Analysis of Depressive and Non-Depressive Tweets Using Multi-Layer Perceptron (MLP)

CS 584 Final Project Report

Kiran Kumar Nimmakayala
Department of Computer Science
Emory University
kiran.kumar.nimmmakayala@emory.edu

Abstract

This paper investigates the characteristics and distributions of depressive and non-depressive tweets in India between December 2019 and December 2020. Utilizing sentiment scores derived from TextBlob based on prevalent negative and positive lexicons identified via SentiWord and academic research, this study aims to apply advanced natural language processing (NLP) and machine learning techniques to analyze patterns in the sentiments expressed in tweets. Our analysis not only contributes to the understanding of public sentiment but also enhances methodologies in sentiment analysis using deep learning.

1 Introduction

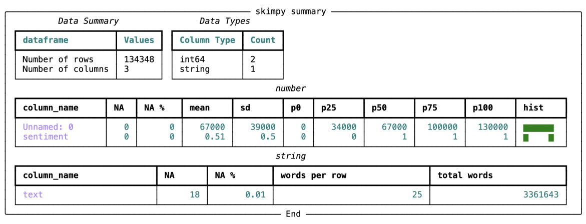
Social media platforms, particularly Twitter, have become vital in understanding public health issues like depression. The ubiquitous nature of such platforms provides a continuous stream of data that, when analyzed, can reveal insights into the public's mental health status over time. This study focuses on a dataset containing depressive and non-depressive tweets originating from India, aiming to detect patterns and potential triggers in public sentiment related to mental health issues.

2 Dataset and Experimental Setup

2.1 Dataset Description

The dataset comprises 134,348 tweets, which have been filtered to include only those that utilize the top 250 most frequently used words from both negative and positive lexicons. These tweets have been further categorized into depressive and non-depressive classes based on sentiment scores calculated using TextBlob, a Python library specifically tailored for the pre-processing and analysis of textual data.

As depicted in Figure 1, the dataset consists of 134,348 tweets, with a near-even distribution of



```
───────────────────────────── skimpy summary ─────────────────────────────
         Data Summary                Data Types
 ┏━━━━━━━━━━━━━━━━━━━┳━━━━━━━━┓ ┏━━━━━━━━━━━━━┳━━━━━━━┓
 ┃ dataframe         ┃ Values ┃ ┃ Column Type ┃ Count ┃
 ┡━━━━━━━━━━━━━━━━━━━╇━━━━━━━━┩ ┡━━━━━━━━━━━━━╇━━━━━━━┩
 │ Number of rows    │ 134348 │ │ int64       │ 2     │
 │ Number of columns │ 3      │ │ string      │ 1     │
 └───────────────────┴────────┘ └─────────────┴───────┘
                                   number
 ┏━━━━━━━━━━━━━━┳━━━━┳━━━━━━┳━━━━━━━┳━━━━━━━┳━━━━┳━━━━━━━┳━━━━━━━┳━━━━━━━━┳━━━━━━━━┳━━━━━━━━┓
 ┃ column_name  ┃ NA ┃ NA % ┃ mean  ┃ sd    ┃ p0 ┃ p25   ┃ p50   ┃ p75    ┃ p100   ┃ hist   ┃
 ┡━━━━━━━━━━━━━━╇━━━━╇━━━━━━╇━━━━━━━╇━━━━━━━╇━━━━╇━━━━━━━╇━━━━━━━╇━━━━━━━━╇━━━━━━━━╇━━━━━━━━┩
 │ Unnamed: 0   │  0 │    0 │ 67000 │ 39000 │  0 │ 34000 │ 67000 │ 100000 │ 130000 │ ██████ │
 │ sentiment    │  0 │    0 │  0.51 │   0.5 │  0 │     0 │     1 │      1 │      1 │ █    █ │
 └──────────────┴────┴──────┴───────┴───────┴────┴───────┴───────┴────────┴────────┴────────┘
                                   string
 ┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━┳━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━┓
 ┃ column_name                ┃ NA       ┃ NA %        ┃ words per row                 ┃ total words                ┃
 ┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━╇━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━┩
 │ text                       │       18 │        0.01 │                            25 │                    3361643 │
 └────────────────────────────┴──────────┴─────────────┴───────────────────────────────┴────────────────────────────┘
─────────────────────────────────── End ───────────────────────────────────
```

Figure 1: Summary statistics of the dataset used in the study, showing the distribution of sentiment and the total number of words across the tweets.

sentiment, and a substantial total word count, signifying a rich corpus for analysis.

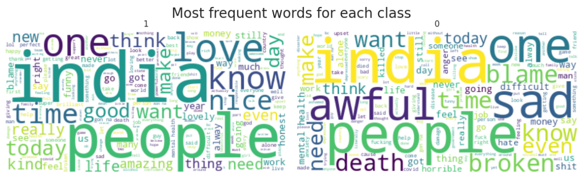


Figure 2: Most frequent words for each class.

As shown in Figure 2, the most frequent words used in depressive and non-depressive tweets exhibit a distinct distribution, reflecting the underlying sentiment.

2.2 Experimental Tools

For the analysis and modeling phases of our experiments, we utilized a suite of Python libraries renowned for their efficacy in data handling and machine learning applications:

- **TfidfVectorizer** from **Scikit-learn** for transforming text data into a matrix of TF-IDF features.
- Deep learning frameworks such as **TensorFlow** and **PyTorch** for constructing and training sophisticated neural network models.
- Evaluation metrics from **Scikit-learn** such as **accuracy score, f1 score, precision-recall**

curve, and **roc curve** to thoroughly assess model performance.

These tools were selected for their efficiency and effectiveness in handling specific challenges in natural language processing and machine learning, ensuring precision and depth in our analytical results.

3 Methodology

This section details the methods employed in preprocessing the data, constructing the model architectures, and outlining the training and evaluation procedures.

3.1 Data Pre-processing

The pre-processing stage is critical for enhancing the performance of our machine learning models. Initially, tweets are cleansed to remove URLs, usernames, and special characters, effectively reducing extraneous content. Each tweet is then tokenized into individual words, and stopwords are eliminated to decrease noise in the dataset. To further refine the text data, we apply TF-IDF vectorization, transforming the tweets into numerical representations that emphasize the most significant words while mitigating the influence of less informative, frequently occurring words.

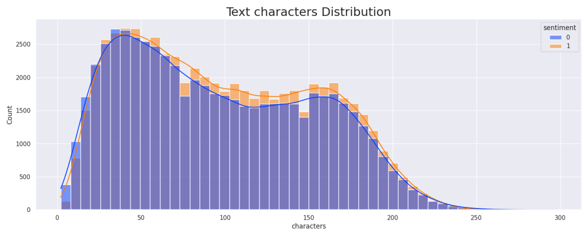


Figure 3: Distribution of characters in tweets.

Figure 3 illustrates the character count distribution across the tweets, indicating the typical tweet length in our dataset.

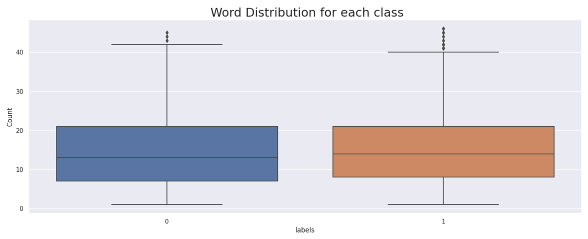


Figure 4: Word distribution for each class after preprocessing.

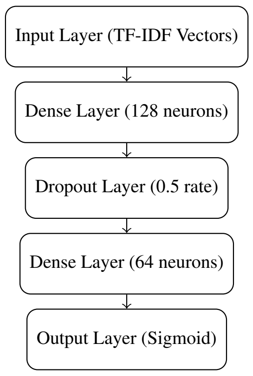
The pre-processing steps' effectiveness is evidenced in Figure 4, where the word count for each class is visualized post-vectorization.

3.2 Model Architectures

We explored sophisticated model architectures using TensorFlow and PyTorch frameworks to assess their efficacy in sentiment analysis tasks on the cleaned tweet dataset.

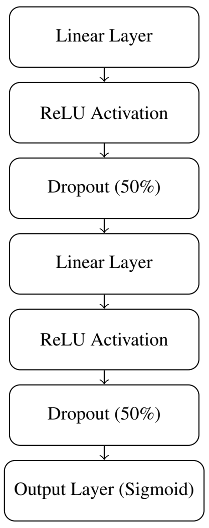
3.2.1 TensorFlow MLP Model

Below is the flow diagram of our TensorFlow model architecture, illustrating the sequential arrangement from the input to the output layer:



3.2.2 PyTorch MLP Model

The following diagram represents the architecture of our PyTorch model, featuring a class-based definition that facilitates encapsulation of the components:



3.3 Training and Evaluation

Both models were trained using a batch size of 32 and a binary cross-entropy loss function, suitable for binary classification problems. Optimization was conducted using the Adam optimizer at a learning rate of 0.001, over 50 epochs with early stopping to prevent overtraining.

Evaluation was performed on a reserved test set, representing 20% of the total data. We employed a range of metrics, including accuracy, precision, recall, and F1-score, to provide a comprehensive evaluation of the models. These metrics are essential for understanding the models' effectiveness, particularly under the challenge of class imbalance inherent in sentiment analysis.

4 Results

This section presents the performance outcomes of our machine learning models. The following classification reports demonstrate the effectiveness of the TensorFlow and PyTorch frameworks in distinguishing between depressive and non-depressive tweets. These results highlight the capability of each model to accurately classify tweets, providing insights into their precision, recall, and overall accuracy.

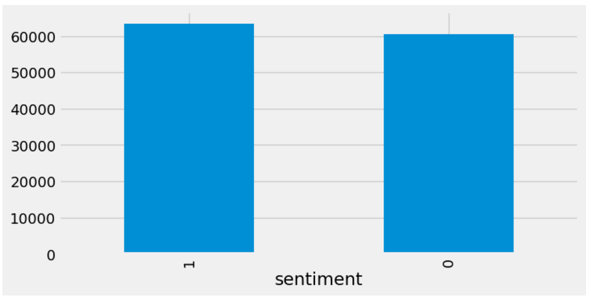


Figure 5: Distribution of classes in the dataset.

The overall distribution of the classes within the dataset is depicted in Figure 5, highlighting the balance between depressive and non-depressive tweets.

4.1 PyTorch Model Classification Report

The PyTorch implementation shows commendable performance with balanced precision and recall across the classes. The classification report is as follows:

Class	Precision	Recall	F1-Score
0	0.87	0.88	0.87
1	0.88	0.87	0.88
Accuracy: 0.87			

4.2 TensorFlow Model Classification Report

The TensorFlow implementation demonstrates slightly higher accuracy than the PyTorch model, particularly in handling class 0 (non-depressive tweets). The details are provided below:

Class	Precision	Recall	F1-Score
0	0.88	0.91	0.89
1	0.91	0.88	0.89
Accuracy: 0.89			

4.3 ROC Curve Analysis

Figure 6 depicts the Receiver Operating Characteristic (ROC) curve for both the PyTorch and TensorFlow models. The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

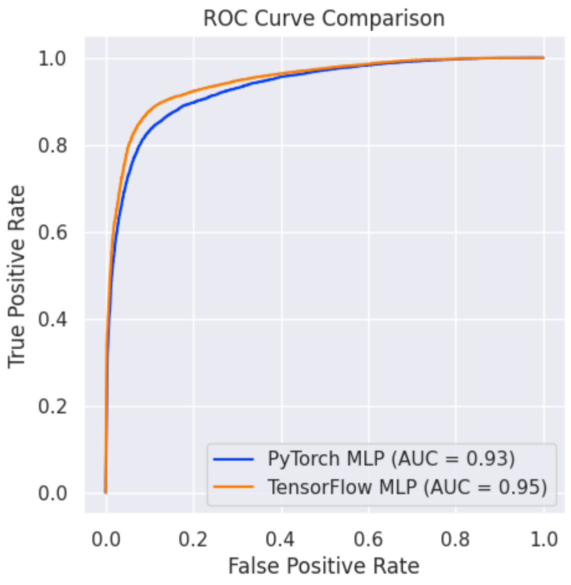


Figure 6: ROC Curve Comparison between the PyTorch and TensorFlow models.

Both models exhibit competitive performance, with the TensorFlow model achieving an AUC of 0.95, which is slightly higher than the PyTorch model's AUC of 0.93. The proximity of both curves to the top left corner indicates high overall accuracy in predictions; however, TensorFlow's closer approach suggests better discrimination between classes. This divergence may reflect differences in model architecture or training processes, indicating TensorFlow's marginal superiority in handling the sentiment classification task presented.

Furthermore, Figure 7 shows the Precision-Recall Curve for both models. The area under these curves also signifies model performance, with higher values indicating greater precision and recall. Again, both models perform well, but the TensorFlow model demonstrates a marginal edge over the PyTorch model, particularly in the area of precision. The high precision across the range of recall levels for both models is critical in the context of mental health, where the accurate classification of depressive tweets is paramount.

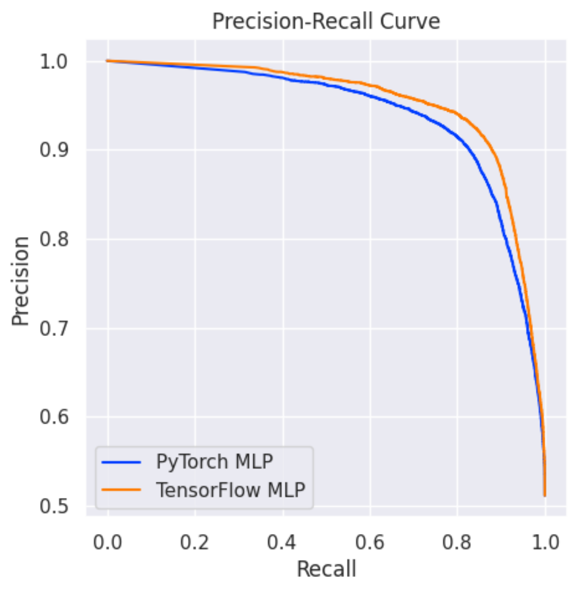


Figure 7: Precision-Recall Curve for the PyTorch and TensorFlow models.

These curves collectively provide comprehensive insight into the models' performance, highlighting their effectiveness in sentiment analysis tasks while also pointing towards slight advantages conferred by TensorFlow's model in this specific instance.

5 Discussion

The comparative analysis indicates that both models perform commendably in the sentiment classification task, with marginal differences in their performance metrics. Notably, the PyTorch model exhibits a higher recall for the non-depressive class, indicating enhanced sensitivity to non-depressive tweets. In contrast, the TensorFlow model demonstrates superior precision, thus presenting fewer false positives within depressive tweets. These findings imply that the choice of model may hinge on the particular demands of the sentiment analysis task at hand—whether the primary objective is to minimize false negatives or false positives.

This nuanced understanding of model performance can guide practitioners in selecting the appropriate framework based on the specific goals and constraints of their sentiment analysis application. For instance, in scenarios where missing out on identifying depressive content could have severe implications, a model with higher recall might be preferred.

6 Conclusion

This research underscores the efficacy of machine learning methods in the analysis and interpretation of voluminous social media datasets for applications in mental health surveillance. The promising outcomes of this study advocate for further explorations into refining these computational models. Future research avenues may include the incorporation of multilingual datasets, enhancing the models' capacity to interpret sentiment across diverse linguistic contexts. Such advancements could augment the versatility and robustness of the developed tools, thereby bolstering their applicability in a broad spectrum of real-world scenarios, particularly in global mental health monitoring and intervention efforts.

Overall, the study contributes valuable insights into the development of automated systems capable of detecting mental health cues in social media communication, highlighting the integral role of AI and machine learning in public health informatics.

7 References

References

Depressive/non-depressive tweets data. https://www.kaggle.com/datasets/mexwell/depressivenon-depressive-tweets-data. Kaggle Dataset.

Pytorch. https://pytorch.org/. An open source machine learning framework.

scikit-learn. https://scikit-learn.org/stable/. Machine Learning in Python.

Sentiment analysis with pytorch part 5: Mlp model. https://galhever.medium.com/sentiment-analysis-with-pytorch-part-5-mlp-model-387057

Tensorflow. https://www.tensorflow.org/. An end-to-end open source machine learning platform.

Textblob. https://textblob.readthedocs.io/en/dev/. A library for processing textual data.

(kag; pyt; sci; ten; tex; hev; **?**)