Chatbot for mental health

by

Ali Mohamed Magdi Ali ID:150159

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Bachelor of computer science (SE programme)
in the
Faculty of Computer Science
of the
October University for Modern Sciences and Arts (MSA), EGYPT

Graduation Projects advisor:
Dr.Hesham Mansour

(Feb 2022)

Abstract

One of the most challenges that each work place face are burnout, anxiety ,depression and other mental health problems , these issues lead to reduced productivity, each organization should try to tackle these problems with the help of mental health professionals, by properly extracting the features that identifies mental health problems ,and keep a good record of employees mental health history, collecting the mental health data with a survey is a very cost effective solution yet has very low engagement rate,in this paper we built a chatbot website , that aims to tackle this problem because unlike surveys chatbots are in- teractive, and they try to simulate human conversation to some extent which will has been proven to increase the engagement between potential patients , and the chatbot, which submits the patients responses and predicted mental health case to a professional who can follow up with their assessment,using state of the art machine learning algorithms , and easy to use web based interface, yet we have a very challenging aspect in this project , to build a chatbot is to basically try to solve an AI-complete problem , which means there is no optimal way to build a chat bot so far, so we are going to choose the best approaches that are being used currently to correctly answer all patients mental health queries and also will focus on creating a dynamic interface , that assist the doctor's who intend to use this system, and making sure doctors will have to work with this system along with AI to improve the accuracy of the diagnose that people who suffer from mental health have, because by definition AI complete problems require human computation along with machine learning to build a reliable and robust system.

Acknowledgments

I am heartily thankful to Professor Hesham Mansour my project supervisor , I wouldn't have my project today completed if it wasn't for his great help , he was always encoring me and pushing my limits and I am so grateful for his cool ideas , also sincerely thankful to Dr.Ayman Ezzat as he assisted me from the early stages of the project and helped me figure out what I was missing to build a project that adds value, finally thanks to Dr. Wael Gomaa as he also gave me good recommendations to improve my models, I learnt during this project more than I have imagined thanks to having professors with great minds who always encourage their students to evolve their skills in many fields and reach a documented data driven project that is backed by appendices and results .

Contents

A Collected materials from DIACWOZ website 16

Bibliography

DAIC-WOZ Database Description

This database is part of a larger corpus, the Distress Analysis Interview Corpus (DAIC) (Gratch et al.,2014), that contains clinical interviews designed to support the diagnosis of psychological distress conditions such as anxiety, depression, and post-traumatic stress disorder. These interviews were collected as part of a larger effort to create a computer agent that interviews people and identifies verbal and nonverbal indicators of mental illness (DeVault et al., 2014). Data collected include audio and video recordings and extensive questionnaire responses; this part of the corpus includes the Wizard-of-Oz interviews, conducted by an animated virtual interviewer called Ellie, controlled by a human interviewer in another room. Data has been transcribed and annotated for a variety of verbal and non-verbal features.

This share includes 189 sessions of interactions ranging between 7-33min (with an average of 16min). Each session includes transcript of the interaction, participant audio files, and facial features. For more details please refer to the

17

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction :

Knowledge datasets are found are in README.MD in github

"h ttps://github.com/AliMohamedMagdi/Mental-health-app.git"

Recently we have become aware of the important role of mental health ,the united nations included mental health is the sustainable development goals as a part of the third goal good health and well being, according to the world health organization "one of the leading cause of disability is depression , also suicide is the second leading cause of death for people who are between 15-29 year old, the alarming fact is people facing mental health illness can die early two decades to an illness that can be prevented if discovered early and dealt with properly".

so we aim to create a chatbot that aid users to correctly identify their mental health condition , recording the user responses along with important inferences about the user to aid personalize their conversation such as the user responsive time , and their most common words used during the conversation, importantly I will need to utilize the chatbot Natural language processing capabilities such as mining the patient text and dealing with words who have the same meaning, also making sure to build a system that is dynamic of nature allowing the field expert to expand the chatbot domain of knowledge , by adding new responses and also specifying to the bot important parts of text to be extracted such as medical terms , the intended users of the system , doctor's who run mental health hospitals, or clinics, and the patients who are treated by those doctors, to have their records kept and their symptoms recorded effectively, the main problem we are going to tackle is mining the patient text for important key words, also understanding the user input to answer mental health frequently asked questions in English, making sure that the bot is also able to handle

Arabic questions, by using state of the art natural language processing approaches such as word net , and name entity recognition , also the approach for the website is using model view controller architecture pattern which will aid making an easy to use chatbot that can be easily updated, the natural language processing field has progressed immensely and many useful tools can aid creating this project, along with the great progress in the machine learning field , assuming that the data collected is reliable , the system can classify patients based on their symptoms , the main outcome of this project is the bot is able to answer mental health frequently asked questions In English and Arabic , also able to converse with the user saving their critical data , and offering a diagnosis for depression based on scientific approaches, also the developed system successfully was altered by the admin to learn new words and responses .

Chapter 2

Background

2.1 General Guidelines

dealing with the challenge of making computer understand human language is an AI complete problem , solving it will be solving the Alan-Turing test , which represent a human interrogator speaking with a computer , if the human couldn't tell if he is talking to a computer ,then we pass the Turing test and solve an AI complete problem. that was the reason behind choosing the chatbot will be a knowledge based agent , have a certain knowledge database that can be expanded easily in the field of mental health with the help of mental health professionals, instead of making a general purpose AI that can respond to any question which is practically impossible.

mainly the contribution that I want to make is to make the chat bot knowledge ex- pandable because science evolves every day , and as a result to this evolution , new diseases emerge, that will be the first feature I will focus on , secondly to improve the process of creating the chatbot by offering an API that extracts intents and entities effectively, thirdly I want to feed the chatbot with conversation with depressed people and based on certain features such how fast they respond , the context of their story , and their tone of voice I will determine using state of the art machine learning algorithm, to determine the degree of truth that represent if this person is suffering from depression or not ,the knowledge of the chatbot will be verified from the professional's who assist me in this project, and from world wide trust data set sources.

existing solutions such as MoodKit offers professional psychology programs in the hands of the normal user of each day , this tool is brilliant because it offers mood tracking and surveys that ask the user couple of question and based on it it gives personalized ad vices to improve the mental health of the user, also Woebot if another chatbot solution similar to our project, offering the same tools the MoodKit offers, but unlike those two solution we aim for more free conversation we aim to make a chatbot like Jarvis in and Siri in terms of Natural language processing , and try to make my chatbot respond to non scripted responses , and generalize more , which is the advantage of my project , also we aim to make our chatbot agent mobile agent so it can be integrated to any website and responsive so that it can be used on any smart device , special thanks to Dr.Ahmed Essam El-Mahdy. Psychiatrist Specialized in Pediatric Psychiatry, Family Counseling, Addiction, Toxicology, Adolescent Psychiatry, Pediatric Neurology, Adult Neurology and Adult Psychiatry, as he offered good insights on my domain knowledge and also helped me testing the system , so It can be evolved by a non programming user , also we tested the model prediction result on 2 live depressed people and 5 healthy subjects and it had only one wrong prediction. the system was able to identify the depression form the patient tone of voice, and the methodology will be explained in further chapter.

Chapter 3

Specification - (SRS)

3.1 General Guide Lines

New-awareness Psychological Center (the Center) is a multidisciplinary behavioral health care practice that offers mental health and substance abuse their mission is to maintain a society where every one mental health is addressed , every single human is exposed to stress , sadness ,grief , and addiction , thus it's our job to provide a solution for these problems , and to integrate medicine with technology by building an intelligent chatbot mobile agent that can be used by a doctor , or a nurse who don't have much technical background, and have the ability not only to keep patient record and update them regularly but to offer the patient a conversation solution , and identify from their features the degree of mental illness they have, and suggest treatment as per the doctor who is using the chatbot.

The user interface is a responsive web application that can be installed on any mobile device or a computer, or even a Television , it was built using PHP and JavaScript with the help of a python server.

the python server contains a neural network and word similarity novel algorithm that calculates edit distance between each word and return the average in the whole sentence, if outcome is below a variable threshold the chatbot utters the mapped correct answer , else the chatbot ask the user if the means the second closest answer based on the similarity and the confidence of the neural network model, regrading the neural network it's a 3 layer neural network input layer is the number of intents , hidden layer, and output layer with the number of possible answer, Sigmund is the activation function for the neural network , and the training was optimised using SGD Stochastic gradient descent, for the best results the training was done in 200 Epochs with batch size=20 . at the decision making part , to determine if a patient is depressed or not , I used an official data set from kaggle, and made my prediction using 5 machine learning algorithms such as decision tree and random forest , boosting , and found out the XGB boost has the highest accuracy with 91 so I implemented the final prediction function using XGB boost A specification should tell the reader what the software system is required to do. UMLS+ OVERVIEW Describing what a software system does (specification) and how it does so (design) effectively usually means describing it from more than one viewpoint. Each viewpoint will convey some information about the system that other viewpoints omit.

- The user interface is an interactive
- 

Figure 1 interface

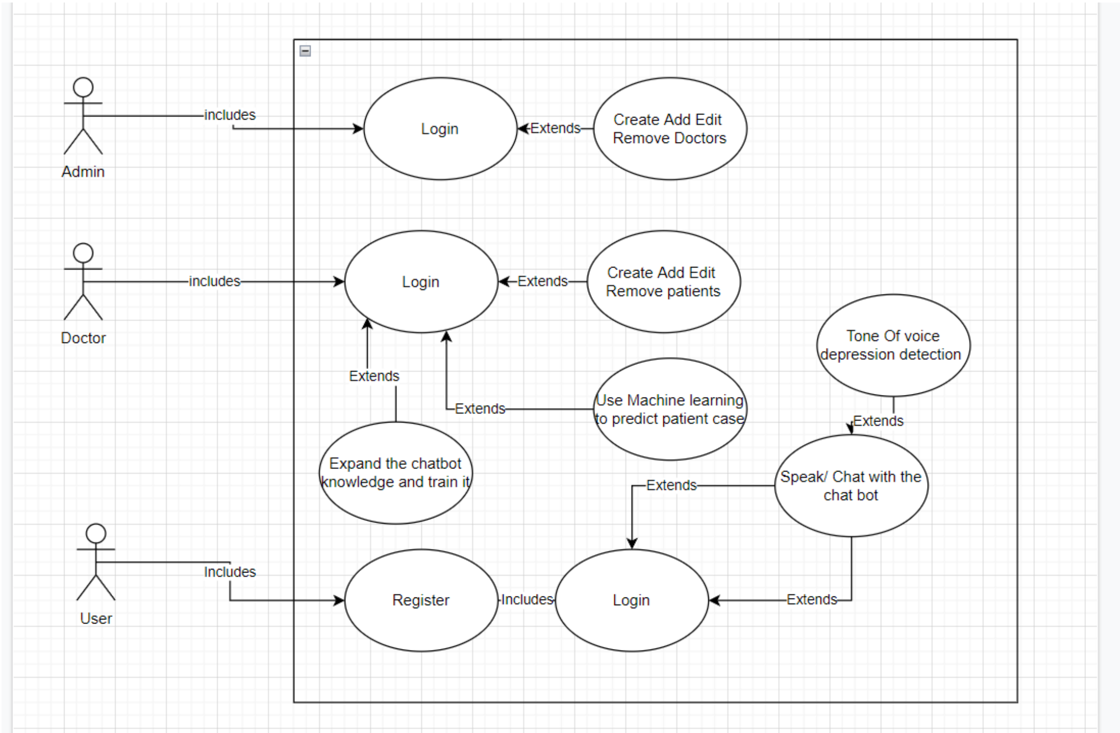
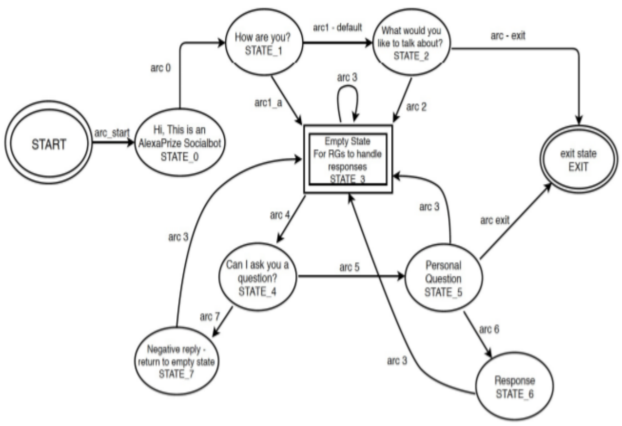


Figure 2 Usecase diagram



•

• Figure 3 State Machine

=

Chapter 4

Design

4.1 Design Guidelines

- In this section we will explain the used materials int this project which varied between data & tools & environments.

I used two different datasets both downloaded from kaggle

first one is from a survey was done to measure frequency of mental health disorders in the work place using the main predictors of mental health in the united states and it's format is comma separated values, "CSV file" ,containing 1263 patient response as a text, for the preprocessing I used a label encoder to make sure each column will be encoded into a number , when it comes to sub-sampling the data set is fairly small and no sub sampling needed

First data set link: https://www.kaggle.com/osmi/mental-health-in-tech-survey

Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere
37	Female	United State	IL	NA	No	Yes	Often
44	M	United State	IN	NA	No	No	Rarely
32	Male	Canada	NA	NA	No	No	Rarely
31	Male	United King	NA	NA	Yes	Yes	Often
31	Male	United State	TX	NA	No	No	Never
33	Male	United State	TN	NA	Yes	No	Sometimes
35	Female	United State	MI	NA	Yes	Yes	Sometimes
39	M	Canada	NA	NA	No	No	Never

Figure 4: Dateset samples "there is more 18 columns in this data set

Second data set is for the sentiment analysis also collected from kaggle , a collection of a many sentences with the corresponding emotion, it's format is text files and size is 1.97MB divided into three sections , training , testing , and validating, each section has about 2000 sentence with their emotions labeled , at the preprocessing I had to perform three stages the first is to remove the stop words" words that is commonly being used at the English language that don't offer any information , such as "the ,is ,and" , and the second stage is removing contraction " combing two words with apostrophe " and the last stage was word lemmatisation to be able to tell that the word better has good as it's "lemma".

Second data set link : https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp

```
i feel low energy i m just thirsty;sadness
i have immense sympathy with the general point but as a possible proto wr
i do not feel reassured anxiety is on each side;joy
i didnt really feel that embarrassed;sadness
i feel pretty pathetic most of the time;sadness
```

Figure 5: Dateset samples

4.2 Tools

- Software
- Library

  - Java Abydyos: is an open source machine learning framework that enables developer to find similarity between sentences
  - Python : a programming language that is considered the perfect fit for machine learning , because it's simple , consistent, and have access to great librarians

Libraries:

  - Pandas is a python library for data processing , manipulation and analysis,
  - Sklearn is a very effective tool in predictive data analysis

3.1.3 Environment

  - Local CPU, Intel i7 processor with 4 cores GPU run much slower than a CPU core
  - Google cloud storage : allows many user to collaborate and access the project at the same time because chat bots require constant updates

3.2 Methods

- In this section we will mention & explain our solution methodology + used approach(algorithms)

3.2.1 System architecture Overview

After collecting the user input we load the first data set shown in figure 5 , at the step of preprocessing we clean the data from noisy input , and encode the categorical data into numbers, make comparison between the variable to be able to decide which attribute to be used based on their importance, the next step is scaling the data , age has too many possible values , scaling age will make sure to for the algorithms to make correct decisions , now we are ready to build a model , several machine learning algorithm will be used , and explained below and the model that offers the highest accuracy will be selected,

Used algorithms for this data set are:

for the attribute selection I chose to build a forest to decide which attributes are more important resulting

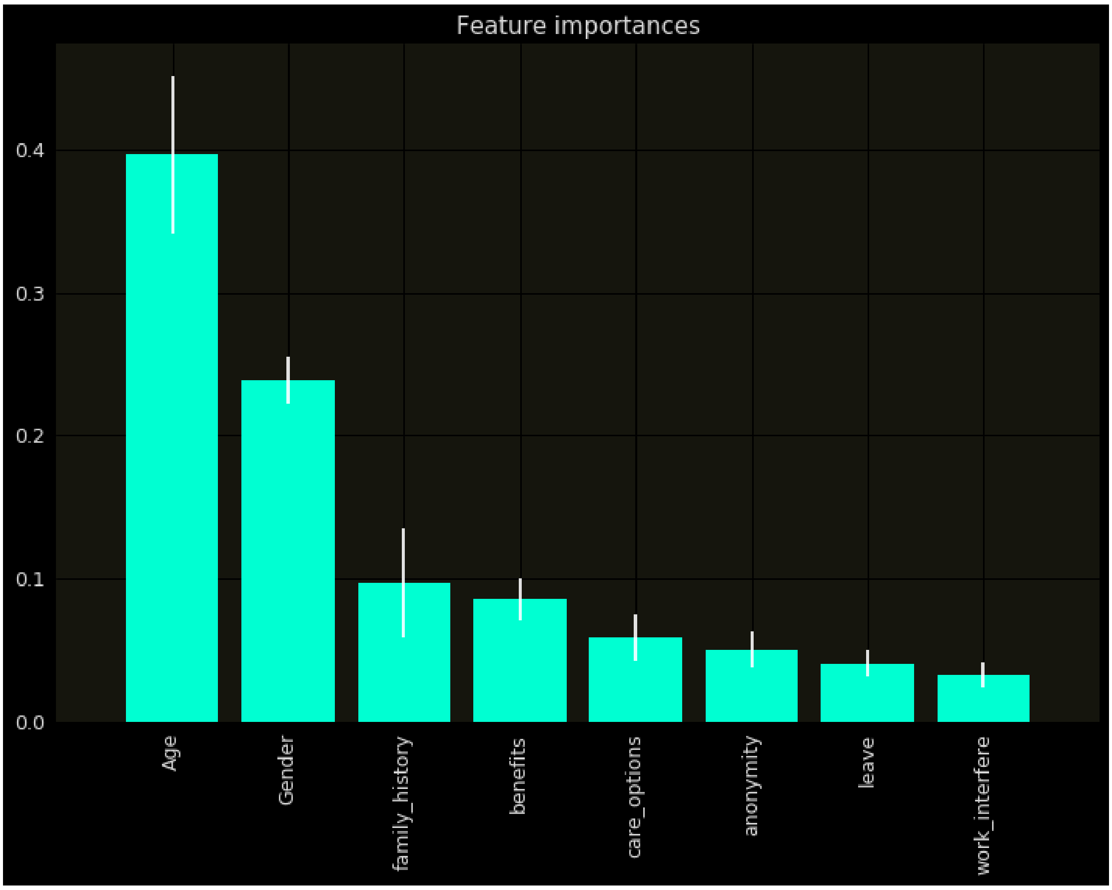


Figure 6: Using Random forest for attribute selection outcome.

For the model building I used:

Nears neighbour because it is very simple and it offered really high accuracy,but the draw back nearest was is very slow with large datasets which is the case in my project.

Decision tree because of it was so easy to visualize the problem , it handled missing data efficiently and it was robust to outliers, gave me higher accuracy than Nears neighbour algorithm,

Adaptive boosting by combining a base learners , boosting offered an advantage, that instead of relying only on one machine learning algorithm , I relied on couple of them “decision trees”and the weighted sum of those learning algorithm became the output of my prediction, this algorithm gave me the highest accuracy that’s why I make my final prediction with it ,

For model evaluation I used confusion matrix which is a technique to evaluate performance for classification algorithms.

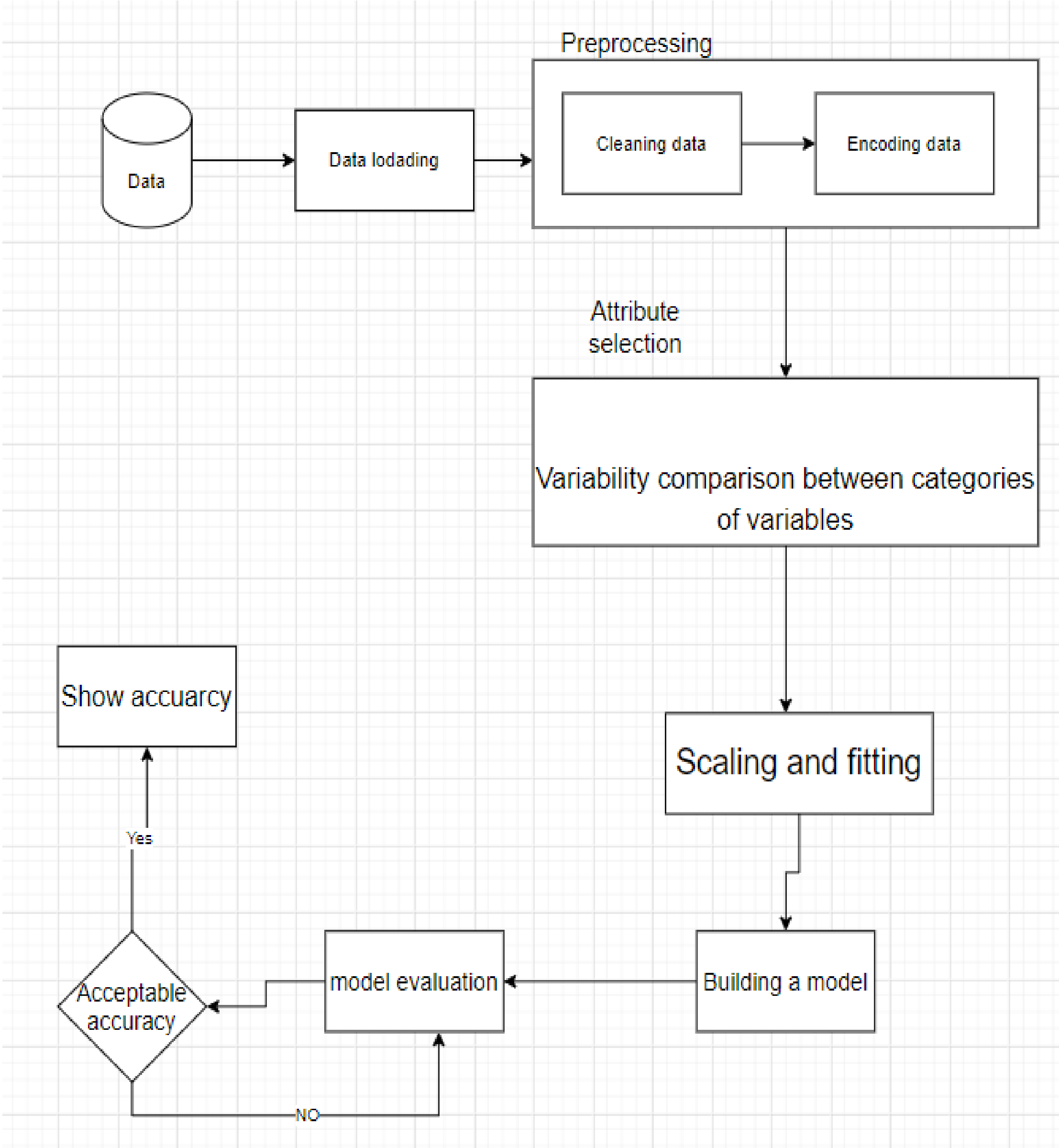


Figure 7: first data set overview

for the second data set we need to classify the emotion the steps are almost the same but at preprocessing stage we remove the words that don’t offer information about the user emotion” Stop-words”, and expand words with apostrophe”expand contradiction”, and lastly we do we try to reduce a given word to it’s given word “lemmatization” , after we split the text into smaller units “ tokenization” , after we build a model and keep evaluating it until we get the acceptable accuracy

The used algorithm at the model building part is BERT “which stands for bidirectional Encoder Representation from Transformers it’s a neural network technique for natural language processing Google itself used BERT at 2019.

For model evaluation I used F1 score which is a statistical analysis to measure test accuracy score.

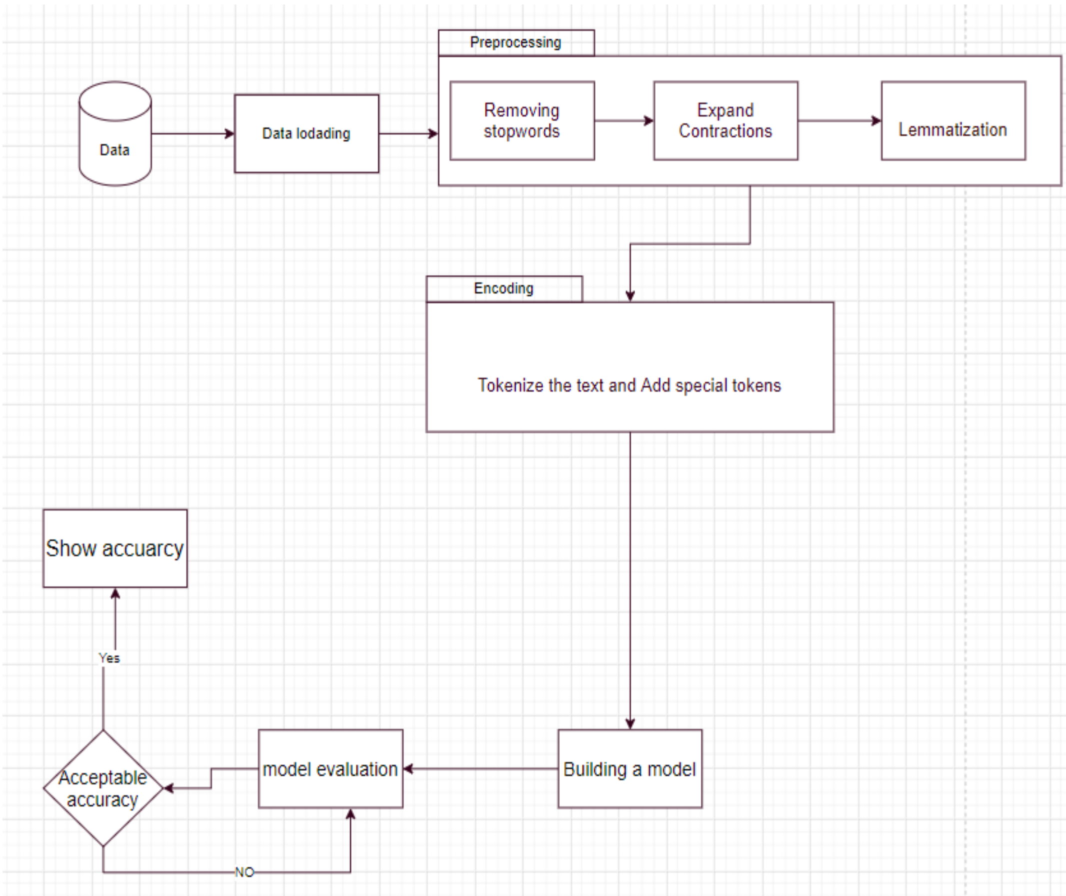


Figure 7: Second data set overview

Chapter 5

Implementation

5.1 General Guidelines

The development of the system took three phases , at the beginning I developed a chat Bot that understands the user input and save that input to a MongoDB instance , this part was made using Abydos frame work and it was essential as the chat bot is the way we collect and process user input, during this phase I needed to train the bot with basic examples of how A human conversation work , by using “intents” A group of sentences that is fed to the chat bot to be able to grasp what the user really wants to say the below figure will show a part of this process

```
- intent: ask_whatspossible
  examples: |
    - Can you explain me in one sentence what you are doing?
    - Can you help me?
    - Could you please show me what you can
    - Great, is there anything else you can do, bot?
```

Figure 9: feeding the bot with the questions

After this I needed to initialize the variables that I need to store for future usage such as the user name and the email address in Abydos they are called slots and they are essential as they represent the chat bot memory, so the data collected from the user can be stored in those slots, and then I created a Mongo Database instance ensuring that those slots are saved for future usage.

```yaml
exercise:
  type: text
  influence_conversation: false
sleep:
  type: text
  influence_conversation: false
diet:
  type: text
  influence_conversation: false
stress:
  type: text
  influence_conversation: false
```

Figure 10: sample of the local memory

```
slots: Object
    time: false
    requested_slot: null
    name: "ali magdy"
    email: "alimagdy94@gmail.com"
    age: "32"
    gender: "male"
    feeling: "fear"
```

Figure 11: sample of the database memory

At the second phase I had to equip the chat bot with the data collected in order to apply the data mining process and deciding to use adaptive boosting for the decision making that was explained earlier in chapter 3 methodology part , after doing so now we have a chat bot able to understand general queries, able to understand the user’s emotion, and also able to classify the user need for treatment or not using their input and the loaded datasets, also important functionalities needed to be implemented , such as the ability for the chat bot to calculate the average time the user takes to respond to the chat bot, and always compare it to the previous time to check if the user is responding faster to the chat bot or not.

At the third phase I had to create a user interface for the chat bot so I built a progressive web application using Laravel and java script to represent the chat bot front end part, the website is pretty informative when it comes to mental health , and has a widget when clicked the chat bot expands and the user is able to converse with the bot using it, the created interface is user friendly and can be customized easily and it will be shown during the system running part.

During the system development we used Abydos , Python , Sklearn ,visual studio code , and Mongo Database.

4.2 System Structure (5~ 6 lines)

- Here we will explain on the coming subsections the overview of the final system we developed by explaining the flow between its components , and in the second subsection we will illustrate the class diagram for the used classes in this system.

4.2.1 System Overview

- The system composed of 4 main stages

The first stage of the system is getting the user input through the user interface at the first step in this stage is applying natural language understand to be able to understand the user's message, the second step is to validate the user input in case of they miss spelled a word , and get the relevant information after this step such as user name , email, and average time took to respond ,then save these information to our mongo data base, at the second stage of the system at the first step we load the emotions data set , and apply preprocessing by going into three phases Removing the words that don't offer information to the meaning, separating the words that has apostrophe, and finally restoring words to their root meaning. After the preprocessing the second step is model creation BERT algorithm,that will allow the system to predict the emotions at this step we keep evaluating the model using F Score till we get the highest score once that happens we save the predicted user emotion, the third Stage first step is to load the patients diagnosis data set ,and applying preprocessing which always takes three different phases first phase is data cleaning , such as incorrect input , miss spelled input , and making sure all the input is in lower case, then dealing with missing data such as filling missing age with the average age of the user's, second phase is data encoding to make sure to encode each category into numeric value, and at the third phase is scaling which is performed for the attribute age specifically, because it has a lot of values which can affect our learning badly.

The second step at the third stage is to select the right attribute to apply learning on I used random forest algorithm for this purpose by using it's importance measure " checking if removing an attribute will increase or decrease the accuracy" and based on this we chose the attributes, and the third step is building a model using "nearest neighbor, decision tree, random forest, and adaptive boosting" and keep evaluating the model till we get the highest accuracy, then save the best resulting model result in the data base.

At the fourth stage based on the user's emotion , time to respond , and diagnosis , the bot offers a response for the user , and the doctor can have access to the system to validate the patients diagnose by checking their conversation with the chat bot , and the outcome of the machine learning process.

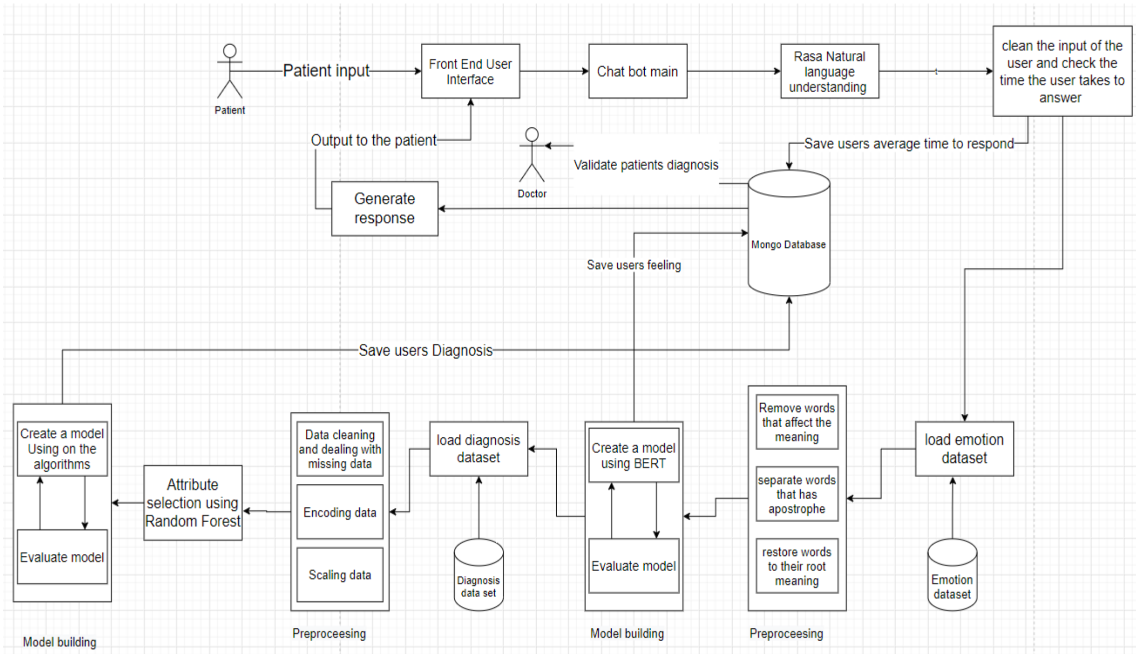


Figure 12: Classification of user emotion and diagnosis architecture overview

4.2.2 Class Diagram

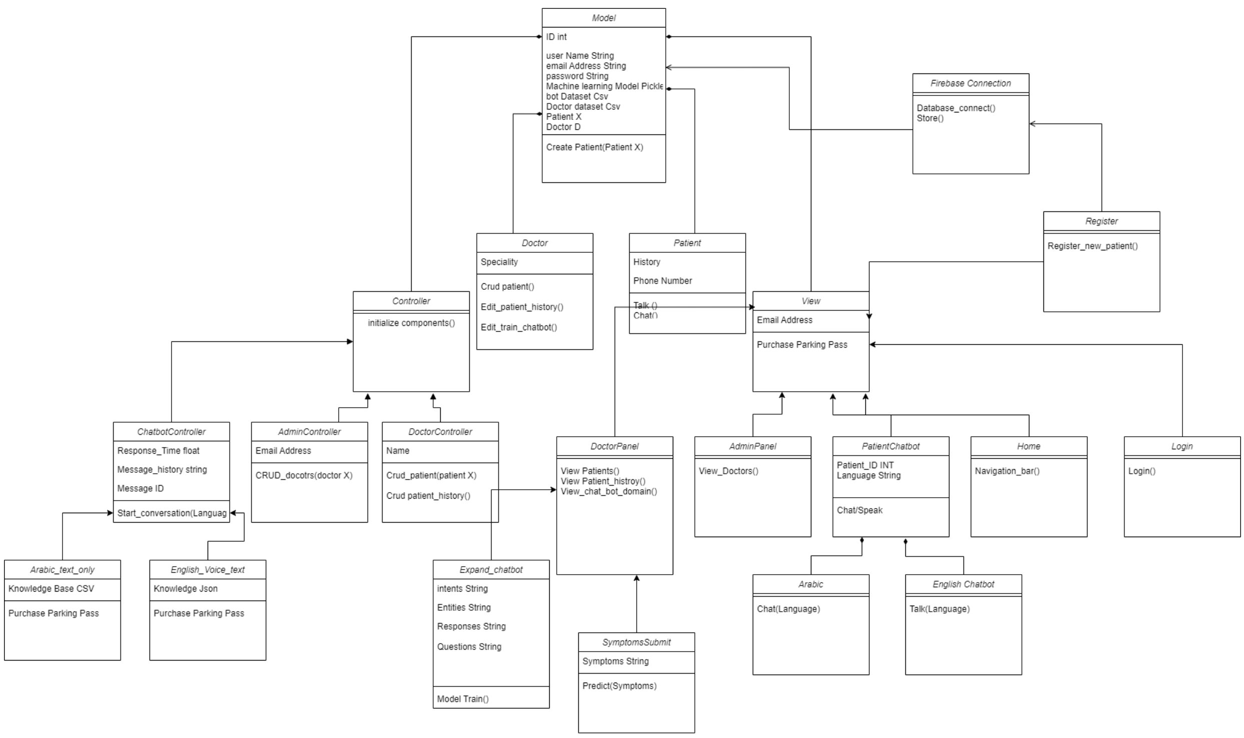


Figure 13 Class Diagram

For each class:

- Mention its Name, its responsibility, its relation with other classes.
- In the level of (attributes and methods) mention its Name , its responsibility , its relation with other attributes
-

Class Name	AdminController
Sub Classes	Doctor, Patient Chatbot actions
responsibility	CRUD doctors and manage the system
Collaboration	Doctor's Model
Attribute	Int ID String Name Int phone number String Gender
Operation	Doctor ADD/REM/EDIT/DEL/ doctor()

- In the level of (attributes and methods), don't explain all of them just select the important of them.

Class Name	DoctorController
Super Classes	Admin Contoller
responsibility	View Patient history and validate the diagnosis generated by the chat bot And also train the chatbot for new sentences

Collaboration	Need to interact with Chatbot actions and get the diagnosis predicted and make sure that this prediction is correct , also interacts with staff with salary to get the working hours and salary
Operations	Int Validate Diagnosis() String offer_treatment()
Class Name	Receptionist
Super Classes	User, Staff with salary
Sub Classes	Invoice
responsibility	Generate invoice for the patient
Collaboration	Interacts with invoice according to the doctor's instructions, also interacts with staff with salary to get the working hours and salary
Attribute	Invoice X
Operations	Void Create invoice() Void delete invoice() Void Modify invoice() Invoice update invoice() Boolean Search_invoice()
Class Name	Patient
Super Classes	User
responsibility	Enter their information and get their mood predicted and their case severity
Collaboration	Need to interact with Chatbot actions , converse with the bot , and have the invoice paid
Attribute	Inherits from user

Operations	Void Pay_Invoice()
Class Name	Invoice details
responsibility	Have the details of the invoice stored such as time of visit the patient case
Collaboration	Need to interact with invoice to get the invoice number and the amount
Attribute	String Patient Case Float Time_of_visit
Class Name	ChatbotController
Super Classes	User
responsibility	Get the patient input and predict what they want to say
Collaboration	Interacts with time improvement to calculate the average time user took to respond to the chat bot ,interacts with mood tracker to interpret the patients mood, also with diagnosis to ask the user set of questions and predict if they require immediate treatment
Attributes	String Latest Message List Time Stamp
Class Name	Diagnose
responsibility	Collect patients personal details and validate it
Collaboration	Need to interact with learning to apply machine learning and predict if the patient require treatment
Attribute	String Self_employment String Family history

	String treatment String work_interference String tech_company String benefits String Care_options String Anonymity String medical_leave String discuss_with_collegue List new_input
Operations	List Validate input(noisy input)
Class Name	Learning
responsibility	Use user's details and time improvment and apply machine learning algorithms to get the highest accuarcy classification for the patients case.
Collaboration	Need to interact with Diagnose to get clean input
Attribute	Data frame train_df List Feature_Cols Dataframe X Dataframe Y
Operations	void evaluateClassModel(Model, ytest , y predict) tuningRandomizedSearchCV(model, param_dist) int KNN() int DecesionTreeClassifer() int RandomForest() int Boosting()

Class Name	Mood checker
responsibility	Predict patient current mood based on their latest response
Collaboration	Need to interact with Chatbot actions to get the latest response.
Attribute	Dataframe X Dataframe Y
Operations	String Predict(user_latest_sentence)
Class Name	Time_improvement
Purpose	Check the time user took to respond to the chat bot
Collaboration	With chat bot action to extract the time stamp each time the patient sends a message
Operations	int Count_Interactions (user_ID) Float Calculate_Averege_time_improvment()

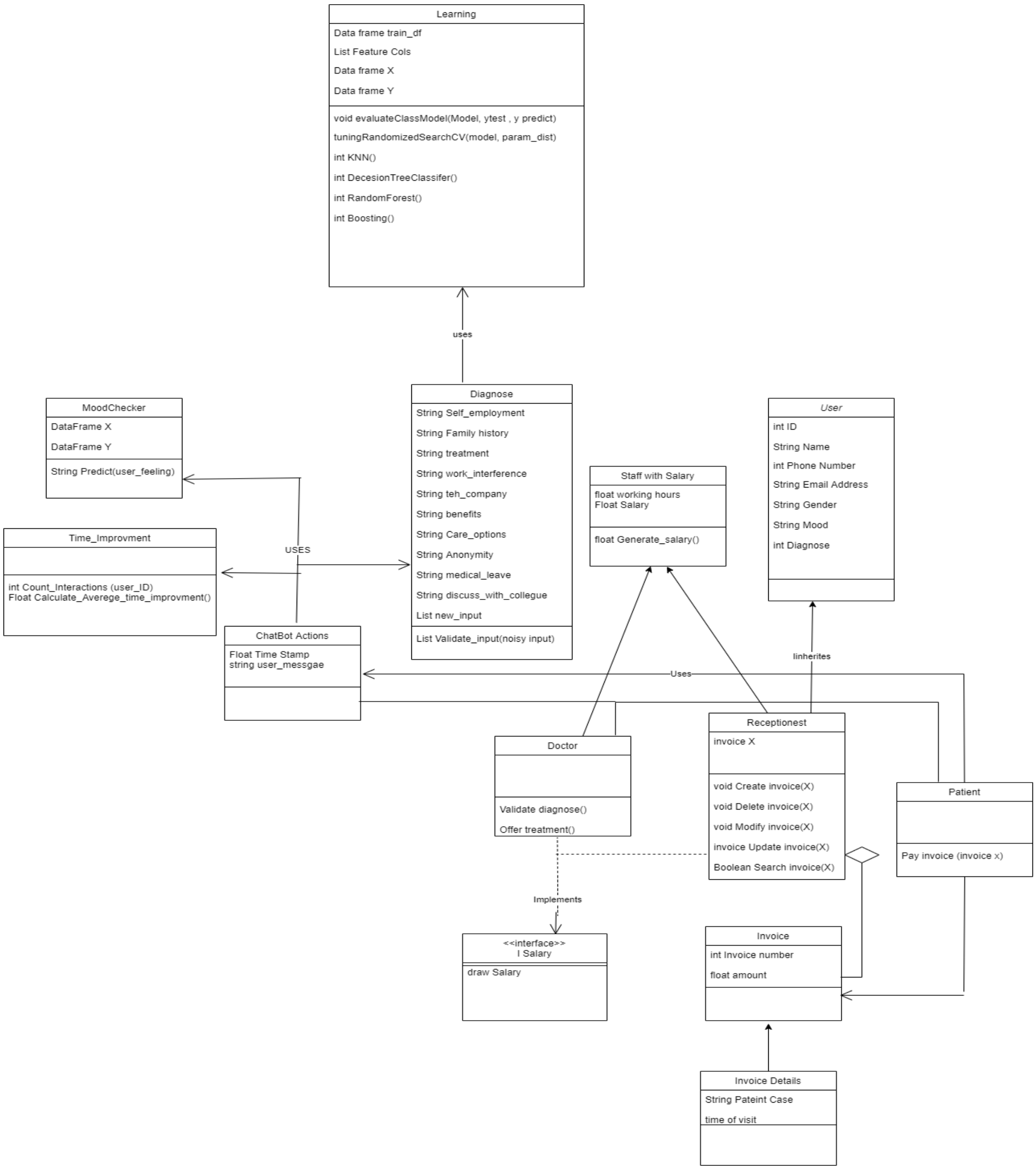


Figure 12: Class Diagram

4.3 System Running

4.3.1 Component A

Login and register

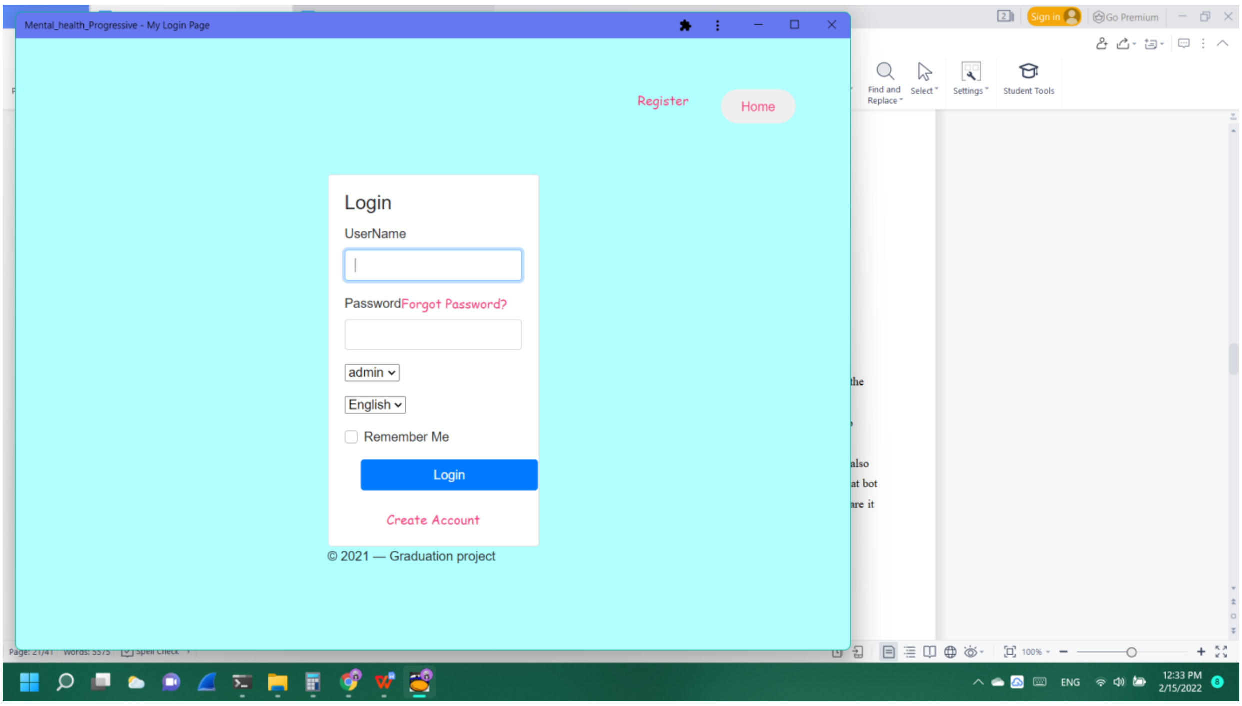


Figure 14 Login page

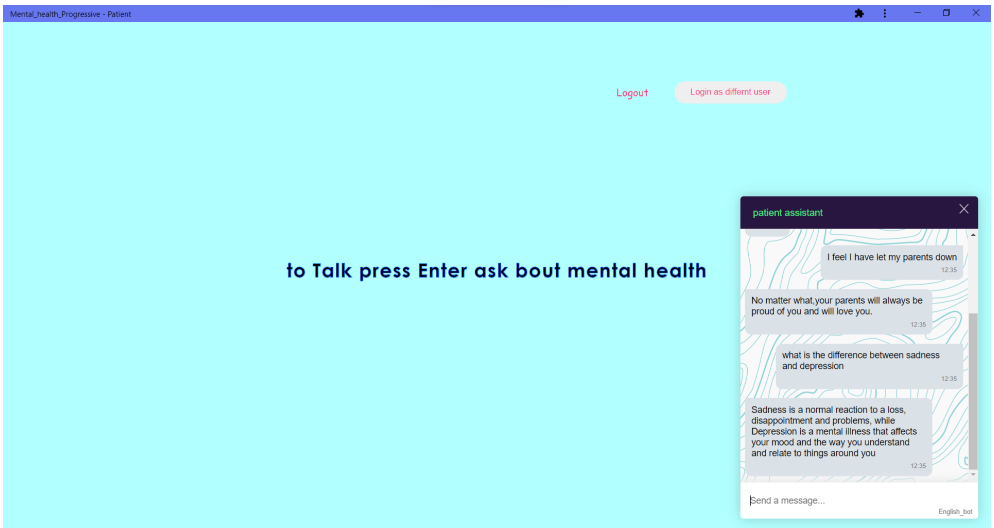


Figure 15 Patient interface

The input function is determined by the patient and the output shows at the same widget Also the second input is: each message the user sends the bot calculates the average time user takes responding to chatbot each time the user sends a message and the output here is a variable in the data base called time changes to true if the user responding quickly to the chatbot and false if the user responding slowly to the bot.

4.3.2 Component B

The input of this function is the user feeling that he inputs and the mood data set to be able to train the model, this function generates output by the end of the conversation responding to the user emotion, in the case shown in the figure below the bot predicted that the user is feeling sadness and responds accordingly.

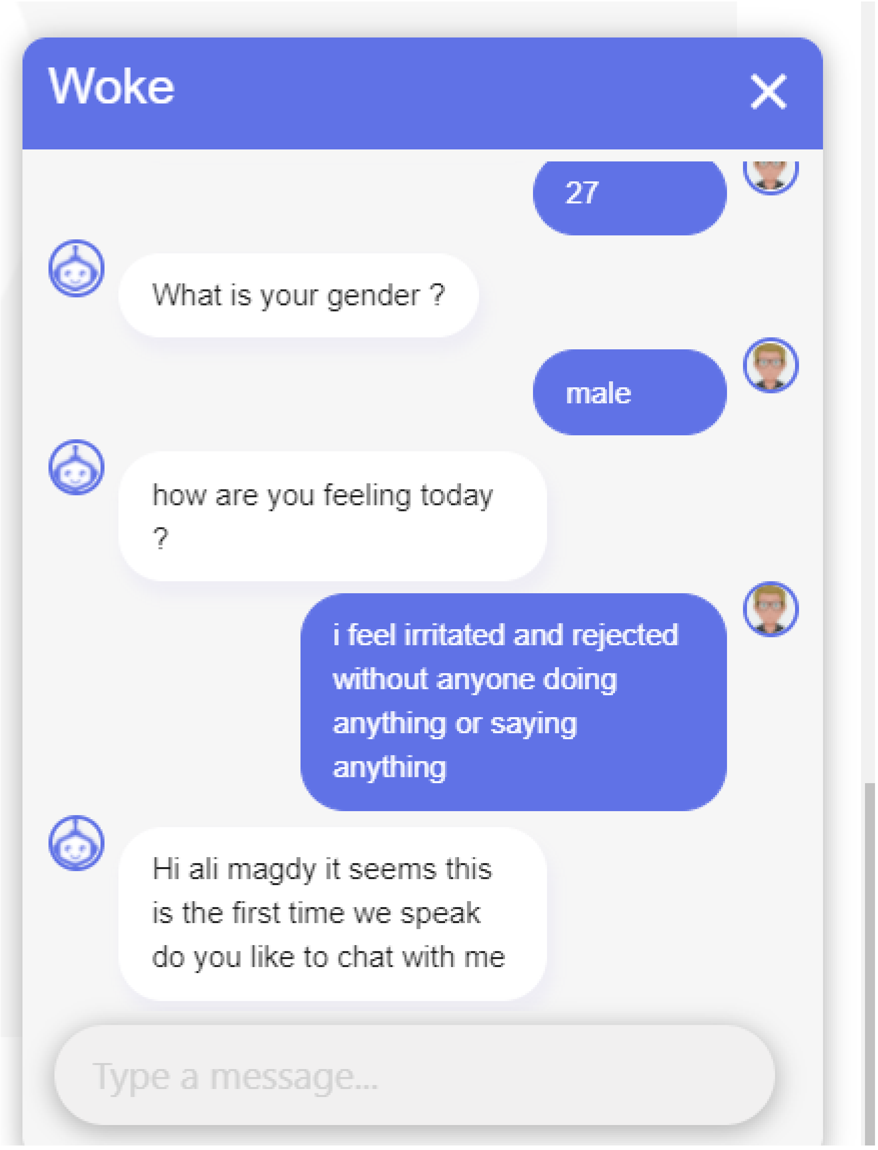


Figure 16: component B input

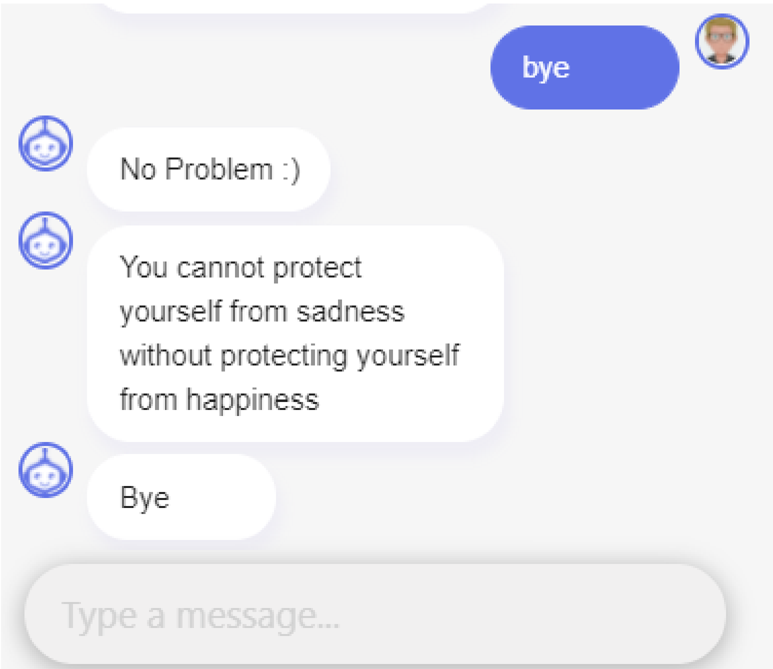


Figure 17: second component input

4.3.3 Component C

The input to this component is the mental health data set that contains the 1000 respondents ,and the user's answer to the questions below and based on the user answers and after applying machine learning the output shows weather the user require immediate treatment or not.

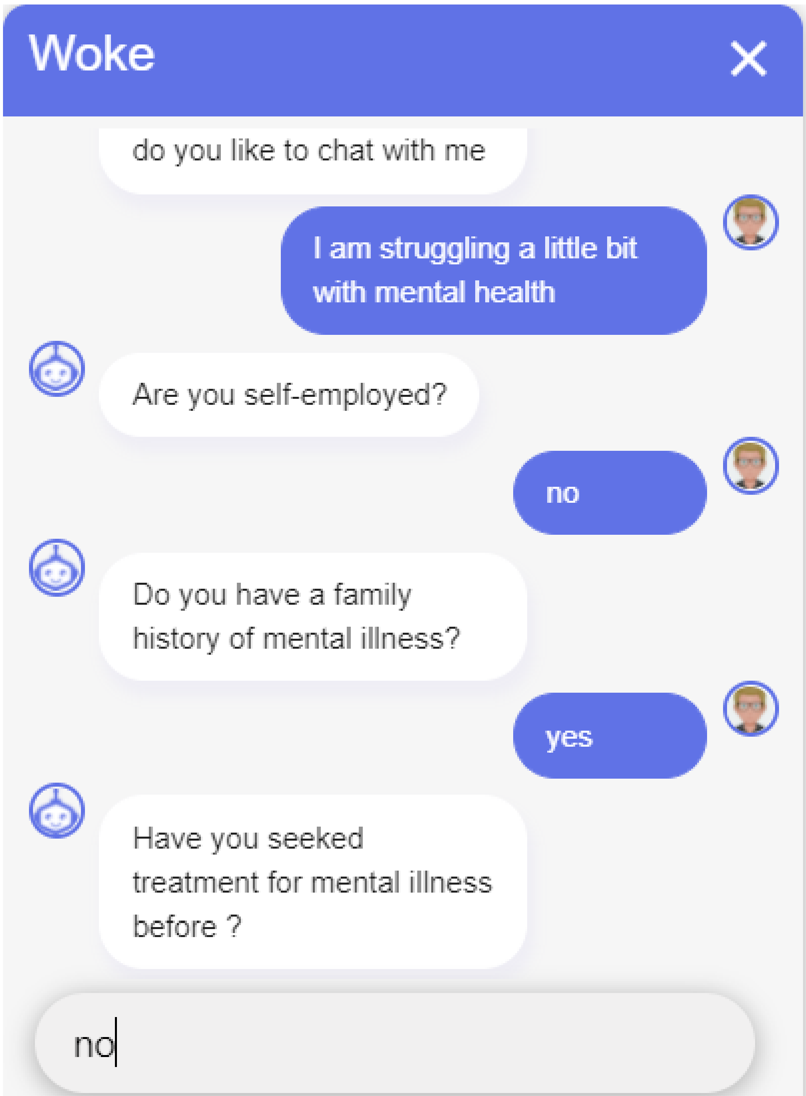


Figure 18: third component input

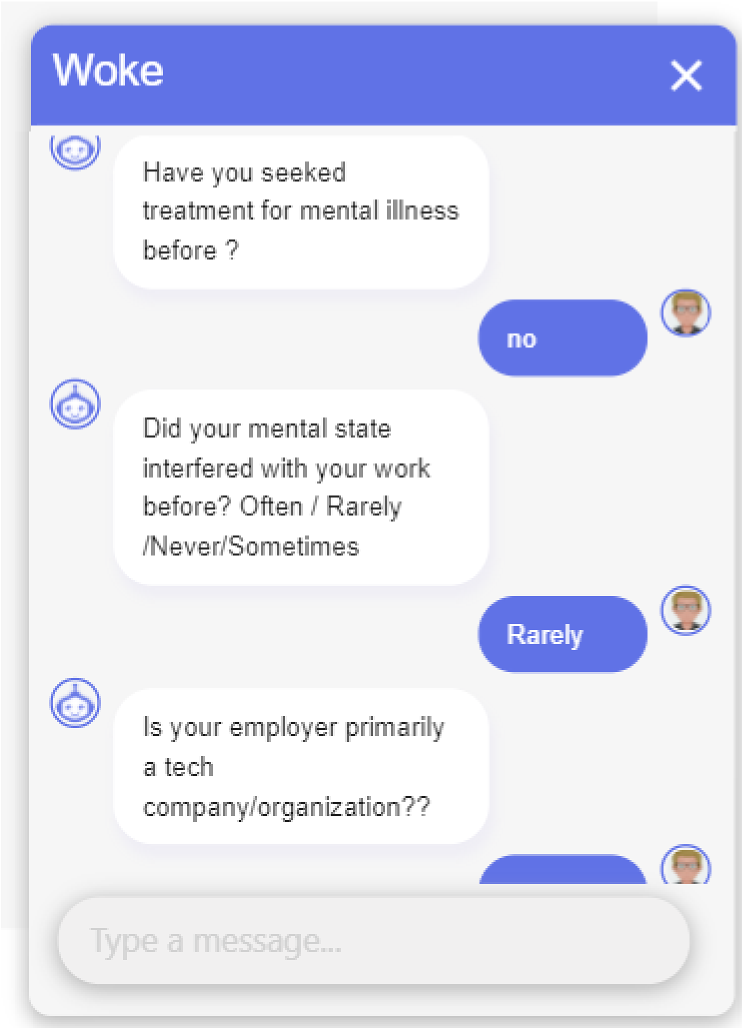


Figure 19: third component input

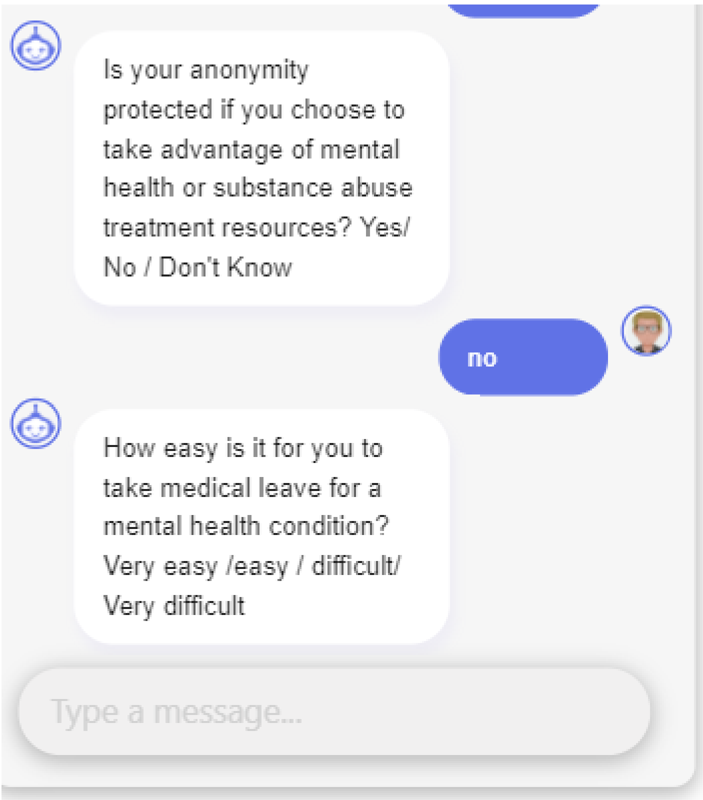


Figure 20: third component input

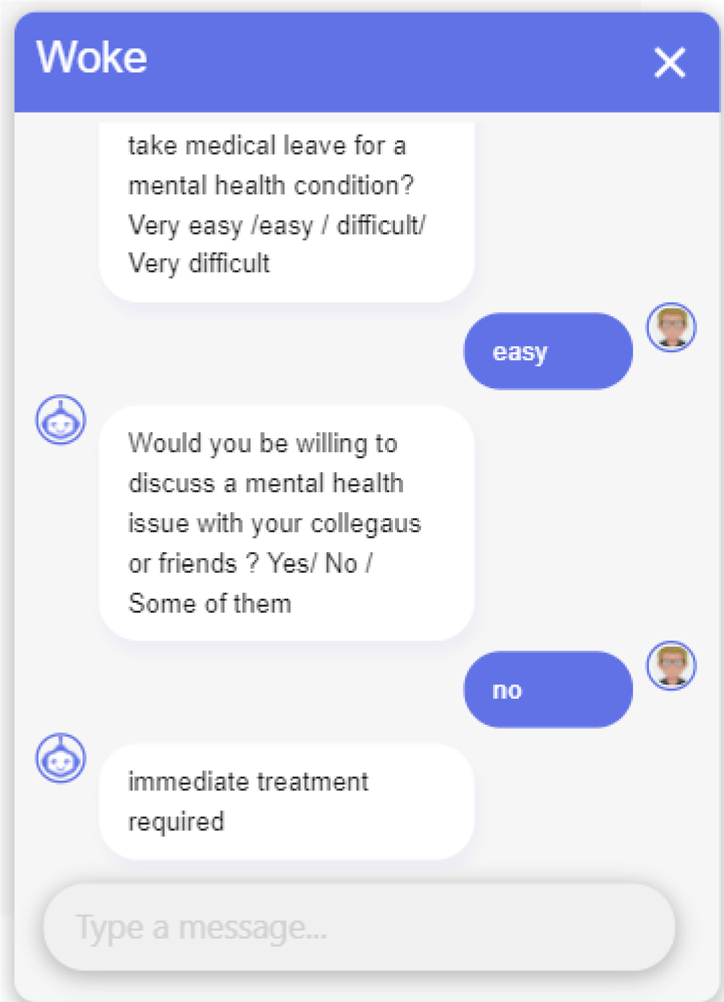


Figure 21: third component output

And also the Admin view who can control doctors add and edit remove them ,

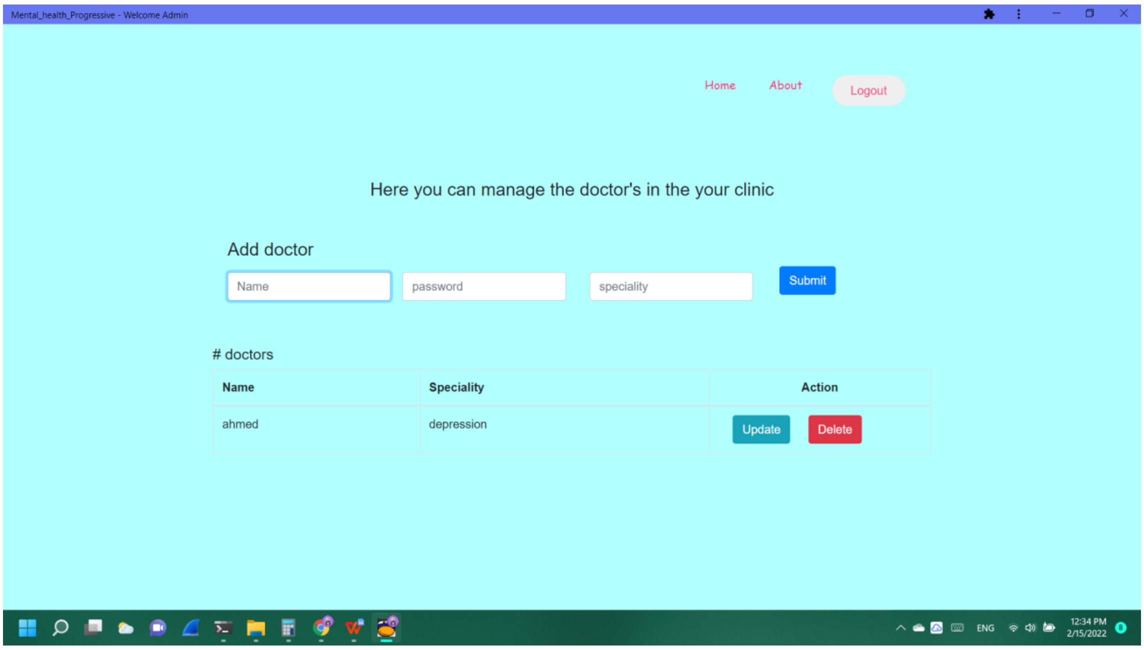


Figure 22 Admin patient control

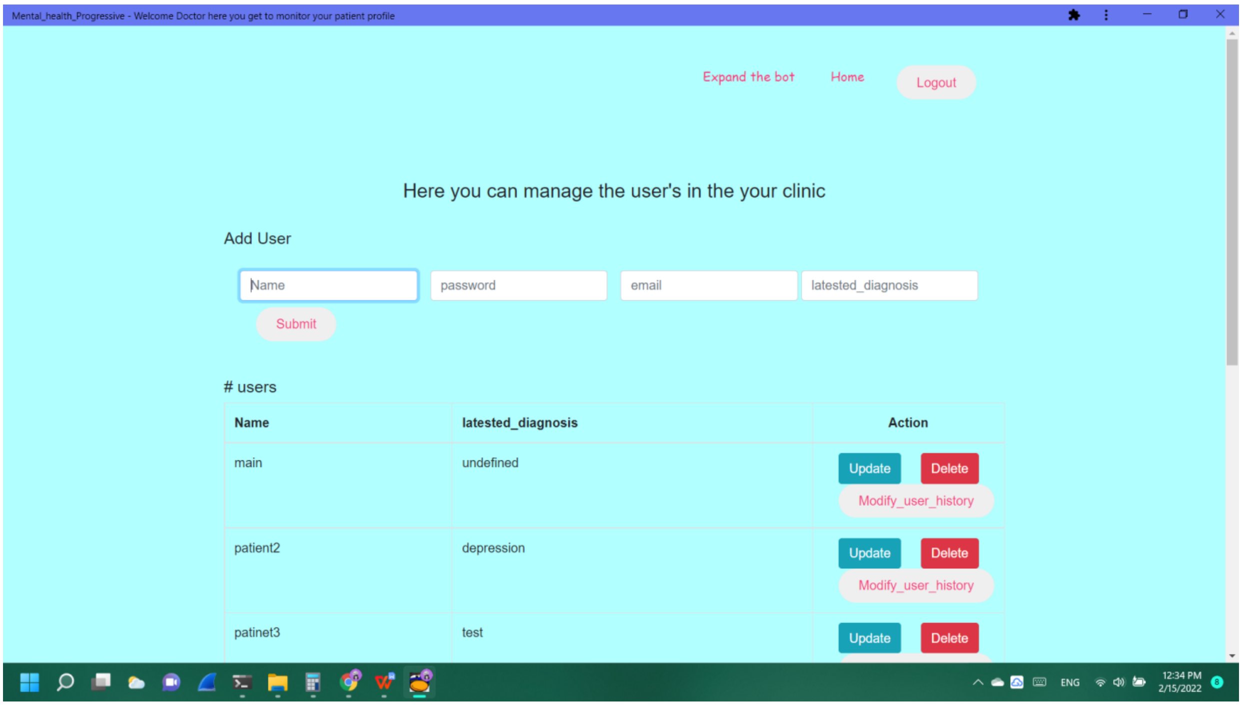


Figure 23 Doctor Control panel

Doctor can Crud patients , edit patient history to cooperate with other doctors

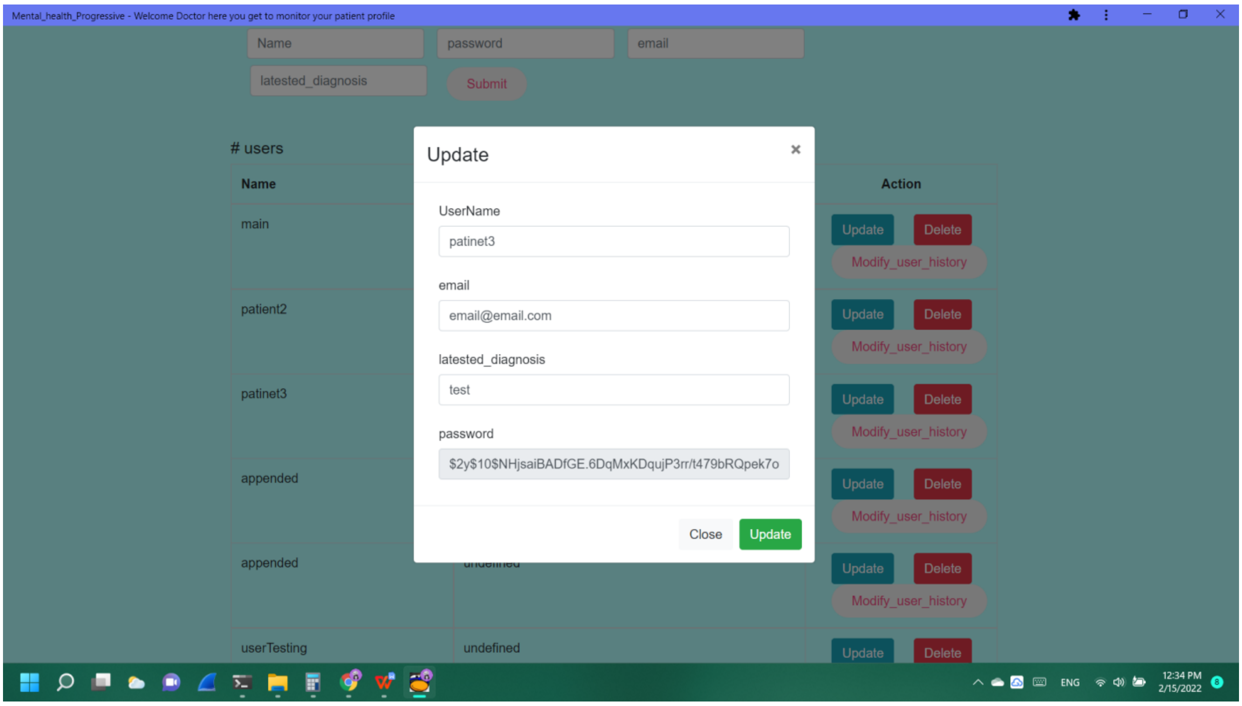
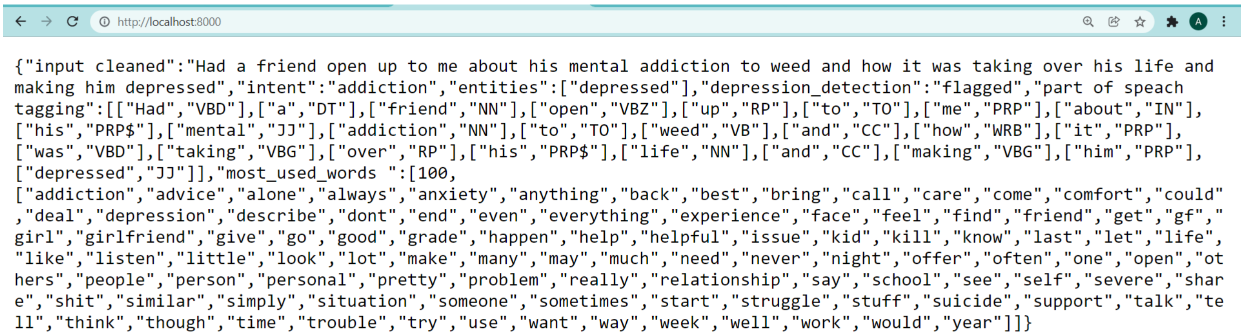


Figure 24

Final component is an API that Encapsulates all of the above



http://localhost:8000

{"input cleaned":"Had a friend open up to me about his mental addiction to weed and how it was taking over his life and making him depressed","intent":"addiction","entities":["depressed"],"depression_detection":"flagged","part of speach tagging":[["Had","VBD"],["a","DT"],["friend","NN"],["open","VBZ"],["up","RP"],["to","TO"],["me","PRP"],["about","IN"],["his","PRP$"],["mental","JJ"],["addiction","NN"],["to","TO"],["weed","VB"],["and","CC"],["how","WRB"],["it","PRP"],["was","VBD"],["taking","VBG"],["over","RP"],["his","PRP$"],["life","NN"],["and","CC"],["making","VBG"],["him","PRP"],["depressed","JJ"]],"most_used_words ":[100,["addiction","advice","alone","always","anxiety","anything","back","best","bring","call","care","come","comfort","could","deal","depression","describe","dont","end","even","everything","experience","face","feel","find","friend","get","gf","girl","girlfriend","give","go","good","grade","happen","help","helpful","issue","kid","kill","know","last","let","life","like","listen","little","look","lot","make","many","may","much","need","never","night","offer","often","one","open","others","people","person","personal","pretty","problem","really","relationship","say","school","see","self","severe","share","shit","similar","simply","situation","someone","sometimes","start","struggle","stuff","suicide","support","talk","tell","think","though","time","trouble","try","use","want","way","week","well","work","would","year"]]}

Figure 25

API to make sure our bot is mobile

Chapter 6

Results and Evaluation

6.1 General Rules

First best Result was using edit distance between sentences which is currently implemented, second best result was using a three layered neural network,

Thir best result is when I used cosine similarity for the Arabic chat bot , but ended up using multi nominal naive bayes,

That chat bot is really robust and intelligence the fall back when the chat bot has no clue what you are saying is based on your conversational context

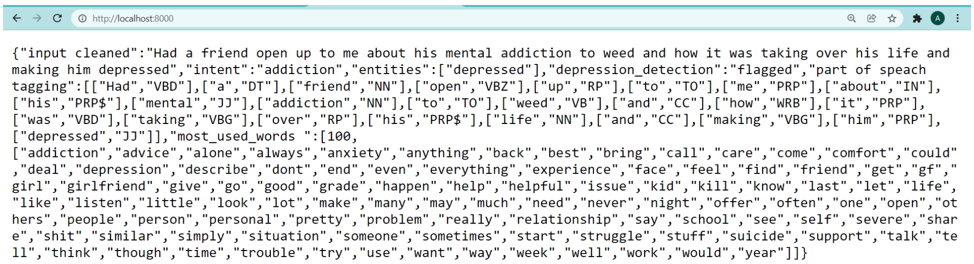
5.2.2 Limitations

The problem domain is AI complete no matter how high my models reached still not enough, The arabic chatbot was built using Naive bayes and it over fitted no matter how hard I try is is still boolean retrieval “just answer what I know”

The English chat bot is more robust and it was trained on a dataset called DIACWOZ to be able to sense depression in the tone of voice of the speaker just by answering couple of questions

5.3 Evaluation

- The built system is really mobile and it can migrate from one network to other very easily also



{"input cleaned":"Had a friend open up to me about his mental addiction to weed and how it was taking over his life and making him depressed","intent":"addiction","entities":["depressed"],"depression_detection":"flagged","part of speach tagging":[["Had","VBD"],["a","DT"],["friend","NN"],["open","VBZ"],["up","RP"],["to","TO"],["me","PRP"],["about","IN"],["his","PRP$"],["mental","JJ"],["addiction","NN"],["to","TO"],["weed","VB"],["and","CC"],["how","WRB"],["it","PRP"],["was","VBD"],["taking","VBG"],["over","RP"],["his","PRP$"],["life","NN"],["and","CC"],["making","VBG"],["him","PRP"],["depressed","JJ"]],"most_used_words ":[100,["addiction","advice","alone","always","anxiety","anything","back","best","bring","call","care","come","comfort","could","deal","depression","describe","dont","end","even","everything","experience","face","feel","find","friend","get","gf","girl","girlfriend","give","go","good","grade","happen","help","helpful","issue","kid","kill","know","last","let","life","like","listen","little","look","lot","make","many","may","much","need","never","night","offer","often","one","open","others","people","person","personal","pretty","problem","really","relationship","say","school","see","self","severe","share","shit","similar","simply","situation","someone","sometimes","start","struggle","stuff","suicide","support","talk","tell","think","though","time","trouble","try","use","want","way","week","well","work","would","year"]]}

Figure API with accuracy measure for every thing summed up

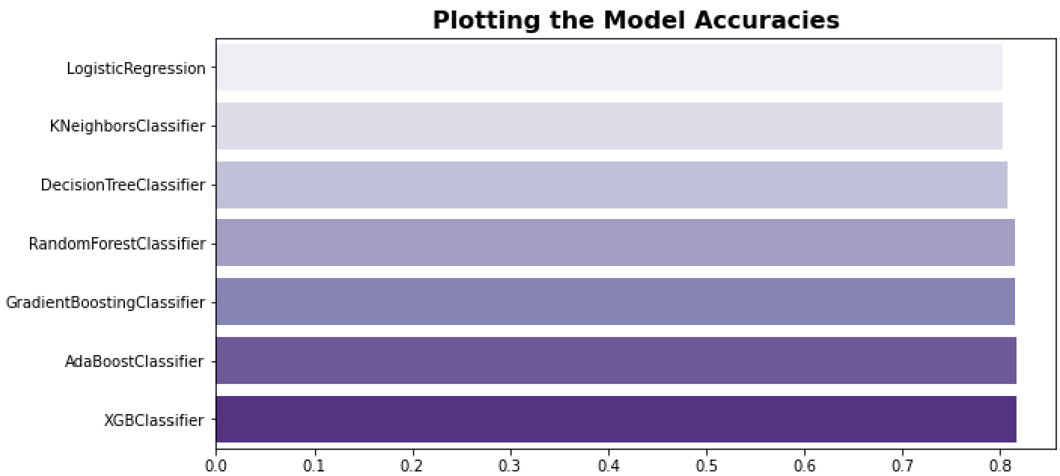


Figure 25

Figure Accuracy Evaluation

Models Number	Training Data	Testing Data	Training time	Accuracy	Validation accuracy
Cosine Similarity	2000	666	instant	80%	60%
Naive Bayes	10000	3333	instant	83%	79%
Neural network	10000	3333	12 Seconds	92%	90.21%
Edit Distance	10000	3333	instant	98%	95%
DIAC	1000000000	3333333	1×1	90%	80%

As shown in the previous table I went with the best outcomes

5.3.2 Time Performance

Regarding the time, it was okay when I handling textual data as shown most of the models trained instantly but training the DIAC WOZ dataset takes a week with a powerful cloud GPU

Chapter 7

Conclusions and Future work

7.1 General Rules

For more personalized experience it is beneficial to have more than one language supported by the chat bot , that will increase the reach the bot will have geographically and provide the answers in the patient's Native language. As It was concluded in the previous section Dr.Mohamed ELShami highlighted the importance of body language when it comes to identifying mental illness ,

we can add a software that is able to identify the human body shape ,and map the key points of our anatomy, a lot of information is embedded in our facial expressions and hand movement which can

Learnt working with machine learning in a way that I couldn't imagine ,In addition to body language using the patients tone of voice can help tremendously, and it is not a hard task with the huge advancement in speech emotion recognition through machine learning.

In order to make the bot effective helping people with mental health , it highly important to collaborate with group of mental health professional's designing appropriate conversation that mimics cognitive behavioral therapy, to extended the bot power from noticing that the patient needs treatment and make it actually treat the patient.

Bibliography

1.Alepis, E., & Virvou, M. (2011). Automatic generation of emotions in tutoring agents for affective e-learning in medical education. Expert Systems with Applications, 38(8): 9840–9847.

2.Ashok, G., Brian, C., Mithun, K., Shanu, S., Abhinaya, S., & Bryan, W. (2015). Using Watson for Enhancing Human-Computer Co-Creativity. AAAI Symposium: 22–29.

3.Avalverde, D. (2019). A Brief History of Chatbots. Perception, Control, Cognition. Retrieved March 9, 2019 from: https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/

4.Ayedoun, E., Hayashi, Y., & Seta, K. (2015). A Conversational Agent to Encourage Willingness to Communicate in the Context of English as a Foreign Language. Procedia Computer Science, 60(1): 1433–1442.

5.Ben Mimoun, Mohammed Slim, & Poncin, I. (2015). A valued agent: How ECAs affect website customers' satisfaction and behaviors. Journal of Retailing and Consumer Services, 26: 70– 82.

6.Chatbot Magazine (2019). A Visual History of Chatbots. Retrieved March 9, 2019 from: https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

7.Colace, F., De Santo, M., Lombardi, M., Pascale, L., Pietrosanto, A. (2018). Chatbot for E-Learning: A Cases Study. International Journal of Mechanical Engineering and Robotics Research Vol. 7, No. 5, September.

8.Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019 from: https://www.youtube.com/watch?v=38sL6pADCog

9.Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning: Routledge. https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

10.Lip ko, H. (2018). Meet Jill Watson: Georgia Tech's first AI teaching assistant. Retrieved on March 9, 2019 from: https://pe.gatech.edu/blog/meet-jill-watson-georgia-techs-first-ai-teaching-assistant.

11.Maruti Techlabs. (2018). Why can chatbots replace Mobile Apps immediately? Retrieved March 9, 2019 from: https://www.marutitech.com/why-can-chatbots-replace-mobile-apps-immediately/