Dataset choice

For the assignment, I selected the **`Mental Health  Dataset`**. This dataset contained a collection of real-world tweets, each labeled with a mental health label. The biomedical dataset consisted of excerpts from biomedical research papers and only 3 classes. The data consisted of explicit diagnostic labels linked to specific organs. Since the mental health dataset consisted of user-generated input and usually an absence of explicit medical terms, I found it, personally, more interesting to train.

Data exploration and Insights

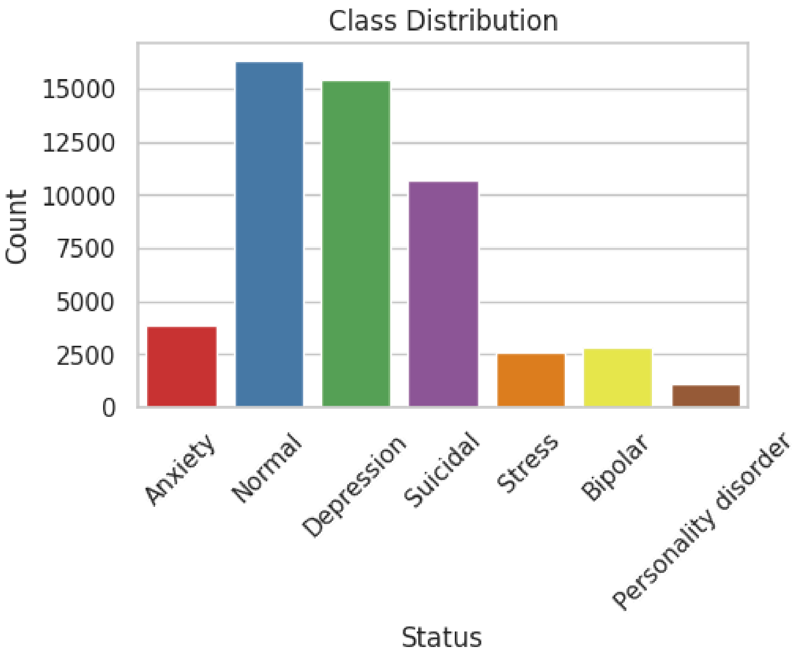
I began by conducting some Exploratory Data Analysis (EDA) . The original dataset contained **53,043 entries**. After removing rows with missing values, this was reduced to **52,681**, which was large enough to train LSTM and BERT models.They appeared to be random across classes. Looking at some examples, The normal labeled tweets were usually about a variety of personal topics, however tweets with mental health labels usually had descriptions of symptoms or negative experiences such as “I have a desk job, I can't sit still, I can't even stand still” or “I try so hard to be happy”,

Class Distribution

The target variable `status` includes 7 classes with the following distribution:

Status	**Count**	**Percentage**
Normal	16,343	31.02%
Depression	15,404	29.24%
Suicidal	10,652	20.22%
Anxiety	3,841	7.29%
Bipolar	2,777	5.27%
Stress	2,587	4.91%

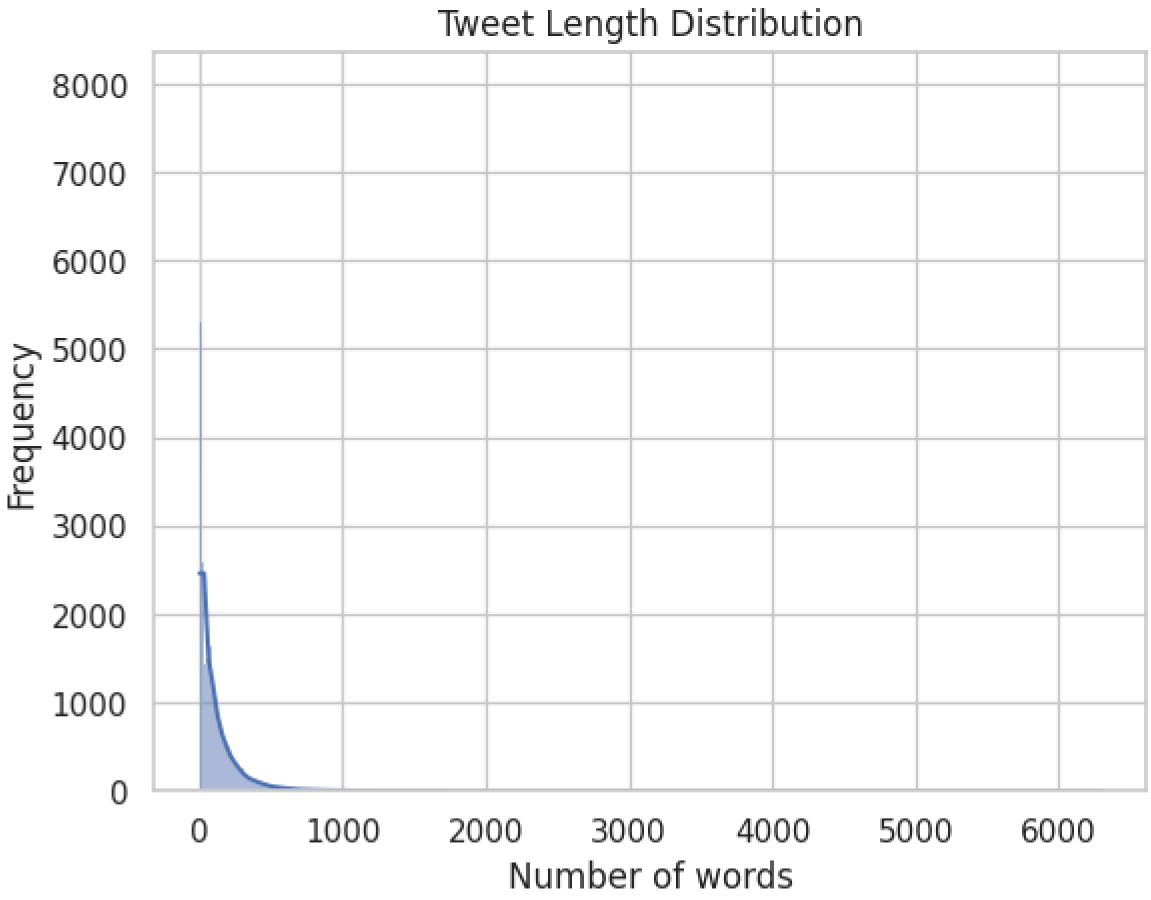
Status	Count	Percentage
Personality disorder	1,077	2.04



There was significant class imbalance in the data, with the top three class(Normal, Depression and Suicidal), accounting for 80% of the distribution and personality disorder only having about 1000 examples.

Text length analysis:

The median tweet was 62 words, however the distribution of this was heavily skewed with an extremely long tail with the largest tweet having 6300 words. To avoid extremely long sequences that slowed down training, especially for LSTM-based models, I chose to exclude tweets longer than 1000 words in my dataset. The length of tweets also differed by class. What was most notable was that tweets labelled normal were significantly smaller than those classified a mental health condition. And of the extreme outliers excluded, Most were labelled **`depressed`** or **`anxiety`**. About 80% of the tweets were less than 150 words, which I later used for setting my parameters.

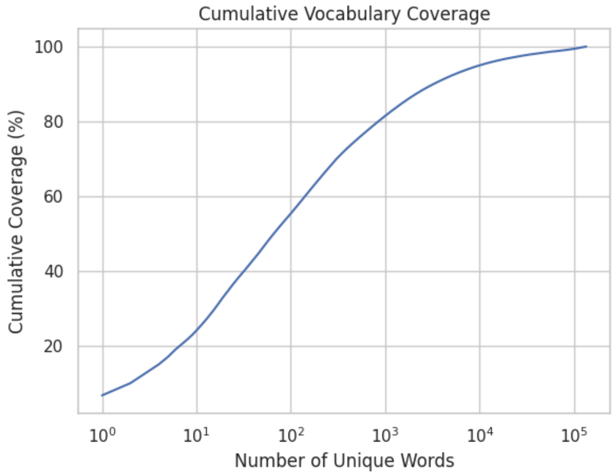


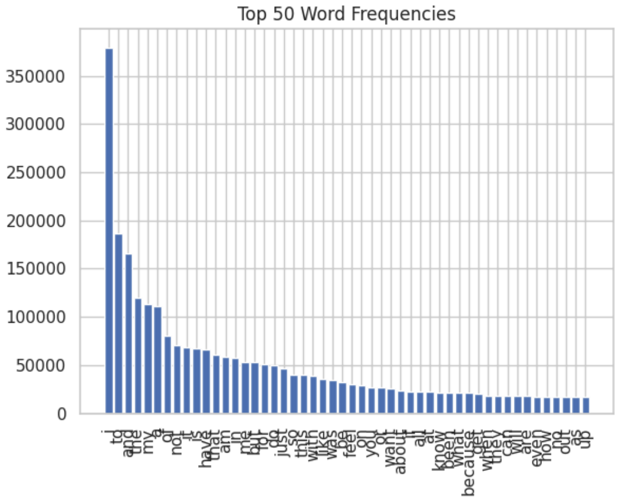
Vocabulary Analysis

To better understand the linguistic diversity of the dataset, I analyzed the vocabulary coverage across different word set sizes.

```
Total unique words: 134519
Total word count: 5666891

Coverage at different vocabulary sizes:
1000 words cover 81.45% of the corpus
5000 words cover 92.19% of the corpus
10000 words cover 94.97% of the corpus
```

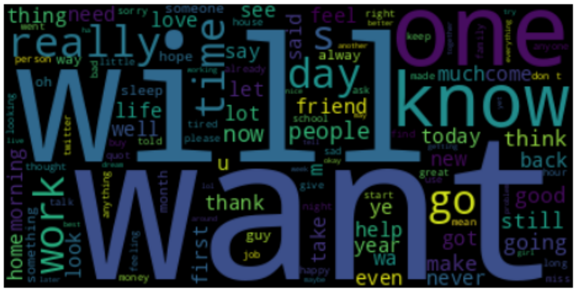




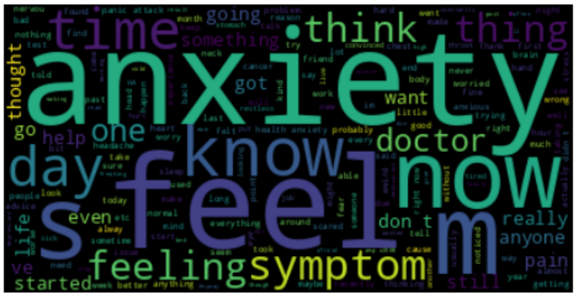
Word Frequency

To identify patterns for each class, I looked at the frequency at which certain words were being used in different classes.

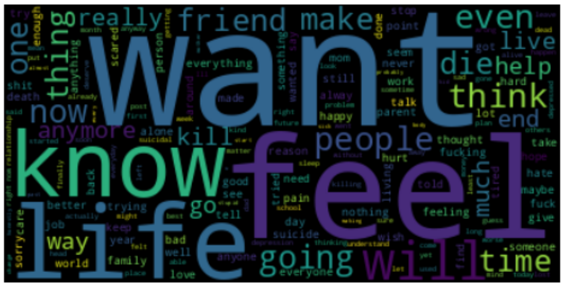
Word Cloud for Normal



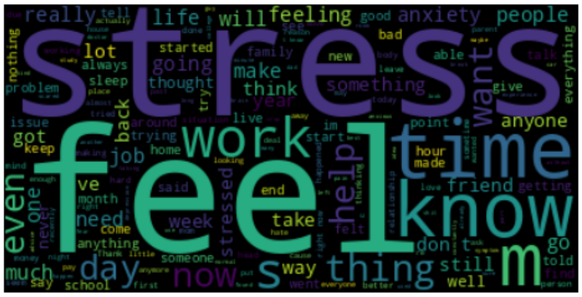
Word Cloud for Anxiety

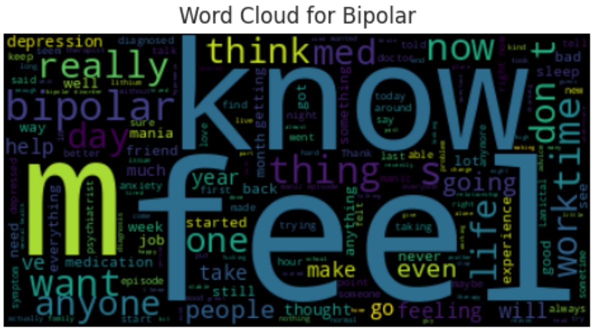


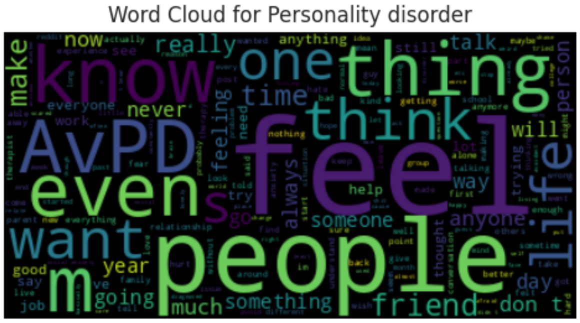
Word Cloud for Suicidal



Word Cloud for Stress







Some words like feel and like were common across classes. However, some words were unique to their class such as “anxiety” for anxiety , and “cannot” and “life” for suicidal.

Preprocessing

LSTM

For my LSTM models i underwent heavy preprocessing of the tweets, these include:

1. Lowercasing: All text was converted to lowercase
2. Accent characters were standardized (e.g., "rosé" → "cafe").
3. Contraction Expansion: I expanded contracted words like “can’t” to “cannot” to ensure consistency.
4. URL Removal: Links were removed
5. All numbers and Punctuations were removed
6. All white spaces and special characters were removed
7. I removed hashtags and mentions which are common in tweets

For comparison, I created two different datasets. In the other,I removed common stopwords such as "i” "me", and "and" using the NLTK stopword list. In my initial training, I did not remove stopwords, since some of them may contain important context which may otherwise be lost. For example, many of the depression tweets were describing first person experiences, which used words like “I”, “me” to describe negative experiences, while many of the tweets labelled as personality disorder describe experiences about other people. Similarly cannot was the fifth most common word in tweets labelled Suicidal. Removing stop words would remove the difference between “can” and “cannot” and “me” and “them”. Since an LSTM is trained to understand context and long term dependencies, it is not immediately clear if removing stop words would be beneficial.

.

I then used the keras's in-built tokenizer, to tokenize the data. This splits texts into words, assigning a number to each word. This is followed by padding/truncating using pad_sequences() to ensure uniform input length for the LSTM.

BERT

For the BERT model, I avoided aggressive preprocessing such as lowercasing, punctuation removal, or stopword filtering. This is intentional, as BERT's tokenizer is designed to handle natural, noisy text using subword tokenization. I usedused the `AutoTokenizer` from Hugging Face's Transformers library to tokenize the text into `input_ids` and `attention_masks`, while also managing special tokens, padding, and truncation automatically. This ensured our text was correctly formatted for input into BERT without losing linguistic nuances.

Model and Training Choice

LSTM

LSTM models are fairly effective for sequential text because they effectively handle long-range dependencies and context. I chose a single layer LSTM as a baseline to evaluate other models.

I used tensorflow's LSTM model on to train the following architecture:

Embedding Layer: 5000 tokens, 128 dimensions:

Based on the vocabulary analysis, 5000 words covered 93% of the training corpus, even before preprocessing.

Layer: 64 hidden units

Dropout layer of 0.5 and a final dropout of 0.3: Prevents overfitting by randomly disabling 50% of neurons during training and 30% of Neurons before the final layer.

Dense Layer- 64 units with a ReLU activation to add non linearity

Dense Output Layer (softmax) of 7 layers: Based on Number of classes

Training Details:

- **Dataset Split:80% training, 10% validation, 10% testing. Fixed random state (`random_state=22`) to ensure reproducibility.**

- **Optimizer and Loss Function:**
  - **Optimizer: Used an Adam optimizer with learning rate (`learning_rate=0.001`) is a common choice in NLP tasks**
  - **Loss Function: Sparse categorical cross-entropy, suitable for multi-class integer-encoded labels.**

- **Training Setup:**
  - **Batch size: 32:** Based on Computational constraints
  - **Epochs: Up to 20 epochs,** with Early Stopping based on validation loss (`patience=3`). In all cases, the model stopped early.

BiLSTM

I then trained a Bidirectional Long Short-Term Memory (BiLSTM), to process the sequence from both directions, forwards and backwards to improve context awareness. I used a similar architecture structure as the LSTM to ensure comparability. The only difference being the layer is twice as large.

Note: I trained each model twice, first with the dataset without stop words removed and then with the dataset with stop words removed.

BERT

For more advanced text understanding, I implemented a **pre-trained BERT-based model** (BERT-base-uncased) using Hugging Face's `transformers` library.

Used Hugging Face's `AutoTokenizer` with truncation and padding (max_length=128), converting text into BERT's expected input format (`input_ids`, `attention_mask`).

Model: `BertForSequenceClassification`
I used the BERT-base version which has:

- 12 Transformer layers

- 768 hidden dimensions
- 12 attention heads
- 110M+ parameters

To ensure consistency and comparability , I used the same 80% training, 10% validation, 10% test split , using the same random seed (`random_state=22`).

- **Training Setup:**
  Batch size: 32 for training and 64 for eval
- **Epochs: 4**

I used the empirically tested learning rate of 2e-5, and saved the best model in the end.

Performance

Base LSTM

The base LTSM model performed decently well, with an overall accuracy of 74.25% and a weighted F1-score of 73.89%.

```
Accuracy:  0.7425
Precision: 0.7403
Recall:    0.7425
F1 Score:  0.7389
```

```
                  precision    recall  f1-score   support

           Anxiety       0.81      0.72      0.76       384
           Bipolar       0.67      0.77      0.71       273
        Depression       0.67      0.75      0.70      1485
            Normal       0.94      0.90      0.92      1681
Personality disorder       0.00      0.00      0.00       107
            Stress       0.40      0.60      0.48       250
          Suicidal       0.68      0.60      0.64      1067

          accuracy                           0.74      5247
         macro avg       0.59      0.62      0.60      5247
      weighted avg       0.74      0.74      0.74      5247
```

However, there were significant differences in performances in different classes. Normal and Anxiety classes were predicted with the highest reliability (F1 scores of **0.92** and **0.76** respectively). The normal class was the most well represented in the dataset. Depression was also well represented and performed moderately well(**0.70).** However, Stress performed poorly(0.48), while the model failed to predict Personality disorder at all(**F1=0)**. Based on the

EDA, Personality disorder seemed to have the most different tweets, they mostly focused on observations on external events rather than feelings. It was also the most underrepresented class and lacked words like stress or anxiety which could easily distinguish it. The **macro-average F1 score (0.59)** is significantly lower than the **weighted average F1 (0.74)**, indicating that performance is skewed toward well-represented classes.

After removing stopwords from the input data, the LSTM model showed a slight improvement in performance, achieving an accuracy of 75.59% and a weighted F1-score of 75.53%

```
Accuracy:  0.7559
Precision: 0.7658
Recall:    0.7559
F1 Score:  0.7553
                      precision    recall  f1-score   support

             Anxiety       0.84      0.73      0.78       400
             Bipolar       0.80      0.74      0.77       250
          Depression       0.76      0.68      0.72      1555
              Normal       0.89      0.93      0.91      1649
Personality disorder       0.41      0.10      0.16       111
              Stress       0.38      0.70      0.49       256
            Suicidal       0.66      0.69      0.68      1027

            accuracy                           0.76      5248
           macro avg       0.68      0.65      0.64      5248
        weighted avg       0.77      0.76      0.76      5248
```

Normal remained the most accurate class, however it saw slightly lower performance(F1 score of **0.91**). Anxiety, Bipolar, and Suicidal classes also saw stable or improved performance (F1 scores of **0.78, 0.77, and 0.68**, respectively). However, stress and personality disorder saw significant improvements, though the former still had very low recall(**0.10)** suggesting it was still missing most of them.Removing stop words proved beneficial, particularly for minority and symptom-driven classes, improving the model’s ability to generalize and extract relevant features.

BiLSTM

The BiLSTM results with stop words included:

```
Accuracy:  0.7604
Precision: 0.7673
Recall:    0.7604
F1 Score:  0.7615

                    precision    recall  f1-score   support

             Anxiety       0.85      0.80      0.82       384
             Bipolar       0.79      0.81      0.80       273
          Depression       0.74      0.64      0.69      1485
              Normal       0.92      0.92      0.92      1681
Personality disorder       0.53      0.48      0.50       107
              Stress       0.50      0.54      0.52       250
            Suicidal       0.61      0.73      0.66      1067

            accuracy                           0.76      5247
           macro avg       0.71      0.70      0.70      5247
        weighted avg       0.77      0.76      0.76      5247
```

Without stop words included:

```
Accuracy:  0.7759
Precision: 0.7744
Recall:    0.7759
F1 Score:  0.7749
                      precision    recall  f1-score   support

             Anxiety       0.78      0.84      0.81       400
             Bipolar       0.84      0.79      0.81       250
          Depression       0.74      0.73      0.73      1555
              Normal       0.91      0.93      0.92      1649
Personality disorder       0.57      0.51      0.54       111
              Stress       0.54      0.55      0.55       256
            Suicidal       0.68      0.66      0.67      1027

            accuracy                           0.78      5248
           macro avg       0.72      0.72      0.72      5248
        weighted avg       0.77      0.78      0.77      5248
```

The BiLSTM shows marginally better overall performance than the LSTM.IWith stopwords retained, the model achieved an **accuracy of 76.04%** and a **weighted F1-score of 76.15%**. After removing stopwords, the performance improved across nearly all metrics, with accuracy rising to **77.59%** and the weighted F1-score increasing to **77.49%**. However, compared to the LSTM, the biggest improvement is that its much more balanced in its classification of various tasks. Personality disorder classification improves with a much higher recall 55% compared 10%. iLSTM’s bidirectional architecture enabled it to consider both past and future context within a sentence, which proved valuable for identifying psychological cues that are not always sequentially linear. Users with Personality Disorder or Stress often write longer, run-on, or erratically structured tweets. This may be missed with the single directional approach of the LSTM. For example, the tweet, “I want to be okay, but I can’t stop crying, I hate myself”, contains multiple emotional cues which may be missed with a unidirectional LSTM. These tweets also contained many stop words, and focusing on the content itself helped improve performance.

BERT

The BERT model’s performance exceeded the BiLSTM in all parameters.

```
Accuracy:  0.8253
Precision: 0.8267
Recall:    0.8253
F1 Score:  0.8250

Classification Report:
                      precision    recall  f1-score   support

             Anxiety       0.92      0.88      0.90       400
             Bipolar       0.82      0.87      0.84       250
          Depression       0.75      0.81      0.78      1555
              Normal       0.96      0.95      0.95      1649
Personality disorder       0.86      0.71      0.78       111
              Stress       0.75      0.78      0.76       256
            Suicidal       0.72      0.65      0.68      1027

            accuracy                           0.83      5248
           macro avg       0.82      0.81      0.81      5248
        weighted avg       0.83      0.83      0.83      5248
```

It achieved an accuracy of **82.53%** and a weighted F1 score of **82.50%.**These scores indicate **robust performance** on the multi-class classification task, significantly outperforming both the LSTM and BiLSTM models. Like the LTSM, it retained high recall and precision for Anxiety and Normal. However, it also **retains solid recall** for low-support classes like *Bipolar* and *Personality Disorder* (87% and 71%, respectively). This highlights BERT’s ability to leverage contextual embeddings and fine-grained language understanding. BERT is pretrained on a large training corpus and is therefore, more familiar with diverse linguistic patterns, allowing it to even identify classes which are not well represented in the training set. It also uses **WordPiece tokenization**, breaking rare or unknown words into meaningful subwords (e.g. *suicidalness → suicidal + ness*).

Explainability Analysis and Class signatures

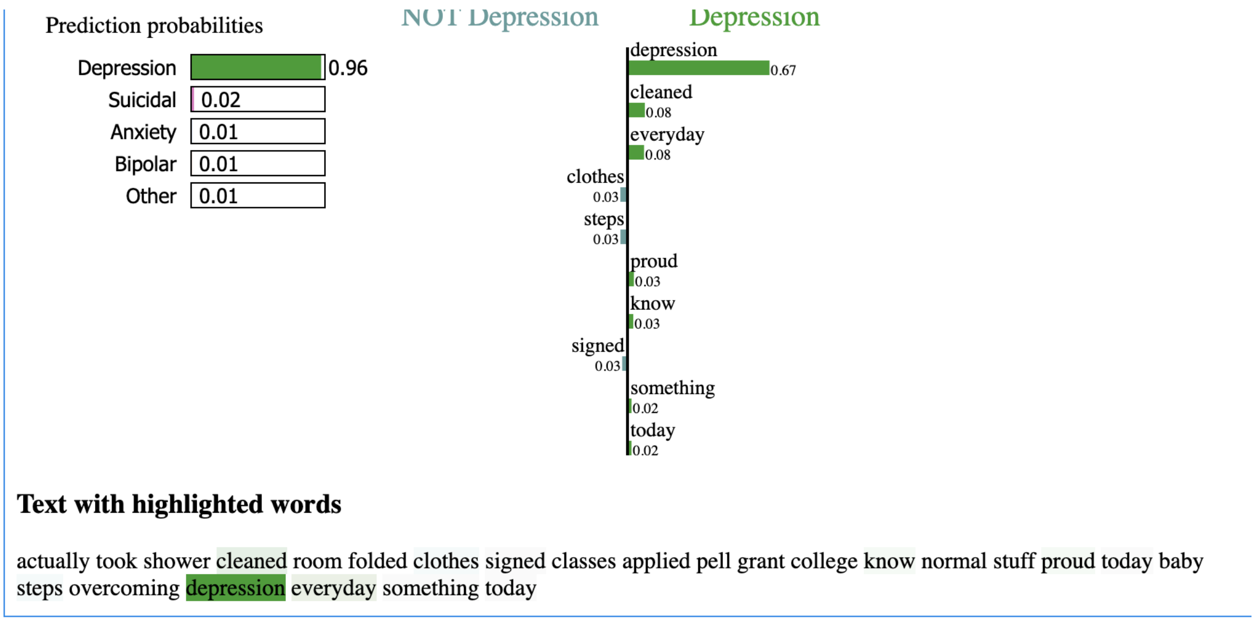
LIME

LIME (Local Interpretable Model-agnostic Explanations), generates an approximation of the model by perturbing the input and observing changes in the output.
I used LIME to generate explanations of some examples of the BiLSTM model and BERT.
Words that contribute both positively or negatively to predicting the class are highlighted

BiLSTM

For example,

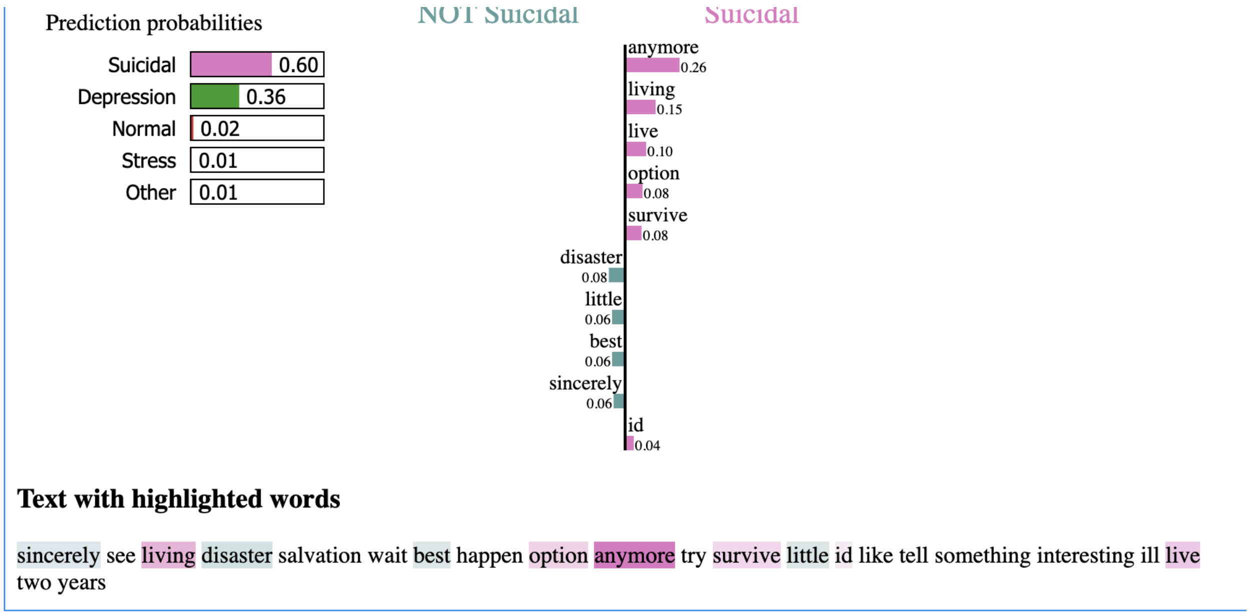
```
True Label: Depression
```



Text with highlighted words

actually took shower cleaned room folded clothes signed classes applied pell grant college know normal stuff proud today baby steps overcoming depression everyday something today

For the following text, the BiLSTM model is classified as being depressed with 96% probabilty. The word “depressed” itself was the most important feature it identified, Suggesting that often people in tweets are acknowledging their mental health condition which the model is identifying.



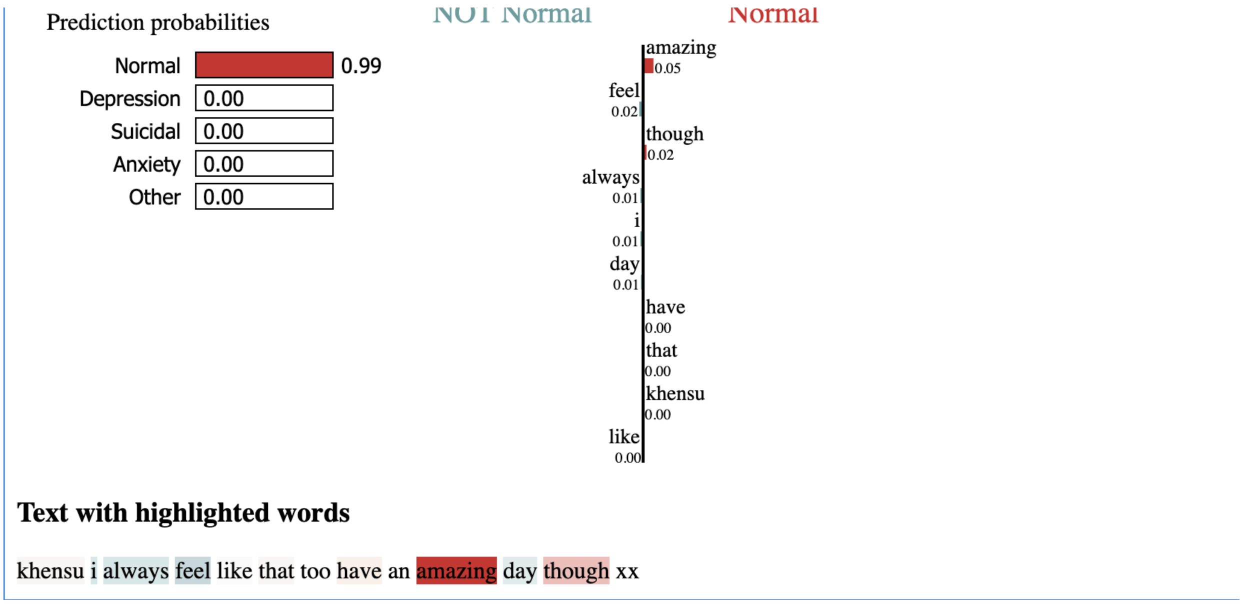
Prediction probabilities

Suicidal 0.60
Depression 0.36
Normal 0.02
Stress 0.01
Other 0.01

Text with highlighted words

sincerely see living disaster salvation wait best happen option anymore try survive little id like tell something interesting ill live two years

In this tweet, it found “anymore”, “live” and “living” as important features for classifying as suicidal. Words like “little” were pushing against the prediction.



Prediction probabilities

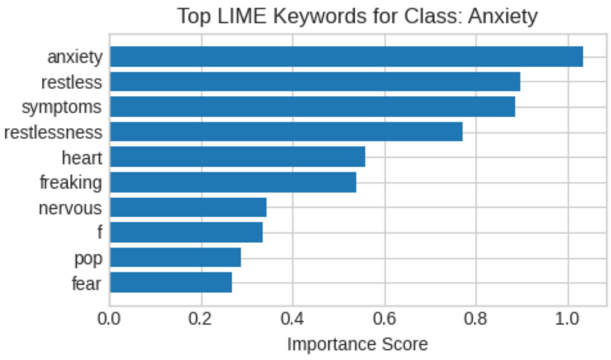
Normal 0.99
Depression 0.00
Suicidal 0.00
Anxiety 0.00
Other 0.00

Text with highlighted words

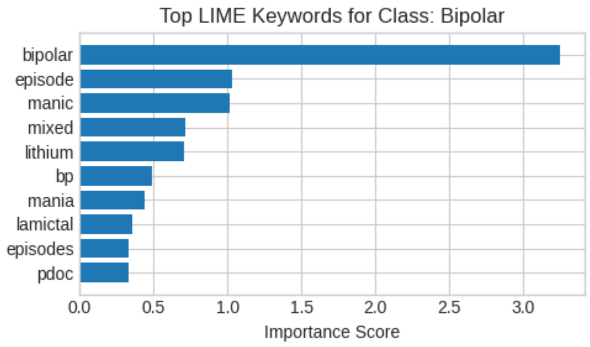
khensu i always feel like that too have an amazing day though xx

Here, words like “amazing” are indicative of normal. Words that tend to show positive emotion are classified as normal.

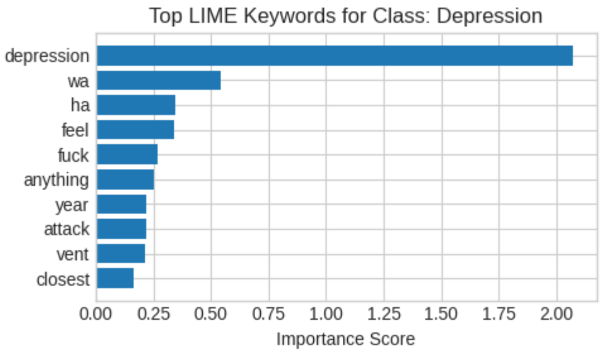
I then ran LIME across multiple representative samples for each class and aggregating the most influential word to identity class-specific keywords that contribute most to the model's predictions. These words can be seen as **class signatures.**



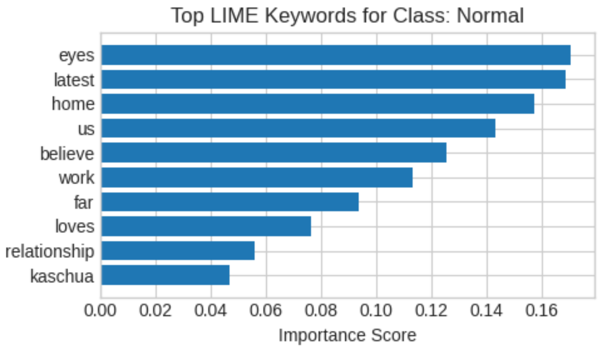
Top LIME Keywords for Class: Anxiety



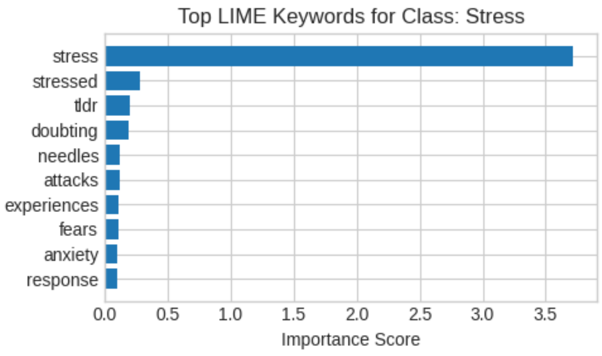
Top LIME Keywords for Class: Bipolar



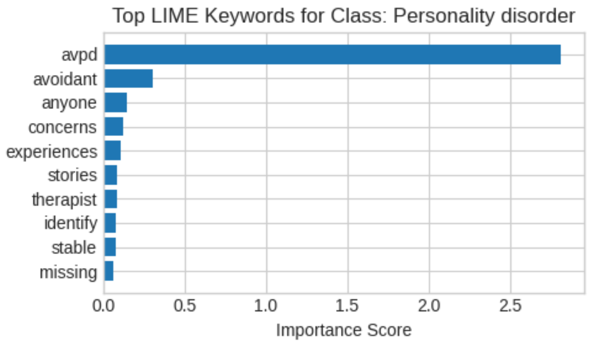
Top LIME Keywords for Class: Depression



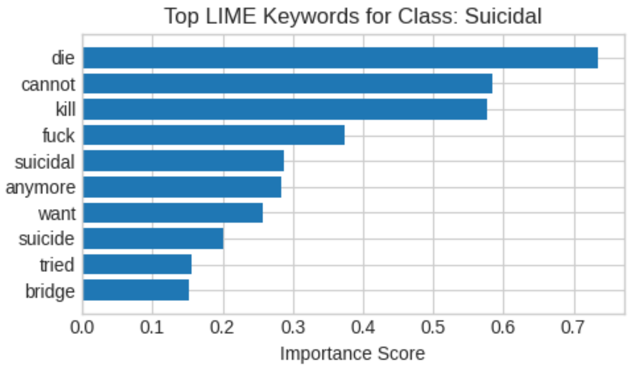
Top LIME Keywords for Class: Normal



Top LIME Keywords for Class: Stress



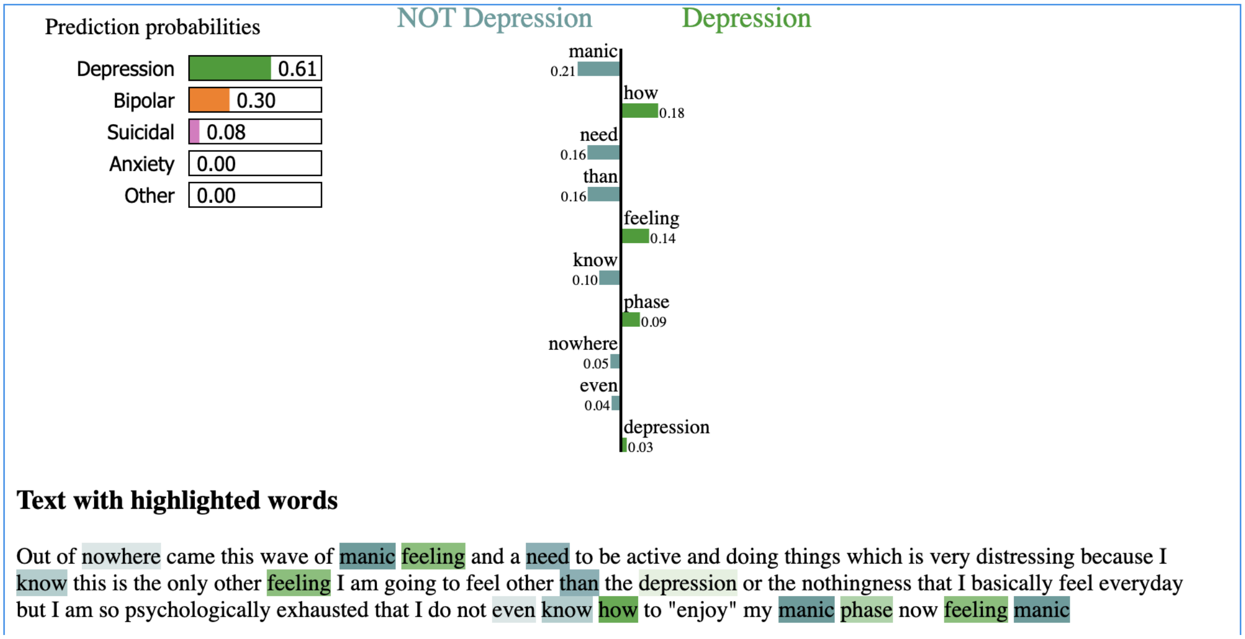
Top LIME Keywords for Class: Personality disorder



Top LIME Keywords for Class: Suicidal

The model was able to identify key words for each mental health diagnosis. In many cases, the diagnosis was included in the statement. For example “depression” was the most important keyword for depression and “anxiety” was the most important for anxiety. In many cases, it was putting out symptoms such as “restless” and “nervous” for anxiety, “avoidant” for personality disorder, and “die” for suicidal. This suggests that users are aware of their symptoms and are using tweets to vent their frustrations. While both "Anxiety" and "Stress" share some overlap, LIME shows that anxiety tweets mention internal symptoms like *heart* or *nervous*, whereas stress tweets focus on external stressors and situational terms.The normal class words were about a variety of things which were largely neutral to positive.

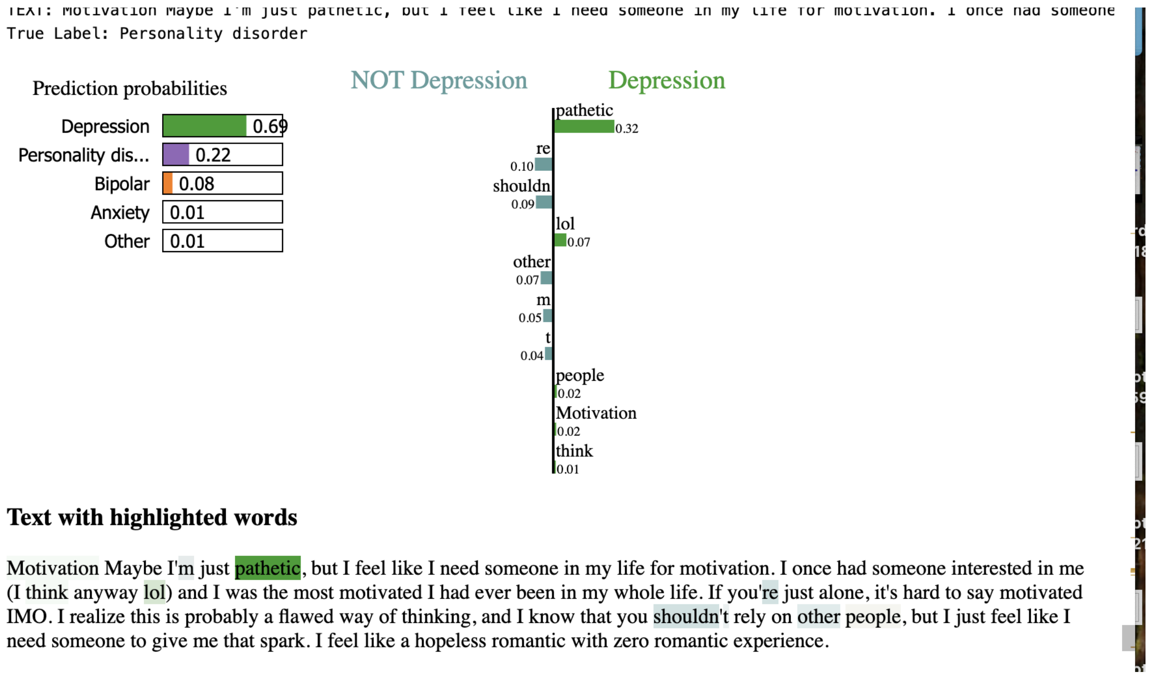
BERT



Text with highlighted words

Out of nowhere came this wave of manic feeling and a need to be active and doing things which is very distressing because I know this is the only other feeling I am going to feel other than the depression or the nothingness that I basically feel everyday but I am so psychologically exhausted that I do not even know how to "enjoy" my manic phase now feeling manic

Similarly, we apply the same methodology for BERT. BERT is able to do a better job distinguishing between classes because it understands context. While manic is mentioned many times it still is able to distinguish between depression and bipolar. The model seems to understand the emotional expressed (e.g., “don’t know how to enjoy”) which is more common with depression.

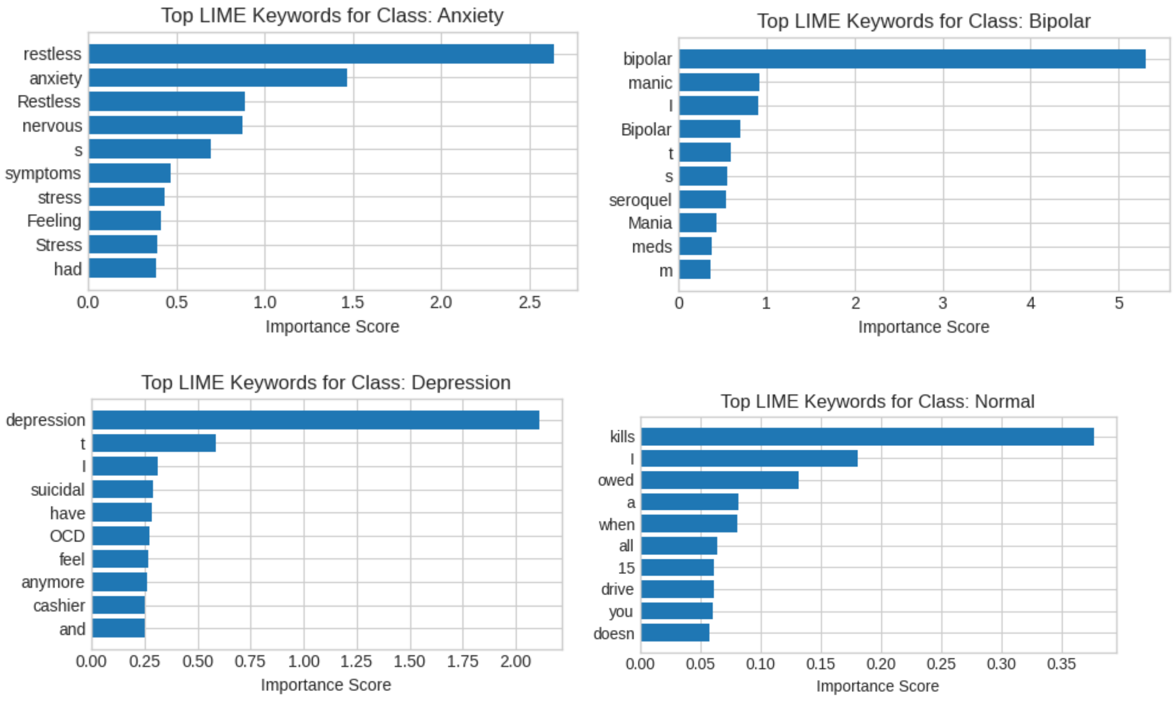


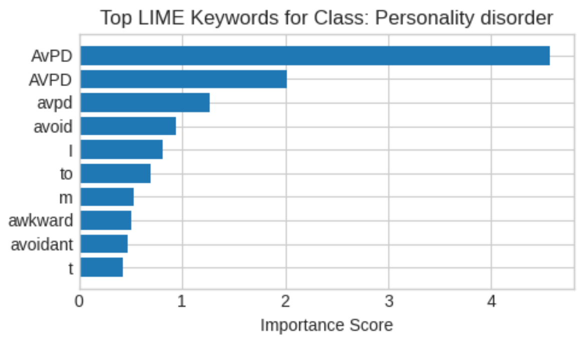
```
TEXT: Motivation Maybe I'm just pathetic, but I feel like I need someone in my life for motivation. I once had someone
True Label: Personality disorder
```

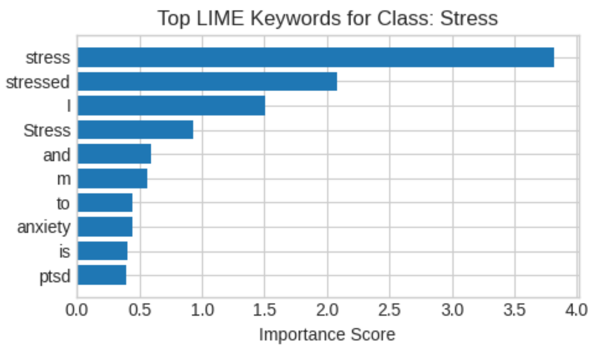
Text with highlighted words

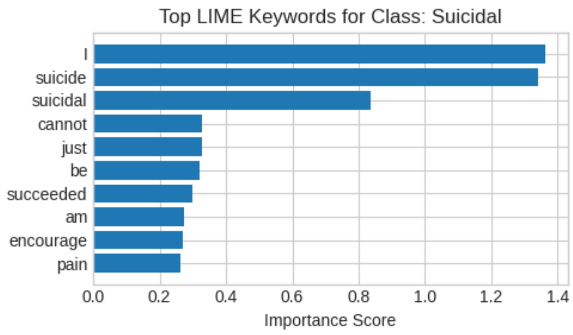
Motivation Maybe I'm just pathetic, but I feel like I need someone in my life for motivation. I once had someone interested in me (I think anyway lol) and I was the most motivated I had ever been in my whole life. If you're just alone, it's hard to say motivated IMO. I realize this is probably a flawed way of thinking, and I know that you shouldn't rely on other people, but I just feel like I need someone to give me that spark. I feel like a hopeless romantic with zero romantic experience.

However, the model was again getting confused with similar words such as pathetic which are indicative of depression.







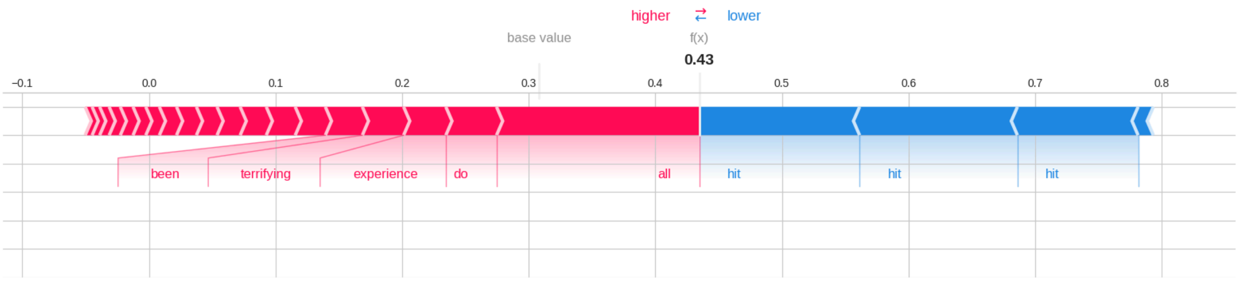


Many of the words overlap between the BERT and BiLSTM modes. The BERT model contains stop words, which are often ranked high in the importance score. BERT doesn't simply assess the importance of individual words in isolation; instead, it evaluates each word in the context of the entire sentence. As a result, stop words like "I" can carry significant weight because of their interactions and relationships with surrounding words in the sentence.

SHAP

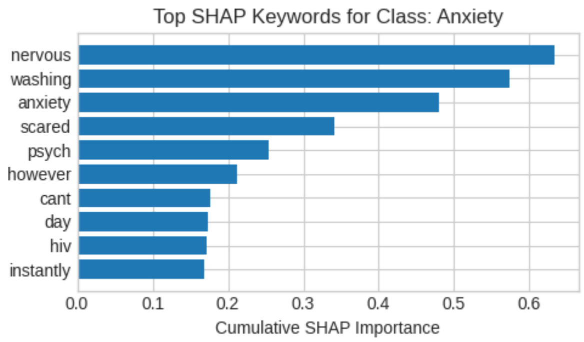
SHAP stands for Shapley Additive exPlanations. It uses game theory to determine importance values.Each prediction is seen as a "payout" in a cooperative game, and the features are "players" that contribute to the payout. SHAP assigns a value to each feature based on how much it contributes to the final prediction, averaged over all possible combinations of features. For computational efficiency, I limited the number of games to 100.

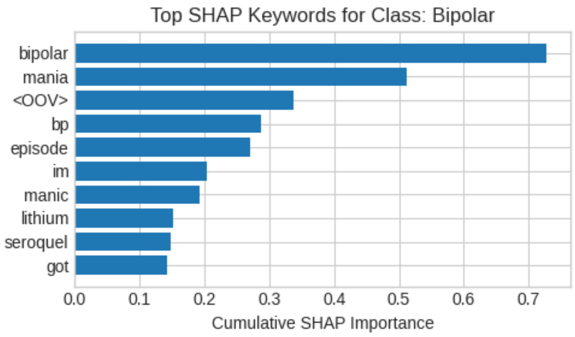
Below is an example of a SHAP visualization on the same sentence:
aware our terrifying hit been they rock like <OOV> the been do to all challenge took to another <OOV> ones am the experience hit dr been hit

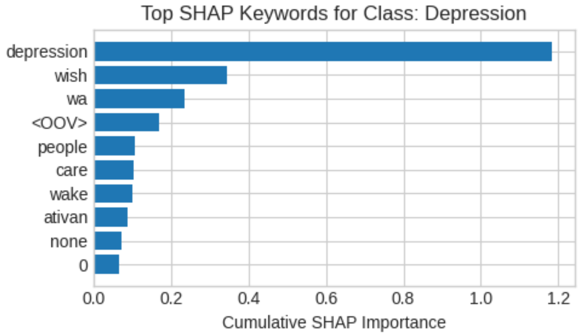


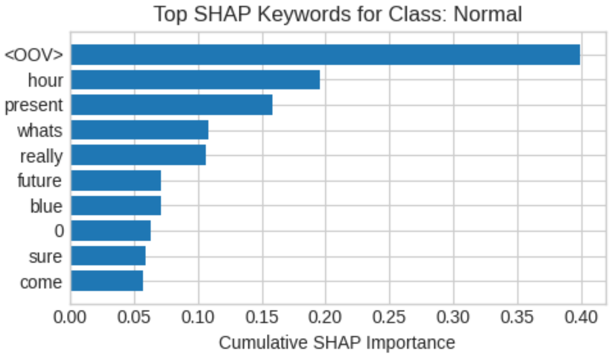
Here red words increase the probability of the predicted class which is depression. Words like experience and “terrifying”,Here <OOV> is the token that keras replaces if not included in the max_words repository.

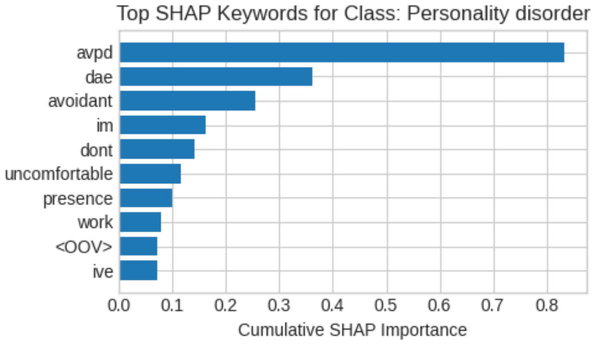
Applying SHAPE on a sample of training data like before, we find the following most important words for each class.

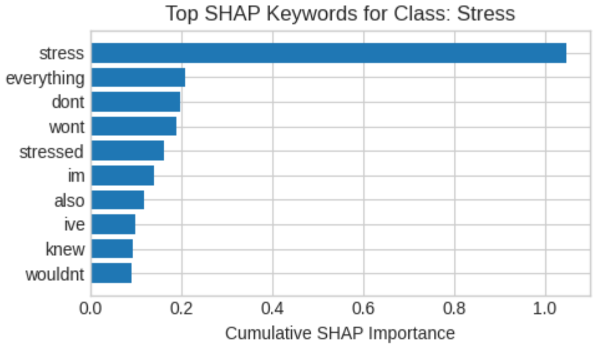


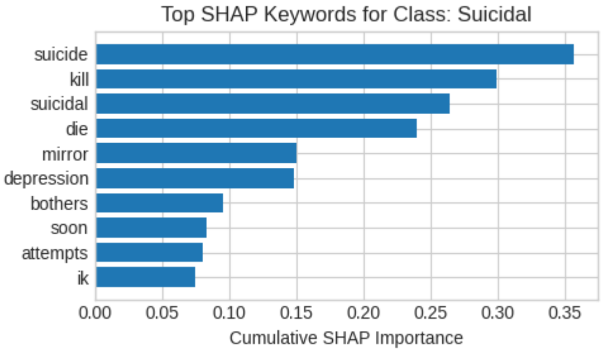












There are some similarities between both LIME and SHAP in terms of the words, they picked up.Both highlight diagnostic terms and feelings, however SHAP was able to pick up on words like “im” and “got”

While LIME focuses on local feature importance, SHAP looks at the global feature contribution to the prediction.

While the model explanations generated using SHAP and LIME provide some insight into why particular predictions are made by the model they are not fully reliable. LIME provides locally linear approximations around a prediction, making its reliability dependent on the choice of neighborhood sampling. When looking at top keywords for each class it's important to remember that the importance is not just attributed to the presence of the word itself but also its importance to surrounding neighbours. Therefore, aggregations miss the nuance and context of the word. Reliability of SHAP explanations for NLP tasks can be affected by tokenization strategies. BERT uses WordPiece tokens, and the presence of `[OOV]` like we saw complicated the understanding of the model.

SHAP offers global and local explanations by attributing numeric importance scores to tokens. It identifies specific keywords that strongly influence the prediction for each mental health class. Interpretability is high when recognisable keywords like “stress” or “die” exist however certain tokens are omitted due to the model design where a fixed vocabulary exists. LIME explanations are easily interpretable due to the importance rankings, clearly highlighting which keywords drive the prediction. However these are linear approximations which do not translate directly to the “reLU” activation in the model.

Both methods identified similar keywords indicative of mental health classes, e.g., "suicide" and "kill" for the Suicidal class, and "bipolar," "manic," "lithium" for Bipolar. differences arose due to how the method calculates importance and the sampling choice.

Reliability may be improved by using domain specific tokenizers as well as a larger vocabulary. For more precision, sampling size should be that of the entire training split.

Challenges Faced and Potential Improvements

By far, the biggest challenge was training the model due to the high amount of compute required. I had to use multiple GPUs using Kaggles’ jupyter server to train the models. This limited the number of times I could train the model. The LSTM model saw a significant improvement by using a bidirectional model as well as by removing stop words. However this improvement was not consistent across classes. A significant challenge was the heavily imbalanced dataset.Minority classes like Stress and PTSD performed badly in particular and the models struggled to generalize well for them.The length was tweets was heavily skewed with some being extremely long, of over 6000 words. This variability complicates sequence modeling, especially for LSTM-based models. Truncating or excluding long texts was necessary but risked losing some valuable content.

The performance of the LSTM models could be improved by increasing the number of layers instead of using a single layer architecture as well as by using an expanded vocabulary set, however this would come at the expense of more compute. Applying oversampling techniques like SMOTE or by under-sampling majority classes could help improve prediction for underrepresented classes.

Running SHAP and LIME on a larger background sample size or on the full training set would lead to more accurate global predictions.