Final Report

Jialu Li & Chengyi Liu

Introduction:

We are data scientists working with a mental health organization. The organization wants to see whether it can identify potential mental health issues from people in their text statements through sentiment analysis. We will work on the sentiment analysis task using natural language processing techniques with machine learning models.

Dataset and features:

Dataset: For this task, we use the mental health dataset available on Kaggle: https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health, which compiles 53,043 raw statements from diverse platforms, including social media posts, Reddit posts, Twitter posts, etc. After cleaning data by removing duplicates and NAs, 51,073 unique samples remain and are classified as shown in Figure 1.

Figure 1: Classification of statement sentiments

status	
Normal	16039
Depression	15087
Suicidal	10641
Anxiety	3618
Bipolar	2501
Stress	2293
Personality disorder	895

Features: The dataset includes a “statement” feature and a target variable, “status”, that consists of seven classes: Anxiety, Normal, Depression, Suicidal, Stress, Bipolar, and Personality disorder.

Model selection:

We select the following three traditional machine learning classification models for this multi-class classification problem. Overall, they are robust to high-dimensional data, like vectorized statements we used as a feature for sentiment analysis. Additionally:

- Random forest can handle non-linear relationships well.
- Logistic regression is more interpretable with assigned probabilities for each class.
- Support vector machine can maximize class separation, which works well for multi-class classification.

However, these models treat words independently, which may fail to capture nuanced sentiments in phrases. Therefore, we also select a Recurrent Neural Network (RNN), Long-Short Term Memory. The LSTM model can maintain memory of previous words and capture the order and context of words in a sentence, making it effective for understanding complex emotions and classifying multiple sentiments.

Feature engineering:

The only feature we used in this project was the “statement” feature, which contained different sentiment statements of people from various platforms. To fit features in models, we vectorize features for traditional machine learning models and tokenize features for the LSTM model before passing them to the LSTM embedding layer.

We also encode classes in the target variable for the LSTM model. Please refer to the following: Anxiety—0, Normal—1, Depression—2, Suicidal—3, Stress—4, Bipolar—5, and Personality disorder—6.

Model hyperparameter tuning:

We use a Randomized search method for hyperparameter tuning for the three traditional machine learning models.

- For Random Forest, we select three parameters:
  - n_estimators (the number of trees),
  - max_depth (the maximum depth of the tree),
  - and criterion (the function to measure the quality of a split).
- For Logistic Regression, we select three parameters:
  - C (inverse of regularization strength),
  - solver (algorithm to use in the optimization problem),
  - and max_iter (the maximum iterations taken for the solvers to converge).
- For SVM, we select three parameters:
  - C (regularization parameter),
  - kernel (kernel type used in the algorithm),
  - and gamma (kernel coefficient).

These parameters are selected based on their influential importance to the model performance, as referred to on the scikit learn algorithm websites, and previous model training experience in the midterm project. As a result, the best parameter sets for the three models are:

- Random Forest: 'n_estimators' = 200, 'max_depth' = None, and 'criterion' = 'gini'
- Logistic Regression: 'solver' = 'liblinear', 'max_iter' = 500, and 'C' = 10
- SVM: 'kernel' = 'linear', 'gamma' = 'scale', and 'C' = 1

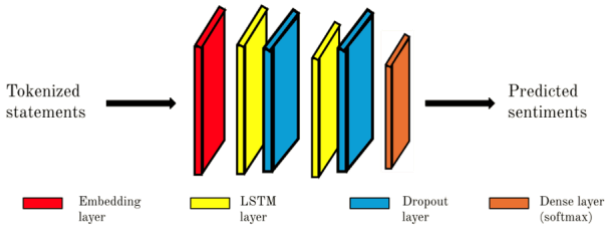
For the LSTM model, we build a simple stacked block LSTM model, containing four different layers: an embedding layer, two LSTM layers, two dropout layers, and a dense layer. The model structure is shown in Figure 2.

- The tokenized statements first pass through the embedding layer to convert each word index into a fixed-sized dense vector.

- Then, vectors go through the first LSTM block, consisting of a 64-unit LSTM layer and a 50% dropout layer, which returns the full sequences and randomly sets 50% of the inputs to 0.
- Then, processed vectors go through the second LSTM block, consisting of a 32-unit LSTM layer and a 50% dropout layer, which returns summarized vectors with 50% of the inputs set to 0.
- Finally, processed vectors pass through the dense layer for multi-class classification. We use the softmax activation function to convert outputs into probability distributions over classes.

We fix the model structure and twist the dropout percentage and training epochs. It turns out that the simple parameter twisting does not affect the network performance a lot. Therefore, we fix the dropout percentage to 50% and the training epochs to 10.

Figure 2: Workflow of the LSTM model



Model evaluation and performance:

We use the classification report metrics to evaluate models, including:

- precision: the proportion of correct positive predictions (true positives),
- recall: the proportion of actual positives that are correctly identified (TP/(TP+FN)),
- F1-score: balances precision and recall,
- support: the number of samples in a class,
- accuracy: the ratio of correct predictions,
- macro average: averaged metrics equally for all classes,
- and weighted average: averaged metrics with class weights.

These metrics are commonly used to evaluate classification models and report different aspects of the model. We chose the accuracy metric to show the overall correctness of the evaluated models, and we will also refer to the weighted average metrics if the accuracy is tied, because the number of samples in each class is imbalanced.

Comparing the accuracy among all models selected, as shown in the figures below, the Logistic Regression model achieves the highest accuracy of 76% and the weighted average score of precision, recall, and F1, 75%, 76%, and 75%.

Classification Report for Random Forest:

	precision	recall	f1-score	support
Anxiety	0.75	0.61	0.67	723
Bipolar	0.86	0.49	0.62	500
Depression	0.61	0.77	0.68	3018
Normal	0.83	0.96	0.89	3208
Personality disorder	1.00	0.18	0.30	179
Stress	0.70	0.18	0.28	459
Suicidal	0.70	0.55	0.62	2128
accuracy			0.72	10215
macro avg	0.78	0.53	0.58	10215
weighted avg	0.73	0.72	0.70	10215

Classification Report for SVM:

	precision	recall	f1-score	support
Anxiety	0.74	0.73	0.74	723
Bipolar	0.79	0.63	0.70	500
Depression	0.70	0.72	0.71	3018
Normal	0.86	0.95	0.91	3208
Personality disorder	0.71	0.37	0.49	179
Stress	0.61	0.44	0.51	459
Suicidal	0.68	0.64	0.66	2128
accuracy			0.75	10215
macro avg	0.73	0.64	0.67	10215
weighted avg	0.75	0.75	0.75	10215

Classification Report for Logistic Regression:

	precision	recall	f1-score	support
Anxiety	0.76	0.72	0.74	723
Bipolar	0.79	0.65	0.71	500
Depression	0.71	0.71	0.71	3018
Normal	0.85	0.96	0.90	3208
Personality disorder	0.67	0.41	0.51	179
Stress	0.60	0.43	0.51	459
Suicidal	0.67	0.65	0.66	2128
accuracy			0.76	10215
macro avg	0.72	0.65	0.68	10215
weighted avg	0.75	0.76	0.75	10215

Classification report for LSTM:

	precision	recall	f1-score	support
0	0.71	0.72	0.72	723
1	0.93	0.91	0.92	3208
2	0.70	0.68	0.69	3018
3	0.63	0.68	0.65	2128
4	0.47	0.30	0.37	459
5	0.58	0.68	0.63	500
6	0.25	0.28	0.26	179
accuracy			0.73	10215
macro avg	0.61	0.61	0.61	10215
weighted avg	0.73	0.73	0.73	10215

Model deployment:

We develop prediction pipelines for both single statement prediction and statement dataset prediction. Please refer to the .py files in the GitHub link. The organization can put raw statements directly in the pipeline, and the statements will be vectorized and predicted automatically. The single statement prediction pipeline will return both statements and predicted sentiments. A new data frame with predicted sentiments will be produced when using the statement dataset prediction pipeline. We use another Kaggle dataset: https://www.kaggle.com/datasets/mdismielhossenabir/sentiment-analysis, to test the statement dataset prediction pipeline.

Recommendations and future work:

The Logistic Regression model achieves an acceptable classification accuracy of 76%. Therefore, we recommend that the mental health organization employ this model for sentiment analysis.

With this trained model, the organization can predict a large number of statements and promptly refer to the predicted sentiment classes. However, we suggest that the organization only use the Logistic Regression model's results for reference, as the accuracy is not high enough to be fully trusted. More detailed analysis and face-to-face diagnosis are needed for those who are identified with mental health issues, and sometimes, those who are incorrectly identified as normal.

In this project, we also tried to implement a transformer-based model, DistilBERT. However, we can only train the model for one epoch due to limited memory, and the performance is not as expected. In the future, if the computational memory permits, we will train the transformer model for more epochs to see if the model can better predict sentiments from people's daily statements for early mental health issue detection.