Cho et al., 2019, proved through their study that SVM is a popular ML technique in the field of mental health. It has been used across all mental health fields, and as was predicted, the majority of SVM classifiers developed in the studies showed high accuracy of more than 75%. Data are sparse in the mental health field, which means that measured feature values are discrete and features only capture a small portion of a data point's properties. One factor that makes SVM effective in this field is the data sparsity. For feature selection, several ML methods, including GBM, have occasionally been used in combination. Because they have the advantage of handling a large number of features simultaneously without feature selection, the ensemble methods of GBM and Random Forest were also used as the primary algorithm to categorize some mental health patients. On average, they demonstrated strong SVM performance, however one classifier's accuracy on the test sample was just under 60%. It means that the ML method alone might not ensure highly accurate categorization. Our analysis only employed KNN and Nave Bayes once with SVM, however in certain circumstances their performance was on par with or better than SVM's.

2.1 Reviewed Systems

iHelpr

In previous projects, Inspire Workplaces and Ulster University collaborated to create the chatbot iHelpr as part of a Knowledge Transfer Partnership (Cameron et al., 2019). On the following topics: stress, anxiety & depression, trauma, sleep, and alcohol, iHelpr offers guided self-assessment. Each section enables the user to finish a self-assessment tool and pinpoint specific areas of difficulty. The Perceived Stress Scale (PSS) for stress, PCL5 for trauma, the Audit questionnaire for alcohol, The Sleep Condition Indicator for sleep, GAD7 for anxiety, and the PHQ-9 for depression are the psychometrically validated instruments that are used. The PSS is typically administered on paper, with the user manually calculating their score after circling their response to each question.

WoeBot

The first system put under a microscope is WoeBot. Woebot bills itself as an automated chatbot that tracks users' moods and provides a space for them to express themselves through therapeutic chats. Woebot is built on a platform of Cognitive Behavioral Therapy (CBT) – a talking therapy described by the UK National Health Service as a talking therapy that can help patients manage mental health conditions by changing the way they think and behave by empower individuals to reinterpret one's negative thoughts into positive ones – NLP, clinical knowledge, and light-hearted daily talk. Woebot runs on a "freemium" business model, rendering limited access to select services for a period of time, after which, payment plans are provided to continue the use of the app. Woebot is a text-based, however is not available on web platforms. Moreover, Woebot’s focus is on talk therapy, and not mental health screening.

Calm

Calm is a multi-award-winning mindfulness app for iOS, Android, and the web. Calm, which was released in 2012, has more 100 million downloads and 700,000 5-star ratings. Hundreds of soothing activities, useful breathing methods, and sleep stories read by celebrities such as Matthew McConaughey and LeBron James are included in the program. Calm has a lot to offer, and the user interface is rather straightforward to use. Plus, they're always adding fresh material.

Headspace

Headspace's daily guided meditation and mindfulness practices can help you discover serenity, wellbeing, and balance. Try any of their new sleep meditations, such as music, natural soundscapes, or storytelling sleepcasts, before going to bed. The program creates tailored plans depending on your input, allowing you to learn the fundamentals of meditation and progress from there. (Gepp K., 2022). Headspace features a text-based conversational format to interact with its users. The app is available on all mobile platforms, but is not web-based. Headspace provides mainly meditational and mindfulness practices, but is unable to perform mental health screening tests.

Wysa

Wysa, like Woebot, is skilled in cognitive behavioral therapy (CBT) and thinking restructuring. The platform is extremely user-friendly, appealing, and simple to navigate. The distinctions between Wysa and Woebot proved initially difficult to discern. Both are CBT-focused chatbots. Daily check-ins are required for both. Both provide pre-filled responses to make check-ins go more smoothly. Wysa was found to possess and lack the same features as that of Woebot. There was also no free unlimited access to all its services entirely.

Reflectly

Reflectly is a simple digital diary at its core, but it offers much more. If nothing else, looking back over earlier journal entries and tracking your mood over time is fascinating, but the app also feels like a genuinely useful mental health support tool. When you need motivating quotes, this app has you covered. A thought-provoking question to get you writing (Wyciślik-Wilson S., 2021). Reflecty possess and lacks the same aforementioned features as Woebot and Wysa. The only distinction lays in its cost. Reflectly offers access to all its services for free, with no in-app purhcases or payment plans.

Joy

Danny Freed, a design engineer and front-end programmer, used artificial intelligence to create Joy, a chatbot. It debuted three weeks ago on Facebook Messenger. After a friend committed suicide, Freed was encouraged to build Joy in the hopes of encouraging people to talk about their mental wellbeing. Unlike the aforementioned reviewed systems, Joy is available on web-based platform. Furthermore, users do not have to pay to gain full access to Joy’s services. However, Joy is unable to screen and evaluate mental disorders, although it is geared towards users diagnosed with obsessive compulsive disorder (OCD).

PAPER	PROBLEM	OBJECTIVE(S)	METHODOLOGY	DATASET	GAP(S)/FUTURE WO

A Mental Health Chatbot with Cognitive Skills for Personalised Behavioural Activation and Remote Health Monitoring - 2022	Unscalable conventional mental healthcare services to address increasing demand.	Present the design and development of a Behavioural Activation (BA) based AI chatbot and participatory evaluation to confirm effectiveness in providing support for mental health.	PRIME framework - an ensemble of machine learning and deep learning algorithms for classification, clustering, association rules - and NLP techniques that impose a structure onto unstructured text generated by free-flowing online mental health support group discussions.	Online discussions	Low acceptabilit towards treatmen Continuous engag and Recurrent em support.
Behavioral Modeling for Mental Health using Machine Learning Algorithms - 2018	Determine the onset of the mental illness in a target group.	Apply various machine learning algorithms such as support vector machines, and logistic regression to identify the state of	The responses obtained from the target group for the designed questionnaire were first subject to unsupervised	Survey questionnaires	This work can be to include diffe sections of the society and categorizing dif mental illness like anxiety, de etc.,

		mental health in a target group.	learning techniques. The labels obtained as a result of clustering were validated by computing the Mean Opinion Score. These cluster labels were then used to build classifiers to predict the mental health of an individual.		
Predicting Mental Health Illness using Machine Learning Algorithms - 2022	Early detection of mental health issues to improve patient quality of life.	Identifying five machine learning techniques and assessing their accuracy in identifying mental health issues using several accuracy criteria.	Data Collection, Data Cleaning, encoding data, Finding Co-variance matrix, Scaling and Fitting, Tuning, Evaluation models, Finding Accuracy, Predicting Data and Results.	Survey questionnaires (undetermined)	The data set use research is very and in the futur large data set can be used, research can be to the same for r accuracy.

An Effective Approach for Mental Health Prediction Using Machine Learning algorithm - 2022	Mental health prediction using machine learning is to manage and detect social network mental disorders (SNMDs)based on the Twitter data	Quantify features and patterns from Twitter to know the symptoms and risk factors of mental disorders by using methods of machine learning.	The proposed methodology is based on KNN classification algorithm, which shows an improvement over one of the existing methodologies which is based on SVM classification algorithm. These algorithms have been executed using a machine-learning tool called SPYDER.	Twitter tweets	Prediction of me health is limite the English lang
Mental Health Diseases Analysis on Twitter using Machine Learning – 2021	Using Twitter Attitude Analysis to predict moods and mental health issues	This paper centres around the conduct impact investigation dependent on heterogeneous well-being information created in online media conditions.	Sentiment quantification through dataset collection, preprocessing, feature engineering, system modeling, and performance evaluation.	Twitter tweets.	Classifying how features concern sentiments, whic positive-negativ neutral, affect opinion and how opinions are.

Evaluating Mental Health using Twitter Data – n.d.	Analysis of mental health phenomena on social media platforms (Twitter).	Show the benefit of social media analysis by comparing traditional sources of research to collected data from Twitter.	Developed a deep neural network to detect Tweet sentiments with the following layers: 1) Embeddings Layer 2) Convolution Neural Network Layer 3) Dropout layer 4) Max Pooling Layer 5) Long Short-Term Memory (LSTM) Layer 6) Dropout layer 7) Dense Layer	Twitter Tweets	Further research usage of differe combinations of neural networ and activation f as well as on th keywords which indicate the dep in the tweets.
Predicting Anxiety, Depression and Stress in Modern Life using Machine Learning Algorithms - 2020	Prediction of depression, stress and anxiety using machine learning algorithms.	This research focused on detecting anxiety, depression and stress using the Depression, Anxiety and Stress Scale questionnaire (DASS 21).	Data were collected from a total of 348 participants via Google forms and subsequently classified using five machine learning algorithms –	Survey question naires	The accuracy of Bayes was found highest, althoug Forest was identified as th model, producing imbalanced class

			namely Decision Tree, Random Forest Tree, Naïve Bayes, Support Vector Machine and KNN.		

2.2 Feature-Based Analysis

Name of System	Mental Health Resource Provision	CBT Treatment	Mental Health Screening
Replika	Yes	Yes	No
Calm	Yes	Yes	No
Wysa	No	Yes	No
Headspace	No	Yes	No
Woe Bot	**No**	**Yes**	**No**
iHelpr	No	Yes	No
Reflectly	No	Yes	No
Happify	No	Yes	No
My Proposed System	**Yes**	**Yes**	**Yes**

CHAPTER THREE

THEORETICAL FRAMEWORK

3.0 Theoretical Framework

The strategies covered in this chapter are those that shall seek to address the research gaps mentioned in chapter 2. The proposed study will be divided into sections that describe the dataset(s), using NLP to perform preprocessing, the splitting, and the machine learning techniques that were used to develop baseline models. The figure below depicts the theoretical framework or architecture of this study.

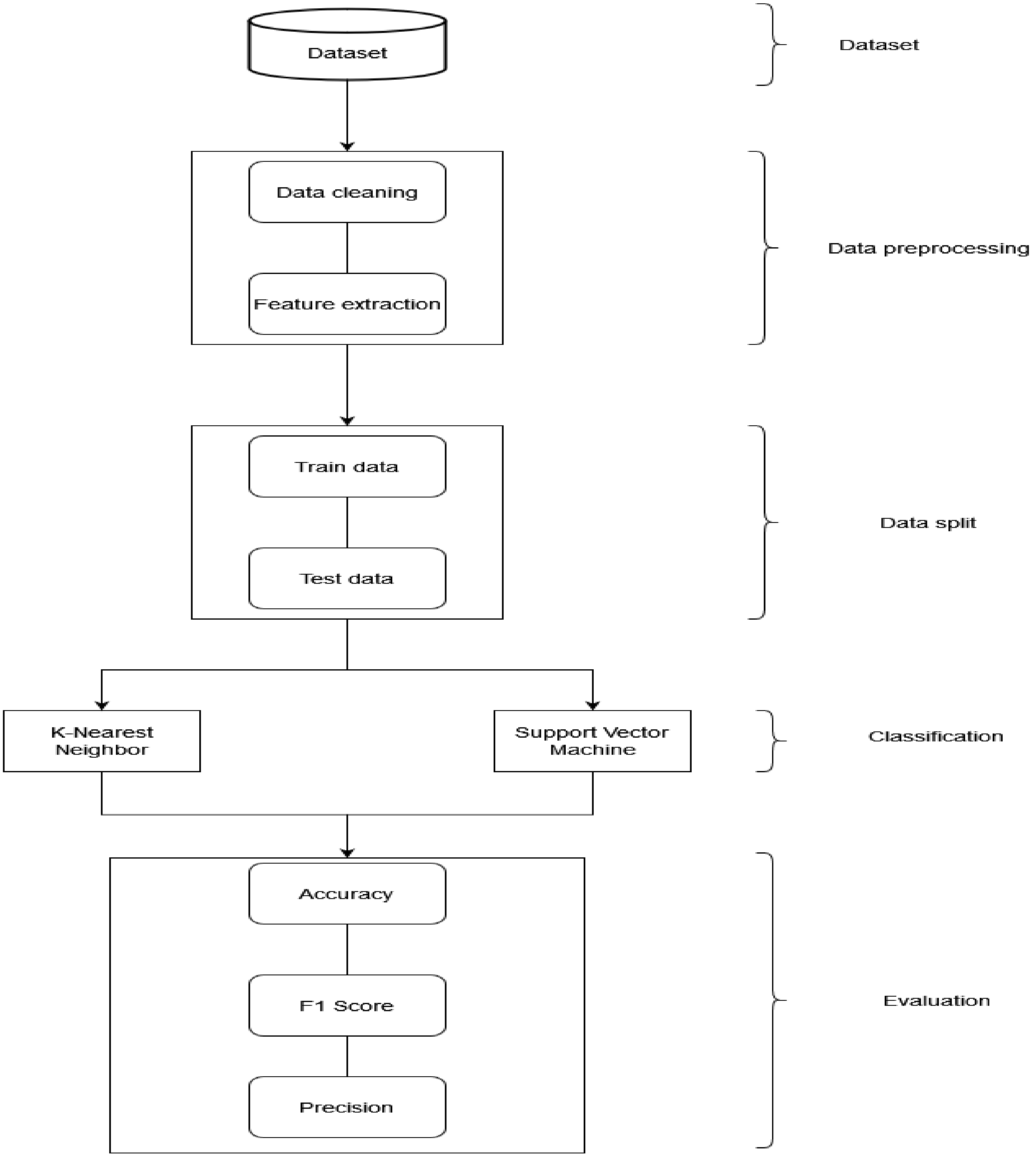


Figure 3.0: Diagram of Proposed Framework

3.1 Dataset Description

The dataset, Symptom Checklist 90 (SCL-90), was acquired through the data repository website Kaggle. It comprises over 4000 entries of participants’ results of the test, as well as their age and gender. The threshold for sound mental wellbeing is 173 points or lower. A higher result is indicative of mental distress.

3.2 Dataset Preprocessing

One of the most important processes is data preprocessing. The data is cleaned in this stage to get rid of redundant information and extra punctuation. However, this dataset was mostly preprocessed, leaving only the task of converting the gender values to 1 and 2, for male and female respectively.

3.3 Data Split

In this data split stage, the ratio used for training and testing the dataset would be 80:20, respectively. This is to ensure the most optimal/accurate predictions.

3.4 Classification

After the dataset is split into training data and testing data, the next step is utilizing three classifiers: K- Nearest Neighbors (KNN), and Support Vector Machine (SVM) to create machine learning models for training and detecting mental health disorders such as depression and anxiety. The aforementioned classifiers are discussed in more detail below:

a) K-Nearest Neighbors:

The K-Nearest Neighbors algorithm, sometimes referred to as KNN or K-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point. Although it can be applied to classification or regression issues, it is commonly employed as a classification algorithm because it relies on the idea that comparable points can be discovered close to one another. Figure 3.1 below demonstrates an instance of KNN's utilization.

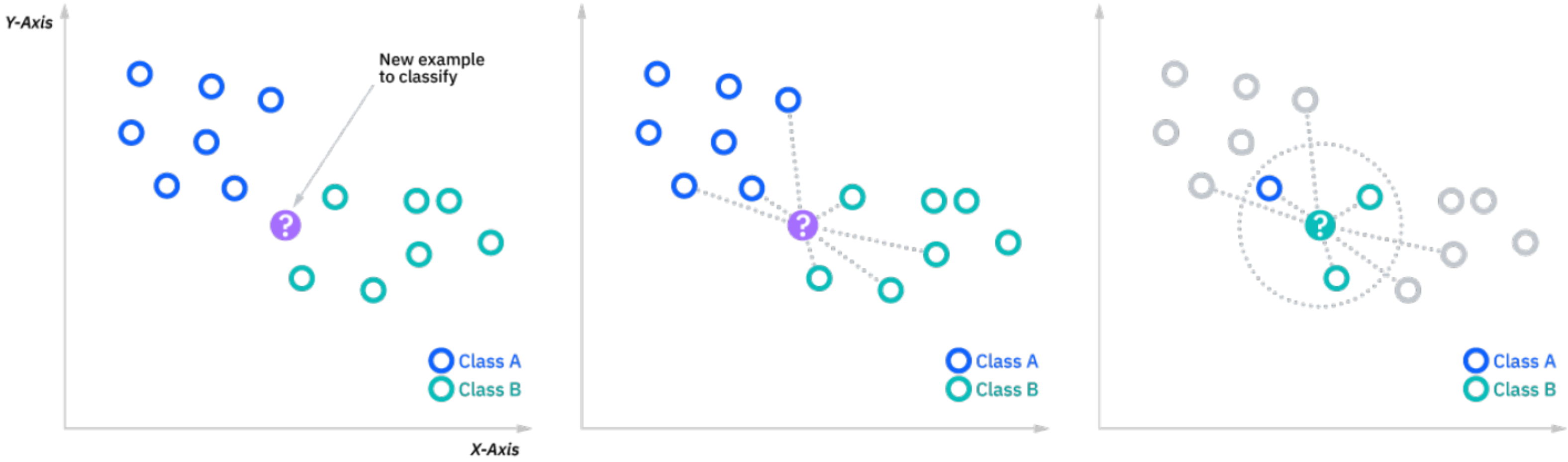
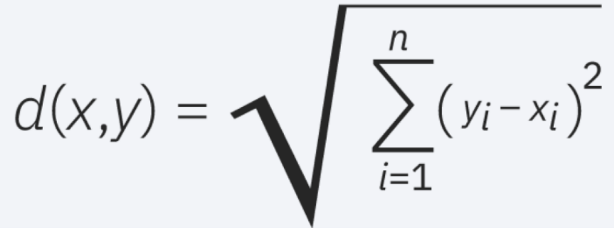


Figure 3.1: KNN Diagram

As shown in Figure 3.1 above, a selected data point among two categories of data points, called Class A and Class

B is to be classified. The distance between the query point and the other data points must be calculated in order to determine which data points are closest to a specific query point. These distance measurements aid in the creation of decision boundaries, which divide query points into various regions. There are numerous distance measures that can be applied, with one of them being Euclidean distance. This distance metric, which can only be applied to real-valued vectors, is the most widely used one. The straight line between the query point and the other point being measured is calculated using the formula below.



$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Figure 3.2: Euclidean Distance Formula

b) Support Vector Machine:

A support vector machine (SVM) is a supervised learning model that employs classification techniques to solve two-group classification problems. An SVM model can classify new text after being given sets of labeled training data for each category. They offer two key advantages over more recent algorithms like neural networks: greater speed and improved performance with fewer samples (in the thousands). As a result, the approach is excellent for text classification issues, where it's typical to only have access to a dataset with a few thousand tags on each sample. Sub-gradient descent and coordinate descent are two recent algorithms for finding the SVM classifier. When working with large, sparse datasets, both techniques have shown to be significantly more effective than the conventional method; sub-gradient methods are particularly effective when there are many training examples, and coordinate descent when the dimension of the feature space is high.

$$f(\mathbf{w}, b) = \left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^{\mathsf{T}}\mathbf{x}_i - b)\right)\right] + \lambda\|\mathbf{w}\|^2.$$

Figure 3.3: Sub-gradient Descent SVM

$$\text{maximize } f(c_1 \ldots c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (x_i \cdot x_j) y_j c_j,$$

$$\text{subject to } \sum_{i=1}^{n} c_i y_i = 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i.$$

Figure 3.4: Coordinate Descent SVM

3.5 Evaluation

This stage involves evaluating the model's performance. The performance of the model will be assessed using the accuracy score, precision, and other widely used assessment measures.

a) **Accuracy:**

One parameter for assessing classification models is accuracy. The percentage of predictions that our model correctly predicted is known as accuracy. The following is the official definition of accuracy:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Accuracy can also be determined in terms of positives and negatives for binary classification, as seen below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP, TN, FP, and FN stand for True Positives, True Negatives, and False Positives, respectively.

b) **F-1 score:**

In binary classification statistical analysis, a test's accuracy is measured by the F-score or F-measure. It is determined by dividing the number of true positive results by the total number of positive results, including those that were mistakenly identified, and by the number of samples that should have been classified as positive. This calculation is done using the test's precision and recall. The following is the F1 Score:

$$\boxtimes\boxtimes 1\ \boxtimes\boxtimes\boxtimes\boxtimes\boxtimes\boxtimes\boxtimes\boxtimes\boxtimes\boxtimes = \frac{\boxtimes\boxtimes\boxtimes\boxtimes}{\boxtimes\boxtimes\boxtimes\boxtimes + \frac{1}{2}(\boxtimes\boxtimes\boxtimes\boxtimes + \boxtimes\boxtimes\boxtimes\boxtimes)}$$

CHAPTER FOUR

ANALYSIS OF PROPOSED SYSTEM

4.0 Overview of the Proposed System

4.0.1 Functional Requirements

- The system shall initiate/engage in conversation with the user about mental health concerns.

- The system shall detect any patterns or signs of mental disorders through targeted questions regarding specific mental illnesses such as depression, stress and anxiety.

- The system shall provide the user with resources upon request or per suggestion aimed at dealing with identified mental health issues.
- The system shall provide a report of identified mental health disorder(s) or detected symptoms to user.

4.0.2 Non-Functional Requirements

- To give the impression that they are conversing with a person rather than a machine, the bot should take a moment before responding.
- The system should be active at all times.
- The system should be able to comprehend shorthand writing, and informal language.
- The system should guard against vulgar language.

4.1 Major Features of the Proposed System

4.1.1 Depression Detection

Through interaction with the user, the chatbot should be able detect low mood and depressive characteristics in the context the conversation. Moreover, by asking specific questions regarding how the user feels at the time, deductions can be made to determine if the user is experiencing a depressive episode.

4.1.2 Stress and Anxiety Detection

The chatbot should similarly be able to identify traits that express anxiety and/or low-high levels of stress in the user. Upon confirmation that the user is experiencing mental distress, resources are subsequently shared as means of relief.

4.1.3 User Interface

The system's user interface will enable users to initiate a conversation with the chatbot in order to discuss mental health concerns. It should also provide the user with a report of previous conversations and the issues(s) identified in that session with the chatbot. Lastly, there should be a dedicated section to mental health resources such as suicide hotlines, articles concerning various mental disorders and coping mechanisms, and names and contact information of available mental health institutions.

4.2 Benefits of the Proposed System

The benefits of the proposed system include:

- Ease of access to mental health resources.
- Detection of common mental disorder, free of cost.
- Promotion of mental health awareness.

4.3 System Context Diagram of the Proposed System

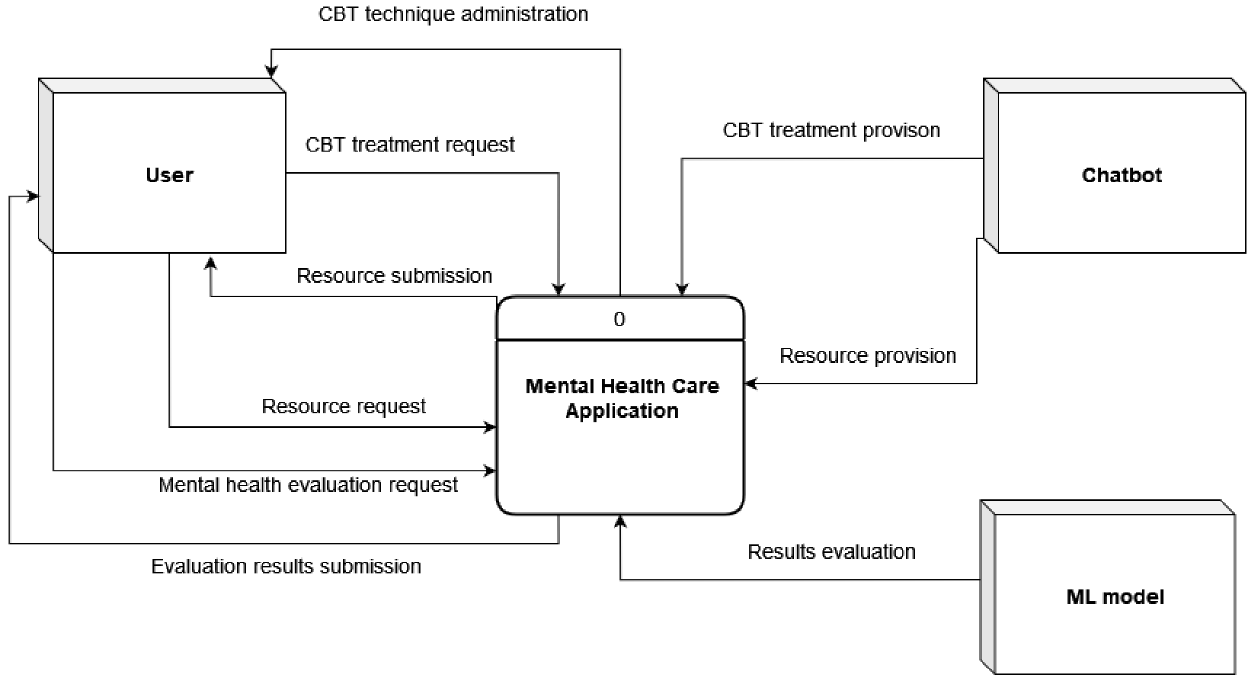


Figure 4.0: System Context Diagram

4.4 Algorithm of the Proposed System

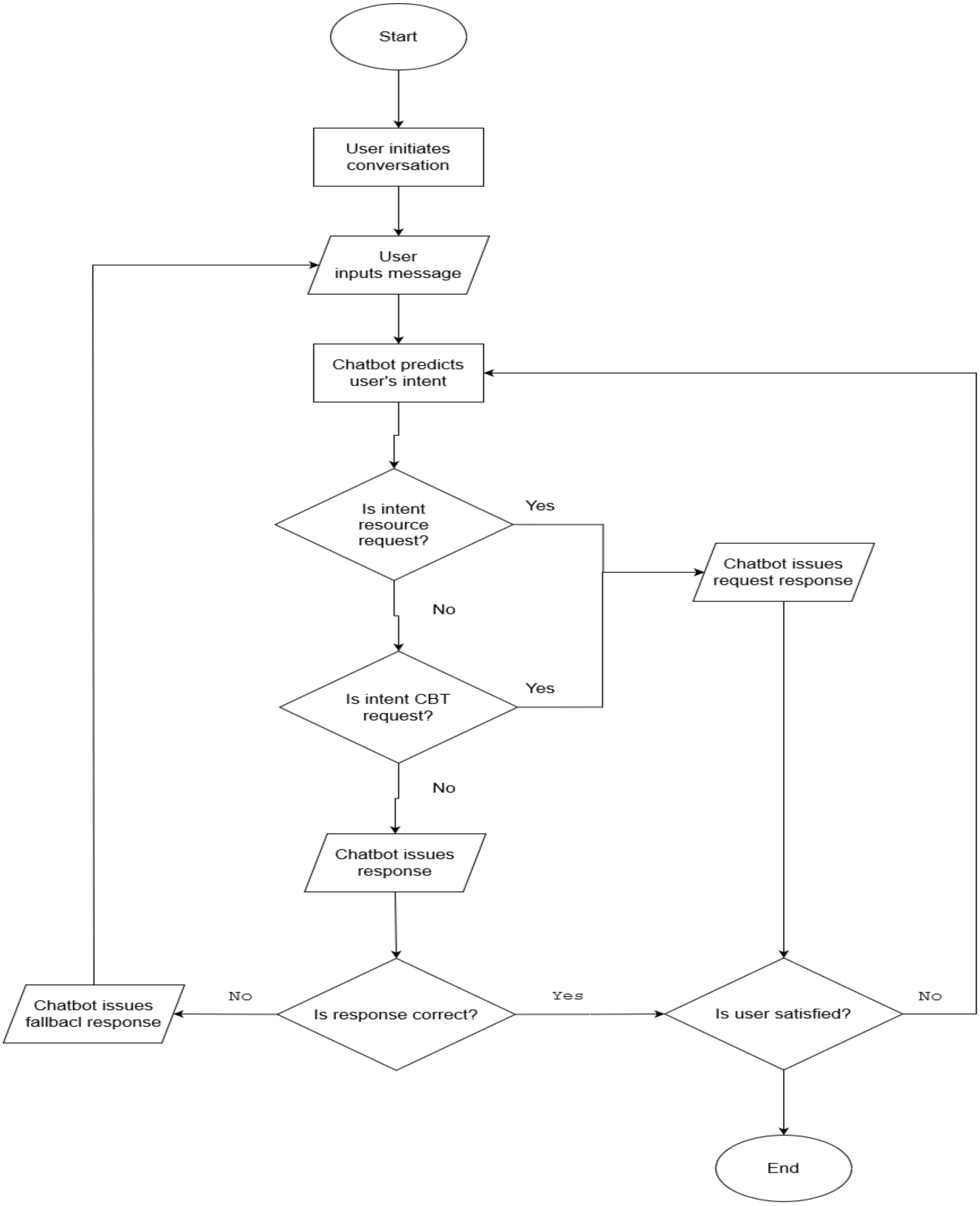

Start
User initiates conversation
User inputs message
Chatbot predicts user's intent
Is intent resource request?
Yes
No
Chatbot issues request response
Is intent CBT request?
Yes
No
Chatbot issues response
Chatbot issues fallbacl response
No
Is response correct?
Yes
Is user satisfied?
No
End

Figure 4.1: Diagram that demonstrates algorithm of proposed system

CHAPTER FIVE

DETAILED DESIGN OF PROPOSED SYSTEM

The system will consist of a Rasa chatbot built with NLU and NLP, and a machine learning model built using TensorFlow and Scikit-learn. The machine learning model will be trained on a dataset of numerous results of the SCL-90 questionnaire, including the age, gender, and choices selected for each of the symptoms in the questionnaire. Two models will be trained using KNN and SVM respectively, with the model trained with SVM performing better than the KNN model based on accuracy, precision, and F1 score.

The chatbot and machine learning model will be integrated into a website using Flask and JavaScript to provide backend functionality. The chatbot will provide preliminary mental healthcare support via CBT treatment, while the machine learning model will evaluate the user's mental health using the SCL-90 questionnaire.

Tools and Techniques Used:

- Rasa: Rasa is an open-source framework for building chatbots using NLU and NLP. It will be used to develop the chatbot component of the system.

- TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It will be used to develop the machine learning model component of the system.

- Scikit-learn: Scikit-learn is an open-source machine learning library for Python. It will be used to preprocess the dataset and train the machine learning models.

- KNN and SVM: The machine learning models were trained using the KNN and SVM algorithms, after which, the results were compared based on accuracy, and f1 score. The model trained using SVM had marginally bettter performance.

- Flask: Flask is a micro web framework written in Python. It will be used to develop the website's backend and handle HTTP requests from the front-end.

- JavaScript: JavaScript is a popular programming language for web development. It will be used to develop the front-end of the website and enable user interactions with the chatbot.

5.1 Functional Processes

- User Input: The system will receive user input in the form of text messages from the user, which is then interpreted and processed by the chatbot to produce appropriate responses.

- NLU and NLP: The chatbot will use NLU and NLP algorithms to analyze the user's input and determine the intent of their message. This will allow the system to provide appropriate responses based on the user's needs. Thusly, python libraries such

- CBT Treatment: The chatbot will provide CBT treatment to users by engaging in interactive conversations aimed at identifying and challenging negative thought patterns and behaviors. The specific CBT treatments are cognitive restructuring and coping strategies.

- SCL-90 Evaluation: The machine learning model will evaluate the user's mental health status based on their responses to the SCL-90 questionnaire. The evaluation will user’s the user's mental health status and recommend appropriate treatment options.

Overall, the system will provide users with access to mental healthcare support and treatment options.

5.2 Data Flow Diagrams (DFDs)

5.2.0 Context Diagram DFD

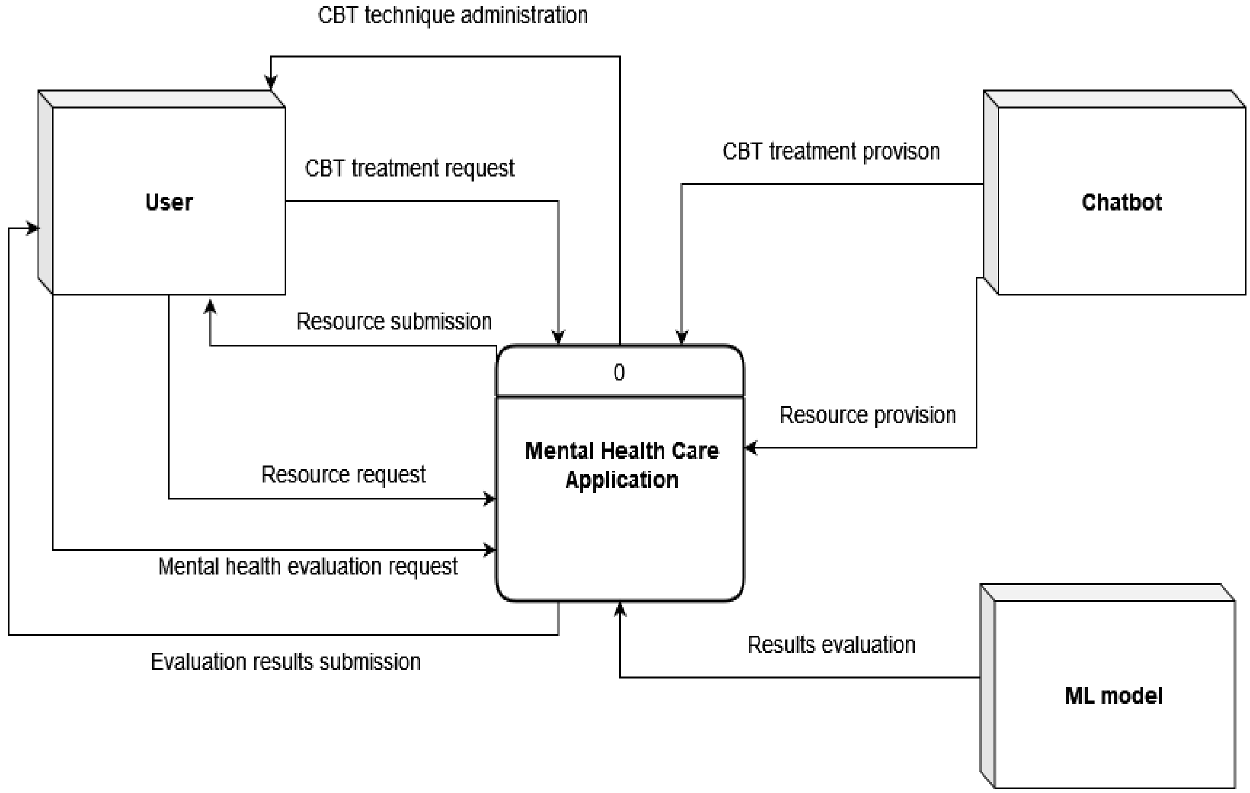


Figure 5.0: Context Diagram DFD of Proposed System

5.2.1 DFD Level 0

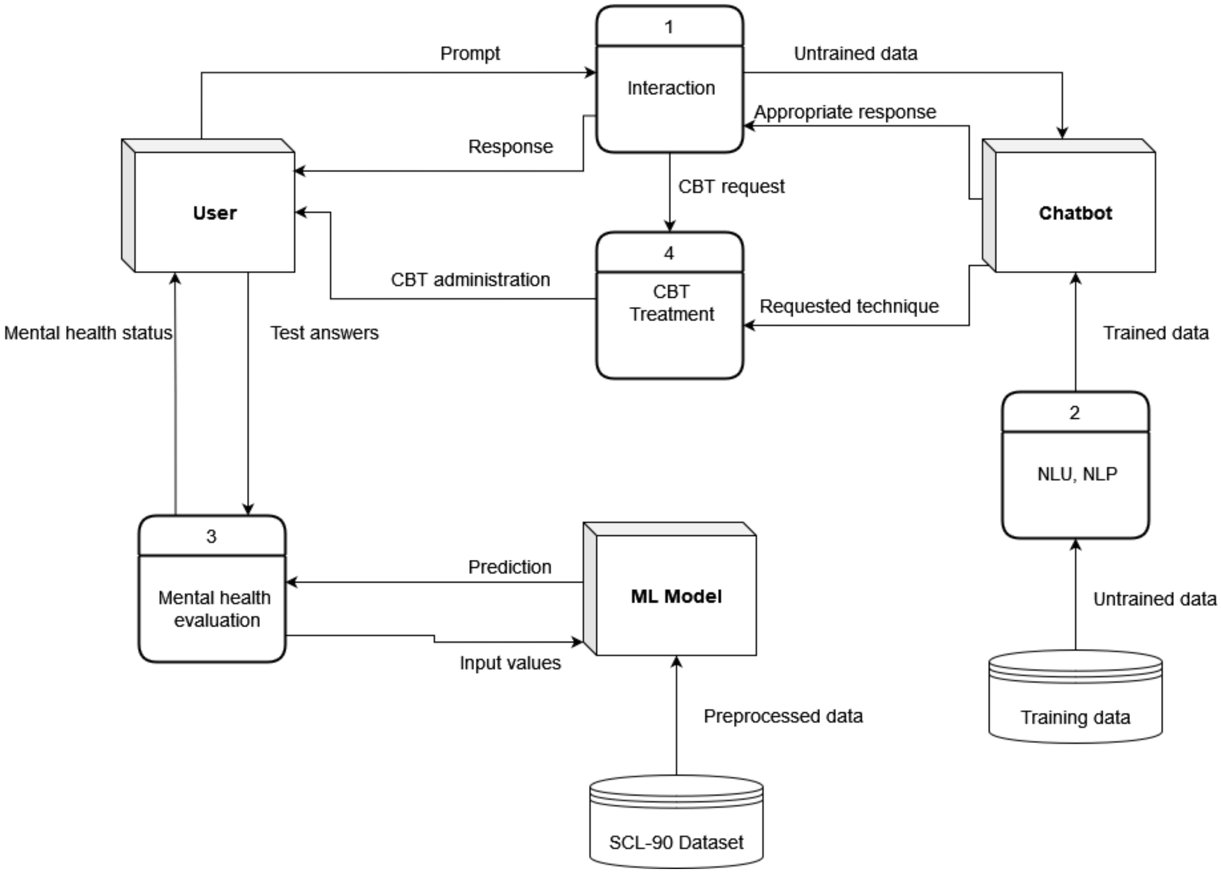


Figure 5.1: DFD Level 0 of Proposed System

5.3 Process Models

5.3.1 Flowchart for DFD 0 - Interaction Process

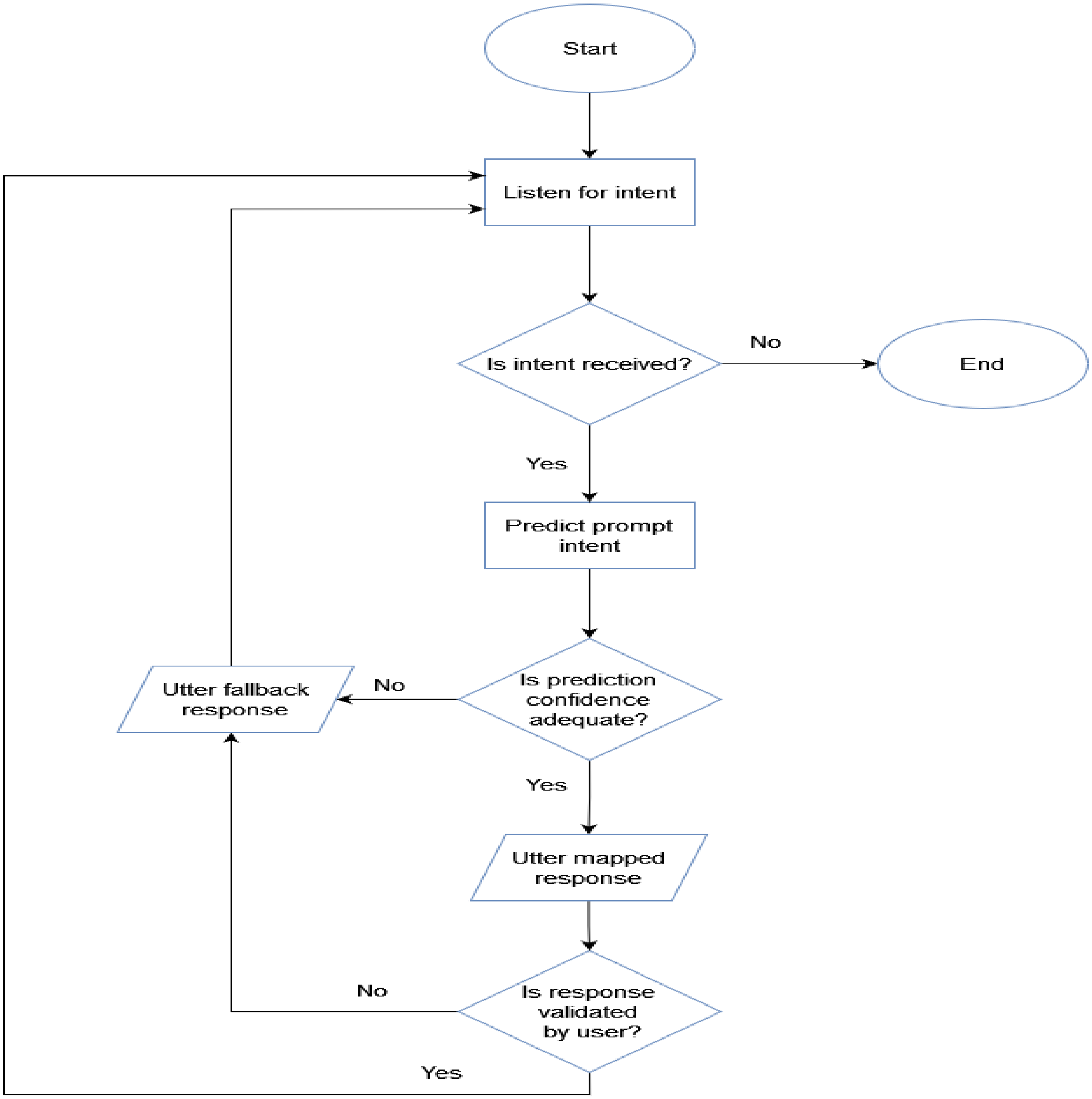


Figure 5.3: Diagram depicting flowchart of interaction process in DFD 0

5.3.2 Flowchart for DFD 0 - CBT Treatment Process

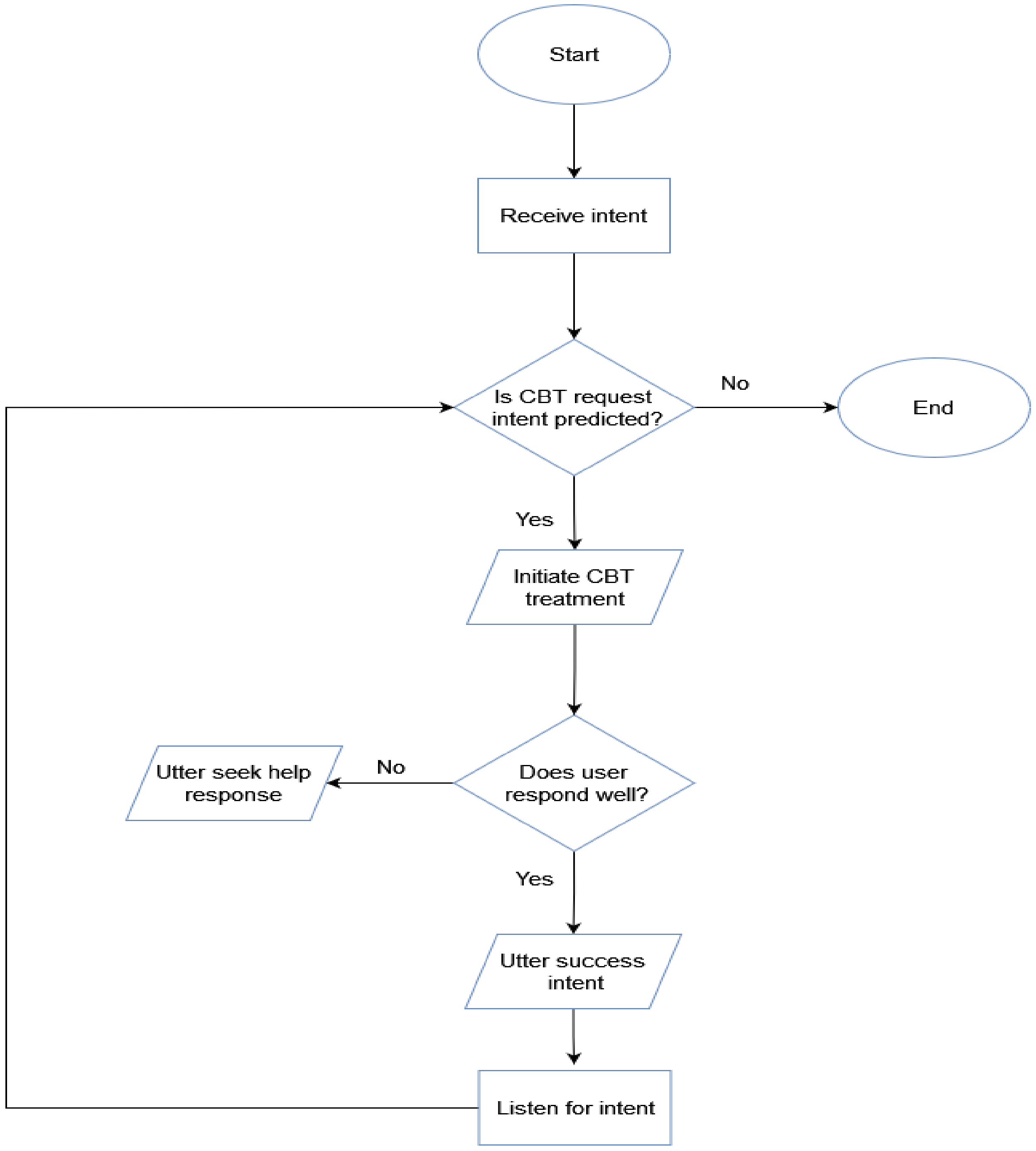


Figure 5.4: Diagram depicting CBT treatment process in DFD 0

5.3.3 Flowchart for DFD 0 - Mental Health Evaluation

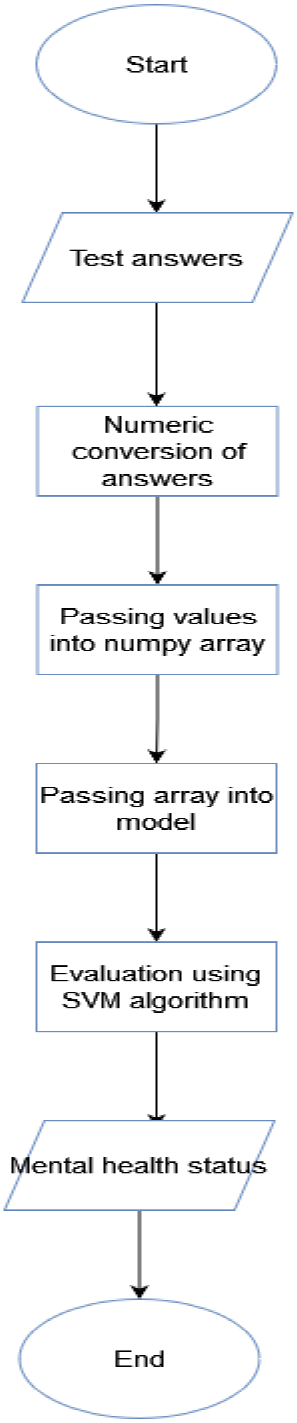

Start
Test answers
Numeric conversion of answers
Passing values into numpy array
Passing array into model
Evaluation using SVM algorithm
Mental health status
End

Figure 5.5: Diagram depicting mental health evaluation process in DFD 0

5.3.4 Story Graph of Training Data for Chatbot

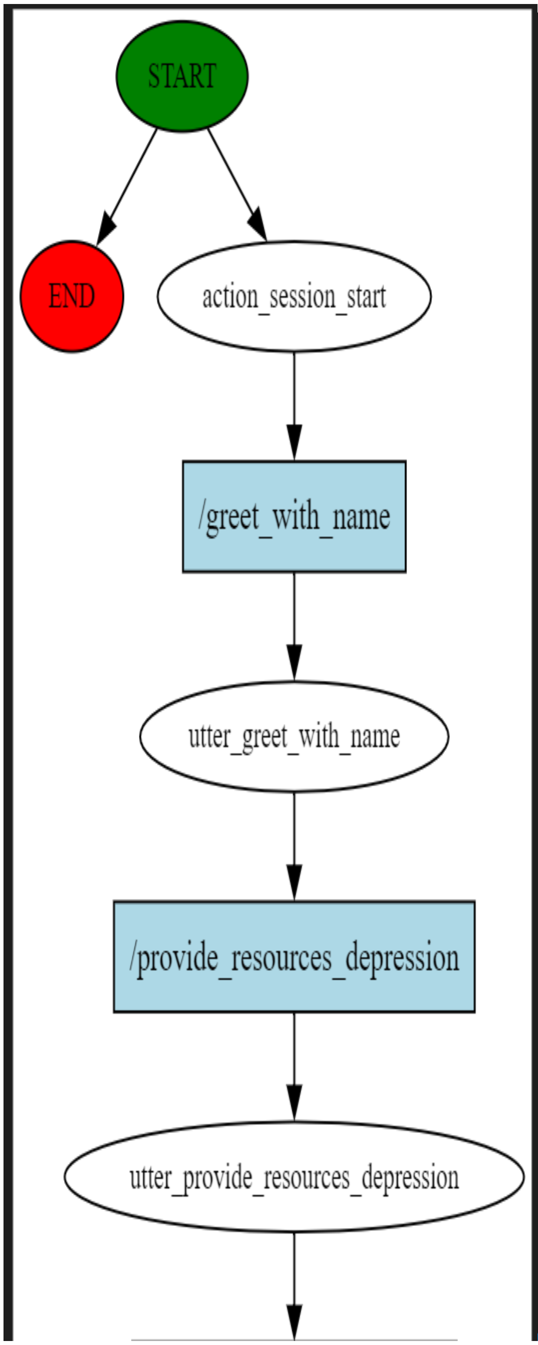


Figure 5.6.1

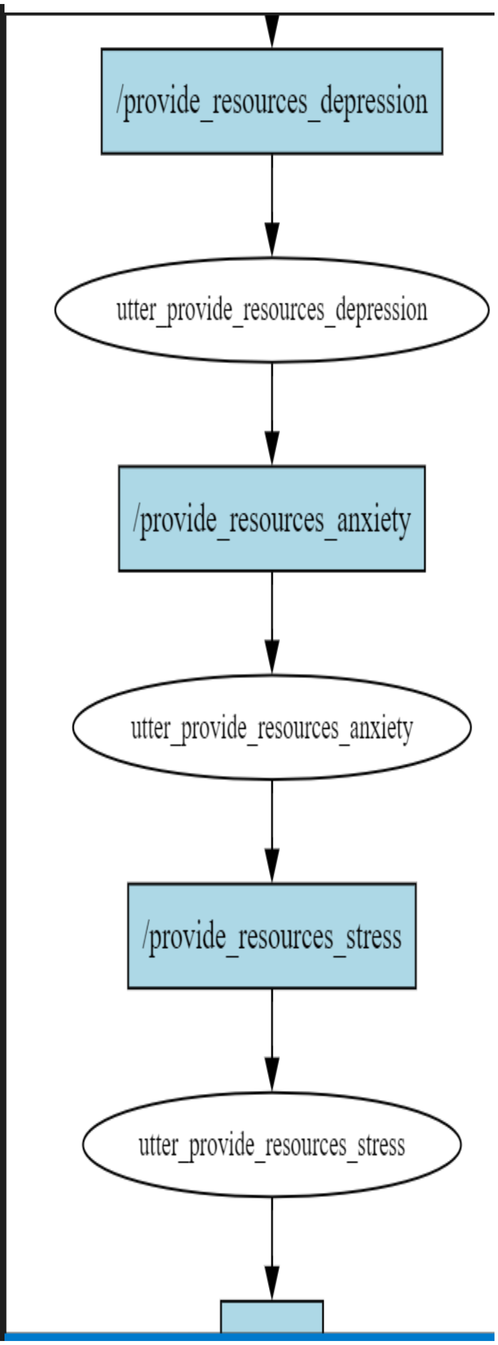

/provide_resources_depression
utter_provide_resources_depression
/provide_resources_anxiety
utter_provide_resources_anxiety
/provide_resources_stress
utter_provide_resources_stress

Figure 5.6.2

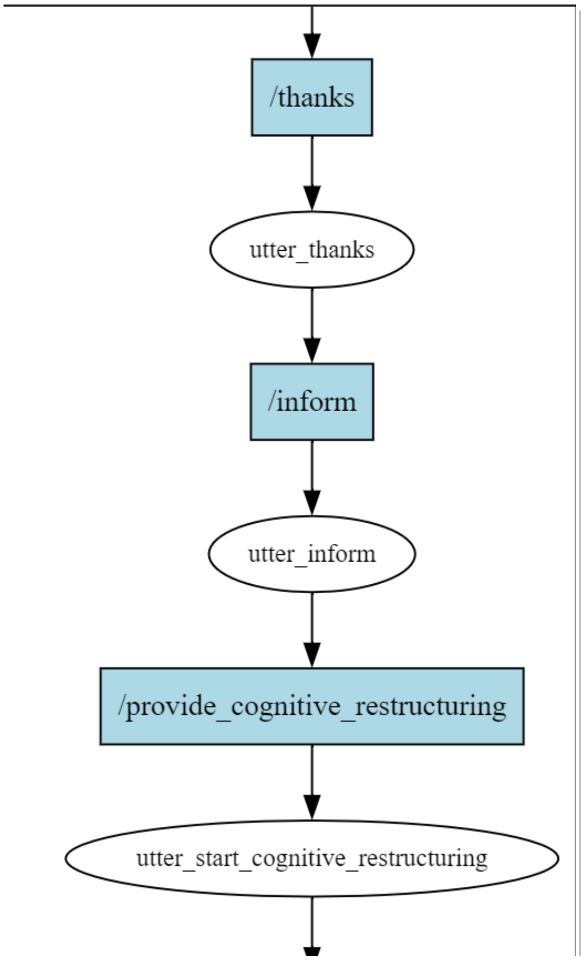

/thanks
utter_thanks
/inform
utter_inform
/provide_cognitive_restructuring
utter_start_cognitive_restructuring

Figure 5.6.3

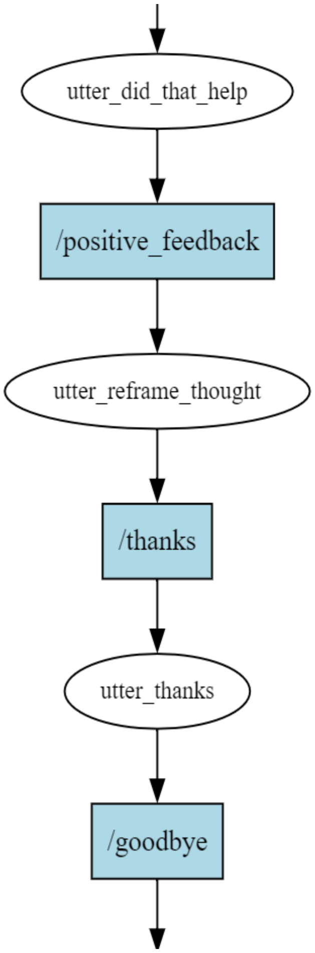


Figure 5.6.4

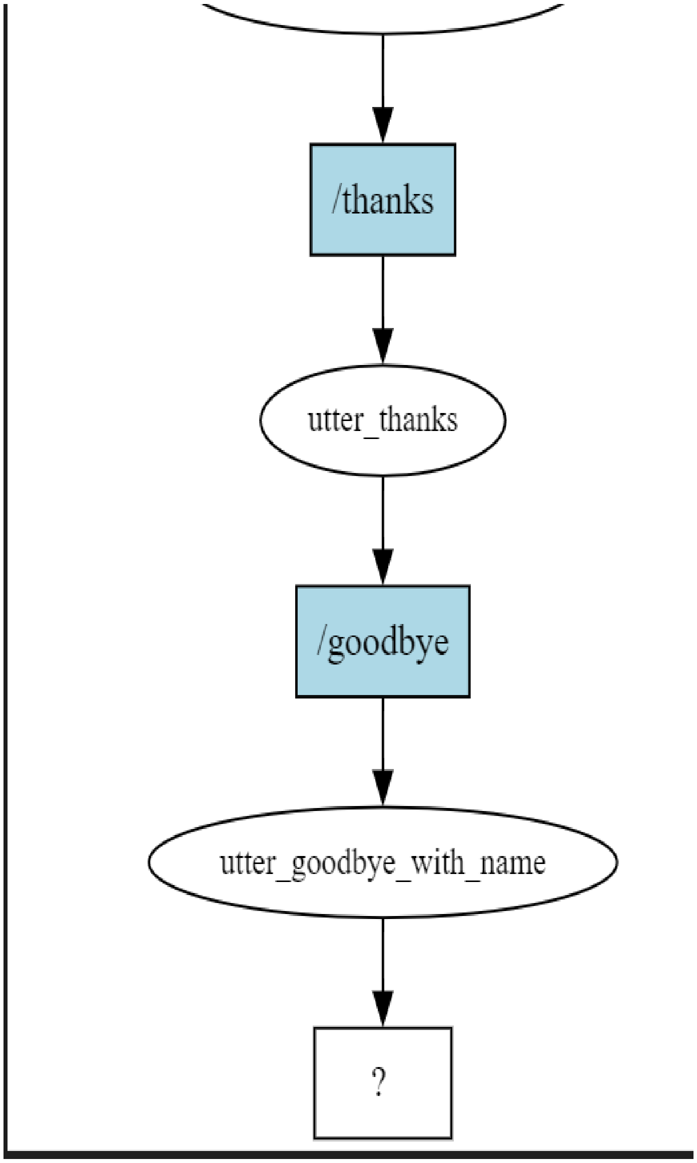


Figure 5.6.5

5.3.5 Use Case Diagram

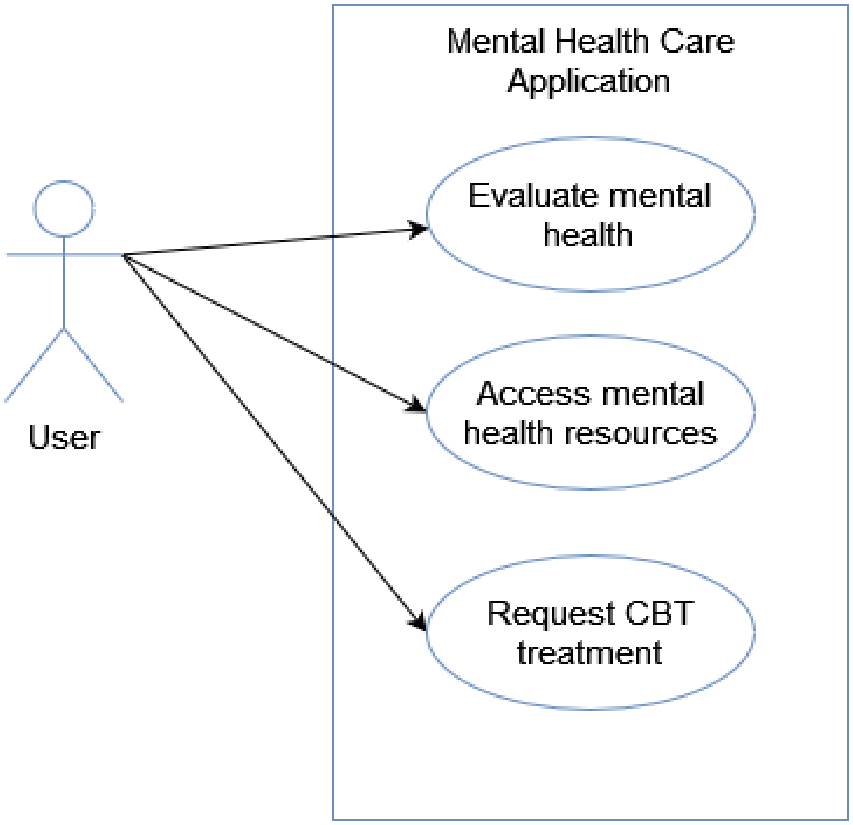

Mental Health Care Application
Evaluate mental health
Access mental health resources
Request CBT treatment
User

Figure 5.7: Use Case Diagram of proposed system

5.4 Sequence Diagram of Chatbot

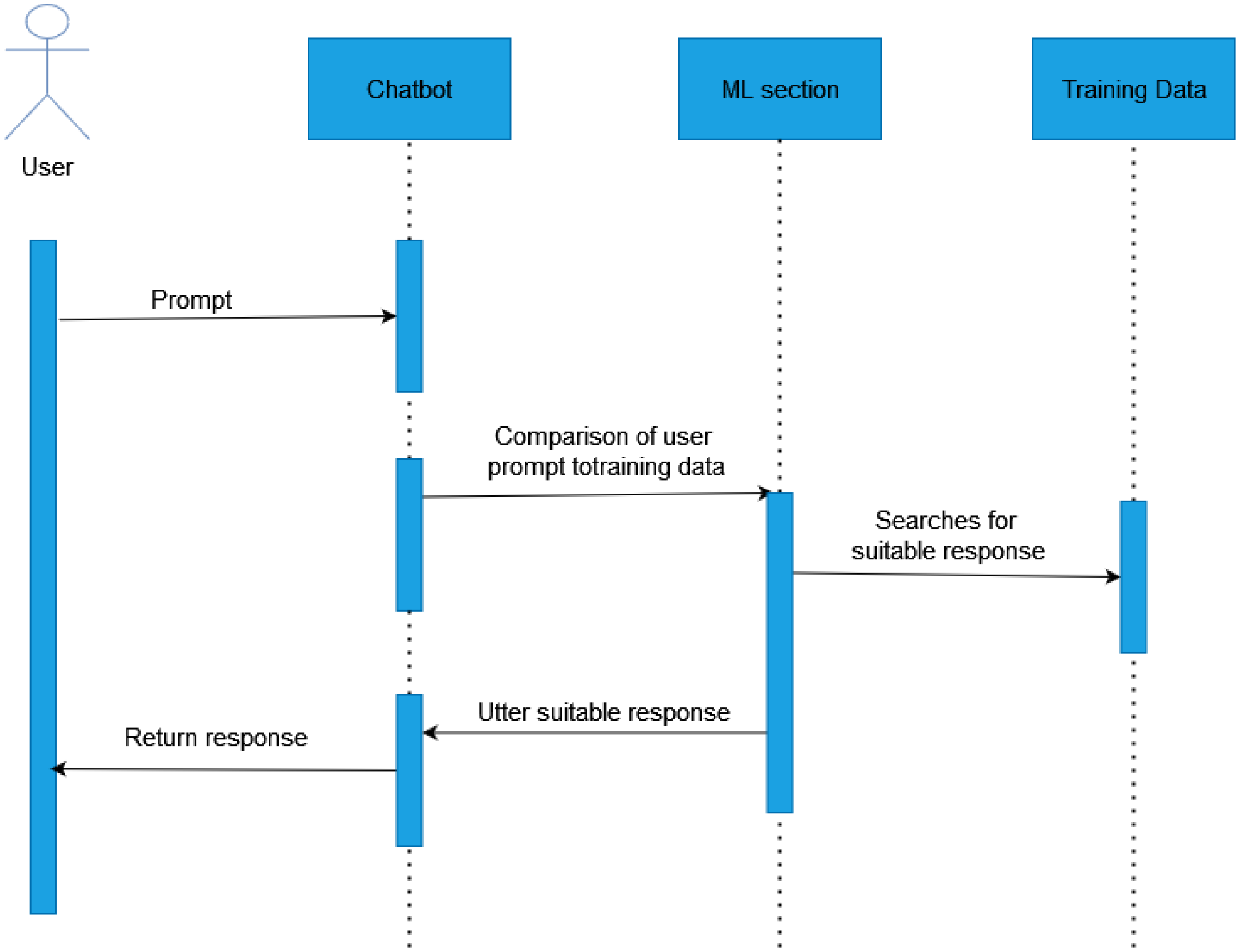


Figure 5.8: Sequence diagram of proposed system's chatbot

CHAPTER SIX

SYSTEM IMPLEMENTATION & TESTING

6.1 Implementation

To implement the proposed system, the following software were required:

- Rasa: The Rasa open-source framework for building chatbots using NLU and NLP was required to develop the chatbot component of the system. The installation process for Rasa involved creating a virtual environment, installing dependencies, and installing Rasa using pip. The specific version of Rasa used was 3.4.4
- Python: Python is a high-level programming language that is widely used for machine learning and web development. It was also required to run the Rasa chatbot, the machine learning models, and the Flask web application. Python can be downloaded and installed from the official Python website.
- TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It was required to develop the machine learning model component of the system. TensorFlow can be installed using pip.
- Scikit-learn: Scikit-learn is an open-source machine learning library for Python. It was required to preprocess

the dataset and train the machine learning models. Scikit-learn can be installed using pip.

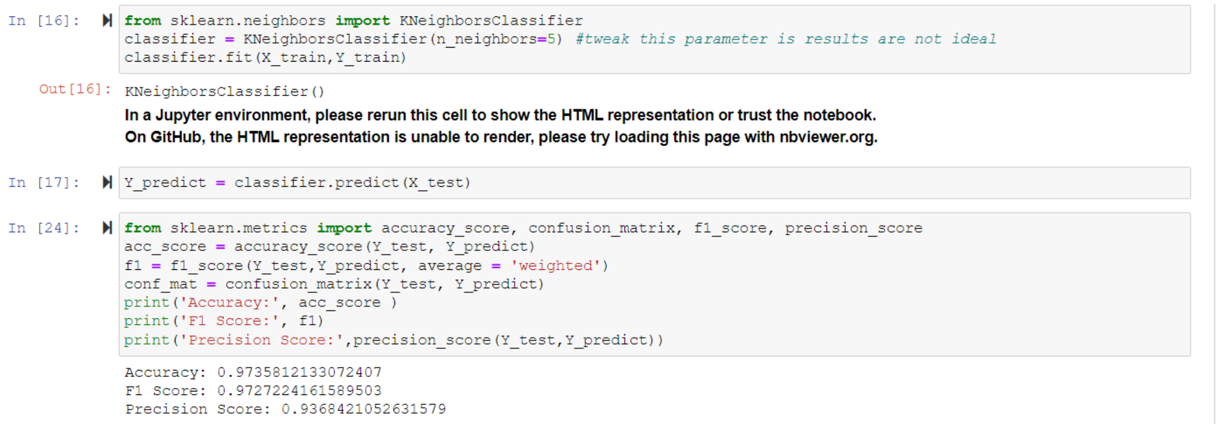
- Flask: Flask is a micro web framework written in Python. It was required to develop the website's backend and handle HTTP requests from the front-end. Flask can be installed using pip.
- JavaScript: JavaScript is a popular programming language for web development. It was required to develop the front-end of the website and enable user interactions with the chatbot. JavaScript does not require installation as it is a client-side language that runs in the user's web browser.
- HTML/CSS: HTML and CSS are markup languages used for creating web pages. They were required to develop the user interface of the website. HTML and CSS do not require installation as they are interpreted by web browsers.e downloaded and installed from the official Git website.
- IDE: Jupyter Notebook was used to build the KNN and SVM train models, and Visual Studio Code was used to develop the chatbot as well as the rest of the system. These IDEs provide features such as code highlighting, debugging, and version control integration, making it easier to develop complex systems.

6.2 Testing

Testing was a necessary step at every stage of the system's development.

6.2.1 Machine Learning Model Testing

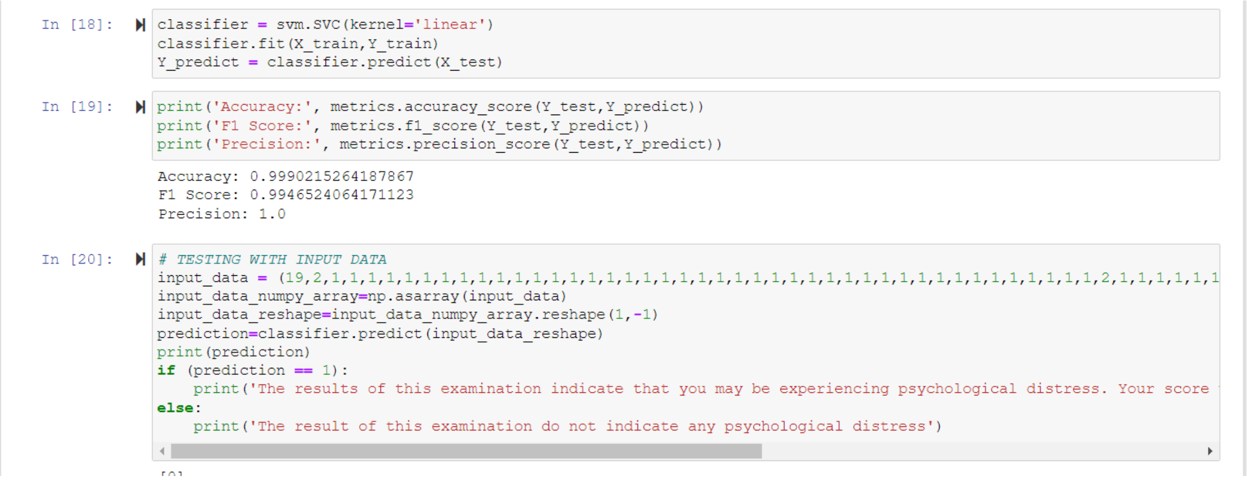
Two models were created using the acquired SCL-90 dataset. The first model was trained using KNN, after which its metrics were measured, namely: accuracy, f1_score and precision. The results were recorded as follows:



```
In [16]:  from sklearn.neighbors import KNeighborsClassifier
          classifier = KNeighborsClassifier(n_neighbors=5) #tweak this parameter is results are not ideal
          classifier.fit(X_train,Y_train)

Out[16]:  KNeighborsClassifier()
          In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
          On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [17]:  Y_predict = classifier.predict(X_test)

In [24]:  from sklearn.metrics import accuracy_score, confusion_matrix, f1_score, precision_score
          acc_score = accuracy_score(Y_test, Y_predict)
          f1 = f1_score(Y_test,Y_predict, average = 'weighted')
          conf_mat = confusion_matrix(Y_test, Y_predict)
          print('Accuracy:', acc_score )
          print('F1 Score:', f1)
          print('Precision Score:',precision_score(Y_test,Y_predict))

          Accuracy: 0.9735812133072407
          F1 Score: 0.9727224161589503
          Precision Score: 0.9368421052631579
```

Figure 6.0: Image depicting a KNN model being trained and its metrics (accuracy, f1 score, and precision) recorded

The second model was created using SVM, and similarly, its metrics were recorded as follows:



```
In [18]:  classifier = svm.SVC(kernel='linear')
          classifier.fit(X_train,Y_train)
          Y_predict = classifier.predict(X_test)

In [19]:  print('Accuracy:', metrics.accuracy_score(Y_test,Y_predict))
          print('F1 Score:', metrics.f1_score(Y_test,Y_predict))
          print('Precision:', metrics.precision_score(Y_test,Y_predict))

          Accuracy: 0.9990215264187867
          F1 Score: 0.9946524064171123
          Precision: 1.0

In [20]:  # TESTING WITH INPUT DATA
          input_data = (19,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1
          input_data_numpy_array=np.asarray(input_data)
          input_data_reshape=input_data_numpy_array.reshape(1,-1)
          prediction=classifier.predict(input_data_reshape)
          print(prediction)
          if (prediction == 1):
              print('The results of this examination indicate that you may be experiencing psychological distress. Your score
          else:
              print('The result of this examination do not indicate any psychological distress')
```

Figure 6.0.1: Image depicting an SVM model being and its metrics (accuracy, f1 score, and precision) recorded

The deciding factor for why the SVM model was chosen was the slightly higher performance, as both models were able to make accurate predictions when the same sets of input data were passed into them.

6.2.2 Chatbot Training and Testing

The system's chatbot was created using the Rasa framework, allowing access to its Rasa Core which houses sophisticated components such as the dialogue policies and nlu pipelines. A training data set comprising 127 stories, 6 rules, and over, including 31 intents and 17 responses was utilized in creating and training the model. The specific components (featurizers, tokenizers, classifiers, etc) used for training and testing the model were as follows:

- Language: This specifies the language that the chatbot will use for processing user input and generating responses. In this case, the language is set to English.

- Pipeline: This defines the sequence of components that will be used to process user input and extract entities. The pipeline includes the following components:

- SpacyNLP: This component uses the Spacy library to perform natural language processing on user input. It uses the "en_core_web_md" model to tokenize the input and extract linguistic features.

- SpacyTokenizer: This component tokenizes user input using the Spacy library.

- RegexFeaturizer: This component extracts features from user input using regular expressions.

- CRFEntityExtractor: This component uses conditional random fields (CRF) to extract entities from user input.

- EntitySynonymMapper: This component maps entity synonyms to their canonical form to improve entity recognition.

- CountVectorsFeaturizer: This component converts user input into a bag-of-words representation to improve intent classification.

- DIETClassifier: This component is a deep learning-based intent classifier that uses a combination of dense and sparse features to classify user input into intents. It is trained using the epochs parameter, which specifies the number of training iterations.

3. Policies: This defines the sequence of policies that will be used to handle user requests and generate responses. The policies include the following:

- MemoizationPolicy: This policy remembers previous conversation turns to improve response selection and reduce redundant actions.

- TEDPolicy: This policy uses a transformer-based neural network to predict the next action based on the current dialogue state and user input. It is trained using the max_history and epochs parameters, which specify the maximum number of previous turns to consider and the number of training iterations, respectively.

- RulePolicy: This policy allows the developer to define rules that map user input to actions, bypassing the intent classification and entity extraction components.

The results of the model trained were recorded as follows for the various components:

DIETClassifier:

- epochs: 100 - This is the number of training epochs (iterations) that were run for this model.

- t_loss=1.86 - This is the training loss, which is a measure of how well the model is fitting the training data. A lower loss value indicates better performance.

- i_acc=0.958 - This is the intent classification accuracy, which is the percentage of correctly classified intents in the test set. A higher accuracy value indicates better performance.

- e_f1=0.94 - This is the entity extraction F1 score, which is a measure of the model's precision and recall for identifying entities in the test set. An F1 score of 1.0 indicates perfect precision and recall, while a lower score indicates lower performance.

TEDPolicy:

- epochs: 100 - This is the number of training epochs (iterations) that were run for this model.

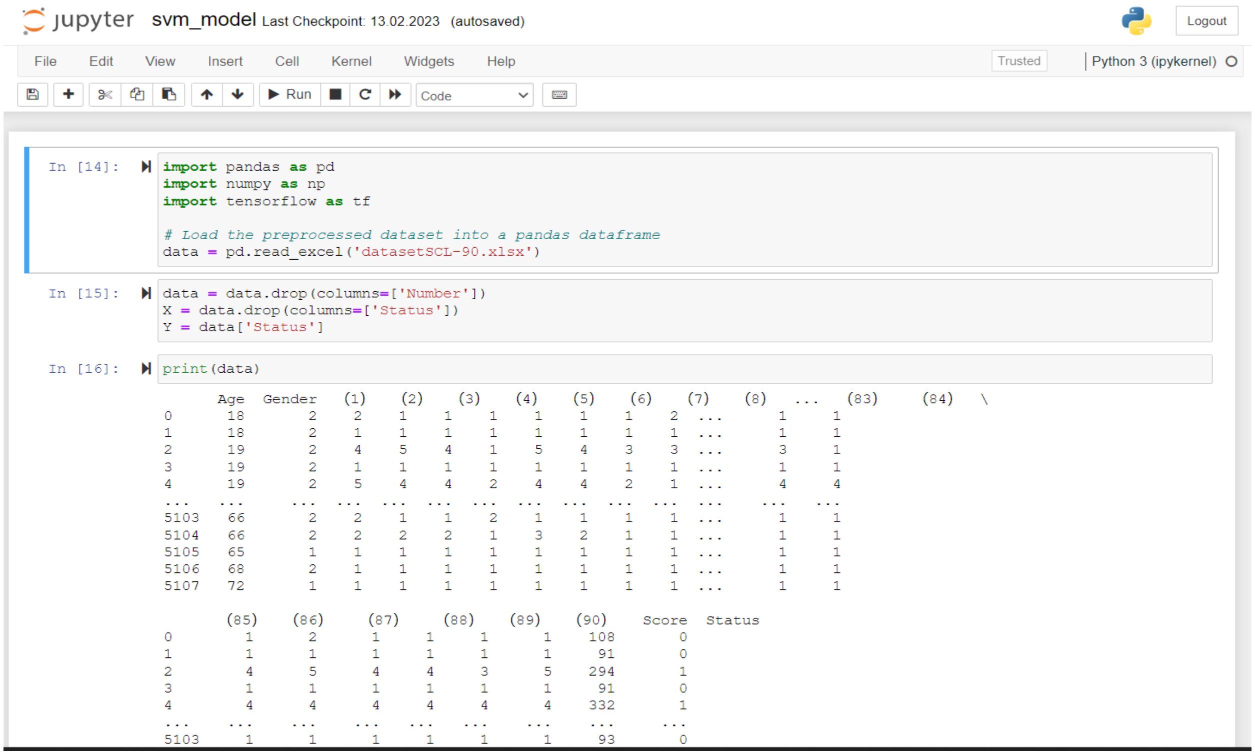
- t_loss=0.793 - This is the training loss, which is a measure of how well the model is fitting the training data. A lower loss value indicates better performance.

- loss=0.363 - This is the overall loss on the validation set, which is used to evaluate the model's performance during training. A lower loss value indicates better performance.

- acc=0.992 - This is the accuracy on the validation set, which is the percentage of correctly predicted actions. A higher accuracy value indicates better performance.

6.3 Sample Codes

6.3.1 SVM Model Creation and Training

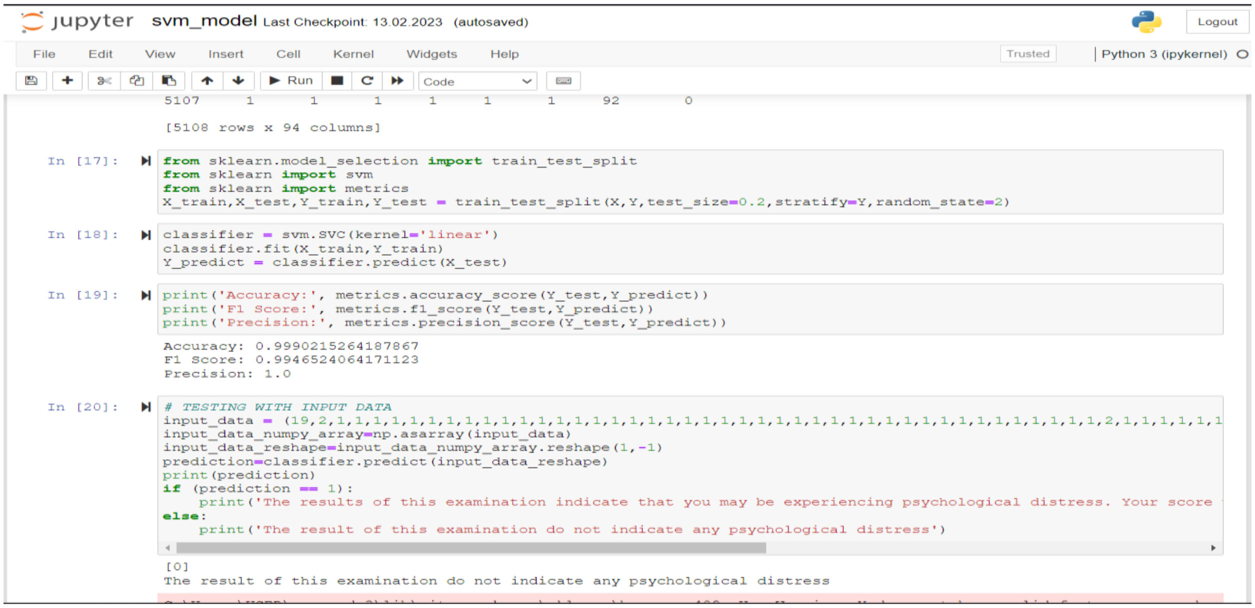


```
In [14]:  import pandas as pd
          import numpy as np
          import tensorflow as tf

          # Load the preprocessed dataset into a pandas dataframe
          data = pd.read_excel('datasetSCL-90.xlsx')
```

```
In [15]:  data = data.drop(columns=['Number'])
          X = data.drop(columns=['Status'])
          Y = data['Status']
```

```
In [16]:  print(data)
```

```
      Age  Gender  (1)  (2)  (3)  (4)  (5)  (6)  (7)  (8)  ...  (83)   (84)  \
0      18       2    2    1    1    1    1    1    1    2  ...     1     1
1      18       2    1    1    1    1    1    1    1    1  ...     1     1
2      19       2    4    5    4    1    5    4    3    3  ...     3     1
3      19       2    1    1    1    1    1    1    1    1  ...     1     1
4      19       2    5    4    4    2    4    4    2    1  ...     4     4
...   ...     ...  ...  ...  ...  ...  ...  ...  ...  ...  ...   ...   ...
5103   66       2    2    1    1    2    1    1    1    1  ...     1     1
5104   66       2    2    2    2    1    3    2    1    1  ...     1     1
5105   65       1    1    1    1    1    1    1    1    1  ...     1     1
5106   68       2    1    1    1    1    1    1    1    1  ...     1     1
5107   72       1    1    1    1    1    1    1    1    1  ...     1     1

      (85)   (86)   (87)   (88)   (89)   (90)   Score  Status
0        1      2      1      1      1      1     108       0
1        1      1      1      1      1      1      91       0
2        4      5      4      4      3      5     294       1
3        1      1      1      1      1      1      91       0
4        4      4      4      4      4      4     332       1
...    ...    ...    ...    ...    ...    ...     ...     ...
5103     1      1      1      1      1      1      93       0
```



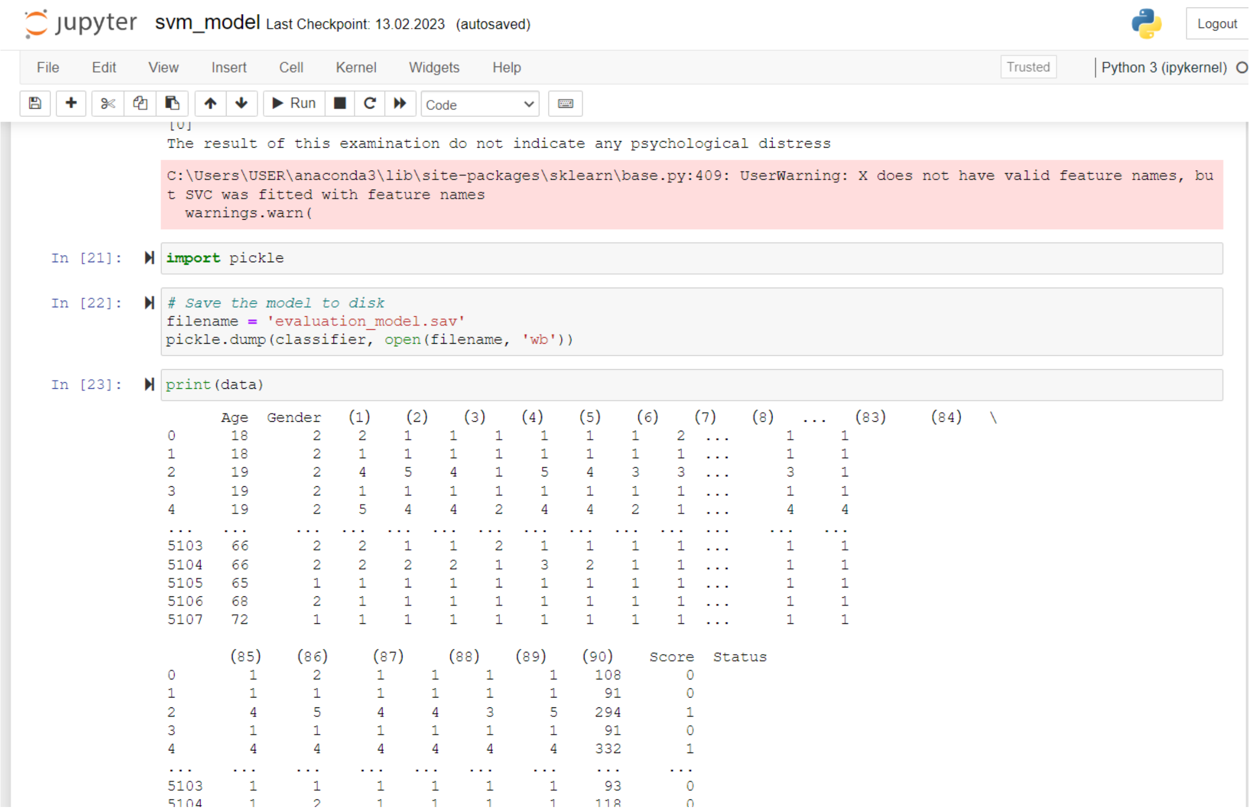
```
5107     1      1      1      1      1      1      92       0

[5108 rows x 94 columns]
```

```
In [17]:  from sklearn.model_selection import train_test_split
          from sklearn import svm
          from sklearn import metrics
          X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,stratify=Y,random_state=2)
```

```
In [18]:  classifier = svm.SVC(kernel='linear')
          classifier.fit(X_train,Y_train)
          Y_predict = classifier.predict(X_test)
```

```
In [19]:  print('Accuracy:', metrics.accuracy_score(Y_test,Y_predict))
          print('F1 Score:', metrics.f1_score(Y_test,Y_predict))
          print('Precision:', metrics.precision_score(Y_test,Y_predict))

          Accuracy: 0.9990215264187867
          F1 Score: 0.9946524064171123
          Precision: 1.0
```

```
In [20]:  # TESTING WITH INPUT DATA
          input_data = (19,2,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,1,1,1,1,1,1
          input_data_numpy_array=np.asarray(input_data)
          input_data_reshape=input_data_numpy_array.reshape(1,-1)
          prediction=classifier.predict(input_data_reshape)
          print(prediction)
          if (prediction == 1):
              print('The results of this examination indicate that you may be experiencing psychological distress. Your score
          else:
              print('The result of this examination do not indicate any psychological distress')

          [0]
          The result of this examination do not indicate any psychological distress
```


jupyter svm_model Last Checkpoint: 13.02.2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
[0]
The result of this examination do not indicate any psychological distress

C:\Users\USER\anaconda3\lib\site-packages\sklearn\base.py:409: UserWarning: X does not have valid feature names, bu
t SVC was fitted with feature names
  warnings.warn(
```

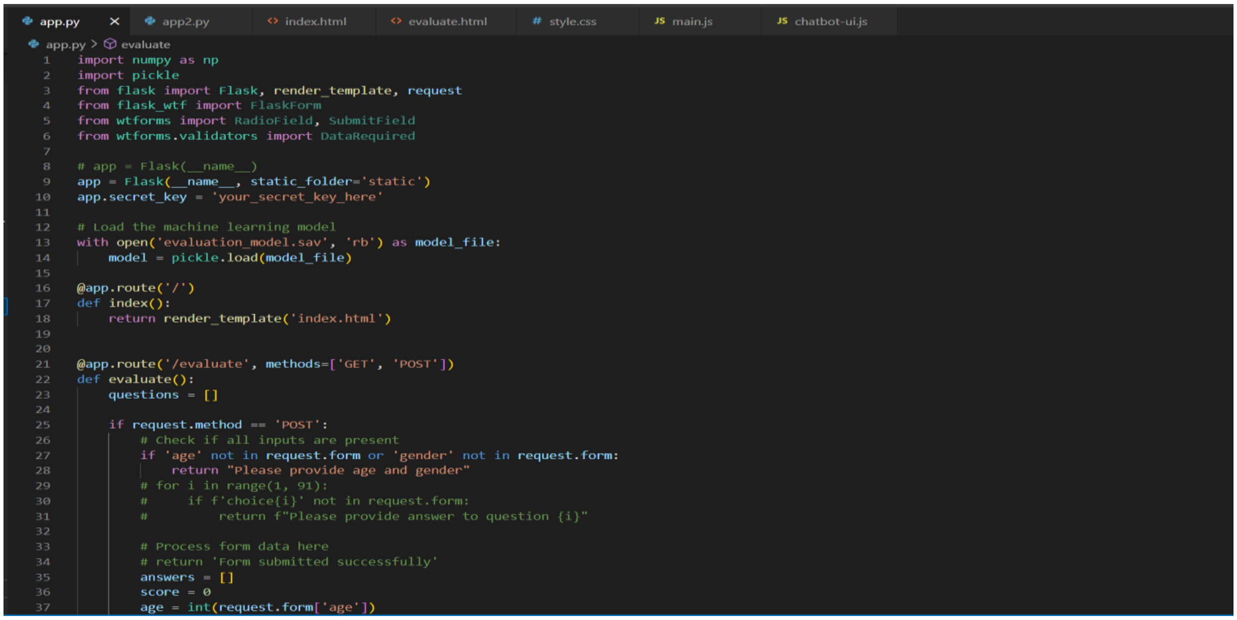
```
In [21]: import pickle
```

```
In [22]: # Save the model to disk
         filename = 'evaluation_model.sav'
         pickle.dump(classifier, open(filename, 'wb'))
```

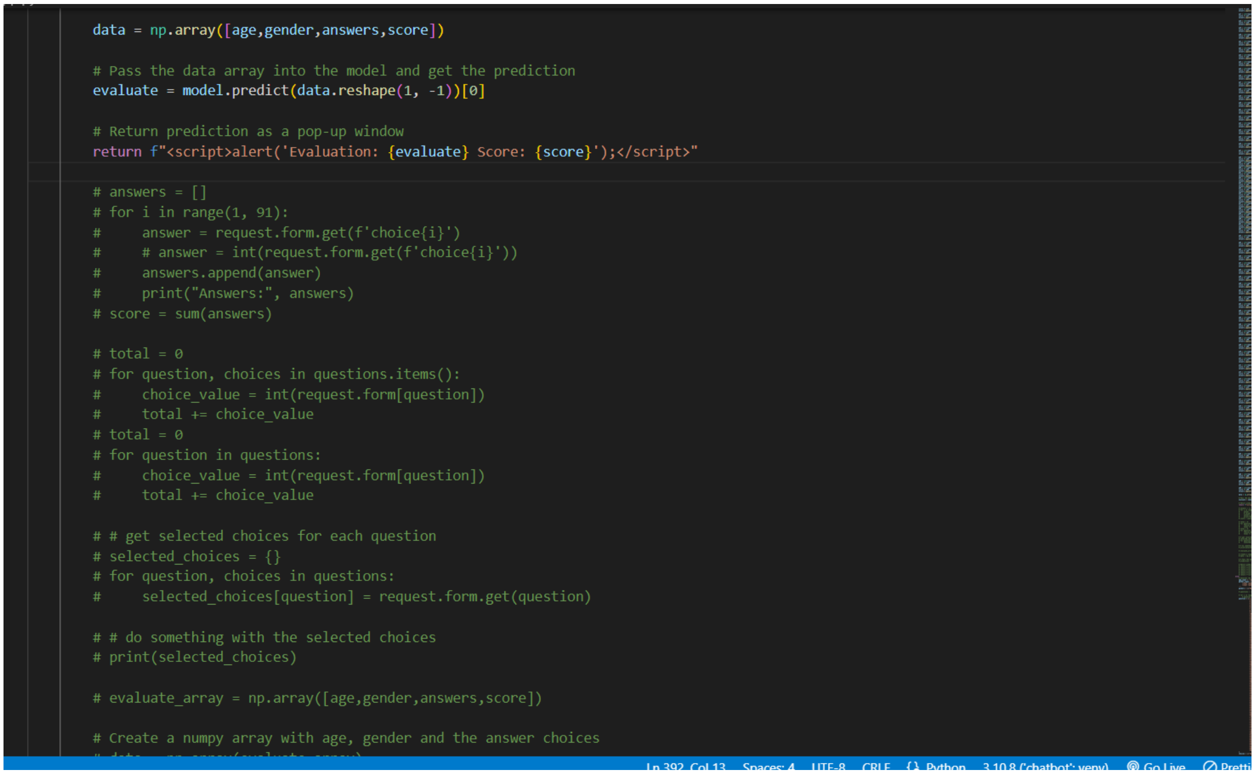
```
In [23]: print(data)
```

```
       Age  Gender  (1)   (2)   (3)   (4)   (5)   (6)   (7)   (8)  ...  (83)    (84)  \
0       18       2    2    1    1    1    1    1    1    2  ...     1     1
1       18       2    1    1    1    1    1    1    1    1  ...     1     1
2       19       2    4    5    4    1    5    4    3    3  ...     3     1
3       19       2    1    1    1    1    1    1    1    1  ...     1     1
4       19       2    5    4    4    2    4    4    2    1  ...     4     4
...    ...     ...  ...  ...  ...  ...  ...  ...  ...  ...  ...   ...   ...
5103    66       2    2    1    1    2    1    1    1    1  ...     1     1
5104    66       2    2    2    2    1    3    2    1    1  ...     1     1
5105    65       1    1    1    1    1    1    1    1    1  ...     1     1
5106    68       2    1    1    1    1    1    1    1    1  ...     1     1
5107    72       1    1    1    1    1    1    1    1    1  ...     1     1

       (85)   (86)    (87)    (88)   (89)   (90)   Score  Status
0        1      2       1      1      1      1    108       0
1        1      1       1      1      1      1     91       0
2        4      5       4      4      3      5    294       1
3        1      1       1      1      1      1     91       0
4        4      4       4      4      4      4    332       1
...    ...    ...     ...    ...    ...    ...    ...     ...
5103     1      1       1      1      1      1     93       0
5104     1      2       1      1      1      1    118       0
```

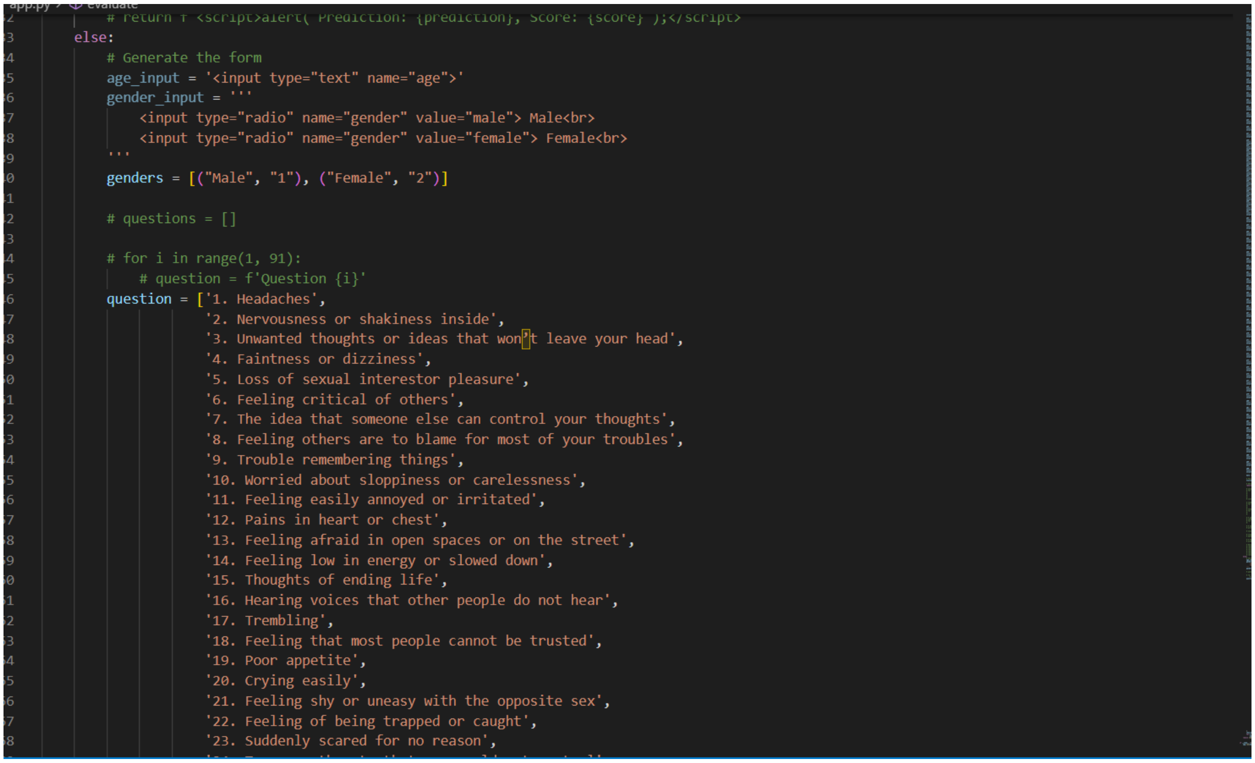
6.3.2 Flask Application



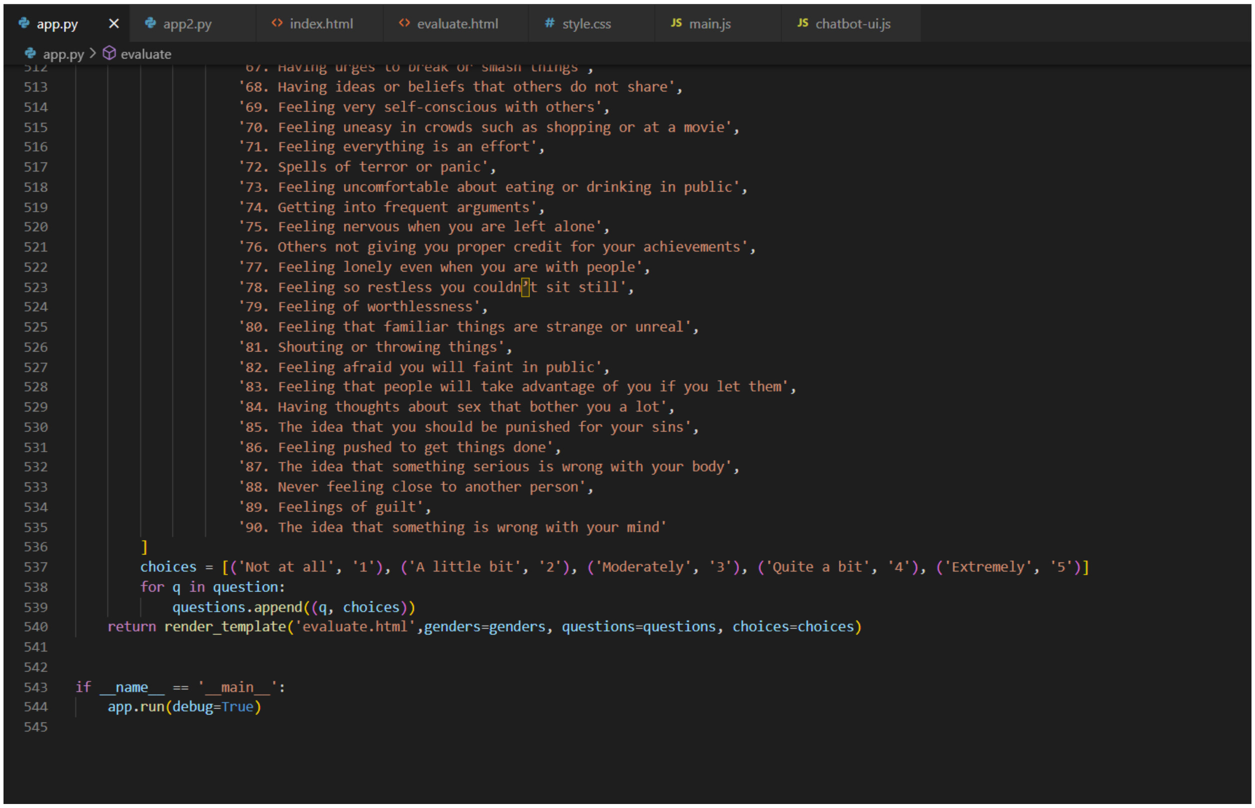
```
import numpy as np
import pickle
from flask import Flask, render_template, request
from flask_wtf import FlaskForm
from wtforms import RadioField, SubmitField
from wtforms.validators import DataRequired

# app = Flask(__name__)
app = Flask(__name__, static_folder='static')
app.secret_key = 'your_secret_key_here'

# Load the machine learning model
with open('evaluation_model.sav', 'rb') as model_file:
    model = pickle.load(model_file)

@app.route('/')
def index():
    return render_template('index.html')


@app.route('/evaluate', methods=['GET', 'POST'])
def evaluate():
    questions = []

    if request.method == 'POST':
        # Check if all inputs are present
        if 'age' not in request.form or 'gender' not in request.form:
            return "Please provide age and gender"
        # for i in range(1, 91):
        #     if f'choice{i}' not in request.form:
        #         return f"Please provide answer to question {i}"

        # Process form data here
        # return 'Form submitted successfully'
        answers = []
        score = 0
        age = int(request.form['age'])
```



```
        data = np.array([age,gender,answers,score])

        # Pass the data array into the model and get the prediction
        evaluate = model.predict(data.reshape(1, -1))[0]

        # Return prediction as a pop-up window
        return f"<script>alert('Evaluation: {evaluate} Score: {score}');</script>"

        # answers = []
        # for i in range(1, 91):
        #     answer = request.form.get(f'choice{i}')
        #     # answer = int(request.form.get(f'choice{i}'))
        #     answers.append(answer)
        #     print("Answers:", answers)
        # score = sum(answers)

        # total = 0
        # for question, choices in questions.items():
        #     choice_value = int(request.form[question])
        #     total += choice_value
        # total = 0
        # for question in questions:
        #     choice_value = int(request.form[question])
        #     total += choice_value

        # # get selected choices for each question
        # selected_choices = {}
        # for question, choices in questions:
        #     selected_choices[question] = request.form.get(question)

        # # do something with the selected choices
        # print(selected_choices)

        # evaluate_array = np.array([age,gender,answers,score])

        # Create a numpy array with age, gender and the answer choices
        # data = np.array(evaluate_array)
```



```
        # return f"<script>alert('Prediction: {prediction}, Score: {score}');</script>"
    else:
        # Generate the form
        age_input = '<input type="text" name="age">'
        gender_input = '''
            <input type="radio" name="gender" value="male"> Male<br>
            <input type="radio" name="gender" value="female"> Female<br>
        '''
        genders = [("Male", "1"), ("Female", "2")]

        # questions = []

        # for i in range(1, 91):
            # question = f'Question {i}'
        question = ['1. Headaches',
                    '2. Nervousness or shakiness inside',
                    '3. Unwanted thoughts or ideas that won’t leave your head',
                    '4. Faintness or dizziness',
                    '5. Loss of sexual interestor pleasure',
                    '6. Feeling critical of others',
                    '7. The idea that someone else can control your thoughts',
                    '8. Feeling others are to blame for most of your troubles',
                    '9. Trouble remembering things',
                    '10. Worried about sloppiness or carelessness',
                    '11. Feeling easily annoyed or irritated',
                    '12. Pains in heart or chest',
                    '13. Feeling afraid in open spaces or on the street',
                    '14. Feeling low in energy or slowed down',
                    '15. Thoughts of ending life',
                    '16. Hearing voices that other people do not hear',
                    '17. Trembling',
                    '18. Feeling that most people cannot be trusted',
                    '19. Poor appetite',
                    '20. Crying easily',
                    '21. Feeling shy or uneasy with the opposite sex',
                    '22. Feeling of being trapped or caught',
                    '23. Suddenly scared for no reason',
```

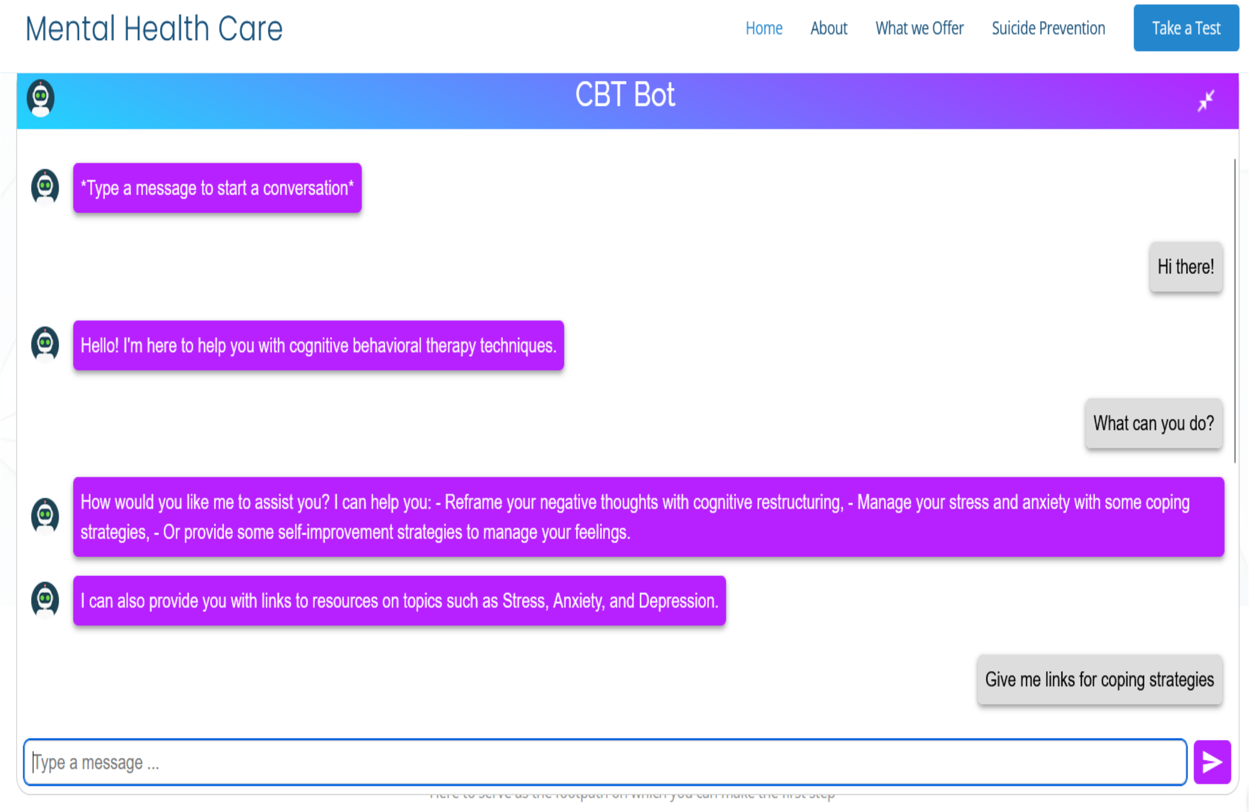


```
                    '67. Having urges to break or smash things',
                    '68. Having ideas or beliefs that others do not share',
                    '69. Feeling very self-conscious with others',
                    '70. Feeling uneasy in crowds such as shopping or at a movie',
                    '71. Feeling everything is an effort',
                    '72. Spells of terror or panic',
                    '73. Feeling uncomfortable about eating or drinking in public',
                    '74. Getting into frequent arguments',
                    '75. Feeling nervous when you are left alone',
                    '76. Others not giving you proper credit for your achievements',
                    '77. Feeling lonely even when you are with people',
                    '78. Feeling so restless you couldn’t sit still',
                    '79. Feeling of worthlessness',
                    '80. Feeling that familiar things are strange or unreal',
                    '81. Shouting or throwing things',
                    '82. Feeling afraid you will faint in public',
                    '83. Feeling that people will take advantage of you if you let them',
                    '84. Having thoughts about sex that bother you a lot',
                    '85. The idea that you should be punished for your sins',
                    '86. Feeling pushed to get things done',
                    '87. The idea that something serious is wrong with your body',
                    '88. Never feeling close to another person',
                    '89. Feelings of guilt',
                    '90. The idea that something is wrong with your mind'
        ]
        choices = [('Not at all', '1'), ('A little bit', '2'), ('Moderately', '3'), ('Quite a bit', '4'), ('Extremely', '5')]
        for q in question:
            questions.append((q, choices))
    return render_template('evaluate.html',genders=genders, questions=questions, choices=choices)


if __name__ == '__main__':
    app.run(debug=True)

```

CHAPTER SEVEN

RESULTS OF TRAINING

The selected components and algorithms chosen to develop the chatbot proved quite efficient, with accuracy levels of 95.8% and 99.2% for the deep learning components used (DIETClassifier and TEDPolicy, respectively). As a result, the intents of new untrained data passed into the model were correctly predicted, and the appropriate responses were issued. This is depicted in Fig 7 and Fig 8 below which shows the conversation results of an instance where new untrained data is introduced.



Type a message to start a conversation

Hi there!

Hello! I'm here to help you with cognitive behavioral therapy techniques.

What can you do?

How would you like me to assist you? I can help you: - Reframe your negative thoughts with cognitive restructuring, - Manage your stress and anxiety with some coping strategies, - Or provide some self-improvement strategies to manage your feelings.

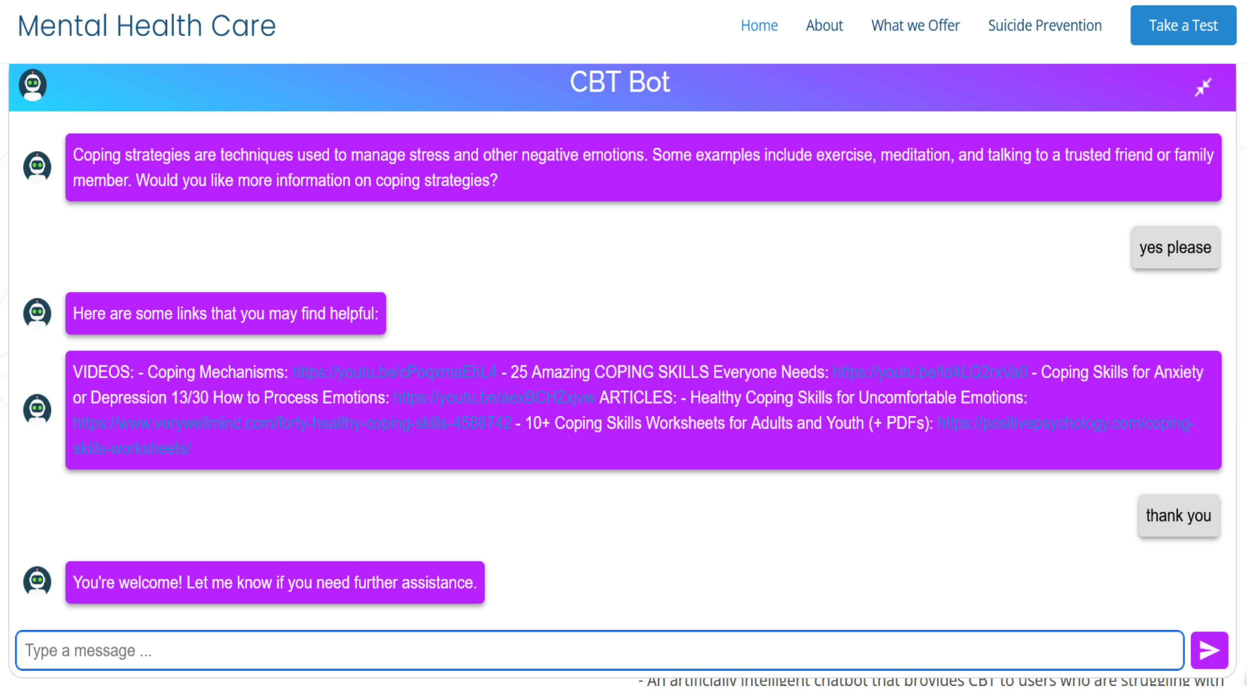
I can also provide you with links to resources on topics such as Stress, Anxiety, and Depression.

Give me links for coping strategies

Type a message ...

Here to serve as the footpath on which you can make the first step

Fig 7. An example of a conversation with CBT Bot with new untrained data "Give me links for coping strategies"



Coping strategies are techniques used to manage stress and other negative emotions. Some examples include exercise, meditation, and talking to a trusted friend or family member. Would you like more information on coping strategies?

yes please

Here are some links that you may find helpful:

VIDEOS: - Coping Mechanisms: https://youtu.be/cPoqxmaEhL4 - 25 Amazing COPING SKILLS Everyone Needs: https://youtu.be/td4LQ2rxVa0 - Coping Skills for Anxiety or Depression 13/30 How to Process Emotions: https://youtu.be/aexBCHZxjvw ARTICLES: - Healthy Coping Skills for Uncomfortable Emotions: https://www.verywellmind.com/forty-healthy-coping-skills-4586742 - 10+ Coping Skills Worksheets for Adults and Youth (+ PDFs): https://positivepsychology.com/coping-skills-worksheets/

thank you

You're welcome! Let me know if you need further assistance.

- An artificially intelligent chatbot that provides CBT to users who are struggling with

Fig 8. An example of CBT Bot accurately issuing the correct response to "Give me links for coping strategies"

The user prompts the chatbot with "Give me links for coping strategies" which is not listed in any of the created intents in the NLU file for the chatbot. However, it was able to accurately predict the new data's intent, issuing the correct response with no issues.

This is indicative of the high performance of the chatbot.

Similarly, the machine learning model trained with SVM, with an accuracy of 99%, for the purpose of mental health evaluation accurately predicts the user's mental health status with the correct input values.

CHAPTER EIGHT

CONCLUSION AND RECOMMENDATION

In conclusion, the project aimed at providing preliminary mental healthcare via a chatbot that offers CBT treatment and a machine learning model that evaluates the user's mental health using SCL-90. The system was designed using Rasa chatbot, Tensorflow, and sklearn machine learning models. The chatbot was trained to recognize 31 intents and 4 entities, namely, resources, name, symptoms, and technique, to provide personalized care to each user. The machine learning model was trained on a dataset of numerous results of the SCL-90, including the age, gender, and choices selected for each of the symptoms in the questionnaire. Two models were trained using KNN and SVM respectively, with the SVM model performing better in terms of accuracy, precision, and F1 score.

The system's architecture was designed to integrate the chatbot and the machine learning model into a website using Flask and Javascript. The chatbot used SpacyNLP, SpacyTokenizer, RegexFeaturizer, CRFEntityExtractor, EntitySynonymMapper, CountVectorsFeaturizer, and DIETClassifier to process the user's input and provide appropriate responses. The machine learning models used TensorFlow and sklearn to classify the user's mental health status based on their SCL-90 results.

Recommendations:

The proposed system can be further improved and enhanced in several ways. Firstly, the system's performance can be improved by training the machine learning model on a larger and more diverse dataset. This would improve the accuracy of the system's diagnosis and increase the effectiveness of the CBT treatment offered. Additionally, the chatbot's conversational abilities can be improved by training it on a larger dataset of user inputs and responses. This would enable the chatbot to recognize and respond to a broader range of user inputs, improving the user experience.

Secondly, the system's privacy and security can be improved by implementing advanced security measures such as end-to-end encryption, secure user authentication, and access control. This would ensure that users' data is protected and secure, reducing the risk of data breaches and unauthorized access to sensitive information.

Finally, the system's accessibility can be improved by making it available in multiple languages and integrating it with assistive technologies such as screen readers and voice recognition software. This would ensure that the system is accessible to a broader range of users, including those with disabilities or limited English proficiency.

In summary, the proposed system has the potential to provide accessible and effective mental healthcare to a broad range of users. By implementing the recommendations outlined above, the system can be further improved and enhanced, providing more accurate diagnoses, improving the user experience, and ensuring user privacy and security.

REFERENCES

Felman, A. (2022, July 29). *What is mental health?* Retrieved October 7, 2022 from https://www.medicalnewstoday.com/articles/154543

Walker, E. R., PhD. (2015, April 1). Mortality in Mental Disorders and Global Disease Burden Implications: A Systematic Review and

Meta-analysis. Psychiatry and Behavioral Health | JAMA Psychiatry | JAMA Network. Retrieved October 7, 2022, from https://jamanetwork.com/journals/jamapsychiatry/fullarticle/2110027

Mohr, D. C., Zhang, M., & Schueller, S. M. (2017). Personal Sensing: Understanding Mental Health Using Ubiquitous Sensors and Machine Learning. Annual review of clinical psychology, 13, 23–47. https://doi.org/10.1146/annurev-clinpsy-032816-044949

Shatte, A., Hutchinson, D., & Teague, S. (2019). Machine learning in mental health: A scoping review of methods and applications. Psychological Medicine, 49(9), 1426-1448. doi:10.1017/S0033291719000151

Cho, G., Yim, J., Choi, Y., Ko, J., & Lee, S. H. (2019). Review of Machine Learning Algorithms for Diagnosing Mental Illness. Psychiatry investigation, 16(4), 262–269. https://doi.org/10.30773/pi.2018.12.21.2