Sentiment Analysis for Mental Health: A Deep Learning Approach

Group ID: sp25_DsoDusKan

Abstract

This report details the analysis of the "Sentiment Analysis for Mental Health" dataset, employing deep learning to classify mental health statuses from textual statements. The dataset includes categories such as Normal, Depression, Suicidal, Anxiety, Bipolar, Stress, and Personality Disorder. We applied data preprocessing, balancing techniques, and developed models including shallow and deep neural networks, and a fine-tuned BERT model. The BERT model achieved an accuracy of 94.5% and a weighted F1 score of 94.5%. This report presents the methodology, results, visualizations, and key findings.

1. Introduction

This comprehensive dataset is a meticulously curated collection of mental health statuses tagged from various statements. The dataset amalgamates raw data from multiple sources, cleaned and compiled to create a robust resource for developing chatbots and performing sentiment analysis. The goal is to predict these statuses using deep learning models using classification methods. The dataset's initial imbalance required preprocessing and balancing to ensure effective model training.

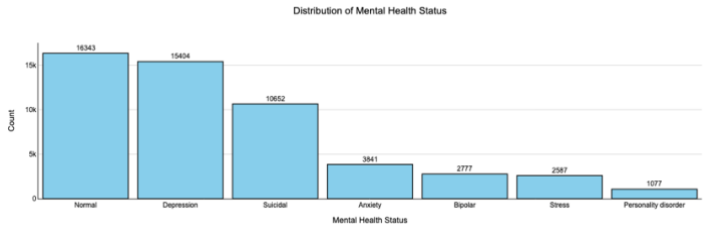
2. Dataset Overview

Initial Class Distribution

The dataset's initial distribution was imbalanced:

- Normal: 16,343
- Depression: 15,404
- Suicidal: 10,652
- Anxiety: 3,841
- Bipolar: 2,777
- Stress: 2,587
- Personality Disorder: 1,077

This imbalance risked underrepresenting minority classes during training.

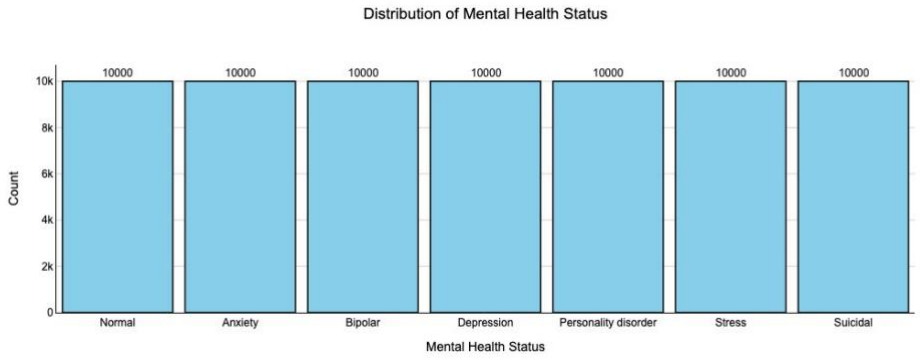


Dataset Balancing

To address the imbalance, we used:

- Under-sampling for majority classes (e.g., Normal, Depression).
- Over-sampling for minority classes (e.g., Personality Disorder, Stress).

Post-balancing, each class had 10,000 samples, ensuring a balanced dataset.



3. Data Preprocessing

Preprocessing steps included:

- Dropping 362 rows with null values in *statement* column
- LabelEncoder was used to convert categorical text labels into numerical values because deep learning models require numerical inputs. This step ensures that the labels are in a machine-readable format before tensor conversion, avoiding complications in model training.

Tokenization and Tensor Conversion:

BERT-based uncased model was chosen because, it has been pre-trained on a large dataset and captures deep contextual meanings of text. The uncased version is used to avoid distinctions between uppercase and lowercase words, which is useful for informal text data.

torch.long ensures compatibility with loss functions like CrossEntropyLoss, which expects long-type labels.

- Created a custom dataset class to convert text and labels into token IDs and attention masks.
- Padded/truncated sequences to a fixed length and returned tensors.
- Used a DataLoader with batch size 32, shuffling data each epoch. The dataset was split into 80% training and 20% testing subsets, with random seeds for reproducibility.

4. Model Development

Shallow Neural Network

A shallow neural network typically has one or two hidden layers between the input and output layers. We implemented an architecture that utilizes a model with a 128-dimensional input. This input is then passed through a linear layer to a hidden dimension, followed by a ReLU activation function. It then maps the result to 7 output classes via another linear layer for classification.

• Hidden dimension of 16 yielded 37% test accuracy and test loss of ≈ 1.61 (SGD, learning rate 0.05, 10 epochs).

• With $n = 256$, accuracy improved to 41.1% when training used SGD (learning rate = 0.05) over 10 epochs with CrossEntropy loss.

Deep Neural Network

The deep neural network with configurable depth (num_layers = d) and width (hidden_dim = p):

The architecture starts with an input layer followed by a ReLU activation, then passes through $(d - 1)$ hidden layers each with ReLU activations. Finally, it outputs predictions through the final output layer.

The best configuration ($d = 3,\ p = 128$) achieved:

• Test loss: 1.385

• Accuracy: 47%

•AUC-ROC:0.82

Why Adding Depth Helped: Adding depth to the network significantly improved performance, with the best deep network ($d = 3,\ p = 128$, test loss: 1.385) outperforming the best shallow network ($n = 256$, test loss 1.5328). Even a single-layer deep network ($d = 1,\ p = 128$, test loss 1.5589) beat the shallow benchmark when compared to an earlier run (1.5774), but with the updated shallow result (1.5328), deeper configurations shine

5. Optimization Techniques

Gradient Descent and Stochastic Gradient Descent (SGD) are optimization algorithms used to train models, but they differ in how they update the model's parameters. Gradient descent uses the entire dataset to compute the gradient at each step, while SGD uses only a single training sample (or a small mini batch).

Gradient Descent (Full Batch)

SGD optimizer with batch size 56,000 and learning rate 0.05 is inefficient for large datasets, resulting in minimal training loss decrease and low test accuracy.

GD showed poor performance with slow training loss decrease, plateauing, and low test accuracy.

Stochastic Gradient Descent (SGD)

To evaluate how batch size impacts training dynamics, convergence, and generalization, we experimented with a set of progressively smaller batch sizes: [56000, 28000, 14000, 7000, 3500]

• Smaller batches reduced training time (592s → 441s) and improved test loss (1.9579 → 1.9437).

• Best accuracy: 20.31% at batch size 3,500.

SGD with Momentum

To evaluate how momentum influences convergence and generalization, we repeated the SGD batch size experiments using SGD with Momentum (momentum=0.9). The rest of the training configuration was kept the same for a fair comparison.

• Best performance (batch size = 3,500): accuracy 30.49%, loss 1.7198, time 439s.

Table 1: Comparison of Optimization Techniques

Method	Batch Size	Accuracy	Loss	Time (s)
Full-batch GD	56,000	11.75%	1.9470	624.04
SGD (no momentum)	3,500	20.31%	1.9437	441.00
SGD + Momentum	3,500	30.49%	1.7198	439.00

6. Advanced Techniques

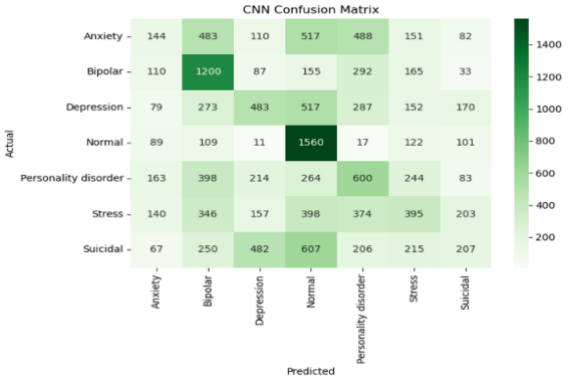
CNN Classifier

A CNN (Convolutional Neural Network) classifier is a type of deep learning model designed to recognize patterns in grid-like data, such as text and images. It uses convolutional layers to automatically extract spatial features, followed by pooling and fully connected layers to classify input into categories.

A custom CNN classifier trained for 10 epochs achieved 32.8% test accuracy and 28.9% weighted F1 score. The confusion matrix of the CNN classifier reveals poor performance across classes. For instance, only 144 out of 1,975 samples were correctly classified as Anxiety, while 517 Anxiety samples were misclassified as Normal.

Classification Report:

	precision	recall	f1-score	support
0	0.18	0.07	0.10	1975
1	0.39	0.59	0.47	2042
2	0.31	0.25	0.28	1961
3	0.39	0.78	0.52	2009
4	0.27	0.31	0.28	1966
5	0.27	0.20	0.23	2013
6	0.24	0.10	0.14	2034
accuracy			0.33	14000
macro avg	0.29	0.33	0.29	14000
weighted avg	0.29	0.33	0.29	14000



Takeaways:

The accuracy is low for CNN Classifier because it is **designed for spatial patterns** (like in images), and while they can process text (via embeddings), they don't capture long-range dependencies or context well.

BERT Model

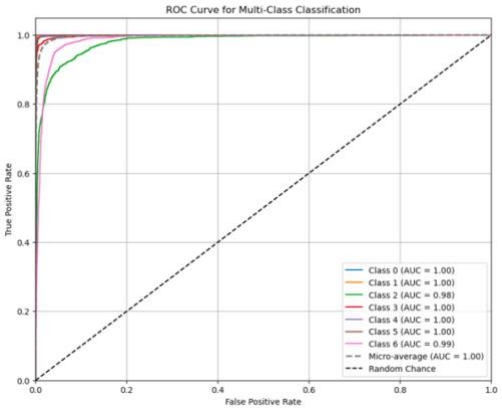
BERT Model reads text bidirectionally (both left-to-right and right-to-left) to grasp context more effectively than previous models. BERT is widely used for tasks like question answering, sentiment analysis, and text classification.

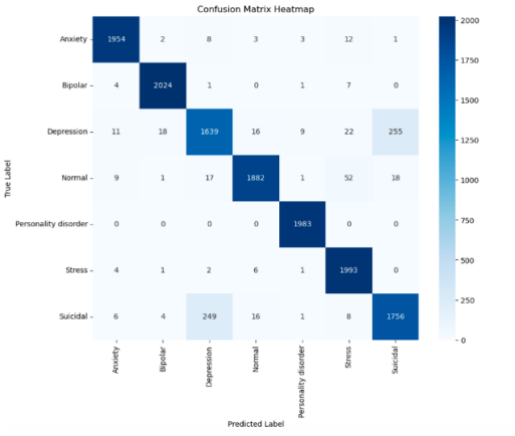
BERT (bert-base-uncased), fine-tuned for 4 epochs with batch size 16, achieved:

• Test accuracy: 94.5%.

• Weighted F1 score: 94.5%.

• Evaluation loss: 0.294.





Results and Discussion

BERT outperformed others with 94.5% accuracy and 94.5% weighted F1 score. Shallow and deep neural networks achieved 37% and 47% accuracy, respectively, while the CNN lagged at 32.8%. Confusion matrices (Figures 1–4) show BERT's superior class discrimination, with near-perfect Personality Disorder classification but some Depression-Suicidal overlap. Optimization experiments highlight mini-batch SGD with momentum as optimal.

Conclusion

BERT's 94.5% accuracy underscores its effectiveness for mental health classification. Dataset balancing ensured fairness, and SGD with momentum (batch size 3,500) optimized performance. The CNN's poor results suggest limitations in textual analysis compared to BERT's transformer architecture. Balancing the dataset improved model fairness across classes.

Future Scope

• Hybrid CNN-transformer models.

• CNN architecture fine-tuning.

• Incorporating metadata or contextual embeddings.

• Real-time mental health screening applications.

References

- Dataset link - Sentiment Analysis for Mental Health Dataset
- Google Colab Notebook with code - sp25_DsoDusKan_FPA05.ipynb
- About BERT - https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/
- Presentation video link - Call with 5305-Deeplearning_ANN-project group-20250504_215840-Meeting Recording.mp4