CMPS 261 – Machine Learning

Report: Sentiment Analysis for Mental Health Using Machine Learning

By Tala Diab and Ralph El Skaff

Notebook: https://github.com/tdyab/sentiment-analysis-261.git

Table of Content

I. Introduction & Objectives

An aspect of overall well-being that is often overlooked or swept under the rug is mental health. People often take serious words spoken by others as a joke when its regarding mental health, this is why being able to classify mental health conditions from text data can aid early intervention.

This report contains the development of a machine learning system made for classifying mental health status based on textual statements. The project uses natural language processing techniques along with machine learning algorithms to categorize statements into eight categories made of mental health conditions: anxiety, bipolar disorder, depression, normal, personality disorder, stress and suicidal tendencies.

The primary objectives of this project include:

- Preprocessing and exploring a dataset of mental health-related statements.
- Developing and evaluating machine learning models for classifying mental health conditions.
- Addressing class imbalance and optimizing model performance.
- Deploying the trained models for practical use.

II. Dataset Overview

The dataset used for the machine learning model consists of 37,130 samples, each containing a statement and the corresponding mental health status. However, the dataset exhibits class imbalance, since conditions such as “Normal” or “Depression” are heavily present compared to “Bipolar” or “Personality disorder.”

Dataset Structure

Column	Description	Data Type	Missing values
Statement	Text describing mental state	String	0
Status	Label (mental health category)	String	0

Class Distribution

Class	Count	Percentage
Normal	11,440	30.8%
Depression	10,785	29.0%
Suicidal	7,460	20.1%
Stress	1,810	4.9%
Anxiety	2,690	7.2%
Bipolar	1,940	5.2%
Personality disorder	755	2.0%

The dataset exhibits class imbalance, which is common in real-world mental health data. The class imbalance was addressed using Synthetic Minority Over-sampling Technique (SMOTE) to ensure fairness during model training. (more on SMOTE in I.ii.2)

III. Methodology

i. Data Preprocessing

Handling missing values: No missing values were found in the statement column. Had missing values existed, we would have dropped rows or replaced it with placeholder text.

Label encoding: Since machine learning models require numerical inputs, the target labels (mental health statuses) were encoded alphabetically into numerical values using LabelEncoder and sklearn.preprocessing. The mapping output was {'Anxiety': 0, 'Bipolar': 1, 'Depression': 2, 'Normal': 3, 'Personality disorder': 4, 'Stress': 5, 'Suicidal': 6}

Train-Test split: The dataset was split into 80% training and 20% testing sets, with stratification to ensure a fair and balanced evaluation and the preserving of class proportions. Resulting in 29,500 samples for training and 7,376 samples for testing.

ii. Feature Extraction

1. TF-IDF Vectorization

Text data was transformed into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF) to find correlations between certain words in statements and the relevant status. TF-IDF was used since it captures word importance relative to the entire collection, as well as penalizing common words (such as "i", "the", "and") while emphasizing meaningful terms.

Parameters used for TF-IDF:

max_features = 5000 (limits dimensionality to the top 5,000 words).

ngram_range = (1, 2) (considers single words and bigrams).

stop_words = 'english' (removes common stopwords).

2. Handling Class Imbalance (SMOTE)

Since the dataset was imbalanced, it would have led to a poor performance on minority classes such as 'Bipolar' and 'Personality disorder'. SMOTE generates synthetic samples for underrepresented classes.

Before: minority classes had very few samples, risk of model ignoring these classes.

After: balanced class distribution, improved model recall for rare conditions.

iii. Model Development

Two machine learning models were developed and evaluated:

1. Logistic Regression

Logistic regression is simple, interpretable and known to work well for text classification. It serves as a baseline for comparing more complex models.

Training configuration: max_iter = 1000 (ensures convergence), solver = 'lbfgs' (works well for multiclass classification).

Performance metrics (full classification report included in sentiment_analysis.ipynb):

Accuracy: 75.3%

Precision: 70% (macro avg)

Recall: 74% (macro avg)

F1-Score: 72% (macro avg)

Best performance was on 'Normal' with f1-score being 0.90, whereas the rarest class, 'Personality disorder' was a struggle resulting in f1-score = 0.63.

2. XGBoost

XGBoost handles non-liner relationships better than logistic regression and its robust against overfitting due to regularization along with supporting imbalanced data via class weighing.

Training configuration: n_estimators = 200 (number of boosting rounds), max_depth = 5 (controls tree complexity), eval_metric = 'mlogloss' (optimizes for multi-class log loss).

Performance metrics (full classification report included in sentiment_analysis.ipynb):

Accuracy: 77.8%

Precision: 79% (macro avg)

Recall: 71% (macro avg)

F1-Score: 74% (macro avg)

XGBoost proved to have better generalization across all classes as well as higher accuracy. However, it still struggles with rarer classes.

iv. Hyperparameter Tuning (GridSearchCV)

To optimize XGBoost, we performed grid search over

n_estimators = [150, 200, 250]

max_depth = [3, 5, 7]

Best parameters found: n_estimators = 200, max_depth = 5.

Impact of Tuning: F1-Score improved by 2% and reduced overfitting.

v. Model Deployment

The following components were serialized using joblib

- Trained models: logistic_regression_model.pkl, xgboost_model.pkl
- Preprocessing tools: tfidf_vectorizer.pkl (converts text to TF-IDF features), label_encoder.pkl (maps labels to numbers)

IV. Challenges & Solutions

Challenges faced included class imbalance which was overcome by using SMOTE to oversample minority classes, high dimensionality in text data which wasn't a problem due to limiting TF-IDF to 5,000 features. Another issue that arose was the model bias toward majority classes and there was an attempt to combat it using class-weighted loss in XGBoost. We also reduced overfitting by tuning max_depth and n_estimators.

Some improvements could be made through using deeper learning models that might be able to capture contextual meaning better than TF-IDF or using ensemble methods and combining logistic regression and XGBoost for better robustness.

V. Conclusion

The project successfully developed a machine learning pipeline for mental health text classification. Some of the key insights were:

- XGBoost outperformed Logistic Regression (77.8% vs 75.3% accuracy).
- SMOTE effectively handled class imbalance
- The system is deployable for real-world applications