AMERICAN UNIVERSITY OF BEIRUT

FACULTY OF ARTS & SCIENCES

CMPS 261– Introduction to Machine Learning

Sentiment Analysis for Mental Health

Yara Al Tassi
Ali Ibrahim Khazaal Fawaz
Rozanne Wehbe

Instructor: Dr. Mohamed A. Kobeissi

April 9, 2025

Github Link: https://github.com/afawaz261/cmps261-project

Contents

0.1 Abstract

This project tackles the multi-class classification of mental health statuses from text data using various machine learning models. After extensive preprocessing and transformation using a TF-IDF vectorizer, the dataset was split into training, validation, and test sets. Due to significant class imbalance, we applied SMOTE (Synthetic Minority Oversampling Technique) to balance the training set. We evaluated five different models: Logistic Regression, Random Forest, Naive Bayes, Neural Network, and XGBoost. The performance of each model was assessed based on accuracy, precision, recall, F1-score, and confusion matrices. XGBoost outperformed all other models with a validation accuracy of 73.5% and a test accuracy of 71.9%, demonstrating its robustness in handling the textual nature and imbalance of the data. The final XGBoost model, along with the TF-IDF vectorizer, was saved for future inference. This report presents a comparative analysis of the models and discusses the challenges encountered throughout the pipeline, including imbalance handling and performance evaluation.

0.2 Introduction

Mental health is a growing concern in today's digital age, with social media and online platforms providing a space where individuals often express their thoughts, emotions, and psychological states. The ability to detect mental health conditions through textual data can play a significant role in early intervention and support. Recent advancements in Natural Language Processing (NLP) and Machine Learning (ML) have enabled the development of automated systems capable of identifying mental health indicators from text.

In this project, we aim to classify user-generated text into one of seven mental health categories: *Normal*, *Depression*, *Suicidal*, *Anxiety*, *Bipolar*, *Stress*, and *Personality disorder*. This task presents multiple challenges, including the high dimensionality and sparse

nature of textual data, overlapping features among classes, and a significant imbalance in class distribution.

To address these issues, we implemented a comprehensive pipeline that includes:

1. text preprocessing
2. feature extraction using TF-IDF
3. data splitting
4. balancing with SMOTE
5. training and evaluating multiple machine learning models:
    - Logistic Regression
    - Random Forest
    - Naive Bayes
    - Neural Network
    - XGBoost

To determine the most effective approach for this classification task. We compared the strengths and weaknesses of each model.

0.3 Project Structure

Originally, we had all of the code in one jupyter notebook; however, that made it hard to understand and run conveniently. We then decided to switch to a modular approach to ensure the maintainability and reproducibility of our code and to ensure that our main notebook is clean and easy to understand.

0.3.1 main.ipynb

This file imports all of the functions from the src package that we created, which contains the functions needed for:

- data cleaning
- text pre-processing
- feature extraction
- model training

0.3.2 src folder

This folder contains the functions used throughout the project.

The **data_cleaning** file contains functions to:

- load the dataset
- handle null values
- check class distribution
- check text length distribution

The **text_preprocessing** file contains functions that:

- tokenize (lemmatize) the text
- extract word frequencies
- visualize word clouds

The **feature_extraction** file contains functions to:

- extract features using Tdidf
- encode labels
- split data
- handle data imbalance

The **model_training** file contains functions that:

- train logistic regression, random forest, naive bayes, CNN, and xgb models
- evaluate the models

- save the models to prevent recomputation
- plot confusion matrices
- create a summary of all of the models performance metrics
- load saved models if they exist

0.3.3 models folder

This contains all of the saved models that we trained that can be loaded instead of recomputing them.

0.3.4 validation_evaluation_reports folder

This contains the evaluation/ classification reports for each model on the validation set.

0.3.5 confusion_matrices folder

This contains the saved confusion matrix of each model (tested on validation set)

0.4 Dataset and Preprocessing

The dataset used in this project consists of user-generated textual data labeled according to seven mental health categories: *Normal*, *Depression*, *Suicidal*, *Anxiety*, *Bipolar*, *Stress*, and *Personality disorder*. A preliminary analysis of the label distribution revealed significant class imbalance, with the majority of samples belonging to the *Normal*, *Depression*, and *Suicidal* categories, while other classes such as *Personality disorder* and *Stress* were severely underrepresented.

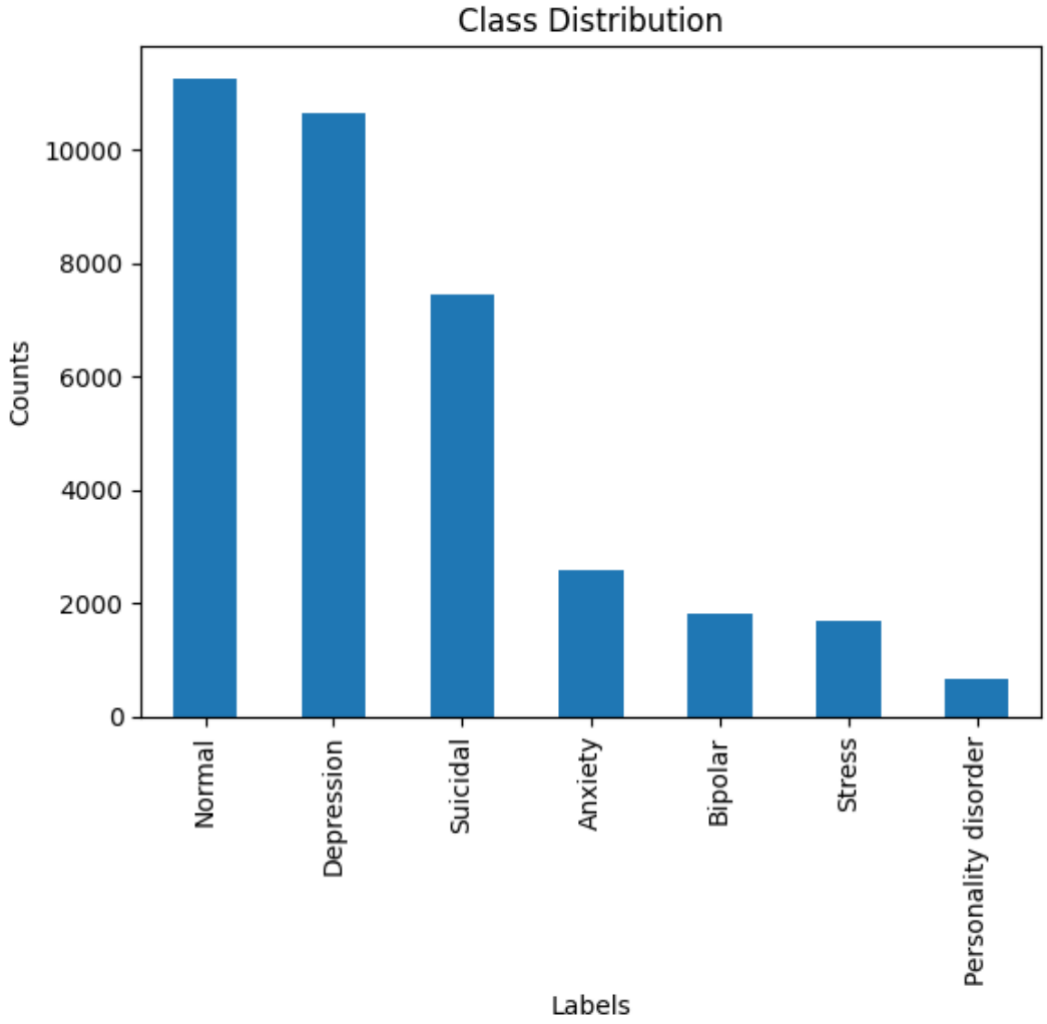

Figure 1: Class distribution of the original dataset, highlighting the imbalance across categories.

Before training any models, several preprocessing steps were applied to clean and standardize the text data. These included:

1. converting all text to lowercase
2. removing punctuation, stopwords, URLs, numbers, and extra whitespace
3. lemmatized each token to reduce it to its base form

These steps ensured that the textual input was normalized and that irrelevant noise was eliminated.

To convert the cleaned text into numerical features suitable for machine learning algorithms, we employed the Term Frequency-Inverse Document Frequency (TF-IDF)

vectorizer. This technique captures the importance of words in each document relative to the entire corpus, resulting in a high-dimensional but informative feature space.

The dataset was then split into training, validation, and test sets using a stratified 60/20/20 split to preserve label distribution across subsets. To address the class imbalance issue, we applied the Synthetic Minority Oversampling Technique (SMOTE) on the training data. This oversampling strategy synthetically generates new samples from the minority classes, leading to a more balanced training set that helps models generalize better across all classes.

Exploratory Visualization

To gain an initial understanding of the linguistic features of each class, we generated word clouds representing the most frequent terms in the tweets for each mental health label. The grid below illustrates common words used in posts labeled as Normal, Suicidal, Depression, Bipolar, Anxiety, Stress, and Personality Disorder.

Figure 2: Word cloud visualization per class label. Dominant terms such as *"feel"*, *"want"*, and *"life"* appear consistently across distress-related labels, while more neutral words are prevalent in the Normal class.

0.5 Methodology

Our approach involved training and evaluating five machine learning models for multi-class text classification: Logistic Regression, Random Forest, Multinomial Naive Bayes,

a Neural Network, and XGBoost. Each model was trained on the TF-IDF vectorized features, using the balanced training set obtained after applying SMOTE.

0.5.1 Model Training

Logistic Regression was trained with `max_iter=1000` to ensure convergence, and used the `random_state=42` for reproducibility.

Random Forest was trained using the default parameters with `random_state=42`. It aggregated multiple decision trees to reduce overfitting and improve generalization.

Multinomial Naive Bayes was chosen for its simplicity and efficiency on text data. It assumes feature independence and is often used as a baseline for text classification tasks.

Neural Network was implemented using TensorFlow's Keras API. The architecture consisted of two fully connected dense layers (with 512 and 256 units respectively), ReLU activation functions, dropout layers (rate 0.3) for regularization, and a final softmax output layer. The model was compiled using the Adam optimizer and trained for 10 epochs with a batch size of 64.

XGBoost, a gradient-boosted tree-based model, was trained with `eval_metric='mlogloss'` and `use_label_encoder=False`, which are standard for multi-class classification. It outperformed all other models on both validation and test accuracy.

0.5.2 Evaluation

All models were evaluated using accuracy, precision, recall, and F1-score on the validation set. Additionally, confusion matrices were generated to better understand misclassifications. The best performing model, XGBoost, was saved using `joblib` along with the fitted TF-IDF vectorizer for future inference.

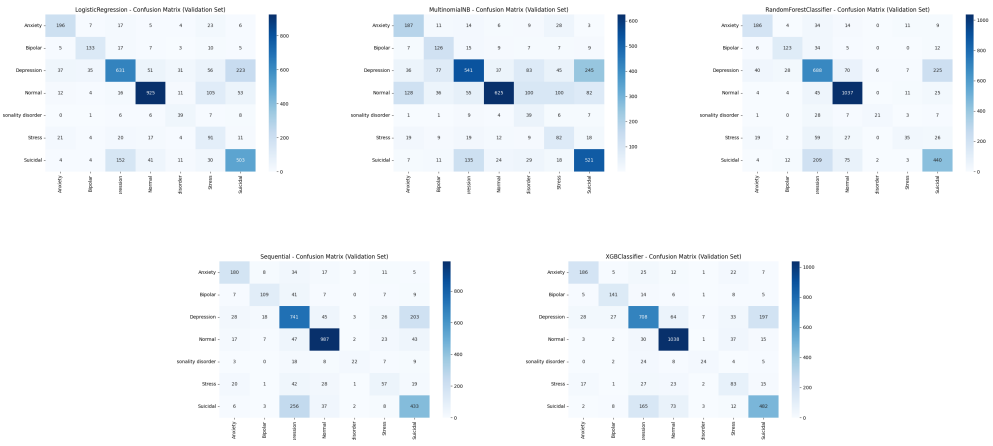

Figure 3: Confusion Matrices

	Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Weighted Precision	Weighted Recall	Weighted F1
4	XGBClassifier	0.7378048780487805	0.6838943984266664	0.6558229695230032	0.6640358498418099	0.7358652728021287	0.7378048780487805	0.735235681838724
0	LogisticRegression	0.6978935698447893	0.6167187801198747	0.6731485621995189	0.6345098386039796	0.7258684455124924	0.6978935698447893	0.7067566791818696
2	NeuralNetwork	0.7009423503325942	0.6594639449435629	0.5892949566554982	0.6143603013367877	0.7002180253163965	0.7009423503325942	0.6984412599728248
3	RandomForestClassifier	0.7012195121951219	0.6726777376633487	0.5834584929527479	0.607547496694739	0.6924705120301705	0.7012195121951219	0.6918956241010151
1	MultinomialNB	0.5878603104212861	0.5036589921701311	0.6082629520748679	0.5186835181657937	0.6699636917371069	0.5878603104212861	0.6064693469807549

Figure 4: Summary of Model Metrics

0.6 Results

As we can observe, although xgboost is the best performing model, it still struggles with classifying "personality disorder" and "stress".

0.6.1 Test Accuracy

Since we chose xgboost, we ran the final predictions on the test set.

As shown, the XGBoost model achieved the highest test accuracy of 0.718, outperforming the other models.

```
Predictions saved to xgb_test_predictions.csv
                     precision    recall  f1-score   support

            Anxiety       0.72      0.74      0.73       259
            Bipolar       0.79      0.69      0.74       180
         Depression       0.70      0.62      0.66      1063
             Normal       0.83      0.92      0.87      1127
Personality disorder       0.64      0.52      0.57        66
             Stress       0.42      0.45      0.43       168
           Suicidal       0.63      0.63      0.63       745

           accuracy                           0.72      3608
          macro avg       0.67      0.65      0.66      3608
       weighted avg       0.72      0.72      0.72      3608
```

Figure 5: xgboost test accuracy

To further assess the best-performing model, we examined its test classification report, including precision, recall, and F1-score across all classes.

The report highlights that while XGBoost performed exceptionally well on the *Normal* class (F1 = 0.88), it struggled with minority classes such as *Personality disorder* and *Stress*, which had lower F1-scores of 0.57 and 0.43 respectively. These results emphasize the challenge of handling imbalanced datasets, even with the application of SMOTE.

Overall, XGBoost demonstrated strong generalization and consistent accuracy across multiple mental health categories, making it the most robust model among those evaluated.

0.7 Discussion

The performance of the five models evaluated—Logistic Regression, Random Forest, Naive Bayes, Neural Network, and XGBoost—revealed important insights into the task of multi-class sentiment classification in the context of mental health.

Overall, **XGBoost** emerged as the best-performing model in terms of both validation and test accuracy, achieving an accuracy of approximately 0.72. This suggests that

XGBoost's ability to handle class imbalance and capture nonlinear relationships made it especially well-suited to the task. The Neural Network and Random Forest models also performed competitively, with accuracies close to 0.70, while Logistic Regression offered slightly lower but consistent results. On the other hand, Naive Bayes underperformed, likely due to its strong assumptions of feature independence, which are rarely satisfied in natural language data.

Despite the use of SMOTE for oversampling, class imbalance continued to affect the overall model performance, especially for underrepresented classes such as *Stress* and *Personality disorder*. These categories had notably lower F1-scores and recall values, indicating that the models struggled to correctly identify samples from these classes. This challenge was further compounded by the subtle linguistic differences between some of the classes, such as *Depression*, *Suicidal*, and *Anxiety*, which often share overlapping emotional and semantic features.

Several directions for improvement can be considered in future work. First, exploring more advanced text representation techniques such as word embeddings (e.g., Word2Vec or BERT) could capture richer semantic relationships between words. Additionally, further hyperparameter tuning, particularly for the XGBoost and Neural Network models, might yield performance gains. Lastly, incorporating more domain-specific linguistic features or external psychological lexicons could enhance the sensitivity of models to subtle variations in mental health-related language.

0.8 Conclusion

In this project, we addressed the challenge of multi-class sentiment classification of mental health-related text using a dataset labeled across seven emotional states. After applying comprehensive preprocessing steps, TF-IDF vectorization, and class balancing via SMOTE, we trained and evaluated five models: Logistic Regression, Random Forest, Naive Bayes, Neural Network, and XGBoost.

Among these, XGBoost consistently outperformed the others on both validation and test sets, demonstrating strong capability in handling high-dimensional textual data and imbalanced classes. While other models like Random Forest and Neural Network showed competitive results, Naive Bayes lagged due to its simplistic assumptions.

Our findings highlight the importance of both model choice and preprocessing strategy in mental health text classification. This study sets a strong baseline for future work that can leverage advanced embeddings and domain-specific features to improve performance, especially on underrepresented and semantically overlapping classes.