Practicum Sprint #9

Mental/Physical Illness Chatbot

Tusheet Goli, Tejas Pradeep, Akshay Sathiya, Pranav Khorana, Sanket Manjesh, Rahul Chawla

tgoli3@gatech.edu, tpradeep8@gatech.edu, asathiya6@gatech.edu, pkhorana3@gatech.edu, smanjesh3@gatech.edu, rchawla36@gatech.edu

1 ACCOMPLISHMENTS THIS WEEK

1.1 Akshay Sathiya's Progress

During this sprint, Akshay continued working on the physical illness prediction (PIP) pipeline. The PIP pipeline consists of two parts, the symptom extraction (SE) model (multi-output classifier consisting of several neural networks), which predicts the symptoms included in a user's message, and the PIP models (random forest and neural network classifiers), each of which uses the predicted symptoms to predict the user's physical illness risks. When using the PIP pipeline for inference, only one of the two PIP models is used (the specific PIP model to be used for inference is set/configured in the code, based on which one helps the PIP pipeline perform better on the updated test suite).

Akshay refactored the code so that the SE and PIP models can be trained independently, instead of together, to speed up the process of tuning hyperparameters for each model. Akshay tuned the hyperparameters for the SE model and both PIP models, which all achieved perfect/near-perfect accuracies and F1 scores on their respective training/testing data and on various folds of their respective datasets. The PIP pipeline as a whole performed well on the updated test suite, so the SE and the PIP models are unlikely to be overfitting despite their perfect/near-perfect accuracies and F1 scores.

Akshay updated the test suite so that it consists of 14 total test cases. The first ten test cases each consist of a message, ground truth symptoms, and the ground truth physical illness that all the ground truth symptoms correspond to. Akshay manually created each test case by referring to the symptoms and physical illnesses in the Kaggle physical diseases dataset (Patil, 2020) and writing a message that described multiple symptoms corresponding to the same physical

illness. This ensures that the test cases are representative of the messages that the app would receive from real users. Akshay configured the response builder to provide the top three physical illness predictions (in decreasing order of probability) for a user's message, and each test case was considered to be passed if the ground truth physical illness was one of the top three predictions.

For both the random forest and neural network PIP models, the PIP pipeline passed eight out of ten test cases, with the ground truth illness showing up as the top prediction for seven of the eight passed test cases and showing up as the bottom/third prediction for one of the eight test cases. The neural network PIP model has more confidence in its predictions (higher probabilities) than the random forest PIP model, so the neural network PIP model has been set as the default PIP model for the PIP pipeline when the PIP pipeline is used for inference.

The last four test cases had messages which did not have any symptom information at all, so no symptoms should be extracted from those messages and the response sent back to the user should not contain any physical illness predictions. Akshay coded up a check in the PIP pipeline to ensure that no physical illness predictions are made for messages that have no extracted symptoms, and the response builder would return an empty string as the PIP response for those messages. For both the random forest and neural network PIP models, the PIP pipeline passed all four no-symptom test cases.

The PIP pipeline's performance on the updated test suite indicates that the SE and PIP models are unlikely to be overfitting and indicates that these models (and the PIP pipeline as a whole) are likely to perform well on messages from real users.

Akshay also configured the response builder to provide precautions in the response that the user can take for each of their predicted physical illness risks to prevent/mitigate those illnesses. Precautions for each physical illness were included in the Kaggle physical diseases dataset (Patil, 2020).

Akshay merged with work with the *main* branch in the GitHub repository. Next sprint, Akshay will help resolve any integration issues that arise with the different parts of the app and work on the Sprint #10 deliverables (demo, documentation, etc.).

1.2 Pranav Khorana's Progress

This week, Pranav made some more progress on integrating the react native UI with the flask backend. He was able to successfully register and authenticate a user using the front end. He also utilized asynchronous function and callbacks in order to wait for the backend to receive messages from the front end, so that the chatbot could formulate a response to send back to the frontend. This involved the use of hooks such as useEffect, useState, and more. This coming week, Pranav will communicate more with his other group members to clarify issues such up to how many messages the backend should receive before sending a response back utilizing the NLP models.

1.3 Rahul Chawla's Progress

During this sprint, Rahul worked on completing the mental illness prediction models. He used the train and test emotions datasets from the previous weeks to create a trained sentiment analysis classifier with sklearn. Once completing the classifier, Rahul created a function to develop a response based on the user's input messages. Depending on the distribution of sentiments with these messages, Rahul wrote the prediction model to indicate if the user had a likelihood of Depression, Anxiety, or Bipolar Disorder.

Furthermore, Rahul worked with the response builder to compile the results into a format that can be easily accessed by other components of the application for integration.

For sprint 10, Rahul will continue to make improvements to the mental health prediction model, which may include fine-tuning how the mental disorders are classified from the user messages. Rahul also plans to work with the others to make any changes needed to this component to allow for a smooth integration of the full app.

1.4 Tusheet Goli's Progress

This week, Tusheet finished integrating the app. He finalized the database schemas and was successfully able to test it with actual data in the tables. The schemas of the tables before testing are shown below in the image.

```
d9dol0u4tdoh24=> \dt
           List of relations
 Schema | Name  | Type  |     Owner
--------+-------+-------+----------------
 public | chat  | table | upgypwamervwra
 public | users | table | upgypwamervwra
(2 rows)


d9dol0u4tdoh24=> \d+
                                    List of relations
 Schema | Name  | Type  |     Owner      | Persistence | Access method |    Size    | Description
--------+-------+-------+----------------+-------------+---------------+------------+-------------
 public | chat  | table | upgypwamervwra | permanent   | heap          | 8192 bytes |
 public | users | table | upgypwamervwra | permanent   | heap          | 8192 bytes |
(2 rows)


d9dol0u4tdoh24=> select * from users;
 user_id | fname | mname | lname | dob | gender | email | password
---------+-------+-------+-------+-----+--------+-------+----------
(0 rows)


d9dol0u4tdoh24=> select * from chat;
 user_id | message | response
---------+---------+----------
(0 rows)
```

```
d9dol0u4tdoh24=> select * from users;
             user_id              |  fname  |  mname   | lname |    dob     | gender |         email         | password
--------------------------------------+---------+----------+-------+------------+--------+-----------------------+----------
 a1d6225f-936d-4893-844d-20571779f8b0 | Tusheet | Sidharth | Goli  | 04/24/1984 | M      | tusheetgoli@yahoo.com | password
(1 row)
```

He also finished coding up all the necessary endpoints to connect the backed and model for both, the Physical Illnesses Prediction Model and the Mental Illnesses Prediction Model. The code can be seen in the api folder in server.py. He tested all the endpoints and ensure they work as intended. The UI team will work on connecting to these endpoints to complete the integration of the application on their end.

For Sprint 10, Tusheet will help both the model or UI teams with any issues they might be facing with the API endpoints to complete a smooth integration on both ends. He will work on adding any additional endpoints deemed necessary/useful by the model or UI team. Most of the work with the backend, database, and API that was assigned to Tusheet is done, and he will work on helping other team members to complete their stuff as well. All of Tusheet's work is pushed to the *main* branch of Github and can be seen there.

https://github.gatech.edu/gt-cs6440-hit-spring2022/Team-2-Mental-Illness-Chatbot/blob/main/server.py

1.5 Tejas Pradeep's Progress

This week Tejas worked extensively with tusheet to integrate the various components of the app. Tejas worked on hosting the database on Heroku and setting up a connection to the database. Further, Tejas worked on building the various endpoints required to link the front end to the model. Tejas further tested the endpoints that exist to ensure that they are working. Our model also had some issues linking to packaging with the physical illness model and Tejas worked on reformatting our codebase to fix packaging issues with the physical illness model. For Sprint 10 Tejas shall help the front end finish their tasks and fully integrate the system.

1.6 Sanket Manjesh's Progress

This week, Sanket helped work on functionality for the mental illness prediction model. He worked with Rahul to use sklearn to build a classifier that could identify if a user was Bipolar, has Anxiety, or had Depression based on the frequency of certain words they said in the chatbot. Sanket plans to also try to sue TextBlob for sentiment analysis and classification of these illnesses to see if he can receive better results.

2 CHALLENGES ENCOUNTERED

None

3 FUTURE PLANS

The team members made significant progress on their tasks and integrated the parts of the project together. The app is now working end-to-end. The team members will test out the app and make any fixes as needed. The team will also work on the Sprint #10 materials (demo, documentation, etc.) over the upcoming days.

4 REFERENCES

1. Patil, P. (2020, May 24). *Disease Symptom Prediction*. Kaggle. Retrieved March 6, 2022, from https://www.kaggle.com/itachi9604/disease-symptom-description-dataset