Machine Learning Sentiment Analysis Report

1st Leon Hoang
Tickle College of Engineering
Knoxville, United States of America
phoang5@vols.utk.edu

2nd Sourya Korisapati
Tickle College of Engineering
Knoxville, United States of America
skorisap@vols.utk.edu

3rd Tristan Horton
Tickle College of Engineering
Knoxville, United States of America
chorto14@vols.utk.edu

4th Tyler Garriott
Tickle College of Engineering
Knoxville, United States of America
tgarrio1@vols.utk.edu

***Abstract*—This essay presents a machine learning framework to augment mental health awareness by identifying subtle linguistic indicators of mental health disorders, leveraging a sentiment analysis dataset and baseline model from Kaggle. Data augmentation through back-translation (initially from English to French and back) doubles the training set and addresses class imbalance. This paper discusses extensions made to the baseline model through an improved data preprocessing method that removes empty strings to reduce noise and additional dataset augmentation through translation into additional Germanic and Romance languages, and it includes explanations of other improvements that were attempted but ultimately discarded for various reasons. This model is not intended to diagnose, but to assist medical professionals by flagging potential indicators of mental health issues for further evaluation.**

***Index Terms*—Natural language processing, Back-translation, TF-IDF vectorization, Logistic Regression, Class Imbalance, Multilingual Translation, Cross-Validation, Bootstrap Confidence Intervals**

I. Introduction

Mental health awareness is becoming increasingly important in today's society. However, the science behind it is still in its infancy. Along with that, the efficacy of detection methods for mental health disorders is sub-par. The warning signs can at times be "small — for example, 'I wish I wasn't here' or 'Nothing matters [—]'" [1] and often go unnoticed by friends, family, and medical professionals. Often, patients don't seek help, an "estimated 61.0% U.S. adults aged 18 or older with major depressive episode[s] received treatment in the past year." [2]

The goal of this project is to use a machine learning algorithm to help detect potential disorders that may not have manifested in obvious ways, which can lead to more accurate diagnoses. The goal is not to diagnose patients, as this can only be done by trained medical professionals who have direct access to patients. Instead, the goal is to make medical professionals aware of mental illness indicators in patients that may have otherwise gone overlooked, allowing them to address the issue with the patient.

The proposed solution uses natural language processing (NLP) to process training data, (phrases and sentences said by patients). The data is pulled from the Kaggle sentiment analysis page and is used to categorize disorders. A machine learning algorithm trained on this data would be able to recognize common words and speech patterns among those suffering from various mental health disorders.

II. Dataset Overview and Exploratory Data Analysis

The dataset used in this project, *Sentiment Analysis for Mental Health*, contains approximately 53,000 English-language statements, each labeled with one of seven mental health categories: Normal, Depression, Suicidal, Anxiety, Stress, Bipolar, and Personality Disorder. The data was open source from the Kaggle repository [3]. Each sample consists of an English statement expressed by a user, along with a label representing the inferred mental health condition.

Initial inspection of the dataset revealed a substantial imbalance in label distribution. The categories Normal, Depression, and Suicidal account for the majority of the samples, while Bipolar and Personality Disorder are significantly underrepresented. This uneven distribution introduces classification bias and motivates using strategies such as data augmentation and weighted loss functions during model training. The raw imbalance is illustrated in Figure 1, which shows the label distribution prior to any preprocessing.

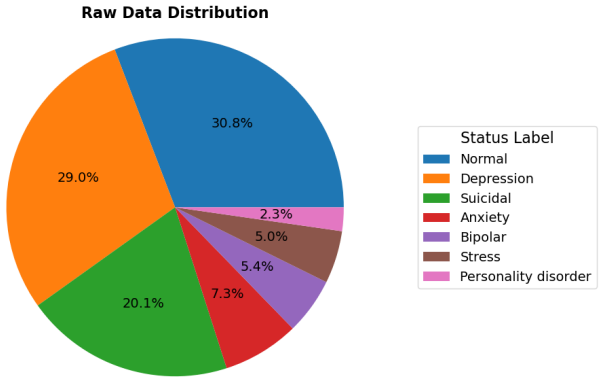

Fig. 1. Data distribution before cleaning

First, the raw data underwent several cleaning steps to ensure consistency and accuracy. All text entries were converted

to lowercase to eliminate case-based variance. Noise such as URLs, punctuation, HTML tags, newline characters, numeric tokens, and bracketed annotations was removed using regular expressions. These steps aimed to reduce irrelevant features and emphasize semantically meaningful content. Finally, any missing or null entries were replaced with empty strings to maintain the integrity of the dataset and prevent vectorization issues.

Next, the cleaned data was preprocessed with Tokenization and stopword removal using the Natural Language Toolkit (NLTK) library. Each statement was broken into individual tokens, and common English stopwords (such as a, an, the, etc.) were removed to focus the model on informative terms. These steps produced a cleaned version of the data suitable for vector representation.

Then, the cleaned and preprocessed data was augmented using a back-translation technique to increase the dataset's linguistic diversity and mitigate class imbalance. Using the TextBlob library, samples were translated from English to French and back to English (failed-to-translated texts were replaced with their original). This augmentation preserved the original semantic content while introducing alternative phrasings, effectively doubling the training data and improving model generalization. It is important to note that both the cleaning and preprocessing steps were reapplied to the new augmented data to maintain the equivalent quality.

The good data was then split into train and test with a standard ratio of 80/20 and random state of 42, ready to be transformed using vectorization from the library Term Frequency-Inverse Document Frequency (TF-IDF). A maximum of 10,000 features was retained, capturing the most relevant unigrams. This representation preserved important lexical distinctions without introducing excessive dimensionality.

Exploratory data analysis revealed that most statements were between 10 and 30 words long, consistent with the brevity expected in social media content. Word cloud visualizations (Fig. 2) highlighted frequently occurring terms such as "feel," "alone," "sad," and "help," indicating a strong emotional connection. In examining class-specific content, significant linguistic overlap was observed between labels such as Suicidal and Anxiety, suggesting that users often express similar symptoms under distinct labels. This ambiguity reflects the real-world comorbidity of mental health conditions and highlights the limitations of purely lexical classification approaches.

These findings informed our modeling choices. The use of TF-IDF was supported by the structured nature of the text, while class imbalance and semantic overlap pointed to the need for both robust baseline models and future context-aware alternatives. Additionally, the observed ambiguity underscores the potential benefits of incorporating contextual embeddings in future iterations.

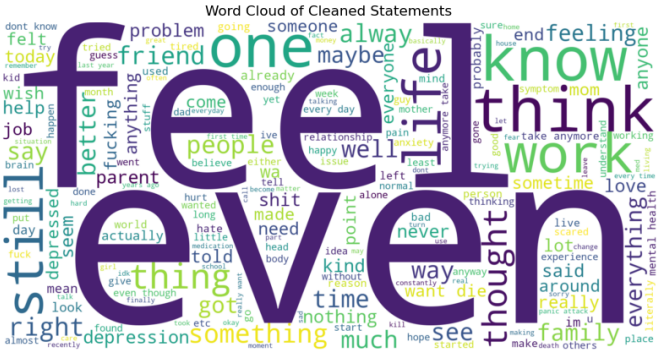

Fig. 2. Word Cloud

III. Baseline Model

The baseline model used in this project was developed by Muhammad Faizan [4] and incorporates standard preprocessing, TF-IDF vectorization, and logistic regression. While other sentiment analysis models utilize a Random Forest Classifier and Naive Bayes Classifier, this Logistic Regression model is more accurate. The techniques in our baseline work together to create a strong model that is simplistic, interpretable, and more accurate than other models. It is the obvious choice.

The model begins with exploratory data analysis and tokenization using NLTK to handle missing values, and cleans up the text data by removing punctuation, stop words, NaN values, links, and numbers. The data is then augmented by paraphrasing the original text. This is done using TextBlob (a Python library for processing textual data) to translate each statement from English to French and back to English. This rephrases the text while preserving the meaning, which increases the diversity and size of the data. The new text is added to the dataset to expose the model to different ways of expressing the same idea. The new data is processed in the same way as the original data, split between the train and testing sets, and vectorized using TF-IDF with 10,000 features, which is done to limit the dimensionality and focus on only the most frequent and meaningful terms (reduce noise).

Once vectorized, the model is trained using a Logistic Regression algorithm, which categorizes the data into seven classes: anxiety, bipolar, depression, normal, personality disorder, stress, and suicidal. The model uses Grid Search with a cross-validation value of 5 for hyperparameter tuning and optimization. The model makes predictions, and its accuracy score is calculated once the hyperparameter tuning is completed. Finally, the confusion matrix and classification report are generated for performance analysis.

With this approach, the model has a strong performance. Its accuracy score is 86.5% (Fig. 3), which, while not the only metric important to determining a model's performance, is still a good indicator of the model's strength. The macro average F1-Score of 0.87 shows that the model's performance is balanced across all classes, regardless of the frequency, and the weighted average of 0.87 further confirms the model's accuracy in classifying the most common categories. The

confusion matrix (Fig 4.) expands on this and offers a more granular view across all categories.

That said, one limitation in the baseline model was its handling of missing values: replacing NaN entries with empty strings introduced zero-vector inputs in TF-IDF potentially adding noise and reducing reliability in edge cases.

Best Parameters:
{'C': 100}
Accuracy Score:
0.8647374870393062
Classification Report:

	precision	recall	f1-score	support
Anxiety	0.92	0.91	0.91	1562
Bipolar	0.93	0.90	0.91	1150
Depression	0.83	0.82	0.83	6182
Normal	0.93	0.95	0.94	6571
Personality disorder	0.85	0.81	0.83	447
Stress	0.89	0.85	0.87	1047
Suicidal	0.77	0.77	0.77	4259
accuracy			0.86	21218
macro avg	0.87	0.86	0.87	21218
weighted avg	0.86	0.86	0.86	21218

Fig. 3. Baseline model Classification Report

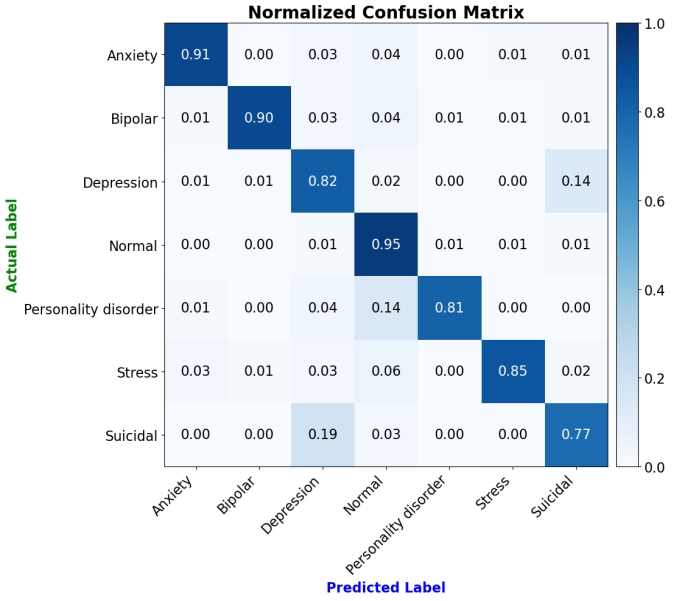

Fig. 4. Baseline model Confusion Matrix

IV. Technical Approach

A. Class Weights

The first strategy we attempted to improve the baseline model was the application of class weighting during training. We did this to attempt to improve the performance of the more important classes such as "Suicidal" and "Depression" as misclassifying a class like that could have real-world consequences. Shifting the model's priorities to increase its sensitivity to these critical labels was done by assigning higher weights to the loss function for those critical classes penalizing incorrect predictions more heavily when those labels are involved.

This adjustment had a clear impact where the most heavily weighted class had a recall for the "Suicidal" class increasing from 0.77 to .80 and its F1-score increasing a little. The weighting of these classes led to an overall drop in accuracy from 0.87 to 0.84 in the weighted model. Several other classes also dropped in terms of performance, especially in recall and F1 scores. For example, in Figure 5 you can see that "Bipolar" has an F1-Score of 0.88 when it originally had a score of 0.92 and in the same figure you can see that "Stress" dropped to 0.77 and "Bipolar" dropped to 0.77. These shifts indicate that the weighting introduced imbalances which boosted minority class sensitivity while undermining general stability.

Best Parameters:
{'C': 100}
Accuracy Score:
0.8420209256291827
Classification Report:

	precision	recall	f1-score	support
0	0.92	0.88	0.90	1555
1	0.94	0.83	0.88	1151
2	0.80	0.80	0.80	6162
3	0.93	0.93	0.93	6541
4	0.86	0.77	0.81	480
5	0.91	0.77	0.83	1068
6	0.71	0.80	0.75	4261
accuracy			0.84	21218
macro avg	0.87	0.82	0.84	21218
weighted avg	0.85	0.84	0.84	21218

Fig. 5. Class weights Classification Report

The changes in different scores indicate a tradeoff, improved minority class performance slightly but degrading overall balance and generalization. Since our goal was to improve the model overall while keeping the model's performance stable we chose to abandon class weighting.

B. BERT

To improve the baseline we explored replacing TF-IDF vectorization with Bidirectional Encoder Representations from Transformers (BERT). BERT is a contextual language model developed by Google that uses a transformer-based architecture and WordPiece tokenization to generate context-aware embeddings for text. Unlike TF-IDF, BERT provides deep representations that can capture the meaning of a word based on its surrounding context, which is very beneficial for a nuanced task such as mental health classification.

To evaluate BERT, we fine-tuned a pre-trained BERT model for our classification task. As shown in the BERT classification report (Figure 6) the model achieved an accuracy of 0.92, greatly outperforming the TF-IDF baseline accuracy of 0.87. It also delivered higher macro and weighted F1-scores 0.93 vs 0.87 in the baseline with improved recall across all classes,

which included a significant gain for the "Suicidal" class going from 0.77 to 0.91. This confirms that BERT could substantially improve performance the model's performance The problem with BERT was the computational demands. Even before expanding the dataset through multilingual augmentation, BERT took 30 minutes to run with three epochs (one complete pass through the entire training dataset during model training). In our case, training the BERT model on the baseline data set, which includes the French language, augmentation took 30 minutes. The 30 minutes is the time it takes to process the entire dataset three times, which is common practice for ensuring the model learns effectively.

This leads to a bottleneck when we expand the dataset with our data augmentation. The size of the training data will grow nearly six times, and that leads to proportional increases in both processing time and memory consumption. Training BERT on the augmented dataset is impractical with run times increasing beyond our ability to iterate effectively. With how big our dataset is after augmentation, we experienced kernel crashes and instability during training from memory limitations which made BERT unviable for us. Given the improvements in accuracy, BERT would be great for future improvements to our model.

```
Epoch 1 - Loss: 0.5547
Epoch 2 - Loss: 0.3077
Epoch 3 - Loss: 0.1700
                     precision    recall  f1-score   support

            Anxiety       0.95      0.96      0.96      1523
            Bipolar       0.97      0.94      0.96      1109
         Depression       0.92      0.88      0.90      6204
             Normal       0.98      0.97      0.98      6505
Personality disorder       0.97      0.90      0.93       424
             Stress       0.88      0.92      0.90      1080
           Suicidal       0.84      0.91      0.87      4228

           accuracy                           0.92     21073
          macro avg       0.93      0.92      0.93     21073
       weighted avg       0.93      0.92      0.92     21073
```

Fig. 6. BERT Classification Report

C. Preprocessing Improvements

As part of our early pipeline tuning, we identified that handling missing values properly had a measurable effect on model stability and accuracy. In the baseline model, missing text entries (NaN) were being converted to empty strings. However, this led to problems: vectorizers like TF-IDF treated empty strings as all-zero vectors, while transformer models like BERT processed them as padding or default embeddings. This introduced noise and potentially biased the model.

To address this, we chose to drop all rows with missing text data–representing just 1% of the dataset–ensuring the training set only included semantically meaningful input-output pairs. This minor preprocessing change reduced noise, improved signal clarity, and ensured the model learned from valid, real examples.

D. Expanded Data Augmentation

To improve the dataset for our model, we implemented a strategy centered around multilingual back-translation using TextBlob like the baseline as a form of further data augmentation. Rather than altering the model's architecture, the baseline already had very strong performance, we decided to enhance the training data in a way that would encourage greater generalization. The baseline model initially only used French as the intermediate language, however, we expanded this to include Italian, Spanish, German, and Portuguese.

We ran the model with different languages such as a few different Asian languages, however, the ones we used for our final worked well due to their linguistic diversity, wide availability in machine translation systems, and contrast with English. These languages also represent a good mix of Romance, and Germanic roots which help introduce a variety of rephrasing and idiomatic structures to the augmented data.

Through back-translation, we translated the original English text to a selected language and then translated it back to English. This creates natural variations in phrasing and structure without changing the core meaning of the text. This augmentation creates a wider range of linguistic patterns while preserving the original sentiment.

In practice, these changes worked to significantly expand the dataset but importantly increase the diversity of the text available. The more robust language and variation allows for more natural language expression to be accounted for; this is especially important for something like this where everyone expresses themselves differently.

V. Results

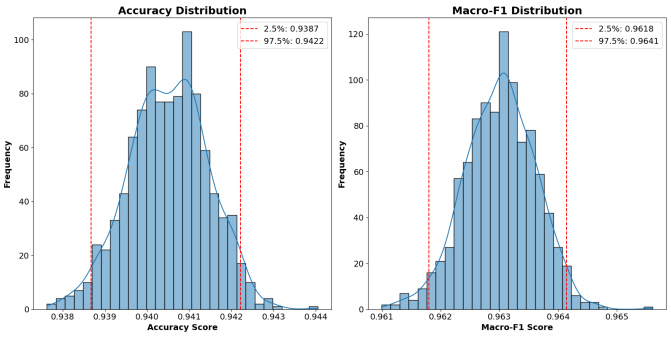

Fig. 7. Final models

The bootstrap histograms in Figure 7 illustrate the distribution of accuracy and macro F1 scores over 1,000 resamples. Our final model achieved a 95% confidence interval of 0.9387 to 0.9422 for accuracy and 0.9618 to 0.9641 for macro F1, indicating high performance and extremely low variability. This tight spread shows that our model generalizes well and performs consistently across different subsets of data.

A. Cross-Validation Statistics

Our final model achieved a cross-validation mean accuracy of 0.8519, which is significantly higher than the baseline

model's 0.7874. Additionally, the standard deviation of our final model is 0.0011 which is much smaller than the baseline model's 0.0025 showing greater stability and less performance fluctuation across the CV folds.

B. Classification Reports

```
Best Parameters:  {'C': 100}

Accuracy Score:  0.9405390869689013

Classification Report:
                     precision    recall  f1-score   support

            Anxiety       0.99      0.99      0.99      4509
            Bipolar       1.00      1.00      1.00      3331
         Depression       0.91      0.91      0.91     18321
             Normal       0.98      0.99      0.99     19890
Personality disorder       1.00      1.00      1.00      1256
             Stress       1.00      0.99      0.99      3078
           Suicidal       0.87      0.86      0.87     12833

           accuracy                           0.94     63218
          macro avg       0.96      0.96      0.96     63218
       weighted avg       0.94      0.94      0.94     63218
```

Fig. 8. Final model Classification Report

The classification report (Fig. 8) shows improvements in our model across the board in precision, recall, and F1-score. An example of this can be seen in "Suicidal," where the F1-score improved from 0.77 in (Fig. 3) to 0.87, and "Personality Disorder" rose from 0.83 to 1.00. Previously, these sections were under performing or imbalanced classes which suggests that our new model is better able to capture the nuanced class patterns after the augmentation.

C. Confusion Matrices

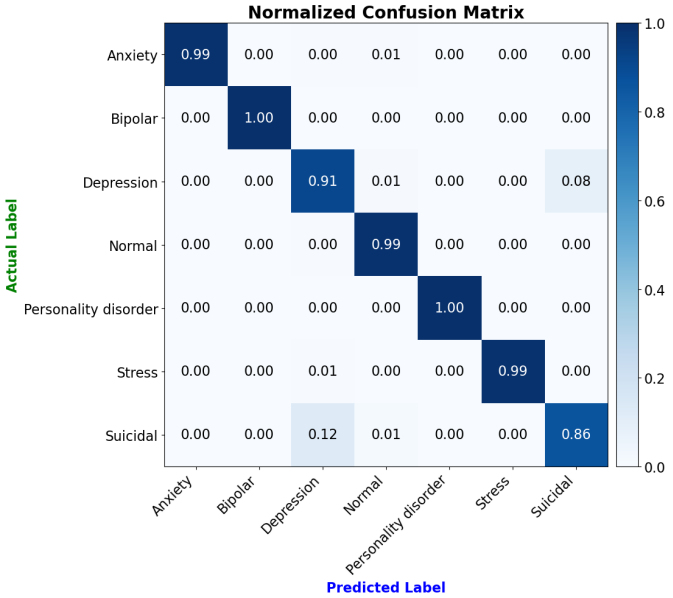

Fig. 9. Enter Caption

The confusion matrix (Fig. 9) visualizes how accurately our model can predict each class. Our results show very strong results along the diagonal with fewer errors, especially in categories such as "Suicidal" and "Depression". This is in contrast to the baseline model (Figure 4), which had more confusion in classes like "Suicidal", "Depression" or "Normal," showing less precision than our final model.

D. Discussion

These results for our final model provide compelling evidence that data augmentation can significantly enhance performance in mental health classification tasks. The tight confidence intervals in the bootstrap analysis and the macro F1 scores indicate very high consistency in the model's predictions across random subsamples. This suggests that the model is able to generalize very well and is not dependent on specific parts of the dataset.

The cross-validation statistics further support this conclusion. Our final model achieved a notably higher mean accuracy of 0.8519 vs 0.7874 and a substantially lower standard deviation of 0.0011 as opposed to 0.0025 signaling better stability across folds. These metrics demonstrate that our changes to the model have reduced variance and increased the robustness of our dataset.

Important areas of improvement are in the critical classes. The F1-score for the "Suicidal" class has improved from 0.77 (Fig. 3) to 0.87 (Fig. 8) and "Personality Disorder" rose from 0.83 (Fig. 3) to 1.00 (Fig. 8). These improvements suggest that the back-translation helped expose the model to more expressive variations of the same sentiment ultimately making them easier to identify.

The performance of our model is encouraging however we are hesitant to say our model is perfect. Our model's near-perfect F1 score could indicate that our data augmentation is leading to overfitting. The possibility of overlap or redundancy in back-translated examples could create patterns this model learns too well, ultimately inflating the confidence without true generalization. Future work would need to explore out-of-sample testing on real-world data to better assess how well our model is generalizing.

Additional factors may have contributed to overfitting. The class imbalance in the dataset meant that classes like "Normal" and "Depression" dominated, potentially biasing the model despite synthetic sampling. Moreover, TF-IDF vectorization—while effective at capturing lexical information—creates a high-dimensional sparse feature space, which may cause the model to latch onto specific tokens that do not generalize well to new data.

Additionally, the tradeoffs explored during our experimentation, especially with class weighting and BERT could be important for the future development of this model. Utilizing BERT in future models could greatly improve performance at the cost of time and resources.

VI. Conclusion

This project set out to expand upon the existing baseline model that already used natural language processing to flag

potential mental health concerns—augmenting, not replacing, clinician judgment. The model came with a robust logistic-regression baseline (TF-IDF on 10,000 features, 80/20 train-test split), grid search, and utilization of the Natural Language Toolkit (NLTK). We explored class weighting, which traded off overall accuracy for minority-class recall, and fine-tuned BERT, achieving 92% accuracy but at a significant computational cost. Ultimately, our most effective pipeline combined expanded back-translation augmentation (French, Italian, Spanish, German, Portuguese) with enhanced preprocessing (dropping NaNs rather than blank-padding). This approach yielded a cross-validation mean accuracy of 0.8519 (σ = 0.0011 vs. 0.7874 ± 0.0025 baseline) and tight bootstrap confidence intervals for accuracy (0.0.9387–0.9422) and macro F1 (0.9630–0.9652). Notably, F1-scores for underrepresented classes climbed—"Suicidal" from 0.77 to 0.87 and "Personality Disorder" from 0.83 to 1.00—demonstrating that strategic data augmentation can mitigate imbalance without sacrificing stability. However, it's important to recognize that such high accuracy may also be indicative of overfitting the training data rather than a true improvement in generalization.

Distribution of Work

Tyler Garriott found the baseline model. He worked on the BERT implementation and weight classes. He was also responsible for the discussion, the technical approach, results parts of this paper, and contributed to the powerpoint.

Leon Hoang was also responsible for writing the dataset report included in this document. Wrote the code for the data augmentation and redesigning of the code. He also created the Github repository and Discord server that we have been using for this project.

Tristan Horton was responsible for summarizing the baseline model that we will expand upon and improve on in this project. Was responsible for proposing the expansions to the model. He was also responsible for updating the midterm paper to reflect the changes made to code.

Sourya Korisapati was responsible for writing the introduction, conclusion, abstract of this report, formatting it and the slide show presentation, as well as writing the "Work Distribution" section. He was also responsible for editing the paper.

All group members contributed to this report in significant and helpful ways.

References

[1] National Alliance on Mental Illness, "Risk of Suicide," *NAMI*, Feb. 12, 2024. https://www.nami.org/About-Mental-Illness/Common-with-Mental-Illness/Risk-of-Suicide/

[2] National Institute of Mental Health, "Major Depression," *National Institute of Mental Health*, Jul. 2023. https://www.nimh.nih.gov/health/statistics/major-depression

[3] Suchintika Sarkar, "Sentiment Analysis for Mental Health," Kaggle, 2023. https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health

[4] Muhammad Faizan, "Sentiment Analysis for Mental Health - NLP," Kaggle, 2023. https://www.kaggle.com/code/muhammadfaizan65/sentiment-analysis-for-mental-health-nlp/notebook