Project 2 Report

PeaceWise – a mental health chatbot

By Adithya Viswanathan (AXV190040)

ML/NLP Techniques

**Support Vector Classifier (SVC)**

One of the models I experimented with was SVC (Support Vector Classifier). Since this is a classification problem (classifying a user question to a category label), SVC would be one of the ideal models. Although SVC is usually for binary classification, it makes use of “one-versus-one” approach for multi-class classification. This is done behind-the-scenes by sklearn training a classifier for every pair of classes. For context, I split the dataset into questions as model input, and tag as model output. Details regarding dataset and its properties will be discussed in “Datasets and Cleaning” section. Then, I chose to train on 80% of data and test on 20%.

This model proved to be very successful, and here is an overview of the performance:

Best parameters: {'C': 4, 'gamma': 0.5, 'kernel': 'sigmoid'}

Accuracy: 0.923

Precision: 0.916

Recall: 0.923

F1-score: 0.912

Hamming Loss: 0.076

Jaccard Similarity: 0.894

Precision, recall, and f1 are standard metrics used for classification, and they represent the predictions of this multi-class classification problem. Nevertheless, they show that the model can predict labels very well. Hamming loss was a new metric which I thought would be useful. It essentially counts how many labels were incorrectly predicted. Since this is a loss metric, a low value is better, and the result shown is very good. Finally, I chose to include Jaccard Similarity, which is very similar to accuracy, but is seen apart since it looks for overlap in label sets, not just individual predictions. Therefore, it is more holistic than accuracy. Overall, the performance is very satisfactory.

**Multinomial Naïve Bayes (MNB)**

For a multi-class classification problem, this is one of the most common solutions. It performs well in large dimensional spaces, and is very efficient for text classification. It relies on the assumption that features (words in sentences) are conditionally independent given the class, which is a reasonable assumption for many text classification problems. It also works well in large datasets due to its favorable scaling properties. One thing I noticed was that the model trained very fast. For context, I split the dataset into questions as model input, and tag as model output. Then, I chose to train on 80% of data and test on 20%. Details regarding dataset and its properties will be discussed in “Datasets and Cleaning” section.

This model resulted in slightly better performance than SVC, which is why I ultimately chose it for my chatbot predictions. Here is an overview of the performance:

Best Parameters: {'alpha': 0.005, 'fit\_prior': True}

Accuracy: 0.949

Precision: 0.957

Recall: 0.949

F1-score: 0.945

Hamming Loss: 0.050

Jaccard Similarity 0.927

Compared to the previous SVC model, the Multinomial Naïve Bayes model performed a bit better. I chose to include both these models in my report as it shows the process I went through to find the optimal model. It is rare for an ML engineer to find the best model on the first try, so I wanted to incorporate different strategies to reveal the reality of building a project from start to finish. Hamming loss was lower than SVC, and Jaccard similarity was a bit higher. These models were trained on identical train sets as well as tested and evaluated on the same test sets. This ensured consistency in results and equal comparison of performance. Overall, the performance is very commendable. For clarity, all chatbot predictions are using this model.

**GridSearchCV**

In the journey to finding the best hyper-parameters for the models mentioned above, GridSearchCV was an enormous asset. It did an exhaustive search over all selected hyper-parameter values and found the best-performing set. The automated tuning process excellently optimized model performance and helped me achieve the best predictions for my data. The “best parameters” listed in the ML models above were derived from GridSearchCV.

**Tokenize sentences**

Prior to feeding in my dataset to the model for training, I iterated through each row and tokenized each sentence. This was done in order to help apply further transformations on the data. The following points will describe subsequent processing steps.

**Lower-case words**

In order to standardize the vocabulary, I lower-cased every word after tokenizing. This will aid in generalizing words that may differ in case due to grammatical reasons, like being the first word in a sentence. In summary, this was done to avoid treating two identical words as different (ex. “My” and “my”).

**Remove stopwords and punctuation**

Stopwords mostly do not contribute to the meaning of the text. Therefore, I removed them and left only the content words. This would help the model focus on the words which make the sentence representative of their topic or meaning. For example, “What is the cause of my stress?” will be reduced to [“cause”, “stress”].

**Lemmatization**

This step was done for a similar reason to lower-casing. This is to normalize words that look different but are indeed the same content word. It also reduces vocabulary size as a result, which helps model efficiency and generalization. For example, it would reduce “running”, “ran”, and “runs” to “run”.

**TF-IDF Vectorizer**

This step was performed in order for the training data to be interpreted by the model during training. TF-IDF vectorizer creates a document-term matrix. In a document-term matrix, each row represents a document, and each column represents a unique term (word) in the vocabulary. This helps the model learn the importance of each term in each sentence and find patterns in higher dimensional space. Although the resulting matrix is sparse for each instance, it lead to impressive results. The label values remain as text due to SVC and MNB’s capability to handle them.

Datasets and Cleaning

**Primary Dataset**

<https://www.kaggle.com/datasets/elvis23/mental-health-conversational-data>

My primary dataset was a mental health conversational Q/A dataset from Kaggle. This dataset contains 80 sets of Q/A, although its actual size was larger because each row actually contained multiple questions and answers. The “Reformatting Data” section talks about how I extended the effective size of this dataset. The features are “patterns” (or questions) and “responses”. The label is “tag”, also known as the type of question (ex. greeting, help, stress, etc.).

**Secondary Dataset**

<https://huggingface.co/datasets/heliosbrahma/mental_health_chatbot_dataset>

My secondary dataset was another mental health Q/A dataset, which I got from HuggingFace. It contained 172 sets of Q/A. My processing in order to bring this to the same structure as primary dataset is described in the “Reformatting Data” sub-section below. Even after adding the second dataset, which brought my total Q/A rows to 250, I felt my data was not big enough for a machine learning model to handle a conversation. Therefore, I performed the following.

**Data Augmentation**

Even after adding my secondary dataset, I felt the size of my dataset to be inadequate. This is when I noticed the questions and responses in my secondary dataset contained only one sentence. Also, some questions in my primary dataset were also lacking variety. This would make generalization tough for my model, as it would only accept the question valid if it appears exactly as it does in my dataset during training. Therefore, I decided to paraphrase all the questions in my secondary dataset and some in my primary dataset. This allowed the model to account for multiple variations in the user’s query for the same question. For example, if the training data had the question “What is mental health?”, I wanted it to also train on the questions “What is the definition of mental health?”, “Define mental health”, and “What constitutes mental health”. This is representative of user queries since each user might word their query differently, and it is my job as a model builder to account for variation in their language. I achieved this augmentation by utilizing a pretrained t5 model on HuggingFace, which takes a question as input, and outputs N paraphrased questions. Here is the [link](https://huggingface.co/mrm8488/t5-small-finetuned-quora-for-paraphrasing). One thing which I would try to improve in future work is the running time of this paraphrasing process. It scales linearly with the number of questions in the dataset, so running time will linearly grow with size of dataset. *Just to clarify, the pretrained model was used only to supplement my dataset from its initial state. This step was not part of my actual prediction task; hence why this is listed under datasets and cleaning topic.*

**Reformatting data**

Now it is time to tackle my primary dataset. I noticed that each row contained multiple questions and answers, so it would be necessary to extract questions into their own rows. This is done so the model only trains on one question and label at a time. The number of responses can still be numerous. I did this by creating a new row for each additional question that appears in a given row in my primary dataset. This resulted in approximately double the rows from the original primary dataset. For my secondary dataset, I had to extract the questions and answers, which were stored in the same column. I used regular expressions to separate the sentences into question and answer. Then, I applied the same transformation I made on my primary dataset; extract questions into their own rows so that each row contained only one question. Doing this to the primary and secondary dataset resulted in massive growth of the dataset size; I increased my dataset from 250 instances to nearly 1200.

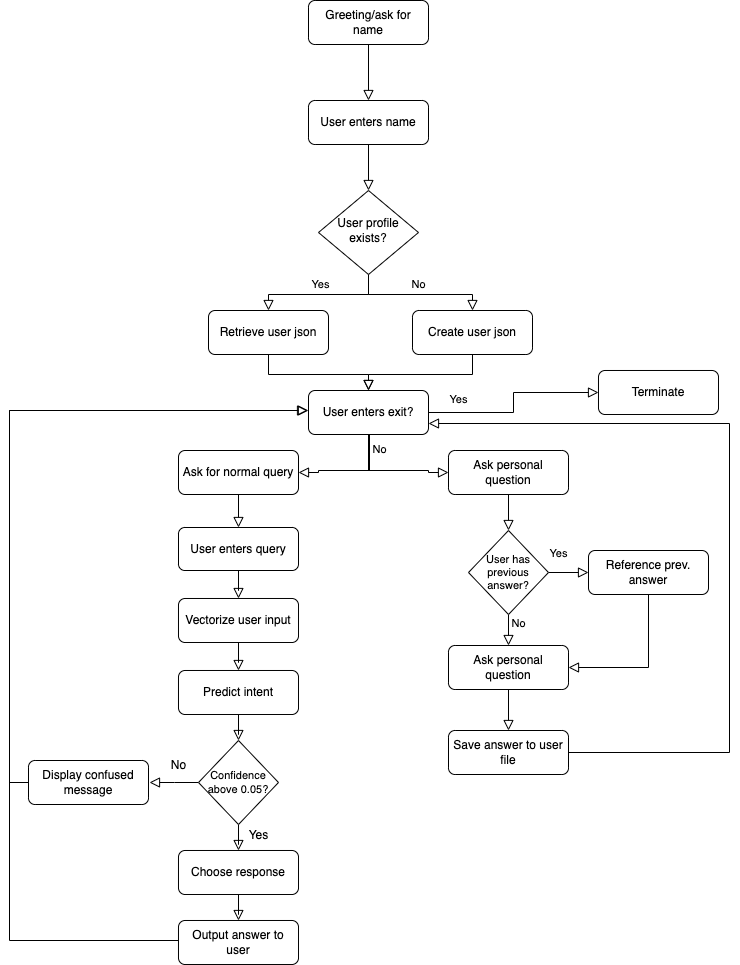
**Data Analysis**

The purpose of data analysis was to learn more about my data before diving into model building. First, I plotted the distribution of my tag column (class label). Specifically, I plotted the top 15 tags. My highest-occurring label was “greeting”. Due to my augmentation, certain Q/A about mental health made it into my distribution as well. Next, I plotted the distribution of the responses column. This was done to see which type of question had the most answers. It ended up being tags like “about”, “casual”, “stressed”, and “anxious”. The fact that mental health tags had a lot of responses was a sign of good training, as it would lead to varied answers to these common mental health questions from the user. Finally, I created a word cloud of the training data (questions) as it visually displays the most common words with relative size.

**Pre-processing**

Finally, in order to prepare my data for inputting into a model, split the data into X (training) and Y (label). X is the “pattern” column and Y is the “tag” column. Then, for each row, I tokenized, lowercased, and lemmatized each word, and removed stopwords. This lead my total training instances to be round 1200. As previously mentioned, I kept training and testing data constant so that future iterations and models would receive the same data for the purpose of equal comparison and analysis afterward.

Dialog Tree



Sample Dialogs

SAMPLE 1

*Welcome! My name is PeaceWise, your mental health assistant. What is your name?*

*> Adithya1*

*Hello Adithya1! Let's talk about mental health and how to keep your mind, body, and spirit peaceful.*

*Start chatting or enter 'quit' to exit the program*

*adithya1 > who are you?*

*PeaceWise > You can call me PeaceWise.*

*adithya1 > who created you?*

*PeaceWise > The real question is: Who created you?*

*adithya1 > I feel scared*

*PeaceWise > It'll all be okay. This feeling is only momentary.*

*adithya1 > how do I manage stress?*

*PeaceWise > Here are some personalized suggestions to help you cope with stress:*

*Just wondering adithya1, what are your interests?You previously said one of your likes was "playing with my dog". What else would you like to add on?*

*Thank you for your response!*

*adithya1 > i feel stressed*

*PeaceWise > Take a deep breath and gather your thoughts. Go take a walk if possible. Stay hydrated*

*Sorry for digress, but can you tell me about your hobbies or interests adithya1?You previously said one of your likes was "playing with my dog". What else would you like to add on?*

*Thank you for your response!*

*adithya1 > i feel ok*

*PeaceWise > Did something happen which made you feel this way?*

*adithya1 > i dont want to talk*

*PeaceWise > Alright no problem. Is there something you want to talk about?*

*adithya1 > no thank you*

*PeaceWise > My pleasure*

*Before we move on, do you mind giving me some feedback about how this conversation is going adithya1?You previously said your feedback was "keep trecking". What else would you like to add on?*

*Thank you for your response!*

*adithya1 > good night*

*PeaceWise > Good night. Sweet dreams.*

*adithya1 > bye*

*PeaceWise > Goodbye! If you ever feel like chatting again or need someone to talk to, don't hesitate to reach out.*

SAMPLE 2

*Welcome! My name is Pandora, your mental health assistant. What is your name?*

*> Monica*

*Hello Monica! I'm your mental health companion. What would you like to know about mental health?*

*Start chatting or enter 'quit' to exit the program*

*monica > Who are you?*

*PeaceWise > Hello there! I am your empathetic Mental Health conversational chatbot, designed to provide personalized suggestions and support. My purpose is to lend a listening ear, offer guidance, and assist you in navigating through any mental health challenges you might be facing. Feel free to share your thoughts and concerns, and I'm here to help you to the best of my abilities. Remember, you are not alone, and I'm here to support you throughout this journey.*

*monica > Who made you?*

*PeaceWise > I was created by Adithya Viswanathan.*

*monica > How are you?*

*PeaceWise > Hello! I am good. How are you feeling today?*

*monica > I am feeling so lonely*

*PeaceWise > Why do you think you feel this way?*

*monica > I feel so worthless*

*PeaceWise > It's only natural to feel this way. Tell me more. What else is on your mind?*

*monica > My dad passed away*

*PeaceWise > I'm sorry to hear that. If you want to talk about it. I'm here.*

*Just wondering monica, what are your interests?*

*Thank you for your response!*

*Before we dive deeper into the previous topic, I wanted to ask about your preferences. Are there any things that you'd rather pass on or avoid?*

*Thank you for your response!*

*monica > Can people with mental illness recover?*

*PeaceWise > When healing from mental illness, early identification and treatment are of vital importance. Based on the nature of the illness, there are a range of effective treatments available. For any type of treatment, it is essential that the person affected is proactive and fully engaged in their own recovery process.*

*Before I forget, I'm curious to know about any activities, foods, or preferences you don't like?You previously said one of your dislikes was "I try to avoid city driving due to traffic". What else would you like to add on?*

*Thank you for your response!*

*monica > Where can I find a support group*

*PeaceWise > There are several places you can explore to find a support group that suits your needs:*

*monica > Thank you for the help*

*PeaceWise > Happy to help!*

*Before we move on, do you mind giving me some feedback about how this conversation is going monica?*

*Thank you for your response!*

*monica > See you later*

*PeaceWise > See you later.*

An appendix for sample user models that were created

|  |  |
| --- | --- |
| A screenshot of a computer program  Description automatically generated | A screenshot of a computer program  Description automatically generated |

Sample user files: abraham.json and monica.json

“Abraham” and “Monica” refers to the name that the user entered when program prompted them in the beginning. This file holds the following fields and information:

* Name: user’s chosen name
* Questions: questions that user has asked
* Likes: information that user has provided about their interests
* Dislikes: information that user has provided about their disinterests
* Feedback: information that user has provided about their experience with chatbot (can be positive or negative comments)

Fields start as empty lists, and can grow according to what questions are asked by user.

Evaluations of the chatbot and analysis of its strengths and weaknesses

Strengths:

* Model can predict well even when query is slightly different than what it has been trained on, and the answer is correct and very relevant
* Answers are coherent and understandable
* Able to handle different examples of small talk and casual talk (greetings, expressing feelings, etc.)
* Response time from user entering query to retrieving answer is almost instantaneous
* Saving user’s information is flawless.
* Getting user’s previous feedback, likes, and dislike answers and asking new question is working well.

Weaknesses:

* Can be more personalized. Currently relying on few versions of greetings, confusion, and personal questions.
* Does not build upon previous mental health queries.
* Unfamiliar questions can be handled better.

Likert-Style Feedback

1. On a scale of 1 to 5, how satisfied are you with the responsiveness of the chatbot?

1 - Very Dissatisfied | 2 – Dissatisfied | 3 - Neutral | 4 – Satisfied | 5 - Very Satisfied

1. How helpful did you find the chatbot's responses?

1 - Not Helpful at All | 2 - Slightly Helpful | 3 - Moderately Helpful | 4 – Helpful | 5 - Extremely Helpful

1. To what extent did the chatbot understand your queries?

1 - Didn't Understand at All | 2 - Partially Understood | 3 - Understood Somewhat | 4 - Mostly Understood | 5 - Completely Understood

1. How satisfied are you with the accuracy of the information provided by the chatbot?

1 - Very Inaccurate | 2 - Somewhat Inaccurate | 3 – Neutral | 4 - Somewhat Accurate | 5 - Very Accurate

Responses:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Person | Question 1 | Question 2 | Question 3 | Question 4 |
| Chandra, family friend | 5 | 4 | 4 | 3 |
| Vishy, father | 4 | 4 | 3 | 4 |
| Aparna, sister | 5 | 4 | 4 | 4 |