NLP Project – Mental Health Text Classifier

Author: Marinescu Alexandru

Group: 407

Table of Contents:

1. Introduction

This is the documentation for a mental health text classifier, one the available topics for the main project for Foundations of NLP.

The purpose is decide if the topic of a given text is related to mental health issues and if this is the case, decide what type of issue if is, from a given span of labels. The resulting models would then be incorporated in a simple API for easier usage.

Example:

Input	Result
A sandwich and french bread sit on a cutting board with an apple. I should prepare the breakfast.	The sentence **wouldn't be related to mental issues**.
After last nights social events, I've decided to buy a stash from them, because life is too stressful.	The sentence is probably **related to drug abuse.**

I decided to choose this subject because mental health is very important and affects our everyday lives. I consider that such a classifier could be helpful in analyzing textual data from different social platforms like X, Facebook, Reddit and identify potential trends and changes in them in order for us to adapt. For example, therapists could try to use such a tool to update their knowledge about different trends and have a stronger base opinion just from a patient's textual data.

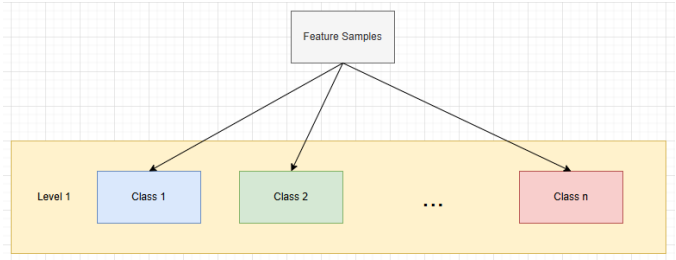
2. Related Work

Before stating any technical work, here is a more extreme, but good usage: **preventing terrorism by analyzing social media data by correlation.**
One such example is the *The Christchurch Attack (New Zealand, 2019).* I will leave it to the reader to delve deeper into the subject. The main points are:

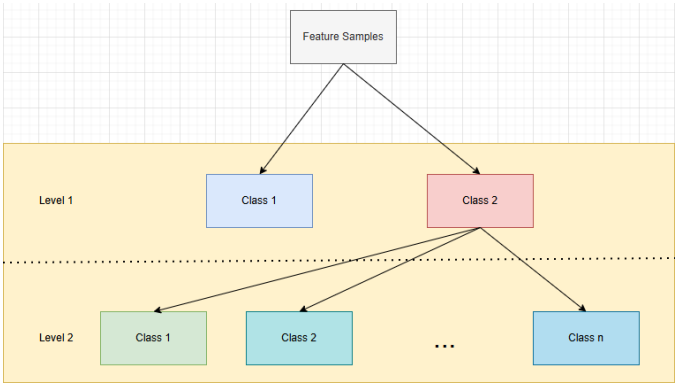
- Terrorist was a 28-year-old Australian man with far-right extremist views.
- Terrorist live-streamed the attack on **Facebook** using a helmet-mounted camera.
- Terrorist posted a **74-page manifesto** with far-right many ideologies on **8chan** (an imageboard website known for hosting extremist content) just before the attack.

Unfortunately, the attack wasn't prevented. Afterwards, a major effort was launched to analyze social media behavior and online extremism. Organizations, such as the Global Internet Forum to Counter Terrorism (GIFCT), have since collaborated with platforms like Facebook, Google, and Twitter.

Coming back to the more technical side, probably the closest piece of work would be [4] **Multi-Class Multi-Level Classification of Mental Health Disorders on Textual Data from Social Media.** The dataset is based on Reddit data categorized on 6 classes, 5 of which are mental illnesses. What is to be noted is the strategy used, where instead of immediately classifying the data for the final result:



The paper follows to decide if the text is first, about a mental illness and then classify it:

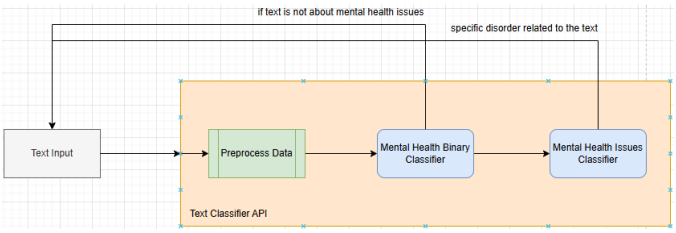


In [5] **Text-Based Data Analysis for Mental Health Using Explainable AI and Deep Learning**, another similar paper, the idea of custom tools for data scraping should be noted. The research follows 2 methods: *deep learning* and *machine learning for text classification* also on 6 classes, 5 of which are mental illnesses. Another aspect to be noted is the use of pretrained models, such as Word2Vec, to extract embeddings from the collected samples.

There are other aspects such as demographics, or language specific expressions that can highly affect one such research.

3. Method

The main idea behind the project is to classify data based on mental issues classes. However, because it was started quite late, there was no time to ask permission for more complex datasets. So, in order to leverage the open source available datasets, the following strategy was used:



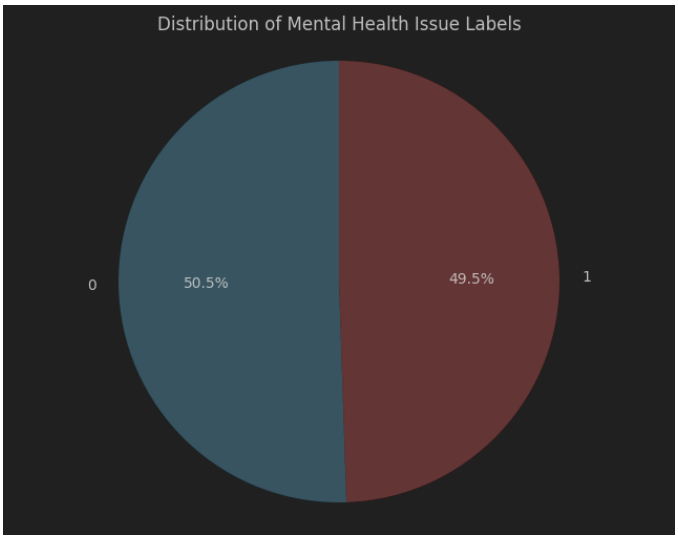
The strategy was used to take advantage of the small size available datasets because it will isolate the context from the non-mental health related texts and hopefully offer more accuracy.

Dataset

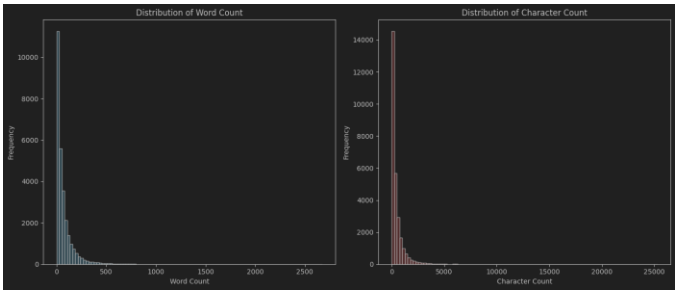
For the binary class (mental issues text vs normal text), the following dataset was used: https://www.kaggle.com/datasets/reihanenamdari/mental-health-corpus/data, which looks like this:

text	label
dear american teens question dutch per…	0
im done trying feel betterthe reason im …	1

While th textual data has spelling/spacing mistakes, it is rich in samples (27972), and well balanced:



However, there is the potential for outliers, since most samples have a word count below 300, with some exceeding this. This is a histogram with bin size of 100, for samples word and character count:



The dataset is mainly in English, but as mentioned before, other aspects, such as demographics, or culture could make this a very specific/biased set.

By curiosity, here are the most common words for each label and the 10 most common PoS (using nltk implementations for tokenization and pos tagging):

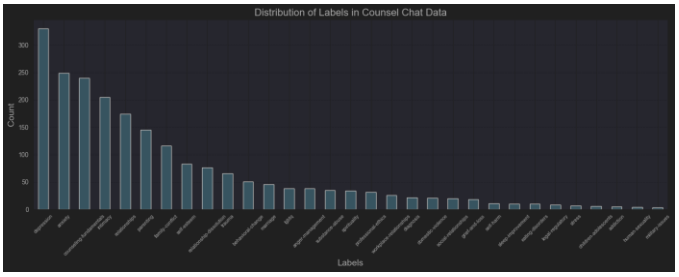
For the multi-class mental issues classification, 2 datasets were used:

- https://github.com/nbertagnolli/counsel-chat/tree/master/data
- https://paperswithcode.com/dataset/dreaddit

The *counsel chat* data was already split into train and test samples, but didn't use the original split. It contains 10 columns:

questionID	A unique question id which is distinct for every question
questionTitle	The title of the question on counsel chat
questionText	The body of the individual's question to counselors
questionLink	A URL to the last location of that question (might not be active)
topic	The topic the question was listed
therapistInfo	The summary of each therapist, usually a name and specialty
therapistURL	A link to the therapist's bio on counselchat
answerText	The therapist response to the question
upvotes	The number of upvotes the answerText received
split	The data split for training, validation, and testing

The text data is in english language with a few spelling mistakes. Just as important, it has 31 labels (disorders) and it's not well balanced:

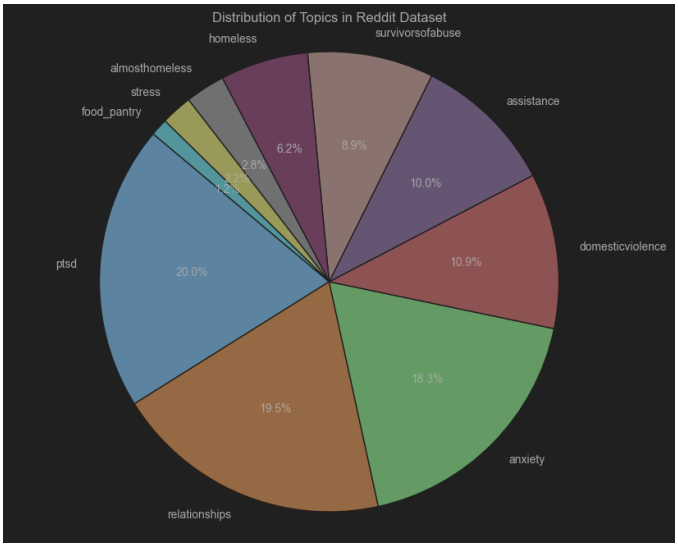


The Dreddit set consists of posts from five different categories of Reddit communities under subreddits related to mental health issues. Example:

subreddit	text
ptsd	He said he had not felt that way...
assistance	Hey there r/assistance, Not sure if this is th...
relationships	until i met my new boyfriend, he is amazing, h...

The text data is in english language and has multiple columns, but these 2 are most important.

If we consider the *subreddit* column as labels, it has 10 labels and it's also not balanced between them:



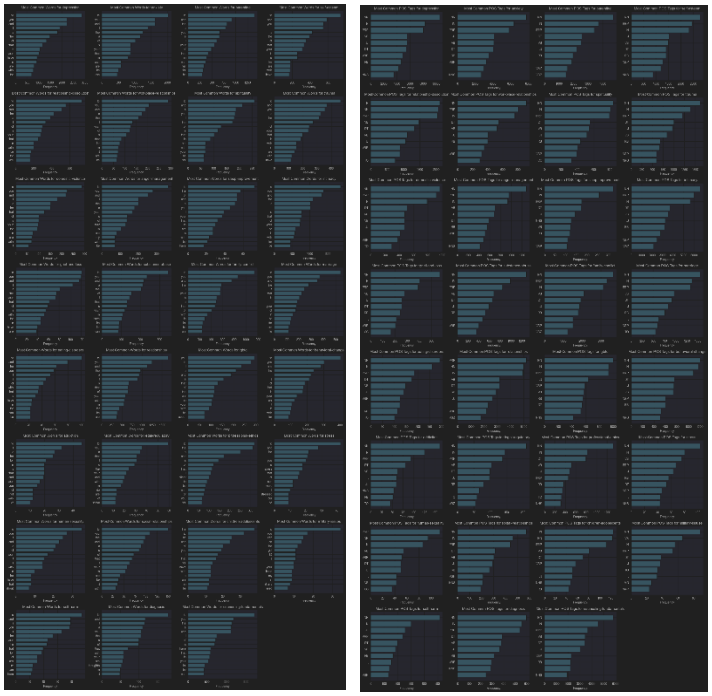
Preprocessing

We'll begin with the particular preprocessing parts, and we'll end with the common preprocessing part.

For the counsel chat data, in order to get the right we context, *questionText* and *answerText* columns were combined under one ***combinedText*** column and the *topic* was also kept.

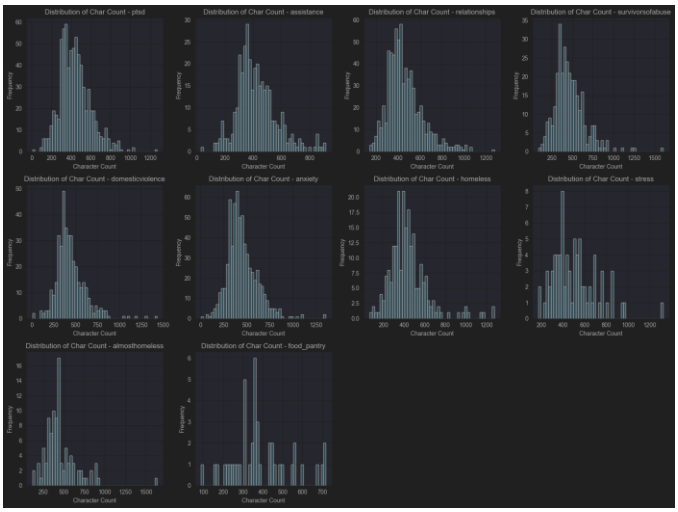
On this, more exploration was done to see word and character count for each label. Below are the histograms for character count for each disorder. For the more plentiful sets, most range around the 1000-2000 character mark, which doesn't indicate long detailed responses.

Similarly as before, here are the most common word for each label and the 10 most common PoS:



A more technical person in social analysis would be able to analyze and find the correlation between the use of certain words and the disorder domain. Most of them look very similar.

The same statistics were also obtained for the reddit dataset:



Character count per post, which revolves around the 500 value, quite standard for a media post.

And these are the most common word for each label and the 10 most common PoS. This shows the similarities between the 2 datasets. This is what allowed the creation of a **combined dataset**, where

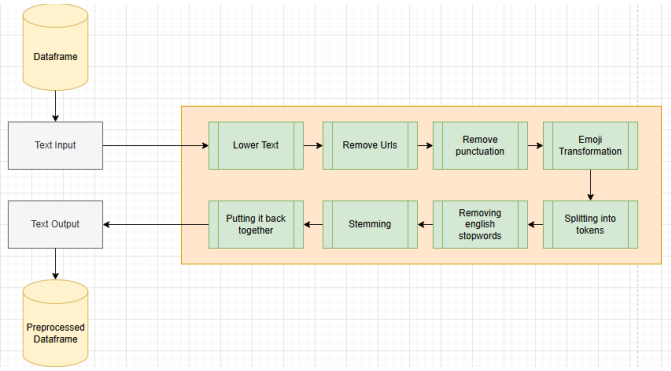
1. For a common label, feature rows were added from both sets
2. Otherwise, add from the specific dataset
3. Keep the 10 most populated labels and their feature data

Those labels are: ['anxiety', 'relationships', 'ptsd', 'domesticviolence', 'assistance', 'depression', 'survivorsofabuse', 'counseling-fundamentals', 'homeless', 'intimacy'].

This can give us 3 dataset options for multiclass analysis, which is a big advantage given the small original datasets.

Now, onto the preprocessing steps common to all datasets, we have:

1. Lower casing the text
2. Removing urls
3. Removing punctuation
4. Emoji transformation
5. Splitting into tokens (words in this case)
6. Removing english stopwords
7. Stemming
8. Putting it back together



This pipeline was applied to all datasets, for both binary and multiclass cases.

Afterwards, data was split into training and testing samples with a test_size = 0.2 of the original data. The labels were also encoded, for better training.

Methods

Neural Network

We'll start with the ***Mental Health Binary Classifier*** (if text is, or isn't related to mental issues).

The training and testing data was tokenized to sequences using *Tokenizer* provided by keras API, and then sequences were padded to reach same dimension. A vocabulary size of 47107 was obtained, which is quite large for the problem.

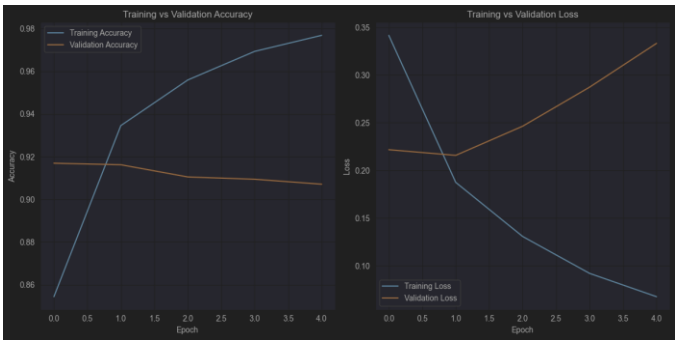
Then, a **convolutional neural network** was used. More designs were utilized, but the overall process wasn't complicated, thanks to the well structured data. This is the final summary:

Layer (type)	Output Shape	Param #
embedding_4 (Embedding)	(None, 25000, 100)	4710700
conv1d_12 (Conv1D)	(None, 24999, 4)	804
max_pooling1d_12 (MaxPooling1D)	(None, 12499, 4)	0
conv1d_13 (Conv1D)	(None, 12498, 8)	72
max_pooling1d_13 (MaxPooling1D)	(None, 6249, 8)	0
conv1d_14 (Conv1D)	(None, 6248, 16)	272
max_pooling1d_14 (MaxPooling1D)	(None, 3124, 16)	0
flatten_4 (Flatten)	(None, 49984)	0
dense_8 (Dense)	(None, 32)	1599520
dropout_3 (Dropout)	(None, 32)	0
dense_9 (Dense)	(None, 1)	33

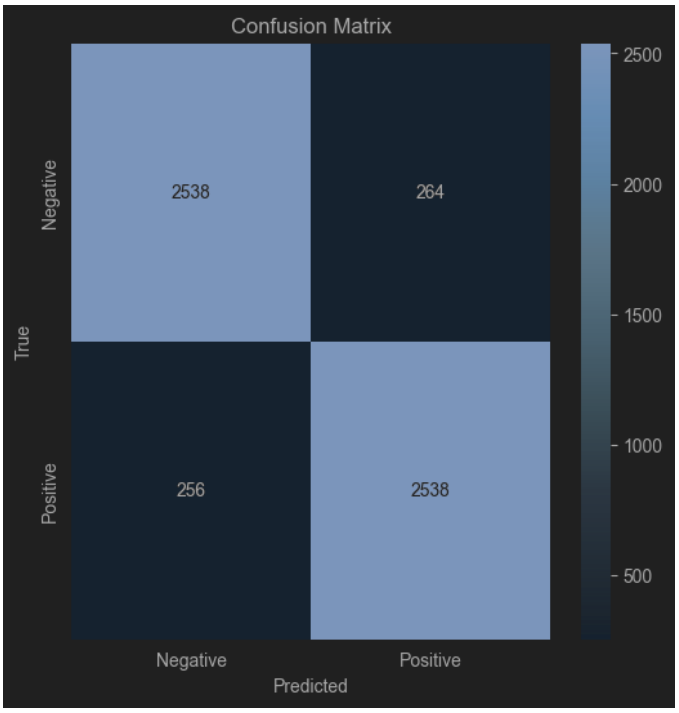
This is a simple model, as it was run on a local machine, but the architecture is quite known.

ADAM was the choice for optimizer and *accuracy* as the main metric. Because of other past attempts and overfitting, a scheduler for the learning was introduced. I hoped that gradually reducing the learning rate as training progressed would ensure the stability in the accuracy (it did, but not much).

This reached ~91% validation accuracy, just from 5 epochs. Learning curves, are not the best however:



And this is the confusion matrix for the cnn model predictions on testing set:



Next, let's continue with ***Mental Health Multiclass Classification***. Multiple attempts have been tried on all 3 datasets

- counsel chat dataset – 10 classes
- reddit dataset
- combined dataset (details in *Preprocessing*)

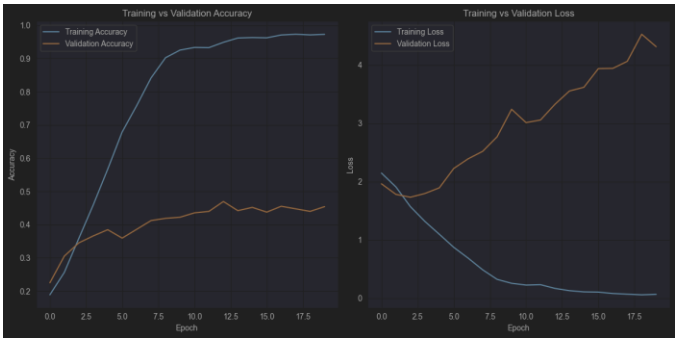
Best results were obtained using the combined dataset. Despite this, the data still wasn't balanced, so **class weights** were calculated using *scikit* implementation and added to the training pipeline of the **convolutional neural network**. More structures were tried, but this had the best result:

Layer (type)	**Output Shape**	**Param #**
embedding_4 (Embedding)	(None, 2500, 128)	1545728
conv1d_12 (Conv1D)	(None, 2496, 64)	41024
max_pooling1d_12 (MaxPooling1D)	(None, 1248, 64)	0
lstm_10 (LSTM)	(None, 256)	328704
dense_23 (Dense)	(None, 64)	16448
dropout_12 (Dropout)	(None, 64)	0
dense_24 (Dense)	(None, 10)	650

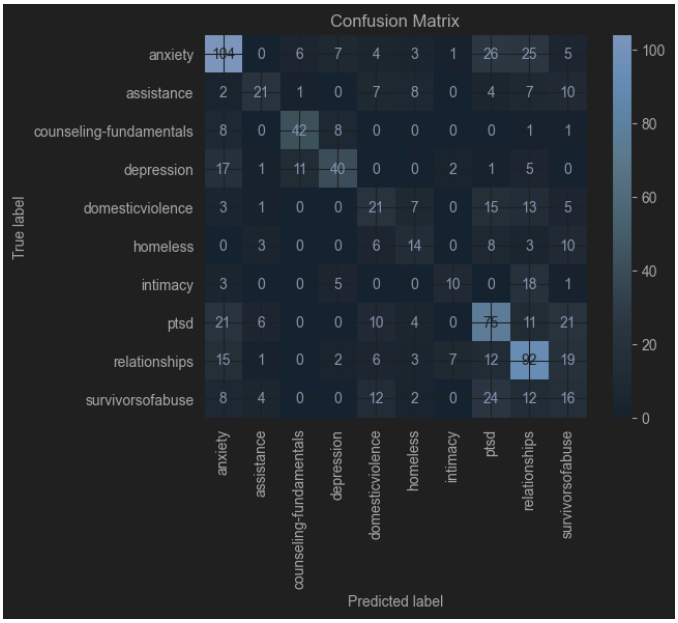
Originally, this started from a cnn, but it was poised with too much overfitting, so changes were applied

LSTM (Long-Short Term Memory) layer had the hope to extract the dependencies not just between the words next to each other, but from the whole context. And *Bidirectional* layer was also tried added to do this both for past and future tokens (hopefully), but it lowered the validation accuracy, so was removed.

The model reached ~46% validation accuracy from 20 epochs. It overfitted significantly (~91% training accuracy), but we'll discuss this in a further section. These are the learning curves:



And this is the confusion matrix for the predictions done on the testing set:



You can see that it does well on some classes and worse on others. There is a reason for this, but it will be discussed later.

Machine Learning Methods

You guessed it, we'll start with the ***Mental Health Binary Classifier***. CNN model was quite successful, but I decided to attempt some machine learning methods as well to see how they compare and in the end, choose the best.

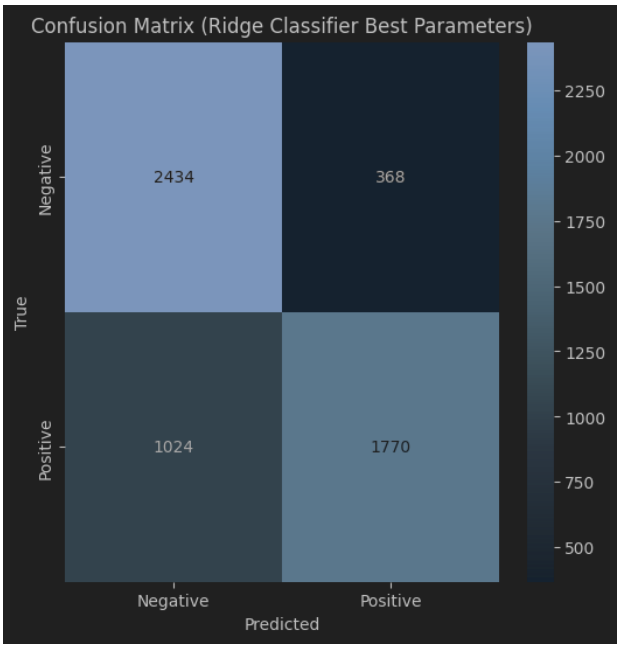
However, the sequences used earlier as tensors won't be useful here. For this, *TfidfVectorizer* from **sklearn** was used to vectorize the initial data. This is particularly useful for handling spelling mistakes, because of a score of IDF, which will give different scores for mistakes as opposed to BoW for example

1.1 Ridge Classifier

A Ridge Classifier is a linear classification model that applies **L2 regularization** (large difference in coefficients will be penalized more – in theory).

Since α (alpha) is the regularization strength (like a weight coefficient in the sum from predictive function), *GridSearchCV* method was used with:

- *RidgeClassifier* from **sklearn** as the model
- 'alpha': [5, 10, 15, 20, 50, 100]
- 'max_iter': [50, 100, 200]
- 'accuracy' as scoring
- the vectorized training data features for fit



The best alpha was 5, and max_iter 50, with the cross-validation accuracy was 0.8961, but the accuracy on testing data predictions was 0.7513.

1.2 Random Forest

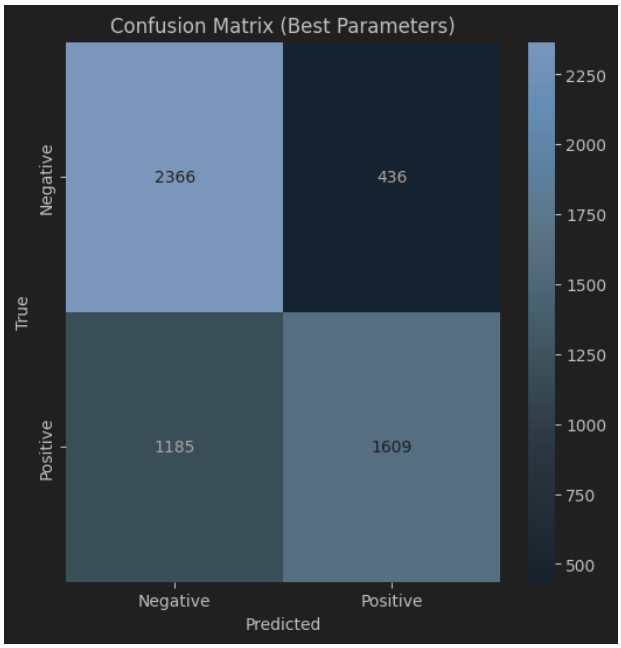
This a learning algorithm that generates multiple decision trees during training and can combine them. The implementation from **sklearn** was used (*RandomForestClassifier*).

Question would be: how many trees (estimators) should be used? Too few would miss features, but too many may include unnecessary features that can add noise.

For this *GridSearchCV* method from **sklearn** was used. This will try all possible combinations of given hyperparameters and does cross-validation for performance between each combination. For this specific search, the following setup was used:

- RandomForestClassifier as the model
- 'n_estimators': [200, 250, 300]
- 'accuracy' as scoring
- the vectorized training data features for fit

The best parameter 300, the cross-validation accuracy was 0.8835, but the accuracy on testing data predictions was 0.7103.



Next, we'll continue with the machine learning methods for ***Mental Health Multiclass Classifier.***

2.1 Logistic Regression

Despite the name, this models the probability of a given input to belong to a specific class by applying a softmax function to the weighted sum of inputs. Ridge classifier is a similar case as well.

Since α (alpha) is the regularization strength (like a weight coefficient in the sum from predictive function), *GridSearchCV* method was used with:

- *LogisticRegression* from **sklearn** as model
- 'C': [10, 15, 30]
- 'max_iter': [10, 20, 30]
- 'accuracy' as scoring
- the vectorized training data features for fit

The best C was 10, and max_iter 10, with the cross-validation accuracy was 0.5825, with a similar accuracy on testing data predictions: 0.56 – 0.57.

2.2 Support Vector Machine (SVM)

It tries to find a hyperplane that best separates the data into classes. SVMs aim to maximize the margin between the hyperplane and the nearest data points from any class. It worked well on a previous project, so I decided to use it here too.

In order to avoid noisy data, slack variables were introduced and we can control this level of regularization using another parameter – C. *GridSearchCV* method was used with:

- *SVC* from **sklearn** as the model
- 'C': [0.1, 1, 10]
- 'kenrel': ['linear', 'rbf']
- 'accuracy' as scoring
- the vectorized training data features for fit

The best C was 1 with a linear kernel and with the cross-validation accuracy was 0.58, with a similar accuracy on testing data predictions of 0.6.

2.3 Random Forest

Used for the same reason as in the binary classification. For this *GridSearchCV* method from **sklearn** was used with:

- *RandomForestClassifier* as the model
- 'n_estimators': [300, 400, 500]
- 'min_samples_split': [4, 5, 6]
- 'accuracy' as scoring
- the vectorized training data features for fit

The best parameter 500, with a minimum number of children per tree of 4. The cross-validation accuracy was 0.5513, with a similar accuracy on testing data predictions of 0.55.

2.4 Ridge Classifier

Used for the same reason as in the binary classification. For this *GridSearchCV* method from **sklearn** was used with:

- *RidgeClassifier* from **sklearn** as the model
- 'alpha': [0.5, 1, 2, 3, 4, 5]
- 'accuracy' as scoring
- the vectorized training data features for fit

The result of the search was 2. The cross-validation accuracy was 0.5805, with a similar accuracy on testing data predictions of 0.59.

Evaluation

For the ***Mental Health Binary Classifier***, we have the following scores:

Method	Accuracy Score
Neural Network	0.91
Random Forest	0.71
Ridge Regression	0.75

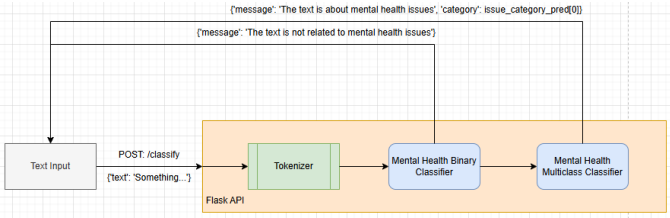
There multiple things to take into consideration, such as inference time and complexity, but for the final implementation, the CNN was chosen for the binary classifier.

For the ***Mental Health Multiclass Classifier***, we have the following scores:

Method	Accuracy Score
Neural Network	0.46
Logistic Regression	0.57
SVM Classifier	0.60
Random Forest	0.55
Ridge Regression	0.59

Yes, despite the fact that SVM had a really high training time (has O(d^2) time complexity), it was the best and inference time is linear which isn't completely bad. This is the choice for this case.

Both of these models have been integrated in ***Flask*** API (just for simple http requests), and exposed on a post endpoint which accepts a text payload. The final flow looks like this:



4. Conclusion

Possible Improvements

We can safely say that most issues were related to the ***Mental Health Multiclass Classifier***. But why?

One reason is the actual data, which had spelling mistakes, was unbalanced, and for the counsel set, 21 out of 31 topics were eliminated along with their data. For spelling, there are spell checkers already implemented for english language, such as **Speller** from *autocorrect*. Should have realized it a bit sooner. For the unbalance in the datasets, some data augmentation techniques could have been attempted, like ***back translation***, or ***synonym replacement***. They depend on the specific scenario.

You might have noticed that the CNN model did poorly in training and evaluation. The chosen architectures aren't necessarily bad, but what impacted the results the most is the embedding of the data. If data features were extracted and represented in a better way, it would have been better. On this case, a more proficient model could be used to extract the embeddings and use those in neural network (this is called *feature extraction*). For example, **BERT** (Bidirectional Encoder Representations from Transformers), that's based on the transformer architecture, could be used to extract those embeddings.

In the end, a personal note of improvement would be the use of utility modules/libraries. I did the research using jupyter notebooks and the API lies in a python file. Throughout those notebooks is a lot of repeated code that I could be extracted in a notebook and reused.

In the future, given some spare time, I'd like to attempt to create an extraction pipeline to establish a more detailed and clean dataset, for this problem. I understand the ethical limitations, but I know that there are some out there which were built in a similar manner (ex: Meta).

Personal Notes

The task was challenging and the mistake was choosing to start it quite late. Despite all of this, I enjoyed it, not only because of the theme, but also because I learned many aspects, such as:

- TF-IDF - Term Frequency Technique
- How punishing data representation can be
- Interpreting scenarios using PoS and most common words
- Classification models

Would have liked to delve deeper into transformer architecture, but it's my fault for starting late. Otherwise, I appreciate the opportunity given by this project.

Limitations

Starting with the actual data, the main limitation is the availability of data in this domain. Most of the suggested resources require approval from authors which could take some time. The open source datasets might not be that well-structured, or they are quite rare for various mental issues.

Last limitation is the actual hardware. I could have used a Google Colab Notebook for this, but apart from the tensorflow related tools, the rest of the model training for machine learning run on CPU. I attempted to use feature extraction using a variant of BERT (bert-base-uncased), and I had to get the embeddings in batches, because there wasn't enough memory to load more at once. It could also be my limited knowledge in using the Colab Notebook and there are more settings to search.

Ethical Statement

Given the theme of the project, questions such as: "what unethical cases could there be?" may arise.

For example, some people could use a similar tool to discriminate against individuals, such as denying them job opportunities, housing, or insurance based on inferred mental health conditions. This is not to be desired. Another case could be to spread misinformation about mental health issues or exploit vulnerable individuals by recommending harmful treatment.

"was any bias introduced in this project?"

Given the datasets used, this would be the case, because all data is english and most of them come from posts from social media, which is a volatile environment, but highly dependent on the user's culture, history, etc. Moreover, only 10 classes were used based on some features, so the model might oversimplify complex mental health issues, reducing nuanced conditions to single categories.

5. Bilbiography

1. Bickert, M., & Fishman, B. (2017, June 15). *How we counter terrorism*. Meta. Hard Questions blog series https://about.fb.com/news/2017/06/how-we-counter-terrorism/

2. Wikipedia contributor. (2019, March 15). *Christchurch mosque shootings*. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Christchurch_mosque_shootings

3. Madan, Rohit. "TF-IDF/Term Frequency Technique: Easiest Explanation for Text Classification in NLP Using Python (Chatbot Training on Words)." *Analytics Vidhya*, 30 May 2019, https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3.

4. Sutranggono, A. N., Sarno, R., & Ghozali, I. (2024). Multi-Class Multi-Level Classification of Mental Health Disorders Based on Textual Data from Social Media. *Journal of Information and Communication Technology, 23*(1), 77–104. https://doi.org/10.32890/jict2024.23.1.4

5. Namdari, R. (n.d.). *Mental Health Corpus: Labeled sentences about depression and anxiety* [Dataset]. Kaggle. Retrieved from https://www.kaggle.com/datasets/reihanenamdari/mental-health-corpus/data

6. Bertagnolli, N. (2020, May 11). *Counsel Chat: Bootstrapping High-Quality Therapy Data*. Towards Data Science. Retrieved from https://towardsdatascience.com/counsel-chat-bootstrapping-high-quality-therapy-data-971b419f33da

7. Turcan, E., & McKeown, K. (2019). *Dreaddit: A Reddit Dataset for Stress Analysis in Social Media*. Retrieved from https://paperswithcode.com/dataset/dreaddit

8. Madan, R. (2019, May 30). *TF-IDF/Term Frequency Technique: Easiest explanation for Text classification in NLP using Python (Chatbot training on words).* Analytics Vidhya. Retrieved from https://medium.com/analytics-vidhya/tf-idf-term-frequency-technique-easiest-explanation-for-text-classification-in-nlp-with-code-8ca3912e58c3

9. Olah, C. (2015, August 27). *Understanding LSTM Networks*. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

10. Liviu P. Dinu (2024–2025). *Introduction of NLP*. Faculty of Mathematics and Computer Science, University of Bucharest., All courses